



# UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE ESTUDIOS SUPERIORES  
CUAUTITLAN

"SISTEMAS DE INFORMACION. CREACION Y PUESTA A PUNTO DE UNA BASE DE DATOS UTILIZANDO ORACLE PARA OFICINA CERO EN SEGUROS TEPEYAC, S. A. DE C. V."

298025

## TRABAJO DE SEMINARIO

QUE PARA OBTENER EL TITULO DE LICENCIADO EN INFORMATICA  
P R E S E N T A :  
ENRIQUE ALBERTO MORENO SOLIS

ASESOR: M.C.C. VALENTIN ROLDAN VAZQUEZ



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).

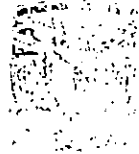
El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición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

FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLAN  
UNIDAD DE LA ADMINISTRACION ESCOLAR  
DEPARTAMENTO DE EXAMENES PROFESIONALES



UNIVERSIDAD NACIONAL  
AUTÓNOMA DE  
MÉXICO

UNIVERSIDAD NACIONAL  
FACULTAD DE ESTUDIOS  
SUPERIORES CUAUTITLAN



DEPARTAMENTO DE  
EXAMENES PROFESIONALES

DR. JUAN ANTONIO MONTARAZ CRESPO  
DIRECTOR DE LA FES CUAUTITLAN  
P R E S E N T E

ATN: Q. Ma. del Carmen García Mijares  
Jefe del Departamento de Exámenes  
Profesionales de la FES Cuautitlán

Con base en el art. 51 del Reglamento de Exámenes Profesionales de la FES-Cuautitlán, nos permitimos comunicar a usted que revisamos el Trabajo de Seminario Sistemas de información.

"Creación y puesta a punto de una Base de Datos utilizando

Oracle para Oficina Cero en Seguros Tepeyac S.A. de C.V."

que presenta el pasante: Enrique Alberto Moreno Solís  
con número de cuenta: 9651072-4 para obtener el título de  
Licenciado en Informática

Considerando que dicho trabajo reúne los requisitos necesarios para ser discutido en el EXÁMEN PROFESIONAL correspondiente, otorgamos nuestro VISTO BUENO

ATENTAMENTE  
"POR MI RAZA HABLARA EL ESPIRITU"

Cuautitlán Izcalli, Méx. a 6 de Septiembre de 2001

| MODULO     | PROFESOR                              | FIRMA          |
|------------|---------------------------------------|----------------|
| <u>I</u>   | <u>M.C.C. Araceli Nivón Zaghi</u>     | <u>[Firma]</u> |
| <u>II</u>  | <u>M.C.C. Valentín Roldán Vázquez</u> | <u>[Firma]</u> |
| <u>III</u> | <u>M.C.C. Miguel Alvarez Pasaye</u>   | <u>[Firma]</u> |

## INDICE

|                   |   |
|-------------------|---|
| INTRODUCCION..... | 3 |
| OBJETIVOS.....    | 6 |
| HIPOTESIS.....    | 7 |

## PARTE I

MARCO TEORICO

|  |    |
|--|----|
| 1 BASES DE DATOS.....  | 9  |
| 1.1 ORIGEN DE LOS DATOS.....                                 | 9  |
| 1.2 BASE DE DATOS.....                                       | 9  |
| 1.3 SISTEMAS DE ADMINISTRACION DE BASES DE DATOS (DBMS)..... | 11 |
| 1.4 ENFOQUE RELACIONAL.....                                  | 13 |
| 2 ORACLE.....  | 14 |
| 2.1 ORACLE SERVER.....                                       | 15 |
| 2.1.1 MECANISMOS DE SEGURIDAD.....                           | 15 |
| 2.1.2 COPIA DE SEGURIDAD Y RECUPERACIÓN.....                 | 16 |
| 2.1.3 GESTIÓN DEL ESPACIO.....                               | 16 |
| 2.1.4 ORACLE LOADER Y SQL * CONNECT.....                     | 16 |
| 2.1.5 HERRAMIENTAS DE DESARROLLO.....                        | 17 |
| 2.1.6 SQL * Plus.....  | 17 |
| 2.1.7 ORACLE REPORTS: EL ESCRITOR DE INFORMES.....           | 18 |
| 2.1.8 DEVELOPER 2000 Y DESIGNER 2000.....                    | 19 |
| 2.2 ARQUITECTURA ORACLE.....                                 | 19 |
| 2.2.1 LOS ESPACIOS DE TABLAS (TABLESPACES).....              | 19 |
| 2.2.2 INSTANCIAS.....  | 21 |
| 2.2.3 USUARIOS.....  | 21 |
| 2.2.4 ÍNDICES.....   | 22 |
| 2.2.5 VISTAS.....  | 22 |
| 2.2.6 PROCEDIMIENTOS Y FUNCIONES.....                        | 22 |
| 2.2.7 PAQUETES, PACKAGES.....                                | 23 |
| 2.2.8 DISPARADORES, TRIGGERS.....                            | 23 |
| 2.2.9 SINÓNIMOS.....   | 23 |
| 2.2.10 PRIVILEGIOS Y ROLES.....                              | 23 |
| 2.2.11 ESTRUCTURAS DE MEMORIA INTERNAS.....                  | 24 |
| 2.2.12 ESTRUCTURAS DE PROCESO.....                           | 24 |
| 2.2.13 FICHEROS.....   | 24 |
| 2.2.14 EL CÓDIGO ORACLE.....                                 | 25 |
| 3 EL SISTEMA OPERATIVO UNIX.....                             | 26 |
| 4 LENGUAJE SQL.....  | 28 |
| 4.1 MANIPULANDO LA INFORMACION.....                          | 29 |
| 4.2 OPERADORES.....  | 31 |
| 4.3 AGRUPACIONES Y ORDENACIONES.....                         | 33 |
| 4.4 TIPOS DE DATOS.....                                      | 36 |

## PARTE II

CASO PRACTICO

|          |   |            |
|----------|---|------------|
| <b>1</b> | <b>DESCRIPCION DEL PROBLEMA</b> .....   | <b>41</b>  |
| <b>2</b> | <b>INSTALACION DE ORACLE</b> .....  | <b>42</b>  |
| 2.1      | CARACTERÍSTICAS DE HARDWARE Y SOFTWARE.....   | 42         |
| 2.2      | CONFIGURANDO EL AMBIENTE.....   | 43         |
| 2.2.1    | TAREAS A CONFIGURAR COMO USUARIO root.....  | 43         |
| 2.2.2    | TAREAS A CONFIGURAR CON EL USUARIO oracle8i.....                                    | 45         |
| 2.2.3    | CONFIGURACION DE TAREAS PARA LOS PRODUCTOS ORACLE.....                              | 46         |
| 2.3      | INSTALACION.....  | 47         |
| 2.3.1    | SELECCIÓN DEL TIPO DE INSTALACION.....  | 47         |
| 2.3.2    | INICIANDO EL INSTALADOR UNIVERSAL DE ORACLE (OUI).....                              | 48         |
| 2.3.3    | EJECUTANDO EL OUI.....  | 49         |
| 2.3.4    | INSTALACION DE LOS PRODUCTOS.....   | 50         |
| <b>3</b> | <b>INSTALACION DE LA BASE DE DATOS</b> .....  | <b>51</b>  |
| 3.1      | DEFINICION DE ESPACIOS.....   | 51         |
| 3.2      | CREANDO SCRIPTS.....  | 52         |
| 3.2.1    | ARCHIVO DE INICIALIZACION.....  | 52         |
| 3.2.2    | CODIGOS.....  | 53         |
| 3.3      | MODIFICACIONES FINALES.....   | 56         |
| <b>4</b> | <b>AMBIENTE DEL USUARIO</b> .....   | <b>59</b>  |
| 4.1      | CREACION DE USUARIOS Y PERFILES.....  | 60         |
| 4.2      | OBTENCION DE LA ESTRUCTURA DE DATOS DE LAS TABLAS DESDE AMBIENTE DE PRODUCCION..... | 62         |
| 4.3      | OBTENCION DE LOS INDICES DE LAS TABLAS.....   | 62         |
| 4.4      | CREACION DE LOS SCRIPTS PARA TABLAS.....  | 64         |
| 4.5      | CREACION DE LOS SCRIPTS PARA INDICES.....   | 65         |
| 4.6      | CREACION DE LIGAS.....  | 66         |
| 4.7      | EJECUCION DE LOS SCRIPTS.....   | 67         |
| 4.8      | PERMISOS PUBLICOS A LAS TABLAS.....   | 68         |
| 4.9      | COPIA DE LOS PROGRAMAS DESDE DESARROLLO.....  | 68         |
| 4.10     | CREACION DE TODOS LOS DEMAS USUARIOS.....   | 69         |
| <b>5</b> | <b>PRINCIPALES TAREAS DE ADMINISTRACION DE LA BASE DE DATOS</b> .....               | <b>77</b>  |
| 5.1      | MANEJO DE LOS SEGMENTOS DESAHER.....  | 78         |
| 5.2      | MANEJO DE LAS TABLAS.....   | 82         |
| 5.3      | MANEJO DE LOS INDICES.....  | 83         |
| 5.4      | MANEJO DE LOS USUARIOS.....   | 86         |
| 5.5      | MANEJO Y REORGANIZACION DE LOS DATOS.....   | 89         |
| 5.6      | RESPALDO.....   | 91         |
|          | <b>CONCLUSION</b> .....   | <b>94</b>  |
|          | <b>ANEXO</b> .....  | <b>95</b>  |
|          | <b>BIBLIOGRAFIA</b> .....   | <b>103</b> |

## INTRODUCCION

Seguros Tepeyac S.A de C.V, como empresa dedicada a la venta de seguros es una de las empresas más grandes en su ramo a nivel Nacional e importante a nivel mundial, por este motivo el manejo de su información resulta una parte estratégica del negocio.

Existen dos ambientes de Base de Datos:

- Producción; Aquí es en donde se manejan los datos reales de la compañía
- Desarrollo: En este ambiente se manejan datos ficticios, aquí se crean los programas y objetos necesarios para después ser traspasados al ambiente de producción.

En el presente trabajo se explicará la instalación de una base de datos para el área de Oficina Cero (Ofi0) en Seguros Tepeyac S.A de C.V en un servidor independiente. La base de datos que se pretende realizar y el nuevo ambiente a crear serán intermedios entre los dos ambientes mencionados, es decir, serán un puente en el cual los usuarios serán los encargados de realizar las pruebas, antes de dar su visto bueno a los programas creados en desarrollo para que pasen a producción. Se esperan grandes ventajas ya que el procedimiento actual de pruebas se lleva en el mismo ambiente que desarrollo lo cual provoca saturación de información y lentitud en el proceso de liberación de programas. El punto crítico esta en la emisión de pólizas ya que este es el módulo que presenta la mayor cantidad de necesidades (ya sea modificación de programas u objetos ya existentes o creación de nuevos), por esto, la creación de la base de Ofi0 será inicialmente con los objetos necesarios para esta parte del proceso.

La base de datos será creada bajo Oracle 8i con una plataforma UNIX versión 11 de HP y haciendo uso de una Máquina HP 9000 K380, ya que estos son los recursos con los que la empresa dispone para el desarrollo de este proyecto.

El presente trabajo dará a conocer principalmente los siguientes temas divididos en dos partes:

## Parte I Marco Teórico.

Esta primera parte da una introducción a algunos temas y conceptos que serán utilizados durante la segunda parte del trabajo como son: conceptos de base de datos, definiciones, UNIX, SQL, y sobre todo arquitectura de Oracle como RDBMS.

## Parte II Caso Práctico:

- a) **INSTALACION DE ORACLE** : Se tomará el proceso de preinstalación (a nivel UNIX), configuración del ambiente, instalación y postinstalación del software.
- b) **INSTALACION DE LA BASE DE DATOS.** Una vez que Oracle esta instalado se hará la creación de la base de datos en la misma máquina, se explicará el proceso de planeación de los espacios, y desarrollo de los scripts necesarios para la creación de los objetos necesarios para el funcionamiento idónea de la base.
- c) **CREACION DEL AMBIENTE DEL USUARIO.** Obtención de los datos necesarios para la recreación de objetos de emisión, formulación de los scripts, copia de los programas y creación de los usuarios y roles necesarios para el ambiente de Ofi0
- d) **PRINCIPALES TAREAS DE ADMINISTRACION DE LA BASE DE DATOS** En este último capítulo se habla del manejo y mantenimiento de los diferentes objetos de la base de datos como son índices, tablas, segmentos deshaces, espacios de tablas, usuarios y perfiles, respaldos de la información.

El traspaso de programas y objetos de Desarrollo a Producción ahora tendrá un paso intermedio llamado Oficina Cero. Este traspaso de programas se llevará a cabo por medio de una solicitud que tendrá un número consecutivo, este número lo tendrá desde que el usuario hace el requerimiento al área de desarrollo hasta la liberación por parte de Base de Datos (DBA's) a producción.

El procedimiento para liberación de requerimientos hacia oficina cero y producción es el siguiente:

El área de desarrollo recibe el requerimiento por parte del usuario final, se desarrolla o modifican programas u objetos necesarios. El área de desarrollo será la responsable de liberar el programa u objeto hacia oficina cero con un número de solicitud, el área de Administración de Base de Datos recibe el requerimiento, hace las modificaciones pertinentes e informa al área de desarrollo. El área de desarrollo informa al usuario que realice sus pruebas y este es el encargado de rechazar o liberar el requerimiento hacia al área de Administración de Base de Datos utilizando el mismo número de solicitud usado al momento de la liberación a oficina cero. Si las modificaciones son aprobadas por el usuario y liberado a producción entonces el área de Base de Datos hace los cambios en producción e informa tanto al usuario como al área de desarrollo cuando estén listos.



---

## OBJETIVOS

### Objetivo General

Crear un ambiente de base de datos bajo Oracle para mejorar las pruebas de los programas de aplicación en Seguros Tepeyac S.A de C.V

### Objetivos Particulares

- I. Beneficiar las pruebas de los programas para que pasen al ambiente de producción funcionando de forma óptima.
- II. Mejorar la administración y control de los objetos creados en ambiente de pruebas y ambiente de producción .
- III. Lograr que el procedimiento de petición, desarrollo y liberación de objetos sea más rápido y sin fallas.
- IV. Mejorar la distribución de los espacios tanto en ambiente Desarrollo, Ofna0 y Producción.
- V. Independizar procesos de ambiente Desarrollo y Pruebas mejorando el rendimiento de equipos.

---

## HIPOTESIS

Si se crea una base de datos específica para el ambiente de pruebas en Seguros Tepeyac S.A de C.V. se verán beneficiados los siguientes aspectos

- Ya que los usuarios harán pruebas de los programas antes de pasarlos a ambiente real (producción), estos programas llegarán a dicho ambiente funcionando como el usuario lo requiera.
- La independencia de los ambientes de pruebas y desarrollo mejorarán el rendimiento de los equipos.
- Al quedar la liberación de los objetos por manos de los DBA's con un número de solicitud, se tendrá un mejor control de los programas, objetos y procedimientos liberados.
- Al poder hacer los usuarios sus propias pruebas mejorará el tiempo de respuesta desde que hagan la petición hasta la liberación del requerimiento en producción.

## PARTE I

### MARCO TEORICO

En esta primera parte encontraremos la descripción y justificación del problema así como conceptos de Bases de datos, Oracle, SQL y UNIX, éstas dos últimas, herramientas indispensables para la creación, puesta a punto y administración de una base de datos bajo Oracle.

## 1 BASES DE DATOS

### 1.1 ORIGEN DE LOS DATOS

Un origen de datos (data source) incluye los datos a los que desea acceder un usuario, así como la información necesaria para acceder a dichos datos, algunos ejemplos de orígenes de datos son:

- Un directorio que contiene una serie de archivos DBASE a los que se desea acceder.
- Una base de datos SQL Server, el servidor en que reside dicha base de datos y la red utilizada para acceder a ese servidor.

La percepción del mundo puede ser descrita como una sucesión de fenómenos, que desde el comienzo de los tiempos el hombre ha tratado de descubrir, ya sea que los entienda completamente o no. La descripción de estos fenómenos es llamada datos. Los datos corresponden a registros de hechos acerca de un fenómeno con lo cual ganamos información sobre el mundo que nos rodea. La Información pasa a ser el incremento del conocimiento que puede ser inferido de los datos. Usualmente el dato y su significado son registrados juntos, ya que el lenguaje natural es lo suficientemente poderoso para hacerlo.

### 1.2 BASE DE DATOS

Una Base de Datos (BD) mantiene y reúne información a través de estructuras de datos, sobre esta estructura de datos se puede actualizar, borrar, etc. la información.

Las BD se mantienen en dispositivos de acceso directo, dentro de un volumen puede haber más de una Base de datos. La BD en sí, esta formada por un conjunto de tablas, consultas e informes, dentro de un volumen pueden existir varias bases de datos que se pueden interrelacionar y debe tener mínimo una tabla, una BD se puede definir como un mantenedor de registros, es decir, un sistema cuyo propósito general es registrar y mantener

---

información, que puede estar relacionada con cualquier cosa u objeto que sea significativo para la organización.

En un sistema de BD se distinguen 4 componentes:

1. Datos
2. Software
3. Hardware
4. Usuarios

☒ **DATOS:** Los datos almacenados se dividen en una o más Bases de Datos. Por lo tanto, una Base de Datos es un recipiente de datos almacenados y en general es *Integrada* como *Compartida*.

- † *BD Integrada:* Se entiende que la base de datos puede considerarse como una unificación de varios archivos de datos independientes de donde se elimina parcial o totalmente cualquier redundancia entre los mismos.
- † *BD Compartida:* Partes individuales de las Bases de Datos pueden compartirse entre varios usuarios distintos en el sentido de que cada uno de ellos puede tener acceso a la misma parte de la Base de datos y usarla con propósitos distintos.

☒ **SOFTWARE:** Entre las bases de datos físicas, es decir, el almacenamiento real de datos y los usuarios, existe un nivel de software que recibe el nombre de DBMS (Sistema manejador de base de datos). Su objetivo es manejar todas las solicitudes formuladas por los usuarios y/o programas de acuerdo a una función general del DBMS, otro objetivo es proteger a los usuarios contra los detalles a nivel de hardware, es decir, el DBMS ofrece una visión a los usuarios que está por encima del hardware y apoya las operaciones de este.

- ☐ **HARDWARE:** Está compuesto por el espacio de disco, tambores, etc., en donde reside la BD, junto con los dispositivos asociados como unidades de control y canales entre otros, ya que la BD es demasiado grande para localizarla en memoria.
  
- ☐ **USUARIOS:** Existen 3 tipos de usuarios en la BD:
  - † *Usuario Final:* Usa la BD realizando consultas a través de un lenguaje de consulta o a través de un programa.
  - † *Segundo Usuario*\*: Programador de aplicaciones, desarrolla los sistemas necesarios para permitir la posibilidad de comunicación o extensión de información desde la Base de Datos.
  - † *Administrador de la Base de Datos(DBA):* Dentro de sus funciones podemos mencionar:
    - ☛ Mantener en forma óptima y eficiente la base de datos controlando procedimientos, instalaciones, procesos, etc.
    - ☛ Realizar funciones de auditoría, manejando la seguridad de la Base de Datos, además de crear usuarios y accesos permitidos.

El DBA es la persona que toma decisiones estratégicas y políticas con respecto a la información de la empresa, además de diseñar los aspectos técnicos necesarios para poner en práctica estas decisiones, es decir, será encargado del control general del problema a nivel técnico.

### 1.3 SISTEMAS DE ADMINISTRACION DE BASES DE DATOS (DBMS)

Es un conjunto de programas que manejan todo el acceso a la BD de la siguiente forma:

- 1.- Un usuario solicita acceso empleando algún sublenguaje de datos.
- 2.- El DBMS interpreta la solicitud y la analiza.

---

\* Para poder ser segundo usuario debe saber usar ciertas aplicaciones.

- 3.- El DBMS inspecciona en orden el esquema externo de ese usuario, la correspondencia externa conceptual asociada, el esquema conceptual, la correspondencia conceptual interna y la estructura de almacenamiento.
- 4.- El DBMS ejecuta las operaciones necesarias sobre la base de datos almacenados.

Dentro de las funciones principales que realiza un DBMS tenemos:

- 1.- **DEFINICION DE DATOS:** Es capaz de aceptar definiciones de datos (esquema externo, conceptual interno y todas las correspondencias asociadas en versión fuente) y convertirlas en una versión de objeto apropiada. Debe incluir componentes de procesadores de lenguajes para cada uno de los diversos lenguajes de definición de datos, además debe tener las definiciones en DBL (Lenguaje de Definición de datos) para poder interpretar y resolver las solicitudes.
- 2.- **MANIPULACION DE DATOS:** Atiende las solicitudes del usuario tales como inserción, eliminación, modificación y extracción, es decir, debe incluir un componente procesador de Lenguajes de Manipulación de Datos (DML). En general las solicitudes en DML pueden ser planificadas y no planificadas.
- 3.- **SEGURIDAD E INTEGRIDAD DE LOS DATOS:** El DBMS supervisa las solicitudes de los usuarios y rechaza los intentos de violar las necesidades de seguridad e integridad de los datos definidos por el DBA.
- 4.- **RECUPERACION Y CONCURRENCIA DE LOS DATOS:** El DBMS cuida el cumplimiento de ciertos controles de recuperación y concurrencia. El Administrador de Transacciones actúa en caso de que el DBMS no funcione.
- 5.- **DICCIONARIO DE DATOS:** El DBMS incluye una función de diccionario de datos (Meta\_Datos), se puede decir que es una Base de Datos del sistema y no del usuario cuyo contenido puede considerarse como "Datos acerca de los Datos" que en el fondo son definiciones de objetos de datos y otros objetos del sistema.

6.- ADMINISTRADOR DE COMUNICACIONES DE DATOS: Las solicitudes de un usuario final son dirigidas a la Base de Datos donde son transmitidas en forma de mensajes de comunicación o a la inversa, las respuestas al usuario son tomadas como mensajes del mismo tipo, todas las transmisiones se efectúan bajo el control de otros grupos de programas llamados Administradores de Comunicación de Datos el cual no forma parte del DBMS, sino que es un programa autónomo pero debe trabajar en forma conjunta con el DBMS para poder satisfacer los requerimientos de los usuarios.

#### 1.4 ENFOQUE RELACIONAL

El modelo relacional es una forma de ver los datos, es decir, una receta para representar los datos mediante tablas y para manipular su representación.

