

BASES DE DATOS

NoSQL

Not Only SQL.

Características:

- No hay relaciones (no hay joins) para que se pueda almacenar la información de forma distribuida. Son escalables.
- No hay restricción.
- Es más rápido.
- Montan un único elemento que se le llama agregado.

Más utilizada para Big Data.

No cumplen el principio ACID.

SQL

Lenguaje de consultas estructurado.

Software para administrar información.

Ventaja poder consultar la información que está guardada.

Los datos van a estar en tablas. Las tablas van a estar en un esquema. Se crean DDL.

Operaciones de tipo CRUD → DML

- Create. Los insert que vamos a realizar para llenar las tablas.
- Read. Select. Consultar la información que se ha guardado en las tablas.
- Update. Update. Actualización.
- Delete. Operaciones de delete.

DML. Data Manipulation

DDL. Data Definition Lenguaje.

Primary Key es el ID. Identificar único de cada registro que hay en la tabla.

Foreign Key. La clave de relación entre las tablas

- MySQL
- Oracle
- SQL Server
- MariaDB

Características:

- Tablas relacionadas entre tablas.
- Tenemos restricciones.
 - En los campos

- Tipos asignados a las columnas
- Primary Key no puede tener duplicados
- Foreign Key tiene que mantener la relación entre las tablas

Impedancia. Los tipos de datos que hay en una tabla a veces no va a coincidir con los elementos que tenemos en un lenguaje de programación. Por ejemplo los booleanos. No son escalables debido a la relación que hay entre las tablas.

Sólo admite información estructurada por lo que no sirve para Big Data. En Big Data pueden ser semi estructurados, no tienen una estructura definida.

Todas cumplen el principio **ACID**.

- Atomicity. Atomicidad. Cuando haya un fallo tiene que echarse todo para atrás. Si todos los pasos seguidos no son correctos no se debería guardar la base de datos. O todo o nada. Lo que definimos en todas las consultas (insert, delete, update), al final se va a ejecutar un Commit. En este punto se queda guardado en la base de datos. Si hay un error en algún punto dentro de la transacción (conjunto de instrucciones ejecutadas que damos como inserts, deletes, updates o selects), todo lo que se ha hecho anteriormente se echa para atrás (rollback). Así aseguramos que la información está correcta.
- Consistency. Consistencia. Me asegura que la información sea del mismo tipo. Los campos tienen que ser consistentes, el mismo tipo y formato.
- Isolation. Aislamiento. Hay ocasiones en las que las bases de datos están siendo utilizadas por diferentes apps. Lo que se intenta es bloquear la tabla cuando una aplicación la está utilizando por lo que no puede utilizarla la APP2 hasta que no termine la APP1.
- Durability. Durabilidad. La información se mantiene ahí, no se pierde.

MySQL Command Line.

```

MySQL 8.0 Command Line Client
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 8.0.34 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

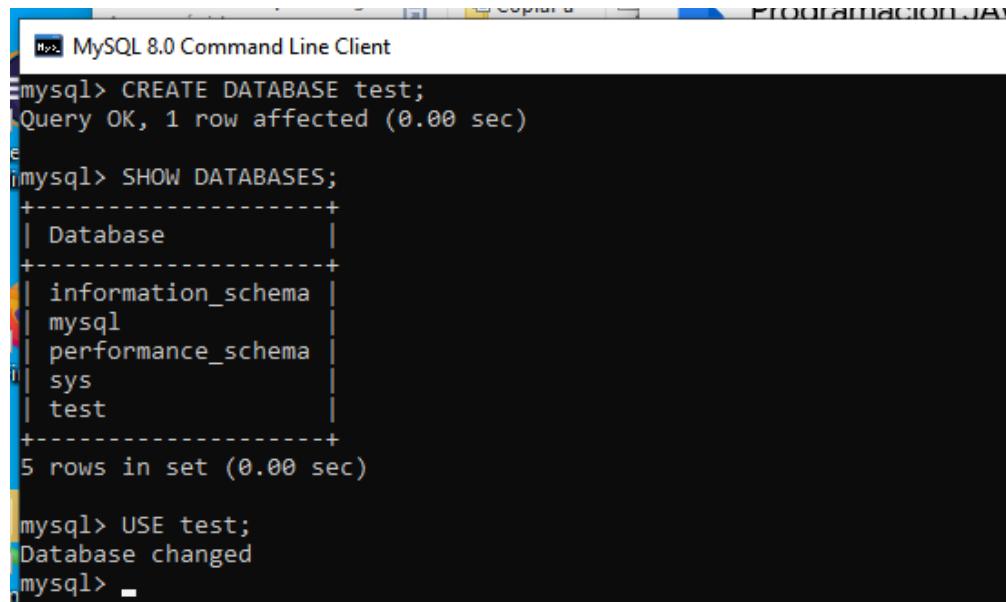
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW DATABASES;
+-----+
| Database      |
+-----+
| information_schema |
| mysql          |
| performance_schema |
| sys            |
+-----+
4 rows in set (0.00 sec)

```

Hay que terminar todos los comandos con ;
 SHOW DATABASES; Muestra las bases de datos
 CREATE DATABASE test; Para crear base de dato

