

41132
38



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES

CAMPUS ARAGON

MIGRACION DE ORACLE 7.3.4 A ORACLE 8.1.6
DE LA BASE DE DATOS TIPO RELACIONAL EN UN
SISTEMA DE PREPAGO DE UNA COMPAÑIA DE
RED CELULAR"

T E S I S
QUE PARA OBTENER EL TITULO DE
INGENIERO EN COMPUTACION
P R E S E N T A N :
RAUL GILDARDO ROSAS LOPEZ
MARIO DE JESUS MARTINEZ CUELLAR

ASESOR:
ING. ROBERTO BLANCO BAUTISTA

MEXICO

NOVIEMBRE 2003

TESIS CON
FALLA DE ORIGEN



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

**TESIS
CON
FALLA DE
ORIGEN**

PAGINACION

DISCONTINUA

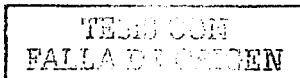
... ..

Dedico a:

Muchas cosas de la vida son gratis y otras se logran con esfuerzo y continua dedicación, esta es una de ellas la cual va acompañada del apoyo incondicional de mis padres, hermanos y el inmenso amor que recibo día a día de mi esposa, la cual me ha brindado la alegría más grande de mi vida, nuestro hijo Sebastián, motivos por los cuales agradezco a Dios de tenerlos conmigo para continuar disfrutando a lado de ellos cada momento en busca de una mejor calidad como ser humano, persona y profesionista que ahora soy.

A nuestra máxima casa de estudios la UNAM que me recibió como un hijo enseñándome el camino de la educación y sabiduría en sus aulas acompañadas de mis profesores y amigos que han sido parte de mi vida y de mi preparación.

Raúl



✓

Dedico a:

A dios, por brindarme la fuerza y la salud para hacer de este momento uno de los más importantes de mi vida.

A mis padres, Mario y Leonor aquellos seres inalcanzables que me dieron la oportunidad de vivir y de quienes estoy muy orgulloso de tenerlos conmigo, aquellas personas que día con día me brindan su apoyo y me guían por el camino de la humildad.

Mil gracias papa y mama, por todo su amor y cariño que he recibido de ustedes durante todos estos años de mi vida.

A mis hermanos, Dany, Pedro, Eder y Gus gracias por su apoyo, cariño y comprensión.

A ti Ofelia, por formar parte de mi vida, por tu amor y por darme la oportunidad de compartir contigo este logro.

A Raúl, mi amigo, por su apoyo.

A la Universidad Nacional Autónoma de México, por brindarme ese tesoro tan preciado:
La Educación.

A todas y cada una las personas que han estado a mi lado, familiares, amigos y maestros gracias por su motivación.

Mil gracias a todos.

Con mucho amor,
Mario de Jesús Martínez Cuellar.

Ji

INDICE

| | |
|--------------|------|
| OBJETIVOS | xiii |
| INTRODUCCION | xv |

CAPITULO I

I. FUNDAMENTOS DE BASES DE DATOS

| | |
|-----------------------------------------------------|----|
| 1.1 Antecedentes Históricos | 19 |
| 1.2 Ventajas e Inconvenientes de las Bases de Datos | 20 |
| 1.3 Modelos de Bases de Datos | 24 |
| 1.3.1 Modelo Relacional | 24 |
| 1.3.2 Modelo Jerárquico | 28 |
| 1.3.3 Modelo de Red | 28 |
| 1.4 Arquitecturas de Bases de Datos | 29 |
| 1.4.1 Arquitectura Cliente-Servidor | 29 |
| 1.4.2 Arquitectura Distribuida | 30 |
| 1.5 Etapas de desarrollo de una Base de Datos | 31 |
| 1.5.1 Modelo Conceptual de Datos | 31 |
| 1.5.1.1 Modelo Entidad - Relación | 33 |
| 1.5.1.1.1 Entidades | 33 |
| 1.5.1.1.2 Atributos | 34 |
| 1.5.1.1.3 Relaciones | 34 |
| 1.5.1.1.3.1 Uno a Uno | 36 |
| 1.5.1.1.3.2 Uno a Muchos | 36 |
| 1.5.1.1.3.3 Muchos a Muchos | 37 |
| 1.5.1.1.4 Identificadores Unicos | 37 |
| 1.5.1.1.5 Normalización | 38 |
| 1.5.1.1.5.1 Primera Forma Normal(PFN) | 39 |
| 1.5.1.1.5.2 Segunda Forma Normal(SFN) | 39 |
| 1.5.1.1.5.3 Tercera Forma Normal(TFN) | 39 |



| | | |
|---------|-----------------------------------------|----|
| I.5.2 | Modelo Lógico de Base de Datos | 40 |
| I.5.2.1 | Mapeo de Entidades-Tablas | 41 |
| I.5.2.2 | Mapeo de Atributos-Columnas | 41 |
| I.5.2.3 | Mapeo de Id Unicos – Llaves Primarias | 42 |
| I.5.2.4 | Mapeo de Relaciones-Llaves Foráneas | 43 |
| I.5.3 | Construcción Física de la Base de Datos | 44 |
| I.6 | Tendencias de las Bases de Datos. | 44 |

CAPITULO II

| | | |
|----------|------------------------------------------------|----|
| I.- | ELEMENTOS DE ORACLE 8 | |
| II.1 | Que es Oracle ? | 49 |
| II.2 | Arquitectura de Oracle | 51 |
| II.2.1 | Instancia | 51 |
| II.2.1.1 | Memoria | 51 |
| II.2.1.2 | Procesos | 53 |
| II.2.2 | Archivos | 57 |
| II.3 | Objetos de Oracle | 60 |
| II.3.1 | Tablas | 60 |
| II.3.2 | Vistas | 61 |
| II.3.3 | Sinónimos | 62 |
| II.3.4 | Indices | 62 |
| II.3.5 | Secuencias | 63 |
| II.3.6 | Roles | 63 |
| II.4 | SQL y SQL Plus | 63 |
| II.4.1 | Que es SQL y SQL Plus ? | 63 |
| II.4.2 | Sentencias SQL | 64 |
| II.4.2.1 | Lenguaje de Definición de Datos (DDL) | 65 |
| II.4.2.2 | Lenguaje de Manipulación de Datos (DML) | 66 |
| II.4.2.3 | Lenguaje de Control de Datos (DCL) | 66 |
| II.4.3 | Selección, Restricción y Ordenamiento de Datos | 67 |
| II.4.3.1 | Cláusula From | 68 |
| II.4.3.2 | Cláusula Where | 68 |



| | | |
|----------|-------------------------------|----|
| II.4.3.3 | Cláusula Order By | 69 |
| II.4.3.4 | Cláusula Distinct | 70 |
| II.4.4 | Funciones | 70 |
| II.4.5 | Operaciones de Conjuntos | 75 |
| II.4.6 | Joins | 76 |
| II.4.6.1 | Que es un Join? | 76 |
| II.4.6.2 | Tipos de Joins | 76 |
| II.4.7 | Subqueries | 79 |
| II.4.8 | Restricciones (Constraints) | 80 |
| II.4.8.1 | Tipos de Restricciones | 80 |
| II.4.9 | Control de Acceso a Usuarios | 81 |
| II.4.9.1 | Cuentas de Usuario | 81 |
| II.4.9.2 | Privilegios del Sistema | 82 |
| II.4.9.3 | Privilegios de Objetos | 83 |
| II.4.9.4 | Auditoria | 84 |

CAPITULO III

| | | |
|-----------|---------------------------------------------------------------------------|-----|
| III.- | MIGRANDO BASES DE DATOS ORACLE | |
| III.1 | Concepto de Migración | 89 |
| III.1.1 | Pasos del Proceso de Migración | 89 |
| III.1.2 | Responsabilidades del Administrador de Base de Datos durante la Migración | 91 |
| III.2 | Preparando la Migración | 92 |
| III.2.1 | Elegir Método de Migración | 92 |
| III.2.1.1 | Utileria de Migración de Oracle Migration Utility | 92 |
| III.2.1.2 | Asistente de Migración de Datos Data Migration Assistant | 95 |
| III.2.1.3 | Export / Import | 99 |
| III.2.1.4 | Copia de Datos | 102 |
| III.2.2 | Requerimientos del Sistema y Recursos Disponibles | 104 |
| III.2.3 | Problemas Comunes durante la Migración | 106 |
| III.3 | Estrategias de Respaldo | 112 |
| III.3.1 | Respaldo en Caliente (Hot Backup) | 112 |

| | |
|------------------------------------------|-----|
| III.3.2 Respaldo en Frío (Cold Backup) | 113 |
| III.3.3 Descarga y Carga de Datos | 113 |
| III.4 Plan de Pruebas | 115 |
| III.4.1 Proceso de Migración | 116 |
| III.4.2 Base de Datos Alterna | 116 |
| III.5 Tareas Posteriores a la Migración | 117 |

CAPITULO IV

IV.- MIGRACION DE LA BASE DE DATOS DEL SISTEMA DE PREPAGO

| | |
|----------------------------------------------------------------------------------------------|-----|
| IV.1 Que es el Sistema de Prepago ? | 123 |
| IV.2 Configuración Actual | 125 |
| IV.2.1 Sistema Operativo | 125 |
| IV.2.2 Base de Datos | 127 |
| IV.3 Análisis de Migración | 134 |
| IV.4 Tareas Previas | 140 |
| IV.4.1 Respaldo de la Base de Datos en Dominio1 vía Hotbackup | 140 |
| IV.4.2 Restauración de Base de Datos en Dominio2 | 143 |
| IV.4.3 Restauración y Sincronización de Base de Datos en Dominio2 | 144 |
| IV.4.4 Paro de Ivr's y Sistema de Bancos en Dominio1 | 145 |
| IV.4.5 Abrir Base de Datos de Dominio2 | 146 |
| IV.4.6 Adición de Espacios de Base de Datos de Dominio2 | 147 |
| IV.5 Proceso de Migración de la Base de Datos en el Dominio2. | 149 |
| IV.5.1 Migración y Actualización de la Base de Datos Oracle 7.3.4 a Oracle 8.1.6 en Dominio2 | 149 |
| IV.5.1.1 Preparación de la Base de Datos de Producción para Migración | 149 |
| IV.5.1.2 Instalación del Software de Oracle8 | 154 |
| IV.5.1.3 Migrando la Base de Datos Oracle7 | 155 |
| IV.5.1.4 Conversión de la Base De Datos | 160 |
| IV.5.2 Actualización de Aplicación | 162 |

| | |
|---------------------------------------------------------------------------|-----|
| IV.5.3 Actualización de la Base de Datos para Versión 7 de Aplicación. | 163 |
| IV.5.4 Reenrutamiento de tráfico hacia Dominio2. | 167 |
| IV.5.5 Pruebas de tráfico. | 168 |
| IV.5.6 Apertura de IVR's, Carga de Fichas y Sistema de Bancos | 169 |
| IV.6 Tareas Posteriores al Proceso de Migración | 170 |
| IV.6.1 Respaldo Total de Base de Datos vía Hotbackup | 170 |
| IV.6.2 Respaldo Total de la Base de Datos Vía Export | 170 |
| IV.6.3 Cambio de Password Usuario OUTLN | 171 |
| IV.6.4 Consistencia de los constraints de fecha de Oracle | 171 |

CAPITULO V

V.- MANTENIMIENTO A LA BASE DE DATOS.

| | |
|---------------------------------------------------------------------------------------------------|-----|
| V.1 Introducción al Mantenimiento y Afinación de Bases de Datos Oracle. | 175 |
| V.2 Herramientas de Monitoreo de Base De Datos | 176 |
| V.3 Manejo de Mensajes del Manejador | 177 |
| V.3.1 Oracle Alert.log | 177 |
| V.3.2 Archivos de Trace | 178 |
| V.3.3 Manejo de Eventos | 178 |
| V.4 Manejo de Utilerías, Vistas dinámicas y de Diccionario para el Monitoreo de la Base de Datos. | 179 |
| V.4.1 Vistas Dinámicas y De Diccionario de Datos | 179 |
| V.4.2 Manejo de Utilerías de Oracle | 180 |
| V.5 Monitoreo y Afinación del Sistema Operativo. | 182 |
| V.6 Monitoreo y Afinación de los Segmentos de Memoria de Oracle. | 183 |
| V.6.1 Area Global del Sistema (SGA) | 184 |
| V.6.1.1 Shared Pool | 184 |
| V.6.1.2 Buffer Cache | 189 |
| V.6.1.3 Redo Log Buffer | 191 |
| V.6.2 Area Global de Programa (PGA) | 193 |
| V.6.3 Areas de Código de Software | 194 |

| | |
|---------------------------------------------------------|-----|
| V.7 Configuración de la Base de datos y Uso de Hardware | 194 |
| V.8 Recomendaciones sobre los Segmentos de Rollback. | 196 |
| V.9 Uso de Particionamiento de Objetos de Oracle. | 198 |
| | |
| CONCLUSIONES | 205 |
| APENDICE I | 209 |
| APENDICE II | 241 |
| GLOSARIO | 259 |
| BIBLIOGRAFIA | 261 |

OBJETIVOS

- Disponibilidad Total del Sistema de Prepago durante el periodo de Migración.
- Asegurar la Integridad de los Datos.
- Disminuir Costos y Tiempos de Espera.
- Lograr una Migración Transparente para el usuario.

TECNOLOGIA
FALLA DE ORIGEN

xiv

TESIS CON
FALLA DE ORIGEN

INTRODUCCION

Hoy en día el uso de los manejadores de Bases de Datos ha incrementado su auge en la gran mayoría de los campos, ya no es un término que resulte desconocido para el ser humano, por el contrario es un concepto que día con día cobra mayor importancia y que se convierte en parte de él.

Los pasos agigantados hacia la vanguardia que dan este tipo de herramientas obliga a que los administradores de Bases de Datos se encuentren actualizados en sus sistemas para lograr con ello el gran desempeño de estos, en base a las nuevas características que brindan estas novedosas versiones.

En el presente trabajo se encuentra la información necesaria requerida para lograr una migración de una versión de estos manejadores, se muestra de manera detallada todos los pasos requeridos para llevar a cabo esta tarea, así como los posibles problemas que se pudieran presentar durante el desarrollo.

Primeramente se muestran algunos conceptos básicos sobre Base de Datos, la manera en que éstas son analizadas y diseñadas; es decir, desde plasmar cada una de las entidades para posteriormente poder transformarlas en tablas que formarán parte de nuestra Base de Datos y a su vez, de nuestro sistema.

Nos brinda conceptos de lo que es nuestro manejador de Base de Datos (Oracle), el lenguaje para poder manipularlo y acceder a los datos; así como operaciones lógicas que se pueden realizar con ellos; además de protección; esto último, con el fin de mantener la integridad y de permitir únicamente el acceso a aquellas personas o aplicaciones que realmente lo requieran.

Entre otras cosas nos presenta también diferentes técnicas para poder realizar la tarea de migración, desde los métodos gráficos como lo es el Asistente de Migración de Datos de Oracle hasta los métodos más complejos como lo es la Utilería de Migración de Oracle, pasando por el Export/Import y la Copia de Datos.

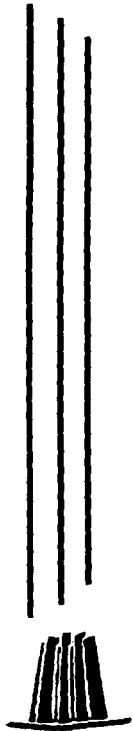
Una de las partes más importantes y sobresalientes de este trabajo es la aplicación de todos los conceptos anteriores. Existe un apartado en el cuál se aplica todo lo anterior a un caso práctico y real; es decir, se realiza paso a paso la migración de la enorme Base de Datos del Sistema de Prepagó.

Por último, se dan a conocer algunas técnicas para el monitoreo y mantenimiento de esta Base de Datos a nivel lógico y físico, es decir a nivel Sistema Operativo incluyendo todos sus componentes tales como discos, memoria, etc. como a nivel de procesos propios de Oracle.

Si alguna de las técnicas presentadas en este trabajo son aplicadas, se podrá obtener gran provecho al tener que realizar alguna tarea de este tipo, sobretodo por las grandes cantidades de información que se manipulan en este tipo de Base de Datos.

TESIS CON

FALTA DE



CAPITULO I

FUNDAMENTOS DE BASE DE DATOS

TESIS CON
FALSA DE ORIGEN

xvii

xviii

TERES CON
FALLA DE ORIGEN



FUNDAMENTOS DE BASES DE DATOS

1.1 ANTECEDENTES HISTORICOS

Antes de la introducción de los sistemas de gestión de Bases de Datos, todos los datos que eran permanentemente almacenados en un sistema informático, tales como las nóminas y los registros de contabilidad, se almacenaban en archivos individuales. Un sistema de gestión de archivos, generalmente proporcionado por el fabricante de la computadora como parte del sistema operativo, era el encargado de llevar la cuenta de los nombres y ubicaciones de los archivos. El sistema de gestión de archivos, básicamente no tenía un modelo de datos; no sabía nada acerca del contenido de los archivos; para el sistema, era lo mismo un archivo que contuviera un documento de texto que un archivo que contuviera datos de nóminas.

El hecho de asociar una estructura de datos con una relación, fué identificado por investigadores y documentado en el año de 1967, por R.E. Levenin y M.E. Maron. Posteriormente, D. L. Childs, en 1968, efectuó una serie de publicaciones al respecto. En Junio de 1970, es publicado en Communications of the ACM un artículo denominado "A Relational Model of Data for Large Shared Data Banks" realizado por Eddgar F. Cood; el cual, dá lugar al inicio del modelo de datos relacional.

Este artículo, es el resultado de un trabajo con un modelo abstracto de la información que se inició en 1968 en el IBM Research Laboratory en San José, California. El objetivo: encontrar un fundamento teórico de los diferentes aspectos de un Sistema Manejador de Bases de Datos (Data Base Management System) completamente ajeno de los aspectos de un proceso físico dentro de una máquina en particular. Este modelo, es el que actualmente se conoce como modelo relacional ó estructuras de datos relacionales.

Una de las partes más importantes de este modelo es la introducción del concepto de tener en la estructura misma operadores relacionales (Selección, Proyección, Unión). Esta fué la álgebra que Cood definió para operar estas estructuras. Estos operadores son los que se utilizan para toda la manipulación de datos en una estructura relacional.

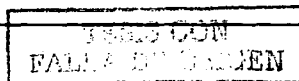
El primer nombre que recibió el lenguaje manejador de datos fué ALPHA y posteriormente cambio a SQL (Structured Query Language). El artículo en referencia originó que se empezaran a diseñar y desarrollar sistemas basados en la teoría del modelo relacional. El primero de los prototipos, fué llamado PRTV (Peterlee Relational Test Vehicle) desarrollado en Inglaterra en el Centro Científico de Peterlee. Estaba muy enfocado a problemas urbanos y de estadísticas de salud. De este prototipo, surgió otro sistema que se generó en Europa, se le conoce como Interactive Management System. Un segundo prototipo, desarrollado por IBM, se llamó XRM (Extended Relational Memory) y fué desarrollado en el Centro Científico de Cambridge, Mass. Posteriormente, en 1978, se tuvo disponible en Yorktown Heights, N.Y., el llamado Q.B.E., (Query By Example).

Este sistema se basa en una idea muy original para su época, la de presentar al usuario un dibujo o un esquema de sus datos en forma de tabla (Columnas y Renglones), en la que esquemáticamente se pueden hacer consultas, actualizaciones o modificaciones a través de lo que podríamos llamar "un ejemplo ilustrado".

Definitivamente el prototipo base y más completo desarrollado por IBM es el famoso Sistema R que se desarrolló en San Jose Research Laboratory. Empezó a ser operativo en el año de 1977. Las versiones de este sistema para la venta como programas-producto, comenzaron a salir por parte de los fabricantes de Software a principios de la década de los 80's. En forma paralela, en la Universidad de Berkeley, se llevó a cabo el desarrollo de otro importante prototipo diseñado por Michael Stonbraker. A este tipo prototipo se le denominó INGRES y su desarrollo concluyó en el año de 1976. Al igual que el Sistema R de IBM, INGRES fué convertido en programa producto y se empezó a comercializar a principios de los 80's.

1.2 VENTAJAS E INCONVENIENTES DE LAS BASES DE DATOS

Las Bases de Datos surgen como respuesta al nuevo planteamiento de los sistemas orientados hacia los datos y tienen el proposito de mejorar la calidad de los sistemas informáticos y aumentar su rendimiento.



A continuación, se muestran las ventajas de los sistemas de bases de datos:

- Independencia de los datos respecto a los procesos.-

La mutua independencia de datos y procesos lleva a que un cambio a estos últimos no imponga un nuevo diseño de la base de datos. Por otra parte, la nueva inclusión de recientes informaciones, desaparición de otras, etc., no deben obligar a alterar los programas. Esta independencia de los tratamientos frente a la estructura de la base de datos, supone una considerable ventaja, al evitar el importante esfuerzo que origina la reprogramación de las aplicaciones cuando se producen cambios en los datos.

La flexibilidad que proporcionan los datos y programas es muy importante para conseguir sin excesivos costos la continua adaptación del sistema de información a la evolución de las organizaciones, aunque en la práctica dicha dependencia no es absoluta, los sistemas de gestión de bases de datos actuales van respondiendo cada vez mejor a las exigencias de los usuarios relativas a la independencia y flexibilidad de los datos.

- Coherencia de los datos.-

Debido a que la información de la base de datos se recoge y almacena una sola vez, en los tratamientos se utilizan los mismos datos, por lo que los resultados de todos ellos son coherentes y perfectamente comparables. Además al no existir (o disminuir en gran medida) la redundancia en los datos, desaparece el problema que se presentaba en el enfoque clásico de que el cambio en un dato obligaba a actualizar una serie de archivos. De esta forma se elimina también el inconveniente de las divergencias en los resultados debidas a actualizaciones no simultáneas en todos los archivos.

- Mejor disponibilidad de los datos para el conjunto de los usuarios.-

Cuando se aplica la metodología de bases de datos, cada usuario ya no es propietario de los datos, puesto que éstos se comparten entre el conjunto de aplicaciones, existiendo una mejor disponibilidad de los datos para todos los que tienen necesidad de ellos, siempre que estén autorizados para su acceso.

Hay también una mejor transparencia respecto a la información existente, ya que todos los datos que se encuentran en la base se deben relacionar en un catálogo o diccionario de puede ser ampliamente difundido y accedido por medios informáticos.

- Mayor valor informativo.-

Puesto que la base de los datos ha de ser reflejo del mundo real, en ella se recogen las interrelaciones entre los datos, por lo que el valor informativo del conjunto es superior a la suma del valor informativo de los elementos individuales que los constituyen; es decir, actúa el efecto de sinergia.

- Mejor y más normalizada documentación de la información, la cual está integrada con los datos.-

En el enfoque clásico, los datos se encuentran separados de su contenido semántico; los primeros se almacenan en archivos y su descripción se hace mediante un lenguaje de programación que se encuentra en los programas. La documentación de los datos realizada por el analista o programador es en general insuficiente. Además, por lo común la estandarización brilla por su ausencia. Este problema se atenúa en gran medida en las bases de datos, ya que en la misma base se incluyen, no sólo los datos, sino también la semántica de los mismos.

- Mayor eficiencia en la recopilación, validación e introducción de los datos en el sistema.-

Al no existir redundancias, los datos se recopilan y validan una sola vez, aumentando así el rendimiento de todo el proceso previo al almacenamiento.

- Reducción del espacio de almacenamiento.-

La desaparición (o disminución) de las redundancias, así como la aplicación de técnicas de compactación, lleva en los sistemas de bases de datos a una menor ocupación de almacenamiento secundario – disco magnético.

Como hemos señalado anteriormente, las bases de datos no sólo presentan ventajas, sino también tienen posibles inconvenientes que es necesario valorar antes de

tomar una decisión relativa a un cambio en la orientación del sistema informático. Entre estos inconvenientes es preciso destacar:

- **Instalación costosa.-**

La implantación de un sistema de base de datos puede llevar consigo un costo elevado, tanto en equipo físico (nuevas instalaciones o ampliaciones), como en el lógico (sistemas operativos, programas, compiladores , etc. necesarios para su uso), además del mismo coste de adquisición y mantenimiento del Sistema de gestión de bases de datos.

- **Personal especializado.-**

Los conocimientos, que resultan imprescindibles para una utilización correcta y eficaz y sobre todo para el diseño y administración de las bases de datos, implican una necesidad de personal especializado.

- **Implementación larga y difícil.-**

Debido a las causas apuntadas anteriormente, la implantación de una base de datos puede convertirse en una tarea larga y laboriosa. Las dificultades que van apareciendo a lo largo de su desarrollo llevan en general a que se superen ampliamente los plazos inicialmente previstos.

- **Falta de rentabilidad a corto plazo.-**

La implantación de un sistema de base de datos, tanto por su costo en personal y en equipos como por el tiempo que tarda en estar operativo, no resulta rentable a corto plazo, sino a medio o, incluso, a largo plazo.

- **Escasa estandarización.-**

Un problema muy importante que se pone de manifiesto en el momento de la creación de una base de datos, es la falta de estandarización que facilite a los usuarios el manejo de los sistemas de bases de datos.



- Defasamiento entre teoría y práctica.-

Al existir un considerable avance de la teoría en relación con la práctica, en muchas ocasiones los usuarios, especialmente los directivos se engañan respecto a las presentaciones reales que pueden proporcionarles los Sistemas manejadores de bases de datos actuales, creyendo que constituyen ya una realidad ciertos aspectos que todavía son sólo teóricos, aunque de aplicación inmediata operativa.

Es evidente que las bases de datos constituyen muchas ventajas siempre y cuando se logre contrarrestar y doblegar a todos sus inconvenientes, ya que de lo contrario sería una gran frustración tanto para directivos como para usuarios de sistemas de bases de datos.

I.3 MODELOS DE BASES DE DATOS

Para establecer la diferencia entre Bases de Datos, no únicamente se puede realizar por sus funciones específicas, sino además por la forma en que modelan los datos.

Existen básicamente 3 modelos de Bases de Datos: El modelo relacional, jerárquico y el modelo de red.

I.3.1 MODELO RELACIONAL

El nacimiento de este modelo como ya se mencionó fué a raíz de la investigación que realizó E.J. Codd, en la cual propone los siguientes objetivos, para este modelo:

- Objetivo de independencia de datos. Proporciona una frontera clara y tangible entre los aspectos lógicos y físicos de un manejador de base de datos.
- Objetivo de Comunicación. Crear un modelo empleando estructuras simples, de tal manera que todo tipo de usuarios (programadores y usuarios finales) tengan una comprensión común de los datos y por lo tanto, puedan comunicarse entre sí sobre los aspectos al menos básicos de la base de datos.
- Objetivo de procesamiento de conjuntos. Introducir conceptos para un lenguaje de alto nivel que permita a los usuarios expresar operaciones sobre grandes volúmenes de datos, en lugar de tener que considerar un registro a la vez.

Además de estos objetivos, hubo otros como el de contar con una sólida fundamentación teórica para la organización y manejo de los datos y eliminar el empleo de conceptos de posicionamiento para direccionar los datos, de modo que se operen convenientemente.

El modelo relacional, es un modelo muy elegante y claro. Ha constituido los cimientos de la industria de las bases de datos durante casi 20 años. Una base de datos de tipo relacional reduce los datos al más simple nivel, tablas de dos dimensiones que contienen columnas y filas de valores de datos. Las tablas son fáciles de visualizar, y los usuarios pueden explicar sus necesidades en términos fáciles de entender. Además de que las bases de datos relacionales cuentan con un vocabulario relativamente limitado.

Comúnmente, el término Base de Datos Relacional se entiende como una colección de tablas bidimensionales ligadas entre sí. Las bases de datos relacionales son un modelo particular de bases de datos que se basan principalmente en lo siguiente:

- Estructuras
- Grupo de operaciones matemáticas basadas en el álgebra relacional
- Integridad de los datos para asegurar la consistencia.

ESTRUCTURAS

- Entidades. Objeto de interés sobre el cual el negocio necesita obtener información.
- Atributos. Información usada para describir a las entidades.
- Relaciones. Asociación que existe entre dos o más entidades. La forma en que una entidad se relaciona con otra. La relación entre dos entidades es definida por un atributo que ambas entidades contengan.

En un sistema relacional, el significado de una columna o fila, no depende de su orden relativo de otra fila o columna. Esto significa, que la recuperación de los datos puede realizarse desde un selecto número de columnas sin tener que recuperar todas las mismas. De igual forma para las filas, se pueden recuperar datos desde un subconjunto de filas sin tener que recuperar todas las filas.



Las bases de datos relacionales reducen el almacenamiento de los datos redundantes. Esto tiene tres beneficios:

- Permite recolocar datos y combinarlos para facilitar una nueva relación.
- La actualización de los datos es sencilla, ya que son pocas las ocurrencias del dato. Esto reduce la aparición de errores originados de la actualización de todas las ocurrencias.
- Se reduce los requerimientos de espacio en disco.

OPERACIONES MATEMATICAS

El Algebra Relacional de E. F. Codd, define tres tipos de operaciones para la recuperación de datos.

- Selección.

Selección, es el proceso de recuperar todas las filas en la cual se especifica algún criterio. Esta es una estrecha definición de selección, y no debe ser confundida con el comando SELECT de SQL el cual hace más que una selección.

- Proyección

Es el proceso de recuperar una o más columnas de una tabla en un orden específico y remover cualquier fila duplicada.

- Unión

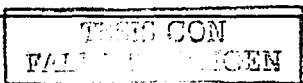
La unión es el proceso de recuperar columnas de diferentes tablas, donde los valores de una columna en común de cada tabla, son iguales. La habilidad de unir dos o más tablas es poderosa. La operación de unión distingue a los sistemas relacionales de los no relacionales.

RESTRICCIONES DE INTEGRIDAD

- Datos Unicos.

Cada entidad debe tener un identificador único así que cada fila es diferente de las demás. En teoría ninguna fila puede ser duplicada en una tabla.

Una tabla puede tener una llave primaria la cual es una columna o un grupo de las mismas cuyo valor único identifica a cada fila.



- **Indices**

Un índice, es un conjunto ordenado de punteros para los datos en una tabla. Se basa en los valores de los datos de una o más columnas de la tabla. Un índice provee dos beneficios:

- Mejora la ejecución, pues hace el acceso a los datos más rápido.
- Asegura unicidad. Una tabla con un índice único no puede tener dos filas con los mismos valores en la columna o columnas que forman la llave del índice.

- **Vistas**

Una vista, es un camino alternativo para la representación de datos que existen en una o más tablas. Puede incluir todas o unas de las columnas de una o más tablas base. Una vista puede estar basada en otra vista o en una combinación de vistas y tablas.

Una vista se ve como una tabla y se puede tratar como tal. El nombre de la vista puede ser usada en mandatos de SQL como si fuera el nombre de una tabla. Algunas operaciones no se pueden hacer sobre una vista, pero no es necesario saber que una aparente tabla es actualmente una vista.

Además puede mostrar al usuario el mismo dato en diferentes sentidos. Los programadores asignados, los administradores de la base de datos, y los usuarios finales pueden ver el dato como una petición de sus necesidades.

El relativamente pequeño número de conceptos existentes en la teoría relacional, ha ayudado a convertir esta técnica en un estándar industrial. Los diseñadores relacionales han sido capaces de aislar la complejidad del desarrollo físico de las bases de datos con respecto al diseño lógico del sistema, proporcionando así una interfaz sencilla para los diseñadores de aplicación.

Durante las dos últimas décadas hemos madurado como industria. Muchos de nosotros hemos sentido la necesidad de contar con un entorno de modelización más rico que se adaptara mejor a los recientes avances que tienen hacia una modelización genérica.



I.3.2 MODELO JERARQUICO

La estructura lógica en la cual se sustenta una base de datos jerárquica, es el árbol. Se compone de un nodo raíz y varios nodos sucesores ordenados jerárquicamente. Cada nodo, representa una entidad y las relaciones entre entidades son la conexiones entre los nodos.

El nodo, colocado en la parte superior es llamado padre y los nodos inferiores son los hijos. En el sistema jerárquico, las conexiones entre archivos no dependen de la información contenida en ellos; se definen al inicio y son fijos. La característica sobresaliente de este modelo es el manejo de la conexión uno a muchos, entre un padre y varios hijos, en otras palabras, cada hijo sólo tiene un padre.

DESVENTAJAS EN EL ENFOQUE JERARQUICO.

- No modela sencillamente las relaciones Muchos a Muchos.
- Anomalías de inserción.
- Anomalías de borrado.
- Anomalías de actualización.
- Se pueden dar consultas inconsistentes.

I.3.3 MODELO DE RED

Los datos se representan como registros ligados formando un conjunto de datos intersectados. La base de datos de red, a diferencia de las jerárquicas, permite cualquier conexión entre entidades, es decir, se pueden representar relaciones de muchos a muchos. En una red, un hijo puede tener varios padres y varios hijos a la vez.

DESVENTAJAS EN EL ENFOQUE DE RED.

- Resulta difícil definir nuevas relaciones.
- Es complicado darle mantenimiento ya que cualquier cambio en la estructura requiere una descarga en los datos.
- Representa desperdicio de recursos.

- Anomalías de inserción.
- Anomalías de borrado.

1.4 ARQUITECTURAS DE BASES DE DATOS

1.4.1 ARQUITECTURA CLIENTE-SERVIDOR

Esta arquitectura de base de datos tradicional, es utilizada por DB2 y las bases de datos de minicomputadoras tales como Oracle, Informix o Ingres. En esta arquitectura tanto como en el DBMS los datos físicos residen en un mainframe o minicomputadora central, junto con el programa de aplicación que acepta entradas desde la terminal del usuario y muestra los datos en la pantalla del usuario.

Supongamos que el usuario teclea una consulta que requiere una búsqueda secuencial en una base de datos, como por ejemplo la petición para hallar la cantidad media de mercancías de todos los pedidos. El DBMS recibe la consulta, explora la base de datos para acceder a cada uno de los registros de datos del disco calcula el promedio y muestra el resultado en la pantalla. Tanto el procesamiento de la aplicación como el de la base de datos se producen en la computadora central, y como el sistema es compartido por muchos usuarios, cada usuario experimenta una degradación del rendimiento cuando el sistema está más cargado.

La arquitectura Cliente-Servidor es una forma de procesamiento distribuido donde las actividades computacionales son compartidas entre computadoras cooperativas y redes de trabajo. Una aplicación es funcionalmente dividida en dos o más programas los cuales ejecutan una diferente computadora y se comunica con otra a través de mensajes por medio de la red.

Los programas clientes corren sobre PC's de usuarios. Los programas servidor corren sobre computadoras más poderosas. El cliente manda una solicitud al servidor y cuando este lo recibe, lo procesa y manda el resultado al cliente nuevamente.

Tipicamente una red de área local contiene más clientes que servidores. Las computadoras de los clientes no pueden compartir sus recursos o usar los recursos de otra computadora cliente.

Así, en una arquitectura Cliente-Servidor, el usuario puede usar toda la capacidad de acceso disponible por el computador servidor, el cual no está disponible localmente en el escritorio del cliente. Una aplicación Cliente-Servidor tiene la capacidad para ambos, para el cliente y para el servidor.

El servidor es usualmente la computadora más poderosa en el área local de la red. Ciertos criterios hacen de algunos sistemas mejor situados para servidores, tales como un alto grado de transferencia de entrada / salida, la velocidad con la cual un servidor procesa las tareas de la red. Alta capacidad de disco y múltiples controladores son otro criterio.

Un servidor es usualmente multitarea, así, este puede servir múltiples clientes simultáneamente.

Una aplicación cliente usa SQL para comunicarse con el servidor de la base de datos. El cliente es responsable por ejecutar la aplicación y operar un comando SQL para el envío a través de la red al servidor de la base de datos. El servidor de la base de datos es responsable por la recepción y ejecución del comando SQL, envío de datos y código de estado de respuesta al cliente.

1.4.2 ARQUITECTURA DISTRIBUIDA

En esta arquitectura las bases de datos están situadas en servidores separados compartiendo datos entre ellos, cada servidor puede estar aislado físicamente de todos los demás sin estar aislado lógicamente de ellos. Una implementación típica de esta configuración es la de servidores situados en las oficinas centrales de una empresa que están comunicados con servidores departamentales situados en regiones alternas. Cada servidor admite aplicaciones de cliente, y también tiene la capacidad de comunicarse con los demás servidores de la red.

Cuando uno de los servidores, envía una solicitud de base de datos a otro servidor, el servidor emisor, actúa como un cliente. El servidor que recibe la solicitud ejecuta la sentencia SQL que se le pasa, y devuelve los resultados y las posibles condiciones de error al emisor.

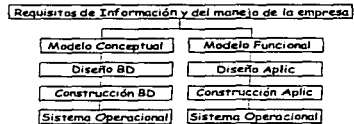
BENEFICIOS DE UN SISTEMA DE BASE DE DATOS DISTRIBUIDA

- Localización Independiente. La localización de la aplicación del cliente es independiente de la localización del dato.
- Localización transparente. El usuario puede acceder la base de datos sin conocimiento de su localización, y se puede mover la base de datos sin tener efectos sobre las aplicaciones de los usuarios.
- Incremento del crecimiento de las aplicaciones. Puede elevar el grado de una aplicación añadiendo más clientes o con la compra de una máquina más poderosa como el servidor de la base de datos.
- Autonomía local. Se puede mantener cada base de datos separadamente de otras bases de datos y administrar los datos compartidos, puede asignar programadores para el soporte de las aplicaciones.
- Independencia de hardware y software. Es posible usar hardware y software para diferentes manufacturas y se puede emparejar para los requerimientos de sus aplicaciones.
- Procesamiento distribuido. El procesamiento y almacenamiento de datos entre muchas computadoras, puede ser compartido.
- Incremento de la disponibilidad y formalidad. Si un servidor de base de datos se cae, solamente sus clientes son afectados, no la organización entera.

1.5 ETAPAS DE DESARROLLO DE UNA BASE DE DATOS

1.5.1 MODELO CONCEPTUAL DE DATOS

El desarrollo de un sistema operacional en una empresa, debe realizarse en partes claras y bien definidas las cuáles se presentan en los siguientes diagramas. Tomaremos los puntos más importantes de estos, para explicarlos.





ESTRATEGIA

Es la primera etapa. En ella, el usuario dá a conocer cómo se llevan a cabo los procesos en una empresa y cómo se maneja la información actualmente. El analista de sistemas puede auxiliarse de cuestionarios para dialogar, con los usuarios. El objetivo de esta etapa es generar el documento de especificación de requerimientos.

La especificación de requerimientos representa el contrato entre el usuario (o cliente) y el equipo que va a desarrollar el sistema. Esta especificación, toma como base la descripción inicial de los usuarios. Los analistas y usuarios juntos, determinan como se llevarán a cabo los procesos donde se maneja información, además determinarán también qué procesos serán automatizados y cuáles no lo serán.

Para el desarrollo de una nueva base de datos se debe de empezar por tomar la estrategia a seguir, estudiando los requerimientos del nuevo proyecto ó sistema, para ello se debe de aprender acerca de las necesidades del negocio.

Una de las principales partes de estudio de la estrategia, es saber con qué tecnología se cuenta. Se debe realizar un aproximado del costo del hardware, software y del desarrollo.

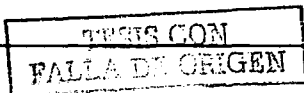
Si una base de datos no se encuentra bien diseñada desde el inicio, esto provocaría que la aplicación no corriera eficientemente.

ANÁLISIS

La fase de análisis, es la investigación a detalle del proceso requerido del negocio (funciones) y la información requerida para ejecutar esas funciones (entidades con sus atributos y sus relaciones).

1.5.1.1 MODELO ENTIDAD-RELACION

En este subcapítulo veremos los conceptos básicos del Modelo Entidad-Relación; en los cuales, nos apoyaremos para la realización del análisis de las bases de datos.



1.5.1.1.1 ENTIDAD

Una entidad, es una persona, objeto o lugar, que cae dentro del alcance del sistema, acerca de la cual el sistema debe mantener, correlacionar y desplegar información. Es algo que es significativo para la problemática tratada y sobre la cual se dice, se conoce, o se tiene información relevante.

Cada entidad, debe tener un nombre único dentro del sistema, lo que indica que no pueden existir dos entidades con el mismo nombre dentro del mismo sistema

Para identificar las entidades dentro del sistema, es necesario formularse las siguientes preguntas:

- ¿ Se le puede dar un nombre ?
- ¿ Tiene asociada información o características de interés ?
- ¿ Cuáles atributos serían identificadores únicos ?
- ¿ Se puede justificar la entidad, dándole una frase ?
- Examine conceptos, ¿ Cuales son significativos ?

Una entidad, es representada por medio de una caja (Cuadrado o Rectángulo) con las esquinas redondeadas, dentro de ella se escribe su nombre con letras mayúsculas y en singular. Si tiene sinónimos (Nombres Alternos para la Entidad) deberán de colocarse entre paréntesis o con diagonal, con la finalidad de no crear otra entidad.

Ejemplo:

Maestro, Estudiante, Curso, Avión, Aeropuerto, Vuelo, Empleado, Departamento.

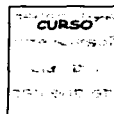
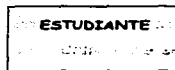
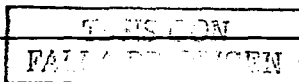


Figura 1.5.1.1.1a Entidades



1.5.1.1.2 ATRIBUTO

Información y datos específicos de una entidad. Es la descripción de las categorías relevantes de datos de una entidad. También representa a la información específica que necesita ser mantenida. En términos generales, es una característica o cualidad de una entidad.

Algo que es importante mencionar es el hecho de que un atributo pertenece a una y sólo una entidad, y no la entidad pertenece a algún atributo.

Ejemplo:

Si consideramos una entidad denominada Estudiante y una denominada Vuelo, algunos de sus atributos podrían ser los siguientes:

Nombre, Número, Edad, Sexo para la entidad Empleado y Número de Vuelo, Compañía, Horario, Origen, Destino para la entidad Vuelo.

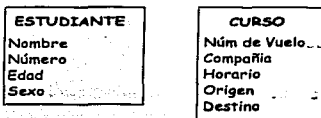


Figura 1.5.1.1.2a Atributos

1.5.1.1.3 RELACIONES

Es una descripción de como las entidades se relacionan. Es cualquier forma significativa en la cual dos entidades pueden ser asociadas. Una relación requiere de una o más entidades, las cuales deben caer dentro del alcance del sistema. Es, una Asociación bidireccional significativa entre dos entidades, o entre una sola entidad.

Es importante mencionar que una relación no es entre atributos de dos entidades, sino entre dos entidades; además de que debe permitir ser expresada en ambos sentidos.



Ejemplo:

Considerando las entidades empleado, salario, avión, vuelo, maestro, estudiante; se pueden deducir las siguientes relaciones:

- A un empleado se le puede asignar un salario.
- A cada avión se le pueden asignar varios vuelos.
- Cada maestro tiene varios estudiantes

Con ello, podemos deducir que un empleado está relacionado con un salario, que un avión esta relacionado con vuelos y que los maestros están relacionados con los estudiantes.

Las convenciones para diagramar las relaciones, son las siguientes:

- Una línea entre dos entidades.
- Nombre de la relación en letras minúsculas
- Opcionalidad
 - Mandatoria (Línea Continua) _____
 - Opcional (Línea Punteada) - - - - -
- Grado
 - Uno a Muchos 
 - Uno y solo uno 

Cada relación necesita 3 piezas de información por separado:

Nombre.- Una descripción de las reglas del negocio asociando a las entidades, por ejemplo enseñado por o asignado a.

Opcionalidad.- La regla del negocio que determina si la relación siempre debe de existir o si no es necesario que exista para cada una de las instancias de la entidad.

Grado.- La regla del negocio que indica cuántas relaciones pueden existir.

CONVENCIONES PARA DIAGRAMAR RELACIONES

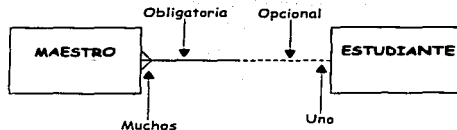


Figura 1.5.1.1.3a Relaciones



De acuerdo a las necesidades de información del negocio existen tres tipos de relaciones las cuales son:

- Relación Uno a Uno.
- Relación Uno a Muchos
- Relación Muchos a Muchos

1.5.1.1.3.1 UNO A UNO

Esta relación tiene un grado de uno a uno y solo uno en ambas direcciones, o sea que es mandatoria en ambos sentidos. Este tipo de relaciones son muy raras, si se identifica este tipo de relaciones, se debería de investigar con mayor profundidad el origen de ella, y si realmente es necesaria para formar parte del sistema o de lo contrario poderla omitir, indefinidamente.

Ejemplo.

Considerando la entidad Bicicleta y la entidad Ciclista podemos deducir la siguiente relación uno a uno:

- Cada bicicleta puede ser conducida por uno y sólo un ciclista
- Cada ciclista puede ser conductor de una y sólo una bicicleta.

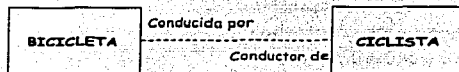
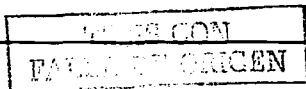


Figura 1.5.1.1.3.1a Relaciones Uno a Uno

1.5.1.1.3.2 UNO A MUCHOS

Esta relación es la más común en los diagramas Entidad-Relación. Tiene un grado de uno a muchos en una dirección y de uno y sólo uno en la otra dirección, generalmente la primera de ella es obligatoria y la segunda es opcional. Por ejemplo se puede dibujar una relación del tipo uno a muchos entre las entidades departamento y empleado la cual se describe de la siguiente forma:

- Cada empleado debe estar asignado a uno y sólo un departamento.
- Cada departamento puede ser responsable de uno a más, empleados.



Ambas sintaxis significan lo mismo que un departamento está asociado con cualquier número de empleados incluyendo cero, y que cualquier empleado está asociado, con un departamento. En notación ERD, esto se dibuja de la siguiente manera:

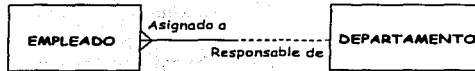


Figura I.5.1.1.3.2a Relación Uno a Muchos

I.5.1.1.3.3 MUCHOS A MUCHOS

Esta relación tiene un grado de uno a muchos en ambas direcciones y son muy comunes al inicio del análisis de un sistema por lo que se requiere de un análisis más profundo para crear entidades intermedias que logren separar la relación de muchos a muchos.

Un ejemplo claro de este tipo de relaciones es la de un estudiante con un curso la cual se describe a continuación:

- Un estudiante puede estar asignado a uno o muchos cursos
- Un curso puede ser tomado por uno o muchos estudiantes.

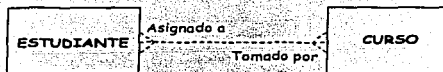


Figura I.5.1.1.3.3a Relación Muchos a Muchos

I.5.1.1.4 IDENTIFICADORES UNICOS

Cada una de las tablas poseen una columna o una combinación de las mismas que permiten identificar cada registro, en la tabla. Este identificador único es denominado Llave Primaria (Primary Key). Esta llave provee la utilidad de distinguir



cada uno de los registros dentro de una tabla. Esto permite al usuario y a la base de datos identificar y localizar algún registro particular dentro de la tabla.

En teoría, ninguna fila puede ser duplicada en una tabla. Considerando una tabla con ventas ordenadas con columna para la fecha, código del producto, cantidad e identificador del cliente. Si diez clientes ordenados son atendidos en la mañana y otros diez atendidos en la tarde, esto crearía dos filas duplicadas en la tabla, una para cada orden. Para asegurar la creación de filas únicas, se debe crear una columna adicional para almacenar la hora en la cual la orden tuvo lugar, o almacenar la generación única de una secuencia de números (tal como el número de factura).

Una tabla puede tener una llave primaria cuyo valor único identifica a cada fila. Otra tabla puede tener llaves foráneas, la cual es una columna o grupo de columnas cuyos valores son iguales a los de esas primeras llaves primarias de una tabla. Una de las reglas de integridad referencial es que el valor de la llave foránea de la tabla debe ser igual al valor de otra llave primaria de la tabla.

Durante la etapa de diseño se debe de analizar cada entidad con el fin de encontrar aquellos atributos que sean candidatos para formar parte de identificadores únicos. Una vez detectados estos atributos sólo se marca del lado izquierdo con un símbolo de # el cual hace referencia que el atributo de su derecha es el identificador único, de esa entidad.

Por ejemplo: Se muestra que la entidad pedido tiene dos atributos que conjuntamente forman un identificador único por fila.

| |
|-------------|
| PEDIDO |
| # Id_Pedido |
| # Región |

Figura 1.5.1.1.4a Llaves Primarias

1.5.1.1.5 NORMALIZACION

Representan el conjunto de normas que nos ayudan a diseñar una estructura de bases de datos óptima para su implementación, gestión y explotación desde distintas

aplicaciones, consiguiendo independencia de las mismas. El creador de estas normas fué E.F.Codd, quién formulo las 3 primeras formas normales (1fn, 2fn y 3fn).

Ventajas

- Permite simplificar, entender mejor y sobretodo implementarse en base de datos relacionales.
- Un modelo normalizado, se traduce sin problemas en un diseño de base de datos relacional normalizada.
- Facilita el proceso de diseño y evita no regresar a etapas anteriores.

1.5.1.1.5.1. PRIMERA FORMA NORMAL (PFN).

Cualquier tabla que cumpla con la definición de relación está en la primera forma normal. Para que una tabla sea una relación debe tener las siguientes características:

- Cada celda de la tabla debe contener un sólo valor, y no se permiten grupos o arreglos como valores.
- Todas las entradas en cualquier columna (atributo) debe ser del mismo tipo.
- Cada columna debe tener un nombre único, pero el orden de las columnas en la tabla es insignificante.

Finalmente, dos renglones en una tabla no pueden ser idénticos, pero el orden de los renglones es insignificante.

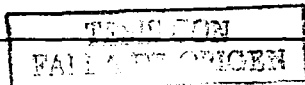
1.5.1.1.5.2 SEGUNDA FORMA NORMAL (SFN).

Una relación, está en segunda forma normal si todos sus atributos no-llave son dependientes de toda la llave.

Según la definición, todas las relaciones que tienen por llave a uno sólo de sus atributos, están automáticamente en segunda forma normal. Si la llave es de un sólo atributo, todos los atributos no-llave son dependientes de toda la llave; ahí no pueden existir dependencias parciales.

1.5.1.1.5.3 TERCERA FORMA NORMAL (TFN).

Una relación está en tercera forma normal si está en segunda forma normal y no tiene dependencias transitivas, es decir, que uno de sus atributos no llave dependa directamente de otro atributo no llave.





Es importante mencionar que una dependencia transitiva, genera anomalías de borrado e inserción de datos en una entidad.

1.5.2 MODELO LOGICO DE BASE DE DATOS

El modelo lógico de una base de datos relacional implica tomar como base un modelo de información, el cual, es una vista de los requerimientos de información del negocio y convertirlo en una representación de software. Esto es, las entidades se convierten en tablas, sus atributos en columnas, los identificadores únicos en llaves primarias y las relaciones en llaves foráneas.

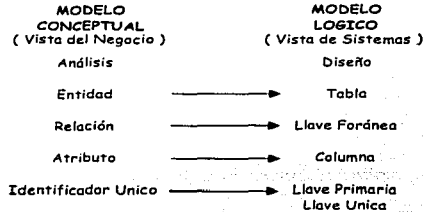


Figura 1.5.2a Modelo Lógico de Base de Datos

Antes de realizar la conversión del modelo entidad-relación en un modelo lógico inicial, es necesario asegurarse de que todos los elementos involucrados estén completos.

- Todas las entidades deben tener su descripción.
 - Todas las entidades deben poseer un nombre.
 - Todas las relaciones muchos a muchos deben ser resueltas.
 - Todas las entidades deben poseer un identificador único.
 - El modelo debe estar normalizado.
 - La Opcionalidad de los atributos debe estar especificada.
- Las reglas de integridad referencial deben estar bien definidas.

1.5.2.1 MAPEO DE ENTIDADES-TABLAS.

Una simple entidad es única y no envuelve ninguna relación. Cada una de las entidades del modelo entidad-relación, deben convertirse en tablas. Cada una de las tablas, deben poseer la facilidad de regresarse a la entidad de la cual, fueron generadas. El nombre de una tabla es asignado por el diseñador, generalmente es el mismo que posee la entidad. con la única diferencia de que éste, es escrito en plural; ya que una tabla contiene un conjunto de registros.

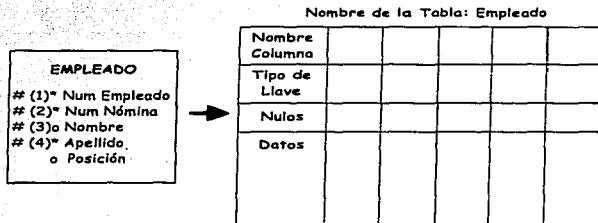


Figura 1.5.2.1a Mapeo de Tablas

1.5.2.2 MAPEO DE ATRIBUTOS - COLUMNAS.

El mapeo de atributos se realiza considerando, los siguientes puntos:

- Todos y cada uno de los atributos de una entidad se listan como encabezados de columnas, con el mismo nombre que poseen en el diagrama entidad-relación.
- Los espacios existentes entre palabras deben ser sustituidos por el símbolo (_).
- Evitar el uso de palabras reservadas de SQL como nombres de columnas.
- Usar abreviaciones consistentes para evitar confusiones.
- No utilizar el uso mismo nombre para columna y tabla.

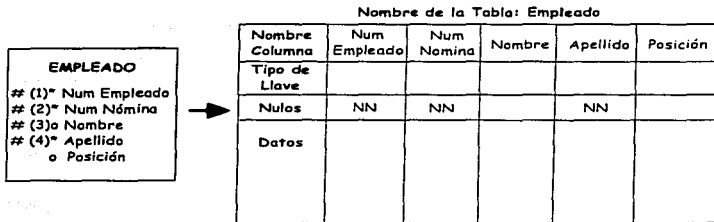


Figura 1.5.2.2 Mapeo de Columnas

1.5.2.3 MAPEO DE IDENTIFICADORES UNICOS – LLAVES PRIMARIAS

Todas las columnas que son candidatas a ser llaves primarias tendrán que ser obligatorias y únicas, o, en otros términos, indispensables para el procesamiento de datos. En caso de que la tabla no posea columnas obligatorias para su uso como llave primaria, será necesario crear llaves surrogadas. Esto es, se crea una columna de referencia únicamente para propósitos técnicos pero no es requerimiento del negocio. Algunas veces, una columna ha sido elegida para ser la llave primaria, pero puede existir otra columna que fué definida como un identificador único secundario. En este caso, los restantes identificadores únicos serán denominados llaves únicas.

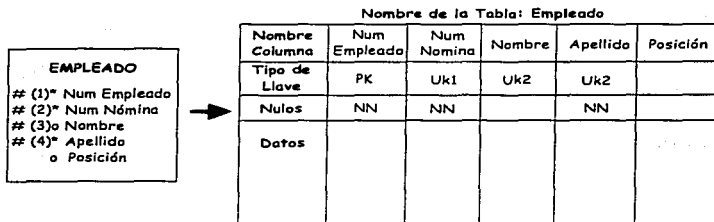
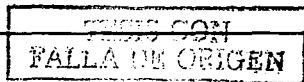


Figura 1.5.2.3 Mapeo de Llaves Primarias



1.5.2.4 MAPEO DE RELACIONES – LLAVES FORANEAS

- Relaciones Muchos a Uno.

Cuando una de las relaciones muchos a uno ha sido designada entre dos tablas, es necesario añadir la columna de llave primaria de la tabla maestra a la tabla detalle. Es necesario determinar un nombre a la nueva columna; generalmente se usa el nombre de la tabla, de la cual se originó. Esta nueva columna es conocida como llave foránea (Fk).

