

Bases de Datos

Carrera Programador full-stack

Actualización de Datos

Actualización de datos

Sintaxis

Sentencias para agregar, modificar y eliminar datos de tablas.

```
INSERT INTO <nombre_tabla> (<columna1>,  
<columna2>, <columna3>, ...)  
VALUES (<valor1>, <valor2>, <valor3>, ...);
```

```
DELETE FROM <nombre_tabla> [WHERE <condicion>]  
[ORDER BY ...] [LIMIT <cantidad_filas>];
```

```
UPDATE <nombre_tabla> SET  
<col_nombre_1>={<valor1>|DEFAULT} [,  
<col_nombre_2>={<valor2>|DEFAULT}]  
[WHERE <condicion>]  
[ORDER BY ...] [LIMIT <cantidad_filas>];
```

Actualización de datos

Función	<code>`INSERT`</code>	<code>`UPDATE`</code>	<code>`DELETE`</code>
Propósito	Agrega nuevos registros (filas) a una tabla.	Modifica los valores existentes en una o más filas de una tabla.	Elimina registros (filas) de una tabla.
Acción	Añade una nueva fila con los valores proporcionados.	Modifica los valores de una o más columnas en una o más filas.	Elimina filas que cumplan con una condición especificada o todas las filas si no se especifica una condición.
Sintaxis básica	<code>`INSERT INTO nombre_tabla (columna1, columna2) VALUES (valor1, valor2);`</code>	<code>`UPDATE nombre_tabla SET columna1 = nuevo_valor1, columna2 = nuevo_valor2 WHERE condicion;`</code>	<code>`DELETE FROM nombre_tabla WHERE condicion;`</code>
Requiere condición	No es necesario.	Se puede especificar una condición para actualizar filas específicas.	Puede o no especificar una condición para eliminar filas.
Puede afectar múltiples filas	Sí, se pueden insertar múltiples filas en una sola sentencia <code>`INSERT`</code> .	Sí, se pueden actualizar múltiples filas que cumplan con la condición.	Sí, se pueden eliminar múltiples filas que cumplan con la condición o todas las filas si no se especifica una condición.

INSERT

Se utiliza para agregar nuevas filas de datos a una tabla determinada.

- Debemos especificar el nombre de la tabla y una lista de valores para la fila.
- Los valores deben suministrarse en el mismo orden en el que se especificaron los atributos correspondientes en el comando CREATE TABLE.

```
INSERT INTO <nombre_tabla> (<columna3>,  
<columna2>, <columna1>,...)  
VALUES (<valor3>, <valor2>, <valor1>,...);
```

INSERT

Ejemplo

Insertar el teléfono móvil 229 - 4639675 para el cliente cuyo número (nro_cliente) es 50.

```
INSERT INTO E01_TELEFONO  
VALUES (229, 4639675, 'M', 50);
```

Una segunda forma de la sentencia INSERT permite especificar explícitamente los nombres de los atributos que se corresponden con los valores suministrados en el comando INSERT.

```
INSERT INTO  
E01_TELEFONO (nro_telefono, tipo,  
codigo_area, nro_cliente)  
VALUES (4547894, 'M', 249, 50);
```

INSERT

Uso	Descripción	Ejemplo
Inserción de múltiples filas	Inserta varios registros en una sola consulta <code>INSERT</code> .	<code>INSERT INTO tabla (columna1, columna2) VALUES (valor1_1, valor1_2), (valor2_1, valor2_2), ...;</code>
Inserción condicional	Permite insertar datos basados en una condición usando <code>WHERE</code> .	<code>INSERT INTO tabla (columna) VALUES (valor) WHERE condicion;</code>
Inserción desde otra tabla o consulta	Permite insertar datos desde otra tabla o mediante una consulta.	<code>INSERT INTO tabla_destino (columna1, columna2) SELECT columna3, columna4 FROM otra_tabla WHERE condicion;</code>
Inserción con valores predeterminados	Permite especificar valores predeterminados para las columnas.	<code>INSERT INTO tabla (columna1, columna2, columna3) VALUES (valor1, valor2, DEFAULT);</code>
Inserción utilizando secuencias (autoincremento)	Permite insertar datos en una columna autoincrementada omitiendo esa columna en la sentencia <code>INSERT</code> .	<code>INSERT INTO tabla (columna1, columna2) VALUES (valor1, valor2);</code> (columna1 es autoincrementada)

UPDATE

Se utiliza para modificar los valores de atributo de una o más filas seleccionadas.

