

SESION 04

DICCIONARIO DE DATOS Y LENGUAJE SQL



Ing. Cesar Hajar
Instructor
chijar@Gmail.com

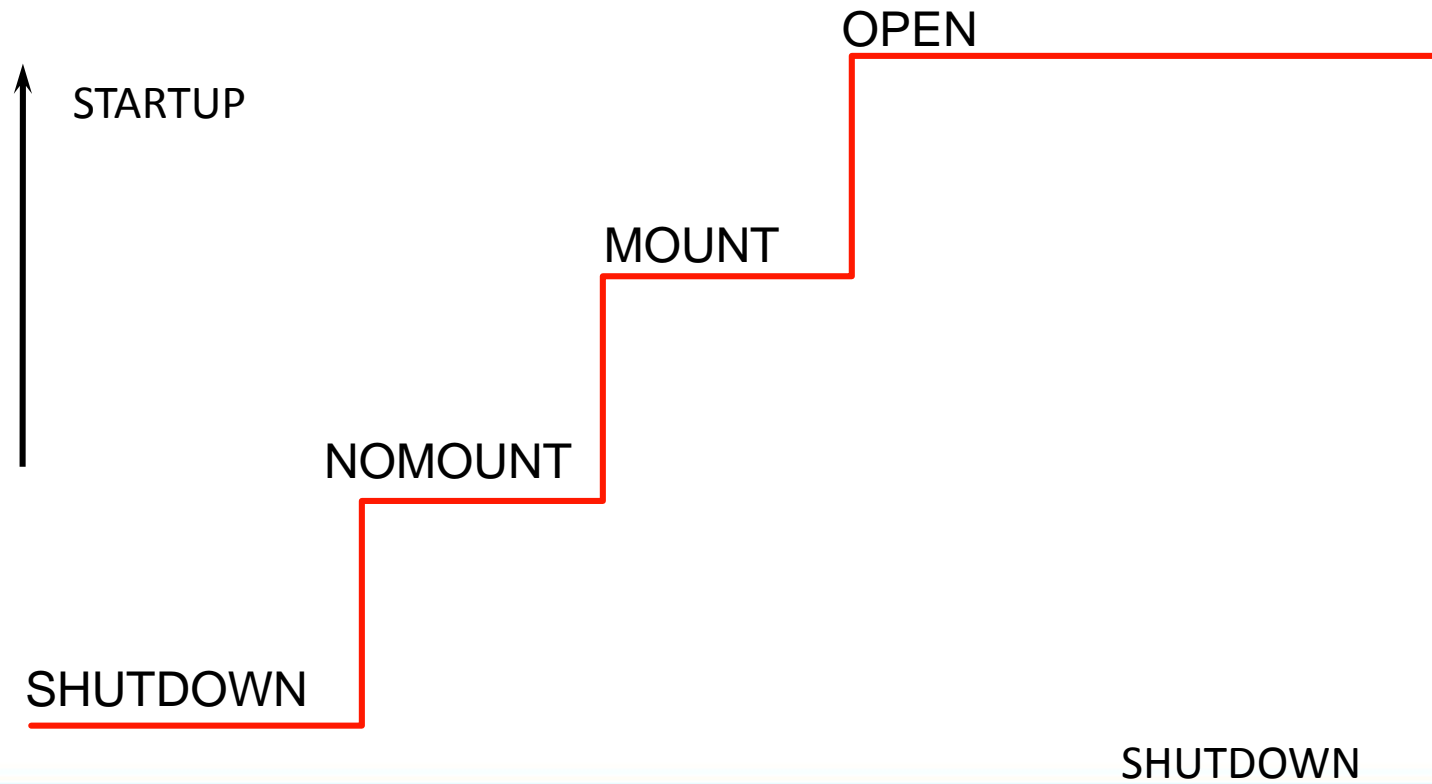


ESTADO DE LA INSTANCIA DE LA BASE DE DATOS (modos de inicio y apagado)