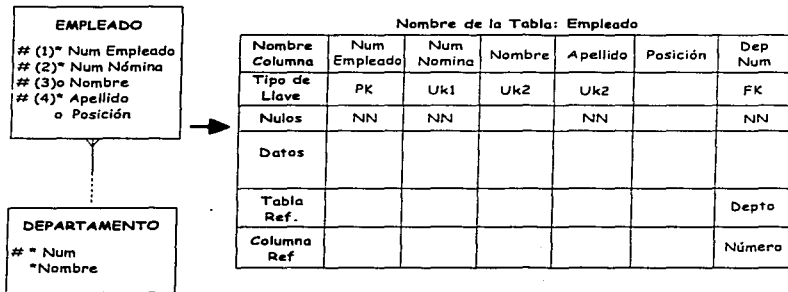


Figura 1.5.2.4 Mapeo de Llaves Foráneas

Relaciones Uno a Uno

Para una relación obligatoria uno a uno, la columna de llave foránea deberá colocarse en la tabla donde la relación obligatoria termina. Esta tabla se convierte en tabla detalle y la otra, en maestra.

Para una relación uno a uno opcional en las dos direcciones, deben de tomarse en cuenta dos consideraciones para determinar en que tabla se colocara la llave foránea.

1.- Si un renglón en una tabla es creado antes que en la otra, entonces esta tabla será tabla maestra y la columna de llave foránea deberá colocarse en la otra tabla.

2.- Si una tabla tendrá menos registros que la otra, entonces esta tabla podría ser la maestra y crear la llave foránea en la otra tabla.

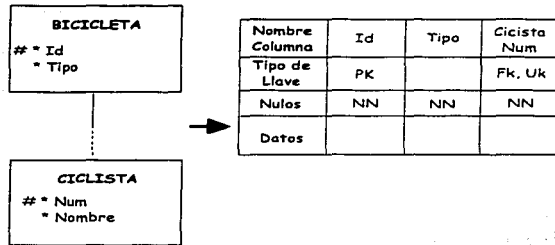


Figura 1.5.2.4b Relación Uno a Uno de Llave Foránea

1.5.3 CONSTRUCCIÓN FÍSICA DE LA BASE DE DATOS

Esta etapa se apoya en el diseño. Consiste en generar en un lenguaje de programación los módulos especificados en la etapa de diseño.

En gran medida esta etapa depende de la habilidad del programador para expresar de una manera eficiente lo contenido en el diseño.

1.6 TENDENCIAS DE LAS BASES DE DATOS.

La historia de las bases de datos se extiende desde mediados de los años sesenta, y se ha caracterizado por su excepcional productividad y su impresionante impacto económico. En efecto, en estos momentos el mercado de las bases de datos ronda, según diversas fuentes, los 8.000 millones de dólares, con un crecimiento de 20 % anual. A estas cantidades hay que añadir las inversiones que hacen los usuarios en aplicaciones desarrolladas sobre sistemas de gestiones de bases de datos.

En los últimos años venimos asistiendo un avance espectacular en la tecnología de bases de datos. Temas que hasta hace poco parecían exclusivos de laboratorios y centros de investigación comienzan a aparecer en las últimas versiones de algunos

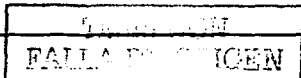
Sistemas de Gestión de Bases de Datos y nuevos productos: Bases de Datos Multimedia, Activas, Deductivas, Orientadas a Objetos, móviles, paralelas, etc. etc.

Esta nueva generación de bases de datos (Tercera) se caracteriza por proporcionar capacidades de gestión de datos al igual que sus predecesoras, permitiendo que grandes cantidades de datos persistentes, sean compartidos por muchos usuarios. También proporcionan gestión de objetos permitiendo tipos de datos mucho más complejos, objetos multimedia, datos derivados y encapsulamiento de la semántica de los datos, así como otras nuevas capacidades. Algunos proporcionan incluso gestión de conocimiento, soportando un gran número de reglas complejas para inferencia automática de información y también para mantener las restricciones de integridad entre datos.

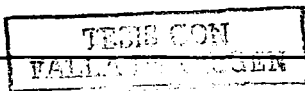
Desde finales de los años 70s la comunidad de investigadores en bases de datos tomó conciencia de la naturaleza fundamentalmente descentralizada de muchas organizaciones y, por lo tanto, de la importancia de acercar los datos a sus usuarios responsables. Surge así la idea de tener bases de datos en múltiples sitios, en lugar de una sólo base de datos centralizada. Entre los beneficios de este enfoque están la mayor disponibilidad de los datos frente a fallas, y la disminución en costos de comunicaciones remotas.

Desde comienzos de los 80's se viene trabajando a nivel investigación en el manejo de consultas y transacciones distribuidas así como en el soporte de múltiples copias de un dato. Los resultados de la investigación, se han venido transfiriendo de los laboratorios a los productos comerciales es así como hoy en día todos los vendedores importantes de Sistemas de Gestión de Bases de Datos ofrecen servicios de distribución.

No obstante lo anterior, son muy pocas las organizaciones que se la han jugado a fondo con la distribución. Esto puede deberse en parte a la falta de experiencia en el manejo de la tecnología, a no disponer de metodologías de diseño de distribución probadas y soportadas con herramientas estilo CASE, y a la escasa transparencia a la distribución ofrecida actualmente por los productos comerciales.



Aunque los conceptos y desarrollos teóricos de las bases de datos distribuidas están maduros desde hace varios años, y los Sistemas de Gestión de Bases de Datos, ofrecen buenas funcionalidades para distribución, son escasas las organizaciones que están utilizando realmente esta tecnología. Se espera que esta situación empiece a cambiar en el mediano plazo. Para ello se requiere de buenas metodologías de diseño de distribución y de una mayor transparencia en la distribución por parte de los Sistemas de Gestión de Bases de Datos.





CAPITULO II

ELEMENTOS DE ORACLE

TESIS CON
FALLA DE OXIGEN

TESIS CON
FALLA DE CUBIEN



ELEMENTOS DE ORACLE 8

II.1 QUE ES ORACLE ?

Esta compañía, fundada en 1977, es la mayor proveedora mundial de "software" de Bases de Datos y servicios para la gestión de información y la segunda mayor compañía mundial de "software". Cuenta con 34.000 empleados en más de 140 países.

ORACLE, es un sistema para la gestión de bases de datos relacionales (SGBDR). Las bases de datos relacionales están construidas en torno al modelo relacional basado en el álgebra relacional. El rigor de la teoría relacional ofrece un marco teórico sólido para la concepción y utilización de las bases de datos. Los lazos entre los diferentes objetos no son definidos físicamente, sino semánticamente lo que permite una gran flexibilidad. Actualmente, el modelo relacional es el más ampliamente aceptado. Sus características principales son:

- Estructuras. Objetos bien definidos que almacenan los datos de la base.
- Operaciones. Acciones que permiten a los usuarios manipular los datos y estructuras de una B.D.
- Reglas de Integridad. Leyes que rigen las operaciones sobre datos y estructuras.

Un SGBDR ofrece importantes beneficios como:

- Independencia entre el almacenamiento físico de los datos y la estructura lógica de la B.D.
- Fácil y variable acceso a todos los datos.
- Gran flexibilidad en el diseño de la B.D. Reduce el almacenamiento de datos y redundancias.

CARACTERISTICAS PRINCIPALES DE ORACLE

- Mecanismos de Seguridad. Oracle, cuenta con mecanismos de seguridad que controlan el acceso a los datos sensibles utilizando un conjunto de privilegios, en función del nombre con el que se conecta a la base de datos. A los usuarios se les concede derechos para consultar, modificar y crear datos. Los clientes usan estos



mecanismos para asegurarse de que ciertos usuarios puedan consultar los datos de carácter sensible, mientras que a otros, se les niega dicha posibilidad.

- Realización de Copias de Seguridad y Recuperación. Las copias de seguridad permiten crear una copia secundaria de los datos almacenados; mientras que los procedimientos de recuperación restauran los datos a partir de una copia de seguridad. Estas dos estrategias, permiten minimizar la pérdida de datos y son demasiado útiles en posibles contingencias.

- Gestión de Espacio. Oracle, ofrece una gestión flexible de espacio, ya que puede asignar cierto espacio de disco para el almacenamiento de los datos y controlar las siguientes asignaciones instruyendo a Oracle sobre cuanto espacio reservar para los requerimientos futuros.

- Conectividad de Carácter Abierto. Oracle, proporciona conectividad hacia y desde software de otros fabricantes, se puede trabajar con información almacenada con otros sistemas de bases de datos como lo es DB2 de IBM, Sybase o Microsoft Access. Al igual se pueden almacenar datos en la base de datos de Oracle y acceder a ellos desde otros software como Visual Basic, Powerbuilder, java, etc.

- Procesamiento de Transacciones On-line (OLTP) a un nivel comparable al ofrecido por un mainframe.

- Aumento del número de usuarios concurrentes.

- Operaciones avanzadas de colas, etc. (Los entornos OLTP son aquellos en que se realizan concurrentemente un gran número de transacciones de corta duración, como puede ser un gran sistema bancario).

- Consulta en Paralelo. Esta opción permite que los clientes aprovechen las ventajas de procesar las consultas en computadoras con más de una unidad central de proceso (CPU); ya que cuando se genera una consulta con esta opción, el servidor divide la tarea en varios procesos que funcionan en forma simultánea, repartíendose el procesamiento de la consulta , los resultados se combinan y se presentan al usuario cuando, están listos.

- Mejor Soporte de Sistemas Distribuidos. Oracle, cuenta con una nueva tecnología de réplica de datos y un lenguaje de programación que realiza la operaciones más rápido, dentro del servidor, ya que actúa directamente con la Base de datos, éste, se denomina PL/SQL.



- Alta Capacidad de Gestión con Tablas e Índices Divididos en Particiones. Oracle, permite dividir las tablas en partes más pequeñas en función de un rango de valores de alguno de sus datos, lo que facilita la administración, mejora la disponibilidad de los datos y aumenta el rendimiento de las consultas.

Oracle, posee múltiples herramientas para el desarrollo de aplicaciones. En el nivel más bajo se encuentran las comúnmente llamadas OCI (Oracle Call Interface o Interfaz de Llamadas de Oracle), Librerías de programación que permiten un control total de la base de datos desde lenguaje C.

II.2 ARQUITECTURA DE ORACLE

II.2.1 INSTANCIA

Una instancia de base de datos, es un conjunto de estructuras de memoria y procesos que acceden a los archivos de datos.

Los parámetros que determinan el tamaño y composición de una instancia, están almacenados en un archivo llamado init.ora. Este archivo, es leído durante el arranque de la base de datos y puede ser modificado por el administrador de base de datos (DBA). Cualquier modificación de este archivo no tiene efecto hasta la siguiente vez que se arranque la base de datos.

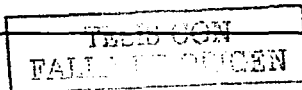
II.2.1.1 MEMORIA

Oracle mantiene dos estructuras principales de memoria: el Area Global de Sistema (System Global Area SGA) y el Area Global de Programa (Program Global Area PGA).

AREA GLOBAL DE SISTEMA

El SGA, es la zona de memoria en la que la base de datos Oracle guarda información sobre su estado. Esta estructura de memoria esta disponible para todos los procesos, por eso se dice que esta compartida. El Area Global de Sistema, sirve además para facilitar la transferencia de información entre usuarios y también almacena la información estructural de la base de datos más frecuentemente requerida.

LA SGA se divide en varias partes:





- **Buffers de Base de Datos (Database Buffer Cache)**

Es el caché que almacena los bloques de datos leídos de los segmentos de datos de la Base de Datos, tales como tablas, índices y clusters. Los bloques modificados, se llaman bloques sucios. El tamaño de buffer caché se fija por el parámetro `DB_BLOCK_BUFFERS` del archivo `init.ora`.

Como el tamaño del buffer suele ser pequeño para almacenar todos los bloques de datos leídos, su gestión se hace mediante el algoritmo LRU.

- **Buffers Redo Log**

Los registros Redo describen los cambios realizados en la Base de Datos y son escritos en los archivos redo log, para que puedan ser utilizados en las operaciones de recuperación de la Base de Datos. Pero antes de ser escritos en los archivos redo log son escritos en un caché de la SGA llamado redo log buffer. El servidor, escribe periódicamente los registros redo log en los archivos redo log. El tamaño del buffer redo log se fija por el parámetro `LOG_BUFFER`.

- **Area de SQL Compartido. (Shared SQL Pool)**

En esta zona, se encuentran las sentencias SQL que han sido analizadas. El análisis sintáctico de las sentencias SQL lleva su tiempo y Oracle mantiene las estructuras asociadas a cada sentencia SQL, analizada durante el tiempo que pueda, para ver si puede reutilizarlas. Antes de analizar una sentencia SQL, Oracle determina si encuentra otra sentencia exactamente igual en la zona de SQL compartido. Si es así, no la analiza y pasa directamente a ejecutar la que mantiene en memoria. De esta manera, se premia la uniformidad en la programación de las aplicaciones. La igualdad se entiende que es lexicográfica, espacios en blanco y variables incluidas.

El contenido de la zona de SQL compartido es:

- Plan de ejecución de la sentencia SQL.
- Texto de la sentencia.
- Lista de objetos referenciados.

Los pasos de procesamiento de cada petición de análisis de una sentencia SQL son:

- Comprobar si la sentencia se encuentra en el área compartida.
- Comprobar si los objetos referenciados son los mismos.
- Comprobar si el usuario tiene acceso a los objetos referenciados.

Si no, la sentencia es nueva, se analiza y los datos de análisis se almacenan en la zona de SQL compartida. También se almacena en la zona de SQL compartido el caché del diccionario. La información sobre los objetos de la Base de Datos se encuentra almacenada en las tablas del diccionario. Cuando ésta información se necesita, se leen las tablas del diccionario y su información se guarda en el caché del diccionario de la SGA.

Este caché, también se administra mediante el algoritmo LRU. El tamaño del caché está gestionado internamente por el servidor, pero es parte del shared pool, cuyo tamaño viene determinado por el parámetro SHARED_POOL_SIZE.

AREA GLOBAL DE PROGRAMA

El PGA, es una área de memoria utilizada por un proceso Oracle. Esta zona de memoria no se puede compartir.

II.2.1.2 PROCESOS

La instancia de Oracle, está compuesta de una serie de procesos que son el enlace entre las estructuras físicas y las estructuras de memoria; cada vez que la instancia ha sido dada de baja, estos procesos mueren y cuando la instancia se dá de alta vuelven a levantarse de forma transparente para el usuario.

A continuación se describen cada proceso y el papel que juega en la gestión de la base de datos. Todo esto se puede ver en la siguiente figura:

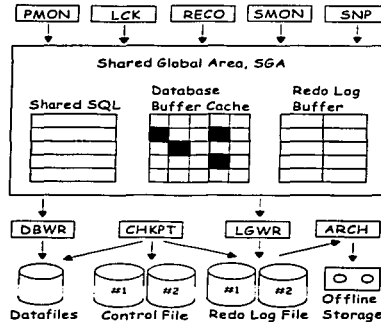


Figura II.2.1.2 Procesos de Oracle

- **Monitor de Sistema. (System Monitor (SMON)).**

El SMON, es el supervisor del sistema y se encarga de todas las recuperaciones que sean necesarias durante el arranque. Esto puede ser necesario, si la Base de Datos dejó de funcionar inesperadamente por fallo físico, lógico u otras causas. Este proceso, realiza la recuperación de la instancia de Base de Datos a partir de los archivos redo log. Además, limpia los segmentos temporales no utilizados y compacta los huecos libres contiguos en los archivos de datos. Este proceso, se despierta regularmente para comprobar si debe intervenir.

- **Monitor de Proceso. (Process Monitor (PMON)).**

Restaura las transacciones no validadas de los procesos de usuario que abortan, liberando los bloqueos y los recursos de la SGA. Asume la identidad del usuario que ha fallado, liberando todos los recursos de la Base de Datos que estuviera utilizando, y anula la transacción cancelada. Este proceso se despierta regularmente para comprobar si su intervención es necesaria.

- **Escritor de Base de Datos (Database Writer (DBWR)).**

El proceso DBWR es el responsable de gestionar el contenido de los buffers de datos y del caché del diccionario. Éste lee los bloques de los archivos de datos y los almacena en la SGA. Luego, escribe en los archivos de datos los bloques cuyo contenido, ha variado. La escritura de los bloques a disco es diferida buscando mejorar la eficiencia de la E/S.

Es el único proceso que puede escribir en la Base de Datos. Esto asegura la integridad. Se encarga de escribir los bloques de datos modificados por las transacciones, tomando la información del buffer de la Base de Datos cuando se valida una transacción. Cada validación, no se lleva a la Base de Datos física de manera inmediata sino que los bloques de la Base de Datos modificados se vuelcan a los archivos de datos periódicamente o cuando sucede algún checkpoint o punto de sincronización.

Los bloques del buffer de la Base de Datos (bloques del segmento de rollback y bloques de datos) menos recientemente utilizados son volcados en el disco continuamente para dejar sitio a los nuevos bloques.

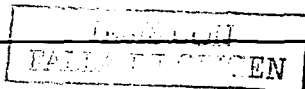
El bloque del segmento de rollback se escribe SIEMPRE antes que el correspondiente bloque de datos.

Múltiples transacciones pueden solapar los cambios en un sólo bloque antes de escribirlo en el disco.

Para que se mantenga la integridad y coherencia de la Base de Datos, todas las operaciones se guardan en los archivos de redo log. El proceso de escritura es asíncrono y pueden realizar grabaciones multibloque para aumentar la velocidad.

- **Escritor de Registro (Log Writer (LGWR)).**

El proceso LGWR es el encargado de escribir los registros redo log en los archivos redo log. Los registros redo log siempre contienen el estado más reciente de la Base de Datos, ya que puede que el DBWR deba esperar para escribir los bloques modificados desde el buffer de datos a los archivos de datos.





Conviene tener en cuenta que el LGWR es el único proceso que escribe en los archivos de redo log y el único que lee directamente los buffers de redo log durante el funcionamiento normal de la Base de Datos.

Coloca la información de los redo log buffers en los archivos de redo log. Los redo log buffers almacenan una copia de las transacciones que se llevan a cabo en la Base de Datos.

Esto se produce:

- A cada validación de transacción, y antes de que se comunique al proceso que todo ha ido bien.
- Cuando se llena el grupo de buffers de redo log
- Cuando el DBWR escribe buffers de datos modificados en disco.

Así, aunque los archivos de Base de Datos no se actualicen en ese instante con los buffers de Base de Datos, la operación queda guardada y se puede reproducir. Oracle no tiene que consumir sus recursos escribiendo el resultado de las modificaciones de los datos en los archivos de datos de manera inmediata. Esto se hace porque los registros de redo log casi siempre tendrán un tamaño menor que los bloques afectados por las modificaciones de una transacción, y por lo tanto el tiempo que emplea en guardarlos es menor que el que emplearía en almacenar los bloques sucios resultado de una transacción; que ya serán trasladados a los archivos por el DBWR. El LGWR es un proceso único, para asegurar la integridad. Es asíncrono. Además permite las grabaciones multibloque.

- Punto de Inspección (Checkpoint (Chkpt)).

Este proceso escribe en los archivos de control los checkpoints. Estos puntos de sincronización son referencias al estado coherente de todos los archivos de la Base de Datos en un instante determinado, en un punto de sincronización. Esto significa que los bloques sucios de la Base de Datos se vuelcan a los archivos de Base de Datos, asegurándose de que todos los bloques de datos modificados desde el último checkpoint se escriben realmente en los archivos de datos y no sólo en los archivos redo log; y que los archivos de redo log también almacenan los registros de redo log hasta este instante. La secuencia de puntos de control se almacena en los archivos de datos, redo log y archivos de control. Los checkpoints se produce cuando:

- Un espacio de tabla se pone inactivo, offline.
- Se llena el archivo de redo log activo.
- Se para la Base de Datos.
- El número de bloques escritos en el redo log desde el último checkpoint alcanza el límite definido en el parámetro LOG_CHECKPOINT_INTERVAL.
- Cuando transcurra el número de segundos indicado por el parámetro LOG_CHECKPOINT_TIMEOUT desde el último checkpoint.

Este proceso está activo si el parámetro CHECKPOINT_PROCESS tiene un valor verdadero.

- Archivador (Archiver (Arch)).

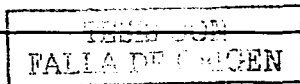
El proceso archivador tiene que ver con los archivos redo log. Por defecto, estos archivos se reutilizan de manera cíclica de modo que se van perdiendo los registros redo log que tienen una cierta antigüedad. Cuando la Base de Datos se ejecuta en modo ARCHIVELOG, antes de reutilizar un archivo redo log realiza una copia del mismo. De esta manera se mantiene una copia de todos los registros redo log por si fueran necesarios para una recuperación.

- Recuperación (Recovery (Reco)).

El proceso de recuperación está asociado al servidor distribuido. En un servidor distribuido los datos se encuentran repartidos en varias localizaciones físicas, y estas se han de mantener sincronizadas. Cuando una transacción distribuida se lleva a cabo puede que problemas en la red de comunicación haga que una de las localizaciones no aplique las modificaciones debidas. Esta transacción dudosa, debe ser resuelta de algún modo, y esa es la tarea del proceso recuperador. Está activo si el parámetro DISTRIBUTED_TRANSACTIONS tiene un valor distinto de 0.

II.2.2 ARCHIVOS

Los archivos de la base de datos guardan información tanto de los datos almacenados en la base de datos como la necesaria para gobernar la propia base de datos. Existen distintos tipos, los cuales se mencionan a continuación:





- Archivos de Datos (Datafiles).

En estos archivos, reside la información de la Base de Datos. Sólo son modificados por el DBWR. A ellos se vuelcan los bloques sucios de la SGA cuando se hace una validación o cuando sucede un checkpoint. Las validaciones de las transacciones no producen un volcado inmediato, sino lo que se conoce por un commit diferido. Toda actualización se guarda en los archivos de redo log, y se lleva a la Base de Datos física cuando tenemos una buena cantidad de bloques que justifiquen una operación de E/S. Almacenan los segmentos (datos, índices, rollback) de la Base de Datos. Están divididos en bloques, cada uno de los cuales se corresponde con un buffer del buffer cache de la SGA. En el bloque de cabecera no se guardan datos de usuario, sino la marca de tiempo del último checkpoint realizado sobre el archivo.

- Archivos de Registro (Redo Log).

En este tipo de archivos, se graba toda operación que se efectúe en la Base de Datos y sirven de salvaguarda de la misma. Tiene que haber por lo menos 2, uno de ellos debe estar activo, online, y se escribe en ellos de forma cíclica. Existe la posibilidad de almacenar los distintos archivos de redo log en el tiempo mediante el modo ARCHIVER. Así, se puede guardar toda la evolución de la Base de Datos desde un punto dado del tiempo. Una opción es la utilización de archivos redo log multiplexados:

- Permite al LGWR escribir simultáneamente la misma información en múltiples archivos redo log.
- Se utiliza para protegerse contra fallos en el disco.
- Da una alta disponibilidad a los archivos redo log activos u online.

Esto se hace definiendo el número de grupos y de miembros de archivos redo log que van a funcionar en paralelo:

Grupos: Funcionan como archivos redo log normales, uno de ellos está activo y el resto espera su turno.

Miembros: Cada escritura de un registro redo log se lleva a cabo en todos los miembros del grupo activo en ese momento.

Cuando se produce algún fallo en los archivos de redo log o en el proceso LGWR:

- Si la escritura en un archivo redo log falla pero el LGWR puede escribir al menos en uno de los miembros del grupo, lo hace, ignorando el archivo inaccesible y registrando un fallo en un archivo de trace.

- Si el siguiente grupo no ha sido archivado (modo ARCHIVELOG) antes del cambio de grupo que lo pone activo, ORACLE espera hasta que se produzca el archivado.

Si fallan todos los miembros de un grupo mientras el LGWR trata de escribir, la instancia se para y necesita recuperación al arrancar.

- Archivos de Control (Control Files).

Mantienen la información física y ruta de todos los archivos que forman la Base de Datos; así como el estado actual de la misma. Son utilizados para mantener la consistencia interna y guiar las operaciones de recuperación. Son imprescindibles para que la Base de Datos se pueda arrancar.

Contienen:

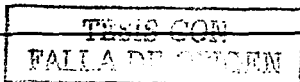
- Información de arranque y parada de la Base de Datos.
- Nombres de los archivos de Datos y Redo log.
- Información sobre los checkpoints.
- Fecha de creación y nombre de la Base de Datos.
- Estado online y offline de los archivos.

Debe haber múltiples copias en distintos discos, mínimo dos, para protegerlos de los fallos de disco. La lista de los archivos de control se encuentra en el parámetro CONTROL_FILES, que debe modificarse con la Base de Datos suspendida.

Hasta aquí, los tipos de archivos que se suelen considerar fundamentales en la arquitectura del SGBD Oracle. Pero existen otros archivos, que aunque no forman parte de la arquitectura Oracle resultan importantes en el uso del SGBD.

- ARCHIVO INIT.ORA

Como parte de la distribución de software, Oracle provee un archivo de parámetros de inicialización llamado init.ora. Este archivo contiene los parámetros del sistema Oracle y debe ser utilizado por el DBA para configurar el SGBD y adecuarlo a una



determinada explotación. Oracle, lee este archivo durante el proceso de arranque para determinar el tamaño de la SGA y encontrar los archivos de control, entre otros menesteres. Como el archivo `init.ora`, es fundamental para el arranque de la Base de Datos, debería ser copiado frecuentemente para protegerlo de posibles pérdidas.

- **ARCHIVOS DE TRACE**

Oracle, crea archivos de texto llamados de trace para ayudar en la diagnóstico de problemas y en el ajuste del SGBD. Cada proceso del servidor, escribe en un archivo de trace asociado cuando es necesario. Los procesos de usuarios también pueden tener asociados archivos de trace. La situación de estos archivos de trace del sistema se especifica por el parámetro `BACKGROUND_DUMP_DEST`, y los de usuario por `USER_DUMP_DEST`. Oracle crea archivos de trace automáticamente cuando ocurre algún error.

Un parámetro muy frecuentemente utilizado por los desarrolladores Oracle es el `SQL_TRACE`, que cuando está puesto a `TRUE` produce que toda sentencia SQL ejecutada genere información en los archivos de trace.

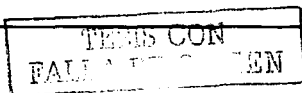
II.3 OBJETOS DE ORACLE

II.3.1 TABLAS

Una tabla, es un objeto de la base de datos que almacena los mismos. Esta formada por varias columnas, cada una de estas columnas tiene un tipo de dato asociado. La información sobre cada tabla se almacena en el diccionario de datos. Con esta información, Oracle puede mantener los datos que residen en las tablas y garantiza que los datos introducidos en cada una de las columnas sean del tipo de dato que se definió en la creación de la tabla.

Los tipos de datos que se pueden usar para la definición de una tabla se muestran en el siguiente esquema.

| TIPO DE DATO | DESCRIPCION |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------|
| <code>VARCHAR2(size)</code> | No se desperdicia memoria, ya que la longitud del campo es del tamaño de datos insertados, el mínimo es 1 el máximo 2000 |
| <code>CHAR(size)</code> | No es flexible y el tamaño del campo está definido por su longitud, mínimo 1 y máximo 255. |



| | |
|----------------|------------------------------------------------------------------------------------------------------------------|
| NUMBER(size) | Es de tipo numérico con una precisión de 38. |
| DATE | Este dato es de tipo fecha y se encuentra dentro de un rango de Enero 1, 4712 B.C. y Diciembre 31, 4712 A.D. |
| LONG | Su límite de longitud es de 2 gigabytes, útil para imágenes y sólo se permite una columna de tipo LONG por tabla |
| RAW y LONG RAW | Utilizado para datos binarios |
| BLOB * | Datos Binarios arriba de 4 Gigabytes |
| BFILE * | Almacenamiento de datos binarios en un archivo externo; arriba de 4 Gigabytes |

II.3.2 VISTAS

Una vista, permite ver una selección personalizada de una o una colección de tablas. A diferencia con una tabla, una vista no contiene ningún dato únicamente contiene una consulta SQL; los datos que se recuperan de esa consulta se presentan, en forma de tabla. En realidad, si uno no ha creado la vista, podría pensar que está trabajando con una tabla, al igual que una tabla se puede insertar, actualizar, borrar y seleccionar datos. Siempre se pueden seleccionar los datos de una vista aunque en algunas situaciones existen restricciones en lo que se refiere a la manipulación de esos datos.

Las vistas pueden proporcionar un nivel adicional de seguridad, por ejemplo se puede tener una tabla que contenga los empleados de una empresa, y al utilizar una vista sobre esta tabla se define que los jefes de departamento, sólo puedan ver la información de sus subordinados.

Las vistas, permiten ocultar complejidad en los datos, ya que se puede recuperar información de la base de datos a partir de dos o más tablas haciendo una asociación. Por ejemplo, se puede disponer de una vista que sea una combinación de una tabla que se llame clientes y de otra tabla identificada como pedidos, de este modo el usuario de la base de datos sólo debería tener que hacer una selección de la vista llamada cliente_pedido, y lo importante es que no necesitaría saber que realmente esté estructurada en dos tablas.

TELEFON
FALLA DE ORIGEN

II.3.3 SINONIMOS

Un sinónimo, es un objeto de la base de datos que permite crear nombres alternativos para las vistas y tablas de Oracle. Se puede decidir establecer sinónimos para una tabla Oracle, por múltiples razones:

- Para ocultar el verdadero creador o nombre de una tabla.
- Por que se quiere ocultar la verdadera localización de una tabla.
- Para proporcionar a los usuarios nombres de tablas menos complicados que los nombre reales.

En otras palabras un sinónimo, es un nombre alternativo de un objeto de la base de datos. Piensese en un sinónimo, como en un apodo para un objeto.

Existen dos tipos de sinónimos:

- Sinónimos Privados. Este tipo de sinónimo es creado cuando un usuario otorga privilegios a otro usuario y sólo este usuario puede usar dicho sinónimo.
- Sinónimos Públicos. Es aquel que todos los usuarios de Oracle pueden usar

II.3.4 INDICES

De la misma forma que el índice de un libro ayuda a buscar rápidamente la información, un índice colocado en una tabla ayuda a recuperar los datos más rápidamente, ya que es una minicopia de una tabla. Las entradas de índice a una tabla permiten a Oracle un acceso rápido a los datos contenidos en las tablas.

Las entradas de índice contienen información sólo sobre las columnas que forman parte del índice, no sobre todas las columnas de la tabla.

Se pueden crear dos tipos de índices: Índices Únicos e Índices No Únicos. Un índice único no permite la duplicación; un índice no único permite la duplicación.

Oracle, permite crear índices concatenados, que se forman con más de una columna.



II.3.5 SECUENCIAS

Una secuencia, es un objeto de base de datos utilizado para generar números secuenciales; es creada por un usuario y puede ser compartida a multiples usuarios. Uno de los usos típicos de las secuencias es crear el valor de la llave primaria, el cual tiene que ser unico para cada uno de los registros. Las definiciones de las secuencias tambien se almacenan en el diccionario de datos. Las secuencias proporcionan una lista consecutiva de números exclusivos que sirven para simplificar las tareas de programación. La primera vez que se llama a una secuencia en una consulta, devuelve un valor, predeterminado. En cada consulta, subsiguiente a la secuencia se obtendrá un valor aumentado en el incremento específico. Las secuencias pueden ser cíclicas, o pueden seguir creciendo hasta alcanzar un valor máximo especificado.

Las secuencias ayudan a reducir la necesidad de las E/S de disco almacenando los números en memoria cache, y poniéndolos a disposición de los sistemas que requieran una clave primaria numerada secuencialmente.

II.3.6 ROLES

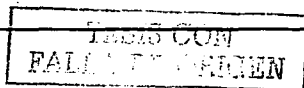
Un rol, es un conjunto de privilegios (Select, Insert, Update, Delete, Create, etc.) que se agrupan y conceden a los usuarios. Una vez que los privilegios se han concedido a un rol, un usuario hereda los privilegios del rol, siendo un miembro del mismo. De esta forma, en lugar de actualizar la cuenta de cada usuario individualmente, basta con gestionar el rol.

II.4 SQL y SQL PLUS

II.4.1 QUE ES SQL y SQL PLUS?

SQL (Lenguaje de Consultas Estructurado . Structured Query Language)

Es el lenguaje que permite la comunicación con el sistema de gestión de base de datos desde cualquier herramienta o aplicación. SQL, es un lenguaje unificado, lo utilizan todo tipo de usuarios desde el administrador de la base de datos (DBA), hasta





el usuario final. Es un lenguaje no procesal, ya que el usuario especifica que quiere, no como ni dónde conseguirlo; además, es relacionamente completo debido a que permite la realización de cualquier consulta de datos.

SQL Plus

Es una herramienta de Oracle que reconoce y envía las sentencias SQL al servidor de Oracle para su ejecución.

TABLA DE COMPARACION DE SQL Y SQL PLUS.

| SQL | SQL PLUS |
|----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|
| Es un lenguaje para comunicarse con el servidor de Oracle y acceder a los datos. | Reconoce sentencias SQL y las envía al servidor. |
| Esta basado en el standard SQL de ANSI | Oracle es el propietario de la interface |
| Manipula datos y definiciones de tablas de la Base de Datos. | No permite la manipulación de valores en la Base de Datos. |
| Las sentencias son introducidas en un buffer. Una o más líneas. | No se almacenan en ningún buffer. Son introducidas una línea a la vez. |
| No tiene un carácter de continuación. | Tiene un carácter de continuación (-) por si la línea de comando es más larga de una línea. |
| Las sentencias no pueden ser abreviadas | Los comandos pueden ser abreviados. |
| Utiliza un carácter de terminación para ejecutar los comandos inmediatamente. | No requiere caracteres de terminación |
| Usa funciones para realizar algún formato. | Usa comandos para formatear datos. |

II.4.2 SENTENCIAS SQL

Las sentencias SQL pertenecen a tres categorías principales: Lenguaje de Definición de Datos (DDL Data Definiton Language), Lenguaje de Manipulación de Datos (DML Data Manipulation Language) y Lenguaje de Control de datos (DCL Data Control Language). Estos tres lenguajes no son similares en sí mismos, sino que es una forma de clasificar las sentencias de lenguaje SQL en función de su cometido. La diferencia principal reside en que el DCL refleja sus efectos en el diccionario de datos y que el DDL además de reflejar dichos cambios, crea objetos en la base de datos; por otro lado el DML es el que permite consultar, insertar, modificar y eliminar la información contenida en los objetos de la base de datos.

Cuando se ejecutan las sentencias DDL de SQL, el Sistema de Gestión de Base de Datos confirma la transacción actual antes y después de cada una de las sentencias DDL. En cambio, las sentencias DML no llevan implícito el commit y se pueden deshacer. Existe pues un problema al mezclar sentencias DML con DDL, ya que estas últimas, pueden confirmar las primeras de manera involuntaria e implícita, lo que en ocasiones puede ser un problema.

SENTENCIAS SQL

| | |
|-----------------------------------------------|-------------------------------------------|
| SELECT | Extracción de Datos |
| INSERT UPDATE DELETE | Lenguaje de Manipulación de Datos (DML) |
| CREATE ALTER DROP RENAME TRUNCATE | Lenguaje de Definición de Datos (DDL) |
| COMMIT ROLLBACK SAVEPOINT | Control de Transacciones |
| GRANT REVOKE | Lenguaje de Control de Datos (DCL) |

II.4.2.1 LENGUAJE DE DEFINICION DE DATOS (DDL).

El DDL permite especificar un esquema de base de datos por medio de un conjunto de definiciones. El resultado de la compilación de instrucciones DDL es un conjunto de objetos de base de datos que se almacenan en el diccionario de datos. Este diccionario de datos es consultado cuando se leen o modifican los datos de la base propia. Entre sus principales tareas se mencionan las siguientes: La creación de un objeto de base de datos así como la modificación y eliminación de este objeto

A continuación se muestra una lista parcial de sentencias DDL.

| SENTENCIA DDL | OBJETIVO |
|-----------------|---------------------------------------------------------------------------|
| Alter Procedure | Recompilar un procedimiento almacenado |
| Alter Table | Añadir o redefinir una columna, modificar la asignación de almacenamiento |
| Analyze | Recoger estadísticas de rendimiento sobre los objetos de |
| Create Table | Crear una Tabla |
| Create Index | Crear Índice |
| Drop Table | Eliminar una Tabla |
| Drop Index | Eliminar un Índice |

II.4.2.2 LENGUAJE DE MANIPULACION DE DATOS (DML).

El lenguaje de manipulación de datos permite insertar, actualizar, borrar y seleccionar los datos de la base de datos. Como su nombre implica, el DML permite trabajar con los contenidos de la base de datos.

| SENTENCIA DML | OBJETIVO |
|---------------|--------------------------------------|
| Insert | Añadir filas de datos a una tabla |
| Delete | Eliminar filas de datos de una tabla |
| Update | Modificar los datos de una tabla |

II.4.2.3 LENGUAJE DE CONTROL DE DATOS (DCL).

Con estas sentencias de control se obtiene la seguridad entre los usuarios que acceden a la base de datos, consiguiendo que sólo ejecuten las tareas que se les tenga permitidas sin afectar la integridad de los objetos así como datos.

| SENTENCIA DCL | OBJETIVO |
|---------------|---------------------------------------------|
| Grant | Conceder privilegios a un usuario o un rol. |
| Revoke | Retirar privilegios a un usuario o un rol. |

Existen otras sentencias que son denominadas de control de transacciones, estas sentencias garantizan la integridad de la base de datos las cuales son:

| SENTENCIA | OBJETIVO |
|-----------|-----------------------------------------------------------------------------|
| Commit | Confirmar como permanentes las modificaciones realizadas |
| Rollback | Deshacer todas las modificaciones realizadas desde la última confirmación. |
| Savepoint | Punto de salvamento al cuál se regresa en caso de deshacer una modificación |

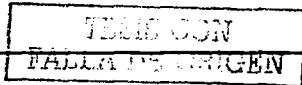
II.4.3 SELECCION, RESTRICCION Y ORDENAMIENTO DE DATOS

La recuperación de los datos en el lenguaje SQL se realiza mediante la sentencia SELECT, seleccionar. Esta sentencia permite indicar al Sistema de Gestión de Base de Datos la información que se quiere recuperar. Esta es la sentencia SQL, más habitual. La sentencia SELECT consta de cuatro partes básicas:

- La cláusula SELECT seguida de la descripción de lo que se desea ver, los nombres de las columnas a seleccionar.
- Esta parte es obligatoria.
- La cláusula FROM seguida de la especificación de las tablas de las que se han de obtener los datos. Esta parte es obligatoria.
- La cláusula WHERE seguida por un criterio de selección, una condición. Esta parte es opcional.
- La cláusula ORDER BY seguida por el criterio de ordenación. Esta parte es opcional.

Un ejemplo de la sintaxis de la sentencia select, se muestra a continuación:

```
SELECT (columna)
FROM (tabla)
[WHERE condición]
[ORDER BY (expresión Columna [ASC | DESC])];
```





II.4.3.1 CLAUSULA FROM

La cláusula FROM, define las tablas de las que se van a seleccionar las columnas. Se puede añadir al nombre de las tablas el usuario propietario de las mismas de la forma usuario.tabla. De esta manera podemos distinguir entre las tablas de un usuario y otro. Oracle siempre considera como prefijo el nombre del propietario de las tablas, aunque no se lo indiquemos. De esta forma dos o más usuarios pueden tener tablas que se llamen igual sin que surjan conflictos operacionales. Si quisiéramos acceder a las filas de la tabla dep del usuario jperez, (ademas de tener privilegios de lectura sobre esa tabla) deberíamos escribir la siguiente sentencia SQL:

```
SQL> select * from jperez.dep;
```

También, se puede asociar un alias a las tablas para abreviar los nombres de las tablas. Un ejemplo se puede ver en la sentencia SQL siguiente:

```
SQL> select d.nombre from dep d;
```

II.4.3.2 CLAUSULA WHERE

En la clausula Where es donde se debe proponer la condición que han de cumplir todas las filas para salir en el resultado de la consulta. La complejidad del criterio de búsqueda es prácticamente ilimitada y en el se pueden conjugar operadores de diversos tipos con funciones de columnas, componiendo funciones más o menos complejas.

Operadores de Comparación

| OPERADOR | OPERACION | EJEMPLO |
|------------|---------------------------|-----------------------------------------|
| = | Igualdad | Select * from emp where cod_dep = 100; |
| !=, <>, ^= | Desigualdad | Select * from emp where cod_dep != 100; |
| < | Menor que | Select * from emp where cod_dep < 200; |
| > | Mayor que | Select * from emp where cod_dep > 200; |
| <= | Menor o Igual | Select * from emp where cod_dep <= 200; |
| >= | Mayor o Igual | Select * from emp where cod_dep >= 200; |
| In | Igual a cualquiera de los | Select * from emp |

| | | |
|---------------|----------------------------------------------------------------------------------------------|----------------------------------------------------------|
| | miembros entre paréntesis | where cod_dep in (200, 300); |
| Not in | Distinto a cualquiera de los miembros entre paréntesis | Select * from emp where cod_dep not in (200, 300); |
| Between | Contenido en el rango | Select * from emp where cod_dep between 100 and 199; |
| Not Between | Fuera del rango | Select * from emp where cod_dep not between 100 and 199; |
| like ' _abc%' | Contiene la cadena abc a partir del segundo carácter y luego cualquier cadena de caracteres. | Select * from emp where nombre like 'Ma%'; |

Operadores Aritméticos

| OPERADOR | OPERACION | EJEMPLO |
|----------|-----------|-----------------------------------------------------------------------|
| + | Suma | Select nombre, salario + comision from emp where oficio = 'VENDEDOR'; |
| - | Resta | Select * from emp where cod_dep != 100; |
| * | Producto | Select * from emp where cod_dep < 200; |
| / | División | Select * from emp where cod_dep > 200; |

Operadores Lógicos

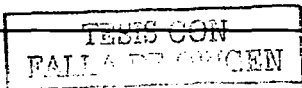
| OPERADOR | OPERACION | EJEMPLO |
|----------|-----------|------------------------------------------------------------------------------|
| AND | Y | Select nombre oficio from emp where salario >= 100 and oficio = 'VENDEDOR'; |
| OR | O | Select nombre oficio from emp where salario >= 100 or oficio = 'VENDEDOR'; |

Operadores de Cadenas de Carácterés

| OPERADOR | OPERACION | EJEMPLO |
|----------|---------------|---------------------------------|
| | Concatenación | Select nombre oficio from emp; |

II.4.3.3 CLAUSULA ORDER BY

Se utiliza para especificar el criterio de ordenación de la respuesta de la consulta. Por defecto, la ordenación es ascendente, aunque se puede especificar un orden descendente. La ordenación se puede establecer sobre el contenido de columnas





o sobre expresiones con columnas. A continuación se puede ver un ejemplo de uso de la cláusula order by en la que se desea obtener un listado de los empleados ordenado de manera descendente por su salario y en caso de igualdad de salario, ordenado ascendentemente por su nombre.

```
SQL> Select nombre,salario from emp order by salario desc, nombre;
```

II.4.3.4 CLAUSULA DISTINCT

Quando se realiza una consulta sobre una tabla en la que se extrae información de varias columnas, puede ocurrir que, si no incluimos la/s columna/s que forman la clave principal, obtengamos filas repetidas en la respuesta.

Si este comportamiento no nos resulta satisfactorio podemos utilizar la cláusula DISTINCT para eliminar las filas duplicadas obtenidas como respuesta a una consulta.

```
SQL> Select distinct oficio from emp;
```

II.4.4 FUNCIONES

Existen en SQL muchas funciones que pueden complementar el manejo de los datos en las consultas. Se utilizan dentro de las expresiones y actúan con los valores de las columnas, variables o constantes.

Se pueden incluir en las cláusulas Select, Where y Order By.

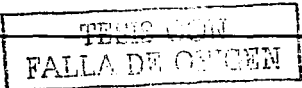
- FUNCIONES NUMERICAS

| FUNCION | COMETIDO | EJEMPLO | RESULTA DO |
|-------------|-------------------------------------------------------------|----------------------------------|---------------|
| ABS (n) | Calcula el valor absoluto de n | Select abs (-15) from dual; | 15 |
| CEIL (n) | Calcula el valor entero inmediatamente sup o igual a n | Select Ceil (15.7) from dual; | 16 |
| FLOOR (n) | Calcula el valor entero inmediatamente inferior o igual a n | Select floor (15.7) from dual; | 15 |

| | | | |
|---------------|--------------------------------------------------------------------------------------------------------------|----------------------------------------|-------|
| MOD(m,n) | Calcula el resto resultante de dividir m entre n | Select mod (11,4) from dual; | 3 |
| POWER(m,n) | Calcula la potencia n-esima de m | Select power (3,2) from dual; | 9 |
| ROUND(m,n) | Calcula el redondeo de m a n decimales. Si n < 0 el redondeo se efectua a por la izquierda del punto decimal | Select round (123.456,1) from dual; | 123.5 |
| SQRT (n) | Calcula la raiz cuadrada de n | Select sqrt (4) from dual; | 2 |
| TRUNC (m,n) | Calcula m truncado a n decimales (n puede ser negativo) | Select trunc (123.456, 1) from dual; | 123.4 |
| SIGN (n) | Calcula el signo de n, devolviendo -1 si n < 0, 0 si n = 0 y 1 si n > 0. | Select sign (-12) from dual; | -1 |

• FUNCIONES DE CARACTER

| FUNCION | COMETIDO | EJEMPLO | RESULTADO |
|------------------------|-----------------------------------------------------------------------------------------------|---------------------------------------------------------------|--------------------|
| CHR (n) | Devuelve el carácter cuyo valor codificado es n. | Select chr (65) from dual; | A |
| ASCII (cad) | Devuelve el valor ascii de cad | Select ascii ('A') from dual; | 65 |
| CONCAT (cad1,cad2) | Devuelve cad1 concatenada con cad2. Esta función es equivalente al operador | Select concat (concat (nombre , 'es '), oficio) from emp; | Cano es Presidente |
| LOWER (cad) | Devuelve la cadena cad con todas sus letras convertidas en minúsculas | Select lower ('MinUsCuLas') from dual; | minúsculas |
| UPPER (cad) | Devuelve la cadena cad con todas sus letras convertidas a mayúsculas. | Select upper ('maYusCuLas') from dual; | MAYUSCULAS |
| INITCAP (cad) | Devuelve cad con el primer carácter en mayúsculas. | Select initcap ('isabel') from dual; | isabel |
| LPAD (cad1, n, cad2) | Devuelve cad1 con longitud de n y ajustada a la derecha rellenando por la izquierda con cad2. | Select lpad ('P', 5, '*') from dual; | ****P |
| RPAD (cad1, n, cad2) | Devuelve cad1 con longitud n, y ajustada a la izquierda rellenando por la derecha con cad2. | Select rpad('P',5,'*') from dual; | P**** |
| REPLACE | Devuelve cad en la que | Select | diego |



| | | | |
|--------------------------|-----------------------------------------------------------------------------------|--------------------------------------------|----|
| (cad, ant, nue) | cada ocurrencia de la cadena ant ha sido sustituida por la cadena nue | replace ('digo', 'i', 'ie') from dual; | |
| SUBSTR (cad, m , n) | Devuelve la subcadena de cad compuesta por n caracteres a partir de la posición m | Select substr('ABCDEFG',3 ,2) from dual; | CD |
| LENGTH (cad) | Devuelve la longitud de cad | Select length ('cadena') from dual; | 6 |

• FUNCIONES DE MANEJO DE FECHAS

| FUNCION | COMETIDO | EJEMPLO | RESULTAD O |
|---------------------------------|--------------------------------------------------------------------------|-----------------------------------------------------------|---------------|
| SYSDATE | Devuelve la fecha y hora del sistema | Select sysdate from dual; | 14-MAR-97 |
| ADD_MONTHS (d, n) | Devuelve la fecha d incrementada en n meses | Select add_months (sysdate, 4) from dual; | 14-JUL-97 |
| LAST_DAY (d) | Devuelve la fecha del ultimo dia del mes d. | Select last_day (sysdate) from dual; | 31-MAR-97 |
| MONTHS BETWEEN (d1, d2) | Devuelve la diferencia en meses entre las fechas d1 y d2. | Select months_between (sysdate, '01-JAN-97') from dual; | 2.43409424 |
| NEXT_DAY (d , cad) | Devuelve la fecha del primer dia de la semana cad despues de la fecha d. | Select next_day (sysdate, 'sunday') from dual; | 16-MAR-97 |

• FUNCIONES DE CONVERSION

| FUNCION | COMETIDO | EJEMPLO | RESULTAD. O |
|--------------------------------|------------------------------------------------------------------------------------------------|------------------------------------------|----------------|
| TO_NUMBE R (cad, fmto) | Convierte la cadena cad a un número, opcionalmente de acuerdo con el formato fmto. | Select to_number ('12345') from dual; | 12345 |
| TO_CHAR (d, fmto) | Convierte la fecha d a una cadena de caracteres opcionalmente con el formato fmto. | Select to_char (sysdate) from dual; | '14-MAR-97' |
| TO_DATE (cad, fmto) | Convierte la cadena cad de tipo varchar2 a fecha opcionalmente de acuerdo con el formato fmto. | Select to_date ('1-JAN-97') from dual; | 01-JAN-97 |



Con las fechas pueden utilizarse varios formatos. Estos, permiten modificar la presentación de una fecha. En la siguiente tabla se presentan algunos formatos de fecha y el resultado que generan.

- Máscaras de Formato Numéricas

| FORMATO | COMETIDO | EJEMPLO | RESULTA DO |
|--------------------|-----------------------------------------------------|----------------------------------------------------|---------------|
| cc ó scc | Valor del siglo | Select to_char (sysdate, cc) from dual; | 20 |
| y,yyy ó sy,yyy | Año con coma, con o sin signo | Select to_char (sysdate, 'y,yyy') from dual; | 1,997 |
| Yyy ó yyy ó yy ó y | Año sin signo con cuatro, tres, dos o un dígito | Select to_char(sysdate, 'yyyy') from dual; | 1997 |
| q | Trimestre | Select to_char (sysdate , 'q') from dual; | 1 |
| ww ó w | Número de la semana del año o del mes | Select to_char (sysdate , 'ww') from dual; | 11 |
| mm | Número del mes | Select to_char (sysdate , 'mm') from dual; | 03 |
| Ddd ó dd ó d | Número del día del año del mes o de la semana | Select to_char (sysdate , 'ddd') from dual; | 073 |
| hh ó hh12 ó hh24 | La hora en formato de 12 o 24 h. | Select to_char (sysdate , 'hh') from dual; | 12 |
| mi | Los minutos de la hora | Select to_char (sysdate , 'mi') from dual; | 15 |
| ss ó ssss | Los segundos dentro del minuto o desde las 0 horas. | Select to_char (sysdate , 'sssss') from dual; | 44159 |

- Máscaras de Formato de Carácter

| FORMATO | COMETIDO | EJEMPLO | RESULTA DO |
|-------------|---------------|------------------------------------|---------------------|
| year ó year | Año en Inglés | Select to_char (sysdate, 'year') | nineteen ninety- |



| | | | |
|-------------|--------------------------------------------------------------|--------------------------------------------------------------|----------------|
| Month ó mon | Nombre del mes o su abreviatura de 3 letras | from dual; Select to_char (sysdate, 'month') from dual; | seven march |
| day ó dy | Nombre del día de la semana o su abreviatura de tres letras. | Select to_char (sysdate, 'day') from dual; | friday |
| a.m. ó p.m. | El espacio del día | Select to_char (sysdate , 'a.m.') from dual; | p.m. |
| b.c. ó a.d. | Indicador del año respecto al del nacimiento de Cristo | Select to_char (sysdate , 'b.c.') from dual; | a.d. |

- FUNCIONES GENERALES

| FORMATO | COMETIDO | EJEMPLO | RESULTADO |
|-----------------------------------------------------|--------------------------------------------------------------|-------------------------------------------------------------------------------|------------------|
| DECODE (var, val1, cod1, val2, cod2, ... defecto) | Convierte el valor de var de acuerdo con la codificación. | Select decode (oficio, 'Presidente', 'P', 'Director', 'D', 'X') from emp | P,D,X |
| GREATEST (exp1, exp2, ...) | Devuelve el mayor valor de una lista | | |
| LEAST (cad, frnto) | Devuelve el menor valor de una lista | | |
| NVL (val, exp) | Devuelve la expresión exp si val es null y val en otro caso. | Select salario + nvl (comision , 0) from dual; | 450000 350000 |

- FUNCIONES DE GRUPO

| FUNCION | COMETIDO | EJEMPLO |
|----------------|----------------------------------------------------------------|-------------------------------------------------------------|
| COUNT (col) | Cuenta el número de filas agrupadas | Select count(nombre), oficio from emp group by oficio; |
| AVG (col) | Calcula el valor medio de todos los valores de la columna col | Select avg (salario), oficio from emp group by oficio; |
| MAX (col) | Calcula el valor máximo de todos los valores de la columna col | Select max (salario), oficio from emp group by oficio; |
| MIN (col) | Calcula el valor mínimo de todos los valores de la columna col | Select min (salario), oficio from emp group by oficio; |
| SUM (col) | Calcula la suma de los valores de la columna col | Select sum (salario), oficio from emp group by oficio; |
| STDDEV (col) | Calcula la desviación | Select stddev (salario), oficio |

| | | |
|---------------------|---------------------------------------------------------------------------------------------|----------------------------------------------------------------------|
| | típica de los valores de la columna col sin tener en cuenta los valores nulos. | from emp group by oficio; |
| VARIANCE (col) | Calcula la varianza de los valores de la columna col sin tener en cuenta los valores nulos. | Select variance (salario) , oficio from emp group by oficio; |

II.4.5 OPERACIONES DE CONJUNTOS

El resultado de cada consulta es un conjunto de filas. Y, con conjuntos, se pueden realizar tres operaciones típicas: la unión, intersección y la diferencia.

UNION (UNION)

Combina todas las filas del primer conjunto con todas, las del segundo. Cualquier fila duplicada se reducirá a una.

INTERSECCIÓN (INTERSECT)

Examinará las filas de los conjuntos de entrada y devolverá aquellas que aparezcan en ambos. Todas las filas duplicadas serán eliminadas antes de la generación del conjunto resultante.

DIFERENCIA (MINUS)

Devuelve aquellas filas que están en el primer conjunto pero no en el segundo. Las filas duplicadas del primer conjunto se reducirán a una fila única antes de empezar la comparación con el segundo campo.

Algunas de las reglas para el manejo de los operadores de conjuntos son las siguientes:

- Pueden ser encadenados en cualquier combinación, siendo evaluados de izquierda a derecha.
- No existe jerarquía de precedencia en el uso de estos operadores, pero puede ser forzada mediante paréntesis.
- Pueden ser empleados con conjuntos de diferentes tablas siempre que se apliquen las siguientes reglas:
 - Las columnas son relacionadas en orden, de izquierda a derecha.

- Los nombres de las columnas son irrelevantes.
- Los tipos de datos deben coincidir.

II.4.6 JOINS

II.4.6.1 QUE ES UN JOIN ?

En Bases de Datos Relacionales, una operación de join consiste en la búsqueda de registros en dos tablas de una base de datos. Las dos tablas deben de tener por lo menos un campo en común; esto es, el campo por medio del cual se realizara el join debe ser miembro de las dos tablas, por lo general son las columnas de llave, primaria y foránea.

Algunas recomendaciones que deben tomarse en cuenta para la creación de joins, son las siguientes:

- Cuando se escriben sentencias `SELECT` para unir dos tablas (join), es necesario anteponer el nombre de la tabla al nombre de la columna que se desea extraer, esto por claridad y por un acceso fácil y rápido a la Base de Datos.
- Si el mismo nombre de una columna esta presente en dos ó mas tablas, debe anteponerse el nombre de la tabla, al nombre de la columna, esto con el fin de identificar correctamente la procedencia de los datos.
- Para unir n número de tablas, es necesario al menos n-1 condiciones de join.

Cuando una condición de Join es invalida u omitida, el resultado es un producto cartesiano en el cual todas las combinaciones de todos los registros son desplegados. Todos los registros de la primer tabla son unidos con todos los registros de la segunda tabla.

