LENGUAJES Y
CIENCIAS DE LA
COMPUTACIÓN
UNIVERSIDAD DE MÁLAGA

Administración de Bases de Datos

Tema 3. Nivel Físico de una Base de Datos (Segunda Parte)

Índice

1. El SGBD ORACLE
2. Tablespaces y Datafiles
3. Vistas Dinámicas del Diccionario
4. Gestión del Espacio Lógico
   a. Bloques de Datos, Extensiones y Segmentos
5. Estructura de la Memoria
6. Estructura de los Procesos
7. Objetos del Esquema. Tablas, Clusters, Índices
8. Administración del SGBD ORACLE
9. Herramientas
10. Iniciar/finalizar ORACLE

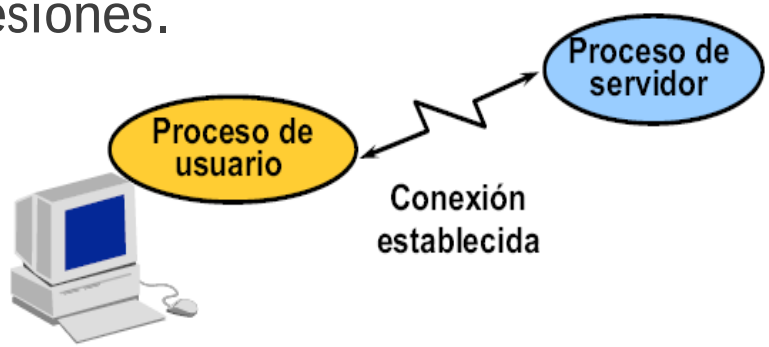
6. Estructura de los Procesos

Estructura de los PROCESOS

- **Todo usuario** que se conecta a Oracle debe **ejecutar dos módulos** de código para acceder a la instancia de la BD Oracle, que se ejecutan como procesos normales bajo el control del S.O.**:**
  - **Aplicación o Herramienta Oracle:** Se trata de una aplicación normal que se conecte a Oracle, o bien una herramienta de Oracle (como Oracle Enterprise Manager o SQL*Plus).
    - Estos programas permiten ejecutar sentencias SQL.
  - **Código del Servidor de Oracle:** Es una parte de Oracle que se ejecuta para el usuario y que interpreta y procesa las órdenes SQL.
- **Sistemas Oracle Multiproceso:** Distintos procesos ejecutan distintas partes de Oracle, además de los procesos de los usuarios.
  - La mayoría de los SGBD son **multiusuario**, ya que es una de sus principales ventajas.
  - Dividiendo el trabajo en **distintos procesos** se consigue un mejor rendimiento dando servicio a múltiples usuarios y aplicaciones.
- **Tipos de Procesos:**
  - Procesos de Usuario (*User Processes*).
  - Procesos de Oracle (Oracle Processes):
    1. Proc. de Servidor (*server*)
    2. Proc. en 2º Plano, o de Fondo (*background*)

Estructura de los PROCESOS

- **Procesos de Usuario:** Cuando el usuario ejecuta una aplicación, o una herramienta de Oracle, se crea un proceso de usuario.
  - **Conexión:** Es una vía de comunicación entre un proceso de usuario y una Instancia. Establece el mecanismo de comunicación.
  - **Sesión:** Es una conexión específica de un usuario a una Instancia a través de un proceso de usuario.
    - Por ejemplo, un usuario establece una **conexión** usando SQL*Plus, después introduce su *username* y *password* y, posteriormente, inicia una **sesión** (a través de la ejecución de un proceso de usuario).
    - Un mismo usuario puede tener varias sesiones.



Estructura de los PROCESOS

- **Procesos de Oracle:** Pueden ser de 2 tipos:
  - **1. Procesos de Servidor:** Se crean para cada una de las aplicaciones de usuario
  - **2. Procesos de *Background*:** Para maximizar el rendimiento y posibilitar el acceso simultáneo de múltiples usuarios
    - Algunos de estos procesos son creados automáticamente cuando se inicia una **Instancia**. No todos ellos están siempre presentes.

Estructura de los PROCESOS

- Procesos de Background:
  - **Database Writer (DBW0 o DBWn):** Escribe el contenido de los *buffers* que han sido modificados a los *datafiles*.
  - **Log Writer (LGWR):** Es el responsable del funcionamiento de los *buffers* del *Redo Log* mediante la escritura de su contenido al fichero de *Redo Log*.
  - **Checkpoint (CKPT):** Cuando se realiza un *checkpoint*, Oracle actualiza las cabeceras de todos los *datafiles* que almacenan datos a causa de este *checkpoint*.
  - **System Monitor (SMON):** Este proceso tiene dos funciones:
    - Maneja la recuperación de la BD a partir del fallo de una Instancia (esto es, cuando las estructuras de memoria y los procesos que componen la Instancia no pueden continuar por algún motivo).
    - Chequea periódicamente los espacios de disco para determinar si une pequeños fragmentos de espacios libres.