- La cláusula **WHERE** selecciona las filas que se van a modificar.
- La cláusula **SET** especifica los atributos que se modificarán y sus nuevos valores.

```
UPDATE <nombre_tabla>  
SET <col_nombre_1>={<valor1>|DEFAULT} [,  
<col_nombre_2>={<valor2>|DEFAULT}]  
[WHERE <condicion>]  
[ORDER BY ...]  
[LIMIT <cantidad_filas>];
```

Actualización de Datos

Ejemplo

Cambiar el nombre del cliente número “15” por “Juan”.

```
UPDATE e01_cliente
SET
    nombre = 'Juan'
WHERE
    nro_cliente = 15;
```


UPDATE

La actualización del valor de una clave puede propagarse a los valores de la *foreign key* de las filas de otras relaciones en caso de haberse especificado una opción de acción referencial en las restricciones de integridad referencial del DDL.

UPDATE

Actualizar valores en una columna	Cambia el valor de una columna específica en una o más filas.	<code>`UPDATE tabla SET columna = nuevo_valor WHERE condicion;`</code>
Actualizar múltiples columnas	Actualiza varios campos a la vez en una fila.	<code>`UPDATE tabla SET columna1 = valor1, columna2 = valor2 WHERE condicion;`</code>
Actualizar basado en una subconsulta	Utiliza una subconsulta en la cláusula <code>`SET`</code> para obtener valores de otra tabla y actualizar en consecuencia.	<code>`UPDATE tabla1 SET columna = (SELECT columna FROM tabla2 WHERE condicion) WHERE condicion;`</code>
Actualizar con valores calculados	Utiliza expresiones para calcular nuevos valores durante la actualización.	<code>`UPDATE tabla SET columna = columna * 1.1 WHERE condicion;`</code>
Actualizar basado en un patrón o substring	Actualiza registros que coincidan con un patrón específico o contengan ciertos substrings.	<code>`UPDATE tabla SET columna = REPLACE(columna, 'viejo', 'nuevo') WHERE columna LIKE '%viejo%';`</code>
Actualizar usando valores de otras filas	Utiliza valores de otras filas para actualizar una columna.	<code>`UPDATE tabla t1 SET columna = (SELECT MAX(valor) FROM otra_tabla t2 WHERE t2.clave = t1.clave) WHERE condicion;`</code>
Actualizar basado en una condición compleja	Utiliza condiciones complejas en la cláusula <code>`WHERE`</code> para actualizar filas que cumplan con criterios específicos.	<code>`UPDATE tabla SET columna = nuevo_valor WHERE (condicion1 AND condicion2) OR condicion3;`</code>

DELETE

El comando DELETE elimina filas de una tabla.

- Con la cláusula WHERE se seleccionan las filas que se van a eliminar.

```
DELETE FROM <nombre_tabla>  
[WHERE <condicion>]  
[ORDER BY ...]  
[LIMIT <cantidad_filas>];
```

DELETE

Ejemplo

Borrar todos los teléfonos del cliente número “20”.

```
DELETE FROM e01_telefono  
WHERE  
    nro_cliente = 20;
```

DELETE

La eliminación se puede propagar a filas de otras tablas si se han especificado opciones de acciones referenciales en las restricciones de integridad referencial del DDL.