Un producto cartesiano tiende a generar un número excesivo de registros, los cuales raramente pueden ser utilizados; para evitar esto, es necesario incluir una condición de join en la cláusula `WHERE`.

II.4.6.2 TIPOS DE JOINS

Existen 2 tipos principales de Joins: EquiJoin y Non-EquiJoins.

- EQUIJOIN

Consideremos las siguientes tablas: EMP y DEPT

| EMP | | | DEPT | | |
|------------------|--------|--------|------------------|------------|----------|
| EMPNO | ENAME | DEPTNO | DEPTNO | DNAME | LOC |
| 7839 | KING | 10 | 10 | ACCOUNTING | NEW YORK |
| 7698 | BLAKE | 30 | 30 | SALES | CHICAGO |
| 7782 | CLARK | 10 | 10 | ACCOUNTING | NEW YORK |
| 7566 | JONES | 20 | 20 | RESEARCH | DALLAS |
| 7654 | MARTIN | 30 | 30 | SALES | CHICAGO |
| 7499 | ALLEN | 30 | 30 | SALES | CHICAGO |
| 7844 | TURNER | 30 | 30 | SALES | CHICAGO |
| 7900 | JAMES | 30 | 30 | SALES | CHICAGO |
| 7521 | WARD | 30 | 30 | SALES | CHICAGO |
| 7902 | FORD | 20 | 20 | RESEARCH | DALLAS |
| 7369 | SMITH | 20 | 20 | RESEARCH | DALLAS |
| ... | | | ... | | |
| 14 rows selected | | | 14 rows selected | | |

↑ Llave Primaria
↑ Llave Foránea

Figura II.4.6.2a Equijoin

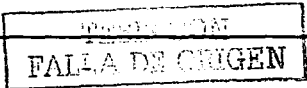
Para determinar el nombre de departamento al que pertenece cada empleado, se tiene que comparar el valor de la columna DEPTNO de la tabla EMP, con el valor de la columna DEPTNO de la tabla DEPT. La relación que existe entre la tabla EMP y DEPT es un equijoin. Los valores en la columna DEPTNO en las dos tablas tienen que ser iguales.

La extracción de registros, utilizando un equijoin es como sigue:

```
SQL> SELECT emp.empno, emp.ename, emp.deptno,
        dept.deptno, dept.loc
FROM emp, dept
WHERE emp.deptno = dept.deptno;
```

| EMPNO | ENAME | DEPTNO | DEPTNO | LOC |
|------------------|-------|--------|--------|----------|
| 7839 | KING | 10 | 10 | NEW YORK |
| 7698 | BLAKE | 30 | 30 | CHICAGO |
| 7782 | CLARK | 10 | 10 | NEW YORK |
| 7566 | JONES | 20 | 20 | DALLAS |
| ... | | | | |
| 14 rows selected | | | | |

Figura II.4.6.2b Equijoin





La cláusula **SELECT** especifica el nombre de las columnas a extraer:

- Nombre del Empleado, Número del Empleado, Número de Departamento, los cuales son columnas pertenecientes a la tabla **EMP**
- Número de Departamento, Nombre del Departamento y Localización; los cuales son columnas de la tabla **DEPT**.

La cláusula **FROM** especifica las dos tablas a las cuales se va a acceder.

La cláusula **WHERE** especifica como las tablas serán unidas. Esto es por medio de la columna **DEPTNO** ya que es común en las dos tablas.

- NON-EQUIJOIN

Consideremos las siguientes tablas: **EMP** y **SALGRADE**

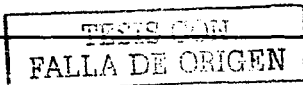
| EMP | | | SALGRADE | | |
|------------------|--------|------|----------|-------|-------|
| EMPNO | ENAME | SAL | GRADE | LOSAL | HISAL |
| 7839 | KING | 5000 | 1 | 700 | 1200 |
| 7698 | BLAKE | 2850 | 2 | 1201 | 1400 |
| 7782 | CLARK | 2450 | 3 | 1401 | 2000 |
| 7566 | JONES | 2975 | 4 | 2001 | 3000 |
| 7654 | MARTIN | 1250 | 5 | 3001 | 9999 |
| 7499 | ALLEN | 1600 | | | |
| 7844 | TURNER | 1500 | | | |
| 7900 | JAMES | 950 | | | |
| ... | | | | | |
| 14 rows selected | | | | | |

Figura II.4.6.2c Non-Equijoin

La relación entre la tabla **EMP** y la tabla **SALGRADE** es un non-equijoin, lo cual significa que ninguna columna en la tabla **EMP** corresponde directamente a una columna en la tabla de **SALGRADE**. La relación existente entre las dos tablas es que la columna **SAL** en la tabla **EMP** se encuentra entre la columna **LOSAL** y **HISAL** de la tabla **SALGRADE**.

La extracción de registros utilizando un non-equijoin es como sigue:

```
SQL> SELECT e.ename, e.sal, s.grade  
FROM emp e, salgrade s
```



WHERE e.sal BETWEEN s.losal AND s.hisal;

| <u>ENAME</u> | <u>SAL</u> | <u>GRADE</u> |
|------------------|------------|--------------|
| JAMES | 950 | 1 |
| SMITH | 800 | 1 |
| ADAMS | 1100 | 1 |
| 14 rows selected | | |

Figura II.4.6.2d Non - Equijoin

El ejemplo crea un non-equijoin para evaluar el grado de salario de los empleados. El salario, tiene que estar entre los rangos menor y mayor del salario.

II.4.7 SUBQUERIES

A veces se han de utilizar en un consulta los resultados de otra consulta, llamada subconsulta. Un ejemplo de esto ocurre cuando queremos conocer los nombres de los empleados cuyo salario está por encima de la media:

```
SQL> SELECT nombre FROM emp
WHERE salario > ( SELECT avg ( salario ) FROM emp );
```

La consulta más interna calcula el salario medio, y la consulta más externa lo utiliza para seleccionar los nombres que ganan más de la media.

El valor de comparación puede ser un valor simple, como en el ejemplo anterior, o un conjunto de valores. Hay que tener en cuenta este detalle ya que el tipo de operador a utilizar varía. En el primer caso se puede utilizar un operador de comparación de carácter aritmético (<, >, etc.). Y en el segundo uno de tipo lógico (IN).

Las subconsultas pueden devolver más de una columna, y se habrán de comparar de manera consecutiva:

- Las columnas de la clausula WHERE de la consulta principal deben estar agrupadas por parentesis.
- Las columnas encerradas entre paréntesis deben coincidir en número y tipo de datos con los datos que devuelve la subconsulta.

El nivel de anidamiento de subconsultas es ilimitado.



II.4.8 RESTRICCIONES (CONSTRAINTS)

Las restricciones o constraints de los datos, se imponen para asegurarnos que los datos cumplen con una serie de condiciones predefinidas para cada tabla. Estas restricciones ayudan a conseguir la integridad referencial: todas las referencias dentro de una Base de Datos son válidas y todas las restricciones se han cumplido.

Las restricciones se van a definir acompañadas por un nombre, lo que permitirá activarlas o desactivarlas según sea el caso; o también mezcladas en la definiciones de las columnas de la tabla.

II.4.8.1 TIPOS DE RESTRICCIONES

NOT NULL (No Nulos)

Establece la obligatoriedad de que esta columna tenga un valor no nulo. Se debe especificar junto a la columna a la que afecta. Los valores nulos no ocupan espacio, y son distintos a 0 y al espacio en blanco. Hay que tener cuidado con los valores nulos en las operaciones, ya que 1 * NULL es igual a NULL.

UNIQUE (Unica)

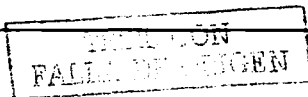
Evita valores repetidos en una columna, admitiendo valores nulos. Oracle crea un índice automáticamente cuando se habilita esta restricción y lo borra al deshabilitarse.

CHECK (Verifica)

Comprueba que se cumpla una condición determinada al rellenar esa columna. Esta condición sólo debe estar construida con columnas de esta misma tabla.

PRIMARY KEY (Llave Primaria)

Establece el conjunto de columnas que forman la llave primaria de esa tabla. Se comporta como única y obligatoria sin necesidad de explicitarlo. Sólo puede existir una llave primaria por tabla. Puede ser referenciada como llave foránea, por otras tablas. Crea un índice automáticamente cuando se habilita o se crea esta restricción.



FOREIGN KEY (Llave Foránea)

Establece que el contenido de esta columna será uno de los valores contenidos en una columna de otra tabla maestra. Esta columna marcada como llave foránea puede ser NULL. No hay límite en el número de llaves foráneas.

La llave foránea puede ser otra columna de la misma tabla. Se puede forzar que cuando una fila de la tabla maestra sea borrada, todas las filas de la tabla detalle cuya llave foránea coincida con la clave borrada se borren también. Esto se consigue añadiendo el comentario de ON DELETE CASCADE en la definición de la llave foránea.

II.4.9 CONTROL DE ACCESO A USUARIOS

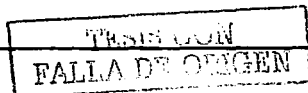
En un ambiente de múltiples usuarios, es necesario mantener la seguridad de acceso y uso de la base de datos. Una base de datos Oracle otorga los siguientes beneficios para mantener la seguridad:

- Control de Acceso a la Base de Datos
- Brindar acceso específico únicamente a ciertos objetos de la base de datos
- Confirmación sobre privilegios otorgados y recibidos por medio del diccionario de datos.
- Creación de sinónimos para los objetos de la base de datos

La seguridad de una base de datos Oracle puede ser clasificada en 2 categorías: seguridad del sistema y seguridad de los datos. La seguridad del sistema, se refiere al acceso y uso de la base de datos a nivel de sistema, esto es, otorgando un user y un password al usuario, un espacio determinado donde estará alojado el usuario, y operaciones que tendrá permitidas el usuario.

II.4.9.1 CUENTAS DE USUARIO

El administrador de la base de datos (DBA), es el encargado de crear usuarios ejecutando la sentencia CREATE USER. En este momento el usuario no posee ningún





privilegio, pero el DBA, puede ya otorgar n número de privilegios al usuario, los cuales determinaran lo que el usuario puede realizar a nivel base de datos.

Un ejemplo de la sintáxis para la creación de un usuario es como sigue:

```
SQL> CREATE USER user  
IDENTIFIED BY password;
```

II.4.9.2 PRIVILEGIOS DEL SISTEMA

Los privilegios son los derechos que se otorgan a un usuario de la base de datos para ejecutar alguna sentencia SQL en particular. El administrador de la base de datos, es el usuario de más alto nivel que posee la habilidad de otorgar acceso a la base de datos a los usuarios y a sus objetos.

Un usuario requiere privilegios del sistema para obtener el acceso a la base de datos y privilegios sobre objetos para poder manipular, estos en la base de datos. Un usuario, es posible que obtenga el privilegio para que él pueda otorgar privilegios a otros usuarios o roles.

Más de 80 privilegios de sistema están disponibles para usuarios y roles y estos principalmente son otorgados por el administrador de la base de datos.

Un ejemplo de la sintaxis para otorgar privilegios de sistema a un usuario es como sigue:

```
SQL> GRANT privilege system  
TO user;
```

Algunos privilegios de sistema son los siguientes:

| PRIVILEGIOS DEL SISTEMA | OPERACIONES AUTORIZADAS |
|-------------------------|-----------------------------------------------------|
| CREATE USER | Otorga el privilegio de creación de usuarios |
| DROP USER | Elimina usuarios |
| DROP ANY TABLE | Elimina tablas |
| BACKUP ANY TABLE | Copia de una tabla con la utilidad export de Oracle |
| CREATE SESSION | Permite la conexión a la Base de Datos |

| | |
|------------------|---------------------------------------------------------------------------------------|
| CREATE TABLE | Crea tablas en el esquema propio de algún usuario |
| CREATE SEQUENCE | Crea secuencias en el esquema propio de un usuario |
| CREATE VIEW | Crea vistas en el esquema propio de algún usuario |
| CREATE PROCEDURE | Crea procedimientos almacenados, funciones o paquetes en el esquema de algún usuario. |

Figura II.4.9.3a Privilegios de Sistema

II.4.9.3. PRIVILEGIOS DE OBJETOS

Un privilegio de objeto, es un derecho para desempeñar una acción en particular sobre una tabla, vista, secuencia o procedimiento específico. Cada uno de los objetos, tiene su particular conjunto de privilegios que pueden ser otorgados sobre él. A continuación se muestra una tabla que contiene los privilegios que pueden ser otorgados a algunos objetos.

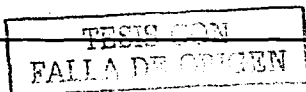
| PRIVILEGIO DE OBJETO | Tabla | Vista | Secuencia | Procedimiento |
|----------------------|-------|-------|-----------|---------------|
| ALTER | X | | X | |
| DELETE | X | X | | |
| EXECUTE | | | | X |
| INDEX | X | | | |
| INSERT | X | X | | |
| REFERENCES | X | | | |
| SELECT | X | X | X | |
| UPDATE | X | X | | |

Figura II.4.9.3a Privilegios de Objetos

Un ejemplo de la sintaxis para otorgar privilegios de objetos a un usuario, es como sigue:

```
SQL> GRANT object privilege
      ON object
      TO user;
```

II.4.9.4 AUDITORIA





La Base de Datos, tiene la capacidad de auditar todas las acciones que tienen lugar dentro de ella. Los registros de auditoría se escriben en una tabla denominada SYS.AUD\$, en el diccionario de datos o en el seguimiento de auditoría (audit trail) del sistema operativo.

Es posible auditar tres tipos distintos de acciones: los intentos de conexión, los accesos a los objetos y las acciones de la base de datos. Durante la realización de las auditorías, el funcionamiento por omisión de la base de datos consiste en registrar tanto las órdenes que han tenido éxito, como las que no lo han tenido; este comportamiento puede modificarse al establecer cada uno de los tipos de auditoría.

Para permitir la auditoría de una base de datos, el archivo INIT.ORA debe contener una entrada para el parámetro AUDIT_TRAIL..

• AUDITORIAS DE ACCESO

Pueden auditar todos los intentos de conexión a la base de datos. La orden para empezar a auditar los intentos de iniciar una sesión es:

```
AUDIT SESSION;
```

Para auditar sólo los intentos de conexión que tengan éxito o fracasen, es necesario utilizar las órdenes siguientes:

```
AUDIT SESSION WHENEVER SUCCESSFUL;
```

```
AUDIT SESSION WHENEVER NOT SUCCESSFUL;
```

Si los registros de auditoría se almacenan en la tabla SYS.AUD\$ podrán examinarse mediante las vistas DBA_AUDIT_SESSION o DBA_AUDIT_CONNECT.

• AUDITORIAS DE ACCION

Es posible auditar cualquier acción que afecte a un objeto de la base de datos (tabla, espacio de tablas, sinónimo, etc). Las acciones que pueden afectar a esos objetos (CREATE, ALTER y DROP) pueden agruparse durante la auditoría.



Cada acción que puede auditarse tiene asignado un código numérico dentro de la base de datos. La vista `AUDIT_ACTIONS` permite acceder a estos códigos.

Una vez conocido el código de acción, puede utilizarse la vista `DBA_AUDIT_OBJECT` para determinar la forma en que la acción ha afectado a un objeto.

- **AUDITORIAS DE OBJETOS**

Además de las acciones de nivel sistema sobre los objetos, también, se pueden auditar las acciones de manejo de datos dirigidas a los objetos. Entre estas acciones, pueden incluirse la auditoría de las operaciones `SELECT`, `INSERT`, `UPDATE` y `DELETE` sobre las tablas. Las acciones de este tipo se auditan de forma muy similar a las auditorías de acción; la única diferencia entre ellas es la adición de una nueva cláusula en la orden `AUDIT`.

La cláusula adicional para las auditorías de objetos es `BY SESSION` o `BY ACCESS`, que especifica si un registro de auditoría debe escribirse una vez para cada sesión (`BY SESSION`) o una vez para cada acceso a un objeto (`BY ACCESS`).

Por ejemplo, si un usuario ejecutara 4 sentencias `UPDATE` distintas sobre la misma tabla, la auditoría `BY ACCESS` daría como resultado la escritura de 4 registros de auditoría (Uno por cada acceso a la tabla). Por otra parte, al auditar la misma situación `BY SESSION` sólo escribiría un registro de auditoría.

Algunos ejemplos de sintaxis de las auditorías de objetos son los siguientes:

`AUDIT INSERT ON user.table;` (Audita todas las operaciones `INSERT`).
`AUDIT ALL ON user.table;` (Audita todas las órdenes que afecten la tabla especificada).
`AUDIT DELETE ON user.table BY SESSION;` (Audita todas las operaciones `DELETE`).

TESIS CON
FALLA DE ORIGEN



CAPITULO III

MIGRANDO BASES DE DATOS ORACLE

TESIS CON
FALSACTE EUGEN



MIGRANDO BASES DE DATOS ORACLE

III.1 CONCEPTO DE MIGRACION

El concepto de migración, es muy usual dentro de los sistemas que se encuentran en desarrollo y producción en el ámbito de la industria y la tecnología de la computación, refleja ventajas en el nuevo uso de otros sistemas o el uso de los mismos pero, con mejoras. Dentro de las Bases de Datos, existe software que es utilizado para el manejo de los datos tales como ORACLE, él cual sufre actualizaciones debido a las tendencias y requerimientos de la tecnología, lo que exige a la mayoría de los sistemas existentes una migración hacia estas nuevas versiones las cuales ofrecen ventajas mejorando muchas veces la funcionalidad.

La migración de una Base de Datos, es el proceso de transformar una versión del manejador de Base de Datos, en alguna otra versión más actualizada. Por ejemplo, transformar una Base de Datos Oracle 7 en una Base de Datos Oracle 8i.

III.1.1 PASOS DEL PROCESO DE MIGRACIÓN.

Es muy importante comprender cada uno de los pasos de migración antes de proceder a migrar una Base de Datos en algún ambiente productivo. Estos pasos, son aplicables en todos los sistemas operativos, con algunas excepciones posibles en ciertas plataformas específicas.

Una correcta planeación y el uso de herramientas que provee Oracle8i pueden facilitar en gran medida, el proceso de migración. A continuación, se mencionan cada uno de los pasos requeridos para realizar la migración de una Base de Datos:

- 1.- Preparar la Migración
- 2.- Pruebas del proceso de migración
- 3.- Pruebas sobre la Base de Datos Migrada
- 4.- Preparar y preservar recursos de la Base de Datos origen.
- 5.- Migración de la Base de Datos Productiva.
- 6.- Afinación y ajustes de las nuevas características de la nueva versión de Oracle.



Cada uno de estos pasos, incluyen tareas muy específicas a realizar y de suma importancia durante la migración, la descripción de estos pasos se hace con el fin de familiarizar de manera general el proceso de migración.

1.- Preparar la Migración

- La decisión del método de migración que será utilizado, basándose sobre consideraciones involucradas del producto de Base de Datos correspondiente, los objetivos de la migración, y las capacidades y habilidades del dominio de las metodologías de migración.
- Desarrollar un plan de pruebas de migración de la base de datos y producto de Oracle con versión 8i.
- Preparar una estrategia de respaldos así como verificar el tiempo de restauración de Base de Datos y consistencia del sistema.
- Estimar y prevenir los recursos del sistema para la migración.

2.- Pruebas del proceso de migración de Oracle

- Ejecutar el proceso de migración sobre la base de datos de pruebas (versión 7), el cual debe ser generado en un ambiente totalmente alterno lo más cercano al de producción y sobre todo no debe tener interferencia con el actual producto de Oracle versión 7.
- Generar las pruebas correspondientes sobre la migración del producto de Oracle de versión 7 a versión 8i.

3.- Pruebas sobre la Base de Datos Migrada.

- Comparar resultados, que no existan anomalías entre la Base de Datos de versión 7 y la Base de Datos que fué migrada a versión 8i.
- Investigar y anotar todas las anomalías encontradas en el proceso de migración de la Base de Datos e implementar las correcciones al proceso de migración final.



- Repetir estos tres pasos tanto como sea necesario hasta que el proceso de migración sea completamente satisfactorio.

4.- Preparar y preservar recursos de la Base de Datos origen.

- Preparar la Base de Datos de producción de manera apropiada para que la migración a Oracle 8i sea exitosa.
- Calendarizar el tiempo fuera de servicio requerido para el respaldo y migración del ambiente productivo para versión 8i.

5.- Migración de la Base de Datos Productiva.

- Aplicar el procedimiento generado durante los pasos uno, dos y tres para migrar la Base de Datos de Oracle versión 7 a versión 8i.
- Después de la migración realizar un respaldo total del producto y la Base de Datos.

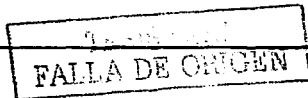
6.- Afinación y ajustes de las nuevas características de la nueva versión de Oracle.

- Afinar la base de datos de acuerdo a las nuevas características que ofrece la versión de Oracle 8i.
- Desarrollar nuevos procedimientos de administración si son necesarios.
- Afinar la migración de los usuarios de base de datos así como sus aplicaciones.

III.1.2 RESPONSABILIDADES DEL ADMINISTRADOR DE LA BASE DE DATOS (DBA) DURANTE LA MIGRACION.

El Administrador de Base de Datos (DBA) usualmente está involucrado en cada uno de los pasos del proceso de migración, excepto en los pasos que involucran pruebas de la aplicación sobre la Base de Datos migrada.

Los puntos específicos del DBA durante la migración son:





- Conocer a cada una de las personas involucradas en el proceso de migración y definir claramente las responsabilidades de ellos durante el proceso de migración.
- Afinar las pruebas de migración.
- Calendarizar las pruebas de migración y el proceso de la misma en el ambiente de producción.
- Generar respaldos de pre-migración de versión 7 en producción.
- Generar un procedimiento de respaldos con la nueva versión de Oracle después de la migración.
- Completar satisfactoriamente el proceso de migración.

Este personaje (DBA), en resumen es el responsable de que el proceso de migración se efectúe satisfactoriamente.

III.2 PREPARANDO LA MIGRACION

III.2.1 ELEGIR METODO DE MIGRACION

Existen diversos métodos que provee Oracle para realizar la migración de una Base de Datos, a continuación se describen cada uno de ellos:

III.2.1.1 UTILERIA DE MIGRACION DE ORACLE (MIGRATION UTILITY)

La utilidad de migración de Oracle (Migration Utility), es una utilidad a nivel línea de comando que convierte estructuras (Archivos de Datos, Archivos de Control, Segmentos de Rollback) y el diccionario de datos de la Base de Datos a ser migrada, en un formato nuevo utilizado por las Bases de Datos Oracle 8i, cambiando únicamente los encabezados de los archivos de datos (datafiles).

Es importante mencionar que la utilidad de migración de Oracle, no cambia el contenido de los datos de los archivos de datos (datafiles), ni tampoco su formato.

Consideraciones del Sistema y Requerimientos

- Requerimientos de Espacio.

Los binarios de Oracle8i requieren al menos 3 veces más de espacio en disco que los de Oracle7. Esto puede ocasionar que durante el proceso de migración se tenga el problema de no tener el espacio suficiente.

La utilería de migración de Oracle requiere relativamente poco espacio temporal. Únicamente necesita el espacio suficiente en el tablespace de system, para alojar el diccionario de datos tanto de Oracle7 como de Oracle8i simultáneamente. El espacio requerido para alojar el diccionario de datos depende de cuantos objetos existan en la Base de Datos; comúnmente el nuevo diccionario de datos de Oracle8i requiere del doble de espacio que el de Oracle7.

Es importante considerar que los programas de conversión requieren también espacio en el tablespace de system y en los segmentos de rollback. La cantidad exacta de espacio requerido por este tipo de programas varía dependiendo del número de objetos de la Base de Datos.

- Consideraciones Tamaño Del Bloque.

El valor del parámetro DB_BLOCK_SIZE en la Base de Datos de Oracle7 y en la Base de Datos migrada Oracle8i tiene que ser el mismo. Oracle8i requiere como mínimo que el tamaño del bloque sea de 2048 bytes (2Kb). Sin embargo, múltiplos de 2Kb como por ejemplo 4Kb, 8KB, 16KB son aceptables y proveen mayor robustez en la operación de la Base de Datos.

- Consideraciones Para Bases De Datos Distribuidas.

Cuando se realiza una migración de Oracle7 en un ambiente de Bases de Datos Distribuidas, es necesario asegurarse que no existan transacciones pendientes en la vista del diccionario de datos DBA_2PC_PENDING antes de migrar la Base de Datos. Ya que, cuando una Base de Datos es abierta usando la sentencia ALTER DATABASE RESET LOGS y alguna transacción esta pendiente, el manejador mostrará errores.

Si existen transacciones pendientes, estas tendrán que ser resueltas antes de realizar la migración haciendo uso de las sentencias COMMIT FORCE O ROLLBACK FORCE.

TRABAJA CON
FALLA DE ORIGEN



Vista Rápida Al Proceso De Migración

En el Ambiente Oracle7

1.- Se ejecuta la utilería de migración de Oracle, la cual crea un nuevo diccionario de datos basado en el diccionario de datos que se tenía en la versión anterior (Oracle7). Además, un nuevo archivo binario basado en el archivo de control de Oracle7 es creado. Este archivo binario, también es conocido como el archivo de conversión.

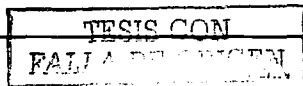
En el Ambiente Oracle8i

1.- Se ejecuta la sentencia ALTER DATABASE CONVERT, la cual crea un nuevo archivo de control basado en el archivo de conversión creado por la utilería de migración de Oracle, convierte todos los encabezados de los archivos de datos que se encuentran en línea al formato de una Base de Datos Oracle8i y termina por montar la Base de Datos Oracle8i.

Los encabezados de los archivos de datos que se encuentran fuera de línea y todos aquellos tablespaces que se encuentran en modo (solo lectura) no son actualizados durante el proceso de migración. Estos encabezados son convertidos posteriormente cuando los archivos de datos son puestos en línea nuevamente y cuando los tablespaces son puestos en modo de lectura - escritura.

2.- Se ejecuta la sentencia ALTER DATABASE OPEN RESETLOGS, la cual automáticamente convierte todos los objetos y usuarios definidos en el nuevo diccionario de datos Oracle8i. Además, en este momento todos los segmentos de Rollback son convertidos, al formato Oracle8i.

3.- Se ejecutan los programas de conversión de Base de Datos los cuales se encuentran en la ruta de \$ORACLE_HOME/rdbms. El primero de ellos es el programa de nombre u0703040.sql. Este programa crea y altera tablas del sistema, borra al usuario MIGRATE y ejecuta los programas catalog.sql y catproc.sql los cuales crean vistas del sistema y todos los paquetes necesarios para el uso de PL/SQL.





VENTAJAS DE LA UTILERIA DE MIGRACION DE ORACLE (MIGRATION UTILITY)

- 1.- El uso de esta utilería es relativamente fácil y rápido, comparado con el uso del método de Export / Import, ya que los objetos del diccionario de datos son únicamente aquellos que son modificados.
- 2.- No existe límite en el tamaño de la Base de Datos que tenga que ser migrada.
- 3.- Requiere relativamente poco espacio adicional en disco comparado con otros métodos de migración.
- 4.- Provee mayor control en el proceso de migración comparado con el Asistente de Migración de Datos de Oracle.

DESVENTAJAS DE LA UTILERIA DE MIGRACION DE ORACLE (MIGRATION UTILITY)

- 1.- Únicamente permite migraciones de Bases de Datos Oracle7 a Bases de Datos Oracle8i.
- 2.- No es posible migrar porciones particulares de alguna Base de Datos. Solamente permite la migración total de la Base de Datos.
- 3.- No permite la migración de Bases de Datos a diferentes plataformas.
- 4.- No permite la actualización de un release a otro. Por ejemplo de Oracle 8.0.5 a Oracle 8.1.5

III.2.1.2 ASISTENTE DE MIGRACION DE DATOS DE ORACLE (DATA MIGRATION ASSISTANT)

El Asistente de Migración de Datos de Oracle (Data Migration Assistant) brinda una interface gráfica amigable, la cual sirve como guía para el Administrador de Bases de Datos (DBA) a través de todo el proceso de migración. Este asistente de migración lo que hace es ejecutar la utilería de migración de Oracle (Migration Utility) en modo oculto, lo cual evita que el DBA tenga que ejecutar los comandos en línea.



Los requerimientos del sistema, la preparación de la Base de Datos Oracle7 y la instalación del software de Oracle8, son iguales, si se utiliza la utilidad de migración de Oracle ó el asistente de migración de datos. Es por ello que nos enfocaremos principalmente al paso de la migración de la Base de Datos.

MIGRANDO LA BASE DE DATOS ORACLE7

- 1.- Iniciar el Asistente de Migración de Datos de Oracle. Esto se realiza ejecutando el comando: `odma`
- 2.- Una vez que la pantalla de bienvenida haya aparecido se deben de seguir las instrucciones que muestran cada una de las pantallas posteriores, brindando la información solicitada; la cual corresponde a indicar correctamente tanto el nuevo Oracle Home y el anterior Oracle Home , asegurarse de que la ruta de algunos archivos como el de inicialización sea la correcta, indicar si el respaldo de la Base de Datos quiere realizarse por medio de esta herramienta, etc.

Al terminar el Asistente de Migración de Datos de Oracle, se procede a realizar lo siguiente:

- 1.- Moverse al siguiente directorio `ORACLE_HOME/rdbms/admin` e iniciar el Server Manager.
- 2.- Conectarse a la Base de Datos como el usuario `INTERNAL`.
- 3.- Iniciar la instancia de la Base de Datos Oracle8i por medio del comando `STARTUP RESTRICT`.
- 4.- Si la Base de Datos tiene la modalidad de replicación instalada, es necesario ejecutar los siguientes scripts:
`catrep.sql`
`r0703040.sql`
- 5.- Dar de Baja la Base de Datos con el comando `SHUTDOWN NORMAL` o `SHUTDOWN IMMEDIATE`.



VENTAJAS DEL ASISTENTE DE MIGRACION DE DATOS (DATA MIGRATION ASSISTANT)

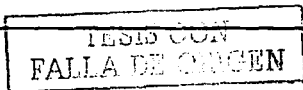
- 1.- Provee una interface gráfica fácil de usar la cual sirve de guía a través de todo el proceso de migración.
- 2.- Requiere una mínima interacción del DBA, comparado con el método de Export/Import o la Copia de Datos.
- 3.- Relativamente rápida, ya que los objetos del diccionario de datos son únicamente los que son alterados.
- 4.- No existe límite en el tamaño de la Base de Datos a ser migrada.
- 5.- Permite la actualización entre releases de una versión. Por ejemplo es posible actualizar una Base de Datos de Oracle 8.0.5 a Oracle 8.1.5.

DESVENTAJAS DEL ASISTENTE DE MIGRACION DE DATOS (DATA MIGRATION ASSISTANT)

- 1.- Provee menos flexibilidad que otros métodos de migración debido a su alto grado de optimización.
- 2.- Provee menos control sobre el proceso de migración.
- 3.- No permite la migración parcial de una Base de Datos.
- 4.- Con la utilización de este método, no es posible realizar la migración de una Base de Datos de una plataforma a otra.
- 5.- Las Bases de Datos con la opción de Servidor en Paralelo instalada, no es posible que sean migradas con este método.

UTILERIA DE MIGRACION VS ASISTENTE DE MIGRACION DE DATOS

- 1.- El asistente de migración de datos de Oracle, provee una interface gráfica la cual sirve como guía a través del proceso de migración, mientras que la utilería de migración de Oracle es una utilería en modo línea de comando. Además, el asistente de migración de datos de Oracle provee gran cantidad de ayuda en línea; lo cual implica que su uso sea más fácil que el utilizar la utilería de migración de Oracle.





2.- El asistente de migración de datos de Oracle es menos flexible que la utilidad de migración de Oracle. Para evitar complejidad, el asistente de migración de datos de Oracle automatiza muchos de los pasos del proceso de migración. En contraste, si es utilizada la utilidad de migración de Oracle muchos de los pasos del proceso de migración tendrán que ser realizados manualmente, lo cual permite que se puedan realizar ajustes durante el proceso de migración, si estos son necesarios. Por supuesto, que el proceso de migración suele tomar mayor tiempo si la utilidad de migración de Oracle es utilizada, ya que muchos pasos se tienen que realizar manualmente y se tienen que tomar muchas más decisiones.

3.- El asistente de migración de datos de Oracle, realiza todos y cada uno de los pasos requeridos en el proceso de migración cada vez que este asistente es ejecutado. Sin embargo, si es necesario salir del asistente por cualquier razón durante el proceso de migración, la base de datos Oracle7 tendrá que ser recuperada del último respaldo disponible que se tenga e iniciar nuevamente desde el principio el proceso de migración. Por el contrario, como en la utilidad de migración de Oracle cada uno de los pasos del proceso de migración están claramente definidos, en el caso de que se tenga que abortar alguno de los pasos, el proceso de migración puede proseguir desde el paso previo sin tener que comenzar desde un principio todo el proceso.

4.- El asistente de migración de datos de Oracle, remueve automáticamente cada uno de los parámetros de inicialización del archivo `init.ora` que son obsoletos para una Base de Datos Oracle8i. En el caso de utilizar la utilidad de migración de Oracle, estos parámetros deben de removerse manualmente ya que la utilidad no altera en lo absoluto el archivo de inicialización.

5.- El asistente de migración de datos de Oracle, no soporta la migración de sistemas con la opción de Servidores en Paralelo (Oracle Parallel Server) instalada; para ello es necesario hacer uso de la utilidad de migración de Oracle.

En general si el DBA prefiere una interface gráfica en vez de líneas de comando y si desea procesos automatizados en los cuales tenga que tomar pocas decisiones, la mejor opción es el asistente de migración de datos de Oracle. Por el contrario, si



prefiere ejecutar líneas de comando en vez de una interface gráfica y sobretodo tener un mayor control durante el proceso de migración, la utilería de migración de Oracle es la mejor opción.

III.2.1.3 EXPORT/IMPORT

Otro de los métodos para realizar una migración de base de datos es utilizando la herramienta de Oracle Export /Import.

EXPORTAR

La utilería export de Oracle lee la base de datos, incluido el diccionario de datos, y escribe el resultado en un archivo binario denominado archivo de volcado para exportación. Esta operación puede realizarse para toda la base de datos, para usuarios específicos o para tablas específicas. Durante las exportaciones, se puede elegir entre exportar o no el diccionario de datos asociado con las tablas, como los privilegios, los índices y las restricciones asociadas a ellos. El archivo que describe la exportación contiene las órdenes necesarias para recrear por completo todos los objetos elegidos.

Las exportaciones de toda la base de datos pueden realizarse para todas las tablas (denominadas exportaciones completas), o sólo para las tablas que han cambiado desde la última exportación.

Existen tres modos de exportar los cuales, se describen a continuación:

- *Exportación en modo tabla*

Cuando se usa la exportación en modo tabla, se le especifica a Oracle los nombres de una o más tablas que se van a exportar. Oracle escribe los datos de la tabla en el archivo de exportación.

Defragmentar un espacio de tablas (tablespaces), puede borrarse y volverse a crear.



- *Exportación en modo base de datos completa.*

Cuando se hace una exportación completa de la base de datos, todas las órdenes de creación de archivos de soporte de la base de datos (archivos de datos, espacios de tablas (tablespace's), segmentos de anulación, etc.) y de datos de los usuarios se escriben en el archivo de exportación para cada usuario de la base de datos, excepto para el usuario SYS . Este archivo de exportación se puede usar para una importación de la base de datos completa.

IMPORTAR

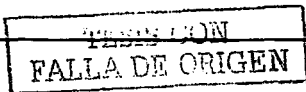
Después de exportados los datos, pueden importarse mediante la utilería Import de Oracle. Esta utilería lee el archivo de volcado para exportación creado por Export y ejecuta la órdenes que hay en él . Por ejemplo, entre estas ordenes puede incluirse una orden CREATE TABLE, seguida de una orden INSERT para cargar datos en la tabla.

Los datos que se han exportado, no tienen que importarse en la misma base de datos, o en el mismo esquema que se haya utilizado para generar el archivo de volcado para exportación. Este archivo puede utilizarse para crear un conjunto duplicado de los objetos exportados en un esquema distinto, o en otra base de datos.

Es posible importar todos los datos exportados o sólo una parte de ellos. Si se importa el archivo de volcado para exportación de una exportación completa, durante la importación se crearán todos los objetos de la base de datos (Incluidos los espacios de tablas, los archivos de datos y los usuarios). No obstante, suele ser útil crear previamente los espacios de tablas y los usuarios para especificar la distribución física de los objetos en la base de datos. Si se van a importar parte de los datos del archivo de volcado para exportación, antes de la importación habrá que configurar los espacios de tablas, los archivos de datos y los usuarios que van a poseer y en los que se van a almacenar esos datos.

Los datos también pueden importarse en una base de datos creada en una versión superior del núcleo de Oracle. Este procedimiento puede utilizarse entre versiones principales consecutivas de Oracle, no se admite la posibilidad inversa.

Los mismos 3 métodos que existen para realizar la exportación de una Base de Datos, pueden ser aplicados para la importación.



El proceso de migración con esta herramienta consiste en completar los siguientes pasos:

- 1.- Generar un Export de la base de datos que va a ser migrada (base de datos origen), al realizar el export de esta base de datos se generan archivos fisicos que contienen la información total o parcial de la base de datos dependiendo de las necesidades de la migración.
- 2.- Se deben de revisar los archivos generados por la herramienta export para validar que no se encuentren dañados.
- 3.- Instalar la nueva versión del producto de Oracle.
- 4.- Crear el ambiente de Oracle con la nueva versión. Este ambiente se convertirá en la base de datos de producción, si este ambiente es creado en el mismo servidor, la base de datos debe de llamarse con el mismo nombre como la base de datos de la versión anterior sin reescribir los archivos de datos.
- 6.- Generar el Import de los archivos de Export dentro del nuevo ambiente de Oracle.
- 7.- Revisar que los mensajes que genera la herramienta de Import no contengan errores.
- 8.- Al finalizar la carga de los datos en el nuevo ambiente es necesario ejecutar las tareas posteriores a la migración las cuales se describen en los siguientes subcapítulos.

Ventajas Del uso de Export/Import

- 1.- Se puede migrar una base de datos de versión Oracle 6 y las versiones de Oracle 7 hasta la versión Oracle8i.
- 2.- Se puede migrar partes específicas de la base de datos.
- 3.- Se pueden defragmentar los archivos de datos (datafiles) de la base de datos y comprimir los datos migrados para generar un mejor funcionamiento.
- 4.- La base de datos puede ser reestructurada en diferentes espacios de tablas (tablespaces) y las tablas pueden ser particionadas.
- 5.- En los casos que sea necesario es posible migrar una base de datos a versiones inferiores por ejemplo de Oracle versión 8i a Oracle versión 7.



Desventajas Del uso de Export/Import.

- 1.- Este método es demasiado lento excepto para bases de datos pequeñas, el tiempo requerido aumenta dependiendo de la cantidad de datos y más si se tienen tipos de datos LONG o si las bases de datos contienen Gigabytes probablemente tome muchas horas.
- 2.- Requiere mucho espacio en disco para realizar la copia de datos en los archivos de Export.
- 3.- Existe una limitante de Oracle al generar los archivos de Export los cuales no deben de ser mayor a 2 Gb, aunque la plataforma del sistema operativo lo permita los mensajes de error son obtenidos al utilizar la herramienta de Import, dando por resultado la incompleta migración de datos.

III.2.1.4 COPIA DE DATOS

Dentro del manejo de datos en la arquitectura de Oracle existe la copia de datos desde una base de datos Oracle a otra base de datos Oracle usando ligas entre estas bases de datos denominados (Database Links), por ejemplo se pueden copiar datos desde una tabla origen de una base de datos a una tabla destino de otra base de datos en diferentes máquinas con el comando COPY de SQL*Plus, o se puede crear nuevas tablas en la base de datos destino y llenar las tablas con datos desde la base de datos origen usando el comando INSERT INTO, o el comando CREATE TABLE... FROM y el comando CREATE TABLE ..AS.

Usando este método de copia de datos se obtiene una gran ventaja que es la defragmentación en los archivos de datos (datafiles) y una reestructuración de la base de datos al crear nuevos espacios de tablas (tablespaces) o modificar tablas o tablespaces existentes, otra ventaja importante dentro de este mecanismo de migración se ve reflejado en crear sólo los objetos o usuarios necesarios de una base de datos específica.

Al utilizar la copia de datos resalta la facilidad de seleccionar los registros específicos de las tablas que serán migradas, este mecanismo es muy eficiente al



querer realizar la depuración de una base de datos ya que se pasa sólo una parte de la tabla de esa base de datos, en contraste si se utiliza Export/Import solo se puede llevar las tablas completas.

Un ejemplo claro de este mecanismo utilizando un caso muy común es crear una tabla (emp_2) que contenga sólo ciertos datos de la tabla origen. En este ejemplo, se utiliza el concepto de database link y se describe el funcionamiento de estos.

```
CREATE TABLE EMP_2
(empno      number(4),
ename      varchar(10),
salario    numberr(15),
depto_num  number(2)
AS SELECT EMPNO, ENAME, SALARIO, DEPTO_NUM
FROM EMPLEADOS@DBLINKV7 WHERE DEPTO_NUM IN (10, 20, 30);
```

En este ejemplo la liga es EMPLEADOS@DBLINKV7 la cual, está formada de dos partes las cuales están divididas por el signo de arroba (@). En la parte izquierda está definido el nombre del objeto que en este caso es una tabla de la base de datos origen y del lado derecho se encuentra la liga que es un objeto que esta creado en la base de datos destino. La definición de este objeto contiene un usuario y password de la base de datos remota además de la definición del servicio de red que utilizara, este es el que se encuentra definido en el archivo de configuración tnsnames.ora el cual contiene el nombre del servidor (host) , base de datos y puerto por el cual tendrán acceso las ligas; de esta manera una base de datos de un servidor puede ejecutar sentencias de SQL*Plus en otra base de datos de otro servidor.



VENTAJAS DE COPIA DE DATOS

- 1.- Los archivos de datos (Datafiles) son defragmentados y los datos son compactados para generar un mejor funcionamiento.
- 2.- La base de datos puede ser reestructurada en diferentes espacios de tablas (Tablespace).
- 3.- Se puede migrar de diferente plataforma de sistema operativo y hardware.
- 4.- Se puede migrar una base de datos de versión Oracle 6, las versiones de Oracle 7, hasta la versión Oracle8i.
- 5.- Se puede migrar partes específicas de la base de datos.

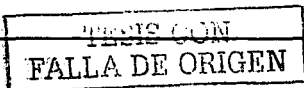
DESVENTAJAS DE COPIA DE DATOS

- 1.- Este método es demasiado lento excepto para bases de datos pequeñas, el tiempo requerido aumenta dependiendo de la cantidad de datos y más si se tienen tipos de datos LONG, si las bases de datos contienen Gigabytes probablemente tome muchas horas,
- 2.- Requiere que las bases de datos tanto la de origen como la destino deben de estar accesibles durante la copia de datos.

III.2.2 REQUERIMIENTOS DEL SISTEMA Y RECURSOS DISPONIBLES

Un adecuado análisis de los requerimientos del sistema y de los recursos disponibles, es un punto muy importante para una migración exitosa de la Base de Datos. Cada uno de los métodos de migración posee sus propios requerimientos, sin embargo es necesario considerar lo siguiente al realizar la evaluación de cada uno de ellos:

- Configuración tanto en el Sistema Operativo como en el hardware.
- Tamaño de la Base de Datos a ser migrada.
- Posibles ajustes en el tamaño de la Base de Datos considerando las nuevas características de Oracle8i.



Es importante considerar que los archivos binarios de Oracle8i requieren al menos 3 veces más de espacio que los archivos binarios utilizados por Oracle7. Este incremento requiere de atención especial ya que en sistemas grandes cientos de ejecutables son generados. El espacio requerido por estos archivos ejecutables depende también de las opciones (Oracle Servidor en Paralelo, SQL Net) que vayan a ser instaladas en el ambiente de Oracle8i.

Además, el diccionario de datos de Oracle8i requiere del doble de espacio que el utilizado por Oracle7 en el tablespace de SYSTEM. Si se tiene planeado el uso de la utilería de migración de Oracle, es posible estimar los requerimientos de espacio para el tablespace de SYSTEM, ejecutando la utilería con la opción de CHECK_ONLY. Oracle8i, también requiere 2 veces más de memoria RAM que Oracle7.

En la figura que a continuación se muestra se observan las diferencias en los requerimientos del sistema entre Oracle8i y Oracle7.

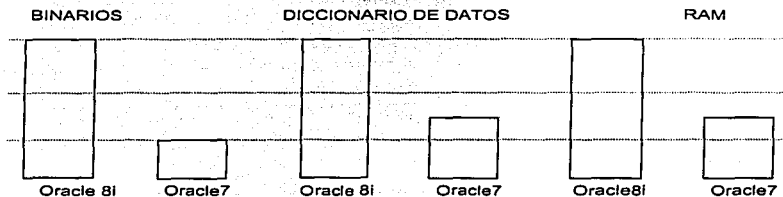


Figura III.2.2a Requerimientos de Espacio

Los requerimientos de memoria del sistema pueden estimarse considerando los siguientes factores:

- El tamaño promedio de procesos de Oracle que incluyen algún cursor abierto y una área de ordenamiento.
- El número pico de usuarios concurrentes conectados al sistema.
- El promedio de memoria de Oracle utilizado por una aplicación front-end.



III.2.3 PROBLEMAS COMUNES DURANTE LA MIGRACION

Cuando se realiza la migración de una base de datos suelen ocurrir errores derivados de muchas causas, sobre todo en los momentos de finalizar la migración; estos errores en la mayoría de los casos ocurren por descuidos en las primeras tareas realizadas durante la migración, las cuales son muy comunes en la etapa de pruebas.

Si la migración de la base de datos se realiza tanto con la Utileria de Migración de Oracle (Migration Utility) o el Asistente de Migración de Datos (Data Migration Assistant) suelen presentarse los siguientes errores:

- Problemas Generales en la Migración.

Los problemas generales en la migración ocurren cuando es ejecutado ya sea Migration Utility o Oracle Data Migration Assistant, pero los problemas en realidad son originados por la configuración de la base de datos. Tanto para el Migration Utility o el Assistant, es necesario que la base de datos posea un buen desempeño antes de realizar la migración. Típicamente cuando ocurre un error tanto en el Migration Utility o el Assistant la migración se detiene y se muestra en pantalla uno o más mensajes de error dependiendo de la causa.

Si se utiliza el Oracle Data Migration Assistant y ocurre un error durante la migración es necesario restaurar la base de datos con el respaldo de la versión anterior antes de volver a ejecutar el asistente.

- Insuficiente Espacio en el Tablespace de System.

Este problema puede regresar el siguiente mensaje:

```
ORA-00604: error occurred at recursive SQL level 1  
ORA-01653: unable to extend table SYS by 473 in tablespace SYSTEM
```

Para este problema es necesario agregarle espacio al tablespace de SYSTEM o el tablespace donde se encuentran las tablas de sistema. En lo que sea posible, se



recomienda que se verifique el espacio que está asignado a los tablespaces principalmente a los temporales y a los de los rollback segments, con esta simple validación de espacio en la base de datos se evitan problemas como el anterior.

- Incorrecta Definición en el Parámetro de AUDIT_TRAIL

Este problema puede regresar el siguiente mensaje:

```
ORA-00604: error occurred at recursive SQL level num
ORA-01552: cannot use system rollback segment for non-system tablespace 'name'
ORA-02002: error while writing to audit trail
```

Este problema se puede presentar por las siguientes dos condiciones:

- El parámetro de inicialización AUDIT_TRAIL está definido como DB o como TRUE.

- La tabla SYS.AUD\$ está creada en el tablespace diferente al de System.

Para corregir este problema es necesario completar los siguientes pasos:

1.- Dar de baja la base de datos si se encuentra abierta.

2.- Colocar el parámetro de AUDIT_TRAIL en el archivo de inicialización con el siguiente valor.

```
AUDIT_TRAIL = NONE
```

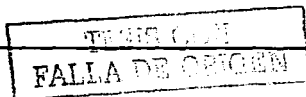
3.- Es necesario volver a ejecutar la Utilería de Migración o restaurar la base de datos de un respaldo si es utilizado el Oracle Data Migration Assistant.

- Validar el parámetro de OPTIMAL de los Segmentos de Rollback de System.

Este problema puede regresar el siguiente mensaje:

```
ORA-01562: failed to extend rollback segment (id = 0)
ORA-01628: max # extents (n) reached for rollback segment SYSTEM
```

Este mensaje indica que los segmentos de rollback de SYSTEM son demasiado pequeños para completar la tarea de migración. Se debe de estar completamente





seguro que los segmentos de rollback son lo suficientemente grandes para realizar la tarea de migración satisfactoriamente.

La solución para este problema consiste en los siguientes pasos:

- 1.- Establecer el valor de nulo en el parámetro OPTIMAL de los segmentos de rollback.
- 2.- Duplicar el parámetro de NEXT EXTENT en los segmentos de rollback de system.
- 3.- Duplicar el valor de MULTIPLIER.
- 4.- Agregar espacio en el tablespace de System.

- El Valor definido de **MULTIPLIER** es pequeño.

Este problema puede regresar el siguiente mensaje:

ORA-01632: max # extents (%s) reached in index %s.%s

Al usar la herramienta de Migration Utility, esta usa por default el valor de 15 para el parámetro opcional de MULTIPLIER y este valor es pequeño, para corregir este problema solo es necesario incrementar este valor opcional.

Mig MULTIPLIER=30

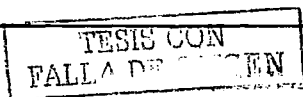
- **Problemas con la instrucción de ALTER DATABASE CONVERT.**

Cuando se utiliza la Utileria de Migración y se ejecuta el comando de ALTER DATABASE CONVERT suelen aparecer mensajes de error en la migración, típicamente la conversión se detiene y aparecen uno o más mensajes de error, estos mensajes se describen a continuación ya que son los más comunes durante la ejecución de este comando.

a) Problemas con la existencia de los control files de Oracle7.

Este problema puede regresar el siguiente mensaje:

ORA-00200: cannot create control file name





ORA-00202: controlfile: name
ORA-27038: skgfrcre: file exists

Para este problema es necesario que sean renombrados o removidos los control files de la versión anterior de Oracle antes de usar el comando de ALTER DATABASE CONVERT, con esto se evita este mensaje de error.

b) Si La Base de Datos es inicializada en otro modo diferente al NOMOUNT.

Este problema puede regresar el siguiente mensaje:

ORA-00227: corrupt block detected in controlfile: (block num, # blocks num)
ORA-00202: control file: '%s'

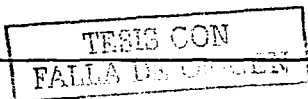
Este mensaje indica que la base de datos fué inicializada en otro modo diferente al NOMOUNT, para este problema es necesario que sean renombrados o removidos los archivos de control de la versión anterior de Oracle antes de usar el comando de ALTER DATABASE CONVERT, además que la base de datos debe de ser inicializada en modo NOMOUNT antes de utilizar esta instrucción.

c) Problemas en la Conversión de Archivos

Este problema puede regresar el siguiente mensaje:

ORA-00404: convert file not found: name
ORA-27037: unable to obtain file status

Cuando aparece este mensaje en las plataformas Unix hace mención al archivo *convsi.dbf*, el cual es generado por la Utilería de Migración, este archivo debe de moverse a la ruta de ORACLE_HOME/dbs en el directorio del ambiente de ORACLE8i, antes de volver a correr el comando de ALTER DATABASE CONVERT se debe verificar que este archivo se encuentre en la ruta mencionada.



d) El nombre de la Base de Datos no es Correcto

Este problema puede regresar el siguiente mensaje:

ORA-01103: database name 'string' in controlfile is not 'string'

Cuando aparece este mensaje es necesario encontrar en dónde el nombre de la base de datos esta incorrecto, sólo existen dos partes en las cuales revisar este error, cada una de ellas se refiere a que el nombre de la base de datos especificado en el archivo de inicialización, en el parámetro DB_NAME no esta igual que el nombre de la base de datos en el archivo de conversión *convsid.dbf*. Otra razón, consiste en que la variable de ambiente ORACLE_SID en la instancia del Oracle8i no esta igual que en el archivo de conversión *convsid.dbf*.

Para corregir este problema sólo se debe de estar seguro que el nombre de base de datos este especificado correctamente en las siguientes tres partes:

- En la variable de ambiente ORACLE_SID.
- Verificar el DB_NAME en el archivo de inicialización de la instancia.
- En la parte del sid del archivo de *convsid.dbf*.

Por ejemplo, si en la variable de ambiente ORACLE_SID o en la parte de DB_NAME del archivo de inicialización esta definido el nombre de la base de datos como DB1, entonces el archivo de conversión (*convsid.dbf* se debe de llamarse de la siguiente manera: *convDB1.dbf*

e) Problemas de integridad de Versión en los Datafiles.

Este problema puede regresar el siguiente mensaje:

ORA-01122: datafile name - failed verification check
ORA-01110: data file name: str
ORA-01211: Oracle7 data file is not from migration to Oracle8i

TESIS CON
FALLA DE VERIFICACION

Este mensaje indica que la base de datos fue abierta con la versión de Oracle 7 después de ejecutar la herramienta de Migración, esto significa que los datafiles han cambiado en el diccionario de datos. Para corregir este problema es necesario dar de baja la base de datos, para posteriormente renombrar los control file creados por la

instrucción de ALTER DATABASE CONVERT con diferentes nombres, para posteriormente restaurar la copia generada de los control files antes de usar la instrucción STARTUP NOMOUNT del proceso de migración que deberá repetirse este proceso de migración. Si no se tiene respaldo de los control files, será necesario restaurar la base de datos del respaldo generado antes de la migración y volver a realizar este proceso.

f) Volver a Ejecutar la instrucción ALTER DATABASE CONVERT.

Este problema puede regresar el siguiente mensaje:

```
ORA-01122: datafile name - failed verification check
ORA-01110: data file name: str
ORA-01202: wrong incarnation of this file - wrong creation time
```

Este error normalmente indica que la sentencia de ALTER DATABASE CONVERT fué usada previamente pero fallo. Si se encuentra con este error se debe de proseguir con el siguiente paso de la migración, el cual es el reset de los logs usando el siguiente comando ALTER DATABASE OPEN RESETLOGS, de cualquier manera si se encuentran problemas posteriores al correr esta instrucción y la base de datos no es migrada es necesario restaurar la base de datos de un respaldo generado antes de dar inicio a la migración y volver a ejecutar el proceso de migración completo. En estos casos se recomienda revisar nuevamente el proceso de migración y encontrar de que manera evitar los errores durante el proceso.

TRABAJE CON
FALLA DE ORIGEN

III.3 ESTRATEGIAS DE RESPALDO

III.3.1 RESPALDO EN CALIENTE (HOT BACKUP)

Las copias de seguridad en caliente, pueden utilizarse con cualquier base de datos Oracle que se esté ejecutando en modo ARCHIVELOG. En este modo, se copian y almacenan los archivos de registro en línea (RedoLog Files), con lo que se crea un registro completo de todas las transacciones llevadas a cabo dentro de la Base de Datos.

Oracle, escribe en los archivos de registro (RedoLog Files) de forma cíclica; después de llenar el primer archivo de registro empieza a escribir en el segundo, hasta que lo llena, después escribe en el tercero y así sucesivamente. Cuando se llena el último, archivo de registro, el proceso Escritor de Registro (Log Writer) empieza a sobrescribir el contenido del primero.

Cuando Oracle se ejecuta en modo ARCHIVELOG, el proceso Archivador (Archiver) realiza una copia de cada uno de los archivos de registro antes de sobrescribirlo, dichos archivos suelen escribirse en un dispositivo de disco, aunque también pueden escribirse directamente en un dispositivo de cinta magnética, lo cual no es recomendable ya que requiere demasiado tiempo.

