

Arquitectura del DBMS Oracle

guía de iniciación

Autor: Jorge Sánchez (www.jorgesanchez.net) año 2004
e-mail: <mailto:info@jorgesanchez.net>



Este trabajo está protegido bajo una licencia de **Creative Commons** del tipo **Attribution-NonCommercial-ShareAlike**.

Para ver una copia de esta licencia visite:

<http://creativecommons.org/licenses/by-nc-sa/2.0/>

o envíe una carta a:

**Creative Commons, 559 Nathan Abbott Way, Stanford, California
94305, USA.**



C O M M O N S D E E D

Los contenidos de este documento están protegidos bajo una licencia de **Creative Commons** del tipo **Attribution-Noncomercial-Share Alike**. Con esta licencia:

Eres libre de:

- Copiar, distribuir y mostrar este trabajo
- Realizar modificaciones de este trabajo

Bajo las siguientes condiciones:



Attribution (Reconocimiento). Debe figurar siempre el autor original de este trabajo



Noncommercial (No comercial). No puedes utilizar este trabajo con propósitos comerciales.



Share Alike (Compartir igual). Si modificas, alteras o construyes nuevos trabajos a partir de este, debes distribuir tu trabajo con una licencia idéntica a ésta

- Si estas limitaciones son incompatible con tu objetivo, puedes contactar con el autor para solicitar el permiso correspondiente
- No obstante tu derecho a un uso justo y legítimo de la obra, así como derechos no se ven de manera alguna afectados por lo anteriormente expuesto.

Esta nota no es la licencia completa de la obra, sino una traducción del resumen en formato comprensible del texto legal. La licencia original completa (jurídicamente válida y pendiente de su traducción oficial al español) está disponible en

<http://creativecommons.org/licenses/by-nc-sa/2.0/legalcode>

índice

| | |
|---|-----------|
| índice | 5 |
| notas sobre el funcionamiento de los RDBMS | 7 |
| introducción..... | 7 |
| diccionario de datos | 7 |
| conexión a un sistema gestor de bases de datos | 8 |
| herramientas que proporcionan los RDBMS | 8 |
| términos relacionados con los sistemas gestores de bases de datos | 9 |
| aproximación a Oracle | 11 |
| base de datos objeto-relacional..... | 11 |
| breve historia de Oracle..... | 11 |
| otros servidores | 12 |
| estrategia de Oracle | 12 |
| servidor Oracle | 14 |
| introducción..... | 14 |
| elementos del servidor Oracle | 14 |
| conexiones..... | 14 |
| estructura de las bases de datos Oracle | 15 |
| instancia de la base de datos | 15 |
| procesamiento de instrucciones SQL | 17 |
| conceptos del sistema Oracle | 19 |
| almacenamiento..... | 19 |
| transacciones | 20 |
| usuarios..... | 20 |
| pérdidas de información | 21 |
| bases de datos distribuidas..... | 22 |
| herramientas de Oracle | 22 |

notas sobre el funcionamiento de los RDBMS

introducción

Un RDBMS es un Sistema Gestor de Bases de Datos Relacionales. Se trata de software capaz de producir, manipular y gestionar bases de datos de tipo relacional.

Es un software que se antepone a los datos de una base de datos, de modo que cualquier acceso a los datos pasa por una petición al RDBMS que éste gestiona a fin de realizar la operación más conveniente sobre esa petición.

Prácticamente es un Sistema Operativo diseñado para el control del acceso a los datos. Para conseguir este control, todo RDBMS posee una serie de subsistemas que se encargan de gestionar cada servicio. Algunos de estos subsistemas son:

- ⊙ **Sistema de gestión de la memoria.** Encargado de decidir que parte de la memoria se dedica a cada tarea del RDBMS. Su función es que haya suficiente memoria para que el RDBMS funcione eficazmente y a la vez nunca dejar menos memoria de la que necesita el Sistema Operativo para que la máquina funcione.
- ⊙ **Gestión de Entrada y Salida.** Para conseguir que los accesos a los datos sean adecuados.
- ⊙ **Procesador de lenguajes.** Para interpretar las instrucciones SQL (o de otros lenguajes válidos) que los usuarios lanzan a la base de datos.
- ⊙ **Control de procesos.** Gestiona los programas en ejecución necesarios para el funcionamiento de la base de datos.
- ⊙ **Control de la red.** Para gestionar las conexiones a la base de datos desde la red y evitar problemas a la base de datos en caso de desconexión.
- ⊙ **Control de transacciones.** Permite gestionar las transacciones (series de operaciones que se pueden anular o llevar a cabo al final).

diccionario de datos

La gran ventaja de los RDBMS consiste en que permiten gestionar los datos de forma lógica, se utilizan estructuras más abstractas para los datos, a fin de evitar utilizar el complicado entramado físico que posee una base de datos.

