

Módulo de adaptación

Master en Business Intelligence y Big Data

PROFESOR/A
Antonio Sarasa Cabezuelo

Inserción de filas en una tabla

- Para poder consultar los datos de una base de datos hay que introducirlos con la sentencia **INSERT INTO VALUES:**

```
INSERT INTO nombre_tabla [(columnas)]  
{VALUES ({v1 | DEFAULT | NULL}, ...,  
{vn/DEFAULT/NULL}) | <consulta>};
```

Inserción de filas en una tabla

- Los valores v_1, v_2, \dots, v_n se deben corresponder con las columnas de la tabla especificada y deben estar en el mismo orden, a menos que las volvamos a poner a continuación del nombre de la tabla. En este último caso, los valores se deben disponer de forma coherente con el nuevo orden.
- Si se quiere especificar que un valor por omisión se usa la palabra reservada DEFAULT, y si se trata de un valor nulo se usa la palabra reservada NULL.

Inserción de filas en una tabla

- Observar que para insertar más de una fila con una sola sentencia, se deben obtener los datos mediante una consulta a otras tablas.

Inserción de filas en una tabla

- Por ejemplo si se quiere insertar en una tabla clientes que tiene las columnas :nif, nombre_cli, codigo_cli, telefono, direccion, ciudad, se podría hacer de dos formas:

```
INSERT INTO clientes  
VALUES (10, 'Mercadona', '122233444-C', 'Gran vida 8',  
'Madrid', DEFAULT);
```

Inserción de filas en una tabla

O bien:

```
INSERT INTO clientes(nif, nombre_cli, codigo_cli, telefono,  
direccion, ciudad)  
VALUES ('122233444-C', 'Mercadona', 10, DEFAULT, 'Gran vida  
8', 'Madrid');
```

Inserción de filas en una tabla

- Insertar un préstamo en la relación Préstamo

```
INSERT INTO Prestamo  
VALUES ('Navacerrada', 'Pepe Pérez', 125.000)
```

- También es posible obtener los datos mediante una consulta SELECT que actúe como proveedor de datos.

```
INSERT INTO Prestamo  
SELECT * FROM Nuevos_Prestamos
```

Borrado de filas de una tabla

- Para borrar valores de algunas filas de una tabla se usa la sentencia **DELETE**:

```
DELETE FROM nombre_tabla [WHERE  
condiciones];
```

- Observar que si no se utiliza la cláusula **WHERE** se borran todas las filas de la tabla, en cambio si se utiliza **WHERE** entonces solo se borran aquellas filas que cumplen las condiciones especificadas.

Borrado de filas de una tabla.

- Por ejemplo si se quieren borrar todas las filas de la tabla proyectos se usaría la sentencia:

```
DELETE FROM proyectos;
```

- Sin embargo si solo se quieren borrar las filas de la tabla en las que el valor de la columna cliente vale 12, entonces se usaría la sentencia:

```
DELETE FROM proyectos WHERE codigo_cliente = 12;
```

Borrado de filas de una tabla.

- Borrar todos los clientes que tengan un préstamo no registrado en la relación Préstamo.

```
DELETE  
FROM Clientes  
WHERE Clientes.NumPrestamo NOT IN  
(SELECT NumPrestamo FROM Prestamo)
```

Modificación de filas de una tabla

- Para modificar los valores de algunas filas de una tabla se usa la sentencia **UPDATE**:

```
UPDATE nombre_tabla  
SET columna = {expresión | DEFAULT | NULL}  
[, columna = {expr | DEFAULT | NULL} ...]  
WHERE condiciones;
```

- La cláusula SET indica qué columna modificar y los valores que puede recibir, y la cláusula WHERE especifica qué filas deben actualizarse.

Modificación de filas de una tabla.

- Por ejemplo si se quiere inicializar el sueldo de todos los empleados del proyecto 2 en 500 euros:

```
UPDATE empleados SET sueldo = 500  
WHERE num_proyec = 2;
```

Modificación de Datos

- La parte **WHERE** es opcional y, si no se especifica, se actualizarán todas las tuplas de la tabla.

```
UPDATE Prestamo  
SET importe=200.000  
WHERE NumPrestamo='P-170'
```

Modificación de Datos

- La clausula **WHERE** admite consultas anidadas

Modificar todos los prestamos cuya sucursal hay sido cerrada a la sucursal 'Centro'.

```
UPDATE Prestamo  
SET sucursal= 'Centro'  
WHERE sucursal IN  
      (SELECT sucursal  
       FROM Sucursales_Cerradas)
```