Contenido

[UD01 2](#_Toc58089688)

[Tipos de ficheros, tipos de soporte, métodos de acceso y parámetros 2](#_Toc58089689)

[Qué es una base de datos y sus ventajas 4](#_Toc58089690)

[Distintos modelos de BBDD 6](#_Toc58089691)

[Que es SGBD. Funciones, componentes y tipos 8](#_Toc58089692)

[Bases de datos centralizadas y distribuías 10](#_Toc58089693)

[Sistemas gestores de bases de datos comerciales y libres 11](#_Toc58089694)

[UD02 11](#_Toc58089695)

[En run SQL command lite 11](#_Toc58089696)

[Creando y modificando tablas 13](#_Toc58089697)

[Ejemplo Clínica veterinaria 13](#_Toc58089698)

[Ejemplo coches 15](#_Toc58089699)

# UD01

## Tipos de ficheros, tipos de soporte, métodos de acceso y parámetros

* FICHEROS: conjunto de información relacionada, tratada como un todo y organizada de forma estructurada

## Qué es una base de datos y sus ventajas

+

**Base de datos:** Es una colección de datos relacionados lógicamente entre sí, con una definición y descripción comunes y que están estructurados de una determinada manera. Es un conjunto estructurado de datos que representa entidades y sus interrelaciones, almacenados con la mínima redundancia y posibilitando el acceso a ellos eficientemente por parte de varias aplicaciones y usuarios.

## Distintos modelos de BBDD

## Que es SGBD. Funciones, componentes y tipos

**Sistema Gestor de Base de Datos**: Conjunto coordinado de programas, procedimientos, lenguajes, etc., que suministra, tanto a los usuarios no informáticos, como a los analistas programadores, o al administrador, los medios necesarios para describir y manipular los datos contenidos en la base de datos, manteniendo su integridad, confidencialidad y seguridad.

DDL

DML

DCL

## Bases de datos centralizadas y distribuías

## Sistemas gestores de bases de datos comerciales y libres

# UD02

Usuario: system

Contraseña: 14Febrero

WORKSPACE: PAULA

USERNAME: [GRANADARODRIGUEZ.PAULA@GMAIL.COM](mailto:GRANADARODRIGUEZ.PAULA@GMAIL.COM)

PASS: 14Febrero

## En run SQL command lite

Lo abrimos y nos conectamos



|  |
| --- |
| SQL> CONNECT SYSTEM/14Febrero |

Creamos el tablespace, buscando la carpeta correcta.

|  |
| --- |
| CREATE TABLESPACE NOMBRE DATAFILE 'C:\oraclexe\app\oracle\oradata\XE\NOMBRE.DBF' SIZE 5M;  *Tablespace created.* |

Para borrar un tablespace

|  |
| --- |
| DROP TABLESPACE NOMBRE INCLUDING CONTENTS AND DATAFILES; |

Para ver los usuarios

|  |
| --- |
| SELECT \* FROM ALL\_USERS; |

Borrar un usuario

|  |
| --- |
| DROP USER NOMBRE CASCADE; |

Creamos el usuario PAULA para el tablespace BIBLIOTECA

|  |
| --- |
| CREATE USER PAULA identified by pass default TABLESPACE BIBLIOTECA; |

De otorgamos privilegios

|  |
| --- |
| GRANT CONNECT, RESOURCE TO PAULA; |

Salimos

|  |
| --- |
| Exit; |

Entramos en SQL Developer



Introducimos los datos:

* Nombre de conexión: BIBLIOTECA
* Usuario: PAULA
* Contraseña: pass

Ejemplo:

|  |
| --- |
| SQL> CONNECT SYSTEM/14Febrero  Connected.  SQL> CREATE TABLESPACE CLINICA\_VETERINARIA DATAFILE 'C:\oraclexe\app\oracle\oradata\XE\CLINICA\_VETERINARIA.DBF' SIZE 2M;  Tablespace created.  SQL> GRANT CONNECT, RESOURCE TO PAULA;  Grant succeeded.  SQL> Exit; |