Es posible realizar copias de seguridad del sistema de archivos de una base de datos mientras se encuentra abierta, siempre y cuando se este ejecutando en modo ARCHIVELOG. Para ello, hay que poner cada uno de los espacios de tablas (Tablespace) en un estado de copia de seguridad, realizar entonces la copia de seguridad de los archivos de datos y restablecer después el espacio de tablas a su estado normal. Por último se realiza una copia física del archivo de control.

La base de datos puede recuperarse por completo a partir de una copia de seguridad de este tipo, y también puede llevarse a cualquier instante en el tiempo, utilizando los archivos de registro que anteriormente han sido archivados. Cuando la base de datos se abra, se habrán restablecido las transacciones confirmadas que hubiera en la base de datos, en dicho instante.

Mientras la base de datos está abierta se realiza la copia de seguridad de los siguientes archivos:



- Archivos de Datos (Data Files)
- Archivos de Registro (RedoLog Files)
- Archivos de Control (Control File). Esta copia se realiza mediante una orden especial.

III.3.2 RESPALDO EN FRÍO (COLD BACKUP)

Las copias de seguridad en frío se producen cuando la base de datos se ha apagado en forma normal (es decir, no por un fallo en la instancia), esto es con el comando shutdown immediate o shutdown normal. Mientras la Base de Datos se encuentra cerrada, se realiza la copia de seguridad de los siguientes archivos:

- Archivos de Datos (Data Files)
- Archivos de Control (Control Files)
- Archivos de Registro (RedoLog Files)
- Archivo INIT.ORA (Opcional)

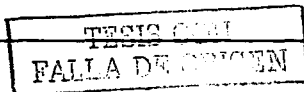
Al realizar la copia de seguridad de estos archivos mientras la base de datos está cerrada, se proporciona una imagen completa e íntegra de la Base de Datos tal como era en el instante de su cierre. El conjunto completo de archivos puede recuperarse en una fecha posterior, a partir de las copias de seguridad y la base de datos podrá funcionar.

Es más fácil realizar la copia de seguridad de los archivos de datos, si la estructura de directorios que están asignados a la base de datos se encuentra organizada de una manera coherente.

III.3.3 DESCARGA Y CARGA DE DATOS

DESCARGA DE DATOS.

La descarga de información de la Base de Datos, consiste en realizar una consulta directa de la información existente en la Base de Datos por medio de una sentencia SELECT, delimitando cada uno de los campos por medio de algún carácter especial ya sea una coma (,), un pipe (|), etc. Estos datos son almacenados en un archivo de texto.





A continuación se muestra un ejemplo de un programa para la descarga de información de una tabla de la Base de Datos.

```
SET PAGESIZE 0
SET LINESIZE 150

SPOOL /my_home/descarga1.txt

SELECT Nombre||'|'||
       Apellido_Pat||'|'||
       Apellido_Mat||'|'||
       Dirección||'|'||
       Teléfono||'|'||
FROM Cliente;
SPOOL OFF
```

CARGA DE DATOS.

SQL Loader es una herramienta de Oracle que permite cargar datos desde archivos planos y colocarlos en la Base de Datos Oracle, de acuerdo con las instrucciones que recibe del archivo de control. El archivo de control le comunica dónde colocar los datos, y le describe los tipos de datos que se van a cargar. Puede filtrar registros (es decir, no cargar registros que no son correctos), cargar datos en múltiples tablas al mismo tiempo, y generar una clave única o manipular los datos antes de introducirlos en una tabla Oracle.

SQL Loader esta compuesto de 5 archivos principales:

- Archivo de Control de Carga

El archivo de control, contiene las especificaciones acerca de los datos que serán cargados, por ejemplo la localización del archivo de descarga de datos, que tablas serán afectadas durante la carga, especificaciones de tipo de columnas y datos, el carácter de separación entre columnas, etc.

- Archivo de Descarga de Datos

El archivo de descarga de datos contiene los datos que serán cargados dentro de la base de datos. Los datos pueden estar en formato de texto.

- Archivo de Salida de Carga

El archivo de salida registra todas las transacciones de carga realizadas por SQL Loader. Por ejemplo, contiene el número de registros que fueron exitosamente cargados, el número de registros que fueron descartados, etc. El archivo de salida únicamente es creado si se especifica en la línea de comando.

- Archivo de Descartar

El archivo de descartar contiene todos aquellos registros que no cumplieron con alguna de las especificaciones del archivo de control. Este, es generado únicamente si se especifica en el archivo de control. Aún, si el archivo de control especifica un archivo de descartar, y todos los registros cumplen con las especificaciones, SQL Loader no genera un archivo de descartar.

- Archivo de Registros Erróneos

El archivo de registro erróneos contiene datos que SQL Loader rechaza. La principal razón por la que los registros son rechazados, es debido a que poseen un tipo de dato erróneo. Oracle automáticamente crea este tipo de archivos cuando existen registros que son rechazados.

III.4 PLAN DE PRUEBAS

Es muy importante poseer un completo plan de pruebas, que permita la validación de cada una de las etapas del proceso de migración. La rigurosa y correcta puesta en marcha de este plan de pruebas, asegura que el proceso de migración de la Base de Datos de producción esté completamente entendido, que es predecible y por supuesto será exitoso.

Debe de considerarse en gran medida la importancia de un plan de pruebas y no hacerse a un lado, ya que de ello depende el éxito de la migración. Es por ello que se recomienda realizar tantas pruebas como sea necesario antes de proceder a migrar la Base de Datos de producción.



III.4.1 PROCESO DE MIGRACIÓN

Crear un ambiente de pruebas alterno que no interfiera con la Base de Datos que se encuentra en el ambiente productivo. Cabe mencionar que el ambiente de pruebas dependerá del método de migración que haya sido elegido.

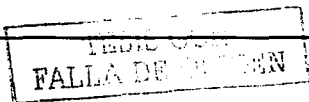
- Si se tiene planeado elegir la utilería de migración de Oracle o el asistente de migración de datos de Oracle, es necesario crear una versión de pruebas de la Base de Datos que será migrada.
- Si se ha elegido el método Export / Import, entonces se tienen que realizar exportaciones e importaciones de pequeñas cantidades de datos

Es importante mencionar que una excelente prueba de migración, consiste en crear un ambiente alterno con una copia idéntica de la Base de Datos que se encuentra en el ambiente productivo.

Las pruebas de migración deben de realizarse tantas veces como sea necesario, y no se debe de migrar el ambiente productivo hasta que no se tengan completamente definidos los detalles en cada uno de los pasos del proceso de migración.

III.4.2 BASE DE DATOS ALTERNA

- Desarrollar el plan de pruebas en la Base de Datos origen (Oracle 7) y en la Base de Datos migrada (Oracle 8i), comparando resultados.
- Realizar pruebas de operación sobre la Base de Datos migrada, es decir, probar las aplicaciones sobre la Base de Datos Oracle 8i; esto con el fin de asegurarse que la Base de Datos funciona correctamente.
- Es posible realizar pruebas de las nuevas funcionalidades que ofrece Oracle 8i, considerando que la adición de alguna de estas características en la Base de Datos no interfiera en el funcionamiento de alguna aplicación.





III.5 TAREAS POSTERIORES A LA MIGRACIÓN

Al terminar satisfactoriamente el proceso de migración es necesario realizar las siguientes tareas posteriores:

Respaldo de la base de datos con la nueva versión de Oracle.

Esta tarea consiste en realizar un respaldo completo de la Base de Datos de producción recién migrada, el cual debe de contener todos los archivos de datos, archivos de registro, archivos de control y archivo de inicialización. Este respaldo puede

ser con la herramienta Export o realizar un respaldo en frío o caliente, y una vez que se tenga el respaldo puede ser usado en cualquier contingencia. Cambiar el password del usuario OUTLN.

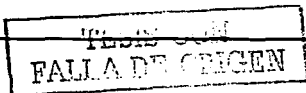
El usuario OUTLN es creado automáticamente durante la instalación de Oracle8i, este usuario tiene los privilegios de DBA por tal motivo se debe de cambiar el password, ya que implica un punto de seguridad sobre la Base de Datos, para evitar este hueco es necesario utilizar la instrucción de ALTER USER sobre el usuario OUTLN, de la siguiente manera dentro del prompt de SQL*PLUS:

```
ALTER USER OUTLN IDENTIFIED BY nuevo_password
```

Checar la consistencia de los constraints de Fecha de Oracle.

Después de la migración es necesario revisar la consistencia de todos los objetos de la Base de Datos, uno de los mas importantes son los constraints referentes al manejo de las fechas, para este problema Oracle proporciona un programa el cual verifica la consistencia de estos objetos, además si encuentra objetos dañados los marca como inválidos para ser posteriormente analizados. Para correr este programa es necesario seguir los siguientes pasos:

1.- Conectarse con el usuario de administración de Oracle y colocarse en la ruta \$ORACLE_HOME/rdbms/admin.





- 2.- Inicializar una sesión de Server Manager con el siguiente comando svrmgrl en Oracle 8i.
- 3.- Conectarse a la instancia de la base de datos de la siguiente manera:
SVRMGR> connect internal
- 4.- Teclar las siguientes instrucciones.

```
SVRMGR> SPOOL utlresult.log  
SVRMGR> @utlconst.sql  
SVRMGR> SPOOL OFF
```

Después de ejecutar el script se utlconst.sql, este genera un archivo de mensajes con el nombre de utlresult.log, el cual contiene todos los constraints inválidos, el programa no corrige los constraints por lo tanto es necesario borrarlos y reconstruirlos.

Reconstruir los Índices Bitmap.

Durante la migración algunos índices de bitmap pueden quedar inválidos, es necesario encontrar cuáles índices resultan ser dañados esto usando la siguiente sentencia:

```
SELECT index_name, index_type, table_owner, status  
FROM dba_indexes  
WHERE index_type = 'BITMAP'  
AND status = 'UNUSABLE';
```

Los índices que son mostrados inválidos deben de ser reconstruidos para lograr la consistencia de todos los objetos en la base de datos.

Cargar los servicios de JAVA para Oracle.

En el momento de instalar la nueva versión de Oracle si se escoge la opción de Java es necesario instalar el programa de Oracle llamado ODCI.jar y CartridgeSevices.jar. Para realizar esta tarea se deben de ejecutar las siguientes instrucciones desde la línea de comandos.

```
loadjava -user sys/PASSWORD -resolve -synonym -grant public  
-verbose ORACLE_HOME/vobs/jlib/ODCI.jar
```

```
loadjava -user sys/PASSWORD -resolve -synonym -grant public  
-verbose ORACLE_HOME/vobs/jlib/CartridgeServices.jar
```

Con estos programas se instalan las clases y se crean los sinónimos, los cuales deben de ser bajo el esquema de SYS.

TESIS CON
FALLA DE ORIGEN

TESIS CON
FALLA DE ORIGEN



CAPITULO IV

MIGRACION DE LA BASE DE DATOS DEL SISTEMA DE PREPAGO

TESIS CON
FALLA DE ORIGEN

122

TESIS CON
FALLA DE ORIGEN



MIGRACION DE LA BASE DE DATOS DEL SISTEMA DE PREPAGO

IV.1 ¿ QUE ES EL SISTEMA DE PREPAGO ?.

El sistema de prepago es una aplicación de tasación en tiempo real que permite a los operadores de redes inalámbricas ofrecer un servicio de telefonía celular prepagada utilizando distintos elementos tales como una Red Celular Movil (Mobile Switching Center MSC) o un Registro de Localización (Home Location Registry HLR). El Sistema de Prepago permite que cada uno de los subscriptores posea una cuenta de débito, la cual le permite realizar y recibir llamadas las cuales serán descontadas de su saldo actual.

Periódicamente, un subscriptor puede ser avisado de que su cuenta esta por terminarse o que el periodo de expiración de esta misma se esta aproximando, este problema puede solucionarse ingresando una nueva ficha del sistema de prepago con el fin de aumentar el saldo de su cuenta.

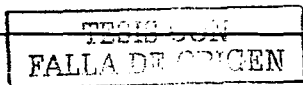
Existen diversos elementos que interactúan dentro de lo que es el Sistema de Prepago, cada uno de ellos se describe a continuación.

RED CELULAR MÓVIL (Mobile Switching Center)

Las redes celulares móviles están compuestas de tres componentes principales, Estos son: Registros de Localización (Home Location Registrys HLR), Radiobases (Radio Base Stations RBS) y Estaciones Móviles. La red, es modular en diseño y puede ser adaptada a varios requerimientos de capacidad añadiendo más MSCs, HLRs y RBSs, junto con canales de radio y equipo de transmisión.

UNIDAD DE RESPUESTA INTERACTIVA (Interactive Voice Response IVR)

La unidad de respuesta interactiva de voz (IVR), es un dispositivo de multitarea/multiaplicación el cual opera en una plataforma Intel x86 usando UNIXWare, con tarjetas de interface a troncales de cada central celular. El IVR, envía mensajes a los subscriptores y recupera información desde el teléfono celular durante la llamada,





como también sobre los procedimientos de administración. Por ejemplo, si un suscriptor requiere saber su saldo, el suscriptor cuestiona al IVR y este a su vez al PPAS, donde esta la Base de Datos, el PPAS le envía la información al IVR y este a su vez envía un mensaje al suscriptor informándole sobre su saldo.

SISTEMA DE ADMINISTRACION DE PREPAGO (Prepaid Administration System PPAS)

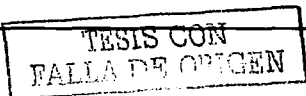
La plataforma PPAS es una plataforma abierta cliente/servidor. El PPAS se conecta vía TCP/IP a los dispositivos de mensajes interactivos (IVR) y con clientes de Windows para proporcionar sesiones de trabajo a usuarios del sistema de prepago. La conexión con la central celular se hace a través de terminales de señalización SS7 conectadas al HLR (Home Location Registry) donde se hace la tasación en tiempo real y todo el control de la llamada.

SISTEMA DE SEÑALIZACIÓN DE CANAL COMÚN (Common Channel Signaling System No. 7)

Es un estándar global de comunicaciones definido por la Unión Internacional de Telecomunicaciones (ITU) y por el Sector de Estandarizaciones de Telecomunicaciones (ITU-T). El estándar define los procedimientos y protocolos por el cual los elementos de la red intercambian información para efectos inalámbricos (Celular) y alámbricos, es decir para llevar a cabo llamadas, enrutamiento y control . El ITU define el SS7 siguiendo las normas nacionales así como los estándares definidos por American National Standard Institute (ANSI) y Bell Communications Research (Bellcore).

PROTOCOLO TCP/IP

TCP/IP son las siglas de Protocolo de Transmisión y Control y Protocolo de Internet (Transmission Control Protocol / Internet Protocol). TCP/IP es un protocolo punto a punto, orientado a conexión e implantado en una amplia gama de enlaces de comunicación que se utiliza para interconectar múltiples redes de





área local (Local Area Network LAN) . Ofrece varias facilidades a los usuarios para controlar la transferencia de datos, la fiabilidad, el control de flujo, las conexiones y la seguridad.

IV.2 CONFIGURACIÓN ACTUAL

El proceso de migración de Base de Datos, implica poseer toda la información necesaria con respecto a la configuración tanto de hardware como de software del servidor; de esta manera es posible tener una correcta planeación y ejecución de dicho proceso en cada una de sus etapas.

IV.2.1 SISTEMA OPERATIVO

El hardware es de suma importancia debido a que de ello depende en gran medida el buen funcionamiento de cualquier aplicación y Base de Datos que este instalada, para lograr esto, es necesario tener una adecuada configuración del sistema operativo acorde a las necesidades, además de tenerlo actualizado independientemente de la plataforma que se este manejando.

Es necesario que el DBA cuente con los conocimientos básicos de sistema operativo, además de que durante todo el proceso de migración trabaje de manera cercana con la persona responsable del servidor, quien proporcionara la configuración y funcionamiento de la plataforma.

Las características de la plataforma de prepago en la cual se realizará la migración de la Base de Datos, es la siguiente:

- Servidor modelo Sun Enterprise 10000.
- 38 Cpu's de 400 Mhz y 8 Mb de Memoria Cache
- 38 Gbytes de Memoria RAM
- 36 System Boards
- 31 Discos Totales de 18 Gb tipo Sun A-5000 de los cuales 23 discos son asignados para Base de Datos.



El espacio total que posee el servidor se encuentra distribuido en 3 principales grupos de file systems que corresponden al sistema operativo, Base de Datos y aplicación.

FILE SYSTEM (SISTEMA OPERATIVO)

| FILE SYSTEM | ESPACIO TOTAL (KBYTES) | ESPACIO ASIGNADO (KBYTES) | ESPACIO LIBRE (KBYTES) | % OCUPACION |
|-------------|--------------------------|-----------------------------|--------------------------|-------------|
| / | 482824 | 192007 | 242535 | 45% |
| /usr | 2055705 | 756891 | 1237143 | 38% |
| /var | 2055705 | 2055705 | 314824 | 85% |
| /opt | 2055705 | 329025 | 1665009 | 17% |
| /home | 3565158 | 2584468 | 924306 | 74% |
| /tmp | 16822248 | 532288 | 16289960 | 4% |

FILE SYSTEM (APLICACION)

| FILE SYSTEM | ESPACIO TOTAL | ESPACIO ASIGNADO | ESPACIO LIBRE | % OCUPACION |
|-------------|---------------|------------------|---------------|-------------|
| /prepay | 2088960 | 1789524 | 287267 | 87% |
| /provision | 7340032 | 1060211 | 5887366 | 16% |
| /cdr | 20971520 | 7359476 | 13612044 | 36% |
| /newnet | 1048576 | 198007 | 797429 | 20% |

FILE SYSTEM (BASE DE DATOS)

| FILE SYSTEM | ESPACIO TOTAL | ESPACIO ASIGNADO | ESPACIO LIBRE | % OCUPACION |
|-------------|---------------|------------------|---------------|-------------|
| /oracle | 2097152 | 1274089 | 771635 | 63% |
| /u01 | 10485760 | 8822116 | 1611720 | 85% |
| /u02 | 10485760 | 10294088 | 185746 | 99% |
| /u03 | 10485760 | 7713786 | 2685416 | 75% |
| /u04 | 10485760 | 10242952 | 235284 | 98% |
| /u05 | 10485760 | 10242696 | 235470 | 98% |

TESIS CON
FALLA DE ORIGEN



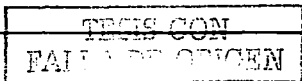
| | | | | |
|-----------|----------|----------|----------|-----|
| /u06 | 10485760 | 10255748 | 222888 | 98% |
| /u07 | 10485760 | 10294088 | 185746 | 99% |
| /u08 | 10485760 | 8462588 | 1960388 | 82% |
| /u09 | 10485760 | 10294184 | 185652 | 99% |
| /u10 | 10485760 | 10242664 | 235502 | 98% |
| /u11 | 10485760 | 10255588 | 222980 | 98% |
| /u12 | 10485760 | 10242920 | 235316 | 98% |
| /u13 | 10485760 | 10294024 | 185810 | 99% |
| /u14 | 10485760 | 9994538 | 475934 | 96% |
| /u15 | 10485760 | 10242632 | 235534 | 98% |
| /u16 | 10485760 | 10242696 | 235470 | 98% |
| /u17 | 10485760 | 6147504 | 4202688 | 60% |
| /tmpcpt | 10485760 | 10242584 | 235582 | 98% |
| /tmpusr | 10485760 | 10242576 | 235590 | 98% |
| /tmpapp | 10485760 | 8194536 | 2219628 | 79% |
| /rollback | 35362816 | 20482720 | 14763912 | 59% |
| /redo | 17681408 | 10666540 | 6905324 | 61% |
| /rpt | 6291456 | 3894104 | 2247576 | 64% |
| /exports | 20971520 | 17212856 | 3701540 | 83% |
| /hotbck1 | 17825792 | 16537752 | 1267928 | 93% |
| /hotbck2 | 17825792 | 14545596 | 3229004 | 82% |
| /hotbck3 | 17825792 | 17793872 | 31480 | 99% |

IV.2.2 BASE DE DATOS

La Base de Datos de este sistema tiene una configuración compleja debido a la magnitud de las transacciones y a la gran cantidad de datos que maneja la aplicación. Es por ello que todos los archivos de datos (datafiles) se encuentran distribuidos en los 22 filesystems con el fin de distribuir la carga en cada uno de los discos y así evitar su contención.

ESPACIO DE BASE DE DATOS.

| ESPACIO TOTAL (KBytes) | ESPACIO ASIGNADO (KBytes) | ESPACIO LIBRE (Kbytes) |
|-----------------------------|--------------------------------|-----------------------------|
| 414000000 | 345596928 | 68403072 |





Los archivos de configuración de la Base de Datos se encuentran en los siguientes directorios.

```
/oracle/ora8/dbs/initppay.ora  
/oracle/ora8/dbs/configppay.ora  
/oracle/ora8/network/admin/listener.ora  
/oracle/ora8/network/admin/tnsnames.ora
```

Estos archivos contienen la definición del manejo de la instancia, los cuales se mencionan a continuación:

1.- El espacio de memoria (SGA) de la Base de Datos se encuentra distribuido de la siguiente manera:

| PARAMETRO | VALOR |
|------------------|---------------|
| Fixed Size | 69,312 |
| Variable Size | 273,780,736 |
| Database Buffers | 1,785,856,000 |
| Redo Buffers | 10,256,384 |
| Total de SGA | 2,069,962,432 |

2.- La Base de Datos cuenta con tres archivos de control los cuales son los siguientes:

```
/u01/oradata/ppay/ctrl1ppay.ctl  
/u06/oradata/ppay/ctrl2ppay.ctl  
/u11/oradata/ppay/ctrl3ppay.ctl
```

3.- El modo en el cual se encuentra la Base de Datos está definido en los siguientes parámetros del archivo de configuración. Esto significa que la Base de Datos esta en modo archive; es decir que, existe un respaldo de cada uno de los archivos log, en la ruta definida.

```
log_archive_start = true  
log_archive_dest = /redo/oradata/archive
```




4.- La Base de Datos cuenta con 20 segmentos de rollback, definidos en el parámetro rollback_segments del archivo de configuración.

rollback_segments=

(r01,r02,r03,r04,r05,r06,r07,r08,r09,r10,r11,r12,r13,r14,r15,r16,r17,r18,r19,r20)

5.- La Base de Datos cuenta con 10 grupos de archivos de Log, cada grupo está compuesto por 1 miembro de 500 Mb distribuidos de la siguiente manera:

| GRUPO | MIEMBRO |
|-------|----------------------------------|
| 1 | /redo/oradata/ppay/log05ppay.dbf |
| 2 | /redo/oradata/ppay/log06ppay.dbf |
| 3 | /redo/oradata/ppay/log07ppay.dbf |
| 4 | /redo/oradata/ppay/log08ppay.dbf |
| 5 | /redo/oradata/ppay/log01ppay.dbf |
| 6 | /redo/oradata/ppay/log02ppay.dbf |
| 7 | /redo/oradata/ppay/log03ppay.dbf |
| 8 | /redo/oradata/ppay/log04ppay.dbf |
| 9 | /redo/oradata/ppay/log09ppay.dbf |
| 10 | /redo/oradata/ppay/log10ppay.dbf |

TABLESPACES QUE COMPONEN LA BASE DE DATOS

La Base de Datos de esta aplicación consta de objetos tales como tablas, secuencias, índices, procedimientos, etc. , estos objetos son alojados en el espacio lógico llamado tablespaces, estos tablespaces están creados por datafiles los cuales son archivos físicos que se encuentran distribuidos en los filesystem.

Además de los objetos de la aplicación, la Base de Datos utiliza objetos propios para el manejo y funcionalidad de los datos, estos objetos son los segmentos de rollback, el diccionario de datos, los archivos de registro (Redo Logs), etc.

A continuación se muestra una lista de los tablespaces que componen la Base de Datos:

TRABAJANDO CON
FALLAS DE ORIGEN



| TABLESPACE | KBYTES | USED | FREE | PCT_USED | MAX_FREE |
|------------|------------|------------|-----------|----------|----------|
| IDATA01 | 4,096,000 | 2,889,632 | 1,206,368 | 70.55 | 990808 |
| IDATA02 | 1,792,000 | 121,216 | 1,670,784 | 6.76 | 1012304 |
| IDATA03 | 3,072,000 | 563,224 | 2,508,776 | 18.33 | 870392 |
| IDATA04 | 2,048,000 | 1,446,488 | 601,512 | 70.63 | 368616 |
| IDATA05 | 3,072,000 | 1,228,824 | 1,843,176 | 40.00 | 614392 |
| IDATA06 | 3,072,000 | 102,424 | 2,969,576 | 3.33 | 1023992 |
| IDATA07 | 3,072,000 | 307,224 | 2,764,776 | 10.00 | 921592 |
| IDATA08 | 2,560,000 | 266,448 | 2,293,552 | 10.41 | 1023992 |
| IDATA09 | 13,312,000 | 5,012,048 | 8,299,952 | 37.65 | 1433592 |
| IDATA10 | 14,848,000 | 12,185,696 | 2,662,304 | 82.07 | 870392 |
| IDATA11 | 12,824,576 | 11,161,720 | 1,662,856 | 87.03 | 307192 |
| IDATA12 | 2,764,800 | 2,334,776 | 430,024 | 84.45 | 225272 |
| IDATA13 | 1,536,000 | 512,024 | 1,023,976 | 33.33 | 358392 |
| IDATA14 | 3,072,000 | 2,457,648 | 614,352 | 80.00 | 102392 |
| IDATA15 | 1,536,000 | 774,584 | 761,416 | 50.43 | 270232 |
| IDATA16 | 32,768,000 | 28,917,896 | 3,850,104 | 88.25 | 1638392 |
| IDATA17 | 2,048,000 | 573,136 | 1,474,864 | 27.99 | 369512 |
| IINDEX01 | 1,920,000 | 669,304 | 1,250,696 | 34.86 | 1023736 |
| IINDEX02 | 896,000 | 74,368 | 821,632 | 8.30 | 416464 |
| IINDEX03 | 1,536,000 | 194,584 | 1,341,416 | 12.67 | 471032 |
| IINDEX04 | 1,280,000 | 329,504 | 950,496 | 25.74 | 505240 |
| IINDEX05 | 2,048,000 | 614,432 | 1,433,568 | 30.00 | 460792 |
| IINDEX06 | 1,536,000 | 51,224 | 1,484,776 | 3.33 | 501752 |
| IINDEX07 | 1,536,000 | 307,224 | 1,228,776 | 20.00 | 409592 |
| IINDEX08 | 1,280,000 | 737,520 | 542,480 | 57.62 | 327640 |
| IINDEX09 | 7,168,000 | 3,649,536 | 3,518,464 | 50.91 | 665592 |
| IINDEX10 | 6,144,000 | 5,433,368 | 710,632 | 88.43 | 215032 |
| IINDEX11 | 3,840,000 | 3,297,320 | 542,680 | 85.87 | 409592 |
| IINDEX12 | 768,000 | 522,264 | 245,736 | 68.00 | 143352 |
| IINDEX13 | 768,000 | 512,024 | 255,976 | 66.67 | 102392 |
| IINDEX14 | 2,048,000 | 1,710,128 | 337,872 | 83.50 | 184312 |
| IINDEX15 | 1,280,000 | 780,272 | 499,728 | 60.96 | 481112 |
| IINDEX16 | 14,336,000 | 11,901,416 | 2,434,584 | 83.02 | 102400 |
| IINDEX17 | 1,024,000 | 201,112 | 822,888 | 19.64 | 205760 |



| | | | | | |
|----------|------------|-----------|------------|-------|--------|
| IRPT01 | 3,072,000 | 2,136,496 | 935,504 | 69.55 | 681176 |
| IUSER01 | 1,024,000 | 847,448 | 176,552 | 82.76 | 176264 |
| RBS | 20,480,000 | 5,734,480 | 14,745,520 | 28.00 | 409592 |
| SYSTEM | 512,000 | 77,888 | 434,112 | 15.21 | 417056 |
| TEMP_APP | 8,192,000 | 4,710,432 | 3,481,568 | 57.50 | 798712 |
| TEMP_RPT | 11,264,000 | 9,973,808 | 1,290,192 | 88.55 | 225272 |
| TEMP_USR | 11,264,000 | 6,512,688 | 4,751,312 | 57.82 | 921600 |

La relación que existe entre cada tablespace y datafiles es la siguiente:

| TABLESPACE | DATAFILE | | | | |
|------------|------------------------|------------------------|------------------------|--|--|
| SYSTEM | /u01/sysppay.dbf | | | | |
| IDATA01 | /u04/ldata011ppay.dbf | /u04/ldata014ppay.dbf | /u04/ldata015ppay.dbf | | |
| | /u09/ldata012ppay.dbf | /u09/ldata016ppay.dbf | /u14/ldata013ppay.dbf | | |
| IDATA02 | /u04/ldata021ppay.dbf | /u09/ldata022ppay.dbf | /u09/ldata024ppay.dbf | | |
| | /u14/ldata023ppay.dbf | | | | |
| IDATA03 | /u01/ldata031ppay.dbf | /u06/ldata032ppay.dbf | /u11/ldata033ppay.dbf | | |
| IDATA04 | /u04/ldata041ppay.dbf | /u09/ldata042ppay.dbf | /u14/ldata043ppay.dbf | | |
| | /u14/ldata044ppay.dbf | | | | |
| IDATA05 | /u01/ldata051ppay.dbf | /u06/ldata052ppay.dbf | /u11/ldata053ppay.dbf | | |
| IDATA06 | /u01/ldata061ppay.dbf | /u06/ldata062ppay.dbf | /u11/ldata063ppay.dbf | | |
| IDATA07 | /u01/ldata071ppay.dbf | /u06/ldata072ppay.dbf | /u11/ldata073ppay.dbf | | |
| IDATA08 | /u03/ldata084ppay.dbf | /u04/ldata081ppay.dbf | /u09/ldata082ppay.dbf | | |
| | /u14/ldata083ppay.dbf | | | | |
| IDATA09 | /u02/ldata091ppay.dbf | /u02/ldata094ppay.dbf | /u07/ldata092ppay.dbf | | |
| | /u07/ldata095ppay.dbf | | | | |
| | /u12/ldata093ppay.dbf | /u12/ldata096ppay.dbf | /u13/ldata097ppay.dbf | | |
| IDATA10 | /u02/ldata109ppay.dbf | /u04/ldata101ppay.dbf | /u04/ldata104ppay.dbf | | |
| | /u04/ldata105ppay.dbf | | | | |
| | /u06/ldata1011ppay.dbf | /u07/ldata107ppay.dbf | /u08/ldata107ppay.dbf | | |
| | /u09/ldata102ppay.dbf | | | | |
| | /u09/ldata106ppay.dbf | /u11/ldata108ppay.dbf | /u13/ldata1010ppay.dbf | | |
| | /u14/ldata103ppay.dbf | | | | |
| IDATA11 | /u03/ldata111ppay.dbf | /u03/ldata1111ppay.dbf | /u03/ldata116ppay.dbf | | |
| | /u03/ldata117ppay.dbf | | | | |
| | /u05/ldata114ppay.dbf | /u05/ldata115ppay.dbf | /u06/ldata1112ppay.dbf | | |

REGISTRO
FALLA DE ORIGEN

| | | | | |
|----------|------------------------|------------------------|------------------------|--|
| | /u06/ldata1113ppay.dbf | | | |
| | /u06/ldata1114ppay.dbf | /u06/ldata1118ppay.dbf | /u08/ldata1115ppay.dbf | |
| | /u08/ldata112ppay.dbf | | | |
| | /u10/ldata119ppay.dbf | /u13/ldata1110ppay.dbf | /u13/ldata113ppay.dbf | |
| IDATA12 | /u02/ldata121ppay.dbf | /u02/ldata124ppay.dbf | /u07/ldata122ppay.dbf | |
| | /u07/ldata125ppay.dbf | | | |
| | /u09/ldata126ppay.dbf | /u12/ldata123ppay.dbf | /u13/ldata127ppay.dbf | |
| IDATA13 | /u02/ldata131ppay.dbf | /u07/ldata132ppay.dbf | /u12/ldata133ppay.dbf | |
| IDATA14 | /u01/ldata146ppay.dbf | /u02/ldata141ppay.dbf | /u0/ldata142ppay.dbf | |
| | /u12/ldata143ppay.dbf | | | |
| | /u12/ldata144ppay.dbf | /u12/ldata145ppay.dbf | | |
| IDATA15 | /u03/ldata151ppay.dbf | /u08/ldata152ppay.dbf | /u13/ldata153ppay.dbf | |
| IDATA16 | /u05/ldata161ppay.dbf | /u05/ldata164ppay.dbf | /u05/ldata169ppay.dbf | |
| | /u10/ldata162ppay.dbf | | | |
| | /u10/ldata165ppay.dbf | /u10/ldata167ppay.dbf | /u13/ldata1611ppay.dbf | |
| | /u15/ldata1610ppay.dbf | | | |
| | /u15/ldata163ppay.dbf | /u15/ldata166ppay.dbf | /u15/ldata168ppay.dbf | |
| | /u16/ldata1612ppay.dbf | | | |
| | /u16/ldata1613ppay.dbf | /u16/ldata1614ppay.dbf | /u16/ldata1615ppay.dbf | |
| | /u17/ldata1616ppay.dbf | | | |
| | /u17/ldata1617ppay.dbf | /u17/ldata1618ppay.dbf | | |
| IDATA17 | /u16/ldata171ppay.dbf | /u16/ldata173ppay.dbf | /u17/ldata172ppay.dbf | |
| | /u17/ldata174ppay.dbf | | | |
| IINDEX01 | /u04/iindex011ppay.dbf | /u04/iindex014ppay.dbf | /u09/iindex012ppay.dbf | |
| | /u14/iindex013ppay.dbf | | | |
| | /u14/iindex015ppay.dbf | | | |
| IINDEX02 | /u04/iindex021ppay.dbf | /u09/iindex022ppay.dbf | /u14/iindex023ppay.dbf | |
| | /u14/iindex024ppay.dbf | | | |
| IINDEX03 | /u01/iindex031ppay.dbf | /u06/iindex032ppay.dbf | /u11/iindex033ppay.dbf | |
| IINDEX04 | /u04/iindex041ppay.dbf | /u09/iindex042ppay.dbf | /u13/iindex044ppay.dbf | |
| | /u14/iindex043ppay.dbf | | | |
| IINDEX05 | /u01/iindex051ppay.dbf | /u01/iindex054ppay.dbf | /u06/iindex052ppay.dbf | |
| | /u11/iindex053ppay.dbf | | | |
| IINDEX06 | /u01/iindex061ppay.dbf | /u06/iindex062ppay.dbf | /u11/iindex063ppay.dbf | |
| IINDEX07 | /u01/iindex071ppay.dbf | /u06/iindex072ppay.dbf | /u11/iindex073ppay.dbf | |
| IINDEX08 | /u03/iindex084ppay.dbf | /u04/iindex081ppay.dbf | /u09/iindex082ppay.dbf | |

ESTAS CON
FALLA DE ORIGEN



| | | | |
|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------|-------------------------------------------------------------------------------|
| | /u14/iindex083ppay.dbf | | |
| IINDEX09 | /u02/iindex091ppay.dbf /u07/iindex095ppay.dbf /u12/iindex093ppay.dbf | /u02/iindex094ppay.dbf | /u0/iindex092ppay.dbf |
| IINDEX10 | /u04/iindex101ppay.dbf /u13/iindex106ppay.dbf /u14/iindex103ppay.dbf | /u09/iindex102ppay.dbf | /u09/iindex104ppay.dbf |
| IINDEX11 | /u03/iindex111ppay.dbf /u12/iindex114ppay.dbf /u13/iindex113ppay.dbf | /u08/iindex112ppay.dbf | /u08/iindex115ppay.dbf |
| IINDEX12 | /u02/iindex121ppay.dbf | /u07/iindex122ppay.dbf | /u12/iindex123ppay.dbf |
| IINDEX13 | /u02/iindex131ppay.dbf | /u07/iindex132ppay.dbf | /u12/iindex133ppay.dbf |
| IINDEX14 | /u02/iindex141ppay.dbf /u07/iindex142ppay.dbf /u12/iindex143ppay.dbf | /u02/iindex144ppay.dbf | /u03/iindex146ppay.dbf |
| IINDEX15 | /u03/iindex151ppay.dbf /u13/iindex153ppay.dbf | /u08/iindex152ppay.dbf | /u08/iindex154ppay.dbf |
| IINDEX16 | /u05/iindex161ppay.dbf /u05/iindex167ppay.dbf /u06/iindex1614ppay.dbf /u10/iindex162ppay.dbf /u10/iindex165ppay.dbf /u15/iindex163ppay.dbf /u15/iindex166ppay.dbf /u17/iindex1616ppay.dbf | /u05/iindex1610ppay.dbf | /u05/iindex164ppay.dbf /u09/iindex1612ppay.dbf |
| IINDEX17 | /u16/iindex171ppay.dbf /u17/iindex174ppay.dbf | /u16/iindex173ppay.dbf | /u17/iindex172ppay.dbf |
| IRPT01 | /u11/irpt011ppay.dbf | /u11/irpt012ppay.dbf | |
| IUSER01 | /u01/iuser01ppay.dbf | | |
| RBS | /rollback/rbs01ppay.dbf /rollback/rbs04ppay.dbf /rollback/rbs05ppay.dbf /rollback/rbs08ppay.dbf /rollback/rbs09ppay.dbf | /rollback/rbs02ppay.dbf | /rollback/rbs03ppay.dbf /rollback/rbs06ppay.dbf /rollback/rbs07ppay.dbf |
| TEMP_APP | /tmpapp/temp_app01ppay.dbf /tmpapp/temp_app03ppay.dbf | /tmpapp/temp_app02ppay.dbf | /tmpapp/temp_app04ppay.dbf |

TEMP APP
FALLA DE ORIGEN



| | | |
|----------|-----------------------------|-----------------------------|
| TEMP_RPT | /tmpmprl/temp_rpt01ppay.dbf | /tmpmprl/temp_rpt02ppay.dbf |
| | /tmpmprl/temp_rpt03ppay.dbf | |
| | /tmpmprl/temp_rpt04ppay.dbf | /tmpmprl/temp_rpt05ppay.dbf |
| | /u08/temp_rpt06ppay.dbf | |
| TEMP_USR | /tmpmprl/temp_usr06ppay.dbf | /tmpusr/temp_usr01ppay.dbf |
| | /tmpusr/temp_usr02ppay.dbf | |
| | /tmpusr/temp_usr03ppay.dbf | /tmpusr/temp_usr04ppay.dbf |
| | /tmpusr/temp_usr05ppay.dbf | |

IV.3 ANÁLISIS DE MIGRACIÓN

Una de las partes fundamentales en cualquier proceso de migración de una Base de Datos radica en el análisis y decisión del método por medio del cual se llevará a cabo la migración, ya que de ello depende el control y rapidez con que se ejecute exitosamente, dicho proceso. En la mayoría de los casos, el tiempo que se invierte en la migración es de suma importancia debido a que en la actualidad muchas de las aplicaciones son críticas; es decir de 7 x 24; lo cual significa que el tiempo que una aplicación se encuentre fuera de servicio, impacta profundamente el servicio que dicha aplicación está brindando. Es por ello que se debe de tener claramente definido el tipo de aplicación que se está manejando, los objetivos y tiempos propuestos; en base a ello, a las ventajas y desventajas de cada método de migración se puede hacer la elección correcta.

En el capítulo anterior se describió de manera detallada y funcional los métodos existentes para llevar a cabo una migración de Base de Datos, los cuales, fueron:

- Migration Utility
- Data Migration Assistant
- Export/Import
- Copia de Datos

Ya se ha descrito el tipo de Base de Datos con la que se trabaja, debido a sus características y las pruebas realizadas con cada uno de los métodos en cuanto a tiempos requeridos para cada una de las tareas tanto previas como durante y después de la migración, se ha elegido que la migración de la Base de Datos se realice por medio de la utilidad de **Oracle Migration Utility**.



Es importante mencionar que en algunas de las tareas previas y posteriores al proceso de migración, se aplican otros métodos tales como export/import y copia de

datos. Esta decisión, fué tomada gracias a las pruebas realizadas en el ambiente alterno, ya que de esta manera se pudo observar la funcionalidad de cada una de los métodos a fin de poder aplicar sus ventajas en diferentes puntos de nuestro proceso de migración.

Una vez elegido el método de migración y después de un exhaustivo plan de pruebas se llega a los resultados de que este mecanismo requiere de aproximadamente ;5 horas para ser ejecutado; lo cual implica que la aplicación no brindará servicio durante todo este tiempo impactando de manera considerable, el estado financiero de la empresa, lo cual es inaceptable por parte del personal responsable del proyecto. Debido a ello, es necesario implementar un nuevo esquema para el proceso de migración, el cual consiste en lo siguiente:

- 1.- Disponer del hardware necesario para poseer un servidor con las mismas características que el productivo, considerando espacio en disco, cpu's y memoria.
- 2.- Configurar el hardware de la misma manera como se encuentra el ambiente productivo, es decir, contar con la misma estructura de file systems, para generar una replica de la Base de Datos productiva.

Estos 2 requerimientos, fueron necesarios ya que la implantación del nuevo esquema de migración consiste en lo siguiente:

- 1.- Se realiza un respaldo en caliente (Hotbackup) de la Base de Datos en el ambiente productivo (Dominio1) y se lleva al ambiente alterno (Dominio2).
- 2.- Se sincroniza esta Base de Datos Dominio2) y se aplica el proceso de migración sobre el Dominio2.
- 3.- El Dominio1 continúa brindando servicio.
- 4.- Al finalizar la migración, únicamente se realiza un cambio de aplicación entre Dominios, siendo ahora el Dominio2 el productivo y viceversa.

Durante el tiempo de migración se generaron transacciones las cuales hasta este momento no se encuentran reflejadas en el ambiente migrado, para corregir esto, es necesario realizar la extracción de los datos y aplicarlos en el nuevo ambiente.

Con la implantación de este esquema y gracias a las ventajas que ofrece la utilería de migración de Oracle (Migration Utility) es posible cumplir con los objetivos y



reducir a cero el tiempo de negación de servicio, además de cumplir con los tiempos mínimos requeridos por parte del área responsable de este proyecto.

ESQUEMA 1.

CONFIGURACIÓN ACTUAL

- Dominio1

Se posee un servidor Sun Enterprise 10k con un Sistema Operativo Sun Solaris 2.5 y una Base de Datos Oracle 7.3.4, la cual se encuentra espejeada hacia 2 arreglos de discos. La versión de la aplicación es 6.0

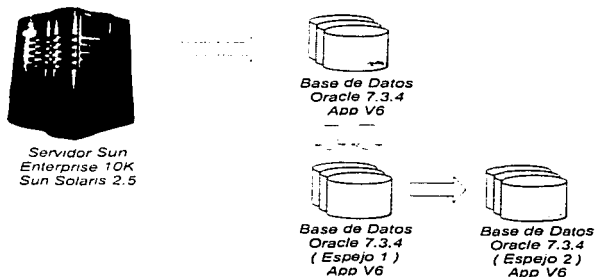


Figura IV.3a Configuración Actual Sistema de Prepago

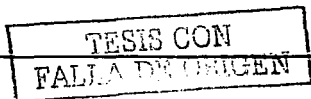
En este Dominio, se encuentra totalmente la producción del Sistema.

ESQUEMA 2

ESTRATEGIA DE MIGRACIÓN

- Dominio1.

Se desasocia el espejo 2 del Dominio1 y se asocia al Dominio2.





Se obtiene un Hotbackup de la Base de Datos en el Dominio1 y éste es restaurado en el Dominio2; mientras que el Dominio1 aún continúa con la producción de nuestro sistema.

- Dominio2

El Dominio2 se encuentra con Sistema Operativo Sun Solaris 2.8, el cuál, fué instalado como tarea previa y asociado con una replica de la Base de Datos (Oracle 7.3.4) de Dominio1. Se encuentra listo para aplicar el proceso de migración.

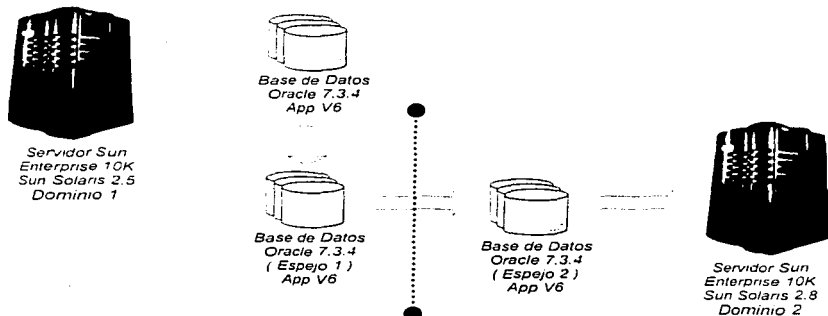


Figura IV.3b Estrategia de Migración

ESQUEMA 3

PROCESO DE MIGRACIÓN REALIZADA.

- Dominio1.

Se tiene el Dominio1 actualmente productivo con el Sistema Operativo 2.5 y Base de Datos Oracle 7.3.4. La producción aún continúa en este Dominio.

- Dominio2.



Se ha realizado exitosamente el proceso de migración en el Dominio2 y este ahora posee Sistema Operativo Sun Solaris 2.8, una Base de Datos Oracle 8.1.6. y la versión 7.0 de nuestra Aplicación



Servidor Sun Enterprise 10K Sun Solaris 2.5 Dominio 1



Base de Datos Oracle 7.3.4 App V6



Servidor Sun Enterprise 10K Sun Solaris 2.8 Dominio 2



Base de Datos Oracle 8.1.6 App V7



Base de Datos Oracle 7.3.4 (Espejo 1) App V6

Figura IV.3c Proceso de Migración Realizada

ESQUEMA 4

CONFIGURACION DEL SISTEMA FINAL

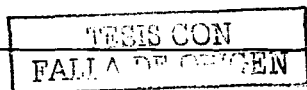
Dominio1.

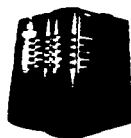
Este Dominio queda con la Base de Datos Oracle 7.3.4 y el Sistema Operativo Sun Solaris 2.5

Dominio2.

La migración se ha llevado a cabo, tanto de nuestro Sistema Operativo y Base de Datos, como también de la Aplicación. Se ha realizado el switcheo del Sistema y ahora la producción de nuestro sistema se encuentra en este Dominio.

Por último, únicamente se hace un espejo de la Base de Datos como seguridad del sistema.

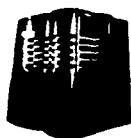




Servidor Sun Enterprise 10K
Sun Solaris 2.5
Dominio 1



Base de Datos Oracle 7.3.4
App V6



Servidor Sun Enterprise 10K
Sun Solaris 2.8
Dominio 2



Base de Datos Oracle 8.1.6
App V7

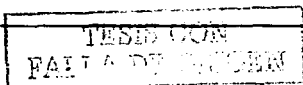


Base de Datos Oracle 8.1.6
(Espejo 1)
App V7

Figura IV.3d Configuración del Sistema Final

La migración de esta Base de Datos, se resume en las siguientes tareas a realizar además de proporcionar el tiempo requerido por cada una de las actividades teniendo un plan estricto a seguir y así lograr que la migración se realice de una manera exitosa.

| # | Actividad | Duración (minutos) | Comentarios |
|---|--------------------------------------------------------------------------------|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | Respaldo de la Base de Datos en Dominio1 via HotBackup (Version 7.3.4) | 120 | Se realiza un hot backup de la base de datos en los file system /hotbck1 /hotbck2 y /hotbck3, con la finalidad de restaurarla en dominio2. |
| 2 | Transferencia de Hot Backup a Dominio2 | 10 | Mediante sistema operativo se realiza una desasociación de los file systems hotbckX y se asocian al dominio2. |
| 3 | Restauración de respaldo en Dominio2 | 120 | Se restaura la Base de Datos en las rutas correspondientes en el Dominio2. |
| 4 | Sincronización de Base de Datos en Dominio2 | 900 | Transferencia y Aplicación de archivos Archive en Base de Datos Dominio2. |
| 5 | Paro de Ivrs y Sistema de Bancos | 15 | Se prohíbe el acceso a las tablas de fichas |
| 6 | Abrir Base de Datos. | 30 | Se abre la Base de Datos. |
| 7 | Adición de Espacios a la Base de Datos en Dominio2 | 60 | Se adiciona espacio a algunos tablespaces. |
| 8 | Actualización del Software de Oracle y Migración de Base de Datos en Dominio2. | 180 | Tener compartido el cd de Oracle |
| 9 | Actualización de Aplicación | 30 | Opción 4 del menú de actualización de |



| | | | |
|----|---------------------------------------------------------------------------|-----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 10 | Actualización de la Base de Datos para Versión 7 de Aplicación. | 180 | <p>Aplicación</p> <p>- Opción 5 del menú de actualización de Aplicación</p> <p>- En esta actividad se incluye el procedimiento de la descarga de la tabla de Fichas.</p> |
| 11 | Reenrutamiento de tráfico hacia Dominio2. Pruebas de tráfico. | 120 | Se realizarán las pruebas de tráfico para validar que la Aplicación este cobrando bien las llamadas y estas puedan realizarse sin problemas. En paralelo se realizará un reporte en el dominio1 para obtener el tráfico no facturado por la actualización en Dominio2. |
| 12 | Apertura de IVR's, Carga de Fichas y Sistema de Bancos | 20 | Se abre el acceso para el abono de fichas. Se requiere también probar el correcto funcionamiento para cargar fichas con la nueva versión de Aplicación. |
| 13 | Respaldo de la Base de Datos vía HotBackup (Versión Oracle8) en Dominio2. | 120 | Se realiza un hot backup de toda la base de datos en los file system /hotbck1 /hotbck2 y /hotbck3, con la finalidad de recuperarla en caso de corrupción total |
| 14 | Respaldo de la Base de Datos vía Export (Versión Oracle 8) en Dominio2. | 120 | Se realiza un export de la Base de Datos para recuperar objetos en caso de corrupción parcial. |
| 15 | Ejecución de Programa para aplicar descuentos en dominio2 | 120 | Se ejecuta un script, el cual realizara las operaciones de substracción a los números del dominio2 que realizaron llamadas en dominio1, mientras el proceso de migración se realizaba. |
| 16 | Sincronización de Mirror | 300 | Se desasocia el espejo en el dominio1 y se asocia en Dominio2. |

IV.4 TAREAS PREVIAS

IV.4.1 RESPALDO DE LA BASE DE DATOS EN DOMINIO1 VIA HOTBACKUP.

Para el respaldo de la Base de Datos se va a hacer uso de uno de los métodos mencionados en el capítulo anterior, este respaldo es vía Hot Backup en el dominio1 actualmente productivo y se aloja en los file systems /hotbck1, /hotbck2, /hotbck3 y /hotbck4, con la finalidad de restaurarse en Dominio2.

Para este respaldo, se utilizan los siguientes programas los cuales se encuentran automatizados y se deben de ejecutar de la siguiente manera:

1.- Poner en modo de respaldo todos los tablespaces de la Base de Datos.



Para esta tarea se deben de ejecutar dos programas, el primero tiene como resultado generar los comandos que serán utilizados para iniciar el respaldo. El primer programa es el siguiente y se encuentra en la siguiente ruta, ya que estos programas tienen extensión .sql y se deben de correr bajo el prompt sqlplus.

```
/hotbck1/HOTBACKUP/genera_beginbackup.sql  
/hotbck1/HOTBACKUP/beginbackup.sql
```

Ver detalle en **APENDICE I IV.4.1.1**

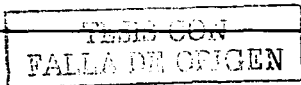
2.- Realizar la copia física de todos los datafiles de la Base de Datos.

Para la copia de todos los archivos de la Base de Datos primero se debe de generar un pipe en las rutas de respaldo para ello se ejecutan varios programas.

- Generar pipes para datafiles en la ruta de respaldo /hotbck1
/hotbck1/HOTBACKUP/generapipesdb1.sql
- Generar pipes para datafiles en las rutas de respaldo restantes (hotbck2, hotbck3 y hotbck4)
/hotbck1/HOTBACKUP/copia_pipes.sh
- Creación de Pipes para datafiles.
/hotbck1/HOTBACKUP/pipes_dbf1.sh
/hotbck1/HOTBACKUP/pipes_dbf2.sh
/hotbck1/HOTBACKUP/pipes_dbf3.sh
/hotbck1/HOTBACKUP/pipes_dbf4.sh
- Generar pipes para logfiles en las rutas de respaldo (/hotbck1 /hotbck2, /hotbck3, /hotbck4)
/hotbck1/HOTBACKUP/gen_pipeslogs.sh
- Creación de Pipes para Logfiles.
/hotbck1/HOTBACKUP/pipes_logs.sh

Ver detalle en **APENDICE I IV.4.1.2**

Pipe. Archivo especial que es utilizado para transferir datos entre 2 procesos. Uno de ellos, escribe en este archivo, mientras que el otro lee de él.





Después de que los pipes están creados en las rutas del respaldo se ejecuta el siguiente programa el cual utiliza el comando dd de unix para la copia de archivos a nivel sistema operativo donde se indica el archivo de entrada y el archivo de salida el cual contiene la copia; además de ello se le agrega otro comando de unix (cat) con el cual se comprime el archivo de salida utilizando menos espacio que el archivo origen de esta manera una Base de Datos de gran tamaño puede ser respaldada en pocos gigabytes.

```
/hotbck1/HOTBACKUP/gen_dd_resp.sql  
/hotbck1/HOTBACKUP/gen_dd_resp_logs.sql  
nohup /hotbck1/HOTBACKUP/dd_resp.sh &  
nohup /hotbck1/HOTBACKUP/dd_resp_logs.sh &
```

Ver detalle en APENDICE I IV.4.1.3

Por ultimo, sólo es necesario que los pipes que fueron generados para el respaldo se deben de borrar con el siguiente programa.

```
/hotbck1/HOTBACKUP/borra_pipes.sh
```

Ver detalle en APENDICE I IV.4.1.4

3.- Poner todos los tablespaces en modo de termino de respaldo.

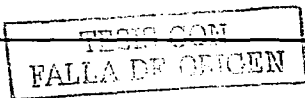
Para esta tarea se deben de ejecutar dos programas en el siguiente orden:

```
/hotbck1/HOTBACKUP/gen_endbackup.sql  
/hotbck1/HOTBACKUP/endbackup.sql
```

Ver detalle en APENDICE I IV.4.1.5

4.- Realizar una copia de los archivos de control de la Base de Datos.

Es necesario también respaldar los archivos de control en forma física a otra ruta de la siguiente manera:





```
cp -p /u01/oradata/ppay/ctrl1ppay.ctl /hotbck1/HOTBACKUP/ctrl1ppay.ctl  
cp -p /u02/oradata/ppay/ctrl2ppay.ctl /hotbck1/HOTBACKUP/ctrl2ppay.ctl
```

o ejecutando la siguiente sentencia a nivel Oracle, la cual permite recuperarlos del trace generado.

```
Alter database backup control_file to trace;  
ó  
Alter database backup controlfile to '/hotbck1/oradata/ctrl2ppay.ctl';
```

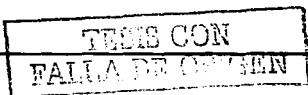
NOTA: Deben incluirse todos los archivos que muestran las vistas dba_data_files y v\$logfile. Si se desea que todos los datafiles sean respaldados en forma paralela es necesario correr cada una de las líneas en background.

IV.4.2 RESTAURACIÓN DE BASE DE DATOS EN DOMINIO2

1.- Transferencia de Respaldo.

Una vez finalizado y validado el respaldo en caliente en dominio1 se procede a su transferencia al dominio2. Esta tarea se puede realizar utilizando diferentes técnicas de transferencia de archivos, tales como ftp, asociando y desasociando discos, respaldo y restauración en cintas; cada una de estas técnicas tienen ventajas y desventajas una sobre la otra dependiendo de la cantidad de archivos y sus tamaños que sean transferidos. En este caso la transferencia será por medio de comandos a nivel Sistema Operativo, los cuales consisten en desasociar los discos del dominio1 y asociarlos al dominio2, la ventaja de esta técnica radica en el tiempo requerido para la transferencia de los archivos del respaldo ya que son alrededor de 200 Gb de información y son transferidos en solo 10 minutos.