Los componentes de la estructura son básicamente Relación, Tupla, Cardinalidad, Atributo, Grado, Dominio y Clave Primaria.

Una relación corresponde a lo que se conoce como tabla o entidad; una tupla corresponde a una fila de esa tabla, un atributo corresponde a una columna de la misma tupla, el número de tuplas se denomina cardinalidad, el número de atributos se llama grado, la clave primaria es un identificador único para la tabla, es decir, una columna o combinación de una columna con la siguiente propiedad "NUNCA EXISTEN DOS FILAS DE LA TABLA CON EL MISMO VALOR DE UNA COLUMNA O COMBINACION DE VARIAS COLUMNAS", el Dominio es una combinación de valores de los cuales uno o más atributos obtienen sus valores reales.

La estructura de un enfoque relacional son las asociaciones entre tuplas y se representan únicamente por valores de datos en las columnas sacadas de un dominio común.



| TERMINO RELACIONAL | TERMINO INFORMAL    |
|--------------------|---------------------|
| RELACION           | Tabla               |
| TUPLA              | Fila o registro     |
| CARDINALIDAD       | Número de filas     |
| ATRIBUTO           | Campo o columna     |
| GRADO              | Número de columna   |
| CLAVE PRIMARIA     | Identificador único |
| DOMINIO            | Valores legales     |

TABLA 1. Comparación de términos

Existen dos tipos de relaciones:

1.- *Relaciones reales*: Corresponde al concepto de Tabla, es decir, una relación autónoma cuya importancia está dada por el diseñador para un uso específico dentro de una aplicación

2.- *Relaciones virtuales*: Una vista es una relación derivada que posee su propio nombre representada dentro del sistema exclusivamente mediante su definición en término de otras relaciones, no posee datos almacenados propios, separados y distinguibles a diferencia de las relaciones reales.

## 2 ORACLE

En sus inicios, Oracle era simplemente una empresa de bases de datos relacionales, las bases de datos relacionales eran una forma de pensar sobre cómo debían estructurarse y almacenarse los datos. La clave de este tipo de base de datos consiste en entender las relaciones existentes entre los datos, y en estructurar la BD para que refleje dichas relaciones. De esta forma, puede construirse una BD que resista el paso del tiempo, el objetivo de una BD relacional consiste en construir una base de datos en la que sólo cambien los datos pero no en la estructura de la misma.

Hoy en día, Oracle Corporation, con una base en Redwood Shores, California, fabrica productos de software y ofrece servicios para la gestión electrónica de la

información. Oracle es un suministrador de software para computadores de ámbito mundial, con unos ingresos actuales superiores a 3,000 millones de dólares y con negocios en más de 90 países de todo el mundo.

## 2.1 ORACLE SERVER

Oracle Server es un moderno entorno de gestión de la información. Es un almacén para grandes cantidades de datos y le proporciona al usuario un rápido acceso a dichos datos. Oracle Server ofrece la compartición de datos entre las aplicaciones con lo cual la información está almacenada en un lugar y se utiliza en muchos sistemas. Oracle Server puede ejecutarse en docenas de plataformas distintas, y admite las siguientes configuraciones:

- Basado en anfitrión: Los usuarios se conectan directamente al mismo computador en el que se encuentra la base de datos.
- Cliente / servidor: Los usuarios acceden a la base de datos desde su computadora independiente.
- Procedimiento distribuido: Los usuarios acceden a una base de datos que reside en más de un computador. La base de datos está repartida en más de una máquina y los usuarios no son conscientes de la ubicación física de los datos con los que trabajan

Oracle vende su tecnología de servidor con varias opciones adicionales que mejoran su capacidad. El producto básico ofrece todas las funciones necesarias para los requisitos de la mayoría de los clientes de Oracle. Cuando los clientes necesitan las opciones procedimental (procedural), distribuida (distributed), consulta en paralelo (parallel query) o servidor en paralelo (parallel server) para cumplir con los requisitos de sus negocios, sólo tiene que comprarlas además del producto Server

### 2.1.1 MECANISMOS DE SEGURIDAD

Los sofisticados mecanismos de seguridad de Oracle permiten controlar el acceso a datos delicados mediante una gran variedad de privilegios, a los usuarios se les conceden

derechos para examinar, modificar y crear datos. Los clientes utilizan estos mecanismos para asegurarse que los usuarios determinados puedan acceder a los datos delicados, que estarán prohibidos para el resto de los usuarios.

### 2.1.2 COPIA DE SEGURIDAD Y RECUPERACIÓN

Oracle proporciona unas sofisticadas rutinas de copia de seguridad y recuperación. La copia de seguridad crea una copia secundaria de los datos de Oracle, mientras que la recuperación reestablece una copia de los datos a partir de dicha copia de seguridad. Con la estrategia de copia de seguridad y recuperación de Oracle se minimiza la pérdida de datos y el tiempo de indisponibilidad en caso de que se produzcan problemas.

### 2.1.3 GESTIÓN DEL ESPACIO

Oracle ofrece una gestión flexible del espacio. Es posible asignar espacio en disco para el almacenamiento de datos y controlar las asignaciones subsiguientes indicándole a Oracle cuánto espacio debe reservar para los requisitos futuros.

### 2.1.4 ORACLE LOADER Y SQL \* CONNECT

En el pasado, Oracle tenía una herramienta especial denominada Oracle Data Loader (ODL) o Cargador de Datos de Oracle, que con el paso del tiempo ha evolucionado hasta convertirse en Oracle Loader, una herramienta que permite cargar datos en una base de datos Oracle en una forma muy rápida. Una gran cantidad de sistemas sigue utilizando el enfoque tradicional. Una BD relacional es una buena herramienta, pero para la migración de los sistemas tradicionales a los sistemas relacionales hace falta convertir los datos de los sistemas existentes, ese es el cometido de Oracle Loader, es una herramienta que permite introducir datos en una base de datos de Oracle.

SQL \* Connect es una herramienta que permite que una base de datos Oracle pueda conectarse y comunicarse con una base de datos tradicional bajo cualquier plataforma.

Una vez que los datos se encuentran en una base de datos Oracle, con las nuevas herramientas es más sencillo generar informes y pantallas para manejar su contenido. También es una realidad que esta migración puede tardar años en conseguirse. Oracle ha desarrollado herramientas para transferir sus datos como Oracle Loader y herramientas para comunicarse con una base de datos tradicional, como SQL \* Connect.

### 2.1.5 HERRAMIENTAS DE DESARROLLO

Oracle Server, denominado habitualmente el motor de la base de datos, admite una amplia gama de herramientas de desarrollo, herramientas de consulta para el usuario final, aplicaciones listas para usar y herramientas de gestión de la información de ámbito de empresa.

### 2.1.6 SQL \* Plus

SQL (Structured Query Language) es el estándar de la industria adoptado por todos los vendedores de base de datos. SQL \* Plus de Oracle es un superconjunto del SQL estándar; se ajusta a las normas de un lenguaje compatible con SQL y tiene varias características adicionales específicas de Oracle.

SQL está adaptado para aprovechar la potencia del modelo relacional, como todos los datos están almacenados mediante relaciones, es posible trabajar con los datos en conjuntos, en contraposición con la utilización de filas independientes de datos. Con SQL \* Plus desaparece la mayoría del trabajo asociado con la recuperación de datos de una base de datos tradicional, por ejemplo, ya no se lee un registro, en su lugar, se escribe un programa que se encarga de todos los registros asociados con una entidad. Ningún registro es distinto de los demás. En SQL \* Plus cualquier acción que decida ejecutar se llevará a cabo como un conjunto completo. SQL se explicará más adelante en éste capítulo de una manera más detallada.

### 2.1.6.1 ORACLE FORMS

Teniendo una poderosa BD y complementada con un potente lenguaje para la creación de informes (SQL \* Plus), es natural que Oracle se complemente con un generador de pantalla ya que es la principal forma que tienen los usuarios de tratar con la base de datos. Dicha herramienta es Oracle Forms. Se ejecuta en un entorno de interfaz gráfica de usuario. (GUI; Graphical User Interface), con el aspecto y funcionamiento de Microsoft Windows.

Los desarrolladores de aplicaciones diseñan las pantallas de introducción de datos y de consulta con Oracle Forms, los usuarios finales pueden utilizar dichas pantallas para manejar los datos en la base de datos Oracle. La interfaz con el usuario está controlada por eventos, por teclas de función o por el teclado numérico. Como se muestra en la siguiente Tabla, con Oracle Forms pueden escribirse tres tipos de módulos que, agrupados, ofrecen al desarrollador y al usuario final una solución completa de introducción de datos de empresa basada en formularios.

| TIPO DE PROGRAMA | CONTENIDO  |
|------------------|--|
| Formularios      | Permiten la actualización, creación y eliminación de datos dentro de los objetos de Oracle             |
| Menú             | Define una amplia variedad de menús principales y un subconjunto opcional de submenús                  |
| Biblioteca       | Es un almacén para el código centralizado PL/SQL al que acceden otros tipos de módulos de Oracle Forms |

TABLA 2. Tipos de módulos de Oracle Forms

### 2.1.7 ORACLE REPORTS: EL ESCRITOR DE INFORMES

Teniendo una poderosa BD relacional, y complementada con un conjunto de herramientas de desarrollo. Oracle siguió expandiéndose. Aunque SQL \*Plus es una buena herramienta para la creación de informes, no se diseñó sólo para eso. Así que Oracle desarrolló varias herramientas sólo para generar informes, entre las que se incluyen

---

SQL\*QMX, RPT y Easy \*SQL. Oracle Reports ( y su predecesor SQL \* Reportwriter) es el buque insignia de Oracle en el tema de la escritura de informes.

Oracle Reports ( a partir de la versión 2.0 ) es un verdadero entorno multimedia de generación de informes, permite incluir imágenes, sonido y gráficos, y los informes pueden presentarse en varios colores y con una amplia variedad de tipos de letra. Con Oracle Reports es muy sencillo crear los estilos de salida de informes más populares. Con sus potentes funciones predeterminadas, los desarrolladores pueden crear informes maestros de detalle, de matriz y circulares casi sin programar.

### 2.1.8 DEVELOPER 2000 Y DESIGNER 2000

Developer 2000 y Designer 2000 ofrecen una solución completa cuando los clientes de Oracle necesitan diseñar, programar, implementar y mantener sistemas. Ofrecen un rápido desarrollo de aplicaciones en un entorno Windows cliente/servidor. Sus funciones avanzadas incorporan Reingeniería de Proceso de Negocios (BPR); y mecanismos para aprovechar el procesamiento del servidor que puede realizarse mediante el motor de base de datos de Oracle; ambos productos tienen un alto grado de integración con Oracle Server. El procesamiento se divide entre el servidor y el cliente utilizando procedimientos almacenados y disparadores (triggers).

## **2.2 ARQUITECTURA ORACLE**

### 2.2.1 LOS ESPACIOS DE TABLAS (TABLESPACES)

Un espacio de tablas es una división lógica de la BD, cada BD tiene al menos uno llamado SYSTEM. Un espacio de tablas puede pertenecer sólo a una BD y los espacios de tablas se utilizan para mantener juntos los datos de usuarios o de aplicaciones para facilitar su mantenimiento o mejorar las prestaciones del sistema.

Cuando se crea una tabla se debe indicar el espacio de tablas al que se destina, por defecto se depositan en el espacio de tablas SYSTEM, este espacio de tablas es el que

---

contiene el diccionario de datos, por lo que conviene reservarlo para el uso del servidor, y asignar las tablas de usuario a otro espacio de tablas.

Cuando se crea un espacio de tablas se le asigna un espacio en disco que Oracle reserva inmediatamente, se utilice o no, si este espacio inicial se ha quedado pequeño Oracle puede gestionar el crecimiento dinámico de los ficheros sobre los que se asientan los espacios de tablas, esto elimina la posibilidad de error en las aplicaciones por fallos de dimensionamiento inicial. Los parámetros de crecimiento del tamaño de los espacios de tablas se especifican en la creación de los mismos.

Dentro de cada espacio de tabla se pueden almacenar objetos de distinta naturaleza: tablas, índices, etc, pero no se deben mezclar, se necesita una manera de separarlos, y eso son los *segmentos*.

Un segmento está contenido en su totalidad en un espacio de tabla. Un segmento está constituido por un conjunto de extensiones, que no son más que grupos de bloques de disco ORACLE contiguos. Cuando se borra un segmento, el espacio es devuelto al espacio de tabla.

Todos los datos de la BD están almacenados en segmentos, existiendo cinco tipos:

- ☐ De datos: almacenan las tablas.
- ☐ De índices: permiten un acceso rápido a los datos dependiendo de la cantidad de los mismos, las consultas que sólo hacen referencia a columnas indexadas se resuelven en el índice. Establecen un control de unicidad (los índices son automáticos cuando se definen claves primarias). Cada índice ocupa un segmento independiente del segmento de datos y deberían estar en un espacio de tablas distinto al de los datos para mejorar el rendimiento.
- ☐ De *rollback*: son objetos internos de la BD que permiten efectuar la restauración de las transacciones no validadas asegurando la consistencia en lectura., son tan importantes que una BD no puede arrancar si no puede acceder al menos a un segmento de *rollback*.

Si la BD tiene múltiples espacios de tablas, deben existir al menos dos segmentos de *rollback* y cada segmento de *rollback* debe tener al menos dos extensiones, reutilizables de manera cíclica. Estos segmentos son un objeto compartido de la BD, aunque se puede asignar un segmento de *rollback* particular a una transacción dada.

- Temporales: son creados por Oracle para un uso temporal cuando debe realizar una ordenación que no le cabe en memoria, y en las operaciones: `create index`, `order by`, `group by`, `distinct`, `union`, `intersect`, `minus`. Son eliminados cuando la sentencia finaliza.
- De *bootstrap*: Se crea en SYSTEM y contiene definiciones del diccionario para sus tablas, que se cargan al abrir la BD, no requiere ninguna acción por parte del DBA y no cambia de tamaño.

### 2.2.2 INSTANCIAS

Para permitir el acceso a los datos, Oracle utiliza un conjunto de procesos que son compartidos por todos los usuarios. Además, existen estructuras de memoria que son utilizadas para almacenar los datos más recientemente solicitados a la BD. Una *instancia* de BD es el conjunto de estructuras de memoria y de procesos que acceden a los ficheros de datos.

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

### 2.2.3 USUARIOS

Una cuenta de usuario no es una estructura física de la BD, pero está relacionada con los objetos de la misma. Existen dos usuarios especiales: SYS y SYSTEM. El usuario SYS posee las tablas del diccionario de datos; que almacenan información sobre el resto de las estructuras de la BD. El usuario SYSTEM posee las vistas que permiten acceder a las tablas del diccionario, para el uso del resto de los usuarios de la BD.



Todo objeto creado en la BD se crea por un usuario, en un espacio de tablas y en un fichero de datos determinado. Toda cuenta de la BD puede estar unida a una cuenta del Sistema Operativo, lo que permite a los usuarios acceder a la cuenta de la BD sin dar la clave de acceso. Cada usuario puede acceder a los objetos que posea o a aquellos sobre los que tenga derecho de acceso.

#### 2.2.4 ÍNDICES

Un índice es una estructura de la BD utilizada para agilizar el acceso a una fila de una tabla. Cada fila tiene un identificador de fila llamado ROWID, que determina el fichero, bloque y fila dentro del bloque donde está almacenada la fila. Cada entrada del índice consiste en un valor clave y una ROWID. Cada una de estas entradas se almacena en un árbol B<sup>+</sup>. Los índices se crean automáticamente cuando se define una restricción UNIQUE o PRIMARY KEY.

#### 2.2.5 VISTAS

Conceptualmente, una vista puede considerarse como una máscara que se extiende sobre una o más tablas, de modo que cada columna de la vista se corresponde con una o más columnas de las tablas subyacentes. Cuando se consulta una vista, esta traspasa la consulta a las tablas sobre las que se asienta. Las vistas no se pueden indexar, no generan almacenamiento de datos y sus definiciones se almacenan en el diccionario de datos.

#### 2.2.6 PROCEDIMIENTOS Y FUNCIONES

Un procedimiento es un bloque de código PL/SQL, que se almacena en el diccionario de datos y que es llamado por las aplicaciones. Se pueden utilizar para implementar seguridad, no dando acceso directamente a determinadas tablas si no es a través de procedimientos que acceden a esas tablas. Cuando se ejecuta un procedimiento se ejecuta con los privilegios de su propietario. La diferencia entre un procedimiento y una función es que ésta última puede devolver valores.

### 2.2.7 PAQUETES, PACKAGES

Se utilizan para agrupar procedimientos y funciones. Los elementos dentro de los paquetes pueden ser públicos o privados. Los públicos pueden ser llamados por los usuarios, los privados están ocultos a los usuarios y son llamados por otros procedimientos.

### 2.2.8 DISPARADORES, TRIGGERS

Son procedimientos que son ejecutados cuando se produce un determinado evento en la BD. Se pueden utilizar para mejorar y reforzar la integridad y la seguridad de la BD.

### 2.2.9 SINÓNIMOS

Para identificar completamente un objeto dentro de una BD se necesita especificar el nombre de la máquina, el nombre del servidor, el nombre del propietario y el nombre del objeto. Para hacer transparente todo esto al usuario se pueden utilizar los sinónimos que apuntarán a los objetos y si el objeto cambia de lugar o propietario, sólo habrá que modificar el sinónimo.

Existen sinónimos públicos y privados, los públicos son conocidos por todos los usuarios de una BD y los privados son locales a un usuario.

### 2.2.10 PRIVILEGIOS Y ROLES

Para que un objeto pueda ser accedido por un usuario debe de tener otorgado ese privilegio. Ejemplos de privilegios son INSERT, SELECT, UPDATE, EXECUTE, entre otros.

Los roles son grupos de privilegios que pueden ser utilizados para facilitar la gestión de los privilegios, los privilegios se pueden otorgar a un role, y los roles pueden ser otorgados a múltiples usuarios.

### 2.2.11 ESTRUCTURAS DE MEMORIA INTERNAS

Oracle mantiene dos estructuras principales de memoria: el Área Global de Programa, (*Program Global Area, PGA*); y el Área Global del Sistema, (*System Global Area, SGA*) o también conocida como *Shared Global Area*.

El PGA es la zona de memoria de cada proceso Oracle. No está compartida y contiene datos e información de control de un único proceso.

El SGA es la zona de memoria en la que la BD Oracle guarda información sobre su estado. Esta estructura de memoria está disponible para todos los procesos, por eso se dice que está compartida.

### 2.2.12 ESTRUCTURAS DE PROCESO

El servidor se vale de una serie de procesos que son el enlace entre las estructuras físicas y de memoria, esto son:

El monitor del sistema (SYSMON)

El monitor de procesos (PMON)

El escritor de la base de datos (DBWR)

Escritor de lógicos (LGWR)

Checkpoint (CHKPT)

Archivador (ARCH)

Recuperador (*RECO*)

### 2.2.13 FICHEROS

*Ficheros redo log*: En ellos se graba toda operación que se efectue en la BD 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.

*Ficheros de control*: Mantienen la información física de todos los ficheros que forman la BD, camino incluido; así como el estado actual de la BD. Son utilizados para mantener la consistencia interna y guiar las operaciones de recuperación. Son imprescindibles para que la BD se pueda arrancar. Contienen:

- ☒ Información de arranque y parada de la BD.
- ☒ Nombres de los archivos de la BD y *redo log*.
- ☒ Información sobre los *checkpoints*.
- ☒ Fecha de creación y nombre de la BD.
- ☒ Estado *online* y *offline* de los archivos.

*Fichero init.ora:* Como parte de la distribución software, Oracle provee de un fichero de parámetros de inicialización llamado *init.ora*. Este fichero contiene los parámetros del sistema Oracle y debe ser utilizado por el DBA para configurar el Sistema Gestor de Base de Datos (SGBD) y adecuarlo a una determinada explotación. Oracle lee este fichero durante el proceso de arranque para determinar el tamaño de la SGA y encontrar los ficheros de control. Como el fichero *init.ora* es fundamental para el arranque de la BD, debería ser copiado frecuentemente para protegerlo de posibles pérdidas.

#### 2.2.14 EL CÓDIGO ORACLE

Cuando el software Oracle se instala en un sistema, se crean subdirectorios y archivos, dependientes todos ellos del sistema operativo (S.O.) Por ejemplo, en el S.O. Unix, y todos los subdirectorios Oracle se encuentran dentro del directorio principal ORACLE\_HOME. Todos estos subdirectorios contienen ficheros ejecutables y *scripts* que son cruciales para el funcionamiento y la administración del SGBD, y es lo que se conoce por el código Oracle. Entre ellos, una herramienta es fundamental en las tareas de administración y puesta en marcha de la BD: *server manager*, *svrmgr*.

### 3 EL SISTEMA OPERATIVO UNIX

El Sistema Operativo UNIX fue creado a finales de la década de los 60 sobre la base de varios trabajos realizados conjuntamente por el MIT y Laboratorios BELL. Dichos trabajos (proyecto MULTICS) iban encaminados a la creación de un macrosistema de computación que diera servicio a miles de usuarios. Si bien el proyecto fracasó, posiblemente por intentar abarcar demasiado contando con unos elementos hardware limitados en esa época, influyó decisivamente sobre la evolución de los sistemas informáticos posteriores.

Un antiguo integrante de dicho proyecto, Ken Thompson, desarrolló por su cuenta un sistema operativo monousuario con la característica principal de un sistema de archivos jerárquico. El sistema encontró muchos entusiastas y se hizo portable al reescribirse casi íntegramente en lenguaje "C", y se suministró en código fuente a las Universidades como objeto de formación, así, la Universidad de California en Berkeley retocó dicho sistema (fundamentalmente comunicaciones y diversas utilidades como el editor "vi")

Los siguientes conceptos son comunes para todos los sistemas UNIX, por lo cual se puede afirmar que éstos componen las características principales de UNIX :

**KERNEL** : Este es el componente principal del sistema operativo. Se encarga de asignar tareas y manejar el almacenamiento de datos. El usuario rara vez opera directamente con el Kernel, que es la parte residente en memoria del sistema operativo.

**SHELL** : Esta es la utilidad que procesa las peticiones de los usuarios. Cuando alguien teclea un comando en la terminal, el shell interpreta el comando y llama el programa deseado. También es un lenguaje de programación de alto nivel que puede utilizarse en la combinación de programas de utilidad para crear aplicaciones completas.

