

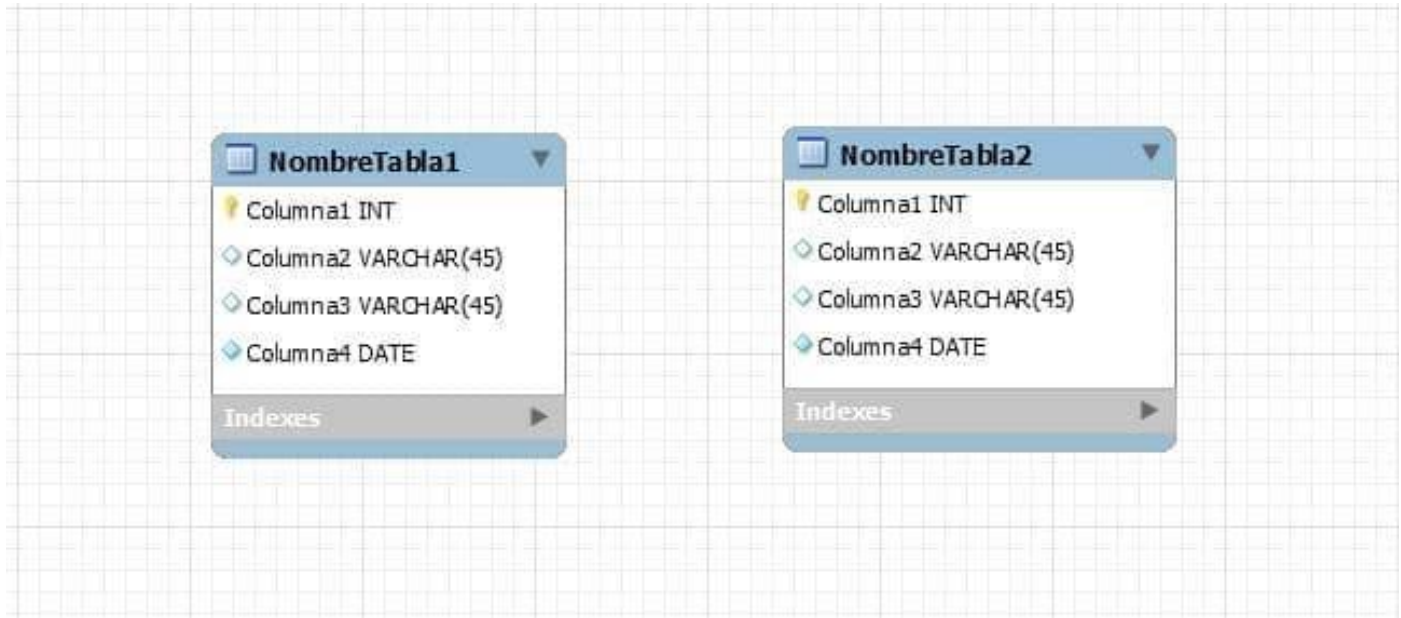
SQL ANSI

Lenguaje de Definición de Datos DML



— Sentencias
INSERT – UPDATE – DELETE

Insertar, actualizar y borrar datos en MySQL



Para agregar datos a las tablas se utiliza la instrucción **Insert into**, este comando es uno de los más usados en los diferentes gestores de Base de Datos. Para insertar los registros se puede hacer de uno en uno, o agregar varios registros a través de una misma instrucción insert into.

Sintaxis de Insert into en MySQL

La sintaxis para usar la instrucción insert into en una tabla de MySQL es la siguiente:

```
INSERT INTO "NombreTabla" ("PrimeraColumna", "SegundaColumna", etc)  
VALUES ("Dato1", "Dato2", etc);
```

Sintaxis de las Sentencias **INSERT**

NombreTabla: Es el nombre de la tabla en la que insertaremos registros.

PrimeraColumna, SegundaColumna,...: Son las columnas de la tabla en la que vamos a insertar registros.

"Dato1", "Dato2" ,...: Son los valores que vamos a guardar en cada columna especificada.

Es importante mencionar que la sintaxis anterior se puede reducir en los casos que vamos a agregar datos a todas las columnas, ya que podríamos plantearlo de la forma siguiente:

```
INSERT INTO "NombreTabla" VALUES ("Dato1", "Dato2", etc);
```

Cuando aplicamos esta sintaxis, debemos respetar la estructura de la tabla y además después del comando values enviar todos los datos para cada columna en el orden correcto, ya que dicha sintaxis indica que vamos a insertar registros a todas las columnas, por lo tanto debemos enviar los datos exactamente como los hemos especificado al momento de crearla.

