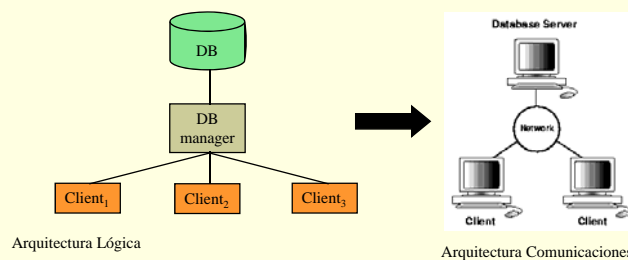


Bases de Datos

Introducción a ORACLE

- Sistema Gestor de Bases de Datos Relacionales (SGBDR), que maneja distintos tipos de datos:
 - Tradicionales como INTEGER, VARCHAR2, DATE, ..., etc.
 - Nuevos y complejos: imágenes, docs, mapas, video, audio, ..., etc.
- Entorno multiusuario (Cliente/Servidor).



Bases de Datos

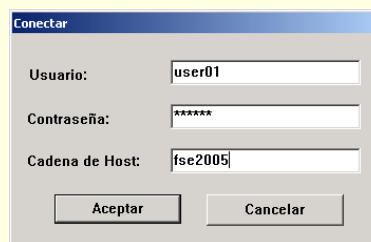
ORACLE y su SQL+

- **ORACLE:** Sistema Gestor de Base de Datos Relacional
 - Versátil
 - Probada eficiencia y escalabilidad
 - Amplia difusión
 - Basado en el lenguaje de datos PL/SQL (extensión de SQL)
- **SQL-PLUS:** herramienta interfaz de Oracle
 - permite conectarse al Servidor Oracle
 - envía sentencias SQL para su ejecución y muestra los resultados
 - contiene su propio lenguaje de comandos (locales)
 - permite la ejecución de *scripts*
 - *script:* fichero de texto que contiene un conjunto de sentencias SQL

Introducción al SQL+

Identificación de Usuario

- Una máquina con el Sw cliente de Oracle tendrá el SQL+ y los archivos de configuración que posibilitan la conexión
- Al arrancar el SQL+, la primera operación será la conexión
- Para ello, SQL+ solicitará al usuario:
 - nombre
 - contraseña
 - cadena de conexión (identificación servidor)



The screenshot shows a dialog box titled "Conectar" with three input fields: "Usuario:" containing "user01", "Contraseña:" containing "*****", and "Cadena de Host:" containing "fse2005". At the bottom, there are two buttons: "Aceptar" and "Cancelar".

Introducción al SQL+

Cambio de Contraseña

```
Oracle SQL*Plus
Fichero  Editar  Buscar  Opciones  Ayuda

SQL*Plus: Release 9.0.1.0.1 - Production on Mar Ene 27 17:32:31 2004
(c) Copyright 2001 Oracle Corporation. All rights reserved.

Conectado a:
Oracle9i Enterprise Edition Release 9.0.1.1.1 - Production
With the Partitioning option
JServer Release 9.0.1.1.1 - Production

SQL> password
Cambiando la contraseña para USER01
Contraseña antigua: *****
Contraseña nueva: *****
Confirmar contraseña nueva: *****
Contraseña cambiada
SQL> |
```

Introducción al SQL+

- Permite introducir una sentencia SQL.
- Tiene un buffer que almacena la última sentencia ejecutada
- Proporciona un editor de línea para modificar el buffer (para modificar y volver a ejecutar la última sentencia)
- También permite la ejecución de *scripts* (ficheros *.sql)
- Formatea los resultados de las consultas en informes básicos

Introducción al SQL+ Comandos SQL+ (I)

- Comandos de sesión
 - `connect [username[/password]][@cadena de conexión]`
conectar a un servidor
 - `disconnect` termina conexión actual
 - `quit / exit` salir (termina conexión y sesión)
 - `host` devuelve el control al S.O. (vuelve con `EXIT`)

Introducción al SQL+ Comandos SQL+ (II)

- Operaciones con el Buffer
 - **Ed** invoca al editor
 - **R | Run** ejecuta el contenido del buffer
 - **L | List** lista el contenido del buffer (completo)
 - **L n | L n m | ...**
lista el contenido de las líneas que se especifican.