El shell puede soportar múltiples usuarios, múltiples tareas y múltiples interfaces para sí mismo. Los dos shells más populares son el **BourneShell** (System V) y el **Cshell**

---

(BSD UNIX), debido a que usuarios diferentes pueden usar diferentes shells al mismo tiempo, entonces el sistema puede aparecer diferente para usuarios diferentes. Existe otro shell conocido como KornShell (así llamado en honor de su diseñador), que es muy popular entre los programadores.

**PROGRAMAS DE UTILIDAD (UTILERIAS)** : El Sistema Operativo UNIX incluye una gran variedad de programas de utilidad que pueden ser fácilmente adaptadas para realizar tareas específicas. Estas utilerías son flexibles, adaptables, portables y modulares, y pueden ser usadas junto con filtros y redireccionamientos para hacerlos más poderosos.

**SISTEMA MULTIUSUARIOS** : Dependiendo del equipo disponible, un UNIX puede soportar desde uno hasta más de 100 usuarios, ejecutando cada uno de ellos un conjunto diferente de programas.

**SISTEMA MULTITAREAS** : UNIX permite la realización de más de una tarea a la vez. Pueden ejecutarse varias tareas en su interior, mientras se presta toda la atención al programa desplegado en la terminal.

**ESTRUCTURA DE ARCHIVOS** : La estructura de archivos del UNIX está pensada para facilitar el registro de una gran cantidad de archivos. Utiliza una estructura jerárquica o de árbol que permite a cada usuario poseer un directorio principal con tantos subdirectorios como desee; UNIX también permite a los usuarios compartir archivos por medio de enlaces (*links*), que hacen aparecer los archivos en más de un directorio de usuario. Además, UNIX permite proteger los archivos del usuario contra el acceso por parte de otros usuarios.

**ENTRADA Y SALIDA INDEPENDIENTE DEL DISPOSITIVO** : Los dispositivos (como una impresora o una terminal) y los archivos en disco son considerados como archivos por UNIX. Cuando se da una instrucción al UNIX puede indicársele que envíe el resultado a cualquiera de los diversos dispositivos o archivos. Esta desviación recibe el nombre de *redireccionamiento* de la salida.

En forma similar, la entrada de un programa puede redireccionarse para que venga de un archivo en disco. En el UNIX, la entrada y la salida son independientes del dispositivo, pueden redireccionarse hacia o desde cualquier dispositivo apropiado.

COMUNICACIÓN ENTRE PROCESOS : UNIX permite el uso de conductos y filtros en la línea de comandos. Un **conducto** (*pipe*) redirige la salida de un programa para que se convierta en entrada de otro. Un **filtro** es un programa elaborado para procesar un flujo de datos de entrada y producir otro de datos de salida. Los conductos y filtros suelen usarse para unir utilerías y realizar alguna tarea específica.

#### 4 LENGUAJE SQL

SQL es un lenguaje estándar y universal que se utiliza para acceder a las bases de datos, nos da la posibilidad de modificar, borrar, crear y consultar los registros de una base de datos, crear estructuras nuevas para guardar datos, modificar las estructuras ya existentes entre otras.

SQL es un estándar adoptado por todos los fabricantes de software para realizar operaciones sobre una BD. Por lo tanto, las sentencias que se usan para consultar o modificar datos en SQL Server son las mismas que las que se usan con Oracle o con Informix. Con esto se consigue que no sea tan complicado cambiar de un lenguaje de programación de bases de datos a otro.

Otra de las ventajas que existen es que las instrucciones básicas que se necesitan para leer y modificar los datos de una base de datos son muy pocas, de hecho, se tiene solamente una instrucción para insertar datos, para modificar, borrar y leer registros. A estas cuatro instrucciones les acompañan otras para realizar otras operaciones auxiliares.

## 4.1 MANIPULANDO LA INFORMACION

Antes de empezar con la manipulación de los datos de una tabla, existen 2 tipos de operaciones que se pueden realizar en SQL:

- ▣ Lenguaje de Definición de Datos (DDL). El DDL (Data Definition Language), está formado por un conjunto de instrucciones que lo que hacen es definir y modificar los objetos que se tienen en la base de datos. Por ejemplo existen las instrucciones DDL para crear las tablas, otras para añadir campos a las tablas, modificar el tamaño de dichos campos, eliminar las tablas de la base de datos, etc. En definitiva, son instrucciones para definir la estructura de la base de datos.
- ▣ Lenguaje de Manipulación de Datos (DML). Con las instrucciones de este lenguaje lo que se hace es crear, modificar, consultar o borrar los datos. Solo se actúa sobre los datos, no se actúa nunca sobre la estructura ya que para eso están las instrucciones DDL. La siguiente tabla muestra las principales instrucciones de tipo DML:

| OPERACIÓN A REALIZAR         | INSTRUCCIÓN SQL |
|------------------------------|-----------------|
| Consultar datos de una tabla | Select          |
| Insertar datos en una tabla  | Insert          |
| Modificar datos de una tabla | Update          |
| Borrar filas de una tabla    | Delete          |

TABLA 3. Instrucciones DML

**Insert :** Con esta instrucción se introducen datos nuevos en una tabla.

La sintaxis básica de la instrucción insert es:

```
INSERT INTO tabla (campo01, campo02, campo03, ...)
VALUES (valor01, valor02, valor03, ...)
```

Se debe poner el nombre de la tabla en la que se quiere insertar y luego entre paréntesis la lista de los campos de dicha tabla. Luego en la opción de VALUES, se pone la



lista de valores que se van a guardar en cada campo y se corresponden en el mismo orden, es decir, en campo01 se guardará el valor valor01, en el campo02 el valor02, etc.\*

**Select** : Ahora que ya se tienen datos, estos se pueden consultar utilizando la instrucción **SELECT**.

La sintaxis básica de la instrucción **SELECT** es:

```
SELECT campo01, campo02, ...
FROM tabla;
```

Con esta instrucción lo que se obtiene son los registros (o filas) de la tabla "tabla" y por cada una de estas filas nos muestra el valor de los campos que le hemos indicado.

**Update** : Ahora que ya se cuenta con algunos datos, estos pueden también ser modificados.

La sintaxis básica de la instrucción **update** es:

```
UPDATE tabla
SET campo01 = valor01,
Campo02 = valor02,
Campo03 = valor03, ...
Campo21 = valor21
WHERE condicion;
```

Para indicar qué campos se quieren modificar se listan poniendo el nombre del campo y el nuevo valor que se quiere que vaya en él. Para indicar qué filas (o registros) se quieren modificar se puede añadir la cláusula **WHERE** con alguna condición.

**Delete**: Finalmente se pueden borrar filas (o registros). Para esto se usa el **delete** cuya sintaxis básica es:

```
DELETE FROM tabla WHERE condición;
```

En esta instrucción no hay la posibilidad de indicar qué campos se desean borrar ya que lo que hace es borrar filas enteras.\*

---

\* Algunos datos van entre comillas simples porque el tipo de dato que se introduce es un string, es decir, un grupo de caracteres

## 4.2 OPERADORES

Los operadores se necesitan para realizar comparaciones son: =, <>, !=, <=, >=, <, >, LIKE, los operadores lógicos: OR, AND, NOT y los operadores aritméticos +, -, \*, /.

Los operadores del mismo tipo tienen preferencia de izquierda a derecha.

**Operadores Lógicos:** Los operadores lógicos son AND, OR, NOT. Estos operadores comparan resultados lógicos de la siguiente manera:

| AND       |           | RESULTADO |
|-----------|-----------|-----------|
| VERDADERO | VERDADERO | VERDADERO |
| FALSO     | FALSO     | FALSO     |
| FALSO     | VERDADERO | FALSO     |

TABLA 4. La cláusula AND

Si comparamos dos condiciones con AND, el resultado solamente será verdadero si ambas condiciones son verdaderas, o dicho de otra forma, si comparamos dos condiciones con AND, si una de las dos es falsa ya no nos importa el resultado de la otra ya que el resultado será falso.

| OR        |           | RESULTADO |
|-----------|-----------|-----------|
| VERDADERO | VERDADERO | VERDADERO |
| FALSO     | VERDADERO | VERDADERO |
| FALSO     | FALSO     | FALSO     |

TABLA 5. La cláusula OR

\* Se debe poner atención a el número de registros afectados al final de la transacción ya que se suele cometer errores en la condición, para deshacer el cambio se utiliza rollback y para aceptarlo commit

Si comparamos dos condiciones con OR, el resultado solamente será falso si ambas condiciones son falsas, o dicho de otra forma, si comparamos dos condiciones con OR, con que una de ellas sea verdadera el resultado es verdadero.

## NOT

Devuelve el estado contrario de una condición. Es decir, NOT verdadero es falso y NOT falso es verdadero.

## Operadores Aritméticos

Los operadores aritméticos son suma (+), resta (-), multiplicación (\*) y división (/)

## Preferencia entre los operadores

La preferencia entre los operadores indica el orden en el que se van a evaluar las condiciones cuando están involucrados varios de los operadores. Los operadores están ordenados por Niveles de preferencia, y dentro de cada nivel, se evalúan primero los que están más a la izquierda dentro de la expresión. Los niveles son:

|                       |                             |
|-----------------------|-----------------------------|
| *, /                  | Multiplicación y División   |
| +, -,                 | Suma, Resta y Concatenación |
| =, <, <=, >, >=, Like | Comparativos                |
| Not                   |                             |
| And                   |                             |
| Or                    |                             |

TABLA 6. Preferencia entre los operadores

Así, si por ejemplo hacemos:  $2 + 4 + 3 * 5$ , como la multiplicación es lo que más peso tiene primero se haría la multiplicación y luego las sumas, por lo que el resultado será  $2 + 4 + 15 = 21$ .

**Operadores Avanzados:** Los operadores avanzados son: UNION, UNION ALL, INTERSEC, MINUS, BETWEEN.

- ☐ **Union:** Cuando se realiza una unión entre registros de dos tablas, el resultado que nos da es las filas de la primera más las filas de la segunda pero las filas que están repetidas solo salen una vez. Al hacer una unión de dos tablas, el número y tipo de los campos que se selecciona en la primera SELECT debe coincidir con el número y el tipo de los campos seleccionados en la segunda
  
- ☐ **Union All:** Cuando se realiza una union all entre registros de dos tablas, el resultado que nos da es las filas de la primera más las filas de la segunda y aparecen todas las filas repetidas que hagan falta.
  
- ☐ **Intersec:** Cuando se realiza una intersección entre registros de dos tablas, el resultado que nos da es las filas de la primera tabla que también se encuentran en la segunda .
  
- ☐ **Minus:** Cuando se realiza una operación minus entre registros de dos tablas, el resultado que nos da es las filas de la primera que no están en la segunda. Esta operación nos viene muy bien cuando debemos tener dos tablas iguales por algún motivo y queremos estar seguros de que realmente lo son.
  
- ☐ **Between:** Con este operador se compara si un dato está entre dos valores. Si el dato está entre los dos valores con los que se compara, ambos inclusive, nos devuelve verdadero, y si no lo está, nos devuelve falso\*. Por lo tanto, una comparación del tipo:
  - 7 between 2 and 10 es verdadero.
  - 7 between 1 and 6 es falso.
  - 7 between 7 and 8 es verdadero.

### 4.3 AGRUPACIONES Y ORDENACIONES

**Ordenaciones:** Cuando se aplica la orden SELECT de una tabla, no se sabe el orden en el que se van a mostrar los datos. No influye para nada el orden en el que se han ido introduciendo los datos en la tabla. Por lo general, los motores de bases de datos que usan SQL, como Oracle, no aseguran nunca el orden en el que aparecerán los datos al hacer una

select, existe una forma para conseguir que estos datos aparezcan ordenado por uno o varios campos y en orden ascendente o descendente de la siguiente manera:

```
ORDER BY campo01, campo02, campo03 ...
```

Por defecto, si no se indica nada especial ordenará ascendentemente primero por los valores del campo01, para todos los registros que tiene el mismo valor en campo01 ordenará ascendentemente por el valor de campo02 y así sucesivamente. Si se quiere ordene descendentemente por alguno de los campos se añade después de dicho campo la palabra reservada "DESC".

Aunque parezca que la ordenación no tenga ningún misterio existe un punto para tener en cuenta. Cuando se realiza una select y se ordena el resultado, el servidor de bases de datos, lleva todas las filas seleccionadas a una zona especial de la memoria en la que realiza la ordenación (tablespace TEMP). Si no tiene suficiente memoria, tendrá que utilizar el disco como caché para ordenar. Ordenar siempre resulta costoso al servidor.

**Agrupaciones:** Se utiliza la instrucción GROUP BY campo1, campo2, etc. pudiendo agregar una función sobre el campo, por medio de esta instrucción se obliga a que el resultado nos lo de agrupado por cierto campo.

**Cláusula having:** Hay ocasiones en que al hacer una agrupación interesa filtrar los datos devueltos dependiendo el valor que obtenga función aplicada a la agrupación (suma, media, máximo, etc). HAVING es un filtro de la agrupación. Primero hace la agrupación y luego nos muestra los registros que cumplen la condición del HAVING.

Es importante distinguir entre la cláusula HAVING y la cláusula WHERE. Con la cláusula WHERE, se filtran los registros que se quieren utilizar en la orden SELECT mientras que con la cláusula HAVING se filtran las agrupaciones que se han obtenido. Se puede utilizar cláusulas WHERE con cláusulas HAVING combinadas.

---

\* Hay que tener en cuenta al hacer un Between entre fechas que una fecha tiene horas, minutos y segundos.

**Inserts Masivas:** Anteriormente se explico cómo insertar registros nuevos en una tabla con la instrucción INSERT, sin embargo, existe la posibilidad de realizar INSERTS masivas en una tabla partiendo de una selección de datos previa de otra tabla (o de ella misma) con la siguiente sintaxis:

```
INSERT INTO tabla (campo01, campo02, campo03)
VALUES ( SELECT campo04, campo05, campo06 FROM tabla2);
```

Es obligatorio que se seleccione el mismo número de campos que los campos que se van a insertar, también es obligatorio que los tipos de los campos que se seleccionan y sus correspondientes de la tabla en la que se inserta sean iguales o compatibles, es decir, que si el campo04 de la tabla2 es carácter y se va a insertar en el campo01 de tabla y es numérico, nos dará un error.

**Joins:** Hay ocasiones en las que, dependiendo de la consulta, nos vemos obligados a obtener información que realmente está en dos (o más) tablas distintas. Cuando entre estas dos tablas existen uno o varios campos que las pueden relacionar, se puede hacer un join entre dichas tablas.

La sintaxis que se puede apreciar en el join es bastante sencilla. Por un lado se seleccionan los campos que nos interesan en la cláusula SELECT, luego, en la cláusula FROM se deben indicar las dos tablas a relacionar, y finalmente lo más importante, se deben igualar los campos de las dos tablas que nos sirven para relacionarlas.\*

**Outer Joins:** Son una clase especial de los joins. Cuando se hace join normal el proceso que realiza el motor de sql recorre uno a uno todos los registros de la primera tabla y luego los une con los registros de la segunda tabla que tengan el mismo código que los relaciona.

Se utiliza operador "(+)". Este operador se debe poner junto a la columna de la tabla del join que sabemos que puede que no tenga su equivalente en la otra tabla, por lo

---

\* Si se olvida igualar los campos a relacionar en las tablas el resultado será un producto cartesiano de ambas tablas.

tanto, se pueden registros de una tabla que no tengan asignado ningún registro de la tabla con la que se quiere relacionar así que en la columna que los relaciona de la tabla principal se pone el operador "+" para indicarle al motor que cuando se encuentre un caso de estos no desechar la fila, que debe mostrarla pero con los datos de departamento a NULL.

#### 4.4 TIPOS DE DATOS

**CHAR:** Las columnas de las tablas que se definen de tipo char contienen cadenas de caracteres de tamaño fijo. Por lo tanto, si decimos que el nombre del empleado es un char(30), en cada nuevo registro que se crea, se reserva un espacio de 30 bytes para guardar el nombre del empleado. Cuando el nombre que se va a guardar en este campo es de tamaño menor, se rellena con blancos a la derecha hasta ocupar los 30 bytes. Es un tipo de datos que no se utiliza demasiado, en su lugar se utiliza el varchar2

**VARCHAR2:** Las columnas de las tabla definidas varchar2 contienen cadenas de caracteres de tamaño variable. Por lo tanto, si decimos que el nombre del empleado es un varchar2(30), en cada nuevo registro que se crea, se reserva el espacio justo para guardar el nombre que estamos introduciendo. Si posteriormente se cambia el nombre del empleado por otro nombre mayor, el motor es el que encarga de buscar sitio para guardar los nuevos caracteres que no entraban en el espacio anterior. Este es el tipo de dato más común para las cadenas de caracteres.

Alguna de las funciones que se pueden utilizar para manipular cadenas de caracteres son:

*Concat*(cadena1, cadena2) se utiliza para concatenar dos strings

*Lpad* (cadena1, número, cadena2) se utiliza para cuando la cadena1 ocupa menos posiciones que el valor indicado por número, en cuyo caso se rellena por la izquierda con el valor de la cadena2 hasta ocupar ese número de caracteres. *Lpad*('hola',10,'as') = 'asasashola'

*Initcap* (cadena) devuelve la cadena pero con la primera letra de cada palabra en mayúsculas y el resto en minúscuas. *Initcap*('HOLA mundo') = 'Hola Mundo'

*Replace*(cadena, cadena a reemplazar, cadena3). Sustituye todas las apariciones de la cadena a reemplazar por la cadena3. *Replace*('Hola mundo soy el segundo','undo','as') = 'Hola mas soy el segas'

*Rpad* (cadena1, número, cadena2) es como el *lpad* pero los espacios que se rellenan son los que sobran por la derecha.

*Substr*(cadena1, número1, número2) devuelve los primeros número2 caracteres de la cadena1 que hay a partir del carácter número1 empezando desde la izquierda. Si no se especifica número2 se devuelven todos los caracteres hasta el final del string. Si número1 es negativo se empiezan a contar los caracteres desde el final del string hacia el principio.

También hay funciones que se aplican a los strings y que nos devuelven números:

*Instr*(cadena1,cadena2,n,m). Busca en la cadena1 la cadena2 a partir de la posición "n" de la cadena1 y devuelve l posición en la que se encuentra. Si además indicamos un "m" mayor que 1 nos busca la ocurrencia número "m" de la cadena2 en la cadena1. Por defecto "n" y "m" son 1 por lo que busca la primera aparición de la cadena2 en la cadena1 y devuelve esta posición. *Instr*('1z2y1a2b1z2y','y',1,1) = 4. *Instr*('1z2y1a2b1z2y','y',6,1) = 6. *Instr*('1z2y1a2b1z2y','2y',1,2) = 11

*Length*(cadena) nos devuelve el número de caracteres que forman la cadena.

**NUMBER:** Las columnas de las tablas que definimos de tipo number se utilizan para guardar números. La sintaxis que se usa para definir una variable o columna de tipo number es: number(p,s) donde 'p' es la precisión y 's' es la escala.

También pueden aparecer definiciones del tipo number(p) en la cual se supone que la escala es 0, o sea que es lo mismo que poner number(p,0).

La escala es el número de posiciones decimales que tiene nuestro valor mientras que la precisión menos la escala será la cantidad de posiciones enteras de nuestro valor, aunque no es del todo correcto.



Así, si por ejemplo tenemos el valor

| A representar | Definición    | Guardado           |
|---------------|---------------|--------------------|
| 12,345.89     | Number (9)    | 12,345             |
| 12,345.89     | Number (7,2)  | 12,345.89          |
| 12,345.89     | Number (8,1)  | 12,345.9           |
| 12,345.89     | Number (6,2)  | Error de Precisión |
| 12,345.89     | Number (7,-2) | 12,300             |

TABLA 7. Ejemplos de formatos numéricos

En esta tabla de ejemplos se pueden apreciar tres puntos importantes, en primer lugar, cuando el valor a representar tiene más decimales que los que permite su definición, éstos se redondean a la escala de la definición, pero no da ningún error, en segundo lugar, cuando el número de posiciones enteras (precisión – escala) no es suficiente para guardar el valor, el motor devuelve un error y por último, las escalas negativas sirven para redondear el número a la decena, centena, millar ...

Alguna funciones que se pueden utilizar para manipular los números son :

**Abs** nos devuelve el valor absoluto de un número

**Round** (número, decimales) nos redondea un número al número de decimales que le indiquemos.

**Trunc** (número, decimales) nos trunca un número al número de decimales que le indiquemos.

**Sign** nos devuelve -1 si el número es negativo, 0 cuando vale 0 y 1 cuando es mayor que 0.

**Mod** (número1, número2) nos devuelve el resto de dividir el número1 entre el número2 y si el número2

**S 0** nos devuelve el número1.

**DATE:** Las columnas de las tablas de tipo date se utilizan para guardar fechas. Cuando se utiliza este tipo de dato, en el valor guardado se tienen el siglo, el año, el mes, el día, la hora, el minuto y el segundo de la fecha guardada.

Las fechas se pueden manipular con operaciones aritméticas. Así pues, podemos obtener el día siguiente a una fecha sumándole uno.

La función *sysdate* nos da la fecha y la hora actual.

La función *last\_day(d)* nos da el último día del mes de la fecha que se le pasa, por lo que podemos saber por ejemplo los días que faltan para fin de mes con un *last\_day(sysdate) - sysdate*.

La función *next\_day(fecha, día)* nos devuelve la fecha del siguiente día a la fecha que le damos que es el día de la semana que le indicamos en "día". Por ejemplo si hoy es miércoles, el resultado de *next\_day(01/01/2000,'MONDAY')* será 6/01/2000.

La función *trunc(fecha)* nos devuelve la fecha pero sin las horas, minutos y segundos.

## PARTE II

### CASO PRACTICO

En esta parte se explica el proceso de instalación y puesta a punto de una base de datos bajo Oracle 8i en un servidor HP9000 para el ambiente de pruebas (Ofna0) en Seguros Tepeyac S.A.

---

## 1 DESCRIPCIÓN DEL PROBLEMA

El problema radica en que los programas y objetos creados para la base de datos real de la compañía (Producción) no pasan siempre como deberían de pasar debido a la falta de pruebas previas al traspaso, ocasionando que algunos procesos fallen en el ambiente de Producción y causando pérdida de tiempo en recuperar datos perdidos.

