



Laboratorio 6

Quicklink no disponible

El Herramienta de aprendizaje externa no está disponible.
Volver

Enlace directo: [Video Update y Delete](#) 

Quicklink no disponible

El Herramienta de aprendizaje externa no está disponible.
Volver

Enlace directo: [Video Alter Table](#) 



En este laboratorio realizaremos modificaciones en los datos que hemos guardado usando las sentencias UPDATE y DELETE, y modificaciones en los esquemas de las tablas usando la sentencia ALTER TABLE.

Sentencia UPDATE

La sintaxis básica del UPDATE es:

```
UPDATE [ schema. ] table_name [ alias ]  
SET { column = { expression | ( subquery ) | DEFAULT }  
      | ( column [, column ]... ) = ( subquery ) } [ , ... ]  
[ WHERE condition ]
```

expression es una expresión que incluye el uso de funciones, operadores, columnas o valores literales.

condition es una expresión que al evaluarse entrega un valor booleano

La cláusula UPDATE indica la tabla que se va a modificar.

La cláusula SET indica la(s) columna(s) cuyos valores se van a modificar. Los valores que se van a asignar se especifican con una expresión o con un subquery. El subquery debe retornar exactamente un valor para cada fila que se va a actualizar.

La cláusula WHERE permite restringir las filas que se van a actualizar. Solo se actualizan aquellas para las que la condición especificada es verdadera.

Ejemplos:

Dado el siguiente modelo relacional, se dan algunos ejemplos de uso del UPDATE.

- Cambiar la descripción del producto de código 58

```
UPDATE producto
SET     descripcion = 'Azucar'
WHERE  cod = 58;
```

- Cambiar el porcentaje de descuento otorgado a los clientes de la ciudad de Cali

```
UPDATE cliente
SET     descuento = 0.10
WHERE  codCiud IN (SELECT codCiud FROM ciudad WHERE
nombre='Cali');
```

- Cambiar en la tabla compra el precio de los productos vendidos el día de hoy, tomando el precio de la tabla producto

```
UPDATE (SELECT c.fecha, c.precio AS precio_compra,
              p.precio AS precio_prod
        FROM  compra c JOIN producto p ON (c.prod=p.cod)) cp
SET     cp.precio_compra = cp.precio_prod
WHERE  TRUNC(fecha) = TRUNC(CURRENT_DATE);
```

Sentencia DELETE

La sintaxis básica del DELETE es:

DELETE FROM [<i>schema.</i>] <i>table_name</i> [<i>alias</i>]
[WHERE <i>condition</i>]

La cláusula DELETE indica la tabla donde se van a borrar registros.

La cláusula WHERE permite seleccionar las filas que se van a borrar. Solo se actualizan aquellas para las que la condición especificada es verdadera.

Ejemplos:

Dado el modelo relacional de la figura anterior, se dan algunos ejemplos de uso del DELETE.

- Borrar un registro de la tabla compra

```
DELETE FROM compra
WHERE  cc = 256 AND prod = 58
AND TRUNC(fecha) = '20/08/2020';
```

- Borrar las compras del día de hoy del cliente Carlos Torres

```
DELETE FROM compra
WHERE  cc IN (SELECT cc FROM cliente WHERE nombre = 'Carlos Torres')
AND    TRUNC(fecha) = '14/08/2020';
```

Sentencia ALTER TABLE

La sintaxis básica del ALTER TABLE es:

```
ALTER TABLE [schema.]table_name action
```

action es:

```
RENAME TO new_table_name
RENAME COLUMN column TO new_name
ADD ( column data_type [ inline_constraint [ ... ] ] [, ...] )
DROP( column [, ...] ) [CASCADE CONSTRAINTS]
MODIFY( column [ data_type ] [ inline_constraint [ ... ] ] [, ...] )
ADD table_constraint [, ...]
DROP PRIMARY KEY [ CASCADE ]
DROP UNIQUE ( column [, ...] )
DROP CONSTRAINT constraint_name [ CASCADE ]
```

data_type es el tipo de dato (dominio) de la columna^[1].

