

SQL

Esquema.

Para este ejercicio trabajaremos con el esquema dado por las siguientes tablas:

- Cervezas(nombre, tipo, grados, ciudad_origen);
- EnStock(nombre_cerveza, ciudad, cantidad, precio_unitario);

1. Creando una base de datos y poniendo datos

Tu primera prueba será crear una base de datos de acuerdo a el esquema de arriba. La sintaxis para crear tablas en SQL es la siguiente (después de cada sentencia SQL debes poner un punto y coma, ‘;’):

```
CREATE TABLE <Nombre> (<atributo1> tipo, ... , <atributoN> tipo)
```

Por ejemplo,

```
CREATE TABLE EnStock (nombre_cerveza varchar(20), ciudad varchar(20),  
                        cantidad integer, precio_unitario float)
```

También es posible especificar valores por defecto:

```
CREATE TABLE EnStock (nombre_cerveza varchar(20), ciudad varchar(20) ,  
                        cantidad integer, precio_unitario float DEFAULT 2000)
```

En este caso el precio por defecto al crear una tupla es 2000.

1. Crea la otra tabla.

Acá va una lista de algunos tipos de datos disponibles

- CARACTERES (STRINGS)
 - char(20) – largo fijo
 - varchar(20) – largo variable (más usado en la practica)
- NÚMEROS
 - int, smallint, float, real – etc.

- TIEMPOS Y FECHAS

- DATE – YYYY-MM-DD
- TIME – hora formato 24hrs.
- TIMESTAMP – fecha + hora

- Hay muchos otros, ver el estándar SQL

2. Use la siguiente instrucción para ver el contenido de las tablas que acabas de crear. Están vacías, obviamente.

```
SELECT * FROM Cervezas
```

```
SELECT * FROM EnStock
```

2. Lamentablemente, ¡faltaron las llaves! Hay que borrar las tabas. Para borrar una tabla se usa

```
DROP TABLE EnStock
```

2. Ahora crea las tablas con llaves (recuerde también el valor por defecto en EnStock. Suponiendo que *atributoI* es llave, la sintaxis es

```
CREATE TABLE <Nombre> (<atributo1> tipo, . . . ,  
                        <atributoI> tipo PRIMARY KEY, . . . , <atributoN> tipo)
```

O bien, si la llave está compuesta por más de un atributo (en este caso *atributoI* y *atributoJ*):

```
CREATE TABLE <Nombre> (<atributo1> tipo, . . . ,  
                        <atributoN> tipo, PRIMARY KEY(atributoI,atributoJ))
```

1.1. INSERT y DELETE

La forma básica de insertar:

```
INSERT INTO <Nombre> VALUES (<valoratributo1> , . . . , <valoratributoN>)
```

Por ejemplo (los strings pueden ir entre comillas simples o dobles),

```
INSERT INTO Cervezas VALUES ('Cristal' , 'lager', 4.3, 'Santiago')
```

Pero también puedes decidir llenar solo algunos valores:

```
INSERT INTO EnStock (nombre_cerveza, ciudad) VALUES ('Cristal' , 'Temuco')
```

1. Ejecute la instrucción anterior. ¿Qué queda en la tabla **EnStock**? Notar que el precio toma el valor por defecto, pero la cantidad queda indefinida y toma un valor *nulo*. ¡Vamos a discutir esto con detalle en el curso!
2. Vamos a ver qué pasa cuando intentamos agregar dos veces la misma cerveza. Ejecute el siguiente comando y vea lo que pasa. ¿Qué ocurrió?

```
INSERT INTO Cerveza VALUES ('Cristal' , 'lager', 3, 'Colina')
```

3. En cambio, intente con

```
INSERT INTO Cerveza VALUES ('Cristal light', 'lager', 3, 'Colina')
```

4. Finalmente decidimos eliminar a Cristal (hay que ser francos, no es la mejor Cerveza). Para eso borramos todas las tuplas cuyo nombre contenga Cristal:

```
DELETE FROM Cerveza  
WHERE nombre LIKE '%Cristal%'
```

Ojo que se borraron ambas cervezas. En este caso el **LIKE** funciona como una igualdad, pero podemos agregar:

- El comando **'%'** que significa *cualquier secuencia de caracteres*
- El comando **'_'** que significa *cualquier carácter* (pero solo uno, como **'ni_'**)

1.2. Otros Comandos

Hay una serie de otros comandos que puedes usar. Acá van algunos:
Eliminar un atributo:

```
ALTER TABLE Cervezas DROP COLUMN grados
```

El comando de arriba arroja un error en SQLite3. Esto es por que este sistema no soporta comandos para remover atributos, pero casi todos los motores sí lo soportan. Y agregar uno:

```
ALTER TABLE Cervezas ADD COLUMN grados
```

Modificar datos:

```
UPDATE Cervezas SET grados = 4 WHERE nombre = 'Cristal'
```