Se propone la creación de una base de datos para pruebas (Ofi0), en la cual los usuarios tendrán la oportunidad de verificar que los programas y objetos sean traspasados al ambiente de Producción de acuerdo a las necesidades. Esta nueva base tomará algunos datos de la base de Producción, pretendiendo que sea lo más parecida a la base real pero en una dimensión menor.

Se da la posibilidad de hacer esta instalación debido a que la empresa cuenta con equipo de características técnicas óptimas para la creación de esta base de datos, la cual no requiere algo muy poderoso debido al tamaño que se pretende que tenga. Este equipo es un servidor HP9000 cuyas características se mencionan más adelante en la parte de la instalación.

Se empleará Oracle 8i debido a que Oracle es el software utilizado para las bases de datos existentes en la empresa (ambiente de Desarrollo y Producción) aunado a que es un software perfectamente configurable y que se adapta a las necesidades de la empresa. El sistema de información que actualmente utiliza la empresa está desarrollado por programadores de la misma bajo las herramientas que Oracle proporciona, lo cual crea una relación casi perfecta entre los programas y la Base de Datos.

Se utilizará UNIX debido a que es el sistema operativo por defecto de HP9000 (HP-UX) y también porque con la experiencia de las bases de datos existentes se ha visto que UNIX y Oracle son completamente compatibles y juntos forman una herramienta poderosa y flexible a las necesidades.

---

A continuación se describe el proceso de creación de la base de datos de Ofi0. dividiéndolo de la siguiente manera:

1. Instalación del software Oracle
2. Creación de la Base de Datos
3. Configuración del Ambiente
4. Principales tareas de Administración de la base de datos.

## 2 INSTALACION DE ORACLE

### 2.1 CARACTERÍSTICAS DE HARDWARE Y SOFTWARE

Como se mencionó, la base de datos y el ambiente a crear debe ser igual en estructura tanto al ambiente de desarrollo como al de producción, con la diferencia en espacio ya que en desarrollo y Ofi0 no se requerirá insertar todos los datos de producción para realizar la pruebas.

Oracle requiere de un servidor potente, sobre todo debido a la cantidad de información y a la cantidad de usuarios que va a acceder a la base de datos de forma concurrente.

El servidor que se va a utilizar es ya parte de la empresa, al igual que los discos duros por lo que es algo que ya se tiene y que además se sabe que se trata de equipos que soportan la base de datos sin ningún problema pues son tecnológica y técnicamente equipos nuevos.

Las características son las siguientes:

HP 9000 MOD. K380  
Procesador a 550 MHZ  
Memoria RAM 2.0 GB  
Discos Duros SCSI 100GB  
Unidad DVD 6x/32  
Tarjeta de red HP-PB 100 Base TX card  
Unidad de cinta HP C1537A

## 2.2 CONFIGURANDO EL AMBIENTE

### 2.2.1 TAREAS A CONFIGURAR COMO USUARIO *root*

#### Configurando el Kernel de UNIX para oracle

Aquí se configuran los parámetros de IPC (Comunicación Entre Procesos) para acomodar la estructura de la SGA (Area Global Compartida) del Oracle Server. No se podrá arrancar la base de datos si el sistema no tiene la memoria compartida adecuada para la SGA.

- 1) Se verifica el estado de la memoria compartida.\*

```
IPC status from /dev/kmem as of Tue May 29 14:15:33 2001
T      ID      KEY          MODE          OWNER         GROUP
Message Queues:
q      0 0x3c1c0465  -Rrw--w--w-   root          root
q      1 0x3e1c0465  --rw-r--r--   root          root
Shared Memory:
m      0 0x2f100002  --rw-----   root          sys
m      1 0x411c0611  --rw-rw-rw    root          root
m      2 0x4e0c0002  --rw-rw-rw    root          root
m      3 0x41200353  --rw-rw-rw    root          root
Semaphores:
s      0 0x2f100002  --ra-ra-ra-   root          sys
s      1 0x411c0611  --ra-ra-ra-   root          root
s      2 0x4e0c0002  --ra-ra-ra-   root          root
s      3 0x41200353  --ra-ra-ra-   root          root
s      4 0x00446f6e  --ra-r--r--   root          root
s      5 0x00446f6d  --ra-r--r--   root          root
s      6 0x01090522  --ra-r--r--   root          root
```

- 2) Configurar los parámetros de Kernel correspondiente al:

- a) tamaño máximo de un segmento de memoria compartida (SHMMAX)
- b) tamaño mínimo de un segmento de memoria compartida (SHMMIN)
- c) máximo número de identificadores de memoria compartida en el sistema (SHMMNI)

\* Se utiliza el comando *ipcs* para obtener la lista de la memoria compartida actual del sistema obteniendo un número de identificación y su dueño.

- d) máximo número de segmentos de memoria compartida que un proceso de usuario puede utilizar (SHMSEG)

A continuación se muestran los valores que se configuraron al Kernel que son los parámetros que recomienda Oracle.

|        |     |
|--------|-----|
| SHMMAX | 1G  |
| SHMMIN | 1   |
| SHMMNI | 100 |
| SHMSEG | 10  |

Después de modificar los parámetros del Kernel es necesario reiniciar el sistema.

### Creación de los puntos de montaje

En la máquina se crearon los puntos de montaje en la siguiente ruta\*:

```
SOFTWARE → /oracle8i/
BASE      → /oracle8i/app/oracle/oracle/product/8.1.5i- (ORACLE_BASE)
```

### Creación del grupo para administración de la base

Para poder instalar Oracle es necesario la creación de un usuario de Oracle el cual es oracle8i, se da de alta en el archivo `/etc/passwd`.

Es también necesario la creación de un grupo dentro del archivo `/etc/group` llamado dba, grupo al que se asocia el usuario oracle8i y mi usuario con privilegios de root, *enrique*, los usuarios dados de alta dentro de este grupo deben ser exclusivamente usuarios para administrar las bases de datos.

---

\* Oracle requiere por lo menos dos puntos de montaje, uno para el software y por lo menos uno más para los archivos de la base de datos.

## Creación de un grupo para el OUI (Instalador Universal de Oracle)

Se crea un grupo llamado *oinstall*. Este grupo contendrá el usuario *orainventory*. Esta cuenta de usuario es la que corre el proceso de instalación y debe ser obligatoriamente miembro de este grupo.

## Creación de la cuenta UNIX que contendrá al usuario Oracle

La cuenta *oracle8i* es la cuenta UNIX dueña del software Oracle después de la instalación. A esta cuenta se le dieron los siguientes valores dentro de su archivo *.profile* en de su directorio HOME

```
ORACLE_BASE=/oracle8i/app/oracle/oracle      ;      export
ORACLE_HOME=$ORACLE_BASE/product/8.1.5i     ;      export
ORA_NLS33=/ocommon/nls/admin/data           ;      export ORA_NLS33
HOME=/oracle8i                               ;      export HOME
TERM=ansi                                     ;      export TERM
PATH=$PATH:$ORACLE_HOME/bin:/bin:/usr/bin:/usr/ccs/bin:$HOME/bin ;
export P
SHLIB_PATH=$ORACLE_HOME/lib:/usr/lib        : export SHLIB_PATH
ORACLE_SID=OF10                             ;      export ORACLE_SID
NLS_LANG=AMERICAN_AMERICA.WE8PC850         ;      export NLS_LANG
ORA_NLS33=$ORACLE_HOME/ocommon/nls/admin/data ;      export ORA_NLS33
PS1=TRON2000-'$PWD' -                       ;      export PS1
set -o vi
umask 022
.profile: END
```

La cuenta *oracle8i* solamente es utilizada para la administración de la base de datos del software de Oracle, root no debe ser utilizado como la cuenta Oracle.

### 2.2.2 TAREAS A CONFIGURAR CON EL USUARIO oracle8i

#### Permisos para la creación de archivos

Se teclea el comando *umask 022* de UNIX a la cuenta *oracle8i* para asegurarse que el grupo y los demás usuarios tengan permisos de ejecución, pero no permisos de escritura, en los archivos que OUI creará.



### 2.2.3 CONFIGURACION DE TAREAS PARA LOS PRODUCTOS ORACLE

#### Almacenamiento

Tipo de almacenamiento: Se utilizara raw devices\* para los ficheros de control, ficheros redo log y ficheros para espacios de tablas de la base de datos.

Tamaño de los archivos : Los archivos de la base de datos serán 10Mb mas pequeños que el tamaño del raw device. El tamaño del archivo de control es determinado por Oracle.

#### 2.2.3.1 Pasos a ejecutar como usuario root

1. Creación de raw devices: Se hace por medio del SAM (System Administrator) o por medio del comando de UNIX :

```
# lvcreate
```

Se crean inicialmente 10 raw devices para la base de datos bajo el directorio /dev/vg01/, utilizando como nombre del raw device 1 (rlvol001) al raw device 8 (rlvol008), distribuidos como se muestra en la siguiente Tabla.

| RAW DEVICE         | TAMAÑO |
|--------------------|--------|
| /dev/vg01/rlvol001 | 500MB  |
| /dev/vg01/rlvol002 | 500MB  |
| /dev/vg01/rlvol003 | 500MB  |
| /dev/vg01/rlvol004 | 800MB  |
| /dev/vg01/rlvol005 | 1 GB   |
| /dev/vg01/rlvol006 | 800MB  |
| /dev/vg01/rlvol007 | 2 GB   |
| /dev/vg01/rlvol008 | 1.5GB  |

TABLA 1. Espacios de los raw devices

\* Raw device, los espacios son creados de forma cruda, es decir, del espacio total (100 Gb) se va tomando el espacio conforme se necesite, no conforme al tamaño de los discos.

2. Se verifica que la red este funcionando correctamente por medio de la transferencia de un archivo de prueba (Con FTP\* a la dirección IP del servidor 10.0.2.112)

## 2.3 INSTALACION

### 2.3.1 SELECCIÓN DEL TIPO DE INSTALACION

Oracle tiene tres categorías de instalación: Enterprise Edition, Client y Oracle Programmer, cada una de ellas nos da tres tipos de instalación Típica, Mínima y Personalizada. Se debe de seleccionar la combinación de instalación adecuada según el propósito y los requerimientos. En la tabla 2 se muestran las diferentes herramientas que son instaladas con cada categoría.

| Productos                                      | Enterprise Edition | Client | Programmer |
|--|--------------------|--------|------------|
| Oracle 8i Server                               | X                  |        |            |
| Oracle 8i Client                               |                    | X      |            |
| Net8 Assistant                                 | X                  | X      | X          |
| Oracle Call Interface                          | X                  | X      | X          |
| Oracle Advanced Security Option-Export Edition | X                  | X      |            |
| Oracle Object Type Translator                  | X                  | X      | X          |
| Pro * C  |                    |        | X          |
| Pro * Cobol                                    |                    |        | X          |
| Pro * Fortran                                  |                    |        | X          |
| OUI  | X                  | X      | X          |
| SQL * Plus                                     | X                  | X      |            |
| Utilerias                                      | X                  | X      |            |
| Net8 Server                                    | X                  |        |            |
| Net8 Cliente                                   | X                  | X      | X          |
| Administrador de Conexiones                    | X                  |        |            |
| Agente Inteligente                             | X                  |        |            |
| Oracle Names                                   | X                  |        |            |
| Parallel Server                                | X                  |        |            |
| Asistente para configurar la base de datos     | X                  |        |            |
| Asistente para migración de datos              | X                  |        |            |
| Administrador de almacenamiento                | X                  |        |            |

\* El File Transfer Protocol (FTP) permite la transferencia de archivos entre maquinas con igual o diferente plataforma.

|                   |   |   |   |
|-------------------|---|---|---|
| Oracle Audio Data | X |   |   |
| Documentación HP  | X | X |   |
| SQLJ Traductor    | X |   |   |
| SQLJ Runtime      | X |   |   |
| JDBC/OCI          | X | X | X |
| JDBC/THIN         | X |   |   |
| SQLJ              | X | X |   |
| EJB/CORBA         | X | X |   |
|                   |   |   |   |

TABLA 2. Instalador Universal de Oracle: Categorías de Instalación del Producto

En este caso debido a que lo que se desea instalar es el software Oracle para posteriormente hacer la instalación de una base de datos en el servidor entonces se realizará la instalación Enterprise Edition de tipo personalizada (custom), seleccionando todas las herramientas que se tienen disponibles.

### 2.3.2 INICIANDO EL INSTALADOR UNIVERSAL DE ORACLE (OUI)

#### Montaje del CD-ROM Oracle8i

El CD-ROM de Oracle está en formato ISO 9660-RockRidge\*, debe de ser montado manualmente con el usuario *root*.

- 1) Se inserta por medio del editor vi el la siguiente línea en el archivo */etc/pfs\_fstab*

```
/dev/dsk/c4t2d0 /SD_CDROM pfs-rrip xlat=unix 0
```

lo cual indica lo siguiente

```
/dev/dsk/c42t0      -> Archivo del dispositivo del CD-ROM
/SD_CDROM          -> Punto de montaje
pfs-rrip xlat=unix 0 -> Formato ISO9660 con extensión RockRidge
```

- 2) Se ejecuta el archivo *pfs\_mountd* con el siguiente comando

```
# nohup /usr/sbin/pfs_mountd &
```

- 3) Se ejecuta también el siguiente comando

```
# nohup /usr/sbin/pfsd &
```

\* ISO 9660 es un sistema de archivos estándar que permite leer el mismo CD ROM en diferentes plataformas.

- 4) Se ejecuta el siguiente archivo para montar el CD-ROM

```
# /usr/sbin/pfs_mount/SD_CDROM
```

- 5) Se sale de la cuenta del root

```
# exit
```

- 6) Se cambia de directorio a /SD\_CDROM, que es el aparente sistema de archivos en donde se monto el CD-ROM

```
# cd /SD_CDROM
```

### 2.3.3 EJECUTANDO EL OUI

Ya montado el CD-ROM se inicia el OUI

- 1) Accesar como usuario oracle8i en una máquina que permita xwindows, es decir, un entorno UNIX Gráfico (máquina con IP10.0.2.124) ya que el OUI de Oracle 8i incluye esa posibilidad haciendo la instalación más amigable:

```
# su - oracle8i
```

- 2) Ir al directorio /SD\_CDROM

```
# cd /SD_CDROM
```

Ejecutamos el comando siguiente para verificar el contenido

```
# ll
```

Obteniendo:

```
TRON2000-/SD_CDROM-ll
total 34
d----- 2 root      root      18432 Mar 11  1999 RR_MOVED
drwxr-xr-x 2 25882    users     2048 Feb 25  1999 Translations
drwxr-xr-x 2 25882    users     4096 Mar  8  1999 doc
-rw-r--r-- 1 25882    users     3622 Mar  9  1999 index.htm
drwxr-xr-x 2 25882    users     2048 Mar 11  1999 install
-rwxr-xr-x 1 25882    users       468 Mar 11  1999 runInstaller
drwxr-xr-x 2 25882    users     2048 Mar 11  1999 stage
TRON2000-/SD_CDROM-
```

3) Se ejecuta el OUI con el comando siguiente respetando minúsculas y mayúsculas:

```
# ./runInstaller
```

NOTA: Si manda *error DISPLAY not set. Please set the DISPLAY and try again.*

: *Read-only file system*

Se ejecuta el comando con el que se exporta el ambiente gráfico y la instalación para ser realizada de forma remota desde la máquina con IP 10.0.2.124

```
# export DISPLAY = 10.0.2.124 : 0.0
```

### 2.3.4 INSTALACION DE LOS PRODUCTOS

- 1) En la pantalla de bienvenida se da click en el botón siguiente para iniciar la instalación
- 2) Se da el valor del ORACLE\_HOME en el destino y el punto de montaje del CD en el Origen
- 3) En la ventana de Productos Disponibles se selecciona **Enterprise Edition** como categoría y **Custom** (personalizada) como el tipo de la instalación
- 4) Se seleccionan todas las opciones haciendo una instalación completa, esto ocupa aproximadamente 1 GB de espacio, al darle continuar manda una barra de estado del 0 al 100%.
- 5) Después de que OUI terminó el 100% de la instalación de archivos se da la opción finalizar volviendo a la línea de comandos UNIX y se ejecuta el shell *root.sh*, este shell que es creado por mismo OUI en el directorio de ORACLE\_HOME, para ejecutarlo se accesa al sistema como usuario *root*. Por medio de este shell se dan los permisos pertinentes a los productos Oracle.

```
# cd $ORACLE_HOME
```

```
# ./root.sh
```

Una vez ejecutados estos pasos nos indica que la instalación es exitosa y esta listo el software Oracle para realizar la instalación de la base de datos, instalación que se explica a continuación.

### 3 INSTALACION DE LA BASE DE DATOS

#### 3.1 DEFINICION DE ESPACIOS

Se debe primero definir como se va a dividir la información en la base de datos. En toda base de datos Oracle existe por lo menos un tablespace llamado SYSTEM que contendrá el diccionario de datos.

En la base de datos Ofi0 configuran los siguientes tablespaces

| TABLESPACE   | RAW DEVICE         | TAMAÑO  | DESCRIPCION                                      |
|--------------|--------------------|---------|--|
| SYSTEM       | /dev/vg01/rlvol001 | 490 MB  | CONTENDRA EL DICCIONARIO DE DATOS                |
| TEMP         | /dev/vg01/rlvol002 | 490 MB  | TABLESPACE POR DEFAULT PARA SEGMENTOS TEMPORALES |
| ROLLBACKS    | /dev/vg01/rlvol003 | 490 MB  | ESPACIO PARA SEGMENTOS ROLLBACK                  |
| USERS        | /dev/vg01/rlvol004 | 790 MB  | ESPACIO PARA SEGMENTOS DE LOS DESARROLLADORES    |
| CATALOGOS    | /dev/vg01/rlvol005 | 1014 MB | CONTIENE SEGMENTOS PARA TABLAS DE CATALOGOS      |
| INDCATALOGOS | /dev/vg01/rlvol006 | 790 MB  | CONTIENE LOS DATOS DE INDICES PARA CATALOGOS     |
| TAEMISION    | /dev/vg01/rlvol007 | 2038 MB | CONTIENE DATOS DE TABLAS PARA EMISION            |
| INDEMISION   | /dev/vg01/rlvol008 | 1526 MB | PARA DATOS DE INDICES DE TABLAS DE EMISION       |

TABLA3. Distribución de los tablespaces

Como se puede observar se utilizarán al principio aproximadamente 7.5 GB para TABLESPACES, recordando que después se crearán los demás tablespaces para tablas que no son de emisión.

## 3.2 CREANDO SCRIPTS

### 3.2.1 ARCHIVO DE INICIALIZACION

Después de instalar el software de Oracle 8i, Oracle crea automáticamente el siguiente archivo:

```
/oracle8i/app/oracle/oracle/product/8.1.5i/dba/init.ora
```

El archivo init.ora es un ejemplo con los valores que se deben configurar para levantar la instancia, este archivo se toma como base para la creación de la instancia de Ofi0. El archivo después de modificarlo queda de la siguiente forma:

```
db_name                =      OFI0          #NOMBRE DE LA BASE DE DATOS

#####
#      AQUÍ SE CONFIGURAN LOS DESTINOS DE LOS ARCHIVOS DE CONTROL      #
#      ASI COMO DE LOS DIRECTORIOS DUMP (PARA ERRORES DE ORACLE,      #
#      DE BASE DE DATOS O DE USUARIOS)                                #
#####

control_files          =      (
    /oracle8i/admin/OFI0/controlf/control01.ctl,
    /oracle8i/admin/OFI0/controlf/control02.ctl,
    /oracle8i/admin/OFI0/controlf/control03.ctl)

background_dump_dest   = /oracle8i/admin/OFI0/bdump
core_dump_dest         = /oracle8i/admin/OFI0/cdump
user_dump_dest         = /oracle8i/admin/OFI0/udump
db_block_size          = 8192          # TAMAÑO DEL BLOQUE

#####
#      AQUÍ SE CONFIGURA EL ORDEN EN QUE LOS SEGMENTOS ROLLBACK      #
#      SERAN ACCESADOS POR LA BASE DE DATOS                            #
#####

rollback_segments      = ()#(R01, R02, R03, R04)
db_files                = 90          #NUMERO MAXIMO DE DATAFILES
db_file_multiblock_read_count= 32
db_block_buffers       = 8000 #TAMAÑO DE CACHE DE BUFFER
shared_pool_size       = 87000000    #TAMAÑO DEL SGA
log_checkpoint_interval = 10000
processes              = 500 #NUMERO DE PROCESOS SIMULTANEOS
dml_locks              = 2500 #NUMERO MAX BLOQUEOS DE TABLAS SIM
log_buffer             = 256000 #TAMAÑO DEL BUFFER REG ACTIVIDAD
sequence_cache_entries = 500
max_dump_file_size     = 10240 #TAMAÑO MAX DEL ARCHIVO DUMP
```

```
# global_names= TRUE
checkpoint_process          = TRUE
sort_area_size             = 181072#MEMORIA PARA ORDENAMIENTO X USER
transactions                = 200
v733_plans_enabled         = TRUE
utl_file_dir               = *
_db_block_cache_protect    = TRUE
distributed_transaction     = 4
distributed_lock_timeou    = 2
```

Después de crear el archivo `initOFI0.ora` es necesario la creación de la base de datos. Los siguientes scripts se corren por medio del Server Manager entrando con el siguiente comando:

```
# svrmgrl
```

se hace la conexión a la base de datos con el usuario 'internal', usuario que se crea automáticamente al instalar el software de oracle.

```
# connect internal
```

Ya conectados a la base de datos es entonces deben se corren un conjunto de scripts creados especialmente para la creación de la base de datos de oficina cero.