USE test; Para modificarlo



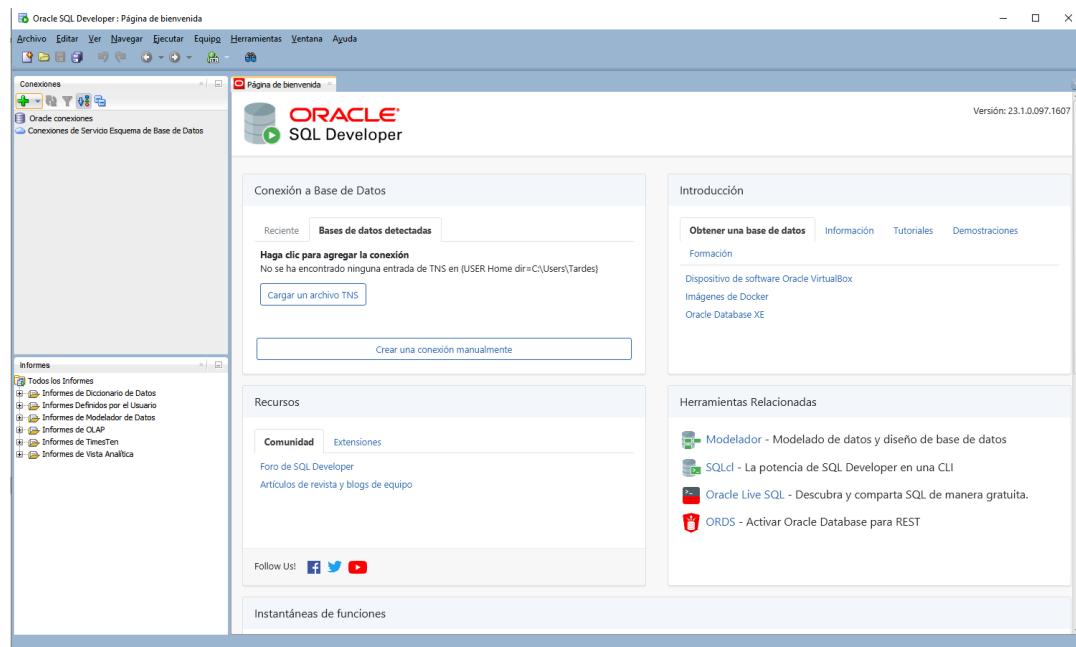
```
MySQL 8.0 Command Line Client

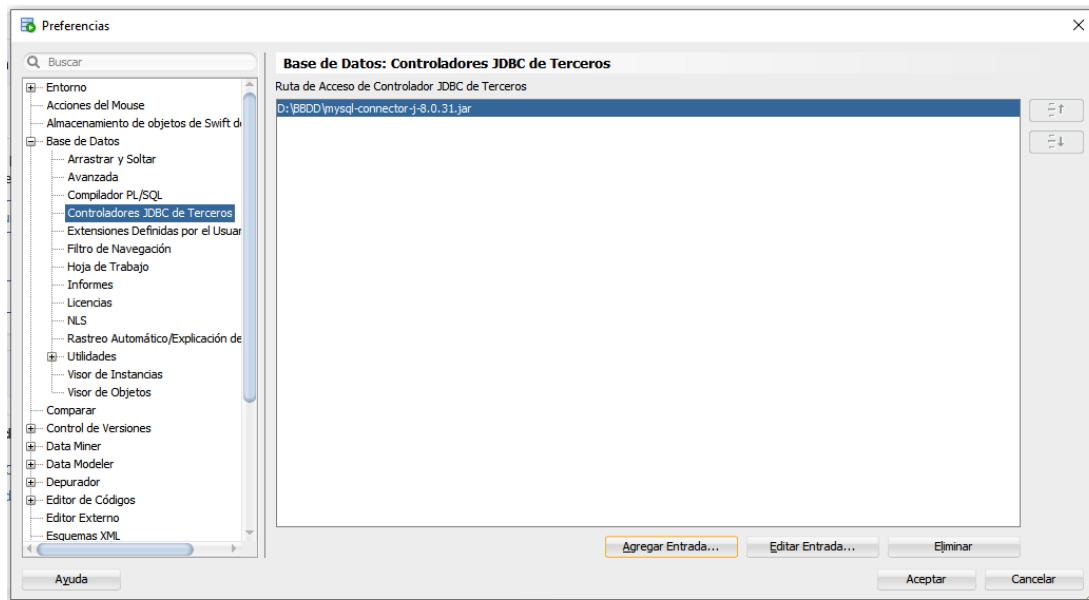
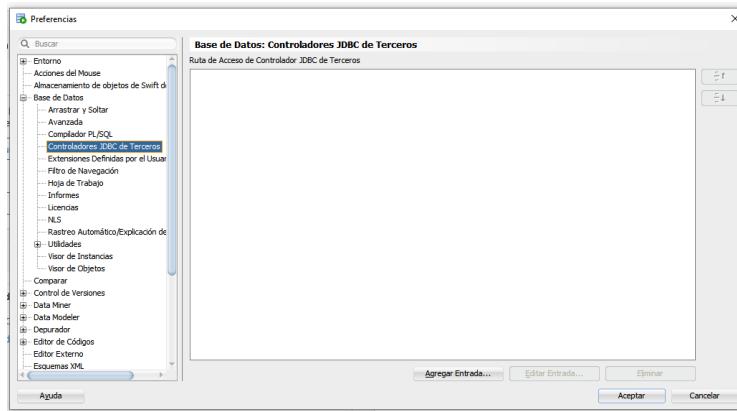
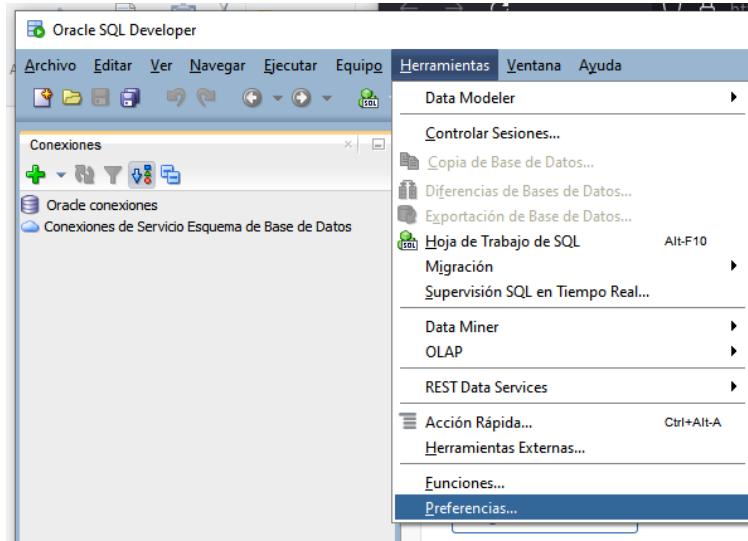
mysql> CREATE DATABASE test;
Query OK, 1 row affected (0.00 sec)

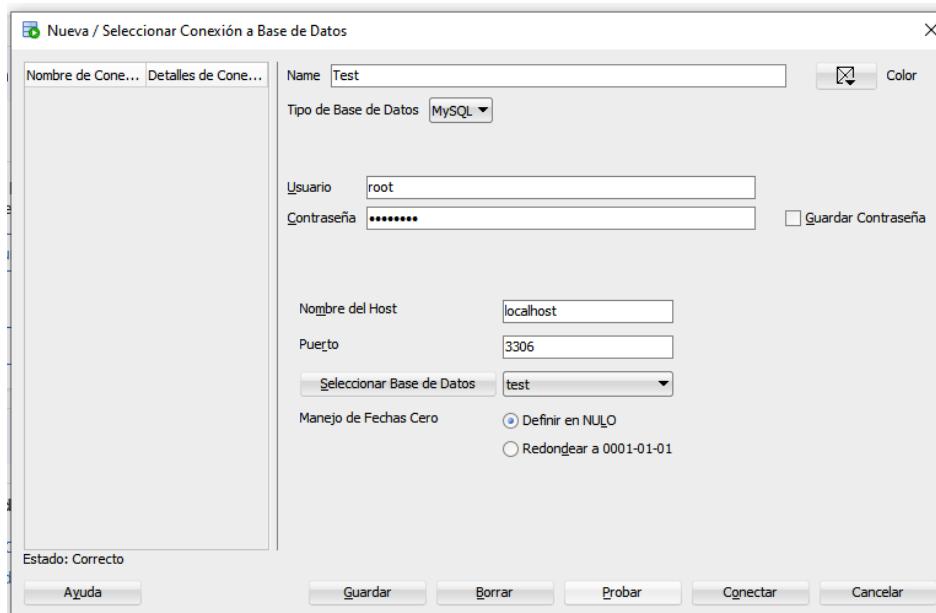
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| test |
+-----+
5 rows in set (0.00 sec)

mysql> USE test;
Database changed
mysql>
```