El diccionario de datos agrupa los metadatos de una base de datos. En este diccionario aparecen todos los objetos de la base de datos; con su nombre, función, control de acceso (seguridad) y correspondencia física en los archivos de datos.

Cada vez que llega al gestor de bases de datos una petición sobre datos de una base de datos, el RDBMS abre el diccionario de datos para comprobar los metadatos relacionados con la petición y resolver si hay permiso de uso y donde localizar físicamente los datos requeridos.

conexión a un sistema gestor de bases de datos

Normalmente cualquier DBMS funciona como servidor, programa que está en ejecución esperando peticiones de conexión al sistema. En cada intento de conexión el sistema verificará qué usuario intenta conectar y si tiene permiso se produce la conexión.

En la conexión el usuario puede ejecutar peticiones sobre la base de datos en el lenguaje, o lenguajes, que el DBMS sea capaz de traducir.

Esto permite centralizar la información ya que el servidor se puede encontrar absolutamente alejado del usuario que intenta acceder. De modo que el usuario puede estar en un ordenador y el servidor en otro.

conexión local

Se trata de una conexión en la cual el servidor de base de datos y el usuario que intenta conectar están en la misma máquina. No hace falta control de red, pero limita el uso de la base de datos a la máquina en la que el servidor está instalado.

cliente / servidor

Se trata del método más común de trabajo. El servidor de bases de datos lanza un proceso en la máquina central (servidor) desde la que se gestionan las bases de datos. Este proceso está a la escucha de nuevos usuarios, cuando estos llegan se produce una conexión que permite que el servidor y el cliente se comuniquen.

La ventaja de esta implementación reside en que se permite centralizar el sistema de datos, lo que facilita su control. Por otro lado eso permite una accesibilidad a la base de datos desde distintas máquinas

herramientas que proporcionan los RDBMS

Para el uso de las bases de datos, los RDBMS proporcionan diversas herramientas, que además tienen finalidades distintas en función de qué tipo de usuario las utiliza:

- **Herramientas de instalación.** Instaladores para facilitar la tarea de realizar la siempre difícil instalación del producto de base de datos.
- **Herramientas de gestión de red.** Que permiten que el gestor de base de datos sea correctamente accesible desde la red, así como gestionar el correcto flujo de información sobre la red que integra al RDBMS.
- **Herramientas de conexión en el lado del cliente.** Los programas que permiten a los usuarios conectar a la base de datos para lanzar las instrucciones que se deseen.
- **Herramientas de desarrollo.** Que facilitan la labor de crear aplicaciones para una base de datos.
- **Herramientas CASE.** Para poder realizar diseños completos de aplicaciones de datos.
- **Herramientas de administración.** Que permiten una más fácil realización de las tareas administrativas.
- **Herramientas de recuperación en caso de desastre.**

- ⊙ **Herramientas de copia de seguridad**
- ⊙ **Herramientas de importación de datos.**
- ⊙ **Herramientas de globalización**
- ⊙ **Herramientas de creación de aplicaciones hacia Internet**

términos relacionados con los sistemas gestores de bases de datos

- ⊙ **Memoria.** La memoria interna del ordenador (normalmente la RAM). Un recurso muy importante de las bases de datos, ahí se almacenan los últimos datos utilizados.
- ⊙ **Procesos.** Programa que se encuentra en ejecución. Un RDBMS lanza muchos procesos que gestionan todos los subsistemas de control necesarios para el buen funcionamiento de la base de datos.
- ⊙ **Sistema de archivos.** Los dispositivos y software (sistema operativo) encargados de almacenar físicamente la información en un ordenador. En este sistema se almacenan los datos y los programas que forman el gestor de base de datos.
- ⊙ **Red.** Sistema que conecta distintas máquinas. Cada red utiliza un protocolo concreto de comunicaciones, por lo que el RDBMS utilizado debe también utilizar esos protocolos.
- ⊙ **SQL.** El lenguaje de consulta más popular de los sistemas de base de datos relacionales. Permite ejecutar cualquier petición sobre el servidor de bases de datos. Todo RDBMS que se precie permite realizar conexiones que leen e interpretan instrucciones SQL.

aproximación a Oracle

base de datos objeto-relacional

Las bases de datos relacionales incorporan sólo estructuras estáticas de datos, las llamadas tablas. Es suficiente para modelizar problemas de datos muy grandes, pero no permite aplicar técnicas de programación orientada a objetos.

Hay una serie de bases de datos llamadas orientadas a objetos, donde el elemento de trabajo en lugar de la tabla es el objeto, que incorpora datos y procedimientos. Pero este enfoque todavía está en estudio y no parece clara su ventaja sobre el enfoque relacional.