Modos inicio de una Base de Datos



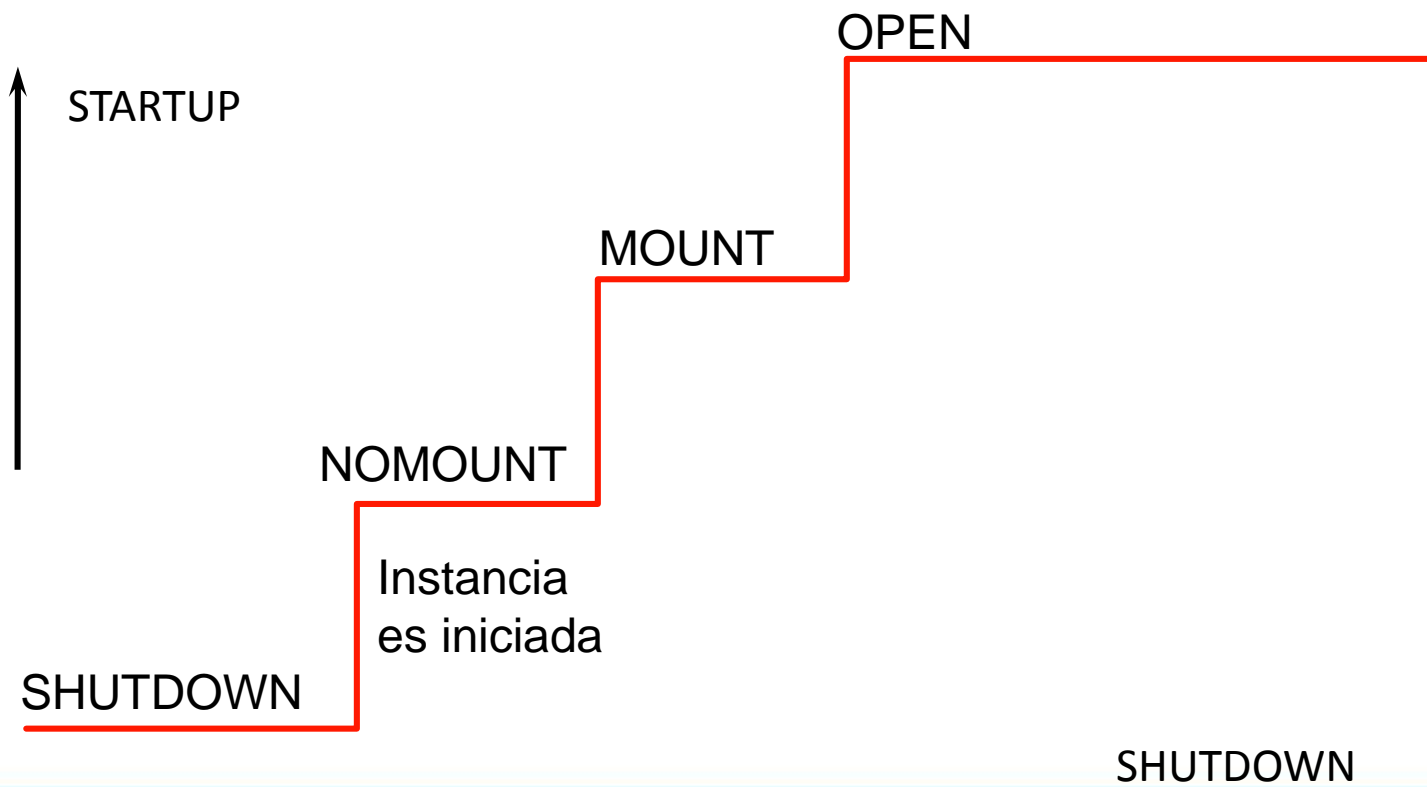
Starting Up a Database STARTUP



Modos inicio de una Base de Datos



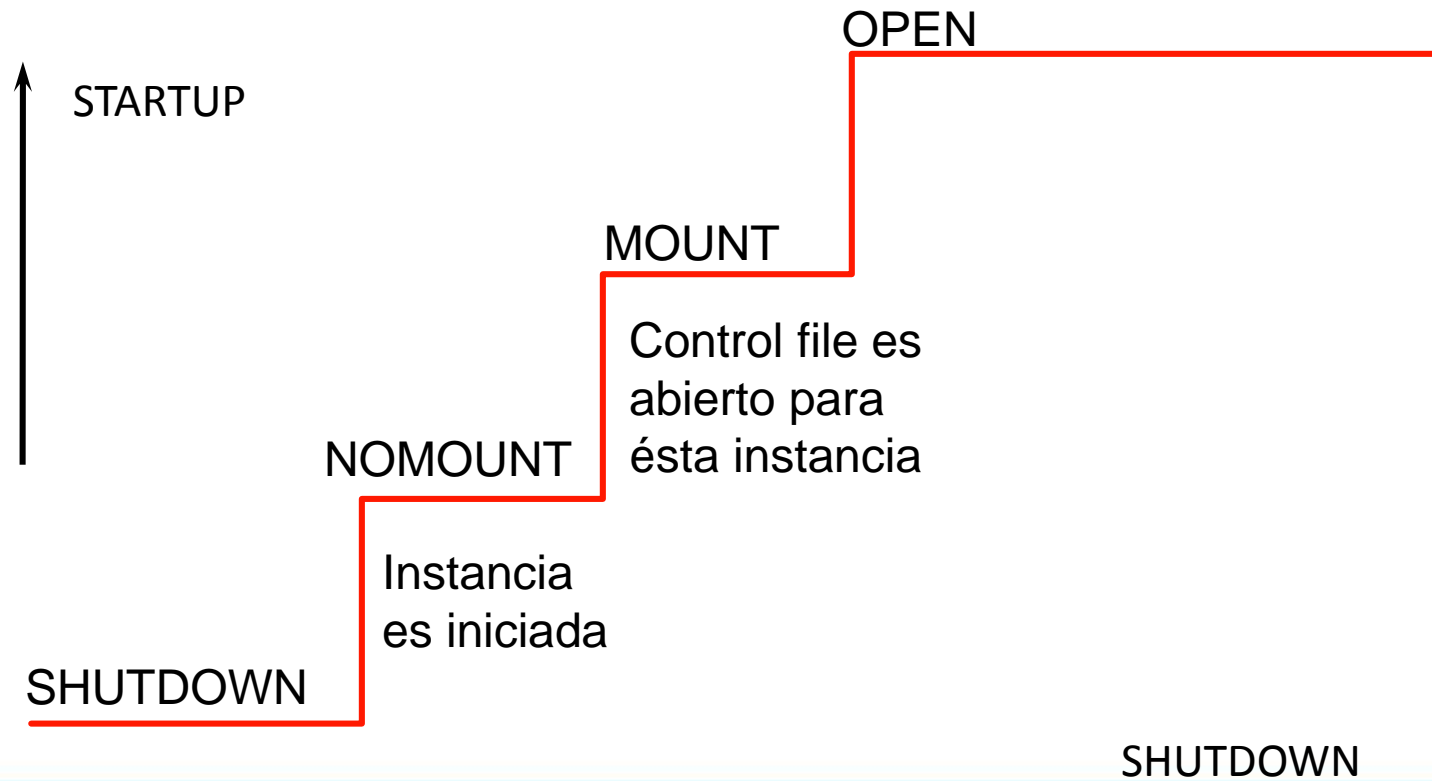
Starting Up a Database NOMOUNT



Modos inicio de una Base de Datos



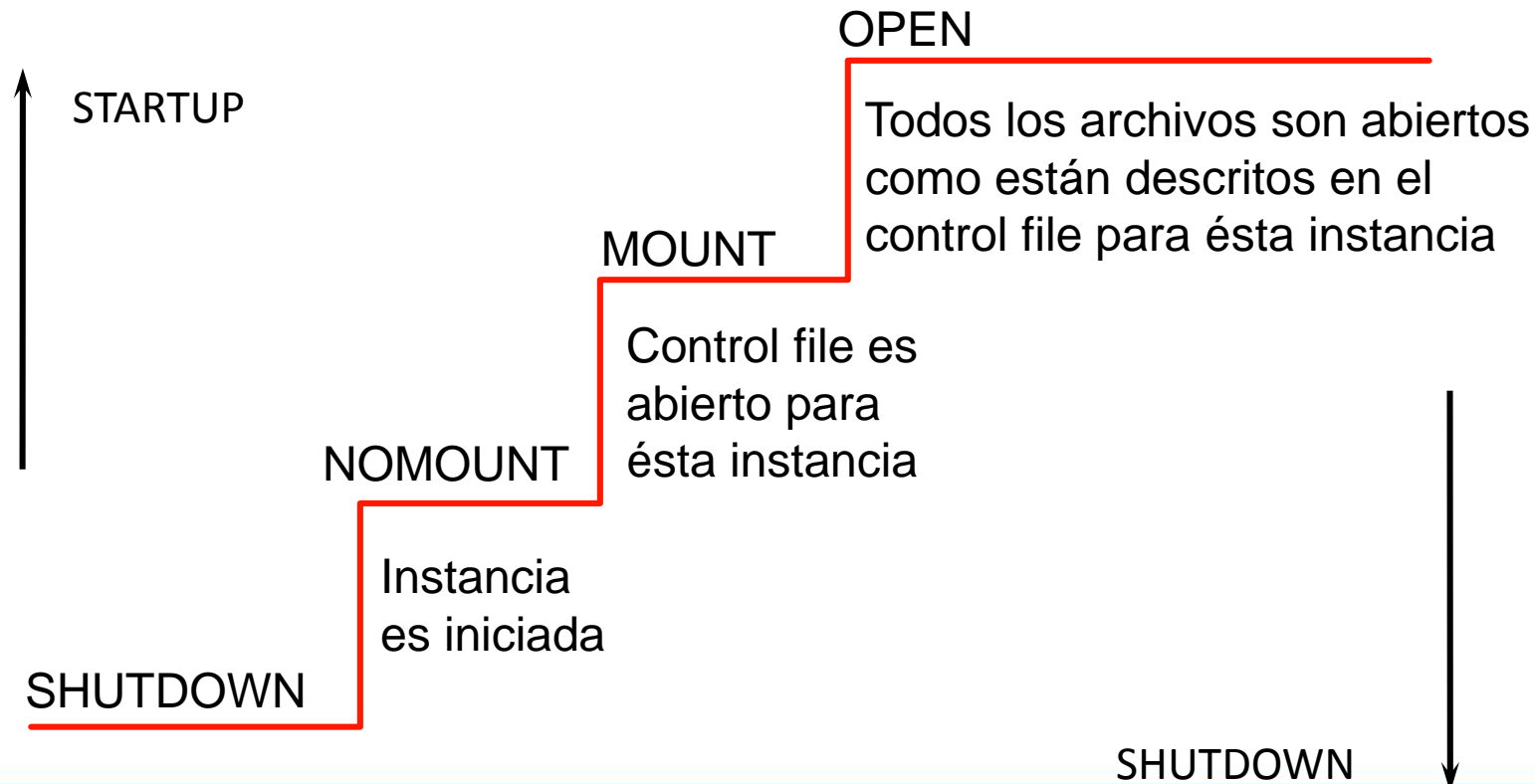
Starting Up a Database MOUNT



Modos inicio de una Base de Datos



Starting Up a Database OPEN





Comando **ALTER DATABASE**

- El comando ALTER DATABASE puede ser usado para cambiar el estado de una base de datos.
- Hay que considerar que solo es factible un avance hacia adelante en la secuencia de estados de inicio de una base de datos.
- SHUTDOWN → NOMOUNT → MOUNT → OPEN

Modos inicio de una Base de Datos



Comando **ALTER DATABASE**

- Opciones de uso del comando:
- NOMOUNT:
- ALTER DATABASE MOUNT
- ALTER DATABASE OPEN
- MOUNT:
- ALTER DATABASE OPEN