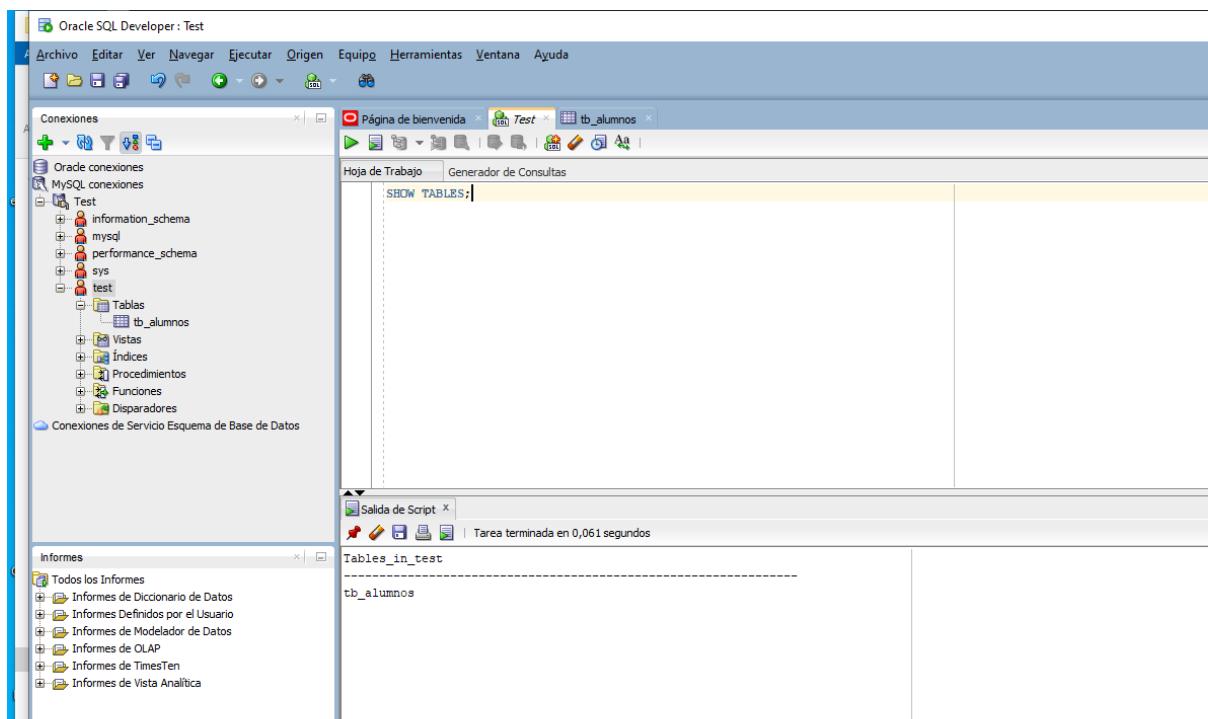
Oracle SQL Developer



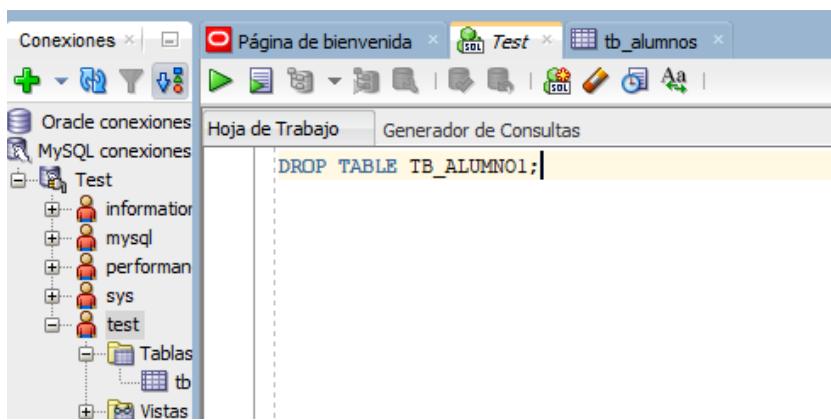




SHOW TABLES; (botón play) Mirar tablas creadas

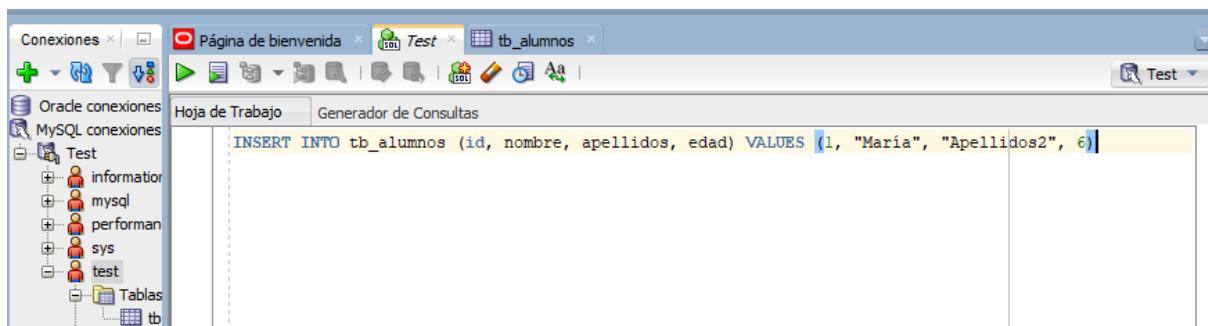


Drop - borra la tabla
Truncate - deja la tabla vacía

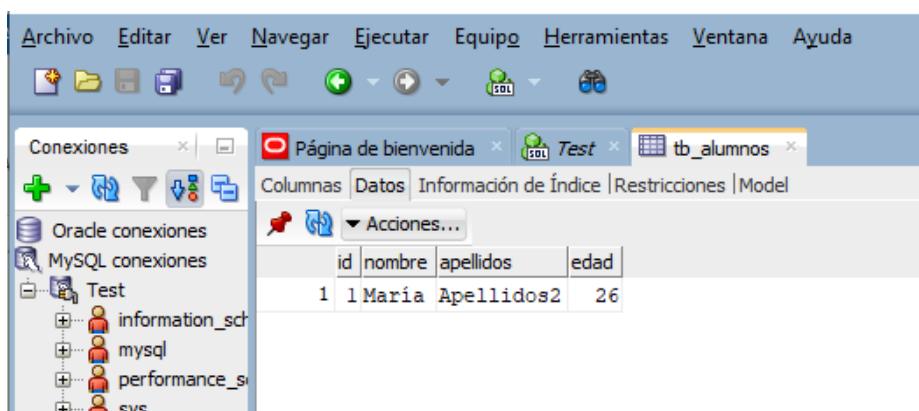


Cómo insertar información en una tabla

Insert. Siempre tiene que haber el mismo número de datos que de valores.



Para ver los datos insertados:



Si intentamos insertar otro dato con el mismo id nos da error.

The screenshot shows the MySQL Workbench interface. In the top tab bar, the active tab is 'tb_alumnos'. The main workspace contains a SQL editor with the following query:

```
INSERT INTO tb_alumnos (id, nombre, apellidos, edad) VALUES (1, "Pedro", "Apellidos1", 28)
```

Below the SQL editor is a 'Salida de Script' (Script Output) pane. It displays the output of the query execution:

```
Tables_in_test
tb_alumnos

1 fila insertadas.

Error que empieza en la linea: 1 del comando -
