



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES

ARAGÓN

**“MIGRACIONES DE BASES DE DATOS ORACLE DE
LAS VERSIONES 9I Y 10G HACIA LA VERSIÓN 11G
R2”**

CASO PRACTICO

**QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN**



P R E S E N T A N :
L U I S Á L V A R O V Á Z Q U E Z
L A B R A

C A R L O S A L B E R T O
M O R A L E S R O M E R O

MÉXICO, 2012.



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.

MIGRACIÓN DE BASES DE DATOS
ORACLE DE LAS VERSIONES 9I Y 10G HACIA LA
VERSIÓN 11G R2

Con tenido

INTRODUCCIÓN.....	5
CONCEPTOS	5
¿POR QUÉ SE MIGRA?.....	7
Caducidad del soporte.	7
Nueva versión de la base de datos	8
MÉTODOS DE MIGRACIÓN DE BASE DE DATOS ORACLE	8
Sencillas	8
Complejas	9
ESTRUCTURA DEPARTAMENTAL.....	10
Capitulo 1	14
CAPITULO 1 DISEÑO.....	15
1.1 VERSIONES DE LAS BASES	15
1.2 ESTRUCTURA DE ALMACENAMIENTO.....	15
1.3 DISTRIBUCIÓN EN LOS NUEVOS SERVIDORES DE BASE DE DATOS	21
Capitulo 2	22
CAPITULO 2 ETAPAS DE PRUEBA.....	23
2.1 INSTALACIÓN DE SOFTWARE ORACLE EN LAS ZONAS.....	23
2.2 INSTALACIÓN DE ORACLE 11GR2 Y LA CREACIÓN DE GRUPOS DE DISCOS PARA ASM EN LOS SERVIDORES.....	26
2.3 INTALACIÓN DE GRID INFRAESTRUCTURE PARA ASM	37
2.4 CREACIÓN DE INSTANCIAS DE BASE DE DATOS	46
2.5 MIGRACIÓN DE LA BASE DE DATOS LOBORH.....	56
2.6 MIGRACIÓN DE LA BASE DE DATOS LOBOSYS.....	82
2.7 MIGRACIÓN DE LA BASE DE DATOS LOBORH08.....	85
2.8 MIGRACIÓN DE LA BASE DE DATOS LOBORH09.....	88
2.9 MIGRACIÓN DE LA BASE DE DATOS SIA	91
2.10 MIGRACIÓN DE LA BASE DE DATOS SIAADM	96
2.11 MIGRACIÓN DE LA BASE DE DATOS SISADM	100
2.12 MIGRACIÓN DE LA BASE DE DATOS SIAM	105
2.13 MIGRACIÓN DE LA BASE DE DATOS VIDECUFI	108
2.14 MIGRACIÓN DE LA BASE DE DATOS REMESAS	111
2.15 MIGRACIÓN DE LA BASE DE DATOS MAE	114
2.16 MIGRACIÓN DE LA BASE DE DATOS ESTAD	117
2.17 MIGRACIÓN DE LA BASE DE DATOS SIO.....	120
2.18 MIGRACIÓN DE LA BASE DE DATOS SIOBK2.....	123
Capitulo 3	126

CAPITULO 3 MIGRACIÓN A PRODUCCIÓN	127
3.1 MIGRACIÓN DE PRODUCCIÓN DE SIO	127
CONCLUSIONES.....	131

INTRODUCCIÓN

En este proyecto se va a realizar un estudio detallado y descripción de todo el procedimiento que conlleva la migración de un sistema de base de datos Oracle 9.2. y 10.2 hacia Oracle 11.2.

Además de la migración en sí se plantea el problema con otros condicionantes. Se plantea realizar, además de la migración de la base de datos, un cambio de almacenamiento, inicialmente en file system, a Automatic Storage Management (ASM).

Para conseguir estos objetivos se configurarán dos servidores con S.O Solaris y que a su vez cada uno estará dividido en dos zonas, en las que se instalará y configurará la nueva versión de Base de Datos.

Del mismo modo, se aporta información sobre las razones por las que la empresa nos solicita realizar dicha migración.

CONCEPTOS

Antes de introducirnos a lo que son las migraciones de bases de datos Oracle es importante mencionar algunos conceptos de interés para una mejor comprensión del proyecto y para, en un caso hipotético, utilizarlo como referencia si no se recuerda algo en concreto durante la lectura del mismo.

Se hace especial hincapié en que los conceptos son relacionados con Oracle. La forma en la que se recomienda que se organice su software, elementos típicos en sus instalaciones y, en especial, las herramientas que se van a utilizar a lo largo de todo este trabajo.

-Base de datos

Una base de datos es una colección de datos tratados como una unidad. El objetivo de una base de datos es almacenar y recuperar información relacionada. En general, un servidor gestiona de forma fiable una gran cantidad de datos en un entorno multiusuario en el cual muchos de estos pueden acceder simultáneamente a los mismos datos. Todo esto es cumplido mientras se proporciona un rendimiento de alto nivel. Además, una base de datos proporciona también prevención de accesos no autorizados y soluciones eficientes para errores de recuperación.

-Automatic Storage Management – ASM

ASM automatiza y simplifica la distribución de los datafiles, control files y log files. Los ficheros de la BD son automáticamente distribuidos por todos los discos disponibles, y el almacenamiento de la BD es rebalanceado siempre que la configuración del almacenamiento cambia. Proporciona redundancia a través de mirror (espejo o duplicación) de los ficheros de la BD, e incrementa su rendimiento mediante la automatización de la distribución de los archivos de la BD por todos los discos disponibles.

-Tablespace:

Un tablespace es una unidad lógica de almacenamiento dentro de una base de datos Oracle. Es lógico, porque el tablespace no es visible en el sistema de archivos de la máquina en la que reside la base de datos. Un tablespace, a su vez, se compone de al menos un datafile que, a su vez, se encuentren físicamente en el sistema de archivos del servidor.

-Datafiles:

Todas las bases de datos Oracle tienen al menos un datafile físico. Los datafiles contienen todos los datos de la base de datos. Los datos de las estructuras lógicas de la base de datos, tales como las tablas o los índices, están físicamente almacenados en datafiles asignados para la base de datos.

Las características de los datafiles son:

- Un datafile sólo puede ser asociado con una única base de datos.
- Los datafiles pueden tener ciertas características para que automáticamente se extiendan cuando la base de datos se quede sin espacio.
- Uno o más datafiles forman una unidad lógica de almacenamiento llamada tablespace.

-Instancia de Base de datos

La base de datos Oracle está compuesta por un conjunto de archivos del sistema operativo que contienen datos introducidos por usuarios o aplicaciones e información estructural acerca de la base de datos en sí misma llamados metadata. La información es almacenada de forma persistente en estos archivos.

Para permitir a los usuarios o aplicaciones ver o actualizar los datos en la base de datos, Oracle debe levantar un conjunto de procesos, llamados procesos de background, y asignar una cantidad de memoria para ser utilizada durante la operación con la base de datos. Los procesos de background y la memoria asignada por Oracle conjuntamente forman la instancia. Una instancia debe ser levantada para leer y escribir información en la base de datos.

Cuando la instancia de la base de datos no está disponible, los datos están a salvo en la base de datos pero no se puede acceder a ellos a través de usuarios ni de aplicaciones.

Las propiedades de una instancia de base de datos están especificadas utilizando los parámetros de inicialización de la instancia. Cuando la instancia es levantada, un archivo de parámetros de inicialización es leído y la instancia es configurada de acuerdo a éste.

-ORACLE_HOME

El ORACLE_HOME es una variable de ambiente que se utiliza para establecer y definir la ruta de acceso de inicio de Oracle.

El directorio ORACLE_HOME tendrá los subdirectorios, binarios, archivos ejecutables, programas, scripts, etc. para la base de datos Oracle.

Este directorio puede utilizarse por cualquier usuario que tenga permisos de utilizar la base de datos determinada.

Si la variable ORACLE_HOME se define como una variable de ambiente, entonces durante el proceso de instalación, la ruta de inicio en Oracle se establecerá en el directorio definido como predeterminado. Si la variable no está definida, el Oracle tendrá su propia ubicación predeterminada.

-ORACLE_BASE:

El ORACLE_BASE también es una variable de entorno para definir el directorio de nivel de base/raíz donde tendrá el árbol de directorios de bases de datos Oracle: ORACLE_HOME definido bajo el directorio ORACLE_BASE.

-ORACLE_SID:

El Sistema de Identificación de Oracle ó Oracle System ID(SID) se utiliza para identificar de forma exclusiva una determinada base de datos en un sistema. Por esta razón, uno no puede tener más de una base de datos con el mismo SID en un sistema informático.

-oraInventory:

El oraInventory, contiene la información relacionada con todos los productos Oracle instalados en un servidor. Su presencia no es requerida para la operatividad del software pero sí cuando queremos actualizar el software, por ello es una buena práctica respaldarlo cada vez que instalemos o actualicemos el software Oracle.

-DDL

Un lenguaje de definición de datos (Data Definition Language, DDL por sus siglas en inglés) es un lenguaje proporcionado por el sistema de gestión de base de datos que permite a los usuarios de la misma llevar a cabo las tareas de definición de las estructuras que almacenarán los datos así como de los procedimientos o funciones que permitan consultarlos.

¿POR QUÉ SE MIGRA?

Las razones que llevan a migrar una base de datos que está trabajando a pleno rendimiento en todos los entornos se pueden resumir en dos: caducidad del soporte y aparición de nuevas versiones de la base de datos con nuevas mejoras y funcionalidades que, se sobreentiende, proporcionan una mejora en el rendimiento y mayores funcionalidades.

Caducidad del soporte.

Oracle ofrece una serie de servicios para los productos que tiene soportados. Todos estos servicios enfocados a la infraestructura de IT de la empresa que los contrata para soportar las demandas de su negocio.

Todos los productos tienen una caducidad en cuanto a soporte se refiere. Esto hace referencia a varios conceptos. Por un lado se hablaría de las actualizaciones y desarrollo de mejoras del producto contratado y por otro de los servicios de soporte propiamente dichos.

Anualmente se puede hacer actualizaciones o upgrades de los productos contratados mientras se mantiene el soporte. Es decir, Oracle se compromete a trabajar en la mejora constante del producto contratado hasta que aparezca una nueva versión que de soporte a la actual.

Mientras se mantenga este soporte, se proporciona el servicio de resolución de

problemas 24 x 7 (7 días - 24 horas) a través de Metalink (procedimiento de contacto por defecto) y a través del teléfono ó email (sólo en ocasiones excepcionales). Estos dos procedimientos son el medio de contacto con los analistas que la empresa dispone para la resolución de los distintos casos que pueden aparecer y que proporcionan una respuesta basada en el estudio del caso concreto y de las mejores prácticas recomendadas.

Una vez que haya aparecido una nueva versión de un producto estos servicios de soporte se mantendrán durante un tiempo para que las empresas que lo consideren puedan hacer una transacción hacia la nueva versión. Una vez pasado el período los servicios de actualización de software y de soporte se suspenderán.

Nueva versión de la base de datos

La aparición de nuevas versiones en el software es algo inevitable y con ellas las nuevas funcionalidades y posibilidades para el rendimiento de nuestra base de datos y, en consecuencia, de nuestros sistemas.

Con estas nuevas versiones, las compañías que utilizan el producto, buscan un mayor rendimiento de sus sistemas, por lo que deciden llevar sus bases de datos a la versión más reciente si es que sus aplicaciones lo permiten.

MÉTODOS DE MIGRACIÓN DE BASE DE DATOS ORACLE

Oracle provee muchos métodos para realizar esta tarea, de nuevo, todo depende de como la queramos hacer y depende de los requerimientos que tengamos. Pero existen soluciones sencillas y soluciones complejas, vamos a ver algunas de ellas de manera general y sin entrar en los detalles de cada una.

Sencillas

Copia Directa

Suponiendo que el nuevo servidor es exactamente igual que el anterior respecto a versiones de software (sistema operativo, base de datos, parches, etc) en teoría si copiamos todos los archivos de datos, control files, redo logs, etc la base de datos debería de funcionar correctamente. Personalmente yo no utilizaría esta opción por que siempre falta algo, un parche, un parámetro, etc.

Export/Import (DataPump en 10g)

Este podría ser también complejo pero normalmente se hace un export del usuario(s) de la aplicación y se importa en la otra base de datos, creo que es de las más sencillas que puede existir además que podemos hacer el export de una versión anterior a la nueva base de datos por ejemplo de 9i a 10g. El único problema de esta opción es que debemos tener más o menos el mismo espacio en disco que la base de datos actual para poder hacer el export; por ejemplo en una base de datos de 10Gb tal vez si es conveniente hacerlo pero en una de 600Gb o de Tb ya no es tan práctico hacerlo, y si además se tiene que transferir el archivo(s) del export pues sería muchísimo más tardado.

Complejas

RMAN Duplicate.

RMAN es una utilitaria de Oracle para realizar backups/restore. Una opción que es muy interesante es el "duplicate" y como su nombre lo dice duplica una base de datos a otro servidor. Básicamente como funciona es de la siguiente manera: se hace un backup de la base de datos "fuente/original", una vez con el backup guardado en el tape/disco/etc el resto del trabajo se hace en el servidor "destino/nuevo", en el destino se debe tener el software de Oracle instalado, hay que configurar el tnsnames.ora para tener acceso a la base "fuente", se copia el init.ora del fuente al destino (para tener la misma configuración) y se hacen los cambios necesarios al archivo (cambiar el destino de los control files, memoria, etc), se agregan al init.ora 2 parámetros para el duplicate (db_file_name_convert y log_file_name_convert), se crea el spfile y se crea un script de RMAN para duplicar la base de datos. El script se ejecuta en la base de datos destino y básicamente el script o RMAN buscan el backup que hicimos de la fuente y hace un restore al destino de todos los archivos, hace un recover de la base hasta la hora/scn/etc que le digamos y deja la nueva base de datos lista para trabajar con ella. Personalmente este es un método preferido para hacer "refresh" de ambientes por ejemplo de producción a QA o desarrollo, es rápido (bueno depende de ciertos factores red/servidor/discos/etc) y sencillo ya que una vez teniendo los scripts es solo cuestión de ejecutarlos y RMAN se encarga de todo.

NOTA.- Muchísimos pasos no son mencionados y cierta configuración se tiene que hacer para que esto funcione correctamente.

Backup/Restore

Es muy similar al Duplicate pero casi todos los pasos los tenemos que hacer manualmente, hacer un backup del fuente, hacer el restore en el destino, recrear control files, hacer recover, etc.

Copy a nivel SAN.

Esta opción es ideal cuando se tienen bases de datos grandes (GB a TB+), en esencia la copia de la información se hace a nivel de hardware/discos. Si la base de datos fuente y destino están en una SAN entonces solo hay que copiar los discos que contienen la base de datos fuente a los discos que van a contener la base de datos destino. Todo esto se hace por medio del sistema operativo y regularmente lo hacen los encargados de los servidores. La participación del DBA en esta opción es mínima, solo hay que poner los tablespaces en la base de datos fuente en modo backup, se hace la copia a nivel hardware, una vez terminada la copia se quita el modo backup de la fuente, se envían los archivos del archive de la fuente al destino (de preferencia se hace un log switch) y en el destino se recrean los control files si es necesario y se hace un recover (aquí va a preguntar por los archives que acabamos de copiar) y listo, la BD queda funcional.

Estas son algunas de las formas en que se puede migrar, copiar, mover y recuperar una base de datos.

PRESENTACION DEL PROYECTO

Las migraciones que a continuación se detallan, fueron elaboradas para la CONDUSEF, y se decide hacerlo de forma lógica (import/export) ya que después de hacer el análisis se llega a la conclusión de que la forma más rápida para hacerlo es mediante la migración de los schemas que son utilizados por sus aplicaciones, así,

toda la basura que existe en sus bases de datos debido a la creación de algunos objetos para pruebas fue eliminada al pasar a la nueva versión en producción.

Para hacer notar la importancia del proyecto, se agrega una pequeña reseña de lo que es la CODUSEF:

La Comisión Nacional para la Protección y Defensa de los Usuarios de Servicios Financieros, es un Organismo público descentralizado, que protege los derechos de los usuarios de servicios financieros. Al mismo tiempo, fomenta el desarrollo de una cultura educativa en materia de operación y alcances de las Instituciones Financieras.

La Condusef funciona como intermediario, cuando existen diferencias en la interpretación de los lineamientos o compromisos estipulados al adquirir un servicio o un producto financiero.

Atiende reclamaciones relacionadas a las prácticas indebidas de las Instituciones Financieras.

Verifica que la información utilizada para la publicidad de los servicios y productos financieros sea veraz y no induzca al consumo mediante interpretaciones erróneas.

Así mismo, pone a disposición de los usuarios gran cantidad de información relacionada a los productos y servicios financieros; proporciona otras herramientas como estadísticas, cuadros comparativos de Instituciones bancarias, Afores, Aseguradoras y otras compañías del sector bursátil; al igual que simuladores y calculadoras.

ESTRUCTURA DEPARTAMENTAL

La siguiente tabla muestra la estructura organizacional de CONDUSEF destacando el Departamento de Redes y Telecomunicaciones, el cual es el encargado de administrar las bases de datos que van a ser migradas.

CVE_AREA	CVE_AREA_SUP	DESCRIPCION
2000000		PRESIDENCIA
2010000	2000000	SECRETARIA TECNICA
2100000	2000000	SECRETARIA DE LA JUNTA DE GOBIERNO
2110000	2100000	PROSECRETARIA DE LA JUNTA DE GOBIERNO
2200000	2000000	VICEPRESIDENCIA TECNICA
2210000	2200000	DIRECCION GENERAL DE EDUCACION FINANCIERA
2211000	2210000	DIRECCION DE ANALISIS DE SERVICIOS Y PRODUCTOS FINANCIEROS
2211001	2211000	DEPARTAMENTO DE ESTUDIOS DE SECTOR BANCARIO Y BURSATIL
2211002	2211000	DEPARTAMENTO DE ESTUDIOS SECTOR AHORRO Y CREDITO NO BANCARIO
2212000	2210000	DIRECCION DE PROMOCION Y DESARROLLO EDUCATIVO
2212001	2212000	DEPARTAMENTO DE CONTENIDOS EDUCATIVOS
2212002	2212000	DEPARTAMENTO DE PUBLICACIONES
2212003	2212000	DEPARTAMENTO DE VINCULACION Y EVENTOS
2213000	2210000	DIRECCION DE COMUNICACION SOCIAL
2213001	2213000	DEPARTAMENTO DE COMUNICACION INTERNA Y DISTRIBUCION
2213002	2213000	DEPARTAMENTO DE DESARROLLO DE MEDIOS DE PROMOCION
2220000	2200000	DIRECCION GENERAL DE DESARROLLO FINANCIERO, ESTADISTICO Y OPERATIVO
2221000	2220000	DIRECCION DE VALUACION Y DESARROLLO FINANCIERO

2221001	2221000	DEPARTAMENTO DE VALUACION CONTABLE Y FINANCIERA
2221002	2221000	DEPARTAMENTO DE ANALISIS FINANCIERO
2221003	2221000	DEPARTAMENTO DE ADMINISTRACION DE SIMULADORES DE SERVICIOS FINANCIEROS
2222000	2220000	DIRECCION DE DESARROLLO Y EVALUACION DEL PROCESO OPERATIVO
2222001	2222000	DEPARTAMENTO DE REGISTRO DE INSTITUCIONES FINANCIERAS
2222002	2222000	DEPARTAMENTO DE DESARROLLO DE HERRAMIENTAS
2222003	2222000	DEPARTAMENTO DE CALIDAD EN EL SERVICIO DE ATENCION A USUARIOS
2222004	2222000	DEPARTAMENTO DE PROCESOS CON INSTITUCIONES FINANCIERAS
2223000	2220000	DIRECCION DE INFORMACION Y DESARROLLO ESTADISTICO
2223001	2223000	DEPARTAMENTO DE INDICES DE RECLAMACION
2223002	2223000	DEPARTAMENTO DE INFORMACION OPERATIVA
2223003	2223000	DEPARTAMENTO DE INFORMACION FINANCIERA
2230000	2200000	DIRECCION GENERAL DE EVALUACION Y VIGILANCIA
2231000	2230000	DIRECCION DE EVALUACION DE PRODUCTOS DE CREDITO Y CAPTACION
2231001	2231000	DEPARTAMENTO DE BANCOS Y SOFOMES
2231002	2231000	DEPARTAMENTO DE ORGANIZACIONES Y ACTIVIDADES AUXILIARES Y CREDITO POPULAR
2232000	2230000	DIRECCION DE EVALUACION DE PRODUCTOS DE SEGUROS, FIANZAS Y AHORRO
2232001	2232000	DEPARTAMENTO DE SEGUROS Y FIANZAS
2232002	2232000	DEPARTAMENTO DE AHORRO Y BURSATIL
2233000	2230000	DIRECCION DE VIGILANCIA, CUMPLIMIENTO Y ENLACE CON AUTORIDADES
2233001	2233000	DEPARTAMENTO DE CUMPLIMIENTO
2233002	2233000	DEPARTAMENTO DE ENLACE INSTITUCIONAL
2300000	2000000	VICEPRESIDENCIA JURIDICA
2310000	2300000	DIRECCION GENERAL DE DICTAMINACION Y SUPERVISION
2310100	2310000	SUBDIRECCION DE INSPECCION
2311000	2310000	DIRECCION DE VERIFICACION, CONTROL NORMATIVO Y UNIDADES ESPECIALIZADAS
2311001	2311000	DEPARTAMENTO DE ENLACE CON UNIDADES ESPECIALIZADAS
2311002	2311000	DEPARTAMENTO DE CONTROL NORMATIVO
2312000	2310000	DIRECCION DE DICTAMINACION
2312001	2312000	DEPARTAMENTO DE DICTAMINACION EN MATERIA DE BANCOS Y ORGANIZACIONES AUXILIARES
2312002	2312000	DEPARTAMENTO DE DICTAMINACION EN MATERIA DE SEGUROS Y FIANZAS
2312003	2312000	DEPARTAMENTO DE DICTAMINACION EN MATERIA DE SAR
2312004	2312000	DEPARTAMENTO DE CONTROL Y SEGUIMIENTO DE DICTAMENES
2320000	2300000	DIRECCION GENERAL DE DEFENSORIA, INTERVENTORIA Y CONSULTIVA
2321000	2320000	DIRECCION DE DEFENSA A USUARIOS
2321001	2321000	DEPARTAMENTO DE DEFENSORIA EN MATERIA DE BANCOS
2321002	2321000	DEPARTAMENTO DE DEFENSORIA EN MATERIA DE SEGUROS Y FIANZAS
2321003	2321000	DEPARTAMENTO DE DEFENSORIA EN MATERIA DE ORGANIZACIONES Y ACTIVIDADES AUXILIARES DE CREDITO
2321004	2321000	DEPARTAMENTO DE ASUNTOS ADMINISTRATIVOS, LABORALES Y FAMILIARES
2322000	2320000	DIRECCION DE INTERVENTORIA Y ASUNTOS PENALES
2322001	2322000	DEPARTAMENTO DE PROCEDIMIENTOS
2322002	2322000	DEPARTAMENTO DE DENUNCIAS
2323000	2320000	DIRECCION CONSULTIVA
2323001	2323000	DEPARTAMENTO DE PROCEDENCIA
2323002	2323000	DEPARTAMENTO DE CONSULTAS
2330000	2300000	DIRECCION GENERAL DE SERVICIOS LEGALES
2331000	2330000	DIRECCION DE ARBITRAJE Y SANCIONES
2331001	2331000	DEPARTAMENTO DE SANCIONES A INSTITUCIONES DE CREDITO Y ORGANIZACIONES AUXILIARES DE CREDITO
2331002	2331000	DEPARTAMENTO DE ARBITRAJE Y CONDONACION DE MULTAS
2331003	2331000	DEPARTAMENTO DE SANCIONES A SOFOMES Y OTROS INTERMEDIARIOS

		FINANCIEROS
2332000	2330000	DIRECCION CONTENCIOSA
2332001	2332000	DEPARTAMENTO DE REMATE DE VALORES
2332002	2332000	DEPARTAMENTO DE AMPAROS Y APELACIONES
2332003	2332000	DEPARTAMENTO DE CONTROL LEGAL
2332004	2332000	DEPARTAMENTO DE DEFENSA LEGAL Y DENUNCIAS PENALES
2333000	2330000	DIRECCION DE RECURSOS DE REVISION
2333001	2333000	DEPARTAMENTO DE RECURSOS DE LA LEY DE PROTECCION Y DEFENSA AL USUARIO DE SERVICIOS FINANCIEROS
2333002	2333000	DEPARTAMENTO DE RECURSOS DE LA LEY DE TRANSPARENCIA Y ORDENAMIENTO DE LOS SERVICIOS FINANCIEROS
2333003	2333000	DEPARTAMENTO DE RECURSOS DE LA LEY DE INSTITUCIONES DE CREDITO
2334000	2330000	DIRECCION DE DISPOSICIONES, CONVENIOS Y CONTRATOS
2334001	2334000	DEPARTAMENTO DE REGISTRO DE CONTRATOS
2334002	2334000	DEPARTAMENTO DE ANALISIS LEGAL
2334003	2334000	DEPARTAMENTO DE DISPOSICIONES Y DESARROLLO NORMATIVO INTERNO
2334004	2334000	DEPARTAMENTO DE DISPOSICIONES Y DESARROLLO NORMATIVO EXTERNO
2400000	2000000	VICEPRESIDENCIA DE DELEGACIONES
2410000	2400000	DIRECCION GENERAL DE DELEGACIONES NORTE SUR
2410101	2410000	DELEGACION BAJA CALIFORNIA
2410102	2410000	DELEGACION BAJA CALIFORNIA SUR
2410103	2410000	DELEGACION CAMPECHE
2410104	2410000	DELEGACION CHIAPAS
2410105	2410000	DELEGACION CHIHUAHUA
2410106	2410000	DELEGACION CD. JUAREZ
2410107	2410000	DELEGACION GUERRERO
2410108	2410000	DELEGACION NUEVO LEON
2410109	2410000	DELEGACION OAXACA
2410110	2410000	DELEGACION QUINTANA ROO
2410111	2410000	DELEGACION SINALOA
2410112	2410000	DELEGACION SONORA
2410113	2410000	DELEGACION TABASCO
2410114	2410000	DELEGACION TAMAULIPAS
2410115	2410000	DELEGACION VERACRUZ
2410116	2410000	DELEGACION YUCATAN
2411000	2410000	DIRECCION DE ENLACE Y ANALISIS REGIONAL NORTE SUR
2420000	2400000	DIRECCION GENERAL DE DELEGACIONES CENTRO OCCIDENTE
2420101	2420000	DELEGACION AGUASCALIENTES
2420102	2420000	DELEGACION COAHUILA
2420103	2420000	DELEGACION COLIMA
2420104	2420000	DELEGACION DURANGO
2420105	2420000	DELEGACION EDO. MEXICO
2420106	2420000	DELEGACION GUANAJUATO
2420107	2420000	DELEGACION HIDALGO
2420108	2420000	DELEGACION JALISCO
2420109	2420000	DELEGACION METROPOLITANA NORTE
2420110	2420000	DELEGACION METROPOLITANA ORIENTE
2420111	2420000	DELEGACION METROPOLITANA SUR
2420112	2420000	DELEGACION MICHOACAN
2420113	2420000	DELEGACION MORELOS
2420114	2420000	DELEGACION NAYARIT
2420115	2420000	DELEGACION PUEBLA
2420116	2420000	DELEGACION QUERETARO

2420117	2420000	DELEGACION SAN LUIS POTOSI
2420118	2420000	DELEGACION TLAXCALA
2420119	2420000	DELEGACION ZACATECAS
2421000	2420000	DIRECCION DE ENLACE Y ANALISIS REGIONAL CENTRO OCCIDENTE
2422000	2420000	DELEGACION METROPOLITANA CENTRAL
2422001	2422000	DEPARTAMENTO DE ATENCION TELEFONICA Y ELECTRONICA
2422002	2422000	DEPARTAMENTO DE CONCILIACION EN MATERIA DE INTERMEDIARIOS FINANCIEROS BANCARIOS Y NO BANCARIOS
2422003	2422000	DEPARTAMENTO DE CONCILIACION EN MATERIA DE SEGUROS, FIANZAS Y SAR
2422004	2422000	DEPARTAMENTO DE ATENCION PERSONALIZADA
2500000	2000000	VICEPRESIDENCIA DE PLANEACION Y ADMINISTRACION
2510000	2500000	DIRECCION GENERAL DE PLANEACION
2510100	2510000	SUBDIRECCION DE INFORMATICA Y TELECOMUNICACIONES
2510101	2510100	DEPARTAMENTO DE SOPORTE TECNICO
2510102	2510100	DEPARTAMENTO DE REDES Y TELECOMUNICACIONES
2510103	2510100	DEPARTAMENTO DE DESARROLLO Y MANTENIMIENTO DE SISTEMAS
2510104	2510100	DEPARTAMENTO DE MONITOREO Y VIGILANCIA CCTV Y CAT
2511000	2510000	DIRECCION DE FINANZAS
2511001	2511000	DEPARTAMENTO DE TESORERIA
2511002	2511000	DEPARTAMENTO DE CONTROL PRESUPUESTAL
2512000	2510000	DIRECCION DE PROGRAMACION Y CONTROL
2512001	2512000	DEPARTAMENTO DE CONTABILIDAD
2512002	2512000	DEPARTAMENTO DE PLANEACION Y CONTROL DE INFORMACION
2520000	2500000	DIRECCION GENERAL DE PERSONAL Y ORGANIZACION
2520100	2520000	SUBDIRECCION DE ORGANIZACION Y SELECCION DE PERSONAL
2520101	2520100	DEPARTAMENTO DE ESTUDIOS ORGANIZACIONALES
2520102	2520100	DEPARTAMENTO DE METODOS Y PROCEDIMIENTOS
2520103	2520100	DEPARTAMENTO DE SELECCION DE PERSONAL
2521000	2520000	DIRECCION DE ADMINISTRACION DE PERSONAL
2521100	2521000	SUBDIRECCION DE PRESTACIONES Y GESTION DE PUESTOS
2521101	2521100	DEPARTAMENTO DE PAGOS Y TABULADORES
2521102	2521100	DEPARTAMENTO DE ANALISIS Y GESTION DE PUESTOS
2522000	2520000	DIRECCION DE DESARROLLO DE PERSONAL
2522100	2522000	SUBDIRECCION DE CAPACITACION
2530000	2500000	DIRECCION GENERAL DE BIENES Y SERVICIOS
2531000	2530000	DIRECCION DE ADQUISICIONES
2531001	2531000	DEPARTAMENTO DE ALMACEN E INVENTARIOS
2531002	2531000	DEPARTAMENTO DE LICITACIONES, CONTRATACIONES Y COMPRAS
2532000	2530000	DIRECCION DE SERVICIOS GENERALES
2532001	2532000	DEPARTAMENTO DE ADMINISTRACION Y CONTROL DE SERVICIOS
2532002	2532000	DEPARTAMENTO DE MANTENIMIENTO Y CONSERVACION
2533000	2530000	DIRECCION DE GESTION Y CONTROL DOCUMENTAL
2533100	2533000	SUBDIRECCION DE GESTION DOCUMENTAL
2533101	2533100	DEPARTAMENTO DE ADMINISTRACION DOCUMENTAL
2600000	2000000	ORGANO INTERNO DE CONTROL
2610000	2600000	AREA DE AUDITORIA PARA EL DESARROLLO Y MEJORA DE LA GESTION PUBLICA
2620000	2600000	AREA DE AUDITORIA INTERNA
2630000	2600000	AREA DE RESPONSABILIDADES Y QUEJAS

Ahora se sabe que todo el desarrollo del trabajo es relacionado a la CONDUSEF, por lo que no será necesario mencionarlo durante la presentación de las migraciones.

Capítulo 1

DISEÑO

DISEÑO

En este capítulo se mostrara toda la información referente a las instancias de bases de datos, que van a ser migradas, como lo es la versión del motor de cada base existente, la estructura de almacenamiento así como la estructura de distribución de los nuevos servidores con relación a la instancia de base de datos a migrar.

1.1 VERSIONES DE LAS BASES

Las siguientes tablas muestran las diferentes versiones de las bases de datos del cliente que se desean migrar a la última versión de oracle, las cuales estarán distribuidas en dos servidores y cada servidor estará virtualizado en dos zonas.

DOMINIO 0

Zona 1

No	Instancia	Versión
1	SIOBK2	10.2.0.3

Zona 2

No	Instancia	Versión
1	SIO	10.1.0.2.0
2	VIDECUFI	10.1.0.2.0
3	REMESAS	10.1.0.2.0
4	ESTAD	10.1.0.2.0
5	SIAM	10.1.0.2.0
6	MAE	10.1.0.2.0

DOMINIO 1

Zona 1

No	Instancia	Versión
1	LOBORH	9.2.0.6
2	LOBOSYS	9.2.0.6
3	LOBORH08	9.2.0.6
4	LOBORH09	9.2.0.6

Zona 2

No	Instancia	Versión
1	SIA	9.2.0.6
2	CONBAN	9.2.0.6
3	SIAADM	9.2.0.6
4	ADQUI	9.2.0.6
5	SRCPROD	9.2.0.6

Nota: Todas las instancias se migraran a la versión 11gR2 de oracle con almacenamiento en ASM.

1.2 ESTRUCTURA DE ALMACENAMIENTO

Las tablas que a continuación se muestran detallan el nombre de los Tablespace, Mb usados hasta el momento, Mb libres, porcentaje usado y el tamaño estimado de disco duro destinado para los grupos de ASM, tomando en cuenta el crecimiento de las bases de datos para que en la migración de cada una de ellas se tenga el triple del tamaño actualmente utilizado y así evitar problemas de espacio en un lapso de tiempo estimado a tres años.

SIOBK2

TABLESPACE_NAME	MB_USED	MB_FREE	PERCENT_USED	SIZE_ASM
USER	241.25	1	99.59	723.75
SYSTEM	550	50.0625	90.9	1650
SYSAUX	490	60.75	87.6	1470
SIODATA	12288	1877.8125	84.72	36864
UNDOTBS1	510	439.1875	13.88	1530
TOTAL	14079.25	2428.81	75.34	42237.75

LOBORH

TABLESPACE_NAME	MB_USED	MB_FREE	PERCENT_USED	SIZE_ASM
EXAMPLE	148.75	0.125	99.92	446.25
SYSTEM	580	2.5625	99.56	1740
XDB	46.875	2.625	94.4	140.625
CWMLITE	20	2.25	88.75	60
PAO_DATA	50	19.6875	60.63	150
PAO_IDX	50	22.5	55	150
PERFSTAT	500	241.1875	51.76	1500
DRSYS	20	10.3125	48.44	60
ODM	20	10.6875	46.56	60
LOBORH	5626.8125	3416.4375	39.28	16880.4375
USERS	25	21.0625	15.75	75
UNDOTBS1	7089	6322.4375	10.81	21267
UNDOTBS02	8680	8345.5625	3.85	26040
TOOLS	10	9.9375	0.63	30
INDX	25	24.9375	0.25	75
TOTAL	22891.44	18452.3125	47.71	68674.3125

LOBOSYS

TABLESPACE_NAME	MB_USED	MB_FREE	PERCENT_USED	SIZE_ASM
EXAMPLE	148.75	0.125	99.92	446.25
PERFSTAT	500	1.1875	99.76	1500
SYSTEM	430	8.9375	97.92	1290
XDB	46.875	2.625	94.4	140.625
CWMLITE	20	2.25	88.75	60
DRSYS	20	10.3125	48.44	60
ODM	20	10.6875	46.56	60
UNDOTBS1	410	392.6875	4.22	1230
TOOLS	10	9.9375	0.63	30
INDX	25	24.9375	0.25	75
USERS	25	24.9375	0.25	75
LOBOSYS	1024	1023.375	0.06	3072
TOTAL	2679.625	1512	48.43	8038.875

LOBO08

TABLESPACE_NAME	MB_USED	MB_FREE	PERCENT_USED	SIZE_ASM
LOBORH	3178.5625	1.75	99.94	9535.6875
EXAMPLE	148.75	0.125	99.92	446.25
SYSTEM	530	5.6875	98.93	1590
XDB	46.875	2.625	94.4	140.625
CWMLITE	20	2.25	88.75	60
DRSYS	20	10.3125	48.44	60
ODM	20	10.6875	46.56	60
UNDOTBS1	540	523.6875	3.02	1620
TOOLS	10	9.9375	0.63	30
INDX	25	24.9375	0.25	75
USERS	25	24.9375	0.25	75
TOTAL	4564.188	616.9375	52.83	13692.5625

LOBO09

TABLESPACE_NAME	MB_USED	MB_FREE	PERCENT_USED	SIZE_ASM
EXAMPLE	148.75	0.125	99.92	446.25
SYSTEM	520	1.25	99.76	1560
XDB	46.875	2.625	94.4	140.625
CWMLITE	20	2.25	88.75	60
LOBORH	3408	984.3125	71.12	10224
DRSYS	20	10.3125	48.44	60
ODM	20	10.6875	46.56	60
TOOLS	10	9.9375	0.63	30
UNDOTBS1	3395	3376.6875	0.54	10185
INDX	25	24.9375	0.25	75
USERS	25	24.9375	0.25	75
TOTAL	7638.625	4448.0625	50.056	22915.875

SIO

TABLESPACE_NAME	MB_USED	MB_FREE	PERCENT_USED	SIZE_ASM
SIO_INDEX	11902	5.875	99.95	35706
USERS	75.75	0.6875	99.09	227.25
SYSAUX	1120	81.5625	92.72	3360
SIO_DATA	9299	712.6875	92.34	27897
STAT_DATA	150	20.1875	86.54	450
UNDOTBS_SIO	120	20	83.33	360
PA_DATA	50	19.625	60.75	150
SYSTEM	1000	445.5625	55.44	3000
PA_INDX	50	22.3125	55.38	150
UNDOTBS1	2048	1765.375	13.8	6144
UNDO_TEMP_SIO	100	98.6875	1.31	300
SIOUNE_DATA	512	511.9375	0.01	1536
TOTAL	26426.75	3704.5	61.72	79280.25

VIDECUFI

TABLESPACE_NAME	MB_USED	MB_FREE	PERCENT_USED	SIZE_ASM
SYSTEM	460	9.125	98.02	1380
SYSAUX	930	70.0625	92.47	2790
DESACUFI_DATA	100	60.125	39.88	300
USERS	5	4.5625	8.75	15
VIDECUFI_DATA	150	148.6875	0.88	450
UNDOTBS1	6795	6779.5625	0.23	20385
VIDECUFI_INDEX	50	49.9375	0.13	150
TOTAL	8490	7122.06	34.34	25470

REMESAS

TABLESPACE_NAME	MB_USED	MB_FREE	PERCENT_USED	SIZE_ASM
SYSTEM	450	7.9375	98.24	1350
SYSAUX	900	54.25	93.97	2700
REMESAS_DATA	100	89.1875	10.81	300
USERS	5	4.5625	8.75	15
REMESAS_INDEX	25	24.9375	0.25	75
UNDOTBS1	5225	5216.1875	0.17	15675
TOTAL	6705	5397.06	35.37	20115

SIAM

TABLESPACE_NAME	MB_USED	MB_FREE	PERCENT_USED	SIZE_ASM
SYSTEM	450	4.3125	99.04	1350
SYSAUX	340	5.5625	98.36	1020
UNDOTBS1	25	17.5625	29.75	75
SIAM	100	91.125	8.88	300
USERS	5	4.5625	8.75	15
TOTAL	920	123.13	48.96	2760

MAE

TABLESPACE_NAME	MB_USED	MB_FREE	PERCENT_USED	SIZE_ASM
SYSTEM	450	6.375	98.58	1350
SYSAUX	340	18.25	94.63	1020
UNDOTBS1	25	17.375	30.5	75
USERS	5	4.5625	8.75	15
MAEDATA	10240	10239.125	0.01	30720
TOTAL	11060	10285.69	46.49	33180

ESTAD

TABLESPACE_NAME	MB_USED	MB_FREE	PERCENT_USED	SIZE_ASM
SYSTEM	450	8.1875	98.18	1350
SYSAUX	330	50.8125	84.6	990
ESTAD_DATA	500	358.625	28.28	1500
USERS	5	4.5625	8.75	15
UNDOTBS1	125	114.5625	8.35	375
TOTAL	1410	536.75	45.632	4230

SIA

TABLESPACE_NAME	MB_USED	MB_FREE	PERCENT_USED	SIZE_ASM
EXAMPLE	148.75	0.125	99.92	446.25
TARIFICA	249	3.6875	98.52	747
SYSTEM	480	21.375	95.55	1440
XDB	46.875	2.625	94.4	140.625
SIADATA_M	500	51.125	89.78	1500
CWMLITE	20	2.25	88.75	60
SICOFIP_DATA	700	81	88.43	2100
SIADATA_H	700	191.4375	72.65	2100
ODM	15	5.6875	62.08	45
USERS	400	160.5625	59.86	1200
STAT1_DTA	500	231.1875	53.76	1500
DRSYS	20	10.3125	48.44	60
SIADATA_S	50	26.9375	46.13	150
SIADATA_F	50	30.125	39.75	150
SICOFIP_INDEX	600	385.25	35.79	1800
UNDOTBS1	415	390.6875	5.86	1245
GESTION_DATA	20	19.5	2.5	60
TOOLS	10	9.9375	0.63	30
INDX	25	24.9375	0.25	75
FOGLIGHT	100	99.9375	0.06	300
SIADATA_RH	100	99.9375	0.06	300
MV04	512	511.9375	0.01	1536
USER04	512	511.9375	0.01	1536
STAT_DATA	512	511.9375	0.01	1536
INDX04	1024	1023.938	0.01	3072
TOTAL	7709.63	4408.4	43.33	23128.875

CONBAN

TABLESPACE_NAME	MB_USED	MB_FREE	PERCENT_USED	SIZE_ASM
EXAMPLE	148.75	0.125	99.92	446.25
PERFSTAT	500	1.8125	99.64	1500
SYSTEM	430	7.8125	98.18	1290
XDB	46.875	2.625	94.4	140.625
CWMLITE	20	2.25	88.75	60
CONBAN_DATA	80	15.0625	81.17	240
DRSYS	20	10.3125	48.44	60
ODM	20	10.6875	46.56	60
SIIE	100	65.6875	34.31	300
UNDOTBS1	415	402.6875	2.97	1245
TOOLS	10	9.9375	0.63	30
INDX	25	24.9375	0.25	75
USERS	25	24.9375	0.25	75
TOTAL	1840.63	578.88	53.50	5521.875

SIAADM

TABLESPACE_NAME	MB_USED	MB_FREE	PERCENT_USED	SIZE_ASM
EXAMPLE	148.75	0.125	99.92	446.25
SYSTEM	430	9.375	97.82	1290
XDB	46.875	2.625	94.4	140.625
CWMLITE	20	2.25	88.75	60
USERS	25	9.125	63.5	75
DRSYS	20	10.3125	48.44	60
ODM	20	10.6875	46.56	60
SIAADM	10	7.875	21.25	30
UNDOTBS1	420	402.6875	4.12	1260
TOOLS	10	9.9375	0.63	30
INDX	25	24.9375	0.25	75
TOTAL	1175.63	489.94	51.42	3526.875

ADQUI

TABLESPACE_NAME	MB_USED	MB_FREE	PERCENT_USED	SIZE_ASM
EXAMPLE	148.75	0.125	99.92	446.25
SYSTEM	430	10.25	97.62	1290
XDB	46.875	2.625	94.4	140.625
CWMLITE	20	2.25	88.75	60
DRSYS	20	10.3125	48.44	60
ODM	20	10.6875	46.56	60
DGBS_DATA	20	16.625	16.88	60
UNDOTBS1	415	397.6875	4.17	1245
SALA	100	99.25	0.75	300
TOOLS	10	9.9375	0.63	30
USERS	25	24.875	0.5	75
INDX	25	24.9375	0.25	75
CIS_DATA	100	99.9375	0.06	300
TOTAL	1380.63	709.5	38.38	4141.875

SRCPROD

TABLESPACE_NAME	MB_USED	MB_FREE	PERCENT_USED	SIZE_ASM
EXAMPLE	148.75	0.125	99.92	446.25
SYSTEM	430	10.25	97.62	1290
XDB	46.875	2.625	94.4	140.625
CWMLITE	20	2.25	88.75	60
DRSYS	20	10.3125	48.44	60
ODM	20	10.6875	46.56	60
SALA	50	38	24	150
DGBS_DATA	20	17.5	12.5	60
UNDOTBS1	415	397.6875	4.17	1245
BDRSC	100	96.4375	3.56	300
TOOLS	10	9.9375	0.63	30
INDX	25	24.9375	0.25	75
USERS	25	24.9375	0.25	75
CIS_DATA	50	49.9375	0.13	150
TOTAL	1380.63	695.63	37.23	4141.875

1.3 DISTRIBUCIÓN EN LOS NUEVOS SERVIDORES DE BASE DE DATOS

En el siguiente cuadro se muestra la forma en que quedaran distribuidas las instancias en cada una de las zonas en los nuevos servidores de base de datos.