Modos de apagado de una Base de Datos

Modo de apagado	A	I	T	N
Permite nuevas conexiones	NO	NO	NO	NO
Espera a que las sesiones actuales terminen	NO	NO	NO	SI
Espera hasta que las transacciones actuales terminen	NO	NO	SI	SI
Fuerza un checkpoint y cierra los archivos	NO	SI	SI	SI

- Modos de apagado:

- A = ABORT
- I = IMMEDIATE
- T = TRANSACTIONAL
- N = NORMAL

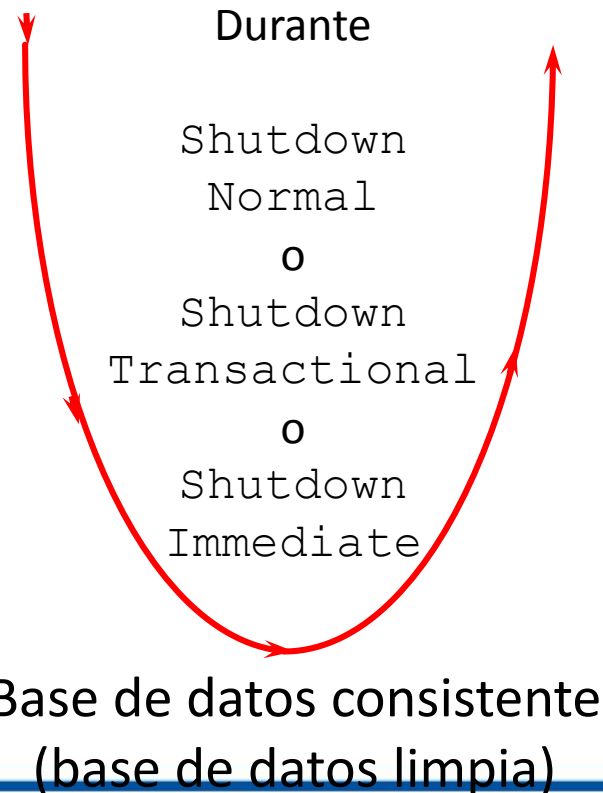
Modos de apagado de una Base de Datos



Opciones de SHUTDOWN

Hacia abajo:

- Los cambios NO CONFIRMADOS son deshechos (solo para IMMEDIATE y TRANSACTIONAL)
- Database buffer cache escritos a los data files
- Recursos liberados



Hacia arriba:

- La instancia no requiere recuperación

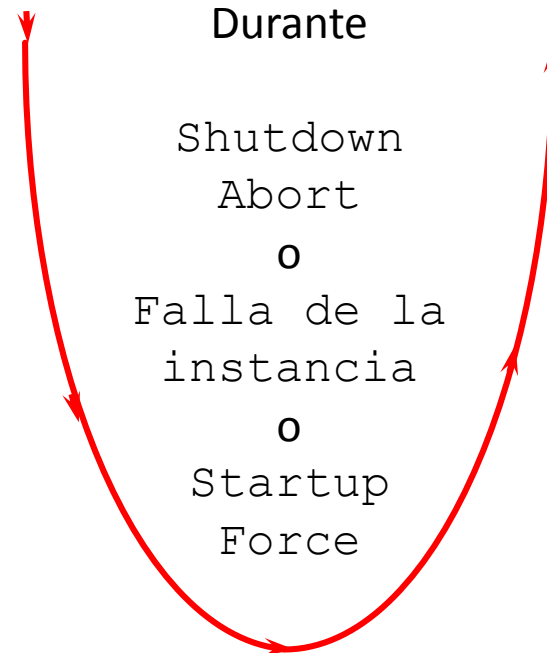
Modos de apagado de una Base de Datos



Opciones de SHUTDOWN

Hacia abajo:

- Los buffers modificados no son escritos a los data files.
- Los cambios no confirmados no son deshechos.(rolled back)



Hacia arriba:

- Archivos redo log en línea usados para re aplicar cambios.
- Segmentos UNDO usados para deshacer cambios no confirmados.
- Recursos liberados.

Base de datos inconsistente
(base de datos sucia)



- La administración de la instancia incluye el manejo del SGA y de sus principales procesos: SMON, PMON, DBWn, LgWr, ArcWn, CKPT
- Las Base de Datos deben estar en estado OPEN para poder trabajar.
- Se recomienda hacer un SHUTDOWN IMMEDIATE cuando se desea apagar una base de datos.
- Evite usar la sentencia SHUTDOWN ABORT.



ARCHIVOS Y PARAMETROS DE CONFIGURACIÓN

Archivos y parámetros de configuración

Listado

- ☐ Parameter file
- ☐ Alert log.
- ☐ Trace log



Archivos y parámetros de configuración

Listado



Pfile

El nombre viene de *Parameter File* y es un archivo de **texto**

Spfile

El nombre proviene de *Server Parameter File*, y es un archivo **binario** asociado a la base de datos. Se pueden modificar parámetros dinámicamente sin tener que poner stop.



Archivos y parámetros de configuración



Parámetros Estáticos: cuando se realiza el cambio debe de reiniciarse la instancia.

Dinámicos: no es necesario reiniciar la instancia.





Archivos y parámetros de configuración



opción	uso
SHOW PARAMETER nombre	En SQL*Plus, nos muestra los parámetros referentes a la sesión actual
SHOW SPPARAMETER nombre	En SQL*Plus nos enseña los valores de los parámetros almacenados en el SPFILE
SELECT * FROM V\$PARAMETER WHERE UPPER(name) LIKE '%nombre%'	Vista del diccionario de datos con los valores actuales de los parámetros
SELECT * FROM V\$SPPARAMETER WHERE UPPER(name) LIKE '%nombre%'	Vista del diccionario de datos con los valores de los parámetros en el SPFILE
SELECT * FROM V\$SYSTEM_PARAMETER WHERE UPPER(name) LIKE '%nombre%'	Valores de los parámetros que afectan a la instancia actual de la base de datos



Archivos y parámetros de configuración

parámetro	valor
DB_NAME	Nombre de la base de datos.
DB_DOMAIN	Dominio al que pertenece la base de datos.
DB_UNQ_NAME	Nombre único de base de datos. Se usa para diferenciar instancias con el mismo nombre de base de datos en sistemas distribuidos.
COMPATIBLE	Versión de Oracle Database compatible con la instancia de base de datos
CONTROL_FILES	Ruta a los archivos de control (pueden ser varios valores)
DIAGNOSTIC_DEST	Ruta, por defecto, inicial a los archivos log y de traza.
LOG_ARCHIVE_DEST_N	Ruta en la que se almacenará el archivo histórico redo log nº <i>n</i>
DB_RECOVERY_FILE_DEST	Ruta al directorio de recuperación de la base de datos
DB_RECOVERY_FILE_DEST_SIZE	Tamaño del área Flash de recuperación
DB_CREATE_FILE_DEST	Ruta para los archivos de datos gestionados por Oracle. Por defecto vale NULL y la ruta será la habitual (según el modelo OFA).
DB_FILES	Máximo número de archivos de base de datos que pueden ser abiertos por Oracle