INSERT INTO tb_alumnos (id, nombre, apellidos, edad) VALUES (1, "Pedro", "Apellidos1", 28)
Error en la linea de comandos : 1 Columna : 1
Informe de error -
Error SQL: Duplicate entry '1' for key 'tb_alumnos.PRIMARY'
```

Para ver las restricciones

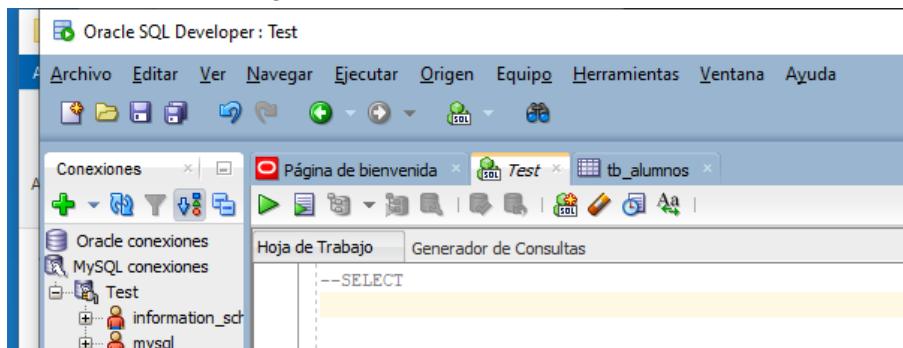
The screenshot shows the MySQL Workbench interface with the 'tb_alumnos' table selected. The 'Restricciones' tab is active. The table data is as follows:

CONSTRAINT_NAME	CONSTRAINT_TYPE
1 apellidos	UNIQUE
2 PRIMARY	PRIMARY KEY

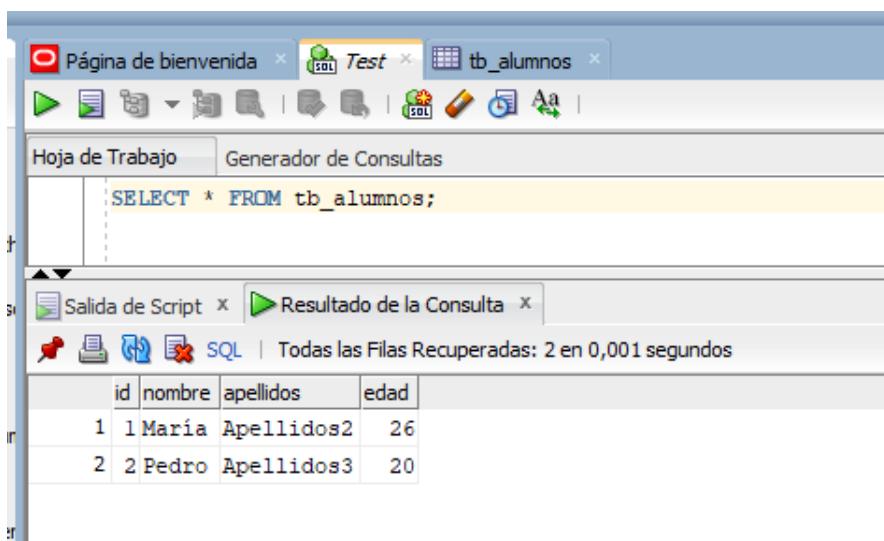
Para guardar

The screenshot shows the MySQL Workbench interface with the 'tb_alumnos' table selected. A 'Guardar' (Save) dialog box is open in the foreground. The dialog box shows the save location as 'C:\Users\Tardes' and the file name as 'Test.sql'. The 'Tipo de Archivo' (File Type) is set to 'Script SQL (*.sql)'. The background shows the MySQL Workbench interface with the 'Conexiones' (Connections) and 'Informes' (Reports) panes visible.

