

UNIVERSIDAD AUTÓNOMA DE QUERÉTARO

BIBLIOTECA

FACULTAD DE INFORMATICA

Universidad Autónoma de Querétaro
Facultad de Informática

_____ 2002
_____ 2002
_____ 2002
_____ 2002

**Administración de Base de Datos Aplicada a un Sistema
Manejador de Bases de Datos (Oracle)**

TESINA

Que para obtener el título de Licenciado en Informática

Presenta

Luis Eduardo Rodríguez Fernández de Arteaga

Asesor: I.S.C. Jabel Reséndiz González

Santiago de Querétaro, Qro., a 11 de Marzo del 2002



TS
005.74
R696a

F07140

TS
005.74
R696a

F07140



UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
BIBLIOTECA
FACULTAD DE INFORMÁTICA

UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
BIBLIOTECA
FACULTAD DE INFORMÁTICA

No. Adq. F07140
Clasif. TS 005.74
Cutter R696a



CARTA DE ACEPTACIÓN

Por este medio, se otorga constancia de aceptación de tesina para obtener el título de Licenciado en Informática, que presenta el pasante **LUIS EDUARDO RODRÍGUEZ FERNÁNDEZ DE ARTEAGA** con el tema denominado “*Administración de Base de Datos Aplicada a un Sistema Manejador de Bases de Datos (Oracle)*”.

Este trabajo fue desarrollado como una investigación derivada del curso de titulación “**ADMINISTRACIÓN DE BASE DE DATOS**”, dando cumplimiento a uno de los requisitos contemplados en el artículo 34 del reglamento de titulación vigente, en lo referente a la opción de titulación por realización y aprobación de cursos de actualización.

Se extiende la presente para los fines legales a que haya lugar y para su inclusión en todos los ejemplares impresos de la tesina, a los diez días del mes de abril del dos mil dos.

ATENTAMENTE


ING. JABEL RESÉNDIZ GONZÁLEZ
PROFR. CURSO DE TITULACIÓN

INDICE

INDICE	1
ÍNDICE DE FIGURAS Y TABLAS	4
ÍNDICE DE GRÁFICOS	4
ÍNDICE DE TABLAS.....	4
INTRODUCCIÓN	5
1 CONCEPTOS BÁSICOS DE ADMINISTRACIÓN DE BASES DE DATOS	10
1.1 CONCEPTOS INICIALES	14
1.1.1 INSTALACIÓN DEL SOFTWARE DE LA BASE DE DATOS:.....	16
1.2 PLANEACIÓN DE LA BASE DE DATOS	17
1.3 CREACIÓN DE LA BASE DE DATOS	19
1.3.1 PREREQUISITOS DE CREACIÓN.....	20
1.3.2 PASOS PARA LA CREACIÓN DE LA BASE DE DATOS.....	21
1.3.3 PARÁMETROS DE LA BASE DE DATOS.....	23
1.3.4 EJEMPLO DE CREACIÓN DE UNA BASE DE DATOS.....	28
1.3.5 CONSIDERACIONES DESPUÉS DE CREAR UNA BASE DE DATOS.....	29
1.3.6 ARRANCAR LA BASE DE DATOS.....	30
1.3.7 OPCIONES DE ARRANQUE.....	31
1.3.7.1 ARRANQUE NORMAL.....	31
1.3.7.2 ARRANQUE CON MONTAJE.....	31
1.3.7.3 ARRANQUE SIN MONTAJE.....	31
1.3.7.4 ARRANQUE RESTRINGIDO.....	32
1.3.7.5 ARRANQUE FORZADO.....	32
1.3.8 CERRAR LA BASE DE DATOS.....	33
1.3.8.1 SHUTDOWN IMMEDIATE.....	34
1.3.8.2 SHUTDOWN TRANSACTIONAL.....	34
1.3.8.3 SHUTDOWN ABORT.....	35
2. ADMINISTRACIÓN DE LAS ESTRUCTURAS DE ALMACENAMIENTO Y DICCIONARIOS DE DATOS	37
2.1 ADMINISTRACIÓN DE LAS ESTRUCTURAS DE ALMACENAMIENTO	37
2.1.1 ESTRUCTURAS LÓGICAS DE LA BASE DE DATOS.....	37
2.1.1.1 TABLESPACES.....	37
2.1.1.2 MANEJO DE TABLESPACES.....	39
2.1.1.3 ESQUEMAS Y OBJETOS DE ESQUEMA (SCHEMAS Y SCHEMA OBJECTS).....	46

2.1.1.4 BLOQUES.	47
2.1.1.5 EXTENTS.	47
2.1.1.6 SEGMENTOS.	47
2.1.1.7 SEGMENTOS DE DATOS.	48
2.1.1.8 SEGMENTO DE ÍNDICE.	49
2.1.1.9 SEGMENTO DE ROLLBACK.	49
2.1.1.10 SEGMENTO TEMPORAL.	50
2.1.2 ESTRUCTURAS FÍSICAS DE LA BASE DE DATOS.	51
2.1.2.1 DATAFILES (ARCHIVOS DE DATOS)	51
2.1.2.2 MANEJO DE LOS DATAFILES (ARCHIVOS DE DATOS).	52
2.1.2.3 ARCHIVOS REDO LOG.	54
2.1.2.4 ARCHIVOS DE CONTROL.	55
2.1.2.5 MANEJO DE LOS ARCHIVOS DE CONTROL.	56
2.2 UTILERÍAS IMPORT Y EXPORT.	59
2.2.1 IMPORT Y EXPORT.	59
2.2.2 MODOS DE EXPORTACIÓN E IMPORTACIÓN.	63
2.2.2.1 EXPORTACIÓN EN MODO TABLA.	63
2.2.2.2 EXPORTACIÓN EN MODO USUARIO.	63
2.2.2.3 EXPORTACIÓN DE LA BASE DE DATOS COMPLETA.	64
2.2.2.4 IMPORTACIÓN EN MODO TABLA.	64
2.2.2.5 IMPORTACIÓN EN MODO USUARIO.	65
2.2.2.6 IMPORTACIÓN DE LA BASE DE DATOS COMPLETA.	65
2.3 MIGRACIÓN.	66
2.4 UTILIZACIÓN DEL DICCIONARIO DE DATOS.	69
2.4.1 TIPOS DE VISTAS DEL DICCIONARIO DE DATOS.	69
2.4.1.1 LAS VISTAS MÁS ÚTILES PARA ADMINISTRACIÓN.	70
3 ADMINISTRACIÓN DE LA SEGURIDAD.	72
3.1 MANEJO DE USUARIOS.	72
3.1.1 AUTENTIFICACIÓN DE USUARIOS.	72
3.1.1.1 AUTENTIFICACIÓN DESDE LA BASE DE DATOS.	72
3.1.1.2 AUTENTIFICACIÓN EXTERNA.	73
3.1.1.3 AUTENTIFICACIÓN A NIVEL ORGANIZACIÓN.	73
3.1.2 CREACIÓN DE USUARIOS.	74
3.1.3 ALTERAR USUARIOS.	75
3.1.4 ELIMINAR USUARIOS.	76
3.2 MANEJO DE PROFILES (PERFILES).	77
3.2.1 CREACIÓN DE PROFILES.	77
3.2.2 ASIGNACIÓN DE PROFILES.	78
3.2.3 ALTERAR PROFILES.	79
3.2.4 ELIMINAR UN PROFILE.	79
3.2.5 LISTANDO LA INFORMACIÓN DE USUARIOS Y PROFILES.	81

3.3 MANEJO DE PRIVILEGIOS Y ROLES.	82
3.3.1 PRIVILEGIOS	82
3.3.1.1 PRIVILEGIO DE SISTEMA.....	83
3.3.1.2 PRIVILEGIOS DE OBJETOS.....	83
3.3.2 MANEJO DE ROLES.	84
3.3.2.1 CREANDO UN ROL.....	84
3.3.2.2 ELIMINAR ROLES.....	85
3.3.3 OTORGANDO PRIVILEGIOS Y ROLES.....	85
3.3.4 REVOCAR PRIVILEGIOS Y ROLES.....	87
3.3.5 LISTANDO INFORMACIÓN DE PRIVILEGIOS Y ROLES.....	88
3.4 TIPOS DE USUARIOS.	89
3.5 MAS CONCEPTOS SOBRE SEGURIDAD.	92
3.5.1 ESTRATEGIAS DE RESPALDO.....	92
CONCLUSIONES Y COMENTARIO PERSONAL	94
GLOSARIO.....	96
BIBLIOGRAFIA.	97

ÍNDICE DE FIGURAS Y TABLAS

Índice de Gráficos

1.1 Arranque de la Base de Datos	33
2.1 Relación entre bases de datos, tablespaces, y archivos de datos (Datafiles)	38
2.2 Esquema de los Tablespace, Datafiles y Objetos Esquema	46
2.3 Segmentos, Extents y Bloques	48
2.4 Segmentos de Rollback	50
2.5 Estructura de la Base de Datos y Posición de los Data Files en la Misma	51
2.6 Exportación de la base de datos	60
2.7 Importación de la base de datos	60
2.8 Pasos de la migración	66

Índice de Tablas

1.1 Opciones de La Instrucción Shutdown	36
2.1 Tipos de vistas	69
2.2 Vistas DBA mas utiles	70
2.3 Vistas v\$ mas comunes	71

INTRODUCCIÓN.

Hoy en día la información en la vida diaria de muchos de los seres humanos que habitamos este planeta, forma un papel importante ya sea en nuestros trabajos, nuestra educación, o nuestros pasatiempos. La información nos permite saber lo que esta pasando en nuestro entorno y poder prever acciones a futuro o simplemente enterarse de lo que sucede en el mundo a nuestro alrededor.

Al darse esta importancia de la información los sistemas informáticos vienen a tener casi la misma importancia, ya que son los que nos permiten tener acceso a esta información de manera fácil y rápida.

En el mundo moderno donde las organizaciones manejan grandes volúmenes de información, la cual muchas veces es crucial para el desempeño o el éxito de las organizaciones o donde la tecnología y la informática se han adueñado de parte de nuestra cultura, la información se vuelve un recurso que no podemos dejar por un lado.

Los datos que generan la información de las organizaciones deben de ser manejados de manera correcta para que no existan perdidas en este recurso tan importante. Las bases de datos nos permiten tener un control sobre esta información.

Una base de datos es un conjunto de datos relacionados entre sí. Por otro lado un sistema manejador de bases de datos o sistema de gestión de bases de datos consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a esos datos.

Un sistema manejador de bases de datos es un conjunto de programas que permite a los usuarios crear y mantener una base de datos. Facilitando el proceso de definir, construir, y manipular bases de datos para diversas aplicaciones.

El objetivo de un sistema de gestión de bases de datos es proporcionar un entorno que sea tanto práctico como eficiente de usar en el almacenamiento de la información de la base de datos. Proporcionando fiabilidad en la información almacenada a pesar de las fallas que se puedan presentar y de que existan personas que deseen violar su seguridad.

En los sistemas de procesamiento de archivos típicos los registros permanentes son almacenados en varios archivos y se escriben diferentes programas de aplicación para extraer y añadir registros a los archivos que deben de ser. Lo cual genera problemas como redundancia en los datos, dificultad de acceso, aislamiento de los datos, problemas de integridad, problemas de atomicidad, problemas de seguridad, etc. Este problema son eliminados o previstos en un sistema manejador de bases de datos.

Los datos que se manejan en las bases de datos de las organizaciones generalmente son muy importantes para la operación del negocio, su toma de decisiones, y para llevar a cabo con éxito sus procedimientos administrativos y por ende los objetivos organizacionales. El sistema manejador de bases de datos nos permite tener este control sobre los datos. Pero por la misma importancia de ellos se tiene que tener mucho cuidado en su uso.

Una de las razones para utilizar un sistema manejador de bases de datos es para tener un control centralizado tanto de los datos como de los programas que accedan a esos datos. La persona que tiene el control central sobre el sistema es el administrador de bases de datos (ADB ó DBA Data Base Administrator).

En las organizaciones de buen tamaño normalmente muchas personas utilizan los mismos recursos informáticos para esto se requiere un persona que sea la autoridad que supervise, coordine y controle los recursos. Hablando de bases de datos el primer recurso que se debe cuidar es la base de datos, además del buen funcionamiento del sistema manejador de base de datos.

La administración de bases de datos es un conjunto de tareas que realiza el administrador de bases de datos (DBA) las cuales son actividades técnicas de la base de datos que permiten el mejor desempeño de la base de datos.

La administración de bases de datos es muy extensa ya que esta relacionada con casi todos los aspectos que engloban un sistema de bases de datos. Dentro de ella engloba conceptos como instalación de la base de datos, manejo del almacenamiento, manejo de la seguridad, manejo del funcionamiento de la base de datos y de la integridad de los datos contenidos en ella.

Al referirme a la administración de bases de datos aplicadas a un sistema manejador de bases de datos me estoy refiriendo a dos cosas que voy a hablar de algunas de las tareas principales que realiza un administrador de bases de datos y por otro lado que estas actividades van a estar enfocadas a su aplicación en un manejador de bases de datos que en este caso es Oracle.

Oracle es un sistema manejador de bases de datos que además de tener la base de datos en si tiene muchas herramientas para la administración, desarrollo y modelado de los sistemas de las organizaciones.

En este trabajo trato de abarcar aspectos importantes de la administración de bases de datos por supuesto no abarco todos los temas que atañen a un administrador de bases de datos pero trato de que el lector conozca y entienda lo que es la administración de bases de datos y su aplicación en Oracle.

En la primera parte menciono conceptos básicos que se deben de tener en cuenta en el trabajo de administración de bases de datos haciendo énfasis en las actividades que realiza el administrador. Así como también incluyo algunos conceptos útiles para la lectura posterior a través de este trabajo y de la inicialización de la base de datos.

Posteriormente hablo sobre el almacenamiento sus aspectos por la parte física y lógica, sus estructuras y algunos otros temas relacionados. Menciono también la función del diccionario de datos.

Posteriormente hablo de la seguridad y aspectos relacionados con ella. Tener cuidado de la seguridad es una de las tareas mas importantes del administrador de bases de datos. En esta parte incluyo algunos conceptos de manejo de usuarios y privilegios así como la recuperación y respaldo de la información.

Durante todo el trabajo aunque es de administración de bases de datos hago mucho hincapié en las actividades del administrador de bases de datos. Esto es obvio, ya que estoy hablando del actor principal de este tema.

Podría decirse que el administrador de base de datos es el líder que coordina el manejo de la información en las organizaciones. Por esto, tiene que ser una persona responsable y consiente de su trabajo.

El administrador de bases de datos se encarga de autorizar el acceso a la base de datos, de coordinar y vigilar su empleo, y de adquirir los recursos necesarios de hardware y software.

El administrador de bases de datos es la persona responsable cuando surgen problemas como violaciones a la seguridad o respuesta lenta del sistema.

Además del administrador de bases de datos existen algunos otros tipos de usuarios de un sistema de bases de datos como son:

- Diseñadores de bases de datos, los cuales se encargan de identificar los datos que se guardaran en las bases de datos.
- Usuarios finales que necesitan tener acceso a la base de datos para hacer consultas, actualizar información y generar reportes.

- Analistas de sistemas y programadores de aplicaciones, estos determinan los requerimientos de los usuarios finales y determinan que transacciones van a satisfacer estos requerimientos. Estos requerimientos los implementan en forma de programas. Conociendo la gama de capacidades del sistema manejador de bases de datos.

Es necesario saber que los objetos son representaciones del mundo real en software. Las bases de datos Oracle están compuestas de muchos objetos que son los que sirven para almacenar o manejar la información.

Como objetos de la base de datos tenemos a las tablas, índices, procedimientos, constraints, triggers, etc. Los cuales siempre le pertenecen a un usuario, el cual, tiene siempre todos los derechos sobre ellos.

El objetivo general de este trabajo es mostrar un poco de lo que en Oracle hay para hacer por parte de un administrador de bases de datos.

1 CONCEPTOS BÁSICOS DE ADMINISTRACIÓN DE BASES DE DATOS.

El Administrador de Bases de Datos (DBA) contribuye al funcionamiento efectivo de todos los sistemas de la base de datos. Tiene el control central sobre el sistema de base de datos proporcionando soporte a todos los usuarios, resolviendo los problemas técnicos que surgen con el software de la base de datos.