### 3.2.2 CODIGOS

A continuación se muestran los *scripts* de creación de una BD llamada OFI0 en el orden en el que son ejecutados desde el Server Manager:

```
crdbOFI0.sql
```

*Script* de creación de la BD llamada OFI0.

```
catalog.sql
```

Crea el diccionario de datos, este script lo crea automático Oracle al instalar el software.

```
crea_tablespaces.sql
```

Crea los 7 tablespaces restantes (excepto el SYTEM que se crea en al crear la base)



```
crea_rollbacks.sql
```

Crea los 4 rollbacks que serán usados para la base de datos

```
CrdbOF10.sql
```

```
REM * script de creación para bd OF10
REM * archivo: crdbOF10.sql

set termout on
set echo on
spool crdbOF10.log

REM * creación inicial de la BD
REM * crear la BD, con los ficheros de control
REM * especificados en el fichero initOF10.ora
REM * ORACLE_SID (nombre de la base de datos) debe ser igual a OF10
REM *
REM * arrancar la instancia de BDOF10 con el fichero initOF10.ora
REM *

connect internal
startup pfile=$ORACLE_HOME/dbs/initOF10.ora nomount /*Lee el archivo de
inicialización initOF10.ora especificando que la base no ha sido
montada.*/

REM * crea la BD OF10

create database OF10
controlfile reuse # continuan usando los archivos de control
maxinstances 8
maxlogfiles 32 # máximo número de archivos de log
character set WE8PC850 # lenguaje
datafile
    '/dev/vg01/rlvol001' size 490M # Tablespace System
logfile
    '/dev/vg01/rlvoljtrnredo01' size 18M,
    '/dev/vg01/rlvoljtrnredo02' size 18M

disconnect

spool off

REM * la BD deberá estar arrancada en este momento.

catalog.sql

REM * crear el diccionario
REM * el fichero catalog.sql viene con la instalación del software
Oracle.

@ $ORACLE_HOME/rdbms/admin/catalog.sql
```

```
crea_rollback.sql
```

```
REM * Crear segmentos rollback
REM * crear 4 segmentos rollback en el tablespace RBS
REM * 1 segmento rollback por cada 4 acciones concurrentes.
REM * No mas de 50 segmentos rollback, tamaño óptimo 20M.
REM * Todos los segmentos rollback del mismo tamaño.
REM * Las extensiones posibles de igual tamaño por segmento rb

connect internal

CREATE ROLLBACK SEGMENT R01 STORAGE (INITIAL 5M NEXT 5M OPTIMAL 20M)
TABLESPACE ROLLBACKS;
CREATE ROLLBACK SEGMENT R02 STORAGE (INITIAL 5M NEXT 5M OPTIMAL 20M)
TABLESPACE ROLLBACKS;
CREATE ROLLBACK SEGMENT R03 STORAGE (INITIAL 5M NEXT 5M OPTIMAL 20M)
TABLESPACE ROLLBACKS;
CREATE ROLLBACK SEGMENT R04 STORAGE (INITIAL 5M NEXT 5M OPTIMAL 20M)
TABLESPACE ROLLBACKS;
ALTER ROLLBACK SEGMENT R01 ONLINE;
ALTER ROLLBACK SEGMENT R02 ONLINE;
ALTER ROLLBACK SEGMENT R03 ONLINE;
ALTER ROLLBACK SEGMENT R04 ONLINE;

REM * Modificar los usuarios SYS y SYSTEM.
REM * designar tablespace temporal a TEMP
REM * designar tablespace por defecto para todos los usuarios USERS

alter user sys temporary tablespace TEMP;
alter user system default tablespace USERS temporary tablespace TEMP;

REM * Para cada usuario DBA, ejecutar el script de
REM * creacion de sinonimos. No olvidar ejecutarlo para cada
REM * usuario DBA creado en el futuro.

connect system/manager
@ $ORACLE_HOME/rdbms/admin/catdbsyn.sql
```

Después de correr estos scripts la base de datos esta lista para ser utilizada para crear las tablas que se utilizarán en el sistema para pruebas, es decir, las tablas de emisión y catálogos en sus tablespaces correspondientes.

### 3.3 MODIFICACIONES FINALES

Por último es modificado el siguiente archivo:

```
/oracle8i/app/oracle/oracle/product/8.1.5i/network/admin/tnsnames.ora
```

Este archivo contiene información necesaria para poder acceder desde una máquina a otras bases de datos de forma remota( bases distribuidas), se configura el nombre de la

base de datos, el protocolo, la IP y el puerto por medio del cual se accederá , quedando de la siguiente forma:

```
#
# Installation Generated NetV2 Configuration by Enrique Moreno
# archivo: Tnsnames.ora
#
#<oracle_sid> =
# (DESCRIPTION =
# (ADDRESS = (PROTOCOL= TCP)(Host= contab)(Port= 1521))
# (CONNECT_DATA = (SID = <oracle_sid>))
# )

PRD =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS =
        (PROTOCOL = tcp)
        (HOST = 10.0.2.102)
        (PORT = 1521)
      )
    )
    (CONNECT_DATA =
      (SID = B)
    )
  )
DESA =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS =
        (PROTOCOL = tcp)
        (HOST = 10.0.2.104)
        (PORT = 1521)
      )
    )
    (CONNECT_DATA =
      (SID = DESA)
    )
  )
OFIO =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS =
        (PROTOCOL = tcp)
        (HOST = 10.0.2.108)
        (PORT = 1521)
      )
    )
    (CONNECT_DATA =
      (SID = OFIO)
    )
  )
```

En donde:

```

OFIO =          *** Nombre del servicio
(DESCRIPTION = *** No necesario, una pequeña
                descripción de la base
  (ADDRESS_LIST =
    (ADDRESS =
      (PROTOCOL = tcp) *** Protocolo para conectarse
      (HOST = 10.0.2.108) *** Dirección de la máquina en
                          la que esta la base de datos
      (PORT = 1521) *** Puerto para conexión
    )
  )
(CONNECT_DATA =
  (SID = OFIO) *** Nombre de la base de datos
)
)

```

En este archivo encontramos parámetros de configuración para poder hacer ligas hacia diferentes bases de datos, son varias las bases de datos a las que se requiere tener conexión, el nombre de las principales bases de datos con las que tendremos conexión serán:

- B → Base de datos de Producción
- OFIO → Base de datos de Oficina Cero
- DESA → Base de datos de Desarrollo

Este archivo es necesario para poder hacer la creación de ligas entre bases de datos (database links\*).

Una vez realizada la configuración de este archivo la base de datos de oficina cero esta lista para ser utilizada y para poder crear las tablas y objetos que serán necesarios para emisión.

---

\* Las ligas se utilizan para poder intercambiar información entre objetos en diferentes bases de datos.

## 4 AMBIENTE DEL USUARIO

Después de configurar la base de datos y estando en línea, entonces se crean las tablas que serán usadas para el módulo de Emisión.

En los dos ambientes que ahora mismo existen se tiene creado un usuario Oracle 'ops\$espania' que es el dueño de todas las tablas, índices y demás objetos que se utilizan de forma pública actualmente en Producción y Desarrollo, este usuario deberá ser también el dueño de las tablas que se crearán en Ofi0.

En la empresa todo el sistema llamado TRON esta desarrollado por programadores pertenecientes al grupo MAPFRE, grupo de aseguradoras europeo al que pertenece la compañía, TRON fue creado con las herramientas que Oracle proporciona para este fin, esta es una de las razones principales de la creación de la base de datos de ofi0, ya que al ser el sistema desarrollado por el grupo frecuentemente es necesario modificarlo con la finalidad de mejorarlo y adaptarlo a Tepeyac, ya sea simplemente por la creación de un índice nuevo, para mejorar velocidad en las búsquedas o la creación y modificación de tablas debido a la modificación o creación de programas nuevos.

La base de datos que se está creando se pretende que sea lo mas parecida posible a la base de datos de Producción simplemente más pequeña, dentro de la base de datos de Producción contamos con alrededor de 4000 tablas como se muestra a continuación:

```
SQL> select count(*) from dba_tables
2 /

COUNT(*)
-----
4040
```

Estas tablas ocupan un espacio aproximado de 200 GB mostrado a continuación

```
SQL> select sum(bytes)/1073741824 GB from dba_segments
2 where segment_type = 'TABLE'
3 /

GB
-----
196.62
```

Debido a que sería muy difícil la recreación de todas las tablas al mismo tiempo, se crearán en la nueva instancia de inicio solamente las tablas del módulo de Emisión de Pólizas ya que estas son las más utilizadas en los programas además de que son las que frecuentemente necesitan modificaciones para los programas respectivos, aunque hay que tomar en cuenta que todas las tablas serán creadas en su momento.

Para cada módulo se tienen las tablas que le pertenecen, en el caso de Emisión con sus catálogos son alrededor de 150 tablas de las cuales ya se conoce su nombre y estructura de cada una de ellas. La forma en que se realiza la recreación es la siguiente

- 1) Creación de usuarios y perfiles
- 2) Obtención de estructura de los datos de las tablas desde ambiente producción
- 3) Obtención de los índices de las tablas, con estructura de producción
- 4) Creación de los Scripts necesarios para recreación de las tablas
- 5) Creación de Script para creación de índices
- 6) Creación por medio de un Shell de UNIX las tablas y los índices.
- 7) Premisos a los usuarios de pruebas sobre las tablas

#### 4.1 CREACION DE USUARIOS Y PERFILES

Los primeros usuarios que se crean son los de los administradores de la base de datos con atributos dba, ya con estos usuarios este funcionando correctamente se crea el rol\_pruebas y el rol\_desarrollo\*

El rol\_desarrollo tiene los siguientes permisos:

```
PRIVILEGE
-----
CREATE SESSION
CREATE TABLE
CREATE PROCEDURE
EXECUTE ANY PROCEDURE
SELECT ANY TABLE
CONNECT
```

\* Un Rol es un conjunto de permisos que al ser asignado a un usuario, el usuario obtiene todos los permisos que pertenezcan al Rol.

El rol `rol_pruebas` tiene los siguientes permisos:

```
PRIVILEGE
-----
CREATE SESSION
CREATE TABLE
CREATE PROCEDURE
EXECUTE ANY PROCEDURE
SELECT ANY TABLE
DELETE ANY TABLE
UPDATE ANY TABLE
INSERT ANY TABLE
```

Como se puede observar la diferencia entre el rol `rol_desarrollo` y el rol `rol_pruebas` es que el rol `rol_pruebas` ya puede hacer inserción, borrado y actualización en cualquier tabla.

El rol `rol_desarrollo` tiene todos los permisos en la instancia de desarrollo pues es el rol asignado a todos los desarrolladores, en Ofi0 el rol asignado a los usuarios designados para las pruebas es el rol `rol_pruebas`.

La forma como se crea un role y un usuario la podemos ver más a detalle más adelante en la parte de Tareas de Administración de la Base de Datos.

Ya creados los roles entonces es necesario la creación de un usuario para la creación de las tablas, éste es el mencionado `ops$espania` con la siguiente sintaxis

```
SQL> create user ops$espania identified by *****
  2 temporary tablespace temp
  3 default tablespace users
```

```
SQL> Grant rol_pruebas to ops$espania
```

Después se le otorga una cuota<sup>1</sup> al usuario `espania` con el commando

```
Alter user ops$espania
Quota unlimited on taemision, indemision, catalogos, indcatalogos
```

Además de estos dos roles al usuario `ops$espania` se le dan permisos de

```
DROP TABLE
CREATE INDEX
DROP INDEX
DROP INDEX
```

<sup>1</sup> Quota: espacio asignado a un usuario en determinado espacio de tablas.

#### 4.2 OBTENCION DE LA ESTRUCTURA DE DATOS DE LAS TABLAS DESDE AMBIENTE DE PRODUCCION

Existe una tabla del diccionario de datos llamada dba\_segments con la cual se puede obtener la información que necesitamos para la creación. La información es obtenida de la base de datos de producción con la siguiente consulta:

```
SQL> SELECT SEGMENT_NAME, BYTES, OWNER
2 FROM DBA_SEGMENTS WHERE SEGMENT_TYPE='TABLE'
3 AND SEGMENT_NAME IN
4 ('A2000020','A2000160','A2000162','A2500271','A2040200','A2000280','A
2500360','A2600005','A2600006','A2990900','A2600002','A2000265','A200
0060','A2990131',
.....)
```

Así se obtiene el nombre de la cada tabla de emisión, el tamaño y el dueño de la misma, dando la siguiente información pero de todas las tablas:

| SEGMENT_NAME | BYTES      | OWNER       |
|--------------|------------|-------------|
| A2990132     | 3440640    | OPS\$ESPAÑA |
| A2600002     | 16384      | OPS\$ESPAÑA |
| A2600005     | 16384      | OPS\$ESPAÑA |
| A2600006     | 24576      | OPS\$ESPAÑA |
| A2000020     | 5845983232 | OPS\$ESPAÑA |
| A2301302     | 11067392   | OPS\$ESPAÑA |
| A1001303     | 1171456    | OPS\$ESPAÑA |
| A2301303     | 516096     | OPS\$ESPAÑA |
| A2000265     | 14548992   | OPS\$ESPAÑA |
| A2000160     | 16384      | OPS\$ESPAÑA |
| A2990900     | 16384      | OPS\$ESPAÑA |
| A2000250     | 629145600  | OPS\$ESPAÑA |

TABLA4. Datos de las tablas

Los nombres de las tablas ya son conocidos y fueron proporcionados por los desarrolladores de la compañía.

#### 4.3 OBTENCION DE LOS INDICES DE LAS TABLAS

Para obtener los datos de los índices de las tablas que se crearán es necesario correr el siguiente join entre las tablas dba\_indexes y dba\_segments



```

SQL> select b.index_name,
2  a.table_owner, a.table_name,
3  a.column_name, a.column_position ,c.bytes,
4  b.uniqueness
5  from dba_ind_columns a, dba_indexes b, dba_segments c
6  where b.index_name=a.index_name
7  and b.table_name=a.table_name
8  and b.table_owner=a.table_owner
9  and b.owner=a.table_owner
10 and c.owner=a.table_owner
11 and c.owner=a.table_owner
12 and c.segment_name=b.index_name
13 and c.segment_name=a.index_name
14 and b.table_name
15 in('A2000020', 'A2000160', 'A2000162', 'A2500271', 'A2040200', 'A2000280',
16  'A2500360',
17  'A2600005', 'A2600006', 'A2990900', 'A2600002', 'A2000265', 'A2000060', 'A2
18  990131',
19  'A2990132', 'A2301303', 'A2200010', 'A2200022', 'A2200021', 'A2302208', 'A2
20  200020',
21  ..... )

```

Cómo se puede ver, la información que aquí se obtiene es el nombre del índice, la tabla a la que pertenece, las columnas que ocupa con su posición dentro del índice, el tipo de índice y el dueño dando como resultado la siguiente tabla:

| INDEX_NAME  | TABLE_OWNER | TABLE_NAME | COLUMN_NAME | COLUMN POSITION | BYTES   | UNIQUENES |
|-------------|-------------|------------|-------------|-----------------|---------|-----------|
| II_A2600002 | OPS\$ESPAÑA | A2600002   | NUM_POLI    | 1               | 3153920 | NONUNIQUE |
| II_A2600002 | OPS\$ESPAÑA | A2600002   | NUM_END     | 2               | 3153920 | NONUNIQUE |
| II_A2600002 | OPS\$ESPAÑA | A2600002   | NUM_SECU    | 3               | 3153920 | NONUNIQUE |
| II_A2600002 | OPS\$ESPAÑA | A2600002   | COD_CIA     | 4               | 3153920 | NONUNIQUE |
| II_A2600005 | OPS\$ESPAÑA | A2600005   | NUM_POLI    | 1               | 3153920 | NONUNIQUE |
| II_A2600005 | OPS\$ESPAÑA | A2600005   | NUM_END     | 2               | 3153920 | NONUNIQUE |
| II_A2600005 | OPS\$ESPAÑA | A2600005   | LINEA_TXT   | 3               | 3153920 | NONUNIQUE |
| II_A2600005 | OPS\$ESPAÑA | A2600005   | COD_CIA     | 4               | 3153920 | NONUNIQUE |
| II_A2600006 | OPS\$ESPAÑA | A2600006   | NUM_POLI    | 1               | 122880  | NONUNIQUE |
| II_A2600006 | OPS\$ESPAÑA | A2600006   | NUM_END     | 2               | 122880  | NONUNIQUE |
| II_A2600006 | OPS\$ESPAÑA | A2600006   | COD_CIA     | 3               | 122880  | NONUNIQUE |
| II_A2990132 | OPS\$ESPAÑA | A2990132   | COD_CIA     | 1               | 1482752 | NONUNIQUE |
| II_A2990132 | OPS\$ESPAÑA | A2990132   | NUM_POLI    | 2               | 1482752 | NONUNIQUE |
| II_A2990132 | OPS\$ESPAÑA | A2990132   | NUM_END     | 3               | 1482752 | NONUNIQUE |

TABLA5. Información de los índices

#### 4.4 CREACION DE LOS SCRIPTS PARA TABLAS

Ya que se tiene toda la información necesaria para la creación de las tablas entonces se pueden crear los scripts necesarios, un script creará las tablas otro los índices.

Dentro de la creación de las tablas se debe de poner el nombre de la tabla, el storage (valores de inicio) y la copia de la estructura de la tabla de producción como sigue:

```
Create table A2990132
Storage (Initial 1M next 1M pctincrease 0 minextents 1)
Pctfree 10 pctused 90
Tablespace TAEMISION
As select * from A2990132@hpt520 -Es la liga hacia la base de producción
Where 1=2
```

Como se puede observar en el ejemplo se crea la tabla A2990132, dentro de sus valores de Storage tenemos un valor **initial** que es el tamaño inicial para la tabla, al consultar en producción el tamaño de esta se obtuvo e un tamaño de 3M, como se comentó, esta base de datos será igual a la de producción pero más pequeña por lo que se le da un valor inicial más pequeño que en producción.

El siguiente valor, **next**, se dio un valor igual al inicial, este valor nos indica que si la tabla necesita crecer reservará otro 1M para el siguiente crecimiento.

El **pctincrease 0** es el porcentaje de incremento del valor **next**, es decir que si es necesario un tercer crecimiento este será del mismo tamaño que el anterior.

El valor **minextents** sirve para decir que esta tabla por lo menos crecerá una vez.

Los siguientes valores **pctfree** y **pctused** indican el valor reservado para operaciones de inserción y modificación de registros y el porcentaje que se puede utilizar con registros respectivamente.

La orden `as select`, hace una selección por medio de una liga `a2990132@prod` a la misma tabla pero de producción, esto es para que tenga la misma estructura, condicionando que `l=2`, es decir que copie la estructura pero no los datos.

El nombre del script para la creación de las tablas es 'tablas.sql'.

#### 4.5 CREACION DE LOS SCRIPTS PARA INDICES

Al igual que el script para tablas, debemos de crear el script para los índices de las tablas con la información obtenida anteriormente, el script se forma de la siguiente forma:

```

Create index I1_A2600002 on a2600002
(num_poll, num_end, num_secu, cod_cia)
storage(initial 1M next 1M pctincrease 0)
pctfree 5
tablespace INDEMISSION

```

Este índice se crea basándonos en la información siguiente previamente obtenida.

| Índice      | Dueño       | Tabla   | Columna | Orde | Tamaño  | Tipo      |
|-------------|-------------|---------|---------|------|---------|-----------|
| I1_A2600002 | OPS\$ESPANI | A260000 | NUM_POL | 1    | 3153920 | NONUNIQUE |
| I1_A2600002 | OPS\$ESPANI | A260000 | NUM_END | 2    | 3153920 | NONUNIQUE |
| I1_A2600002 | OPS\$ESPANI | A260000 | NUM_SEC | 3    | 3153920 | NONUNIQUE |
| I1_A2600002 | OPS\$ESPANI | A260000 | COD_CIA | 4    | 3153920 | NONUNIQUE |

La sintaxis contiene al comando `create index` más el nombre del índice `on` nombre de la tabla en la que se va a crear.

Entre paréntesis se ponen las columnas que se incluirán en el índice tomando el orden con el que aparecen en la columna `column_position` de la información obtenida.

El `storage` se obtiene de la misma forma que el de una tabla con la diferencia en el valor de `pctfree` pues debe de tomarse en cuenta que el `pctused` lo asigna por default Oracle y que los índices una vez que crecen ya no decrecen hasta su recreación.

Por último se especifica que el índice sea creado en el `tablespace` que se creó para índices de emisión llamado INDEMISION.

El script creado para la creación de los índices se llama 'indices.sql'

#### 4.6 CREACION DE LIGAS

Es necesario que nuestra base de datos de oficina cero pueda ser vista tanto por al área de Desarrollo como la de Producción, esto porque muchas veces será necesario el traspaso de datos o de objetos de una base a otra, ya sea entre Desarrollo y Oficina Cero o entre Oficina Cero y Producción.

Para que nuestra base de datos pueda ser vista desde desarrollo es necesario crear una liga, la liga se crea como sigue:

```
Acceso con mi usuario ops$enrique que tiene entre sus permisos CREATE PUBLIC
DATABASE LINK
```

La instrucción para poder crear una liga es la siguiente:

```
CREATE PUBLIC DATABASE LINK DESA
CONNECT TO SYSTEM IDENTIFIED BY MANAGER
USING DESA
```

Con este comando le estamos pidiendo que se cree la liga DESA hacia la base de datos remota DESA y diciendo que al momento de conectarse se conecte como usuario 'system' para que puedan ser vistos todos los objetos y tener todos los permisos. El servicio DESA en la instrucción `using` debe de obtenerse del archivo `TNSNAMES` de la base de datos de desarrollo en el renglón `SID`.

Las ligas creadas en la base son las siguientes

DESA → Liga creada desde oficina cero para poder ver la base de datos de desarrollo

PROD → Liga creada desde oficina cero para poder ver la base de datos de producción

OFI0 → Liga creada desde producción para poder ver la base de datos de oficina cero  
 OFI0 → Liga creada desde desarrollo para poder ver la base de datos de oficina cero.

Ya con las ligas creadas, la nueva base de datos de Ofi0 puede ser vista por las otras dos bases y las bases de Desarrollo y Producción pueden ser vistas por Oficina Cero permitiendo que los datos pueden ser copiados de una base a otra.

Los scripts fueron creados para todas las tablas con sus índices incluyendo las de tipo Catálogo debido a que eran necesarias para la emisión.

#### 4.7 EJECUCION DE LOS SCRIPTS

Una vez que se hacen los dos scripts se corren estos con una shell de UNIX que se forma de la siguiente manera:

```

/*****
**SHELL QUE SIRVE PARA EJECUTAR LA CREACION**
**** DE TABLAS E INDICES DE EMISION ****
*****/
date > recrea_tablas.log
sqlplus $USUTRNORA @tablas >> recrea_tablas.log
sqlplus $USUTRNORA @indices >> recrea_tablas.log
date >> recrea_tablas.log

```

Tanto el script 'tablas' como 'indices' tiene en su primer línea la instrucción `conn ops$espania identified by ops$.....`, comando con el cual se realiza la conexión a la base de datos con el usuario 'ops\$espania' para que los objetos creados sean propiedad de este usuario como en la base de Desarrollo y Oficina Cero.