Oracle aprovecha un nuevo enfoque, el objeto-relacional. Es un punto medio entre los dos anteriores, permite a los usuarios utilizar la base de datos de forma relacional, pero incorpora extensiones de las bases de datos orientadas a objetos.

Oracle incluso soporta el enfoque orientado a objetos. El hecho que de permite los tres objetos hace que se pueda trabajar de forma relacional y añadir cualquier mejora orientada a objetos. Pero el núcleo de Oracle sigue estando pensado para el enfoque relacional.

mejoras procedentes del enfoque orientado a objetos

- ⊙ **tipos abstractos.** Permiten crear dominios complejos para almacenar datos en los atributos de las tablas.
- ⊙ **tablas anidadas.** Permiten seleccionar valores dentro de una lista para un atributo. Representan relaciones uno a varios, donde el contenido de la tabla representa la tabla relacionada con la actual. Sirven por ejemplo para elegir un mes de una lista con los doce meses.
- ⊙ **varrays.** Es una colección de valores que permite seleccionar varios valores relacionados con un atributo de una tabla. Por ejemplo permitiría elegir varias asignaturas en el mismo alumno (lo que está en desacuerdo con el modelo relacional, pero es que en realidad la información del *varray* se almacena aparte)
- ⊙ **objetos grandes.** Se trata de los llamados LOB (*large object binary*). Permiten almacenar textos larguísimos, archivos externos (de los que sólo se almacenará la dirección) y datos binarios.
- ⊙ **vistas-objeto.** Permiten crear objetos a partir de las tablas de datos.

breve historia de Oracle

| | |
|------|--|
| 1978 | Primera versión del sistema Oracle para el sistema PDP-11 de Digital Equipment. Se basaba en las tesis de E. F. Codd y se convierte en el primer RDBMS comercial |
| 1980 | Se crea Oracle Systems, empresa propietaria del DBMS Oracle |
| 1981 | Versión 3 de Oracle, escrita en C (2,5 millones de dólares en ventas) |
| 1984 | Versión 4, conexión cliente / servidor en sistema VAX/VMS. 13 millones de ventas |

arquitectura del DBMS Oracle

aproximación a Oracle

| | |
|------|---|
| 1986 | Versión 5 de Oracle, posibilidad de capacidades distribuidas. 55 millones en ventas |
| 1988 | Versión 6, incluye herramienta CASE, 188 millones de ventas |
| 1991 | Se alcanzan los mil millones de dólares |
| 1993 | Versión 7 de Oracle para el sistema Unix. Se convierte en el mejor gestor de bases de datos enormes |
| 1994 | 2000 millones de ventas, versión de Oracle para PC |
| 1997 | Versión 8, se lanza hacia Internet y adopta Java. 5000 millones en ventas |
| 1998 | Versión de Oracle para Linux |
| 1999 | Versión 8i, mayor integración de Java e Internet. 8000 millones en ventas |
| 2000 | Más de 10.000 millones en ventas, primera solución de bases de datos del mercado. |
| 2001 | Aparece la versión 9i con grandes mejoras en la distribución |
| 2003 | Aparece la versión 10g, con grandes mejoras en cuanto al tamaño de almacenamiento y velocidad. |

otros servidores

Los rivales de Oracle fundamentales hoy en día son:

- ⊙ **Ingres de Computer Associates**
- ⊙ **DB/2 de IBM**
- ⊙ **SQL Server de Microsoft**
- ⊙ **ADABAS de Software AG**
- ⊙ **Informix**
- ⊙ **PostgreSQL (de código abierto)**

estrategia de Oracle

Oracle es una empresa que fabrica diversos productos. Su idea es aportar un producto autosuficiente para el mantenimiento de datos y la creación de aplicaciones basados en estos. Sus tres productos más importantes son:

- ⊙ **Oracle DataBase.** El DBMS Oracle, junto con las herramientas fundamentales para hacer de servidor y los programas clientes necesarios para conectar clientes
- ⊙ **Oracle Application Server.** Servidor de aplicaciones para la creación de programas distribuidos.

- ⦿ **Oracle Developer Suite.** Programas para la generación de aplicaciones rápidas basadas en bases de datos Oracle.

servidor Oracle

introducción

Un servidor Oracle es el software que permite una administración y desarrollo de bases de datos. Tiene tres posibilidades de ejecución:

- **Local o basada en host.** El servidor se ejecuta en la misma máquina en la que se conectan los clientes. La versión personal de Oracle database, produce servidores de este tipo.
- **Cliente-Servidor.** Enfoque más típico. El servidor reside en un ordenador distinto respecto al que los usuarios van a usar para conectarse a la base de datos.
- **Cliente-Servidor de Aplicaciones-Servidor.** Los usuarios acceden a un servidor de aplicaciones (**Oracle Application Server**) que, a su vez, accede al servidor Oracle. Los tres elementos (cliente, servidor de aplicaciones, servidor Oracle) pueden estar en tres máquinas distintas.
-

elementos del servidor Oracle

El servidor Oracle está formado por dos elementos:

- **La instancia de la base de datos.** Consta de **datos** (llamados estructuras de memoria) y de **procesos** en memoria (procesos background) necesarios para dar servicio a los usuarios de la base de datos. Puede haber más de una instancia si se distribuye la base de datos en más de una máquina. Cada instancia abre **una y sólo una** base de datos.
- **Ficheros en disco.** Representan la base de datos en sí. Consta de:
 - ◆ **Estructuras lógicas:** Tablespaces, objetos del esquema de usuario.
 - ◆ **Estructuras físicas:** Los ficheros de datos almacenados en disco. Los ficheros de datos (asociados a los *tablespaces*), los ficheros *redo log* y los ficheros de control

conexiones

Para establecer una **sesión** con la base de datos, el usuario necesita conectar con la instancia de la base de datos. Normalmente esto significa arrancar una herramienta cliente como **SQL*Plus** o ejecutar una aplicación de desarrollo de bases de datos (como **Oracle Forms**); entonces se ejecuta un **proceso de usuario**.

Cuando esto ocurre, en el servidor se establece un **proceso de servidor**. Este proceso es el encargado de comunicar al usuario con la instancia Oracle en nombre del proceso de usuario. Cada vez que el usuario ejecuta instrucciones SQL, éstas son transmitidas a la instancia Oracle por el proceso servidor.

De este modo una conexión es un camino entre un proceso de usuario y un servidor Oracle.

Cada **sesión** es una conexión de un usuario con el servidor Oracle. Un usuario puede establecer múltiples sesiones (si se conecta desde diferentes herramientas y máquinas)

estructura de las bases de datos Oracle

Desde el punto de vista de Oracle, una **base de datos** es una colección de datos tratados como una única unidad. Una base de datos Oracle contiene tres tipos de ficheros:

- ⊙ **Archivos de datos.** Contiene los datos actuales de la base de datos así como el diccionario de datos.
- ⊙ **Archivos rehacer (*redo logs*).** Almacenan datos recuperables en caso de error grave.
- ⊙ **Archivos de control.** Necesarios para mantener la integridad de la base de datos.

Además se utilizan otros archivos de forma auxiliar

- ⊙ **Archivos de parámetros.** Que definen algunas características de una instancia Oracle.
- ⊙ **Archivos de contraseñas.** Que sirven para autenticar a los usuarios.
- ⊙ **Copias de archivos rehacer.** Utilizadas para la recuperación de datos.