Archivos y parámetros de configuración



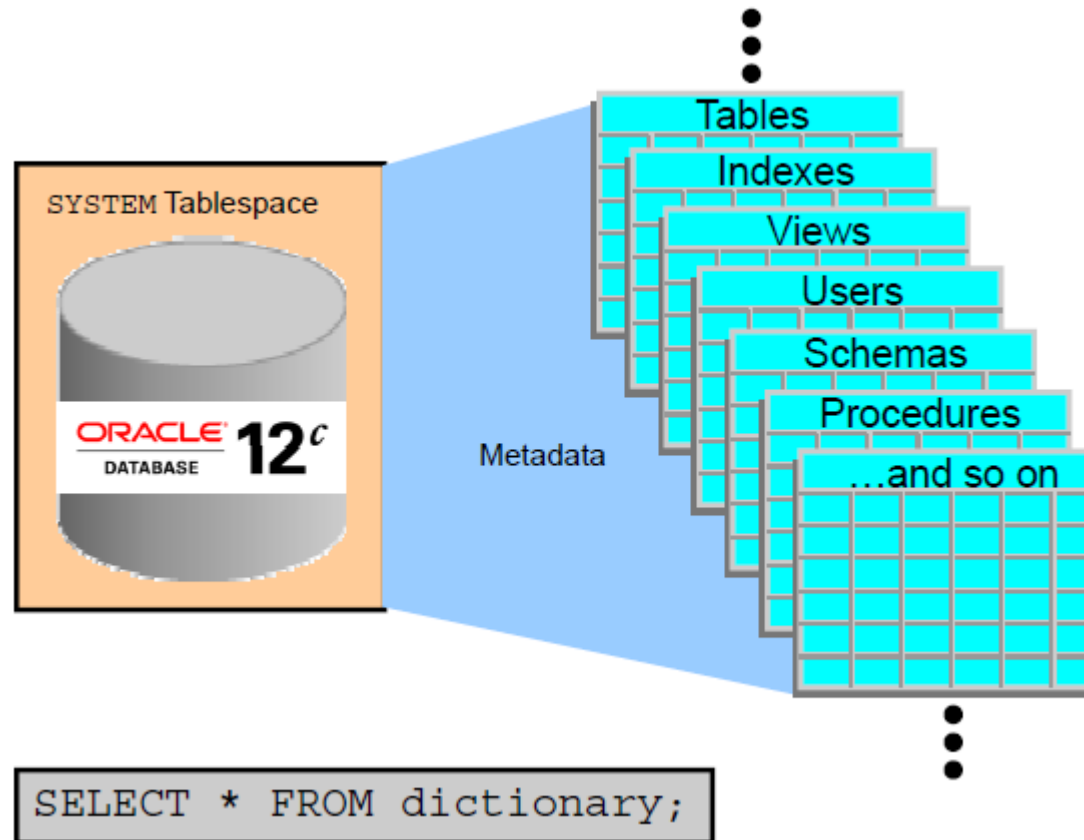
- **Trace files:** Each server and background process can write to an associated trace file. When an internal error is detected by a process, the process dumps information about the error to its trace file. Some of the information written to a trace file is intended for the database administrator, whereas other information is for Oracle Support Services.
- **Alert log files:** These are special trace entries. The alert log of a database is a chronological log of messages and errors. Each instance has one alert log file. Oracle recommends that you review this alert log periodically.



DICCIONARIO DE DATOS (VISTAS ESTATICAS Y DINAMICAS)



Diccionario de datos





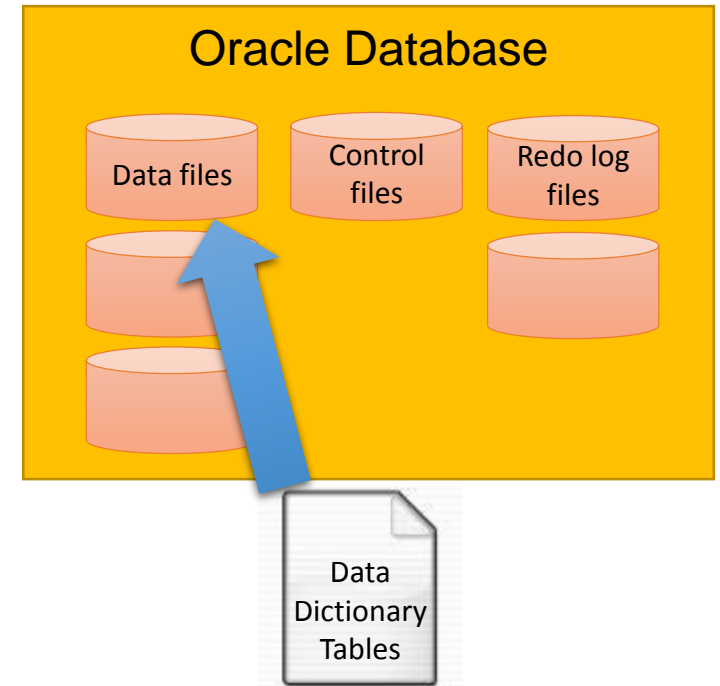
- Es la metadata de la base de datos
- Contiene los nombres y atributos de todos los objetos en la base de datos
- La creación o modificación de cualquier objeto provoca una actualización del diccionario
- Son tablas de solo lectura y vistas.
- Son mantenidas por el Oracle Server.
- Se acceden a través de vistas con la sentencia **SELECT**.
- Pertenecen al usuario **SYS**.
- Se almacenan en el TABLESPACE **SYSTEM**.



Contenido del diccionario de datos



- El diccionario de datos provee información acerca de:
 - Estructuras físicas y lógicas de la base de datos
 - Definiciones y asignaciones de espacio de objetos
 - Restricciones de integridad
 - Usuarios
 - Roles
 - Privilegios
 - Auditoria





Creación de las vistas del diccionario de datos



Script	Propósito	Script
catalog.sql	Comúnmente utilizado para crear las vistas del diccionario de datos y sinónimos.	catalog.sql
catproc.sql	Ejecuta scripts requeridos o usados con PL/SQL	catproc.sql

- Ubicación:
- UNIX: \$ORACLE_HOME/rdbms/admin
- NT: %ORACLE_HOME%\rdbms\admin



Cómo es usado el diccionario de datos



- Usos principales:
 - El servidor Oracle las utiliza para encontrar información sobre:
 - Usuarios
 - Objetos del esquema
 - Estructuras de almacenamiento
 - El servidor Oracle las modifica cuando una sentencia DDL es ejecutada.
 - Los usuarios y DBAs las utilizan como una referencia de solo lectura para obtener información sobre la base de datos.



Categoría de vistas del diccionario de datos



- Tres conjuntos de vistas estáticas
- Distinguidos por su alcance:
 - **DBA**: lo que hay en todos los esquemas
 - **ALL**: lo que el usuario puede acceder y lo que le pertenece
 - **USER**: lo que hay en el esquema del usuario (lo que le pertenece)

DBA_xxx All of the objects in the database

ALL_xxx Objects accessible by the current user

USER_xxx Objects owned by the current user



Ejemplos del diccionario de datos



- Visión general
 - `DICTIONARY`, `DICT_COLUMNS`
- Objetos del esquema
 - `DBA_TABLES`, `DBA_INDEXES`, `DBA_TAB_COLUMNS`, `DBA_CONSTRAINTS`
- Asignación de espacio
 - `DBA_SEGMENTS`, `DBA_EXTENTS`
- Estructuras de la base de datos
 - `DBA_TABLESPACES`, `DBA_DATA_FILES`

Información de tablas: USER_TABLES



```
DESCRIBE user_tables;
```

Name	Null?	Type
TABLE_NAME	NOT NULL	VARCHAR2 (30)
TABLESPACE_NAME		VARCHAR2 (30)
CLUSTER_NAME		VARCHAR2 (30)
IOT_NAME		VARCHAR2 (30)

```
SELECT table_name  
FROM user_tables;
```

TABLE_NAME
COUNTRIES
JOB_HISTORY
EMPLOYEES
JOBS

Información de columnas:

USER TAB COLUMNS



```
DESCRIBE user_tab_columns;
```

Name	Null?	Type
TABLE_NAME	NOT NULL	VARCHAR2 (30)
COLUMN_NAME	NOT NULL	VARCHAR2 (30)
DATA_TYPE		VARCHAR2 (106)
DATA_TYPE_MOD		VARCHAR2 (3)

```
SELECT column_name,data_type  
FROM user_tab_columns  
WHERE table_name='EMPLOYEES';
```

COLUMN_NAME	DATA_TYPE
EMPLOYEE_ID	NUMBER
FIRST_NAME	VARCHAR2
LAST_NAME	VARCHAR2
EMAIL	VARCHAR2



Información de restricciones



- **USER_CONSTRAINTS**: describe las restricciones definidas en las tablas del usuario.
- **USER_CONS_COLUMNS**: describe las columnas que eres propietario y que son especificadas en las restricciones.

```
DESCRIBE user_constraints;
```