El shell anterior lo que hace es primero crear o remplazar el archivo `recrea_tablas.log` con la fecha y hora de inicio, posteriormente ejecuta los archivos `tablas.sql` e `indices.sql` uno por uno, agregando los resultados obtenidos en el mismo archivo `recrea_tablas.log`, esto con el fin de poder visualizar los posibles errores que se pueden obtener de la ejecución, por último se da la instrucción 'date' para guarde la fecha y hora en el que termine.

## 4.8 PERMISOS PUBLICOS A LAS TABLAS

Una vez que estas tablas han sido creadas es necesario dar al rol\_pruebas y al rol\_desarrollo los permisos pertinentes para que puedan seleccionarlas, en el caso de desarrollo, y que puedan realizar inserciones, borrados, selecciones y actualizaciones por parte del rol\_pruebas. Para dar permisos al rol\_pruebas es necesario realizar un script por medio del cual se dan permisos, el script se forma de la siguiente manera por cada una de las tablas creadas:

```
Grant select, insert, delete, update on A2600002 to rol_pruebas
```

Esto se ejecuta desde el usuario de ops\$espania y se debe hacer por cada una de las tablas.

Ya con permisos, se crean los sinónimos públicos a las tablas de la siguiente manera para que cada una de ellas puedan ser vistas de forma pública. Esto con la siguiente sintaxis por cada tabla:

```
Create public synonym A2600002 for ops$espania.A2600002
```

Con esto se pide la creación del sinónimo publico A2600002 para la tabla A2600002 del usuario ops\$espania.

## 4.9 COPIA DE LOS PROGRAMAS DESDE DESARROLLO

Como se comentó el sistema está realizado por desarrolladores de MAPFRE, los programadores ocupan varios tipos de herramientas para su desarrollo comentadas anteriormente, todos los programas se encuentran dentro del directorio:

```
/u/trn30/
```

Es necesario realizar la copia de los programas de la maquina de Producción a nuestro nuevo ambiente por lo que se crea el mismo directorio /u/trn30/. Este subdirectorio es una variable del sistema (ESPANIA\_HOME), que se especifica dentro del archivo .profile de cada usuario de tal forma que los programas al intentar ser ejecutados ya

sea con el comando `run*` de Oracle o por medio del menú de TRON el comando inmediatamente busca el programa dentro de la variable de sistema llamada `$ESPANIA_HOME`, siendo la ruta de esta `/u/trn30/tipo de archivo`, se tienen los siguientes tipos de archivo:

|                           |   |
|---------------------------|---|
| <code>/u/trn30/inp</code> | definiciones de formatos de impresión   |
| <code>/u/trn30/exe</code> | archivos fuente de las formas (FRM)     |
| <code>/u/trn30/frm</code> | archivos de formas                      |
| <code>/u/trn30/pco</code> | archivos fuente de los coboles          |
| <code>/u/trn30/cob</code> | archivos ejecutables de los coboles     |
| <code>/u/trn30/spp</code> | packages y procedimientos               |
| <code>/u/trn30/bin</code> | archivos ejecutables                    |
| <code>/u/trn30/sql</code> | archivos tipo sql                       |
| <code>/u/trn30/ctl</code> | archivos de control                     |
| <code>/u/trn30/sh</code>  | archivos tipo shell                     |
| <code>/u/trn30/pc</code>  | archivos proc                           |
| <code>/u/trn30/ini</code> | archivos de inicialización              |
| <code>/u/trn30/def</code> | definiciones de formatos para impresión |
| <code>/u/trn30/mdf</code> | archivos ejecutables                    |

Para copiar los programas es necesario hacer una transferencia de archivos de la maquina de Producción a la de Oficina Cero por medio de un shell que lo único que hace es conectarse a la máquina de desarrollo por medio de FTP, cambiarse a cada uno de los directorios de programas y ejecutar el comando `mget *` para traer todos los archivos.

#### 4.10 CREACION DE TODOS LOS DEMAS USUARIOS

Se dan de alta tres tipos de usuarios, para programadores (30 usuarios), usuarios de pruebas (13 usuarios), usuarios dba (2 usuarios).

En UNIX cada tipo de usuarios se da de alta en su directorio correspondiente bajo la ruta `/users/`. Se crean los archivos `.profile*` para cada uno de los tipos de usuarios de la siguiente forma:

```

MSPOOL_DIR=/spool/desa/angeromo          ;      export MSPOOL_DIR
LPDEST=PMETSIAU                          ;      export LPDEST
USUTRNORA='ops$angeromo/ops$angeromo'    ;      export
USUORA='ops$angeromo/ops$angeromo'       ;      export USUORA
. /users/mbin/mentorno

```

\* Run : comando de Oracle para ejecutar un archivo ejecutable de Oracle Forms

\* El archivo `.profile` es un archivo que se crea dentro del directorio HOME del usuario y que es leído antes de que el usuario accese al sistema.

```
alias export='exit'  
alias unalias='exit'  
alias w='echo "comando no disponible..."  
alias who='echo "comando no disponible..."  
alias ps='echo "comando no disponible..."  
alias unset='exit'  
alias ksh='exit'  
alias rcp='exit'  
alias matal='exit'  
alias mata='exit'  
alias ftp='exit'  
alias USUTRNORA='exit'  
alias USUORA='exit'  
alias alias='exit'
```

Como se puede observar se creó un directorio para el spool, este directorio se utiliza para almacenar los listados o demás archivos que genere el usuario por medio del uso de la variable \$MSPOOL\_DIR por parte de los programadores, en LPDEST se establece la impresora por default y por último en USUORA y USUTRNORA el usuario y password para la base de datos Oracle, todas estas variables ya son reconocidas por TRON.

También dentro del archivo .profile se manda llamar un archivo mentorno1, mentorno2 o mentorno3, con accesos a desarrollo, oficina cero y a los dos anteriores más producción respectivamente. También se establecen más variables de entorno utilizadas por TRON, el contenido de este archivo se puede encontrar en el anexo con el mismo nombre 'mentorno3' y es muy similar al mismo archivo que ya existe en producción.

Como un ejemplo el usuario que tiene configurado el mentorno3 presenta la siguiente pantalla al acceder al sistema:

Seleccione la base de datos a la que se desea conectar :

|                        |                                 |
|------------------------|---------------------------------|
| 1 - (Des) Desarrollo   | K380I                           |
| 2 - (Of0) Oficina Cero | K380II                          |
| 3 - (Prod) Producción  | T520 (Solo personal autorizado) |

Opción (1) :



Este archivo proporciona seguridad y manejo del rendimiento del servidor ya que no permite que un usuario pueda entrar más de dos veces al sistema ni que en una máquina se pueda entrar con dos usuarios distintos al mismo tiempo.

Después de crear el usuario de UNIX se crea el usuario de Oracle que es el mismo que establece en la variable USUORA del archivo .profile, se le otorga después el rol correspondiente.

El usuario después de ser creado en UNIX y Oracle se debe de dar de alta en algunas tablas para que funcione correctamente estas tablas son:

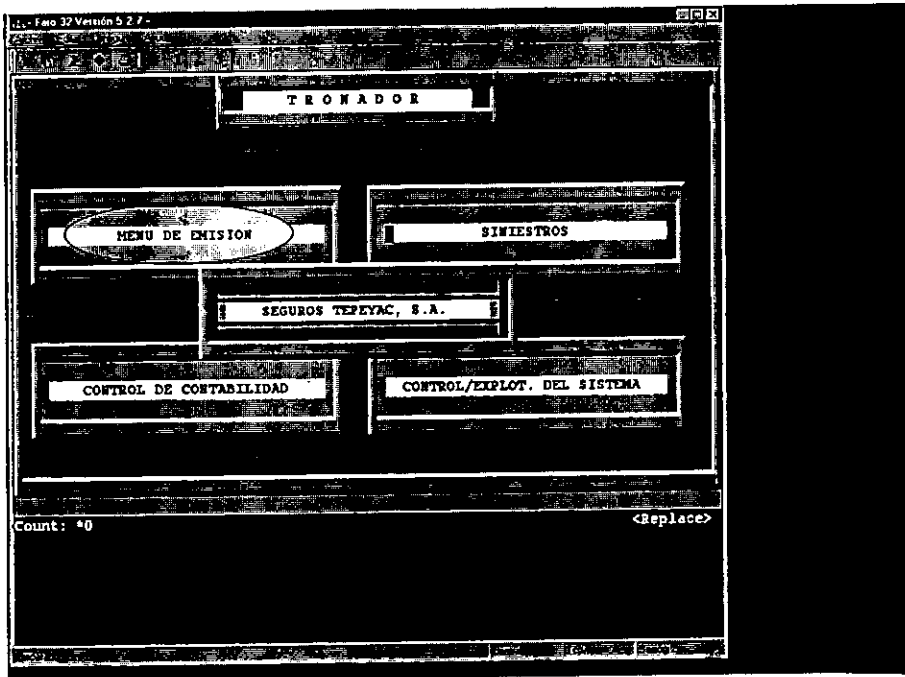
|                  |  |
|------------------|--|
| G1002700         | Tabla que contiene perfil de Oracle y datos del usuario  |
| A5020047         | Se da de alta el MSPool_DIR de UNIX  |
| REHABIL_HUSERS   | Se dan de alta los usuarios para que puedan imprimir desde Oracle  |
| FORMATOS_JETFORM | Se da de alta el usuario con los formatos que utilizará al imprimir  |
| ACCESOS          | Se dan permisos para tener acceso según reglas   |
| A700810          | Se dan de alta para poder emitir pólizas   |
| A700820          | Se dan de alta por reglas  |
| G2120020         | Se dan de alta para poder emitir pólizas   |
| PROGRAMAS        | Aquí se deben dar de alta c/u de los programas que existan en el \$ESPANIA_HOME, se dan de alta una sola vez |
| APSEGPGM         | Se dan de alta los permisos y menús para cada uno de los perfiles  |

Todas estas tablas son creadas con la misma estructura que en Desarrollo en el Tablespace que se creó llamado USERS.

El sistema TRON es de origen español, se instala por medio de un Cd-Rom proporcionado por el grupo MAPFRE, este CD ya está instalado en la máquina en la que se está creando el ambiente de Ofi0, lo que hace su instalación es crear una serie de directorios y archivos ejecutables UNIX para poder ejecutar el menú principal, al acceder un usuario y seleccionar la base de datos a la que se quiere conectar, para acceder al menú ejecuta el archivo menu archivo que fue creado cuando se instaló el software de TRON.

Lo que el usuario ve físicamente al acceder al menú son pantallas realizadas por medio de la herramienta FORMS de Oracle, dentro de la forma principal, se valida el usuario y el password y se hacen llamar todos los demás programas que son necesarios para su ejecución, también son llamadas las reglas para poder acceder a los programas, estas reglas son por ejemplo el poder emitir pólizas de autos para la cual deben de estar dado de alta el usuario en la tabla g2120020.

Para poder acceder a un programa se hace por medio del menú principal y posteriormente a la opción deseada, por ejemplo para acceder a la opción 'Rehabilitación de endosos' el usuario tiene acceso a la aplicación por medio de su usuario y password que se proporcionaron para UNIX, todo el proceso que realizan los archivos .profile y .mentomo es transparente para el usuario de tal forma que después de seleccionar la base de datos lo único que ve es la pantalla principal de TRON que es la que se muestra a continuación.



PANTALLA I MENU PRINCIPAL

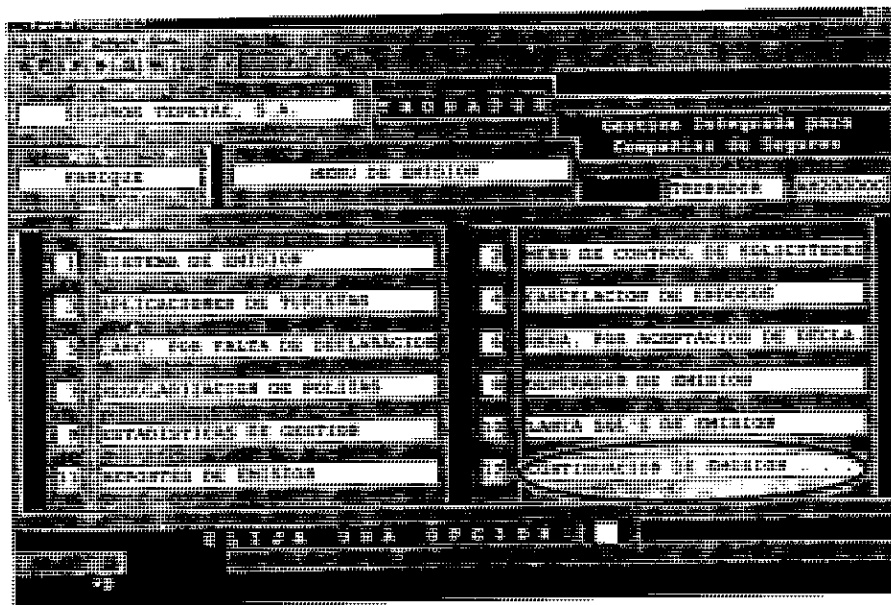
Como un ejemplo se requiere acceder a 'Rehabilitación de endosos', siguiendo la secuencia necesaria.

→ Menu de emision

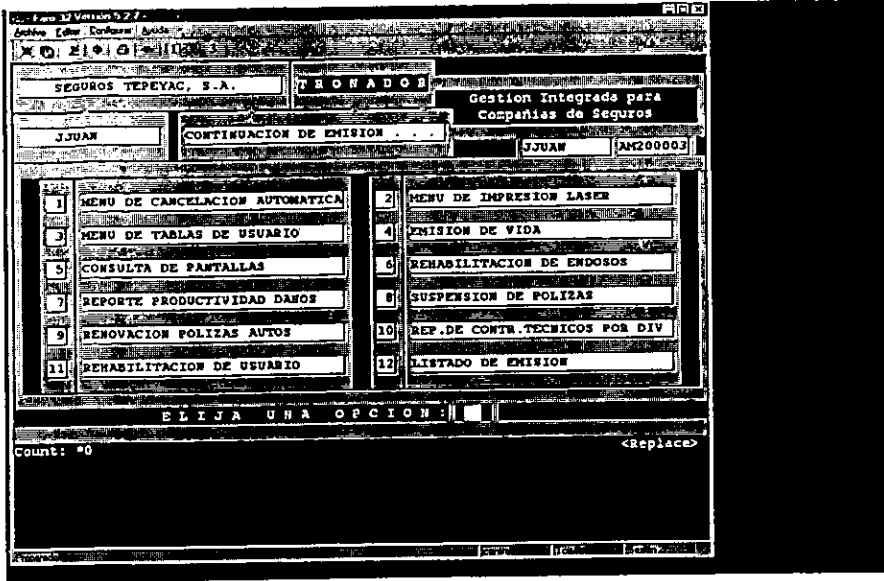
→ Continuación de emisión

→ Rehabilitación de endosos

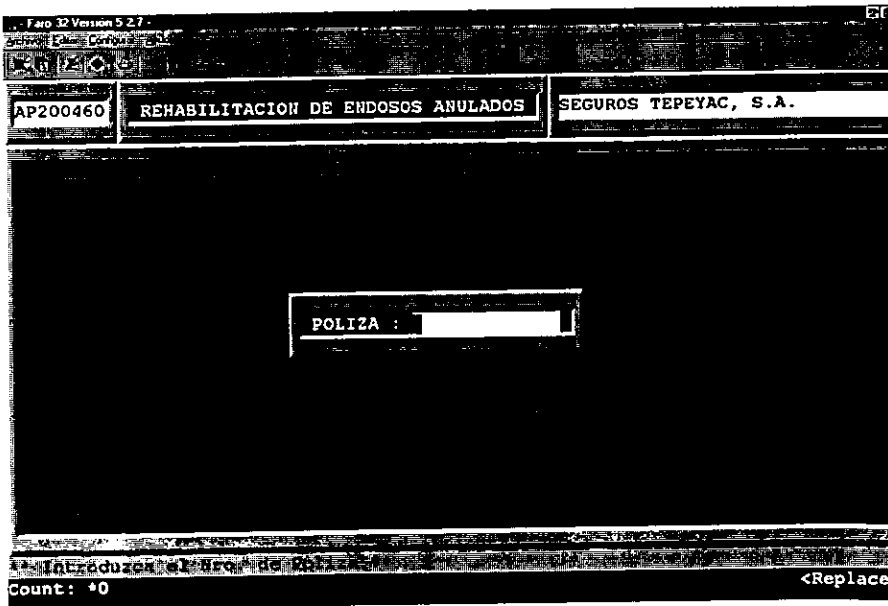
Para esto se accesa a las siguientes pantallas



PANTALLA 2 MENU DE EMISION



PANTALLA 3 CONTINUACION DE EMISION



PANTALLA 4 REHABILITACION DE ENDOSOS

Cómo se puede ver esta pantalla se llama REHABILITACION DE ENDOSOS, el nombre de su archivo FRM e INP en el directorio \$ESPANIA\_HOME es el AP200460.INP y AP200460.FRM que corresponde al nombre real de la forma y aparece arriba del lado izquierdo. Esta información debe de estar dada de alta en la tabla PROGRAMAS que tiene los siguientes campos:

```
SQL> desc programas
Name                               Null?    Type
-----
COD_PROG                           NOT NULL VARCHAR2(12)
DESC_PROG                           NOT NULL VARCHAR2(40)
NOM_OPCION                          VARCHAR2(30)
NUM_OPCION                           NUMBER(2)
PADRE                                VARCHAR2(8)
COD_VERSION                          VARCHAR2(2)
```

Los campo que deben de estar dados de alta para poder acceder a la forma son:

```
cod_prog ='AP200460'
desc_prog='Programa para rehabilitar endosos'
nom_opcion='REHABILITACION DE ENDOSOS ANULADOS'
NUM_OPCION='6'
PADRE='AM200003'
```

El número de opción corresponde al número que aparece en la pantalla 3 correspondiente a Rehabilitación de Endosos que es el número 6.

El campo PADRE se refiere al nombre de la forma por medio de la cual se accedió que en este caso es AM200003 (Continuación de emisión), este nombre aparece en la pantalla 2 en la parte superior derecha.

Para que esta opción pueda ser visualizada se deben de dar de alta en esta tabla todas las pantallas de las cuales depende que son AM00000 que es el menú principal, AM200000 y AM200003 con las mismas características que la AP200460. Si falta alguna simplemente no se visualizará la opción deseada. Además el archivo .inp y .frm deben de existir bajo el directorio UNIX /inp y /frm respectivamente del \$ESPANIA\_HOME.

Para saber a que opciones podrá acceder un usuario es necesario darle el perfil que utilizará en la tabla G1002700, la tabla en donde se dan de alta los programas a los que accederá un usuario dependiendo su perfil es APSEGPGM con los siguientes campos:

```
SQL> desc apsepggm
Name                               Null?    Type
-----
COD_PROG                           NOT NULL VARCHAR2(8)
COD_USR                             NOT NULL VARCHAR2(8)
COD_TIPO                             VARCHAR2(1)
```

El perfil no es mas que un nombre dado de alta en la tabla APSEGPGM en el campo cod\_usr junto con una forma (FRM), forma a la que tendrán acceso todos los usuario que tengan asignado el mismo nombre del perfil en la tabla g1002700 en el campo cod\_perfil.

Siguiendo con el ejemplo pensemos que se trata del usuario de pruebas SILVIARI y queremos tenga un perfil que se llame CONSULTA1 en la tabla G1002700, el cual solamente podrá acceder a la opción de 'Rehabilitación de endosos' ya que únicamente están dadas de alta las formas necesarias para llegar a esta opción para este perfil en la tabla APSEGPGM y pensemos que este perfil lo tienen asignado dos personas más en la tabla G1002700.

Ya que tenemos dados de alta todas las formas necesarios para llegar a la opción 'Rehabilitación de endosos' en la tabla PROGRAMAS se debe de dar de alta el perfil en la tabla APSEGPGM de la siguiente forma

```
COD_PROG='AP200460'
COD_USR ='CONSULTA1'
COD_TIPO='A'
```

En esta tabla se da de alta el perfil CONSULTA1 tantas veces como pantallas necesite es decir, en este ejemplo la tabla APSEGPGM contendrá 4 registros del perfil CONSULTA1. El código del programa o COD\_PROG es el nombre del archivo que se dio de alta en la tabla PROGRAMAS, se debe de dar una sola vez de alta en la tabla programas, el COD\_USR corresponde al nombre del perfil y COD\_TIPO='A' quiere decir que la

pantalla esta abierta para el perfil, si fuera N quiere decir que aunque la tiene dada de alta el perfil, al momento de dar la opción 6 en la pantalla3 mandaría el error 'PANTALLA NO ACTIVA'.

Una vez que ya se tienen dados de alta los programas y el perfil es necesario registrar al usuario en la tabla G1002700 la cual tiene los siguientes campos:

| Name         | Null? | Type          |
|--------------|-------|---------------|
| COD_USER_CIA |       | VARCHAR2 (8)  |
| NOM_USER     |       | VARCHAR2 (20) |
| COD_USER     |       | VARCHAR2 (8)  |
| MENÜ_DEFECTO |       | VARCHAR2 (8)  |
| COD_PERFIL   |       | VARCHAR2 (8)  |

Para seguir con el ejemplo del usuarios SILVIARI se insertaría de la siguiente manera:

```

COD_USER_CIA   ='SILVIARI'
NOM_USER       ='SILVIA PATRICIA RIVERA'
COD_USER       ='ENRIQUE' <<< persona que lo da de alta >>>
MENÜ_DEFECTO  ='AM000000' <<< primer pantalla que verá al acceder al
                sistema >>>
COD_PERFIL     ='CONSULTA1'

```

Con esto se le indica al sistema que al acceder el usuario vea el menú principal y tome las pantallas según estén dadas de alta en APSEGPGM para el perfil correspondiente, en este caso solamente teniendo acceso al menú de Rehabilitación de endosos. Si necesitará acceso a otras opciones solamente es necesario darlos de alta en la tabla PROGRAMAS si no existe y en la tabla APSEGPGM.

## 5 PRINCIPALES TAREAS DE ADMINISTRACION DE LA BASE DE DATOS

El administrador de la base de datos es el responsable del mantenimiento del servidor de Oracle pudiendo así el servidor procesar las peticiones de los usuarios. Es necesario entender la arquitectura de Oracle para un mantenimiento efectivo a la base de datos..