Estructura de los PROCESOS

- **Process Monitor (PMON):** Actúa cuando falla un proceso de usuario, liberando los recursos que tuviera asignados (memoria...) y deshaciendo los cambios que hubiese realizado desde su último **`COMMIT`**.
- **Recoverer (RECO):** Es un proceso de *background* opcional para las BD distribuidas que gestiona las transacciones distribuidas.
- **Archiver (ARCH):** Es opcional. Copia los ficheros del *Redo Log* a un dispositivo predeterminado cuando éste está lleno. Este proceso solamente se presenta cuando el *Redo Log* se usa en modo **`ARCHIVELOG`** y el archivo automático está activado.
- **Lock (LCKn):** Opcional. Solamente para servidores paralelos. Gestiona los bloqueos entre distintas Instancias.
- **Job Queue (SNPn):** Opcional, para BD distribuidas.
- **Queue Monitor (QMNn):** Opcional. Monitoriza el orden de salida de los mensajes.
- **Dispatcher (Dnnn):** Opcional. Permite a los procesos de usuario compartir procesos del servidor, de manera que el servidor pueda soportar un mayor número de usuarios.
- **Shared Server (Snnn):** Cada uno de estos procesos sirve las peticiones de múltiples clientes en configuraciones como la anterior, compartiendo los procesos del servidor.

Estructura de los PROCESOS

- **Listener**:
  - El listener es un proceso que escucha peticiones de conexión de cliente.
  - Se configura en el fichero `listener.ora`, con una dirección de protocolo que identifica la base de datos
  - Por ejemplo:

```
(DESCRIPTION=
(ADDRESS=(PROTOCOL=tcp)(HOST=my-server) (PORT=1521)))
```

  - Además, el listener conecta al usuario a los dispatchers o servidores dedicados (el listener es parte de Oracle Net Services, no de Oracle).
  - Cuando una instancia arranca, el listener establece una ruta de comunicación con Oracle.
  - También puede establecer una comunicación entre bases de datos
  - En bases de datos distribuidas sirve para definir servicios y las instancias que sirven esos servicios
  - Arrancar|parar el listener:
    - Lsnrctl start | stop
    - En Windows también se puede hacer arrancando/parando el servicio

Desde el S.O.

7. Objetos del Esquema. Tablas, Índices, Clusters

TABLAS

- Unidad básica de organización de datos
- Organización:
  - Heap. Las filas no se guardan en ningún orden
  - IOT (Index-Organized Table). Las filas se guardan en el orden de la clave primaria
  - Tablas Externas. Sus metadatos se guardan en la base de datos, pero sus datos, no.
- Los datos pueden ser permanentes o temporales
- Una tabla puede tener columnas virtuales:

```
create table productos (
codigo number primary key,
descripcion varchar2(50),
precio_neto number (9,2),
tipo_iva number(2),
total as (precio_neto+ precio_neto * tipo_iva /100))
```

- Una tabla puede estar partida → PARTITION en función de diversos criterios. Cada partición puede ir a un tablespace distinto

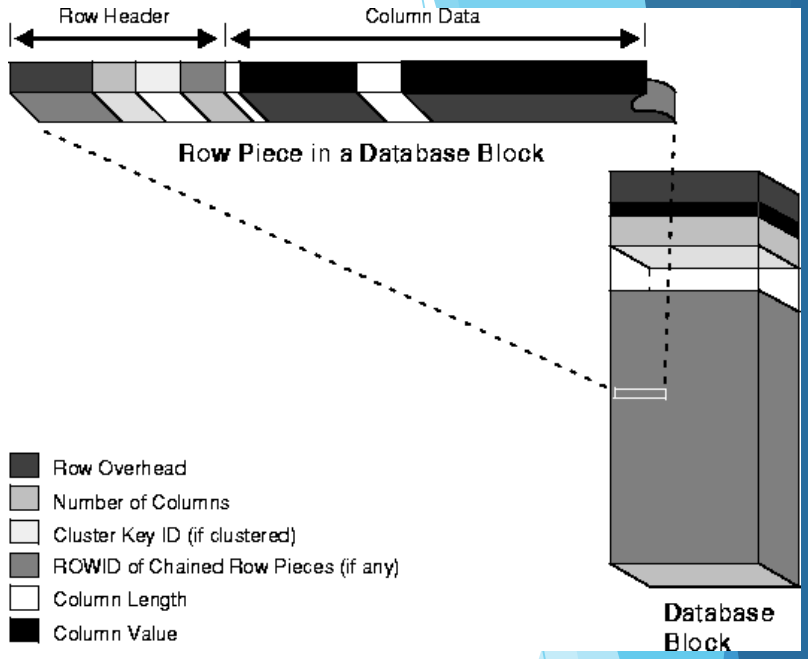
TABLAS

Cuando se **crea**, Oracle asigna un segmento de datos de un *tablespace* para contener los futuros datos de la tabla. Podemos controlar la generación de este espacio a partir de los parámetros correspondientes (**PCTFREE**, **PCTUSED**...).

Oracle almacena habitualmente cada **fila** de una tabla en una única página (***row piece***)**:**

Una fila se fragmenta en varias páginas (varios ***row pieces***) si la fila no puede ser almacenada completamente en la página: Se usan varias páginas, encadenándolas.

Cada fila de una tabla se almacena en dos partes: una que contiene información de cabecera (***row header***) y otra con el contenido de los datos (***column data***).



- ***Row header:*** En una fila de un único bloque ocupa 3 bytes como mínimo.
- ***Column data*****:** Almacena la longitud de cada columna y sus datos.
  - El valor **NULL** se representa almacenando cero en la longitud y nada en los datos.
- Borrar una columna es costoso, pero si se usa la opción **SET UNUSED** de **ALTER TABLE**, sólo marca la columna como borrada posponiendo el borrado.