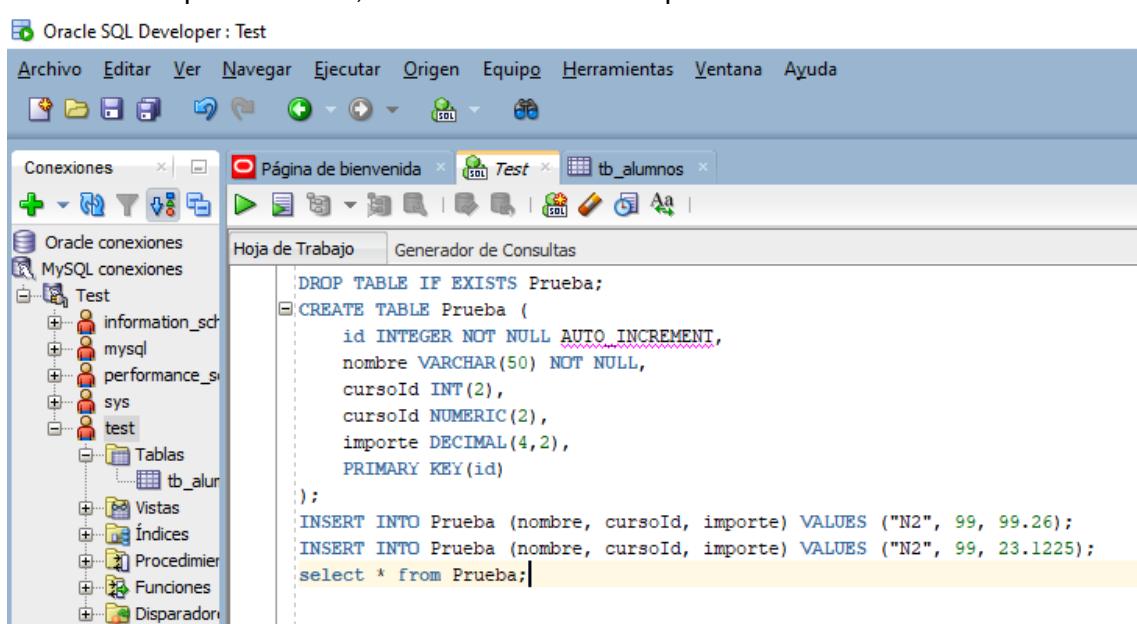
Para comentar (dos guiones)



Para mostrar los datos



Decimal (4 números enteros de los cuales 2 van a ser decimales). Aunque tenga más decimales no pasaría nada, sólo insertaría los dos primeros decimales.



Ejercicio curso

The screenshot shows the MySQL Workbench interface. On the left, the 'Conexiones' sidebar lists 'Oracle conexiones' and 'MySQL conexiones'. Under 'MySQL conexiones', there is a connection named 'Curso' which is expanded to show 'curso' schema, containing 'Tablas' (tb_facilities), 'Vistas', 'Índices', 'Procedimientos', 'Funciones', and 'Disparadores'. Other schemas listed include 'information_schema', 'mysql', 'performance_schema', 'sys', and 'test'. Below the schemas is a 'Test' section with the same schema structure. On the right, the main workspace shows the 'tb_facilities' table with the following data:

id	guid	name	description	category	address	height
1	1 1234567	Planta1	Planta de Urgencias	Categoría especial	Direccion1	25.50
2	2 7654321	Planta2	Planta de Digestivo	Categoría1	Direccion2	55.90
3	3 3547689	Planta3	Planta de Psicología	Categoría2	Direccion3	85.78
4	4 8374019	Planta4	Planta de Neurología	Categoría3	Direccion4	100.90
5	5 4567890	Planta5	Planta de Enfermería	Categoría4	Direccion5	125.30

Below the table, a query editor window titled 'Curso~1' contains the SQL query: `select * from tb_facilities where height > 100`. The results of this query are shown in the 'Resultado de la Consulta' tab, displaying the same five rows as the table.

This screenshot shows another session in MySQL Workbench. The 'Hoja de Trabajo' tab contains the SQL query: `select NAME, ID, HEIGHT from tb_facilities where height > 100`. The results are displayed in the 'Resultado de la Consulta' tab, showing the following data:

name	id	height
1 Planta4	4	100.90
2 Planta5	5	125.30

En Oracle se llaman secuencias como auto_increment.

Para que muestre sólo un tipo de datos.

This screenshot shows a query in MySQL Workbench that retrieves data of a single type. The 'Hoja de Trabajo' tab contains the SQL query: `select ID, NAME from tb_facilities`. The results are shown in the 'Resultado de la Consulta' tab, displaying the following data:

id	name
1	1 Plantal
2	2 Planta2
3	3 Planta3
4	4 Planta4
5	5 Planta5

Se pueden poner más condiciones

The screenshot shows the MySQL Workbench interface. In the top tab bar, there are three tabs: 'Curso1.sql', 'SQL' (highlighted in green), and 'tb_facilities'. Below the tabs is a toolbar with icons for running queries, saving, and zooming. The main area has two panes: 'Hoja de Trabajo' (Worksheet) and 'Generador de Consultas' (Query Builder). The worksheet pane contains the SQL query:select NAME, ID, HEIGTH, CATEGORY from tb_facilities where heigth > 100 AND category='Categoria3'The results pane below it shows the output of the query:| name | id | heigth | category |
| --- | --- | --- | --- |
| 1 | Planta4 | 4 | 100.90 | Categoría3 |

Se pueden ordenar según el campo que quieras de ascendente o descendente, en el caso de un String es alfabético (ascendente a-z y descendente z-a).
Además se pueden limitar los resultados a mostrar (LIMIT)

The screenshot shows two separate sessions in MySQL Workbench. Both sessions have tabs for 'Curso1.sql', 'SQL' (highlighted in green), and 'tb_facilities'. The first session (top) has a worksheet with the following query:1 select NAME, ID, HEIGTH, CATEGORY from tb_facilities
2 where heigth > 100
3 ORDER BY heigth DESC;The results show two rows:| name | id | heigth | category |
| --- | --- | --- | --- |
| 1 | Planta5 | 5 | 125.30 | Categoría4 |
| 2 | Planta4 | 4 | 100.90 | Categoría3 |
The second session (bottom) has a worksheet with this query:1 select NAME, ID, HEIGTH, CATEGORY from tb_facilities
2 where heigth > 100
3 ORDER BY heigth DESC
4 LIMIT 1;The results show one row:| name | id | heigth | category |
| --- | --- | --- | --- |
| 1 | Planta5 | 5 | 125.30 | Categoría4 |

En la creación de una tabla, se puede poner, por ejemplo:
tipo ENUM ('Nave', 'Tanque')

Si quiero modificar una columna se puede poner lo siguiente:

```
UPDATE TB_GUERRERO SET tipo = "Soldado";
```

Si fuera en algunos, se pondría WHERE (y la condición)

Para eliminar toda la tabla

```
DELETE FROM TB_GUERRERO;
```

Si quiero eliminar alguna inserción

```
DELETE FROM TB_GUERRERO WHERE vehiculo_id=2;
```

Para poner una foreign key (restricción) donde queremos que los guerreros tengan una referencia a qué nave pertenecen. Por ello, una vez declarada la foreign key, si hay que borrar una nave, primero se tendrían que eliminar los guerreros que tengan referenciada esa nave y luego la nave de la tabla vehículos.