Introducción al SQL+ Comandos SQL+ (III)

- Operaciones con el texto del Buffer
 - **[A | Append] texto**
concatena el texto a la última línea del buffer
 - **Input texto**
inserta nuevas líneas a continuación de la línea activa
 - **Del | Delete** borra la línea activa
 - **[C | Change] 'txt antiguo' 'txt nuevo'**
cambia el texto antiguo por el nuevo

Introducción al SQL+ Comandos SQL+ (IV)

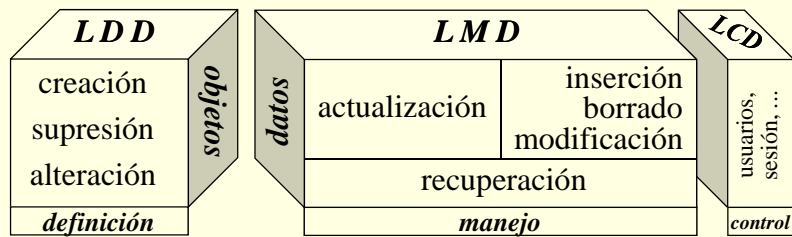
- Operaciones con ficheros
 - **Ed** <fichero> edita el fichero
 - **[Start | @]** <fichero> ejecuta un fichero *.sql
 - **Get** <fichero> introduce el fichero en el buffer sin ejecutarlo
 - **Save** <fichero> guarda el contenido del buffer en el fichero
 - **Save** <fichero> **append** guarda el buffer en el fichero concatenándolo con lo que ya tenía

Introducción al SQL+ Comandos SQL+ (V)

- Operaciones de *spooling*
 - **Spool** <fichero> arranca el spooling (todo lo que aparece por pantalla se guardará en el fichero)
 - **Spool** : muestra el nombre del fichero de salida.
 - **Spool off** : desactiva el spool.
 - **Spool out** : desactiva el spool y manda imprimir el fichero.

Lenguajes del SQL

- Instrucciones del Lenguaje SQL: divididas en dos tipos
 - para la definición de los objetos: **Lenguaje de Definición de Datos**
 - para el manejo de la información: **Lenguaje de Manipulación de Datos**
- Ambos conjuntos son completamente distintos: propósito, acciones, ...



Lenguaje de Definición de Datos

(LDD)

Tipos de Datos Nuevos

- **VARCHAR2 (n)** : alfanumérico (hasta 240)
- **NUMBER (n,p)** : números con signo y punto decimal
 - n = número total de dígitos a almacenar
 - p = dígitos a la derecha del punto decimal (Si se omite solo se podrá almacenar hasta 40 dígitos)
- **DATE** : se almacenan como una cadena de caracteres de 7 bytes (por defecto con formato dd/mm/aa).

Tipos de Datos Nuevos

- **LONG** : cadenas de caracteres de hasta 64 K (solo una por tabla). Normalmente para campos resumen.

Observaciones:

- * No se pueden indexar
- * No se pueden utilizar en cláusulas WHERE
- * No pueden ser seleccionadas en consultas anidadas
- * No pueden aparecer en ningún tipo de expresiones
- * No pueden aparecer en consultas que estén unidas a otras mediante UNION | MINUS | INTERSECT

Tratamiento de Fechas

- **SYSDATE** : fecha actual (SELECT SYSDATE FROM DUAL;)
- Las fechas en ORACLE se almacenan como números (facilidad de conversión a otros formatos).
- Aritmética de fechas:
 - Fecha + X** Añade X días a la fecha seleccionada
 - Fecha - X** Resta X días a la fecha seleccionada
 - Fecha - X/24** Resta X horas a la fecha seleccionada (resultados de tipo fecha)
 - Fecha - Fecha** Devuelve el número de días transcurridos entre dos fechas

Tratamiento de Fechas

- *Conversión de fechas :*
 - TO CHAR (fecha, 'formato')**: Convierte datos de tipo fecha en datos de tipo carácter con el formato indicado.
 - TO DATE (fecha, 'formato')**: Convierte datos de tipo carácter con el formato indicado, en datos de tipo fecha.