TABLAS TEMPORALES

- Sus **datos** duran mientras dure la **transacción o sesión** actuales
- Los **metadatos** son permanentes (su definición no se borra). Por tanto, sí se pueden crear vistas, triggers, índices sobre las tablas temporales
- Sentencia de creación:

```
CREATE GLOBAL TEMPORARY TABLE <tabla>
   ON COMMIT [DELETE|PRESERVE] ROWS;
```

- Debe especificarse si son datos específicos de la transacción o de la sesión:
  - **ON COMMIT DELETE ROWS:** Borra el contenido de la tabla al efectuar un **COMMIT**, es decir, los datos son específicos de la transacción.
  - **ON COMMIT PRESERVE ROWS:** NO borra el contenido de la tabla al hacer **COMMIT**, sino que los borra al finalizar la sesión.
  - Son datos privados de cada sesión: No pueden compartirse.

TABLAS EXTERNAS

- Accede a datos en fuentes externas como si los datos estuvieran en una tabla de la base de datos. Se puede utilizar SQL, PL/SQL y Java para consultar los datos externos.
- Se puede crear una tabla externa, copiar el archivo en la ubicación especificada en la definición de la tabla externa, y utilizar SQL para consultar los registros en el archivo de texto
- Se usan mucho para Data Warehouse en operaciones ETL (Extract, Transform, Load)
- Se usan sobre todo para leer datos, pero con los ***drivers*** adecuados, también se puede escribir

```
CREATE TABLE (…) ORGANIZATION EXTERNAL (datos del
  fichero)
```

TABLAS ORGANIZADAS POR ÍNDICE

- **IOT (*index-organized tables*):** Son tablas en las que los datos están contenidos en el índice asociado (*B-tree*).
  - Con la sentencia **`CREATE TABLE`** y su cláusula **`ORGANIZATION INDEX`**.
  - Las **modificaciones** en los datos de la tabla (insertar, actualizar o borrar) tienen como efecto una actualización del índice.
  - Realmente, **el índice es la tabla:** En vez de estar compuesto de valor de clave y puntero (**`ROWID`**), se compone de **valor de clave y resto de valores de la fila**.
    - No duplica el almacenamiento de las claves como en una tabla ordinaria con su índice.
  - Son idóneas para **accesos por clave primaria** pero no recomendadas para otro tipo de accesos.
  - Pueden crearse **índices adicionales** sobre este tipo de tablas para acceder eficientemente por otras columnas.
  - **Diferencias principales** entre estas tablas y las tablas ordinarias**:**

Tabla Ordinaria	Tablas Organizadas por Índice
`ROWID` identifica una fila.	La llave primaria identifica una fila.
Llave primaria opcional.	Llave primaria obligatoria.
Acceso por el **`ROWID`**.	Acceso por la llave primaria.
Análisis secuencial para recuperar todas las filas.	Un análisis completo del índice recupera todas las filas ordenadas por la PK.
Pueden almacenarse en un cluster.	No pueden almacenarse en un cluster.
Pueden contener columnas de tipos **`LONG`** y LOB.	Pueden contener columnas LOB, pero no **`LONG`**.

ÍNDICES

- Estructura opcional asociada con una tabla o un *cluster*.
  - Se crean sobre **una o varias columnas**, para acelerar la ejecución de sentencias SQL.
    - Tras cada columna puede especificarse **`ASC`** o **`DESC`**.
  - Una tabla puede tener cualquier **cantidad de índices**, si la combinación de columnas en cada uno sea diferente. Incluso, cambiando el orden:
    - Ejemplos:
      - **`CREATE INDEX Pieza_idx1 ON Pieza (Nombre, Cantidad);`**
      - **`CREATE INDEX Pieza_idx2 ON Pieza (Cantidad, Nombre);`**
  - Podemos utilizar distintos **Tipos de Índices** que permiten una funcionalidad distinta de cara al rendimiento de la BD**:** Árboles B sobre las tablas, árboles B sobre los clusters, índices *hash* sobre clusters, índices de clave inversa e índices de mapas de bits.
  - Los índices son lógica y físicamente **Independientes de los Datos** de las tablas asociadas y son mantenidos dinámica y **automáticamente**.
    - Se puede crear o borrar un índice sin ningún efecto lateral sobre los datos de la tabla u otros índices.