No.	SERVIDOR/DIRECCIÓN IP	INSTANCIA	DISKGROUP ASIGNADO
1	SVRDBSIOBK2/10.33.1.xx	SIOBK2	DATOSD0Z1
2	SVRDBSIO/10.33.1.xx	SIAM VIDECUFI REMESAS MAE ESTAD SIO	DATOSD0Z2
3	SVRDBLOBORH/10.33.1.xx	LOBORH LOBOSYS LOBORH08 LOBORH09	DATOSD1Z1
4	SVRDBSIA/10.33.1.xx	SIA SIAADM CONBAN* ADQUI* SRCPROD*	DATOSD1Z2

Nota: las instancias CONBAN,ADQUI y SRCPROD se migraran a una sola instancia llamada sisadm a petición del cliente.

Ya que se realizo de forma correcta el dimensionamiento de los nuevos servidores apartir de la información presentada se puede continuar con el proceso de migración el cual sera seguida describiendo en los siguientes capitulos.

Capítulo 2

ETAPAS DE PRUEBA

ETAPAS DE PRUEBA

A continuación se mostraran las etapas que se llevaron a cabo en la migración de las bases de datos para CONDUSEF, desde la instalación del software de Oracle 11G R2 en las zonas de los servidores, la creación de las instancias de base de datos, la obtención de los objetos DDL y su creación en los destinos, para posteriormente hacer la migración lógica de los schemas indicados por el cliente.

Debido a que la instalación y creación de los grupos de discos para ASM es idéntica en las 4 zonas de los servidores, se mostrará únicamente una instalación que será la de la zona denominada SVRDBSIO indicando si es que hay algún cambio durante el proceso de instalación.

2.1 INSTALACIÓN DE SOFTWARE ORACLE EN LAS ZONAS

Revisión de requisitos para Oracle 11gR2 en Solaris

Esto se requiere hacer en las cuatro zonas que tenemos, como ejemplo se pondrá solo en la zona SVRDBSIO, conectarse con el usuario root y ejecutar los siguientes comandos:

1. Revisar memoria del server

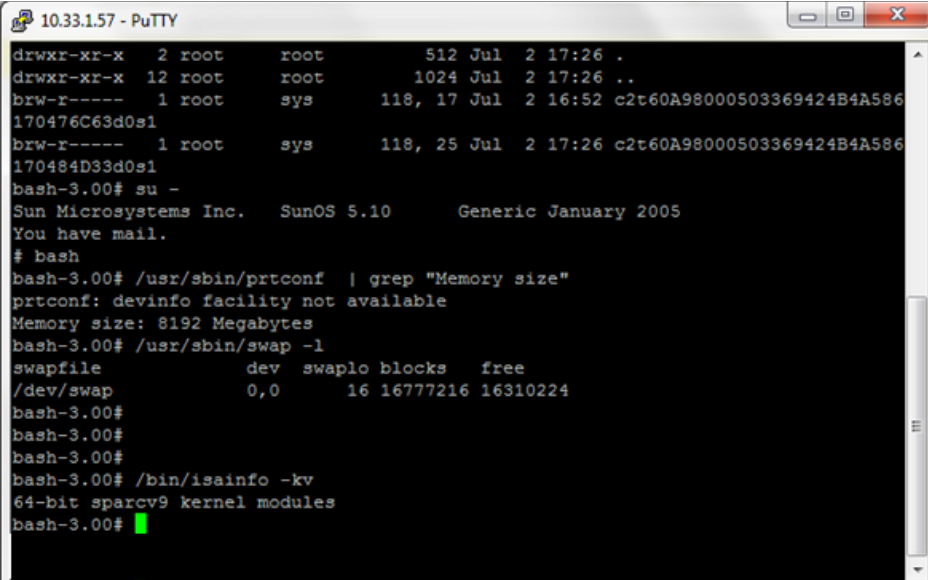
```
/usr/sbin/prtconf | grep "Memory size"
```

2. Revisar Swap

```
/usr/sbin/swap -l
```

3. Checar arquitectura

```
/bin/isainfo -kv
```



```
10.33.1.57 - PuTTY
drwxr-xr-x  2 root   root       512 Jul  2 17:26 .
drwxr-xr-x 12 root   root      1024 Jul  2 17:26 ..
brw-r----- 1 root   sys       118, 17 Jul  2 16:52 c2t60A98000503369424B4A586
170476C63d0s1
brw-r----- 1 root   sys       118, 25 Jul  2 17:26 c2t60A98000503369424B4A586
170484D33d0s1
bash-3.00# su -
Sun Microsystems Inc.  SunOS 5.10      Generic January 2005
You have mail.
# bash
bash-3.00# /usr/sbin/prtconf | grep "Memory size"
prtconf: devinfo facility not available
Memory size: 8192 Megabytes
bash-3.00# /usr/sbin/swap -l
swapfile      dev  swaplo blocks  free
/dev/swap     0,0   16 16777216 16310224
bash-3.00#
bash-3.00#
bash-3.00#
bash-3.00# /bin/isainfo -kv
64-bit sparcv9 kernel modules
bash-3.00#
```

4. Espacio en temporal

```
df -k /tmp
```

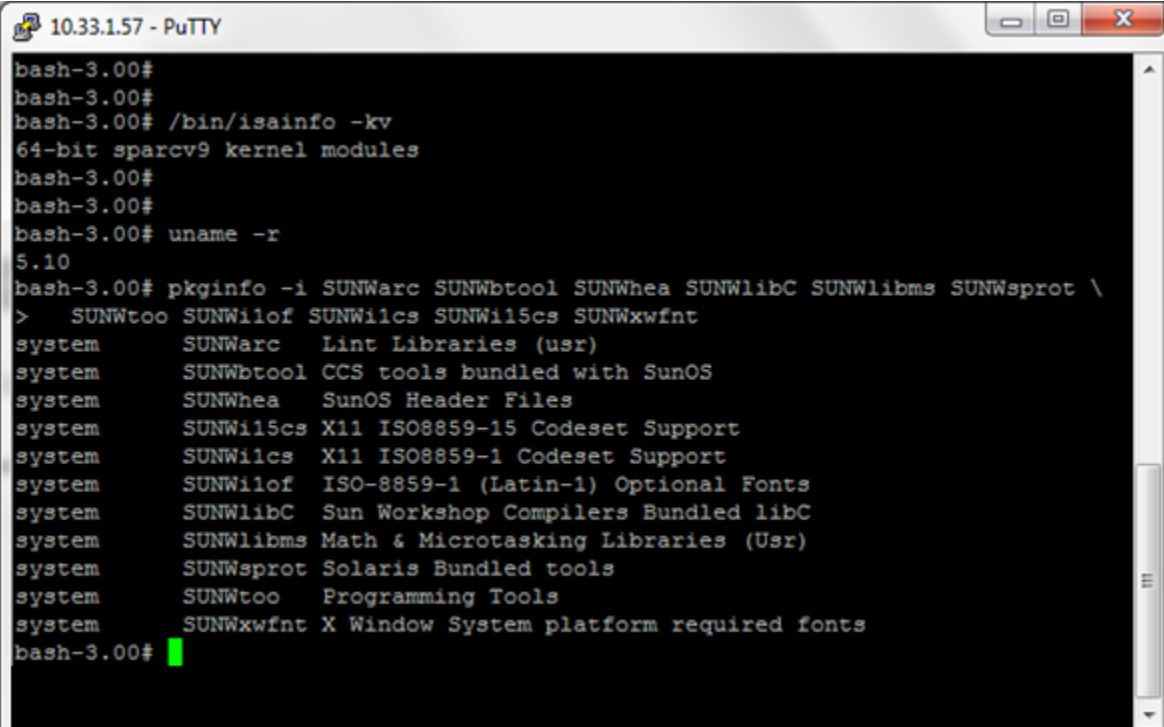
5. Versión de SO

```
uname -r
```


6. Paquetes necesarios

```
SUNWarc
SUNWbtool
SUNWhea
SUNWlibC
SUNWlibm
SUNWlibms
SUNWsprout
SUNWtoo
SUNWilof
SUNWilcs (ISO8859-1)
SUNWi15cs (ISO8859-15)
SUNWxwfnt
SUNWcsl
```

```
$ pkginfo -i SUNWarc SUNWbtool SUNWhea SUNWlibC SUNWlibms SUNWsprout \
  SUNWtoo SUNWilof SUNWilcs SUNWi15cs SUNWxwfnt
```



```
10.33.1.57 - PuTTY
bash-3.00#
bash-3.00#
bash-3.00# /bin/isainfo -kv
64-bit sparcv9 kernel modules
bash-3.00#
bash-3.00#
bash-3.00# uname -r
5.10
bash-3.00# pkginfo -i SUNWarc SUNWbtool SUNWhea SUNWlibC SUNWlibms SUNWsprout \
> SUNWtoo SUNWilof SUNWilcs SUNWi15cs SUNWxwfnt
system      SUNWarc      Lint Libraries (usr)
system      SUNWbtool    CCS tools bundled with SunOS
system      SUNWhea      SunOS Header Files
system      SUNWi15cs    X11 ISO8859-15 Codeset Support
system      SUNWilcs     X11 ISO8859-1 Codeset Support
system      SUNWilof     ISO-8859-1 (Latin-1) Optional Fonts
system      SUNWlibC     Sun Workshop Compilers Bundled libc
system      SUNWlibms    Math & Microtasking Libraries (Usr)
system      SUNWsprout   Solaris Bundled tools
system      SUNWtoo      Programming Tools
system      SUNWxwfnt    X Window System platform required fonts
bash-3.00#
```

7. Revisión de parches:

120753-06: SunOS 5.10: Microtasking libraries (libmtsk) patch

139574-03: SunOS 5.10

```
/usr/sbin/patchadd -p | grep 120753
```

```
/usr/sbin/patchadd -p | grep 139574
```

8. Creación de grupos y usuario de S.O estos mismos pasos se realizarán en las diferentes zonas de los servidores:

```
$ groupadd oinstall
```

```
$ groupadd dba
```

Generar el usuario de sistema orapp y asignarle una contraseña:

```
$ useradd -g oinstall -G dba orapp
```

```
$ passwd orapp
```

```
New Password: *****
```

Asignar permisos a orapp

```
$ chown orapp:oinstall /u01
```

Crear un directorio llamado orapp dentro de la ruta /export/home/, el cual será el home del usuario, asignarle los siguientes permisos:

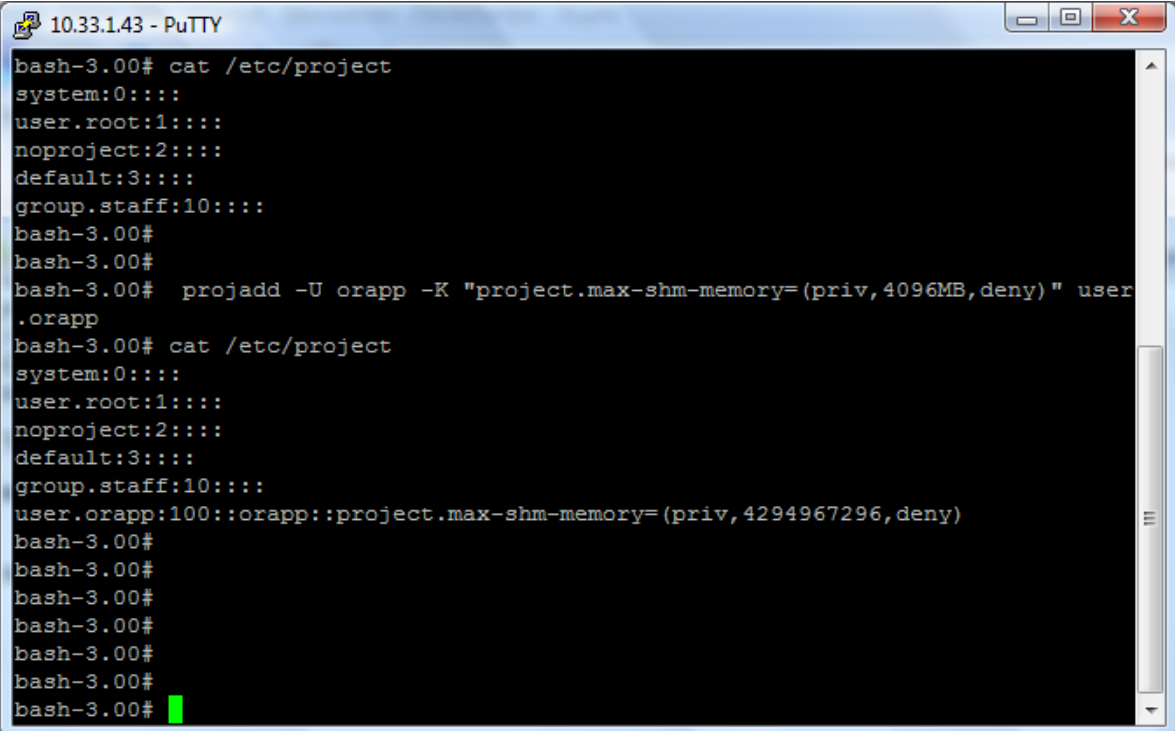
```
$ mkdir -p /export/home/orapp
```

```
$ chown orapp:oinstall /export/home/orapp
```

9. Creación del proyecto para orapp

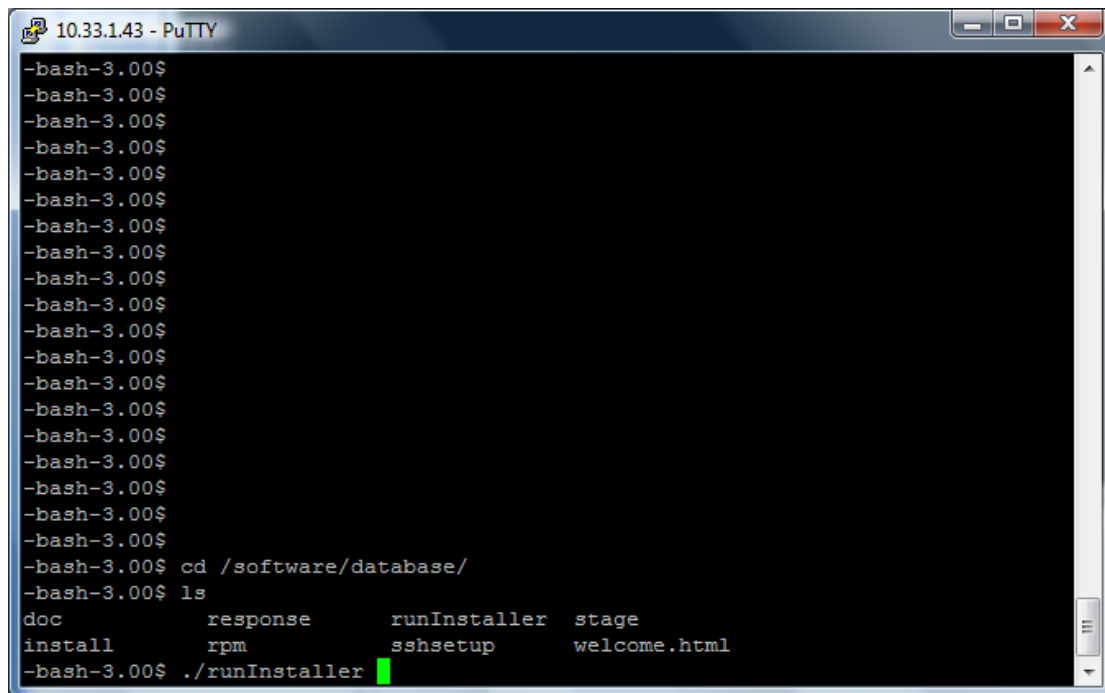
Como usuario root ejecutar lo siguiente en los 4 servidores.

```
$ Projadd -U orapp -K "Project.max-shm-memory=(priv,4096MB,deny)" user.orapp
```



```
10.33.143 - PuTTY
bash-3.00# cat /etc/project
system:0::::
user.root:1::::
noproject:2::::
default:3::::
group.staff:10::::
bash-3.00#
bash-3.00#
bash-3.00# projadd -U orapp -K "project.max-shm-memory=(priv,4096MB,deny)" user.orapp
bash-3.00# cat /etc/project
system:0::::
user.root:1::::
noproject:2::::
default:3::::
group.staff:10::::
user.orapp:100::orapp::project.max-shm-memory=(priv,4294967296,deny)
bash-3.00#
bash-3.00#
bash-3.00#
bash-3.00#
bash-3.00#
bash-3.00#
bash-3.00#
```


3. Regresar al directorio `/export/home/orapp/software` y entrar a la carpeta Database y ejecutar el archivo para instalar el software de Oracle mediante el comando `./runInstaller`.

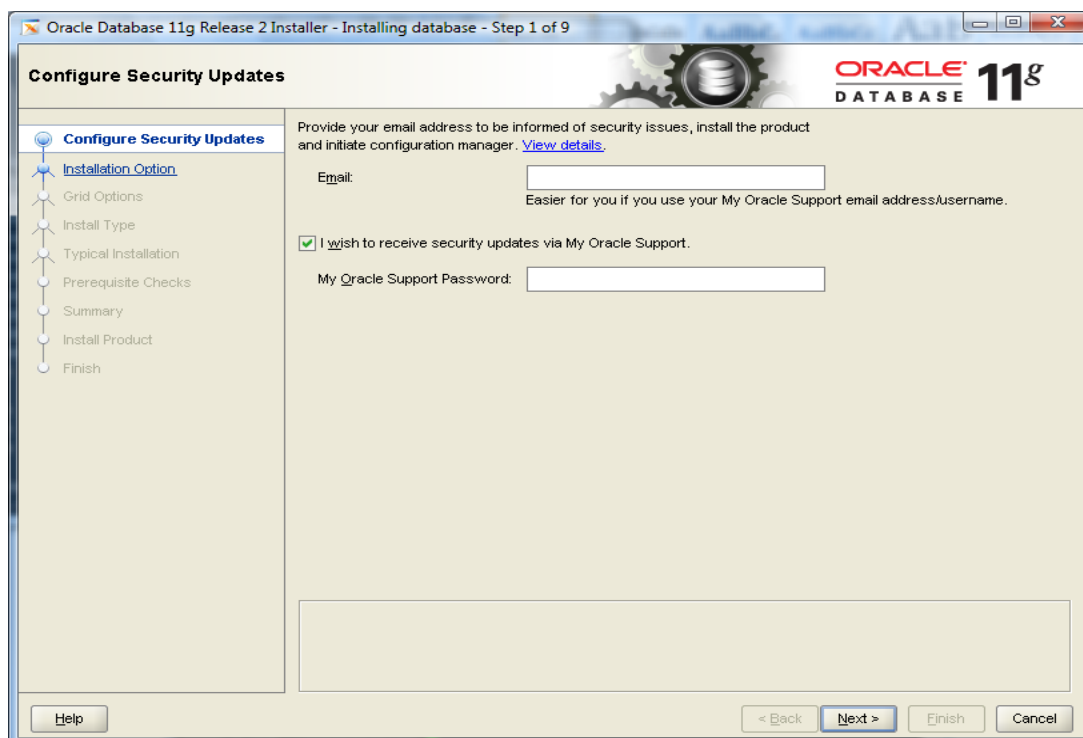


```

10.33.1.43 - PuTTY
-bash-3.00$
-bash-3.00$
-bash-3.00$
-bash-3.00$
-bash-3.00$
-bash-3.00$
-bash-3.00$
-bash-3.00$
-bash-3.00$
-bash-3.00$
-bash-3.00$
-bash-3.00$
-bash-3.00$
-bash-3.00$
-bash-3.00$
-bash-3.00$
-bash-3.00$
-bash-3.00$
-bash-3.00$
-bash-3.00$
-bash-3.00$ cd /software/database/
-bash-3.00$ ls
doc          response    runInstaller stage
install     rpm        sshsetup    welcome.html
-bash-3.00$ ./runInstaller

```

Al ejecutar el comando desplegará el asistente para la instalación del software de Oracle en la que pide que se defina un correo para que mande avisos de actualizaciones, en ambos campos se deja en blanco y se da next para continuar.



Oracle Database 11g Release 2 Installer - Installing database - Step 1 of 9

Configure Security Updates

Provide your email address to be informed of security issues, install the product and initiate configuration manager. [View details.](#)

Email:

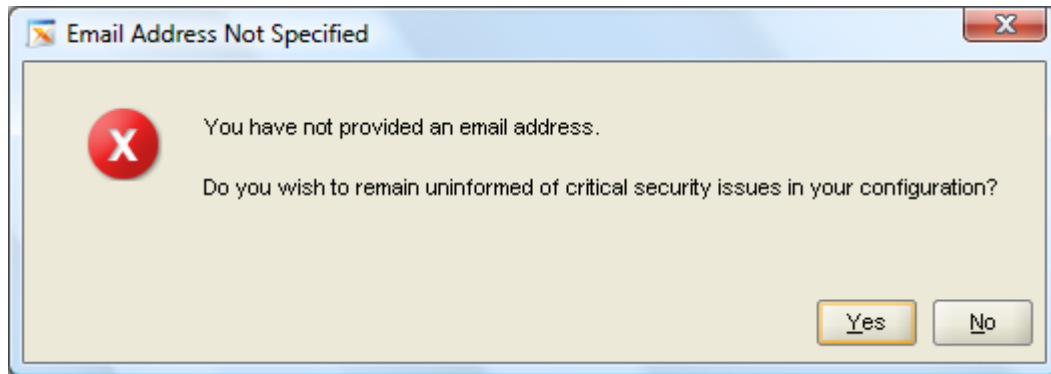
Easier for you if you use your My Oracle Support email address/username.

I wish to receive security updates via My Oracle Support.

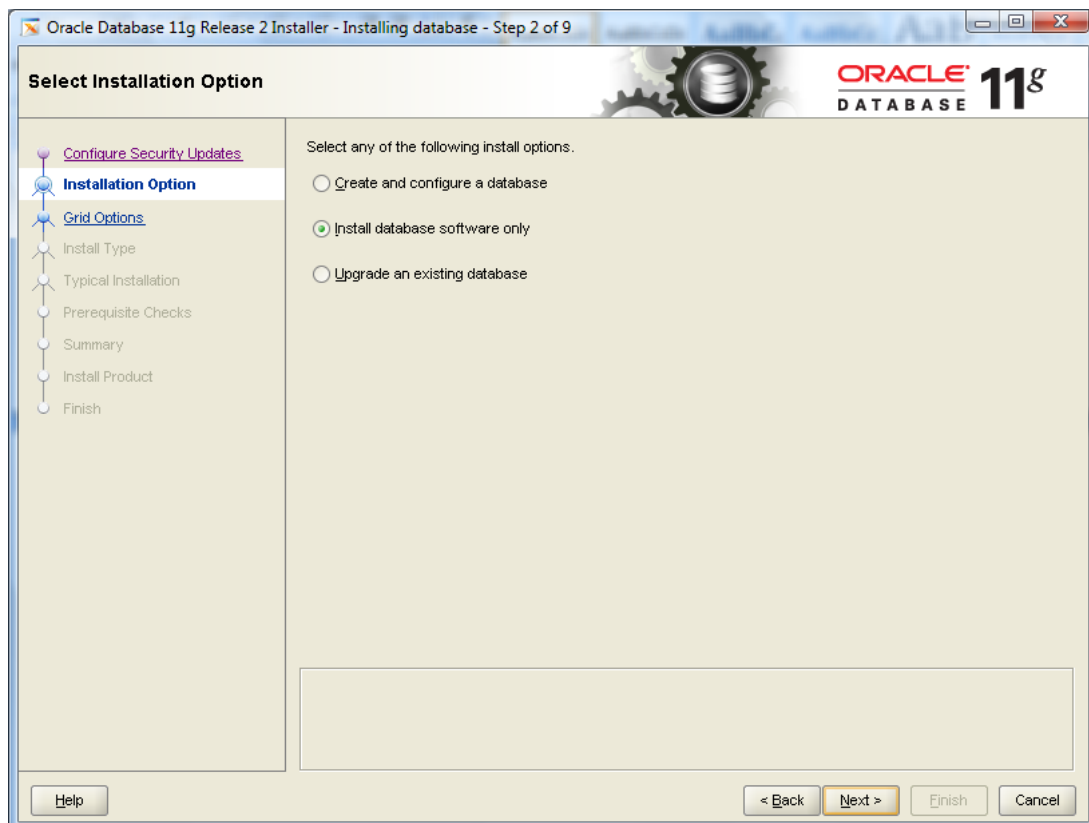
My Oracle Support Password:

Help < Back Next > Finish Cancel

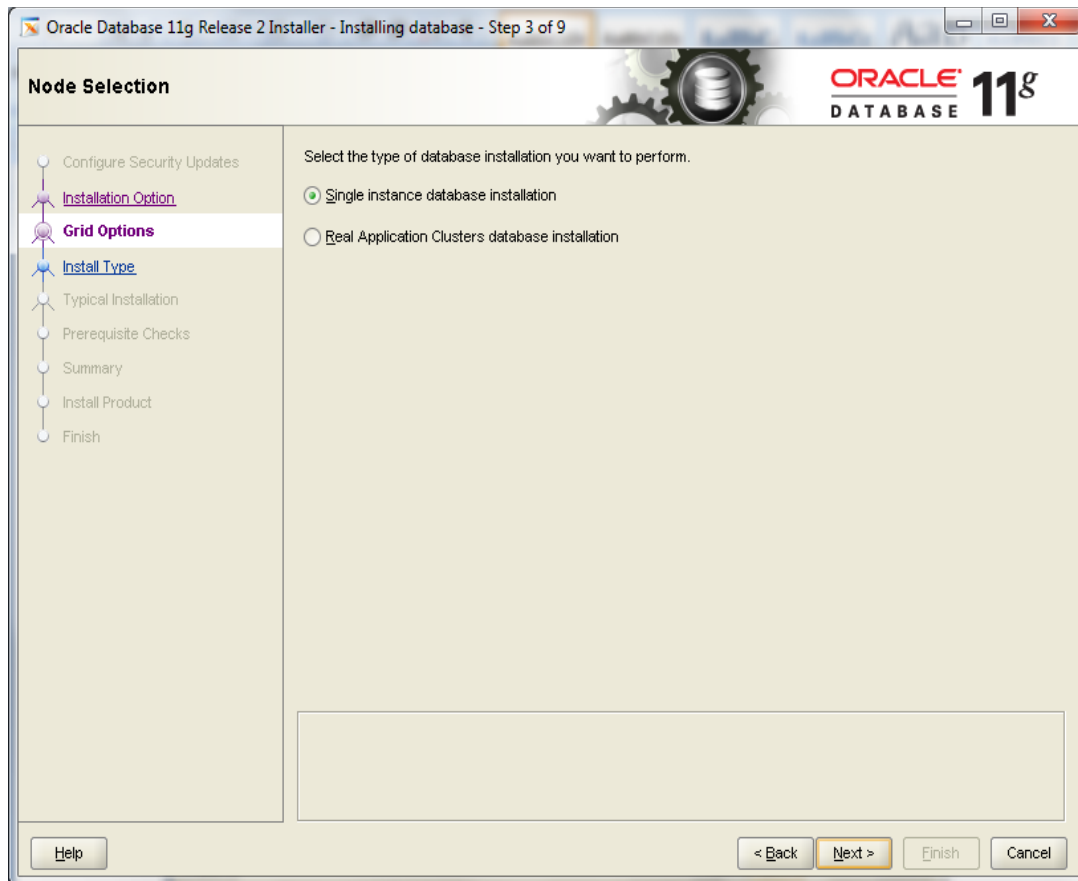
Al dejar dichos campos en blanco el asistente nos mandara un mensaje de alerta le damos en la opción yes para continuar con el proceso.



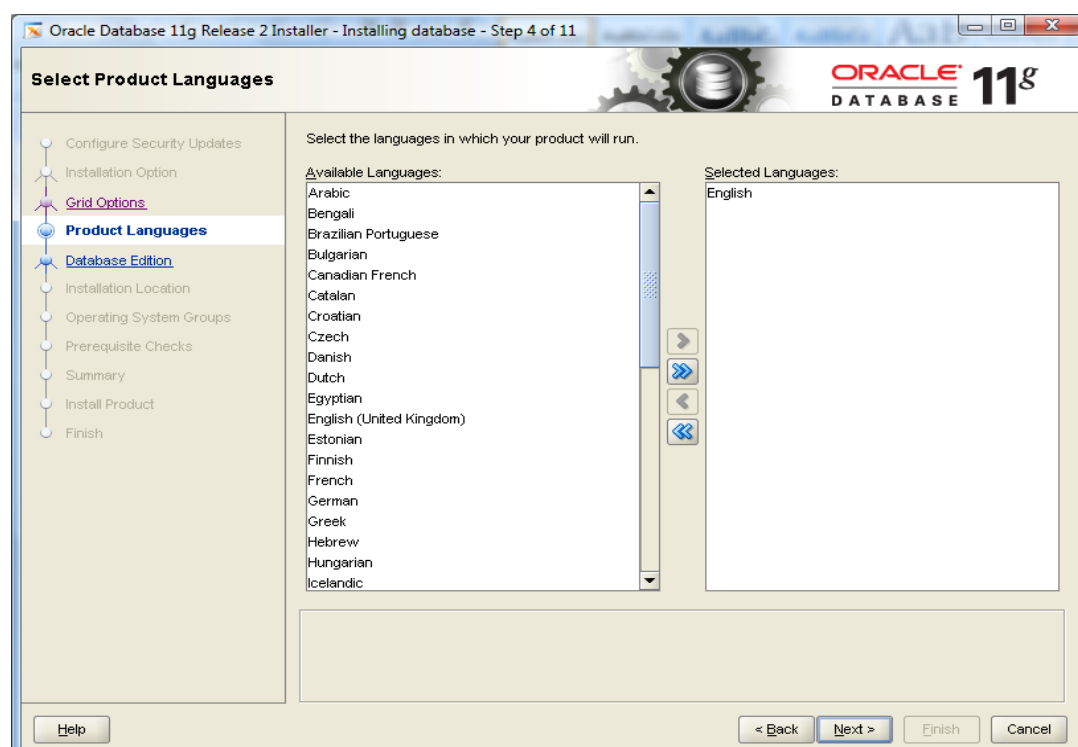
El siguiente paso es elegir que se desea instalar, se elige la opción de solo el software de oracle, las instancias de base de datos se crearán después.



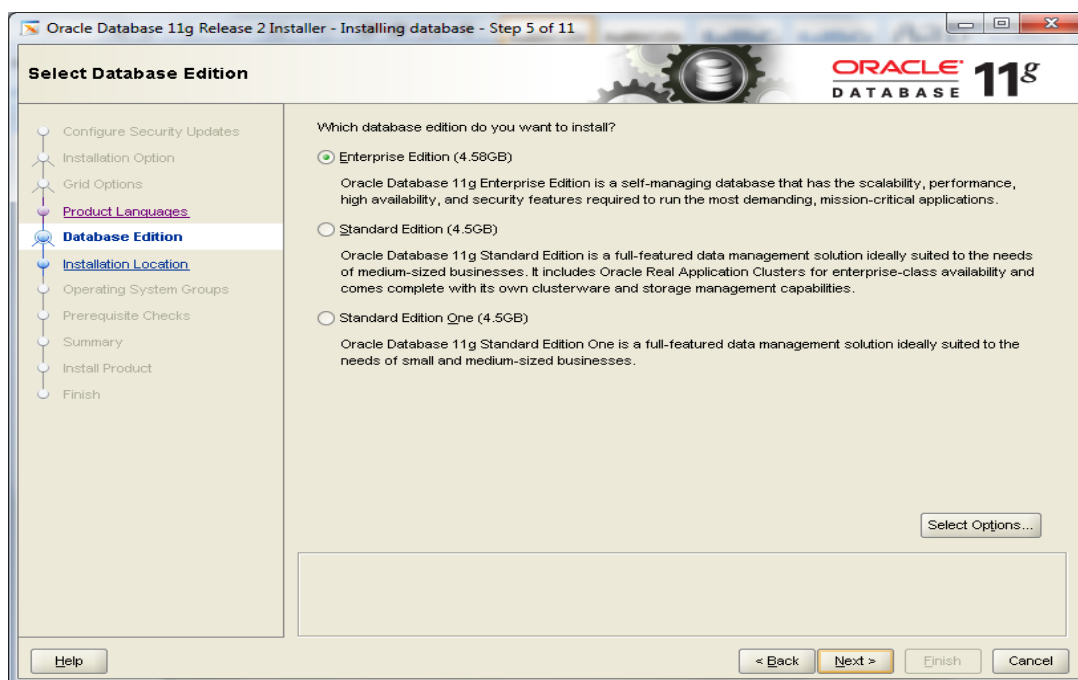
Ahora se elige la opción de single instance Database, ya que sera en standalone, lo cual indica es que es un solo servidor.



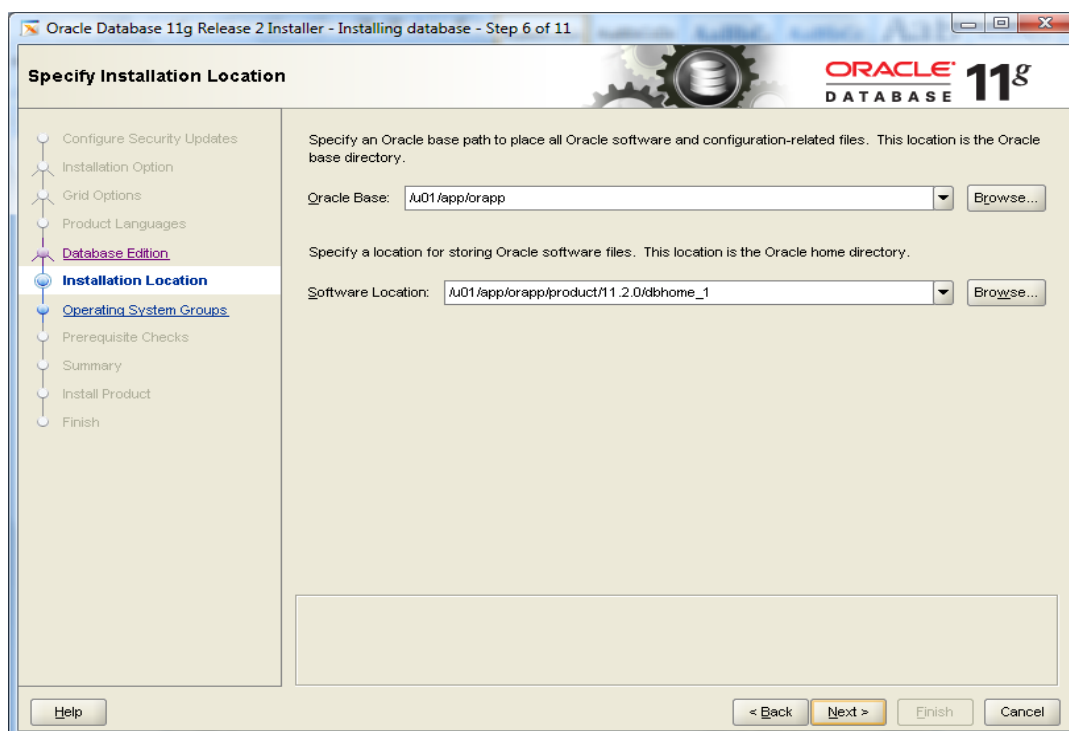
Se define el lenguaje del software, en este caso sólo se selecciona el inglés.



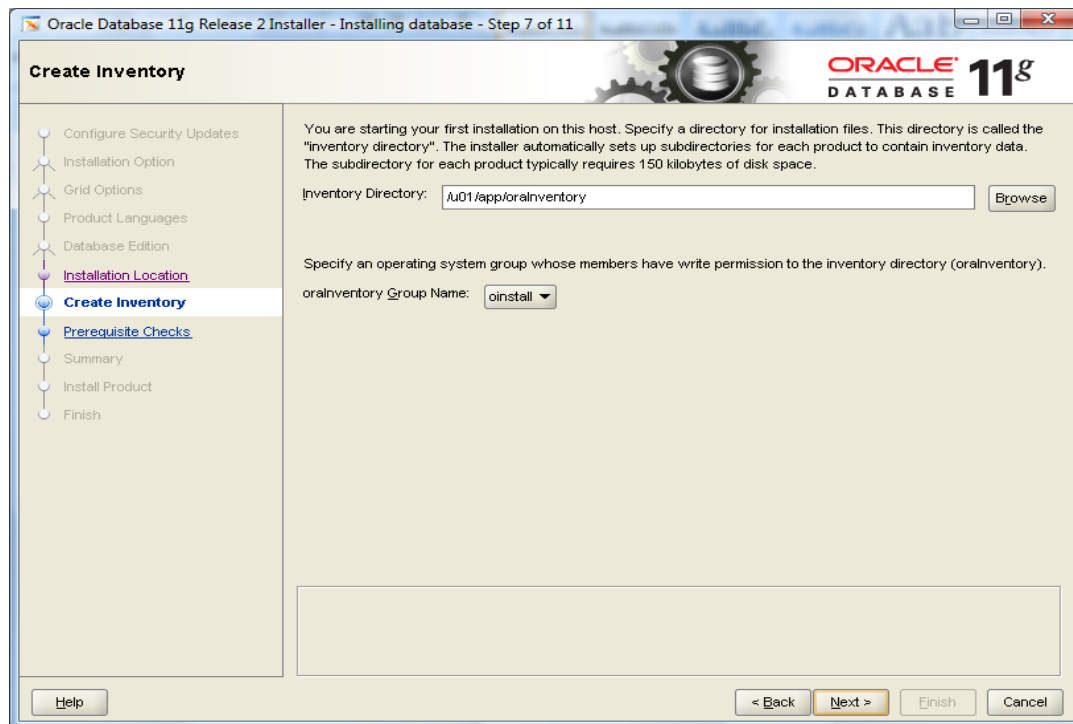
El paso siguiente es definir la edición del software que se va a instalar, se elige la opción Enterprise Edition, esta elección es en base al licenciamiento adquirido por el cliente.



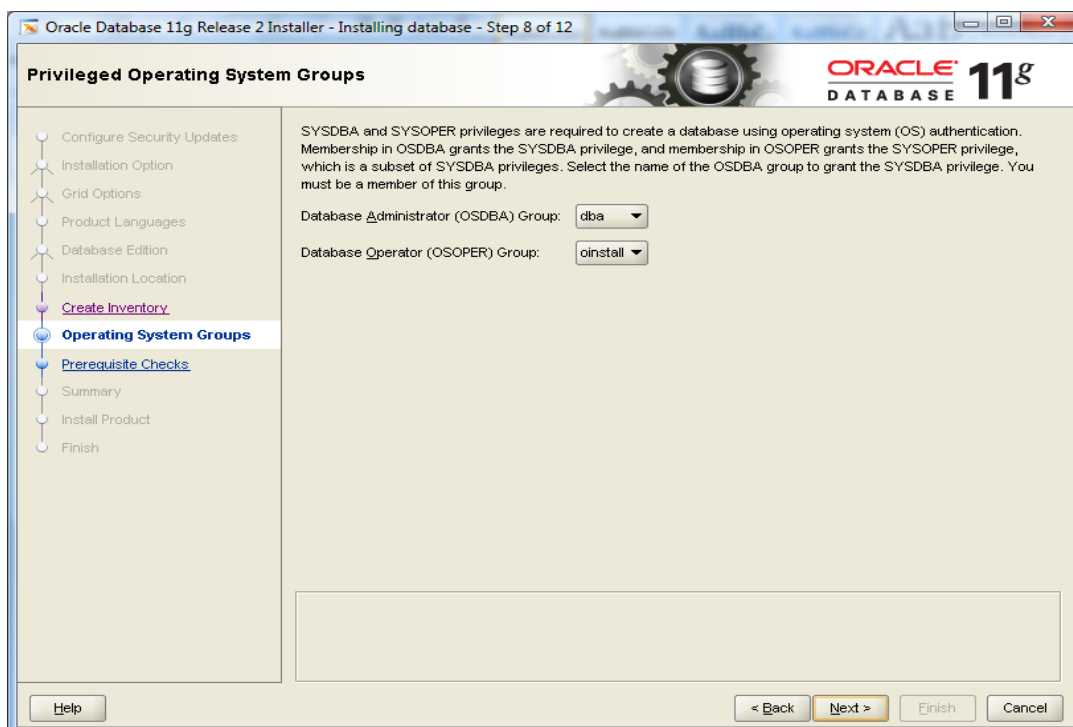
Pasado el punto anterior se define el oracle_base y el Oracle_home, los cuales serán la ruta base y la ruta home de Oracle en la que quedarán instalados los binarios del software.