Entre otras las actividades de las que es responsable el DBA son:

- Monitorear las operaciones diarias de la base de datos. Por ejemplo revisar: cuanto se ha almacenado en espacio de disco, la velocidad de acceso, que el servicio se otorgue de manera correcta, etc.
- Realizar las instalaciones y actualizaciones del software de la base de datos. Ejemplo: Instalar el software del servidor de base de datos, instalar el software para las aplicaciones front-end, y el software de conexión de los clientes con la base de datos. Además actualizar las versiones del software de Oracle, crear la base de datos y abrirla para su uso normal, etc.
- Planear la creación de las bases de datos. Para esto se realizan múltiples actividades como son: Asesorar en el diseño, definir la seguridad, diseñar la base de datos, crear los objetos necesarios en la base de datos, llevar un registro de los scripts que se corren para crear objetos y tablas, calcular el tamaño físico de los datos, revisar de que manera Oracle y sus aplicaciones utilizan de la mejor manera los recursos de hardware, investigar cuantos discos duros están disponibles para las bases de datos y cuantos dispositivos de respaldo se tienen, definir estrategias de respaldo, etc.
- Administrar el almacenamiento físico y lógico. Como ejemplo de estas funciones tenemos: Definir en que unidades lógicas se va almacenar y de que tamaño van a ser estas, definir la estructura lógica del almacenamiento y revisar como la estructura lógica

del almacenamiento va a afectar el desempeño del sistema y las operaciones de manejo de la base de datos. Para esto se debe prever analizando: el desempeño de la computadora ejecutando Oracle, el desempeño durante las operaciones de acceso a los datos y la eficiencia de los procedimientos de respaldo y recuperación.

- Modificar la estructura de la base de datos. Como es: Autorizar modificaciones en la definición de la base de datos solicitadas por los desarrolladores. Además en el momento que ya se creó la base de datos se deben de crear los segmentos de rollback necesarios dependiendo de las operaciones que se estén realizando.
- Manejo de usuarios de la base de datos. Ejemplo: Creación, control, asignación de roles.
- Controlar la seguridad. Ejemplo: Otorgar derechos a usuarios, definir estrategias de recuperación y respaldo.
- Realización de copias de seguridad y recuperación. Ejemplo: Planear y realizar los procedimientos de respaldo y recuperación, elaborar planes de contingencia. En este rubro el DBA debe implementar las estrategias de respaldo y recuperación que planeó con anterioridad, estableciendo un control sobre estas actividades estableciendo calendarios y procedimientos.
- Colaborar con los desarrolladores. Para Estandarizar y dar orden a la base de datos.
- Ajuste del rendimiento. Realizando acciones que mejoren la velocidad de respuesta, rendimiento y eficiencia de la base de datos.

En general el DBA controla las actividades que son más críticas en la base de datos, para esto necesita todos los privilegios del sistema manejador, lo que no se otorga a ningún otro usuario, además debe contar con privilegios en el sistema operativo suficientes para asegurar el buen funcionamiento de la base de datos.

Para realizar muchas de las tareas de la administración en la base de datos el DBA necesita ejecutar comandos del sistema operativo. Para esto se necesita que el DBA tenga una

cuenta de sistema operativo que le permita acceder a los archivos del software del manejador de base de datos.

Hablando de manera practica el DBA es el encargado de las tareas mas técnicas y criticas en la base de datos. Por ejemplo algunos de los casos en los que tiene que actuar el DBA son:

- Recuperar información cuando hay perdida de datos. Digamos que se descompone uno de los discos duros del servidor por lo cual se pierden datos. En este caso se utilizan los respaldos para recuperar la información que se tenia hasta cierto momento del tiempo.
- Reiniciar la base de datos cuando ha ocurrido un error. Llega a suceder que cuando se manejan grandes volúmenes de información y se realizan actualizaciones muy complejas o mal planeadas la base de datos no aguanta y se bloquea por lo que el DBA tiene que reiniciar la base de datos para que los procesos que están bloqueando el sistema se terminen y pueda seguir funcionando la base de datos.
- Controlar y en algunos casos realizar la creación de objetos de la base de datos para verificar que se cumpla con el diseño que se realizó, que no afecte el rendimiento de la base de datos y evitar objetos basura en la base de datos. Algunas veces los desarrolladores al hacer pruebas o realizar procesos poco comunes en la base de datos crean tablas, índices, constraints, y demás objetos de manera temporal los cuales muchas veces olvidan borrar generando basura que gasta espacio en disco. El DBA debe definir estrategias para eliminar estos datos y objetos basura.
- Realizar los respaldos dependiendo de las políticas de la organización. A veces la información es muy cambiantes y se tiene que hacer muchos respaldos en otros casos la información es casi estática por lo que los respaldos no son necesarios con tanta frecuencia, para esto se tiene que hablar con los dueños de la información para definir con que frecuencia respaldar.

- Corregir fallas en el servidor. A veces por una mala instalación hay parámetros de sistema operativo o de la base de datos que no se definen bien lo cual puede ocasionar mal funcionamiento. El DBA debe corregir estos errores.
- Contactar al soporte técnico del manejador de base de datos si es necesario. En el caso de Oracle en el caso que se presenten errores que salgan del conocimiento común del DBA este debe contactar a los servicios de soporte técnico con que cuenta la compañía que produce y vende el manejador.
- Estar en contacto con los coordinadores y directores para determinar a quienes se les otorga derechos y que derechos se les otorga. Esto es por que a la base de datos acezan muchos tipos de usuarios los cuales piden se les de acceso a la información de manera inmediata pero a veces es solo un director de área el que puede autorizar que se le de acceso y derechos a ciertos usuarios. El DBA en algunos casos tiene que agilizar estas autorizaciones para dar un servicio mas rápido a los usuarios.
- Dialogar con los desarrolladores para determinar el tamaño de almacenamiento requerido para albergar la información de los diferentes sistemas, realizando con esto a su vez la definición de los espacios físicos y lógicos que se necesiten.
- Leer la documentación de la base de datos para encontrar mejoras en el rendimiento, ya que muchas veces los errores se dan por no leer a fondo la documentación necesaria.
- Reunirse con los coordinadores de áreas, proyectos y desarrolladores para definir políticas y estándares. Como son nombres de objetos, manera de otorgar derechos, procedimientos de respaldo, procedimientos de seguridad preventivos y correctivos, etc.

1.1 Conceptos Iniciales.

En el caso de Oracle una instancia es un conjunto de procesos de soporte y áreas de memoria asignados para acceder a una base de datos.

Al inicializar o arrancar (startup) la base de datos usualmente se hace sin montar o subir la base de datos esto se hace durante la creación de la base de datos.

Después de arrancar se puede montar (mount) la base de datos para realizar actividades específicas de mantenimiento que revisaremos en su momento. En este momento no esta aun abierta la base de datos.

Cuando una base de datos esta abierta (open) esta en operación normal lo que significa que esta inicializada, la base de datos esta montada y además esta abierta. Al abrir se pone la instancia en un estado que permita a lo usuarios acceder a la base de datos para llevar a cabo su trabajo normal sobre esta realizando operaciones de acceso a los datos.

Una base de datos esta formada por una o mas unidades lógicas de almacenamiento llamadas tablespaces.

Oracle almacena lógicamente los datos en tablespaces y físicamente en archivos de datos asociados con su correspondiente tablespace. En pocas palabras podemos decir que un tablespace contiene uno o más archivos de datos. los archivos de datos son estructuras físicas conformes al sistema operativo que se este utilizando y en las cuales el servidor Oracle corre. Estos contiene toda la información de la base de datos.

Por otro lado lógicamente están divididos los tablespaces en segmentos y a su vez los segmentos están divididos en extents. Todas estas son estructuras lógicas de almacenamiento.

También tenemos los segmentos de rollback que son usados por una transacción que esta realizando cambios en la base de datos. Antes de cambiar los datos o los índices, el viejo valor es guardado en el segmento de rollback. Esto permite al usuario deshacer los cambios hechos.

Todas las bases de datos Oracle tienen un tablespace llamado System el cual se define al crear la base de datos y contiene el diccionario de datos y las unidades de programación como son los procedimientos, los packages o paquetes y los triggers.

Los clusters o grupos son un método opcional para almacenar datos de una tabla. Un cluster es un grupo de tablas que comparte el mismo bloque de datos porque ellas comparten columnas comunes y son frecuentemente usadas juntas.

El diccionario de datos es un conjunto de tablas y vistas que son usadas como una referencia que provee información acerca de la base de datos asociada. Las tablas base del diccionario de datos son creadas durante la creación de la base de datos. Digamos que este diccionario es la principal fuente de información para el servidor, el administrador y los usuarios de la base de datos. Este diccionario de datos es modificado por el servidor Oracle cuando un comando de definición de datos es ejecutado.

El archivo de control es un archivo binario que es necesario para arrancar la base de datos y que la operación de la misma sea exitosa. Este le sirve a la base de datos para localizar los archivos de datos y los archivos redo log. Este archivo de control es actualizado

constantemente durante la operación de la base de datos y debe estar disponible en todo momento. Este archivo también provee información acerca de la consistencia de la base de datos. Si uno de los archivos de control no están disponibles, la base de datos puede no funcionar apropiadamente.

El servidor Oracle mantiene los archivos redo log en línea para minimizar la pérdida de datos en la base de datos. Estos archivos registran todos los cambios hechos en los datos en el buffer cache de la base de datos con algunas excepciones como en el caso de escrituras directas. Los archivos Redo Log son usados en situaciones como en la falla de una instancia para recobrar los datos comprometidos que no han sido escritos en los archivos de datos. Digamos que estos archivos son solo usados para recuperación.

Oracle cuenta con algunas herramientas para administración de bases de datos. Con una interfaz de línea de comandos se tiene el Server Manager que nos permite realizar tareas de administración utilizando comandos. Por otro lado se tiene el programa Oracle Enterprise Manager (OEM) que tiene una interfaz gráfica. Los dos realizan casi las mismas funciones como son: Crear base de datos, arrancar instancias, crear usuarios, dar derechos etc.

1.1.1 Instalación Del Software De La Base De Datos:

La instalación del software de la base de datos es una de las tareas del DBA, la cual varía según el manejador que se quiera instalar, sistema operativo que utilice y la versión del manejador que este instalando.

Para darnos una idea de esta tarea por ejemplo en el caso de Oracle para NT la instalación del software es un proceso parecido a la instalación de cualquier paquete de oficina. Para esto Oracle proporciona un programa instalador universal que sirve para instalar de manera visual y sencilla el software y dentro del mismo proceso crear una base de datos default. Esta tarea es realizada por el DBA ya que es crítica en el funcionamiento de la base de datos.

1.2 Planeación De La Base De Datos.

El DBA tiene la responsabilidad de planear:

- La estructura lógica del almacenamiento.
- El diseño de la base de datos.
- Las estrategias de respaldo de la base de datos.

Es importante planear como la estructura lógica del almacenamiento va a afectar el desempeño del sistema y las operaciones de manejo de la base de datos. Por ejemplo, antes de crear cualquier tablespace para la base de datos, se debe saber cuantos archivos de datos van a formar el tablespace, en que discos duros van a estar almacenados físicamente los archivos de datos y que tipo de información va a ser almacenada en cada tablespace. En Oracle cuando se planean todas las estructuras lógicas de almacenamiento de la base de datos se debe tomar en cuenta los efectos que esta estructura tendrá cuando la base de datos este realmente creada y corriendo. Para esto se deben considerar ciertos puntos para saber que tanto afectara a la base de datos la estructura lógica del almacenamiento. Estos rubros son:

- El rendimiento de la computadora ejecutando Oracle.
- El rendimiento de la base de datos durante las operaciones de acceso a los datos.
- La eficiencia de los procedimientos de respaldo y recuperación de la base de datos.

También se tiene que planear el diseño relacional de los objetos de la base de datos y las características de almacenamiento para cada uno de esos objetos. Planeando las relaciones entre los objetos y el almacenamiento físico de cada objeto antes de crearlo, se puede afectar directamente el rendimiento de la base de datos como una unidad, por esto es que se tiene que planear en función del crecimiento de la base de datos.

En el caso de ambientes distribuidos, esta planeación es extremadamente importante, ya que la localización física de datos de acceso frecuente, pueden afectar dramáticamente el desempeño de la aplicación.

También se tiene que planear estrategias de respaldo. Después de esto se tiene que encontrar que se quiere cambiar en la estructura lógica del almacenamiento que se planeo con anterioridad o en el diseño de la base de datos para mejorar la eficiencia de las operaciones de respaldo.

Para esto el DBA debe haberse preparado bastante en cuestiones de diseño antes de iniciarse en las actividades de administración de la base de datos.

1.3 Creación de la base de datos.

Antes de crear una base de datos se deben de tener algunas consideraciones como son: Los prerrequisitos de creación, si se va a crear una base de datos nueva, o si se va a migrar una versión anterior de la base de datos. La creación de la base de datos prepara varios archivos de sistema operativo para que puedan trabajar juntos como una base de datos.

Crear un base de datos en Oracle incluye las operaciones:

- Crear nuevos archivos de datos o borrar datos que existían en anteriores archivos de datos.
- Crear estructuras de información que Oracle requiere para acceder y usar la base de datos como lo es el diccionario de datos.
- Crear e inicializar los archivos de control y los archivos redo log de recuperación para la base de datos.

Antes de crear la base de datos se debe planear la base de datos que es una de las tareas más importantes del DBA en el caso de la creación de la base de datos se deben considerar los siguientes puntos antes de crear una base de datos:

- Planear sus tablas e índices y estimar cuanto espacio requerirán.
- También se tiene que planear como se va a proteger la nueva base de datos, incluyendo la configuración de sus archivos redo log en línea, el espacio que estos requieren y definir estrategias de respaldo.
- Se tiene que seleccionar un conjunto de caracteres para la base de datos. Después de que ya esta creada la base de datos no se puede cambiar el conjunto de caracteres. Para esto se debe pensar detenidamente que conjunto de caracteres se va a utilizar antes de crear la base de datos. esto es importante ya que todos los datos caracteres incluyendo los datos en el diccionario de datos están guardados en el conjunto de caracteres de la base de datos, además los usuarios van a utilizar este conjunto cuando accesen.

1.3.1 Prerequisitos De Creación.

Para crear la base de datos se requiere:

Los privilegios del sistema operativo asociados con la completa operación del administrador de base de datos.

Suficiente memoria para iniciar la instancia de Oracle.

Suficiente espacio en disco para la base de datos que se planea tener y además para que se pueda ejecutar Oracle.

Por otro lado se tiene que tener en cuenta si se esta creando una base de datos desde cero o si se va a migrar datos a una nueva base de datos.

Al usar una base de datos nueva dependiendo de su sistema operativo una base de datos debe ser creada automáticamente como parte del procedimiento de instalación para Oracle. Se puede utilizar esta base de datos inicial arreglándola conociendo sus requerimientos de manejo de información. Pero también puede olvidarse de esta y crear una o más bases de datos nuevas que reemplacen la que se creo por default.

Si se migra una versión anterior de la base de datos se tiene que considerar si se han estado utilizando versiones anteriores de Oracle, la creación de la base de datos es solo necesaria si se quiere enteramente una nueva base de datos. En el otro caso se puede migrar las bases de datos existentes manejadas por una versión anterior de Oracle y usando en ellas las nueva versión.

1.3.2 Pasos Para La Creación De La Base De Datos.

Estos pasos deberán ser seguidos en el orden que se presentan para poder crear la base de datos y dejarla disponible para el uso del sistema:

1. Respalidar las base de datos existentes.
2. Crear los archivos de parámetros.
3. Editar los nuevos archivos de parámetros.
4. Checar el identificador de instancia de su sistema.
5. Arrancar el SQL*Plus y conectarse a Oracle como SYSDBA.
6. Arrancar una instancia.
7. Crear la base de datos.
8. Respalidar la base de datos.

Estos pasos son para todos los sistemas operativos pero es importante que se revise la documentación específica de cada plataforma para conocer detalles especiales.