Creación de Tablas (I)

```
CREATE TABLE <nombre_de_la_tabla>
(
  <nombre_de_la_columna1> <tipo_de_datos> [(tamaño)] [NOT NULL] [DEFAULT],
  .....
  <nombre_de_la_columnaN> <tipo_de_datos> [(tamaño)] [NOT NULL] [DEFAULT],

  [UNIQUE [nombre]({<lista_de_columnas>})],
  [PRIMARY KEY [nombre]({<lista_de_columnas>})],
  [CHECK [nombre] <condición>],
  [FOREIGN KEY [nombre] ({<lista_de_columnas>})
  REFERENCES <nombre_de_tabla_referenciada> ({<lista_de_columnas>})
  [ON DELETE CASCADE] ] );
```

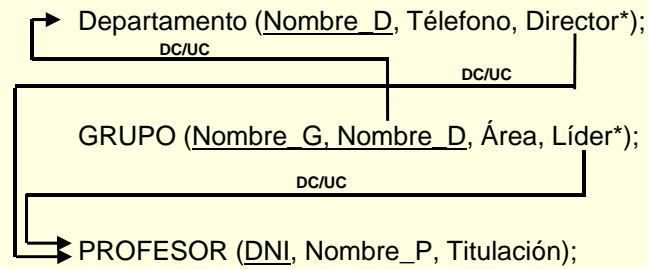
Creación de Tablas (II)

OBSERVACIONES:

- *Los nombres de las tablas de un usuario no se pueden repetir*
- *Los nombres de las columnas deben ser únicos dentro de cada tabla*
- *Los nombres pueden tener como máximo 30 caracteres*
- *Cada tabla puede contener hasta 254 columnas*

Creación de Tablas (III)

■ Ejemplo:



Creación de Tablas (IV)

```
CREATE TABLE PROFESOR
  (DNI          CHAR(9) PRIMARY KEY,
   Nombre_P    VARCHAR2(50),
   Titulación   VARCHAR2(30));

CREATE TABLE DEPARTAMENTO
  (Nombre_D    VARCHAR2(50) PRIMARY KEY,
   Teléfono    NUMBER(9),
   Director    CHAR(9),
   CONSTRAINT C1 FOREIGN KEY (Director)
   REFERENCES PROFESOR (DNI) ON DELETE CASCADE);

CREATE TABLE GRUPO
  (Nombre_G    VARCHAR2(30),
   Nombre_D    VARCHAR2(50),
   Área        VARCHAR2(30),
   Líder       CHAR(9),
   PRIMARY KEY (Nombre_G, Nombre_D),
   CONSTRAINT C2 FOREIGN KEY (Nombre_D)
   REFERENCES DEPARTAMENTO (Nombre_D) ON DELETE CASCADE,
   CONSTRAINT C3 FOREIGN KEY (Líder)
   REFERENCES PROFESOR (DNI) ON DELETE CASCADE);
```

Modificación de Tablas

■ ALTER TABLE

- *Cambiar la definición de una columna para permitir valores nulos o no, así como el tipo de datos*
- *Aumentar o disminuir el tamaño de una columna*
- *Añadir columnas*

Modificación de Tablas

■ ALTER TABLE {ADD|MODIFY|DROP|DISABLE|ENABLE}

```
ALTER TABLE <nombre_de_la_tabla> MODIFY  
  (<nombre_de_la_columna1> <tipo_de_datos> [NULL|NOT NULL] [constraint],  
  .....  
  <nombre_de_la_columnaN> <tipo_de_datos> [NULL|NOT NULL] [constraint]);
```

```
ALTER TABLE <nombre_de_la_tabla> DROP  
  (<constraint> <nombre_de_la_columna|s> CASCADE;
```

```
ALTER TABLE <nombre_de_la_tabla> {DISABLE|ENABLE}  
  (<constraint> <nombre_de_la_columna|s> CASCADE;
```

```
ALTER TABLE Profesor ADD (Sueldo NUMBER(10,3));  
ALTER TABLE Profesor ADD (Dept VARCHAR2(50));
```