ÍNDICES

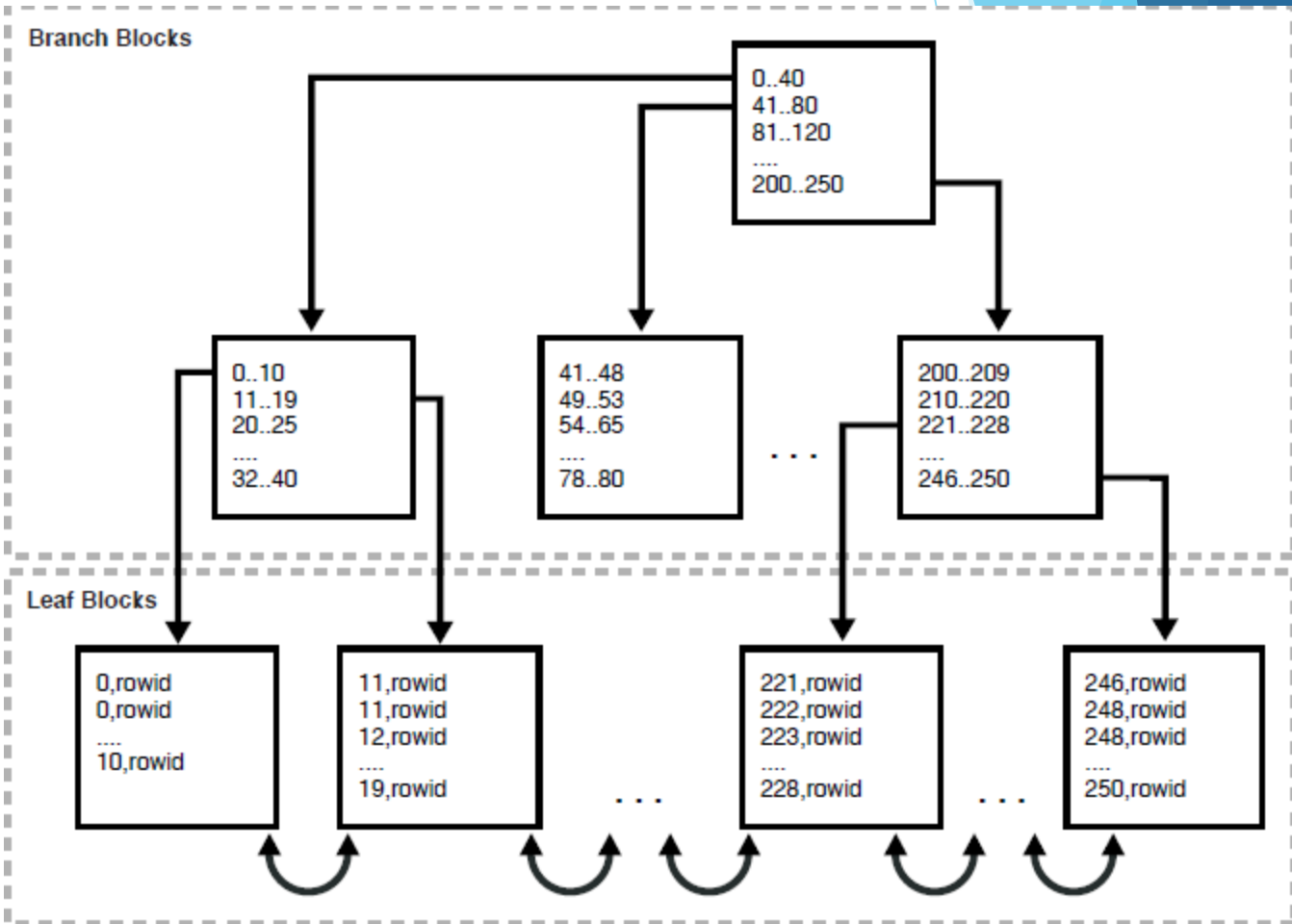
- Pueden ser **únicos** (*unique*) o **no únicos** (*nonunique*), según exijan o no que las columnas del índice admitan o no valores duplicados en distintas filas: **`CREATE [UNIQUE] INDEX...`**
  - Si esa restricción existe en la tabla (**`UNIQUE`**), Oracle crea un índice único automáticamente.
    - Oracle no recomienda crear índices únicos explícitamente.
- Es recomendable que sean **únicos** y que, al menos, se cree **uno por cada clave primaria o externa** de cada tabla, así como por cada columna que contenga valores de búsqueda usuales, sin excedernos.

ÍNDICES

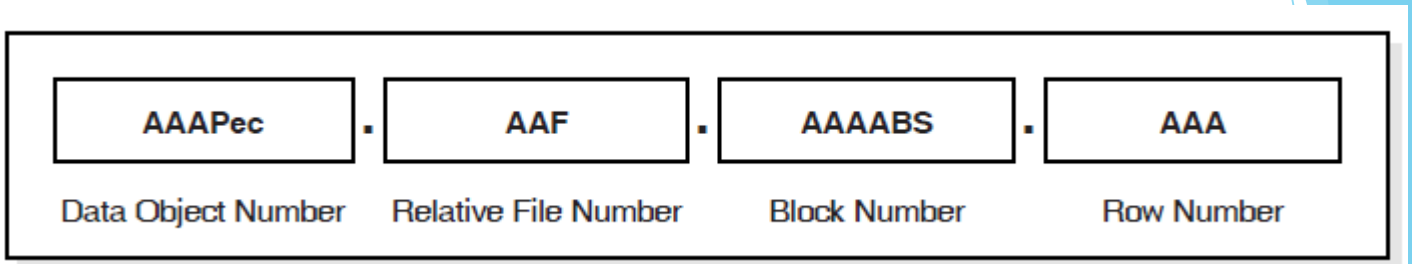
- **Índices Basados en Función (*function-based indexes*):** Se puede construir un índice sobre una función determinista definida por el usuario.
  - **Ej. 1:** Índice que será usado en consultas como la propuesta:
    - `CREATE INDEX idx ON table_1 (a + b * (c - 1), a, b);`
    - `SELECT a FROM table_1 WHERE a + b * (c - 1) < 100;`
  - **Ej. 2:** Para facilitar búsquedas insensibles a mayúsculas/minúsculas:
    - `CREATE INDEX uppercase_idx ON Pieza (UPPER(Nombre));`
    - `SELECT * FROM Pieza WHERE UPPER(Nombre) = 'TORNILLO';`
- **Cuando se Crea un Índice:**
  - Se le asigna un segmento de índice para contener sus valores en el *tablespace* correspondiente.
    - Es preferible que este *tablespace* no sea el mismo en el que está contenida la tabla asociada y que ambos *tablespaces* estén almacenados en discos diferentes, para que Oracle pueda leerlos en paralelo.
  - Al crear un índice, Oracle ordena las columnas del índice y almacena el valor de los índices junto con el **ROWID** de las filas.
  - Los índices pueden crearse en orden ascendente (**ASC**), descendente (**DESC**), comprimidos (**COMPRESS**) o no comprimidos (**NOCOMPRESS**).