inline_constraint permite definir reglas de integridad^[2] como parte de la definición de la columna. Estas reglas pueden ser:

```
[ { NOT NULL | CHECK ( expression ) | DEFAULT default_expr |
  UNIQUE | PRIMARY KEY |
  REFERENCES [refschema.]reftable [ ( refcolumn ) ] [ON DELETE
    {CASCADE|SET NULL} ] } ]
```

table_constraint permite definir reglas de integridad que involucran varias columnas. Estas reglas pueden ser:

```
{ CHECK ( expression ) | UNIQUE ( column [, ... ] ) |
  PRIMARY KEY ( column [, ... ] ) |
  FOREIGN KEY ( column [, ... ] ) REFERENCES reftable [ ( refcolumn [,
  ... ]
    ) ] [ ON DELETE {CASCADE|SET NULL} ] }
```

expression es una expresión que retorna un valor booleano

default_expr es un valor que se asigna a la columna cuando se inserta un registro que no provee ese valor. La expresión puede ser un valor literal, una función, o el llamado a una función (ej. SYSDATE)

reftable, *refcolumn*, *refschema* son, respectivamente, los nombres de la tabla y columna a la que hace referencia, y del esquema donde está la tabla

Ejemplos:

- Cambiar el nombre de la tabla **Compra** por **ComprasCliente**

```
ALTER TABLE Compra RENAME TO ComprasCliente;
```
- Cambiar el nombre del atributo **precio** en la tabla **Producto** por **precioUnitario**

```
ALTER TABLE producto RENAME COLUMN precio TO precioUnitario;
```
- Agregar restricción de **unicidad** a la **descripción** del **producto**

```
ALTER TABLE Producto MODIFY descripción UNIQUE;
```
- Cambiar la definición de la columna **descripción** de la tabla **Producto**

```
ALTER TABLE Producto MODIFY ( descripcion VARCHAR2(60) DEFAULT 'Bonito');
```
- Borrar la columna **descripción** de la tabla **Producto**

```
ALTER TABLE Producto DROP ( descripcion ) CASCADE CONSTRAINTS;
```
- Agregar una llave primaria compuesta en la tabla **OtraTabla**:

```
ALTER TABLE OtraTabla ADD PRIMARY KEY (colTipo, colCodigo);
```

Ejercicios

Usando el esquema de base de datos creado en los laboratorios anteriores, escriba una sentencia SQL para cada uno de los requerimientos de ésta sección.

Nota. La referencia de las sentencias UPDATE y DELETE y otras referencias se encuentran en:

- [Update](#)
- [Delete](#)
- [Tipos de dato](#)
- [Restricciones de integridad](#)

1. Cambie la **direccion** y la **ciudad** del **usuario Pedro Ramos** por 'Av. Sur # 3 -21' y 103 respectivamente.
2. Actualice las compras del 15 de diciembre del 2024, cambie **dirEntrega** y **ciudadEntrega** por la dirección y ciudad del usuario que realizó la compra.**Nota:** aunque en la base de datos solo hay una compra en esa fecha, la sentencia debe funcionar sin importar cuantas compras hubiesen.
3. Borre del carrito de compras del usuario **Camila Garcia** el producto **Auriculares con orejas de gato**
4. En la tabla **Producto** cambiar el nombre del atributo **Categoria** por **codCategoria**
5. Agregar las siguientes restricciones a atributos del modelo:
 - a) En la tabla **Compra** el atributo **ciudadEntrega** debe ser **obligatorio**
 - b) En la tabla **Producto** el atributo **cantidadInv** debe tener por **defecto** el valor cero
 - c) En la tabla **productoCompra** el atributo **Cantidad** debe ser **mayor que cero**
- 6 Agregar a la tabla **Inventario** el atributo **Peso**, un número de 3 dígitos cuyo valor por defecto es 1, y debe ser positivo
7. Se requiere cambiar la clave primaria de la tabla **Compra**, actualmente es un clave compuesta por los atributos (**usuario**, **nrocompra**), para la nueva clave se debe agregar el atributo **idCompra** (número de 5 dígitos declarado como identidad –use la opción ALWAYS para que genere automáticamente el consecutivo cuando se agregue el atributo a la tabla).Tenga en cuenta que la tabla **productoCompra** hace referencia a **Compra**, por lo tanto, en **productoCompra** también cambia la clave foránea y la clave primaria. Este cambio debe realizarse garantizando que la información de los productos de cada compra se conserve sin alteraciones. Sea cuidadoso(a) con el orden en que realiza las modificaciones.

Al finalizar la sesión, cada estudiante debe enviar el script a [Entrega Laboratorio 6](#)