Name	Null?	Type
OWNER	NOT NULL	VARCHAR2 (30)
CONSTRAINT_NAME	NOT NULL	VARCHAR2 (30)
CONSTRAINT_TYPE		VARCHAR2 (1)
TABLE_NAME		VARCHAR2 (30)



Información de restricciones



```
SELECT table_name,constraint_name,constraint_type
FROM user_constraints
WHERE table_name='EMPLOYEES';
```

TABLE_NAME	CONSTRAINT_NAME	CONSTRAINT_TYPE
EMPLOYEES	EMP_MANAGER_FK	R
EMPLOYEES	EMP_JOB_FK	R
EMPLOYEES	EMP_DEPT_FK	R
EMPLOYEES	EMP_LAST_NAME_NN	C
EMPLOYEES	EMP_EMAIL_NN	C
EMPLOYEES	EMP_HIRE_DATE_NN	C
EMPLOYEES	EMP_JOB_NN	C
EMPLOYEES	EMP_JOB_NN	C
EMPLOYEES	EMP_EMAIL_UK	U
EMPLOYEES	EMP_EMP_ID_PK	P



Ejemplos de tablas de rendimiento estático



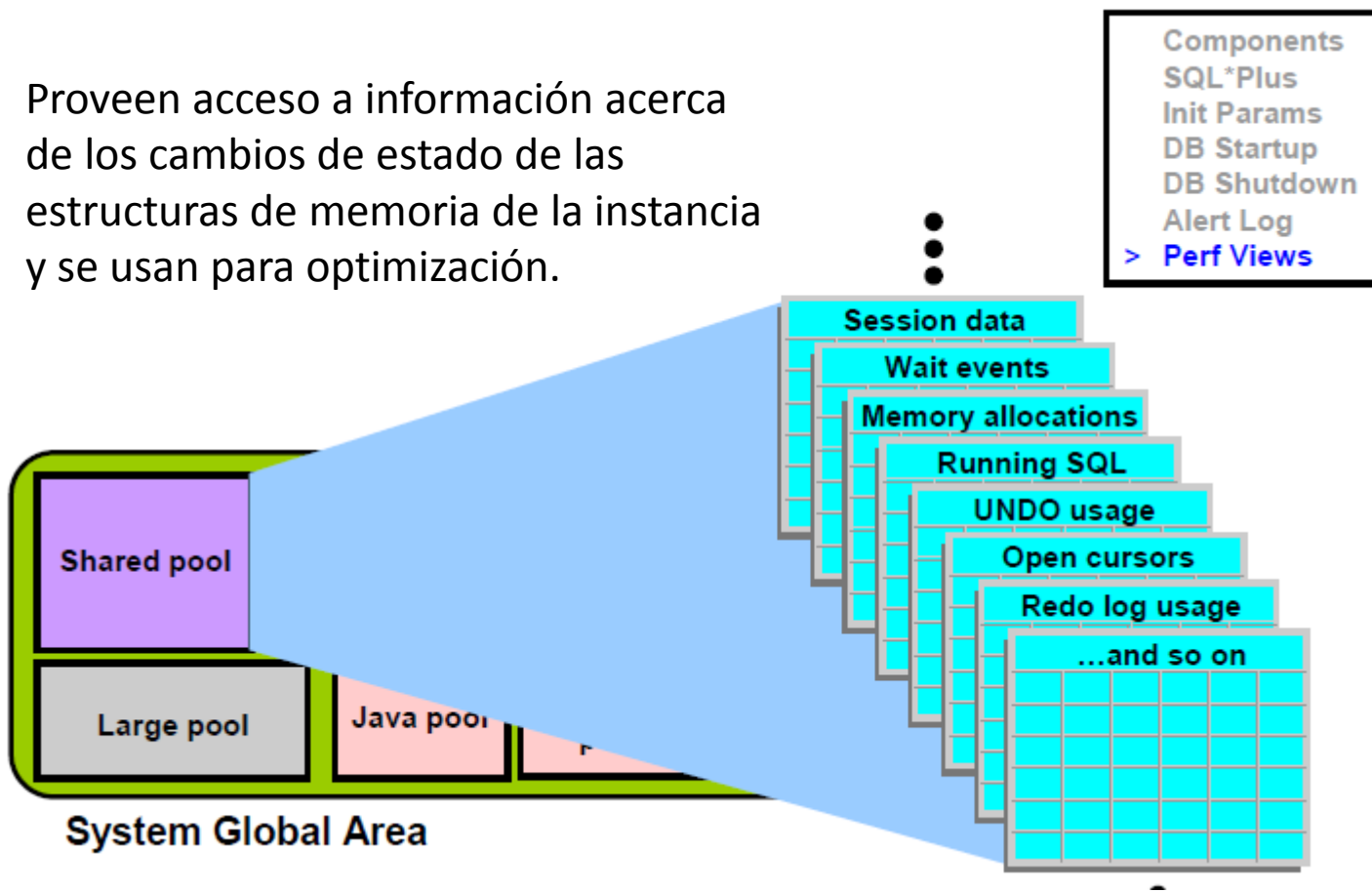
Vistas estáticas a usar con el prefijo USER_ALL_ o USER_	Uso
TABLES	Muestra todas las tablas accesibles desde nuestro usuario.
COLUMNS	Muestra las columnas de las tablas.
CONSTRAINTS	Restricciones
COLS_CONSTRAINTS	Restricciones indicando las columnas implicadas
VIEWS	Vistas
TRIGGERS	Lanzadores de código
SEQUENCES	Secuencias
SYNONYMS	Sinónimos
TAB_COLUMNS	Columnas de las tablas
TYPES	Tipos definidos en la base de datos
VARRAYS	Información sobre los datos de tipo VARRAY definidos en la base de datos
OBJECTS	Objetos de esquema
CATALOG	Lista tablas, vistas, sinónimos y secuencias



Vistas de rendimiento dinámico



Proveen acceso a información acerca de los cambios de estado de las estructuras de memoria de la instancia y se usan para optimización.





Vistas de rendimiento dinámico



- Tablas virtuales
- Registra la actividad actual de la base de datos
- Se actualiza continuamente mientras la base de datos está en funcionamiento
- La información se accede desde la memoria y archivo de control
- Se utiliza para monitorear y afinar la base de datos



Información almacenada



- Las vistas dinámicas de performance incluyen información acerca de:
 - Sesiones
 - Estado de los archivos
 - Progreso de las tareas y procesos
 - Bloqueos
 - Estado de backup
 - Uso de memoria
 - Parámetros de sistema y sesión
 - Ejecución de SQL
 - Estadísticas y métricas



Ejemplos de tablas de rendimiento dinámico

Vistas dinámicas	Uso
V\$INSTANCE	Muestra el estado de la instancia de la base de datos
V\$PARAMETER	Lista de los parámetros y sus valores actualmente en funcionamiento
V\$system_parameter	Parámetros del sistema
V\$SPPARAMETER	Muestra el valor de los parámetros en el archivo SPFILE de parámetros actual
V\$DATABASE	Información sobre la base de datos activa
V\$SESSION	Sesiones activas
V\$PROCESS	Procesos activos
V\$SESS_IO	Contiene estadísticas de entrada/salida por cada usuario
V\$SESSION_LONGOPS	Estado de las operaciones que tardan más de seis segundos en ejecutarse
V\$SYSSTAT	Estadísticas de la sesión
V\$RESOURCE_LIMIT	Información sobre el uso actual y máximo disponible de cada recurso
V\$SQLAREA	Estadísticas sobre el área compartida de SQL (en el Shared Pool de la SGA)



- Lista sentencias SQL cuyo tiempo de ejecución supera los 200000

```
SQL> SELECT sql_text, executions FROM v$sql  
WHERE cpu_time > 200000;
```

- Lista sesiones que tienen más de un día conectados

```
SQL> SELECT * FROM v$session WHERE machine =  
'EDRSR9P1' and logon_time > SYSDATE - 1;
```

- Lista las sesiones bloqueadas

```
SQL> SELECT sid, ctime FROM v$lock  
WHERE block > 0;
```