Existe otra opción de insertar registros y es mencionando columnas específicas, para este caso debemos tomar en cuenta que las columnas que omitimos puedan quedar nulas, es decir que le hayamos agregado la propiedad not null.

Insertar datos en una tabla Mysql

Ya habiendo conocido la sintaxis ahora nos interesa poder agregar los registros, para ello lo haremos con la tabla Alumnos, vamos a tomar en cuenta que tiene los siguientes campos:

IdAlumno

Nombres

Apellidos

Edad

Direccion_Residencia

El código necesario para insertar un único registro en la tabla Alumnos es el siguiente:

MySQL

```
INSERT INTO Alumnos (IdAlumno, Nombres, Apellidos, Edad, Direccion_Residencia) VALUES  
( '0101', 'Franklin1', 'Garcia', '25', 'avenida 01');
```

Código Ansi SQL para insertar registros en Mysql

Si queremos agregar varios registros a través de un mismo insert, basta con agregar una coma en los valores que le enviamos en values, y especificar los datos a insertar.

MySQL

```
INSERT INTO Alumnos (IdAlumno, Nombres, Apellidos, Edad, Direccion_Residencia)
VALUES
('0102', 'Franklin1', 'Garcia', '25', 'avenida 01'),
('0103', 'Franklin2', 'Garcia', '25', 'avenida 02'),
('0104', 'Franklin3', 'Garcia', '25', 'avenida 03');
```


Inserción de Filas

- ✓ Insertar una nueva fila que contenga los valores de cada columna.
- ✓ Mostrar valores en el orden por defecto de las columnas de la tabla.
- ✓ Opcionalmente, mostrar la lista de columnas en la cláusula INSERT.
- ✓ Encerrar valores de caracteres

```
INSERT INTO departments(department_id,  
                        department_name, manager_id, location_id)  
VALUES (70, 'Public Relations', 100, 1700);
```

1 rows inserted

```
INSERT INTO  table [(column [, column...])]  
VALUES      (value [, value...]);
```

Actualizar datos en una tabla en Mysql

Sintaxis de Sentencias UPDATE

- ✓ Modificar los valores existentes en una tabla con la sentencia UPDATE
- ✓ Actualizar más de una fila cada vez (si es necesario).

```
UPDATE      table
SET         column = value [, column = value, ...]
[WHERE      condition];
```

En la sintaxis:

table es el nombre de la tabla.

column es el nombre de la columna de la tabla que se debe rellenar.

value es el valor o subconsulta correspondiente para la columna.

condition identifica las filas que se deben actualizar y se compone de nombres de columna, expresiones, constantes, subconsultas y operadores de comparación.

Para confirmar la operación de actualización, consulte la tabla para visualizar las filas actualizadas.

Actualización de filas en una tabla

- ✓ Si se especifica la cláusula `WHERE`, se modifican los valores de una fila o varias filas específicas:

```
UPDATE employees|  
SET    department_id = 50  
WHERE employee id = 113;
```

1 rows updated

- ✓ Si se omite la cláusula `WHERE`, se modifican los valores de todas las filas de la tabla:

```
UPDATE    copy_emp  
SET       department_id = 110;
```

22 rows updated

- ✓ Especificar `SET column_name= NULL` para actualizar un valor de columna a NULL.

Borrado de filas en una tabla en Mysql

Sentencia DELETE

Puede eliminar filas existentes de una tabla mediante la sentencia `DELETE`

```
DELETE [FROM] table  
[WHERE condition];
```

En la sintaxis:

table es el nombre de la tabla.

condition identifica las filas que se deben suprimir y se compone de nombres de columna, expresiones, constantes, subconsultas y operadores de comparación.

Nota: si no se suprime ninguna fila, se devuelve el mensaje "0 rows deleted" (en el separador Script Output de Workbench).

Sentencia DELETE

Supresión de filas de tablas

- ✓ Se suprimen filas concretas si se especifica la cláusula

```
DELETE FROM departments  
WHERE department_name = 'Finance';
```

1 rows deleted

- ✓ Se suprimen todas las filas de la tabla si omite la cláusula WHERE

```
DELETE FROM copy_emp;
```

22 rows deleted

Sentencia TRUNCATE

- ✓ Elimina todas las filas de una tabla, dejando la tabla vacía y la estructura de la misma intacta.
- ✓ Es una sentencia de lenguaje de definición de datos (DDL) en lugar de una sentencia DML; no se puede deshacer fácilmente.

Sintaxis:

```
TRUNCATE TABLE table_name;
```

✓ Ejemplo:

```
TRUNCATE TABLE copy_emp;|
```