Sugerencia para la foreign key es ponerlo en el nombre del campo (fk_)

```
fk_vehiculo_guerra INTEGER NOT NULL,  
FOREIGN KEY (fk_vehiculo_guerra) REFERENCES TB_VEHICULOS_GUERRA (id)
```

Para poner alias a las tablas

```
SELECT vehiculo.nombre FROM TB_VEHICULOS_GUERRA as vehiculo;
```

o

```
SELECT v.nombre FROM TB_VEHICULOS_GUERRA as v;
```

Para que nos saque el nombre del guerrero y que nos indique en qué nave se encuentra

```
SELECT v.nombre, g.nombre FROM TB_VEHICULOS_GUERRA as v
```

```
INNER JOIN TB_GUERRERO as g ON g.fk_vehiculo_guerra = v.id;
```

Para saber el número de registros de una tabla

```
SELECT COUNT (ID) FROM TB_VEHICULOS_GUERRA;
```

Bases de datos que tiene, en qué casos se usa. Hacer ejemplo tras la instalación.

```
--CONSULTAS CON CONDICIONES
SELECT * FROM TB_GUERRERO
WHERE FK_VEHICULO_GUERRA = 1
ORDER BY ID DESC, NOMBRE ASC
LIMIT 3;
```

Busca en la tabla guerreros dándole varias condiciones:
donde el vehículo de guerra sea 1, ordenado por id descendente, con el nombre ascendente
y un límite de 3 registros.

Sumatorio de todas las fuerzas y la media de todos los registros.

The screenshot shows a MySQL Workbench interface. In the SQL editor, there is a single query:

```
--CONSULTAS CON CONDICIONES
SELECT SUM(FUERZA), AVG(RESISTENCIA) FROM TB_GUERRERO;
```

Below the editor is the "Resultado de la Consulta" (Result of the Query) window. It displays the following information:

- Icon bar: Includes icons for error, warning, and SQL.
- Status message: "SQL | Todas las Filas Recuperadas: 1 en 0,003 segundos"
- Table header: "SUM(FUERZA) AVG(RESISTENCIA)"
- Data row: "1 120 4.0000"

GROUP BY - Operación de agregación/agrupación. Agrupar registro a través de un campo que tenga el mismo valor y hacer una operación con esos mismos datos.
Por ejemplo, tener el sumatorio de las fuerzas de los guerreros del vehículo 1 y los guerreros del vehículo 2.

The screenshot shows a MySQL Workbench interface. In the SQL editor, there is a query:

```
--CONSULTAS CON CONDICIONES
SELECT SUM(FUERZA) as SUMA_FUERZA, AVG(RESISTENCIA), VEHICULO_ID
FROM TB_GUERRERO
GROUP BY VEHICULO_ID;
```

Below the editor is the "Resultado de la Consulta" (Result of the Query) window. It displays the following information:

- Icon bar: Includes icons for error, warning, and SQL.
- Status message: "SQL | Todas las Filas Recuperadas: 2 en 0,001 segundos"
- Table header: "SUMA_FUERZA AVG(RESISTENCIA) vehiculo_id"
- Data rows:

	SUMA_FUERZA	Avg(RESISTENCIA)	vehiculo_id
1	50	5.0000	1
2	70	3.0000	2

Sólo se puede usar el campo el mismo campo del Group by y/o alguna acción (sumatorio, media) de un grupo de datos.

```
--CONSULTAS CON CONDICIONES
SELECT SUM(FUERZA) as SUMA_FUERZA, AVG(RESISTENCIA), VEHICULO_ID, NOMBRE
FROM TB_GUERRERO
GROUP BY VEHICULO_ID;
```

Resultado de la Consulta

SQL | En Ejecución:SELECT SUM(FUERZA) as SUMA_FUERZA, AVG(RESISTENCIA), VEHICULO_ID, NOMBRE FROM TB_GUERRERO GROUP BY VEHICULO_ID;

Expression #4 of SELECT list is not in GROUP BY clause and contains nonaggregated column 'curso.TB_GUERRERO.nombre' which is not functionally dependent on columns in GROUP BY clause.

Cuando queremos que sólo nos muestre el sumatorio de la fuerza y la media de los guerreros de uno de los vehículos.

```
--CONSULTAS CON CONDICIONES
SELECT SUM(FUERZA) as SUMA_FUERZA, AVG(RESISTENCIA), VEHICULO_ID
FROM TB_GUERRERO
WHERE ID = 1
GROUP BY VEHICULO_ID;
```

Resultado de la Consulta

SQL | Todas las Filas Recuperadas: 1 en 0,001 segundos

	SUMA_FUERZA	Avg(RESISTENCIA)	vehiculo_id
1	5	5.0000	1

Si queremos indicar de los guerreros que quiero se puede indicar entre paréntesis. En este caso sólo los del vehículo 1 (guerreros del 1 al 10).

```
--CONSULTAS CON CONDICIONES
SELECT SUM(FUERZA) as SUMA_FUERZA, AVG(RESISTENCIA), VEHICULO_ID
FROM TB_GUERRERO
WHERE ID IN (1,2,3,4,5,6,7,8,9,10)
GROUP BY VEHICULO_ID;
```

Resultado de la Consulta

SQL | Todas las Filas Recuperadas: 1 en 0,001 segundos

	SUMA_FUERZA	Avg(RESISTENCIA)	vehiculo_id
1	50	5.0000	1

Si indicamos los guerreros del 1 al 11, como el 11 es de la nave 2 nos muestra el sumatorio y la media de los guerreros indicados.

```
--CONSULTAS CON CONDICIONES
SELECT SUM(FUERZA) as SUMA_FUERZA, AVG(RESISTENCIA), VEHICULO_ID
FROM TB_GUERRERO
WHERE ID IN (1,2,3,4,5,6,7,8,9,10,11)
GROUP BY VEHICULO_ID;
```