Modificación de Tablas

■ **Observaciones:**

- **MODIFY:** para modificar el tipo o el tamaño de una columna ésta no debe contener valores

modificar una columna NULL a NOT NULL implica que no haya ninguna fila con valor NULL para esa columna, o bien que la tabla esté vacía

Borrado de Tablas

DROP TABLE <nombre_de_tabla>

Cuando se ejecuta el comando DROP:

- *La definición de la tabla se elimina del catálogo.*
- *El espacio que ocupaba la tabla se recupera para otros elementos de la base de datos*
- *Se borran todas las filas de la tabla sin ninguna alerta*
- *La ejecución de este comando elimina la tabla definitivamente*

Renombrado de Tablas

- **RENAME** <antiguo_nombre> TO <nuevo_nombre>

Cuando se ejecuta el comando RENAME:

- *El nuevo nombre se almacena en el catálogo y la tabla no podrá volver a ser referenciada con el nombre antiguo*
- *No se produce ninguna repercusión sobre los datos*
- *El propietario de la tabla es el único que puede cambiar el nombre*

Secuencias

- **Secuencias.** Generación de números secuenciales (autonuméricos) para las columnas

```
CREATE SEQUENCE <nombre_de_secuencia>  
[INCREMENT BY entero]  
[START WITH entero]  
[MAXVALUE entero|NOMAXVALUE]  
[MINVALUE entero|NOMINVALUE]  
[CYCLE|NOCYCLE]  
[CACHE entero|NOCACHE]  
[ORDER|NORDER]
```

Lenguaje de Manipulación de Datos

(LMD)

Introducción

- Inserción de tuplas
 - Borrado de tuplas
 - Modificación de tuplas
 - Consultas
- } Actualizaciones

Actualización: Inserción

■ Inserción de valores :

```
INSERT INTO <nombre_de_la_tabla>  
[<nombre_de_la_columna1>, ..., <nombre_de_la_columnaN>]  
VALUES (<valor1>, ..., <valorN>);
```

Tipos:

- * Inserción total de columnas : - el orden de los valores debe coincidir con el de la definición de las columnas.
- si algún valor es nulo se debe especificar con la constante NULL.
- * Inserción parcial de columnas: - se deben especificar las columnas (orden).

Actualización: Inserción

■ Inserción de tuplas mediante una subconsulta

```
INSERT INTO <nombre_de_tabla>  
[(<nombre_de_columna1>, ..., <nombre_de_columnaN>)]  
SELECT <lista_de_columnas_y_expresiones>  
FROM <lista_de_tablas>  
WHERE <criterios_de_selección>;
```

- * Se insertan los datos que figuran en la lista de tablas y que cumplan los criterios de selección.
- * El orden de las columnas en la cláusula SELECT debe coincidir con el orden de las columnas en la cláusula INSERT.

Actualización: Inserción

- Inserción de tuplas sin especificar el valor de las columnas

```
INSERT INTO <nombre_de_la_tabla>  
VALUES ('&columna1', '&columna2'....., '&columnaN');
```

- Consideraciones generales

* Cuando se utilice el comando INSERT parcial, todas las columnas que no estén especificadas tomarán valores nulos.

* Los valores de tipo CHAR y DATE deben especificarse con comillas simples.

