

Bases de Datos I

Unidad VIII

**Lenguaje de Definición de Datos
SQL/DML**

INSERT, UPDATE y DELETE



SQL/DML

- **Lenguaje de manipulación de datos (LMD) o Data Manipulation Language (DML).**
- **Formado por sentencias que permiten la manipulación de la información almacenada en la base de datos.**
- **Contiene sentencias para agregar (INSERT), modificar (UPDATE), eliminar (DELETE) y consultar (SELECT) de la información.**
- **En realidad no son las únicas, también existen sentencias de borrado completo (TRUNCATE) y mezcla (MERGE).**

Sentencias

- **INSERT**

Permite agregar filas a una tabla de la base de datos.

- **UPDATE**

Permite modificar las filas de una tabla de la base de datos.

- **DELETE**

Permite borrar las filas de una tabla de la base de datos.

Sintaxis básica INSERT

- **INSERT INTO [tabla] [(cols)]
VALUES [(fila1), (fila2)];**
- **Agrega a la tabla [tabla], una o varias filas según lo especificado.**

Ejemplo

- Suponiendo que tenemos definida en nuestra base de datos, la siguiente tabla:

```
CREATE TABLE producto(  
    codigo INTEGER PRIMARY KEY,  
    descripcion VARCHAR(50) NOT NULL,  
    precio NUMERIC(10,2) NOT NULL DEFAULT 0,  
    alta DATE DEFAULT CURRENT_DATE  
);
```

Ejemplo INSERT

- **Agregamos una fila, indicando los valores para cada de las columnas:**

```
INSERT INTO producto (codigo, descripcion, precio, alta)  
VALUES(1, 'Aceite', 30, '2014-10-10');
```

Ejemplo INSERT

- **Agregamos una fila, indicando los valores para cada de las columnas:**

```
INSERT INTO producto  
VALUES(1, 'Aceite', 30, '2014-10-10');
```

- **Nótese que no se indicaron las columnas de la tabla, ya que se respetó el orden de definición de las mismas.**

Ejemplo INSERT

- **Agregamos una fila, dejando algunas columnas a sus valores por defecto:**

```
INSERT INTO producto(codigo, descripcion, precio)  
VALUES(1, 'Aceite', 30);
```

- **Nótese que no se indicó valor para la columna fecha. Si la columna no tiene valor por defecto, tomará por defecto nulo (NULL).**

Ejemplo INSERT

- **Agregamos dos filas, dejando algunas columnas a sus valores por defecto:**

```
INSERT INTO producto(descripcion, precio, codigo)  
VALUES('Aceite', 30, 1),('Jugo', 20, 2);
```

- **Nótese que al indicar las columnas, no fue necesario respetar el orden de definición de las mismas.**

Sintaxis básica UPDATE

- **UPDATE [tabla] SET**
[asignaciones]
WHERE [condiciones];
- **Modifica una o varias columnas, de todas las filas de la tabla que cumplan con las condicione especificadas.**
- **ATENCIÓN: si no se especifica una condición, modifica todas las filas!!!**

Ejemplo UPDATE

- **Modificamos la descripción del producto con código 1:**

```
UPDATE producto SET  
    descripcion = 'Aceite de girasol'  
WHERE codigo = 1;
```

Ejemplo UPDATE

- **Modificamos la descripción y el precio del producto con código 2:**

```
UPDATE producto SET  
    descripcion = 'Jugo de naranja',  
    precio = 40  
WHERE codigo = 2;
```

- **Nótese que cada una de las asignaciones, se separan con una coma (,).**

Ejemplo UPDATE

- **Modificamos la fecha al valor por defecto del producto con código 2:**

```
UPDATE producto SET  
    alta = DEFAULT  
WHERE codigo = 2;
```

- **Nótese que se usó la palabra clave DEFAULT, en vez de una expresión en la asignación.**