## Creando y modificando tablas

### Ejemplo Clínica veterinaria

|  |
| --- |
| --creamos las tablas  CREATE TABLE CLIENTES (  --Dos clientes no pueden llamarse igual.      NOMBRE VARCHAR2(10) UNIQUE,      APELLIDO1 VARCHAR2(15),      APELLIDO2 VARCHAR2(15),      NIF VARCHAR2(9),      NUMERO\_MAXIMO\_IMPAGADOS NUMBER(2),      SEXO VARCHAR2(1),      --el dominio del atributo sexo es M (mujer) y H (hombre).      CONSTRAINT clientes\_sexo\_CK CHECK (SEXO IN ('H','M')),      --el atributo Dirección no puede estar vacío      DIRECCION VARCHAR2(30)NOT NULL,      CONSTRAINT clientes\_nif\_PK PRIMARY KEY(NIF)  );  CREATE TABLE ANIMALES (      --Dos animales no pueden llamarse de la misma forma.      NOMBRE\_ANIMAL VARCHAR2(10) UNIQUE,      --Podremos diferenciar las tuplas de la tabla ANIMALES por el Código del Animal      COD\_ANIMAL NUMBER(5) PRIMARY KEY,      NIF\_CLIENTE VARCHAR2(9),      FECHA\_NACIMIENTO DATE,      FECHA\_1\_VACUNA DATE,      --En un animal, el número de chip es un dato que no puede faltar      NUM\_CHIP NUMBER(3) NOT NULL,          CONSTRAINT animal\_nif\_FK FOREIGN KEY (NIF\_CLIENTE)          --Se debe cumplir la regla de integridad referencial          REFERENCES CLIENTES (NIF) ON DELETE CASCADE,      --La Fecha de la 1ª vacuna del animal nunca puede ser menor que la Fecha de Nacimiento      CONSTRAINT animal\_fechas\_CK CHECK(FECHA\_1\_VACUNA>FECHA\_NACIMIENTO)  );  CREATE TABLE CONSULTAS (      MOTIVO VARCHAR2(10),      DIAGNOSTICO VARCHAR2(15),      TRATAMIENTO VARCHAR2(15),      NUM\_CONSULTA NUMBER PRIMARY KEY,      DIRECCION VARCHAR2(30),      FECHA\_CONSULTA DATE,      IMPORTE NUMBER(5),      ID\_ANIMAL NUMBER(5),      --La consulta debe asociarse a un animal antes de que se le pueda dar de alta.          CONSTRAINT consultas\_animales\_FK FOREIGN KEY (ID\_ANIMAL)              REFERENCES ANIMALES (COD\_ANIMAL)  );    --Crea un nuevo atributo llamado NUMERO\_DIENTES de tipo numérico en la tabla ANIMALES.  ALTER TABLE ANIMALES ADD NUMERO\_DIENTES NUMBER(3);  --Modifica el campo que has creado anteriormente para que el NUMERO\_DIENTES del animal esté comprendido entre 2 y 30 dientes.  ALTER TABLE ANIMALES ADD          CONSTRAINT animales\_dientes\_CK CHECK(NUMERO\_DIENTES BETWEEN 2 AND 30);   --Modifica el campo IMPORTE de la tabla CONSULTAS de manera que solo pueda haber importes de 40, 50 o 60 €.  ALTER TABLE CONSULTAS ADD          CONSTRAINT consultas\_importe\_CK CHECK (IMPORTE IN (40, 50, 60));  --No podemos añadir un cliente si su número máximo de impagados es mayor que 5.  ALTER TABLE CLIENTES ADD CONSTRAINT clientes\_max\_impagosCK CHECK(NUMERO\_MAXIMO\_IMPAGADOS >=5);  --limina la restricción que controla los valores que puede tomar el atributo Sexo.  ALTER TABLE CLIENTES DROP CONSTRAINT clientes\_sexo\_CK;  --Elimina columna Dirección de tabla CLIENTES.  ALTER TABLE CLIENTES DROP COLUMN DIRECCION;  --Cambia la clave primaria de la tabla CLIENTES por Nombre y Apellidos.  --Primero debemos eliminar la restricción donde se establecía la clave primaria de la tabla:  ALTER TABLE CLIENTES DROP CONSTRAINT clientes\_nif\_PK CASCADE;  --Ahora creamos una nueva:  ALTER TABLE CLIENTES ADD CONSTRAINT clientes\_nif\_PK PRIMARY KEY(NOMBRE, APELLIDO1, APELLIDO2);  --Renombra la tabla CLIENTES por PROPIETARIOS  RENAME CLIENTES TO PROPIETARIOS;  --Elimina la tabla CONSULTAS.  DROP TABLE CONSULTAS;  --Crea un usuario con tu nombre y clave ABC y dale todos los privilegios sobre la tabla ANIMALES  CREATE USER NOMBREDELALUMNO IDENTIFIED BY ABC;      GRANT ALL ON ANIMALES TO NOMBREDELALUMNO;  --Ahora al usuario anterior quítale permisos para modificar o actualizar la tabla ANIMALES  REVOKE ALTER, UPDATE ON ANIMALES FROM NOMBREDELALUMNO; |