* La inserción de datos de tipo DATE cuyo formato no coincida con el que tiene por defecto el SGBD se insertarán utilizando la función
TO_DATE ('fecha_a_insertar', 'formato')

Actualización: Borrado

- Borrado de todas las tuplas

```
DELETE [FROM] <nombre_de_tabla>
```

- Borrado de tuplas con un criterio

```
DELETE [FROM] <nombre_de_tabla> [WHERE <criterio_de_selección>]
```


Actualización: Modificación

■ Modificación de una tupla

```
UPDATE <nombre_de_tabla>  
SET columna =<nuevo_valor> [, columna = <nuevo_valor>]  
[WHERE <critero_de_selección_del_registro>];
```

■ Modificación de varias tuplas

```
UPDATE <nombre_de_tabla>  
SET columna = (SELECT columna y/o expresión FROM  
[WHERE <critero_de_selección_del_registro>]);
```

Actualización: Modificación

■ Consideraciones generales

- * Si se omite la cláusula WHERE se modifican todas las filas de la tabla.
- * El nuevo valor de la columna se puede obtener mediante expresiones, constantes o subconsultas.
- * Cuando se utilizan subconsultas, estas deben recuperar tantas columnas como se estén modificando.

Consultas

■ Consulta

```
SELECT { * | [DISTINCT] <columna> [, <columna> ] | <expresión> }  
FROM <tabla> [, <tabla> ]  
    [WHERE <condición> [<condición> ]  
    [GROUP BY <columna> [, <columna> ]  
    [HAVING <condición> [<condición> ] ]  
    [ORDER BY <columna> [, <columna> ] [ASC|DESC] ] ] ;
```

Ejemplos:

```
SELECT * FROM PROFESOR;  
SELECT SYSDATE FROM DUAL;  
SELECT Rownum, Nombre_P, Titulación  
FROM PROFESOR;
```

Consultas

■ Consideraciones

* Entre las condiciones del WHERE deberán incluirse las condiciones de combinación de las tablas :

- Combinación común : operador de igualdad.
- Combinación no común: operadores <, >, <>, BETWEEN,...
- Autocombinación : combinación de una tabla consigo misma.
- Combinación exterior : selecciona, además, las filas de una tabla que no tienen correspondencia con alguna de la otra. (columnaI = columnaJ(+)).

Consultas

■ Consideraciones

<condición> :

- condiciones booleanas (<, >, =, ...IS NULL|NOT NULL, LIKE).
- condiciones booleanas encadenadas por OR o AND.
- <columna> IN | NOT IN <rango_de_valores>.
- <columna> BETWEEN min_val AND max_val.

Ejemplos:

```
SELECT * FROM Profesor WHERE DNI=XXXXXXXX;
```

```
SELECT Nombre_P FROM Profesor  
WHERE Titulación <> 'XXXXXXXX';
```

```
SELECT * FROM Grupo WHERE Área IS NULL;
```

```
SELECT Nombre_P FROM Profesor  
WHERE DNI IN (SELECT Líder FROM Grupo);
```

```
SELECT Nombre_P P, Nombre_D D FROM Profesor P, Departamento D  
WHERE P.DNI IN (SELECT Líder FROM Grupo);
```

Consultas

■ Consideraciones

- GROUP BY <columna> [, <columna>] : agrupa el resultado de la consulta por las columnas especificadas.
- HAVING <condición> [<condición>] : restricciones de los grupos de salida (análoga a WHERE).
- ORDER BY <columna> [, <columna>] [ASC|DESC] : variación del orden por defecto.

Consultas

■ Consideraciones

* El orden de ejecución es:

- Se eligen las tuplas que cumplen las condiciones de la cláusula WHERE
- Se realizan los grupos basados en la cláusula GROUP BY
- Se calculan los resultados de las funciones de grupo para cada grupo
- Se eliminan aquellos grupos que no cumplen la condición del HAVING
- Y luego se ordenan por las columnas especificadas en el ORDER BY

Consultas

Ejemplos:

```
SELECT COUNT(*) FROM Profesor;  
SELECT COUNT(*) FROM Profesor GROUP BY Titulación;  
SELECT SUM(Sueldo) FROM Profesor GROUP BY Titulación  
HAVING COUNT(*)>10;  
SELECT MAX(Sueldo), MIN(Sueldo), AVG(Sueldo)  
FROM Profesor WHERE Titulación = 'Doctorado'  
GROUP BY Dept;
```