

SQL_LITE

Crear un documento que recoja los siguientes apartados:

- 1- Instalación del servidor SQLite. Incluye las pantallas del proceso

```
C:\Users\Claudia Borrachero\Desktop\STUDY\1º DAW\BASE DE DATOS\CASA\sqlite\sqlite-tools-win32-x86-3310100>sqlite3.exe
SQLite version 3.31.1 2020-01-27 19:55:54
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite>
```

- 2- Instalación de un cliente SQLite. Puede ser tanto gráfico como por consola. Incluye las pantallas del proceso.
- 3- Equivalente SQLite a los siguientes comandos SQL:
 - show databases; **.databases**
 - show tables; **.tables**
 - use; **.open**
 - create database; **sqlite3 test.db**
 - drop database; **There is no DROP DATABASE command in SQLite. If you want to drop a database, all you have to do is to delete the database file.**
- 4- Comparar los tipos de datos existentes en MySQL con los tipos de datos existentes en SQLite
 - a) **SQLite** admite estos tipos de datos: Blob, Integer, Null, Text, Real.

TEXT: variable de tipo texto que se almacena en formato UTF, aquí tenemos agrupados los diferentes tipos de datos **SQL** para las variables de texto: **CHARACTER(20)**, **VARCHAR(255)**, **VARYING CHARACTER(255)**, **NCHAR(55)**, **NATIVE CHARACTER(70)**, **NVARCHAR(100)**, **TEXT** y **CLOB**.

NUMERIC: **NUMERIC**, **DECIMAL(10,5)**, **BOOLEAN**, **DATE** y **DATETIME**

INTEGER: es un entero con signo que se almacena con una longitud que va en función del tipo de dato definido: **INTEGER**, **TINYINT**, **SMALLINT**, **MEDIUMINT**, **BIGINT**, **UNSIGNED BIG INT**, **INT2** y **INT8**

REAL: es un dato de tipo float, sus diferentes versiones variarán en la precisión: **REAL, DOUBLE, DOUBLE PRECISION y FLOAT**

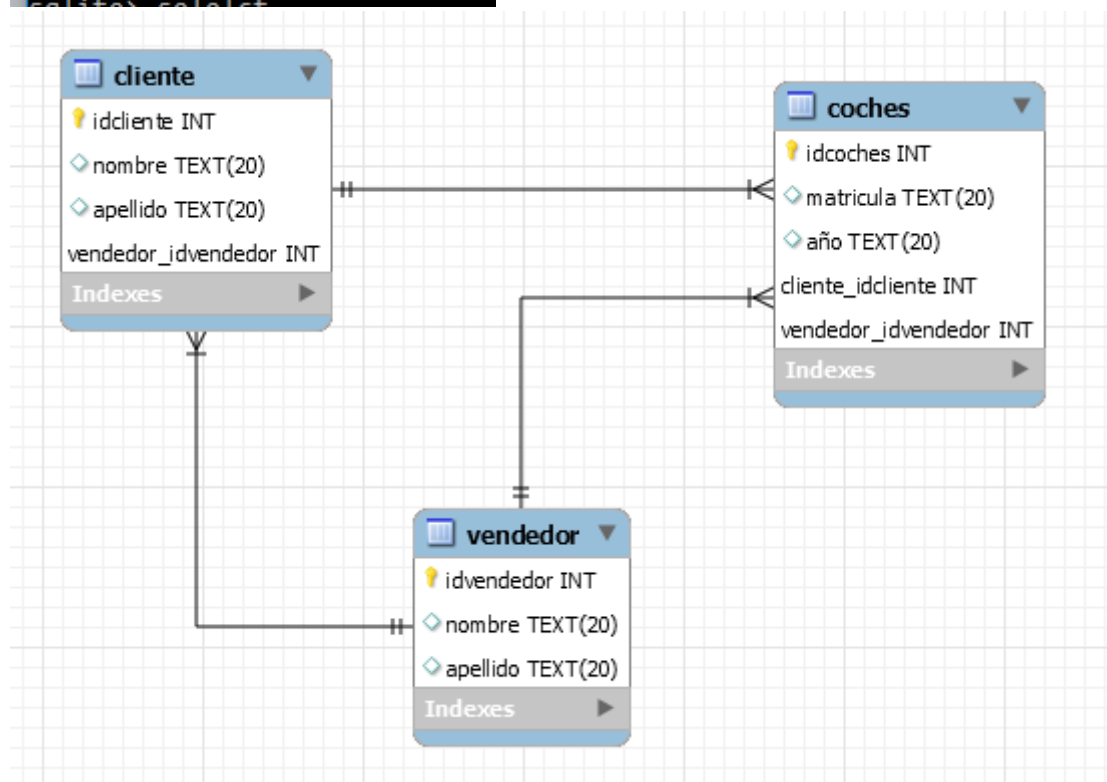
BLOB: los datos se almacenan en el mismo formato en que se introducen.

