

## □ Insert:

Servidor: 127.0.0.1 > Base de datos: bd\_pedidos > Tabla: clientes

Examinar Estructura SQL Buscar Insertar Exportar Importar Privilegios Operaciones Seguimiento Disparadores

```
1 #1))INSERT INTO tabla (lista de campos) VALUES (lista de valores). Esta se utiliza solo para rellenar campos específicos de una tabla.
2
3 #Ejemplo:
4
5 INSERT INTO clientes ('cli_codigo', 'cli_empresa', 'cli_direccion', 'cli_poblacion', 'cli_telefono', 'cli_responsable', 'cli_historial')
6 VALUES ('Aeiou', 'empresa', 'C/X', 'Madrid', '45', 'pepe', '');
7
8
9 #2))INSERT INTO tabla VALUES (lista de valores ordenados por aparicion y separados por coma). Esta se utiliza para rellenar todos los campos de una tabla.
10 #Asignariamos valor null a los campos que no nos interese llenar
11
12 #Ejemplo:
13
14 INSERT INTO clientes
15 VALUES ('BBB', 'ACME', 'C/YY', 'TOLEDO', '49', 'JOSE', null);
16
17 #3))INSERT INTO tabla VALUES
18 #           (lista de valores ordenados por aparicion y separados por coma).
19 #           (lista de valores ordenados por aparicion y separados por coma).
20 #           (lista de valores ordenados por aparicion y separados por coma).
21 #Este nos sirve para rellenar todos los datos de una tabla varias veces.
22
23
24 #Ejemplo:
25
26 INSERT INTO clientes VALUES
27     ('ccc', 'AE', 'C/Yxp', 'TOLETUM', '79', 'jorge', null),
28     ('dd', 'ACer', 'C/u', 'TOLEDO', '55', 'patri', null),
29     ('Bee', 'renolt', 'Yey', 'madrid', '38', 'JOSE', null);
30
31 #4)) INSERT INTO clientes SET      -----> FORMATO PROPIO DE MYSQL, NO SE SUELTI USAR
32 #           campo = valor
33
34 #Ejemplo
35 INSERT INTO clientes SET
36     `cli_codigo` = 'rrrrr',
37     `cli_empresa` = 'vvvvv',
38     `cli_direccion` = 'C/eros',
39     `cli_poblacion` = 'huesca',
40     `cli_telefono` = '444444444',
41     `cli_responsable` = 'pedro'
```

Consola

Buscar

## □ Update:

```
1 #ACTUALIZAR REGISTROS (UPDATE)
2 #UPDATE tabla SET
3 #    campo = valor1,
4 #    campo = valor2,
5 #    campo = valor3,
6 #    campo = valor4,
7 #WHERE condicion
8 #OJO!!!!!! -----> SI NO PONEMOS EL WHERE, ACTUALIZA TODOS LOS REGISTROS DE LA TABLA. PONER SIEMPRE EL WHERE!!
9
10 #EJEMPLO:
11
12 UPDATE clientes SET
13     cli_historial = 'UNICO'
14 WHERE cli_codigo = 'AAAA' | ]
```

## Delete:

```
1 #ELIMINAR REGISTROS (DELETE)
2 #DELETE FROM tabla
3 #WHERE condicion
4 #OJO!!!!!! -----> SI NO PONEMOS EL WHERE, ELIMINA TODOS LOS REGISTROS DE LA TABLA. PONER SIEMPRE EL WHERE
5
6 #EJEMPLO:
7
8 DELETE FROM clientes
9 WHERE cli_historial = 'UNICO'|
```