Ejemplo UPDATE

- **Aumentamos un 20% el precio de todos los aceites:**

UPDATE producto **SET**

precio = precio * 1.2

WHERE descripcion **LIKE** '%Aceite%';

- **El operador LIKE evalúa a verdadero, cuando la cadena de caracteres izquierda, concuerda con el patrón de la derecha. El símbolo % significa 0 o más caracteres y el símbolo _, significa un caracter.**

Ejemplo UPDATE

- **Aumentamos un 20% el precio de todos los productos:**

```
UPDATE producto SET  
precio = precio * 1.2;
```

- **Nótese que este ejemplo se omitió la cláusula WHERE, lo que es igual a WHERE TRUE. Por lo que todas las filas de la tabla son modificadas**

Sintaxis básica DELETE

- **DELETE**
FROM [tabla]
WHERE [condicion];
- **Borra las filas de la tabla, que cumplen con la condición especificada.**
- **ATENCIÓN: si no se especifica una condición, borra todas las filas!!!**

Ejemplo DELETE

- **Borramos el producto con código 1:**

```
DELETE FROM producto  
WHERE codigo = 1;
```

- **Como realizamos la selección por una columna que es clave primaria, esta sentencia borra 0 o 1 fila.**

Ejemplo DELETE

- **Borramos todos los aceites:**

DELETE FROM producto

WHERE descripcion **LIKE** '%Aceite%';

- **El operador LIKE evalúa a verdadero, cuando la cadena de caracteres izquierda, concuerda con el patrón de la derecha. El símbolo % significa 0 o más caracteres y el símbolo _, significa un caracter.**

Ejemplo DELETE

- **Borramos todos los productos:**

DELETE FROM producto;

- **Nótese que este ejemplo se omitió la cláusula WHERE, lo que es igual a WHERE TRUE. Por lo que todas las filas de la tabla son eliminadas.**
- **Para borrar todo el contenido de una tabla existe la sentencia TRUNCATE.**

Sobre la sentencia DELETE

- **La sentencia DELETE realiza un borrado físico de los datos.**
- **Una vez borrados no se pueden recuperar (a menos que tengamos una copia de resguardo).**
- **En la practica habitual, es una operación que raramente se utiliza en el día a día.**
- **Por lo general se realiza un borrado lógico de los datos.**

¿Qué es el borrado lógico de datos?

- **Se utiliza algún tipo de marca, para denotar cuando algún dato ha perdido validez.**
- **Se debe tener en cuenta al consultar los datos, solo incluir aquellos que tiene validez aún.**
- **Esta práctica dejará de ser habitual el día que todos los SGBD implementen el estándar SQL:2011 de bases de datos temporales.**

Ejemplo - Borrado lógico

- Podemos redefinir nuestra tabla de productos de la siguiente forma:

```
CREATE TABLE producto(  
    codigo INTEGER PRIMARY KEY,  
    descripcion VARCHAR(50) NOT NULL,  
    precio NUMERIC(10,2) NOT NULL DEFAULT 0,  
    alta DATE NOT NULL DEFAULT CURRENT_DATE,  
    baja DATE  
);
```

Ejemplo - Borrado lógico

- Para consultar los productos:

```
SELECT *  
FROM producto  
WHERE baja IS NULL;
```

- Para borrar un producto:

```
UPDATE producto SET  
  baja = CURRENT_DATE  
WHERE codigo = 779368;
```

Ejemplo - Borrado lógico

- Para consultar los productos:

```
SELECT *  
FROM producto  
WHERE baja IS NULL;
```

- Para borrar un producto:

```
UPDATE producto SET  
  baja = CURRENT_DATE  
WHERE codigo = 779368;
```


Bibliografía

- **SQL-99 Complete, Really. 1999. Peter Gultzan y Trudy Pelzer.**
- **PostgreSQL Introduction and Concepts. 2001. Momjian, Bruce.**
- **PostgreSQL 9.6 Documentation. 2016. The PostgreSQL Global Development Group.**