Cómo se crean los índices: Árboles B

- Branch Blocks: Para la búsqueda
- Leaf Blocks: Almacena los valores



Formato del ROWID



- Data Object Number: Identifica el segmento
- Relative File number: Identifica el Datafile dentro del Tablespace
- Block Number: Número de datablock dentro del datafile
- Row Number: Fila dentro del datablock

ÍNDICES DE MAPAS DE BITS

- **(CREATE BITMAP INDEX)**
  - Para cada valor de clave se utiliza un **mapa de bits en vez de el ROWID.**
  - Cada **bit del mapa de bits** corresponde a un **ROWID**, y vale **1** si la fila pertenece a ese valor clave, y **0** si no pertenece.
    - Una función de mapeo convierte la posición del bit en un **ROWID**

Ejemplo: - TABLA:

CODIGO	E_CIVIL	...	PROVINCIA
101	Soltero	...	Málaga
102	Casado	...	Madrid
103	Soltero	...	Barcelona
104	Divorciado	...	Barcelona
105	Soltero	...	Madrid
106	Casado	...	Madrid

- ÍNDICE:

VALOR	MAPA DE BITS					
Barcelona	0	0	1	1	0	0
Madrid	0	1	0	0	1	1
Málaga	1	0	0	0	0	0

- Si el número de valores distintos de clave es pequeño (la cardinalidad de la columna es baja), los índices de mapas de bits son muy eficientes:
- Un índice **no** puede ser **BITMAP** y **UNIQUE** a la vez.
- En general, para columnas cuyos valores se repiten más de cien veces.
- Para columnas con valores con un bajo índice de repetición, también se muestra muy eficiente en casos en los que la definición del índice responde a **numerosas condiciones** de una cláusula **WHERE**, ya que las filas que satisfacen algunas, pero no todas las condiciones, son filtradas y desconsideradas antes de que la tabla sea accedida.

Uso de índices

- Oracle utiliza automáticamente cada índice cuando lo necesita y lo actualiza cuando que se modifica la tabla
- Para saber si lo está usando utilizar EXPLAIN PLAN:

```
explain plan for select * from pieza where upper(nombre)='TUERCA';
select * from table(dbms_xplan.display);
Plan hash value: 2998577752
```

```
-------------------------------------------------------------------------------------------
| Id  | Operation                    | Name          | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |               |     1 |    20 |     2   (0)| 00:00:01 |
|   1 |  TABLE ACCESS BY INDEX ROWID| PIEZA         |     1 |    20 |     2   (0)| 00:00:01 |
|*  2 |   INDEX RANGE SCAN           | UPPERCASE_IDX |     1 |       |     1   (0)| 00:00:01 |
-------------------------------------------------------------------------------------------

 Predicate Information (identified by operation id):
---------------------------------------------------

    2 - access(UPPER("NOMBRE")='TUERCA')
```

CLUSTERS

- Grupo de tablas que comparten las mismas páginas porque tienen alguna columna en común y suelen usarse juntas.
  - **Ejemplo:** Si construimos un *cluster* con las tablas **EMPLEADOS** y **DEPARTAMENTOS** utilizando la columna común **CODIGO_DEPARTAMENTO**, los empleados de un mismo departamento, junto con los datos del departamento, compartirían las mismas páginas, y cada valor clave del *cluster* se almacenará sólo una vez.
    - Si cambia el valor clave del *cluster* para una fila, Oracle la realojará.
- **Pueden ser:**
  - ***Cluster de índice***, para las columnas de la clave del *cluster*.
  - ***Cluster HASH*:** Se accede a los datos mediante una **Función *hash*:** Función que se aplica a la columna o columnas de la clave del *cluster*, para obtener la página que corresponde a las filas de esa clave.
    - Esto ahorra tener que leer el índice para localizar o insertar un dato, ya que la función *hash* NO requiere lecturas de disco.

8. Administración del SGBD ORACLE

Administración del SGBD Oracle

- **Funciones del Administrador o DBA:**
  - **Instalación y actualización del *software* de Oracle** (servidor y aplicaciones).
    - Cambiar claves iniciales de las 2 cuentas DBA que Oracle crea automáticamente al crear una BD**:**
      - **SYS:** Todas las tablas y vistas del diccionario de datos pertenecen al esquema de **SYS**, y nadie debería modificarlas. Tampoco se deben crear nuevas tablas de la BD en las cuentas del DBA.
      - **SYSTEM:** Crea nuevas tablas y vistas con información administrativa.
  - **Evaluación del *hardware*:** Evaluar discos, memoria… asignar espacios de almacenamiento y planificar requerimientos futuros.
  - Planificación de los **parámetros de creación** de la BD.
  - **Creación de la BD:**
    - Estructuras de almacenamiento (*tablespaces*…) .
    - Implementación del diseño de la BD: Objetos (tablas, vistas…) y restricciones.
  - Modificar la **estructura de la BD** cuando sea necesario.
  - **Apertura y cierre** de la BD.
  - **Gestión de usuarios** y Sistemas de **seguridad:** Permisos y Roles.
  - **Auditoría:** Controlar y monitorizar el acceso a la BD.
  - **Copias de seguridad** (*backup*) y sus recuperaciones (*recovery*).
  - **Afinamiento** de la BD (optimizar su rendimiento).

Administración del SGBD Oracle

- **Utilidades de Administración:**
  - **SQL*Loader:** Se usa para cargar datos desde ficheros del S.O. a tablas de la base de datos.
    - Puede usarse por usuarios y administradores.
    - Permite especificar el formato de entrada de los datos.
  - **Export e Import:** Permiten mover datos entre distintas BD Oracle.
    - **Export** guarda los datos en ficheros, e **Import** lee esos ficheros y carga los datos en las tablas: Pueden usarse como medios para *backup*.
    - Puede ser entre versiones de Oracle de distintos S.O.
- Ver las **Versiones de los distintos productos Oracle** (núcleo, PL/SQL...): Se puede hacer consultando la vista del diccionario de datos llamada:

  `PRODUCT_COMPONENT_VERSION`.