DELETE

Uso	Descripción	Ejemplo
Eliminar todos los registros de una tabla	Elimina todos los registros de una tabla sin especificar una condición.	<code>`DELETE FROM tabla;`</code>
Eliminar registros basados en una subconsulta	Utiliza una subconsulta en la cláusula <code>`WHERE`</code> para eliminar registros basados en los resultados de otra consulta.	<code>`DELETE FROM tabla WHERE columna IN (SELECT columna FROM otra_tabla WHERE condicion);`</code>
Eliminar registros basados en otra tabla	Elimina registros de una tabla basándote en registros de otra tabla.	<code>`DELETE FROM tabla1 WHERE EXISTS (SELECT * FROM tabla2 WHERE tabla1.columna = tabla2.columna);`</code>
Eliminar registros duplicados	Elimina registros duplicados de una tabla.	<code>`DELETE t1 FROM tabla t1 JOIN tabla t2 ON t1.columna = t2.columna WHERE t1.id > t2.id;`</code>
Eliminar con límites (versiones recientes de MySQL)	Utiliza <code>`LIMIT`</code> para especificar cuántos registros quieres eliminar.	<code>`DELETE FROM tabla LIMIT 10;`</code>

Bases de Datos

Carrera Programador full-stack

Actualización de Datos

Ejercicios

1. Insertar el producto "turrón" de la empresa "misky" con un precio de 230\$ y un stock de 100 unidades.
2. Actualizar el código de área por "526" de los teléfonos que tenían código de área "551".
3. Borrar el producto insertado en el punto 1.

Bases de Datos

Carrera Programador full-stack

Actualización de Datos

Sentencias de Actualización de Datos

Repaso

- **INSERT.** Se utiliza para agregar nuevas filas de datos a una tabla determinada.

Sintaxis:

```
INSERT INTO <nombre_tabla> (<columna1>,  
<columna2>, <columna3>, ...)  
VALUES (<valor1>, <valor2>, <valor3>, ...);
```

Ejemplo:

```
INSERT INTO E01_TELEFONO(nro_telefono, tipo,  
codigo_area, nro_cliente) VALUES  
(4547894, 'M', 249, 50);
```

Sentencias de Actualización de Datos

Repaso

- **UPDATE.** Se utiliza para modificar los valores de atributo de una o más filas seleccionadas.

Sintaxis:

```
UPDATE <nombre tabla> SET  
<col nombre 1>={<valor1>| DEFAULT} [,  
<col nombre 2>={<valor2>| DEFAULT} ]  
[ WHERE <condición> ]  
[ ORDER BY ... ] [ LIMIT <cantidad_filas> ] ;
```

Ejemplo:

```
UPDATE e01_cliente SET nombre = 'Juan '  
WHERE nro_cliente = 15;
```

Sentencias de Actualización de Datos

Repaso

- **DELETE.** Elimina filas de una tabla.

Sintaxis:

```
DELETE FROM <nombre_tabla> [WHERE  
<condicion>] [ORDER BY ...] [LIMIT  
<cantidad_filas>];
```

Ejemplo:

```
DELETE FROM e01_telefono  
WHERE nro_cliente = 20;
```

Lenguaje SQL

Algunas funciones del estándar SQL son:

- **DDL**
 - Definición de datos:
 - Creación de tablas (CREATE)
 - Modificación de tablas (ALTER)
 - Eliminación de tablas (DROP)
- **DML**
 - Consulta de datos
 - Selección (SELECT)
 - **Actualización de los datos**
 - Inserción (INSERT)
 - Actualización (UPDATE)
 - Eliminación (DELETE)

Bases de Datos

Carrera Programador full-stack

Actualización de Datos (Resolución)

INSERT

Problema: Insertar el producto "turron" de la empresa "misky" con un precio de 4\$ y un stock de 100 unidades.

Solución:

```
INSERT INTO
    e01_producto (codigo_producto, marca,
        nombre, descripcion, precio, stock)
VALUES (102, "Misky", "turron",
    "turron de mani", "4", 100);
```

UPDATE

Problema: Actualizar el código de área por "526" de los teléfonos que tenían código de área "551"

Solución:

```
UPDATE e01_telefono  
SET  codigo_area = 526  
WHERE  codigo_area = 551;
```


DELETE

Problema: Borrar el producto insertado en 1

Solución:

```
DELETE FROM e01_producto  
WHERE nombre = 'turron'  
      AND marca = 'misky';
```