instancia de la base de datos

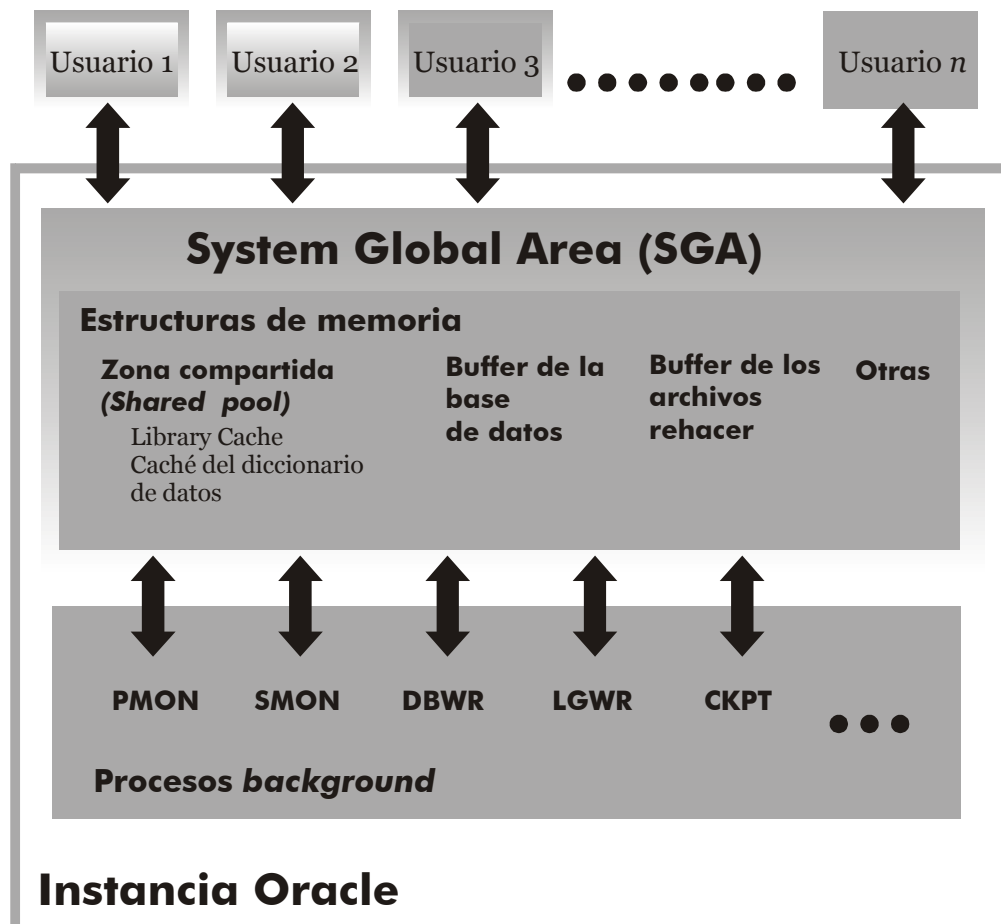


Ilustración 1, Estructura de la instancia de Oracle

arquitectura del DBMS Oracle

servidor Oracle

La instancia de la base de datos es uno de los dos elementos de cualquier base de datos Oracle. Sirve para gestionar los datos de la base de datos y proporcionar servicio a los usuarios que acceden a la misma.

Está compuesta de:

- **Estructuras en memoria.**
- **Procesos en segundo plano (*background*).**

estructuras en memoria

SGA

Es la abreviatura de *System Global Area*, Área Global de Sistema. Está situada al inicio de los datos de la instancia y contiene los datos e información de control de la instancia.

Está formada por las siguientes estructuras:

- **Shared pool, fondo común compartido.** Almacena las últimas instrucciones SQL y PL/SQL ejecutadas. Posee dos estructuras internas:
 - ◆ **Caché de instrucciones (*Library cache*).** Almacena las últimas instrucciones SQL y PL/SQL ejecutadas. Administra los datos mediante algoritmo LRU.
 - ◆ **Caché del diccionario de datos.** Almacena las últimas definiciones de la base de datos utilizadas (tablas, índices, privilegios, usuarios,...) Cada vez que una instrucción utiliza un nombre de la base de datos (tabla, índice,...) se comprueba en el diccionario de datos y se almacena en este caché. De este modo la siguiente vez no hace falta acceder al diccionario de datos real.
- **Caché buffer de la base de datos.** Almacena los últimos bloques de datos accedidos por los usuarios.
- **Buffer de archivo rehacer.** Almacena los últimos cambios realizados a los bloques de datos de la base de datos.
- **Large pool.** Opcional. Se utiliza como memoria de sesión y para realizar operaciones de *backup*.
- **Java pool.** Opcional. Se utiliza como caché de los comandos Java.
- **Otras estructuras**

PGA

Zona global de los programas (*Program Global Area*). En ella se almacenan los datos correspondientes a un proceso (sólo un proceso puede utilizar esta área). Incluye:

- **Áreas de ordenación.** Para acelerar las tareas de ordenación de datos.
- **Información de sesión.** Usuario, privilegios,...
- **Estado del cursor.** Tareas SQL actualmente en ejecución
- **Espacio de pila.** Variables y otros datos

En Oracle los procesos pueden ser de estos tipos:

- **Proceso de usuario.** Lanzado por el usuario para pedir interacción con la base de datos.

- ⊙ **Proceso de servidor.** Hacen de enlace entre los procesos de usuarios y el servidor Oracle. Se utilizan como manejadores de los procesos de usuario. Los comandos de usuario se envían a estos procesos que se encargan de solicitar peticiones a la base de datos mediante el interfaz de programas de Oracle (**OPI**, *Oracle Program Interface*).
- ⊙ **procesos en segundo plano (*background*).** Cada instancia de Oracle arranca una serie de procesos *background*. Los procesos obligatorios son
 - ◇ **DBWR (*DataBase WRiter*).** Proceso encargado de escribir en los ficheros de datos los buffers más antiguos de la memoria, para que la base de datos vaya almacenando los cambios.
 - ◇ **LGWR (*LoG WRiter*).** Escribe los datos a los ficheros rehacer (*redo*) desde la caché de archivos rehacer.
 - ◇ **CKPT.** Actualiza todas las cabeceras de los ficheros de datos para que aparezca la nueva disposición de datos. Esto ocurre cuando se genera un **punto de comprobación**.
 - ◇ **SMON (*System MONitor*).** Permite recuperar la instancia de la base de datos en caso de caída fatal (cuando el sistema falla por ejemplo). Cuando se reinicia de nuevo la instancia de la base de datos,
 - ◇ **PMON (*Process MONitor*).** Es el encargado de gestionar adecuadamente los procesos que fallan. Ante caídas de procesos, PMON se encarga de restaurar los datos adecuadamente.
 - ◇ **SQL *Net Listener.** Es el encargado de encaminar por una red solicitudes de un cliente a un servidor de base de datos Oracle. Este proceso *escuchador (listener)* está tanto en el cliente como en el servidor. Puede encaminar solicitudes que se dirigen a varias instancias.
 - ◇ **Otros**

procesamiento de instrucciones SQL

Para poder ejecutar SQL sobre la base de datos, hay que conectar con la instancia Oracle de la base de datos, lo cual requiere la comunicación entre un proceso cliente y el servidor (el proceso cliente puede ser una instancia de SQL*Plus por ejemplo). Los componentes utilizados por Oracle para procesar el SQL dependen del código enviado:

- ⊙ Las consultas devuelven filas
- ⊙ Las instrucciones DML (Lenguaje de Manipulación de Datos) graban cambios
- ⊙ la instrucción **commit** asegura el proceso de la transacción

Pero de manera general los pasos en ese proceso son:

- 1 > El usuario abre la herramienta que permite el envío de peticiones SQL (por ejemplo SQL*Plus)
- 2 > El usuario introduce su nombre de usuario y contraseña

arquitectura del DBMS Oracle

servidor Oracle

- 3> Oracle consulta el diccionario de datos para verificar la existencia del usuario y para validar su permiso de conexión. Si lo tiene, se produce la conexión
- 4> El usuario escribe la instrucción SQL (supongamos que es una instrucción de modificación)
- 5> Oracle traduce la instrucción con el analizador de instrucciones (devolvería un error si la instrucción no es válida)
- 6> Oracle traduce los nombres usados en la instrucción con la ayuda del diccionario de datos.
- 7> Si es una instrucción de mostrar datos (SELECT), comprueba si otros usuarios han enviado hace poco esa misma instrucción, eso lo comprueba en el caché de instrucciones de la SGA. Si la instrucción está ahí coge los resultados del buffer caché de la base de datos.
- 8> Si la instrucción conlleva cambios, el servidor bloquea las filas que se modificarán.
- 9> La base de datos graba los cambios (si les hubo) y actualiza los archivos deshacer
- 10> La base de datos graba los nuevos valores para los datos
- 11> Oracle libera del bloqueo los registros
- 12> El usuario recibe un mensaje de éxito

conceptos del sistema Oracle

almacenamiento

jerarquía lógica y física de los datos

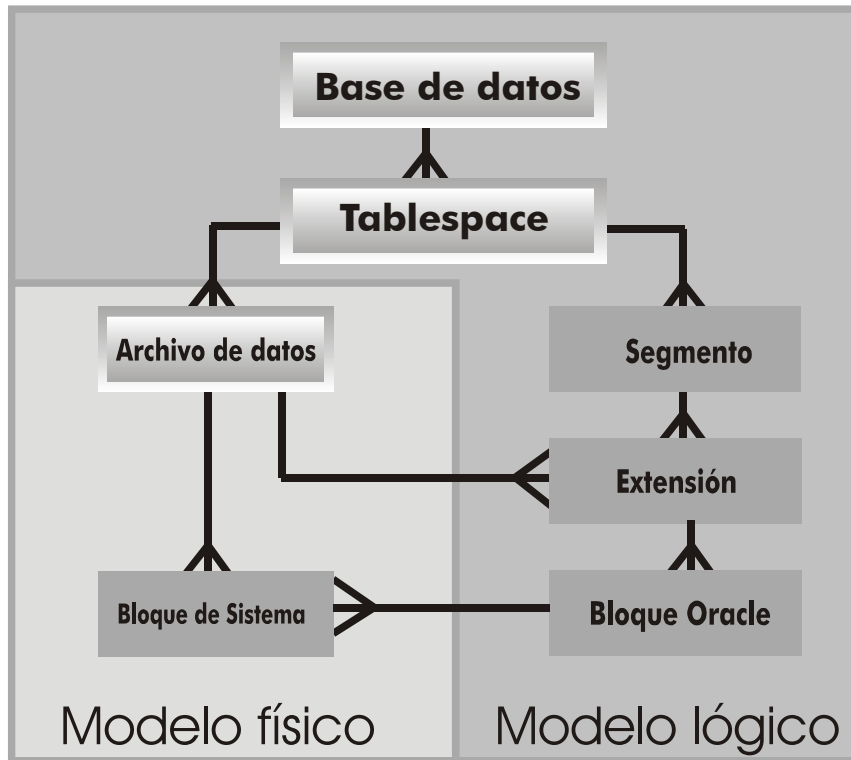


Ilustración 2, Modelo entidad / relación de la estructura de los datos de una base de datos Oracle

Una base de datos tiene una estructura lógica (que se manipula mediante comandos) y una estructura física (la que realmente se almacena en disco).