Una de las herramientas más importantes y útiles en la administración de la base de datos es el Server Manager, por medio de esta herramienta se corren la gran mayoría de procesos para la administración de la base de datos.

Dentro de la administración son importantes los siguientes aspectos:

Manejo de los segmentos deshacer

Manejo de tablas

Manejo de índices

Manejo de los usuarios, perfiles, privilegios y roles

Manejo y Reorganización de los datos

Respaldo de la información

## 5.1 MANEJO DE LOS SEGMENTOS DESAHER

Los segmentos deshacer son fundamentales para el funcionamiento óptimo de la base de datos. Un segmento rollback (rbs) es utilizado para almacenar la imagen anterior cuando un proceso esta realizando cambios en los registros de la base de datos.

Una transacción puede utilizar solamente un segmento rollback (rbs) para almacenar todos sus registros que podrá recuperar, varias operaciones pueden escribir en el mismo segmento simultáneamente.

Cuando se hacen cambios en un registro de la base de datos, la imagen anterior es guardada en el segmento rollback. Si esta transacción es recuperada (con la instrucción rollback), el valor del rbs es escrito nuevamente al registro, recuperando el valor inicial.

En Ofi0, la base de datos fue creada con un total de cuatro segmentos rollback, los cuatro del mismo tamaño, algunas veces los procesos suelen fallar debido a que el rollback se llena y no tiene más lugar hacia donde crecer por lo que una de las tareas del DBA es la de verificar que los procesos no realicen demasiados cambios sin grabarlos (commit). Al



realizar un commit después de una transacción, este inmediatamente quita la imagen anterior del registro en el rollback, liberándole espacio, de tal forma que si se realizan muchas operaciones en el rollback sin realizar commit el proceso fracasará y los cambios realizados no serán grabados.

Los segmentos rollback después de ser creados deben de ser puestos en línea para que puedan ser usados, esto se hizo en el script rollbacks al momento de crear la base de datos, para que los segmentos rbs sean puestos en línea cada que arranca la base de datos estos deben de ser especificados en el archivo de inicialización, en este caso en `initOFIO.ora`, el orden en que son puestos es el orden en el que serán utilizados, en nuestro caso el orden fue el siguiente:

```
rollback_segments          =(R01, R02, R03, R04)
```

Esta línea fue comentada en el archivo `initOFIO0.ora` al momento de la creación de la base de datos porque al querer arrancar la primera vez, la base intenta poner en línea todo lo que se especifique aquí y al todavía no estar creados los segmentos rollback mandaría error y no levantaría, después de creados entonces sí son especificados en el archivo de inicialización.

En caso de que algún segmento quede fuera de línea en algún momento se utiliza el comando para ponerlo nuevamente en línea:

```
ALTER ROLLBACK SEGMENT R01 ONLINE
```

En donde `R01` es el nombre del segmento. Para poner el segmento fuera de línea se substituye `ONLINE` por `OFFLINE`.

Esto es muy útil cuando se va a correr un proceso que sabemos necesita un rollback especial por el número de registros que inserta, borra o actualiza. Para este tipo de procesos se puede crear un rollback `R05` el cual no sería puesto en línea hasta el momento que se

corra el proceso, sin ser dado de alta en el archivo de inicialización. Cuando se corra el proceso se utiliza el comando ALTER para ponerlo ONLINE y también el comando :

```
SET TRANSACTION USE ROLLBACK SEGMENT R05
```

Cuando terminemos de utilizarlo o bien una vez que la transacción ha sido asignada a este segmento se pone nuevamente OFFLINE para que no sea tomado posteriormente por otras transacciones. En caso de que haya sido creado para este proceso y ya no se vaya a utilizar entonces damos el siguiente comando para borrarlo:

```
DROP ROLLBACK SEGMENT R05
```

Para conocer en cualquier momento el estado de los rbs se puede dar el siguiente comando:

```
SELECT segment_name SEGMENTO, tablespace_name TSPACE, owner DUENO,
status ESTADO
FROM dba_rollback_segs
```

Obteniendo:

| SEGMENTO | TSPACE | DUENO | ESTADO  |
|----------|--------|-------|---------|
| SYSTEM   | SYSTEM | SYS   | ONLINE  |
| R01      | RBS    | SYS   | ONLINE  |
| R02      | RBS    | SYS   | ONLINE  |
| R03      | RBS    | SYS   | ONLINE  |
| R04      | RBS    | SYS   | ONLINE  |
| R05      | RBS    | SYS   | OFFLINE |

6 rows selected.

La columna DUENO (OWNER) indica el tipo de rollback

- SYS se refiere a un segmento privado
- PUBLIC se refiere a un segmento publico

Se pueden utilizar las vistas V\$ROLLSTAT y V\$ROLLNAME para obtener estadísticas de los rollbacks utilizados actualmente por la instancia:

```
SELECT n.name NOMBRE, s.extents INCREMENTOS, s.rssize TAMANO, s.optsize
OPTIMO, s.hwmsize MAXIMO, s.xacts TRANSAC, s.status ESTADO
from v$rollname n, v$rollstat s
where n.usn=s.usn
```

Obteniendo:

| NOMBRE | INCREMENTOS | TAMANO   | OPTIMO   | MAXIMO   | TRANSAC | ESTADO  |
|--------|-------------|----------|----------|----------|---------|---------|
| SYSTEM | 11          | 892928   |          | 892928   | 0       | ONLINE  |
| R01    | 4           | 20963328 | 20971520 | 20963328 | 0       | ONLINE  |
| R02    | 4           | 20963328 | 20971520 | 47177728 | 0       | ONLINE  |
| R03    | 4           | 20963328 | 20971520 | 20963328 | 0       | ONLINE  |
| R04    | 4           | 20963328 | 20971520 | 20963328 | 0       | ONLINE  |
| R05    | 4           | 20963328 | 20971520 | 20963328 | 0       | OFFLINE |

Se hace el join por medio de la columna 'usn' que es el número que ocupa el rollback o número de identificación.

En algunas ocasiones algunas transacciones pueden quedar bloqueadas, bloqueando el rollback en el que estaban siendo guardadas, esto puede ser debido a varias cosas, la más frecuente es que un registro haya sido subido a rollback para ser modificado y antes de que sea dado commit ( validación ) otro usuario también modifica el registro al mismo tiempo y da commit antes que el primer usuario provocando que la imagen que tiene el primer usuario en el rollback ya no pueda ser recuperada mandando msj. ORA - 01555 SNAPSHOT TOO OLD. Al suceder esto tanto el usuario como el segmento que fue utilizado por el usuario quedan bloqueadas.

El siguiente query nos ayuda a encontrar transacciones bloqueadas:

```
SELECT s.sid, s.serial#, t.start_time,
t.xidusn RB#, s.username USUARIO
FROM v$session s, v$transaction t, v$rollstat r
WHERE s.saddr = t ses_addr
AND t.xidusn = r.usn
AND ((r.curext = t.start_uxext-1) OR
((r.curext = r.extents-1) AND t.start_uxext=0))
```

## 5.2 MANEJO DE LAS TABLAS

Existen varias tablas y vistas del diccionario de datos que son útiles para el manejo de las tablas e índices de usuario, entre las principales encontramos la DBA\_TABLES, esta tabla es importante para el manejo de todas las tablas, entre sus campos que más nos pueden interesar encontramos OWNER, TABLE\_NAME, TABLESPACE\_NAME, INITIAL\_EXTENT, NEXT\_EXTENT, MIN\_EXTENTS, MAX\_EXTENTS, PCT\_INCREASE, NUM\_ROWS.

Con un query sencillo podemos obtener la información de una tabla en específico, por ejemplo la tabla A2000030 una de las más importantes para la emisión de pólizas:

```
SELECT table_name, owner, tablespace_name, next_extent, max_extents
FROM dba_tables WHERE table_name = 'A2000030'
```

Obteniendo como resultado:

| TABLE_NAME | OWNER         | TABLESPACE_NAME | INITIAL_EXTENT | NEXT_EXTENT | MAX_EXTENTS |
|------------|---------------|-----------------|----------------|-------------|-------------|
| A2000030   | OPSS\$ESPANIA | TAEMISION       | 3424256        | 3424256     | 505         |

Con la información obtenida sabemos que se tiene una tabla de ESPANIA en el espacio de tablas TAEMISION con un incremento de 3424256 bytes y un con 505 incrementos como máximo en un tamaño inicial de 3424256.

Los valores de esta tabla pueden ser modificados por medio de la siguiente instrucción:

```
ALTER TABLE ops$espania.A2000030
STORAGE (next_extent 1M)
```

Se pueden modificar los valores de NEXT\_EXTENT y MAX\_EXTENT, no así el OWNER, TABLESPACE o INITIAL, estos son valores que una vez que se creó la tabla no se pueden modificar a menos que se cree, pero pueden ser tomados de la tabla DBA\_TABLES.

La recreación de una tabla consiste, como su nombre lo indica, en crear una tabla con el mismo nombre y con la misma estructura de otra, la cual será eliminada después de crear la nueva.

La tabla DBA\_SEGMENTS puede ser también útil ya que por medio de su campo BYTES se obtiene el tamaño que actualmente tiene la tabla. La administración de las tablas implica varias cuestiones entre ellas los parámetros que se les darán de inicio, el manejo de espacios, el borrado y la recreación.

En cuanto al manejo de los espacios destaca que el DBA debe de tener en cuenta las proporciones en que las tablas irán creciendo para saber cuanto tiempo estará la tabla sin crecer, hasta cuantas veces puede crecer, hasta que tamaño puede alcanzar en el tablespace, para así saber cuando la tabla debe ser recreada ya sea por fragmentación o porque ya no puede crecer más en el tablespace.

Cuando se toma la decisión de recrear una tabla entonces es necesario la obtención de los datos de las tablas y vistas del diccionario de datos ya mencionadas planeando los nuevos espacios que tendrá la tabla y sus respectivos incrementos.

### 5.3 MANEJO DE LOS INDICES

Hablando de los índices se tiene una tabla que es base para administrarlos, la tabla DBA\_INDEXES, en esta tabla encontramos los siguientes campos como los más importantes:

|                 |   |  |
|-----------------|---|--|
| OWNER           | → | Dueño del índice                               |
| INDEX_NAME      | → | Nombre del índice                              |
| TABLE_OWNER     | → | Dueño de la tabla a la que pertenece el índice |
| TABLE_NAME      | → | Tabla a la que pertenece                       |
| TABLE_TYPE      | → | Tipo de tabla                                  |
| UNIQUENESS      | → | Es o no es único                               |
| TABLESPACE_NAME | → | Espacio de tablas en el que esta               |
| INITIAL_EXTENT  | → | Tamaño inicial                                 |
| NEXT_EXTENT     | → | Tamaño de los incrementos                      |
| MIN_EXTENTS     | → | Mínimo de incrementos                          |

|              |   |   |
|--------------|---|---|
| MAX_EXTENTS  | → | Máximo de incrementos                           |
| PCT_INCREASE | → | Porcentaje de aumento den cada incremento       |
| PCT_FREE     | → | Porcentaje de espacio libre para modificaciones |

Existe otra tabla llamada DBA\_IND\_COLUMNS esta tabla proporciona la siguiente información:

|                 |   |                        |
|-----------------|---|------------------------|
| INDEX_OWNER     | → | Dueño del índice       |
| INDEX_NAME      | → | Nombre del índice      |
| TABLE_OWNER     | → | Dueño de la tabla      |
| TABLE_NAME      | → | Nombre de la tabla     |
| COLUMN_NAME     | → | Nombre de cada columna |
| COLUMN_POSITION | → | Posición de la columna |
| COLUMN_LENGTH   | → | Tamaño de la columna   |

Haciendo una combinación entre estas dos tablas podemos obtener información valiosa de un índice de cierta tabla, siguiendo con la tabla A2000030 podemos ejecutar la siguiente consulta:

```
SELECT b.index_name, b.table_name, c.bytes,
a.column_name,a.column_position, b.uniqueness, b.tablespace_name
FROM dba_ind_columns a, dba_indexes b, dba_segments c
WHERE b.index_name = a.index_name
AND b.table_name = a.table_name
AND b.table_owner = a.table_owner
AND b.owner = a.table_owner
AND c.owner = a.table_owner
AND c.owner = a.table_owner
AND c.segment_name = b.index_name
AND c.segment_name = a.index_name
AND a.table_owner = 'OPS$ESPANIA'
AND a.table_name = 'A2000030'
```

Obteniendo la siguiente información:

| INDEX_NAME   | TABLE_NAME | BYTES   | COLUMN_NAME    | COLUMN_POSITION | UN | TABLESPACE |
|--------------|------------|---------|----------------|-----------------|----|------------|
| I15_A2000030 | A2000030   | 1761280 | NUM_SECU_POL   | 1               | NO | INDEMISION |
| I15_A2000030 | A2000030   | 1761280 | NUM_END        | 2               | NO | INDEMISION |
| I15_A2000030 | A2000030   | 1761280 | COD_RAMO       | 3               | NO | INDEMISION |
| I15_A2000030 | A2000030   | 1761280 | FECHA_VIG_END  | 4               | NO | INDEMISION |
| I15_A2000030 | A2000030   | 1761280 | FECHA_VENC_END | 5               | NO | INDEMISION |
| I8_A2000030  | A2000030   | 1638400 | COD_DIV_REG    | 1               | NO | INDEMISION |
| I8_A2000030  | A2000030   | 1638400 | COD_DIV_PROV   | 2               | NO | INDEMISION |

|             |          |         |                |   |               |
|-------------|----------|---------|----------------|---|---------------|
| I8_A2000030 | A2000030 | 1638400 | COD_OFI_AGENTE | 3 | NO INDEMISION |
| I8_A2000030 | A2000030 | 1638400 | COD_PROD       | 4 | NO INDEMISION |
| I8_A2000030 | A2000030 | 1638400 | NUM_POL1       | 5 | NO INDEMISION |

Relacionando con la tabla DBA\_EXTENTS podemos ver que el índice I15\_A2000030 tiene un tamaño de 1761280, y que esta conformado por 5 columnas, no es índice único, el espacio de tablas es INDEMISION y el orden de las columnas indexadas, toda esta información es muy útil para la recreación de este índice en específico.

Existe otra forma de verificar el estado de un índice, esto es por medio de la tabla INDEX\_STATS. Es necesario primero ejecutar el comando ANALYZE INDEX al índice para que esta tabla contenga la información actual, por ejemplo para analizar el estado del índice I1 de la tabla a2000025:

```
ANALYZE INDEX i1_a2000025 VALIDATE STRUCTURE;
```

Posteriormente verificamos el estado del índice en la tabla INDEX\_STATS

```
SELECT blocks, pct_used, distinct_keys, lf_rows, del_lf_rows
FROM index_stats;
```

Obtenemos los siguientes resultados;

| BLOCKS | PCT_USED | DISTINCT_KEYS | LF_ROWS | DEL_LF_ROWS |
|--------|----------|---------------|---------|-------------|
| 1380   | 52       | 115112        | 115126  | 0           |

La columna blocks contiene el mismo dato que la columna bytes en la tabla dba\_segments pero en bloques, es decir divide los bytes entre el tamaño del block especificado en el archivo init.ora, en este caso 8192 bytes. Es recomendable la recreación del índice cuando el valor obtenido en DEL\_LF\_ROWS excede el 30% del valor de LF\_ROWS, ya que esto significa que existe más del 30% de registros borrados en relación con los registros totales.

Para reconstruir un índice podemos hacerlo de dos formas dependiendo del porqué de su reconstrucción:

- 1) Si el índice se reconstruirá debido a que ya no cabe en el tablespace y queremos que continúe con la misma estructura, entonces se puede utilizar el comando rebuild de la siguiente forma:

```
ALTER INDEX ops$espania.i1_a2000025
REBUILD
TABLESPACE indemision2
```

En este caso no es necesario borrar el índice antes de ejecutar la instrucción anterior ya que el nuevo índice tendrá el mismo nombre que el anterior y con su misma estructura.

Se debe de tener en cuenta que debe existir espacio suficiente en el nuevo tablespace para el índice, tomando este espacio de los datos obtenidos anteriormente

- 2) Cuando queremos recrear un índice por problemas de fragmentación, crecimiento indebido ó demasiados registro borrados, entonces es necesario obtener los índices de las tablas mencionadas anteriormente y después crearlo como índice nuevo. El índice lo podemos borrar con el comando

```
DROP INDEX ops$espania.i1_a2000025
```

Después de borrarlo ya podemos crearlo con la instrucción CREATE INDEX y los parámetros que sean necesarios.

## 5.4 MANEJO DE LOS USUARIOS

Los usuarios pueden ser vistos por medio de la vista del diccionario de datos dba\_users con la siguiente descripción:

|                      |   |  |
|----------------------|---|--|
| USERNAME             | → | Usuario                                |
| USER_ID              | → | Número de identificación               |
| PASSWORD             | → | Password encriptado                    |
| DEFAULT_TABLESPACE   | → | Espacio de tablas asignado por defecto |
| TEMPORARY_TABLESPACE | → | Espacio de tablas temporal asignado    |



---

|         |   |                     |
|---------|---|---------------------|
| CREATED | → | Usuario que lo creo |
| PROFILE | → | Perfil asignado     |

Son creados 45 usuarios de los cuales 13 son para pruebas, 2 para DBA's y 30 desarrolladores.

Existen además de los roles por default de Oracle los dos roles creados para la instancia Ofi0, rol\_desarrollo y el rol\_pruebas. El primero es otorgado a todos los usuarios de desarrollo y el segundo para los 13 usuarios de pruebas. Esta información se puede obtener de la tabla de diccionario de datos DBA\_ROLE\_PRIVS. La cual tiene la siguiente estructura

|              |   |                                      |
|--------------|---|--------------------------------------|
| GRANTEE      | → | Usuarios que tiene el rol            |
| GRANTED_ROLE | → | Rol otorgado                         |
| ADMIN_OPTION | → | Opción para poder administrar el rol |
| DEFAULT_ROLE | → | S o N como su role por defecto       |

La creación del role en este caso llamado rol\_pruebas se hace por medio de la instrucción:

```
CREATE ROLE rol_pruebas
```

Y después solamente se le van dando los permisos a este rol según los tipos de acceso que necesite con la sig. instrucción:

```
GRANT CONNECT TO rol_pruebas
```

Si simplemente queremos ver los roles existentes se tiene la vista dba\_roles con la siguiente estructura:

|                   |   |                                    |
|-------------------|---|------------------------------------|
| ROLE              | → | Nombre del role                    |
| PASSWORD_REQUIRED | → | S o N requiere password al acceder |

Si queremos conocer los privilegios que tiene cada role, esta información se puede obtener de las vistas:

#### DBA\_TAB\_PRIVS

|            |   |                                    |
|------------|---|------------------------------------|
| GRANTEE    | → | Usuario que adopta el permiso      |
| OWNER      | → | Dueño de la tabla afectada         |
| TABLE_NAME | → | Tabla a la que se le da el permiso |
| GRANTOR    | → | Usuario que otorga el permiso      |
| PRIVILEGE  | → | Permiso que da el usuario          |

GRANTABLE → S o N puede dar el permiso a los demás

#### DBA\_SYS\_PRIVS

GRANTEE → Usuario que adopta el permiso  
 PRIVILEGE → Permiso que da el usuario  
 ADMIN\_OPTION → Para poder dar el mismo permiso a otro

Los usuarios creados para DBA simplemente es necesario otorgarles el rol DBA, rol que existe en la tabla DBA\_ROLES y que es el rol del super usuario, es decir, incluye todos los privilegios.

Así como utilizamos el comando GRANT, existe también el REVOKE que quita permisos otorgados a cierto role o usuario. Si se quiere dar un permiso a todos los usuarios se utiliza el comando

```
GRANT SELECT, UPDATE, INSERT, DELETE on A2000025 TO PUBLIC
```

Con esta instrucción todos los usuarios pueden realizar inserción, borrado, selección y actualización de registros en la tabla A2000025.

Si se requieren dar los mismos permisos pero a un solo usuario anteriormente creado como ops\$araceli se utiliza el comando siguiente:

```
GRANT SELECT, UPDATE, INSERT, DELETE on A2000025 TO ops$araceli
```

Los únicos usuarios que pueden crear objetos en Ofi0 aparte de los DBA son los que están en el rol\_desarrollo y solamente pueden crear en el espacio de tablas USERS. Para darles este atributo es suficiente el siguiente comando:

```
ALTER USER ops$araceli QUOTA 20M ON users
```

Limitando al usuario a tener hasta 20 Megabytes de espacio disponible en el tablespace users, un usuario puede tener cuotas diferentes en varios tablespaces siempre evitando que llene el espacio de tablas con sus objetos. No es necesario establecer una cuota ni en el tablespace temporal ni en el rollback pues su espacio es público.

Para saber las cuotas que tiene un usuario y en que espacio de tablas podemos utilizar la siguiente instrucción que hace uso de la vista DBA\_TS\_QUOTAS:

```
SELECT tablespace_name, max_bytes, bytes
FROM dba_ts_quotas
WHERE username='OPSSCARMENAR'
```

Analizando el resultado obtenemos que el usuario OPSSCARMENAR tiene una cuota de 20 M en el tablespace USERS. En el caso de que la columna max\_bytes nos de cómo resultado '-1' indica que su cuota no tiene límite, el valor de bytes es el espacio que hasta ahora ocupa el esquema\* de su usuario.

| TABLESPACE_NAME | MAX_BYTES | BYTES |
|-----------------|-----------|-------|
| DESARROLLO      | 20480     | 16384 |

Los usuarios pueden ser borrados con el comando:

```
DROP USER ops$araceli
```

O puede ser borrado junto con todos sus objetos (esquema) por medio de la instrucción:

```
DROP USER ops$araceli CASCADE
```

## 5.5 MANEJO Y REORGANIZACION DE LOS DATOS

El manejo de los espacios es fundamental para una buena administración de la base de datos, la reorganización de los datos consta de dos aspectos

- La creación de objetos
- La recreación de objetos

\* El esquema lo forma una colección de objetos como tablas, vistas, clusters, packages y procedimientos asociados a un usuario en particular

Dentro de la creación de los objetos es fundamental el saber lo más exacto posible las dimensiones que tendrán los objetos al ser creados y la dimensión en la que irán creciendo, esto, para poder dar los valores de storage adecuados. Un calculo mal hecho sobre un objeto puede causar que el objeto llene el espacio de tablas causando un bloqueo en el sistema al querer insertar registros. En el caso de la base de datos de OFI0 se tiene planeado un crecimiento de objetos aproximado al crecimiento de los mismos en desarrollo.