Ver detalle en APENDICE I IV.4.2.1





2.- Restauración física de Datafiles en Dominio2.

Una vez que el Hot Backup se tiene en el dominio2 y que se ha validado que se encuentra completo y sin errores, se procede a iniciar la restauración de la Base de Datos. Básicamente, el proceso consiste en recuperar cada uno de los datafiles en sus rutas correspondientes y esto se realiza ejecutando los scripts que continuación se mencionan:

```
/hotbck1/HOTBACKUP/gen_dd_restor_dbf.sql
/hotbck1/HOTBACKUP/gen_dd_restor_logs.sql
nohup /hotbck1/HOTBACKUP/dd_restor_dbf.sh >>
/hotbck1/HOTBACKUP/dd_restor_dbf.out &
nohup /hotbck1/HOTBACKUP/dd_restor_logs.sh >>
/hotbck1/HOTBACKUP/dd_restor_logs.out &
```

Ver detalle en APENDICE I IV.4.2.2

IV.4.3 RESTAURACION Y SINCRONIZACIÓN DE BASE DE DATOS EN DOMINIO2

Cuando cada uno de los archivos de datos (Datafiles), que componen la Base de Datos se encuentran en sus rutas correspondientes y en los archivos de salida (dd_restor_dbf.out y dd_restor_logs.out) de la ejecución de los scripts que contienen los comandos dd's no se encuentra ningún error de dispositivo Lleno ó Error de Escritura; se procede a realizar la restauración y sincronización de la Base de Datos la cuál consiste en lo siguiente:

1.- Copia de archivos de control a rutas correspondientes

```
cp -p /hotbck1/HOTBACKUP/ctrl1ppay.ctl /u02/oradata/ppay/ctrl1ppay.ctl
cp -p /hotbck1/HOTBACKUP/ctrl2ppay.ctl /u03/oradata/ppay/ctrl2ppay.ctl
```

2.- Iniciar una sesión de Server Manager.

```
# svrmgrl
SVRMGR> connect internal
```

3.- Montar la Instancia.

```
SVRMGR> startup mount
```




4.- Deshabilitar la bandera de recuperación en automático.

```
SVRMGRL> set autorecovery off
```

5.- Iniciar recuperación y sincronización de Base de Datos.

```
SVRMGRL> recover database using backup controlfile until cancel;
```

Al ejecutar la sentencia anterior, la Base de Datos inicia su proceso de restauración y sincronización solicitando archivos log para que las transacciones contenidas en ellos, sean aplicadas. Es importante mencionar que cada uno de estos archivos log se encuentran en el file system /archive de Dominio1 y estos deben de ser transferidos por medio de ftp hacia el Dominio2 para ser aplicados uno por uno conforme los solicite la recuperación.

Este proceso de sincronización finaliza hasta que se logra aplicar todos y cada uno de los logs generados en Dominio1.

Ver detalle en **APENDICE I IV.4.3.1**

IV.4.4 PARO DE IVR'S Y SISTEMA DE BANCOS EN DOMINIO1

Los Ivr's deben de ser dados de baja al igual que el ingreso de saldo por medio de los bancos en el Dominio1, esto con el fin de evitar que existan actualizaciones a las tablas de pagos y estas no puedan reflejarse en el Dominio2.

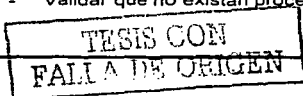
El paro de estos sistemas consiste en lo siguiente:

1.- Paro de IVR's

- Abrir una sesión de unix y conectarse como el usuario dueño de la aplicación

```
SunOS 5.8  
login: pp  
Password: XXXXXXXX
```

- Eliminar procesos de IVR's
for x in `ps -ef | grep ivrHandl | grep -v grep | awk '{ print \$2 }'
do
kill -9 \$x
done
- Validar que no existan procesos





ps -ef | grep ivrHandl

- Conectarse a cada IVR's con el usuario root y darlo de baja
SunOS 5.8
login: root
Password: XXXXXXXX

\$ init 0

2.- Paro de Aplicación de Bancos.

En lo que se refiere al paro de la aplicación de bancos, unicamente se cierran las conexiones a la Base de Datos dando de baja el listener que atiende esta aplicación; esto se realiza de la siguiente manera:

- Abrir una sesión de Unix con el usuario Oracle
SunOS 5.8
login: oracle
Password: XXXXXXXX
- Una vez dentro del Sistema Operativo, teclear el siguiente comando:
\$ isnrctl stop LSNR1

El dar de baja la aplicación de Bancos es responsabilidad del área de Informática.

IV.4.5 ABRIR BASE DE DATOS DE DOMINIO2

Una vez que en el Dominio1 se ha detenido el ingreso de saldo por medio de IVR y el sistema de bancos no acepta alguna transacción, además que se han aplicado todos y cada uno de los archivos log generados en el Dominio1, se procede a abrir la Base de Datos.

1.- Apertura de Base de Datos en Dominio2

```
alter database open resetlogs;
```

En este momento, la Base de Datos en Dominio2 se encuentra recuperada y sincronizada; es decir lista para iniciar el proceso de migración en ella.



IV.4.6 ADICIÓN DE ESPACIOS DE BASE DE DATOS DE DOMINIO2

Debido a que durante este proceso de migración no solamente se migra la versión del manejador de Oracle, sino también la versión del sistema operativo y la versión de la aplicación, es necesario adicionar espacio a los tablespaces que se encuentran relacionados con las tablas nuevas que serán adicionadas a la aplicación y también para todas aquellas tablas que ya existían en la versión anterior pero que ahora sufren modificaciones.

Los tablespaces a los que se les adicionará espacio son los siguientes:

IDATA01.
IDATA02.
IDATA03.
IDATA09.
IDATA10.
ICONV01.
IINDEX03
IINDEX04
RBS01.
RBS02.
RBS03.
RBS04.
IDATA16.
IINDEX16

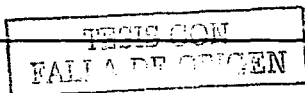
La adición de espacios a cada uno de los tablespaces anteriores, se realiza por medio de la siguiente sentencia:

```
$ sqlplus system  
password: XXXXX
```

```
SQL> Alter tablespace IDATA16 add datafile  
/u01/oradata/ppay/idata1601ppay.dbf size 2000mb;
```

Es necesario crear 2 nuevos tablespaces, uno de datos y uno para índices ya que ahí se alojarán tablas nuevas.

```
TABLESPACE IDATA20.  
TABLESPACE IINDEX06.
```





La creación del tablespace se realizará ejecutando la siguiente sentencia:

```
$ sqlplus system  
password: XXXXX
```

```
SQL> CREATE TABLESPACE IDATA20  
  DATAFILE 'u16/oradata/ppay/ldata201ppay.dbf' SIZE 2000M  
  DEFAULT STORAGE(INITIAL 100M NEXT 100M MINEXTENTS 1 MAXEXTENTS  
  UNLIMITED  
  PCTINCREASE 0) ONLINE;
```

```
ALTER TABLESPACE IDATA20
```

```
  ADD DATAFILE 'u37/oradata/ppay/ldata202ppay.dbf'  
  SIZE 500M REUSE;
```

```
ALTER TABLESPACE IDATA20
```

```
  ADD DATAFILE 'u37/oradata/ppay/ldata203ppay.dbf'  
  SIZE 500M REUSE;
```

```
ALTER TABLESPACE IDATA20
```

```
  ADD DATAFILE 'u37/oradata/ppay/ldata204ppay.dbf'  
  SIZE 500M REUSE;
```

```
-----  
-- TABLESPACES DE INDICES --  
-----
```

```
CREATE TABLESPACE IINDEX06  
  DATAFILE 'u38/oradata/ppay/iindex061ppay.dbf'  
  SIZE 250M REUSE  
  DEFAULT STORAGE(INITIAL 50M  
  NEXT 20M  
  MINEXTENTS 1  
  MAXEXTENTS UNLIMITED  
  PCTINCREASE 0)
```

```
ONLINE;
```

```
ALTER TABLESPACE IINDEX06
```

```
  ADD DATAFILE 'u38/oradata/ppay/iindex062ppay.dbf'  
  SIZE 250M REUSE;
```

```
ALTER TABLESPACE IINDEX06
```

```
  ADD DATAFILE 'u38/oradata/ppay/iindex063ppay.dbf'  
  SIZE 250M REUSE;
```

```
ALTER TABLESPACE IINDEX06
```

```
  ADD DATAFILE 'u38/oradata/ppay/iindex064ppay.dbf'  
  SIZE 250M REUSE;
```



IV.5 PROCESO DE MIGRACIÓN DE LA BASE DE DATOS EN EL DOMINIO2.

IV.5.1 MIGRACIÓN Y ACTUALIZACIÓN DE LA BASE DE DATOS ORACLE 7.3.4 A ORACLE 8.1.6 EN DOMINIO2.

IV.5.1.1 PREPARACIÓN DE LA BASE DE DATOS DE PRODUCCIÓN PARA MIGRACIÓN.

1.- Asegurarse de que los archivos de datos (Datafiles) y los tablespaces se encuentran en línea, o fuera de línea normal. Además de ello, determinar si alguno de los archivos de datos requieren de recuperación y esto se puede realizar por medio de la siguiente sentencia de sqlplus.

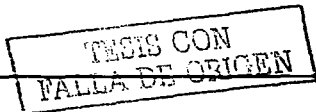
```
$sqlplus system/xxxxx  
SQL> SELECT * FROM v$recover_file;
```

El mensaje que se debe de recibir de esta sentencia es: " Cero Registros Seleccionados ", lo cual significa que todos los archivos de datos (Datafiles) están en línea o fuera de línea en forma normal. Si algún archivo de datos es listado, se tiene que restaurar antes de migrar la Base de Datos.

2.- No debe de existir algún usuario o role con el nombre de OUTLN, porque este esquema es creado automáticamente cuando Oracle8i, es instalado. En caso de que exista, es necesario que el usuario o el role sea borrado y recreado con un nombre distinto.

Para verificar si existe un usuario o role con estas características, se teclea la siguiente sentencia:

```
SQL> SELECT username FROM dba_users WHERE username = 'OUTLN';  
SQL> SELECT role FROM dba_roles WHERE role = 'OUTLN';
```





3.- Asegurarse de que no exista algún usuario o role con el nombre de MIGRATE, ya que la utilería de migración de Oracle crea este esquema y lo utiliza para reemplazar cualquier usuario, o role con este nombre y al finalizar lo elimina del sistema.

Para verificar si existe un usuario o role de nombre MIGRATE, se teclea la siguiente sentencia:

```
SQL> SELECT username FROM dba_users WHERE username = 'MIGRATE';  
SQL> SELECT role FROM dba_roles WHERE role = 'MIGRATE';
```

4.- Los Segmentos de Rollback de system no deben de tener establecido el parámetro de OPTIMAL, ya que esto ocasiona problemas durante el proceso de migración. Para ello, se consulta la vista v\$rollname y v\$rollstat de la siguiente manera:

```
SQL> SELECT a.usn, a.name, b.optsize  
FROM v$rollname a, v$rollstat b  
WHERE a.usn = b.usn AND name = 'SYSTEM';
```

Si existe algún valor en la columna de optsize, debe de establecerse el valor de nulo para estos segmentos de rollback y esto se realiza con la siguiente sentencia:

```
SQL> ALTER ROLLBACK SEGMENT SYSTEM STORAGE ( OPTIMAL NULL );
```

Una vez que el proceso de migración sea completado, puede este valor volver a su normalidad.

5.- Incrementar el máximo número de extents para el segmento de rollback de SYSTEM, alterando el parámetro de MAXEXTENTS en la cláusula de STORAGE de la sentencia ALTER ROLLBACK SEGMENT. El siguiente es un ejemplo de la sentencia ALTER ROLLBACK SEGMENT.

```
SQL> ALTER ROLLBACK SEGMENT system STORAGE ( MAXEXTENTS 505  
);  
SQL> ALTER ROLLBACK SEGMENT system STORAGE ( NEXT 10M );
```



Es posible que sea necesario adicionar más espacio a los segmentos de rollback de system para que el proceso de migración sea exitoso. Si el espacio no es suficiente se encontrarán graves problemas durante el proceso de migración.

6.- Verificar que el tablespace asociado a los usuarios de system y sys sea el tablespace de SYSTEM, además de que el tablespace temporal también sea el tablespace de SYSTEM, de lo contrario correr los siguientes comandos de sqlplus, esto se hace de la siguiente manera.

```
$ sqlplus system  
password: XXXXX
```

```
SQL> ALTER USER SYSTEM DEFAULT TABLESPACE SYSTEM TEMPORARY  
TABLESPACE SYSTEM;  
SQL> ALTER USER SYS DEFAULT TABLESPACE SYSTEM TEMPORARY  
TABLESPACE SYSTEM;
```

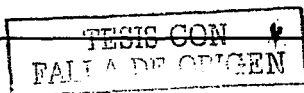
Además, se debe de cambiar uno de los parámetros de almacenamiento de este tablespace ejecutando los siguientes comandos:

```
SQL> ALTER TABLESPACE SYSTEM DEFAULT STORAGE ( PCTINCREASE 0 );
```

7.- Modificar las variables de ambiente para no tener problemas posteriores correspondientes a direccionamiento de rutas, para esto se deben de seguir los siguientes puntos:

7.1 Revisar el archivo .profile del usuario oracle y todas las variables definidas deben de moverse a otro archivo (oracle_profile) como se menciona a continuación.

```
$ su - oracle  
$ cp -p .profile .profile.before70  
$ vi .profile  
<remover variables de ambiente>  
$ vi cfg/oracle_profile  
<agregar variables de ambiente>
```





Asegurarse que la siguiente variable de ambiente esté definida en el archivo 'oracle_profile'.

```
TNS_ADMIN=$ORACLE_HOME/network/admin;export TNS_ADMIN
```

7.2 Es necesario generar una copia de seguridad de todos los archivos de configuración de la Base de Datos de versión 7. Esto se realiza de la siguiente manera.

```
$ su - oracle
$ cd /oracle/ora734/dbs
$ cp -p initppay.ora initppay.ora.save734
$ cp -p configppay.ora configppay.ora.save734
$ cd ~/cfg
$ cp -p oracle_profile oracle_profile.save734
```

7.3 Crear un nuevo directorio 'ora8' y un archivo temporal del profile del usuario Oracle, en el cual se cambiarán todas las referencias de 'ora734' a 'ora8'.

```
$ cd /oracle
$ mkdir ora8
$ cd ~/cfg
$ cp -p oracle_profile oracle_profile.ora8
$ edit oracle_profile.ora8
1,$s/ora734/ora8/g
wq
```

7.4 Definir la variable de ambiente ORA_NLS33 hacia el directorio de trabajo del ambiente de Oracle7. y agregarla también en el archivo 'oracle_profile' de Oracle 8:

```
$ export ORA_NLS33=$ORACLE_HOME/migrate/nls/admin/data
$ cd ~/cfg
$ edit oracle_profile.ora8
a
export ORA_NLS33=$ORACLE_HOME/migrate/nls/admin/data
.
wq

$ edit oracle_profile
a
export ORA_NLS33=$ORACLE_HOME/migrate/nls/admin/data
.
wq
```




7.5 Asegurarse que la variable de ambiente NLS_LANG hace referencia al Carácter Set que se utiliza actualmente en la Base de Datos. Además, es necesario poner esta variable en el archivo de 'oracle_profile' de la versión Oracle8.

Para verificar el Carácter Set de la Base de Datos, se ejecuta la siguiente sentencia:

```
$sqlplus system  
passwd: xxxxxx
```

```
SQL> SELECT * FROM v$nls_parameters  
WHERE parameter = 'NLS_LANGUAGE'  
OR parameter = 'NLS_TERRITORY'  
OR parameter = 'NLS_CHARACTERSET';
```

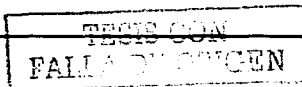
Un ejemplo del resultado de esta consulta es el siguiente:

| PARAMETER | VALUE |
|------------------|-----------------|
| NLS_LANGUAGE | MEXICAN SPANISH |
| NLS_TERRITORY | MEXICO |
| NLS_CHARACTERSET | WE8DEC |

Los valores que muestra esta sentencia son usados para definir la variable NLS_LANG. En este caso el valor de la variable sería el siguiente:

```
$ NLS_LANG="MEXICAN SPANISH_MEXICO.WE8DEC"  
$ export NLS_LANG  
$ cd ~/cfg  
$ edit oracle_profile.oracle8  
  a  
  NLS_LANG="MEXICAN SPANISH_MEXICO.WE8DEC" ; export  
NLS_LANG  
  wq  
  
$ edit oracle_profile  
  a  
  export NLS_LANG="MEXICAN SPANISH_MEXICO.WE8DEC"  
  wq
```

8.- Dar de baja la Base de Datos de manera normal utilizando las sentencias SHUTDOWN NORMAL o SHUTDOWN IMMEDIATE. Es importante mencionar que no





se debe de utilizar la sentencia SHUTDOWN ABORT para dar de baja la Base de Datos, ya que esto ocasiona graves problemas debido a que los archivos de datos necesitarían proceso de restauración.

IV.5.1.2 INSTALACIÓN DEL SOFTWARE DE ORACLES

- 1.- Se debe de ejecutar el script (oraInstRoot.sh) con el usuario de 'root'.

```
$ su -  
$ cd /home/oracle  
$ ORACLE_SID=PPAY; export ORACLE_SID  
$ ./oraInstRoot.sh
```

- 2.- Verificar las nuevas variables de ambiente de Oracle 8i usando los comandos 'set y 'grep de unix, para ello se debe de abrir una sesión de Unix y conectarse con la cuenta del usuario oracle.

```
$ su - oracle  
$ cd ~/cfg /home/oracle/cfg  
$ PATH=/usr/bin:./bin:/usr/ucb:/etc/oracle/ora8/bin; export PATH  
$ . oracle_profile.ora8  
$ set | grep -i oracle
```

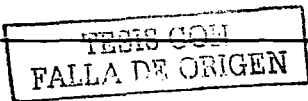
Verificar que las siguientes variables, se encuentren definidas de la siguiente manera:

```
LOGNAME=oracle  
ORACLE_BASE=/oracle  
ORACLE_DOC=/oracle/doc  
ORACLE_HOME=/oracle/ora8  
ORACLE_PATH=/oracle/ora8/bin:/usr/bin:/usr/sbin:/opt/bin  
ORACLE_REPORT=/home/oracle/reports  
ORACLE_SID=ppay  
ORA_NLS33=/oracle/ora8/migrate/nls/admin/data
```

La variable LD_LIBRARY_PATH debe de hacer referencia al directorio '/oracle/ora8/lib'.

- 3.- Verificar que el cdrom del producto de Oracle 8i Enterprise 32 bit se encuentra cargado en la unidad de cd-rom del Dominio2 en la siguiente ruta: /cdrom. Conectarse con el usuario oracle para iniciar el Instalador Universal de Oracle:

```
$ su - oracle
```





```
$ cd /cdrom/cdrom0
```

NOTA: Si se va a usar una sesión de X-windows, se debe de definir las variables de ambiente de DISPLAY para que no mande problemas de errores de sesiones, de la siguiente manera:

```
$DISPLAY=host:0.0; export DISPLAY  
$ xhost +  
$ ./runInstaller -silent -responseFile ~/prepay.rsp
```

donde: host es la dirección IP de la computadora donde se ejecutará el instalador

Nota: Este paso lleva más de una hora ya que esta instalando el producto de la nueva versión, el mensaje que envía al finalizar es el siguiente: **ORACLE INSTALLATION SUCCESFULL.**

4.- Al finalizar esta tarea abrir una sesión de unix y conectarse con la cuenta de root para ejecutar el script 'root.sh' de la siguiente manera:

```
$ su - root  
$ cd /oracle/ora8  
$ ./root.sh  
$ exit
```

IV.5.1.3 MIGRANDO LA BASE DE DATOS ORACLE7

1.- Con las variables de ambiente haciendo referencia al \$ORACLE_HOME de Oracle8i, para ello se utilizan los comandos de 'set' y 'grep' de unix

```
$ su - oracle  
$ cd ~/cfg  
$ export PATH=/usr/bin:./bin:/usr/ucb:/etc:/oracle/ora8/bin  
$ oracle_profile.ora8  
$ set | grep -i oracle
```

Verificar que las siguientes variables, se encuentren de la siguiente manera:

```
LOGNAME=oracle  
ORACLE_BASE=/oracle  
ORACLE_DOC=/oracle/doc  
ORACLE_HOME=/oracle/ora8  
ORACLE_PATH=/oracle/ora8/bin:/usr/bin:/usr/sbin:/opt/bin
```



```
ORACLE_REPORT=/home/oracle/reports
ORACLE_SID=ppay
ORA_NLS33=/oracle/ora8/migrate/nls/admin/data
```

La variable LD_LIBRARY_PATH debe de hacer referencia al directorio '/oracle/ora8/lib'.

2.- Ejecutar la utileria migprep indicando el nuevo Oracle Home (Oracle8i) y el anterior Oracle Home (Oracle7).

```
$ ORACLE_HOME/bin/migprep nuevo_oracle_home viejo_oracle_home
$ ORACLE_HOME/bin/migprep /oracle/ora8 /oracle/ora734
```

El script de 'migprep' al ser ejecutado manda un mensaje haciendo referencia a las variables de ambiente las cuales se deben de validar a continuación se muestra un ejemplo.

```
Ejemplo:
NEW_ORACLE_HOME=/oracle/ora8
OLD_ORACLE_HOME=/oracle/ora734
is this ok [y]/n? y
```

3.- Modificar las siguientes variables de ambiente para que hagan referencia al Oracle Home del ambiente Oracle7.

```
$ cd ~/cfg
$ export PATH=/usr/bin:./bin:/usr/ucb:/etc:/oracle/ora734/bin
$ . oracle_profile
$ set | grep -i oracle
```

Verificar que las siguientes variables, se encuentren de la siguiente manera:

```
LOGNAME=oracle
ORACLE_BASE=/oracle
ORACLE_DOC=/oracle/doc
ORACLE_HOME=/oracle/ora734
ORACLE_PATH=/oracle/ora734/bin:/usr/bin:/usr/sbin:/opt/bin
ORACLE_REPORT=/home/oracle/reports
ORACLE_SID=ppay
ORA_NLS32=/oracle/ora734/ocommon/nls/admin/data
ORA_NLS33=/oracle/ora734/migrate/nls/admin/data
```

La variable LD_LIBRARY_PATH debe de hacer referencia al directorio '/oracle/ora734/lib'.



4.- Verificar que exista suficiente espacio en el tablespace de SYSTEM para que no tener problemas en la migración para esta validación se realiza lo siguiente:

```
$ mkdir /tmp/dbmigrate
$ echo $NLS_LANG
Nota: Verificar que la variable NLS_LANG este correcta:
```

```
$ mig CHECK_ONLY=TRUE SPOOL="/tmp/dbmigrate/oraclemigchk.log"
```

Al finalizar este script será mostrado el espacio estimado para el catálogo de la version Oracle8i y además el espacio libre del tablespace de SYSTEM, si el valor requerido por el catálogo de Versión 8 es mayor que el espacio libre, será necesario agregarle mas espacio al tablespace.

Salida del comando:

```
v8 catalog space requirement: 12902400
free space found: 255467520
```

5.- Verificar que la Base de Datos no esté corriendo de lo contrario, darla de baja, de la siguiente manera:

```
$svrmgrl
SVRMGRL> shutdown immediate
```

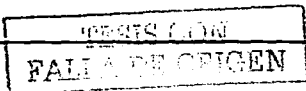
6.- Ejecutar la utileria de migración de Oracle y definir el parametro de MULTIPLIER con un valor de 30.

```
..mig MULTIPLIER=30 SPOOL="/tmp/dbmigrate/migrate.log"
```

7.- Cuando la utileria de migración haya finalizado exitosamente, es necesario cambiar la referencia de las variables de ambiente al nuevo oracle_profile y al ambiente de Oracle8i. Para validar las variables se utilizan los comandos 'set' y 'grep' de unix.

```
..$ORACLE_HOME/ocommon/nls/admin/data
```

```
$ ORA_NLS33=/oracle/ora8/ocommon/nls/admin/data; export ORA_NLS33
$ cd -/cfg
$ edit oracle_profile.oracle8
/NLS33=
s/migrate/ocommon/
```





wq

```
$ cd -/cfg
$ export PATH=/usr/bin:./bin:/usr/ucb:/etc:/oracle/ora8/bin
$ . oracle_profile.oracle8
$ set | grep -i oracle
```

Verificar que las siguientes variables se encuentren de la siguiente manera:

```
LOGNAME=oracle
ORACLE_BASE=/oracle
ORACLE_DOC=/oracle/doc
ORACLE_HOME=/oracle/ora8
ORACLE_PATH=/oracle/ora8/bin:/usr/bin:/usr/sbin:/opt/bin
ORACLE_REPORT=/home/oracle/reports
ORACLE_SID=ppay
ORA_NLS32=/oracle/ora8/ocommon/nls/admin/data
ORA_NLS33=/oracle/ora8/ocommon/nls/admin/data
```

La variable LD_LIBRARY_PATH debe de hacer referencia al directorio '/oracle/ora8/lib'.

8.- Copiar los archivos de configuración y de conversión de Oracle 7 a Oracle8 y renombrar el archivo de sqlnet.ora, esto de la siguiente manera:

```
$ cd /oracle/ora734/dbs
$ cp -p convppay.dbf /oracle/ora8/dbs
$ cp -p initppay.ora /oracle/ora8/dbs
$ cp -p configppay.ora /oracle/ora8/dbs
$ cp -p orapwppay /oracle/ora8/dbs
$ cd /oracle/ora734/network/admin
$ cp -p listener.ora /oracle/ora8/network/admin
$ cp -p tnsnames.ora /oracle/ora8/network/admin
```

9.- Renombrar los archivos de control de la Base de Datos de versión 7, en necesario revisar el archivo de configuración (/oracle/ora8/dbs/configppay.ora) para determinar la localización de estos archivos:

```
$ cd /oracle/ora8/dbs
$ more /oracle/ora8/dbs/configppay.ora
#
# $Header: cnfg.orc 1.1 95/02/27 12:14:25 wyim Osd<unix> $ Copyr (c) 1992 Oracle
#
# cnfg.ora - instance configuration parameters
```



```
control_files          = (/u03/oradata/ppay/ctrl1ppay.ctl,
                        /u27/oradata/ppay/ctrl2ppay.ctl,

# Below for possible futura use...
#init_sql_files       = (?/dbs/sql.bsq,
#                       ?/rdbsms/admin/catalog.sql,
#                       ?/rdbsms/admin/expview.sql)
background_dump_dest  = /oracle/admin/ppay/bdump
core_dump_dest        = /oracle/admin/ppay/cdump
user_dump_dest        = /oracle/admin/ppay/udump
#log_archive_dest     = /oracle/admin/ppay/arch/arch.log
#db_block_size        = <blocksize>
db_name               = ppay
```

```
$ cd /u03/oradata/ppay
```

```
$ ls
```

```
ctrl1ppay.ctl  idata061ppay.dbf  iindex051ppay.dbf  itrain01ppay.dbf
idata031ppay.dbf  idata071ppay.dbf  iindex061ppay.dbf  iuser01ppay.dbf
idata051ppay.dbf  iindex031ppay.dbf  iindex071ppay.dbf  sysppay.dbf
```

```
$ mv ctrl1ppay.ctl ctrl1ppay.ctl.734
```

```
$ ls
```

```
ctrl1ppay.ctl.734  idata061ppay.dbf  iindex051ppay.dbf  itrain01ppay.dbf
idata031ppay.dbf  idata071ppay.dbf  iindex061ppay.dbf  iuser01ppay.dbf
idata051ppay.dbf  iindex031ppay.dbf  iindex071ppay.dbf  sysppay.dbf
```

```
$ cd /u27/oradata/ppay
```

```
$ ls
```

```
ctrl2ppay.ctl  idata062ppay.dbf  iindex052ppay.dbf
idata032ppay.dbf  idata072ppay.dbf  iindex062ppay.dbf
idata052ppay.dbf  iindex032ppay.dbf  iindex072ppay.dbf
```

```
$ mv ctrl2ppay.ctl ctrl2ppay.ctl.734
```

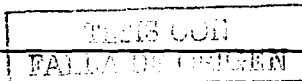
```
$ ls
```

```
ctrl2ppay.ctl.734  idata062ppay.dbf  iindex052ppay.dbf
idata032ppay.dbf  idata072ppay.dbf  iindex062ppay.dbf
idata052ppay.dbf  iindex032ppay.dbf  iindex072ppay.dbf
```

10.- En la versión de Oracle 8i algunos parámetros del archivo de inicialización son cboletes, esto es debido a diversas razones:

- 1.- La versión del manejador. Ahora se posee una superior
- 2.- Algunos parámetros únicamente cambian de nombre, tal es el caso del db_writer que ahora en Oracle 8i recibe el nombre de db_writer_processes.

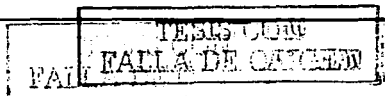
Es por ello que tienen que modificarse. Los parámetros que son modificados son los siguientes:



| Descripción de cambios | |
|-----------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| /var/opt/oracle/oratab | <p>Cambiar todas las referencias de ora734 a ora8 Cambiar la línea 'ppay:/oracle/ora733:Y' a 'ppay:/oracle/ora8:Y' Cambias todas las líneas que empiecen con 'ebu' al final por perles ':N' Por ejemplo: cambiar la línea. 'ebu:/oracle/ora734:Y' a 'ebu:/oracle/ora8:N'</p> |
| /oracle/ora8/network/admin/listener.ora | Cambiar todas las referencias de 'ora734' a 'ora8' |
| /oracle/ora8/dbs/initppay.ora | <p>Cambiar todas las referencias de 'ora734' a 'ora8'</p> <p>Cambiar de 'compatible = 7.3.0.0.0' a 'compatible = 8.1.0.0.0'</p> <p>Agregar las siguientes líneas } agregar al principio</p> <pre> db_domain=world hash_multiblock_io_count = 1 db_writer_processes = 10 </pre> <p>Remove las líneas de los siguientes parámetros o comentar las líneas:</p> <pre> Checkpoint_process log_simultaneous_copies log_small_entry_max_size sequence_cache_entries log_archive_buffers log_archive_buffer_size unlimited_rollback_segments </pre> <p>NOTA: Los siguientes parámetros son obsoletos en Oracle 8.1.6</p> <pre> db_writers async_write spin_count sort_read_fac sequence_cache_hash_buckets </pre> |
| /oracle/ora8/dbs/configppay.ora | Cambiar todas las referencias de 'ora734' a 'ora8' |
| /home/oracle/cfg/oracle_profile | <p>Cambiar las referencias de TNS_ADMIN de la siguiente manera:</p> <p>De: TNS_ADMIN=/oracle/ora734/network/admin/ A: TNS_ADMIN=\$ORACLE_HOME/network/admin</p> |

IV.5.1.4 CONVERSION DE LA BASE DE DATOS.

1.- Moverse al siguiente directorio: \$ORACLE_HOME/rdbms/admin.





2.- Iniciar una sesión en server manager y conectarse como usuario INTERNAL.

```
$svrmgrl
```

```
SVRMGR> CONNECT INTERNAL
```

3.- Iniciar la instancia de Base de Datos Oracle8i sin montar la Base de Datos. Esto se realiza con el comando

```
SVRMGR> STARTUP NOMOUNT;
```

4.- La creación de nuevos archivos de control para Oracle8i y conversión de los encabezados de los archivos de datos de todos los tablespaces que se encuentran en línea, se realiza ejecutando el siguiente comando:

```
SVRMGR> ALTER DATABASE CONVERT;
```

La exitosa ejecución de este comando implica un punto de no retorno a la Base de Datos Oracle7. Para ello sería necesario restaurarla del último respaldo disponible.

5.- Abrir la Base de Datos.

```
SVRMGR> ALTER DATABASE OPEN RESETLOGS;
```

Cuando la Base de Datos es abierta todos los segmentos de rollback que están en línea, son convertidos al formato de Oracle8i.

6.- Al finalizar la conversión de la Base de Datos es necesario ejecutar una de las utilerías de Oracle para la nueva versión, esto se realiza de la siguiente manera:

```
SVRMGR> @utltp.sql
```

7.- Ejecución del programa de conversión U0703040.SQL. Este programa crea y altera algunas tablas de sistema; además de borrar al usuario MIGRATE. También ejecuta los programas CATALOG.SQL Y CATPROC.SQL.

8.- Dar de baja la Base de Datos Oracle8i con el comando SHUTDOWN NORMAL o SHUTDOWN IMMEDIATE. La ejecución de este comando ayuda a que la Base de Datos realice algunas actividades de limpieza, con lo cual se garantiza la consistencia de la nueva Base de Datos Oracle8i.



9.- Modificar el Oracle profile con las variables de ambiente de la nueva versión y definir las ligas para esta versión:

```
$ cd ~/cfg
$ rm -f oracle_profile
$ mv oracle_profile.oracle8 oracle_profile
$ cd
$ . profile
$ ln -s /oracle/ora8/lib/libcintsh.so.8.0 /oracle/ora8/lib/libcintsh.so.1.0
```

IV.5.2 ACTUALIZACIÓN DE APLICACIÓN.

Una vez finalizada la migración de la versión de Oracle, se procede a la migración de la versión de la aplicación, para ello es necesario preparar el ambiente y ejecutar los programas proporcionados por el proveedor. Esto se hace de la siguiente manera:

Nota: Esta tarea se realiza con la cuenta de la aplicación.

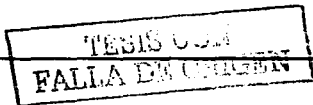
```
$ su - pp
$ cd $PPHOME/etc
$ chmod 755 prepay_profile
$ cp -p prepay_profile prepay_profile.60
$ cp -p $HOME/.profile $HOME/.profile.before70
```

Editar el archivo prepay_profile y cambiar la definición de la variable ORACLE_HOME. Ejecutarlo para que las variables de ambiente, apunten a la nueva ruta de oracle y así poder conectarse a la Base de Datos de la nueva versión, esto de la siguiente manera:

```
$ edit prepay_profile
s/ORACLE_HOME=/oracle/ora8
wq
$ cd
$ . profile
```

Es necesario compilar todos los triggers de la Base de Datos, en caso de que se encuentren inválidos será necesario reconstruirlos. Para ello se ejecutan los siguientes comandos:

```
$su - pp
$ sqlplus pp/<oracleuserpassword>
```





```
SQL> alter trigger pp.CELL_SITE_INS compile;
SQL> alter trigger pp.CELL_SITE_UPD compile;
SQL> alter trigger pp.TAX_PLAN_DEL compile;
SQL> alter trigger pp.TAX_PLAN_UPD compile;
```

```
SQL> select object_name, object_type
       from dba_objects
       where owner='PP' and status='INVALID'
       order by object_type;
```

```
SQL> exit
```

Una vez que se han realizado estos pasos ya se puede ejecutar el programa proporcionado por el proveedor de la siguiente manera:

```
$ cd /provision/work
$ ./ppupgrade.ksh 600
```

Ver detalle en APENDICE I IV.5.2.1

IV.5.3 ACTUALIZACIÓN DE LA BASE DE DATOS PARA VERSIÓN 7 DE APLICACIÓN.

Es necesario realizar algunos cambios dentro de la Base de Datos, estos cambios se deben a que la aplicación en su nueva versión trabaja con algunas modificaciones sobre algunos campos de varias tablas, estos cambios los realiza un programa proporcionado por el proveedor pero en ciertos casos fue necesario que se realizaran tareas para automatizar y agilizar dichos cambios

Actualización de la tabla de fichas.

Debido a que esta tabla contiene demasiada información aproximadamente 50 millones de registros, por lo tanto no es factible realizar el comando ALTER sobre la tabla, es por ello que es necesario implantar otra forma para realizar las modificaciones. Esto consiste en descargar los datos de las tablas en archivos planos, realizar el cambio de estructura de la tabla sin registros y por ultimo cargar los datos. Se realiza de este modo ya que la utilidad export/import de Oracle no es capaz de soportar archivos físicos de un tamaño mayor a 2 Gb.

TRABAJANDO
FALLA DE ORIGEN



A continuación se muestra la forma en que se realiza esta actividad.

1.- Descarga de la tabla de fichas. Para realizar la descarga es necesario ejecutar 9 programas los cuales descargan en archivos planos la tabla por períodos de fechas, el primer programa descarga los datos correspondientes al año 1999, los siguientes tres los datos del año 2000, los otros 4 datos del año 2001 y el último programa los datos de lo que va del año 2002. Estos programas se ejecutan de la siguiente manera:

```
$ cd /home/oracle/FICHAS_MIGRA_V7/UNLOADS
$ nohup fichas_A.sh &
$ nohup fichas_B.sh &
$ nohup fichas_C.sh &
$ nohup fichas_D.sh &
$ nohup fichas_E.sh &
$ nohup fichas_F.sh &
$ nohup fichas_G.sh &
$ nohup fichas_H.sh &
```

Ver detalle en **APENDICE I IV.5.3.1**

2.- Modificar la estructura de la tabla de fichas y prepararla para la carga de datos. Para ello es necesario que el tablespace donde se aloja la tabla no tenga problemas de particionamiento para evitar que la carga de datos sea lenta. Después de verificar lo anterior se procede a crear la tabla con la nueva estructura de campos usando los siguientes programas:

```
$ sqlplus system/xxxxx
SQL> alter tablespace IDATA09 coalesce;
SQL> exit
```

Nota: En el `ldata09` es donde reside la tabla de fichas

```
$ cd /home/oracle/FICHAS_MIGRA_V7/LOAD
```

```
$ sqlplus
USER: PP/<PASSWORD>
```

```
SQL> @crea_fichas_inv_no_index.sql
```

```
      DROP TABLE FICHAS CASCADE CONSTRAINTS
```



```
ERROR at line 1:  
ORA-00942: table or view does not exist
```

```
Table created.
```

```
SQL> @crea_fichas_inv_no_index.sql  
Table dropped.  
Table created.
```

Ver detalle en APENDICE I IV.5.3.2

3.- En este momento ya se encuentra creada la tabla de fichas pero no contiene los constraints (PHOCDIN0_T, PHOCDINV) ni los índices (PHOCDIN1_T, PHOCDIN3_T, PHOCDIN4_T) los cuales impactarían en la carga de datos por lo tanto serán creados al final de la carga. Para verificarlo se ejecutan dos programas en los que el resultado no debe de contener ninguno de estos constraints ni índices; esto de la siguiente manera :

```
Ssqlplus system/xxxxx  
SQL> @check_constraints.sql  
SQL> @check_index.sql
```

Ver detalle en APENDICE I IV.5.3.3

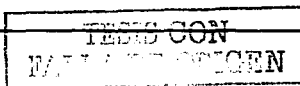
4.- Ya que no existen constraints e índices se procede a realizar la carga de datos en forma paralela ejecutando el siguiente programa:

```
S cd /home/oracle/FICHAS_MIGRA_V7/LOAD  
S nohup load_fichass.sh &
```

Ver detalle en APENDICE I IV.5.3.4

5.- Sólo falta revisar los archivos .log los cuales, contienen el resultado de la carga.

Ver detalle en APENDICE I IV.5.3.5





6.- Validar que la tabla se haya cargado completa para proceder a crear los índices de esta tabla de la siguiente manera:

```
$ sqlplus
```

```
Enter user-name: pp  
Enter password:
```

```
SQL> select count(*) from fichas;
```

```
  COUNT(*)  
-----  
42,606,096
```

Para la creación de los índices solo es necesario ejecutar un programa:

```
$ sqlplus  
Enter user-name: pp  
Enter password: xxxxxx
```

```
SQL> @add_indexes.sql
```

Ver detalle en APENDICE I IV.5.3.6

Por último, sólo es necesario revisar que los índices se encuentren creados sobre la tabla de fichas, como se muestra a continuación;

```
$ sqlplus  
Enter user-name: system  
Enter password: xxxxxx
```

```
SQL> select index_name, table_name, tablespace_name from dba_indexes  
       where table_name = 'FICHAS' AND OWNER = 'PP';
```

| INDEX_NAME | TABLE_NAME | TABLESPACE_NAME |
|------------|------------|-----------------|
| PHOCDI0_T | FICHAS | IINDEX09 |
| PHOCDI1_T | FICHAS | IINDEX09 |
| PHOCDI3_T | FICHAS | IINDEX09 |
| PHOCDI4_T | FICHAS | IINDEX09 |

```
SQL> commit;
```

```
Commit complete.
```

TESIS CON
FALLA DE ORIGEN



```
SQL> select CONSTRAINT_NAME, TABLE_NAME from user_constraints  
where TABLE_NAME = 'FICHAS' ;
```

| CONSTRAINT_NAME | TABLE_NAME |
|-----------------|------------|
| SYS_C005793 | FICHAS |
| SYS_C005794 | FICHAS |
| SYS_C005795 | FICHAS |
| SYS_C005796 | FICHAS |
| SYS_C005797 | FICHAS |
| SYS_C005798 | FICHAS |
| SYS_C005799 | FICHAS |
| SYS_C005800 | FICHAS |
| SYS_C005801 | FICHAS |
| SYS_C005802 | FICHAS |
| SYS_C005803 | FICHAS |
| SYS_C005804 | FICHAS |
| SYS_C005805 | FICHAS |
| SYS_C005806 | FICHAS |
| SYS_C005807 | FICHAS |
| SYS_C005808 | FICHAS |
| SYS_C005809 | FICHAS |
| PHOCDIN0_T | FICHAS |
| PHOCDIN0 | FICHAS |

19 rows selected.

```
SQL> exit
```

El último cambio que se debe de generar a la Base de Datos es habilitar la opción de Archive tecleando los siguientes comandos:

```
Ssu - oracle  
Ssvrmgrl
```

```
SVRMGRL> shutdown  
SVRMGRL> startup mount  
SVRMGRL> alter database archive log on;  
SVRMGRL> alter database open;
```

IV.5.4 REENRUTAMIENTO DE TRÁFICO HACIA DOMINIO2.

Hasta este momento el proceso de migración se ha ejecutado de manera exitosa, se tiene en el Dominio2 la Base de Datos migrada a versión Oracle 8i, junto con la nueva versión de la aplicación de Prepago.



El siguiente paso, consiste en reenrutar cada una de las centrales celulares hacia el Dominio2, quien en este momento se encuentra listo para atender el tráfico.

Este reenrutamiento es labor de la gente de centrales y consiste básicamente en ejecutar comandos para apuntar la central celular hacia un nuevo destino; que en este caso es el Dominio2.

De manera simultánea, se realiza un reporte en el Dominio1 de todas las llamadas ingresadas las cuáles serán llevadas al Dominio2 y aplicadas; para con ello lograr la transparencia para el usuario y evitar las pérdidas financieras que el proceso de migración, hubiera implicado.

Ver detalle APÉNDICE I IV.5.4.1

IV.5.5 PRUEBAS DE TRÁFICO.

Una vez que el tráfico celular está enrutado hacia el Domino2 y éste es el encargado de realizar el cobro de las llamadas en tiempo real; se realizan pruebas de tráfico para validar que las llamadas se enlacen de forma correcta y que los cobros entre una y otra sean los adecuados.

Estas pruebas consisten principalmente en realizar llamadas en diversas modalidades; esto es:

- Llamadas locales entre celulares
- Llamadas de larga distancia entre celulares
- Llamadas locales entre teléfonos fijos y celulares
- Llamadas de Larga distancia entre teléfonos fijos y celulares

Se posee una matriz de casos de tráfico la cual contiene pruebas de las llamadas anteriormente mencionadas, una vez que la matriz ha sido cubierta en su totalidad, y que no existen problemas de tasación; se dice que la aplicación esta funcionando de manera correcta.

Todo este tipo de pruebas son realizadas por gente encargada de la administración de la aplicación, únicamente el personal de Base de Datos y Sistema Operativo juega un papel de apoyo, en caso de que sea necesario.



IV.5.6 APERTURA DE IVR'S, CARGA DE FICHAS Y SISTEMA DE BANCOS

La apertura de los IVR's consiste en dar de alta cada uno de estos equipos para que establezcan sus conexiones hacia el Dominio2 y el ingreso de ficha puede funcionar; esta tarea, se realiza de la siguiente manera:

- Conectarse a cada IVR con el usuario root y darlo de alta.

SunOS 5.8

Login: root

Password: XXXXXX

\$

Una vez abiertos los IVR's, se realizan algunas pruebas de carga de fichas. Estas pruebas consisten en que teléfonos pertenecientes a la aplicación de prepago puedan ingresar sin problema sus fichas y con ello aumentar su saldo. Para ello realizan una marcación especial y retroalimentando de información al IVR por medio del teclado se realiza la carga de la ficha al teléfono.

Por ultimo se contacta a la gente de aplicación de Bancos, quienes son los encargados de dar de alta su aplicación, obviamente el administrador de Base de Datos se debe de cerciorar que el listener encargado de atender esta aplicación se encuentre dado de alta.

- Verificación de Listener Bancos

- Abrir una sesión unix con el usuario Oracle en Dominio2

- SunOS 5.8

- Login: root

- Password: XXXXXX

- \$

- Verificar listener

- ps -ef | grep LSNR1

- Alta de Listener

- \$!snrcctl start LSNR1

TESIS CON
FALLA DE ORIGEN



IV.6 TAREAS POSTERIORES AL PROCESO DE MIGRACIÓN

IV.6.1 RESPALDO TOTAL DE BASE DE DATOS VIA HOT BACKUP

Una vez que se ha realizado la migración y que la aplicación esta funcionando correctamente es necesario realizar un respaldo en caliente de la Base de Datos de manera inmediata para proteger toda la plataforma. Este respaldo se realiza tomando nuevamente los programas que fueron utilizados anteriormente en la ejecución del primer respaldo previo a la migración.

IV.6.2 RESPALDO TOTAL DE LA BASE DE DATOS VÍA EXPORT

Para realizar el respaldo de la Base de Datos vía export es necesario generar scripts para ejecutar esta tarea de manera automática, debido a las características de la Base de Datos este respaldo se realizará sólo a nivel objeto el cual respaldara todas las tablas del usuario de la aplicación de esta manera la información a nivel lógico se tendrá en archivos de manejo de oracle.

La sentencia utilizada para la generación del respaldo vía export es la siguiente:

- Creación de pipe

```
/usr/bin/mknod pipenombre_tabla p
```

- Lectura de información

```
cat pipenombre_tabla | compress > nombre_tabla.dmp.Z &
```

- Export

```
/oracle/ora734/bin/exp          user/password          tables=nombre_tabla  
file=pipenombre_tabla          buffer=10000000        indexes=y  
logs=respaldo.nombre_tabla.log rows=y consistent=y grants=y compress=y
```

TECNOLOGIA
FALLA DE ORIGEN



IV.6.4 CAMBIO DE PASSWORD USUARIO OUTLN

El usuario OUTLN es creado automáticamente durante la instalación de Oracle8i, este usuario tiene los privilegios de DBA por lo cual se debe de cambiar su password, o en su defecto eliminar el usuario. Para ello es necesario conectarse a la Base de Datos y dentro del prompt de SQL*PLUS ejecutar la siguiente sentencia con el password correspondiente:

```
Sqlplus > ALTER USER OUTLN IDENTIFIED BY *****;
```

IV.6.5 CONSISTENCIA DE LOS CONSTRAINTS DE FECHA DE ORACLE.

Después de la migración, es necesario revisar la consistencia de todos los objetos de la Base de Datos, entre los más importantes se encuentran los constraints referentes al manejo de las fechas. Para este problema Oracle proporciona un programa el cual verifica la consistencia de estos objetos, si encuentra objetos dañados los marca como inválidos para posteriormente analizarlos. Para correr este programa es necesario realizar lo siguiente:

```
$ cd /oracle/ora8/rdbms/admin  
$ svrmgrl  
SVRMGRL> connect / as sysdba  
SVRMGRL> startup  
SVRMGRL> @utlconst.sql  
exit
```

La ejecución del programa utlconst.sql, genera un archivo de mensajes con el nombre de utlresult.log, el cual contiene todos los constraints inválidos. Cabe mencionar que el programa no corrige los constraints por lo tanto es necesario borrarlos y reconstruirlos.

TESIS CON
FALLA DE ORIGEN



CAPITULO V

MANTENIMIENTO DE LA BASE DE DATOS

TESIS CON
FALSA DE ORIGEN

TEXAS COM
FALL AND WINTER



MANTENIMIENTO A LA BASE DE DATOS.

V.1 INTRODUCCIÓN AL MANTENIMIENTO Y AFINACIÓN DE BASES DE DATOS ORACLE.

El mantenimiento de una Base de Datos es una tarea larga e importante para el buen funcionamiento del sistema. La tarea de mantenimiento consiste en monitorear todos los acontecimientos dentro de la Base de Datos, los cuales son analizados para posteriormente aplicar los cambios necesarios en la arquitectura del manejador; a esta última tarea se le denomina afinación.

La afinación es responsabilidad de cada una de las áreas involucradas, es decir, Sistema Operativo, Base de Datos y Aplicación; lo cual significa que deben de pensar en conjunto acerca de la afinación e implantar las mejoras al sistema correspondientes a cada uno.

Es importante mencionar que el monitoreo y la afinación no se debe de iniciar si no se tiene una clara idea del resultado que se pretende alcanzar. Por otro lado este proceso de monitoreo y afinación es iterativo, es decir, son tareas que se deben de realizar día con día, para lograr el máximo desempeño del sistema.

Una buena estrategia de afinación consiste en una constante investigación y monitoreo del sistema, basada en aumentar y disminuir parámetros de la instancia y la Base de Datos en el archivo de configuración.

El monitoreo y la afinación deben de iniciar por:

*EL DISEÑO
LAS APLICACIONES
LA MEMORIA
ENTRADAS Y SALIDAS
CONTENCION*

Es importante considerar que herramientas de monitoreo del sistema provistas por el Sistema Operativo pueden ser usadas para verificar el desempeño de eventos tales como:

TESIS CON
FALLA DE ORIGEN



- Paginación y Swapeo
- Espacio de Swap
- Contención de recursos de la máquina en aplicaciones No-Oracle y en diversos procesos.

Dentro del monitoreo y afinación de la instancia de Oracle algunos de los objetivos que se desea alcanzar con la afinación son:

- Accesar al menor número de bloques
- Tratar de que las operaciones necesarias puedan ser efecutadas en memoria
- Compartir el código de los comandos (no hacer distintas versiones para un comando)
- Hacer respaldos y estrategias de recuperación lo más rápido posible.

V.2 HERRAMIENTAS DE MONITOREO DE BASE DE DATOS

Para lograr los objetivos antes descritos, Oracle proporciona herramientas que se mencionan, a continuación:

ORACLE ENTERPRISE MANAGER PERFORMANCE PACK

El Oracle Performance Pack, es un poderoso conjunto de herramientas de monitoreo del performance de la Base de Datos que junto con el Enterprise Manager provee información gráfica en tiempo real.

ORACLE EXPERT

Este es un asistente en la configuración de la Base de Datos autorizado para optimizar el performance, brindando una colección y evaluación de las características de desempeño de las Bases de Datos existentes.

ORACLE LOCK MANAGER

Autorizada para monitorear los bloqueos, con mecanismos que previenen interacciones destructivas entre usuarios accesando, a los mismos recursos. Con ventanas que despliegan una lista multi-columnas que incluyen un registro por cada bloqueo actual en la Base de Datos.



ORACLE PERFORMANCE MANAGER

Propvee la habilidad de monitorear el performance de la Base de Datos en tiempo real; por medio de docenas de planos predefinidos para desplegar una gran variedad de estadísticos del performance de la Base de Datos con respecto a los usuarios, tablespaces, redo logs, buffers, caches e I/O.

ORACLE TOP SESSIONS

Oracle Top Sessions permite monitorear como se usan las conexiones por usuarios, y los recursos utilizados por cada sesión en tiempo real de la Instancia.

ORACLE TRACE

Se puede monitorear con Oracle Trace, el performance por colecciones de datos acerca de eventos que pasen en las aplicaciones.

ORACLE TABLESPACE MANAGER

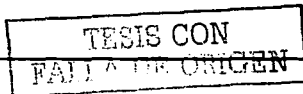
Autorizada para monitorear el almacenamiento de la Base de Datos. Despliega una vista con información de los tablespaces usados y usa la característica de coalescing, para unir los bloques adyacentes.

V.3 MANEJO DE MENSAJES DEL MANEJADOR

Para iniciar el monitoreo de la Base de Datos, es necesario conocer primero los archivos donde el manejador de Base de Datos arroja todos los mensajes sobre sus actividades, estos archivos son:

V.3.1 Oracle Alert

La gran mayoría de las actividades de la Base de Datos residen en el archivo alert.log en forma cronológica, para cada una de las instancias. Es recomendable verificar constantemente el contenido de este archivo en busca de eventos y errores. Además, se recomienda borrarlo periódicamente o administrar su información en forma de histórico.





Si ocurre un error cuando la instancia se encuentra corriendo, los mensajes de error son escritos en este archivo. Los mensajes que arroja la Base de Datos y que requieren mayor atención son los errores internos los cuales son registrados como un **ORA-600** y errores de corrupción de datos **ORA-1578**.

Por otro lado, el archivo `alert.log`, almacena todas las actividades ejecutadas sobre la estructura y parámetros de la Base de Datos; es decir, sentencias como `CREATE DATA BASE`, `STARTUP`, `SHUTDOWN`, `ARCHIVE LOG` y `RECOVER` son almacenadas aquí.

Cuando se inicializa la instancia si el archivo de `alert` no está creado el Servidor de Oracle crea uno por default.

Este archivo se encuentra en la siguiente ruta: `/oracle/ora8/rdbms/log/pppayalert.log`

Ver Detalle en APENDICE II V.3.1.1

V.3.2 Archivos de Trace.

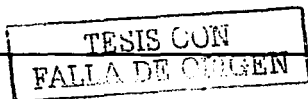
Oracle genera archivos llamados `trace`, en los cuales son almacenados errores comunes relacionados con procesos de `background`. Todos estos archivos son encontrados en el parámetro conocido como `BACKGROUND_DUMP_DEST`. Los archivos `trace` generados por los usuarios se encuentran en la dirección indicada por el parámetro `USER_DUMP_DEST` estos parámetros se encuentran definidos en el archivo de `init.ora` el cual se localiza en la siguiente ruta: `/oracle/ora8/dbs/initppay.ora`,

```
$ more /oracle/ora8/dbs/initppay.ora
```

Ver Detalle en APENDICE II V.3.2.1

V.3.3 Manejo de Eventos.

Los eventos que ocurren sobre la Base de Datos pueden ser monitoreados a través de las vistas dinámicas `v$`, las cuales se van modificando mediante los sucesos que ocurren en la instancia; los eventos más comunes son los siguientes:



Procesos de Esperas en el Buffer. El proceso de DBWR tarda mucho tiempo para realizar sus escrituras por lo tanto es necesario incrementar el número de checkpoint.

Lecturas secuenciales en los Archivos de Datos. Existe contención en el I/O , por lo tanto es necesario distribuir la carga

Para monitorear estos eventos es necesario ejecutar queries sobre las siguientes vistas:

V\$SESSION_WAIT, V\$SYSTEM_EVENT, V\$SESSION_EVENT.

Para llevar estadísticas de tiempo se inicializan parámetros en el init.ora como TIMED_STATISTICS=TRUE o se pueden llevar a cabo dinámicamente ALTER SYSTEM SET TIMED_STATISTICS= TRUE / FALSE.. O bien, para el cálculo de estadísticas dentro de la Base de Datos, se utiliza el comando ANALYZE, el cual nos brinda información referida a Num_rows, blocks (high water mark), empty_blocks, avg_space, chain_cnt, avg_row_len

Ver Detalle en APENDICE II V.3.3.1

V.4 MANEJO DE UTILERIAS, VISTAS DINÁMICAS Y DE DICCIONARIO PARA EL MONITOREO DE BASE DE DATOS

V.4.1 Vistas Dinámicas y de Diccionario de Datos.