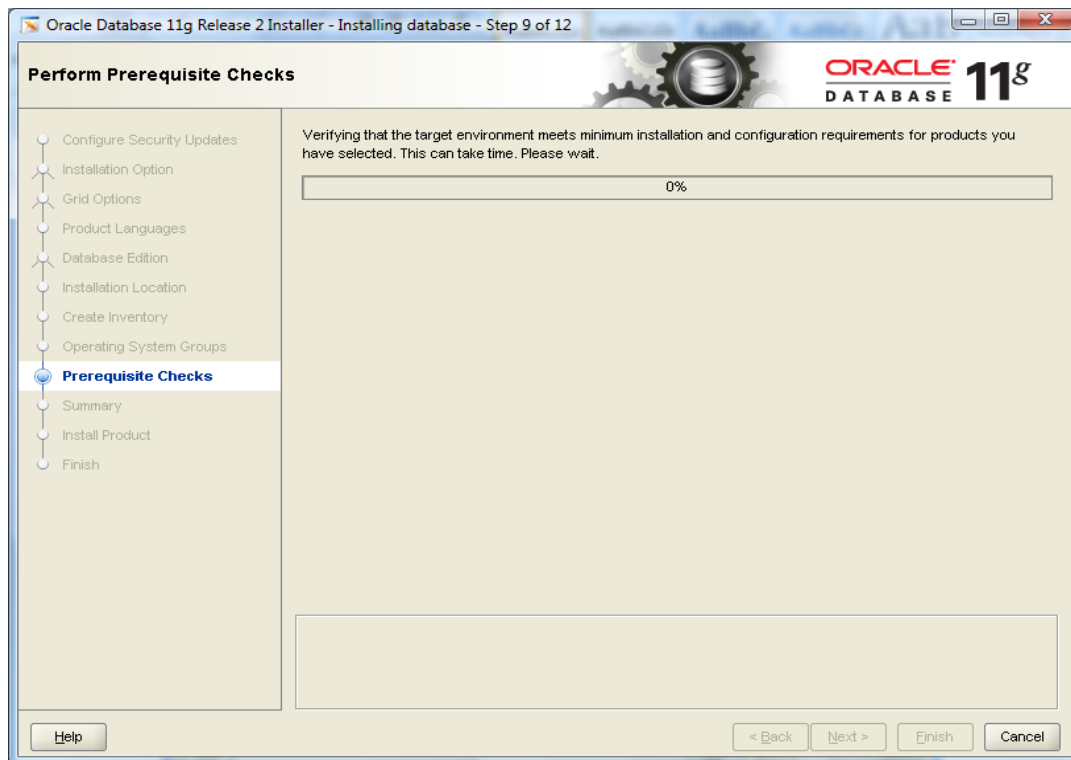
En el paso 7 se define la ruta del directorio oraInventory, así como el grupo propietario de S.O. el cual será oinstall.



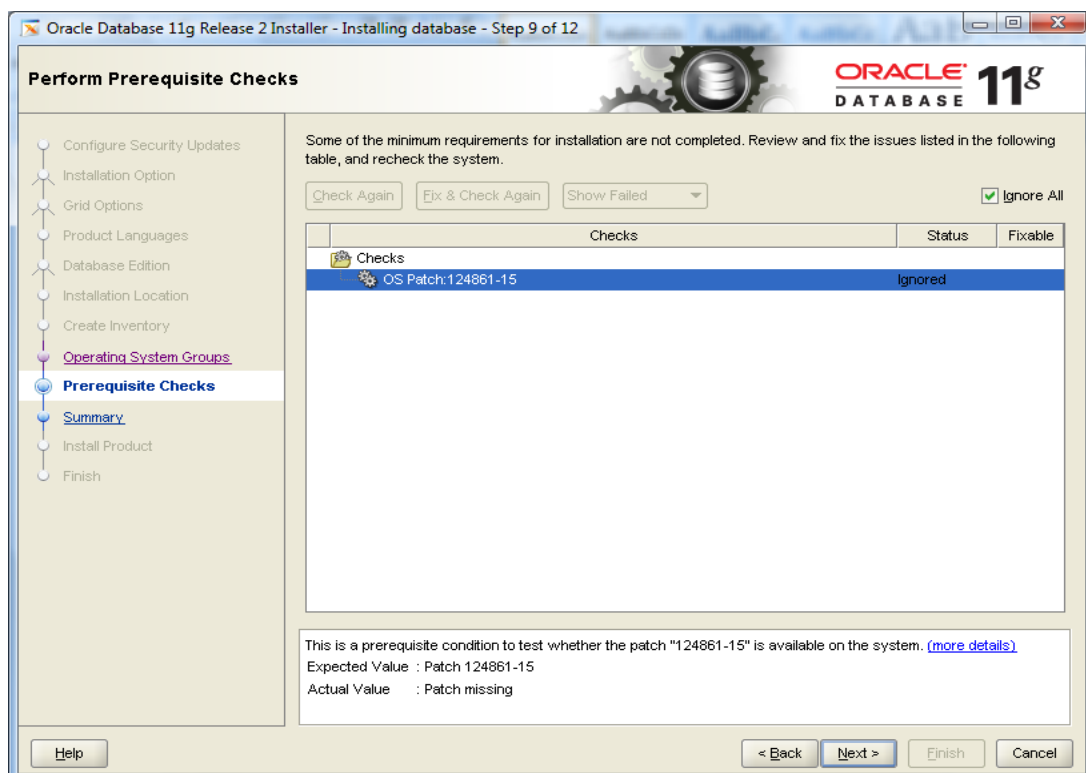
En el paso 8 se definen los grupos que tienen los privilegios a nivel de S.O. como se muestra a continuación.



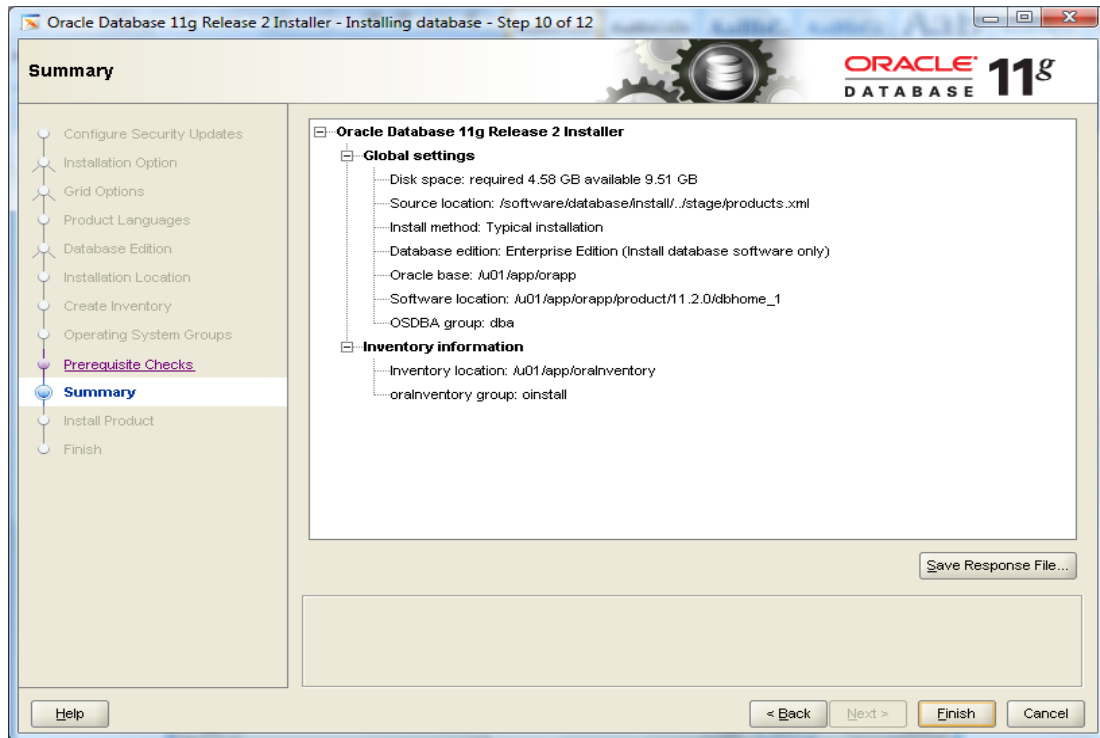
En el paso 9 el asistente realizara una revisión de requisitos.



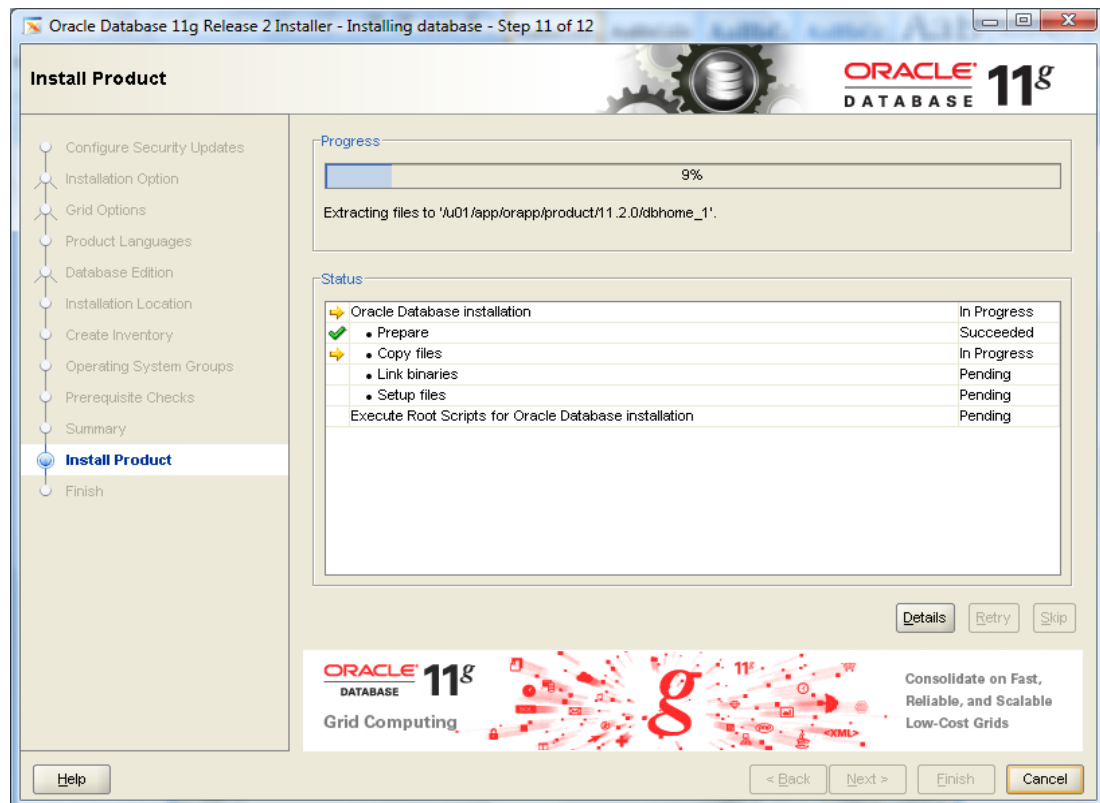
Al término de la revisión se indica que falló al validar un parche pero este no tiene problema ya que es opcional.



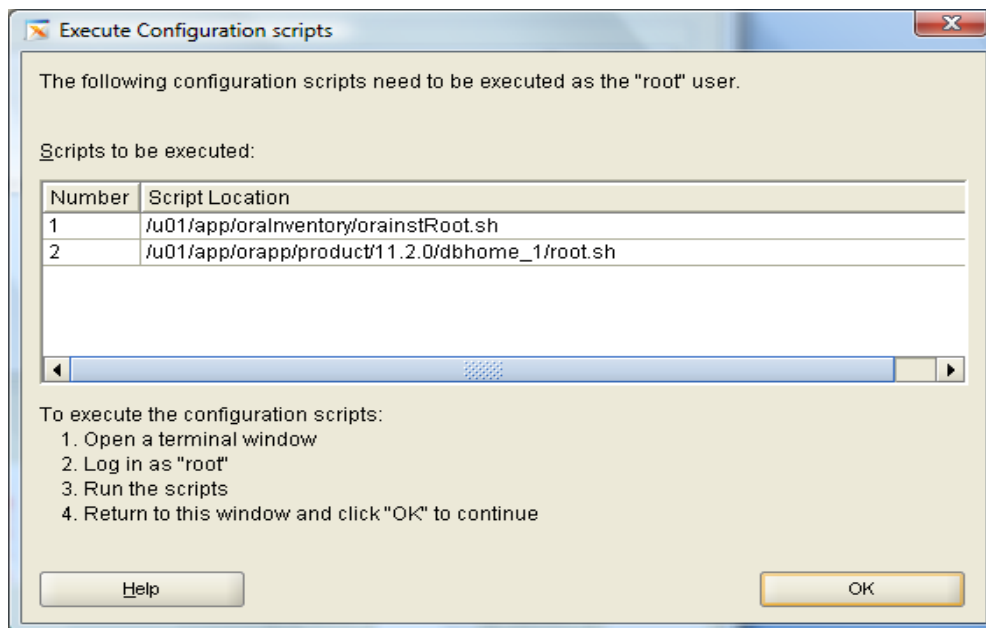
En el paso 10 el asistente nos muestra un resumen antes de empezar la instalación, dar clic en finish para empezar la instalación.



En el paso 11 muestra el avance de la instalación.



Antes de terminar la instalación el asistente pedirá que se ejecuten dos scripts como usuario root.



Para realizar dicha acción conectarse en una nueva terminal con dicho usuario y ejecutar ambos scripts como se muestra en las siguientes pantallas.

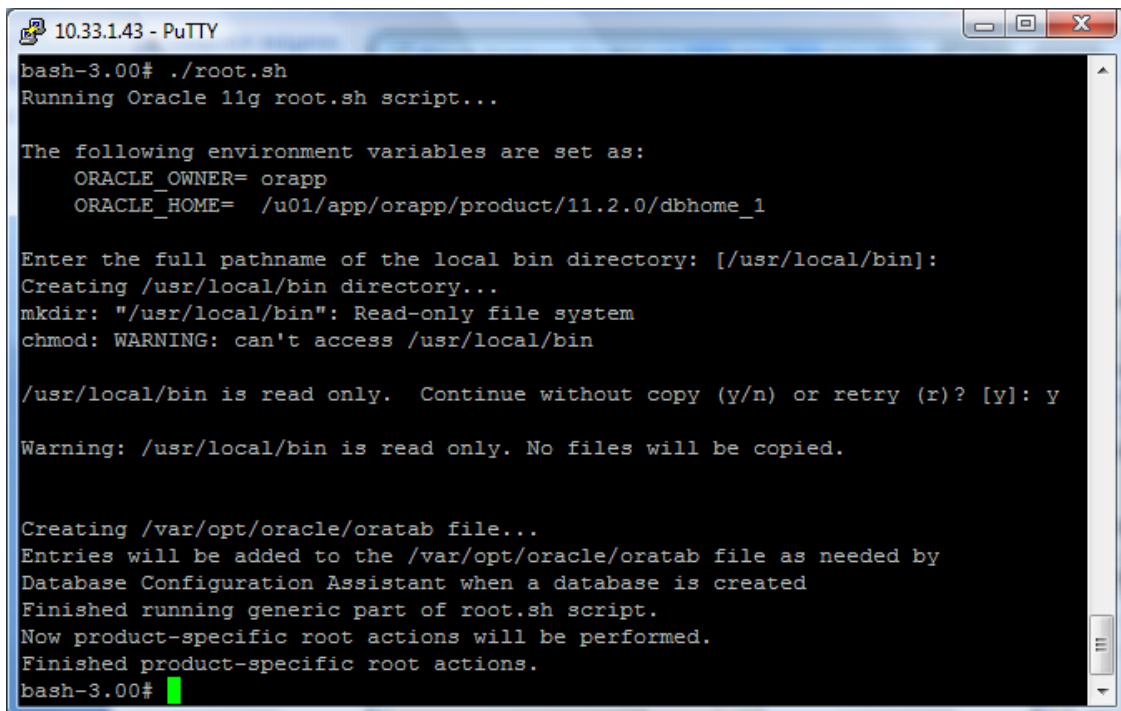
```

10.33.143 - PuTTY
-bash-3.00$
-bash-3.00$
-bash-3.00$
-bash-3.00$
-bash-3.00$
-bash-3.00$ You can find the log of this install session at:
/u01/app/orainventory/logs/installActions2010-07-06_02-36-54PM.log
-bash-3.00$
-bash-3.00$ su -
Password:
Sun Microsystems Inc. SunOS 5.10 Generic January 2005
# bash
bash-3.00# cd /u01/app/orainventory/
bash-3.00# ./orainstRoot.sh
Changing permissions of /u01/app/orainventory.
Adding read,write permissions for group.
Removing read,write,execute permissions for world.

Changing groupname of /u01/app/orainventory to oinstall.
The execution of the script is complete.
bash-3.00#
bash-3.00#
bash-3.00#

```

Para el segundo cambiarse a la ubicación del script y ejecutarlo.



```

10.33.1.43 - PuTTY
bash-3.00# ./root.sh
Running Oracle 11g root.sh script...

The following environment variables are set as:
  ORACLE_OWNER= orapp
  ORACLE_HOME=  /u01/app/orapp/product/11.2.0/dbhome_1

Enter the full pathname of the local bin directory: [/usr/local/bin]:
Creating /usr/local/bin directory...
mkdir: "/usr/local/bin": Read-only file system
chmod: WARNING: can't access /usr/local/bin

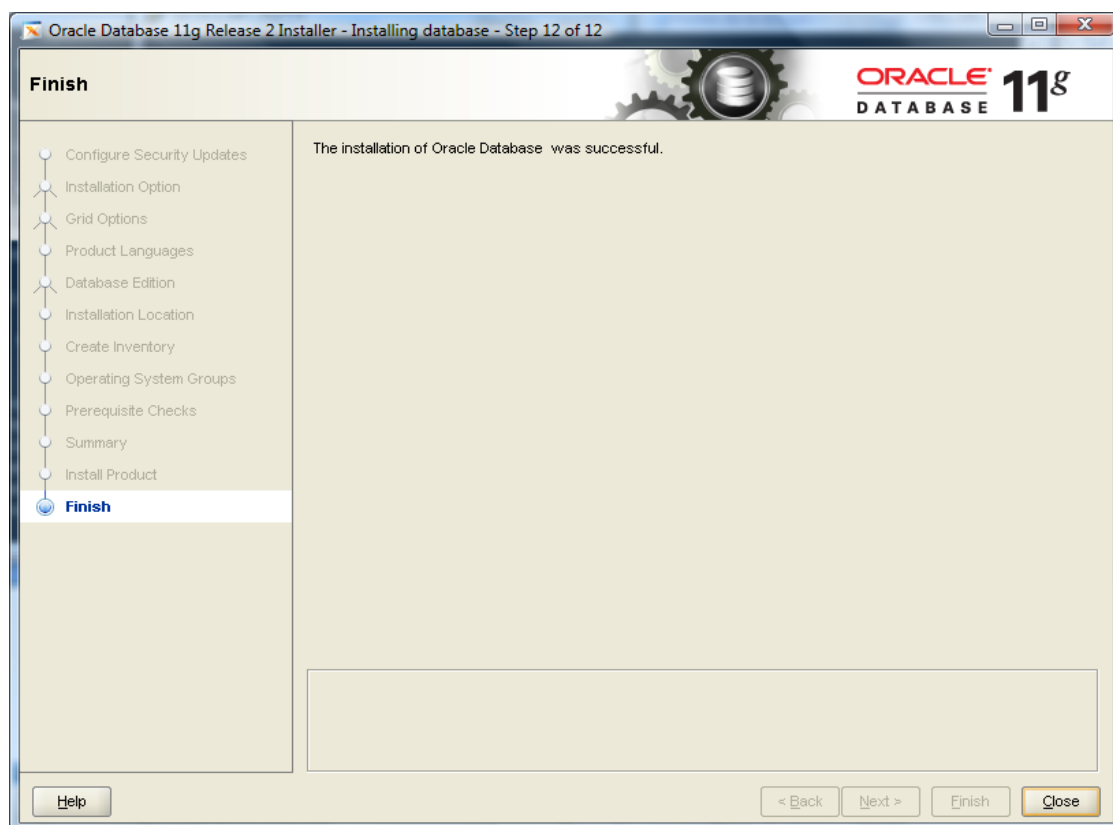
/usr/local/bin is read only. Continue without copy (y/n) or retry (r)? [y]: y

Warning: /usr/local/bin is read only. No files will be copied.

Creating /var/opt/oracle/oratab file...
Entries will be added to the /var/opt/oracle/oratab file as needed by
Database Configuration Assistant when a database is created
Finished running generic part of root.sh script.
Now product-specific root actions will be performed.
Finished product-specific root actions.
bash-3.00#

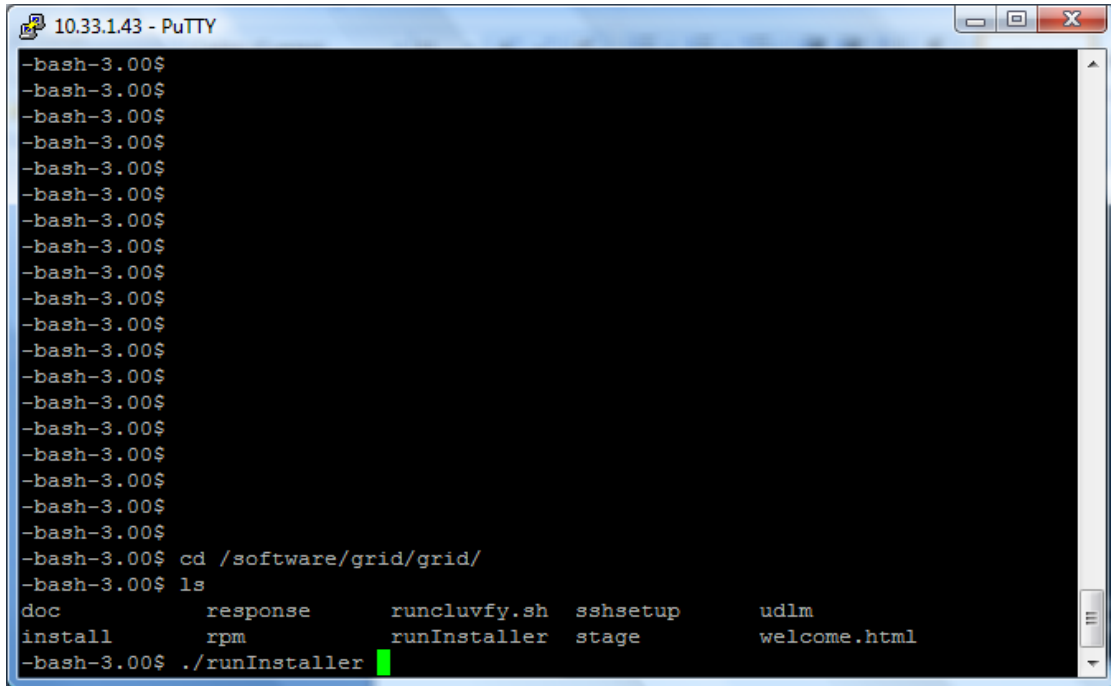
```

Ya realizado lo anterior se da OK en la ventana del asistente para continuar con la instalación, una vez terminada la instalación dar clic en CLOSE.

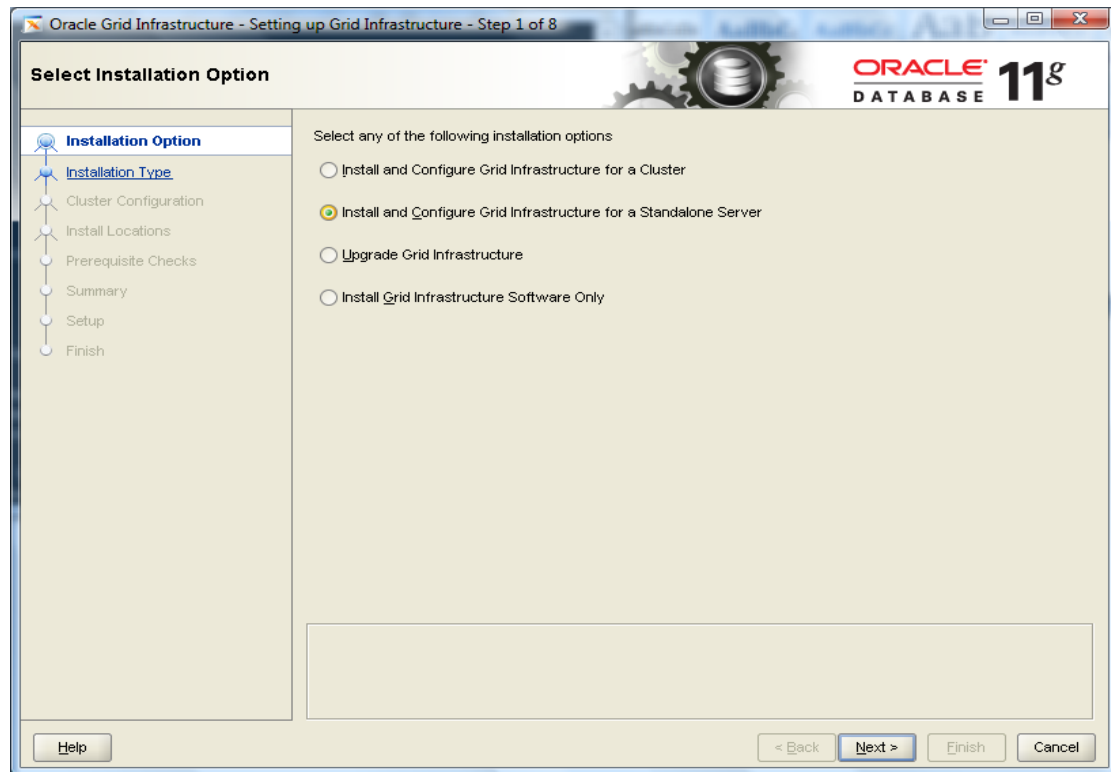


2.3 INTALACIÓN DE GRID INFRASTRUCTURE PARA ASM

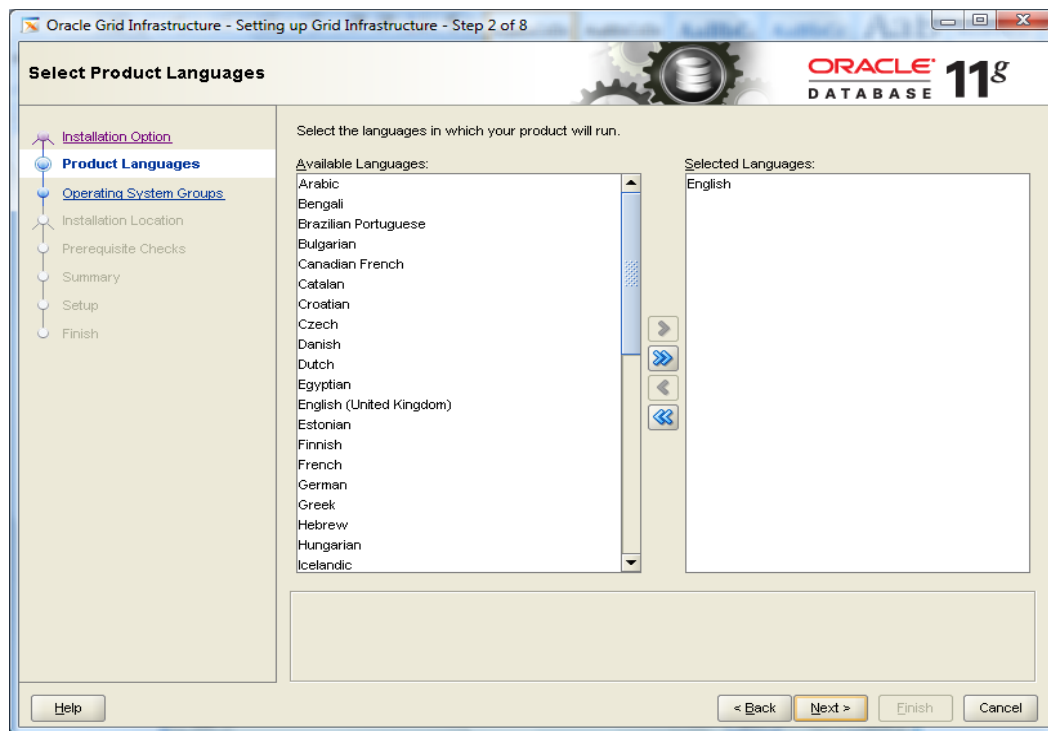
Para la configuración de ASM ingresar a /software/grid y ejecutar el runInstaller



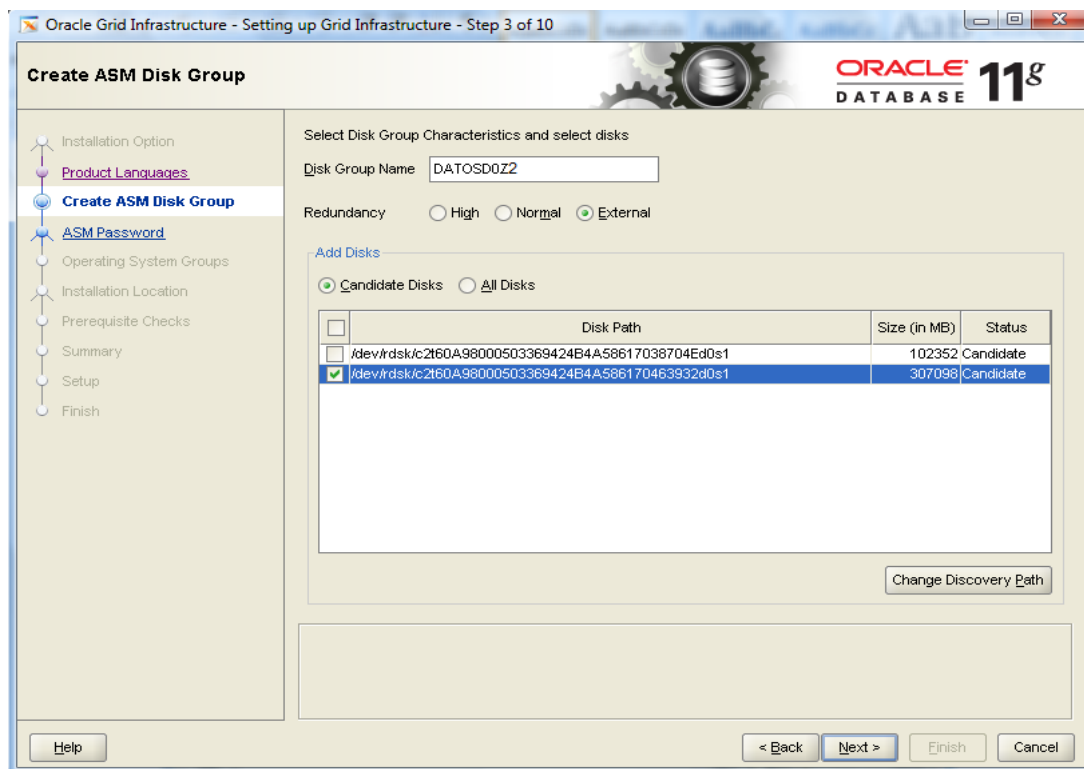
Paso 1: Elegir la opción de instalar y configurar grid infrastructure for a Standalone server.



Paso2: Definir el lenguaje del producto el cual sera inglés



Paso 3: Se define el nombre del grupo de discos para datos el cual sera DATOSD0Z2, redundancia externa y seleccionar el disco con mas espacio.



Se debe recalcar que en cada zona el nombre de los grupos de discos para datos cambia, los cuales se nombran en la siguiente tabla.

ZONA	NOMBRE DEL GRUPO
SVRDBSIO	DATOSD0Z2
SRVDBSIOBK2	DATOSD0Z1
SVRDBLOBORH	DATOSD1Z1
SRVDBSIA	DATOSD1Z2

Paso 4: Se define el password para asm el cual es ***** , como se muestra

Oracle Grid Infrastructure - Setting up Grid Infrastructure - Step 4 of 10

Specify ASM Password

The new Automatic Storage Management (ASM) instance requires its own SYS user with SYSASM privileges for administration. Oracle recommends that you create a less privileged ASMSNMP user with SYSDBA privileges to monitor the ASM instance.

Specify the password for these user accounts.

Use different passwords for these accounts

SYS Password Confirm Password

ASMSNMP

Use same passwords for these accounts

Specify Password: ***** Confirm Password? *****

Messages:

Specify Password:[INS-30011] The password entered does not conform to the Oracle recommended standards.

Help < Back Next > Finish Cancel

Paso 5: Definir los privilegios de grupos para ASM, en los tres campos se deja a oinstall.

Oracle Grid Infrastructure - Setting up Grid Infrastructure - Step 5 of 10

Privileged Operating System Groups

Select the name of the operating system group of which you are a member to be used for OS authentication to Automatic Storage Management (ASM).

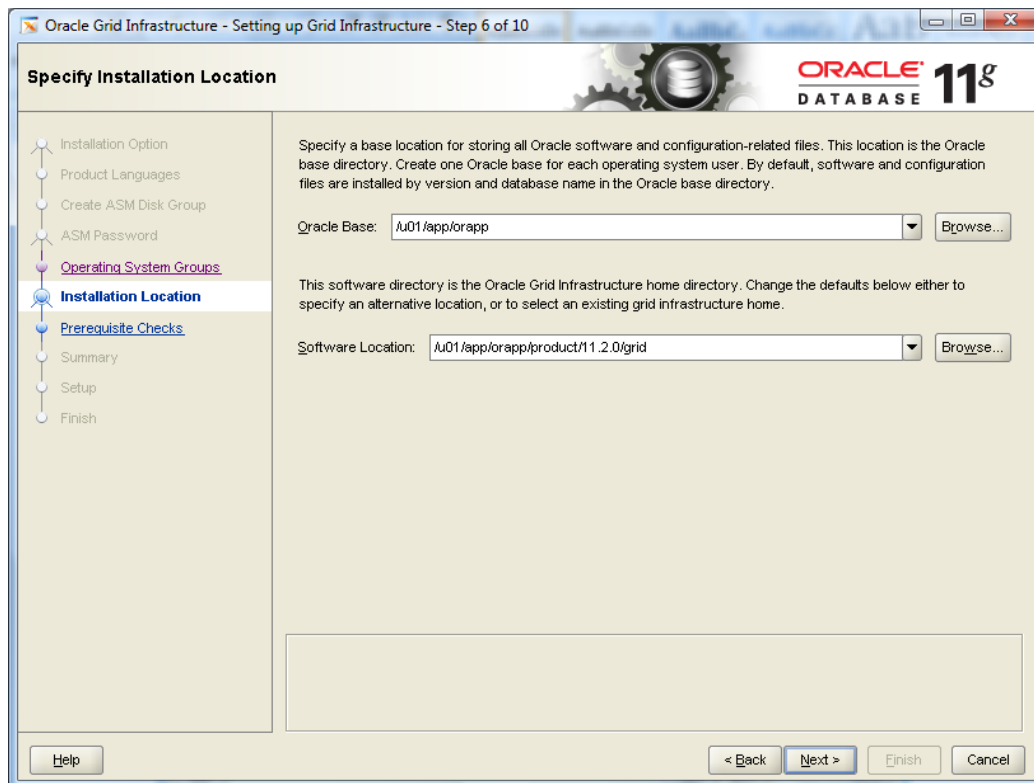
ASM Database Administrator (OSDBA) Group oinstall

ASM Instance Administration Operator (OSOPER) Group oinstall

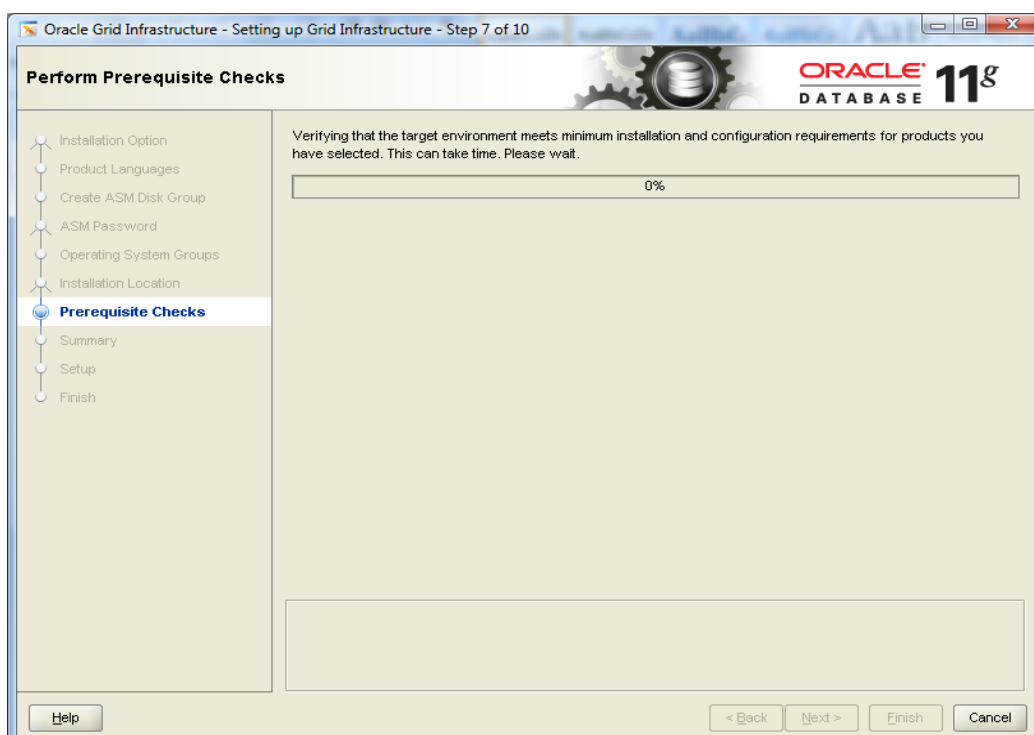
ASM Instance Administrator (OSASM) Group oinstall

Help < Back Next > Finish Cancel

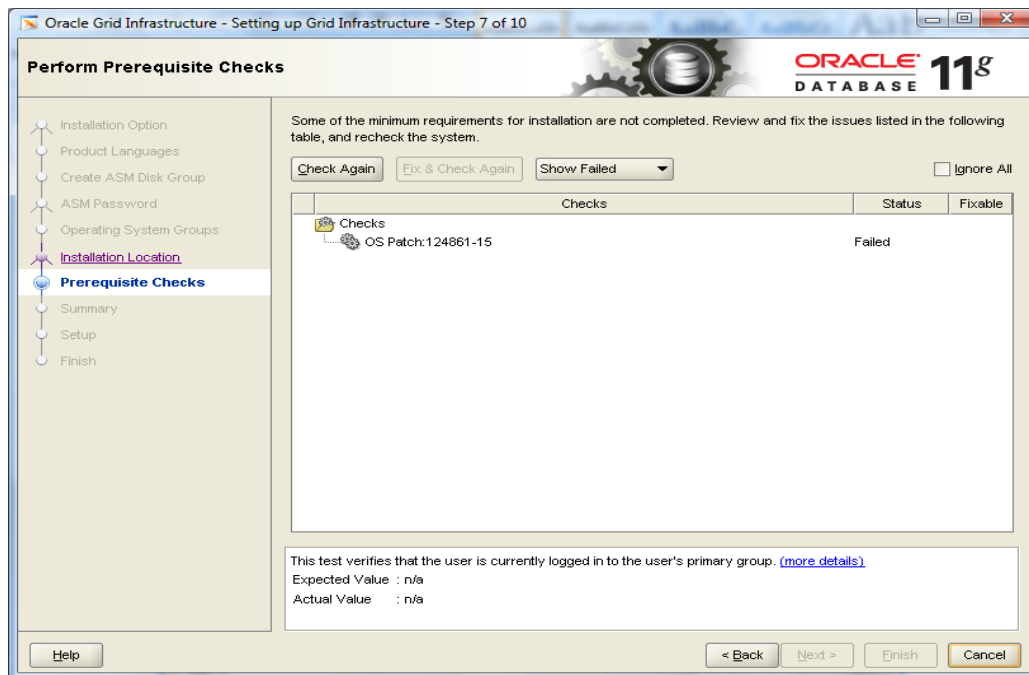
Paso 6: Dejar el Oracle Base y el Oracle_home para el Grid, como aparece a continuación.