Resultado de la Consulta x

SQL | Todas las Filas Recuperadas: 2 en 0,001 segundos

	SUMA_FUERZA	Avg(RESISTENCIA)	vehiculo_id
1	50	5.0000	1
2	7	3.0000	2

No se pueden consultar datos a partir de un alias.

```
--CONSULTAS CON CONDICIONES
SELECT SUM(FUERZA) as SUMA_FUERZA, AVG(RESISTENCIA), VEHICULO_ID
FROM TB_GUERRERO
WHERE ID BETWEEN 1 AND 11
GROUP BY VEHICULO_ID;
```

sultado de la Consulta x

SQL | Todas las Filas Recuperadas: 2 en 0,001 segundos

	SUMA_FUERZA	Avg(RESISTENCIA)	vehiculo_id
1	50	5.0000	1
2	7	3.0000	2

CÓMO ACTUALIZAR Y ELIMINAR DATOS

Para borrar todos los registros donde el vehículo sea 1.

```
--DELETES
DELETE FROM TB_GUERRERO WHERE VEHICULO_ID=1;
```

Si quiero borrar el último registro

```
--DELETES
DELETE FROM TB_GUERRERO WHERE ID = (SELECT ID FROM TB_GUERRERO ORDER BY ID DESC, LIMIT 1);
```

Si quiero modificar se utiliza el UPDATE con SET. Dependiendo el qué le pondremos condiciones.

```
--UPDATES
UPDATE TB_GUERRERO
SET FUERZA=4, RESISTENCIA=6
WHERE NOMBRE='GUERRERO1';

1 fila actualizadas.
```

Actualizar varios registros del vehículo uno y mostrar sólo los 5 primero ordenador por resistencia descendente, fuerza ascendente y nombre descendente. Primero tendrá en cuenta la fuerza al ser la primera condición y luego la resistencia y, por último, el nombre.

```
UPDATE TB_GUERRERO
SET FUERZA=7, RESISTENCIA=2
WHERE NOMBRE='GUERRERO10';

SELECT ID, NOMBRE, FUERZA, RESISTENCIA, VEHICULO_ID FROM TB_GUERRERO
WHERE VEHICULO_ID = 1
ORDER BY FUERZA DESC, RESISTENCIA ASC, NOMBRE ASC
LIMIT 5;
```

id	nombre	fuerza	resistencia	vehiculo_id
1	10 GUERRERO10	7	2	1
2	3 GUERRERO3	7	3	1
3	4 GUERRERO4	5	4	1
4	2 GUERRERO2	5	5	1
5	5 GUERRERO5	5	5	1

Si cambiamos y ponemos primero la resistencia, será el primer parámetro que tendrá en cuenta. Después la fuerza y el nombre.

```

UPDATE TB_GUERRERO
SET FUERZA=7, RESISTENCIA=2
WHERE NOMBRE IN ('GUERRERO10','GUERRERO6');

SELECT ID, NOMBRE, FUERZA, RESISTENCIA, VEHICULO_ID FROM TB_GUERRERO
WHERE VEHICULO_ID = 1
ORDER BY RESISTENCIA DESC, FUERZA DESC, NOMBRE ASC
LIMIT 5;

```

Salida de Script | Resultado de la Consulta | Resultado de la Consulta 1

SQL | Todas las Filas Recuperadas: 5 en 0,001 segundos

id	nombre	fuerza	resistencia	vehiculo_id
1	GUERRERO1	4	6	1
2	GUERRERO2	5	5	1
3	GUERRERO5	5	5	1
4	GUERRERO6	5	5	1
5	GUERRERO7	5	5	1

Para cambiar varios a la vez:

```

UPDATE TB_GUERRERO
SET FUERZA=7, RESISTENCIA=2
WHERE NOMBRE IN ('GUERRERO10', 'GUERRERO6');

```

Para que muestren todos los datos de la biblioteca (toda la información de la biblioteca incluyendo la dirección)

```

SELECT biblio.id, biblio.nombre, dir.tipo_direccion, dir.direccion, dir.ciudad, dir.provincia,
dir.cp FROM TB_BIBLIOTECA as biblio JOIN TB_DIRECCION as dir ON fk_direccion =
dir.id

```

Si solo se escribe el ON lo que hace es que relaciona la biblioteca 1 con todas las direcciones, la biblioteca 2 con todas las direcciones, etc.

Para añadir una columna más en la tabla

```

ALTER TABLE TB_LIBROS ADD precio INT NOT NULL DEFAULT 1 AFTER fk_biblioteca;

```

Para modificar el nombre de la columna

```

ALTER TABLE TB_LIBROS ADD precio INT NOT NULL DEFAULT 1 AFTER fk_biblioteca;
ALTER TABLE TB_LIBROS RENAME COLUMN PRECIO TO precio;

```

```
SELECT if(heigth>100,"MUY ALTO","MUY BAJO"), heigth*1.20
FROM TB_FACILITIES
```

Resultado de la Consulta

SQL | Todas las Filas Recuperadas: 5 en 0,037 segundos

	if(heigth>100,"MUY ALTO","MUY BAJO")	heigth*1.20
1	MUY BAJO	30.6000
2	MUY BAJO	67.0800
3	MUY BAJO	102.9360
4	MUY ALTO	121.0800
5	MUY ALTO	150.3600

Lo que está entre paréntesis funciona como un ternario.

Cuando se realiza algún group by se tiene que poner alguna operación de agregación (media, sumatorio, etc.) o algo que esté en la columna.

Función Round → redondea a los decimales que le indicamos

```
Query query = em.createQuery(
    "select po.tipo, round (avg(po.precio),2) as precioMedio, sum(po.unidades) "
    + "as totalUnidades from ProductoOracle po group by po.tipo");
```