Administración del SGBD Oracle

- Existen **Dos Privilegios Importantes** para la Administración (no son roles)**:**
  - **SYSOPER:** Puede hacer casi todas las tareas de administración, excepto conceder la administración a otros, crear una BD y poco más.
  - **SYSDBA:** Contiene todos los privilegios del sistema **WITH ADMIN OPTION** (incluyendo **SYSOPER**) y permite usar **CREATE DATABASE**.
  - Se **conceden** normalmente : **GRANT SYSDBA TO scott;**
  - Y se **revocan** de similar forma : **REVOKE SYSDBA FROM scott;**
  - Usuario se puede **conectar** con**: CONNECT scott/tiger AS SYSDBA**
  - Se conecta al esquema por defecto (**PUBLIC** y **SYS** respectivamente), y no al esquema asociado al usuario, por lo que éste no podrá ver sus tablas sin cualificarlas con su nombre de usuario.

Crear una BD Oracle

- **Creación de la Base de Datos:** Pasos para crear una BD Oracle:
  - **Realizar Copias de Seguridad de otras posibles BD preexistentes.**
  - **Crear el Fichero de Parámetros:** Las instancias se arrancan a partir de un fichero de parámetros.
    - Para cada base de datos, debe tenerse un fichero de parámetros de este tipo.
      - Oracle suministra un fichero de parámetros de inicialización por defecto que puede ser editado y modificado para cada base de datos.
    - **Los parámetros son:**
      - **DB_NAME:** Nombre local de la base de datos que no puede ser cambiado.
      - **DB_DOMAIN:** Localización lógica de la BD en la red (por ejemplo SCI.UMA.ES). En combinación con **DB_NAME** debe identificar un único nombre en la red.
      - **CONTROL_FILES:** Si no especifica nada, Oracle creará uno por defecto. Se recomienda tener al menos dos ficheros de control en distintos discos.
      - **DB_BLOCK_SIZE:** Es el tamaño de página de la BD. Por defecto es el mismo tamaño que el de un bloque del S.O. (4096 ó 8192 bytes).
      - **PROCESSES:** Máximo número de procesos que pueden conectarse a la BD simultáneamente. Debe incluir los 5 procesos de *background* más uno por cada usuario. Si se estiman 50 usuarios concurrentes como máximo, este valor puede ser de 55.
      - **ROLLBACK_SEGMENTS:** Es una lista de los segmentos de *rollback* que la Instancia asigna cuando arranca la BD.

Fichero de parámetros

- Server Parameter File (SPFILE). Por defecto.
  - Sólo uno por base de datos. Debe residir en el mismo host que la base de datos
  - Leído y escrito por el SGBD (no por aplicaciones cliente)
  - Es binario y no puede ser modificado externamente
  - Se puede modificar un parámetro con un comando SQL y almacenarlo en el SPFILE
    `ALTER SYSTEM SET parametro=valor SCOPE=SPFILE;`
- Fichero de Parámetros de Texto (PFILE)
  - Fichero de texto con una lista de parámetros y sus valores
  - Puede residir en el ordenador donde se ejecute la aplicación cliente que arranca la BD
  - Para modificarlo se hace desde editor (ALTER SYSTEM no lo modifica)
- Los parámetros estáticos de la base de datos se han de modificar en los ficheros de parámetros
- Crear PFILE una vez arrancada la BD con el SPFILE:
  `Create pfile from spfile`

Crear una BD Oracle (continuación)

- **Arrancar la Instancia:** Mediante la instrucción **STARTUP** de SQL*Plus
- **Crear la BD:** Se hace mediante la instrucción **CREATE DATABASE**, lo que provoca que se realicen las siguientes operaciones automáticamente:
  - Crea los ficheros de datos para la BD (*datafiles*)**: ALTER TABLESPACE**
  - Crea los ficheros de control (*control files*)**: CREATE CONTROL FILE**
  - Crea los registros de rehacer (*redo log*).
  - Crea el *tablespace* SYSTEM y el segmento de *rollback* SYSTEM**: CREATE TABLESPACE**
  - Crea el diccionario de datos.
  - Crea a los usuarios **SYS** y **SYSTEM**.
  - Especifica el conjunto de caracteres que se almacenarán.
  - Monta y abre la BD para su uso.
- **Realizar una Copia de Seguridad de la BD.**

Algunas Sentencias SQL del DBA

- El **Comando CREATE** para **Crear** objetos puede sustituirse por **DROP** y **ALTER** para las **Borrar** y **Modificar** el objeto en cuestión**:**
  - **CREATE USER:** Crea un usuario, una cuenta para acceder a la BD.
  - **CREATE ROLE:** Crea un conjunto de privilegios con un nombre.
  - **CREATE SYNONYM:** Crea un sinónimo. Puede establecerse como sinónimo público (sinónimo accesible para todos los usuarios).
  - **CREATE TABLESPACE:** Crea un *tablespace*, espacio en la BD que puede contener objetos.
  - **CREATE ROLLBACK SEGMENT:** Crea un segmento de anulación (*rollback*), un objeto donde Oracle almacena los datos para deshacer modificaciones.
  - **GRANT:** Otorga roles y permisos (o privilegios) del sistema o de objetos a usuarios. Los privilegios se retiran con el comando **REVOKE**.
  - **ANALYZE:** Almacena o borra en el Diccionario de Datos estadísticas sobre el objeto que se especifique. Por ejemplo, para una tabla el resultado se almacenará en **USER_TABLES**.
  - **AUDIT:** Realiza un seguimiento sobre las operaciones ejecutadas o sobre objetos accedidos (usuario, tipo de operación, objeto implicado, fecha y hora). Para detener la auditoria usar **NOAUDIT**. Los datos se guardan en tablas del diccionario con el texto **AUDIT_** en su nombre, como **DBA_AUDIT_OBJECT**, **DBA_AUDIT_TRAIL**...