Oracle despliega todas las estadísticas del sistema en la vista **V\$SYSSTAT**, y usa muchas otras vistas para el monitoreo y performance. Se puede consultar éstas vistas para buscar los totales acumulativos desde que la instancia se ha iniciado pero no resulta tan útil si la instancia, es raramente dada de baja, por que las estadísticas pueden cubrir un largo período y tener un pequeño significado.

Las tablas dinámicas de monitoreo son continuamente actualizadas mientras la Base de Datos está abierta y en uso. Estas tablas dinámicas son identificadas por el prefijo V\$

En las vistas se encuentra información sobre los componentes del servidor:





Instancia / Base de Datos:

V\$DATABASE, V\$INSTANCE, V\$OPTION, V\$PARAMETER,
V\$BACKUP, V\$PROCESS, V\$WAITSTAT

Memoria:

V\$SYSTAT, V\$BUFFER_POOL_STATISTICS, V\$DB_OBJECT_CACHE,
V\$SGASTAT, V\$ROWCACHE, V\$LIBRARYCACHE

Disco:

V\$DATAFILE, V\$FILESTAT, V\$LOG, V\$LOG_HISTORY, V\$DBFILE

Sesiones / Usuario:

V\$LOCK, V\$PROCESS, V\$OPEN_CURSOR, V\$SORT_USAGE,
V\$SESSION, V\$SESSTAT, V\$SESSION_EVENT, V\$SESSION_WAIT

Contención:

V\$LOCK, V\$ROLLNAME, V\$WAITSTAT, V\$LATCH, V\$ROLLSTAT

NOTA: Estas vistas dinámicas, son las más comunes ya que existen mas de 500 en todo el diccionario de datos.

Para recolectar información adecuada sobre la Base de Datos es necesario utilizar el comando `ANALYZE` para que llene de estadísticas todas las tablas del diccionario de datos.

Las tablas más frecuentes del diccionario de datos para el monitoreo son:

DBA_TABLES, DBA_TAB_COLUMNS, DBA_INDEXES, DBA_PARTITIONS,
DBA_IND_PARTITIONS.

V.4.2 Manejo de Utilerías de Oracle.

Se puede reunir información para un monitoreo sobre un tiempo definido, para ello Oracle proporciona unos programas (*utlbstat.sql* y *utlestat.sql*) los cuales se dedican a esta colecta y así producir un reporte.

Estos programas se encuentran en el siguiente directorio `$ORACLE_HOME / rdbms / admin`. Deben de ejecutarse en una sesión de server manager, conectado con el usuario SYS. La ejecución de estos programas brindan mejores y más resultados si se define en el archivo de inicialización.

Si la Base de Datos se da de baja cuando se están corriendo estas 2 utilerías, las estadísticas no tendrán significado y se tendrá que volver a repetir.

A continuación se menciona una breve descripción de cada uno de estos programas:

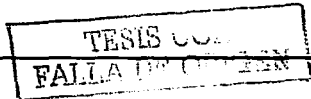
utlibstat.sql.- Genera un conjunto de tablas y vistas en el esquema de sys, con nombres de objetos que contienen la palabra begin, por ejemplo STAT\$BEGIN_FILE. Estos objetos son rellenos con estadísticas iniciales de monitoreo de la Base de Datos.

utlestat.sql.- Crea un conjunto de tablas y vistas en el esquema de sys, con nombres de objetos que contienen la palabra end, por ejemplo STAT\$END_STATS. Estos objetos son rellenos con las estadísticas finales de monitoreo de la Base de Datos. Además, crea un conjunto de tablas que contienen las diferencias entre las estadísticas iniciales y finales, por ejemplo STAT\$\$FILE junto con el archivo report.txt en el directorio actual.

El reporte de estadísticos del archivo report.txt contiene las siguientes secciones:

- Estadísticos del Library Caché
- Estadísticos del Sistema, muestra por transacción y por sesión
- Eventos de espera
- Estadísticos del DBWR
- Estadísticos de I / O, por archivo y por tablespace
- Estadísticos de Latch
- Inicialización de Parámetros
- Estadísticos del Diccionario Caché
- Los tiempos de inicio y fin de corrida de los archivos.

Muchos de los estadísticos capturados en el report.txt informan sobre las más pequeñas observaciones. Estas, son verificadas por desarrolladores del kernel quienes chequean sus códigos para afinaciones internas de Oracle8i.





V.5 MONITOREO Y AFINACION DEL SISTEMA OPERATIVO.

El administrador del sistema operativo es el responsable de monitorear y afinar el sistema operativo; se enfoca principalmente en identificar como las aplicaciones le afectan al funcionamiento de la plataforma. Cuando se monitorea el sistema operativo las partes a considerar son:

Uso de CPU

El monitoreo del CPU en el sistema es de gran relevancia, el CPU debe siempre estar en trabajo constante pero el porcentaje que no debe de sobrepasar es de 90 % de su uso, ya que de lo contrario traería, problemas los cuales se ven reflejados hasta en una caída del servidor e inclusive en una corrupción de la Base de Datos.

Los procesos que están ligados a la Base de Datos normalmente si ocupan bastante CPU pero no se deben de elevar mucho ya que de lo contrario los problemas se encuentran en los parámetros y funcionamiento de la Base de Datos. El punto ideal es de tener de 25 % a 40% de tiempo corriendo las tareas de sistema operativo y el otro 60% a 75% del tiempo de CPU corriendo tareas de otras aplicaciones

Uso de Memoria

El manejo de la memoria en el sistema operativo, es de gran relevancia ya que influye en el funcionamiento ideal de la Base de Datos, el manejo de los parámetros que componen el SGA de la instancia se ven reflejados en la memoria de Sistema operativo; ya que el SGA se inserta sobre la memoria RAM del servidor, por lo tanto es necesario trabajar en equipo en el buen diseño de los parámetros de la Base de Datos para que esta trabaje sin problemas y sin afectar al sistema operativo.

Niveles de I/O

El administrador del sistema operativo define el funcionamiento del I/O, de los discos. Para ello es necesario balancear la carga a través de los controladores de los discos. El monitoreo del I/O consiste en obtener los datos de cuantas lecturas y escrituras se realizan en cada uno de los discos para así validar que objetos de la Base de Datos se encuentran relacionados a este disco y así poder reagruparlos y distribuirlos en otros discos.

Traffic en la red

El administrador del sistema operativo es quien se encarga del monitoreo de la red, si existen problemas en la red, Oracle tendrá problemas en las transacciones realizadas por los usuarios hacia el servidor, estos mensajes son grabados en los archivos log del manejador.

Cada una de estas partes pueden ser monitoreadas tanto por personal que administra el sistema operativo como personal que administra la Base de Datos, cada uno de ellos cuenta con comandos y herramientas independientes.

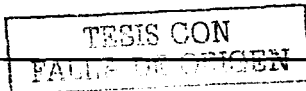
Ver Detalle en APENDICE II IV.5.1

V.6 MONITOREO Y AFINACIÓN DE LOS SEGMENTOS DE MEMORIA DE ORACLE.

El monitoreo y la afinación de los segmentos de memoria de una Base de Datos, juegan un papel muy importante, ya que de ello depende en gran medida el buen funcionamiento de la Base de Datos. Oracle utiliza la memoria para almacenar información como lo es el código de programa que esta siendo ejecutado, información acerca de los sesiones conectadas ya sea activas o inactivas, el estado actual de una sentencia sql, información de bloqueos, etc.

Las estructuras de memoria asociadas con Oracle incluyen las siguientes:

- Area Global del Sistema (SGA)
 - Shared Pool
 - Buffer Cache
 - Redo Log Buffer
- Area de Programa Global (PGA)
- Areas de Código de Software





V.6.1 AREA GLOBAL DEL SISTEMA (SYSTEM GLOBAL AREA SGA)

El Area Global del Sistema, es un grupo de estructuras de memoria compartidas que contienen datos y controlan la información de una determinada instancia de Oracle. Por ejemplo, si existen varios usuarios concurrentemente conectados a la Base de Datos, la información que se encuentre en el Area Global del Sistema es compartida entre ellos.

El Area Global del Sistema contiene las siguientes estructuras de datos:

- Shared Pool
- Library Cache
- Data Dictionary Cache
- Buffer Cache
- Redo Log Buffer

V.6.1.1 SHARED POOL

El Shared Pool esta compuesto de 3 áreas principales: library cache, dictionary cache, y estructuras de control. El tamaño total del shared pool lo determina el parámetro `shared_pool_size` en el archivo de inicialización; su valor por default es de 3,500,000 bytes.

Una correcta asignación de tamaño al shared pool reduce los consumos de tiempo en al menos 3 partes principales:

- 1.- El tiempo de parseo se evita si la sentencia SQL se encuentra actualmente en el shared pool; lo cual reduce el consumo de CPU.
- 2.- Se reduce la sobrecarga en la memoria de la aplicación ya que todas las aplicaciones utilizan las mismas sentencias sql que se encuentran en el shared pool.
- 3.- Existe una reducción en los recursos de I/O, ya que los elementos del diccionario que se encuentran en el shared pool no requieren de acceso a disco.

MANEJO DE DATOS EN EL SHARED POOL.

Dentro del Shared pool algunos de sus caches son dinámicos, es decir incrementan o decrementan sus tamaños como sea necesario. Estos caches dinámicos, incluyen el library cache y el data dictionary cache. Si el tamaño del shared pool alcanza

TESIS CON
FALLA DE ORIGEN

su límite máximo los objetos contenidos en él son excluidos, por lo tanto es una señal de que el tamaño del shared pool resulta insuficiente.

Para la gran mayoría de las aplicaciones, el tamaño del shared pool es crítico para el desempeño de Oracle ya que en él se encuentra el data dictionary cache y todos los bloques de PL/SQL, que incluyen procedimientos, funciones, paquetes, triggers, etc. Si el shared pool es demasiado pequeño, entonces el servidor dedicará sus recursos al manejo de dicho espacio limitado, lo cual consume gran cantidad de recursos de CPU provocando contención.

Para observar el nivel de carga (hit ratio) del data dictionary cache y del library cache se pueden ejecutar las siguientes sentencias:

```
SELECT (SUM(PINS - RELOADS)) / SUM(PINS) "LIB CACHE"
FROM V$LIBRARYCACHE;
```

```
SELECT (SUM(GETS - GETMISSES - USAGE - FIXED)) / SUM(GETS) "ROW CACHE"
FROM V$ROWCACHE;
```

En general la salida de estos queries muestran el número de repaseos reflejados en el library cache. Si el valor es cercano a 1, no es necesario incrementar el tamaño del Shared Pool.

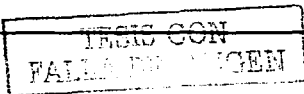
Para verificar la cantidad de memoria libre en el Shared Pool, es necesario ejecutar el siguiente comando:

```
SELECT * FROM V$SGASTAT WHERE NAME = 'FREE MEMORY';
```

Si siempre existe memoria disponible dentro del shared pool, no se tendrá beneficio si esta se incrementa. Cada vez que un elemento es cargado en el shared pool, este no puede ser movido, lo cual implica que la memoria disponible se vuelve discontinua y el shared pool se fragmente.

LIBRARY CACHE

El library cache contiene todos los cursores SQL, los programas PL/SQL, además de información descriptiva acerca de los objetos del esquema.





La actividad del library cache, puede monitorearse por medio de la vista dinámica v\$librarycache, la cual refleja toda la actividad desde el último startup de la Base de Datos.

Cada uno de los registros contenidos en esta vista reflejan la actividad de elementos tales como:

- Sql Area
- Tabla / Procedimiento
- Body
- Trigger

Definidos en la columna namespace.

Existen otras columnas como:

Pins.- Ejecuciones de un elemento en el library cache.

Reloads.- Representa el número de veces en que un elemento no se encontró al momento de ser ejecutado.

El siguiente query muestra la actividad del library cache:

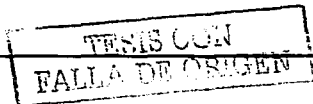
```
SELECT SUM(PINS) "EXECUTIONS",  
SUM(RELOADS) "CACHE MISSES WHILE EXECUTING"  
FROM V$LIBRARYCACHE;
```

La salida del query se observa como sigue:

```
EXECUTIONS CACHE MISSES WHILE EXECUTING  
-----  
320871      549
```

Lo anterior, se interpreta de la siguiente manera:

- La suma de la columna executions indica que las sentencias SQL, los programas PL/SQL y las definiciones de los objetos fueron accedidos un total de 320, 871 veces.
- La suma de la columna cache misses while executing indica que 549 de esas ejecuciones tuvieron que ser reparseadas o implicaron para Oracle la carga de la definición de ese objeto nuevamente, debido a que el elemento ya no se encontraba en el library cache.
- El porcentaje de pérdida de ejecuciones o repaseo es del 0.17% .



VISTAS DINÁMICAS PARA EL LIBRARY CACHE

V\$SQL_AREA

Llenan estadísticos acerca de todos los cursores compartidos, y los primeros 80 caracteres de cada comando.

V\$SQL

Idéntico a v\$sql_area, excepto que estos muestran un registro por cada versión de un comando sql.

V\$SQLTEXT

El texto completo sin truncarlo, en múltiples registros si fuese necesario.

V\$DB_OBJECT_CACHE

Objetos, paquetes, tablas, sinónimos cuando éstos son referenciados en comandos SQL.

V\$LIBRARYCACHE

Estadísticos sobre el manejo del library cache.

V\$SGASTAT

Tamaño de todas las estructuras del SGA

V\$SHARED_POOL_RESERVED

Información acerca de la lista reservada

PARAMETROS QUE AFECTAN AL LIBRARY CACHE

SHARED POOL SIZE

El tamaño del shared pool se define en bytes.

SHARED_POOL_RESERVED_SIZE

El tamaño de ésta lista reservada, en un área de library cache para objetos largos. Objetos pequeños no son llamados en esta área por no fragmentar la lista reservada. El default es 0 (no reserved list). El máximo será la mitad de SHARED_POOL_SIZE.

SHARED_POOL_RESERVED_MIN_ALLOC

El valor mínimo en bytes para objetos los cuales puede usar dicha lista reservada. Objetos grandes pueden usar la lista reservada, si esta no puede buscar

TESIS CON
FALLA DE ENTRENAMIENTO



lugar en otra parte sobre la lista libre. El valor por default es 5000. Los valores deberán ser menores que el `SHARED_POOL_RESERVED_SIZE`, si éste no es 0.

PAQUETE `DBMS_SHARED_POOL`

Este paquete es para asegurar que paquetes largos sean pineados o incrustados en el library cache, y no sean envejecidos. Corrido por `dbmspool.sql`, en algunas versiones se necesita el script `prvtpool.sql` (éste no es corrido en `catproc`)

Para fijar un paquete en el shared pool, usar

```
SQL > EXECUTE dbms_shared_pool.keep ( 'package_name' );  
SQL > EXECUTE dbms_shared_pool.unkeep ( 'package_name' );
```

Para verificar si un objeto ha sido marcado 'kept', consulta la columna `KEPT` en `V$DB_OBJECT_CACHE`.

La vista `V$DB_OBJECT_CACHE` se puede usar para verificar el monto de la memoria compartida usada por un objeto en cache.

DICTIONARY CACHE

La vista dinamica que permite monitorear la actividad del cache de diccionario de datos (data dictionary cache) se llama `v$rowcache`. Cada uno de los registros contenidos en esta tabla contienen estadísticas de un elemento del data dictionary. Las siguientes columnas reflejan el uso y el grado de efectividad del data dictionary.

PARAMETER. Identifica un elemento en particular del data dictionary.

GETS.- Representa el total de peticiones de información sobre un elemento en específico.

GETMISSES.- El número de peticiones que resultaron en pérdidas en el cache.

El siguiente query muestra la actividad del data dictionary cache:

```
SELECT SUM(GETS) "DATA DICTIONARY GETS",  
SUM(GETMISSES) "DATA DICTIONARY CACHE GET MISSES"  
FROM V$ROWCACHE;
```



La salida se interpreta de la siguiente manera:

DATA DICTIONARY GETS DATA DICTIONARY CACHE GET MISSES

1439044

3120

La suma de la columna GETS indica que existieron 1,439,044 peticiones al cache del diccionario de datos.

La suma de la columna GETMISSES indica que 3120 de las peticiones al diccionario de datos, resultaron en pérdidas.

El porcentaje entre GETMISSES y GETS es de 0.2%

V.6.1.2 BUFFER CACHE

El Buffer Cache, es una porción del SGA que contiene copias de los bloques de datos que son leídos de los archivos de datos. Todos y cada uno de los procesos de los usuarios que se encuentran conectados a la Base de Datos de una manera concurrente, comparten el acceso al Buffer Cache. Los buffers en este cache se encuentran organizados en 2 listas: la lista de escrituras y la lista LRU (Least Recently Used).

La lista de escritura contiene todos aquellos buffers sucios (dirty buffer), los cuales contienen datos que han sido modificados pero que aún no han sido llevados a disco.. La lista LRU contiene buffers libres, pinned buffers (actualmente ocupados o accesados) y buffers sucios (saturados de información) los cuales no han sido movidos aun a la lista de escritura.

La primera vez que un proceso de usuario de Oracle requiere una pieza de datos, este es buscado en el buffer cache; si el proceso encuentra el dato en el cache (cache hit), el dato es leído directamente de la memoria. Si el proceso no encuentra el dato en el cache (cache miss), éste, debe de ser copiado desde un datafile al buffer cache antes de ser accesado.

El parámetro de inicialización DB_BLOCK_BUFFERS especifica el número de buffers en el buffer cache. Cada buffer en el cache es del tamaño de un bloque de

TESIS CON
FALLA DE CONCEN



Oracle, especificado en el parámetro de inicialización DB_BLOCK_SIZE, debido a ello, cada buffer en el cache solo puede contener un bloque de datos.

Oracle almacena estadísticas que reflejan el acceso a los datos en una vista llamada v\$sysstat. Estas estadísticas son comunmente utilizadas, para afinar el Buffer Cache. Entre los campos más significantes de esta vista se encuentra:

DB BLOCK GETS.- La suma de estos valores representa el número total, de peticiones por datos.

CONSISTENT GETS.- El valor incluye las peticiones satisfechas accedando a los buffers en memoria.

PHISICAL READS.- El número total de peticiones de datos que resultan en acceso a los datafiles.

Con el siguiente query podemos obtener algunas estadísticas:

```
SELECT NAME, VALUE
FROM V$SYSSTAT
WHERE NAME IN ('DB BLOCK GETS', 'CONSISTENT GETS', 'PHYSICAL READS');
```

Solving Memory Allocation Problems

La salida se observa como sigue:

| NAME | VALUE |
|-----------------|--------|
| DB BLOCK GETS | 85792 |
| CONSISTENT GETS | 278888 |
| PHYSICAL READS | 23182 |

El hit ratio para el buffer cache se calcula con la siguiente formula:

Hit Ratio = 1 - (lecturas fisicas / (db blocks gets + consistent gets))

De acuerdo a este ejemplo el hit ratio del buffer cache es de 94%.

Es importante considerar como regla general, que el incremento de los DB_BLOCK_BUFFERS sólo se llevará a cabo cuando:

- El cache hit ratio sea menos de 0.9%
- El incremento previo de DB_BLOCK_BUFFERES fue exitoso.



Existe un programa de nombre CATPARR.SQL el cual crea una vista de nombre vSBH la cual muestra el número de archivo y el número de bloques que actualmente se encuentran dentro del SGA.

```
SELECT file#, COUNT(block#), COUNT (DISTINCT file# || block#)
FROM VSBH
GROUP BY file#;
```

Si el hit ratio es bajo, es decir, menor del 60 o 70%, esto implica que debe de incrementarse el número de buffers en el cache para aumentar el performance.

Por otro lado si el hit ratio es alto, implica que es lo suficientemente grande para mantener la mayor cantidad de datos que son frecuentemente accedados. En este caso, es posible reducir el tamaño del cache y seguir manteniendo un buen desempeño.

V.6.1.3 REDO LOG BUFFER

El redo log buffer es un espacio circular dentro de la SGA que contiene información acerca de los cambios realizados en la Base de Datos. La información que se encuentra almacenada es la necesaria para reconstruir los cambios realizados en la Base de Datos por sentencias tales como INSERTS, UPDATES, DELETES, CREATE, ALTER u operaciones DROP.

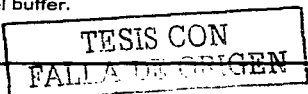
El parámetro de inicialización LOG_BUFFER, determina el tamaño del redo log buffer. En general, un valor alto reduce el I/O particularmente si existen numerosas transacciones o algunas otras que tomen demasiado tiempo.

La estadística REDO BUFFER ALLOCATION RETRIES, refleja el número de veces en que un proceso de usuario espera por espacio en el redo log buffer. Esta estadística, está disponible en la vista dinámica v\$SYSSTAT.

El siguiente query permite monitorear dicha estadística:

```
SELECT NAME, VALUE
FROM v$SYSSTAT
'WHERE NAME = 'REDO BUFFER ALLOCATION RETRIES';
```

El valor obtenido en la columna value debe de estar cerca de 0. Si este valor incrementa constantemente, implica que el proceso ha tenido que esperar cierto tiempo por espacio en el buffer.





ARCHIVOS, GRUPOS Y MIEMBROS

Los Redo log files, están organizados en grupos. Cada grupo debe tener uno ó más miembros. Todos los miembros de un grupo deben tener contenidos idénticos. Por seguridad se recomienda tener 2 ó más miembros en cada grupo, para que en caso de desastre, siempre se cuente con una copia segura.

Los Redo log files en el mismo grupo deben ser idealmente separados, en dispositivos rápidos debido a que el LGWR escribe en ellos continuamente. Cualquier espera para el " log file parallel write " en la vista V\$SYSTEM_EVENT indica un posible problema de I/O con los archivos logs.

EVITANDO ESPERAS DEL CHECKPOINT

El LGWR escribe para los grupos de redo log en círculo, uno tras otro, cuando un grupo se llena, Oracle debe ejecutar un checkpoint. Esto significa:

- El LGWR fluye del log buffer hacia el disco.
- DBWR escribe todos los bloques sucios a los datafiles
- LGWR o CKPT actualizan los encabezados de los datafiles

Los checkpoints normalmente asignan otro trabajo para continuar al mismo tiempo, aunque en el transcurso del checkpoint se generan muchas escrituras a disco, pero si DBWR no ha sido finalizado la fase de checkpoint a un archivo y LGWR necesita el archivo de nuevo. El LGWR tiene que esperar, para evitar ello:

- No esperas para el checkpoint
- Largos archivos de redo log, para disminuir el número de checkpoints.
- Más grupos de redo log para incrementar el tiempo antes que el LGWR necesite sobre escritura.
- Verificar el tiempo de switches de log en el alert.log
- Verificar mensajes de error " checkpoint not complete; unable to allocate file ". Esto significa que LGWR ha esperado por el checkpoint para finalizar.

REGULANDO CHECKPOINTS

DBWR debe siempre ejecutar un checkpoint al final de cada grupo de redo log. Frecuentes checkpoints significan cortas recuperaciones de instancias, pero más escrituras por DBWR y LGWR o CKPT (para los data file headers).

Si la prioridad es tener un performance eficiente, escoge un tamaño de redo log file como que suficientes checkpoints pasen continuamente para causar una sensible, pero baja en tiempo de respuesta, pero no muy continuas.

EL PARÁMETRO LOG_CHECKPOINT_INTERVAL

Este parámetro asigna el número de bloques de sistema operativo entre cada checkpoint.

EL PARÁMETRO LOG_CHECKPOINT_TIMEOUT

Este parámetro fija el número de segundos entre cada checkpoint, el valor por default es cero, el cual significa que esto no tiene impacto.

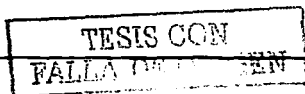
REDUCIENDO REDO

Los Redo log files siempre están ocupados, pero éstos son unos pocos pasos que pueden tomarse para reducir su contenido. Usar direct path llamando sin archiving (modo operativo de la Base de Datos) o en modo irrecuperable

- Si se usa Sql loader en modo de path directo y no se hace archive, no son generados los redo, si ocurre una falla de instancia, se debe correr el proceso de nuevamente.
- Si se usa Sql loader en modo de path directo y se hace archive, es posible seleccionar el modo UNRECOVERABLE.
- Crear objetos en paralelo en modo UNRECOVERABLE
- Si existe la arquitectura Parallel Query, se pueden crear tablas e índices en paralelo, utilizando la opción UNRECOVERABLE. Ellos, tienen que ser recreados si ocurre una falla de instancia antes de completar el comando.

V.6.2 AREA GLOBAL DE PROGRAMA (PGA)

El área global de programa, es una región de memoria que contiene datos o información de control de algún proceso ya sea de servidor o de background. Esta es una área no compartida a la cuál un proceso pueda escribir, únicamente el código de Oracle puede leer o escribir en esta área.





Una porción de PGA es alojada por Oracle cuando un usuario se conecta a la Base de Datos y una sesión es creada.

Los contenidos del PGA varían dependiendo de la instancia, pero siempre contienen un espacio de pila (stack space), el cual es memoria alojada para contener las variables, arreglos y otra información de una sesión.

Los parámetros de inicialización OPEN_LINKS y DB_FILES afectan los tamaños del PGA.

V.6.3 AREAS DE CODIGO DE SOFTWARE

Porciones de memoria utilizadas para almacenar código que esta siendo ejecutado o que va a serlo. Las áreas de software son comúnmente estáticas en tamaño, únicamente son actualizadas cuando el software es actualizado o reinstalado. Las áreas de software son únicamente de lectura y pueden ser instaladas ya sea compartidas o no compartidas.

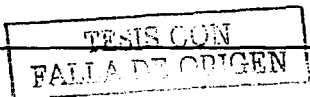
V.7 CONFIGURACIÓN DE LA BASE DE DATOS Y USO DE HARDWARE

Las Bases de Datos son configuradas y distribuidas en base a las características de cada servidor por eso la importancia de un buen diseño para su creación y distribución. La relación que existe entre una Base de Datos y el Sistema Operativo se encuentra reflejada en los tablespaces, que están definidos en la Base de Datos de manera lógica, y de manera física en archivos de datos. Estos últimos, son los que se distribuyen y definen su tamaño a nivel Sistema Operativo.

Cada Base de Datos debe de tener tablespaces específicos para alojar sus objetos tales como:

- *Objetos de Diccionario de Datos.* Generalmente estos objetos surgen cuando se crea la instancia, para ello es necesario crear un tablespace llamado SYSTEM donde estarán alojados.

- *Segmentos de Rollback.* Para estos segmentos de la Base de Datos es necesario crear uno o varios tablespaces (RBS1, RBS2, RBS3, RBS4).





- *Segmentos de Temporal.* Todas las transacciones que necesiten espacios temporales, ya sea para realizar ordenamientos dentro de la Base de Datos requieren de uno o varios tablespaces temporales (TEMP).

- *Tablas.* En necesario crear varios tablespaces para el almacenamiento de las tablas dependiendo de su actividad, si son históricas de catálogo o dinámicas, estos tablespaces son denominados normalmente (IDATA01, IDATA02, IDATA03, IDATA04, IDATA05)

- *Indices.* Son creados para almacenar los índices de todas las tablas de la Base de Datos. Estos son denominados comúnmente (IINDEX01, IINDEX02, IINDEX03, IINDEX04, IINDEX05).

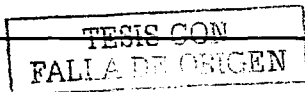
Para el monitoreo de todos los tablespaces de la Base de Datos es necesario consultar algunas tablas del diccionario de datos, en las cuales se encuentran las relaciones reales del Sistema Operativo y la Base de Datos.

Por ejemplo, se pueden encontrar información del espacio libre y ocupado por cada uno de los tablespaces, y con ello determinar si es necesario adicionarle espacio a un tablespace en particular.

Al monitorear el Sistema Operativo podemos encontrar que algunos discos tienen contención de lecturas y escrituras, para corregir esto es necesario primero encontrar que archivos de datos están alojados en ese disco para posteriormente saber a que tablespace corresponden y al final saber que tipo de objetos de Oracle están almacenados en ese tablespace y así poder afinar esos objetos. Muchas veces solo es necesario distribuir los archivos de datos en diferentes discos y evitar la contención.

Ver Detalle en APENDICE II V.7.1

Otra de las relaciones de la Base de Datos y el Sistema Operativo son el manejo de los archivos de Redo Log, los cuales son escritos por el proceso de background de Oracle llamado LGWR el cual baja de los segmentos de memoria de Redo log buffer todas los cambios sufridos en la Base de Datos. Es importante el monitoreo de estos archivos tanto a nivel Base de Datos como de Sistema operativo ya que tiene un alto grado de escrituras por ambas partes y la definición de óptimos parámetros y buen





tamaño para estos archivos implica un buen funcionamiento en toda la plataforma. Para el monitoreo de estos archivos a nivel Base de Datos, es necesario ejecutar queries en las vistas de VSLOG, VSLOGFILE.

Ver Detalle en APENDICE II V.7.2

Si la configuración de ARCHIVE esta habilitada en la Base de Datos, es necesario destinar la copia de estos archivos en discos diferentes para evitar contención en el Sistema Operativo. Para monitorear esta configuración es necesario revisar los parámetros del archivo de init.ora en el cual se definen las rutas para guardar el respaldo de estos archivos; si esta opción se encuentra habilitada es necesario también definir un tiempo adecuado en el checkpoint de la Base de Datos ya que es el tiempo en el cual se estará leyendo de memoria las actividades realizadas y almacenándolas en los archivos de redo log.

Ver Detalle en APENDICE II V.7.3

V.8 RECOMENDACIONES SOBRE LOS SEGMENTOS DE ROLLBACK.

Cuando una transacción hace un cambio a uno o varios registros de una tabla, la información antes del cambio es guardada en los segmentos de rollback, si la transacción no se confirma el valor original es regresado a la tabla mediante estos segmentos; los cuales son usados en forma circular. Si se subestiman el número y tamaño de los segmentos que necesitas se degrada el performance y pueden generar errores. Si la instancia falla cuando una transacción esta en proceso, el servidor de Oracle realiza rollback de las transacciones que no terminaron de manera normal; esto es en forma automática al inicializar la instancia. Todas las transacciones de DML generan una imagen de rollback.

Objetivos del Monitoreo y afinación de Segmentos de Rollback

Las transacciones nunca deben de esperar para acceder a segmentos de Rollback

Esto depende de si existen suficientes segmentos de rollback



Los segmentos de Rollback no deben extenderse durante una corrida normal

Esto depende del número de extents por segmento

No son transaccionales, sin embargo grandes o excepcionales, siempre deben correr fuera del espacio de Rollback

Esto significa que los segmentos de Rollback deben ser medidos correctamente.

Los lectores siempre deben ver las imágenes de consistencia de lectura que necesitan

Esto depende de tanto números como tamaños de los segmentos de Rollback

¿ Cuantos Segmentos de Rollback ?

- Se recomienda un segmento de Rollback para diez usuarios concurrentes, en aplicaciones OLTP.

- Para trabajo en batch, se puede recomienda asignar un segmento de Rollback para cada trabajo concurrente.

Midiendo Segmentos de Rollback

El buen funcionamiento de los segmentos de rollback depende de los parámetros con los cuales son creados, las recomendaciones para la creación de los segmentos son las siguientes:

1.- Escoger el parámetro de almacenamiento INITIAL de la lista (8kb, 16kb, 32kb, 64kb) para las pequeñas transacciones y de 128Kb, 256kb, 512kb, 1mb, 2mb, 4mb para transacciones largas

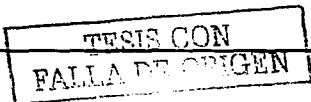
2.- Usar el mismo valor para NEXT e INITIAL, el tamaño de PCTINCREASE es 0, y todos los extents deben de ser del mismo tamaño del NEXT.

3.- Hacer todos los segmentos de Rollback del mismo tamaño. Si no se hace ellos probablemente llegarán a ser del mismo tamaño sobre el tiempo de cualquier manera. El más pequeño será extendido por ser usado por transacciones largas.

4.- Escoger el parámetro de MINEXTENTS a 20 para asegurar que los segmentos de Rollback no encojan hacia abajo de 20 extents. Tomar en consideración la necesidad de consultas concurrentes.

5.- Forzar un segmento de Rollback a regresar al tamaño optimal (definido por el parámetro optimal) siempre que no haya transacciones activas, por el comando:

```
SQL> ALTER ROLLBACK SEGMENT rollback_name SHRINK;
```





Monitoreando los Segmentos de Rollback.

Es importante, saber como monitorear los segmentos de rollback en una Base de Datos que se encuentra en funcionamiento, por lo tanto es necesario ejecutar queries sobre las vistas del sistema `V$ROLLSTAT`, `V$WAITSTAT`. Cualquier valor que no sea cero en la columna `UNDO HEADER` de la vista `V$WAITSTAT` ó en la columna `WAITS` de la vista `V$ROLLSTAT`, indican contención para bloques de encabezado de segmentos de Rollback.

El `report.txt`, producido por `utlbbstat / utllestat`, también tiene una sección de estadísticos de Rollback.

Ver Detalle en APENDICE II V.8.1

Posibles Problemas

Transacciones Largas: Si una transacción es excepcionalmente larga, esto puede fallar porque no hay más espacio en el tablespace para que el segmento de rollback se puede expandir. Para este error se necesitan segmentos grandes de rollback, o más espacio en el tablespace.

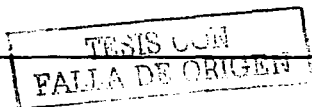
Foto vieja (Snapshot Too Old): Si una consulta falla con el siguiente mensaje de error (`ORA-1555: snapshot too old rollback segment too small`), la imagen del rollback necesitada para la lectura de consistencia probablemente ha sido sobrescrita por una transacción activa. Para resolver este error, se necesitan grandes segmentos de rollback ó incrementar el número de ellos.

Para transacciones largas se puede usar la siguiente sintáxis, para asignarla a un segmento de rollback, de específico:

```
SQL> SET TRANSACTION USE ROLLBACK SEGMENT
nombre_rollback_segment ;
```

V.9 USO DE PARTICIONAMIENTO DE OBJETOS DE ORACLE.

El particionamiento o división por bloques de tablas e índices en una Base de Datos significa descomponer alguno de estos objetos en pequeñas y más manejables piezas de datos, denominadas particiones. Una vez que las particiones están definidas,



las sentencias SQL pueden acceder y manipular las particiones ya sea como tablas o índices completos.

Tablas e índices particionados son comúnmente usados en aplicaciones de Datawarehouse, las cuáles almacenan y acumulan grandes cantidades de datos históricos.

Métodos de Particionamiento

Existen 2 métodos principales de particionamiento: por rango, el cual consiste en particionar los datos de la tabla o índice por medio de un rango de valores y el método aleatorio (hash). Este último consiste en fragmentar (particionar) la tabla o índice por medio de una función aleatoria.

Todas las particiones de una tabla o índice comparten los mismos atributos lógicos aunque sus atributos físicos pueden ser diferentes. Por ejemplo, todas las particiones en una tabla comparten las mismas columnas y las definiciones de los constraints, pero las especificaciones de almacenamiento y otros atributos físicos como PCTFREE, PCTUSED, etc. pueden ser distintos.

PARTICIONAMIENTO POR RANGO.

Los rangos en las particiones son definidos por la especificación de particionamiento para una tabla o índice.

PARTITION BY RANGE (lista de columnas)

y las especificaciones de particionamiento para cada partición individual:

VALUES LESS THAN (lista de valores)

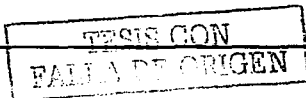
DONDE:

(lista de columnas). La lista de columnas que determina la partición a la que un registro de tabla o índice pertenece.

(lista de valores). Es una lista de valores para cada una de las columnas en la lista anterior.

A continuación se muestra un ejemplo del particionamiento de una tabla.

```
CREATE TABLE sales
  ( invoice_no    NUMBER,
    sale_year    INT NOT NULL,
```





```
sale_month INT NOT NULL,  
sale_day INT NOT NULL )  
PARTITION BY RANGE (sale_year, sale_month, sale_day)  
( PARTITION sales_q1 VALUES LESS THAN (1997, 04, 01)  
TABLESPACE tsa,  
PARTITION sales_q2 VALUES LESS THAN (1997, 07, 01)  
TABLESPACE tsb,  
PARTITION sales_q3 VALUES LESS THAN (1997, 10, 01)  
TABLESPACE tsc,  
PARTITION sales_q4 VALUES LESS THAN (1998, 01, 01)  
TABLESPACE tsd );
```

```
CREATE TABLE sales  
( invoice_no NUMBER,  
sale_year INT NOT NULL,  
sale_month INT NOT NULL,  
sale_day INT NOT NULL )  
STORAGE (INITIAL 100K NEXT 50K) LOGGING  
PARTITION BY RANGE ( sale_year, sale_month, sale_day)  
( PARTITION sales_q1 VALUES LESS THAN ( 1999, 04, 01 )  
TABLESPACE tsa STORAGE (INITIAL 20K, NEXT 10K),  
PARTITION sales_q2 VALUES LESS THAN ( 1999, 07, 01 )  
TABLESPACE tsb,  
PARTITION sales_q3 VALUES LESS THAN ( 1999, 10, 01 )  
TABLESPACE tsc,  
PARTITION sales_q4 VALUES LESS THAN ( 2000, 01, 01 )  
TABLESPACE tsd)  
ENABLE ROW MOVEMENT;
```

La opción de ENABLE ROW MOVEMENT permite la migración de un registro a una nueva partición si este es actualizado.

```
CREATE INDEX month_ix ON sales(sales_month)  
GLOBAL PARTITION BY RANGE(sales_month)  
(PARTITION pm1_ix VALUES LESS THAN (2)  
PARTITION pm2_ix VALUES LESS THAN (3)  
PARTITION pm3_ix VALUES LESS THAN (4)  
PARTITION pm4_ix VALUES LESS THAN (5)  
PARTITION pm5_ix VALUES LESS THAN (6)  
PARTITION pm6_ix VALUES LESS THAN (7)  
PARTITION pm7_ix VALUES LESS THAN (8)  
PARTITION pm8_ix VALUES LESS THAN (9)  
PARTITION pm9_ix VALUES LESS THAN (10)  
PARTITION pm10_ix VALUES LESS THAN (11)  
PARTITION pm11_ix VALUES LESS THAN (12)  
PARTITION pm12_ix VALUES LESS THAN (MAXVALUE));
```

TESIS CON
FALLA DE ORIGEN



PARTICIONAMIENTO ALEATORIO

Este modo de crear particiones, es usado cuando los datos no permiten particiones por medio de rangos. Un particionamiento aleatorio es la mejor opción cuando:

- No se sabe de antemano la cantidad de datos van a ser particionados dentro de algún rango específico.
- El tamaño de las particiones por rango difiere notablemente entre una y otra.

Ejemplos de Particionamiento Aleatorio

En este ejemplo se indica el número de particiones que se desean y el sistema le asigna un nombre por default a cada una de las particiones.

```
CREATE TABLE dept (deptno NUMBER, deptname VARCHAR(32))  
PARTITION BY HASH(deptno) PARTITIONS 16;
```

Los nombres de las particiones así como algunos parámetros de almacenamiento son indicados.

```
CREATE TABLE dept (deptno NUMBER, deptname VARCHAR(32))  
STORAGE (INITIAL 10K)  
PARTITION BY HASH(deptno)  
(PARTITION p1 TABLESPACE ts1, PARTITION p2 TABLESPACE ts2,  
PARTITION p3 TABLESPACE ts1, PARTITION p4 TABLESPACE ts3);
```

```
CREATE TABLE product( ... )  
STORAGE (INITIAL 10M)  
PARTITION BY HASH(column_list)  
( PARTITION p1 TABLESPACE h1,  
PARTITION p2 TABLESPACE h2 );
```

Ventajas del Uso de Particionamiento

- **Reducción de Tiempo de No Disponibilidad en Mantenimiento.**

El hecho de que una tabla o índice se encuentre particionado, permite que algunas operaciones referentes al manejo de datos tales como carga de datos, creación de índices y depuración de tablas resulten en gran medida en reducción de tiempos. Esto debido a que únicamente se opera sobre la partición específica y no sobre toda la tabla



o índice. Además de ello, las operaciones de mantenimiento pueden desarrollarse en diferentes particiones de manera simultánea.

- **Reducción de Tiempo de No Disponibilidad por Falta de Datos.**

Algunas operaciones de mantenimiento son operaciones no planeadas, las cuales requieren de operaciones de recuperación por causa de fallas de hardware o software. El almacenar particiones en diferentes tablespaces provee los siguientes beneficios:

El tiempo de No Disponibilidad en la ejecución de la sentencia *recovery* reduce el tiempo ya que la recuperación de un tablespace más pequeño es más rápida.

El monto de datos no disponibles es más pequeño, ya que únicamente el tablespace en cuestión es llevado al status de fuera de línea. Lo cual implica que las operaciones de usuarios o de mantenimiento pueden acceder a otra partición.

- **Desempeño y Rapidez de Consultas a los Datos.**

Las consultas que se realizan a tablas demasiado largas presentan problemas de desempeño. Una consulta que requiere un barrido de una tabla puede tomar demasiado tiempo, ya que ella, verificará cada uno de los registros de la tabla.

Con el uso del particionamiento una consulta únicamente puede barrer la partición que corresponda a los datos que requiere en lugar de recorrer la tabla por completo. Con ello se mejoran los tiempos de respuesta, además de reducir el espacio de temporal para aquellas consultas que realizan ordenamientos.

- **Desempeño en la lectura y Escritura**

El uso de particionamiento permite el control de como los datos pueden ser distribuidos a través, de los dispositivos físicos. Para balancear la lectura y escritura a disco es posible especificar donde almacenar cada una de las particiones de una tabla o índice.



OPERACIONES DE MANTENIMIENTO EN LAS PARTICIONES

ALTER TABLE ...

| | |
|--------------------------|---------------------------|
| Agregar Particiones | <i>Add Partition</i> |
| Partición Coalesce | <i>Coalesce Partition</i> |
| Borrar Particiones | <i>Drop Partition</i> |
| Intercambiar Particiones | <i>Exchange Partition</i> |
| Mover Particiones | <i>Move Partition</i> |
| Renombrar Particiones | <i>Rename Partition</i> |
| Agrandar Particiones | <i>Split Partition</i> |
| Truncar Particiones | <i>Truncate Partition</i> |

ALTER INDEX ...

| | |
|---------------------------------|--------------------------|
| Borrar Partición de Índice | <i>Drop Partition</i> |
| Reconstruir partición de Índice | <i>Rebuild Partition</i> |
| Renombrar partición de Índice | <i>Rename Partition</i> |
| Alargar partición de Índice | <i>Split Partition</i> |

TESIS CON
FALLA DE ORIGEN

RESERVA
FALLA DE ORIGEN

204



CONCLUSIONES

A lo largo de este trabajo se han expuesto muchos de los conceptos relacionados a las Bases de Datos Oracle 8i, desde los más básicos hasta los que se convierten en el lenguaje cotidiano de los administradores de las mismas (DBA's). Resulta satisfactorio el hecho de que todos y cada uno de los conocimientos teóricos que se presentaron en los tres primeros capítulos, permitieron llevarse a la práctica obteniendo los resultados propuestos.

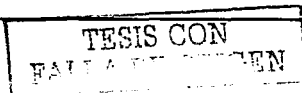
Después de una serie de pruebas en dominios alternos y maquetas, y a la utilización de diversas técnicas para la manipulación de los datos, finalmente pudo llevarse y aplicarse cada uno de los pasos del proceso de migración en la Base de Datos Productiva; quedando en claro, que los objetivos planteados en un principio se cumplieron en forma excelente durante todo el proceso de migración de la Base de Datos del Sistema de Prepago de la empresa de Red Celular.

La Disponibilidad del sistema se mantuvo en un 98%, lo cual fue un porcentaje aceptado por los altos directivos de la empresa. Únicamente se tuvo una pérdida del servicio de aproximadamente 30 minutos lo cuál resulta intrascendente por la hora en que se suscito este evento.

En lo que se refiere a la reducción de costos y tiempos, estos fueron mínimos comparados con las pérdidas que se hubieran presentado en caso de no brindar el servicio a los usuarios finales durante el tiempo de ejecución de todo el proceso de migración.

Obviamente, todo este proceso fué completamente transparente para el usuario, ya que el servicio siempre estuvo presente y operando funcional y dinámicamente.

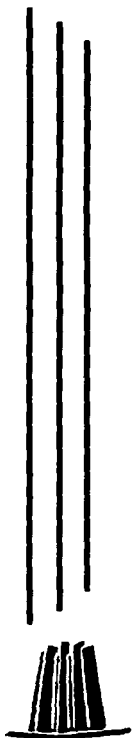
Por otro lado cabe mencionar la gran potencialidad que brinda este manejador de Base de Datos, que es Oracle; el cuál es muy robusto y además de proporcionar diferentes métodos para el desempeño de múltiples tareas operacionales.





Por último podemos mencionar que el trabajo no concluye aquí, ya se dió un gran paso llevando esta enorme cantidad de datos a una nueva versión; el trabajo continúa, esta Base de Datos seguirá operando conllevando esto a una tarea ardua y difícil:

Su administración, mantenimiento y Operación en función del esquema Oracle8i.



**APENDICE I
CAPITULO IV**

TESIS CON
FALLA DE JUDICEN

TESIS CON
FALLA DE ORIGEN

APENDICE I

IV.4.1.1 /oracle/home/genera_beginbackup.sql

```

set pages 0
set feedback off

spool /hotbck1/HOTBACKUP/beginbackup.sql
select 'alter tablespace '||tablespace_name||' begin backup;'
from dba_tablespaces
order by tablespace_name;

select 'exit' from dual;
spool off
exit
    
```

/oracle/home/beginbackup.sql

```

alter tablespace IDATA01 begin backup;
alter tablespace IDATA02 begin backup;
alter tablespace IDATA03 begin backup;
alter tablespace IDATA04 begin backup;
alter tablespace IDATA05 begin backup;
alter tablespace IDATA06 begin backup;
alter tablespace IDATA07 begin backup;
alter tablespace IDATA08 begin backup;
alter tablespace IDATA09 begin backup;
alter tablespace IDATA10 begin backup;
alter tablespace IDATA11 begin backup;
alter tablespace IDATA12 begin backup;
alter tablespace IDATA13 begin backup;
alter tablespace IDATA14 begin backup;
alter tablespace IDATA15 begin backup;
alter tablespace IDATA16 begin backup;
alter tablespace IDATA17 begin backup;
alter tablespace IDATA18 begin backup;
alter tablespace IDATA19 begin backup;
alter tablespace IINDEX01 begin backup;
alter tablespace IINDEX02 begin backup;
alter tablespace IINDEX03 begin backup;
alter tablespace IINDEX04 begin backup;
alter tablespace IINDEX05 begin backup;
alter tablespace IRPT01 begin backup;
alter tablespace SYSTEM begin backup;
alter tablespace TEMP_APP begin backup;
alter tablespace TRBS01 begin backup;
alter tablespace TRBS02 begin backup;
alter tablespace TRBS03 begin backup;
alter tablespace TRBS04 begin backup;
exit
    
```

TESIS CON
FALLA DE ORIGEN

**IV.4.1.2 /oracle/home/generapipesdb1.sh**

```
set pages 0
set feedback off
spool /hotbck1/HOTBACKUP/pipes_dbf1.sh
select 'mknod /hotbck1/ora_backup/'||file_id||'.pipe p'
from dba_data_files
order by file_id;
spool off
exit
```

/hotbck1/HOTBACKUP/copia_pipes.sh

```
cp -p /hotbck1/HOTBACKUP/pipes_dbf1.sh /hotbck1/HOTBACKUP/pipes_dbf2.sh
cat /hotbck1/HOTBACKUP/pipes_dbf2.sh | sed "s/hotbck1/hotbck2/g" >>
/hotbck1/HOTBACKUP/dbf2
mv /hotbck1/HOTBACKUP/dbf2 /hotbck1/HOTBACKUP/pipes_dbf2.sh

cp -p /hotbck1/HOTBACKUP/pipes_dbf1.sh /hotbck1/HOTBACKUP/pipes_dbf3.sh
cat /hotbck1/HOTBACKUP/pipes_dbf3.sh | sed "s/hotbck1/hotbck3/g" >>
/hotbck1/HOTBACKUP/dbf3
mv /hotbck1/HOTBACKUP/dbf3 /hotbck1/HOTBACKUP/pipes_dbf3.sh

cp -p /hotbck1/HOTBACKUP/pipes_dbf1.sh /hotbck1/HOTBACKUP/pipes_dbf4.sh
cat /hotbck1/HOTBACKUP/pipes_dbf4.sh | sed "s/hotbck1/hotbck4/g" >>
/hotbck1/HOTBACKUP/dbf4
mv /hotbck1/HOTBACKUP/dbf4 /hotbck1/HOTBACKUP/pipes_dbf4.sh
```

/hotbck1/HOTBACKUP/pipes_dbf1.sh

```
mknod /hotbck1/ora_backup/1.pipe p
mknod /hotbck1/ora_backup/2.pipe p
mknod /hotbck1/ora_backup/3.pipe p
.
.
mknod /hotbck1/ora_backup/320.pipe p
```

/hotbck1/HOTBACKUP/pipes_dbf2.sh

```
mknod /hotbck2/ora_backup/1.pipe p
mknod /hotbck2/ora_backup/2.pipe p
mknod /hotbck2/ora_backup/3.pipe p
.
.
mknod /hotbck2/ora_backup/320.pipe p
```

/hotbck1/HOTBACKUP/pipes_dbf3.sh

```
mknod /hotbck3/ora_backup/1.pipe p
mknod /hotbck3/ora_backup/2.pipe p
mknod /hotbck3/ora_backup/3.pipe p
.
.
mknod /hotbck3/ora_backup/320.pipe p
```

```
/hotbck1/HOTBACKUP/pipes_dbf4.sh
mknod /hotbck4/ora_backup/1.pipe p
mknod /hotbck4/ora_backup/2.pipe p
mknod /hotbck4/ora_backup/3.pipe p
```

```
mknod /hotbck4/ora_backup/320.pipe p
```

```
/hotbck1/HOTBACKUP/gen_pipeslogs.sh
```

```
for x in 1 2 3 4 5 6 7 8 9 10 11 12
do
echo "mknod /hotbck1/ora_backup/"$x"Abf.pipe p" >>
/hotbck1/HOTBACKUP/pipes_logs.sh
echo "mknod /hotbck1/ora_backup/"$x"Bbf.pipe p" >>
/hotbck1/HOTBACKUP/pipes_logs.sh
echo "mknod /hotbck2/ora_backup/"$x"Abf.pipe p" >>
/hotbck1/HOTBACKUP/pipes_logs.sh
echo "mknod /hotbck2/ora_backup/"$x"Bbf.pipe p" >>
/hotbck1/HOTBACKUP/pipes_logs.sh
echo "mknod /hotbck3/ora_backup/"$x"Abf.pipe p" >>
/hotbck1/HOTBACKUP/pipes_logs.sh
echo "mknod /hotbck3/ora_backup/"$x"Bbf.pipe p" >>
/hotbck1/HOTBACKUP/pipes_logs.sh
echo "mknod /hotbck4/ora_backup/"$x"Abf.pipe p" >>
/hotbck1/HOTBACKUP/pipes_logs.sh
echo "mknod /hotbck4/ora_backup/"$x"Bbf.pipe p" >>
/hotbck1/HOTBACKUP/pipes_logs.sh
done
```

```
/hotbck1/HOTBACKUP/pipes_logs.sh
```

```
# hotbck1
```

```
mknod /hotbck1/ora_backup/1Abf.pipe p
mknod /hotbck1/ora_backup/1Bbf.pipe p
mknod /hotbck1/ora_backup/2Abf.pipe p
mknod /hotbck1/ora_backup/2Bbf.pipe p
mknod /hotbck1/ora_backup/3Abf.pipe p
mknod /hotbck1/ora_backup/3Bbf.pipe p
```

```
mknod /hotbck1/ora_backup/12Abf.pipe p
mknod /hotbck1/ora_backup/12Bbf.pipe p
```

```
# hotbck2
```

```
mknod /hotbck2/ora_backup/1Abf.pipe p
mknod /hotbck2/ora_backup/1Bbf.pipe p
mknod /hotbck2/ora_backup/2Abf.pipe p
mknod /hotbck2/ora_backup/2Bbf.pipe p
mknod /hotbck2/ora_backup/3Abf.pipe p
mknod /hotbck2/ora_backup/3Bbf.pipe p
```

```
mknod /hotbck2/ora_backup/12Abf.pipe p
mknod /hotbck2/ora_backup/12Bbf.pipe p
```

```
# hotbck3
mknod /hotbck3/ora_backup/1Abf.pipe p
mknod /hotbck3/ora_backup/1Bbf.pipe p
mknod /hotbck3/ora_backup/2Abf.pipe p
mknod /hotbck3/ora_backup/2Bbf.pipe p
mknod /hotbck3/ora_backup/3Abf.pipe p
mknod /hotbck3/ora_backup/3Bbf.pipe p

.
.
.
mknod /hotbck3/ora_backup/12Abf.pipe p
mknod /hotbck3/ora_backup/12Bbf.pipe p
```

```
# hotbck4
mknod /hotbck4/ora_backup/1Abf.pipe p
mknod /hotbck4/ora_backup/1Bbf.pipe p
mknod /hotbck4/ora_backup/2Abf.pipe p
mknod /hotbck4/ora_backup/2Bbf.pipe p
mknod /hotbck4/ora_backup/3Abf.pipe p
mknod /hotbck4/ora_backup/3Bbf.pipe p

.
.
.
mknod /hotbck4/ora_backup/12Abf.pipe p
mknod /hotbck4/ora_backup/12Bbf.pipe p
```