- b) **MySQL** admite los tipos de datos mencionados a continuación: Tinyint, Smallint, Mediumint, Int, Bigint, Double, Float, Real, Decimal, Double precision, Numeric, Timestamp, Date, Datetime, Char, Varchar, Year, Tinytext, Tinyblob, Blob, Text, MediumBlob, MediumText, Enum, Set , Longblob, Longtext.

- 5- Crear una base de datos que tenga al menos 3 tablas (deben estar todas relacionadas) y que esté rellena con al menos 5 filas (las tres tablas). Hacer una copia de seguridad de dicha base de datos y adjuntarla al documento

```
C:\Users\Claudia Borrachero\Desktop\STUDY\1º DAW\BASE DE DATOS\CASA\sqlite>sqlite3 test.db
SQLite version 3.31.1 2020-01-27 19:55:54
Enter ".help" for usage hints.
sqlite>
```

```
filename: test.db
sqlite> .tables
cliente  coches  vendedor
sqlite> .schema
```



6- Sobre las base de datos anterior realizar los siguientes tipos de consultas:

- Un select sobre una única tabla con todos los campos

```
sqlite> select * from coches;  
1|1234ABC|2015|1|1  
2|1234DEF|2016|2|2  
3|1234GHI|2017|3|3  
4|1234JKL|2018|4|4  
5|1234MNN|2019|5|5  
sqlite>  
sqlite> select * from cliente;  
1|Ana|Álvarez|1  
2|Beatriz|Bob|2  
3|Claudia|Borrachero|3  
5|Daniela|Dani|4  
5|Estefanía|Esteban|5  
sqlite> select * from vendedor;  
1|Abel|Abelardo  
2|Bob|Bobby  
3|Carlos|Charles  
4|Daniel|Dani  
5|Esteban|Estebitan  
sqlite>
```

- Un select sobre una única tabla que muestre sólo un campo de la tabla

```
sqlite> select nombre from vendedor;  
Abel  
Bob  
Carlos  
Daniel  
Esteban  
sqlite>
```

- Un select sobre una única tabla con una condición.

```
sqlite> select idcliente from cliente where idcliente>2;  
3  
5  
5  
sqlite>
```

- Un select que utilice un Join. Comparar dichas consultas con sus equivalentes en MySQL.

```
sqlite> select v.nombre from vendedor v inner join cliente c on v.idvendedor=c.vendedor_idvendedor;  
Abel  
Bob  
Carlos  
Daniel  
Esteban  
sqlite>
```

En MySQL se realizan las mismas consultas.