1. Respalidar cualquier base de datos existente. Se recomienda mucho que se respalden completamente las bases de datos existentes antes de crear una nueva , en caso de que la creación accidentalmente afecte archivos existentes. Este respaldo debe incluir archivos de parámetros, datos, de redo log y de control.
2. Crear los archivos de parámetros. La instancia para cualquier base de datos Oracle es iniciada usando un archivo de parámetros. Cada base de datos en el sistema debe tener al menos un archivo de parámetros hecho a la medida que corresponda solo a esa base de datos. No se debe de usar el mismo para varias bases de datos. Para crear el archivo de parámetros se debe hacer desde el sistema operativo para hacer una copia del archivo de parametros que se provee con el software de Oracle. Cuando esto se realice se debe

dar un nombre nuevo a esta copia del archivo, por último se tiene que editar este archivo para la nueva base de datos.

3. Editar el nuevo archivo de parámetros. Para crear una nueva base de datos debe inspeccionar y editar los parámetros del nuevo archivo de parámetros.
4. Revisar el indicador de instancia de su sistema. Si se tiene otras bases de datos hay que revisar el identificador de la instancia, el cual debe corresponder con el nombre de la base de datos para prevenir confusiones con el nombre de las instancias de otras bases de datos que estén corriendo concurrentemente en el sistema.
5. Arrancar SQL*Plus y conectarse a Oracle como SYSDBA. En este paso simplemente hay que entrar al prompt de SQL*Plus y teclear la instrucción:
connect username/password as sysdba
6. Inicializar la instancia. Se puede inicializar una instancia sin montar la base de datos, normalmente, se hace esto solo durante la creación de la base de datos. Para esto se usa la instrucción STARTUP con la opción NOMOUNT.
Hasta este momento no hay todavía una base de datos solo están inicializados y preparados para la creación de una nueva base de datos un área global de sistema y procesos que corren en background.
7. Crear la base de datos. Para crear una nueva base de datos se usa la instrucción de SQL CREATE DATABASE, opcionalmente poniendo como parámetro, el nombre de la base de datos, el máximo número de archivos, sus nombres y sus tamaños.

Cuando se ejecuta la instrucción Create Database, Oracle realiza las siguientes operaciones:

- a) Crear los archivos de datos.
- b) Crear los archivos de control.

- c) Crear los archivos redo log de recuperación.
- d) Crea el Tablespace SYSTEM y el segmento de rollback SYSTEM.
- e) Crea el diccionario de datos.
- f) Crea los usuarios SYS y SYSTEM.
- g) Especifica el conjunto de caracteres que almacena datos en la base de datos.
- h) Monta y abre la base de datos para su uso.

Hay que estar seguro que los nombres dados a los archivos de datos y redo log no tengan conflicto con archivos de otra base de datos.

8. Respalda la base de datos. Se tiene que hacer un respaldo completo de la base de datos para asegurarnos que usted tiene un conjunto de archivos completo del cual va a recobrar si una falla en el medio ocurre.

1.3.3 Parámetros de la Base de Datos.

Estos parámetros son los que se especifican en el archivo de parámetros a continuación explicaremos algunos de ellos, como son:

- CONTROL_FILES
- DB_BLOCK_SIZE
- PROCESSES
- ROLLBACK_SEGMENTS
- DB_BLOCK_BUFFERS

- IFILE
- BD_FILES
- SHARED_POOL_SIZE
- DML_LOCKS
- LOG_BUFFER
- GLOBAL_NAMES

Para observarlos de manera practica a continuación se muestra un ejemplo de un archivo de parámetros para la base de datos de una empresa cualquiera:

La localización de este archivo de parámetros es dependiente del sistema operativo que se utilice. A manera de ejemplo en el caso de UNIX se encuentra en la ruta:

```
/export/home/oracle/app/oracle/admin/server1/pfile/
```

Aunque esto puede variar en algunas ocasiones. Ahora veremos el ejemplo del archivo de parámetros:

ifile=/export/home/oracle/app/oracle/admin/server1/pfile/config1.ora

rollback_segments = (r01,r02,r03,r04)

tuning parameters

db_files = 80

db_block_buffers = 200

shared_pool_size = 3500000

processes = 50

dml_locks = 100

log_buffer = 8192

global_names = TRUE

CONTROL_FILES.- Este parámetro se incluye como un conjunto de valores como una lista de nombres de archivos de control. Si se quiere que Oracle cree los nuevos archivos de sistema operativo cuando cree los archivos de control, para esto se debe estar seguro que el nombre de los archivos de control no concuerde con cualquier nombre de archivo que exista actualmente en nuestro sistema. Si se quieren usar los mismos simplemente se deja el nombre igual a como están en el sistema operativo.

Si no se especifican nombres para los archivos de control, Oracle utiliza nombres por default.

DB_BLOCK_SIZE.- Con este parámetros se especifica el tamaño de los bloques de datos. Este tamaño es específico de cada sistema operativo. El tamaño típico es de 2 o 4 Kb. En algunos casos un tamaño de bloque grande proporciona mas eficiencia en la escritura del disco y la memoria.

DB_BLOCK_BUFFERS .- Este parámetro determina el numero de buffers del buffer cache de el área global del sistema. Este numero de buffers afecta el desempeño de la cache. Mientras se tenga una cache más grande se reduce él numero de escrituras y modificaciones a disco. Pero una muy grande cache induce a tomar mucha memoria.

Se necesita estimar él numero de bloques de datos que nuestra base de datos accesa mas frecuentemente incluyendo tablas índices y segmentos de rollback. Este estimado es una aproximación del mínimo numero de buffers que el cache debe tener. En algunos casos 1000 o 2000 es un mínimo practico.

PROCESSES.- Este parámetro especifica él numero de procesos del sistema operativo que pueden ser conectados a Oracle actualmente. El valor de este parámetros debe incluir 5 para los procesos background y uno para cada proceso de usuario. Aunque también incluyen los procesos que corren solos para saber cuantos son los indicados hay que consultar la documentación específica de Oracle para su sistema operativo.

ROLLBACK_SEGMENTS.- En este parámetro se especifica una lista de los segmentos de rollback que una instancia de la base de datos adquiere en el arranque de la base de datos.

IFILE.- Especifica si se va a utilizar otro archivo de parámetros además del que manda llamar la instrucción.

BD_FILES.- Especifica el número máximo de archivos de la base de datos que pueden ser abiertos al mismo tiempo, para esta base de datos. El número máximo depende del sistema operativo para esto tendrá que consultar la documentación específica de su sistema operativo. Si el tamaño máximo de data files es muy chico no se pueden adicionar más datafiles durante la operación de la base de datos sin darla de baja, si este número es muy alto se consume memoria innecesariamente. Lo ideal es que al ser un número chico aumenta el desempeño de la base de datos.

SHARED_POOL_SIZE.- Especifica el tamaño en bytes del área de memoria compartida conteniendo cursores compartidos y procedimientos almacenados. Se recomienda que si es un sistema multiusuario se utilice lo más que se pueda para mejorar el desempeño ya que en el shared pool se guardan los cursores compartidos, procedimientos almacenados, y estructuras de control mientras estas son utilizadas.

DML_LOCKS.- Especifican el número máximo de bloqueos, uno por cada tabla modificada. En cada transacción el valor deberá ser igual al total máximo de bloqueos en tablas que actualmente están siendo usadas por instrucciones DML de los usuarios.

Por ejemplo si 3 usuarios están modificando datos en una tabla entonces un valor de 3 sería suficiente.

LOG_BUFFER .- Especifica la cantidad de memoria en bytes que es usada para guardar los redo log, justo antes de guardarlos en los archivos redo log. Mientras más grande sea esta se realizarán menos escrituras a disco. Pero su tamaño también es limitado por el espacio en

memoria. Esta cantidad también depende del sistema operativo. Para lo cual hay que consultar la documentación específica.

GLOBAL_NAMES .- Especifican si una liga a la base de datos es requerida para tener el mismo nombre de la base de datos a la cual va a ser conectada.

1.3.4 Ejemplo De Creación De Una Base De Datos.

A continuación mostrare un ejemplo de la instrucción CREATE DATABASE:

```
CREATE DATABASE prueba
  DATAFILE 'pru_system' SIZE 10M
  LOGFILE GROUP 1 ('pru_log1a', 'pru_log1b') SIZE 500K,
  GROUP 2 ('pru_log2a', 'pru_log2b') SIZE 500K;
```

Los valores de las opciones MAXLOGFILES, MAXLOGMEMBERS, MAXDATAFILES, MAXLOGHISTORY, y MAXINSTANCES asumen sus valores default, los cuales dependen del sistema operativo. la base de datos es montada en el modo default NOARCHIVELOG y EXCLUSIVE y luego es abierta.

La información contenida en la instrucción del ejemplo anterior crean una base de datos con las siguientes características:

Se crea una base de datos llamada PRUEBA:

El tablespace SYSTEM de la nueva base de datos consta de un archivo de datos o datafile de 10MB llamado PRU_SYSTEM.

La nueva base de datos tiene dos grupos redo log en línea, cada uno conteniendo dos partes de 500KB.

La nueva base de datos no sobre escribirá ningún archivo de control existente especificado en el archivo de parámetros.

1.3.5 Consideraciones Después De Crear Una Base De Datos.

Después de que usted crea una base de datos, la instancia queda corriendo y la base de datos esta abierta y disponible para su uso normal. Si una o más bases de datos existen en nuestro sistema de base de datos, hay que especificar en el archivo de parámetros que se usara con alguna base de datos corriendo al mismo tiempo.

La nueva base de datos tiene solo dos usuarios, SYS y SYSTEM. Los passwords para estos nombres de usuario deben ser cambiados después de que la base de datos fue creada.

1.3.6 Arrancar La Base De Datos.

Ya creamos la base de datos ahora es el momento de inicializar la instancia para que la base de datos quede montada en esta y abierta para su operación.

Para esto se ejecuta el server manager y se introduce la orden `CONNECT INTERNAL/Oracle` para poder conectarnos a la base de datos lo que es necesario para poder arrancar o inicializar la instancia. Se debe conectar a una base de datos con la contraseña que es la palabra después de la diagonal. Solo así podrá arrancar la instancia.

Después para iniciar teclee la instrucción `STARTUP`.

A partir de este momento se ha arrancado la base de datos, también se puede decir que esta abierta o activada después de haber realizado con éxito la instrucción. Pero arrancar la base de datos con Server Manager es en realidad un proceso de 3 pasos, o sea que `STARTUP` tiene tres instrucciones dentro de si, las cuales son:

```
STARTUP NOMOUNT
ALTER DATABASE MOUNT
ALTER DATABASE OPEN
```

Cuando la base de datos se arranca pero existió algún error en alguno de los pasos del proceso o sea que no se realizo con éxito la instrucción. En este caso la base de datos permanece en el ultimo estado que se realizo con éxito.

1.3.7 Opciones De Arranque.

1.3.7.1 Arranque normal.

Para este caso se utiliza la instrucción: `STARTUP NORMAL`, normalmente se omite la palabra normal ya que si esta se tiene el mismo efecto.. Digamos que este es el modo de arranque predeterminado donde se arranca la instancia, se monta la base de datos y se abre quedando lista para su operación normal.

1.3.7.2 Arranque con montaje.

Este modo se utiliza para cambiar el estado de archivado de la base de datos, para realizar una recuperación, renombrar archivos de datos, adicionar, eliminar o renombre archivos redo log, activar o desactivar las opciones de los archivos redo log, etc.

La instrucción es: `STARTUP MOUNT` después de esto la base de datos ha quedado en estado montado y los usuarios no pueden iniciar sesión.

1.3.7.3 Arranque sin montaje.

Este modo se usa para crear de nuevo un archivo de control, o para volver a crear la base de datos partiendo de cero. Esta opción se utiliza generalmente cuando se esta creando la base de datos. La base de datos no esta abierta por lo que no pueden acceder los usuarios. La instrucción para esta caso es:

```
STARTUP NOMOUNT
```

1.3.7.4 Arranque restringido.

Este modo se usa para arrancar la base de datos pero su acceso se restringe a un conjunto privilegiado de usuarios que se haya definido. La base de datos queda arrancada, montada y abierta pero solo para los usuarios privilegiados o DBA's. Este modo esta hecho para realizar operaciones como: Realizar mantenimiento de la estructura como reconstruir índices, realizar importación y exportación de datos, para cargar datos con el SQL Loader, y para impedir temporalmente que usuarios típicos usen los datos.

La instrucción es: `STARTUP RESTRICT`

1.3.7.5 Arranque forzado.

Esta opción se utiliza raras ocasiones, cuando no se es capaz de cerrar una base de datos. Esta opción cierra la base de datos con un shutdown abort (opción que veremos mas adelante) y luego la arranca sin opciones, realizando al arrancar la misma función que startup normal.

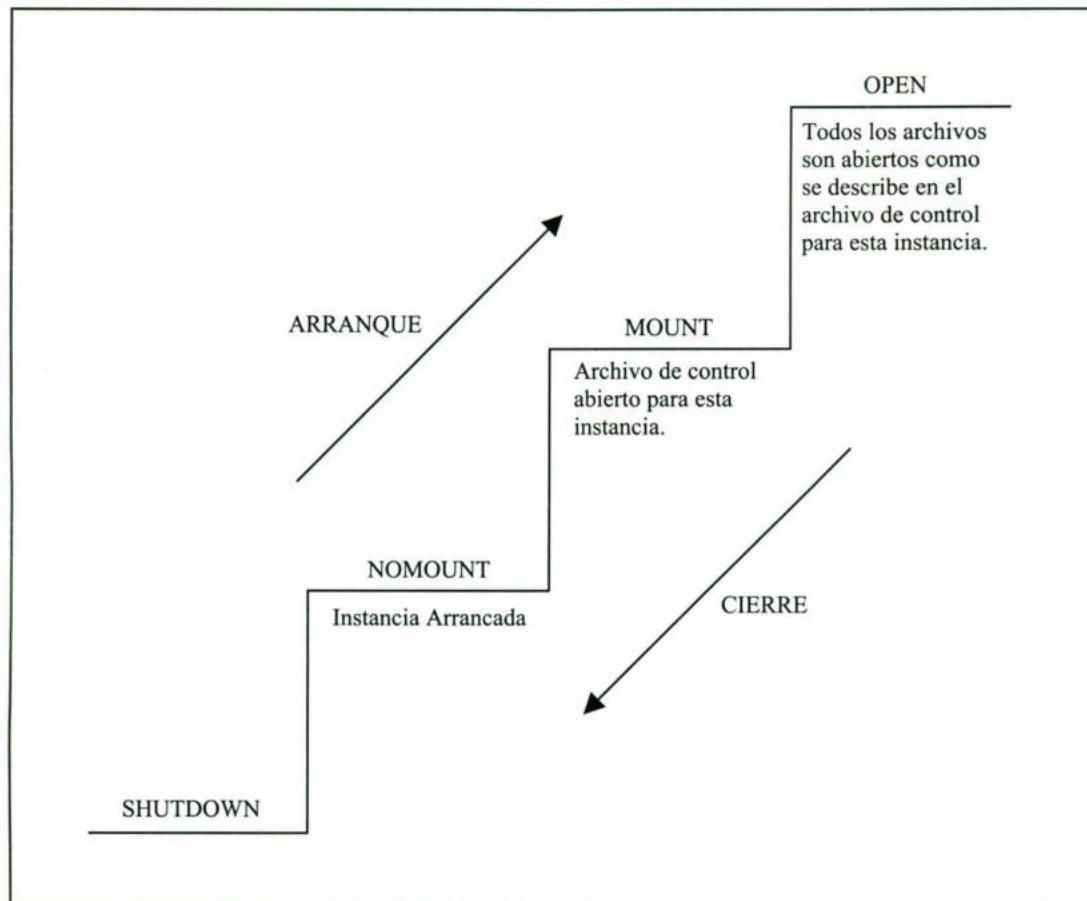


Fig. 1.1 Arranque de la Base de Datos.

1.3.8 Cerrar la base de datos.

Para cerrar la base de datos en condiciones normales se tiene la instrucción `SHUTDOWN NORMAL` aunque el utilizar solo la instrucción `SHUTDOWN` tiene el mismo efecto. Esta instrucción cierra la base de datos, la desmonta y apaga la instancia.

El cierre normal de la base de datos se da bajo las siguientes condiciones:

- No se permiten nuevas conexiones después de que se ejecuta la instrucción.
- Oracle espera a que todos los usuarios conectados se desconecten.