IV.4.1.3 /oracle/home/gen_dd_resp.sql

```
set pages 0
set linesize 120
set feedback off
```

```
spool /hotbck1/HOTBACKUP/dd_resp.sh
```

```
select 'dd if=||file_name||' of=||
decode(mod(file_id,4),0
'/hotbck1/ora_backup/'||file_id||'.pipe bs=10240k &||chr(10)
||'cat /hotbck1/ora_backup/'||file_id||'.pipe | compress -c ->
'/hotbck1/ora_backup/'||file_id||'.dbf.Z &' ,1,
'/hotbck2/ora_backup/'||file_id||'.pipe bs=10240k &||chr(10)
||'cat /hotbck2/ora_backup/'||file_id||'.pipe | compress -c ->
'/hotbck2/ora_backup/'||file_id||'.dbf.Z &' ,2,
'/hotbck3/ora_backup/'||file_id||'.pipe bs=10240k &||chr(10)
||'cat /hotbck3/ora_backup/'||file_id||'.pipe | compress -c ->
'/hotbck3/ora_backup/'||file_id||'.dbf.Z &' ,3,
'/hotbck4/ora_backup/'||file_id||'.pipe bs=10240k &||chr(10)
||'cat /hotbck4/ora_backup/'||file_id||'.pipe | compress -c ->
'/hotbck4/ora_backup/'||file_id||'.dbf.Z &')||chr(10)
||decode(mod(file_id,50),0,'wait',null)
from dba_data_files
order by file_id
/
```

```
spool off
exit
```



```
/hotbck1/HOTBACKUP/gen_dd_resp_logs.sql
```

```
set pages 0  
set linesize 120  
set feedback off
```

```
spool /hotbck1/HOTBACKUP/dd_resp_logs.sh
```

```
select 'dd if='||member||' of='||  
decode(mod(group#,4),0,  
'/hotbck1/ora_backup'||group#||'A'||substr(member,length(member)-1,2))||'.pipe  
bs=10240k &')||chr(10)  
||'cat /hotbck1/ora_backup'||group#||'A'||substr(member,length(member)-1,2))||'.pipe |  
compress -c -> /hotbck1/ora_backup'||group#||'A'||substr(member,length(member)-  
1,2))||'.dbf.Z &'.1,  
'/hotbck2/ora_backup'||group#||'A'||substr(member,length(member)-1,2))||'.pipe  
bs=10240k &')||chr(10)  
||'cat /hotbck2/ora_backup'||group#||'A'||substr(member,length(member)-1,2))||'.pipe |  
compress -c -> /hotbck2/ora_backup'||group#||'A'||substr(member,length(member)-  
1,2))||'.dbf.Z &'.2,  
'/hotbck3/ora_backup'||group#||'A'||substr(member,length(member)-1,2))||'.pipe  
bs=10240k &')||chr(10)  
||'cat /hotbck3/ora_backup'||group#||'A'||substr(member,length(member)-1,2))||'.pipe |  
compress -c -> /hotbck3/ora_backup'||group#||'A'||substr(member,length(member)-  
1,2))||'.dbf.Z &'.3,  
'/hotbck4/ora_backup'||group#||'A'||substr(member,length(member)-1,2))||'.pipe  
bs=10240k &')||chr(10)  
||'cat /hotbck4/ora_backup'||group#||'A'||substr(member,length(member)-1,2))||'.pipe |  
compress -c -> /hotbck4/ora_backup'||group#||'A'||substr(member,length(member)-  
1,2))||'.dbf.Z &'.4,  
from v$logfile  
where member like '/logspayA/oradata/ppay/%'  
/
```

```
select 'dd if='||member||' of='||  
decode(mod(group#,4),0,  
'/hotbck1/ora_backup'||group#||'B'||substr(member,length(member)-1,2))||'.pipe  
bs=10240k &')||chr(10)  
||'cat /hotbck1/ora_backup'||group#||'B'||substr(member,length(member)-1,2))||'.pipe |  
compress -c -> /hotbck1/ora_backup'||group#||'B'||substr(member,length(member)-  
1,2))||'.dbf.Z &'.1,  
'/hotbck2/ora_backup'||group#||'B'||substr(member,length(member)-1,2))||'.pipe  
bs=10240k &')||chr(10)  
||'cat /hotbck2/ora_backup'||group#||'B'||substr(member,length(member)-1,2))||'.pipe |  
compress -c -> /hotbck2/ora_backup'||group#||'B'||substr(member,length(member)-  
1,2))||'.dbf.Z &'.2,  
'/hotbck3/ora_backup'||group#||'B'||substr(member,length(member)-1,2))||'.pipe  
bs=10240k &')||chr(10)  
||'cat /hotbck3/ora_backup'||group#||'B'||substr(member,length(member)-1,2))||'.pipe |  
compress -c -> /hotbck3/ora_backup'||group#||'B'||substr(member,length(member)-  
1,2))||'.dbf.Z &'.3,  
'/hotbck4/ora_backup'||group#||'B'||substr(member,length(member)-1,2))||'.pipe  
bs=10240k &')||chr(10)
```



```
!|'cat /hotbck4/ora_backup/|group#||'B'||substr(member,length(member)-1,2)||'.pipe |
compress -c -> /hotbck4/ora_backup/|group#||'B'||substr(member,length(member)-
1,2)||'.dbf.Z &')
from v$logfile
where member like '/logsppayB/oradata/ppay/%'
/
```

```
spool off
exit
```

/oracle/home/dd_resp.sh

El contenido de este script es extenso debido a que contiene la forma de respaldo de los 320 datafiles que componen la Base de Datos, para efectos de simplificación solo se muestran 5 de ellos:

```
dd if=/u16/oradata/ppay/sysppay.dbf of=/hotbck2/ora_backup/1.pipe bs=10240k &
cat /hotbck2/ora_backup/1.pipe | compress -c -> /hotbck2/ora_backup/1.dbf.Z &

dd if=/u39/oradata/ppay/idata1727ppay.dbf of=/hotbck3/ora_backup/2.pipe bs=10240k &
cat /hotbck3/ora_backup/2.pipe | compress -c -> /hotbck3/ora_backup/2.dbf.Z &

dd if=/u27/oradata/ppay/idata0813ppay.dbf of=/hotbck4/ora_backup/3.pipe bs=10240k &
cat /hotbck4/ora_backup/3.pipe | compress -c -> /hotbck4/ora_backup/3.dbf.Z &

dd if=/u39/oradata/ppay/idata1740ppay.dbf of=/hotbck1/ora_backup/4.pipe bs=10240k &
cat /hotbck1/ora_backup/4.pipe | compress -c -> /hotbck1/ora_backup/4.dbf.Z &

dd if=/u49/oradata/ppay/iindex0116ppay.dbf of=/hotbck2/ora_backup/5.pipe bs=10240k &
cat /hotbck2/ora_backup/5.pipe | compress -c -> /hotbck2/ora_backup/5.dbf.Z &
```

/hotbck1/HOTBACKUP/dd_resp_logs.sh

```
dd if=/logsppayA/oradata/ppay/log7ppay.dbf of=/hotbck4/ora_backup/7Abf.pipe bs=10240k &
cat /hotbck4/ora_backup/7Abf.pipe | compress -c -> /hotbck4/ora_backup/7Abf.dbf.Z &

dd if=/logsppayA/oradata/ppay/log12ppay.dbf of=/hotbck1/ora_backup/12Abf.pipe bs=10240k &
cat /hotbck1/ora_backup/12Abf.pipe | compress -c -> /hotbck1/ora_backup/12Abf.dbf.Z &

dd if=/logsppayA/oradata/ppay/log8ppay.dbf of=/hotbck1/ora_backup/8Abf.pipe bs=10240k &
cat /hotbck1/ora_backup/8Abf.pipe | compress -c -> /hotbck1/ora_backup/8Abf.dbf.Z &

dd if=/logsppayA/oradata/ppay/log10ppay.dbf of=/hotbck3/ora_backup/10Abf.pipe bs=10240k &
cat /hotbck3/ora_backup/10Abf.pipe | compress -c -> /hotbck3/ora_backup/10Abf.dbf.Z &

dd if=/logsppayA/oradata/ppay/log9ppay.dbf of=/hotbck2/ora_backup/9Abf.pipe bs=10240k &
cat /hotbck2/ora_backup/9Abf.pipe | compress -c -> /hotbck2/ora_backup/9Abf.dbf.Z &

dd if=/logsppayA/oradata/ppay/log11ppay.dbf of=/hotbck4/ora_backup/11Abf.pipe bs=10240k &
cat /hotbck4/ora_backup/11Abf.pipe | compress -c -> /hotbck4/ora_backup/11Abf.dbf.Z &

dd if=/logsppayB/oradata/ppay/log7ppay.dbf of=/hotbck4/ora_backup/7Bbf.pipe bs=10240k &
cat /hotbck4/ora_backup/7Bbf.pipe | compress -c -> /hotbck4/ora_backup/7Bbf.dbf.Z &
```



```
dd if=/logsppayB/oradata/ppay/log8ppay.dbf of=/hotbck1/ora_backup/8Bbf.pipe bs=10240k &
cat /hotbck1/ora_backup/8Bbf.pipe | compress -c -> /hotbck1/ora_backup/8Bbf.dbf.Z &

dd if=/logsppayB/oradata/ppay/log10ppay.dbf of=/hotbck3/ora_backup/10Bbf.pipe bs=10240k &
cat /hotbck3/ora_backup/10Bbf.pipe | compress -c -> /hotbck3/ora_backup/10Bbf.dbf.Z &

dd if=/logsppayB/oradata/ppay/log9ppay.dbf of=/hotbck2/ora_backup/9Bbf.pipe bs=10240k &
cat /hotbck2/ora_backup/9Bbf.pipe | compress -c -> /hotbck2/ora_backup/9Bbf.dbf.Z &

rd if=/logsppayB/oradata/ppay/log11ppay.dbf of=/hotbck4/ora_backup/11Bbf.pipe bs=10240k &
cat /hotbck4/ora_backup/11Bbf.pipe | compress -c -> /hotbck4/ora_backup/11Bbf.dbf.Z &

dd if=/logsppayB/oradata/ppay/log12ppay.dbf of=/hotbck1/ora_backup/12Bbf.pipe bs=10240k &
cat /hotbck1/ora_backup/12Bbf.pipe | compress -c -> /hotbck1/ora_backup/12Bbf.dbf.Z &
```

IV.4.1.4 /hotbck1/HOTBACKUP/borra_pipes.sh

```
rm /hotbck1/ora_backup/*.pipe
rm /hotbck2/ora_backup/*.pipe
rm /hotbck3/ora_backup/*.pipe
rm /hotbck4/ora_backup/*.pipe
```

IV.4.1.5 /hotbck1/HOTBACKUP/gen_endbackup.sql

```
set pages 0
set feedback off
```

```
..pool /hotbck1/HOTBACKUP/endbackup.sql
select 'alter tablespace '|tablespace_name||' end backup;'
from dba_tablespaces
order by tablespace_name;
select 'exit' from dual;
spool off
exit
```

/hotbck1/HOTBACKUP/endbackup.sql

```
alter tablespace IDATA01 end backup;
alter tablespace IDATA02 end backup;
alter tablespace IDATA03 end backup;
alter tablespace IDATA04 end backup;
alter tablespace IDATA05 end backup;
alter tablespace IDATA06 end backup;
alter tablespace IDATA07 end backup;
alter tablespace IDATA08 end backup;
alter tablespace IDATA09 end backup;
alter tablespace IDATA10 end backup;
alter tablespace IDATA11 end backup;
alter tablespace IDATA12 end backup;
alter tablespace IDATA13 end backup;
alter tablespace IDATA14 end backup;
alter tablespace IDATA15 end backup;
alter tablespace IDATA16 end backup;
alter tablespace IDATA17 end backup;
```

TEBIS CON
FALLA DE ORIGEN



```
alter tablespace IDATA18 end backup;
alter tablespace IDATA19 end backup;
alter tablespace IINDEX01 end backup;
alter tablespace IINDEX02 end backup;
alter tablespace IINDEX03 end backup;
alter tablespace IINDEX04 end backup;
alter tablespace IINDEX05 end backup;
alter tablespace IRPT01 end backup;
alter tablespace SYSTEM end backup;
alter tablespace TEMP_APP end backup;
alter tablespace TRBS01 end backup;
alter tablespace TRBS02 end backup;
alter tablespace TRBS03 end backup;
alter tablespace TRBS04 end backup;
exit
```

!V.4.2.1 Transferencia de Datos.

A continuación se describe la forma de transferencia, la cual fue proporcionada por parte del Area de Sistema Operativo:

1.- Desatachar el Plex del volumen en Dominio1

```
$ vxplex -g oradg dis hotbck1_01
$ vxplex -g oradg dis hotbck2_01
$ vxplex -g oradg dis hotbck3_01
$ vxplex -g oradg dis hotbck4_01
```

2.- Crear nuevo volumen en Dominio1

```
$ vxmake -g oradg -V fsgen vol hotbck1new
$ vxmake -g oradg -V fsgen vol hotbck2new
$ vxmake -g oradg -V fsgen vol hotbck3new
$ vxmake -g oradg -V fsgen vol hotbck4new
```

3.- Asociar el plex al nuevo volumen en Dominio1

```
$ vxplex -g oradg att hotbck1new hotbck1_01
$ vxplex -g oradg att hotbck1new hotbck2_01
$ vxplex -g oradg att hotbck1new hotbck3_01
$ vxplex -g oradg att hotbck1new hotbck4_01
```

4.- Habilitar el nuevo volumen en Dominio1

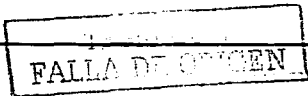
```
$ vxvol -g oradg start hotbck1new
$ vxvol -g oradg start hotbck2new
$ vxvol -g oradg start hotbck3new
$ vxvol -g oradg start hotbck4new
```

5.- Asignación del Tamaño al Nuevo Volumen en Dominio1

```
$ vxvol -g oradg set len=longitud hotbck1new
$ vxvol -g oradg set len=longitud hotbck2new
$ vxvol -g oradg set len=longitud hotbck3new
$ vxvol -g oradg set len=longitud hotbck4new
```

6.- Crear y Validar File Systems en Dominio1

```
$ fsck -F vxfs /dev/vxvdsd/oradg/hotbck1new
```





```
$ fsck -F vxfs /dev/vx/vdsk/oradg/hotbck2new
$ fsck -F vxfs /dev/vx/vdsk/oradg/hotbck3new
$ fsck -F vxfs /dev/vx/vdsk/oradg/hotbck4new
```

7.- Guardar configuración del volumen en Dominio1

```
$ vxprint -g oradg -hmvps hotbck1 > /home/root/SHELLS/hotbck1_vol.txt
$ vxprint -g oradg -hmvps hotbck2 > /home/root/SHELLS/hotbck2_vol.txt
$ vxprint -g oradg -hmvps hotbck3 > /home/root/SHELLS/hotbck3_vol.txt
$ vxprint -g oradg -hmvps hotbck4 > /home/root/SHELLS/hotbck4_vol.txt
```

8.- Crear nuevo grupo en Dominio2

```
$ vxdg -n oradgnew
```

9.- Remover plexes del grupo en Dominio1.

```
$ vxedit -g oradg -rf rm /hotbck1new
$ vxedit -g oradg -rf rm /hotbck2new
$ vxedit -g oradg -rf rm /hotbck3new
$ vxedit -g oradg -rf rm /hotbck4new
```

10.- Remover discos del grupo en Dominio1

```
$ vxdg -g oradg rmdisk nombrelogico del disco
```

11.- Agregar Discos al nuevo grupo en Dominio2

```
$ vxdg -g oradgnew adddisk nombrelógico nombrefísico
```

12.- Crear nuevos volúmenes en Dominio2

```
$ vxmake -g oradgnew -d /home/root/SHELLS/hotbck1_vol.txt
$ vxmake -g oradgnew -d /home/root/SHELLS/hotbck2_vol.txt
$ vxmake -g oradgnew -d /home/root/SHELLS/hotbck3_vol.txt
$ vxmake -g oradgnew -d /home/root/SHELLS/hotbck4_vol.txt
```

13.- Asociar el nuevo grupo al Dominio2

```
$ vxdg -import oradgnew
```

14.- Verificar que los volúmenes se encuentren activos.

```
$ vxprint -lvh
```

15.- Montar los volúmenes a los file systems

```
$ mount -f vxfs /dev/vx/oradgnew/hotbck1new /hotbck1
$ mount -f vxfs /dev/vx/oradgnew/hotbck2new /hotbck2
$ mount -f vxfs /dev/vx/oradgnew/hotbck3new /hotbck3
$ mount -f vxfs /dev/vx/oradgnew/hotbck4new /hotbck4
```

Plex. Porciones de espacio físico en Disco
Volumen. Conjunto de Plexes.

IV.4.2.2 /hotbck1/HOTBACKUP/gen_dd_restor_dbf.sql

Scripts que generan dd's para restauración de datafiles de la Base de Datos.

set pages 0



```
set linesize 120
set feedback off
```

```
spool /hotbck1/HOTBACKUP/dd_restor_dbf.sh
```

```
select 'zcat '|decode(mod(file_id,4),0,'/hotbck1/ora_backup/'||file_id||'.dbf.Z >
'|'/hotbck1/ora_backup/'||file_id||'.pipe &'||chr(10)
||'dd if=/hotbck1/ora_backup/'||file_id||'.pipe of= '|file_name||' bs=10240k &' ,1,
'/hotbck2/ora_backup/'||file_id||'.dbf.Z > '|'/hotbck2/ora_backup/'||file_id||'.pipe &'||chr(10)
||'dd if=/hotbck2/ora_backup/'||file_id||'.pipe of= '|file_name||' bs=10240k &' ,2,
'/hotbck3/ora_backup/'||file_id||'.dbf.Z > '|'/hotbck3/ora_backup/'||file_id||'.pipe &'||chr(10)
||'dd if=/hotbck3/ora_backup/'||file_id||'.pipe of= '|file_name||' bs=10240k &' ,3,
'/hotbck4/ora_backup/'||file_id||'.dbf.Z > '|'/hotbck4/ora_backup/'||file_id||'.pipe &'||chr(10)
||'dd if=/hotbck4/ora_backup/'||file_id||'.pipe of= '|file_name||' bs=10240k &' )
from dba_data_files
order by mod(file_id,4)
/
```

```
spool off
exit
```

```
/hotbck1/HOTBACKUP/gen_dd_restor_logs.sh
```

Scripts que generan dd's para restauración de logfiles de la Base de Datos.

```
set pages 0
set linesize 120
set feedback off
```

```
spool /hotbck1/HOTBACKUP/dd_restor_logs.sh
```

```
select 'zcat '|decode(mod(group#,4),0,
'/hotbck1/ora_backup/'||group#||'A' ||substr(member,length(member)-1,2)||'.dbf.Z >
'|'/hotbck1/ora_backup/'||group#||'A' ||substr(member,length(member)-1,2)||'.pipe &'||chr(10)
||'dd if=/hotbck1/ora_backup/'||group#||'A' ||substr(member,length(member)-1,2)||'.pipe
of= '|member||' bs=10240k &' ,1,
'/hotbck2/ora_backup/'||group#||'A' ||substr(member,length(member)-1,2)||'.dbf.Z >
'|'/hotbck2/ora_backup/'||group#||'A' ||substr(member,length(member)-1,2)||'.pipe &'||chr(10)
||'dd if=/hotbck2/ora_backup/'||group#||'A' ||substr(member,length(member)-1,2)||'.pipe
of= '|member||' bs=10240k &' ,2,
'/hotbck3/ora_backup/'||group#||'A' ||substr(member,length(member)-1,2)||'.dbf.Z >
'|'/hotbck3/ora_backup/'||group#||'A' ||substr(member,length(member)-1,2)||'.pipe &'||chr(10)
||'dd if=/hotbck3/ora_backup/'||group#||'A' ||substr(member,length(member)-1,2)||'.pipe
of= '|member||' bs=10240k &' ,3,
'/hotbck4/ora_backup/'||group#||'A' ||substr(member,length(member)-1,2)||'.dbf.Z >
'|'/hotbck4/ora_backup/'||group#||'A' ||substr(member,length(member)-1,2)||'.pipe &'||chr(10)
||'dd if=/hotbck4/ora_backup/'||group#||'A' ||substr(member,length(member)-1,2)||'.pipe
of= '|member||' bs=10240k &' )
from v$logfile
where member like '/logspayA/oradata/ppay/%'
/
```

```
select 'zcat '|decode(mod(group#,4),0,
'/hotbck1/ora_backup/'||group#||'B' ||substr(member,length(member)-1,2)||'.dbf.Z >
'|'/hotbck1/ora_backup/'||group#||'B' ||substr(member,length(member)-1,2)||'.pipe &'||chr(10)
```

**TESIS CON
FALLA DE ORIGEN**

```

||'dd if=/hotbck1/ora_backup/'||group#||'B'||substr(member,length(member)-1,2)||'.pipe
of=||member||' bs=10240k &'.1,
'/hotbck2/ora_backup/'||group#||'B'||substr(member,length(member)-1,2)||'.dbf.Z >
'||/hotbck2/ora_backup/'||group#||'B'||substr(member,length(member)-1,2)||'.pipe &'||chr(10)
||'dd if=/hotbck2/ora_backup/'||group#||'B'||substr(member,length(member)-1,2)||'.pipe
of=||member||' bs=10240k &'.2,
'/hotbck3/ora_backup/'||group#||'B'||substr(member,length(member)-1,2)||'.dbf.Z >
'||/hotbck3/ora_backup/'||group#||'B'||substr(member,length(member)-1,2)||'.pipe &'||chr(10)
||'dd if=/hotbck3/ora_backup/'||group#||'B'||substr(member,length(member)-1,2)||'.pipe
of=||member||' bs=10240k &'.3,
'/hotbck4/ora_backup/'||group#||'B'||substr(member,length(member)-1,2)||'.dbf.Z >
'||/hotbck4/ora_backup/'||group#||'B'||substr(member,length(member)-1,2)||'.pipe &'||chr(10)
||'dd if=/hotbck4/ora_backup/'||group#||'B'||substr(member,length(member)-1,2)||'.pipe
of=||member||' bs=10240k &'
from v$logfile
where member like '/logsppayB/oradata/ppay/%'
/
spool off
exit

```

IV.4.2.2 /hotbck1/HOTBACKUP/dd_restor_dbf.sh

El contenido de este script es extenso debido a que contiene la forma de restauración de los 320 datafiles en el dominio2 que componen la Base de Datos, para efectos de simplificación solo se muestran 5 de ellos:

```

zcat /hotbck1/ora_backup/4.dbf.Z > /hotbck1/ora_backup/4.pipe &
dd if=/hotbck1/ora_backup/4.pipe of=/u39/oradata/ppay/idata1740ppay.dbf bs=10240k &

zcat /hotbck1/ora_backup/24.dbf.Z > /hotbck1/ora_backup/24.pipe &
rd if=/hotbck1/ora_backup/24.pipe of=/temppay/oradata/ppay/temp_app08ppay.dbf bs=10240k &

zcat /hotbck1/ora_backup/216.dbf.Z > /hotbck1/ora_backup/216.pipe &
dd if=/hotbck1/ora_backup/216.pipe of=/u24/oradata/ppay/iindex053ppay.dbf bs=10240k &

zcat /hotbck1/ora_backup/208.dbf.Z > /hotbck1/ora_backup/208.pipe &
dd if=/hotbck1/ora_backup/208.pipe of=/u24/oradata/ppay/iindex051ppay.dbf bs=10240k &

zcat /hotbck1/ora_backup/204.dbf.Z > /hotbck1/ora_backup/204.pipe &
dd if=/hotbck1/ora_backup/204.pipe of=/u29/oradata/ppay/idata099ppay.dbf bs=10240k &

```

/hotbck1/HOTBACKUP/dd_restor_logs.sh

```

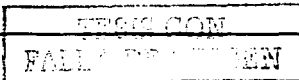
zcat /hotbck4/ora_backup/7Abf.dbf.Z > /hotbck4/ora_backup/7Abf.pipe &
dd if=/hotbck4/ora_backup/7Abf.pipe of=/logsppayA/oradata/ppay/log7ppay.dbf bs=10240k &

zcat /hotbck1/ora_backup/12Abf.dbf.Z > /hotbck1/ora_backup/12Abf.pipe &
dd if=/hotbck1/ora_backup/12Abf.pipe of=/logsppayA/oradata/ppay/log12ppay.dbf bs=10240k &

zcat /hotbck1/ora_backup/8Abf.dbf.Z > /hotbck1/ora_backup/8Abf.pipe &
dd if=/hotbck1/ora_backup/8Abf.pipe of=/logsppayA/oradata/ppay/log8ppay.dbf bs=10240k &

zcat /hotbck3/ora_backup/10Abf.dbf.Z > /hotbck3/ora_backup/10Abf.pipe &
dd if=/hotbck3/ora_backup/10Abf.pipe of=/logsppayA/oradata/ppay/log10ppay.dbf bs=10240k &

```





```
zcat /hotbck2/ora_backup/9Abf.dbf.Z > /hotbck2/ora_backup/9Abf.pipe &
dd if=/hotbck2/ora_backup/9Abf.pipe of=/logsppayA/oradata/ppay/log9ppay.dbf bs=10240k &

zcat /hotbck4/ora_backup/11Abf.dbf.Z > /hotbck4/ora_backup/11Abf.pipe &
dd if=/hotbck4/ora_backup/11Abf.pipe of=/logsppayA/oradata/ppay/log11ppay.dbf bs=10240k &

zcat /hotbck4/ora_backup/7Bbf.dbf.Z > /hotbck4/ora_backup/7Bbf.pipe &
dd if=/hotbck4/ora_backup/7Bbf.pipe of=/logsppayB/oradata/ppay/log7ppay.dbf bs=10240k &

zcat /hotbck1/ora_backup/8Bbf.dbf.Z > /hotbck1/ora_backup/8Bbf.pipe &
dd if=/hotbck1/ora_backup/8Bbf.pipe of=/logsppayB/oradata/ppay/log8ppay.dbf bs=10240k &

zcat /hotbck3/ora_backup/10Bbf.dbf.Z > /hotbck3/ora_backup/10Bbf.pipe &
dd if=/hotbck3/ora_backup/10Bbf.pipe of=/logsppayB/oradata/ppay/log10ppay.dbf bs=10240k &

zcat /hotbck2/ora_backup/9Bbf.dbf.Z > /hotbck2/ora_backup/9Bbf.pipe &
dd if=/hotbck2/ora_backup/9Bbf.pipe of=/logsppayB/oradata/ppay/log9ppay.dbf bs=10240k &

zcat /hotbck4/ora_backup/11Bbf.dbf.Z > /hotbck4/ora_backup/11Bbf.pipe &
dd if=/hotbck4/ora_backup/11Bbf.pipe of=/logsppayB/oradata/ppay/log11ppay.dbf bs=10240k &

zcat /hotbck1/ora_backup/12Bbf.dbf.Z > /hotbck1/ora_backup/12Bbf.pipe &
dd if=/hotbck1/ora_backup/12Bbf.pipe of=/logsppayB/oradata/ppay/log12ppay.dbf bs=10240k &
```

IV.4.3.1 Sincronización de Base De Datos

```
S svrmgrl
SVRMGR> connect internal

SVRMGR> recover database using backup controlfile until cancel;
```

La ejecución del comando anterior inicia la recuperación de la Base de Datos por medio de la aplicación de archivos " archive ". La solicitud de estos archivos se muestra como sigue:

```
ORA-00279: change 3145633785 generated at 10/27/2001 00:05:24 needed for thread 1
ORA-00289: suggestion : /oradata/1_7.dbf
ORA-00280: change 3145633785 for thread 1 is in sequence #8
Specify log: (<RET>=suggested | filename | AUTO | CANCEL)
/oradata/1_7.dbf
```

```
ORA-00279: change 3145633785 generated at 10/27/2001 00:15:24 needed for thread 1
ORA-00289: suggestion : /oradata/1_8.dbf
ORA-00280: change 3145633785 for thread 1 is in sequence #8
Specify log: (<RET>=suggested | filename | AUTO | CANCEL)
/oradata/1_8.dbf
```

De aquí en adelante se tienen que aplicar los archivos hasta el punto que se desee.

- Terminar Recuperación



Al aplicar todos los archivos deseados se procede a terminar la recuperación con el siguiente comando.

```
ORA-00279: change 3145633785 generated at 10/27/2001 00:25:24 needed for thread 1
ORA-00289: suggestion : /oradata/1_9.dbf
ORA-00280: change 3145633785 for thread 1 is in sequence #9
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
CANCEL
```

IV.5.2.1

La ejecución de comando de conversión de la aplicación muestra las tablas que serán modificadas además de los tablespaces en donde viven:

| Tablespace name | # bytes | # used | # free | %free |
|-----------------|-----------|-----------|-----------|-------|
| DDATA01 | 104857600 | 679936 | 104177664 | 99,4 |
| DDATA01 | 209715200 | 237568 | 209477632 | 99,9 |
| DDATA01 | 314572800 | 57344 | 314515456 | 100 |
| DDATA01 | 524288000 | 8192 | 524279808 | 100 |
| DDATA01 | 524288000 | 166633472 | 357654528 | 68,2 |
| DDATA01 | 734003200 | 8192 | 733995008 | 100 |
| DDATA01 | 209715200 | 8192 | 209707008 | 100 |
| DDATA01 | 104857600 | 102817792 | 2039808 | 1,9 |
| DDATA02 | 104857600 | 10493952 | 94363648 | 90 |
| DDATA02 | 209715200 | 8192 | 209707008 | 100 |
| DDATA02 | 104857600 | 10493952 | 94363648 | 90 |
| DDATA02 | 104857600 | 20979712 | 83877888 | 80 |
| DDATA02 | 104857600 | 8192 | 104849408 | 100 |
| DDATA03 | 209715200 | 184827904 | 24887296 | 11,9 |
| DDATA03 | 209715200 | 200302592 | 9412608 | 4,5 |
| DDATA04 | 104857600 | 22945792 | 81911808 | 78,1 |
| IINDEX01 | 104857600 | 16801792 | 88055808 | 84 |
| IINDEX01 | 104857600 | 52436992 | 52420608 | 50 |
| IINDEX02 | 104857600 | 57679872 | 47177728 | 45 |
| IINDEX02 | 209715200 | 8192 | 209707008 | 100 |
| IINDEX03 | 104857600 | 103825408 | 1032192 | 1 |
| IINDEX03 | 209715200 | 195043328 | 14671872 | 7 |
| IINDEX03 | 104857600 | 95649792 | 9207808 | 8,8 |
| IINDEX03 | 209715200 | 78651392 | 131063808 | 62,5 |
| IINDEX04 | 104857600 | 76095488 | 28762112 | 27,4 |
| ITRAIN | 104857600 | 8192 | 104849408 | 100 |
| RBS | 419430400 | 19177472 | 400252928 | 95,4 |
| SYSTEM | 268435456 | 53321728 | 215113728 | 80,1 |
| TEMP | 157286400 | 417792 | 156868608 | 99,7 |

30 rows selected.

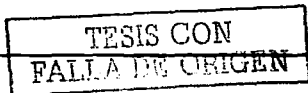
Las siguientes tablas de gran crecimiento y los índices son localizados en el tablespace:

Tablespace DDATA02 & IINDEX02
CARGO_DE_LLAMADAS y DETALLE_DE_LLAMADAS

Usar el tbs por Default (Y/N) : N

Introduce tbs por tablas:

Introduce tbs por Indices:



**IDATA16
IINDEX03**

Tablespace DDATA03 & IINDEX03
AJUSTE_DETALLE_DE_LLAMADAS
STATUS
CIRCULO_DE_LLAMADAS
SUBSCRIPTORES
Usar el tbs por Default (Y/N) : N
Introduce tbs por tablas:
Introduce tbs por indices:

**IDATA19
IINDEX06**

Tablespace DDATA04 IINDEX04
PROFILE
NOMBRE_DE_CARGO_DE_LLAMADAS
PLAN_DE_MARCACION
DESCUENTO_CIRCULO_DE_LLAMADA
TIPO_DE_LOCALIZACION_DE_LLAMADA
REGLAS_PLAN_DE_MARCACION
CODIGO_GL
Usar el tbs por Default (Y/N) : N
Introduce tbs por tablas:
Introduce tbs por indices:

**IDATA02
IINDEX02****IV.5.3.1**

```
$ more fichas_A.sh
echo "<< BEGIN fichas1.sql >> ' >> nohup.out
date >> nohup.out
sqlplus -s <user> /<password> @fichas1.sql
compress /u36/oradata/DESC_MIGRA_PCI9/R9_fichas1.unl
echo << END fichas1.sql >> ' >> nohup.out
date >> nohup.out
```

\$ more fichas1.sql

```
ALTER SESSION SET NLS_DATE_FORMAT='YYYY-MM-DD HH24:MI:SS';
SET HEADING OFF
SET HEADSEP OFF
SET TERMOUT OFF
SET VERIFY OFF
SET FEEDBACK OFF
SET SERVEROUTPUT ON
SET TRIMOUT ON
SET TRIMSPOOL ON
SET NULL ""
SET LINESIZE 999
SET PAGESIZE 0
```

TESIS CON
FALLA DE ORIGEN



SPOOL /u36/oradata/DESC_MIGRA_PCI9/R9_fichas1.uni

```

SELECT
P.CARD_CODE
P.SEQUENCE_NUMBER
P.ACCESS_NUMBER
P.RETAILER_ID
substr(P.ACCESS_CODE,1,13)
P.AMOUNT_EXPIRATION_PERIOD
P.CARD_EXPIRATION_DATE
P.CARD_AMOUNT
P.STATUS
P.USED_DATE
P.DISTRIBUTE_DATE
P.LAST_STATUS_DATE
P.TERMINATION_REASON_CODE
P.CREATION_DATE
P.UPDATED_WHEN
P.UPDATED_BY
P.FK_BATCH_NUMBER
P.FK_COMPANY_NUMBER
P.OP_UNIT_ID
FROM FICHAS P
WHERE TO_CHAR(P.CREATION_DATE,'YYYY') = '1999'
/
SPOOL OFF
CLEAR COLUMNS

```

\$ more fichas_B.sh

```

echo '<< BEGIN fichas2.sql >>' >> nohup.out
date >> nohup.out
sqlplus -s <user>/<password> @fichas2.sql
echo '<< END fichas2.sql >>' >> nohup.out
date >> nohup.out

```

\$ more fichas2.sql

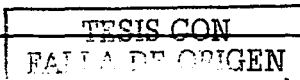
ALTER SESSION SET NLS_DATE_FORMAT='YYYY-MM-DD HH24:MI:SS';

```

SET HEADING OFF
SET HEADSEP OFF
SET TERMOUT OFF
SET VERIFY OFF
SET FEEDBACK OFF
SET SERVEROUTPUT ON
SET TRIMOUT ON
SET TRIMSPOOL ON
SET NULL ""
SET LINESIZE 999
SET PAGESIZE 0

```

SPOOL /u36/oradata/DESC_MIGRA_PCI9/R9_fichas2.uni





```
SELECT
P.CARD_CODE
P.SEQUENCE_NUMBER
P.ACCESS_NUMBER
P.RETAILER_ID
substr(P.ACCESS_CODE,1,13)
P.AMOUNT_EXPIRATION_PERIOD
P.CARD_EXPIRATION_DATE
P.CARD_AMOUNT
P.STATUS
P.USED_DATE
P.DISTRIBUTE_DATE
P.LAST_STATUS_DATE
P.TERMINATION_REASON_CODE
P.CREATION_DATE
P.UPDATED_WHEN
P.UPDATED_BY
P.FK_BATCH_NUMBER
P.FK_COMPANY_NUMBER
P.OP_UNIT_ID
FROM FICHAS P
WHERE TO_CHAR(P.CREATION_DATE,'YYYYMMDD') between '20000101' and '20000430'
/
SPOOL OFF

CLEAR COLUMNS

EXIT
```

\$ more fichas_C.sh

```
echo '<< BEGIN fichas3.sql >>' >> nohup.out
date >> nohup.out
sqlplus -s <user/<password> @fichas3.sql
echo '<< END fichas3.sql >>' >> nohup.out
date >> nohup.out
```

\$ more fichas3.sql

```
ALTER SESSION SET NLS_DATE_FORMAT='YYYY-MM-DD HH24:MI:SS';
```

```
SET HEADING OFF
SET HEADSEP OFF
SET TERMOUT OFF
SET VERIFY OFF
SET FEEDBACK OFF
SET SERVEROUTPUT ON
SET TRIMOUT ON
SET TRIMSPOOL ON
SET NULL ""
SET LINESIZE 999
```




```
SET PAGESIZE 0
```

```
SPOOL /u36/oradata/DESC_MIGRA_PC19/R9_fichas3.unl
```

```
SELECT
P.CARD_CODE
P.SEQUENCE_NUMBER
P.ACCESS_NUMBER
P.RETAILER_ID
substr(P.ACCESS_CODE,1,13)
P.AMOUNT_EXPIRATION_PERIOD
P.CARD_EXPIRATION_DATE
P.CARD_AMOUNT
P.STATUS
P.USED_DATE
P.DISTRIBUTE_DATE
P.LAST_STATUS_DATE
P.TERMINATION_REASON_CODE
P.CREATION_DATE
P.UPDATED_WHEN
P.UPDATED_BY
P.FK_BATCH_NUMBER
P.FK_COMPANY_NUMBER
P.OP_UNIT_ID
FROM FICHAS P
WHERE TO_CHAR(P.CREATION_DATE,'YYYYMMDD') between '20000501' and '20000831'
/
SPOOL OFF
```

```
CLEAR COLUMNS
```

```
EXIT
```

```
$ more fichas_D.sh
```

```
echo '<< BEGIN fichas4.sql >>' >> nohup.out
date >> nohup.out
sqlplus -s <user>/<password> @fichas4.sql
echo '<< END fichas4.sql >>' >> nohup.out
date >> nohup.out
```

```
$ more fichas4.sql
```

```
ALTER SESSION SET NLS_DATE_FORMAT='YYYY-MM-DD HH24:MI:SS';
```

```
SET HEADING OFF
SET HEADSEP OFF
SET TERMOUT OFF
SET VERIFY OFF
SET FEEDBACK OFF
SET SERVEROUTPUT ON
SET TRIMOUT ON
SET TRIMSPOOL ON
```

TESIS CON
FALLA DE ORIGEN



```
SET NULL ""
SET LINESIZE 999
SET PAGESIZE 0
```

```
SPOOL /u36/oradata/DESC_MIGRA_PCI9/R9_fichas4.unl
```

```
SELECT
P.CARD_CODE
P.SEQUENCE_NUMBER
P.ACCESS_NUMBER
P.RETAILER_ID
substr(P.ACCESS_CODE,1,13)
P.AMOUNT_EXPIRATION_PERIOD
P.CARD_EXPIRATION_DATE
P.CARD_AMOUNT
P.STATUS
P.USED_DATE
P.DISTRIBUTE_DATE
P.LAST_STATUS_DATE
P.TERMINATION_REASON_CODE
P.CREATION_DATE
P.UPDATED_WHEN
P.UPDATED_BY
P.FK_BATCH_NUMBER
P.FK_COMPANY_NUMBER
P.OP_UNIT_ID
FROM FICHAS P
WHERE TO_CHAR(P.CREATION_DATE,'YYYYMMDD') between '20000901' and '20001231'
```

```
/
```

```
SPOOL OFF
```

```
CLEAR COLUMNS
```

```
EXIT
```

```
$ more fichas_E.sh
```

```
echo '<< BEGIN fichas5.sql >>' >> nohup.out
date >> nohup.out
sqlplus -s <user>/<password> @fichas5.sql
echo '<< END fichas5.sql >>' >> nohup.out
date >> nohup.out
```

```
$ more fichas5.sql
```

```
ALTER SESSION SET NLS_DATE_FORMAT='YYYY-MM-DD HH24:MI:SS';
```

```
SET HEADING OFF
SET HEADSEP OFF
SET TERMOUT OFF
SET VERIFY OFF
```



```

SET FEEDBACK OFF
SET SERVEROUTPUT ON
SET TRIMOUT ON
SET TRIMSPOOL ON
SET NULL ""
SET LINESIZE 999
SET PAGESIZE 0

```

SPOOL /u36/oradata/DESC_MIGRA_PCI9/R9_fichas5.unl

```

SELECT
P.CARD_CODE
P.SEQUENCE_NUMBER
P.ACCESS_NUMBER
P.RETAILER_ID
substr(P.ACCESS_CODE,1,13)
P.AMOUNT_EXPIRATION_PERIOD
P.CARD_EXPIRATION_DATE
P.CARD_AMOUNT
P.STATUS
P.USED_DATE
P.DISTRIBUTE_DATE
P.LAST_STATUS_DATE
P.TERMINATION_REASON_CODE
P.CREATION_DATE
P.UPDATED_WHEN
P.UPDATED_BY
P.FK_BATCH_NUMBER
P.FK_COMPANY_NUMBER
P.OP_UNIT_ID
FROM FICHAS P
WHERE TO_CHAR(P.CREATION_DATE,'YYYYMMDD') between '20010101' and '20010228'

```

```

/
SPOOL OFF
CLEAR COLUMNS
EXIT

```

\$ more fichas_F.sh

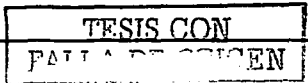
```

echo '<< BEGIN fichas6.sql >>' >> nohup.out
date >> nohup.out
sqlplus -s <user>/<password> @fichas6.sql
echo '<< END fichas6.sql >>' >> nohup.out
date >> nohup.out

```

\$ more fichas6.sql

ALTER SESSION SET NLS_DATE_FORMAT='YYYY-MM-DD HH24:MI:SS';



```

SET HEADING      OFF
SET HEADSEP     OFF
SET TERMOUT     OFF
SET VERIFY      OFF
SET FEEDBACK    OFF
SET SERVEROUTPUT ON
SET TRIMOUT     ON
SET TRIMSPOOL   ON
SET NULL        ""
SET LINESIZE    999
SET PAGESIZE    0
    
```

SPOOL /u36/oradata/DESC_MIGRA_PC19/R9_fichas6.uni

```

SELECT
P.CARD_CODE
P.SEQUENCE_NUMBER
P.ACCESS_NUMBER
P.RETAILER_ID
substr(P.ACCESS_CODE,1,13)
P.AMOUNT_EXPIRATION_PERIOD
P.CARD_EXPIRATION_DATE
P.CARD_AMOUNT
P.STATUS
P.USED_DATE
P.DISTRIBUTE_DATE
P.LAST_STATUS_DATE
P.TERMINATION_REASON_CODE
P.CREATION_DATE
P.UPDATED_WHEN
P.UPDATED_BY
P.FK_BATCH_NUMBER
P.FK_COMPANY_NUMBER
P.OP_UNIT_ID
FROM FICHAS P
'WHERE TO_CHAR(P.CREATION_DATE,'YYYYMMDD') between '20010301' and '20010430'
    
```

/
SPOOL OFF

CLEAR COLUMNS

EXIT

\$ more fichas_G.sh

```

echo '<< BEGIN fichas7.sql >>' >> nohup.out
date >> nohup.out
sqlplus -s <user>/<password> @fichas7.sql
echo '<< END fichas7.sql >>' >> nohup.out
date >> nohup.out
    
```

TESIS CON
FALLA DE ORIGEN

\$ more fichas7.sql

ALTER SESSION SET NLS_DATE_FORMAT=YYYY-MM-DD HH24:MI:SS;

```

SET HEADING      OFF
SET HEADSEP      OFF
SET TERMOUT      OFF
SET VERIFY       OFF
SET FEEDBACK     OFF
SET SERVEROUTPUT ON
SET TRIMOUT      ON
SET TRIMSPOOL    ON
SET NULL         ""
SET LINESIZE     999
SET PAGESIZE     0
    
```

SPOOL /u36/oradata/DESC_MIGRA_PC19/R9_fichas7.unl

```

SELECT
P.CARD_CODE                ||''||
P.SEQUENCE_NUMBER          ||''||
P.ACCESS_NUMBER            ||''||
P.RETAILER_ID              ||''||
substr(P.ACCESS_CODE,1,13) ||''||
P.AMOUNT_EXPIRATION_PERIOD ||''||
P.CARD_EXPIRATION_DATE     ||''||
P.CARD_AMOUNT              ||''||
P.STATUS                    ||''||
P.USED_DATE                 ||''||
P.DISTRIBUTE_DATE          ||''||
P.LAST_STATUS_DATE         ||''||
P.TERMINATION_REASON_CODE  ||''||
P.CREATION_DATE            ||''||
P.UPDATED_WHEN             ||''||
P.UPDATED_BY               ||''||
P.FK_BATCH_NUMBER          ||''||
P.FK_COMPANY_NUMBER        ||''||
P.OP_UNIT_ID                ||''||
FROM FICHAS P
WHERE TO_CHAR(P.CREATION_DATE,'YYYYMMDD') between '20010501' and '20010630'
    
```

/
SPOOL OFF

CLEAR COLUMNS

EXIT

\$ more fichas_H.sh

```

echo '<< BEGIN fichas8.sql >>' >> nohup.out
date >> nohup.out
sqlplus -s <user>/<password> @fichas8.sql
    
```

TESIS CON
FALLA DE ORIGEN



```
echo '<< END fichas8.sql >>' >> nohup.out  
date >> nohup.out
```

```
$ more fichas8.sql
```

```
ALTER SESSION SET NLS_DATE_FORMAT='YYYY-MM-DD HH24:MI:SS';
```

```
SET HEADING OFF  
SET HEADSEP OFF  
SET TERMOUT OFF  
SET VERIFY OFF  
SET FEEDBACK OFF  
SET SERVEROUTPUT ON  
SET TRIMOUT ON  
SET TRIMSPOOL ON  
SET NULL ''  
SET LINESIZE 999  
SET PAGESIZE 0
```

```
SPOOL /u36/oradata/DESC_MIGRA_PC19/R9_fichas8.unl
```

```
SELECT  
P.CARD_CODE  
P.SEQUENCE_NUMBER  
P.ACCESS_NUMBER  
P.RETAILER_ID  
substr(P.ACCESS_CODE,1,13)  
P.AMOUNT_EXPIRATION_PERIOD  
P.CARD_EXPIRATION_DATE  
P.CARD_AMOUNT  
P.STATUS  
P.USED_DATE  
P.DISTRIBUTE_DATE  
P.LAST_STATUS_DATE  
P.TERMINATION_REASON_CODE  
P.CREATION_DATE  
P.UPDATED_WHEN  
P.UPDATED_BY  
P.FK_BATCH_NUMBER  
P.FK_COMPANY_NUMBER  
P.OP_UNIT_ID  
FROM FICHAS P  
WHERE TO_CHAR(P.CREATION_DATE,'YYYYMMDD') between '20010701' and '20010930'
```

```
/
```

```
SPOOL OFF
```

```
CLEAR COLUMNS
```

```
EXIT
```

IV.5.3.2

\$more crea_fichas_inv_no_index.sql

```
DROP TABLE FICHAS CASCADE CONSTRAINTS ;

CREATE TABLE FICHAS (
  CARD_CODE          CHAR(16)  NOT NULL,
  SEQUENCE_NUMBER    NUMBER(8)  NOT NULL,
  ACCESS_NUMBER      NUMBER(15) NOT NULL,
  RETAILER_ID        NUMBER(8)  NOT NULL,
  ACCESS_CODE        CHAR(20)  DEFAULT ''
) NOT NULL,
  AMOUNT_EXPIRATION_PERIOD NUMBER(3)  NOT NULL,
  CARD_EXPIRATION_DATE   DATE          NOT NULL,
  CARD_AMOUNT            NUMBER(13,4) NOT NULL,
  STATUS                 NUMBER(2)    NOT NULL,
  USED_DATE              DATE          NOT NULL,
  DISTRIBUTE_DATE       DATE          NOT NULL,
  LAST_STATUS_DATE      DATE          NOT NULL,
  TERMINATION_REASON_CODE NUMBER(2)  NOT NULL,
  CREATION_DATE         DATE          NOT NULL,
  UPDATED_WHEN         DATE          NOT NULL,
  UPDATED_BY           CHAR(8)      NOT NULL,
  FK_BATCH_NUMBER      NUMBER(8),
  FK_COMPANY_NUMBER    NUMBER(2),
  OP_UNIT_ID           NUMBER(2)  DEFAULT 0 NOT NULL
)
TABLESPACE IDATA09 PCTFREE 30
STORAGE(INITIAL 102400K NEXT 102400K PCTINCREASE 0
MINEXTENTS 1 MAXEXTENTS 2147483645)
PARALLEL (DEGREE 1 INSTANCES 1) NOCACHE ;
```

IV.5.3.3 Validación de constraints e índices.

```
$ sqlplus system/xxxx
SQL> @check_constraints.sql
SQL> select CONSTRAINT_NAME, TABLE_NAME from dba_constraints
2 where TABLE_NAME = 'FICHAS' and owner = 'PP'
3 ;
```

| CONSTRAINT_NAME | TABLE_NAME |
|-----------------|------------|
| SYS_C005793 | FICHAS |
| SYS_C005794 | FICHAS |
| SYS_C005795 | FICHAS |
| SYS_C005796 | FICHAS |
| SYS_C005797 | FICHAS |
| SYS_C005798 | FICHAS |
| SYS_C005799 | FICHAS |
| SYS_C005800 | FICHAS |
| SYS_C005801 | FICHAS |
| SYS_C005802 | FICHAS |
| SYS_C005803 | FICHAS |



```
SYS_C005804      FICHAS
SYS_C005805      FICHAS
SYS_C005806      FICHAS
SYS_C005807      FICHAS
SYS_C005808      FICHAS
SYS_C005809      FICHAS
```

17 rows selected.

SQL> @check_index.sql

```
SQL> select index_name, table_name, tablespace_name from dba_indexes
 2 where table_name = 'FICHAS' AND OWNER = 'PP';
```

no rows selected

SQL> exit

IV.5.3.4 Cargar los datos de la tabla de fichas.

\$ more load_fichas.sh

```
nohup fichas.2000 lod &
nohup fichas.2000A lod &
nohup fichas.2000B lod &
nohup fichas.2001 lod &
nohup fichas.2001A lod &
nohup fichas.2001B lod &
nohup fichas.2001C lod &
```

\$ more fichas.2000 lod

```
sqlldr <user>/<password> data=/u39/oradata/DESC_ PCI3/R123_fichas_inventory1.unf
control=fichas.2000.ctl log=fichas_2000.log
```

\$ more fichas.2000.ctl

```
OPTIONS (DIRECT=TRUE, PARALLEL=TRUE)
LOAD DATA
APPEND
PRESERVE BLANKS
INTO TABLE fichas
FIELDS TERMINATED BY '|'
```

```
(CARD_CODE          CHAR(16),
SEQUENCE_NUMBER     INTEGER EXTERNAL,
ACCESS_NUMBER       INTEGER EXTERNAL,
RETAILER_ID         INTEGER EXTERNAL,
ACCESS_CODE         CHAR(20),
AMOUNT_EXPIRATION_PERIOD INTEGER EXTERNAL,
CARD_EXPIRATION_DATE DATE(19) "yyyy-mm-dd hh24:mi:ss",
CARD_AMOUNT         INTEGER EXTERNAL,
STATUS              INTEGER EXTERNAL,
USED_DATE           DATE(19) "yyyy-mm-dd hh24:mi:ss",
DISTRIBUTE_DATE     DATE(19) "yyyy-mm-dd hh24:mi:ss",
LAST_STATUS_DATE    DATE(19) "yyyy-mm-dd hh24:mi:ss",
TERMINATION_REASON_CODE INTEGER EXTERNAL,
```




```

CREATION_DATE      DATE(19) "yyyy-mm-dd hh24:mi:ss",
UPDATED_WHEN       DATE(19) "yyyy-mm-dd hh24:mi:ss",
UPDATED_BY         CHAR(8),
FK_BATCH_NUMBER    INTEGER EXTERNAL,
FK_COMPANY_NUMBER  INTEGER EXTERNAL,
OP_UNIT_ID         INTEGER EXTERNAL)
    
```

IV.5.3.5

\$ ps -fe|grep sqldr

```

oracle 7106 7098 0 11:35:53 pts/7 0:01 sqldr <user>/<password> data=/exports/y
oracle 7107 7099 0 11:35:53 pts/7 0:01 sqldr <user>/<password> data=/exports/y
oracle 7108 7101 0 11:35:53 pts/7 0:01 sqldr <user>/<password> data=/exports/y
oracle 7109 7100 0 11:35:53 pts/7 0:01 sqldr <user>/<password> data=/exports/y
oracle 7870 2239 0 11:36:15 pts/7 0:00 grep sqldr
oracle 7110 7102 0 11:35:53 pts/7 0:01 sqldr <user>/<password> data=/exports/y
oracle 7111 7103 0 11:35:53 pts/7 0:01 sqldr <user>/<password> data=/exports/y
    
```

\$ ls -lrt

```

-rw-r--r-- 1 oracle dba 0 Feb 23 11:35 fichas_2001.log
-rw-r--r-- 1 oracle dba 0 Feb 23 11:35 fichas_2001A.log
-rw-r--r-- 1 oracle dba 0 Feb 23 11:35 fichas_2001B.log
-rw-r--r-- 1 oracle dba 0 Feb 23 11:35 fichas_2000A.log
-rw-r--r-- 1 oracle dba 0 Feb 23 11:35 fichas_2000B.log
-rw-r--r-- 1 oracle dba 0 Feb 23 11:35 fichas_2000.log
-rw-r--r-- 1 oracle dba 0 Feb 23 11:35 fichas_2001C.log
    
```

\$ more fichas_2000.log

```

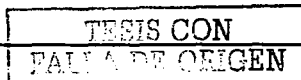
SQL*Loader: Release 8.1.6.0.0 - Production on
(c) Copyright 1999 Oracle Corporation. All rights reserved.
Control File: fichas.2000.ctl
Data File: /u39/oradata/DESC_MIGRA_PCI3/R123_fichas1.unl
Bad File: R123_fichas1.bad
Discard File: none specified
    
```

(Allow all discards)

```

Number to load: ALL
Number to skip: 0
Errors allowed: 50
Continuation: none specified
Path used: Direct - with parallel option.
    
```

Table FICHAS, loaded from every logical record.
 Insert option in effect for this table: APPEND



| Column Name | Position | Len | Term | Encl | Datatype |
|--------------------------|----------|-----|------|------|----------------------------|
| CARD_CODE | FIRST | 16 | | | CHARACTER |
| SEQUENCE_NUMBER | NEXT | * | | | CHARACTER |
| ACCESS_NUMBER | NEXT | * | | | CHARACTER |
| RETAILER_ID | NEXT | * | | | CHARACTER |
| ACCESS_CODE | NEXT | 20 | | | CHARACTER |
| AMOUNT_EXPIRATION_PERIOD | NEXT | * | | | CHARACTER |
| CARD_EXPIRATION_DATE | NEXT | 19 | | | DATE yyyy-mm-dd hh24:mi:ss |
| CARD_AMOUNT | NEXT | * | | | CHARACTER |
| STATUS | NEXT | * | | | CHARACTER |
| USED_DATE | NEXT | 19 | | | DATE yyyy-mm-dd hh24:mi:ss |
| DISTRIBUTE_DATE | NEXT | 19 | | | DATE yyyy-mm-dd hh24:mi:ss |
| LAST_STATUS_DATE | NEXT | 19 | | | DATE yyyy-mm-dd hh24:mi:ss |
| TERMINATION_REASON_CODE | NEXT | * | | | CHARACTER |
| CREATION_DATE | NEXT | 19 | | | DATE yyyy-mm-dd hh24:mi:ss |
| UPDATED_WHEN | NEXT | 19 | | | DATE yyyy-mm-dd hh24:mi:ss |
| UPDATED_BY | NEXT | 8 | | | CHARACTER |
| FK_BATCH_NUMBER | NEXT | * | | | CHARACTER |
| FK_COMPANY_NUMBER | NEXT | * | | | CHARACTER |
| OP_UNIT_ID | NEXT | * | | | CHARACTER |

Table FICHAS:

1152592 Rows successfully loaded.
 0 Rows not loaded due to data errors.
 0 Rows not loaded because all WHEN clauses were failed.
 0 Rows not loaded because all fields were null.