BASE DE DATOS

```
CREATE TABLE vendedor (  
  `idvendedor` INTEGER NOT NULL,  
  `nombre` TEXT NOT NULL,  
  `apellido` TEXT NOT NULL,  
  PRIMARY KEY (`idvendedor`));
```

```
CREATE TABLE cliente (  
  `idcliente` INTEGER NOT NULL,  
  `nombre` TEXT NOT NULL,  
  `apellido` TEXT NOT NULL,  
  `vendedor_idvendedor` INTEGER NOT NULL,  
  PRIMARY KEY (`idcliente`, `vendedor_idvendedor`),  
  CONSTRAINT `fk_cliente_vendedor1`  
    FOREIGN KEY (`vendedor_idvendedor`)  
    REFERENCES `vendedor` (`idvendedor`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION);
```

```
CREATE TABLE coches (  
  `idcoches` INTEGER NOT NULL,  
  `matricula` TEXT NOT NULL,  
  `año` TEXT NOT NULL,  
  `cliente_idcliente` INTEGER NOT NULL,  
  `vendedor_idvendedor` INTEGER NOT NULL,  
  PRIMARY KEY (`idcoches`, `cliente_idcliente`, `vendedor_idvendedor`),  
  CONSTRAINT `fk_coches_cliente`  
    FOREIGN KEY (`cliente_idcliente`)  
    REFERENCES `cliente` (`idcliente`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_coches_vendedor1`  
    FOREIGN KEY (`vendedor_idvendedor`)  
    REFERENCES `vendedor` (`idvendedor`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION);
```

```
INSERT INTO `vendedor` (`idvendedor`, `nombre`, `apellido`) VALUES (1, 'Abel',  
'Abelardo');
```

```
INSERT INTO `vendedor` (`idvendedor`, `nombre`, `apellido`) VALUES (2, 'Bob',  
'Bobby');
```

```
INSERT INTO `vendedor` (`idvendedor`, `nombre`, `apellido`) VALUES (3, 'Carlos', 'Charles');
```

```
INSERT INTO `vendedor` (`idvendedor`, `nombre`, `apellido`) VALUES (4, 'Daniel', 'Dani');
```

```
INSERT INTO `vendedor` (`idvendedor`, `nombre`, `apellido`) VALUES (5, 'Esteban', 'Estebitan');
```

```
INSERT INTO `cliente` (`idcliente`, `nombre`, `apellido`, `vendedor_idvendedor`) VALUES (1, 'Ana', 'Álvarez', 1);
```

```
INSERT INTO `cliente` (`idcliente`, `nombre`, `apellido`, `vendedor_idvendedor`) VALUES (2, 'Beatriz', 'Bob', 2);
```

```
INSERT INTO `cliente` (`idcliente`, `nombre`, `apellido`, `vendedor_idvendedor`) VALUES (3, 'Claudia', 'Borrachero', 3);
```

```
INSERT INTO `cliente` (`idcliente`, `nombre`, `apellido`, `vendedor_idvendedor`) VALUES (5, 'Daniela', 'Dani', 4);
```

```
INSERT INTO `cliente` (`idcliente`, `nombre`, `apellido`, `vendedor_idvendedor`) VALUES (5, 'Estefanía', 'Esteban', 5);
```

```
INSERT INTO `coches` (`idcoches`, `matricula`, `año`, `cliente_idcliente`, `vendedor_idvendedor`) VALUES (1, '1234ABC', '2015', 1, 1);
```

```
INSERT INTO `coches` (`idcoches`, `matricula`, `año`, `cliente_idcliente`, `vendedor_idvendedor`) VALUES (2, '1234DEF', '2016', 2, 2);
```

```
INSERT INTO `coches` (`idcoches`, `matricula`, `año`, `cliente_idcliente`, `vendedor_idvendedor`) VALUES (3, '1234GHI', '2017', 3, 3);
```

```
INSERT INTO `coches` (`idcoches`, `matricula`, `año`, `cliente_idcliente`, `vendedor_idvendedor`) VALUES (4, '1234JKL', '2018', 4, 4);
```

```
INSERT INTO `coches` (`idcoches`, `matricula`, `año`, `cliente_idcliente`, `vendedor_idvendedor`) VALUES (5, '1234MNÑ', '2019', 5, 5);
```