La recreación de objetos depende del tipo de objeto que se quiera recrear, las tablas y los índices manejan sus espacios de diferente manera pues mientras las tablas al borrarles datos liberan espacio que puede ser reutilizado, los índices una vez que crecen no pierden tamaño hasta su recreación, hablando de los espacios de tablas se necesita saber también el tamaño que podrán alcanzar los objetos creados en él para saber cuanto tiempo pueden permanecer sin agregares espacio.

Se puede agregar más espacio a un tablespace se hace por medio de la inserción de datafiles, con el siguiente comando

```
ALTER tablespace ADD datafile '/dev/vg01/rlvol010' size 500M
```

Antes de agregar espacio a un tablespace es necesario valorar el estado en el que se encuentra ya que muchas veces es mejor hacer la recreación de objetos para liberar espacio, algunas otras veces nos damos cuenta de que existen objetos que ya no se usan por lo que lo mejor es respaldarlos y borrarlos del espacio de tablas, los objetos que nos pueden dar mayor espacio libre después de su recreación son los índices.

Para ver la cantidad de espacio libre en un tablespace nos es útil la tabla dba\_free\_space, para saber los espacios libres de un tablespace podemos utilizar la instrucción siguiente

```
select bytes tamaño, tablespace_name tabspace
from dba_free_space
where tablespace_name=(TAEMISION)
```

El resultado obtenido es algo similar a esto

```
TAMANO      TABSPACE
-----
1,142,006,784 TAEMISION
```

Se deben de tomar en cuenta varios aspectos para la recreación de los objetos, el tamaño que ocupan, el espacio libre que se tiene, la fragmentación y el uso que se tenga sobre ellos.

## 5.6 RESPALDO

Para realizar el respaldo es necesario montar el filesystem `bcrespl` que es un sistema de archivos que pertenece al a máquina de Producción y que fin creado con la finalidad de respaldar aquí información para posteriormente ser subida a cintas magnéticas.

El respaldo consta de dos partes, primero se corre un script que se diseño llamado `tablas_espania.sql` y después se corre el segundo script para las demás tablas de los usuarios. El primer script es el siguiente:

```
set head off
set pagesize 49000
set linesize 150
spool tablas_espania.sh
select
'exp file='||table_name||' tables='||table_name||' rows=y buffer=10000000
userid
='||chr(39)||'ops$espania/spo$espania' ||chr(39)||'
compress '||table_name||'.dmp'
from all_tables
where owner='OPS$ESPANIA'
/
spool off
```

Este script lo que hace es generar un archivo `.sh` el cual va formando instrucciones como la siguiente:

```
exp file=A1000108 tables=A1000108 rows=y
buffer=10000000 userid='ops$espania/spo$espania'
compress A1000108.dmp
```

El script generará una de estas instrucciones por cada una de las tablas para el usuario `espania`, la instrucción `export`, propia de Oracle, al ejecutarla crea una archivo plano dentro del `bcrespl` por cada una de las tablas del usuario `espania`, copiando tanto

estructura como los datos, el archivo plano lo comprime con la herramienta compress (comprimir) de UNIX y lo deja con una extensión .dmp.

Ya que se respaldan estas tablas entonces se hace el mismo procedimiento pero ahora para respaldar las tablas de los demás usuarios por medio del archivo `tablas_usuarios.sql` que es el siguiente:

```
set pagesize 1000
set linesize 150
set head off
set echo off
spool tablas_usuarios.sh
select
'exp file='||username||' owner='||chr(39)||username||chr(39)||' rows=y
buffer=10000000 log='||username||' userid=emoreno/emoreno
compress '||username||'.dmp
'
from dba_users
where username not in ('SYS','SYSTEM','OPS$ESPAÑA')
/
select
'exp file='||username||' owner='||chr(39)||username||chr(39)||' rows=n
buffer=10000000 log='||username||' userid=jjuan/jjdbacm
compress '||username||'.dmp
'
from dba_users
where username = 'OPS$ESPAÑA'
spool off
/
```

Este segundo script lo que hace es formar un segundo archivo .sh el cual generará un archivo plano por cada usuario excepto espania igualmente con extensión .dmp y otro archivo plano de la pura estructura de las tablas de espania sin registros (rows=n).

Ya con los dos shells formados entonces se ejecutan por medio del archivo `lanza_respaldo` que contiene lo siguiente:

```
date > respaldo.log
sh tablas_usuarios.sh >> respaldo.log
sh tablas_espania.sh >> respaldo.log
date >> respaldo.log
```

Se genera un archivo `respaldo.log` para poder ver todo lo sucedido con los dos scripts del respaldo, con esto tendremos un respaldo completo de la base de datos de Ofi0 y para recuperarlo solamente será necesario el comando `import` de Oracle.

Una vez que se realizaron los dos respaldos en el sistema de archivos `beresp1`, entonces es necesario subir los archivos creados a unidad de cinta con el comando `tar` de UNIX.

En el anexo podemos encontrar el código de algunos programas de tipo `sql` que nos son útiles para administrar la bases de datos.

## CONCLUSION

La creación de la base de datos de Oficina Cero tuvo una duración de aproximadamente dos meses desde su planeación hasta su implementación.

La nueva Base de Datos ha logrado mejorar las expectativas que se plantearon al principio del proyecto, ahora el control de los programas y objetos corre a cargo únicamente del área de administración de la base de datos mejorando la calidad en los procesos y un mejor manejo de espacios en los equipos, aprovechando los recursos que se tenía sin utilizar.

Mejóro significativamente las pruebas que hacen los usuarios antes de traspasar objetos, el porcentaje de fallas ha bajado significativamente provocando que los usuarios se sientan mejor atendidos ya que el tiempo de respuesta mejoró desde el requerimiento hasta la solución.

El ambiente de Oficina Cero vino a dar a Seguros Tepeyac mayor seguridad y confianza en sus procesos, manejo y almacenamiento de su información ahora depende de mi persona como DBA el mantener que el manejo de información sea el óptimo.



ANEXO

Mentorno

```

#          *****
#          * <<<<< VARIABLES DEL SISTEMA >>>>> *
#          *****

#          *****
#          * MENTORNO, para todos los usuarios de desarrollo *
#          *****

#          *****
#          *
#          * "MENTORNO", el original fue creado por DBAs y este es *
#          * adecuado, para trabajar en la maquina nueva de ofi0 *
#          * K380. Ahora maneja basicamente el ambiente de oficina *
#          * cero *
#          *
#          * Enrique Alberto Moreno Solís mayo 2001 *
#          *
#          *****

asignar_variables () {
MENTORNO=S ; export MENTORNO
ENTORNO_TRONADOR=tepeyac ; export ENTORNO_TRONADOR
LISTA_EQUIPOS="central" ; export LISTA_EQUIPOS
gestion=/desatron/desa/gestion ; export gestion
APL_HOME=$gestion/aplicacion ; export APL_HOME
AUT_HOME=$gestion/autoriza ; export AUT_HOME
BLQ_HOME=$gestion/bloqueos ; export BLQ_HOME
HIST_HOME=$gestion/historico ; export HIST_HOME
TMPDIR=/spool/tmpdir ; export TMPDIR
TZ=CST6CDT ; export TZ
ESPAÑA_HOME=/u/trn30 ; export ESPAÑA_HOME
EDITOR=vi ; export EDITOR
CONEXION_RED=S ; export CONEXION_RED
ORAPIPES=v1 ; export ORAPIPES
ORACLE_HOME=/oradesatron/ora734/app/oracle/product/7.3.4
export ORACLE_HOME
ORACLE_PATH=.:$HOME/inp:$HOME/frm:$ESPAÑA_HOME/frm ; export
ORACLE_PATH
ORATERMPATH=$ORACLE_HOME/forms30/admin/resource ; export ORATERMPATH
COBOL_PATH=.:$HOME/cob:$ESPAÑA_HOME/cob ; export COBOL_PATH
LPI_PATH=.:$HOME:$HOME/lpi:$ESPAÑA_HOME/lpi ; export LPI_PATH
PLUS_PATH=.:$HOME/sql:$ESPAÑA_HOME/sql ; export PLUS_PATH
REGLAS_HOME=$HOME/reg ; export REGLAS_HOME
MANDA=rcp ; export MANDA
LANGUAGE=American_America.WE8PC850 ; export LANGUAGE
NLS_LANG=American_America.WE8PC850 ; export NLS_LANG
MSPOOL_LISTAR="N"; export MSPOOL_LISTAR

NEWSYS_DIR=/users/newsys/ ; export NEWSYS_DIR
USER=`logname` ; export USER

```

```

USUARIO=`logname`           ;      export USUARIO
COMEXT=remsh                 ;      export COMEXT
MAQ=central                  ;      export MAQ
RESP=cp                      ;      export RESP
CAMB=mv                      ;      export CAMB
USUARIOM=root1              ;      export USUARIOM
CAMOD=chmod_entorno         ;      export CAMOD
CAMOW=chown_entorno         ;      export CAMOW
CAMGP=chgrp                 ;      export CAMGP
#####

#LIBORACLE="-L/respa/ora73/lib -lsql -lsqlnet -lncl -lsqlnet
/respa/ora73/lib/libclient.a -lcommon -lgeneric -lsqlnet -lncl -lsqlnet
/respa/ora73/lib/libclient.a -lcommon -lgeneric -lepc -lnlsrt13 -lc3v6 -
lcore3 -lnlsrt13 -lcore3 -lnlsrt13 -lcl -lm -lm"
LIBORACLE="-L$ORACLE_HOME/lib -lsql -lsqlnet -lncl -lsqlnet
$ORACLE_HOME/lib/libclient.a -lcommon -lgeneric -lepc -lnlsrt13 -lc3v6 -
lcore3 -lnlsrt13 -lcore3 -lnlsrt13 -lcl -lm -lm"

export LIBORACLE
}
#####
### CREA DIRECTORIOS ENTORNO #####
#####
verifica_cod () {
    for A in 0 1 2 3 4 5 6 7 8 9 0
    do
        [ "$1" = $A ] && return 0
    done
    echo "Codigo Invalido. Eliminando Sesion."
    sleep 1
    kill -9 0
}

# *****
# * Bloque, para limite de sesiones por usuario *
# *****

#sesion() {
quien=`whoami`
if [ `who|grep -c $quien` -ge 3 ]
then
    banner " Atencion"
    echo "Solo tiene derecho a dos sesiones activas."
    echo "Superado limite de usuarios concurrentes."
    exit
fi
#}
banner mbase

mbase () {
    echo
    echo "Seleccione la base de datos a la que se desea conectar : "

```

```

echo
echo "1 - (Des) Desarrollo                150"
echo "2 - (Of0) Oficina Cero             150"
echo "3 - (Bak) Replica Produccion       T500 (Solo personal
autorizado)"
echo
echo "Opcion (1) : \c"
read A
echo
quiensoy=`whoami`
if [ "$A" = "1" -o "$A" = "" ]
then
    echo "Ha seleccionado Desarrollo I50 *"
    ORACLE_SID=DESA                export ORACLE_SID
    ORACLE_HOME=/oracle8i/app/oracle/oracle/product/8.1.5i/
    export ORACLE_HOME
    TWO_TASK=DESA
    export TWO_TASK
    HOME=/desatron/desa/$quiensoy
    export HOME
    cd
    ESPANIA_HOME=/u/trn30 ;        export ESPANIA_HOME
    NEWSYS_DIR=/users/newsys/      ; export NEWSYS_DIR
    PATH=/bin:.$HOME/bin:$HOME/sbin:$u/sbin:$ORACLE_HOME/bin
    export PATH
    PATH=$PATH:/users/sbin:$ESPANIA_HOME/bin ;export PATH
    PATH=$PATH:$NEWSYS_DIR/bin:$HOME/bin:$ORACLE_HOME/forms
    export PATH
    PATH=$PATH:$ORACLE_HOME/formsnew ; export PATH
    PATH=$PATH:$ESPANIA_HOME/cbin:$ESPANIA_HOME/cob ; export
PATH
    PATH=$PATH:$HOME/exe:$ESPANIA_HOME/exe:$HOME/com ; export
PATH
    PATH=$PATH:$ESPANIA_HOME/com:$ORACLE_HOME/bin2 ; export
PATH
    MSPool_DIR=$MSPool_DIR ;        export MSPool_DIR
    BASE=D71
    fi
if [ "$A" = "2" ]
then
    echo "Ha seleccionado Oficina Cero I50"
    ORACLE_SID=OFI0
    export ORACLE_SID
    ORACLE_HOME=/oradesatron/ora734/app/oracle/product/7.3.4
    TWO_TASK=OFI0
    export TWO_TASK
    export ORACLE_HOME
    HOME=/desatron/ofio/$quiensoy
    export HOME
    cd
    ESPANIA_HOME=/u/trn30 ;        export ESPANIA_HOME
    NEWSYS_DIR=/users/newsys/      ; export NEWSYS_DIR
    PATH=/bin:.$HOME/bin:$HOME/sbin:$u/sbin:$ORACLE_HOME/bin
    export PATH
    PATH=$PATH:/users/sbin:$ESPANIA_HOME/bin ; export PATH

```

```

PATH=$PATH:$NEWSYS_DIR/bin:$HOME/bin:$ORACLE_HOME/forms
export PATH
PATH=$PATH:$ORACLE_HOME/frm ; export PATH
PATH=$PATH:$ESPAÑA_HOME/cbin:$ESPAÑA_HOME/cob ; export
PATH
PATH=$PATH:$HOME/exe:$ESPAÑA_HOME/exe:$HOME/com ; export
PATH
PATH=$PATH:$ESPAÑA_HOME/com:$ORACLE_HOME/bin2 ; export
PATH
MSPOOL_DIR=$MSPOOL_DIR ; export MSPOOL_DIR
BASE=OFIO
fi
if [ "$A" = "3" ]
then
TWO_TASK=PRD
export TWO_TASK
ORACLE_HOME=/oracle8i/app/oracle/oracle/product/8.1.5i/
export ORACLE_HOME
HOME=/users/$quiensoy
export HOME
cd
ESPAÑA_HOME=/u/trn30 ; export ESPAÑA_HOME
NEWSYS_DIR=/users/newsys/ ; export NEWSYS_DIR
PATH=/bin:$u/mbin:$ORACLE_HOME/bin ; export PATH
PATH=$PATH:/users/mbin:$ESPAÑA_HOME/bin ; export PATH
PATH=$PATH:$NEWSYS_DIR/bin:$HOME/bin:$ORACLE_HOME/forms
export PATH
PATH=$PATH:$ORACLE_HOME/formsnew ; export PATH
PATH=$PATH:$ESPAÑA_HOME/cbin:$ESPAÑA_HOME/cob ; export
PATH
PATH=$PATH:$HOME/exe:$ESPAÑA_HOME/exe:$HOME/com ; export
PATH
PATH=$PATH:$ESPAÑA_HOME/com:$ORACLE_HOME/bin2 ; export
PATH
MSPOOL_DIR=$MSPOOL_DIR ; export MSPOOL_DIR
BASE=CENTRAL
ORACLE_HOME=/oracle8i/app/oracle/oracle/product/8.1.5i/
export ORACLE_HOME
echo "ATENCIÓN: Ha seleccionado CENTRAL"
fi
export BASE
echo

PATH=$PATH:/u/mbin:$ORACLE_HOME/bin2
export PATH
#[ ! -d $HOME/inp ] && mcreadir
export PS1
mbase
asignar_variables
chingon () {
MENTORNO='' ; export MENTORNO
}
normal () {
MENTORNO=S ; export MENTORNO
}

```



```

        tablespace_name tabspace
from all_indexes
where table_name=upper('&1')
/
SELECT SEGMENT_NAME NOMBRE,
       bytes
from   dba_segments
where  segment_name in (select index_name from all_indexes
                        where table_name=upper('&&1'))
/
select table_name tabla,
       index_name indice,
       column_name columna,
       column_position posicion
from   dba_ind_columns
where  TABLE_NAME=upper('&&1')
ORDER BY INDEX_NAME, COLUMN_POSITION
/
spool off
!listar $MSPOOL_DIR/busca_indice.lst
exit

```

#### Ve\_datafiles.sql

```

*****
Este archivo permite ver los datafiles de la bases de datos así*** **
      como sus tamaños y el tablespace al que pertenecen      ***
***                Enrique A. Moreno Solís                ***
*****
col nombre          for a33
col status          for a10
col tamanno        for 99,999,999,999
col tabspace       for a13
break on tablespace on report
compute sum of tamanno on tablespace
compute sum of tamanno on report
spool $MSPOOL_DIR/datafiles
select file_name nombre,
       status,
       bytes tamanno,
       tablespace_name tablespace
from dba_data_files
order by tablespace_name, file_name asc
/
spool off
!listar $MSPOOL_DIR/datafiles.lst

```

#### esp\_por\_tablespace.sql

```

*****
Este archivo permite ver el espacio disponible total y por *** **
      fragmento de cada tablespace                        *** **
      Enrique A. Moreno Solís                            ***
*****

```

```
col tamaño for 999,999,999,999,999
col tabspace for a15
break on report
compute sum of tamaño on report
select bytes tamaño, tablespace_name tabspace
from dba_free_space
where tablespace_name=upper('&tabspace')
order by tablespace_name, bytes asc
```

## lock\_por\_objeto.sql

```
*****
Este reporte permite ver cuando un objetos esta bloqueado *** *** y
quien lo esta bloqueando así como el número de sesión del *** ***
usuario que lo esta bloqueando ***
*** Enrique A. Moreno Solís ***
*****
col object_name for a10
col session_id for 99999999999
col oracle_username for a15
col os_user_name for a12
SET PAUSE ON PAUSE "Oprime [ENTER] para continuar ..."
SET PAGESIZE 10
SELECT A.OBJECT_ID,
       A.OBJECT_NAME,
       B.SESSION_ID,
       B.ORACLE_USERNAME,
       B.OS_USER_NAME,
       B.LOCKED_MODE,
       B.PROCESS
FROM DBA_OBJECTS A, V$LOCKED_OBJECT B
WHERE A.OBJECT_ID = B.OBJECT_ID
AND A.OBJECT_NAME = '&OBJETO'
```

## Obj\_por\_nombre.sql

```
*****
Archivo que nos permite ver la información más importante de un***
*** Objeto cualquiera que se le introduzca, útil para admin. ***
*** Enrique A. Moreno S. ***
*****
COLUMN NOMBRE FOR A25 JUSTIFY CENTER
COLUMN TIPO FOR A8 JUSTIFY CENTER
COLUMN USUARIO FOR A13 JUSTIFY CENTER
COLUMN TABSPACE FOR A15 JUSTIFY CENTER
COLUMN ext FOR 999,999,999 JUSTIFY CENTER
COLUMN nex_ext FOR 999,999,999 JUSTIFY CENTER
COLUMN TAMANO FOR 999,999,999,999,999 JUSTIFY CENTER
SET PAGESIZE 1000
SET LINESIZE 150
TTITLE "OBJETOS POR NOMBRE " SKIP 4
SPOOL $MSPOOL_DIR/OBJXNOMBRE.lst
select segment_name nombre,
       segment_type tipo,
       owner usuario,
```

```

        bytes tamaño,
        extents ext,
        next_extent nex_ext,
        tablespace_name tablespace
from dba_segments
where segment_name like upper('%&1%')
order by owner,tablespace_name, bytes
/
SPOOL OFF
!listar $MSPOOL_DIR/OBJXNOMBRE.lst

```

#### ve\_privilegios.sql

```

*****
Archivo que permite ver que permisos se tienen sobre un objeto,***
***      quién otorgo el permiso, y el dueño del objeto      ***
***      Enrique A. Moreno Solís                             ***
*****
COLUMN Aquien          FOR A15
COLUMN duenno          FOR A13
COLUMN tabla           FOR A13
COLUMN otorgo          FOR A13
COLUMN priv            FOR A07
COLUMN opc             FOR A02
break on duenno on tabla on otorgo on aquien
SELECT
    OWNER    Duenno,
    TABLE_NAME tabla,
    GRANTOR  otorgo,
    GRANTEE  Aquien,
    PRIVILEGE priv,
    GRANTABLE opc
FROM    DBA_TAB_PRIVS
WHERE   TABLE_NAME=UPPER('&1')

```



---

**BIBLIOGRAFIA****ORACLE8: DATABASE ADMINISTRATION****Volume 1 – STUDENT GUIDE**

Schwinn, Ulrike  
Venkatachalam, Vijayanandan  
Ed. University México  
1a. Edición  
México 1998.

**ORACLE8: DATABASE ADMINISTRATION****Volume 2 – STUDENT GUIDE**

Schwinn, Ulrike  
Venkatachalam, Vijayanandan  
Ed. University México  
1a. Edición  
México 1998.

**PUESTA A PUNTO DE ORACLE8**

Adolki  
Anand & Velpurri  
Ed. McGraw-Hill  
1ª. Edición  
España 1998

**RESOLUCION DE PROBLEMAS EN ORACLE**

Corey, Michael  
Abbey, Michael  
Dechichio, Daniel  
Abramson, Ian  
Ed. McGraw-Hill  
1ª. Edición  
España 1998

**ORACLE8 PROGRAMACION PL/SQL**

Urman, Scott  
Ed. MacGraw-Hill  
1ª Edición  
España 1998

**ORACLE 8i INSTALLATION GUIDE FROM HP 9000 SERVERS AND WORKSTATIONS**

ORACLE  
Estados Unidos, Febrero 1999

**UNIX FOR ORACLE DBAs**

Burleson, Donald K.  
O'Reilly & Associates  
Estados Unidos 2000

**ORACLE INTERNALS: TIPS, TRICKS AND TECHNIQUES FOR DBAs**

Burleson, Donald K.  
Auerbach Publications  
Estados Unidos, Julio 2001

**ORACLE, MANUIAL DEL ADMINISTRADOR**

Loney, Kevin  
Ed. McGraw-Hill/Interamericana de España  
1ª. Edición  
España, 1999

**UNIX Y LINUX GUIA PRACTICA**

Sánchez, Sebastián  
Ed. RA-MA  
2ª. Edición.  
México, 1999

**MICROSOFT SQL SERVER 2000, MANUAL DEL ADMINISTRADOR**

Stanek William  
Ed McGraw-Hill  
1ª Edición  
España 2000

**PROGRAMACION AVANZADA EN UNIX**

Canosa, Jose M  
Ed. McGraw-Hill  
1ª Edición  
España 1999