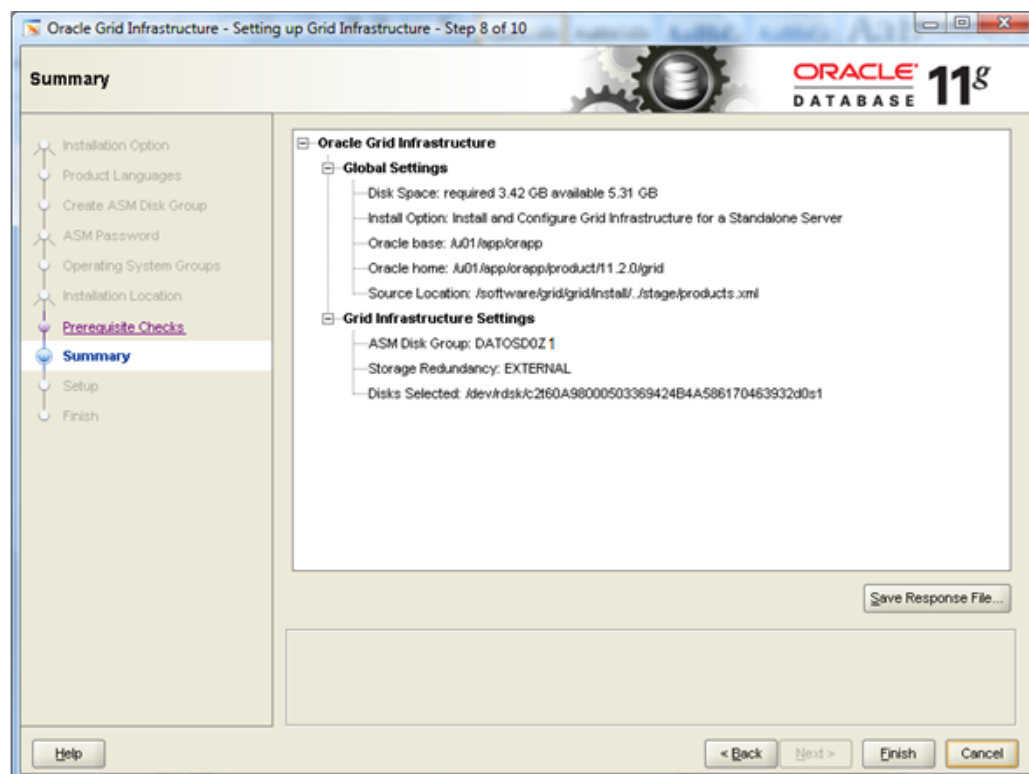
Paso 7: El asistente realizará una revisión de los pre-requisitos antes de empezar la instalación.



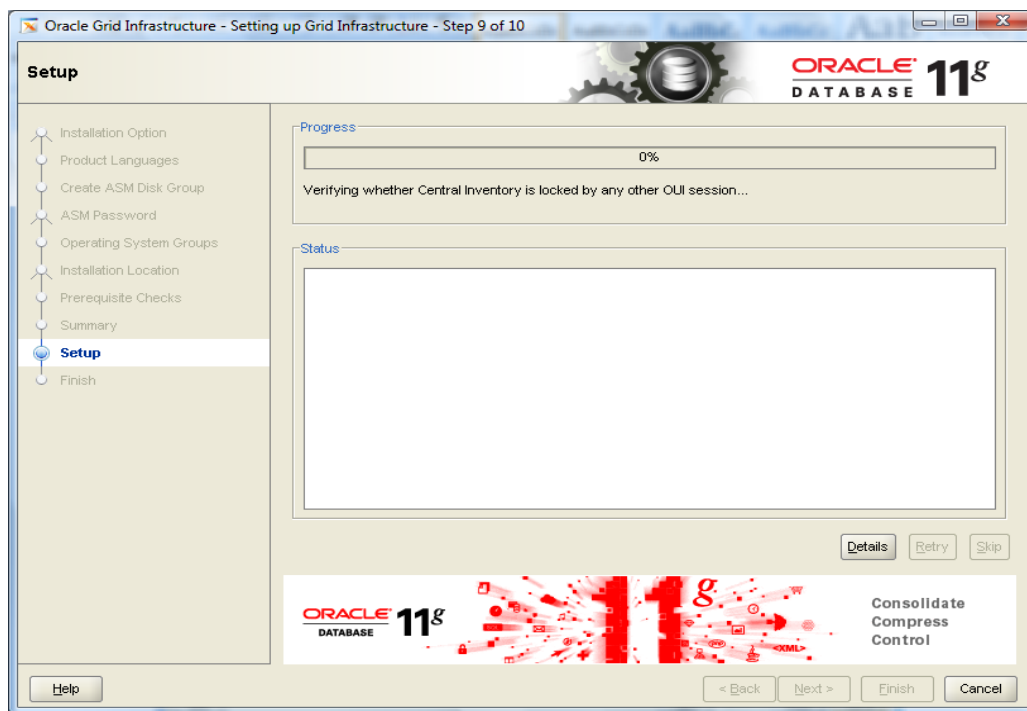
Terminado la revisión de pre-requisitos el asistente manda la siguiente pantalla en la cual muestra que no tenemos un parche instalado, el cual es opcional así que se habilita la casilla Ignore All y dar clic en siguiente.



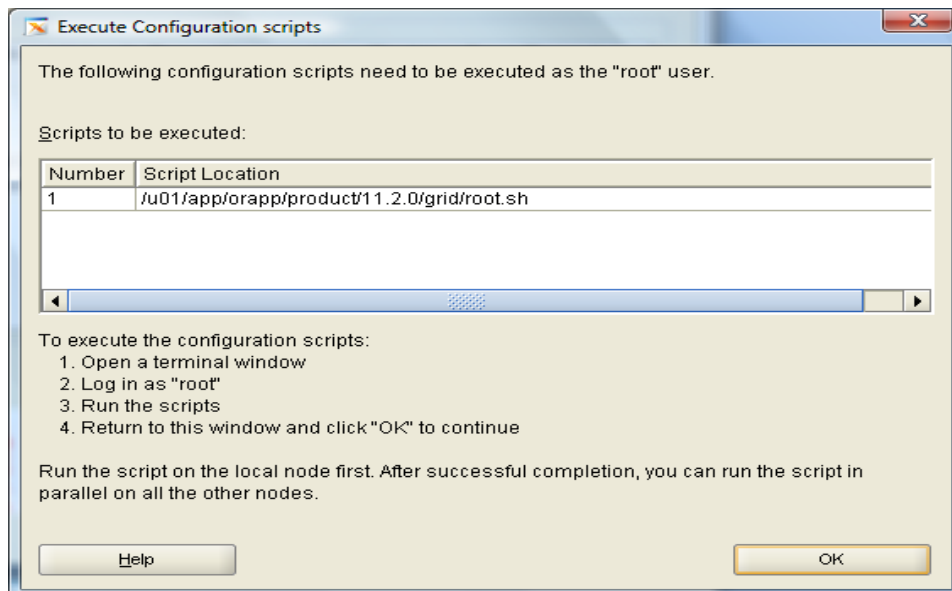
Paso 8: El asistente mandara un resumen antes de empezar la instalación, dar clic en Finish para empezar con ella.



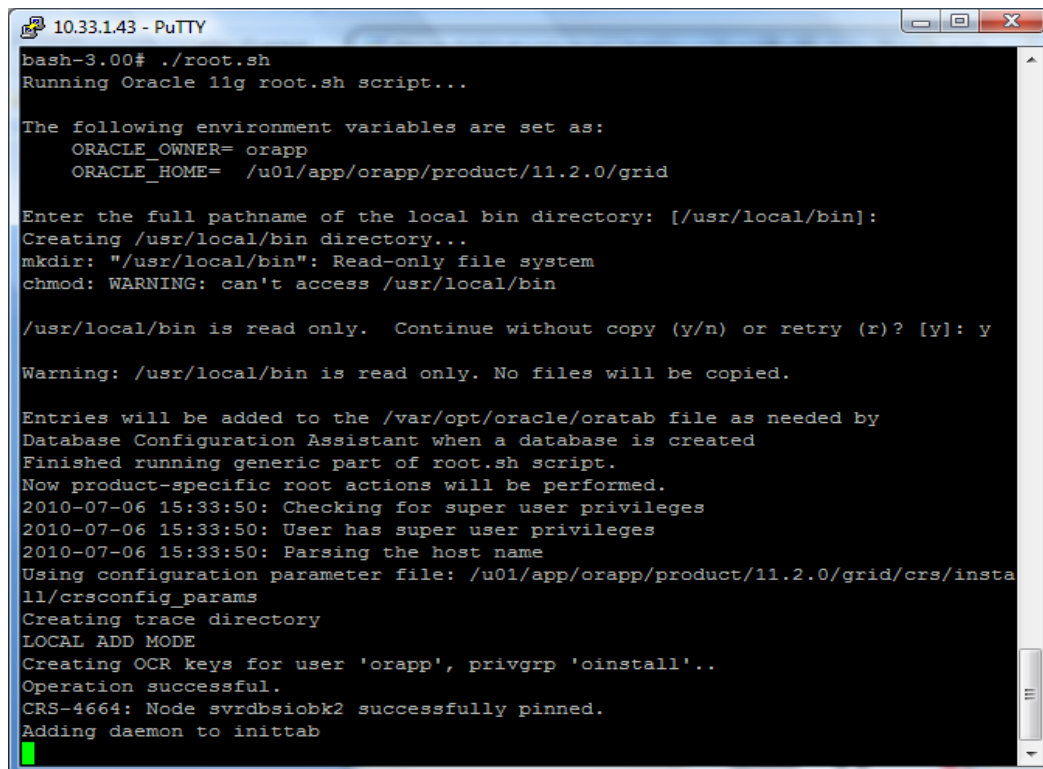
Paso 9: En este paso el asistente muestra el avance de la instalación.



Antes de terminar el asistente pide ejecutar un script como usuario root



Para realizar dicha petición en una nueva ventana y con el usuario root dirigirse hasta la ruta donde se encuentra dicho script y ejecutarlo como se muestra a continuación.



```

10.33.1.43 - PuTTY
bash-3.00# ./root.sh
Running Oracle 11g root.sh script...

The following environment variables are set as:
  ORACLE_OWNER= orapp
  ORACLE_HOME=  /u01/app/orapp/product/11.2.0/grid

Enter the full pathname of the local bin directory: [/usr/local/bin]:
Creating /usr/local/bin directory...
mkdir: "/usr/local/bin": Read-only file system
chmod: WARNING: can't access /usr/local/bin

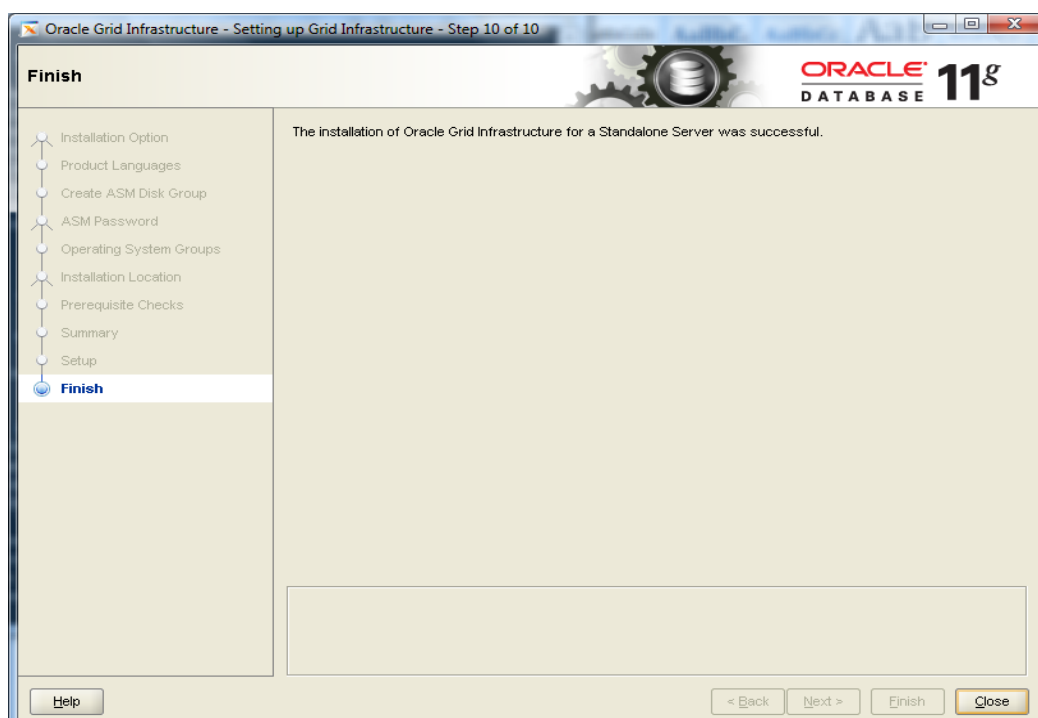
/usr/local/bin is read only. Continue without copy (y/n) or retry (r)? [y]: y

Warning: /usr/local/bin is read only. No files will be copied.

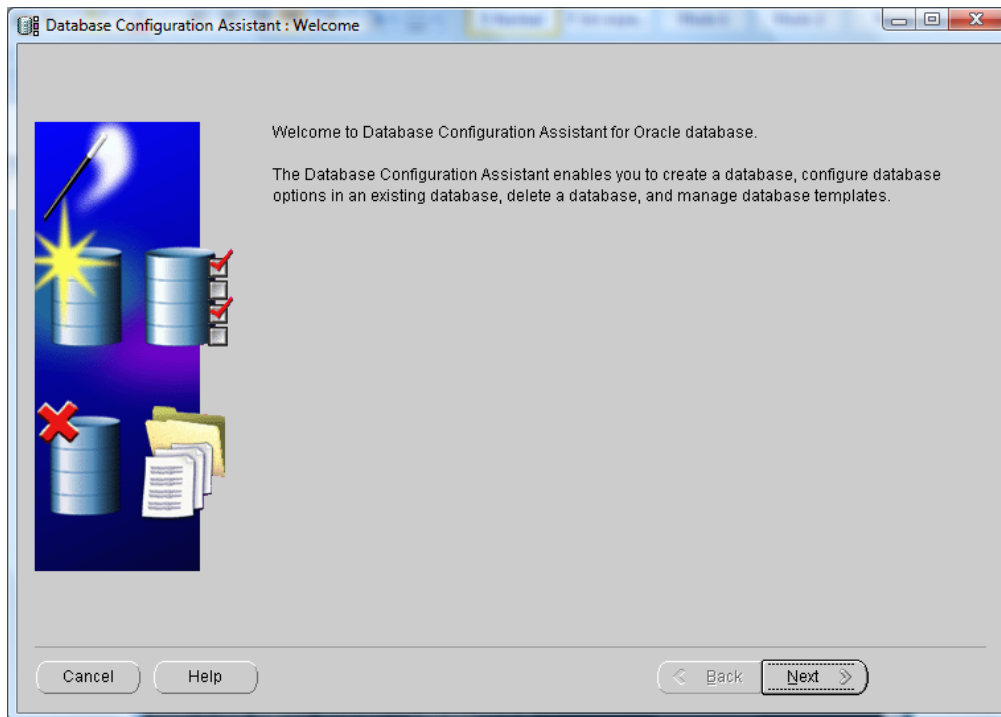
Entries will be added to the /var/opt/oracle/oratab file as needed by
Database Configuration Assistant when a database is created
Finished running generic part of root.sh script.
Now product-specific root actions will be performed.
2010-07-06 15:33:50: Checking for super user privileges
2010-07-06 15:33:50: User has super user privileges
2010-07-06 15:33:50: Parsing the host name
Using configuration parameter file: /u01/app/orapp/product/11.2.0/grid/crs/install/crsconfig_params
Creating trace directory
LOCAL ADD MODE
Creating OCR keys for user 'orapp', privgrp 'oinstall'..
Operation successful.
CRS-4664: Node svrdbsiobk2 successfully pinned.
Adding daemon to inittab

```

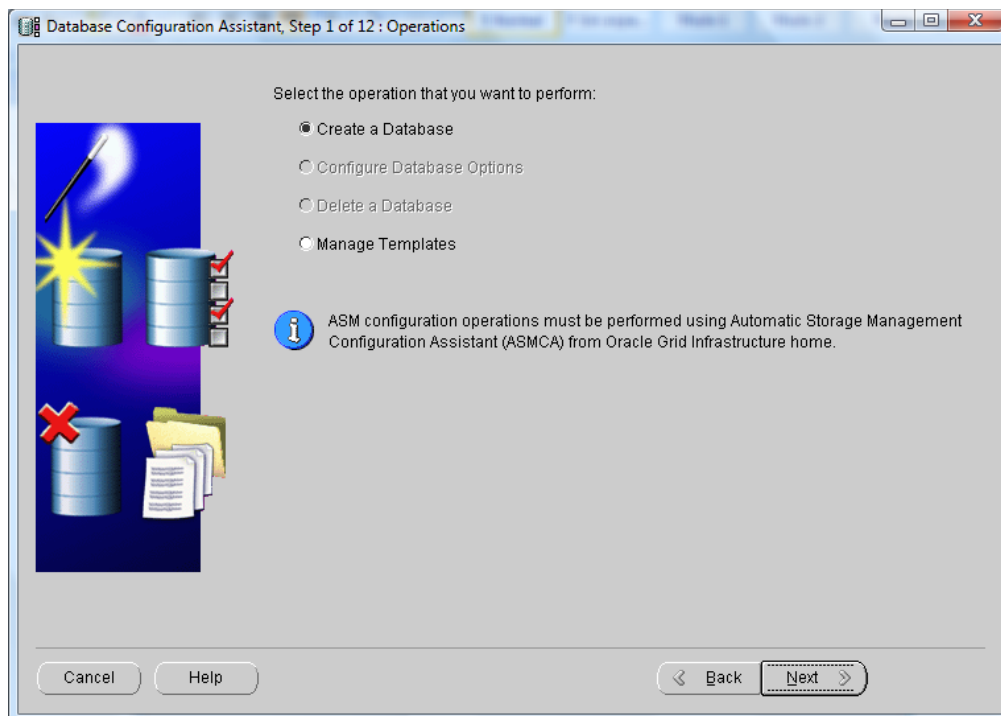
Al haber ejecutado el script damos clic en OK en el mensaje del asistente donde pide ejecutar lo anterior para continuar con la instalación, una vez terminada damos clic en Close para terminar



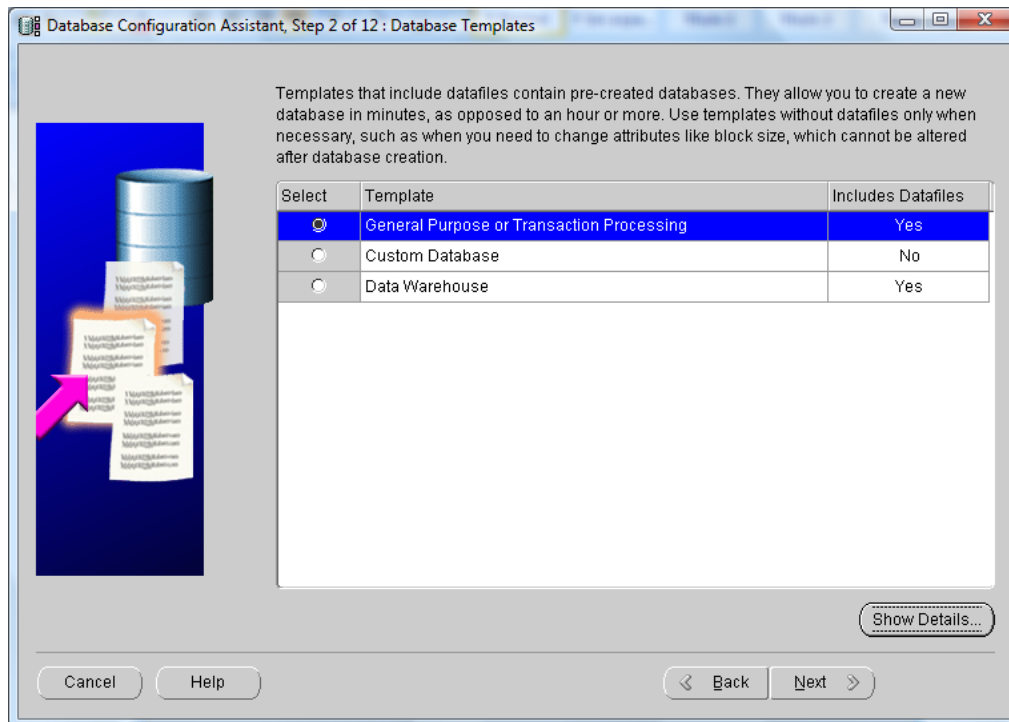
2. El asistente mostrara la pantalla de bienvenida, dar clic en next.



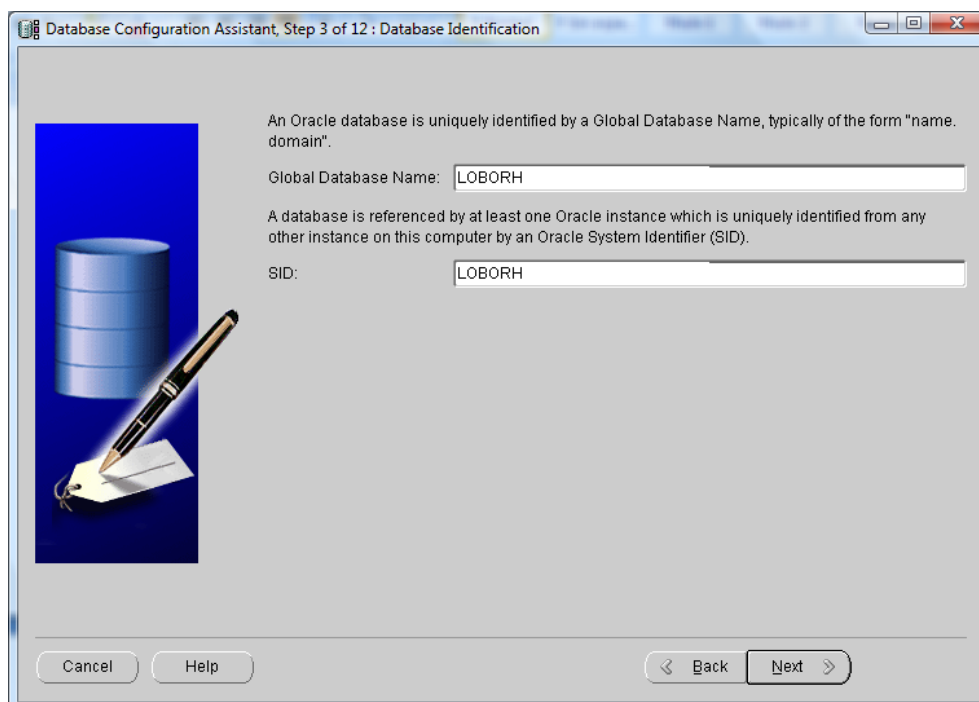
3. Seleccionar la opción Create a Database y dar clic en next.



4. Elegir la opción General Purpose or Transaction Processing del menú de plantillas de base de datos.

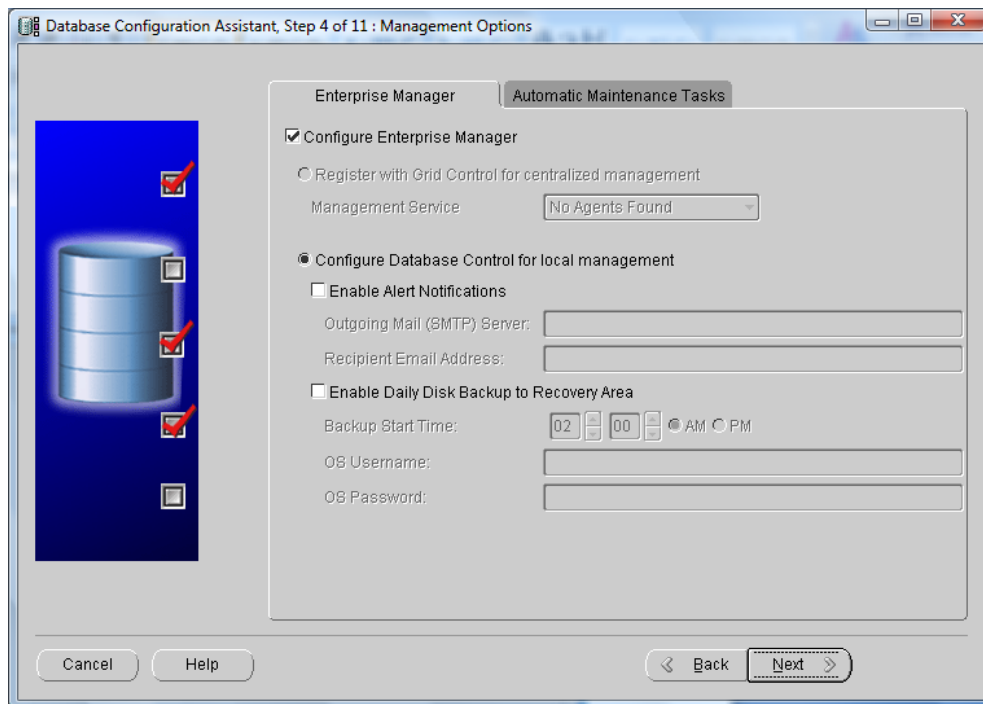


5. Ingresar el nombre global de la instancia y el SID igual que en la base origen.

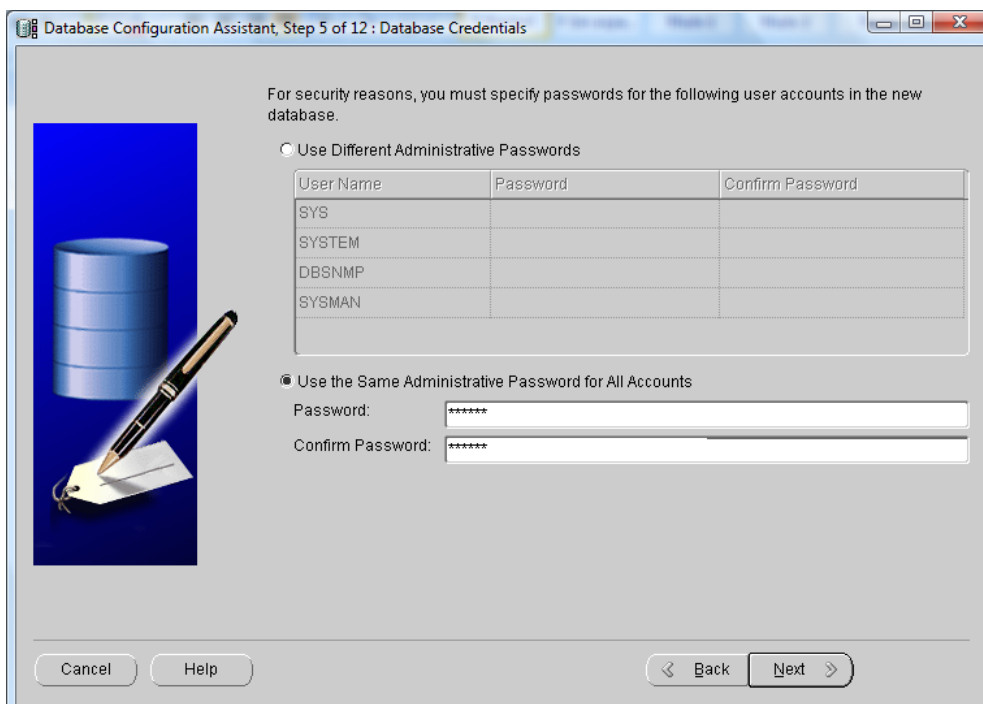


NOTA:Al final de la creación de la instancia se encuentra un cuadro que indica los distintos nombres de las bases de datos que se crearán, el cambio correspondiente a esta pantalla se encuentra en la columna “NOMBRE DE LA INSTANCIA”.

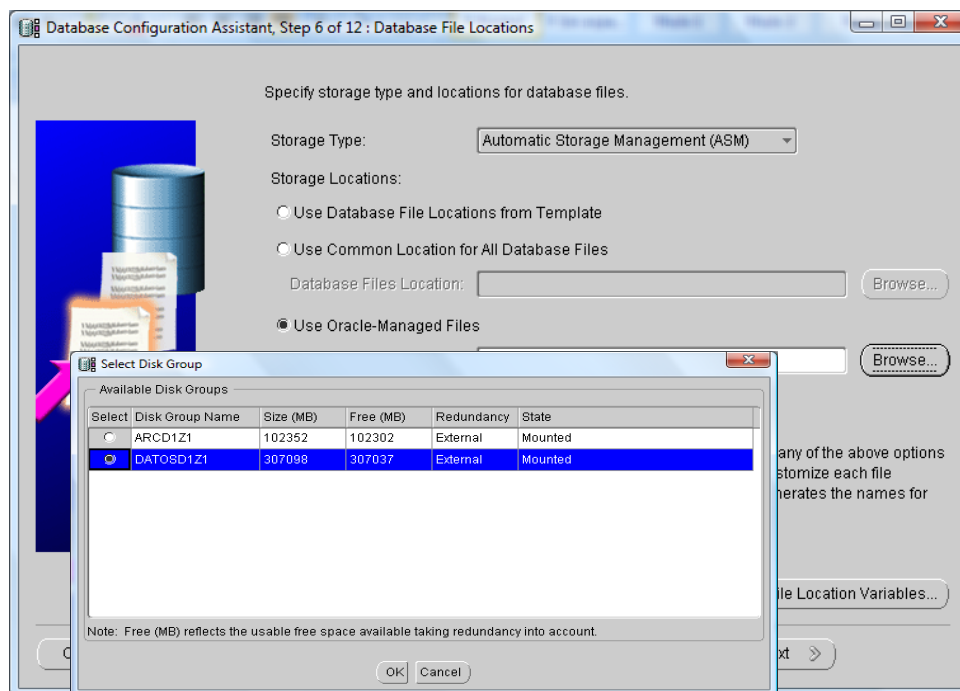
6. Dejar habilitado por default la opción de opciones de Administración.



7. Definir la contraseña para los diferentes usuarios de administración, la cual será la misma para todos.

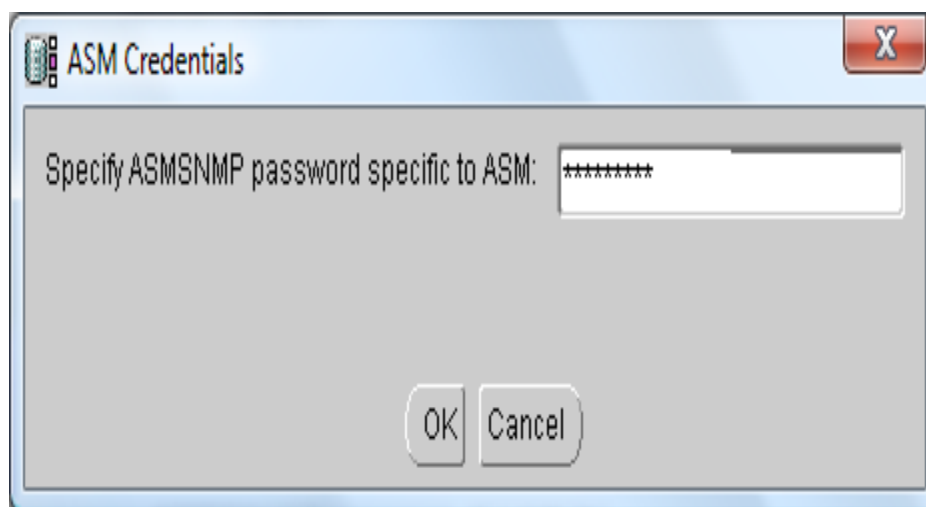


8. En las opciones de ubicación de archivos de la base de datos seleccionar el tipo como ASM, en el campo use Oracle-Managed Files dar clic en browse y seleccionar el disco llamado DATOSD1Z1 y dar ok

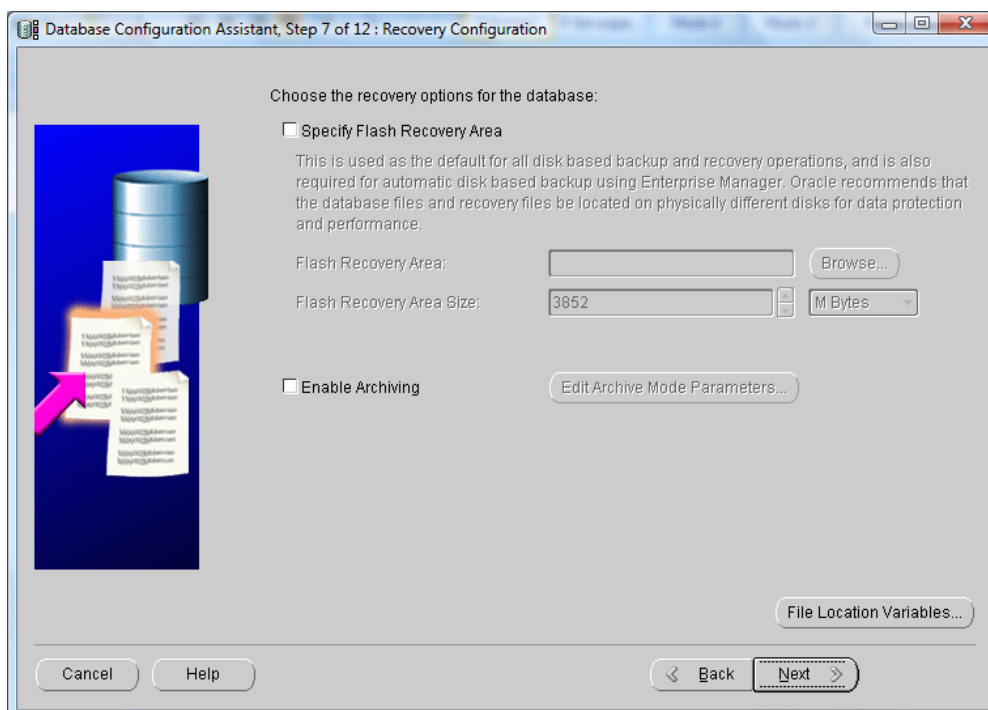


NOTA: Esta es otra de las pantallas que tiene cambio de acuerdo a la instancia que se crea, se identificará en el cuadro como "GRUPO DE DISCOS".

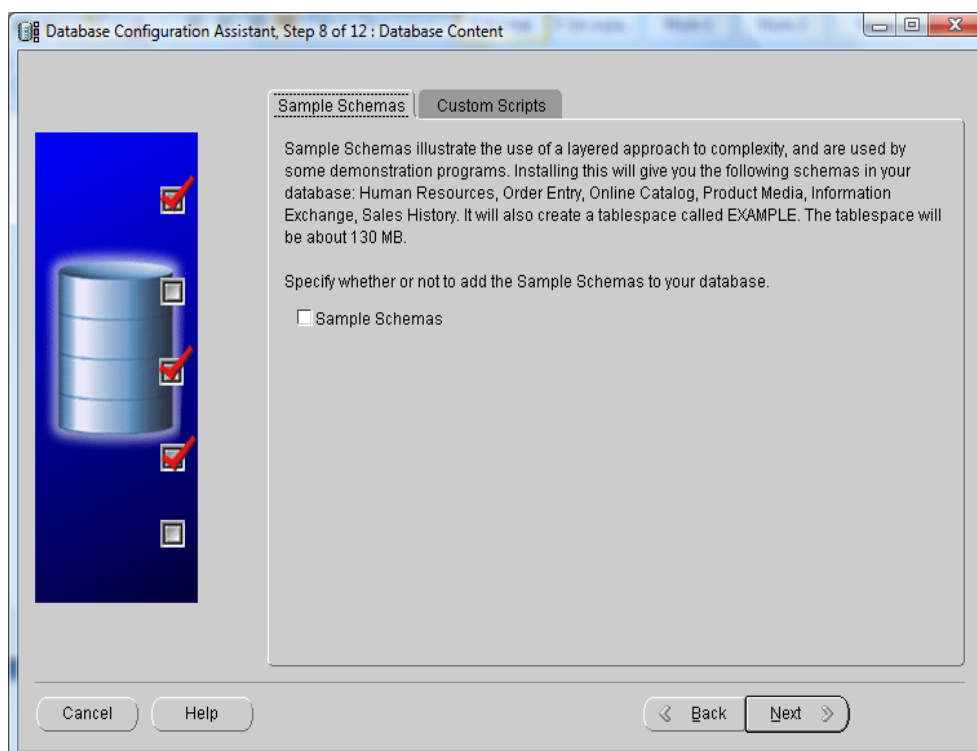
9. Dar clic en next en la pantalla, pedirá ingresar la clave de la instancia de asm.



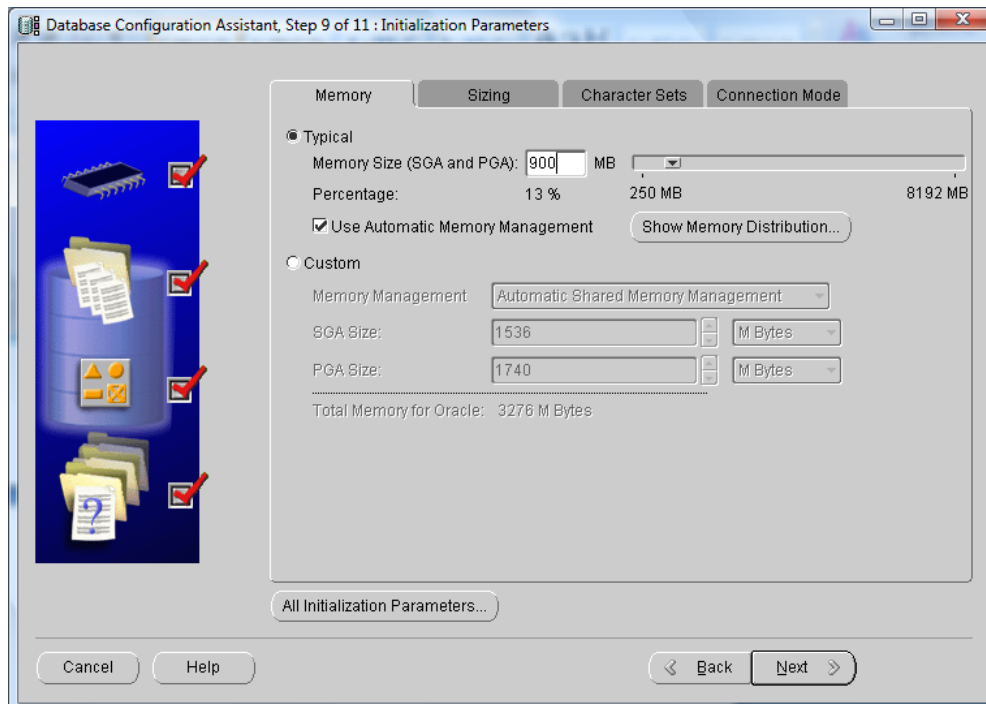
10. Deshabilitar la opción de Specify Flash Recovery Area, dar clic en next.



11. Dejar sin habilitar la opción de schemas de ejemplo.

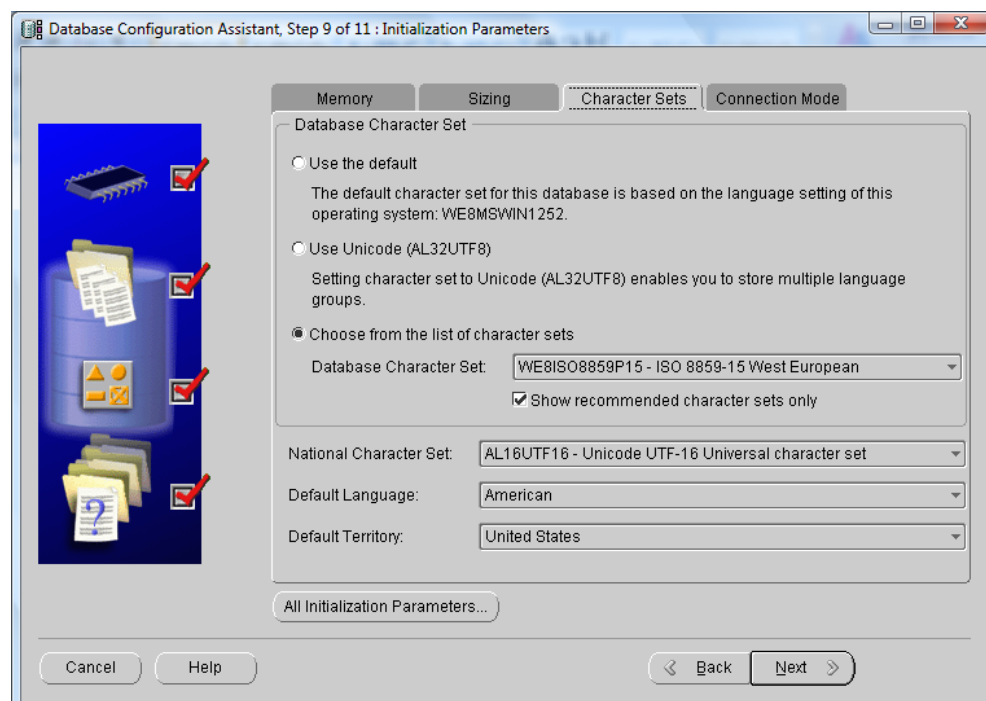


12. Definir el tamaño de memoria para la SGA en 900 MB.



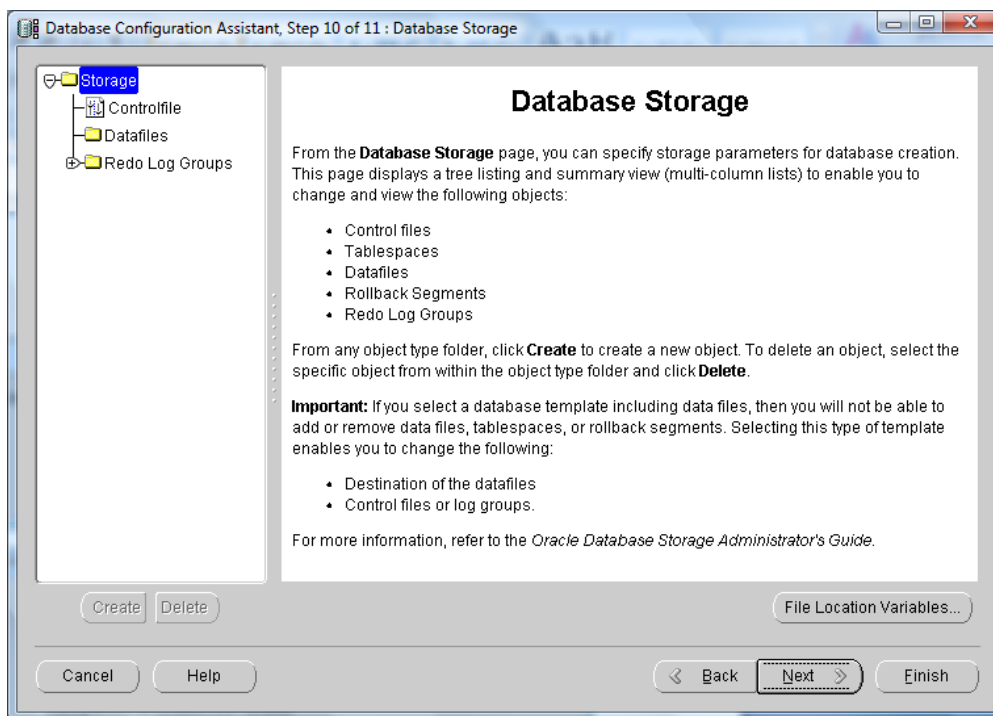
NOTA: Para cada base de datos se asigna un espacio de memoria distinto, el cual se identificara en el cuadro como “SGA”.