Conectarse a una base de datos de otra máquina

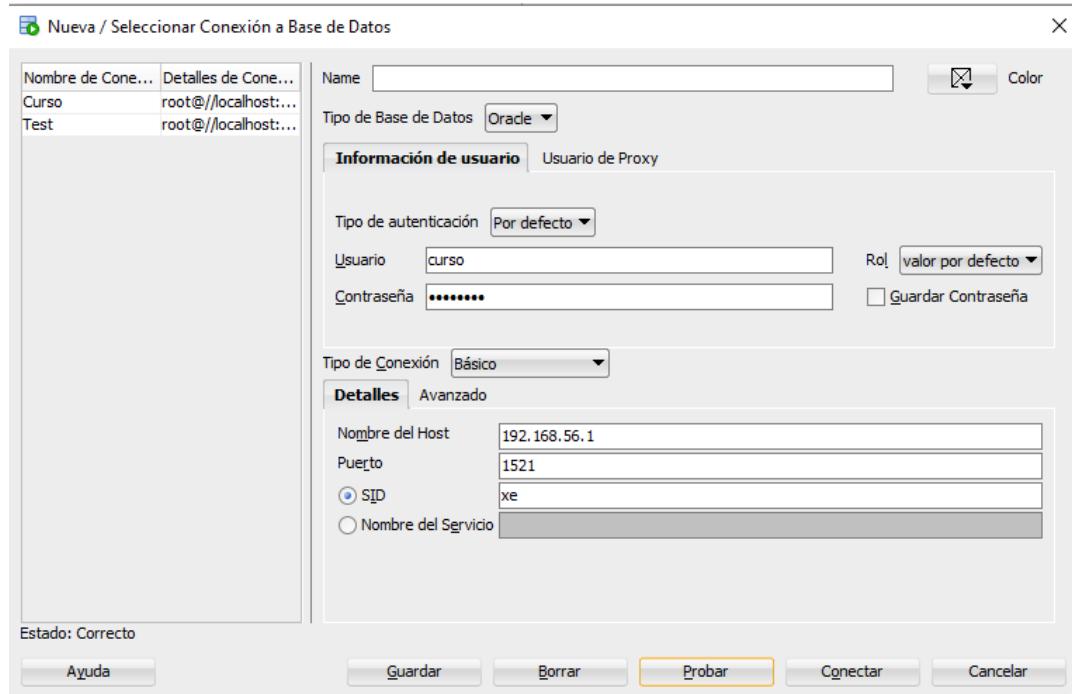
```
Símbolo del sistema
Microsoft Windows [Versión 10.0.19045.3693]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Tardes>ping 192.168.56.1

Haciendo ping a 192.168.56.1 con 32 bytes de datos:
Respuesta desde 192.168.56.1: bytes=32 tiempo<1m TTL=128

Estadísticas de ping para 192.168.56.1:
Paquetes: enviados = 4, recibidos = 4, perdidos = 0
(0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
Mínimo = 0ms, Máximo = 0ms, Media = 0ms

C:\Users\Tardes>
```



`SELECT SYSDATE FROM DUAL;`

Tabla dual, una tabla que tiene Oracle Sysdate, se puede extraer mucha información pero que no sabes las columnas que tiene.

SECUENCIAS

Select SYSDATE FROM DUAL;

`SELECT HIBERNATE_SEQUENCE.NEXTVAL FROM DUAL` - Indica cuál es el siguiente valor a incluir en la secuencia, mueve el cursor al siguiente pero no inserta nada, me devuelve el siguiente valor. Si tengo 3 datos, el nextval me mueve el cursor al 4 y ya se incluiría el valor en el 5.

En un insert, por ejemplo en usuario para que continúe la secuencia

`INSERT INTO TB_USUARIO (id, apellidos, dni, FECHA_ALTA, nombre) values (hibernate_sequence.nextval, "APE1", "", sysdate, "NOMBRE");`

`SELECT HIBERNATE_SEQUENCE.CURRVAL FROM DUAL` -

Nos da el valor actual de la secuencia. Primero hay que hacer un nextval.

En el ejercicio UAUUNUANOTMain

Si pones `@OneToOne (cascade = {CascadeType.ALL}, optional = false)` obliga a que se le ponga dirección a persona de forma obligatoria.

Ejercicio BIBLIOTECA

```
CREATE TABLE TB_DIRECCION (
    id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    tipo_direccion ENUM('calle', 'plaza', 'travesía', 'avenida', 'carretera', 'otro') NOT NULL
COMMENT "Calle, plaza, travesía, avenida, carretera, otro.",
    direccion VARCHAR(40),
    ciudad VARCHAR(20),
    provincia VARCHAR(20),
    cp NUMERIC(5)
);
```

```
CREATE TABLE TB_BIBLIOTECA (
    id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(20) NOT NULL,
    fk_direccion INT NOT NULL,
    FOREIGN KEY (fk_direccion) REFERENCES TB_DIRECCION (id)
);
```

```
CREATE TABLE TB_LIBROS (
    id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    titulo VARCHAR(30) NOT NULL,
    autor VARCHAR(20),
    ISBN NUMERIC(10),
    fk_biblioteca INT NOT NULL,
    FOREIGN KEY (fk_biblioteca) REFERENCES TB_BIBLIOTECA (id)
);
```

```
INSERT INTO TB_DIRECCION (tipo_direccion, direccion, provincia, ciudad, cp) VALUES
("Calle","Direccion1","Provincia1","Ciudad1",11111 );
INSERT INTO TB_DIRECCION (tipo_direccion, direccion, provincia, ciudad, cp) VALUES
("Travesía","Direccion2","Provincia2","Ciudad2",22222 );
INSERT INTO TB_DIRECCION (tipo_direccion, direccion, provincia, ciudad, cp) VALUES
("Avenida","Direccion3","Provincia3","Ciudad3",33333 );
```

```
INSERT INTO TB_BIBLIOTECA (nombre, fk_direccion) VALUES ("Biblioteca1",2);
INSERT INTO TB_BIBLIOTECA (nombre, fk_direccion) VALUES ("Biblioteca2",1);
INSERT INTO TB_BIBLIOTECA (nombre, fk_direccion) VALUES ("Biblioteca3",3);
```

```
INSERT INTO TB_LIBROS (titulo,autor,ISBN,fk_biblioteca) VALUES  
("Titulo1","Autor1",1234567890,2);  
INSERT INTO TB_LIBROS (titulo,autor,ISBN,fk_biblioteca) VALUES  
("Titulo2","Autor2",0987654321,2);  
INSERT INTO TB_LIBROS (titulo,autor,ISBN,fk_biblioteca) VALUES  
("Titulo3","Autor3",0977644321,2);  
INSERT INTO TB_LIBROS (titulo,autor,ISBN,fk_biblioteca) VALUES  
("Titulo4","Autor4",1277634021,1);  
INSERT INTO TB_LIBROS (titulo,autor,ISBN,fk_biblioteca) VALUES  
("Titulo5","Autor5",4297604322,1);  
INSERT INTO TB_LIBROS (titulo,autor,ISBN,fk_biblioteca) VALUES  
("Titulo6","Autor6",7291305847,1);  
INSERT INTO TB_LIBROS (titulo,autor,ISBN,fk_biblioteca) VALUES  
("Titulo7","Autor7",8231408940,3);  
INSERT INTO TB_LIBROS (titulo,autor,ISBN,fk_biblioteca) VALUES  
("Titulo8","Autor8",7190307862,3);  
INSERT INTO TB_LIBROS (titulo,autor,ISBN,fk_biblioteca) VALUES  
("Titulo9","Autor9",3240305747,3);
```