

Curso 1º SMR

Módulo: AO

Actividades 3 Bases de datos: Creamos Tablas mediante SQL

¿Qué pretendemos conseguir?

Ser capaces de crear Tablas en la base de datos mediante SQL.
Ser capaces de modificar Tablas mediante SQL.
Ser capaces de crear Foreign Keys mediante SQL.

¿Qué necesitamos?

Openoffice Base.
Haber realizado las actividades 1 y 2 de Bases de Datos.

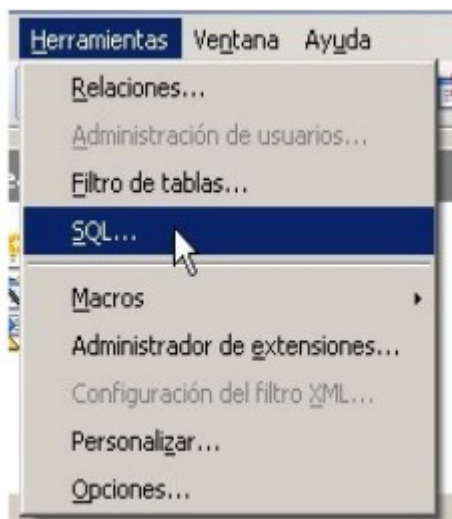
Creación de Tablas mediante SQL en BASE

Antes de nada crearemos la base de datos tal y como hemos aprendido en las actividades anteriores.

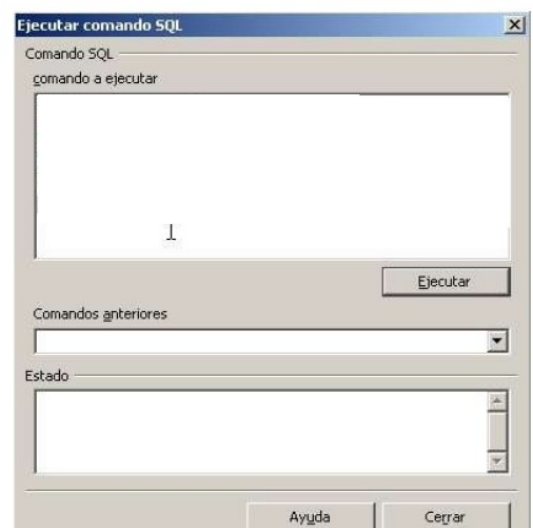
Actividad 1: Busca información sobre el el comando CREATE DATABASE de MySQL.

A la base de datos la llamaremos EMPRESA.

En el menú principal de BASE pincharemos en Herramientas/SQL



Nos aparecerá la siguiente ventana:



En el apartado "comando a ejecutar" es donde vamos a introducir las consultas SQL.

Actividad 2: Introduce el siguiente código

```
CREATE TABLE Empleados ( Id_Empleado INTEGER PRIMARY  
KEY, NOMBRE VARCHAR(40), TELEFONO VARCHAR(10) )
```

Después de esto, si no se ha producido ningún error sintáctico, aparecerá un mensaje en Estado como el siguiente:

"1.- comando ejecutado correctamente".

Este comando ha creado la tabla pero no la vemos. Para solucionarlo: Ver ---> Actualizar Tablas desde el menú principal de BASE.

Actividad 3: Crea las siguientes tablas:

TABLA NUM_VENTA

Campos:

```
N_VENTA Entero  
N_EMPLEADO Entero  
Fecha Fecha (DATE)  
N_CLIENTE Entero
```

Como primary key: N_VENTA

TABLA PRODUCTOS

Campos:

```
N_PRODUCTO Entero  
DESCRIPCION VARCHAR (20)  
PRECIO_UNIDAD DECIMAL (4,2)
```

Como primary key: N_PRODUCTO

TABLA ALUMNOS

Campos:

```
Id_Alumno Entero  
Nombre VARCHAR(40)  
Apellido 1 VARCHAR(40)  
Apellido 2 VARCHAR(40)  
SEXO CHAR(1)  
Fecha_Nac DATE
```

Como Primary Key: Id_Alumno

Modificación de tablas ya creadas

Para modificar tablas ya creadas utilizaremos la sentencia ALTER TABLE, con ella podremos añadir, modificar y borrar campos en las tablas previamente creadas.

Añadir nuevos campos:

```
ALTER TABLE <NOMBRE TABLA> ADD COLUMN <campo> <tipo campo>
```

Ejemplo:

```
ALTER TABLE NUM_VENTA ADD COLUMN MASTERCARD VARCHAR(20)
```

Actividad 4: En la tabla ALUMNOS del ejercicio 3 añade el campo Repetidor del tipo CHAR(1)

Actividad 5: ¿Es necesario actualizar tablas para que el cambio sea visible?