13. Se define el set de caracteres con los siguientes valores.

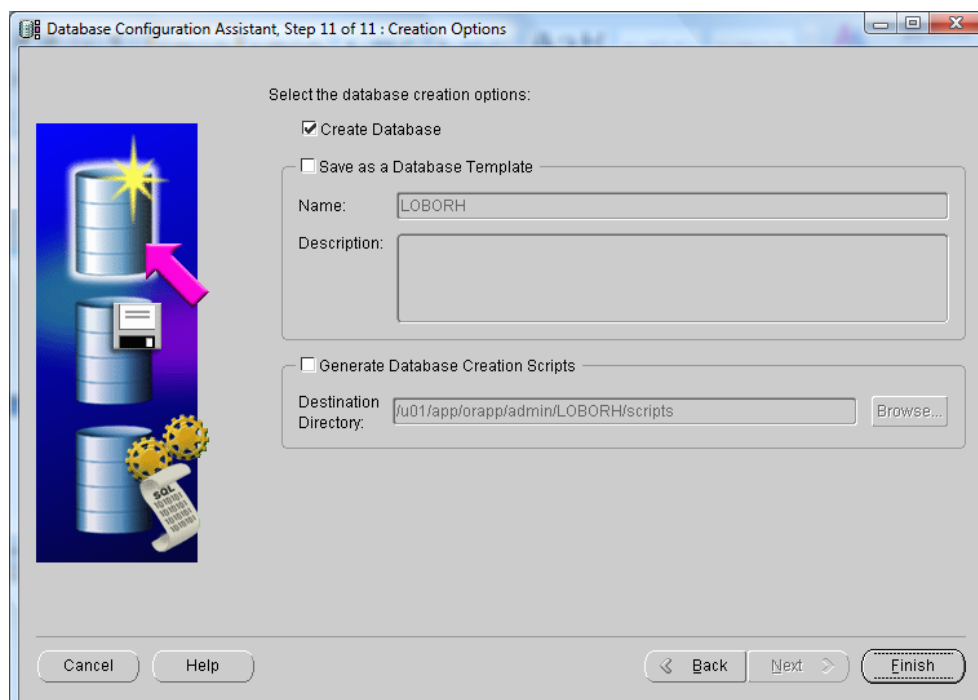


NOTA: El set de caracteres puede variar en las instancias, será identificado en el cuadro como “SET DE CARACTERES”.

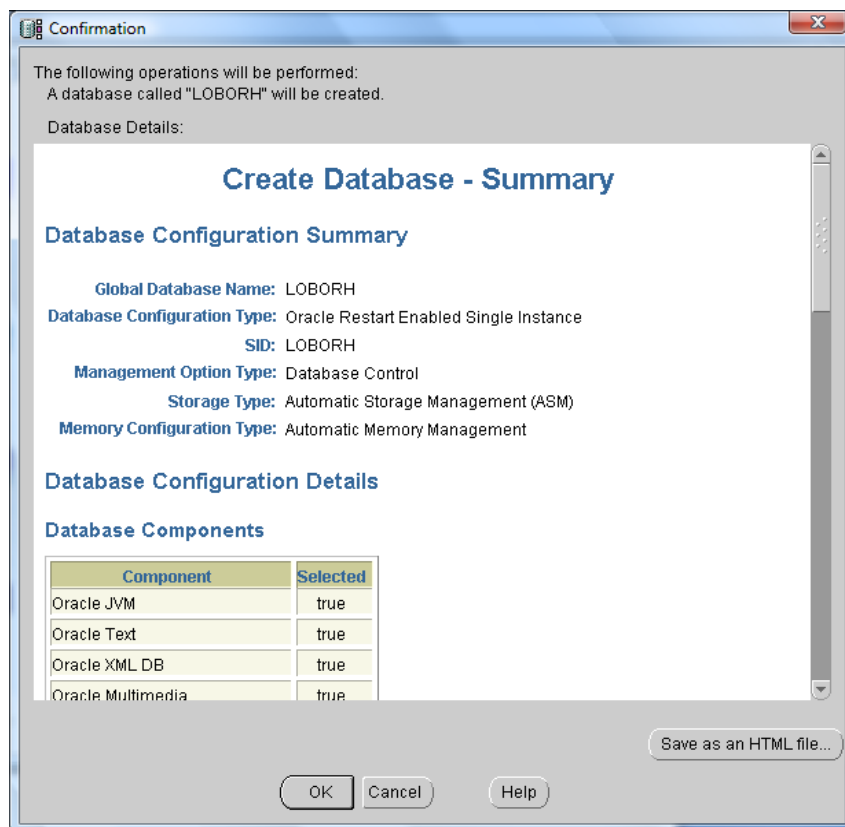
14. Antes de empezar la creación el asistente manda un resumen de la estructura de almacenamiento que tendrá la nueva instancia.



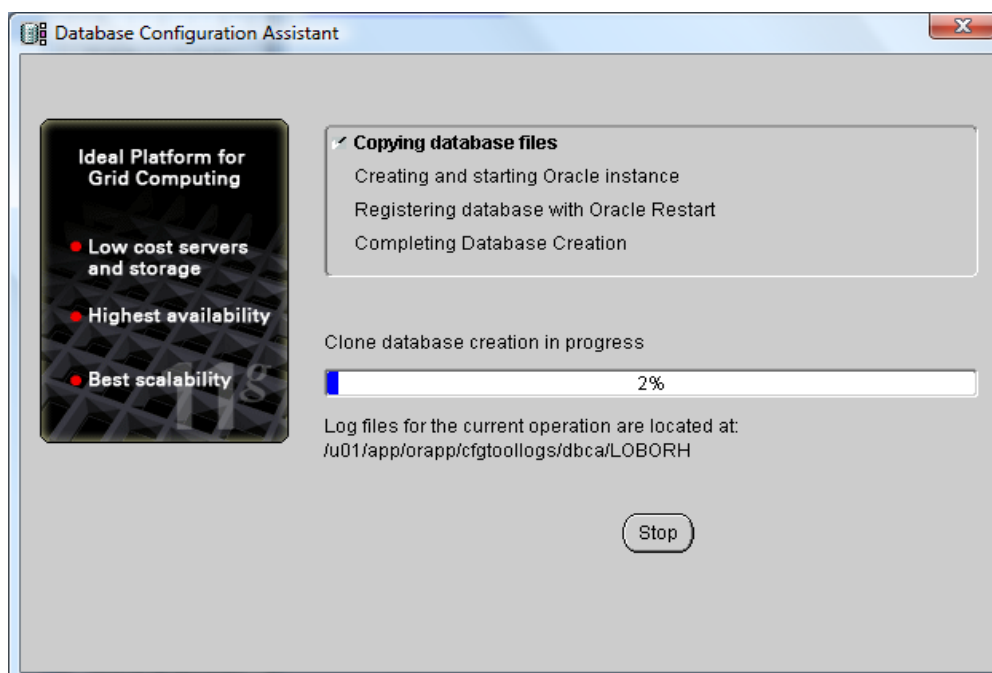
15. Dar clic en el botón Finish para empezar la creación de la nueva instancia.



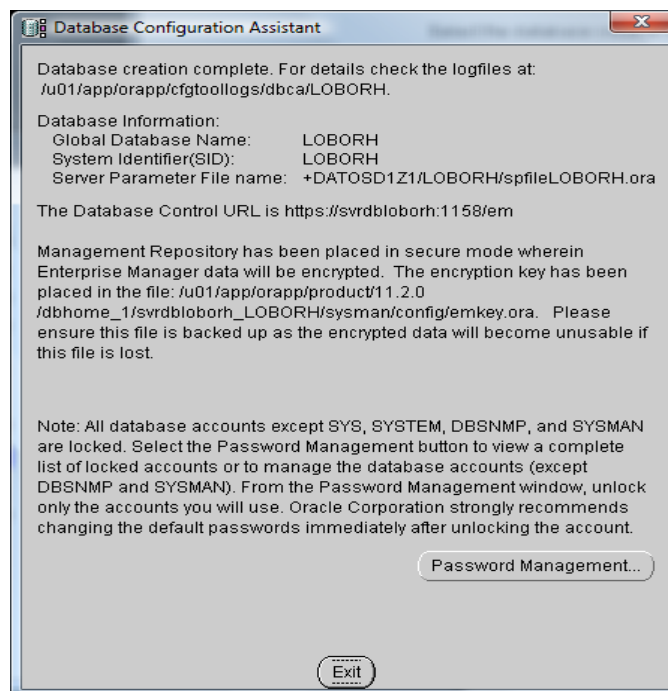
16. El asistente desplegará un resumen detallado de la creación de la instancia dar OK.



17. Proceso de creación de la nueva instancia.



18. Al termino de la creación desplegara el asistente la siguiente pantalla dar clic en Exit para terminar.



NOTA: La pantalla anterior varía de acuerdo al nombre de la instancia que se creo y algunas de sus características.

En el siguiente cuadro se mencionan los cambios que se presentan durante la creación de las distintas bases de datos.

NOMBRE DE LA INSTANCIA	GRUPO DE DISCOS	SGA	SET DE CARACTERES
LOBORH	DATOSD1Z1	900 MB	WE8ISO8859P15
LOBOSYS	DATOSD1Z1	500 MB	WE8ISO8859P15
LOBORH08	DATOSD1Z1	1024 MB	WE8ISO8859P15
LOBORH09	DATOSD1Z1	1024 MB	WE8ISO8859P15
SIA	DATODD1Z2	1536 MB	WE8MSIN1252
SIAADM	DATOSD1Z2	800 MB	WE8MSIN1252
SISADM	DATOSD1Z2	1024 MB	WE8MSIN1252
SIAM	DATOSD0Z2	1024 MB	WE8ISO8859P15
VIDECUFI	DATOSD0Z2	1024 MB	WE8ISO8859P15
REMESAS	DATOSD0Z2	1024 MB	WE8ISO8859P15
MAE	DATOSD0Z2	800 MB	WE8ISO8859P15
ESTAD	DATOSD0Z2	800 MB	WE8ISO8859P15
SIO	DATOSD0Z2	1024 MB	WE8ISO8859P15
SIOBK2	DATOSD0Z1	1536 MB	WE8ISO8859P15

Hasta este punto se tiene ya instalado el motor y la instancia de base de datos en la version 11g Release 2 lo siguiente es obtener los datos en la BD origen, este proceso de describira en el siguiente punto.

2.5 MIGRACIÓN DE LA BASE DE DATOS LOBORH

OBTENCIÓN DE DATOS EN LA BASE ORIGEN.

Para realizar dicha tarea es necesario ejecutar unos scripts en la base origen de acuerdo a la siguiente tabla.

Tabla de ejecución de scripts

No.	Información	Script
1	tablespaces	> busca_ts.sh > crea_ts2.sh
2	perfiles	> Crprofs.sql
3	roles	> Crroles.sql
4	usuarios	> Crusers.sql
5	grants de objetos	> Crgrants.sql
6	grants de sistema	> Crgrantspostsys.sql
7	links	> Crlinks.sql
8	sinónimos	> Crsyns.sql

La ejecución del script es como sigue: busca_ts.sh + (+DATOSD1Z1) + instancia a nivel de S.O., este nos permite extraer todos los tablespaces de la base de datos, el contenido de dicho script es el siguiente:

```
if [ $# != 2 ]
then
    echo
    echo "NUMERO DE PARAMETROS ERRONEO"
    echo
    echo "$SINTAXIS $0 SID DISKGROUP"
    echo
    exit;
fi

DG=$2

sqlplus "/as sysdba" << EOF
set head off
set feed off
set pagesize 0
spool datafiles.txt
select TABLESPACE_NAME||'|' ||$DG||'|' ||sum(BYTES/(1024*1024)) MBYTES from dba_data_files
group by TABLESPACE_NAME;
exit
EOF

cat datafiles.txt | grep -v ">" > datafiles_$.txt
rm -f datafiles.txt
```

Después se ejecuta el siguiente script de la tabla como sigue crea_ts2.sh + archivo generado del script anterior, el contenido de dicho script es el siguiente:

```
#!/bin/sh

if [ $# != 1 ]
then
    echo
    echo "NUMERO DE PARAMETROS ERRONEO"
    echo
    echo "SINTAXIS $0 lista_datafiles"
    echo
    exit;
fi

SID=`echo $1 | cut -d"_" -f2 | cut -d "." -f1`

echo $SID

for i in `cat $1`
do
    tablesp=`echo $i|awk -F"|" '{ print $1}'`
    datafile=`echo $i|awk -F"|" '{print $2}'`
    espacio=`echo $i|awk -F"|" '{ print $3}'`

    #tablesp=`echo $i|cut -d"|" -f1`
    #datafile=`echo $i|cut -d "|" -f2`
    #espacio=`echo $i|cut -d"|" -f3`

    echo $tablesp $datafile $espacio
    echo "
CREATE TABLESPACE ${tablesp}
DATAFILE '$datafile' SIZE $espacio M REUSE AUTOEXTEND OFF
EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO;
" >> crea_ts_${SID}.sql
Done
```

Para ejecutar los siguientes scripts es necesario conectarse a una sesión de sql con el usuario sysdba, una vez en la sesión de sql, ejecutar el script Crprofs.sql, dicho script nos permite sacar todos los perfiles existentes en la base de datos, su contenido es el siguiente.

```
rem $DBA/crprofs.sql
rem
rem This script generates another script that will include all the CREATE
rem PROFILE statements for those in the database.
rem
rem This script must be run by a user with the DBA role under Oracle7.
rem
rem Assumes that there is an OPS$ account for the user running this script.
rem
rem Created 09/09/97 by Brian Lomasky
rem
set verify off
set feedback off
set echo off
set pagesize 0
set termout on
select 'Creating profile build script...' from dual;

create table prof_temp (      lineno NUMBER,
                             text VARCHAR2(80));

declare
    cursor prof_cursor is select
        profile,
        resource_name,
        limit
    from sys.dba_profiles
    where profile <> 'DEFAULT'
    order by 1, 2;
```

```

lv_profile          sys.dba_profiles.profile%TYPE;
lv_resource_name    sys.dba_profiles.resource_name%TYPE;
lv_limit           sys.dba_profiles.limit%TYPE;
lv_lineno          number;
a_lin              varchar2(80);
prev_profile       sys.dba_profiles.profile%TYPE;

function wri(x_lin in varchar2, x_str in varchar2,
            x_force in number) return varchar2 is
begin
    if length(x_lin) + length(x_str) > 80
    then
        lv_lineno := lv_lineno + 1;
        insert into prof_temp values (lv_lineno, x_lin);
        if x_force = 0
        then
            return '    ' || x_str;
        else
            lv_lineno := lv_lineno + 1;
            if substr(x_lin,1,2) = '  '
            then
                insert into prof_temp values (
                    lv_lineno, x_str);
            else
                insert into prof_temp values (
                    lv_lineno, '    ' || x_str);
            end if;
            return '';
        end if;
    else
        if x_force = 0
        then
            return x_lin || x_str;
        else
            lv_lineno := lv_lineno + 1;
            insert into prof_temp values (
                lv_lineno, x_lin || x_str);
            return '';
        end if;
    end if;
end wri;

begin
a_lin := '';
prev_profile := '@';
open prof_cursor;
loop
    fetch prof_cursor into
        lv_profile,
        lv_resource_name,
        lv_limit;
    exit when prof_cursor%NOTFOUND;
    if prev_profile = lv_profile then
        a_lin := wri(a_lin, '    ' || lv_resource_name, 0);
        a_lin := wri(a_lin, ' ', 0);
        a_lin := wri(a_lin, lv_limit, 1);
    else
        if prev_profile != '@' then
            a_lin := wri(a_lin, ';', 1);
        end if;
        a_lin := wri(a_lin, 'CREATE PROFILE ', 0);
        a_lin := wri(a_lin, lv_profile, 0);
        a_lin := wri(a_lin, ' limit', 1);
        a_lin := wri(a_lin, '    ' || lv_resource_name, 0);
        a_lin := wri(a_lin, ' ', 0);
        a_lin := wri(a_lin, lv_limit, 1);
        prev_profile := lv_profile;
    end if;
end loop;
close prof_cursor;
if prev_profile != '@' then
    a_lin := wri(a_lin, ';', 1);
end if;

end;
/
set termout off
set heading off

```

```

spool cr_prof.sql
select 'rem cr_prof.sql' from dual;
select 'rem' from dual;
select 'rem ***** All profiles for database ' || name from v$database;
select 'rem' from dual;
select 'set feedback off' from dual;
select text from prof_temp order by lineno;
spool off
drop table prof_temp;
set termout on
select 'Created cr_prof.sql...' from dual;
set termout off
exit

```

El siguiente script a ejecutar es el Crroles.sql nos permite sacar todos los roles , el contenido de dicho script es el siguiente:

```

rem $DBA/crroles.sql
rem
rem This script generates another script that will include all the CREATE ROLE
rem statements for those in the database.
rem
rem This script must be run by a user with the DBA role under Oracle7.
rem
rem Assumes that there is an OPS$ account for the user running this script.
rem
rem Created 09/09/97 by Brian Lomasky
rem
set verify off
set feedback off
set echo off
set pagesize 0
set termout on
define q=chr(39)
select 'Creating role build script...' from dual;

create table role_temp (      lineno NUMBER,
                             text VARCHAR2(80));

declare
    cursor role_cursor is select
        name,
        password
        from sys.user$
        where type# = 0 and name not in ('ADAMS', 'CTXSYS', 'DBSNMP', 'HR',
'LBACSYS', 'MDSYS', 'ODM', 'ODM_MTR',
'ORDPLUGINS', 'ORDSYS', 'OUTLN', 'PM', 'QS', 'QS_ADM', 'QS_CB',
'QS_CBADM', 'QS_CS', 'QS_ES',
'QS_OS', 'QS_WS', 'SCOTT', 'SH', 'SYS', 'SYSTEM', 'WKPROXY', 'WKSYS',
'WMSYS', 'XDB', 'ANONYMOUS',
'DIP',
'DMSYS', 'EXFSYS', 'MDDATA', 'MGMT_VIEW', 'OE', 'OLAPSYS', 'SI_INFORMTN_SCHEMA',
'AQ_ADMINISTRATOR_ROLE', 'DBA', 'EXP_FULL_DATABASE', 'IMP_FULL_DATABASE', 'EXECUTE_CATA
LOG_ROLE',
'CONNECT', 'RESOURCE', 'OEM_MONITOR', 'OLAP_DBA', 'OLAP_USER', 'WKUSER', 'SELECT_CATALO
G_ROLE')
        order by 1;

    lv_name          sys.user$.name%TYPE;
    lv_password      sys.user$.password%TYPE;
    lv_lineno        number;
    a_lin            varchar2(80);

    function wri(x_lin in varchar2, x_str in varchar2,
        x_force in number) return varchar2 is
    begin
        if length(x_lin) + length(x_str) > 80
        then
            lv_lineno := lv_lineno + 1;
            insert into role_temp values (lv_lineno, x_lin);
            if x_force = 0

```

```

        then
            return '    ' || x_str;
        else
            lv_lineno := lv_lineno + 1;
            if substr(x_lin,1,2) = '  '
            then
                insert into role_temp values (
                    lv_lineno, x_str);
            else
                insert into role_temp values (
                    lv_lineno, '    ' || x_str);
            end if;
            return '';
        end if;
    else
        if x_force = 0
        then
            return x_lin || x_str;
        else
            lv_lineno := lv_lineno + 1;
            insert into role_temp values (
                lv_lineno, x_lin || x_str);
            return '';
        end if;
    end if;
end wri;

begin
    a_lin := '';
    open role_cursor;
    loop
        fetch role_cursor into
            lv_name,
            lv_password;
        exit when role_cursor%NOTFOUND;
        a_lin := wri(a_lin, 'CREATE ROLE ', 0);
        a_lin := wri(a_lin, lv_name, 0);
        if nvl(lv_password, 'NO') = 'NO' then
            a_lin := wri(a_lin, ' not identified;', 1);
        elsif lv_password = 'EXTERNAL' then
            a_lin := wri(a_lin, ' identified externally;', 1);
        else
            a_lin := wri(a_lin, ' identified by values ', 0);
            a_lin := wri(a_lin, &q || lv_password || &q || ';', 1);
        end if;
    end loop;
    close role_cursor;
end;
/

set termout off
set heading off
spool cr_role.sql
select 'rem cr_role.sql' from dual;
select 'rem' from dual;
select 'rem ***** All roles for database ' || name from v$database;
select 'rem' from dual;
select 'set feedback off' from dual;
select text from role_temp order by lineno;
spool off
drop table role_temp;
set termout on
select 'Created cr_role.sql...' from dual;
set termout off
exit

```

Ahora ejecutar el script Crusers.sql para sacar todos los usuarios, su contenido es el siguiente:

```

rem $DBA/crusers.sql
rem
rem This script generates another script that will include all the CREATE USER
rem statements for those in the database (except for the SYS user).
rem
rem Note: Don't forget the user grants and synonyms.

```

```

rem
rem This script must be run by a user with the DBA role under Oracle7.
rem
rem Created 10/21/96 by Brian Lomasky
rem
set verify off
set feedback off
set echo off
set pagesize 0
set termout on
define q=chr(39)
select 'Creating user build script...' from dual;

create table u_temp ( lineno NUMBER,
                    text VARCHAR2(80));

declare
    cursor user_cursor is select
        username,
        password,
        default_tablespace,
        temporary_tablespace,
        profile
    from sys.dba_users
    where username not in ('ADAMS', 'CTXSYS', 'DBSNMP', 'HR', 'LBACSYS',
'MDSYS', 'ODM', 'ODM_MTR',
'ORDPLUGINS', 'ORDSYS', 'OUTLN', 'PM', 'QS', 'QS_ADM', 'QS_CB',
'QS_CBADM', 'QS_CS', 'QS_ES',
'QS_OS', 'QS_WS', 'SCOTT', 'SH', 'SYS', 'SYSTEM', 'WKPROXY', 'WKSYS',
'WMSYS', 'XDB', 'ANONYMOUS',
'DIP',
'DMSYS','EXFSYS','MDDATA','MGMT_VIEW','OE','OLAPSYS','SI_INFORMTN_SCHEMA',

        'AQ_ADMINISTRATOR_ROLE','DBA','EXP_FULL_DATABASE','IMP_FULL_DATABASE','EXECUTE_CATALOG_ROLE',

        'CONNECT','RESOURCE','OEM_MONITOR','OLAP_DBA','OLAP_USER','WKUSER','SELECT_CATALOG_ROLE')
    order by username;

    cursor quo_cursor (c_user varchar2) is select
        tablespace_name, max_bytes
    from sys.dba_ts_quotas
    where username = c_user;

    lv_username          sys.dba_users.username%TYPE;
    lv_password          sys.dba_users.password%TYPE;
    lv_default_tbl       sys.dba_users.default_tablespace%TYPE;
    lv_temp_tbl          sys.dba_users.temporary_tablespace%TYPE;
    lv_profile           sys.dba_users.profile%TYPE;
    lv_tablespace_name   sys.dba_ts_quotas.tablespace_name%TYPE;
    lv_max_bytes         sys.dba_ts_quotas.max_bytes%TYPE;
    lv_string            varchar2(80);
    lv_lineno            number;

    procedure write_out is
    begin
        lv_lineno := lv_lineno + 1;
        insert into u_temp (lineno, text) values (lv_lineno, lv_string);
    end;

begin
    lv_lineno := 0;
    open user_cursor;
    loop
        fetch user_cursor into
            lv_username,
            lv_password,
            lv_default_tbl,
            lv_temp_tbl,
            lv_profile;
        exit when user_cursor%NOTFOUND;
        lv_string := 'CREATE USER '||lv_username||
            ' identified by values '||&q||lv_password||&q;
        write_out;
        lv_string := '        default tablespace '||lv_default_tbl;
        write_out;
    end loop;
end;

```

```

lv_string := '    temporary tablespace '||lv_temp_tbl;
write_out;
open quo_cursor(lv_username);
loop
    fetch quo_cursor into lv_tablespace_name, lv_max_bytes;
    exit when quo_cursor%NOTFOUND;
    if lv_max_bytes = -1 then
        lv_string := '    quota unlimited on '||
            lv_tablespace_name;
    elsif mod(lv_max_bytes, 1048576) = 0 then
        lv_string := '    quota '||
            lv_max_bytes/1048576||
            'M on '||lv_tablespace_name;
    elsif mod(lv_max_bytes, 1024) = 0 then
        lv_string := '    quota '||lv_max_bytes/1024||
            'K on '||lv_tablespace_name;
    else
        lv_string := '    quota '||lv_max_bytes||
            ' on '||lv_tablespace_name;
    end if;
    write_out;
end loop;
close quo_cursor;
lv_string := '    profile '||lv_profile||';';
write_out;
end loop;
close user_cursor;
end;
/
set termout off
set heading off
spool cr_user.sql
select 'rem cr_user.sql' from dual;
select 'rem' from dual;
select 'rem ***** All users for database ' || name from v$database;
select 'rem' from dual;
select 'set feedback off' from dual;
select text from u_temp order by lineno;
spool off
drop table u_temp;
set termout on
select 'Created cr_user.sql...' from dual;
set termout off
exit

```

El siguiente script a ejecutar es el Crgrants.sql para sacar los grants de objetos su contenido se muestra a continuación:

```

rem $DBA/crgrants.sql
rem
rem This script generates another script that will include all the GRANT
rem statements for those in the database (except for the SYS user).
rem
rem Note that this script will include the OBJECT privilege grants, as well
rem as the SYSTEM privilege grants in the single output file. You may wish
rem to edit the output if you need to separate the different types of grants.
rem
rem This script must be run by a user with the DBA role under Oracle7.
rem
rem Assumes that there is an OP$ account for the user running this script.
rem
rem Created 10/30/97 by Brian Lomasky
rem
set verify off
set feedback off
set echo off
set pagesize 0
set serveroutput on size 100000
set termout on
define q=chr(39)
select 'Creating grant build script...' from dual;

create table g_temp ( lineno NUMBER,

```

```

text VARCHAR2(80));

declare
cursor obj_cursor is select
  grantee,
  owner,
  table_name,
  privilege,
  decode(grantable,'YES',' WITH GRANT OPTION;',';')
from sys.dba_tab_privs
where grantor = 'PRUEBA'
order by 2, 3, 1, 4;

cursor col_cursor is select
  grantee,
  owner,
  table_name,
  column_name,
  privilege,
  decode(grantable,'YES',' WITH GRANT OPTION;',';')
from sys.dba_col_privs
where grantor = 'PPRUEBA'
order by 2, 3, 4, 5, 1;

cursor sys_cursor is select
  grantee,
  privilege,
  decode(admin_option,'YES',' WITH ADMIN OPTION;',';')
from sys.dba_sys_privs
where grantee not in ('ADAMS', 'CTXSYS', 'DBSNMP', 'HR', 'LBACSYS',
'MDSYS', 'ODM', 'ODM_MTR',
'ORDPLUGINS', 'ORDSYS', 'OUTLN', 'PM', 'QS', 'QS_ADM', 'QS_CB',
'QS_CBADM', 'QS_CS', 'QS_ES',
'QS_OS', 'QS_WS', 'SCOTT', 'SH', 'SYS', 'SYSTEM', 'WKPROXY', 'WKSYS',
'WMSYS', 'XDB', 'ANONYMOUS',
'DIP',
'DMSYS', 'EXFSYS', 'MDDATA', 'MGMT_VIEW', 'OE', 'OLAPSYS', 'SI_INFORMTN_SCHEMA',
'AQ_ADMINISTRATOR_ROLE', 'DBA', 'EXP_FULL_DATABASE', 'IMP_FULL_DATABASE', 'EXECUTE_CATALOG_ROLE',
'CONNECT', 'RESOURCE', 'OEM_MONITOR', 'OLAP_DBA', 'OLAP_USER', 'WKUSER', 'SELECT_CATALOG_ROLE')
order by 1, 2;

cursor role_cursor is select
  grantee,
  granted_role,
  decode(admin_option,'YES',' WITH ADMIN OPTION;',';')
from sys.dba_role_privs
where grantee not in ('ADAMS', 'CTXSYS', 'DBSNMP', 'HR', 'LBACSYS',
'MDSYS', 'ODM', 'ODM_MTR',
'ORDPLUGINS', 'ORDSYS', 'OUTLN', 'PM', 'QS', 'QS_ADM', 'QS_CB',
'QS_CBADM', 'QS_CS', 'QS_ES',
'QS_OS', 'QS_WS', 'SCOTT', 'SH', 'SYS', 'SYSTEM', 'WKPROXY', 'WKSYS',
'WMSYS', 'XDB', 'ANONYMOUS',
'DIP',
'DMSYS', 'EXFSYS', 'MDDATA', 'MGMT_VIEW', 'OE', 'OLAPSYS', 'SI_INFORMTN_SCHEMA',
'AQ_ADMINISTRATOR_ROLE', 'DBA', 'EXP_FULL_DATABASE', 'IMP_FULL_DATABASE', 'EXECUTE_CATALOG_ROLE',
'CONNECT', 'RESOURCE', 'OEM_MONITOR', 'OLAP_DBA', 'OLAP_USER', 'WKUSER', 'SELECT_CATALOG_ROLE')
order by 1, 2;

cursor pwd_cursor (c_user varchar2) is
  select password from sys.dba_users where username = c_user;

lv_grantee          sys.dba_tab_privs.grantee%TYPE;
lv_owner            sys.dba_tab_privs.owner%TYPE;
lv_table_name       sys.dba_tab_privs.table_name%TYPE;
lv_column_name      sys.dba_col_privs.column_name%TYPE;
lv_privilege        sys.dba_tab_privs.privilege%TYPE;
lv_granted_role     sys.dba_role_privs.granted_role%TYPE;
lv_grantable        varchar2(19);
lv_string           varchar2(80);

```



```

lv_lineno          number;
a_lin              varchar2(80);
prev_grantee       sys.dba_tab_privs.grantee%TYPE;
prev_own           sys.dba_tab_privs.owner%TYPE;
alter_user         sys.dba_tab_privs.owner%TYPE;
prev_owner         sys.dba_tab_privs.owner%TYPE;
prev_table_name    sys.dba_tab_privs.table_name%TYPE;
prev_column_name   sys.dba_col_privs.column_name%TYPE;
prev_grantable     varchar2(19);
privs              varchar2(100);
user_password      sys.dba_users.password%TYPE;
connect_pwd        varchar2(10);

function wri(x_lin in varchar2, x_str in varchar2,
            x_force in number) return varchar2 is
begin
    if length(x_lin) + length(x_str) > 80
    then
        lv_lineno := lv_lineno + 1;
        insert into g_temp values (lv_lineno, x_lin);
        if x_force = 0
        then
            return '    '||x_str;
        else
            lv_lineno := lv_lineno + 1;
            if substr(x_lin,1,2) = '  '
            then
                insert into g_temp values (
                    lv_lineno, x_str);
            else
                insert into g_temp values (
                    lv_lineno, '    '||x_str);
            end if;
            return '';
        end if;
    else
        if x_force = 0
        then
            return x_lin||x_str;
        else
            lv_lineno := lv_lineno + 1;
            insert into g_temp values (
                lv_lineno, x_lin||x_str);
            return '';
        end if;
    end if;
end wri;

begin
a_lin := '';
lv_lineno := 0;
prev_grantee := '@';
prev_own := '@';
prev_owner := '';
prev_table_name := '';
prev_grantable := '';
privs := '';
a_lin := wri(a_lin, 'rem *** Object Privileges ***', 1);
a_lin := '';
open obj_cursor;
loop
    fetch obj_cursor into
        lv_grantee,
        lv_owner,
        lv_table_name,
        lv_privilege,
        lv_grantable;
    exit when obj_cursor%NOTFOUND;
    if prev_grantee = lv_grantee and prev_owner = lv_owner and
       prev_table_name = lv_table_name and
       prev_grantable = lv_grantable
    then
        /* Skip if duplicate privilege found */
        if instr(privs, lv_privilege) = 0 then
            a_lin := wri(a_lin, '    '||lv_privilege, 0);
            privs := privs || lv_privilege;
        end if;
    end if;
end loop;
end;

```

```

else
  if prev_grantee != '@' then
    a_lin := wri(a_lin, ' ON', 0);
    a_lin := wri(a_lin, ' '||prev_owner||'. '||
      prev_table_name, 0);
    a_lin := wri(a_lin, ' TO', 0);
    a_lin := wri(a_lin, ' '||prev_grantee, 0);
    a_lin := wri(a_lin, prev_grantable, 1);
  end if;
  /* */
  /* Since we can not grant privileges on objects we */
  /* do not own, or have been granted privilege to */
  /* with grant option, we need to build in the */
  /* connect strings to switch to the user's schema */
  /* who owns the object */
  /* */
  if lv_owner != prev_own then
    /* Reset password if we changed it */
    if prev_own != '@' then
      a_lin := wri(a_lin, 'connect /', 1);
      if user_password != '<password>' then
        a_lin := wri(a_lin,
          'alter user '||
            lower(prev_own)||
              ' identified by' ||
                ' values '||&q||
                  user_password||&q||';',
          1);
      else
        a_lin := wri(a_lin,
          'alter user '||
            lower(prev_own)||
              ' identified by' ||
                ' <password>';', 1);
      end if;
    end if;
    open pwd_cursor(lv_owner);
    fetch pwd_cursor into user_password;
    if pwd_cursor%NOTFOUND then
      user_password := '<password>';
      dbms_output.put_line(
        '*****> Warning: Username '||
          lv_owner||
            ' not found in DBA_USERS!!');
    end if;
    close pwd_cursor;
    a_lin := wri(a_lin, ' ', 1);
    a_lin := wri(a_lin,
      'rem -----', 1);
    a_lin := wri(a_lin, ' ', 1);
    alter_user := lower(lv_owner);
    if user_password = '<password>' then
      connect_pwd := user_password;
      a_lin := wri(a_lin, 'alter user '||
        alter_user||
          ' identified by <password>';',
        1);
    else
      connect_pwd := 'xyzyzy';
      a_lin := wri(a_lin, 'alter user '||
        alter_user||
          ' identified by xyzyzy;';', 1);
    end if;
    a_lin := wri(a_lin, 'connect '||alter_user||
      '/'||connect_pwd, 1);
    prev_own := lv_owner;
  end if;
  a_lin := wri(a_lin, 'GRANT ', 0);
  a_lin := wri(a_lin, lv_privilege, 0);
  prev_grantee := lv_grantee;
  prev_owner := lv_owner;
  prev_table_name := lv_table_name;
  prev_grantable := lv_grantable;
  privs := lv_privilege;
end if;
end loop;
close obj_cursor;

```

```

if prev_grantee != '@' then
    a_lin := wri(a_lin, ' ON', 0);
    a_lin := wri(a_lin, ' '||prev_owner||'.'||prev_table_name, 0);
    a_lin := wri(a_lin, ' TO', 0);
    a_lin := wri(a_lin, ' '||prev_grantee, 0);
    a_lin := wri(a_lin, prev_grantable, 1);
end if;
/* Reset password if we changed it */
if prev_own != '@' then
    a_lin := wri(a_lin, 'connect /', 1);
    if user_password != '<password>' then
        a_lin := wri(a_lin, 'alter user '||lower(prev_own)||
            ' identified by values '||&q||user_password||
            &q||';', 1);
    else
        a_lin := wri(a_lin, 'alter user '||lower(prev_own)||
            ' identified by <password>;', 1);
    end if;
end if;
-----
a_lin := wri(a_lin, 'rem *** Column Privileges ***', 1);
a_lin := '';
prev_grantee := '@';
prev_own := '@';
prev_owner := '';
prev_table_name := '';
prev_column_name := '';
prev_grantable := '';
privs := '';
open col_cursor;
loop
    fetch col_cursor into
        lv_grantee,
        lv_owner,
        lv_table_name,
        lv_column_name,
        lv_privilege,
        lv_grantable;
    exit when col_cursor%NOTFOUND;
    if prev_grantee = lv_grantee and prev_owner = lv_owner and
        prev_table_name = lv_table_name and
        prev_column_name = lv_column_name and
        prev_grantable = lv_grantable
    then
        /* Skip if duplicate privilege found */
        if instr(privs, lv_privilege) = 0 then
            a_lin := wri(a_lin, ' '||lv_privilege, 0);
            privs := privs || lv_privilege;
        end if;
    else
        if prev_grantee != '@' then
            a_lin := wri(a_lin, ' ON', 0);
            a_lin := wri(a_lin, ' '||prev_owner||'.'||
                prev_table_name, 0);
            a_lin := wri(a_lin, ' TO', 0);
            a_lin := wri(a_lin, ' '||prev_grantee, 0);
            a_lin := wri(a_lin, prev_grantable, 1);
        end if;
        /* */
        /* Since we can not grant privileges on objects we */
        /* do not own, or have been granted privilege to */
        /* with grant option, we need to build in the */
        /* connect strings to switch to the user's schema */
        /* who owns the object */
        /* */
        if lv_owner != prev_own then
            /* Reset password if we changed it */
            if prev_own != '@' then
                a_lin := wri(a_lin, 'connect /', 1);
                if user_password != '<password>' then
                    a_lin := wri(a_lin,
                        'alter user '||
                        lower(prev_own)||
                        ' identified by' ||
                        ' values '||&q||
                        user_password||&q||';',
                        1);
                end if;
            end if;
        end if;
    end if;
end loop;

```

```

else
    a_lin := wri(a_lin,
                'alter user '||
                lower(prev_own)||
                ' identified by' ||
                ' <password>';', 1);
end if;
end if;
open pwd_cursor(lv_owner);
fetch pwd_cursor into user_password;
if pwd_cursor%NOTFOUND then
    user_password := '<password>';
    dbms_output.put_line(
        '*****> Warning: Username '||
        lv_owner||
        ' not found in DBA_USERS!!');
end if;
close pwd_cursor;
a_lin := wri(a_lin, ' ', 1);
a_lin := wri(a_lin,
            'rem -----', 1);
a_lin := wri(a_lin, ' ', 1);
alter_user := lower(lv_owner);
if user_password = '<password>' then
    connect_pwd := user_password;
    a_lin := wri(a_lin, 'alter user '||
                alter_user||
                ' identified by <password>';',
                1);
else
    connect_pwd := 'xyzyz';
    a_lin := wri(a_lin, 'alter user '||
                alter_user||
                ' identified by xyzyz;';', 1);
end if;
a_lin := wri(a_lin, 'connect '||alter_user||
            '/'||connect_pwd, 1);
prev_own := lv_owner;
end if;
a_lin := wri(a_lin, 'GRANT ', 0);
a_lin := wri(a_lin, lv_privilege, 0);
a_lin := wri(a_lin, ' ('||lv_column_name||')', 0);
prev_grantee := lv_grantee;
prev_owner := lv_owner;
prev_table_name := lv_table_name;
prev_column_name := lv_column_name;
prev_grantable := lv_grantable;
privs := lv_privilege;
end if;
end loop;
close col_cursor;
if prev_grantee != '@' then
    a_lin := wri(a_lin, ' ON', 0);
    a_lin := wri(a_lin, ' '||prev_owner||'.'||prev_table_name, 0);
    a_lin := wri(a_lin, ' TO', 0);
    a_lin := wri(a_lin, ' '||prev_grantee, 0);
    a_lin := wri(a_lin, prev_grantable, 1);
end if;
/* Reset password if we changed it */
if prev_own != '@' then
    a_lin := wri(a_lin, 'connect /', 1);
    if user_password != '<password>' then
        a_lin := wri(a_lin, 'alter user '||lower(prev_own)||
                    ' identified by values '||&q||user_password||
                    &q||';', 1);
    else
        a_lin := wri(a_lin, 'alter user '||lower(prev_own)||
                    ' identified by <password>';', 1);
    end if;
end if;
end if;
-----
a_lin := wri(a_lin, 'rem *** System Privileges ***', 1);
a_lin := '';
a_lin := wri(a_lin, 'connect /', 1);
prev_grantee := '@';
prev_grantable := '';
open sys_cursor;

```

```

loop
    fetch sys_cursor into
        lv_grantee,
        lv_privilege,
        lv_grantable;
    exit when sys_cursor%NOTFOUND;
    if prev_grantee = lv_grantee and prev_grantable = lv_grantable
    then
        a_lin := wri(a_lin, ', ' || lv_privilege, 0);
    else
        if prev_grantee != '@' then
            a_lin := wri(a_lin, ' TO', 0);
            a_lin := wri(a_lin, ' ' || prev_grantee, 0);
            a_lin := wri(a_lin, prev_grantable, 1);
        end if;
        a_lin := wri(a_lin, 'GRANT ', 0);
        a_lin := wri(a_lin, lv_privilege, 0);
        prev_grantee := lv_grantee;
        prev_grantable := lv_grantable;
    end if;
end loop;
close sys_cursor;
if prev_grantee != '@' then
    a_lin := wri(a_lin, ' TO', 0);
    a_lin := wri(a_lin, ' ' || prev_grantee, 0);
    a_lin := wri(a_lin, prev_grantable, 1);
end if;
-----
a_lin := wri(a_lin, 'rem *** Role Privileges ***', 1);
a_lin := '';
prev_grantee := '@';
prev_grantable := '';
open role_cursor;
loop
    fetch role_cursor into
        lv_grantee,
        lv_granted_role,
        lv_grantable;
    exit when role_cursor%NOTFOUND;
    if prev_grantee = lv_grantee and prev_grantable = lv_grantable
    then
        a_lin := wri(a_lin, ', ' || lv_granted_role, 0);
    else
        if prev_grantee != '@' then
            a_lin := wri(a_lin, ' TO', 0);
            a_lin := wri(a_lin, ' ' || prev_grantee, 0);
            a_lin := wri(a_lin, prev_grantable, 1);
        end if;
        a_lin := wri(a_lin, 'GRANT ', 0);
        a_lin := wri(a_lin, lv_granted_role, 0);
        prev_grantee := lv_grantee;
        prev_grantable := lv_grantable;
    end if;
end loop;
close role_cursor;
if prev_grantee != '@' then
    a_lin := wri(a_lin, ' TO', 0);
    a_lin := wri(a_lin, ' ' || prev_grantee, 0);
    a_lin := wri(a_lin, prev_grantable, 1);
end if;
end;
/
set termout off
set heading off
spool cr_grant.sql
select 'rem cr_grant.sql' from dual;
select 'rem' from dual;
select 'rem ***** All grants for database ' || name from v$database;
select 'rem' from dual;
select 'set feedback off' from dual;
select text from g_temp order by lineno;
spool off
drop table g_temp;
set termout on
select 'Created cr_grant.sql...' from dual;
set termout off
exit