Consideraciones de las vistas de rendimiento



- El dueño de las vistas es el usuario SYS
- Vistas diferentes están disponibles en diferentes momentos:
 - Cuando la instancia ha levantado
 - La base de datos es montada
 - La base de datos es abierta
- Puede consultar la vista V\$FIXED_TABLE para consultar todas las vistas de rendimiento
- Estas vistas son siempre referidas como vistas “V-dólar” (v\$)
- No garantizan consistencia de lectura debido a que la data es dinámica



DQL – CONSULTAS BASICAS



DQL – CONSULTAS BASICAS



Data manipulation language (DML)

SELECT

INSERT

UPDATE

DELETE

MERGE

Data Control Language (DCL)

GRANT

REVOKE

Data definition language (DDL)

CREATE

ALTER

DROP

RENAME

TRUNCATE

COMMENT

Transaction control

COMMIT

ROLLBACK

SAVEPOINT



Sentencia SELECT



```
SELECT *|{[DISTINCT] column [alias],...}  
FROM table;
```

- SELECT identifica las columnas a mostrar
- FROM identifica las tablas que tiene esas columnas



Escribiendo sentencias SQL



- Las sentencias SQL no distingue mayúsculas y minúsculas.
- Las sentencias SQL pueden ser ingresadas en una o más líneas
- Las palabras claves no pueden abreviarse o separan a través de líneas.
- Las cláusulas se colocan generalmente en líneas separadas.
- Las sangrías se utilizan para mejorar la lectura.
- En SQL Developer, las sentencias SQL pueden ser opcionalmente terminado con un punto y coma (;). Punto y coma se requieren al ejecutar varias instrucciones SQL.
- En SQL * Plus, NO se requiere un punto y coma (;) para terminar cada sentencia SQL.



Expresiones aritméticas



- Se pueden crear expresiones con números y fechas utilizando operadores aritméticos.

Operador	Descripción
+	Suma
-	Resta
*	Multiplicación
/	División



Expresiones aritméticas



- Uso de operadores aritméticos

```
SELECT last_name, salary, salary + 300  
FROM employees;
```

- Precedencia de operadores

```
SELECT last_name, salary, 12*salary+100  
FROM employees;
```

```
SELECT last_name, salary, 12*(salary+100)  
FROM employees;
```



Definición de valor NULO



- NULL es un valor que no está disponible, sin asignar, es desconocido o inaplicable.
- Null no es el mismo que cero o un espacio en blanco.

```
SELECT last_name, salary, salary + 300  
FROM employees;
```



Definiendo alias de columna



- Un alias de columna:
 - Renombra un encabezado de columna
 - Es útil cuando se hacen cálculos
 - Sigue inmediatamente al nombre de la columna (también puede haber la palabra clave opcional AS entre el nombre de la columna y el alias)
 - Requiere comillas dobles si contiene espacios o caracteres especiales, o si se trata de mayúsculas y minúsculas





Usando alias de columnas



	 NAME	 COMM
1	King	(null)
2	Kochhar	(null)
3	De Haan	(null)
4	Hunold	(null)

```
SELECT last_name AS name,  
commission_pct comm  
FROM employees;
```

	 Name	 Annual Salary
1	King	288000
2	Kochhar	204000
3	De Haan	204000
4	Hunold	108000

```
SELECT last_name "Name" , salary*12  
"Annual Salary"  
FROM employees;
```



Operadores de concatenación



- Un operador de concatenación:
 - Enlaza columnas o cadenas de caracteres a otras columnas
 - Está representado por dos barras verticales (||)
 - Crea una columna resultante que es una expresión de caracteres

```
SELECT last_name||job_id AS "Employees"  
FROM   employees;
```

	Employees
1	AbelSA_REP
2	DaviesST_CLERK
3	De HaanAD_VP
4	FordIT_PROG

Filas duplicadas



- El resultado por defecto de las consultas son todas las filas, incluyendo filas duplicadas

```
SELECT department_id  
FROM employees;
```

	DEPARTMENT_ID
1	90
2	90
3	90
4	50

```
SELECT DISTINCT department_id  
FROM employees;
```

	DEPARTMENT_ID
1	(null)
2	90
3	20
4	110



Limitar filas seleccionadas



- Restringir las filas que son retornadas utilizando la cláusula WHERE

```
SELECT *|{ [DISTINCT] column [alias],...}  
FROM table  
[WHERE logical expression(s)];
```

- La cláusula WHERE sigue a la cláusula FROM



Operadores de comparación



Operador	Significado
=	Igual a
>	Mayor que
>=	Mayor o igual que
<	Menor
<=	Menor o igual que
<>	Diferente
BETWEEN ... AND ...	Entre dos valores
IN (set)	Valores en lista
Like	Parecido
Is Null	Es un valor nulo

Definir condiciones utilizando operadores lógicos



Operador	Significado
AND	Retorna VERDADERO si los dos componentes de la condición son verdaderos
OR	Retorna VERDADERO si uno de los componentes de la condición es verdadera
NOT	Retorna VERDADERO si la condición es falsa