1.3.8.1 Shutdown Immediate.

Esta opción se usa en caso de:

Que un apagón de energía eléctrica esta por ocurrir.

Que la base de datos o una de sus aplicaciones no estén funcionando correctamente.

Con esta opción se dan las siguientes condiciones:

Cualquier transacción no comprometida es abortada con rollback.

Oracle no espera a que los usuarios se desconecten da rollback a sus transacciones y los desconecta.

La instrucción es: SHUTDOWN IMMEDIATE.

1.3.8.2 Shutdown Transactional.

Cuando se desea hacer un shutdown planeado de una instancia permitiendo que terminen las transacciones que están en proceso. Después de ejecutar la instrucción ningún usuario puede iniciar una nueva transacción en esa instancia, si se intenta iniciar una nueva transacción los usuarios son desconectados automáticamente. Después de terminar con todas las transacciones los usuarios son desconectados. La instrucción en este caso es:

SHUTDOWN TRANSACTIONAL

1.3.8.3 Shutdown Abort.

Se puede cerrar la base de datos instantáneamente abortando la instancia de la base de datos. Esta opción se utiliza en las siguientes situaciones:

- La base de datos o una de sus aplicaciones están funcionando irregularmente y no funciona ningún otro tipo de shutdown.
- Cuando se viene una falta de energía eléctrica en escaso un minuto.
- Cuando se tiene problemas al arrancar una instancia de la base de datos.

Esta opción arroja los siguientes resultados:

- Las instrucciones cliente de SQL que estaban siendo procesadas en ese momento son terminadas inmediatamente.
- Las transacciones no comprometidas se pierden sin ejecutar un rollback.
- Implícitamente desconecta todos los usuarios conectados.

Si no funcionan las opciones normal e inmediata se debe abortar la base de datos. La instrucción en este caso es:

SHUTDOWN ABORT

Opciones de la Instrucción Shutdown

	Abort	Inmediate	Transaccional	Normal
PERMITE NUEVAS CONEXIONES.	No	No	No	No
Espera hasta que las sesiones actuales terminen.	No	No	No	Si
Espera hasta que las transacciones actuales terminen.	No	No	Si	Si
Forza un checkpoint y cierra los archivos.	No	Si	Si	Si

Tabla 1.1 Opciones de La Instrucción Shutdown.

2. ADMINISTRACIÓN DE LAS ESTRUCTURAS DE ALMACENAMIENTO Y DICCIONARIOS DE DATOS.

2.1 Administración De Las Estructuras De Almacenamiento.

En esta sección analizaremos las estructuras de almacenamiento con las que cuentan las base de datos Oracle. Inicialmente se tiene estructuras físicas y lógicas.

2.1.1 Estructuras Lógicas De La Base De Datos.

Las estructuras lógicas de la base de datos incluyen Tablespaces (Espacios de tablas), schema objects (objetos de esquema), blocks (bloques de datos), extents (extensiones de datos) y segments (segmentos).

2.1.1.1 Tablespaces.

Una base de datos esta dividida en unidades lógicas de almacenamiento llamadas tablespaces, las cuales agrupan estructuras lógicas relacionadas.

La relación entre bases de datos, tablespaces y archivos de datos se muestra en la siguiente figura:

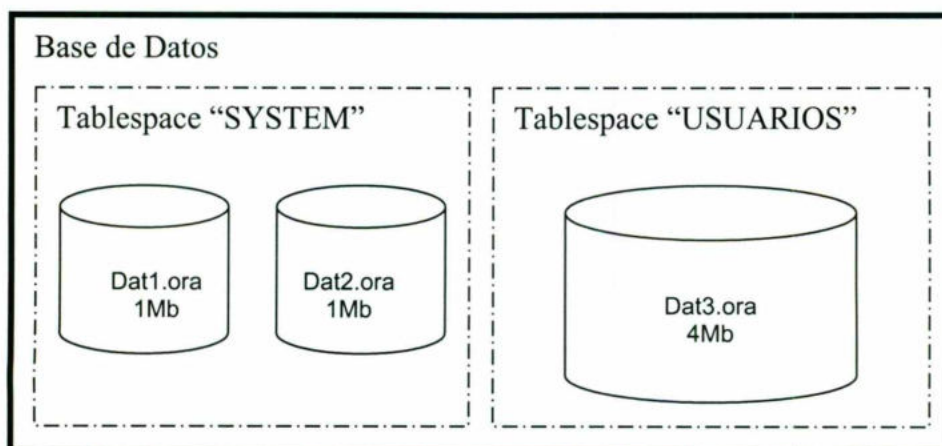


Figura 2.1 Relación entre bases de datos, tablespaces, y archivos de datos (Datafiles).

Esta figura nos muestra que cada base de datos está dividida en uno o más tablespaces. Por otro lado uno o más archivos de datos son creados explícitamente para cada tablespace para almacenar físicamente los datos de todas las estructuras lógicas en un tablespace.

El tamaño combinado de los archivos de datos de un tablespace es la capacidad total de almacenamiento del tablespace. Por ejemplo en el diagrama el tablespace "System" tiene 2 Mb de espacio en dos archivos de datos, el tablespace "Usuarios" tiene 4 Mb en un archivo de datos, la capacidad combinada de los tablespaces es el total de la capacidad de almacenamiento de la base de datos.

Un tablespace puede estar online (En línea o accesible) o offline (No accesible o fuera de línea). Un tablespace está normalmente en línea con lo cual sus usuarios pueden acceder a la información dentro de él. Sin embargo algunas veces un tablespace puede estar offline produciendo que una porción de la base de datos no esté disponible mientras se permite el

acceso normal al resto de la base de datos. Esto hace que muchas tareas administrativas se realicen más fácilmente.

Una base de datos consiste en al menos un tablespace llamado SYSTEM. Se puede decir que este es un tipo de tablespace, el cual se requiere en todas las bases de datos para su operación. Este tiene las siguientes características:

- Contiene la información del diccionario de datos, la definición de los procedimientos almacenados, packages y triggers.
- Contiene el segmento de rollback SYSTEM.
- En teoría no debe contener datos de usuario pero esto tiene que ser especificado.

También los tablespace que no son system los cuales tienen las siguientes características:

- Permiten más flexibilidad en la administración de base de datos.
- Pueden almacenar segmentos de rollback, segmentos temporales, datos de aplicación e índices de aplicación.

2.1.1.2 Manejo de tablespaces.

Antes de crear un tablespace debemos de saber que existen parámetros de almacenamiento por default que se pueden especificar para objetos que serán creados en un tablespace. Esto quiere decir que podemos dar un tamaño por default a estos objetos. Estos parámetros se pueden especificar al crear el tablespace, si no se especifican en este momento los parámetros de almacenamiento toman valores por default.

Para crear un tablespace se utiliza la instrucción CREATE TABLESPACE. Por ejemplo:

```
CREATE TABLESPACE tablespace_2
  DATAFILE 'diska:tablespace_file2.dat' SIZE 20M
  DEFAULT STORAGE (INITIAL 10K NEXT 50K
                  MINEXTENTS 1 MAXEXTENTS 999)
  ONLINE;
```

Esta instrucción crea un tablespace llamado tablespace_2 con un datafile llamado tablespace_file2.dat de 20 Mb de tamaño.

```
CREATE TABLESPACE tablespace_2
  DATAFILE 'diska:tablespace_file2.dat' SIZE 20M
```

Definiendo los parámetros de almacenamiento el tamaño del primer extent va a ser de 10 Kb y el del segundo de 50 Kb.

```
DEFAULT STORAGE (INITIAL 10K NEXT 50K
```

Luego le decimos el numero de extents asignados cuando se crea un segmento que va a ser de 1 y también le decimos el numero maximo de extents que un segmento puede tener.

```
MINEXTENTS 1 MAXEXTENTS 999)
```

Por ultimo indicamos que el tablespace queda para su uso después de creado con:

```
ONLINE;
```

Sintaxis:

```
CREATE TABLESPACE tablespace
    DATAFILE nombre_archivo [clausula_autoextenderse]
    [,nombre_archivo [clausula_autoextenderse]...
    [MINIMUM EXTENT integer [K] [M] ]
    [DEFAULT especificacion_de parametros_de_almacenamiento]
    [PERMANENT | TEMPORARY]
    [ONLINE | OFFLINE]
```

tablespace .- Es el nombre del tablespace.

DATAFILE nombre_archivo.- Especifica el o los datafiles que formaran el tablespace.

clausula_autoextenderse.- Activa o desactiva que los datafiles extiendan su tamaño.
AUTOEXTEND mas ON para activar y OFF para desactivar.

NEXT.- Especifica el espacio en disco máximo permitido para asignar a los datafiles.

UNLIMITED.- Permite que no haya limite en la asignación de espacio al datafile.

MINIMUM EXTENT.- Controla la fragmentación del espacio libre en el tablespace asegurándose que la longitud de cada extent tenga como mínimo esa cantidad y que el esa longitud siempre sea un múltiplo del numero dado.

ONLINE .- Produce que el tablespace este disponible para los usuarios inmediatamente después de creado. Este es por default.

OFFLINE .- Hace que el tablespace no este disponible para los usuarios inmediatamente despues de su creación. En la vista del diccionario de datos `DBA_TABLESPACES` se puede ver si esta offline o online.

PERMANENT.- Especifica que el tablespace será usado para contener objetos permanentes. Esta opción la toma por default.

TEMPORARY .- Especifica que el tablespace será usado para contener objetos temporales. Esta opción la revisaremos en la siguiente sección.

DEFAULT especificacion_de_parametros_de_almacenamiento.- Especifica con la instrucción `DEFAULT STORAGE` los parámetros de almacenamiento por default para los objetos en el tablespace. Sus opciones son:

INITIAL .- Especifica el tamaño en bytes del primer extent de un objeto. Oracle asigna espacio a este extent cuando se crea un objeto del esquema. Se usan la K y M para especificar si son kilobytes o megabytes.

NEXT.- Especifica el tamaño en bytes del siguiente extent que será asignado a un objeto. De nuevo K Y M sirven para especificar el tamaño ya sea kilobytes o megabytes.

PCTINCREASE.- Especifica el porcentaje por el cual crecerán el tercer y subsecuentes extents.

MINEXTENTS.- Especifica el numero total de extents asignados cuando el objeto es creado. Este parámetro permite asignar mucho espacio cuando un objeto es creado. El valor mínimo y default es 1.

MAXEXTENTS.- Especifica el numero total de extents incluyendo el primero, que Oracle puede asignar para el objeto. El valor mínimo es de 1.

UNLIMITED.- Especifica que los extents deben ser asignados tanto como sea necesario. Esto se recomienda para reducir la fragmentación.

Si se desea mejorar la concurrencia de las operaciones de ordenamiento múltiple y reducir el gasto en manejo de espacio se pueden crear tablespaces temporales. No se pueden almacenar objetos permanentes en un tablespace temporal.

Para decirle a la base de datos que el tablespace que estamos creando es temporal lo especificamos con la cláusula TEMPORARY como vimos con anterioridad.

```
CREATE TABLESPACE tablespace TEMPORARY;
```

Para modificar un tablespace existente y convertirlo en uno temporal lo podemos hacer con la instrucción alter. Ejemplo:

```
ALTER TABLESPACE tablespace TEMPORARY;
```

La instrucción ALTER TABLESPACE nos sirve para cambiar algunos de los parámetros u opciones que tiene un tablespace ya creado. Algo de lo que se puede hacer con esta instrucción es:

Alterar los parámetros de almacenamiento para objetos que se creen en un futuro en ese tablespace.

También se puede alterar la disponibilidad de la base de datos (online y offline). Cualquier tablespace se pone online al abrir la base de datos. El tablespace system siempre esta online ya que Oracle siempre tiene que acceder al diccionario de datos. Ejemplo de esta instrucción es:

```
ALTER TABLESPACE users ONLINE;
```

Por otro lado también se puede poner un tablespace offline esto se permite por las siguientes razones:

- Para hacer que una porción de la base de datos no esta disponible mientras el resto sí lo esta.
- Para realizar un respaldo offline de un tablespace.
- Para hacer que una aplicación y su grupo de tablas queden temporalmente no disponibles mientras se actualiza y da mantenimiento a la aplicación.

```
ALTER TABLESPACE users OFFLINE;
```

Otra opción es hacer que un tablespace sea de solo lectura con la instrucción READ ONLY para prevenir escritura en los datafiles del tablespace y después respaldar. El ejemplo de la instrucción se presenta a continuación:

```
ALTER TABLESPACE usuario READ ONLY;
```

Por el contrario cuando se quiere que un tablespace de solo lectura se convierta en un o de lectura escritura se tiene la instrucción READ WRITE, EJEMPLO:

```
ALTER TABLESPACE usuario READ WRITE;
```

Para poder poner de solo lectura un tablespace se necesitan ciertos requisitos que son:

- El tablespace debe estar online.
- Se debe estar seguro que no hay información para deshacer cambios que necesite ser aplicada al tablespace, por lo cual no se puede hacer read only un tablespace con segmentos de rollback.
- Se debe terminar todos los respaldos antes de convertir un tablespace a solo lectura.

También se puede eliminar un tablespace y su contenido con la instrucción DROP TABLESPACE. El tablespace system es el único que no se puede borrar. Pero se debe tener en cuenta que si se elimina un tablespace los datos no se pueden recuperar.

Los datafiles del tablespace permanecen por lo que se tiene que borrar directamente desde el sistema operativo. En la vista del diccionario de datos DBA_TABLESPACES un tablespace eliminado aparece como INVALID.

Ahora Mostrare el ejemplo de la instrucción agregando la cláusula INCLUDING CONTENTS que permite borrar los segmentos contenidos en el tablespace:

```
DROP TABLESPACE users
    INCLUDING CONTENTS;
```

Para poder ver información acerca de los tablespace se tiene las vistas del diccionario de datos que se muestran a continuación:

- DBA_TABLESPACES.- Esta vista nos sirve para listar los nombres y los parámetros de almacenamiento por default de todos los tablespace en la base de datos.
- DBA_DATA_FILES.- esta vista nos sirve para saber los nombres, tamaños, y tablespaces asociados de una base de datos.
- DBA_FREE_SPACE.- Sirve para saber la cantidad de espacio disponible en los extents libres de cada tablespace de la base de datos.

2.1.1.3 Esquemas y objetos de esquema (Schemas and Schema Objects).

Un esquema (schema) es una colección de objetos de la base de datos. Los objetos de esquema (Schema Objects) son estructuras lógicas que refieren directamente a los datos de la base de datos. Los objetos de esquema incluyen estructuras como lo son tablas, vistas, secuencias, procedimientos almacenados, sinónimos, índices, etc. No hay una relación entre los esquemas y los tablespaces, los objetos de un esquema pueden estar almacenados en diferentes tablespaces y un tablespace puede almacenar objetos de diferentes esquemas.