```

El siguiente es Crgrantspostsys.sql para sacar los grants de sistema y su contenido es el siguiente:

```

rem $DBA/crgrants.sql
rem
rem This script generates another script that will include all the GRANT
rem statements for those in the database (except for the SYS user).
rem
rem Note that this script will include the OBJECT privilege grants, as well
rem as the SYSTEM privilege grants in the single output file. You may wish
rem to edit the output if you need to separate the different types of grants.
rem
rem This script must be run by a user with the DBA role under Oracle7.
rem
rem Assumes that there is an OPS$ account for the user running this script.
rem
rem Created 10/30/97 by Brian Lomasky
rem
set verify off
set feedback off
set echo off
set pagesize 0
set serveroutput on size 100000
set termout on
define q=chr(39)
select 'Creating grant build script...' from dual;

create table g_temp ( lineno NUMBER,
                    text VARCHAR2(80));

declare
  cursor obj_cursor is select
    grantee,
    owner,
    table_name,
    privilege,
    decode(grantable,'YES',' WITH GRANT OPTION;',';')
  from sys.dba_tab_privs
  where grantor in ('SYS', 'SYSTEM')
  and grantee not in ('ADAMS', 'CTXSYS', 'DBSNMP', 'HR', 'LBACSYS',
'MDSYS', 'ODM', 'ODM_MTR',
'ORDPLUGINS', 'ORDSYS', 'OUTLN', 'PM', 'QS', 'QS_ADM', 'QS_CB',
'QS_CBADM', 'QS_CS', 'QS_ES',
'QS_OS', 'QS_WS', 'SCOTT', 'SH', 'SYS', 'SYSTEM', 'WKPROXY', 'WMSYS',
'WMSYS', 'XDB', 'ANONYMOUS',
'DIP',
'DMSYS', 'EXFSYS', 'MDDATA', 'MGMT_VIEW', 'OE', 'OLAPSYS', 'SI_INFORMTN_SCHEMA', 'PUBLIC',
'LOGSTDBY_ADMINISTRATOR',

'AQ_ADMINISTRATOR_ROLE', 'DBA', 'EXP_FULL_DATABASE', 'IMP_FULL_DATABASE', 'EXECUTE_CATALOG_R
OLE',

'CONNECT', 'RESOURCE', 'OEM_MONITOR', 'OLAP_DBA', 'OLAP_USER', 'WKUSER', 'SELECT_CATALOG_ROLE'
)
    order by 2, 3, 1, 4;

  cursor col_cursor is select
    grantee,
    owner,
    table_name,
    column_name,
    privilege,
    decode(grantable,'YES',' WITH GRANT OPTION;',';')
  from sys.dba_col_privs
  where grantor = 'PPRUEBA'
  order by 2, 3, 4, 5, 1;

  cursor sys_cursor is select
    grantee,
    privilege,
    decode(admin_option,'YES',' WITH ADMIN OPTION;',';')
  from sys.dba_sys_privs
  where grantee = 'PRUEBAO'
  order by 1, 2;

  cursor role_cursor is select
    grantee,

```

```

        granted_role,
        decode(admin_option, 'YES', ' WITH ADMIN OPTION;', ';')
    from sys.dba_role_privs
    where grantee = 'PRUEBAO'
    order by 1, 2;

cursor pwd_cursor (c_user varchar2) is
    select password from sys.dba_users where username = c_user;

lv_grantee          sys.dba_tab_privs.grantee%TYPE;
lv_owner            sys.dba_tab_privs.owner%TYPE;
lv_table_name       sys.dba_tab_privs.table_name%TYPE;
lv_column_name      sys.dba_col_privs.column_name%TYPE;
lv_privilege        sys.dba_tab_privs.privilege%TYPE;
lv_granted_role     sys.dba_role_privs.granted_role%TYPE;
lv_grantable        varchar2(19);
lv_string            varchar2(80);
lv_lineno           number;
a_lin               varchar2(80);
prev_grantee        sys.dba_tab_privs.grantee%TYPE;
prev_own            sys.dba_tab_privs.owner%TYPE;
alter_user          sys.dba_tab_privs.owner%TYPE;
prev_owner          sys.dba_tab_privs.owner%TYPE;
prev_table_name     sys.dba_tab_privs.table_name%TYPE;
prev_column_name    sys.dba_col_privs.column_name%TYPE;
prev_grantable      varchar2(19);
privs               varchar2(100);
user_password       sys.dba_users.password%TYPE;
connect_pwd         varchar2(10);

function wri(x_lin in varchar2, x_str in varchar2,
            x_force in number) return varchar2 is
begin
    if length(x_lin) + length(x_str) > 80
    then
        lv_lineno := lv_lineno + 1;
        insert into g_temp values (lv_lineno, x_lin);
        if x_force = 0
        then
            return '    ' || x_str;
        else
            lv_lineno := lv_lineno + 1;
            if substr(x_lin, 1, 2) = '  '
            then
                insert into g_temp values (
                    lv_lineno, x_str);
            else
                insert into g_temp values (
                    lv_lineno, '    ' || x_str);
            end if;
            return '';
        end if;
    else
        if x_force = 0
        then
            return x_lin || x_str;
        else
            lv_lineno := lv_lineno + 1;
            insert into g_temp values (
                lv_lineno, x_lin || x_str);
            return '';
        end if;
    end if;
end wri;

begin
    a_lin := '';
    lv_lineno := 0;
    prev_grantee := '@';
    prev_own := '@';
    prev_owner := '';
    prev_table_name := '';
    prev_grantable := '';
    privs := '';
    a_lin := wri(a_lin, 'rem *** Object Privileges ***', 1);
    a_lin := '';
    open obj_cursor;

```

```

loop
  fetch obj_cursor into
    lv_grantee,
    lv_owner,
    lv_table_name,
    lv_privilege,
    lv_grantable;
  exit when obj_cursor%NOTFOUND;
  if prev_grantee = lv_grantee and prev_owner = lv_owner and
    prev_table_name = lv_table_name and
    prev_grantable = lv_grantable
  then
    /* Skip if duplicate privilege found */
    if instr(privs, lv_privilege) = 0 then
      a_lin := wri(a_lin, ' ' || lv_privilege, 0);
      privs := privs || lv_privilege;
    end if;
  else
    if prev_grantee != '@' then
      a_lin := wri(a_lin, ' ON', 0);
      a_lin := wri(a_lin, ' ' || prev_owner || '.' ||
        prev_table_name, 0);
      a_lin := wri(a_lin, ' TO', 0);
      a_lin := wri(a_lin, ' ' || prev_grantee, 0);
      a_lin := wri(a_lin, prev_grantable, 1);
    end if;
    /* */
    /* Since we can not grant privileges on objects we */
    /* do not own, or have been granted privilege to */
    /* with grant option, we need to build in the */
    /* connect strings to switch to the user's schema */
    /* who owns the object */
    /* */
    if lv_owner != prev_own then
      /* Reset password if we changed it */
      if prev_own != '@' then
        a_lin := wri(a_lin, 'connect / as sysdba', 1);
        if user_password != '<password>' then
          a_lin := wri(a_lin,
            'alter user ' ||
            lower(prev_own) ||
            ' identified by' ||
            ' values ' || &q ||
            user_password || &q || ';',
            1);
        else
          a_lin := wri(a_lin,
            'alter user ' ||
            lower(prev_own) ||
            ' identified by' ||
            ' <password>', 1);
        end if;
      end if;
    end if;
    open pwd_cursor(lv_owner);
    fetch pwd_cursor into user_password;
    if pwd_cursor%NOTFOUND then
      user_password := '<password>';
      dbms_output.put_line(
        '*****> Warning: Username ' ||
        lv_owner ||
        ' not found in DBA_USERS!!!');
    end if;
    close pwd_cursor;
    a_lin := wri(a_lin, ' ', 1);
    a_lin := wri(a_lin,
      'rem -----', 1);
    a_lin := wri(a_lin, ' ', 1);
    alter_user := lower(lv_owner);
    if user_password = '<password>' then
      connect_pwd := user_password;
      a_lin := wri(a_lin, 'alter user ' ||
        alter_user ||
        ' identified by <password>',
        1);
    else
      connect_pwd := 'xyzyy';
      a_lin := wri(a_lin, 'alter user ' ||

```



```

                alter_user||
                ' identified by xyzzy;', 1);
            end if;
            a_lin := wri(a_lin, 'connect '||alter_user||
                '/ as sysdba' || connect_pwd, 1);
            prev_own := lv_owner;
        end if;
        a_lin := wri(a_lin, 'GRANT ', 0);
        a_lin := wri(a_lin, lv_privilege, 0);
        prev_grantee := lv_grantee;
        prev_owner := lv_owner;
        prev_table_name := lv_table_name;
        prev_grantable := lv_grantable;
        privs := lv_privilege;
    end if;
end loop;
close obj_cursor;
if prev_grantee != '@' then
    a_lin := wri(a_lin, ' ON', 0);
    a_lin := wri(a_lin, ' ' || prev_owner || '.' || prev_table_name, 0);
    a_lin := wri(a_lin, ' TO', 0);
    a_lin := wri(a_lin, ' ' || prev_grantee, 0);
    a_lin := wri(a_lin, prev_grantable, 1);
end if;
/* Reset password if we changed it */
if prev_own != '@' then
    a_lin := wri(a_lin, 'connect / as sysdba', 1);
    if user_password != '<password>' then
        a_lin := wri(a_lin, 'alter user ' || lower(prev_own) ||
            ' identified by values ' || &q|| user_password ||
            &q||';', 1);
    else
        a_lin := wri(a_lin, 'alter user ' || lower(prev_own) ||
            ' identified by <password>', 1);
    end if;
end if;
-----
a_lin := wri(a_lin, 'rem *** Column Privileges ***', 1);
a_lin := '';
prev_grantee := '@';
prev_own := '@';
prev_owner := '';
prev_table_name := '';
prev_column_name := '';
prev_grantable := '';
privs := '';
open col_cursor;
loop
    fetch col_cursor into
        lv_grantee,
        lv_owner,
        lv_table_name,
        lv_column_name,
        lv_privilege,
        lv_grantable;
    exit when col_cursor%NOTFOUND;
    if prev_grantee = lv_grantee and prev_owner = lv_owner and
        prev_table_name = lv_table_name and
        prev_column_name = lv_column_name and
        prev_grantable = lv_grantable
    then
        /* Skip if duplicate privilege found */
        if instr(privs, lv_privilege) = 0 then
            a_lin := wri(a_lin, ', ' || lv_privilege, 0);
            privs := privs || lv_privilege;
        end if;
    else
        if prev_grantee != '@' then
            a_lin := wri(a_lin, ' ON', 0);
            a_lin := wri(a_lin, ' ' || prev_owner || '.' ||
                prev_table_name, 0);
            a_lin := wri(a_lin, ' TO', 0);
            a_lin := wri(a_lin, ' ' || prev_grantee, 0);
            a_lin := wri(a_lin, prev_grantable, 1);
        end if;
        /* */
        /* Since we can not grant privileges on objects we */

```

```

/* do not own, or have been granted privilege to */
/* with grant option, we need to build in the */
/* connect strings to switch to the user's schema */
/* who owns the object */
/* */
if lv_owner != prev_own then
  /* Reset password if we changed it */
  if prev_own != '@' then
    a_lin := wri(a_lin, 'connect / as sysdba', 1);
    if user_password != '<password>' then
      a_lin := wri(a_lin,
        'alter user '||
        lower(prev_own)||
        ' identified by' ||
        ' values '||&q||
        user_password||&q||';',
        1);
    else
      a_lin := wri(a_lin,
        'alter user '||
        lower(prev_own)||
        ' identified by' ||
        ' <password>', 1);
    end if;
  end if;
  open pwd_cursor(lv_owner);
  fetch pwd_cursor into user_password;
  if pwd_cursor%NOTFOUND then
    user_password := '<password>';
    dbms_output.put_line(
      '*****> Warning: Username '||
      lv_owner||
      ' not found in DBA_USERS!!!');
  end if;
  close pwd_cursor;
  a_lin := wri(a_lin, ' ', 1);
  a_lin := wri(a_lin,
    'rem -----', 1);
  a_lin := wri(a_lin, ' ', 1);
  alter_user := lower(lv_owner);
  if user_password = '<password>' then
    connect_pwd := user_password;
    a_lin := wri(a_lin, 'alter user '||
      alter_user||
      ' identified by <password>',
      1);
  else
    connect_pwd := 'xyzyz';
    a_lin := wri(a_lin, 'alter user '||
      alter_user||
      ' identified by xyzyz;', 1);
  end if;
  a_lin := wri(a_lin, 'connect '||alter_user||
    '/'||connect_pwd, 1);
  prev_own := lv_owner;
end if;
a_lin := wri(a_lin, 'GRANT ', 0);
a_lin := wri(a_lin, lv_privilege, 0);
a_lin := wri(a_lin, '('||lv_column_name||')', 0);
prev_grantee := lv_grantee;
prev_owner := lv_owner;
prev_table_name := lv_table_name;
prev_column_name := lv_column_name;
prev_grantable := lv_grantable;
privs := lv_privilege;
end if;
end loop;
close col_cursor;
if prev_grantee != '@' then
  a_lin := wri(a_lin, ' ON', 0);
  a_lin := wri(a_lin, ' '||prev_owner||'.'||prev_table_name, 0);
  a_lin := wri(a_lin, ' TO', 0);
  a_lin := wri(a_lin, ' '||prev_grantee, 0);
  a_lin := wri(a_lin, prev_grantable, 1);
end if;
/* Reset password if we changed it */
if prev_own != '@' then

```

```

a_lin := wri(a_lin, 'connect / as sysdba', 1);
if user_password != '<password>' then
    a_lin := wri(a_lin, 'alter user '||lower(prev_own)||
        ' identified by values '||&q||user_password||
        &q||';', 1);
else
    a_lin := wri(a_lin, 'alter user '||lower(prev_own)||
        ' identified by <password>;', 1);
end if;
end if;
-----
a_lin := wri(a_lin, 'rem *** System Privileges ***', 1);
a_lin := '';
a_lin := wri(a_lin, 'connect / as sysdba', 1);
prev_grantee := '@';
prev_grantable := '';
open sys_cursor;
loop
    fetch sys_cursor into
        lv_grantee,
        lv_privilege,
        lv_grantable;
exit when sys_cursor%NOTFOUND;
if prev_grantee = lv_grantee and prev_grantable = lv_grantable
then
    a_lin := wri(a_lin, ', '||lv_privilege, 0);
else
    if prev_grantee != '@' then
        a_lin := wri(a_lin, ' TO', 0);
        a_lin := wri(a_lin, ' '||prev_grantee, 0);
        a_lin := wri(a_lin, prev_grantable, 1);
    end if;
    a_lin := wri(a_lin, 'GRANT ', 0);
    a_lin := wri(a_lin, lv_privilege, 0);
    prev_grantee := lv_grantee;
    prev_grantable := lv_grantable;
end if;
end loop;
close sys_cursor;
if prev_grantee != '@' then
    a_lin := wri(a_lin, ' TO', 0);
    a_lin := wri(a_lin, ' '||prev_grantee, 0);
    a_lin := wri(a_lin, prev_grantable, 1);
end if;
-----
a_lin := wri(a_lin, 'rem *** Role Privileges ***', 1);
a_lin := '';
prev_grantee := '@';
prev_grantable := '';
open role_cursor;
loop
    fetch role_cursor into
        lv_grantee,
        lv_granted_role,
        lv_grantable;
exit when role_cursor%NOTFOUND;
if prev_grantee = lv_grantee and prev_grantable = lv_grantable
then
    a_lin := wri(a_lin, ', '||lv_granted_role, 0);
else
    if prev_grantee != '@' then
        a_lin := wri(a_lin, ' TO', 0);
        a_lin := wri(a_lin, ' '||prev_grantee, 0);
        a_lin := wri(a_lin, prev_grantable, 1);
    end if;
    a_lin := wri(a_lin, 'GRANT ', 0);
    a_lin := wri(a_lin, lv_granted_role, 0);
    prev_grantee := lv_grantee;
    prev_grantable := lv_grantable;
end if;
end loop;
close role_cursor;
if prev_grantee != '@' then
    a_lin := wri(a_lin, ' TO', 0);
    a_lin := wri(a_lin, ' '||prev_grantee, 0);
    a_lin := wri(a_lin, prev_grantable, 1);
end if;

```

```

end;
/

set termout off
set heading off
spool cr_grantpost.sql
select 'rem cr_grantpost.sql' from dual;
select 'rem' from dual;
select 'rem ***** All grants for database ' || name from v$database;
select 'rem' from dual;
select 'set feedback off' from dual;
select text from g_temp order by lineno;
spool off
drop table g_temp;
set termout on
select 'Created cr_grantpost.sql...' from dual;
set termout off
exit

```

El siguiente script a ejecutar es Crlinks.sql el cual nos permite sacar todos los db_link de la base, su contenido se muestra a continuación.

```

rem $DBA/crlinks.sql
rem
rem This script generates another script that will include all the CREATE
rem DATABASE LINK statements for those in the database.
rem
rem This script must be run by a user with the DBA role under Oracle7.
rem
rem (The SYS.LINK$ table is used, since the password column is not visible
rem in the DBA_DB_LINKS view).
rem
rem Last Change 05/05/97 by Brian Lomasky
rem
set verify off
set feedback off
set echo off
set pagesize 0
set serveroutput on size 100000
set termout on
define q=chr(39)
select 'Creating database link build script...' from dual;

create table l_temp ( lineno NUMBER,
                    text VARCHAR2(80));

declare
  cursor link_cursor is select
    u.name,
    l.name,
    l.userid,
    l.password,
    l.host
  from sys.link$ l, sys.user$ u
  where l.owner# = u.user#
  order by u.name, l.name;

  cursor pwd_cursor (c_user varchar2) is
    select password from sys.dba_users where username = c_user;

alter_user      sys.user$.name%TYPE;
prev_owner     sys.user$.name%TYPE;
lv_owner       sys.user$.name%TYPE;
lv_db_link     sys.link$.name%TYPE;
lv_username    sys.link$.userid%TYPE;
lv_password    sys.link$.password%TYPE;
lv_host       sys.link$.host%TYPE;
user_password  sys.dba_users.password%TYPE;
connect_pwd    varchar2(10);
a_lin         varchar2(80);
lv_lineno      number;

function wri(x_lin in varchar2, x_str in varchar2,
            x_force in number) return varchar2 is

```

```

begin
  if length(x_lin) + length(x_str) > 80
  then
    lv_lineno := lv_lineno + 1;
    insert into l_temp values (lv_lineno, x_lin);
    if x_force = 0
    then
      return '    ' || x_str;
    else
      lv_lineno := lv_lineno + 1;
      if substr(x_lin,1,2) = '  '
      then
        insert into l_temp values (
          lv_lineno, x_str);
      else
        insert into l_temp values (
          lv_lineno, '    ' || x_str);
      end if;
      return '';
    end if;
  else
    if x_force = 0
    then
      return x_lin || x_str;
    else
      lv_lineno := lv_lineno + 1;
      insert into l_temp values (
        lv_lineno, x_lin || x_str);
      return '';
    end if;
  end if;
end wri;

begin
  lv_lineno := 0;
  a_lin := '';
  prev_owner := '@';
  open link_cursor;
  loop
    fetch link_cursor into
      lv_owner,
      lv_db_link,
      lv_username,
      lv_password,
      lv_host;
  exit when link_cursor%NOTFOUND;
  /* */
  /* Since we can not create a database link in another user's */
  /* schema, we need to build in the connect strings to switch */
  /* to the user's schema who owns the private database link */
  /* */
  if lv_owner != prev_owner then
    /* Reset password if we changed it */
    if prev_owner != '@' then
      a_lin := wri(a_lin, 'connect /', 1);
      if user_password != '<password>' then
        a_lin := wri(a_lin, 'alter user ' ||
          lower(prev_owner) ||
          ' identified by values ' || &q ||
          user_password || &q || ';', 1);
      else
        a_lin := wri(a_lin, 'alter user ' ||
          lower(prev_owner) ||
          ' identified by <password>;',
          1);
      end if;
    end if;
  end if;
  if lv_owner = 'PUBLIC' then
    open pwd_cursor('SYSTEM');
    fetch pwd_cursor into user_password;
    if pwd_cursor%NOTFOUND then
      user_password := '<password>';
      dbms_output.put_line(
        '*****> Warning: Username ' ||
        'SYSTEM' ||
        ' not found in DBA_USERS!!!');
    end if;
  end if;
end loop;

```

```

else
    open pwd_cursor(lv_owner);
    fetch pwd_cursor into user_password;
    if pwd_cursor%NOTFOUND then
        user_password := '<password>';
        dbms_output.put_line(
            '*****> Warning: Username '||
            lv_owner||
            ' not found in DBA_USERS!!!');
    end if;
end if;
close pwd_cursor;
a_lin := wri(a_lin, ' ', 1);
a_lin := wri(a_lin, 'rem -----',
    1);
a_lin := wri(a_lin, ' ', 1);
if lv_owner = 'PUBLIC' then
    alter_user := 'system';
else
    alter_user := lower(lv_owner);
end if;
if user_password = '<password>' then
    connect_pwd := user_password;
    a_lin := wri(a_lin, 'alter user '||alter_user||
        ' identified by <password>;', 1);
else
    connect_pwd := 'xyzyz';
    a_lin := wri(a_lin, 'alter user '||alter_user||
        ' identified by xyzyz;', 1);
end if;
a_lin := wri(a_lin, 'connect '||alter_user||'/'||
    connect_pwd, 1);
if lv_owner = 'PUBLIC' then
    prev_owner := 'system';
else
    prev_owner := lv_owner;
end if;
end if;
if lv_owner = 'PUBLIC' then
    a_lin := wri(a_lin, 'CREATE PUBLIC DATABASE LINK '||
        lv_db_link, 1);
else
    a_lin := wri(a_lin, 'CREATE DATABASE LINK '||
        lv_db_link, 1);
end if;
a_lin := wri(a_lin, ' ', 0);
if lv_username != ' ' then
    a_lin := wri(a_lin, ' connect to '||lv_username
        ||' identified by '||lv_password, 0);
end if;
if lv_host != ' ' then
    a_lin := wri(a_lin, ' using '||&q||lv_host||&q, 0);
end if;
a_lin := wri(a_lin, ';', 1);
end loop;
close link_cursor;
/* Reset password if we changed it */
if prev_owner != '@' then
    a_lin := wri(a_lin, 'connect /', 1);
    if user_password != '<password>' then
        a_lin := wri(a_lin, 'alter user '||lower(prev_owner)||
            ' identified by values '||&q||user_password||
            &q||';', 1);
    else
        a_lin := wri(a_lin, 'alter user '||lower(prev_owner)||
            ' identified by <password>;', 1);
    end if;
end if;
a_lin := wri(a_lin, 'exit', 1);
end;
/

set termout off
set heading off
spool cr_link.sql
select 'rem cr_link.sql' from dual;
select 'rem' from dual;

```

```

select 'rem ***** All database links for database ' || name from v$database;
select 'rem' from dual;
select 'set feedback off' from dual;
select text from l_temp order by lineno;
spool off
drop table l_temp;
set termout on
select 'Created cr_link.sql...' from dual;
set termout off
exit

```

Por último ejecutar el script llamado Crsyns.sql el cual extraerá todos los sinónimos existentes en la base de datos su contenido se muestra a continuación.

```

rem $DBA/crsyns.sql
rem
rem This script generates another script that will include all the CREATE
rem SYNONYM statements for those in the database.
rem
rem This script must be run by a user with the DBA role under Oracle7.
rem
rem Created 10/21/96 by Brian Lomasky
rem
set verify off
set feedback off
set echo off
set pagesize 0
set termout on
select 'Creating database synonym build script...' from dual;

create table syn_temp (
    lineno NUMBER,
    text VARCHAR2(80));

declare
    cursor syn_cursor is select
        owner,
        synonym_name,
        table_owner,
        table_name,
        db_link
        from sys.dba_synonyms
        where owner='PUBLIC'
        and table_owner not in ('ADAMS', 'CTXSYS', 'DBSNMP', 'HR', 'LBACSYS',
'MDSYS', 'ODM', 'ODM_MTR',
'ORDPLUGINS', 'ORDSYS', 'OUTLN', 'PM', 'QS', 'QS_ADM', 'QS_CB',
'QS_CBADM', 'QS_CS', 'QS_ES',
'QS_OS', 'QS_WS', 'SCOTT', 'SH', 'SYS', 'SYSTEM', 'WKPROXY', 'WKSYS',
'WMSYS', 'XDB', 'ANONYMOUS',
'DIP',
'DMSYS', 'EXFSYS', 'MDDATA', 'MGMT_VIEW', 'OE', 'OLAPSYS', 'SI_INFORMTN_SCHEMA',
'AQ_ADMINISTRATOR_ROLE', 'DBA', 'EXP_FULL_DATABASE', 'IMP_FULL_DATABASE', 'EXECUTE_CATALOG_ROLE',
'CONNECT', 'RESOURCE', 'OEM_MONITOR', 'OLAP_DBA', 'OLAP_USER', 'WKUSER', 'SELECT_CATALOG_ROLE')
        order by owner, synonym_name;

    lv_owner          sys.dba_synonyms.owner%TYPE;
    lv_synonym_name   sys.dba_synonyms.synonym_name%TYPE;
    lv_table_owner    sys.dba_synonyms.table_owner%TYPE;
    lv_table_name     sys.dba_synonyms.table_name%TYPE;
    lv_db_link        sys.dba_synonyms.db_link%TYPE;
    lv_lineno         number;
    a_lin             varchar2(80);

function wri(x_lin in varchar2, x_str in varchar2,
    x_force in number) return varchar2 is
begin
    if length(x_lin) + length(x_str) > 80
    then
        lv_lineno := lv_lineno + 1;
        insert into syn_temp values (lv_lineno, x_lin);
    end if;
end;

```

```

        if x_force = 0
        then
            return '    '||x_str;
        else
            lv_lineno := lv_lineno + 1;
            if substr(x_lin,1,2) = '  '
            then
                insert into syn_temp values (
                    lv_lineno, x_str);
            else
                insert into syn_temp values (
                    lv_lineno, '    '||x_str);
            end if;
            return '';
        end if;
    else
        if x_force = 0
        then
            return x_lin||x_str;
        else
            lv_lineno := lv_lineno + 1;
            insert into syn_temp values (
                lv_lineno, x_lin||x_str);
            return '';
        end if;
    end if;
end wri;

begin
a_lin := '';
lv_lineno := 0;
open syn_cursor;
loop
    fetch syn_cursor into
        lv_owner,
        lv_synonym_name,
        lv_table_owner,
        lv_table_name,
        lv_db_link;
    exit when syn_cursor%NOTFOUND;
    if lv_owner = 'PUBLIC' then
        a_lin := wri(a_lin, 'CREATE PUBLIC SYNONYM ', 0);
    else
        a_lin := wri(a_lin, 'CREATE SYNONYM '||
            lv_owner || '.', 0);
    end if;
    a_lin := wri(a_lin, lv_synonym_name, 0);
    a_lin := wri(a_lin, ' for ', 0);
    if lv_db_link != ' ' then
        a_lin := wri(a_lin, lv_table_owner || '.' ||
            lv_table_name || '@' || lv_db_link, 0);
    else
        a_lin := wri(a_lin, lv_table_owner || '.' ||
            lv_table_name, 0);
    end if;
    a_lin := wri(a_lin, ';', 1);
end loop;
close syn_cursor;
a_lin := wri(a_lin, 'exit', 1);
end;
/
set termout off
set heading off
spool cr_syn.sql
select 'rem cr_syn.sql' from dual;
select 'rem' from dual;
select 'rem ***** All synonyms for database ' || name from v$database;
select 'rem' from dual;
select 'set feedback off' from dual;
select text from syn_temp order by lineno;
spool off
drop table syn_temp;
set termout on
select 'Created cr_syn.sql...' from dual;
set termout off
exit

```


CREAR UN RESPALDO DE LA INSTANCIA

Hacer el respaldo mediante un export de la base de datos en el origen.

Script para hacer el import

```
#!/bin/bash

ORACLE_HOME=/oracle9i/product/Ora9i
PATH=$ORACLE_HOME/bin:$PATH
ORACLE_SID=LOBORH
NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P15
NLS_LANGUAGE=AMERICAN_AMERICA.WE8ISO8859P9

export ORACLE_HOME PATH ORACLE_SID NLS_LANG NLS_LANGUAGE

cd /sia/profiles/admin/orapp/respaldos
/oracle9i/product/Ora9i/bin/exp system/oracle buffer=600000 file=
/bkloborhP060710.dmp log=/loborh.log OWNER=SQLUSU2001,OMWB_EMULATION
```

Si dicho respaldo no existiera se saca uno mediante el siguiente comando:

-En caso de ser completo:

```
$ exp login/password@servicio full=y file=NomArchivo.dmp
log=NomLog.log
```

-Si es por esquema.

```
$ exp usuario/password@servicio owner=usuario file=usuario.dmp
log=usuario.log
```

CREACIÓN DE OBJETOS EN LA INSTANCIA DESTINO

En la base destino crear los objetos DDL con los scripts que se obtienen en la ejecución de scripts en la base origen, así que declaramos las variables ORACLE_HOME, ORACLE_SID y EL PATH para la instancia LOBORH y ejecutar los scripts, esto es en el orden siguiente:

No.	Informacion	Script
1	Tablespace	crea_ts_”instancia”.sql
2	Perfiles	cr_prof.sql
3	Roles	cr_role.sql
4	Usuarios	cr_user.sql
5	Grants de sistema	cr_grantpost.sql
6	Grants de objetos	cr_grant.sql

REALIZACIÓN DEL IMPORT EN EL SERVIDOR DESTINO

Script para hacer el import

```
#!/bin/bash

ORACLE_HOME=/u01/app/orapp/product/11.2.0/dbhome_1
PATH=$ORACLE_HOME/bin:$PATH
ORACLE_SID=LOBORH
NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P15
NLS_LANGUAGE=AMERICAN_AMERICA.WE8ISO8859P9

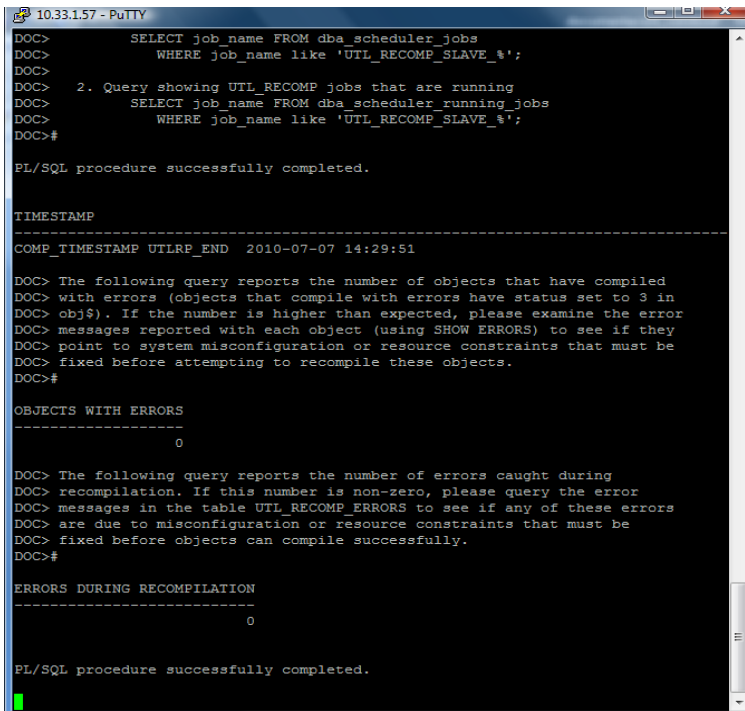
export ORACLE_HOME PATH ORACLE_SID NLS_LANG NLS_LANGUAGE

cd /export/home/orapp/SCRIPTS/LOBO_dmp
imp system/oracle buffer=600000
file=/export/home/orapp/SCRIPTS/LOBO_dmp/bkloborhP060710.dmp
log=/export/home/orapp/SCRIPTS/LOBO_dmp/loborh.log
fromuser=SQLUSU2001,OMWB_EMULATION touser=SQLUSU2001,OMWB_EMULATION
```

Después de terminar de realizar los import se ejecuta el script cr_syn.sql para la generación de los sinónimos.

Por último recompilar los objetos invalidos ejecutando el script utlrp.sql ubicado en

\$ORACLE_HOME/rdbms/admin



```
10.33.157 - PuTTY
DOC> SELECT job_name FROM dba_scheduler_jobs
DOC> WHERE job_name like 'UTL_RECOMP_SLAVE_%';
DOC>
DOC> 2. Query showing UTL_RECOMP jobs that are running
DOC> SELECT job_name FROM dba_scheduler_running_jobs
DOC> WHERE job_name like 'UTL_RECOMP_SLAVE_%';
DOC>#
PL/SQL procedure successfully completed.

TIMESTAMP
-----
COMP_TIMESTAMP UTLRP_END 2010-07-07 14:29:51

DOC> The following query reports the number of objects that have compiled
DOC> with errors (objects that compile with errors have status set to 3 in
DOC> obj$). If the number is higher than expected, please examine the error
DOC> messages reported with each object (using SHOW ERRORS) to see if they
DOC> point to system misconfiguration or resource constraints that must be
DOC> fixed before attempting to recompile these objects.
DOC>#

OBJECTS WITH ERRORS
-----
0

DOC> The following query reports the number of errors caught during
DOC> recompilation. If this number is non-zero, please query the error
DOC> messages in the table UTL_RECOMP_ERRORS to see if any of these errors
DOC> are due to misconfiguration or resource constraints that must be
DOC> fixed before objects can compile successfully.
DOC>#

ERRORS DURING RECOMPILATION
-----
0

PL/SQL procedure successfully completed.
```

Por último se verifican los objetos inválidos para cada usuario tanto en la base origen como en la base destino, esto es mediante el siguiente query:

```
SQL> select object_name, object_type, status from dba_objects
where owner='ESQUEMA' and status='INVALID';
```

2.6 MIGRACIÓN DE LA BASE DE DATOS LOBOSYS

OBTENCIÓN DE DATOS EN LA BASE ORIGEN.

Para realizar esta tarea es necesario ejecutar los scripts en la base origen de acuerdo a la siguiente tabla.

Tabla de ejecución de scripts

No.	Información	Script
1	tablespaces	> busca_ts.sh > crea_ts2.sh
2	perfiles	> Crprofs.sql
3	roles	> Crroles.sql
4	usuarios	> Crusers.sql
5	grants de objetos	> Crgrants.sql
6	grants de sistema	> Crgrantspostsys.sql
7	links	> Crlinks.sql
8	sinónimos	> Crsyns.sql

CREAR UN RESPALDO DE LA INSTANCIA

Hacer el export de los esquemas a migrar en la base LOBOSYS origen con el siguiente script

```
export NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P9
export NLS_LANGUAGE=AMERICAN_AMERICA.WE8ISO8859P9
ORACLE_HOME=/oracle9i/product/Ora9i
export ORACLE_HOME
ORACLE_BASE=/oracle9i/product/Ora9i
export ORACLE_BASE
ORACLE_SID=LOBOSYS
export ORACLE_SID

DATE=`date '+bklobosysP%d%m%y.dmp'`
DATELOG=`date '+bklobosysP%d%m%y.log'`

echo
cd /sia/profiles/admin/orapp/respaldos
/oracle9i/product/Ora9i/bin/./exp system/lobosys@LOBOSYS
OWNER=SQLSU2001,OMWB_EMULATION
FILE=/sia/profiles/admin/orapp/respaldos/resp_migra/lobosys.dmp
LOG=/sia/profiles/admin/orapp/respaldos/resp_migra/lobosys.log
STATISTICS=NONE
```

CREACIÓN DE OBJETOS EN LA INSTANCIA DESTINO

En la base destino crear los objetos DDL con los scripts que se obtienen en la ejecución de scripts en la base origen, así que declaramos las variables ORACLE_HOME, ORACLE_SID y EL PATH para la instancia LOBOSYS y ejecutar los scripts, esto es en el orden siguiente:

No.	Informacion	Script
1	Tablespace	crea_ts_”instancia”.sql
2	Perfiles	cr_prof.sql
3	Roles	cr_role.sql
4	Usuarios	cr_user.sql
5	Grants de sistema	cr_grantpost.sql
6	Grants de objetos	cr_grant.sql

REALIZACIÓN DEL IMPORT EN EL SERVIDOR DESTINO PARA LA INSTANCIA LOBOSYS

Script para hacer el import:

```
#!/bin/bash

ORACLE_HOME=/u01/app/orapp/product/11.2.0/dbhome_1
PATH=$ORACLE_HOME/bin:$PATH
ORACLE_SID=LOBOSYS
NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P15
NLS_LANGUAGE=AMERICAN_AMERICA.WE8ISO8859P9

export ORACLE_HOME PATH ORACLE_SID NLS_LANG NLS_LANGUAGE

cd /export/home/orapp/SCRIPTS/LOBO_dmp
imp system/oracle buffer=600000
file=/export/home/orapp/SCRIPTS/LOBO_dmp/lobosys.dmp
log=/export/home/orapp/SCRIPTS/LOBO_dmp/lodosys.log
fromuser=SQLUSU2001,OMWB_EMULATION touser=SQLUSU2001,OMWB_EMULATION
```

Después de terminar de realizar el import se ejecuta el script cr_syn.sql para la generación de los sinónimos.

Un punto importante antes de terminar es realizar la recompilación de los objetos inválidos ejecutando el script utlrp.sql ubicado en \$ORACLE_HOME/rdbms/admin

```

10.33.157 - PuTTY
DOC> SELECT job_name FROM dba_scheduler_jobs
DOC> WHERE job_name like 'UTL_RECOMP_SLAVE_%';
DOC>
DOC> 2. Query showing UTL_RECOMP jobs that are running
DOC> SELECT job_name FROM dba_scheduler_running_jobs
DOC> WHERE job_name like 'UTL_RECOMP_SLAVE_%';
DOC>#

PL/SQL procedure successfully completed.

TIMESTAMP
-----
COMP_TIMESTAMP UTLRP_END 2010-07-07 14:29:51

DOC> The following query reports the number of objects that have compiled
DOC> with errors (objects that compile with errors have status set to 3 in
DOC> obj$). If the number is higher than expected, please examine the error
DOC> messages reported with each object (using SHOW ERRORS) to see if they
DOC> point to system misconfiguration or resource constraints that must be
DOC> fixed before attempting to recompile these objects.
DOC>#

OBJECTS WITH ERRORS
-----
0

DOC> The following query reports the number of errors caught during
DOC> recompilation. If this number is non-zero, please query the error
DOC> messages in the table UTL_RECOMP_ERRORS to see if any of these errors
DOC> are due to misconfiguration or resource constraints that must be
DOC> fixed before objects can compile successfully.
DOC>#

ERRORS DURING RECOMPILATION
-----
0

PL/SQL procedure successfully completed.

```

Por último se verifican los objetos inválidos para cada usuario tanto en la base origen como en la base destino, esto es mediante el siguiente query:

```
SQL> select object_name, object_type, status from dba_objects
where owner='ESQUEMA' and status='INVALID';
```

Una vez concluido la recompilación y la revisión de los objetos inválidos dentro de la nueva base, se hace la entrega de la misma al cliente para que realice las pruebas necesarias como lo son: la conexión de sus aplicaciones, ejecución de consultas y procesos para validar que no presente ningún error, una vez finalizadas dichas pruebas se da por concluido el proceso de migración de dicha instancia.

2.7 MIGRACIÓN DE LA BASE DE DATOS LOBORH08

OBTENCIÓN DE DATOS EN LA BASE ORIGEN.

Para obtenerlos es necesario ejecutar los scripts en la base origen de acuerdo a la siguiente tabla.

