

A blurred background image showing a person wearing glasses and a blue checkered shirt, looking at a computer screen. The screen displays multiple windows of code or database management software. A laptop keyboard is visible in the foreground.

Implementación y gestión de bases de datos

Construcción de una base de datos en un gestor de bases de datos MySQL

Construcción de una base de datos en un gestor de bases de datos MySQL

1. Creamos nuestra base de datos, en este caso la llamaremos facturación, la instrucción SQL es la siguiente:

"CREATE DATABASE *facturacion*".

Figura 1

Construcción de una base de datos

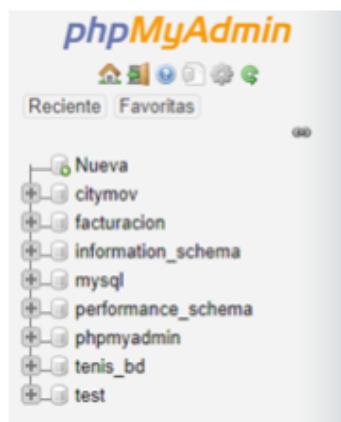


The screenshot shows a MySQL query editor window. At the top, it says "Ejecute la(s) consulta(s) SQL en el servidor '127.0.0.1':". Below that, there is a text area containing the SQL command: "1 CREATE DATABASE Facturacion". At the bottom of the window, there are three buttons: "Limpiar", "Formato", and "Obtener consulta almacenada automáticamente".

Ejecutamos la sentencia, y creamos nuestra base de datos, como se ve en la ilustración.

Figura 2

Motor phpMyAdmin de MySQL



2. Para eliminar una base de datos la instrucción SQL es la siguiente:

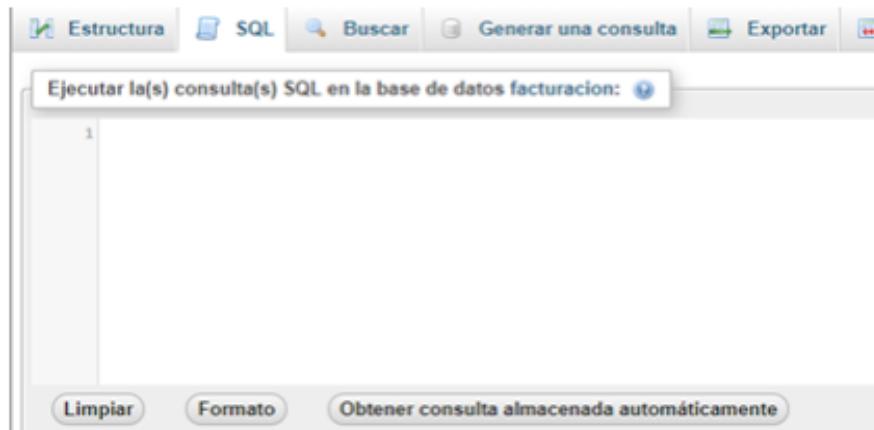
"DROP TABLE facturacion"

Se escribe los comandos seguido del nombre de la base de datos a eliminar, tener en cuenta que, al realizar la eliminación de la base de datos se elimina todo por completo tablas y sus registros.

3. Procedemos crear las tablas (entidades); para la creación de las tablas hay que estar ubicado en la base de datos donde las vamos a crear, en nuestro caso "facturacion", empezamos construyendo las tablas que no tienen llaves foráneas, en nuestro caso las entidades "Vendedor" y "Almacén", y por último seguimos la línea de construcción dependiendo de las relaciones entre las entidades. (seguirá la entidad "Cliente" y por último la entidad "Ventas").

Figura 3

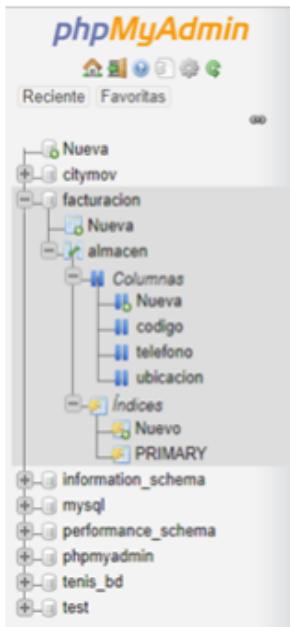
Selección de base de datos facturación



La instrucción SQL para la creación de la tabla Almacén es la siguiente:

```
"CREATE TABLE almacen ( codigo int,
                      ubicacion varchar(100),
                      telefono varchar(100),
                      CONSTRAINT PRIMARY KEY (codigo))"
```

Ejecutamos la sentencia y se crea la tabla almacén (ver ilustración), toda tabla de una base de datos lleva una llave primaria que es el identificador con respecto a las demás tablas.