### Ejemplo coches

|  |
| --- |
| *CREATE* *TABLE* CLIENTES (  *--En la tabla CLIENTES, el atributo Nombre no puede estar vacío*      NOMBRE *VARCHAR2*(10) *NOT* *NULL*,      APELLIDO1 *VARCHAR2*(15),      APELLIDO2 *VARCHAR2*(15),      NIF *VARCHAR2*(9),  *--Dos clientes no pueden tener igual número de teléfono.*      TELEFONO *VARCHAR2*(10) UNIQUE,  *CONSTRAINT* cliente\_nif\_PK *PRIMARY* *KEY*(NIF)  );  *CREATE* *TABLE* VENTAS (      MARCA *VARCHAR2*(10),      MODELO *VARCHAR2*(10),  *--Podremos diferenciar las tuplas de la tabla VENTAS por la matricula.*      MATRICULA *VARCHAR2*(8) *PRIMARY* *KEY*,      NIF\_CLIENTE *VARCHAR2*(9),      FECHA\_COMPRA *DATE*,  *--En un coche, el número de bastidor es un dato que no puede faltar*      NUM\_BASTIDOR *NUMBER*(3) *NOT* *NULL*,  *--La revisión debe asociarse a un coche antes de que se le pueda dar de alta.*  *--Se debe cumplir la regla de integridad referencial.*  *CONSTRAINT* ventas\_nif\_FK *FOREIGN KEY* (NIF\_CLIENTE)  *REFERENCES* CLIENTES (NIF) *ON DELETE CASCADE*    );  *CREATE* *TABLE* REVISIONES (      TIPO\_REVISION *VARCHAR2*(10),      TURNO *VARCHAR2*(1),  *--Podremos diferenciar las tuplas de la tabla REVISIONES por el Identificativo*      NUM\_REVISION *NUMBER*(6) *PRIMARY* *KEY*,      KMS\_ACTUALES *NUMBER*(9),      KMS\_PROX\_REV *NUMBER*(9),      FECHA\_REVISION *DATE*,      IMPORTE *NUMBER*(5),      MATRICU *VARCHAR2*(8),  *--Los Km de la próxima revisión han de ser siempre mayores que los Km actuales*  *CONSTRAINT* REVISIONES\_KMS\_CK *CHECK*(KMS\_PROX\_REV*>*KMS\_ACTUALES),  *--En la tabla REVISIONES el dominio del atributo TURNO es M (mañana) y T (tarde).*  *CONSTRAINT* REVISIONES\_TURNO\_CK *CHECK* (TURNO *IN* ('M','T')),  *--La revisión debe asociarse a un coche antes de que se le pueda dar de alta.*  *CONSTRAINT* REVISIONES\_VENTAS\_FK *FOREIGN KEY* (MATRICU)  *REFERENCES* VENTAS (MATRICULA)  );  *--Crea un nuevo atributo llamado PRECIO de tipo numérico en la tabla VENTAS.*  *ALTER* *TABLE* VENTAS ADD PRECIO *NUMBER*(7);  *--Modifica el campo que has creado anteriormente para que el PRECIO del coche esté comprendido entre 10.000 y 30.000€.*  *ALTER* *TABLE* VENTAS ADD  *CONSTRAINT* ventas\_precio\_CK *CHECK*(PRECIO *BETWEEN* 10000 *AND* 30000);  *--Modifica el campo IMPORTE de la tabla REVISIONES de manera que solo pueda haber importes de 100 o 200 €.*  *ALTER* *TABLE* REVISIONES ADD  *CONSTRAINT* revisiones\_importe\_CK *CHECK* (IMPORTE *IN* (100, 200));  *--No podemos añadir una revisión si su importe excede los 10.000€.*  *ALTER* *TABLE* REVISIONES ADD *CONSTRAINT* revisiones\_importe2\_CK *CHECK*(IMPORTE *>*10000);  *--Elimina la restricción que controla los valores que puede tomar el atributo TURNO.*  *ALTER* *TABLE* REVISIONES *DROP* *CONSTRAINT* REVISIONES\_TURNO\_CK;  *--Elimina columna TELEFONO de tabla CLIENTES.*  *ALTER* *TABLE* CLIENTES *DROP* COLUMN TELEFONO;  *--Cambia la clave primaria de la tabla CLIENTES por Nombre y Apellidos*  *--Primero debemos eliminar la restricción donde se establecía la clave primaria de la tabla:*  *ALTER* *TABLE* CLIENTES *DROP* *CONSTRAINT* cliente\_nif\_PK CASCADE;  *--Ahora creamos una nueva:*  *ALTER* *TABLE* CLIENTES ADD *CONSTRAINT* client\_nif\_PK *PRIMARY* *KEY*(NOMBRE, APELLIDO1, APELLIDO2);  *--Elimina la tabla REVISIONES.*  *DROP* *TABLE* REVISIONES;  *--Renombra la tabla CLIENTES por PROPIETARIOS.*  RENAME CLIENTES *TO* PROPIETARIOS; |