Bind array size not used in direct path.
 Space allocated for memory besides bind array: 0 bytes

Total logical records skipped: 0
 Total logical records read: 1152592
 Total logical records rejected: 0
 Total logical records discarded: 0

Run began on
 Run ended on

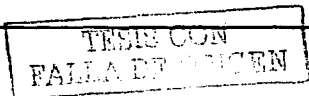
Elapsed time was:
 CPU time was:

IV.5.3.6

\$ more add_indexes.sql
 SPOOL add_indexes.log

```
set timing on
SET HEADING OFF
SET ECHO OFF
alter session set sort_area_size=52428800;
PROMPT Begin of the process
SELECT TO_CHAR(SYSDATE, 'YYYY/MM/DD HH24:MI:SS')
FROM SYS.DUAL;
```

SET HEADING ON



```

SET ECHO ON
ALTER TABLE FICHAS
ADD CONSTRAINT PHOCDINO_T
PRIMARY KEY
(CARD_CODE)
USING INDEX
TABLESPACE IINDEX04 PCTFREE 30
STORAGE(INITIAL 50M NEXT 50M PCTINCREASE 0 );

```

```

SET HEADING OFF
SET ECHO OFF
PROMPT Checkpoint 0

```

```

SELECT TO_CHAR(SYSDATE, 'YYYY/MM/DD HH24:MI:SS')
FROM SYS.DUAL;

```

```

ALTER TABLE FICHAS ADD CONSTRAINT PHOCDINV
FOREIGN KEY (FK_COMPANY_NUMBER, FK_BATCH_NUMBER)
REFERENCES FICHAS_BATCH (COMPANY_NUMBER, BATCH_NUMBER)
ON DELETE CASCADE;

```

```

SET HEADING OFF
SET ECHO OFF
PROMPT Checkpoint 1
SELECT TO_CHAR(SYSDATE, 'YYYY/MM/DD HH24:MI:SS')
FROM SYS.DUAL;

```

```

CREATE UNIQUE INDEX PHOCDIN1_T ON
FICHAS(Access_CODE)
TABLESPACE IINDEX03 PCTFREE 30
STORAGE(INITIAL 50M NEXT 50M PCTINCREASE 0 )
PARALLEL (DEGREE 16);

```

```

SET HEADING OFF
SET ECHO OFF
PROMPT Checkpoint 2

```

```

SELECT TO_CHAR(SYSDATE, 'YYYY/MM/DD HH24:MI:SS')
FROM SYS.DUAL;

```

```

CREATE INDEX PHOCDIN3_T ON
FICHAS(SEQUENCE_NUMBER)
TABLESPACE IINDEX04 PCTFREE 30
STORAGE(INITIAL 50M NEXT 50M PCTINCREASE 0 )
PARALLEL (DEGREE 20)
;

```

```

SET HEADING OFF
SET ECHO OFF
PROMPT Checkpoint 3
SELECT TO_CHAR(SYSDATE, 'YYYY/MM/DD HH24:MI:SS')
FROM SYS.DUAL;

```

TESIS CON
FALL 2014

```
CREATE INDEX PHOCCIN4_T ON
  FICHAS(USED_DATE)
  TABLESPACE INDEX03 PCTFREE 30
  STORAGE(INITIAL 50M NEXT 50M PCTINCREASE 0 )
  PARALLEL (DEGREE 16);
```

```
SET HEADING OFF
SET ECHO OFF
PROMPT End of the process
SELECT TO_CHAR(SYSDATE, 'YYYY/MM/DD HH24:MI:SS')
FROM SYS.DUAL;
```

SPOOL OFF

EXIT

IV.5.4.1

DROP TABLE DELTA_ADJUSTMENT CASCADE CONSTRAINTS;

```
CREATE TABLE DELTA_ADJUSTMENT (
  NO                NUMBER(6)   NOT NULL,
  ACCESS_NUMBER     NUMBER(15)  NOT NULL,
  TRANSACTION_NUMBER NUMBER(6)   NOT NULL,
  BALANCE           NUMBER(13,4) NOT NULL,
  CALL_TOTAL        NUMBER(13,4) NOT NULL,
  FINAL_BALANCE     NUMBER(13,4) NOT NULL,
  UPDATED_WHEN      DATE         NOT NULL,
```

CONSTRAINT NO_DELAD

PRIMARY KEY (NO)

USING INDEX

TABLESPACE INDEX16 PCTFREE 30

STORAGE (INITIAL 5M NEXT 2M PCTINCREASE 0))

TABLESPACE IDATA16

PCTFREE 10

PCTUSED 40

INITRANS 1

MAXTRANS 255

STORAGE (

INITIAL 50M

NEXT 10M

MINEXTENTS 1

MAXEXTENTS 2147483645

FREELISTS 1 FREELIST GROUPS 1)

NOCACHE;

--Realiza el ajuste de cobros de llamadas
 --efectuadas durante el periodo de migración
 --Oracle 7.3.4 a Oracle 8.1.6

```
SET Pagesize 0;
SET Linesize 200;
SET Verify On;
```



```
SET Feedback On;  
SET ServerOutput ON Size 1000000;
```

```
DECLARE
```

```
v_fecha      VARCHAR2(22);  
v_accessnum  NUMBER(15);  
v_transnum   NUMBER(6);  
v_calltotal  NUMBER(13,4);  
v_balance    NUMBER(13,4);  
v_i          NUMBER(9) := 0;
```

```
CURSOR numbers IS
```

```
SELECT access_number, call_total  
FROM rated_call  
WHERE creation_date between to_date('20010505 17:00:00','YYYYMMDD HH24:MI:SS')  
and to_date('20010506 13:00:00','YYYYMMDD HH24:MI:SS')  
AND call_total <> 0  
ORDER BY 1;
```

```
BEGIN
```

```
SELECT to_char(sysdate,'YYYY/MM/DD HH24:MI:SS')  
INTO v_fecha  
FROM dual;  
DBMS_OUTPUT.PUT_LINE('Begin Call Adjustment: '||v_fecha);
```

```
OPEN numbers;
```

```
LOOP
```

```
FETCH numbers INTO v_accessnum, v_calltotal;  
EXIT WHEN numbers%NOTFOUND;
```

```
SELECT balance  
INTO v_balance  
FROM access_number  
WHERE access_number = v_accessnum;
```

```
IF v_balance <> 0 THEN
```

```
SELECT max(transaction_number)  
INTO v_transnum  
FROM adjustment_transaction  
WHERE access_number = v_accessnum;
```

```
IF v_transnum = 0 OR v_transnum IS NULL THEN
```

```
v_transnum := 1;
```

```
ELSE
```

```
v_transnum := v_transnum + 1;
```

```
END IF;
```

```
INSERT INTO ADJUSTMENT_TRANSACTION  
VALUES (v_accessnum,v_transnum,v_calltotal*(-  
1),'ADJPRE',SYSDATE,'3100605600060000','PRUEBA AJUSTE PERIODO DELTA','DELTA');  
COMMIT;
```

```
IF v_balance >= v_calltotal THEN
```

TESIS CON
FALLA DE ORIGEN

```

UPDATE ACCESS_NUMBER
SET BALANCE = BALANCE - v_calltotal, Updated_By = 'delta', Updated_When =
SYSDATE
WHERE Access_Number = v_accessnum;
COMMIT;
ELSE
UPDATE ACCESS_NUMBER
SET BALANCE = 0, Updated_By = 'delta', Updated_When = SYSDATE
WHERE Access_Number = v_accessnum;
COMMIT;
END IF;

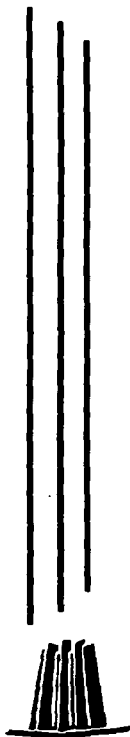
INSERT INTO DELTA_ADJUSTMENT
VALUES (v_i,v_accessnum,v_transnum,v_balance,v_calltotal,(v_balance -
v_calltotal),sysdate);
COMMIT;
ELSE
INSERT INTO DELTA_ADJUSTMENT
VALUES (v_i,v_accessnum,-1,v_balance,0,0,sysdate);
COMMIT;
END IF;

v_i := v_i + 1;
END LOOP;
CLOSE numbers;

DBMS_OUTPUT.PUT_LINE('Total Adjustment Calls: '||v_i);
SELECT to_char(sysdate,'YYYY/MM/DD HH24:MI:SS')
INTO v_fecha
FROM dual;
DBMS_OUTPUT.PUT_LINE('End Call Adjustment: '||v_fecha);
END;
/
EXIT

```

TESIS CON
FALLA DE ORIGEN



APENDICE II
CAPITULO V

TESIS CON
FALLA DE ORIGEN

1951
FALLA DE ORIGEN

240

APÉNDICE II

3.1.1

Alert.log

```

Tue Apr 9 23:25:19 2002
Starting ORACLE instance (normal)
Tue Apr 9 23:25:33 2002
LICENSE_MAX_SESSION = 0
LICENSE_SESSIONS_WARNING = 0
LICENSE_MAX_USERS = 0
Starting up ORACLE RDBMS Version: 8.1.6.3.0.
System parameters with non-default values:
  processes = 1000
  timed_statistics = TRUE
  shared_pool_size = 250000000
  shared_pool_reserved_size= 30965760
  enqueue_resources = 2500
  nls_language = MEXICAN SPANISH
  nls_territory = MEXICO
  disk_asynch_io = FALSE
  tape_asynch_io = FALSE
  control_files = /u03/oradata/ppay/ctrl1ppay.ctl, /u27/oradata/ppay/ctrl2ppay.ctl
  db_block_buffers = 215000
  db_block_size = 8192
  db_block_lru_latches = 88
  db_writer_processes = 10
  compatible = 8.1.0.0.0
  log_archive_start = TRUE
  log_archive_dest = /archive/oradata/archive
  log_buffer = 10240000
  log_checkpoint_interval = 327689
  log_checkpoint_timeout = 600
  db_files = 512
  db_file_multiblock_read_count= 64
  log_checkpoints_to_alert = TRUE
  dml_locks = 1000
  rollback_segments = R01, R02, R04, R05, R06, R07, R08, R09, R10, R11, R12, R13, R14,
R15,
R16, R17, R18, R19, R20, R21, R22, R23, R24, R25, R26, R27, R28, R29, R30, R31, R32, R33,
R34, R35,
R36, R37, R38, R39, R40, R41, R42, R43, R44, R45
  max_enabled_roles = 50
  db_domain = world
  audit_trail = DB
  sort_area_size = 5120000
  db_name = ppay
  ifile = /oracle/ora8/dbs/configppay.ora
  optimizer_mode = choose
  utl_file_dir = /prepay/ppptools, /prepay/pptools/call_totals_npa, /prepay/pptools/tmp_
npa
  parallel_min_servers = 8
  parallel_max_servers = 50
  hash_multiblock_io_count = 1

```

TESIS CON
FALLA DE ORIGEN



```
background_dump_dest = /oracle/admin/ppay/bdump
user_dump_dest        = /u08/oradata/dump
max_dump_file_size    = 10240
core_dump_dest        = /oracle/admin/ppay/cdump
PMON started with pid=2
DBW0 started with pid=3
DBW1 started with pid=4
DBW2 started with pid=5
DBW3 started with pid=6
DBW4 started with pid=7
DBW5 started with pid=8
DBW6 started with pid=9
DBW7 started with pid=10
DBW8 started with pid=11
DBW9 started with pid=12
LGWR started with pid=13
CKPT started with pid=14
SMON started with pid=15
RECO started with pid=16
Tue Apr 9 23:25:36 2002
ARCH: STARTING ARCH PROCESSES
ARC0 started with pid=17
Tue Apr 9 23:25:36 2002
ARCH: STARTING ARCH PROCESSES COMPLETE
Tue Apr 9 23:25:36 2002
ARC0: Archival started
Tue Apr 9 23:25:37 2002
alter database mount
Tue Apr 9 23:25:42 2002
Successful mount of redo thread 1, with mount id 3552168502.
Tue Apr 9 23:25:42 2002
Database mounted in Exclusive Mode.
Completed: alter database mount
Tue Apr 9 23:25:42 2002
alter database open
Tue Apr 9 23:25:52 2002
Thread 1 opened at log sequence 7907
  Current log# 2 seq# 7907 mem# 0: /u35/oradata/ppay/log2ppay.dbf
  Current log# 2 seq# 7907 mem# 1: /u36/oradata/ppay/log2ppay.dbf
Successful open of redo thread 1.
Tue Apr 9 23:25:52 2002
SMON: enabling cache recovery
SMON: enabling tx recovery
Tue Apr 9 23:25:55 2002
Updating 8.1.6.3.0 NLS parameters in sys.props$
-- adding 8.1.6.3.0 NLS parameters.
Completed: alter database open

Thu Apr 18 22:30:45 2002
ARC0: Beginning to archive log# 5 seq# 8264
Thu Apr 18 22:30:45 2002
alter tablespace DBTEST01 begin backup
Thu Apr 18 22:30:47 2002
Completed: alter tablespace DBTEST01 begin backup
Thu Apr 18 22:30:47 2002
```



alter tablespace ICONV01 begin backup
Completed: alter tablespace ICONV01 begin backup
Thu Apr 18 22:30:48 2002
alter tablespace IDATA01 begin backup
Completed: alter tablespace IDATA01 begin backup
Thu Apr 18 22:30:52 2002
alter tablespace IDATA02 begin backup
Completed: alter tablespace IDATA02 begin backup
Thu Apr 18 22:30:53 2002
alter tablespace IDATA03 begin backup
Thu Apr 18 22:30:54 2002
ARC0: Completed archiving log# 5 seq# 8264
Thu Apr 18 22:30:55 2002
Completed: alter tablespace IDATA03 begin backup
Thu Apr 18 22:30:55 2002
alter tablespace IDATA04 begin backup
Completed: alter tablespace IDATA04 begin backup
Thu Apr 18 22:31:05 2002
alter tablespace IDATA05 begin backup
Thu Apr 18 22:35:14 2002
Completed: alter tablespace IDATA05 begin backup
Thu Apr 18 22:35:14 2002
alter tablespace IDATA06 begin backup
Completed: alter tablespace IDATA06 begin backup
Thu Apr 18 22:35:19 2002
alter tablespace IDATA07 begin backup
Completed: alter tablespace IDATA07 begin backup
Thu Apr 18 22:35:23 2002
alter tablespace IDATA08 begin backup
Thu Apr 18 22:35:34 2002
Completed: alter tablespace IDATA08 begin backup
Thu Apr 18 22:35:34 2002
alter tablespace IDATA09 begin backup
Completed: alter tablespace IDATA09 begin backup
Thu Apr 18 22:35:41 2002
alter tablespace IDATA10 begin backup
Completed: alter tablespace IDATA10 begin backup
Thu Apr 18 22:35:42 2002
alter tablespace IDATA11 begin backup
Completed: alter tablespace IDATA11 begin backup
Thu Apr 18 22:35:42 2002
alter tablespace IDATA12 begin backup
Completed: alter tablespace IDATA12 begin backup
Thu Apr 18 22:35:43 2002
alter tablespace IDATA13 begin backup
Completed: alter tablespace IDATA13 begin backup
Thu Apr 18 22:35:44 2002
alter tablespace IDATA14 begin backup
Completed: alter tablespace IDATA14 begin backup
Thu Apr 18 22:35:44 2002
alter tablespace IDATA15 begin backup
Completed: alter tablespace IDATA15 begin backup
Thu Apr 18 22:35:46 2002
alter tablespace IDATA16 begin backup
Thu Apr 18 22:35:59 2002

RECIBO CON
FALLA DE ORIGEN



Completed: alter tablespace IDATA16 begin backup
Thu Apr 18 22:35:59 2002
alter tablespace IDATA18 begin backup
Completed: alter tablespace IDATA18 begin backup
Thu Apr 18 22:36:02 2002
alter tablespace IDATA19 begin backup
Completed: alter tablespace IDATA19 begin backup
Thu Apr 18 22:36:06 2002
alter tablespace IDATA20 begin backup
Thu Apr 18 22:36:17 2002
Completed: alter tablespace IDATA20 begin backup
Thu Apr 18 22:36:17 2002
alter tablespace IINDEX01 begin backup
Completed: alter tablespace IINDEX01 begin backup
Thu Apr 18 22:36:23 2002
alter tablespace IINDEX02 begin backup
Completed: alter tablespace IINDEX02 begin backup
Thu Apr 18 22:36:24 2002
alter tablespace IINDEX03 begin backup
Thu Apr 18 22:36:43 2002
Completed: alter tablespace IINDEX03 begin backup
Thu Apr 18 22:36:43 2002
alter tablespace IINDEX04 begin backup
Thu Apr 18 22:36:50 2002
Completed checkpoint up to RBA [0x2049.2.10], SCN: 0x0001.3986fdd1
Thu Apr 18 22:36:50 2002
Completed: alter tablespace IINDEX04 begin backup
Thu Apr 18 22:36:50 2002
alter tablespace IINDEX05 begin backup
Completed: alter tablespace IINDEX05 begin backup
Thu Apr 18 22:36:54 2002
alter tablespace IINDEX06 begin backup
Completed: alter tablespace IINDEX06 begin backup
Thu Apr 18 22:36:56 2002
alter tablespace IINDEX07 begin backup
Thu Apr 18 22:37:16 2002
Completed: alter tablespace IINDEX07 begin backup
Thu Apr 18 22:37:16 2002
alter tablespace IRPT01 begin backup
Completed: alter tablespace IRPT01 begin backup
Thu Apr 18 22:37:19 2002
alter tablespace PLAN_MIXTO begin backup
Completed: alter tablespace PLAN_MIXTO begin backup
Thu Apr 18 22:37:20 2002
alter tablespace RBS1 begin backup
Completed: alter tablespace RBS1 begin backup
Thu Apr 18 22:37:25 2002
alter tablespace RBS2 begin backup
Completed: alter tablespace RBS2 begin backup
Thu Apr 18 22:37:27 2002
alter tablespace RBS3 begin backup
Completed: alter tablespace RBS3 begin backup
Thu Apr 18 22:37:29 2002
alter tablespace RBS4 begin backup
Completed: alter tablespace RBS4 begin backup

DEIS OUR
FALLA DE SERGEN



Thu Apr 18 22:37:32 2002
alter tablespace RBS5 begin backup
Thu Apr 18 22:37:44 2002
Completed: alter tablespace RBS5 begin backup
Thu Apr 18 22:37:44 2002
alter tablespace STATSPACK begin backup
Completed: alter tablespace STATSPACK begin backup
Thu Apr 18 22:37:45 2002
alter tablespace SYSTEM begin backup
Completed: alter tablespace SYSTEM begin backup
Thu Apr 18 22:37:46 2002
alter tablespace TEMP_PPAY begin backup
Completed: alter tablespace TEMP_PPAY begin backup
Thu Apr 18 22:50:13 2002
Beginning log switch checkpoint up to RBA [0x204a.2.10], SCN: 0x0001.398925f0
Thread 1 advanced to log sequence 8266
Current log# 1 seq# 8266 mem# 0: /u35/oradata/ppay/log1ppay.dbf
Current log# 1 seq# 8266 mem# 1: /u38/oradata/ppay/log1ppay.dbf
Thu Apr 18 22:50:13 2002
ARC0: Beginning to archive log# 6 seq# 8265
Thu Apr 18 22:51:22 2002
ARC0: Completed archiving log# 6 seq# 8265
Thu Apr 18 22:58:48 2002
Completed checkpoint up to RBA [0x204a.2.10], SCN: 0x0001.398925f0
Thu Apr 18 23:22:09 2002
Beginning log switch checkpoint up to RBA [0x204b.2.10], SCN: 0x0001.398b8fae
Thread 1 advanced to log sequence 8267
Current log# 2 seq# 8267 mem# 0: /u35/oradata/ppay/log2ppay.dbf
Current log# 2 seq# 8267 mem# 1: /u38/oradata/ppay/log2ppay.dbf
Thu Apr 18 23:22:09 2002
ARC0: Beginning to archive log# 1 seq# 8266
Thu Apr 18 23:22:37 2002
ARC0: Completed archiving log# 1 seq# 8266
Thu Apr 18 23:32:42 2002
Completed checkpoint up to RBA [0x204b.2.10], SCN: 0x0001.398b8fae
Fri Apr 19 00:01:45 2002
alter tablespace DBTEST01 end backup
Fri Apr 19 00:01:45 2002
Completed: alter tablespace DBTEST01 end backup
Fri Apr 19 00:01:45 2002
alter tablespace ICONV01 end backup
Completed: alter tablespace ICONV01 end backup
Fri Apr 19 00:01:46 2002
alter tablespace IDATA01 end backup
Completed: alter tablespace IDATA01 end backup
Fri Apr 19 00:01:46 2002
alter tablespace IDATA02 end backup
Completed: alter tablespace IDATA02 end backup
Fri Apr 19 00:01:46 2002
alter tablespace IDATA03 end backup
Completed: alter tablespace IDATA03 end backup
Fri Apr 19 00:01:47 2002
alter tablespace IDATA04 end backup
Completed: alter tablespace IDATA04 end backup
Fri Apr 19 00:01:48 2002

TESIS CON
FALLA DE ORIGEN



alter tablespace IDATA05 end backup
Completed: alter tablespace IDATA05 end backup
Fri Apr 19 00:01:48 2002
alter tablespace IDATA06 end backup
Completed: alter tablespace IDATA06 end backup
Fri Apr 19 00:01:49 2002
alter tablespace IDATA07 end backup
Completed: alter tablespace IDATA07 end backup
Fri Apr 19 00:01:50 2002
alter tablespace IDATA08 end backup
Completed: alter tablespace IDATA08 end backup
Fri Apr 19 00:01:51 2002
alter tablespace IDATA09 end backup
Completed: alter tablespace IDATA09 end backup
Fri Apr 19 00:01:52 2002
alter tablespace IDATA10 end backup
Completed: alter tablespace IDATA10 end backup
Fri Apr 19 00:01:52 2002
alter tablespace IDATA11 end backup
Completed: alter tablespace IDATA11 end backup
Fri Apr 19 00:01:52 2002
alter tablespace IDATA12 end backup
Completed: alter tablespace IDATA12 end backup
Fri Apr 19 00:01:52 2002
alter tablespace IDATA13 end backup
Completed: alter tablespace IDATA13 end backup
Fri Apr 19 00:01:52 2002
alter tablespace IDATA14 end backup
Completed: alter tablespace IDATA14 end backup
Fri Apr 19 00:01:52 2002
alter tablespace IDATA15 end backup
Completed: alter tablespace IDATA15 end backup
Fri Apr 19 00:01:53 2002
alter tablespace IDATA16 end backup
Completed: alter tablespace IDATA16 end backup
Fri Apr 19 00:01:55 2002
alter tablespace IDATA18 end backup
Completed: alter tablespace IDATA18 end backup
Fri Apr 19 00:01:55 2002
alter tablespace IDATA19 end backup
Completed: alter tablespace IDATA19 end backup
Fri Apr 19 00:01:56 2002
alter tablespace IDATA20 end backup
Completed: alter tablespace IDATA20 end backup
Fri Apr 19 00:01:57 2002
alter tablespace IINDEX01 end backup
Completed: alter tablespace IINDEX01 end backup
Fri Apr 19 00:01:58 2002
alter tablespace IINDEX02 end backup
Completed: alter tablespace IINDEX02 end backup
Fri Apr 19 00:01:58 2002
alter tablespace IINDEX03 end backup
Completed: alter tablespace IINDEX03 end backup
Fri Apr 19 00:02:01 2002
alter tablespace IINDEX04 end backup



Completed: alter tablespace IINDEX04 end backup
Fri Apr 19 00:02:02 2002
alter tablespace IINDEX05 end backup
Completed: alter tablespace IINDEX05 end backup
Fri Apr 19 00:02:02 2002
alter tablespace IINDEX06 end backup
Completed: alter tablespace IINDEX06 end backup
Fri Apr 19 00:02:03 2002
alter tablespace IINDEX07 end backup
Completed: alter tablespace IINDEX07 end backup
Fri Apr 19 00:02:03 2002
alter tablespace IRPT01 end backup
Completed: alter tablespace IRPT01 end backup
Fri Apr 19 00:02:04 2002
alter tablespace PLAN_MIXTO end backup
Completed: alter tablespace PLAN_MIXTO end backup
Fri Apr 19 00:02:04 2002
alter tablespace RBS1 end backup
Completed: alter tablespace RBS1 end backup
Fri Apr 19 00:02:04 2002
alter tablespace RBS2 end backup
Completed: alter tablespace RBS2 end backup
Fri Apr 19 00:02:04 2002
alter tablespace RBS3 end backup
Completed: alter tablespace RBS3 end backup
Fri Apr 19 00:02:05 2002
alter tablespace RBS4 end backup
Completed: alter tablespace RBS4 end backup
Fri Apr 19 00:02:05 2002
alter tablespace RBS5 end backup
Completed: alter tablespace RBS5 end backup
Fri Apr 19 00:02:05 2002
alter tablespace STATSPACK end backup
Completed: alter tablespace STATSPACK end backup
Fri Apr 19 00:02:05 2002
alter tablespace SYSTEM end backup
Completed: alter tablespace SYSTEM end backup
Fri Apr 19 00:02:06 2002
alter tablespace TEMP_PPAY end backup
Completed: alter tablespace TEMP_PPAY end backup
Fri Apr 19 00:02:07 2002
alter database backup controlfile to '/hotbck1/ora_backup/ctrl1ppay.ctf'
Fri Apr 19 00:02:09 2002
Completed: alter database backup controlfile to '/hotbck1/ora
Fri Apr 19 00:02:09 2002
alter database backup controlfile to '/hotbck2/ora_backup/ctrl1ppay.ctf'
Completed: alter database backup controlfile to '/hotbck2/ora
Fri Apr 19 00:02:11 2002
alter database backup controlfile to '/hotbck3/ora_backup/ctrl1ppay.ctf'
Completed: alter database backup controlfile to '/hotbck3/ora
Fri Apr 19 00:02:13 2002
alter database backup controlfile to trace
Completed: alter database backup controlfile to trace
Fri Apr 19 00:02:14 2002
Beginning log switch checkpoint up to RBA [0x204c.2.10], SCN: 0x0001.398da5f5

TESIS CON
FALLA DE ORIGEN



Thread 1 advanced to log sequence 8268
Current log# 3 seq# 8268 mem# 0: /u35/oradata/ppay/log3ppay.dbf
Current log# 3 seq# 8268 mem# 1: /u38/oradata/ppay/log3ppay.dbf
Fri Apr 19 00:02:14 2002
ARC0: Beginning to archive log# 2 seq# 8267
Fri Apr 19 00:02:31 2002
ARC0: Completed archiving log# 2 seq# 8267
Fri Apr 19 00:09:41 2002
alter tablespace idata16 coalesce
Fri Apr 19 00:09:47 2002
Completed: alter tablespace idata16 coalesce
Fri Apr 19 00:10:03 2002
alter tablespace iindex03 coalesce
Fri Apr 19 00:10:06 2002
Completed: alter tablespace iindex03 coalesce
Fri Apr 19 00:12:42 2002
Completed checkpoint up to RBA [0x204c.2.10], SCN: 0x0001.398da5f5
Fri Apr 19 00:13:32 2002
alter tablespace idata16 coalesce
Fri Apr 19 00:13:32 2002
Completed: alter tablespace idata16 coalesce
Fri Apr 19 00:13:53 2002
alter tablespace iindex03 coalesce
Fri Apr 19 00:13:53 2002
Completed: alter tablespace iindex03 coalesce
Fri Apr 19 00:29:08 2002
Beginning log switch checkpoint up to RBA [0x204d.2.10], SCN: 0x0001.398ed5ab
Thread 1 advanced to log sequence 8269
Current log# 4 seq# 8269 mem# 0: /u35/oradata/ppay/log4ppay.dbf
Current log# 4 seq# 8269 mem# 1: /u38/oradata/ppay/log4ppay.dbf

Tue Apr 23 11:24:40 2002
ARC0: Beginning to archive log# 6 seq# 8493
Tue Apr 23 11:25:06 2002
ARC0: Completed archiving log# 6 seq# 8493
Tue Apr 23 11:34:00 2002
Completed checkpoint up to RBA [0x212e.2.10], SCN: 0x0001.3c25dc9d
Tue Apr 23 11:53:31 2002
Beginning log switch checkpoint up to RBA [0x212f.2.10], SCN: 0x0001.3c2a8aaf
Thread 1 advanced to log sequence 8495
Current log# 2 seq# 8495 mem# 0: /u35/oradata/ppay/log2ppay.dbf
Current log# 2 seq# 8495 mem# 1: /u38/oradata/ppay/log2ppay.dbf
Tue Apr 23 11:53:31 2002
ARC0: Beginning to archive log# 1 seq# 8494
Tue Apr 23 11:53:57 2002
ARC0: Completed archiving log# 1 seq# 8494
Tue Apr 23 12:02:39 2002
Completed checkpoint up to RBA [0x212f.2.10], SCN: 0x0001.3c2a8aaf

FAIL



V.3.2.1 Archivo init.ora

```
# SHeader: initx.ora 1.1 95/02/27 12:14:56 wyim Osd<unix> $ Copyr (c) 1992 Oracle
# include database configuration parameters
ifeil = /oracle/ora8/dbs/configpppay.ora
rollback_segments = (R01, R02, R04, R05, R06, R07, R08, R09, R10, \
R11, R12, R13, R14, R15, R16, R17, R18, R19, R20, \
R21, R22, R23, R24, R25, R26, R27, R28, R29, R30, \
R31, R32, R33, R34, R35, R36, R37, R38, R39, R40, \
R41, R42, R43, R44, R45)
db_domain = world
hash_multiblock_io_count = 1
db_files = 512
db_block_size = 8192
# db_writers = 1 # Tunning 21/08/2000
# db_writers = 10 # Tunning 27/08/2000
db_writer_processes = 10
# async_write = false # Tunning 27/08/2000
optimizer_mode = choose
db_file_multiblock_read_count = 64 # Tunning 21/08/2000
db_block_buffers = 215000 # Tuning 20/05/2001
db_block_lru_latches = 88 # Tuning 21/08/2000
shared_pool_size = 250000000 # Tunning 20/05/2001
shared_pool_reserved_size = 30965760 # Tunning 20/05/2001
# log_checkpoint_interval = 10000
# log_checkpoint_interval = 100000000
log_checkpoint_interval = 327689
# checkpoint_process = true
log_checkpoint_timeout = 600 # Cambio a 600 6/nov/2001 para agllizar
# recuperaciones en caso de failover
log_checkpoints_to_alert = true
# spin_count = 8000 # Tunning 21/08/2000
processes = 1000 # PPAS
dml_locks = 1000 # PPAS

# log_buffer = 8192 # SMALL
# log_buffer = 32768 # MEDIUM
# log_buffer = 163840 # LARGE
# log_buffer = 2097152
log_buffer = 10240000 # Tunning 21/08/2000
# log_simultaneous_copies = 4
# log_simultaneous_copies = 88 # Tunning 21/08/2000
enqueue_resources = 2500
# enqueue_resources = 800 # Tunning 21/08/2000
# sort_area_size = 3297152
sort_area_size = 5120000 # Tunning 21/08/2000
audit_trail = db # if you want auditing
timed_statistics = true # if you want timed statistics
max_dump_file_size = 10240 # limit trace file size to 5 Meg each
user_dump_dest = /u08/oradata/dump
log_archive_start = true # if you want automatic archiving
# log_archive_buffers = 1 # minimize impact on performance
# log_archive_buffer_size = 35 # max buffer size
log_archive_dest = /archive/oradata/archive
# compatible = 7.3.0.0.0
```



```
# compatible = 7.3.4.0.0          # Tuning 21/08/2000
compatible = 8.1.0.0.0          # Tuning 21/08/2000
# global_names = TRUE
NLS_LANGUAGE = "MEXICAN SPANISH"
NLS_TERRITORY = MEXICO
max_enabled_roles = 50
utl_file_dir = /prepay/pptools
utl_file_dir = /prepay/pptools/call_totals_npa
utl_file_dir = /prepay/pptools/tmp_npa
#remote_login_passwordfile=exclusive

disk_asynch_io = FALSE
tape_asynch_io = FALSE
parallel_max_servers=50
parallel_min_servers=8
```

V.3.3.1 QUERYS

Los valores están dados en centésimas de segundo:

```
SQL> SELECT event, total_waits, total_timeouts, time_waited, average_wait
FROM v$system_event;
```

A nivel sesión:

```
SQL> SELECT * FROM v$session_event WHERE sid=10;
```

Eventos en espera:

```
SQL> SELECT sid, seq#, event, wait_time, state FROM v$session_wait;
```

QUERYS

Los siguientes queries muestran información de monitoreo importantes para cualquier base de datos los cuales se han logrado en base a la experiencia y no se encuentran disponibles ni en libros ni manuales.

Session.sql

```
alter session set nls_date_format = 'dd/mm/yyyy hh24:mi:ss';
set echo off
set heading on
set feedback on
set verify off
set termout on
set pagesize 50000
set linesize 300
set wrap off
set trimout on
set trimspool on
set serveroutput off
col sid for 99999
col serial# for 99999
col username for a8
col osuser for a8
col spid for 99999
col machine for a12
col logon_time for a16
```

col status for a8

--SPOOL \$HOME/reports/session

```
SELECT vs.sid SID, vs.serial# SERIAL#, vs.username USERNAME,
       vs.osuser OSUSER, vp.spid SPID, vs.machine MACHINE,
       logon_time LOGON_TIME,
       vs.status STATUS
FROM   v$session vs, v$process vp
WHERE  vs.osuser <> 'pp'
AND    vs.type = 'USER'
AND    vs.paddr = vp.addr
ORDER BY STATUS, LOGON_TIME
/
```

--SPOOL OFF

```
sql_activos.sql
ALTER SESSION SET NLS_DATE_FORMAT = 'DD/MM/YYYY HH24:MI:SS';
SET ECHO OFF
SET HEADING ON
SET FEEDBACK ON
SET VERIFY OFF
SET TERMOUT ON
SET PAGESIZE 50000
SET LINESIZE 300
SET WRAP OFF
SET TRIMOUT ON
SET TRIMSPOOL ON
SET SERVEROUTPUT OFF
COL SID FOR 99999
COL SERIAL# FOR 99999
COL USERNAME FOR A8
COL OSUSER FOR A8
COL SPID FOR 99999
COL MACHINE FOR A12
COL LOGON_TIME FOR A16
COL STATUS FOR A8
```

--SPOOL \$HOME/reports/sql_activos.unl

```
select a.sid,
       a.serial#,
       -- d.spid, a.username,a.osuser, a.machine,a.terminal,a.logon_time,
       b.sql_text
from   v$session a, v$sqltext b, v$process d
       ..here
       a.sql_address = b.address
AND    a.paddr = d.addr
AND    a.sql_hash_value = b.hash_value
AND    a.osuser <> 'pp'
AND    a.type = 'USER'
AND    a.status = 'ACTIVE'
AND    a.username <> 'SYSTEM'
order by a.sid, a.serial#, b.piece
/
```

--SPOOL OFF

TESIS CON
FALLA DE ORIGEN

**sql_user.sql**

```
DEFINE usersid=&1;
DEFINE userserial=&2;

SELECT a.sid SID, a.serial# SERIAL#, a.username USERNAME,
       b.sql_text
FROM v$session a, v$sqltext b
WHERE a.sid = &usersid
AND a.serial# = &userserial
AND a.sql_address = b.address
AND a.sql_hash_value = b.hash_value
ORDER BY a.sid, a.username, b.piece;
```

V.5.1 COMANDOS DE UNIX PARA EL MONITOREO.

En esta parte se muestran algunos comandos de Unix para el monitoreo de las partes más importantes de sistema operativo.

```
$ sar 1 20
```

```
SunOS domain5 5.8 Generic_111850-02 sun4u 06/26/02
```

| | %usr | %sys | %wio | %idle |
|----------|------|------|------|-------|
| 07:44:12 | | | | |
| 07:44:14 | 81 | 19 | 0 | 0 |
| 07:44:18 | 79 | 21 | 0 | 0 |
| 07:44:19 | 79 | 21 | 0 | 0 |
| 07:44:20 | 80 | 20 | 0 | 0 |
| 07:44:21 | 82 | 18 | 0 | 0 |
| 07:44:22 | 82 | 18 | 0 | 0 |
| 07:44:24 | 81 | 19 | 0 | 0 |
| 07:44:25 | 83 | 17 | 0 | 0 |
| 07:44:26 | 81 | 19 | 0 | 0 |
| 07:44:27 | 82 | 18 | 0 | 0 |
| 07:44:28 | 81 | 19 | 0 | 0 |
| 07:44:29 | 81 | 19 | 0 | 0 |
| 07:44:30 | 80 | 20 | 0 | 0 |
| 07:44:31 | 81 | 19 | 0 | 0 |
| 07:44:32 | 81 | 19 | 0 | 0 |
| 07:44:33 | 83 | 17 | 0 | 0 |
| 07:44:34 | 81 | 19 | 0 | 0 |
| 07:44:36 | 81 | 19 | 0 | 0 |
| 07:44:37 | 82 | 18 | 0 | 0 |
| 07:44:38 | 80 | 20 | 0 | 0 |
| Average | 81 | 19 | 0 | 0 |

```
$ ps -ef | grep if | grep of | grep -v grep | more
```

```
oracle 19890      1  0 07:38:17 ?        0:04 dd if=/u43/oradata/ppay/ldata1616ppay.dbf
of/hotbck3/ora_backup/346.pipe bs=10
```



```

oracle 19807      1  0 07:38:17 ?      0:04 dd if=/u30/oradata/ppay/trbs026ppay.dbf
of=/hotbck1/ora_backup/289.pipe bs=1024
oracle 20010     1  0 07:38:18 ?      0:03 dd if=/logspayB/oradata/ppay/log10ppay.dbf
of=/hotbck3/ora_backup/10Bbf.pipe b
oracle 19862     1  0 07:38:17 ?      0:04 dd if=/u43/oradata/ppay/idata1613ppay.dbf
of=/hotbck2/ora_backup/333.pipe bs=10
oracle 19920     1  0 07:38:18 ?      0:04 dd if=/pagos1/oradata/ppay/idata201ppay.dbf
of=/hotbck1/ora_backup/378.pipe bs=
oracle 19941     1  0 07:38:18 ?      0:05 dd if=/pagos2/oradata/ppay/idata208ppay.dbf
of=/hotbck4/ora_backup/385.pipe bs=
oracle 19892     1  0 07:38:17 ?      0:04 dd if=/u50/oradata/ppay/iindex079ppay.dbf
of=/hotbck4/ora_backup/347.pipe bs=10
oracle 19995     1  0 07:38:18 ?      0:04 dd if=/logspayA/oradata/ppay/log10ppay.dbf
of=/hotbck3/ora_backup/10Abf.pipe b
oracle 19880     1  0 07:38:17 ?      0:04 dd if=/u43/oradata/ppay/idata1615ppay.dbf
of=/hotbck4/ora_backup/341.pipe bs=10
oracle 19784     1  0 07:38:17 ?      0:06 dd if=/u29/oradata/ppay/idata1742ppay.dbf
of=/hotbck3/ora_backup/265.pipe bs=10
oracle 19827     1  0 07:38:17 ?      0:04 dd if=/u37/oradata/ppay/idata166ppay.dbf
of=/hotbck3/ora_backup/310.pipe bs=102
oracle 19792     1  0 07:38:17 ?      0:04 dd if=/u03/oradata/ppay/trbs045ppay.dbf
of=/hotbck4/ora_backup/282.pipe bs=1024
oracle 19865     1  0 07:38:17 ?      0:04 dd if=/u50/oradata/ppay/iindex076ppay.dbf
of=/hotbck3/ora_backup/334.pipe bs=10
oracle 19868     1  0 07:38:17 ?      0:04 dd if=/u45/oradata/ppay/idata1617ppay.dbf
of=/hotbck4/ora_backup/335.pipe bs=10
oracle 19843     1  0 07:38:17 ?      0:04 dd if=/u38/oradata/ppay/idata1611ppay.dbf
of=/hotbck4/ora_backup/321.pipe bs=10
oracle 19767     1  0 07:38:17 ?      0:05 dd if=/u35/oradata/ppay/idata163ppay.dbf
of=/hotbck4/ora_backup/266.pipe bs=102
oracle 19762     1  0 07:38:17 ?      0:05 dd if=/u35/oradata/ppay/idata162ppay.dbf
of=/hotbck2/ora_backup/264.pipe bs=102
oracle 19973     1  0 07:38:18 ?      0:04 dd if=/pagos4/oradata/ppay/iindex067ppay.dbf
of=/hotbck3/ora_backup/396.pipe bs
oracle 19804     1  0 07:38:17 ?      0:04 dd if=/u30/oradata/ppay/trbs025ppay.dbf
of=/hotbck4/ora_backup/288.pipe bs=1024
oracle 19786     1  0 07:38:17 ?      0:04 dd if=/u35/oradata/ppay/idata164ppay.dbf
of=/hotbck2/ora_backup/280.pipe bs=102
oracle 20019     1  0 07:38:18 ?      0:03 dd if=/logspayB/oradata/ppay/log12ppay.dbf
of=/hotbck1/ora_backup/12Bbf.pipe b
oracle 19813     1  0 07:38:17 ?      0:04 dd if=/u31/oradata/ppay/iindex0336ppay.dbf
of=/hotbck3/ora_backup/294.pipe bs=1
oracle 20007     1  0 07:38:18 ?      0:03 dd if=/logspayB/oradata/ppay/log8ppay.dbf
of=/hotbck1/ora_backup/8Bbf.pipe bs=

```

\$ /usr/ucb/ps -aguxww | head -30

```

USER      PID %CPU %MEM    SZ  RSS TT      S  START TIME COMMAND
root      26601  3.1  0.0 1264  952 pts/7    O  07:37:38 19:23 /usr/bin/gzip
/ufsdump/actual/opt.dump          /ufsdump/actual/root.dump          /ufsdump/actual/usr.dump
/ufsdump/actual/var.dump
oracle   27966  2.2 7.220702722057376 ?      O  06:30:19 53:36 oracleppay
(DESCRIPTION=(LOCAL=YES)(ADDRESS=(PROTOCOL=beq)))

```

```

oracle 18485 1.9 7.120556562040648 ? O 01:00:03 83:21 oracleppay
(DESCRIPTION=(LOCAL=YES)(ADDRESS=(PROTOCOL=beq)))
oracle 8760 1.6 7.120554082042040 ? O 02:05:00 143:46 oracleppay
(DESCRIPTION=(LOCAL=YES)(ADDRESS=(PROTOCOL=beq)))
pp 18330 1.0 0.15436021520 ? S 01:00:03 230:53 PPBR7200 ppbr7200
repdw 22011 0.9 0.0 928 704 ? O 07:57:19 0:06 grep ORA-
/rpl/DWH/dw_inventario_20020625.unl
root 22021 0.7 0.0 2984 2880 pts/8 O 07:57:25 0:05 /usr/ucb/ps -aguxww
pp 8036 0.6 2.4675968673904 ? S Jun 25 505:44 preRater ppsr7100
pp 9641 0.6 2.4676328674264 ? S Jun 25 504:42 preRater ppsr7100
root 3 0.3 0.0 0 0 ? S Apr 06 11956:33 fflush
oracle 29955 0.3 7.120629682048416 ? S 03:15:45 52:55 oracleppay
(DESCRIPTION=(LOCAL=YES)(ADDRESS=(PROTOCOL=beq)))
oracle 29469 0.3 7.120713282057152 ? S 03:15:19 54:33 oracleppay
(DESCRIPTION=(LOCAL=YES)(ADDRESS=(PROTOCOL=beq)))
pp 7780 0.3 3.1890936884904 ? S Jun 25 349:36 sharedDataServer
root 29583 0.2 0.1 7384 6720 ? S 07:16:05 2:06 bpbkar -r 2147483647 -o
-cint domain4 -class P4 exp -sched P4_exp_schf -st FULL -bpstart_to 300 -bpend_to 300 -
read_to 300 -use_optm -b domain4_1025093760 -kl 200 -fso
pp 13242 0.2 0.1 6696 6008 ? S Jun 25 232:10 InterfaceServer
pp 12894 0.2 0.1 5224 4536 ? S Jun 25 152:12 ss7TcapParser
pp 13200 0.2 0.1 6696 6008 ? S Jun 25 231:15 InterfaceServer
    
```

\$ swap -s

total: 15137768k bytes allocated + 478176k reserved = 15615944k used, 15040680k available

\$ vmstat

```

procs memory page disk faults cpu
r b w swap free re mf pi po fr de sr m0 m1 m3 m4 in sy cs us sy ld
5 4 0 13956648 12100312 0 0 0 0 0 0 0 0 1 0 3 0 4294967196 0 0 -49 -10 -67
    
```

\$ iostat

```

tty md0 md1 md3 md4 cpu
'in tout kps tps serv kps tps serv kps tps serv kps tps serv us sy wt id
1 155 4 1 23 0 0 23 38 3 18 0 0 39 39 8 10 43
    
```

\$ uptime

8:06am up 80 day(s), 12:12, 7 users, load average: 4.96, 5.97, 7.11

\$ ps -ef | grep sqplus | grep -v sqplus

```

reportes 8555 11311 0 07:28:24 ? 0:00 sqplus /NOLOG
@/users/reportes/scripts/PL_SQL/report_escalera.sql
repdw 11656 10447 0 07:30:27 ? 0:00 sqplus /NOLOG
@/users/repdw/scripts/DW/report_dw.sql
    
```

V.7.1

QUERYS

Tablespace.sql

ALTER SESSION SET NLS_NUMERIC_CHARACTERS=',';



```

set term on
set echo off
set pagesize 80
set linesize 80
col name head "TSPACE" for a20
col Kbytes for 99,999,999
col USED for 99,999,999
col free for 99,999,999
col Pct_used for 99.99
title center 'TABLESPACE STATISTICS'

```

```

select nvl(FULL.tablespace_name,nvl(FREE.tablespace_name,'UNKNOWN')) name
       ,kbytes_used      Kbytes
       ,kbytes_used - nvl(kbytes_free,0) Used
       ,nvl(kbytes_free,0) free
       ,((kbytes_used - nvl(kbytes_free,0))/
         kbytes_used)*100 Pct_used
       ,nvl(max_free,0) Max_free
FROM (select sum(bytes)/1024 Kbytes_free
      ,max(bytes)/1024 max_free
      ,tablespace_name
      from sys.DBA_FREE_SPACE
      group by tablespace_name) FREE,
     ( select sum(bytes)/1024 Kbytes_used
      ,tablespace_name
      from sys.DBA_DATA_FILES
      group by tablespace_name) FULL
where FREE.tablespace_name(+) = FULL.tablespace_name
/

```

| TSPACE | TABLESPACE STATISTICS | | | | | |
|----------|-----------------------|------------|-----------|----------|----------|--|
| | KBYTES | USED | FREE | PCT_USED | MAX_FREE | |
| DBTEST01 | 102,400 | 1,240 | 101,160 | 1.21 | 101160 | |
| ICONV01 | 3,072,000 | 539,780 | 2,532,240 | 17.57 | 1508248 | |
| IDATA01 | 6,144,000 | 5,047,040 | 1,096,960 | 82.15 | 508992 | |
| IDATA02 | 1,536,000 | 322,696 | 1,213,304 | 21.01 | 971888 | |
| IDATA03 | 5,120,000 | 3,314,384 | 1,805,616 | 64.73 | 893512 | |
| IDATA04 | 9,062,400 | 7,326,752 | 1,735,648 | 80.85 | 963992 | |
| IDATA05 | 2,560,000 | 1,882,856 | 677,144 | 73.55 | 460792 | |
| IDATA06 | 5,120,000 | 4,400,336 | 719,664 | 85.94 | 378872 | |
| IDATA07 | 2,611,200 | 2,225,600 | 385,600 | 85.23 | 245752 | |
| IDATA08 | 8,448,000 | 7,162,016 | 1,285,984 | 84.78 | 491512 | |
| IDATA09 | 15,257,600 | 9,535,504 | 5,722,096 | 62.50 | 1314792 | |
| IDATA10 | 1,536,000 | 102,416 | 1,433,584 | 6.67 | 1023992 | |
| IDATA11 | 1,536,000 | 384,288 | 1,151,712 | 25.02 | 1151712 | |
| IDATA12 | 2,048,000 | 347,008 | 1,700,992 | 16.94 | 1353992 | |
| IDATA13 | 2,048,000 | 61,448 | 1,986,552 | 3.00 | 1986552 | |
| IDATA14 | 512,000 | 1,008 | 510,992 | .20 | 510992 | |
| IDATA15 | 2,048,000 | 1,618,184 | 429,816 | 79.01 | 235512 | |
| IDATA16 | 49,152,000 | 42,485,664 | 6,666,336 | 86.44 | 1843192 | |
| IDATA17 | 19,609,600 | 16,957,592 | 2,652,008 | 86.48 | 849912 | |
| IDATA18 | 3,072,000 | 121,400 | 2,950,600 | 3.95 | 398752 | |
| IDATA19 | 2,048,000 | 347,456 | 1,700,544 | 16.97 | 425672 | |
| IINDEX01 | 9,318,400 | 7,207,304 | 2,111,096 | 77.34 | 375504 | |



| | | | | | |
|----------|------------|------------|-----------|-------|---------|
| IINDEX02 | 4,096,000 | 2,313,768 | 1,782,232 | 56.49 | 918464 |
| IINDEX03 | 20,480,000 | 17,384,728 | 3,095,272 | 84.89 | 934312 |
| IINDEX04 | 8,806,400 | 6,275,088 | 2,531,312 | 71.26 | 552952 |
| IINDEX05 | 11,161,600 | 3,246,144 | 7,915,456 | 29.08 | 511992 |
| IINDEX06 | 1,024,000 | 139,304 | 884,696 | 13.60 | 221208 |
| IRPT01 | 4,300,800 | 3,700,472 | 600,328 | 86.04 | 93184 |
| ITRAIN01 | 10,240,000 | 3,117,296 | 7,122,704 | 30.44 | 2047992 |
| RBS1 | 5,120,000 | 2,662,424 | 2,457,576 | 52.00 | 1208312 |
| RBS2 | 5,120,000 | 675,864 | 4,444,136 | 13.20 | 1617912 |
| RBS3 | 5,120,000 | 880,664 | 4,239,336 | 17.20 | 1617920 |
| RBS4 | 5,120,000 | 675,864 | 4,444,136 | 13.20 | 1843192 |
| SYSTEM | 2,072,576 | 91,344 | 1,981,232 | 4.41 | 420080 |
| TMP_TEMP | 9,216,000 | 40 | 9,215,960 | .00 | 102400 |
| TMP_APP | 4,096,000 | 16 | 4,095,984 | .00 | 2047992 |

**V.7.2
QUERYS**

Log_status.sql

```

SPOOL SHOME/reports/log_status
SELECT group#, sequence#, bytes, archived,
       status, first_time
FROM   v$log
/
Spool off

```

Select * from V\$LOGFILE;

GROUP# STATUS MEMBER

```

7      /logsppayA/oradata/ppay/log7ppay.dbf
12     /logsppayA/oradata/ppay/log12ppay.dbf
7      /logsppayB/oradata/ppay/log7ppay.dbf
8      /logsppayA/oradata/ppay/log8ppay.dbf
10     /logsppayA/oradata/ppay/log10ppay.dbf
8      /logsppayB/oradata/ppay/log8ppay.dbf
10     /logsppayB/oradata/ppay/log10ppay.dbf
9      /logsppayA/oradata/ppay/log9ppay.dbf
9      /logsppayB/oradata/ppay/log9ppay.dbf
11     /logsppayA/oradata/ppay/log11ppay.dbf
11     /logsppayB/oradata/ppay/log11ppay.dbf
12     /logsppayB/oradata/ppay/log12ppay.dbf
13     /logsppayA/oradata/ppay/log13ppay.dbf
13     /logsppayB/oradata/ppay/log13ppay.dbf
14     /logsppayA/oradata/ppay/log14ppay.dbf
14     /logsppayB/oradata/ppay/log14ppay.dbf

```

V.7.3

Definición de los parámetros en el archivo de init.ora para habilitar el modo ARCHIVE de la Base de Datos .

```

log_checkpoint_timeout = 300
log_archive_start = true
log_archive_dest = /archive/oradata/archive

```


**v.8.1****QUERYS**

Información acerca de los rollback segments que contiene la Base de Datos
Rollback.sql

```
col segment_name for a8  
col tablespace_name for a10  
select segment_name, tablespace_name, initial_extent,next_extent, status  
from dba_rollback_segs  
/
```

Para saber si no existe contención en los segmentos de rollback se debe de ejecutar el siguiente query y si la columna RATIO es mayor a 5%, esto significa que se tiene contención por lo tanto es necesario crear mas segmentos de rollback.

roll_contention.sql

```
Select sum(waits) * 100/sum(gets) "Ratio",  
       Sum(waits) "Waits", sum(gets) "Gets"  
From V$rollstat;
```

Al ejecutar este query se puede encontrar información que contiene el nombre del usuario y el nombre del segmento de rollback que esta utilizando.

Ses_roll.sql

```
Select s.username, rn.name  
From v$session s, v$transaction t, v$rollstat, v$rollname rn  
Where s.saddr = t.ses_addr  
And t.xidusn = r.usn  
And r.usn = rn.usn;  
COL OB_ID FOR 99999  
COL TYPE FOR A5  
COL NAME FOR A25  
COL SESID FOR 99999  
COL USER FOR A8  
COL OSUSER FOR A8  
COL PROCESS FOR 99999  
COL MODE FOR 99
```

```
SELECT LO.object_id "OB_ID", AO.object_type "TYPE", AO.object_name "NAME",  
       LO.session_id "SESID", LO.oracle_username "USER",  
       LO.os_user_name "OSUSER", LO.process "PROCESS", LO.locked_mode "MODE"  
FROM v$locked_object LO, all_objects AO  
WHERE LO.object_id = AO.object_id  
ORDER BY 3  
/
```


GLOSARIO

| | |
|---------|---------------------------------------------|
| 1FN.- | Primera Forma Normal |
| 2FN.- | Segunda Forma Normal |
| 3FN.- | Tercera Forma Normal |
| ANSI.- | Instituto Nacional de Estándares Americanos |
| ARCH.- | Archivador |
| BD.- | Base de Datos |
| CASE.- | Ingeniería de Software para Desarrollo |
| CPU.- | Unidad Central de Procesamiento |
| CHKPT.- | Punto de Inspección |
| DBA.- | Administrador de Bases de Datos |
| DBMS.- | Sistema Manejador de Bases de Datos |
| DBWR.- | Escritor de Base de Datos |
| DCL.- | Lenguaje de Control de Datos |
| DDL.- | Lenguaje de Definición de Datos |
| DML.- | Lenguaje de Manipulación de Datos. |
| E/S.- | Entrada/Salida |
| ERD.- | Diagrama Entidad – Relación |
| HLR.- | Registro de Localización |
| ITU.- | Unión Internacional de Telecomunicaciones |
| IVR.- | Unidad de Voz Interactiva |
| LAN.- | Red de Area Local |
| LGWR.- | Escritor de Archivos Log |
| LRU.- | Algoritmo del Menos Recientemente Usado |
| MSC.- | Centro de Switcheo Movil |
| OCI.- | Interface de Llamadas de Oracle |
| JLTP.- | Procesamiento de Transacciones en Línea |
| PC.- | Computadora Personal |
| PGA.- | Area Global de Programa |

| | |
|----------|---------------------------------------------------|
| PK.- | Llave Primaria |
| PL/SQL.- | Lenguaje de Programación SQL |
| PMON. | Monitor de Procesos |
| PPAS.- | Sistema de Administración de Prepago |
| RAM.- | Memoria de Acceso Aleatorio |
| RBS.- | RadioBases |
| RECO.- | Restaurador |
| SGA.- | Area Global de Sistema |
| SGBDR.- | Sistema de Gestión de Bases de Datos Relacionales |
| SMON.- | Monitor de Sistema |
| SQL.- | Lenguaje de Consultas Estructurado |
| SS7.- | Sistema de Señalización No. 7 |



BIBLIOGRAFIA

- Oracle Design (Database & Code Design)
Dave Ensor & Ian Stevenson
O'Reilly
Marzo 1997
1a. Edición
- Oracle Performance Tunning
Mark Gurry & Peter Corrigan
O'Reilly
Noviembre 1996
2a. Edición
- Fundamentos de Bases de Datos
Henry F. Korth
Abraham Silberschätz
Mc Graw Hill
2a. Edición
1993
- Diseño de Bases de Datos
Gio Wiederhold
Mc Graw Hill
2a. Edición
Septiembre 1990
- Oracle Manual del Administrador
Kevin Lonely
Mc Graw Hill
1a. Edición
Marzo 1996
- Diseño de Bases de Datos Relacionales
Miguel Castaño
Mario Piattini
Alfaomega
1a. Edición
- Fundamentos y Modelos de Bases Datos
Miguel Castaño
Mario Piattini
Alfaomega
1a. Edición

- Oracle 8i Concepts
Release 2 (8.1.6)
Lance Ashdown, Steve Bobrowski
Oracle Corporation
Diciembre 1999
- Oracle 8i Administrator's Guide
Release 2 (8.1.6)
Alex Tsukerman, Andre Kruglikov
Oracle Corporation
Diciembre 1999
- Oracle 8i Installation Guide
Release 2 (8.1.6) for Sparc Solaris
Jags Srinivasan, Lois Price
Oracle Corporation
Diciembre 1999
- Oracle 8i Migration
Release 3 (8.1.7)
Nipun Agarwal, Karleen Aghevli
Oracle Corporation
Septiembre 2000
- Oracle 8i Designing Tuning for Performance
Release 2 (8.1.6)
T. Akiba, Ahmed Alomari
Oracle Corporation
Diciembre 1999

<http://metalink.oracle.com>

http://download-east.oracle.com/otndoc/oracle9i/901_doc/server.901/a90125/toc.htm

http://download-east.oracle.com/otndoc/oracle9i/901_doc/server.901/a87503/instance.htm#13547

<http://docs.oracle.com>

TESIS CON
FALLA DE ORIGEN