Estructura lógica

Está formada por:

- ⊙ **Tablespaces.** Pertenecen sólo a una base de datos y sirven para agrupar los datos de la base de datos. Cada tablespace está formado físicamente por uno o más archivos de datos. Están divididos en 0 o más segmentos. Se pueden visualizar en línea o fuera de línea y pueden ser activados en sólo lectura o en lectura / escritura.
- ⊙ **Segmento.** Sirven para almacenar las estructuras lógicas de la base de datos (tablas, índices,...). Un tablespace se compone de uno o más segmentos. Pero el mismo segmento no puede estar en más de un tablespace.
- ⊙ **Extensiones.** División que se hace a cada segmento. El DBA puede añadir o quitar extensiones a los segmentos a fin de hacer que ganen o pierdan espacio.

arquitectura del DBMS Oracle

conceptos del sistema

- **Bloque Oracle o bloque de datos.** Es la unidad mínima de datos para Oracle y se corresponde a una o más unidades de datos mínimas del sistema operativo en el que nos encontremos.

Estructura física

- **Archivos de datos.** Son archivos en disco que sirven para almacenar los datos físicamente (en una unidad de disco). Cada archivo de datos pertenece sólo a un tablespace. Su tamaño se puede gestionar.
- **Bloques de sistema.** La división mínima de los datos que hace el sistema operativo.

transacciones

Los cambios en la base de datos no son guardados hasta que tras una serie de instrucciones se decide llevar a cabo esos cambios. Hasta ese momento todo lo realizado se toma como provisional. Un fallo en la máquina permitiría invertir los cambios.

Una transacción son varias operaciones SQL que forman una unidad de trabajo. Comienza cuando una persona se conecta y de ahí hasta que ejecuta la instrucción **commit** (ejecutar la transacción) o **rollback** (anular la transacción). La anulación deja la base de datos en el estado anterior al comienzo de la transacción. Tras un commit o un rollback comienza la siguiente transacción.

En Oracle se admite además el uso de **puntos de ruptura** (checkpoints) para almacenar valores intermedios y volver a cualquier de ellos si interesa. Pero esto ralentiza excesivamente el sistema.

usuarios

Los usuarios son las personas que acceden de una forma o de otra a la base de datos. Cada usuario tiene una vista determinada de la base de datos. Hay varios conceptos sobre los usuarios a tener en cuenta.

privilegios

Controlan el permiso que posee un usuario de ejecutar una determinada instrucción SQL. Un usuario que quiera crear una tabla, deberá tener el privilegio (o permiso) adecuado para ello.

Además se pueden colocar privilegios en los objetos, es decir, un usuario propietario de una tabla puede otorgar privilegios a esa tabla (se trataría de un privilegio a nivel de objeto) para que haya otros usuarios que la puedan usar.

rol

Son agrupaciones de privilegios que facilitan la tarea de gestionar a los usuarios. Así cuando una serie de usuarios van a tener los mismos privilegios, se crea un rol que contenga esos privilegios y a esos usuarios se les asigna el rol.

Oracle proporciona varios roles ya preparados, por ejemplo el rol DBA da privilegio absoluto a un usuario.

esquemas

Los esquemas están asociados a los usuarios. Agrupan los objetos lógicos que pertenecen al usuario. Es decir es el conjunto de tablas, vistas, sinónimos, instantáneas, enlaces de base de datos, procedimientos y funciones, paquetes,...

Cada usuario tiene su propio esquema y, en principio, un usuario no tiene acceso a los elementos de otro usuario, salvo que sea un administrador o que otro usuario ceda el privilegio de utilización de uno o más de sus objetos al resto de usuarios.

pérdidas de información

Es una de las tareas y herramientas fundamentales que nos proporcionan las bases de datos. Hay posibilidad de perder datos de nuestra base de datos por alguna de estas razones:

- ⊙ **Fallo en una instrucción.** Hay instrucciones que pueden provocar la pérdida no deseada de cientos de registros en un momento (DELETE o UPDATE). En ese caso basta ejecutar un rollback antes de que la instrucción se lleve a cabo.
- ⊙ **Fallo en la comunicación.** Si una conexión de usuario se corta anulando el proceso de usuario relacionado. En ese caso, Oracle anula los cambios de la última transacción (el resto de transacciones sí se almacena).
- ⊙ **Caída del servidor.** Puede que la instancia Oracle se deje de ejecutar. En ese caso basta con lanzar de nuevo la instancia. El proceso SMON se encargará de grabar los archivos rehacer y aplica de nuevo las transacciones confirmadas. Se anulan los cambios no confirmados.
- ⊙ **Pérdida de datos en los archivos.** Es el único caso en el que tiene que intervenir el administrador. La única posibilidad es recuperarles de una copia de seguridad

copias de seguridad

Es una de las herramientas fundamentales de toda base de datos. Al hacer la copia de seguridad se nos permite recuperar la información de esa copia. Eso hace que perdamos definitivamente los datos perdidos desde la última copia de seguridad, de ahí la importancia de hacer copia a menudo.