Tabla de ejecución de scripts

No.	Información	Script
1	tablespaces	> busca_ts.sh > crea_ts2.sh
2	perfiles	> Crprofs.sql
3	roles	> Crroles.sql
4	usuarios	> Crusers.sql
5	grants de objetos	> Crgrants.sql
6	grants de sistema	> Crgrantspostsys.sql
7	links	> Crlinks.sql
8	sinónimos	> Crsyns.sql

CREAR UN RESPALDO DE LA INSTANCIA

Hacer el export de los esquemas a migrar en la base LOBORH08 origen con el siguiente script:

```
# variables de ambiente de oracle

export NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P9
export NLS_LANGUAGE=AMERICAN_AMERICA.WE8ISO8859P9
ORACLE_HOME=/oracle9i/product/Ora9i
export ORACLE_HOME
ORACLE_BASE=/oracle9i/product/Ora9i
export ORACLE_BASE
ORACLE_SID=LOBORH08
export ORACLE_SID

/oracle9i/product/Ora9i/bin/./exp system/orapp
OWNER=SQLUSU2001,OMWB_EMULATION
FILE=/oradata/SCRIPTS/LOBORH08/loborh08.dmp
LOG=/oradata/SCRIPTS/LOBORH08/loborh08.log STATISTICS=NONE
```

CREACIÓN DE OBJETOS EN LA INSTANCIA DESTINO

En la base destino crear los objetos DDL con los scripts que se obtienen en la ejecución de scripts en la base origen, así que declaramos las variables ORACLE_HOME, ORACLE_SID y EL PATH para la instancia LOBORH08 y ejecutar los scripts, esto es en el orden siguiente:

No.	Informacion	Script
1	Tablespace	crea_ts_”instancia”.sql
2	Perfiles	cr_prof.sql
3	Roles	cr_role.sql
4	Usuarios	cr_user.sql
5	Grants de sistema	cr_grantpost.sql
6	Grants de objetos	cr_grant.sql

REALIZACIÓN DEL IMPORT EN EL SERVIDOR DESTINO

Script para hacer el import para el schema OMWB_EMULATION:

```
#!/bin/bash
ORACLE_HOME=/u01/app/orapp/product/11.2.0/dbhome_1
PATH=$ORACLE_HOME/bin:$PATH
ORACLE_SID=LOBORH08
NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P15
NLS_LANGUAGE=AMERICAN_AMERICA.WE8ISO8859P9
export ORACLE_HOME PATH ORACLE_SID NLS_LANG NLS_LANGUAGE
cd /export/home/orapp/SCRIPTS/LOBO_dmp
imp system/oracle buffer=600000
file=/export/home/orapp/SCRIPTS/LOBO_dmp/loborh08.dmp
log=/export/home/orapp/SCRIPTS/LOBO_dmp/loborh08.log
fromuser=OMWB_EMULATION touser=OMWB_EMULATION
```

Script para hacer el import para el schema SQLUSU2001:

```
#!/bin/bash
ORACLE_HOME=/u01/app/orapp/product/11.2.0/dbhome_1
PATH=$ORACLE_HOME/bin:$PATH
ORACLE_SID=LOBORH08
NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P15
NLS_LANGUAGE=AMERICAN_AMERICA.WE8ISO8859P9
export ORACLE_HOME PATH ORACLE_SID NLS_LANG NLS_LANGUAGE
cd /export/home/orapp/SCRIPTS/LOBO_dmp
imp system/oracle buffer=600000
file=/export/home/orapp/SCRIPTS/LOBO_dmp/loborh08.dmp
log=/export/home/orapp/SCRIPTS/LOBO_dmp/loborh08.log
fromuser=SQLUSU2001 touser=SQLUSU2001
```

Despues de terminar de realizar los import se ejecuta el script cr_syn.sql para la generaci3n de los sin3nimos cr_syn.sql

Por ultimo recompilar los objetos invalidos ejecutando el script utlrf.sql ubicado en \$ORACLE_HOME/rdbms/admin

```

10.33.157 - PuTTY
DOC>      SELECT job_name FROM dba_scheduler_jobs
DOC>      WHERE job_name like 'UTL_RECOMP_SLAVE_%';
DOC>
DOC>  2. Query showing UTL_RECOMP jobs that are running
DOC>      SELECT job_name FROM dba_scheduler_running_jobs
DOC>      WHERE job_name like 'UTL_RECOMP_SLAVE_%';
DOC>#

PL/SQL procedure successfully completed.

TIMESTAMP
-----
COMP_TIMESTAMP UTLRF_END  2010-07-07 14:29:51

DOC> The following query reports the number of objects that have compiled
DOC> with errors (objects that compile with errors have status set to 3 in
DOC> obj$). If the number is higher than expected, please examine the error
DOC> messages reported with each object (using SHOW ERRORS) to see if they
DOC> point to system misconfiguration or resource constraints that must be
DOC> fixed before attempting to recompile these objects.
DOC>#

OBJECTS WITH ERRORS
-----
0

DOC> The following query reports the number of errors caught during
DOC> recompilation. If this number is non-zero, please query the error
DOC> messages in the table UTL_RECOMP_ERRORS to see if any of these errors
DOC> are due to misconfiguration or resource constraints that must be
DOC> fixed before objects can compile successfully.
DOC>#

ERRORS DURING RECOMPILATION
-----
0

PL/SQL procedure successfully completed.

```

Para terminar se verifican los objetos inválidos para cada usuario tanto en la base origen como en la base destino, esto es mediante el siguiente query:

```
SQL> select object_name, object_type, status from dba_objects
where owner='ESQUEMA' and status='INVALID';
```

Para validar que la migraci3n de la base no presenta errores se hace entrega de la misma, para que el cliente realice diferentes pruebas a esta, una vez terminadas las mismas y el cliente no reporta ning3n problema se da por terminado el proceso de migraci3n de esta base de datos y queda validada para realizar la migraci3n a producci3n.

2.8 MIGRACIÓN DE LA BASE DE DATOS LOBORH09

OBTENCIÓN DE DATOS EN LA BASE ORIGEN.

Para realizar dicha tarea es necesario ejecutar los scripts en la base de datos origen de acuerdo a la siguiente tabla.

Tabla de ejecución de scripts

No.	Información	Script
1	tablespaces	> busca_ts.sh > crea_ts2.sh
2	perfiles	> Crprofs.sql
3	roles	> Crroles.sql
4	usuarios	> Crusers.sql
5	grants de objetos	> Crgrants.sql
6	grants de sistema	> Crgrantspostsys.sql
7	links	> Crlinks.sql
8	sinónimos	> Crsyns.sql

CREAR UN RESPALDO DE LA INSTANCIA

Hacer el Export de los esquemas a migrar en la base LOBORH09 origen con el siguiente script:

```
# variables de ambiente de oracle
export NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P9
export NLS_LANGUAGE=AMERICAN_AMERICA.WE8ISO8859P9
ORACLE_HOME=/oracle9i/product/Ora9i
export ORACLE_HOME
ORACLE_BASE=/oracle9i/product/Ora9i
export ORACLE_BASE
ORACLE_SID=loborh09
export ORACLE_SID

/oracle9i/product/Ora9i/bin/./exp system/orapp
OWNER=SQLUSU2001,OMWB_EMULATION
FILE=/oradata/SCRIPTS/LOBORH09/loborh09.dmp
LOG=/oradata/SCRIPTS/LOBORH09/loborh09.log STATISTICS=NONE
```

CREACIÓN DE OBJETOS EN LA INSTANCIA DESTINO

En la base destino crear los objetos DDL con los scripts que se obtienen en la ejecución de scripts en la base origen, así que declaramos las variables ORACLE_HOME, ORACLE_SID y EL PATH para la instancia LOBORH09 y ejecutar los scripts, esto es en el orden siguiente:

No.	Informacion	Script
1	Tablespace	crea_ts_”instancia”.sql
2	Perfiles	cr_prof.sql
3	Roles	cr_role.sql
4	Usuarios	cr_user.sql
5	Grants de sistema	cr_grantpost.sql
6	Grants de objetos	cr_grant.sql

REALIZACIÓN DEL IMPORT EN EL SERVIDOR DESTINO

Script para hacer el import para el schema SQLUSU2001:

```
#!/bin/bash
ORACLE_HOME=/u01/app/orapp/product/11.2.0/dbhome_1
PATH=$ORACLE_HOME/bin:$PATH
ORACLE_SID=loborh09
NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P15
NLS_LANGUAGE=AMERICAN_AMERICA.WE8ISO8859P9
export ORACLE_HOME PATH ORACLE_SID NLS_LANG NLS_LANGUAGE
cd /export/home/orapp/SCRIPTS/LOBO_dmp

imp system/oracle buffer=600000
file=/export/home/orapp/SCRIPTS/LOBO_dmp/loborh09.dmp
log=/export/home/orapp/SCRIPTS/LOBO_dmp/loborh09.log
fromuser=SQLUSU2001 touser=SQLUSU2001
```

Para el schema OMWB_EMULATION:

```
#!/bin/bash
ORACLE_HOME=/u01/app/orapp/product/11.2.0/dbhome_1
PATH=$ORACLE_HOME/bin:$PATH
ORACLE_SID=loborh09
NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P15
NLS_LANGUAGE=AMERICAN_AMERICA.WE8ISO8859P9

export ORACLE_HOME PATH ORACLE_SID NLS_LANG NLS_LANGUAGE
cd /export/home/orapp/SCRIPTS/LOBO_dmp
imp system/oracle buffer=600000
file=/export/home/orapp/SCRIPTS/LOBO_dmp/loborh09.dmp
log=/export/home/orapp/SCRIPTS/LOBO_dmp/loborh09.log
fromuser=OMWB_EMULATION touser=OMWB_EMULATION
```

Después de terminar de realizar los import se ejecuta el script cr_syn.sql para la generación de los sinónimos.

Por último recompilar los objetos inválidos ejecutando el script utlrp.sql ubicado en \$ORACLE_HOME/rdbms/admin

```

10.33.157 - PuTTY
DOC>      SELECT job_name FROM dba_scheduler_jobs
DOC>      WHERE job_name like 'UTL_RECOMP_SLAVE_%';
DOC>
DOC>  2. Query showing UTL_RECOMP jobs that are running
DOC>      SELECT job_name FROM dba_scheduler_running_jobs
DOC>      WHERE job_name like 'UTL_RECOMP_SLAVE_%';
DOC>#

PL/SQL procedure successfully completed.

TIMESTAMP
-----
COMP_TIMESTAMP UTLRP_END  2010-07-07 14:29:51

DOC> The following query reports the number of objects that have compiled
DOC> with errors (objects that compile with errors have status set to 3 in
DOC> obj$). If the number is higher than expected, please examine the error
DOC> messages reported with each object (using SHOW ERRORS) to see if they
DOC> point to system misconfiguration or resource constraints that must be
DOC> fixed before attempting to recompile these objects.
DOC>#

OBJECTS WITH ERRORS
-----
0

DOC> The following query reports the number of errors caught during
DOC> recompilation. If this number is non-zero, please query the error
DOC> messages in the table UTL_RECOMP_ERRORS to see if any of these errors
DOC> are due to misconfiguration or resource constraints that must be
DOC> fixed before objects can compile successfully.
DOC>#

ERRORS DURING RECOMPILATION
-----
0

PL/SQL procedure successfully completed.

```

Para dar fin a la migración de esta base se verifican los objetos inválidos para cada usuario tanto en la base origen como en la base destino, esto es mediante el siguiente query:

```
SQL> select object_name, object_type, status from dba_objects
where owner='ESQUEMA' and status='INVALID';
```

El número de objetos inválidos debe ser nulo o debe ser el mismo que en la base de datos origen, además de que el cliente tiene que validar que no presente ningún problema con la nueva base de datos, si no reportan ningún problema se da por terminado el proceso de migración.

2.9 MIGRACIÓN DE LA BASE DE DATOS SIA

RECOPIACIÓN DE INFORMACION EN LA BASE DE DATOS ORIGEN

Para recopilar toda la información de objetos creados en la base de datos es necesario ejecutar los siguientes scripts de acuerdo a la siguiente tabla.

Tabla de ejecución de scripts

No.	Información	Script
1	tablespaces	> busca_ts.sh > crea_ts2.sh
2	perfiles	> Crprofs.sql
3	roles	> Crroles.sql
4	usuarios	> Crusers.sql
5	grants de objetos	> Crgrants.sql
6	grants de sistema	> Crgrantspostsys.sql
7	links	> Crlinks.sql
8	sinónimos	> Crsyns.sql

-Se procede a hacer el export de los schemas que se migrarán de la base de datos, el export es el siguiente:

```
export NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P9
export NLS_LANGUAGE=AMERICAN_AMERICA.WE8ISO8859P9
ORACLE_HOME=/oracle9i/product/Ora9i
export ORACLE_HOME
ORACLE_BASE=/oracle9i/product/Ora9i
export ORACLE_BASE
ORACLE_SID=sia
export ORACLE_SID

/oracle9i/product/Ora9i/bin/./exp system/orapp
OWNER=CONSULTA_LOBO,REGISTRO,SIA,SICOFIP,SIA_RECHUM,SIA_RECFIN,SIA_RMSG,
TARIFICA,CONDUSEF,GESTION,SIAFF_MARTAER,INVENTARIOS
FILE=/oradata/SCRIPTS/SIA/sia.dmp LOG=/ oradata/SCRIPTS/SIA/sia..log
STATISTICS=NONE
```

Los schemas a migrar son:

CONSULTA_LOBO, REGISTRO, SIA, SICOFIP, SIA_RECHUM, SIA_RECFIN, SIA_RMSG, TARIFICA, CONDUSEF, GESTION, SIAFF_MASTER, INVENTARIOS.

Ahora se crearán los objetos DDL en la base destino para poder importar los datos de los schemas.

No.	Informacion	Script
1	Tablespace	<i>crea_ts_”instancia”.sql</i>
2	Perfiles	<i>cr_prof.sql</i>
3	Roles	<i>cr_role.sql</i>
4	Usuarios	<i>cr_user.sql</i>
5	Grants de sistema	<i>cr_grantpost.sql</i>
6	Grants de objetos	<i>cr_grant.sql</i>

-Ya que se tiene lo anterior se proceder a hacer el import de los schemas en la base destino, el import para cada esquema se presenta a continuación:

-SIA

```
#!/bin/bash
ORACLE_HOME=/u01/app/orapp/product/11.2.0/dbhome_1
PATH=$ORACLE_HOME/bin:$PATH
ORACLE_SID=sia
NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P15
NLS_LANGUAGE=AMERICAN_AMERICA.WE8ISO8859P9
export ORACLE_HOME PATH ORACLE_SID NLS_LANG NLS_LANGUAGE

cd /migracion/SIA

imp system/oracle buffer=600000 file=sia.dmp log=sia.log fromuser=SIA
touser=SIA
```

-INVENTARIOS

```
#!/bin/bash
ORACLE_HOME=/u01/app/orapp/product/11.2.0/dbhome_1
PATH=$ORACLE_HOME/bin:$PATH
ORACLE_SID=sia
NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P15
NLS_LANGUAGE=AMERICAN_AMERICA.WE8ISO8859P9
export ORACLE_HOME PATH ORACLE_SID NLS_LANG NLS_LANGUAGE

cd /migracion/SIA

imp system/oracle buffer=600000 file=sia.dmp log=sia.log
fromuser=INVENTARIOS touser=INVENTARIOS
```

-SIA_RECFIN

```
#!/bin/bash
ORACLE_HOME=/u01/app/orapp/product/11.2.0/dbhome_1
PATH=$ORACLE_HOME/bin:$PATH
ORACLE_SID=sia
NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P15
NLS_LANGUAGE=AMERICAN_AMERICA.WE8ISO8859P9
export ORACLE_HOME PATH ORACLE_SID NLS_LANG NLS_LANGUAGE

cd /migracion/SIA

imp system/oracle buffer=600000 file=sia.dmp log=sia.log
fromuser=SIA_RECFIN touser=SIA_RECFIN
```

-SIA_RECHUM

```
#!/bin/bash
ORACLE_HOME=/u01/app/orapp/product/11.2.0/dbhome_1
PATH=$ORACLE_HOME/bin:$PATH
ORACLE_SID=sia
NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P15
NLS_LANGUAGE=AMERICAN_AMERICA.WE8ISO8859P9
export ORACLE_HOME PATH ORACLE_SID NLS_LANG NLS_LANGUAGE

cd /migracion/SIA

imp system/oracle buffer=600000 file=sia.dmp log=sia.log
fromuser=SIA_RECHUM touser=SIA_RECHUM
```

-REGISTRO

```
#!/bin/bash
ORACLE_HOME=/u01/app/orapp/product/11.2.0/dbhome_1
PATH=$ORACLE_HOME/bin:$PATH
ORACLE_SID=sia
NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P15
NLS_LANGUAGE=AMERICAN_AMERICA.WE8ISO8859P9
export ORACLE_HOME PATH ORACLE_SID NLS_LANG NLS_LANGUAGE

cd /migracion/SIA

imp system/oracle buffer=600000 file=sia.dmp log=sia.log
fromuser=REGISTRO touser=REGISTRO
```

-SIA_RMSG

```
#!/bin/bash
ORACLE_HOME=/u01/app/orapp/product/11.2.0/dbhome_1
PATH=$ORACLE_HOME/bin:$PATH
ORACLE_SID=sia
NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P15
NLS_LANGUAGE=AMERICAN_AMERICA.WE8ISO8859P9
export ORACLE_HOME PATH ORACLE_SID NLS_LANG NLS_LANGUAGE

cd /migracion/SIA

imp system/oracle buffer=600000 file=sia.dmp log=sia.log
fromuser=SIA_RMSG touser=SIA_RMSG
```

-GESTION

```
#!/bin/bash
ORACLE_HOME=/u01/app/orapp/product/11.2.0/dbhome_1
PATH=$ORACLE_HOME/bin:$PATH
ORACLE_SID=sia
NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P15
NLS_LANGUAGE=AMERICAN_AMERICA.WE8ISO8859P9
export ORACLE_HOME PATH ORACLE_SID NLS_LANG NLS_LANGUAGE

cd /migracion/SIA

imp system/oracle buffer=600000 file=sia.dmp log=sia.log
fromuser=GESTION touser=GESTION
```

-CONSULTA_LOBO

```
#!/bin/bash
ORACLE_HOME=/u01/app/orapp/product/11.2.0/dbhome_1
PATH=$ORACLE_HOME/bin:$PATH
ORACLE_SID=sia
NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P15
NLS_LANGUAGE=AMERICAN_AMERICA.WE8ISO8859P9
export ORACLE_HOME PATH ORACLE_SID NLS_LANG NLS_LANGUAGE

cd /migracion/SIA

imp system/oracle buffer=600000 file=sia.dmp log=sia.log
fromuser=CONSULTA_LOBO touser=CONSULTA_LOBO
```

-TARIFICA

```
#!/bin/bash
ORACLE_HOME=/u01/app/orapp/product/11.2.0/dbhome_1
PATH=$ORACLE_HOME/bin:$PATH
ORACLE_SID=sia
NLS_LANG=AMERICAN AMERICA.WE8ISO8859P15
NLS_LANGUAGE=AMERICAN AMERICA.WE8ISO8859P9
export ORACLE_HOME PATH ORACLE_SID NLS_LANG NLS_LANGUAGE
```

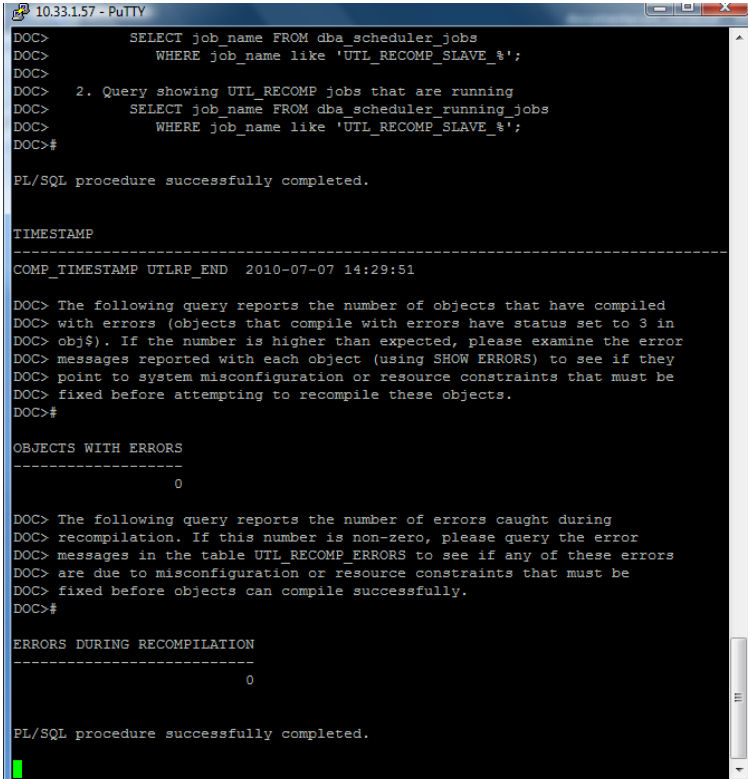
```
cd /migracion/SIA
```

```
imp system/oracle buffer=600000 file=sia.dmp log=sia.log
fromuser=TARIFICA touser=TARIFICA
```

Despues de importar todos los schemas, se ejecuta el script para crear los sinónimos que es cr_syn.sql.

Se recompilan los objetos ejecutando utlrp.sql de la siguiente manera.

```
SQL> @$ORACLE_HOME/rdbms/admin/utlrp.sql
```



```
10.33.157 - PuTTY
DOC> SELECT job_name FROM dba_scheduler_jobs
DOC> WHERE job_name like 'UTL_RECOMP_SLAVE_%';
DOC>
DOC> 2. Query showing UTL_RECOMP jobs that are running
DOC> SELECT job_name FROM dba_scheduler_running_jobs
DOC> WHERE job_name like 'UTL_RECOMP_SLAVE_%';
DOC>#

PL/SQL procedure successfully completed.

TIMESTAMP
-----
COMP_TIMESTAMP UTLRP_END 2010-07-07 14:29:51

DOC> The following query reports the number of objects that have compiled
DOC> with errors (objects that compile with errors have status set to 3 in
DOC> obj$). If the number is higher than expected, please examine the error
DOC> messages reported with each object (using SHOW ERRORS) to see if they
DOC> point to system misconfiguration or resource constraints that must be
DOC> fixed before attempting to recompile these objects.
DOC>#

OBJECTS WITH ERRORS
-----
0

DOC> The following query reports the number of errors caught during
DOC> recompilation. If this number is non-zero, please query the error
DOC> messages in the table UTL_RECOMP_ERRORS to see if any of these errors
DOC> are due to misconfiguration or resource constraints that must be
DOC> fixed before objects can compile successfully.
DOC>#

ERRORS DURING RECOMPILATION
-----
0

PL/SQL procedure successfully completed.
```

Por último se verifican los objetos invalidos para cada usuario tanto en la base origen como en la base destino, esto es mediante el siguiente query:

```
SQL> select object_name, object_type, status from dba_objects
where owner='ESQUEMA' and status='INVALID';
```


2.10 MIGRACIÓN DE LA BASE DE DATOS SIAADM

RECOPIACIÓN DE INFORMACION EN LA BASE DE DATOS ORIGEN

Para recopilar toda la información de objetos DDL creados en la base de datos es necesario ejecutar los siguientes scripts en el orden de la tabla.

Tabla de ejecución de scripts

No.	Información	Script
1	tablespaces	> busca_ts.sh > crea_ts2.sh
2	perfiles	> Crprofs.sql
3	roles	> Crroles.sql
4	usuarios	> Crusers.sql
5	grants de objetos	> Crgrants.sql
6	grants de sistema	> Crgrantspostsys.sql
7	links	> Crlinks.sql
8	sinónimos	> Crsyns.sql

Se procede a hacer el export de los schemas que se migrarán de la base de datos, el export es el siguiente:

```
export NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P9
export NLS_LANGUAGE=AMERICAN_AMERICA.WE8ISO8859P9
ORACLE_HOME=/oracle9i/product/Ora9i
export ORACLE_HOME
ORACLE_BASE=/oracle9i/product/Ora9i
export ORACLE_BASE
ORACLE_SID=siaAdm
export ORACLE_SID

cd /oradata/SCRIPTS/siaAdm
/oracle9i/product/Ora9i/bin/./exp system/orapp
OWNER=SIA,SIA_RMSG,SIA_RECFIN,SIA_RECHUM FILE=siaAdm.dmp
LOG=siaAdm.log STATISTICS=NONE
```

Los schemas a migrar son:

SIA, SIA_RMSG, SIA_RECFIN, SIA_RECHUM.

Ahora se crearán los objetos en la base destino para poder importar los datos de los schemas mediante la ejecución de los siguientes scripts.

No.	Informacion	Script
1	Tablespace	<i>crea_ts_”instancia”.sql</i>
2	Perfiles	<i>cr_prof.sql</i>
3	Roles	<i>cr_role.sql</i>
4	Usuarios	<i>cr_user.sql</i>
5	Grants de sistema	<i>cr_grantpost.sql</i>
6	Grants de objetos	<i>cr_grant.sql</i>

Ya que se tiene lo anterior se proceder a hacer el import de los schemas en la base destino, el import para cada esquema se presenta a continuación :

-SIA

```
#!/bin/bash
ORACLE_HOME=/u01/app/orapp/product/11.2.0/dbhome_1
PATH=$ORACLE_HOME/bin:$PATH
ORACLE_SID=siaADM
NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P15
NLS_LANGUAGE=AMERICAN_AMERICA.WE8ISO8859P9
export ORACLE_HOME PATH ORACLE_SID NLS_LANG NLS_LANGUAGE

cd /migracion/siaAdm

imp system/oracle buffer=600000 file=siaAdm.dmp log=siaAdm.log
fromuser=SIA touser=SIA
```

-SIA_RMSG

```
#!/bin/bash
ORACLE_HOME=/u01/app/orapp/product/11.2.0/dbhome_1
PATH=$ORACLE_HOME/bin:$PATH
ORACLE_SID=siaADM
NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P15
NLS_LANGUAGE=AMERICAN_AMERICA.WE8ISO8859P9
export ORACLE_HOME PATH ORACLE_SID NLS_LANG NLS_LANGUAGE

cd /migracion/siaAdm

imp system/oracle buffer=600000 file=siaAdm.dmp log=siaAdm.log
fromuser=SIA_RMSG touser=SIA_RMSF
```

-SIA_RECFIN

```
#!/bin/bash
ORACLE_HOME=/u01/app/orapp/product/11.2.0/dbhome_1
PATH=$ORACLE_HOME/bin:$PATH
ORACLE_SID=siaADM
NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P15
NLS_LANGUAGE=AMERICAN_AMERICA.WE8ISO8859P9
export ORACLE_HOME PATH ORACLE_SID NLS_LANG NLS_LANGUAGE

cd /migracion/siaAdm

imp system/oracle buffer=600000 file=siaAdm.dmp log=siaAdm.log
fromuser=SIA_RECFIN touser=SIA_RECFIN
```

-SIA_RECHUM

```
#!/bin/bash
ORACLE_HOME=/u01/app/orapp/product/11.2.0/dbhome_1
PATH=$ORACLE_HOME/bin:$PATH
ORACLE_SID=siaADM
NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P15
NLS_LANGUAGE=AMERICAN_AMERICA.WE8ISO8859P9
export ORACLE_HOME PATH ORACLE_SID NLS_LANG NLS_LANGUAGE

cd /migracion/siaAdm

imp system/oracle buffer=600000 file=siaAdm.dmp log=siaAdm.log
fromuser=SIA_RECHUM touser=SIA_RECHUM
```

Despues de importar todos los squemas, se ejecita el script para crear los sinónimos que es cr_syn.sql.

-Se recompilan los objetos ejecutando utlrp.sql de la siguiente manera.

```
SQL> @$ORACLE_HOME/rdbms/admin/utlrp.sql
```

```
10.33.157 - PuTTY
DOC>      SELECT job_name FROM dba_scheduler_jobs
DOC>      WHERE job_name like 'UTL_RECOMP_SLAVE_%';
DOC>
DOC> 2. Query showing UTL_RECOMP jobs that are running
DOC>      SELECT job_name FROM dba_scheduler_running_jobs
DOC>      WHERE job_name like 'UTL_RECOMP_SLAVE_%';
DOC>#
PL/SQL procedure successfully completed.

TIMESTAMP
-----
COMP_TIMESTAMP UTLRP_END 2010-07-07 14:29:51

DOC> The following query reports the number of objects that have compiled
DOC> with errors (objects that compile with errors have status set to 3 in
DOC> obj$). If the number is higher than expected, please examine the error
DOC> messages reported with each object (using SHOW ERRORS) to see if they
DOC> point to system misconfiguration or resource constraints that must be
DOC> fixed before attempting to recompile these objects.
DOC>#

OBJECTS WITH ERRORS
-----
0

DOC> The following query reports the number of errors caught during
DOC> recompilation. If this number is non-zero, please query the error
DOC> messages in the table UTL_RECOMP ERRORS to see if any of these errors
DOC> are due to misconfiguration or resource constraints that must be
DOC> fixed before objects can compile successfully.
DOC>#

ERRORS DURING RECOMPILATION
-----
0

PL/SQL procedure successfully completed.
```

Se verifican los objetos invalidos para cada usuario tanto en la base origen como en la base destino, esto es mediante el siguiente query:

```
SQL> select object_name, object_type, status from dba_objects  
where owner='ESQUEMA' and status='INVALID';
```

Con esto se da por terminado el proceso de migración de la base de datos al nuevo servidor y de versión, el cliente debe validar que no presente ningun problema con la misma antes de poder realizar la migracion de manera productiva.

2.11 MIGRACIÓN DE LA BASE DE DATOS SISADM

NOTA: EN LA SIGUIENTE INSTANCIA SERÁN MIGRADAS TRES BASES DE DATOS PARA FORMAR UNA SOLA, QUE SON CONBAN, ADQUI Y SRCPROD

RECOPIACIÓN DE INFORMACION EN LAS BASES DE DATOS ORIGEN

Para recopilar toda la información de objetos creados en las bases de datos es necesario ejecutar los siguientes scripts, y se hace en el orden en el que se especifica:

Tabla de ejecución de scripts

No.	Información	Script
1	tablespaces	> busca_ts.sh > crea_ts2.sh
2	perfiles	> Crprofs.sql
3	roles	> Crroles.sql
4	usuarios	> Crusers.sql
5	grants de objetos	> Crgrants.sql
6	grants de sistema	> Crgrantspostsys.sql
7	links	> Crlinks.sql
8	sinónimos	> Crsyms.sql

Se procede a hacer el export de los esquemas que se migrarán de la base de datos, el export es el siguiente:

EXPORT DE LA BASE CONBAN

```
export NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P9
export NLS_LANGUAGE=AMERICAN_AMERICA.WE8ISO8859P9
ORACLE_HOME=/oracle9i/product/Ora9i
export ORACLE_HOME
ORACLE_BASE=/oracle9i/product/Ora9i
export ORACLE_BASE
ORACLE_SID=conban
export ORACLE_SID

cd /oradata/SCRIPTS/conban
/oracle9i/product/Ora9i/bin/./exp system/orapp OWNER=CONBAN,SIIE
FILE=conban.dmp LOG=conban.log STATISTICS=NONE
```

EXPORT DE LA BASE ADQUI

```

export NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P9
export NLS_LANGUAGE=AMERICAN_AMERICA.WE8ISO8859P9
ORACLE_HOME=/oracle9i/product/Ora9i
export ORACLE_HOME
ORACLE_BASE=/oracle9i/product/Ora9i
export ORACLE_BASE
ORACLE_SID=ADQUI
export ORACLE_SID

cd /oradata/SCRIPTS/conban
/oracle9i/product/Ora9i/bin/./exp system/orapp OWNER=DGBS,SALA
FILE=ADQUI.dmp LOG=ADQUI.log STATISTICS=NONE

```

EXPORT DE LA BASE SRCPROD

```

export NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P9
export NLS_LANGUAGE=AMERICAN_AMERICA.WE8ISO8859P9
ORACLE_HOME=/oracle9i/product/Ora9i
export ORACLE_HOME
ORACLE_BASE=/oracle9i/product/Ora9i
export ORACLE_BASE
ORACLE_SID=srcprod
export ORACLE_SID

cd /oradata/SCRIPTS/conban
/oracle9i/product/Ora9i/bin/./exp system/orapp OWNER=SRC
FILE=srcprod.dmp LOG=srcprod.log STATISTICS=NONE

```

Los esquemas a migrar son:

CONBAN, SIIE, DGBS, SALA, SRC.

Ahora se crearan los objetos en la base destino para poder importar los datos de los esquemas mediante la ejecución de los siguientes scripts.

No.	Informacion	Script
1	Tablespace	<i>crea_ts_”instancia”.sql</i>
2	Perfiles	<i>cr_prof.sql</i>
3	Roles	<i>cr_role.sql</i>
4	Usuarios	<i>cr_user.sql</i>
5	Grants de sistema	<i>cr_grantpost.sql</i>
6	Grants de objetos	<i>cr_grant.sql</i>

Ya que se tiene lo anterior se procede a hacer el import de los esquemas en la base destino, el import para cada esquema se presenta a continuación:

-CONBAN

```
#!/bin/bash
ORACLE_HOME=/u01/app/orapp/product/11.2.0/dbhome_1
PATH=$ORACLE_HOME/bin:$PATH
ORACLE_SID=sisadm
NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P15
NLS_LANGUAGE=AMERICAN_AMERICA.WE8ISO8859P9
export ORACLE_HOME PATH ORACLE_SID NLS_LANG NLS_LANGUAGE

cd /migracion/conban

imp system/oracle buffer=600000 file=conban.dmp log=conban.log
fromuser=CONBAN touser=CONBAN
```

-SIIE

```
#!/bin/bash
ORACLE_HOME=/u01/app/orapp/product/11.2.0/dbhome_1
PATH=$ORACLE_HOME/bin:$PATH
ORACLE_SID=sisadm
NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P15
NLS_LANGUAGE=AMERICAN_AMERICA.WE8ISO8859P9
export ORACLE_HOME PATH ORACLE_SID NLS_LANG NLS_LANGUAGE

cd /migracion/conban

imp system/oracle buffer=600000 file=conban.dmp log=conban.log
fromuser=SIIE touser=SIIE
```

-SALA

```
#!/bin/bash
ORACLE_HOME=/u01/app/orapp/product/11.2.0/dbhome_1
PATH=$ORACLE_HOME/bin:$PATH
ORACLE_SID=sisadm
NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P15
NLS_LANGUAGE=AMERICAN_AMERICA.WE8ISO8859P9
export ORACLE_HOME PATH ORACLE_SID NLS_LANG NLS_LANGUAGE

cd /migracion/conban

imp system/oracle buffer=600000 file=ADQUI.dmp log=ADQUI.log
fromuser=SALA touser=SALA
```

-DGBS

```
#!/bin/bash
ORACLE_HOME=/u01/app/orapp/product/11.2.0/dbhome_1
PATH=$ORACLE_HOME/bin:$PATH
ORACLE_SID=sisadm
NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P15
NLS_LANGUAGE=AMERICAN_AMERICA.WE8ISO8859P9
export ORACLE_HOME PATH ORACLE_SID NLS_LANG NLS_LANGUAGE

cd /migracion/conban

imp system/oracle buffer=600000 file=ADQUI.dmp log=ADQUI.log
fromuser=DGBS touser=DGBS
```

-SRC

```
#!/bin/bash
ORACLE_HOME=/u01/app/orapp/product/11.2.0/dbhome_1
PATH=$ORACLE_HOME/bin:$PATH
ORACLE_SID=sisadm
NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P15
NLS_LANGUAGE=AMERICAN_AMERICA.WE8ISO8859P9
export ORACLE_HOME PATH ORACLE_SID NLS_LANG NLS_LANGUAGE

cd /migracion/conban

imp system/oracle buffer=600000 file=srcprod.dmp log=srcprod.log
fromuser=SRC touser=SRC
```

Despues de importar todos los squemas, se ejecita el script para crear los sinónimos que es cr_syn.sql.

Se recompilan los objetos ejecutando utlrp.sql de la siguiente manera.

```
SQL> @$ORACLE_HOME/rdbms/admin/utlrp.sql
```

```
10.33.157 - PuTTY
DOC>      SELECT job_name FROM dba_scheduler_jobs
DOC>      WHERE job_name like 'UTL_RECOMP_SLAVE_%';
DOC>
DOC>  2. Query showing UTL_RECOMP jobs that are running
DOC>      SELECT job_name FROM dba_scheduler_running_jobs
DOC>      WHERE job_name like 'UTL_RECOMP_SLAVE_%';
DOC>#
PL/SQL procedure successfully completed.

TIMESTAMP
-----
COMP_TIMESTAMP UTLRP_END  2010-07-07 14:29:51

DOC> The following query reports the number of objects that have compiled
DOC> with errors (objects that compile with errors have status set to 3 in
DOC> obj$). If the number is higher than expected, please examine the error
DOC> messages reported with each object (using SHOW ERRORS) to see if they
DOC> point to system misconfiguration or resource constraints that must be
DOC> fixed before attempting to recompile these objects.
DOC>#

OBJECTS WITH ERRORS
-----
0

DOC> The following query reports the number of errors caught during
DOC> recompilation. If this number is non-zero, please query the error
DOC> messages in the table UTL_RECOMP ERRORS to see if any of these errors
DOC> are due to misconfiguration or resource constraints that must be
DOC> fixed before objects can compile successfully.
DOC>#

ERRORS DURING RECOMPILATION
-----
0

PL/SQL procedure successfully completed.
```


Por último se verifican los objetos invalidos para cada usuario tanto en la base origen como en la base destino, esto es mediante el siguiente query:

```
SQL> select object_name, object_type, status from dba_objects  
where owner='ESQUEMA' and status='INVALID';
```

Al igual que en anteriores migraciones se debe validar que no presente ningun error la base de datos en la nueva versión, esta tarea se debe validar con los propietarios de las tres bases involucradas ya que estas quedaron en una sola instancia.

2.12 MIGRACIÓN DE LA BASE DE DATOS SIAM

OBTENCIÓN DE DATOS EN LA BASE ORIGEN.

Para realizar dicha tarea es necesario ejecutar los scripts en la base origen de acuerdo a la siguiente tabla.

Tabla de ejecución de scripts

No.	Información	Script
1	tablespaces	> busca_ts.sh > crea_ts2.sh
2	perfiles	> Crprofs.sql
3	roles	> Crroles.sql
4	usuarios	> Crusers.sql
5	grants de objetos	> Crgrants.sql
6	grants de sistema	> Crgrantspostsys.sql
7	links	> Crlinks.sql
8	sinónimos	> Crsyns.sql

CREAR UN RESPALDO DE LA INSTANCIA

Hacer el export de los schemas a migrar en la base siam origen con el siguiente script:

```
ORACLE_HOME=/oracle10g/product/Ora10g
export ORACLE_HOME
ORACLE_BASE=/oracle10g/product/Ora10g
export ORACLE_BASE
ORACLE_SID=siam
export ORACLE_SID
NLS_LANGUAGE=AMERICAN_AMERICA.WE8ISO8859P15
export NLS_LANGUAGE
NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P15
export NLS_LANG

cd /home/orapp/migracion/SCRIPTS/siam
/oracle10g/product/Ora10g/bin/./exp system/s1st3m OWNER=SIAM
FILE=/home/orapp/migracion/SCRIPTS/siam/siam-2.dmp
LOG=/home/orapp/migracion/SCRIPTS/siam/siam-2.log STATISTICS=NONE
```

CREACIÓN DE OBJETOS EN LA INSTANCIA DESTINO

En la base destino crear los objetos DDL con los scripts que se obtienen en la ejecución de scripts en la base origen, así que declaramos las variables ORACLE_HOME, ORACLE_SID y EL PATH para la instancia SIAM y ejecutar los scripts, esto es en el orden siguiente:

No.	Informacion	Script
1	Tablespace	crea_ts_”instancia”.sql
2	Perfiles	cr_prof.sql
3	Roles	cr_role.sql
4	Usuarios	cr_user.sql
5	Grants de sistema	cr_grantpost.sql
6	Grants de objetos	cr_grant.sql

Hacer la carga del schema en la nueva base de datos.

```
ORACLE_HOME=/u01/app/orapp/product/11.2.0/dbhome_1
PATH=$ORACLE_HOME/bin:$PATH
ORACLE_SID=siam
NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P15
NLS_LANGUAGE=AMERICAN_AMERICA.WE8ISO8859P9
export ORACLE_HOME PATH ORACLE_SID NLS_LANG NLS_LANGUAGE
```

```
cd /home/orapp/migracion/SCRIPTS/siam
./exp system/oracle FILE=/home/orapp/migracion/SCRIPTS/siam/siam-2.dmp
LOG=/home/orapp/migracion/SCRIPTS/siam/siam-2.log FROMUSER=SIAM
TOUSER=SIAM STATISTICS=NONE
```

Recompilar los objetos inválidos ejecutando el script utlrp.sql ubicado en \$ORACLE_HOME/rdbms/admin

```
10.33.157 - PuTTY
DOC> SELECT job name FROM dba_scheduler_jobs
DOC> WHERE job_name like 'UTL_RECOMP_SLAVE_%';
DOC>
DOC> 2. Query showing UTL_RECOMP jobs that are running
DOC> SELECT job_name FROM dba_scheduler_running_jobs
DOC> WHERE job_name like 'UTL_RECOMP_SLAVE_%';
DOC>#

PL/SQL procedure successfully completed.

TIMESTAMP
-----
COMP_TIMESTAMP UTLRP_END 2010-07-07 14:29:51

DOC> The following query reports the number of objects that have compiled
DOC> with errors (objects that compile with errors have status set to 3 in
DOC> obj$). If the number is higher than expected, please examine the error
DOC> messages reported with each object (using SHOW ERRORS) to see if they
DOC> point to system misconfiguration or resource constraints that must be
DOC> fixed before attempting to recompile these objects.
DOC>#

OBJECTS WITH ERRORS
-----
0

DOC> The following query reports the number of errors caught during
DOC> recompilation. If this number is non-zero, please query the error
DOC> messages in the table UTL_RECOMP_ERRORS to see if any of these errors
DOC> are due to misconfiguration or resource constraints that must be
DOC> fixed before objects can compile successfully.
DOC>#

ERRORS DURING RECOMPILATION
-----
0

PL/SQL procedure successfully completed.
```

Por último se verifican los objetos inválidos para cada usuario tanto en la base origen como en la base destino, esto es mediante el siguiente query:

```
SQL> select object_name, object_type, status from dba_objects  
where owner='ESQUEMA' and status='INVALID';
```

Como se ha realizado durante las migraciones del proyecto se debe realizar una validación por parte del cliente para que se pueda dar por concluido dicho proceso.

2.13 MIGRACIÓN DE LA BASE DE DATOS VIDECUFI

OBTENCIÓN DE DATOS EN LA BASE ORIGEN.

Para realizar dicha tarea es necesario ejecutar scripts en la base origen, conforme a la siguiente tabla.