9. Herramientas

CULTURA GENERAL
- NO ES IMPORTANTE PARA VOSOTROS-

Herramientas ORACLE

- **Enterprise Manager**
  - Tareas administrativas: crear objetos del esquema (tablespaces, tablas e índices),
  - Manejar seguridad de usuarios, backup, y recuperación, importación/exportación de datos.
  - Visualizar estado de rendimiento.
  - http://hostname:portnumber/em

CULTURA GENERAL
- NO ES IMPORTANTE PARA VOSOTROS-

Enterprise Manager

Almacenamiento

Archivos de Control, Tablespaces, Grupos de Tablespaces, Temporales, Archivos de Datos, Segmentos de Rollback, Grupos de Redo Logs, Archive Logs

Configuración de la Base de Datos

Parámetros de Memoria, Gestión de Deshacer, Todos los Parámetros de Inicialización, Uso de Funciones de la Base de Datos

Planificador de Base de Datos

Trabajos, Cadenas, Planificaciones, Programas, Clases de Trabajos, Ventanas, Grupos de Ventanas, Atributos Globales

Gestión de Estadísticas

Repositorio de Carga de Trabajo Automática, Gestionar Estadísticas del Optimizador

Cambiar Base de Datos

Migrar a ASM, Gestionar Tablespace Localmente

Gestor de Recursos

Monitores, Grupos de Consumidores, Asignaciones de Grupos de Consumidores Planes

Políticas

Biblioteca de Políticas, Violaciones de Política

CULTURA GENERAL
- NO ES IMPORTANTE PARA VOSOTROS-

Enterprise Manager

Esquema

Objetos de Base de Datos

Tablas, Índices, Vistas, Sinónimos, Secuencias, Enlaces de Base de Datos, Objetos de Directorio, Reorganizar Objetos

Programas

Paquetes, Cuerpos de Paquetes, Procedimientos, Funciones, Disparadores, Clases, Java, Orígenes Java,

Base de Datos XML

Configuración, Recursos, Listas de Control de Acceso, Esquemas XML, Tablas de Tipo XML, Vistas de Tipo XML

Usuarios y Privilegios

Usuarios, Roles Perfiles, Valores de Auditoría

Vistas Materializadas

Vistas Materializadas, Logs de Vistas Materializadas, Grupos de Refrescamiento

BI & OLAP

Dimensiones, Cubos, Dimensiones de OLAP, Carpetas de Medidas

Tipos Definidos por el Usuario

Tipos de Matrices, Tipos de Objetos, Tipos de Tablas

Administración de Enterprise Manager

Administradores, Planificación de Notificación, Interrupciones

CULTURA GENERAL
- NO ES IMPORTANTE PARA VOSOTROS-

Herramientas ORACLE

- **Administración**
  - Oracle Universal Installer (OUI)
    - Instala el Software de Oracle y las opciones.
  - Database Configuration Assistant (DBCA)
    - Crea una base de datos usando plantillas proporcionadas por Oracle, Permite copiar una base de datos semilla, etc.
  - Database Upgrade Assistant
    - Asistente que guía en el paso de una base de datos existentes a una nueva versión.
  - Oracle Net Manager
    - Guía en la configuración de la red Oracle.

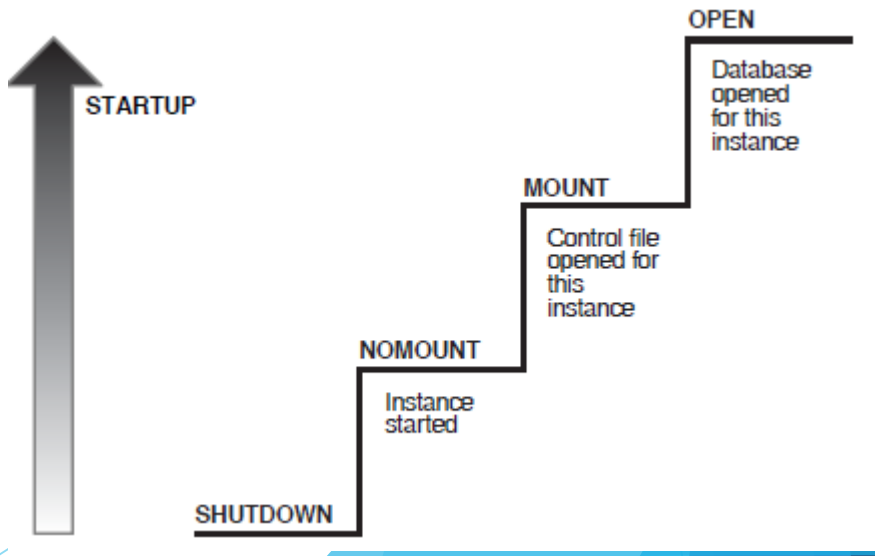
CULTURA GENERAL
- NO ES IMPORTANTE PARA VOSOTROS-