Hay una posibilidad de perder menos información y es hacer que la base de datos se ejecute en modo Archivo Log, lo que significa que se almacena en un archivo de datos especial, los datos que se van rechazando en los registros rehacer por estar llenos. Lo malo de esta opción, es que el servidor funciona más lento; lo bueno es que en caso de desastre se pueden recuperar los datos almacenados.

Hay dos tipos de copia de seguridad:

- ⊙ **En frío.** La copia se realiza tras cortar la instancia de Oracle. La copia de seguridad almacena todos los datos de la base (incluidos los archivos de control y de rehacer).
- ⊙ **En caliente.** Ya que en muchos casos no se puede parar la instancia tan fácilmente (por estar 24 horas al día funcionando). En ese caso es una indicación la que se hace a la base de datos y la copia se realiza desde el sistema operativo (copiando sin más).

Otras posibilidades de copias son:

arquitectura del DBMS Oracle

conceptos del sistema

- ⊙ **Catálogos de copia de seguridad.** Que almacenan información sobre las copias realizadas, fechas, datos, estructura,...
- ⊙ **Copias de seguridad incrementales.** Un problema de las copias es que en bases extensas, la copia tarda muchas horas. Cuando la copia termina la información ya está desfasada. Las copias incrementales sólo almacenan los datos que han cambiado recientemente.

bases de datos distribuidas

Se trata de una base de datos a nivel lógico (los usuarios la manejan como una base de datos normal), pero que en realidad (físicamente) está implementada en varias ubicaciones físicas, incluso en máquinas diferentes y distantes.

Cada máquina ejecuta su propia instancia y conjuntos de archivos y todas se conectan en red para hacer que el usuario no tenga que cambiar su código para reflejar esta distribución. La dificultad de esta estructura suele estar aliviada por medio de instantáneas que graban momentáneamente los datos de las tablas distantes. Permiten trabajar con los datos copiados y se programan para que cada cierto tiempo recojan nuevamente los datos a fin de reflejar sus cambios.

Gracias a las instantáneas no hace falta una sobrecarga tan excesiva de las instantáneas de la base de datos.

herramientas de Oracle

El software del sistema de bases de datos Oracle incorpora herramientas para realizar la mayoría de tareas comunes en una base de datos:

- ⊙ **Oracle Universal Installer.** gestor de instalaciones, controla cada nueva instalación de software Oracle a fin de que se integren de la mejor manera posible
- ⊙ **SQL*plus.** Programa cliente que permite conexión con el servidor Oracle para enviarle secuencias SQL y PL/SQL
- ⊙ **iSQL*plus.** Permite conexiones al servidor Oracle con la misma finalidad que el anterior pero utilizando una navegador de Internet, lo que facilita el trabajo
- ⊙ **SQL*plus WorkSheet.** Permite conexiones al servidor de Oracle, utilizando un entorno más potente (procede del Oracle Enterprise Manager)
- ⊙ **Oracle Enterprise Manager.** Entorno que permite la administración y configuración completa del servidor Oracle.
- ⊙ **SQL*Loader.** Permite cargar en bases de datos de Oracle información que procede de un archivo de texto. Necesaria para utilizar en las bases de datos de Oracle, información que procede de otro software.
- ⊙ **Import/Export.** Para importar y exportar datos entre instancias de Oracle. De un servidor a otro por ejemplo. también se utiliza como herramienta de copia de seguridad.
- ⊙ **Servidor http de Oracle.** Basado en el servidor Apache, permite opciones de documentación y sobre todo la comunicación directa a través de iSQL*Plus con el servidor Oracle sin tener necesidad de instalar software adicional.

- ⊙ **net Manager.** Permite la administración de los servicios de red a fin de configurar las conexiones hacia instancias de Oracle.
- ⊙ **Oracle Forms.** Permite crear aplicaciones visuales sobre bases de datos de Oracle
- ⊙ **Oracle Reports.** Asistente para la producción de informes
- ⊙ **Oracle Designer.** Herramienta CASE de Oracle, para crear esquemas en el ordenador y que el software produzca los resultados del mismo
- ⊙ **Oracle JDeveloper.** Crea aplicaciones Java pensadas para desarrollar formularios sobre datos de Oracle
- ⊙ **Oracle Developer Suite.** Integra todos los componentes anteriores
- ⊙ **Oracle AS (*Application Server*).** Servidor de aplicaciones de Oracle. Permite compilar aplicaciones J2EE
- ⊙ **Pro C/C++** Precompilador de C/C++ para Oracle