Tabla de ejecución de scripts

No.	Información	Script
1	tablespaces	> busca_ts.sh > crea_ts2.sh
2	perfiles	> Crprofs.sql
3	roles	> Crroles.sql
4	usuarios	> Crusers.sql
5	grants de objetos	> Crgrants.sql
6	grants de sistema	> Crgrantspostsys.sql
7	links	> Crlinks.sql
8	sinónimos	> Crsyms.sql

CREAR UN RESPALDO DE LA INSTANCIA

Hacer el export de los esquemas a migrar en la base videcufi origen con el siguiente script:

```
# variables de ambiente de oracle

ORACLE_HOME=/oracle10g/product/Oral0g
export ORACLE_HOME
ORACLE_BASE=/oracle10g/product/Oral0g
export ORACLE_BASE
ORACLE_SID=videcufi
export ORACLE_SID

cd /home/orapp/migracion/SCRIPTS/VIDECUFI
/oracle10g/product/Oral0g/bin/./exp system/slst3m@videcufi
OWNER=admzufi, desacufi
FILE=/home/orapp/migracion/SCRIPTS/VIDECUFI/videcufi.dmp
LOG=/home/orapp/migracion/SCRIPTS/VIDECUFI/videcufi.log
STATISTICS=NONE
```

CREACIÓN DE OBJETOS EN LA INSTANCIA DESTINO

En la base destino crear los objetos DDL con los scripts que se obtienen en la ejecución de scripts en la base origen, así que declaramos las variables ORACLE_HOME, ORACLE_SID y EL PATH para la instancia VIDEUCUFI y ejecutar los scripts, esto es en el orden siguiente:

No.	Informacion	Script
1	Tablespace	crea_ts_”instancia”.sql
2	Perfiles	cr_prof.sql
3	Roles	cr_role.sql
4	Usuarios	cr_user.sql
5	Grants de sistema	cr_grantpost.sql
6	Grants de objetos	cr_grant.sql

REALIZACIÓN DEL IMPORT EN EL SERVIDOR

Script para hacer el import para el schema DESACUFI

```
#!/bin/bash
ORACLE_HOME=/u01/app/orapp/product/11.2.0/dbhome_1
PATH=$ORACLE_HOME/bin:$PATH
ORACLE_SID=videcufi
export ORACLE_HOME PATH ORACLE_SID
cd /migracion/videcufi
imp                               system/oracle                buffer=600000
file=/migracion/videcufi/bkvidecufi130710.dmp lo
g=/migracion/videcufi/videcufi_imp1.log          fromuser=DESACUFI
touser=DESACUFI
```

Para el schema ADMCUFI

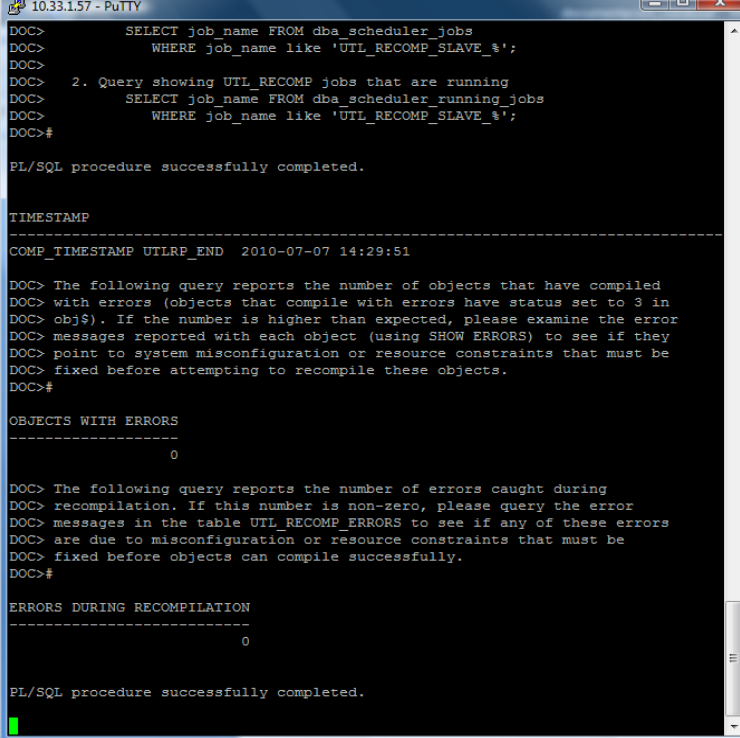
```
#!/bin/bash
ORACLE_HOME=/u01/app/orapp/product/11.2.0/dbhome_1
PATH=$ORACLE_HOME/bin:$PATH
ORACLE_SID=videcufi

export ORACLE_HOME PATH ORACLE_SID
cd /migracion/videcufi
imp                               system/oracle                buffer=600000
file=/migracion/videcufi/bkvidecufi130710.dmp lo
g=/migracion/videcufi/videcufi_imp.log          fromuser=ADMCUFI
touser=ADMCUFI
```

Despues de terminar de realizar los import se ejecuta el script llamado cr_syn.sql para la generación de los sinónimos

Una vez terminado lo anterior, se debe recompilar los objetos inválidos ejecutando el script utlrp.sql ubicado en

\$ORACLE_HOME/rdbms/admin



```

10.33.157 - PuTTY
DOC>      SELECT job_name FROM dba_scheduler_jobs
DOC>      WHERE job_name like 'UTL_RECOMP_SLAVE_%';
DOC>
DOC>  2. Query showing UTL_RECOMP jobs that are running
DOC>      SELECT job_name FROM dba_scheduler_running_jobs
DOC>      WHERE job_name like 'UTL_RECOMP_SLAVE_%';
DOC>#

PL/SQL procedure successfully completed.

TIMESTAMP
-----
COMP_TIMESTAMP UTLRP_END  2010-07-07 14:29:51

DOC> The following query reports the number of objects that have compiled
DOC> with errors (objects that compile with errors have status set to 3 in
DOC> obj$). If the number is higher than expected, please examine the error
DOC> messages reported with each object (using SHOW ERRORS) to see if they
DOC> point to system misconfiguration or resource constraints that must be
DOC> fixed before attempting to recompile these objects.
DOC>#

OBJECTS WITH ERRORS
-----
0

DOC> The following query reports the number of errors caught during
DOC> recompilation. If this number is non-zero, please query the error
DOC> messages in the table UTL_RECOMP ERRORS to see if any of these errors
DOC> are due to misconfiguration or resource constraints that must be
DOC> fixed before objects can compile successfully.
DOC>#

ERRORS DURING RECOMPILATION
-----
0

PL/SQL procedure successfully completed.

```

Por último se verifican los objetos inválidos para cada usuario tanto en la base origen como en la base destino, esto es mediante el siguiente query:

```
SQL> select object_name, object_type, status from dba_objects
where owner='ESQUEMA' and status='INVALID';
```

Antes de dar por concluido el proceso de la migración como se ha mencionado en puntos anteriores el cliente debe validar que dicha migración no presenta errores.

2.14 MIGRACIÓN DE LA BASE DE DATOS REMESAS

OBTENCIÓN DE DATOS EN LA BASE ORIGEN.

Para hacer la tarea es necesario ejecutar los scripts en la base origen de acuerdo a la siguiente tabla.

Tabla de ejecución de scripts

No.	Información	Script
1	tablespaces	> busca_ts.sh > crea_ts2.sh
2	perfiles	> Crprofs.sql
3	roles	> Crroles.sql
4	usuarios	> Crusers.sql
5	grants de objetos	> Crgrants.sql
6	grants de sistema	> Crgrantspostsys.sql
7	links	> Crlinks.sql
8	sinónimos	> Crsyms.sql

CREAR UN RESPALDO DE LA INSTANCIA

Hacer el export de los esquemas a migrar en la base remesas origen con el siguiente script:

```
# variables de ambiente de oracle
ORACLE_HOME=/oracle10g/product/Ora10g
export ORACLE_HOME
ORACLE_BASE=/oracle10g/product/Ora10g
export ORACLE_BASE
ORACLE_SID=remesas
export ORACLE_SID

cd /home/orapp/migracion/SCRIPTS/REMESAS
/oracle10g/product/Ora10g/bin/./exp system/slst3m@remesas
OWNER=ADMREMESAS FILE=remesas.dmp
LOG=/home/orapp/migracion/SCRIPTS/REMESAS/remesas.log STATISTICS=NONE
```


CREACIÓN DE OBJETOS EN LA INSTANCIA DESTINO 10.33.1.44

En la base destino crear los objetos DDL con los scripts que se obtienen en la ejecución de scripts en la base origen, así que declaramos las variables ORACLE_HOME, ORACLE_SID y EL PATH para la instancia REMESAS y ejecutar los scripts, esto es en el orden siguiente:

No.	Informacion	Script
1	Tablespace	<i>crea_ts_”instancia”.sql</i>
2	Perfiles	<i>cr_prof.sql</i>
3	Roles	<i>cr_role.sql</i>
4	Usuarios	<i>cr_user.sql</i>
5	Grants de sistema	<i>cr_grantpost.sql</i>
6	Grants de objetos	<i>cr_grant.sql</i>

REALIZACIÓN DEL IMPORT EN EL SERVIDOR 10.33.1.44

Script para hacer el import para el schema ADMREMESAS

```
#!/bin/bash
ORACLE_HOME=/u01/app/orapp/product/11.2.0/dbhome_1
PATH=$ORACLE_HOME/bin:$PATH
ORACLE_SID=remesas
export ORACLE_HOME PATH ORACLE_SID
cd /migracion/remesas
imp system/oracle buffer=600000 file=/migracion/remesas/remesas.dmp
log= /migracion/remesas/remesas_imp.log fromuser=ADMREMESAS
touser=ADMREMESAS
```

Después de terminar de realizar los import se ejecuta el script *cr_syn.sql* para la generación de los sinónimos.

Para terminar recompilar los objetos inválidos ejecutando el script utlrp.sql ubicado en \$ORACLE_HOME/rdbms/admin

```

10.33.1.57 - PuTTY
DOC>      SELECT job_name FROM dba_scheduler_jobs
DOC>      WHERE job_name like 'UTL_RECOMP_SLAVE_%';
DOC>
DOC> 2. Query showing UTL_RECOMP jobs that are running
DOC>      SELECT job_name FROM dba_scheduler_running_jobs
DOC>      WHERE job_name like 'UTL_RECOMP_SLAVE_%';
DOC>#

PL/SQL procedure successfully completed.

TIMESTAMP
-----
COMP_TIMESTAMP UTLRP_END 2010-07-07 14:29:51

DOC> The following query reports the number of objects that have compiled
DOC> with errors (objects that compile with errors have status set to 3 in
DOC> obj$). If the number is higher than expected, please examine the error
DOC> messages reported with each object (using SHOW ERRORS) to see if they
DOC> point to system misconfiguration or resource constraints that must be
DOC> fixed before attempting to recompile these objects.
DOC>#

OBJECTS WITH ERRORS
-----
0

DOC> The following query reports the number of errors caught during
DOC> recompilation. If this number is non-zero, please query the error
DOC> messages in the table UTL_RECOMP_ERRORS to see if any of these errors
DOC> are due to misconfiguration or resource constraints that must be
DOC> fixed before objects can compile successfully.
DOC>#

ERRORS DURING RECOMPILATION
-----
0

PL/SQL procedure successfully completed.

```

Por último se verifican los objetos inválidos para cada usuario tanto en la base origen como en la base destino, esto es mediante el siguiente query:

```
SQL> select object_name, object_type, status from dba_objects
where owner='ESQUEMA' and status='INVALID';
```

Con esto se da por terminado el proceso de migración de la base de datos al nuevo servidor y de versión, el cliente debe validar que no presente ningún problema con la misma antes de poder realizar la migración de manera productiva.

2.15 MIGRACIÓN DE LA BASE DE DATOS MAE

RECOPIACIÓN DE INFORMACION EN LA BASE DE DATOS ORIGEN

Para recopilar toda la información de objetos creados en la base de datos es necesario ejecutar los siguientes scripts:

Tabla de ejecución de scripts

No.	Información	Script
1	tablespaces	> busca_ts.sh > crea_ts2.sh
2	perfiles	> Crprofs.sql
3	roles	> Crroles.sql
4	usuarios	> Crusers.sql
5	grants de objetos	> Crgrants.sql
6	grants de sistema	> Crgrantspostsys.sql
7	links	> Crlinks.sql
8	sinónimos	> Crsyms.sql

-Se procede a hacer el export de los schemas que se migrarán de la base de datos, el export es el siguiente:

```
# variables de ambiente de oracle
ORACLE_HOME=/oracle10g/product/Ora10g
export ORACLE_HOME
ORACLE_BASE=/oracle10g/product/Ora10g
export ORACLE_BASE
ORACLE_SID=mae
export ORACLE_SID

cd /home/orapp/migracion/SCRIPTS/MAE
/oracle10g/product/Ora10g/bin/./exp system/slst3m@remesas OWNER=MAE
FILE=mae.dmp LOG=/home/orapp/migracion/SCRIPTS/REMESAS/mae.log
STATISTICS=NONE
```

Los esquemas a migrar son:

MAE

Ahora se crearán los objetos en la base destino para poder importar los datos de los schemas mediante la ejecución de los siguientes scripts.

No.	Informacion	Script
1	Tablespace	<i>crea_ts_”instancia”.sql</i>
2	Perfiles	<i>cr_prof.sql</i>
3	Roles	<i>cr_role.sql</i>
4	Usuarios	<i>cr_user.sql</i>
5	Grants de sistema	<i>cr_grantpost.sql</i>
6	Grants de objetos	<i>cr_grant.sql</i>

Ya que se tiene lo anterior se proceder a hacer el import de los esquemas en la base destino, el import para cada esquema se presenta a continuación:

```
#!/bin/bash
ORACLE_HOME=/u01/app/orapp/product/11.2.0/dbhome_1
PATH=$ORACLE_HOME/bin:$PATH
ORACLE_SID=mae
export ORACLE_HOME PATH ORACLE_SID

cd /migracion/mae
imp system/oracle buffer=600000 file=/migracion/remesas/mae.dmp log=
/migracion/remesas/mae.log fromuser=MAE touser=MAE
```

Despues de importar todos los squemas, se ejecita el script para crear los sinónimos que es *cr_syn.sql*:

Se recompilan los objetos ejecutando utlrp.sql de la siguiente manera.

```
SQL> @$ORACLE_HOME/rdbms/admin/utlrp.sql
```

```

10.33.157 - PuTTY
DOC>          SELECT job_name FROM dba_scheduler_jobs
DOC>          WHERE job_name like 'UTL_RECOMP_SLAVE_%';
DOC>
DOC> 2. Query showing UTL_RECOMP jobs that are running
DOC>          SELECT job_name FROM dba_scheduler_running_jobs
DOC>          WHERE job_name like 'UTL_RECOMP_SLAVE_%';
DOC>#

PL/SQL procedure successfully completed.

TIMESTAMP
-----
COMP_TIMESTAMP UTLRP_END 2010-07-07 14:29:51

DOC> The following query reports the number of objects that have compiled
DOC> with errors (objects that compile with errors have status set to 3 in
DOC> obj$). If the number is higher than expected, please examine the error
DOC> messages reported with each object (using SHOW ERRORS) to see if they
DOC> point to system misconfiguration or resource constraints that must be
DOC> fixed before attempting to recompile these objects.
DOC>#

OBJECTS WITH ERRORS
-----
0

DOC> The following query reports the number of errors caught during
DOC> recompilation. If this number is non-zero, please query the error
DOC> messages in the table UTL_RECOMP_ERRORS to see if any of these errors
DOC> are due to misconfiguration or resource constraints that must be
DOC> fixed before objects can compile successfully.
DOC>#

ERRORS DURING RECOMPILATION
-----
0

PL/SQL procedure successfully completed.

```

Por último se verifican los objetos invalidos para cada usuario tanto en la base origen como en la base destino, esto es mediante el siguiente query:

```
SQL> select object_name, object_type, status from dba_objects
where owner='ESQUEMA' and status='INVALID';
```

Para dar por terminado el proceso de la migración, el cliente debe validar que no presente ningun problema con la misma antes de poder realizar la migración de manera productiva.

2.16 MIGRACIÓN DE LA BASE DE DATOS ESTAD

RECOPIACIÓN DE INFORMACIÓN EN LA BASE DE DATOS ORIGEN

Para recopilar toda la información de objetos creados en la base de datos es necesario ejecutar los siguientes scripts:

Tabla de ejecución de scripts

No.	Información	Script
1	tablespaces	> busca_ts.sh > crea_ts2.sh
2	perfiles	> Crprofs.sql
3	roles	> Crroles.sql
4	usuarios	> Crusers.sql
5	grants de objetos	> Crgrants.sql
6	grants de sistema	> Crgrantspostsys.sql
7	links	> Crlinks.sql
8	sinónimos	> Crsyns.sql

Se procede a hacer el export de los esquemas que se migrarán de la base de datos, el export es el siguiente:

```
# variables de ambiente de oracle
ORACLE_HOME=/oracle10g/product/Ora10g
export ORACLE_HOME
ORACLE_BASE=/oracle10g/product/Ora10g
export ORACLE_BASE
ORACLE_SID=estad
export ORACLE_SID

cd /home/orapp/migracion/SCRIPTS/ESTAD
/oracle10g/product/Ora10g/bin/./exp system/s1st3m OWNER=ESTAD
FILE=estad.dmp LOG=estad.log STATISTICS=NONE
```

Los esquemas a migrar son:

ESTAD

Ahora se crearan los objetos en la base destino para poder importar los datos de los esquemas mediante la ejecución de los siguientes scripts.

No.	Informacion	Script
1	Tablespace	crea_ts_”instancia”.sql
2	Perfiles	cr_prof.sql
3	Roles	cr_role.sql
4	Usuarios	cr_user.sql
5	Grants de sistema	cr_grantpost.sql
6	Grants de objetos	cr_grant.sql

-Ya que se tiene lo anterior se proceder a hacer el import de los esquemas en la base destino que está ubicada en el servidor 10.33.1.44, el import para cada esquema se presenta a continuación:

```
#!/bin/bash
ORACLE_HOME=/u01/app/orapp/product/11.2.0/dbhome_1
PATH=$ORACLE_HOME/bin:$PATH
ORACLE_SID=estad
export ORACLE_HOME PATH ORACLE_SID

cd /migracion/estad
imp system/oracle buffer=600000 file=/migracion/estad/estad.dmp log=
/migracion/estad/estad.log fromuser=ESTAD touser=ESTAD
```

-Despues de importar todos los squemas, se ejecita el script para crear los sinónimos que es cr_syn.sql.

-Se recompilan los objetos ejecutando utlrp.sql de la siguiente manera.

```
SQL> @$ORACLE_HOME/rdbms/admin/utlrp.sql
```

```
10.33.157 - PuTTY
DOC> SELECT job_name FROM dba_scheduler_jobs
DOC> WHERE job_name like 'UTL_RECOMP_SLAVE_%';
DOC>
DOC> 2. Query showing UTL_RECOMP jobs that are running
DOC> SELECT job_name FROM dba_scheduler_running_jobs
DOC> WHERE job_name like 'UTL_RECOMP_SLAVE_%';
DOC>#
PL/SQL procedure successfully completed.

TIMESTAMP
-----
COMP_TIMESTAMP UTLRP_END 2010-07-07 14:29:51

DOC> The following query reports the number of objects that have compiled
DOC> with errors (objects that compile with errors have status set to 3 in
DOC> obj$). If the number is higher than expected, please examine the error
DOC> messages reported with each object (using SHOW ERRORS) to see if they
DOC> point to system misconfiguration or resource constraints that must be
DOC> fixed before attempting to recompile these objects.
DOC>#

OBJECTS WITH ERRORS
-----
0

DOC> The following query reports the number of errors caught during
DOC> recompilation. If this number is non-zero, please query the error
DOC> messages in the table UTL_RECOMP_ERRORS to see if any of these errors
DOC> are due to misconfiguration or resource constraints that must be
DOC> fixed before objects can compile successfully.
DOC>#

ERRORS DURING RECOMPILATION
-----
0

PL/SQL procedure successfully completed.
```

-Por último se verifican los objetos invalidos para cada usuario tanto en la base origen como en la base destino, esto es mediante el siguiente query:

```
SQL> select object_name, object_type, status from dba_objects  
where owner='ESQUEMA' and status='INVALID';
```

Con esto se da por terminado el proceso de migración de la base de datos al nuevo servidor y de versión, el cliente debe validar que no presente ningun problema con la misma antes de poder realizar la migracion de manera productiva.

2.17 MIGRACIÓN DE LA BASE DE DATOS SIO

OBTENCIÓN DE DATOS EN LA BASE ORIGEN.

Para realizar dicha tarea es necesario ejecutar estos scripts en la base origen de acuerdo a la siguiente tabla.

Tabla de ejecución de scripts

No.	Información	Script
1	tablespaces	> busca_ts.sh > crea_ts2.sh
2	perfiles	> Crprofs.sql
3	roles	> Crroles.sql
4	usuarios	> Crusers.sql
5	grants de objetos	> Crgrants.sql
6	grants de sistema	> Crgrantspostsys.sql
7	links	> Crlinks.sql
8	sinónimos	> Crsyns.sql

CREAR UN RESPALDO DE LA INSTANCIA

Hacer el export de los schemas a migrar en la base sio origen con el siguiente script:

```
# variables de ambiente de oracle
ORACLE_HOME=/oracle10g/product/Ora10g
export ORACLE_HOME
ORACLE_BASE=/oracle10g/product/Ora10g
export ORACLE_BASE
ORACLE_SID=sio
export ORACLE_SID

cd /home/orapp/migracion/SCRIPTS/sio
/oracle10g/product/Ora10g/bin/./exp system/s1st3m OWNER=SIO
FILE=sio.dmp LOG=sio.log STATISTICS=NONE
```

CREACIÓN DE OBJETOS EN LA INSTANCIA DESTINO

En la base destino crear los objetos DDL con los scripts que se obtienen en la ejecución de scripts en la base origen, así que declaramos las variables

ORACLE_HOME, *ORACLE_SID* y *EL PATH* para la instancia *SIO* y ejecutar los scripts, esto es en el orden siguiente:

No.	Informacion	Script
1	Tablespace	<i>crea_ts_”instancia”.sql</i>
2	Perfiles	<i>cr_prof.sql</i>
3	Roles	<i>cr_role.sql</i>
4	Usuarios	<i>cr_user.sql</i>
5	Grants de sistema	<i>cr_grantpost.sql</i>
6	Grants de objetos	<i>cr_grant.sql</i>

REALIZACION DEL IMPORT EN EL SERVIDOR DESTINO

Script para hacer el import para el schema *SIO*

```
#!/bin/bash
ORACLE_HOME=/u01/app/orapp/product/11.2.0/dbhome_1
PATH=$ORACLE_HOME/bin:$PATH
ORACLE_SID=sio
NLS_LANG=SPANISH_MEXICO.WE8ISO8859P15
export ORACLE_HOME PATH ORACLE_SID NLS_LANG

cd /migracion/sio
imp system/oracle buffer=600000 file=/migracion/sio/sio.dmp
log=/migracion/sio/sio.log fromuser=SIO touser=SIO
```

Después de terminar de realizar los import se ejecuta el script para la generación de los sinónimos.

Antes de terminar se necesita recompilar los objetos inválidos ejecutando el script utlrp.sql ubicado en \$ORACLE_HOME/rdbms/admin

```

10.33.157 - PuTTY
DOC>      SELECT job_name FROM dba_scheduler_jobs
DOC>      WHERE job_name like 'UTL_RECOMP_SLAVE_%';
DOC>
DOC> 2. Query showing UTL_RECOMP jobs that are running
DOC>      SELECT job_name FROM dba_scheduler_running_jobs
DOC>      WHERE job_name like 'UTL_RECOMP_SLAVE_%';
DOC>#

PL/SQL procedure successfully completed.

TIMESTAMP
-----
COMP_TIMESTAMP UTLRP_END 2010-07-07 14:29:51

DOC> The following query reports the number of objects that have compiled
DOC> with errors (objects that compile with errors have status set to 3 in
DOC> obj$). If the number is higher than expected, please examine the error
DOC> messages reported with each object (using SHOW ERRORS) to see if they
DOC> point to system misconfiguration or resource constraints that must be
DOC> fixed before attempting to recompile these objects.
DOC>#

OBJECTS WITH ERRORS
-----
0

DOC> The following query reports the number of errors caught during
DOC> recompilation. If this number is non-zero, please query the error
DOC> messages in the table UTL_RECOMP_ERRORS to see if any of these errors
DOC> are due to misconfiguration or resource constraints that must be
DOC> fixed before objects can compile successfully.
DOC>#

ERRORS DURING RECOMPILATION
-----
0

PL/SQL procedure successfully completed.

```

Por último se verifican los objetos inválidos para cada usuario tanto en la base origen como en la base destino, esto es mediante el siguiente query:

```
SQL> select object_name, object_type, status from dba_objects
where owner='ESQUEMA' and status='INVALID';
```

Con esto se da por terminado el proceso de migración de la base de datos al nuevo servidor y de versión, el cliente debe validar que no presente ningún problema con la misma antes de poder realizar la migración de manera productiva.

2.18 MIGRACIÓN DE LA BASE DE DATOS SIOBK2

RECOPIACIÓN DE INFORMACIÓN EN LA BASE DE DATOS ORIGEN

Para recopilar toda la información de objetos creados en la base de datos es necesario ejecutar los siguientes scripts siguientes:

Tabla de ejecución de scripts

No.	Información	Script
1	tablespaces	> busca_ts.sh > crea_ts2.sh
2	perfiles	> Crprofs.sql
3	roles	> Crroles.sql
4	usuarios	> Crusers.sql
5	grants de objetos	> Crgrants.sql
6	grants de sistema	> Crgrantspostsys.sql
7	links	> Crlinks.sql
8	sinónimos	> Crsyns.sql

Se procede a hacer el export de los esquemas que se migraran de la base de datos, el export es el siguiente:

```
# variables de ambiente de oracle
ORACLE_HOME=/oracle10g/product/Ora10g
export ORACLE_HOME
ORACLE_BASE=/oracle10g/product/Ora10g
export ORACLE_BASE
ORACLE_SID=siobk2
export ORACLE_SID

cd /home/orapp/migracion/SCRIPTS/siobk2
/oracle10g/product/Ora10g/bin/./exp system/s1st3m OWNER=SIO
FILE=siobk2.dmp LOG=siobk2.log STATISTICS=NONE
```

Los esquemas a migrar son:

SIO

Ahora se crearan los objetos en la base destino para poder importar los datos de los esquemas, mediante la ejecución de los siguientes scripts:

<i>No.</i>	<i>Informacion</i>	<i>Script</i>
1	Tablespace	<i>crea_ts_”instancia”.sql</i>
2	Perfiles	<i>cr_prof.sql</i>
3	Roles	<i>cr_role.sql</i>
4	Usuarios	<i>cr_user.sql</i>
5	Grants de sistema	<i>cr_grantpost.sql</i>
6	Grants de objetos	<i>cr_grant.sql</i>

Ya que se tiene lo anterior se proceder a hacer el import de los esquemas en la base destino, el import para cada esquema se presenta a continuación:

```
#!/bin/bash
ORACLE_HOME=/u01/app/orapp/product/11.2.0/dbhome_1
PATH=$ORACLE_HOME/bin:$PATH
ORACLE_SID=siobk2
NLS_LANG=SPANISH_MEXICO.WE8ISO8859P15
export ORACLE_HOME PATH ORACLE_SID NLS_LANG

cd /migracion/siobk2
imp system/oracle buffer=600000 file=/migracion/siobk2/siobk2.dmp
log=/migracion/siobk2/siobk2.log fromuser=SIO touser=SIO
```

-Despues de importar todos los squemas, se ejecita el script para crear los sinónimos que es *cr_syn.sql*.

-Se recompilan los objetos ejecutando utlrlp.sql de la siguiente manera.

```
SQL> @$ORACLE_HOME/rdbms/admin/utlrlp.sql
```

```

10.33.1.57 - PuTTY
DOC>      SELECT job_name FROM dba_scheduler_jobs
DOC>      WHERE job_name like 'UTL_RECOMP_SLAVE_%';
DOC>
DOC> 2. Query showing UTL_RECOMP jobs that are running
DOC>      SELECT job_name FROM dba_scheduler_running_jobs
DOC>      WHERE job_name like 'UTL_RECOMP_SLAVE_%';
DOC>#

PL/SQL procedure successfully completed.

TIMESTAMP
-----
COMP_TIMESTAMP UTLRP_END 2010-07-07 14:29:51

DOC> The following query reports the number of objects that have compiled
DOC> with errors (objects that compile with errors have status set to 3 in
DOC> obj$). If the number is higher than expected, please examine the error
DOC> messages reported with each object (using SHOW ERRORS) to see if they
DOC> point to system misconfiguration or resource constraints that must be
DOC> fixed before attempting to recompile these objects.
DOC>#

OBJECTS WITH ERRORS
-----
0

DOC> The following query reports the number of errors caught during
DOC> recompilation. If this number is non-zero, please query the error
DOC> messages in the table UTL_RECOMP_ERRORS to see if any of these errors
DOC> are due to misconfiguration or resource constraints that must be
DOC> fixed before objects can compile successfully.
DOC>#

ERRORS DURING RECOMPILATION
-----
0

PL/SQL procedure successfully completed.

```

-Por último se verifican los objetos invalidos para cada usuario tanto en la base origen como en la base destino, esto es mediante el siguiente query:

```
SQL> select object_name, object_type, status from dba_objects
where owner='ESQUEMA' and status='INVALID';
```

El cliente debe realizar la validacion de que la migracion de base de datos al nuevo servidor no presente problemas de conexión ó con los datos de la misma. Con lo cual se da por concluido el proceso de migración de dicha base.

Con este ultimo paso se concluye las migraciones de pruebas presentadas al principio del proyecto, en el capitulo siguiente mostraremos el proceso para realizar la migración de una de las bases de datos de forma productiva.

Capitulo 3

MIGRACIÓN A PRODUCCIÓN

MIGRACIÓN A PRODUCCIÓN

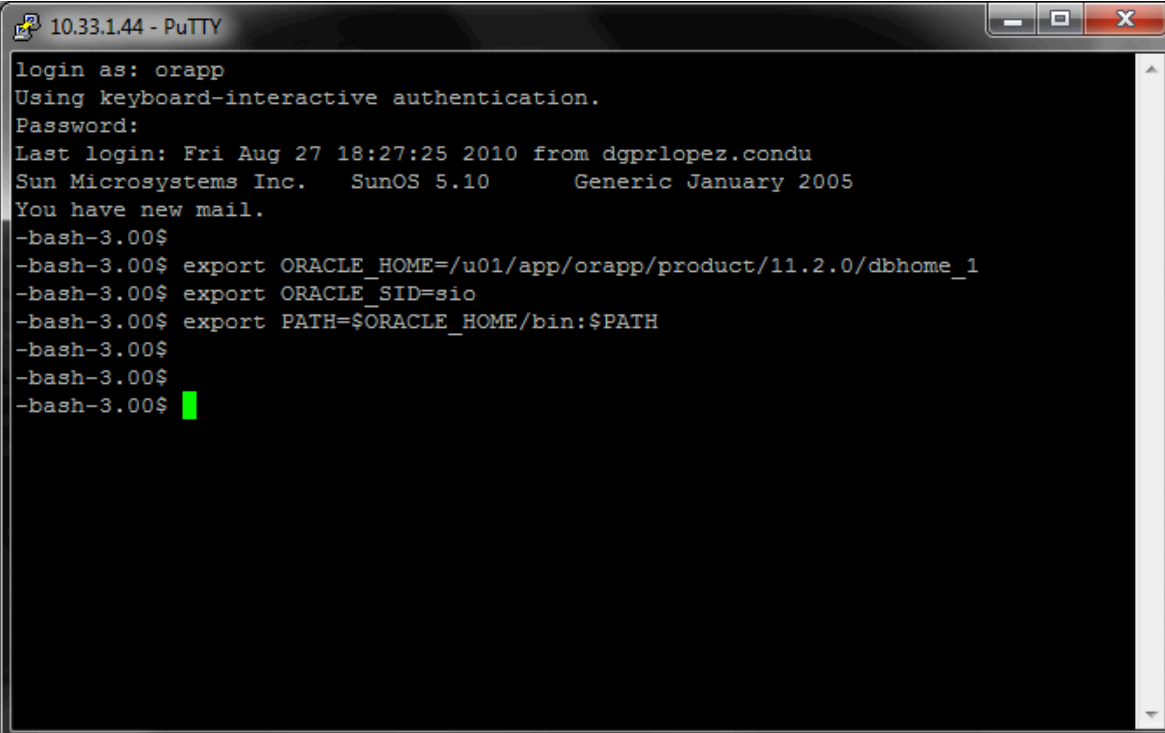
El proceso de migración a producción de las Bases de Datos es mucho mas sencillo, ya que la parte mas complicada es durante la etapa de pruebas, debido a que se tiene que llevar a cabo la instalación y configuración de los nuevos servidores hacia los que se migrara la base de datos. En esta ultima etapa solo será necesario hacer la eliminación de los schemas que se migraron anteriormente, esto es para recrearlos, y una vez que se hace esto, hacer la carga de toda la información perteneciente a dichos schemas.

La migración a producción se hace una vez que han sido aprobadas las Bases de Datos en las etapas de prueba, esta aprobación costa en validar que las aplicaciones utilizadas con las bases funcionan correctamente.

Debido a que en la etapa de pruebas se indica que schemas son los que se migran, y el proceso es el mismo para todas las Bases de Datos, solo se pondrá la migración de una de las Bases mas importantes de la dependencia.

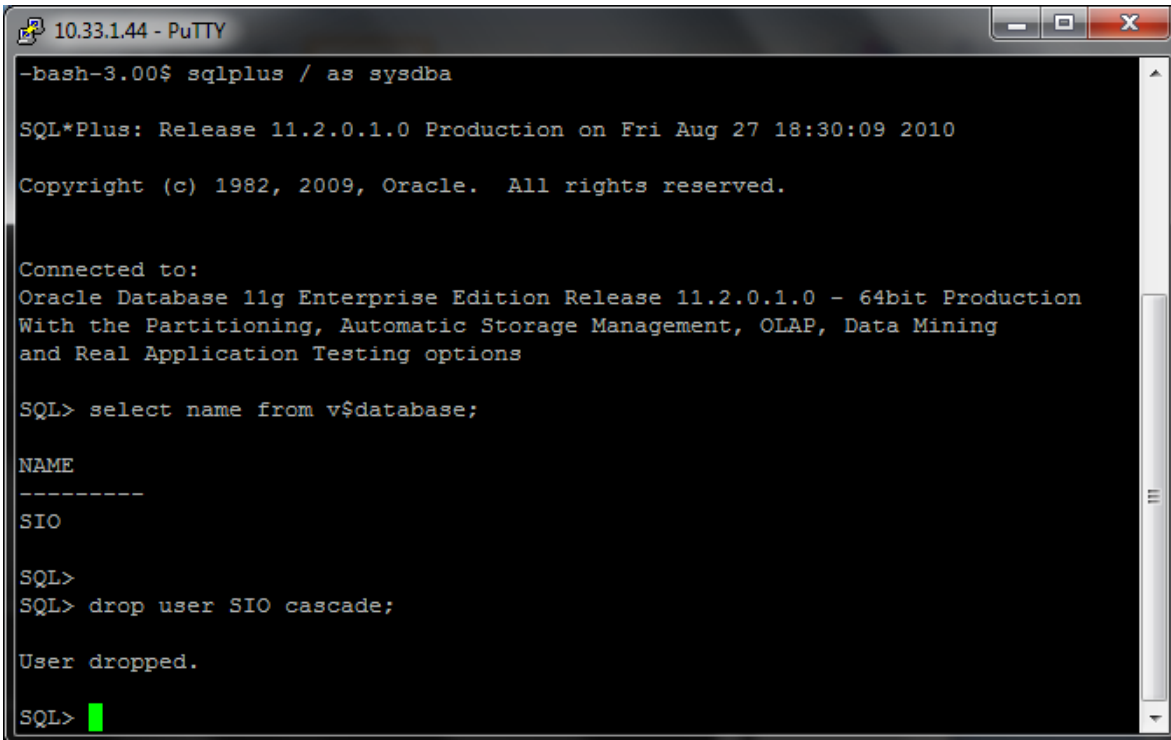
3.1 MIGRACIÓN DE PRODUCCIÓN DE SIO

En el servidor destino borrar los schemas a migrar de sio para ello definir los parámetros ORACLE_HOME, ORACLE_SID, PATH. Una vez definido lo anterior entrar a una sesión de SQL.



```
10.33.144 - PuTTY
login as: orapp
Using keyboard-interactive authentication.
Password:
Last login: Fri Aug 27 18:27:25 2010 from dgprlopez.condu
Sun Microsystems Inc. SunOS 5.10 Generic January 2005
You have new mail.
-bash-3.00$
-bash-3.00$ export ORACLE_HOME=/u01/app/orapp/product/11.2.0/dbhome_1
-bash-3.00$ export ORACLE_SID=sio
-bash-3.00$ export PATH=$ORACLE_HOME/bin:$PATH
-bash-3.00$
-bash-3.00$
-bash-3.00$ █
```


Dentro de la sesión de sqlplus borrar los schemas que se van a migrar mediante el comando drop user:



```

10.33.144 - PuTTY
-bash-3.00$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Fri Aug 27 18:30:09 2010

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, Automatic Storage Management, OLAP, Data Mining
and Real Application Testing options

SQL> select name from v$databases;

NAME
-----
SIO

SQL>
SQL> drop user SIO cascade;

User dropped.

SQL>

```

Ejecutar los scripts para obtener información a migrar en la base origen como se hizo en la etapa de pruebas.

No.	Información	Script
1	tablespaces	> crea_ts.sql
2	perfiles	> Crprofs.sql
3	roles	> Crroles.sql
4	usuarios	> Crusers.sql
5	grants de objetos	> Crgrants.sql
6	grants de sistema	> Crgrantspostsys.sql
7	links	> Crlinks.sql
8	sinónimos	> Crsyns.sql

Por último se recompilan los objetos ejecutando utlrp.sql de la siguiente manera.

SQL> @\$ORACLE_HOME/rdbms/admin/utlrp.sql

```

10.33.1.57 - PuTTY
DOC>      SELECT job_name FROM dba_scheduler_jobs
DOC>      WHERE job_name like 'UTL_RECOMP_SLAVE_%';
DOC>
DOC> 2. Query showing UTL_RECOMP jobs that are running
DOC>      SELECT job_name FROM dba_scheduler_running_jobs
DOC>      WHERE job_name like 'UTL_RECOMP_SLAVE_%';
DOC>#
PL/SQL procedure successfully completed.

TIMESTAMP
-----
COMP_TIMESTAMP UTLRP_END 2010-07-07 14:29:51

DOC> The following query reports the number of objects that have compiled
DOC> with errors (objects that compile with errors have status set to 3 in
DOC> obj$). If the number is higher than expected, please examine the error
DOC> messages reported with each object (using SHOW ERRORS) to see if they
DOC> point to system misconfiguration or resource constraints that must be
DOC> fixed before attempting to recompile these objects.
DOC>#

OBJECTS WITH ERRORS
-----
0

DOC> The following query reports the number of errors caught during
DOC> recompilation. If this number is non-zero, please query the error
DOC> messages in the table UTL_RECOMP_ERRORS to see if any of these errors
DOC> are due to misconfiguration or resource constraints that must be
DOC> fixed before objects can compile successfully.
DOC>#

ERRORS DURING RECOMPILATION
-----
0

PL/SQL procedure successfully completed.

```

Por último se verifican los objetos inválidos para cada usuario tanto en la base origen como en la base destino, esto es mediante el siguiente query:

SQL> select object_name, object_type, status from dba_objects where owner='ESQUEMA' and status='INVALID';

CONCLUSIONES

Los proyectos de migración de bases de datos no son iguales, ya que hay distintos tipos de hacerlo, y la finalidad de estos es diferente en sus distintas ocasiones, ya sea por renovación tecnológica, cambios de versiones o ediciones, del tipo de almacenamiento, etc.

En el proyecto presentado en este caso, las migraciones fueron hechas de manera lógica, esto es cuando solo son migrados tanto la estructura definida (DDL), como los datos manipulados por los usuarios (DML). Se puede observar que todas las migraciones mencionadas son parecidas, la diferencia radica no solo desde el punto de vista tecnológico, sino también en el enfoque que cada equipo de trabajo le da a la planeación, implementación y realización de las migraciones, debido a que cada una tiene sus propios riesgos y necesidades.

La migración de bases de datos es compleja, no solo técnicamente, sino también desde el punto de vista del negocio. Pero también es común que las empresas y sus directivas subestimen los riesgos, costos y duración de este tipo de proyectos. Por lo que se puede apreciar que el impacto de una migración fallida para el negocio puede ser bastante grande y desastroso.

En este trabajo hemos tenido la posibilidad de mostrar una forma de cómo realizar paso por paso la migración de 16 bases de datos de las versiones Oracle 9i y 10g hacia la versión Oracle 11g Release 2, de las cuales, en trece de ellas el procedimiento es el mismo llevándolas de una instancia a otra independiente y en las tres restantes cambia al juntarlas en una misma instancia de base de datos.

La otra parte que se muestra en los procedimientos, es el de almacenar las bases de datos en ASM (Automatic Storage Management), que es una forma de almacenamiento administrada por el propio motor de bases de datos Oracle.

De las cosas que nos deja este proyecto podemos mencionar conocimiento y experiencia, ya que durante el desarrollo del mismo hemos crecido mucho en cuanto a los conocimientos adquiridos durante la carrera, llevándolos a la conjunción con el motor de bases de datos y sobre los productos con los que nos desarrollamos laboralmente, un ejemplo es el llevar la base de datos de File System que es el almacenamiento en disco local, hacia ASM, ya que se llevo una investigación y las pruebas necesarias para hacerlo perfectamente, debido a que se estaba trabajando en un ambiente en el que el objetivo era el de llevarlo a producción, cumpliendo esto satisfactoriamente.

En cuanto a la experiencia que nos otorgo el proyecto podemos decir que en ocasiones se puede seguir un manual y ver que todo sale perfectamente, sin embargo sabemos que la realidad no es así, ya que la presión que ejerce el cliente cuando paga una solución con un costo alto, el encontrarte con issues y tener que resolverlos casi instantáneamente no los contendrá cualquier manual, por lo que nos confiere un valor agregado, ya que en el momento en que se presenta nuevamente, se sabrá de que forma manejarlo sin afectar la operación del cliente y al mismo tiempo ganar mayor prestigio y credibilidad en la forma laboral.

Podemos decir que el desarrollo de este proyecto nos abrió las puertas de una manera en que en la empresa para la que actualmente laboramos se nos comenzaron a asignar una serie de distintos proyectos incluyendo migraciones de bases de datos, implementaciones de esquemas de respaldo, administración, entre otras tareas. Por lo

que este trabajo se tomo como el trampolín para pasar de un estado de becarios a quedar contratados como consultores de la compañía.

Este desarrollo, podra servir como guía para los estudiantes que tengan como tarea el hacer una migración de base de datos Oracle y decidan hacerlo por el metodo de migración logica (export/import). Ya que la forma en que esta explicado todo el proyecto, deja ver de forma clara los pasos a seguir para el procedimiento.