## Select:

The screenshot shows the phpMyAdmin interface for a MySQL database named 'bd\_pedidos'. The current table is 'clientes'. The top navigation bar includes links for 'Examinar', 'Estructura', 'SQL', 'Buscar', 'Insertar', 'Exportar', 'Importar', 'Privilegios', 'Operaciones', 'Seguimiento', and 'Disparadores'. Below the navigation is a search bar and a toolbar with buttons for 'SELECT', 'INSERT', 'UPDATE', 'DELETE', 'Limpiar', 'Formato', and 'Obtener consulta almacenada automáticamente'. A checkbox for 'Enlazar parámetros' is also present. The main area contains a large block of SQL code:

```
1 #SELECT lista de campos FROM lista de tablas
2 #Ejemplo:
3
4 SELECT cli_codigo, cli_poblacion FROM clientes;
5
6 #Comandos que se utilizan comunmente:
7 #DISTINCT: Nos da los distintos datos de una tabla. Por ejemplo, si buscamos poblaciones y hay algunas que se repiten, este comando solo te saca la población una vez.
8 #Ejemplo:
9 SELECT DISTINCT cli_poblacion FROM clientes;
10 #LIKE : Que contenga (aa% o %aa), que sea igual a ( LIKE 'aaa')
11 #ORDER BY: Ordenar por. Se puede ordenar según lista de campos, o por orden ascendente (ASC) o descendente (DESC)
12 #Ejemplo:
13 SELECT cli_codigo, cli_poblacion FROM clientes
14 ORDER BY cli_empresa DESC;
15 #AS: Se utiliza para dar nombre a las listas de campos para dar otro formato
16 #LIMIT: limita los datos mostrados en la búsqueda: En este caso podemos limitar el número de datos (ej.1) o mostrar datos entre varias posiciones (ej.2)
17 #Ejemplo:
18 SELECT cli_codigo, cli_poblacion FROM clientes
19 ORDER BY cli_empresa DESC
20 #ej.1) LIMIT 3
21 #ej.2) LIMIT 2,4
22
```

# Consulta de varias tablas:

```
Examinar Estructura SQL Buscar Insertar Exportar Importar Privilegios Operaciones Seguimiento D
1 #Sacar informacion de varias tablas de una misma base de datos
2
3 # 1) Seleccionar los datos que nos piden en la consulta con el SELECT. Si quisiese sacar un unico valor de cada y que no se repitan, pondriamos DISTINCT seguido del SELECT.
4 #Tambien podemos añadir funciones de agregado, que pueden ser las siguientes:
5 #    1.1) MEDIA (AVG()). Ej: SELECT AVG(producto. prod_precio) AS 'media'
6 #    1.1) SUMA (SUM()). Ej: SELECT SUM(producto. prod_precio) AS 'suma'
7 #    1.1) MINIMO (MIN()). Ej: SELECT MIN(producto. prod_precio) AS 'minimo'
8 #    1.1) MAXIMO (MAX()). Ej: SELECT MAX(producto. prod_precio) AS 'maximo'
9 #    1.1) CUENTA (COUNT()),es decir,que me diga cuantos registros hay de esa búsqueda. Ej: SELECT COUNT(producto. prod_precio) AS 'cuenta'
10 #
11 #SELECT DISTINCT c.cli_empresa, c.cli_poblacion, p.ped_numero,p.ped_fecha
12 SELECT c.cli_empresa, c.cli_poblacion, p.ped_numero,p.ped_fecha
13
14
15 # 2) Seleccionar de que tablas tengo que sacar esa informacion con el FROM
16 FROM clientes as c, pedidos as p
17
18 # 3) Elegir las condiciones de búsqueda. Siempre que saquemos info de más de una tabla, la relación del WHERE será tablas -1. Es decir, si la info sale de dos tablas, el WHERE tendrá solo una condición de igual. Si queremos añadir información, se añade con un AND o un OR. Las condiciones OR se ponen entre paréntesis
19 WHERE
20   c.cli_codigo = p.ped_cli_codigo AND
21   (c.cli_poblacion = 'Madrid' OR c.cli_poblacion = 'Bilbao') AND
22   c.cli_empresa LIKE 'a%'
23
24 # 4) Agrupar según como queremos que nos lo muestre GROUP BY. Cuando ponemos esto, no es necesario poner un ORDER BY después, ya que esta función de group contiene la ordenación
25 GROUP BY c.cli_empresa, p.ped_numero
26
27 # 5) Condiciones aplicadas a las funciones de agrregado HAVING
28 # Este HAVING puede llevar detrás cualquiera de las funciones de agregado descritas en el punto 1 (MAX, MIN, AVG, SUM, COUNT)
29 HAVING COUNT(prod_codigo) > 5
30
31 # 6) Ordenar de arriba a abajo según como queremos que nos lo muestre ORDER BY. Aquí podriamos ordenar en orden ascendente(ASC) o descendente(DS)
32 ORDER BY c.cli_empresa
33
34 # 7) Filtrar resultados de la consulta anterior LIMIT. 2 casos: LIMIT 3 (Saca solo los 3 primeros resultados) o LIMIT 2,3 (Desde el segundo de la lista, me saca 3 resultados)
35 LIMIT 2,3
```