Figura 4*Estructura de base de datos facturación*

El comando para eliminar una tabla de la base de datos es:

"DROP TABLE nombre_de_la_tabla"

Si necesitamos agregar una nueva columna a la tabla la instrucción es la siguiente:

"ALTER TABLE nombre_tabla ADD COLUMN nombre_columna tipo_dato"

Por ejemplo, agregaremos la columna "email" a la tabla "almacen"

"ALTER TABLE almacen ADD COLUMN email varchar(100)"

En nuestro motor phpMyAdmin podemos ver la estructura de una tabla de la base de datos:

Figura 5*Columnas de la base de datos "facturación"*

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
1	codigo	int(11)			No	Ninguna			
2	ubicacion	varchar(100)	latin1_swedish_ci		Sí	Ninguna			
3	telefono	varchar(100)	latin1_swedish_ci		Sí	Ninguna			
4	email	varchar(100)	latin1_swedish_ci		Sí	Ninguna			

Cuando la tabla a crear tiene uno o más llaves foráneas la instrucción SQL es la siguiente:

```
"CREATE TABLE cliente (cedula int,
                      nombre varchar(100),
                      telefono varchar(100),
                      codigo_almacen int,
                      CONSTRAINT pk_cliente PRIMARY KEY (cedula),
                      CONSTRAINT fk_almacen_cliente FOREIGN KEY(codigo_almacen)
                      REFERENCES almacen(codigo))"
```

Figura 6

Script para la construcción de la tabla "Cliente"

Ejecutar la(s) consulta(s) SQL en la base de datos facturación:

```
1 CREATE TABLE cliente ( cedula int,
2                       nombre varchar(100),
3                       telefono varchar(100),
4                       codigo_almacen int,
5                       CONSTRAINT pk_cliente PRIMARY KEY (cedula),
6                       CONSTRAINT fk_almacen_cliente FOREIGN KEY (codigo_almacen) REFERENCES almacen(codigo))|
```

Limpiar Formato Obtener consulta almacenada automáticamente

En la creación de una tabla se crean llaves tanto primarias como foráneas.

- Llave primaria, en nuestra tabla, creada anteriormente, la línea de código que se especifica para la llave primaria es:

"CONSTRAINT pk_cliente PRIMARY KEY (cedula)"

Donde, "pk_cliente" es el nombre que le queremos poner a la llave, "(cedula)" es el campo de nuestra tabla que será la llave primaria.

- Llave foránea, en la tabla "cliente" creada anteriormente la instrucción fue la siguiente:

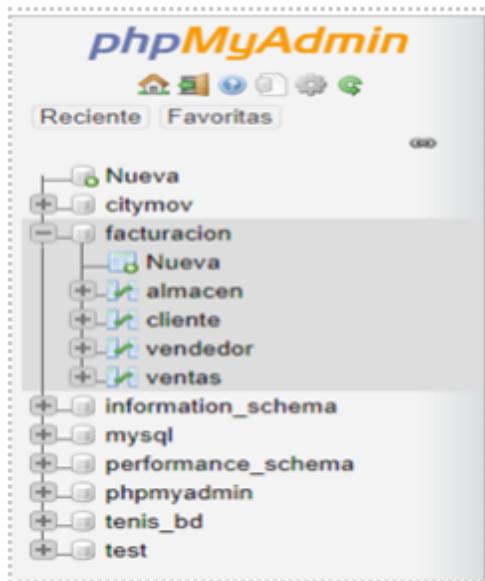
"CONSTRAINT fk_almacen_cliente FOREIGN KEY (codigo_almacen) REFERENCES almacen(codigo))"

Donde, "fk_almacen_cliente" es el nombre que le dimos a nuestra llave foránea, "codigo_almacen" es el campo (debe ser exactamente el mismo tipo de dato con la misma longitud de la tabla relación) de nuestra tabla que se relaciona con la llave primaria de la tabla relación (tabla almacen) , y por último "almacen (codigo)" que procede del comando REFERENCES que se define la relación colocando el nombre de la tabla y su campo de llave primaria como se observa en la línea de código referida anteriormente.

4. Por último, ya tenemos construida nuestra base de datos "facturacion" con todas las tablas correspondiente como esta en el diagrama entidad/relación y sus respectivas llaves.

Figura 7

Estructura de la BD "facturación" y sus tablas



Estructura de la BD "facturación":

Figura 8

Tablas de la BD "facturación"

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar
almacen	Examinar Estructura Buscar Insertar Vaciar Eliminar		InnoDB	latin1_swedish_ci	16 KB	-
cliente	Examinar Estructura Buscar Insertar Vaciar Eliminar		InnoDB	latin1_swedish_ci	32 KB	-
vendedor	Examinar Estructura Buscar Insertar Vaciar Eliminar		InnoDB	latin1_swedish_ci	16 KB	-
ventas	Examinar Estructura Buscar Insertar Vaciar Eliminar		InnoDB	latin1_swedish_ci	48 KB	-
4 tablas Número de filas					112 KB	0 B

Detalles a tener en cuenta al momento de construir bases de datos en los gestores de bases de datos:

- NOT NULL (Para que no se permita un valor nulo o vacío en el campo)
- PRIMARY KEY (identificador de la tabla)
- UNIQUE (Para obligar a que no se repitan valores en esa columna)
- CHECK (Para verificar una condición dada)