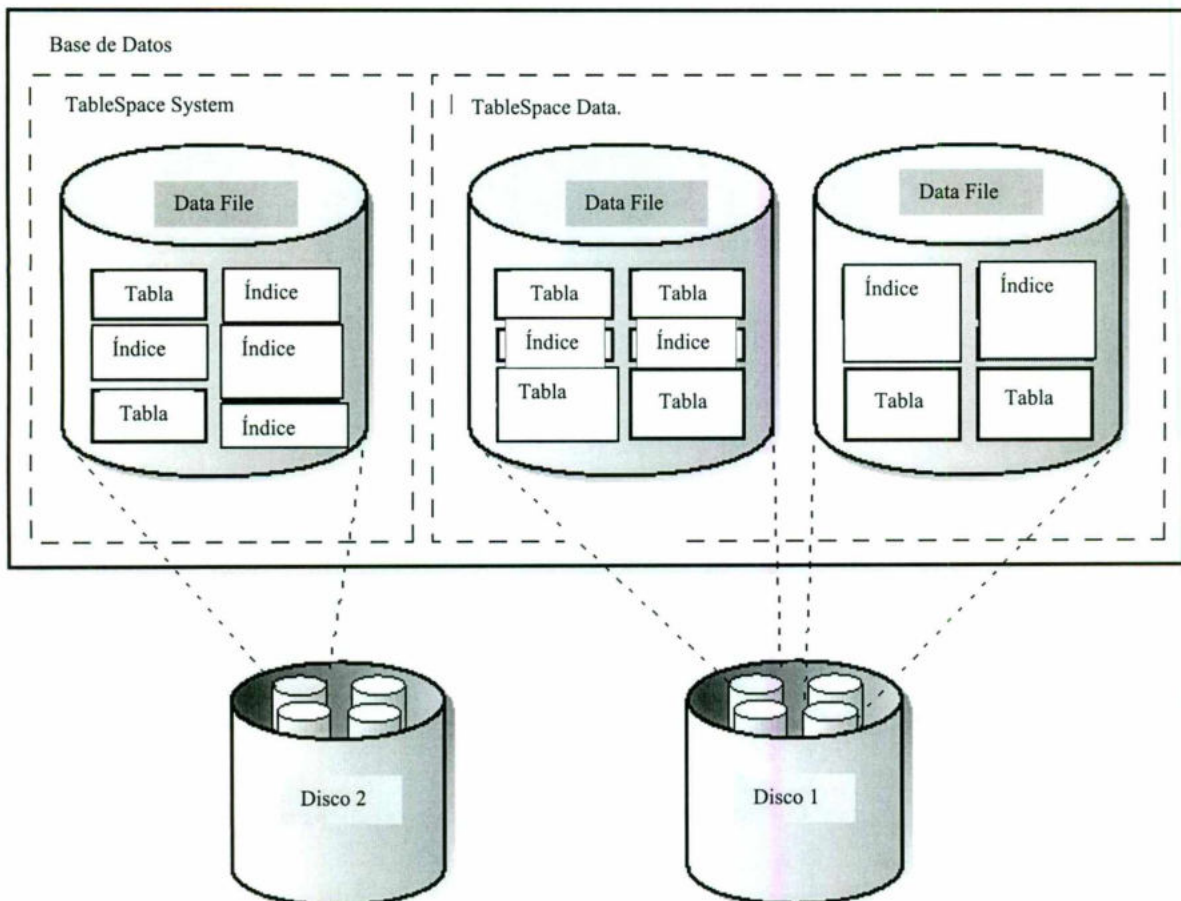


Fig. 2.2 Esquema de los Tablespace, Datafiles y Objetos Esquema.

2.1.1.4 Bloques.

Los datos de la base de datos Oracle son almacenados en bloques (Data Blocks). Un bloque corresponde a un número específico de bytes de espacio físico en disco de la base de datos. El tamaño de los bloques, es específico para cada base de datos cuando es creada. Una base de datos usa y asigna espacio libre de la base de datos en los bloques de datos de Oracle. Para identificar mejor los bloques en la base de datos vea la figura 2.3 y la 2.5.

2.1.1.5 Extents.

El siguiente nivel lógico de la base de datos se llama extent (Extensión). Un extent es un número específico de bloques contiguos, obteniendo un reparto sencillo, usado para almacenar un tipo de información específica. Para identificar mejor los extents en la base de datos vea la figura 2.3 y la 2.5.

2.1.1.6 Segmentos.

El nivel lógico de almacenamiento que está por encima de un extent es llamado segmento. Un segmento es un conjunto de extents asignado para cierta estructura lógica. Existen varios tipos de segmentos los cuales mencionamos enseguida. Para identificar mejor los segmentos en la base de datos vea la figura 2.3 y la 2.5.

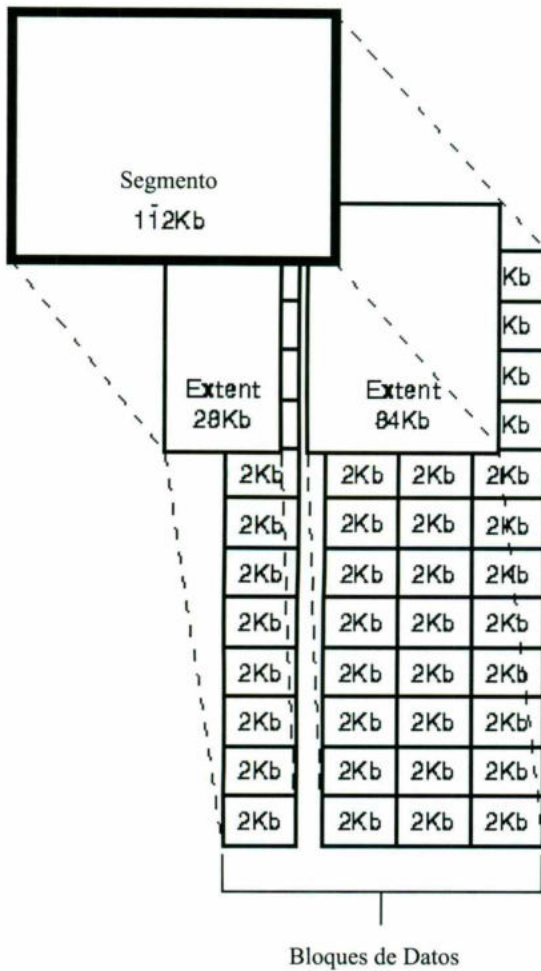


Fig. 2.3 Segmentos, Extents y Bloques.

2.1.1.7 Segmentos de datos.

Cada tabla no agrupada en un cluster tiene un segmento de datos. Todos los datos de la tabla son almacenados en los extents de su segmento de datos. Para una tabla particionada, cada partición tiene un segmento de datos.

Cada cluster tiene un segmento de datos. Los datos de cada tabla en el cluster son almacenados en el segmento de datos del cluster.

2.1.1.8 Segmento de índice.

Cada índice tiene un segmento de índice que almacena todos sus datos. Para un índice particionado, cada partición tiene un segmento de índice.

2.1.1.9 Segmento de rollback.

Uno o mas segmentos de rollback para una base de datos son creados por el DBA para almacenar temporalmente la información de las transacciones que podrían recuperarse. La información en un segmento de rollback es usada:

- Para generar consistencia en la lectura de la información de la base de datos.
- Durante la recuperación de información.
- Para dar un rollback a las transacciones no comprometidas de los usuarios.

Se puede decir que el segmento de rollback es usado para guardar el viejo valor cuando un proceso esta realizando cambios en los datos.

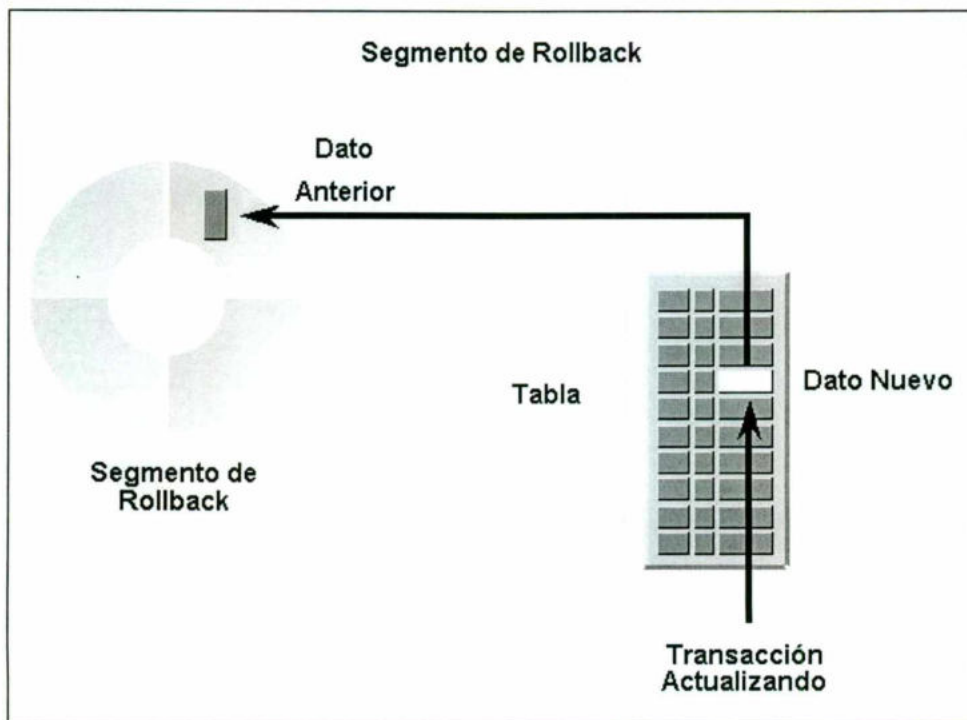


Fig. 2.4 Segmentos de Rollback.

2.1.1.10 Segmento temporal.

Los segmentos temporales son creados por Oracle cuando una instrucción SQL necesita una área de trabajo temporal para completar la ejecución. Cuando la instrucción termina su ejecución, los extents del segmento temporal son regresados a el sistema para su uso futuro.

2.1.2 Estructuras Físicas De La Base De Datos.

2.1.2.1 Datafiles (Archivos de Datos)

Cada base de datos Oracle tiene uno o más datafiles (archivos de datos) físicos. Un datafile de la base de datos contiene todas los datos de la base de datos. Los datos de las estructuras lógicas de la base de datos como son tablas e índices están almacenados físicamente en los archivos de datos asignados para una base de datos.

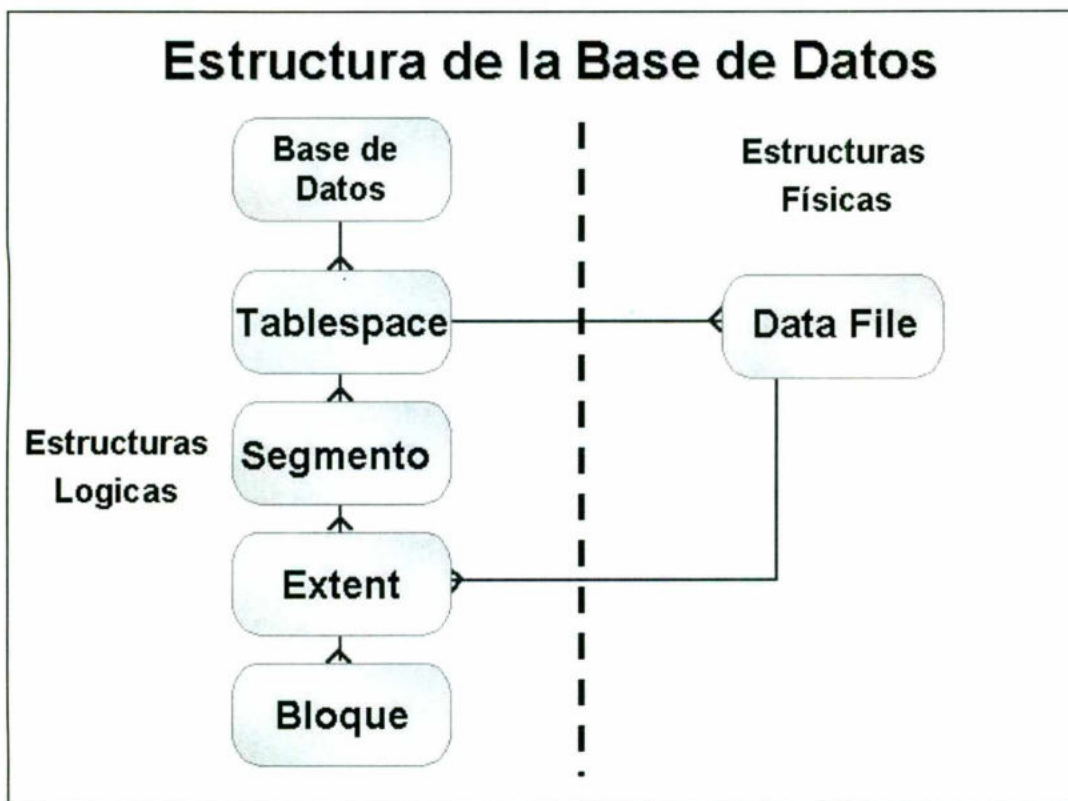


Fig. 2.5 Estructura de la Base de Datos y Posición de los Data Files en la Misma.

Las características de los datafiles son:

Un archivo de datos puede ser asociado con solo una base de datos.

Los datafiles pueden tener cierto conjunto de características que les permiten extenderse automáticamente cuando la base de datos tiene falta de espacio.

Uno o más datafiles forman una unidad lógica de almacenamiento de la base de datos llamada tablespace, como se dijo con anterioridad.

Los datos en los datafiles son leídos, tanto como sea necesario, durante la operación normal de la base de datos y almacenados en la memoria cache de Oracle. Por ejemplo, pensemos que un usuario quiere acceder a algún dato en una tabla de una base de datos. Si la información requerida no esta lista en la memoria cache para la base de datos, esta es leída del datafile apropiado y almacenada en memoria.

Un dato Nuevo o modificado no es necesariamente escrito en un datafile inmediatamente. Para reducir el numero de accesos a disco e incrementar el desempeño, los datos son juntados en memoria y escritos en su apropiado datafile todos a la vez, como un proceso background de Oracle.

2.1.2.2 Manejo de los Datafiles (Archivos de Datos).

Se puede crear y adicionar datafiles a un tablespace para incrementar el espacio total de disco para un tablespace y consecuentemente para la base de datos.

Par esto se utiliza el comando ALTER TABLESPACE, la siguiente instrucción crea un datafile para el tablespace rb_segs:

```
ALTER TABLESPACE rb_segs  
  ADD DATAFILE 'filename1' SIZE 1M;
```


También se puede activar o desactivar el crecimiento automático de un datafile. Para indicar el crecimiento automático se utiliza la instrucción ALTER TABLESPACE. En el ejemplo se activa la autoextensión para el archivo2 en el tablespace usu:

```
ALTER TABLESPACE usu
  ADD DATAFILE 'archivo2' SIZE 10M
  AUTOEXTEND ON
  NEXT 512K
  MAXSIZE 250M;
```

En la siguiente instrucción por el contrario se desactiva el crecimiento automático de archivo2:

```
ALTER DATABASE DATAFILE 'archivo2'
  AUTOEXTEND OFF;
```

También se puede cambiar la longitud fija de un archivo con esta mismo comando como por ejemplo en la instrucción:

```
ALTER DATABASE DATAFILE 'archivo2'
  RESIZE 100M;
```

Para renombrar data files se utiliza también la misma instrucción con la opción RENAME DATAFILE nombre_archivo1 TO nombre_archivo2, ejemplo:

```
ALTER TABLESPACE users
  RENAME DATAFILE 'filename1', 'filename2'
  TO 'filename3', 'filename4';
```

Para ver la información acerca de los datafiles se tiene las vistas del diccionario de datos:

- **DBA_DATA_FILES.**- Esta tabla nos muestra información acerca de los data files como es el nombre del tablespace al cual pertenece, el tamaño de bloque, su status ya sea disponible o no, si es auto extensible, etc.
- **V\$DATAFILE.**- Esta vista contiene información de data files sacada del archivo de control, mostrando si es que esta disponible el data file, su status y en general datos de control para el manejador de base de datos.

2.1.2.3 Archivos Redo Log

Cada base de datos Oracle tiene un conjunto de dos o mas archivos redo log. El conjunto de los archivos redo log para una base de datos es colectivamente conocido como el redo log de la base de datos. Un redo log esta hecho de registros de recuperación, cada uno de los cuales es un grupo de vectores de cambios describiendo un cambio sencillo en la base de datos.

La función primaria de los archivos redo log es registrar todos los cambios hechos a los datos. Debiendo prevenir una falla en los datos modificados de quedar permanentemente escritos e los datafiles, los cambios pueden ser obtenidos del redo log y manejarlos estos como nunca perdidos.

Los archivos redo log son críticos en la protección contra fallas de la base de datos. Para protegerse contra fallas del mismo redo log, Oracle permite multiplexar él redo log para que entonces dos o más copias del redo log puedan ser mantenidas en diferentes discos.

La información del archivo redo log es usada solo para recuperar a la base de datos de una falla del sistema o del medio de almacenamiento que previene a los datos de ser escritos en los datafiles de la base de datos.

Por ejemplo en una inesperada falla en el suministro de energía eléctrica que termine abruptamente con la operación de la base de datos, los datos en memoria no pueden ser escritos en los datafiles y los datos se pierden. Sin embargo cualquier dato perdido puede ser recuperado cuando la base de datos es abierta, después de que la energía eléctrica regresa, aplicando la información en el mas reciente archivo redo log a los datafiles de la base de datos.