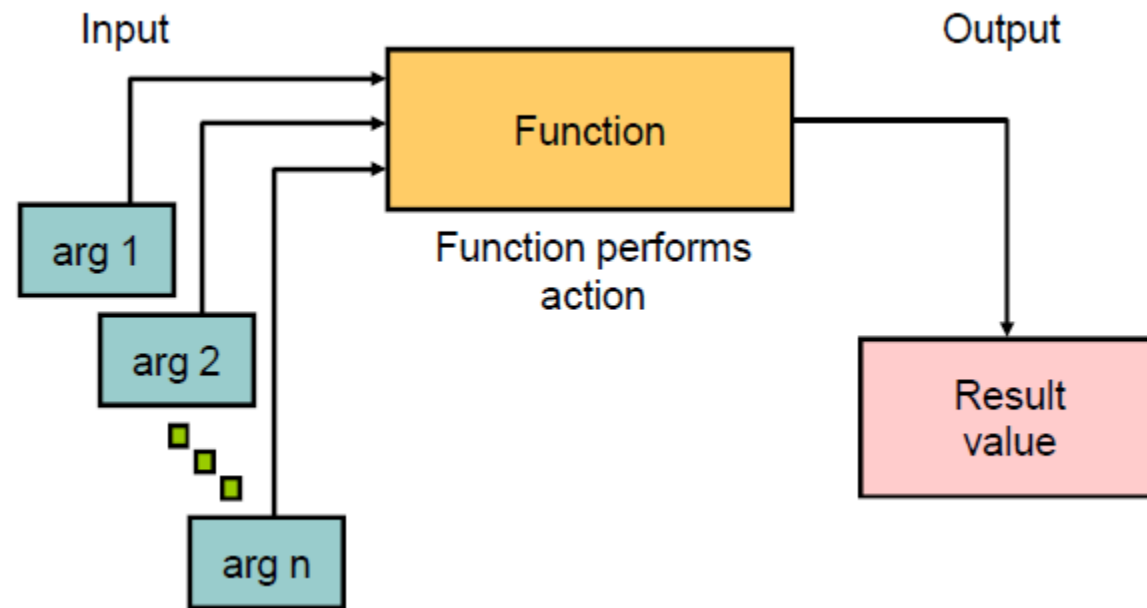


Uso de la cláusula ORDER BY

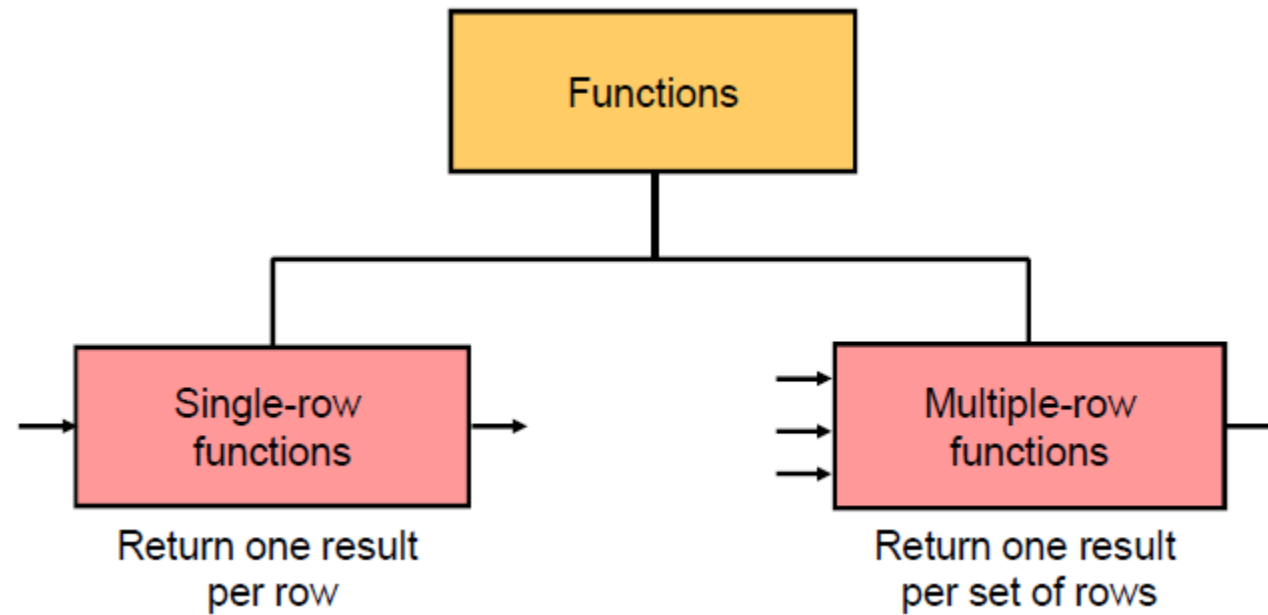


- Ordenar los datos obtenidos con la cláusula ORDER BY
 - ASC: ordena de forma ascendente (valor por defecto)
 - DESC: ordena de forma descendente
 - Se puede ordenar utilizando la posición numérica
 - Se puede ordenar por múltiples columnas

Funciones SQL



Tipos de funciones





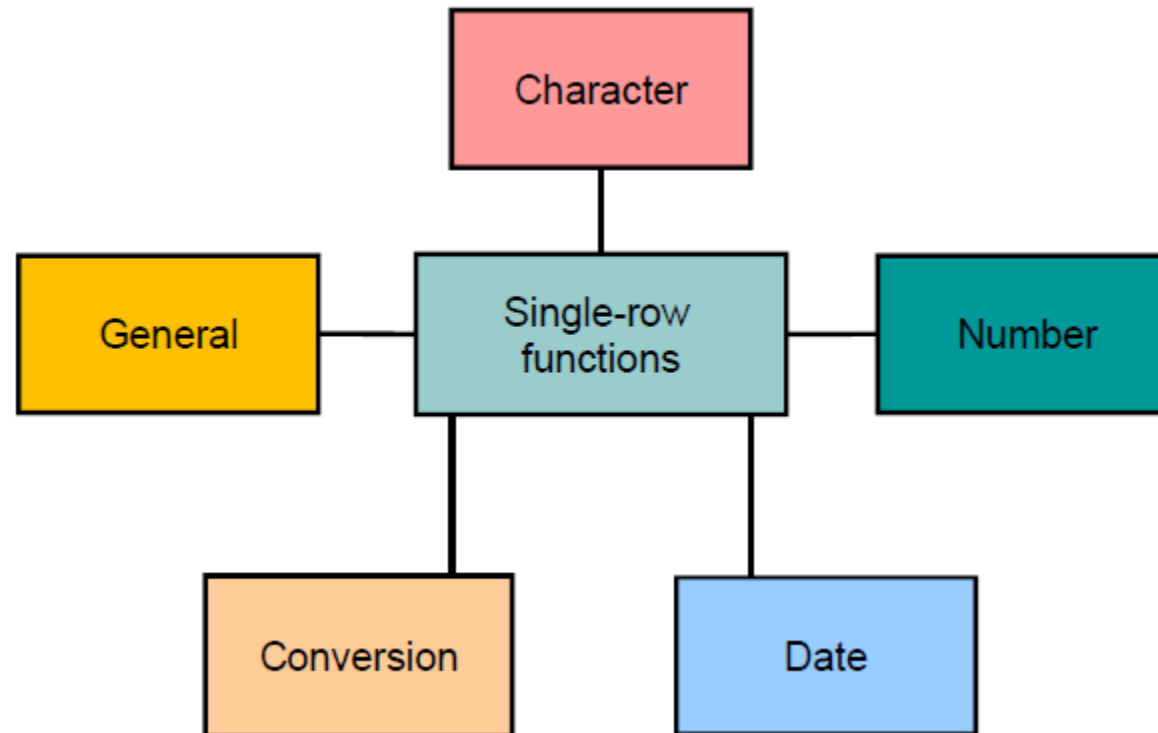
Funciones de fila simple



- Manipulan los elementos de datos
- Acepta argumentos y devuelve un valor
- Se aplica en cada fila que se devuelve
- Retorna un resultado por fila
- Puede modificar el tipo de datos
- Se pueden anidar
- Acepta argumentos que pueden ser una columna o una expresión