10. Iniciar/Finalizar

ORACLE

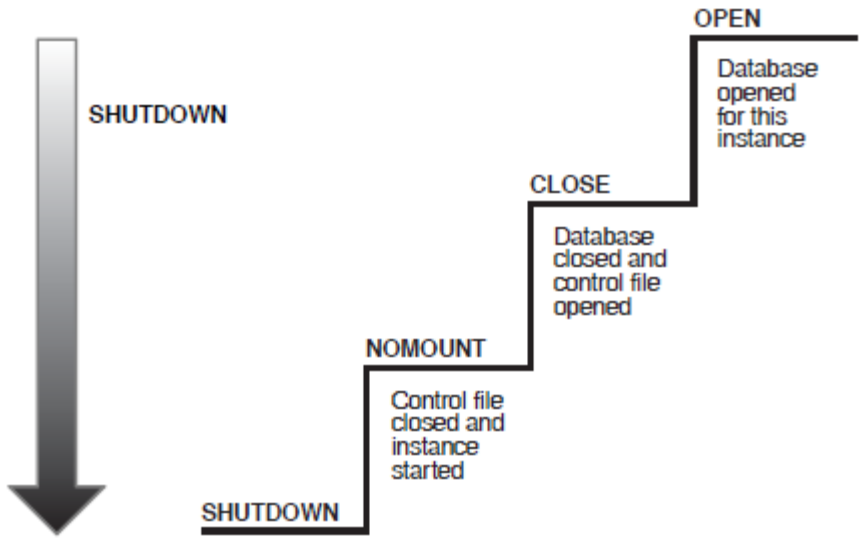
Iniciar/Finalizar ORACLE

- **Inicialización (*Startup*):** Es necesaria para que el SGBD pueda utilizarse.
  - **Crear una Instancia:** Crear el SGA y los procesos de *background (se lee el fichero de parámetros)*.
  - **Montar una BD:** Asocia la instancia ya creada a una BD concreta. Para montar la BD es necesario leer los *ficheros de control*.
  - **Abrir la BD:** Establece la BD como disponible para sus operaciones. Si la base de datos fue cerrada anormalmente, se realiza el *recovery*.



Iniciar/Finalizar ORACLE

- **Finalización (*Shutdown*):** Es el proceso inverso:
  - **Cerrar la BD.**
    - Se escriben los datos de la base de datos y los datos de recovery de la SGA a los datafiles y ficheros de redo, respectivamente.
    - Se cierran los datafiles y ficheros de redo.
    - Los ficheros de control permanecen abiertos
  - **Desmontar la BD.**
    - Se cierran los ficheros de control. La SGA se mantiene en memoria.
  - **Borrar la Instancia Oracle.**
    - La SGA se borra de la memoria



Iniciar/Finalizar ORACLE

- **Permisos necesarios:**
  - SYSDBA y SYSOPER son privilegios especiales del sistema que permiten el acceso a una instancia de base de datos, incluso cuando la base de datos no está abierta. El control de estos privilegios se encuentra fuera de la propia base de datos.
  - Cuando se conecta con el privilegio del sistema SYSDBA, se usa el esquema SYS.
  - Cuando se conecta como SYSOPER, se usa el esquema PUBLIC.
  - Los privilegios de SYSOPER son un subconjunto de los privilegios de SYSDBA.

Apertura y Cierre, con SQL*Plus

- **APERTURA de la BD:**
  - **Con la sentencia STARTUP** (que arranca la instancia).

    STARTUP [PFILE=filename] [EXCLUSIVE] [PARALLEL]
    [MOUNT [dbname] | OPEN [open_options] [dbname] | NOMOUNT]

    - **PFILE:** Especifica el fichero de parámetros.
    - **EXCLUSIVE:** La instancia se asociará a la BD en exclusiva y no permite otras instancias.
    - **PARALLEL:** Si se van a usar varias instancias para acceder a la BD.
    - **MOUNT:** Monta la BD con el nombre **dbname**, pero no la abre. Si no se especifica nombre lo toma del parámetro de inicialización **DB_NAME**.
    - **OPEN:** Monta y abre la BD especificada con las opciones **open_options**:

      READ {ONLY | WRITE [RECOVER]} | RECOVER
    - **NOMOUNT:** No monta (ni abre) la BD.

  - Si se arranca sin montar la BD: **ALTER DATABASE <nombre> MOUNT;**
  - Si montamos la BD sin abrirla : **ALTER DATABASE <nombre> OPEN <modo>;**
    - donde **<modo>** es opcional y puede ser:
      - **READ ONLY:** No puede modificarse. No puede ser **READ WRITE** en otra instancia.
      - **READ WRITE RESETLOG:** Borra toda la información del registro de rehacer.
      - **READ WRITE NORESETLOG:** No borra toda esa información.

Apertura y Cierre, con SQL*Plus

- **CIERRE de la BD:**
  - **SHUTDOWN [NORMAL]:** Por defecto. Espera a que terminen las conexiones (usuarios) y no admite nuevas.
  - **SHUTDOWN TRANSACTIONAL:** No se permiten transacciones nuevas pero se espera a que las que están en curso se cierren.
  - **SHUTDOWN IMMEDIATE:** Se terminan las instrucciones en curso, se desconecta a los usuarios y las transacciones activas hacen *rollback.* Se realiza un *checkpoint* y se cierran los ficheros.
  - **SHUTDOWN ABORT:** El más rápido. Borra la instancia sin cerrar o desmontar la BD. Requiere hacer un *recovery* al arrancar de nuevo (sólo usar en caso de emergencia).