2.1.2.4 Archivos de control.

Cada base de datos Oracle tiene un archivo de control el cual contiene entradas que especifican la estructura física de la base de datos. Por ejemplo estos contiene el siguiente tipo de información.

- Nombre de la base de datos.
- Nombre y localizaciones de los datafiles y archivos redo log de la base de datos.
- Fecha de creación de la base de datos.

Oracle permite multiplexar los archivos de control para protección de ellos mismos.

Cada que una instancia de un base de datos es arrancada, su archivo de control es usado para identificar la base de datos y los archivos redo log que deben estar abiertos para que proceda la operación de la base de datos. Si el maquillaje físico de la base de datos es alterado (por ejemplo, cuando un nuevo datafile o archivo redo log es creado), el archivo de control de la base de datos es automáticamente modificado por Oracle para reflejar los cambios.

2.1.2.5 Manejo de los archivos de control.

Los archivos de control son creados al mismo tiempo que se crea la base de datos. Por default al menos una copia del archivo de control es creada durante la creación de la base de datos. En algunos sistemas operativos se crean una o más copias. Se deben de crear copias de este archivo a manera de respaldo o por si se quiere cambiar algo posteriormente en este archivo.

Si uno quiere ponerle nombre al archivo de control durante la creación de la base de datos solo hay que especificarlo en el archivo de parámetros que se utiliza para crear la base de datos en la opción `CONTROL_FILES`.

Para sacar una copia del archivo de control se hace por medio del sistema operativo y es recomendable que primero se baje la base de datos antes de copiar este archivo.

Se puede crear un archivo de control después de haber creado la base de datos usando el comando `CREATE CONTROLFILE` esto se recomienda cuando:

- Se han dañado todos los archivos de control y no se tiene respaldo.
- Se quiere cambiar alguna de las opciones de la base de datos como el nombre u otros.

Para crear un nuevo archivo de control se tienen que seguir ciertos pasos cuidadosamente como son:

1. Hay que realizar una lista de los datafiles y los archivos redo log de la base de datos para tener un registro de la estructura y poder hacer recuperaciones posteriores.
2. Dar de baja la base de datos.
3. Respalidar todos los datafiles y los archivos redo log.
4. Arrancar una nueva instancia, pero sin, montar ni abrir la base de datos.
5. Crear el nuevo archivo de control con la instrucción `CREATE CONTROLFILE`. Hay que tener cuidado con esta instrucción ya que podría dañar los datafiles y los archivos redo log.
6. Almacenar un respaldo del nuevo archivo de control en otro disco.
7. Editar el archivo de parámetros indicando el nuevo archivo de control.
8. Recuperar la base de datos si es necesario. Es por si se perdieron archivos redo log o datafiles y que el procedimiento que estamos realizando es de recuperación.
9. Abrir la base de datos normalmente.

Y estos fueron los pasos para crear un nuevo archivo de control operación que no es muy común.

También se pueden borrar archivos de control de un base de datos. Se puede eliminar uno de ellos pero tomando en cuenta que en una base de datos debe haber dos o mas archivos de control siempre por lo que debe quedar alguno.

Para esto se siguen los siguientes pasos:

1. Dar de baja la base de datos.
2. Editar el archivo de parámetros quitando el archivo de control de la opción.
CONTROL_FILES.
3. Reiniciar la base de datos.

Después de estos pasos el archivo se puede cambiar de ubicación o borrar desde el sistema operativo.

2.2 Utilerías Import Y Export.

2.2.1 Import y Export.

Las utilidades import y export permiten al DBA y a los desarrolladores realizar copias rápidas y fiables de los datos.

Export hace una copia de los datos y de las estructuras en las que están contenidos y las guarda en un archivo normal de sistema operativo.

Import lee los archivos creados al exportar y coloca los datos y sus estructuras en los archivos de la base de datos.

Los usos principales de estas utilerías son:

- Como parte de procedimientos de respaldo y recuperación.
- Para traspasar datos entre diferentes instancias de Oracle.
- Para traspasar datos de un tablespace a otro.
- Cuando surge la necesidad de reconstruir una base de datos existente, exportar e importar son la única manera de preservar los datos que se tienen actualmente antes de crearla de nuevo.

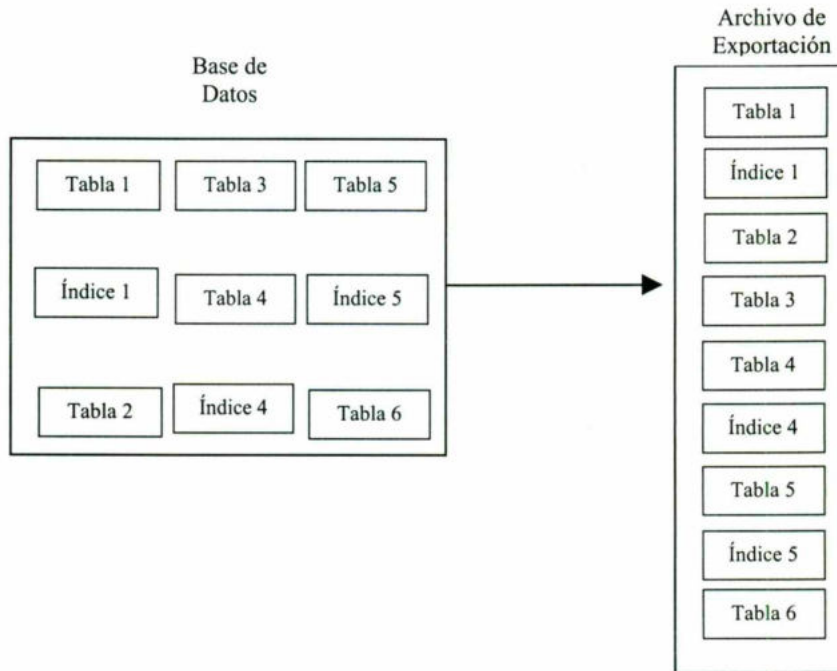


Figura 2.6 Exportación de la base de datos.

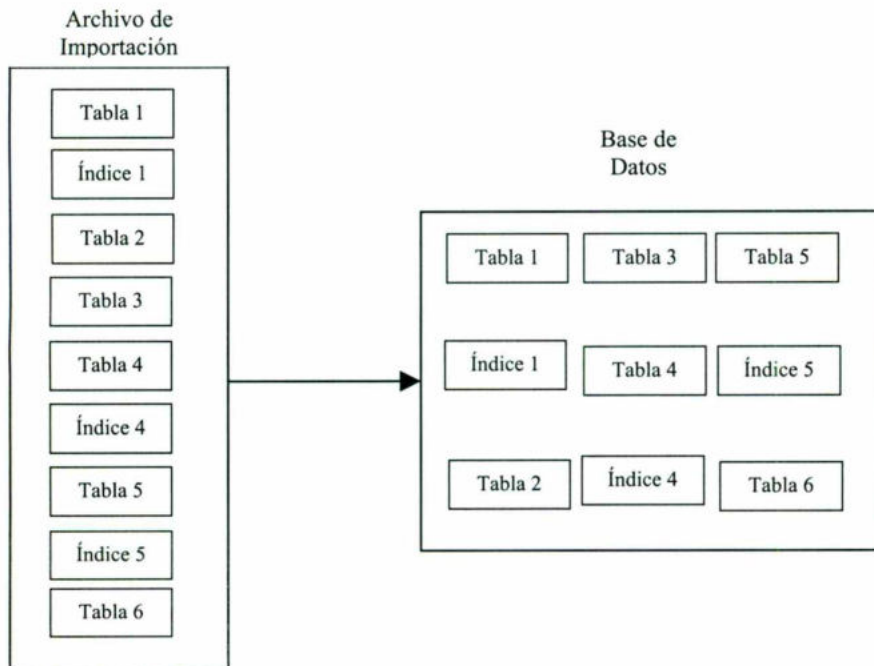


Figura 2.7 Importación de la base de datos.

Un ejemplo de esto es cuando se quieren exportar datos de la base de datos de producción e importar parte o todos los datos a la base de datos de desarrollo.

Import se invoca con el comando imp. Export con el comando exp. Como se ve las instrucciones son diferentes pero tienen más similitudes que diferencias. Para los dos casos hay tres formas de invocarlos las cuales son:

- Con un archivo de parámetros.
- Poniendo los parámetros en la línea de comando.
- Interactuando con el manejador.

La sintaxis de export utilizando archivo de parámetros es:

```
exp nombre_usuario/password PARFILE=nombrearchivo
```

PARFILE.- Es un archivo conteniendo los parámetros que típicamente se utilizan. Si se usan diferentes parámetros para diferentes bases de datos se pueden tener varios archivos de parámetros.

La siguiente opción es que se tecleen los parámetros después de la sentencia:

```
exp nombre_usuario/password <parametros>
```

Después de esta sentencia se colocaría el nombre del parámetro un signo de igual y el valor que debe tomar.

La ultima opción que se tiene para ejecutar el export es de manera interactiva para esto solo se tiene que teclear:

```
exp nombre_usuario/password
```

Y después de esto el export le preguntara en el prompt cada uno de los valores que va a signar a los parámetros.

La primera forma de invocar import utilizando un archivo de parámetros es:

```
imp nombre_usuario/password PARFILE=nombre_archivo
```

Para la segunda forma de invocar import se ponen los parámetros sobre la línea de comandos de manera similar al export poniendo el nombre del parámetro un signo de igual y el valor que se va a asignar al parámetro, todo esto inmediatamente después de la línea siguiente:

```
imp nombre_usuario/password <parametros>
```

Para realizar la invocación interactiva solo teclee:

```
imp nombre_usuario/password
```

Y después el manejador le pedirá cada uno de los parámetros en el prompt.

2.2.2 Modos De Exportación E Importación.

Las utilerías import y export se pueden ejecutar en tres modos dependiendo del objetivo que se siga. Estos modos se muestran a continuación.

2.2.2.1 Exportación en modo tabla.

Cuando se utiliza la exportación en modo tabla se le especifican a Oracle los nombres de una o mas tablas que se van a exportar. Oracle escribe los datos de las tablas en el archivo de exportación. Un ejemplo de esto es la sentencia:

```
Exp userid=pedro/Juárez tables=(feeling,ride,mast)
      File = fm.dmp
```

Esta sentencia exporta las tres tablas que se especificaron que pertenecen al usuario descrito.

2.2.2.2 Exportación en modo usuario.

En este caso Oracle exporta todos los objetos del usuario, incluyendo vistas, sinónimos triggers, etc. Esta modalidad se usa comúnmente para desfragmentar un tablespace. Después de exportar se puede borrar todos los objetos del usuario, el tablespace puede borrarse y volverse a crear. Aquí se muestra un ejemplo de la sentencia:

```
Exp userid=pedro/infante owner=(juan,pablo,jorge,ricardo)
```

Esta instrucción exporta los cuatro usuarios que están entre paréntesis.

2.2.2.3 Exportación de la base de datos completa.

Cuando se hace una exportación completa de la base de datos todas las instrucciones de creación de archivos de soporte como son, datafiles, tablespaces, segmentos de rollback, etc., y datos de usuario se almacenan en el archivo de exportación para cada usuario de la base de datos , excepto para el usuario SYS. Este archivo que se genera se puede utilizar para la importación de la base de datos en otro servidor. Un ejemplo de este tipo de exportación se muestra en la siguiente sentencia:

```
Exp userid=system/not_manager full=y grants=y indexes=y
```

El archivo de datos que se genera le pone el nombre expdat.dmp

2.2.2.4 Importación en modo tabla.

Al utilizar la importación en modo tabla se indican a Oracle los nombres de una o mas tablas que hay que importar. Oracle escribe los datos de las tablas en la base de datos. La sentencia que se muestra nos da el ejemplo de este modo de importación:

```
Imp userid=fernando/gomez  
tables=(impuestos,empleados,salarios)
```

Esta sentencia importa las tablas que están entre paréntesis que pertenecen al usuario fernando.

2.2.2.5 Importación en modo usuario.

En la importación en modo usuario, Oracle importa todos los objetos del usuario, incluyendo vistas, sinónimos, triggers, procedimientos, etc. Esto se utiliza generalmente para crear un tablespace y adicionarle los datos de ese usuario. Un ejemplo de esto es:

```
Imp userid=ruben/morales fromuser=(juan,,pedro)
touser=(juan2,pedro2) file=usu2
```

Importa los usuarios cuyos nombres se encierran entre paréntesis tomándolos del archivo y almacenándolos en los usuarios que en este caso se les va a poner el mismo nombre con un 2 al final.

2.2.2.6 Importación de la base de datos completa.

La importación completa de la base de datos se realiza en dos partes. En la primera todas las instrucciones de creación de archivos de soporte de la base de datos que están en el archivo de exportación se ejecutan. Cuando se completa la primera fase ya se tiene la estructura completa de la base de datos en la nueva base de datos. Posteriormente en la siguiente fase se llevan los objetos de los usuarios a sus respectivos tablespaces.

```
Imp userid=lopez/beto full=y file=todabase
```

Esta instrucción importa toda la base de datos desde el archivo todabase.

2.3 Migración.

En esta sección entendemos como migración la transferencia de datos de una base de datos a otra sin que se pierdan datos.



Figura 2.8 Pasos de la migración.

Para la migración se recomienda seguir ciertos pasos entendiéndolos correctamente. Estos pasos se aplican para cualquier sistema operativo. Los pasos de la migración se muestran gráficamente en la figura 2.2.

A continuación describiremos los pasos para la migración.

1. Decida que método de migración va a utilizar basándose en las consideraciones que se tengan que tener sobre la base de datos de producción, sus objetivos de migración y el conocimiento y capacidades que se tengan de algún método en específico.

Estime los recursos que se necesitan para la migración.

Desarrolle un plan para probar la migración.

2. Pruebe el proceso de migración.

La prueba de migración debe ser realizada en un ambiente creado para probar la migración y no debe interferir con la base de datos de producción.

3. Evaluar la prueba de migración.

Hay que comparar los resultados de la migración verificando que no existan anomalías entre los datos antes de migrar y los datos ya migrados en la prueba.

Investigue la manera de corregir los problemas que surjan en las pruebas de migración. Y repita los pasos 1,2,3 tantas veces como sea necesario hasta que elimine todos los problemas.

4. Prepare y preserve la base de datos origen.

Realice un calendario del tiempo requerido para respaldar y migrar.

Realice un respaldo completo de la base de datos actual.

5. Migrar los datos.

Para migrar la base de datos se puede usar las utilerías Import y export aunque también hay otras herramientas para migración. En el caso del Import y export existen tres pasos básicos:

- a) Exportar los datos de la base de datos que se esta migrando.
- b) Crear una base de datos en la cual se va a importar los datos exportados.
- c) Importar los datos exportados a la nueva base de datos.

6. Afine y ajuste la nueva base de datos.

Si la nueva base de datos es una versión mayor, cheque que características nuevas existen que puedan mejorar su base de datos.

Desarrolle nuevos procedimientos de administración tanto como sea necesario.

El papel del DBA en este proceso es primordial ya que es el encargado de realizarlo y monitorear todos los pasos de la migración.

Los desarrolladores tiene la responsabilidad de verificar que las aplicaciones que corrían sobre la vieja base de datos corran bien en la nueva.

2.4 Utilización Del Diccionario De Datos.

La información del diccionario de datos permite que Oracle gestione sus recursos así como realizar el seguimiento de la información relativa a quien se registra en la base de datos y que archivos se requieren para que la base de datos se ejecute.

El diccionario de datos tiene unas tablas base las cuales solo consulta y modifica Oracle. Para poder consultar la información del diccionario de datos existen las vistas de acceso a usuario que sumarizan y muestran la información de las tablas base del diccionario de datos. La información del diccionario de datos es de solo lectura para los usuarios y hasta para el DBA.

El diccionario de datos reside en el espacio de tablas del sistema. Este pertenece al usuario SYS.

2.4.1 Tipos De Vistas Del Diccionario De Datos.

Las vistas del diccionario de datos están clasificadas según el tipo de información que contienen. En el nombre de cada vista viene un prefijo que es el que identifica el tipo de vista del que se trata seguida de un guión bajo para unirse con el nombre de la vista.