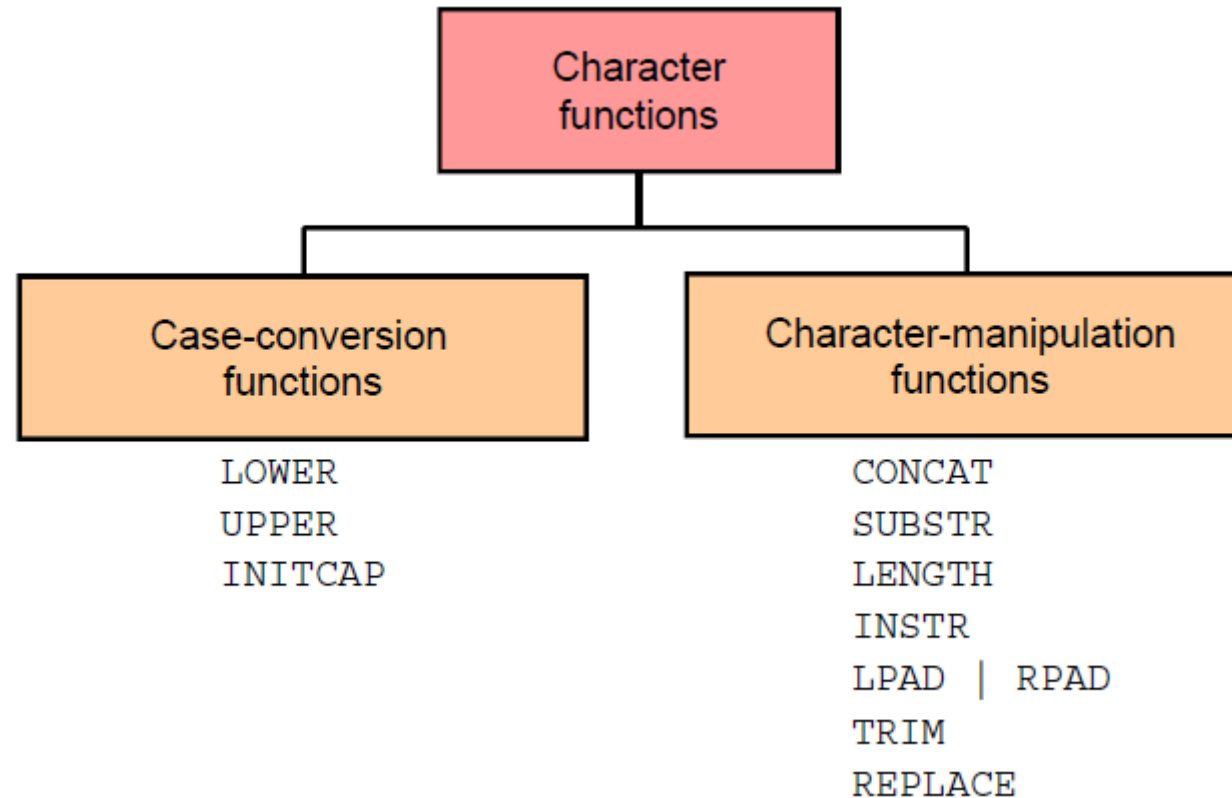


Funciones de fila simple





Funciones tipo caracter





Funciones tipo caracter



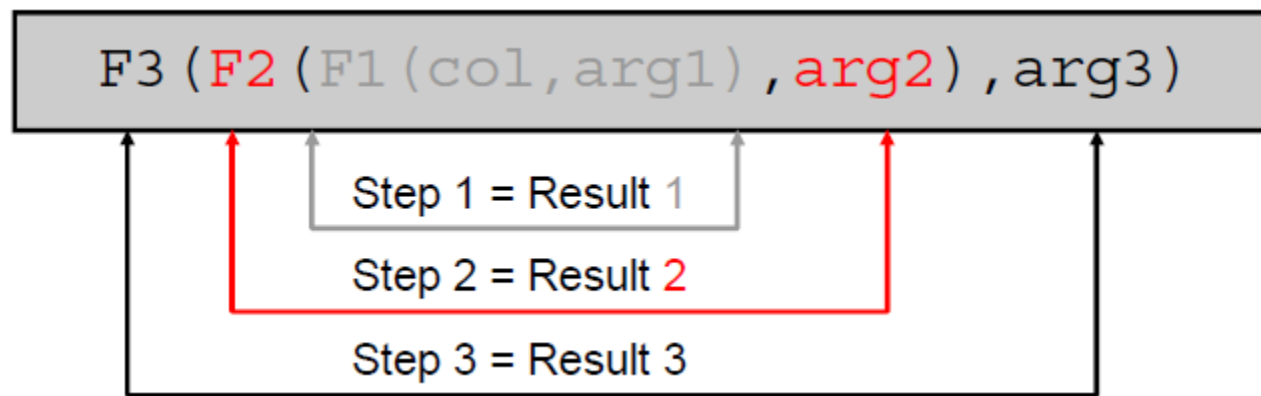
Function	Purpose
<code>LENGTH(column expression)</code>	Returns the number of characters in the expression
<code>INSTR(column expression, 'string', [,m], [n])</code>	Returns the numeric position of a named string. Optionally, you can provide a position <i>m</i> to start searching, and the occurrence <i>n</i> of the string. <i>m</i> and <i>n</i> default to 1, meaning start the search at the beginning of the string and report the first occurrence.
<code>LPAD(column expression, n, 'string')</code> <code>RPAD(column expression, n, 'string')</code>	Returns an expression left-padded to length of <i>n</i> characters with a character expression. Returns an expression right-padded to length of <i>n</i> characters with a character expression.
<code>TRIM(leading/trailing/both, trim_character FROM trim_source)</code>	Enables you to trim leading or trailing characters (or both) from a character string. If <i>trim_character</i> or <i>trim_source</i> is a character literal, you must enclose it in single quotation marks.
<code>REPLACE(text, search_string, replacement_string)</code>	Searches a text expression for a character string and, if found, replaces it with a specified replacement string



Anidando funciones



- Las funciones de fila simple se pueden anidar
- Las funciones anidadas se evalúan desde el nivel más profundo hasta el nivel menos profundo





Funciones numéricas



- ROUND: redondea valores a un decimal específico
- TRUNC: trunca un valor a un decimal específico
- MOD: retorna el residuo de una división

Funciones de fecha



Función	Resultado
MONTHS_BETWEEN	Número de meses entre dos fechas
ADD_MONTHS	Añade meses calendarios a una fecha
NEXT_DAY	Día de la semana de una fecha específica
LAST_DAY	Último día del mes
ROUND	Redondea una fecha
TRUNC	Trunca una fecha
EXTRACT (datetime)	Extrae una porción de la fecha



Funciones de agregación



- Las funciones de grupo operan sobre un grupo de filas y retorna un valor

EMPLOYEES

	DEPARTMENT_ID	SALARY
1	10	4400
2	20	13000
3	20	6000
4	110	12000
5	110	8300
6	90	24000
7	90	17000
8	90	17000
9	60	9000
10	60	6000
...		
18	80	11000
19	80	8600
20	(null)	7000

Maximum salary in
EMPLOYEES table

MAX(SALARY)
24000



Tipos de funciones de grupo



- AVG
- COUNT
- MAX
- MIN
- SUM
- LISTAGG
- STDDEV
- VARIANCE



Creando grupos de datos



EMPLOYEES

	DEPARTMENT_ID	SALARY	
1	10	4400	4400
2	20	13000	9500
3	20	6000	
4	50	2500	3500
5	50	2600	
6	50	3100	
7	50	3500	
8	50	5800	6400
9	60	9000	
10	60	6000	
11	60	4200	10033
12	80	11000	
13	80	8600	
...			
18	110	8300	
19	110	12000	
20	(null)	7000	

Average salary in the
EMPLOYEES table for
each department

	DEPARTMENT_ID	AVG(SALARY)
1	(null)	7000
2	20	9500
3	90	19333.333333333333...
4	110	10150
5	50	3500
6	80	10033.333333333333...
7	10	4400
8	60	6400



Creando grupos de datos: GROUP BY



- Se pueden dividir las filas de una tabla en grupos más pequeños utilizando la cláusula GROUP BY

```
SELECT column, group_function(column)
FROM table
[WHERE condition]
[GROUP BY group_by_expression]
[ORDER BY column];
```



Restringiendo grupos



EMPLOYEES

	DEPARTMENT_ID	SALARY
1	10	4400
2	20	13000
3	20	6000
4	50	2500
5	50	2600
6	50	3100
7	50	3500
8	50	5800
9	60	9000
10	60	6000
11	60	4200
12	80	11000
13	80	8600
...		
18	110	8300
19	110	12000
20	(null)	7000

The maximum salary per department when it is greater than \$10,000

	DEPARTMENT_ID	MAX(SALARY)
1	20	13000
2	90	24000
3	110	12000
4	80	11000



Restringiendo grupos con la cláusula HAVING



Cuando se usa la cláusula HAVING el servidor Oracle restringe grupos de la siguiente forma:

- Las filas son agrupadas
- La función de grupo es aplicada
- Los grupos que coinciden con la cláusula HAVING se muestran en el resultado.

```
SELECT    column, group_function
FROM      table
[WHERE    condition]
[GROUP BY group_by_expression]
[HAVING   group_condition]
[ORDER BY column];
```

Obteniendo datos de múltiples tablas



EMPLOYEES

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	JOB_ID
1	100	Steven	King	AD_PRES
2	101	Neena	Kochhar	AD_VP
3	102	Lex	De Haan	AD_VP
4	103	Alexander	Hunold	IT_PROG
5	104	Bruce	Ernst	IT_PROG
6	105	David	Austin	IT_PROG
7	106	Valli	Pataballa	IT_PROG
8	107	Uiana	Lorentz	IT_PROG
9	108	Nancy	Greenberg	FI_MGR
10	109	Daniel	Faviet	FI_ACCOUNT

JOBS

	JOB_ID	JOB_TITLE
1	AD_PRES	President
2	AD_VP	Administration Vice President
3	AD_ASST	Administration Assistant
4	FI_MGR	Finance Manager
5	FI_ACCOUNT	Accountant
6	AC_MGR	Accounting Manager
7	AC_ACCOUNT	Public Accountant
8	SA_MAN	Sales Manager
9	SA_REP	Sales Representative

↓ ↓

	EMPLOYEE_ID	JOB_ID	JOB_TITLE
1	206	AC_ACCOUNT	Public Accountant
2	205	AC_MGR	Accounting Manager
3	200	AD_ASST	Administration Assistant
4	100	AD_PRES	President
5	101	AD_VP	Administration Vice President
6	102	AD_VP	Administration Vice President
7	109	FI_ACCOUNT	Accountant