Eliminar campos:

```
ALTER TABLE <NOMBRE TABLA> DROP COLUMN <campo>
```

Obviamente, el campo a eliminar debe existir en la tabla.

Ejemplo:

```
ALTER TABLE NUM_VENTA DROP COLUMN MASTERCARD
```

Actividad 6: Elimina el campo Repetidor de la tabla ALUMNOS.

Actividad 7: Elimina el campo N_VENTA de la tabla NUM_VENTA ¿Es posible? ¿Puedes introducir datos en la tabla modificada? ¿Por qué?

Crear primary key's:

```
ALTER TABLE <NOMBRE TABLA> ADD PRIMARY KEY ( Lista de campos)
```

Siempre que hagamos una modificación de este tipo es recomendable "Actualizar Tablas".

Ejemplo:

```
ALTER TABLE NUM_VENTA ADD PRIMARY KEY (Fecha).
```

Con esta misma sentencia podemos crear tablas cuyas primary key estén compuestas por más de un campo. Para ello, primero crearemos la tabla con la sentencia CREATE

TABLE (aquí no indicaremos ningún primary) y, a continuación añadiremos la sentencia ALTER TABLE.

Para enlazar dos sentencias SQL en una misma acción terminaremos las sentencias con el carácter ; (**punto y coma**).

Ejemplo:

```
CREATE TABLE TABLA1 (CAMPO1 CHAR(1), CAMPO2 CHAR(2));  
ALTER TABLE TABLA1 ADD PRIMARY KEY (CAMPO1, CAMPO2);
```

Actividad 8: Crea la tabla TABLA1 con los campos, todos de tipo VARCHAR(15): CAMPO1, CAMPO2, CAMPO3, CAMPO4, CAMPO5. Crea la primary key con: CAMPO1, CAMPO2 y CAMPO3.

Actividad 9: Elimina la primary key de TABLA1.

Actividad 10: Crea una primary key con el CAMPO1 en la tabla TABLA1

Modificar el tipo de campo:

En ocasiones nos será necesario cambiar el tipo de campo: de CHAR a DATE, de CHAR a INTEGER, aumentar el tamaño de un VARCHAR...

Para ello utilizaremos la siguiente sentencia:

```
ALTER TABLE <TABLA> ALTER COLUMN <CAMPO> <NUEVO TIPO  
DE CAMPO>
```

Ejemplo: queremos cambiar el CAMPO4 de TABLA1 (actualmente VARCHAR(15) a INTEGER:

```
ALTER TABLE TABLA1 ALTER COLUMN CAMPO4 INTEGER
```

Actividad 11: Realiza el ejemplo anterior.

Actividad 12: Enuncia algún caso en el que no sea posible realizar ALTER COLUMN e indica el por qué.

Actividad 13: Haz que el CAMPO1, actualmente el primary key de TABLA1 sea INTEGER.

Campo INTEGER generado de manera incremental:

Añadiremos lo siguiente:

```
INTEGER GENERATED BY DEFAULT AS IDENTITY(START WITH  
<número de comienzo>, INCREMENT BY <incremento, si  
es uno esta parte no es necesaria>)
```

Ejemplo:

```
CREATE TABLE TABLA5 (CAMPO1 INTEGER GENERATED BY DEFAULT  
AS IDENTITY (START WITH 0, INCREMENT BY 5))
```

Actividad 14: Crea la TABLA6 con CAMPO1 entero generado por defecto con comienzo en 10 e incremento 2, y CAMPO2, CAMPO3, CAMPO4 todos CHAR (1).

Actividad 15: ¿Quién es el primary key de TABLA6?

Actividad 16: Añade CAMPO5 de tipo VARCHAR (15)

Actividad 17: Elimina el CAMPO3

Actividad 18: Crea la TABLA7 con CAMPO1, CAMPO2, CAMPO3, CAMPO4, CAMPO5, CAMPO6 todos VARCHAR (10)

Actividad 19: Crea un primary key en TABLA7 con CAMPO1, CAMPO2 Y CAMPO5.

Actividad 20: Modifica el primary key de TABLA7 para que sean los CAMPO4, CAMPO5 Y CAMPO6 quienes la formen.

Eliminar TABLAS:

```
DROP TABLE <nombre tabla>
```

Ejemplo: para borrar la tabla "PEDIDOS":

```
DROP TABLE PEDIDOS;
```

Actividad 21: Elimina todas las tablas creadas en esta actividad.

Actividad 22: ¿Es necesario actualizar Tablas para que la Tabla desaparezca del documento?