Los tipos de vistas que se tienen son:

Prefijo	Significado
All	Devuelve la información sobre todos los objetos accesibles para un usuario.
User	Devuelve la información sobre todos los objetos propiedad de un usuario.
Dba	Devuelve una lista completa de la base de datos para todos los usuarios.
V\$	Vistas de rendimiento dinámico actualizadas por Oracle mientras se ejecuta. Al hacer consultas en ellas se puede saber información del rendimiento, de los archivos y de la memoria de la base de datos.

Tabla 2.1 Tipos de vistas.

2.4.1.1 Las vistas más útiles para administración.

Un usuario común no tiene privilegios para acceder a las vistas dba y v\$ que pertenecen al usuario SYS. Para permitirles acceso a estas vistas es necesario otorgarles el privilegio SELECT a cada uno de ellos. La información que contiene es muy importante para el DBA. Estas son las vistas dba más útiles:

Nombre de la Vista	Descripción
DBA_DATA_FILES	Información sobre los datafiles de la base de datos.
DBA_DB_LINKS	Todos los vínculos de la base de datos.
DBA_EXTENTS	Extents que comprenden todos los segmentos de la base de datos.
DBA_FREE_SPACE	Extents libres en todos los espacios de tablas.
DBA_INDEXES	Índices de la base de datos.
DBA_IND_COLUMNS	Columnas que incluyen los índices en todas las tablas.
DBA_OBJECTS	Todos los objetos de la base de datos.
DBA_ROLLBACK_SEGS	Descripción de los segmentos de rollback.
DBA_SEGMENTS	Almacenamiento asignado a todos los segmentos de la base de datos.
DBA_SEQUENCES	Describe las secuencias.
DBA_SYNONYMS	Describe los sinónimos.
DBA_TABLES	Describe todas las tablas.
DBA_TABLESPACES	Describe todos los tablespaces.
DBA_TAB_COLUMNS	Columnas de todas las tablas y vistas de la base de datos.
DBA_TAB_GRANTS	Muestra todos los derechos otorgados sobre los objetos de la base de datos.
DBA_TAB_PRIVS	Todos los privilegios sobre los objetos de la base de datos.
DBA_TS_QUOTAS	Cuotas de espacio de tablas para todos los usuarios.
DBA_USERS	Información de los usuarios.
DBA_VIEWS	Texto de todas la vistas.

Tabla 2.2 Vistas DBA más útiles.

Por otro lado tenemos las vistas v\$ mas útiles las cuales son:

Vista	Contenido
V\$DATAFILE	Información de los datafiles utilizados por la base de datos. Así como la información contenida en los archivos de control.
V\$LIBRARYCACHE	Información sobre la gestión de las ordenes SQL en el área compartida.
V\$LOCK	Información de los bloqueos incorporados a los objetos durante las sesiones de acceso a la base de datos. Los bloqueos se utilizan para que los usuarios cambien los datos de la base de datos que otro usuario ya ha comenzado a modificar.
V\$LOG	Información de los registros de recuperación, extraída de los archivos de control.
V\$LOGFILE	Información sobre la localización de los nombres de archivos del registro de recuperación de la instancia.
V\$PARAMETER	Los valores de todos los parámetros de los archivos de inicialización.
V\$PROCESS	Información relativa a los procesos en curso.
V\$ROLLNAME	Información sobre los segmentos de anulación o segmentos de rollback.
V\$ROLLSTAT	Estadísticas sobre los segmentos de anulación (rollback) en línea.
V\$ROWCACHE	Información sobre el rendimiento/actividad del diccionario de datos en memoria.
V\$SESSION	Información sobre las sesiones activas.
V\$SESSTAT	Estadísticas sobre las sesiones activas de las que informa v\$session.
V\$SQLAREA	Estadísticas sobre los cursores actualmente contenidos en el área compartida. Los cursores son trozos de memoria que usa Oracle para el procesamiento de las ordenes SQL.
V\$STATNAME	El significado de cada estadística de la que informa v\$sesstat.
V\$SYSSTAT	Estadísticas de todo el sistema, en función de las sesiones activas.
V\$WAITSTAT	Detalle de las situaciones producidas cuando más de una sesión quiere acceder a los datos de la base de datos. Puede haber situaciones de espera cuando más de una sesión quiere manipular la misma información en el mismo instante.

Tabla 2.3 Vistas v\$ más comunes.

Para ver que columnas contiene cada una de estas vistas se puede utilizar el comando de SQL*Plus DESCRIBE, pero hay que recordar que de inicio solo el DBA tiene derecho a consultar estas vistas por lo que hay que abrir sesión como administrador.

3 ADMINISTRACIÓN DE LA SEGURIDAD.

3.1 Manejo De Usuarios.

Un usuario es una persona que tiene acceso a la base de datos Oracle. El cual generalmente esta compuesto por un nombre de usuario o login y un password o contraseña.

3.1.1 Autenticación De Usuarios.

Como medio de seguridad se tiene que realizar la identificación para saber que usuario se conecta y que derechos tiene. La autenticación del usuario es asegurarse de que el usuario que se identifico realmente sea la persona que debe usar ese usuario lo cual se hace comúnmente con passwords. Para Oracle dependiendo de cómo se quiera autenticar la identidad de un usuario, se tiene tres maneras de definir usuarios antes de que se les permita crear una sesión:

3.1.1.1 Autenticación desde la Base de datos.

Esto quiere decir que Oracle se encarga de la identificación y autenticación de usuario. Para estos se tiene que especificar un password para el usuario.

La siguiente sentencia muestra la creación de un usuario autenticado por medio de la base de datos.

```
CREATE USER pedro IDENTIFIED BY p2ab1;
```

3.1.1.2 Autenticación Externa.

En este tipo de autenticación la cuenta de usuario es manejada por Oracle pero la administración del password por consiguiente la autenticación del usuario es realizada por un servicio externo el cual puede ser el sistema operativo o un servicio de red. Para este caso no se necesita un password de la base de datos. Cuando este tipo de autenticación se tiene que determinar el parámetro `OS_AUTHENT_PREFIX`. Este prefijo se va utilizar en los nombres de usuarios de Oracle. Este parámetro define el prefijo que Oracle adiciona al inicio de cada nombre de cuenta de usuario del sistema operativo. Cuando un usuario intenta conectarse Oracle compara el nombre que esta después del prefijo con su lista de nombres de usuario.

Por ejemplo, nosotros definimos el prefijo `OPSS$` para este tipo de usuarios. Sí un usuario de sistema operativo con el username "PEDRO" trata de conectarse a una base de datos Oracle y este es autenticado por medio del sistema operativo, Oracle checa que exista un usuario en la base de datos que se llame "OPSS\$PEDRO" y si esto sucede permite conectarse al usuario. Todas las referencias a este usuario van a tener el prefijo `OPSS$`.

Para especificar este tipo de autenticación se tiene que utilizar la opción `EXTERNALLY`

```
CREATE USER scott IDENTIFIED EXTERNALLY;
```

3.1.1.3 Autenticación a nivel organización.

En este caso la cuenta de usuario es mantenida por Oracle, pero el manejo del password y la autenticación la realiza el Oracle Security Service (OSS). Este servicio permite compartir a muchos servidores de bases de datos el manejo de la autenticación de una manera centralizada.

Para esto se tiene la opción GLOBALLY en el comando de creación de usuarios especificando el nombre de usuario externo, como se ve en la siguiente sentencia:

```
CREATE USER pedro IDENTIFIED GLOBALLY as '<nombre externo>';
```

3.1.2 Creación De Usuarios.

Para crear usuarios se tiene la instrucción CREATE USER. Solo el DBA tiene derecho de crear usuarios. Un ejemplo de la creación de usuarios y la explicación de cada una de las opciones se presenta a continuación:

```
CREATE USER escolar
  IDENTIFIED BY maestro
  DEFAULT TABLESPACE data_ts
  TEMPORARY TABLESPACE temp_ts
  QUOTA 100M ON test_ts
  QUOTA 500K ON data_ts
  PROFILE clerk;
```

Los parámetros u opciones de este ejemplo se explican a continuación:

Escolar.- Es el nombre del usuario el cual debe ser único con respecto a los usuarios y roles.

IDENTIFIED.- Esta opción nos permite definir el tipo de autenticación de usuario. Con la opción BY le decimos que se autentifica a través de la base de datos, y después de la opción ponemos el password. La opción EXTERNALLY indica que el usuario se autenticara a través del sistema operativo.

DEFAULT TABLESPACE.- Todo usuario tiene un tablespace por default en el cual se guardan los objetos que se crean para su esquema a los cuales no se les especifica un tablespace para almacenarse. El tablespace por default si no se especifica es el tablespace SYSTEM lo cual no es lo más correcto, lo ideal es, que los objetos de los usuarios no se almacenen en el tablespace SYSTEM.

TEMPORARY TABLESPACE.- Cada usuario tiene también un tablespace temporal cuando un usuario ejecuta una instrucción SQL que requiere un segmento temporal. Este segmento se almacena en el tablespace temporal. También para este el default es el tablespace SYSTEM si no se especifica. Lo ideal sería que se tuviera un tablespace temporal para cada usuario.

QUOTA.- La cuota permite definir la cantidad de almacenamiento físico asignado a un usuario en un tablespace específico de la base de datos. K y M nos permiten definir si son kilobytes y megabytes.

Por default un usuario no tiene cuota en ningún tablespace de la base de datos. Para que un usuario pueda crear objetos se le debe asignar una cuota. Mínimamente se debe de asignar una cuota para el tablespace DEFAULT, y para aquellos tablespaces en los cuales se pretenda crear objetos. También se puede especificar una cuota ilimitada con la opción UNLIMITED permitiendo que no exista restricción en el almacenamiento de objetos para el usuario.

PROFILE.- Permite especificar un profile (perfil). Los profiles los abarcaremos mas adelante.

3.1.3 Alterar Usuarios.

Los usuarios pueden cambiar sus propios passwords con el commando ALTER USER. Para cambiar cualquier otra opción se tiene que tener el privilegio para alterar un usuario.

También se pueden cambiar las opciones de seguridad del usuario con esta instrucción afectando las sesiones futuras y no las actuales.

Un ejemplo de esta instrucción es cuando se quiere cambiar un password como es:

```
ALTER USER escolar IDENTIFIED BY pa2001;
```

Otro uso es cambiar la cuota de almacenamiento para un tablespace como por ejemplo:

```
ALTER USER juan QUOTA 10M ON usu_ts;
```

Las opciones para esta instrucción vendrían siendo las mismas que para crear un usuario permitiendo cambiar estas opciones.

3.1.4 Eliminar Usuarios.

Cuando un usuario es eliminado su esquema y todos sus objetos son removidos del diccionario de datos.

Si lo que se quiere es que el usuario no tenga acceso a sus objetos lo mejor es revocar el derecho de crear sesión del usuario.

Un usuario debe estar desconectado para poder eliminarlo. Y para eliminarlo tenemos el comando DROP USER, se debe usar la opción CASCADE para eliminar el usuario y todos sus objetos asociados y las llaves foráneas que dependen de las tablas del usuario a borrar. Si no se especifica la opción CASCADE y el usuario tiene objetos se producirá un error. A continuación se muestra la instrucción:

```
DROP USER juan CASCADE;
```


3.2 Manejo de Profiles (Perfiles).

Un profile o (perfil) son los límites de consumo de recursos que se asignan a los usuarios de la base de datos Oracle. A menos que el DBA los restrinja se establece por default que los usuarios no tienen ningún límite en cuanto a los recursos que pueden consumir.

Se puede decir que un profile es el nombre de un conjunto de límites de recursos. Se puede asignar un profile a cada usuario, y un profile puede ser asignado a varios usuarios. Por default no se asigna ningún profile a los usuarios.

3.2.1 Creación De Profiles.

Teniendo el privilegio para crear profiles se puede crear un profile con la sentencia `CREATE PROFILE` adicionando los límites de recursos que se quieran.

El siguiente ejemplo crea un profile llamado empleado:

```
CREATE PROFILE empleado LIMIT
  SESSIONS_PER_USER 2
  LOGICAL_READS_PER_SESSION unlimited;
```

`SESSIONS_PER_USER`.- Aquí se agrega esta opción que nos dice cuántas sesiones puede crear el usuario que en este caso son 2.

`LOGICAL_READS_PER_SESSION`.- Lecturas lógicas por sesión, lo que limita el uso del buffer, en este caso se especifican ilimitadas.

Todos los límites de recursos que no se especifiquen toman el conjunto de valores del perfil DEFAULT el cual de inicio no tiene límite para los recursos, por eso es que el DBA debe de especificar estos límites.

3.2.2 Asignación De Profiles.

Después de que un perfil ha sido creado se puede asignar este a los usuarios de la base de datos. Cada usuario puede ser asignado solo a un perfil en cualquier tiempo dado. Si se le asigna un perfil a un usuario que ya tenía uno. El primero se pierde quedando con el perfil que acabamos de asignar. Los perfiles solo pueden ser asignados a usuarios y no a roles o a otros perfiles.

Los perfiles pueden ser asignados usando la sentencia CREATE USER o ALTER USER especificándolo en la opción PROFILE como se ve al final del ejemplo. Aquí tenemos unos ejemplos de la asignación:

```
CREATE USER pablo IDENTIFIED BY pablito
    DEFAULT TABLESPACE datos
    TEMPORARY TABLESPACE tempo
    QUOTA unlimited ON datos
    PROFILE desarrollo;
```

Otro ejemplo de esto es:

```
ALTER USER pablo
    PROFILE desarrollo;
```

3.2.3 Alterar Profiles.

Para alterar los límites de recursos se puede utilizar la sentencia ALTER PROFILE.

Cualquier límite de recursos que se especifique en esta instrucción eliminará el límite que se estableció al definir el perfil inicialmente. Estos cambios toman efecto en la sesión siguiente a la alteración del perfil y no a la sesión activa.

Como ejemplo de la alteración tenemos la sentencia:

```
ALTER PROFILE clerk LIMIT
    CPU_PER_CALL default
    LOGICAL_READS_PER_SESSION 20000;
```

3.2.4 Eliminar un Profile.

Para eliminar un perfil se puede utilizar la sentencia DROP PROFILE. Para poder eliminar un perfil asignado a un usuario es recomendable usar la opción CASCADE.

Como ejemplo, la siguiente sentencia borra un perfil asignado a un usuario:

```
DROP PROFILE clerk CASCADE;
```

El usuario que tenía asignado este perfil automáticamente toma el perfil DEFAULT. El perfil DEFAULT no puede ser eliminado. Los cambios efectuados con la eliminación del perfil se llevan a cabo en la sesión siguiente.

Habilitando o deshabilitando los límites de recursos.

Los límites en los recursos pueden ser habilitados o deshabilitados según lo especifique el DBA.

Para forzar esta habilitación de los límites de los recursos se tienen dos métodos:

Uno se utiliza mientras la base de datos está abierta y es utilizando la sentencia ALTER SYSTEM. Uno va a especificar si a este parámetro se le asigna TRUE o FALSE. Por ejemplo en la sentencia siguiente se fuerza la habilitación de los límites de recursos con TRUE:

```
ALTER SYSTEM  
    SET RESOURCE_LIMIT = TRUE;
```

Esta instrucción no cambia permanentemente los límites de recursos solo lo hace durante la sesión ya que no modifica el archivo de parámetros directamente.

El otro método se utiliza cuando la base de datos está dada de baja cambiando en el archivo de parámetros el parámetro RESOURCE_LIMIT asignándole TRUE o FALSE según se quiera. Al arrancar la base de datos este cambio toma efecto y fuerza la habilitación o deshabilitación de los límites en los recursos.

3.2.5 Listando La Información de Usuarios y Profiles.

Para este fin se tiene algunas vistas del diccionario de datos veremos las principales a continuación:

DBA_USERS .- Esta vista muestra los usuarios y la información asociada a estos.

SYS.DBA_TS_QUOTAS.- Esta lista muestra todas las cuotas de tablespace asignadas específicamente a cada usuario.

SYS.DBA_PROFILES.- En este caso se muestra todos los profiles en la base de datos, así como también información acerca de sus límites.

3.3 Manejo De Privilegios Y Roles.

En esta sección comentaremos los conceptos básicos de los privilegios y los roles. Además mencionamos las instrucciones que nos permiten crear los roles y otorgar los privilegios las cuales se introducen en el SQL*PLUS conectándose como administrador. El SQL*PLUS puede ser utilizado en plataformas UNÍX, Windows NT o Windows 9.x.

3.3.1 Privilegios

Un privilegio es un derecho de ejecución de un particular tipo de sentencia SQL o de acceso a un objeto de otro usuario. Algunos ejemplos de estos derechos son el que un usuario pueda realizar acciones como:

- Conectarse a a base de datos (Create session).
- Crear una tabla.
- Seleccionar filas de una tabla de otro usuario.
- Ejecutar un procedimiento almacenado de otro usuario.

Los derechos se otorgan para que los usuarios puedan realizar las tareas necesarias para completar su trabajo. Estas tareas son específicas y habrá una sola persona que necesite realizar una tarea específica y muchas que realicen varias tareas comunes. El otorgar excesivos privilegios a los usuarios puede comprometer la seguridad.

3.3.1.1 Privilegio de sistema.

Un privilegio de sistema es el derecho para realizar un acción en particular, o sobre algún objeto en particular. Por ejemplo los privilegios de crear tablespaces o de borrar líneas de una tabla cualquiera en la base de datos son privilegios de sistema.

Hay mas de 100 distintos privilegios de sistema. Cada uno de estos permite a un usuario realizar una operación particular de la base de datos.

Los privilegios de sistema pueden ser muy poderosos y deben ser otorgados con cautela a usuarios confiables.

3.3.1.2 Privilegios de objetos.

Un privilegio de objeto de esquema es un derecho para realizar una acción en particular sobre una tabla, vista, secuencia, procedimiento, función o package específico. Diferentes privilegios de objetos están disponibles para diferentes tipos de objetos. Por ejemplo el privilegio de borrar líneas o registros de la tabla de empleados es un privilegio de objeto.

Una manera rápida de otorgar o revocar todos los privilegios disponibles para un objeto es utilizando la palabra ALL o ALL PRIVILEGES.

3.3.2 Manejo de Roles.

Los roles o papeles son grupos de privilegios relacionados que se pueden otorgar a usuarios o a otros roles.

En general los roles se crean para:

- Manejar los privilegios para una aplicación de la base de datos.
- Manejar los derechos de un grupo de usuarios.

Un rol puede opcionalmente requerir autorización cuando un usuario trata de habilitar un rol.

3.3.2.1 Creando un Rol.

Para crearlos se utiliza la sentencia CREATE ROLE.

Inmediatamente después de la creación, un rol no tiene privilegios asociados con el. Para asociar privilegios con un nuevo rol, se deben otorgar privilegios u otros roles a el nuevo rol.

La siguiente instrucción crea el rol “empleado” el cual es autorizado usando el password “sistemas”.

```
CREATE ROLE empleado  
IDENTIFIED BY sistemas;
```

El nombre de un rol debe ser único entre los nombres de usuario y de roles.

3.3.2.2 Eliminar Roles.

Para realizar esta operación se tiene la sentencia DROP ROLE. Aquí muestro un ejemplo:

```
DROP ROLE empleado;
```

Esto se puede usar en el caso de que se quiera inhabilitar el conjunto de privilegios que se otorgaron con el rol. Los objetos no son eliminados al borrar un rol.

3.3.3 Otorgando Privilegios y Roles.

Para otorgar privilegios de sistema o roles a otros roles o usuarios se utiliza la sentencia GRANT.

Para que un usuario otorgue privilegios o roles debe haber obtenido la opción ADMIN OPTION para que todos los privilegios de sistema y roles le sean otorgados. También cualquier usuario con la opción de sistema GRANT ANY ROLE puede otorgar derechos de sistema.

En la siguiente sentencia tenemos ejemplo de la instrucción, en la cual se otorga el derecho de conectarse o crear sesión, y se le otorga el rol contabilidad a el usuario juan:

```
GRANT create session, contabilidad  
TO juan;
```

Los privilegios de objeto no puede ser otorgados junto con privilegios de sistema y roles en la misma sentencia GRANT.

Para otorgar privilegios de objetos a roles y usuarios se usa la misma sentencia GRANT.

Un usuario tiene que cumplir las siguientes condiciones para otorgar un privilegio de objeto:

- Que le pertenezca el objeto especificado.
- Que le hayan sido otorgados sus privilegios de objetos con la opción GRANT OPTION

La opción GRANT OPTION al ser otorgada a un usuario le permite a este que el mismo derecho que le fue otorgado el lo pueda otorgar a cualquier usuario o rol en la base de datos.

En el ejemplo que muestro a continuación se utiliza la sentencia GRANT para otorgar el privilegio de SELECT, INSERT, Y DELETE para todas las columnas de la tabla empleado a los usuarios pedro y pablo:

```
GRANT select, insert, delete ON empleado TO pedro, pablo;
```

Para otorgar derechos sobre una columna se tiene que especificar esta entre paréntesis después del tipo de derecho que se otorga, ejemplo:

```
GRANT INSERT (clave_empleado)  
ON empleado TO pablo;
```

3.3.4 Revocar Privilegios y roles.

Para revocar se utiliza la sentencia REVOKE, para cual se tiene que tener lo derechos correspondientes como lo fue para otorgar derechos.

La siguiente sentencia es un ejemplo de revocación del privilegio de sistema CREATE TABLE, y del rol programador al usuario Luis.

```
REVOKE create table, programador FROM Luis;
```

Para revocar un privilegio de objeto se utiliza la misma instrucción, pero para hacer esto se tiene que ser el usuario que originalmente otorgo el privilegio a ser revocado.

La siguiente sentencia ejemplifica la revocación de objeto:

```
REVOKE select, insert ON salario  
FROM pedro, pablo;
```

Como otro ejemplo se va a revocar todos los derechos al rol finanzas de la tabla salario:

```
REVOKE ALL ON salario FROM finanzas;
```

3.3.5 Listando Información de Privilegios y Roles.

Las vistas mas importantes del diccionario de datos que nos sirven para consultar la información sobre privilegios y roles son:

SYS.DBA_SYS_PRIVS.- Esta vista muestra todos los privilegios de sistema otorgados a usuarios y roles.

SYS.DBA_ROLE_PRIVS.- Esta vista muestra los roles otorgados a usuarios o a otros roles.

SYS.DBA_TAB_PRIVS.- En este caso se muestran los privilegios de objeto otorgados a los usuarios.

SYS.DBA_ROLES.- Lista los roles de la base de datos.

3.4 Tipos de Usuarios.

En esta parte vamos a tratar de manera un poco conceptual pero practica todo lo referente al manejo de usuarios, privilegios y roles, mostrando una forma de aplicación de estos conceptos la cual no se aplica a una plataforma en especial si no que nos da una idea de cómo manejar los usuarios para que nos permitan tener mas seguridad en nuestra base de datos. Estos conceptos se comentan pensando en las herramientas que nos provee Oracle y que ya mencionamos con anterioridad.

Existen muchos tipos de usuarios que ingresan a la base de datos. Los cuales realizan diferentes actividades en la base de datos lo cual a muchos de ellos nos permite agruparlos según sus funciones, los derechos que tengan, y la parte de la base de datos a la cual pueden acceder.

Es importante que se realice esta clasificación de usuarios ya que nos ayuda a incrementar la seguridad, dar un mayor orden en la base de datos y ayuda en gran medida al trabajo del administrador.

Esta clasificación se realiza según las características y necesidades de la organización. Se debe de pensar cuales son los grupos funcionales principales como son: Desarrolladores, DBA's, usuarios de consulta, etc. Muchas veces también se clasifica a los usuarios según la parte de la base de datos a la que tiene derecho a acceder como son usuarios de: recursos humanos, finanzas, contabilidad, almacenes, etc.

Dependiendo de las actividades que realice y el tipo de recursos que utilice cada grupo de usuarios se van definiendo los privilegios que va a tener cada grupo con los cuales se

pueden crear roles y a su vez se puede uno apoyar con perfiles para administrar los recursos de estos grupos de usuarios según sea el caso. Después de tener las clasificaciones se pueden ir creando los usuarios asignándolos a un grupo en específico. Se puede dar el caso de que cierto usuario aunque pertenezca a cierto grupo necesite algunos privilegios extras para desempeñar sus funciones. Si esto sucede serán pocos los casos si se definieron bien las clasificaciones y de ser así es mucho menos trabajo que si no se tuviera un orden y clasificación.

No se puede establecer clasificaciones fijas ya que según las características del negocio son las clasificaciones que se dan. Aun así a continuación se expone un ejemplo que ayuda a observar de manera práctica lo antes mencionado. Estas clasificaciones son tentativas para una compañía cualquiera de mediano tamaño y son un simple ejemplo.

Los tipos de usuarios que se presentan como ejemplo son:

Administrador de la base de datos.- Como es lógico es el que tiene los más altos privilegios en la base de datos y toma las decisiones sobre la administración de la misma. Para esto está en constante comunicación con los ejecutivos y los usuarios que son por así decirlo los dueños de la información. En algunos casos según el tamaño de la organización se puede manejar dos o más DBA's.

Asistentes de Administración.- Estos usuarios ayudan a las tareas de administración y principalmente se dedican a definir los objetos de la base de datos, así como sus constraints, triggers, procedimientos, etc. También puede ayudar en tareas como el respaldo. Estos usuarios van a tener menos privilegios que el DBA, y sus privilegios varían dependiendo de las tareas que les asigne el DBA.

Desarrolladores de aplicaciones.- Estos usuarios generalmente su función es programar los sistemas que se requieren en la organización y que tienen como back end la base de datos.

A estos usuarios generalmente solo se les da derecho a consultar y modificar las tablas. Aunque en algunos casos también pueden modificar la estructura de las tablas. Lo que sí es que deben de ser capaces de crear procedimientos almacenados, packages, triggers, etc. Hay que tener cierto control con estos usuarios ya que al tener ciertos privilegios y ser gente de sistemas se aventura en actividades que pueden provocar desorden en la base de datos como generar mucha información basura o modificar la estructura de tablas. Por lo que se debe de estar muy pendiente de los privilegios que se les otorgan.

Usuarios operativos o de consulta.- Estos usuarios muchas veces son el usuario final al cual se le da nada más acceso para poder consultar la base de datos para tener completo acceso a la información de la cual son dueños y poder generar reportes y consultas que faciliten su trabajo. Si estos usuarios necesitaran modificar datos para eso deben de tener las aplicaciones de captura y mantenimiento de tablas y catálogos.

3.5 Mas Conceptos Sobre Seguridad.

3.5.1 Estrategias De Respaldo.

Una de las tareas de DBA es el respaldo de la información de la base de datos. Esto se hace ya que uno o más archivos físicos pueden dañarse por una falla en el disco. Para recuperar información se requieren los datos respaldados. Hay algunas maneras de respaldar la información.

- Respaldos completos. Estos son respaldos de sistema operativo, respaldando datafiles, archivos redo log, y archivos de control de la base de datos. Este se tiene que realizar cuando la base de datos esta dada de baja.
- Respaldos parciales. Estos son respaldos de sistema operativo de una parte de la base de datos. El respaldo de los datafiles de un tablespace o el respaldo de un archivo de control son ejemplos de respaldos parciales.

Las utilerías Import y Export que revisamos anteriormente nos pueden ayudar a respaldar.

Las estrategias de respaldo se definen para tener un orden al respaldar información de la base de datos. Podría decirse que son procedimientos de respaldo que se definen de manera metódica según las características del negocio y la información.

Estas se definen según las necesidades o las características de información que tenga la organización.

Un ejemplo de esto es:

En algunas empresas muy grandes en las que se tiene información muy importante que se actualiza constantemente se realizan los respaldos siguientes:

- Se hacen respaldos diarios.
- Después de esto se hacen respaldos semanales.
- Posteriormente respaldos mensuales.
- Y además respaldos anuales.

Para estos respaldos ya se tiene a una persona encargada de hacerlos que en determinada hora tiene que hacerlo.

Además se calendarizan los respaldos mensuales o semanales.

Se define que todos los respaldos se hacen en cinta, y se almacenan en una bóveda a prueba de fuego y con código de entrada.

Esto fue en una gran organización pero tal vez en una pequeña empresa solo se hagan respaldos diarios o semanales en disco flexible por que los recursos no alcanzan para mas.

CONCLUSIONES Y COMENTARIO PERSONAL

Como se vio durante el presente trabajo son muchas las actividades que tiene que desarrollar un administrador de bases de datos, por lo que tal vez a veces este personaje tenga que tener ayudantes o que en una organización existan varios administradores de bases de datos.

Por otro lado pienso que el trabajo del administrador de bases de datos no es totalmente técnico ya que aparte de tener que tomar parte en decisiones críticas y tener la responsabilidad por la información de la empresa. Tiene que realizar muchas actividades de planeación para poder definir estrategias, para las actividades de la administración de base de datos. También tiene que tratar mucho con los usuarios que no solo son los empleados de sistemas sino que algunas veces tiene que tratar con usuarios finales y por supuesto con directivos de la empresa.

Pienso que la administración de bases de datos es una actividad que se le tiene que tomar mucha importancia, esto es por que se están realizando actividades críticas para la información de la organización.

Por otro lado se debe hacer caso a las determinaciones que tome el administrador de bases de datos, por que el es la persona que conoce la información y su manejo ya que se puede prestar, que a esta persona la vean como cualquier empleado de sistema, pero en realidad es un tomador de decisiones críticas en una organización.

Para pensar en una buena administración de bases de datos se tienen que analizar algunas cuestiones claves para que el trabajo de administración de la base de datos se realice de la mejor manera.

Yo diría que hay que dar mucho énfasis en el conocimiento y constante capacitación del encargado o encargados de la administración de bases de datos.

Por otro lado no se debe dejar por un lado la planeación de la base de datos ya que si no se tiene una buena planeación los problemas van a surgir posteriormente y el trabajo no tendrá un orden.

Dentro de esta planeación pienso que hay rubros principales donde se tiene que enfocar principalmente la planeación como son:

- El diseño de la base de datos.
- La planeación del almacenamiento con base en este diseño.
- Una estricta definición de políticas de seguridad que contemplen el manejo a usuarios y las estrategias de recuperación y respaldo.

Pienso que la administración de bases de datos tiene mucho futuro ya que mientras exista la información agrupada en bases de datos siempre habrá trabajo para los administradores de bases de datos que en mi opinión son personas muy importantes dentro de la organización moderna.

GLOSARIO.

Commit: Instrucción que da por terminada o compromete una transacción.

Default: Por defecto. Valor automático que toma un objeto cuando no se le asigna ningún valor.

Foreign Key: Llave foránea. Se utiliza para establecer relaciones entre tablas.

Índice: Es un objeto que puede hacer mas rápida una recuperación de registros usando un apuntador.

Instancia: Es una parte de la memoria de la computadora y los procesos necesarios para acceder a una base de datos Oracle.

Objeto Esquema u Objeto de la base de datos: Cada esquema esta asociado a un usuario un objeto esquema puede ser una tabla, un índice, una vista, etc.

Package: Es un objetos de la base de datos que almacena dentro de el procedimientos y funciones.

Privilegio: Es un derecho que se otorga sobre un objeto o una determinada acción en la base de datos.

Profile: (Perfil) Es el nombre de una serie de limites de recursos que se asigna a un usuario.

Redo log: Registros de rehacer.

BIBLIOGRAFIA.

- Abbey, Michael; Abramson, Ian; Corey, Michael J., Oracle 8i Guía de Aprendizaje., 1ª Edición, España, Osborne Mc Graw Hill, 2000.
- Elmasri, Rames; Navathe Shamkant, Sistemas de Bases de datos Conceptos Fundamentales., 2ª. Edición, México, Adison Wesley Longman, 2000.
- Koch, George, Oracle Manual de Referencia, 1ª Edición, España, Mc-Graw Hill Interamericana, 1992.
- Korth, Henry; Silberschatz, Abraham; Sudarshan S, Fundamentos de Bases de Datos., 3ª Edición, España, Mc Graw Hill, 1998.
- Oracle Corporation, Oracle 8i On-Line Generic Documentation CD-ROM, Versión 8.1, E.U.A., Oracle Corporation, 1999.
- Oracle Corporation, <http://www.oracle.com>, 03/05/2001