

BASES DE DATOS

# Manipulación de datos

©Jesús García, 2021 | <http://jgarcia.dev>



BASES DE DATOS

# Manipulación de datos

## Introducción

Inserción

Modificación

Borrado

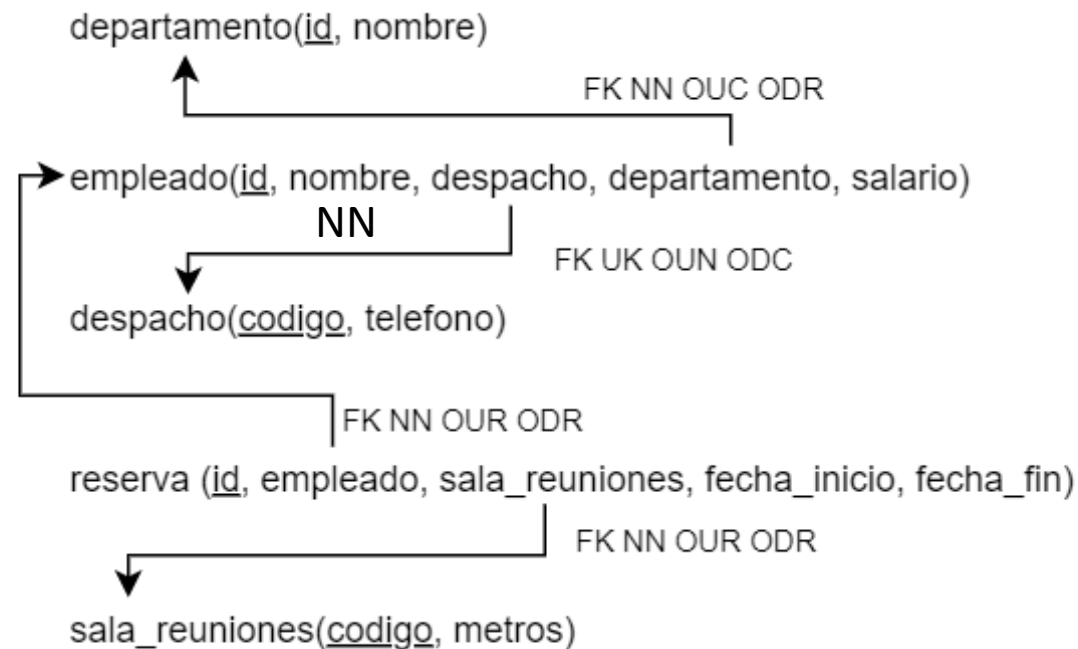
# Introducción

Para la manipulación de los datos de la base de datos se utiliza el lenguaje de manipulación de datos (*Data Manipulation Language* o *DML*).

Así pues, DML se encarga fundamentalmente de la inserción, modificación y eliminación de los datos.

# Introducción

Todos los ejemplos de este tema hacen uso de la base de datos “oficina” cuyo grafo relacional es el siguiente:



- La fecha de fin de reserva debe ser posterior a la fecha de inicio.
- El salario por defecto es 1200€.

Puedes crear la base de datos ejecutando el *script oficina.sql*

BASES DE DATOS

# Manipulación de datos

Introducción

**Inserción**

Modificación

Borrado

# Inserción

Permite insertar una o varias filas en una tabla.

```
INSERT [INTO] nombreDeTabla  
    [ (nombreDeLaColumna [, nombreDeLaColumna] ...) ]  
    {VALUES | VALUE} (listaValores) [, (listaValores) ...];
```

## Cómo interpretar la sintaxis de los comandos

Las llaves {} con una o varias barras | significan opcionalidad. Se debe elegir una de las opciones entre las barras. Por ejemplo, si elegimos VALUES no escribiremos VALUE.

# Inserción

Permite insertar una o varias filas en una tabla.

```
INSERT [INTO] nombreDeTabla  
    [ (nombreDeLaColumna [, nombreDeLaColumna] ... ) ]  
    {VALUES | VALUE} (listaValores) [, (listaValores) ...];
```

Ejemplos:

```
INSERT INTO sala_reuniones VALUES ('SR01', 20.5);  
INSERT INTO departamento VALUES (DEFAULT, 'Ciencias');
```

Puedes encontrar más ejemplos en el fichero *oficina-inserts.sql*

## Cómo comprobar que todo va bien

Para comprobar que las sentencias funcionan correctamente puedes utilizar la sentencia SELECT en su versión más sencilla:

```
SELECT * FROM nombreTabla;
```

Por ejemplo:

```
SELECT * FROM sala_reuniones;
```

Permite comprobar que los datos se han insertado correctamente en la tabla *sala\_reuniones*.



BASES DE DATOS

# Manipulación de datos

Introducción

Inserción

**Modificación**

Borrado

# Modificación

Permite modificar filas en una tabla.

```
UPDATE nombreTabla
```

```
    SET nombreColumna1=valor1 [, nombreColumna2=valor2...]
```

```
    [WHERE condición]
```

En la condición podemos utilizar los operadores lógicos y relacionales:

>, <, >=, !=, <=, =, IS NULL, IS NOT NULL, AND, NOT, OR

Y referirnos a múltiples columnas de la tabla.

# Modificación

Permite modificar filas en una tabla.

```
UPDATE nombreTabla  
    SET nombreColumna1=valor1 [,nombreColumna2=valor2...]  
    [WHERE condición]
```

Ejemplo:

```
UPDATE sala_reuniones  
    SET metros = 30  
    WHERE codigo = 'SR01';
```

# Modificación

Permite modificar filas en una tabla.

```
UPDATE nombreTabla
```

```
    SET nombreColumna1=valor1 [, nombreColumna2=valor2...]
```

```
    [WHERE condición]
```

A la hora de asignar el valor se pueden realizar operaciones matemáticas mediante los operadores aritméticos:

$+, -, *, /$

# Modificación

Permite modificar filas en una tabla.

```
UPDATE nombreTabla  
    SET nombreColumna1=valor1 [, nombreColumna2=valor2...]  
    [WHERE condición]
```

Ejemplo:

```
UPDATE empleado  
    SET salario = salario * 1.15  
    WHERE id > 2;
```

# Modificación

Permite modificar filas en una tabla.

```
UPDATE nombreTabla  
    SET nombreColumna1=valor1 [, nombreColumna2=valor2...]  
    [WHERE condición]
```



Si no indicamos la cláusula WHERE se modificarán todas las filas de la tabla.

Ejemplo:

```
UPDATE sala_reuniones SET metros = 20;
```

Puedes encontrar más ejemplos en el fichero *oficina-updates.sql*

BASES DE DATOS

# Manipulación de datos

Introducción

Inserción

Modificación

**Borrado**

# Borrado

Permite borrar filas en una tabla.

```
DELETE FROM nombreTabla  
      [WHERE condición]
```

En la condición podemos utilizar los operadores lógicos y relacionales:

>, <, >=, !=, <=, =, IS NULL, IS NOT NULL, AND, NOT, OR

Y referirnos a múltiples columnas de la tabla.



# Borrado

Permite borrar filas en una tabla.

```
DELETE FROM nombreTabla  
      [WHERE condición]
```

En la condición podemos utilizar los operadores lógicos y relacionales:

>, <, >=, !=, <=, =, IS NULL, IS NOT NULL, AND, NOT, OR

Y referirnos a múltiples columnas de la tabla.

# Borrado

Permite borrar filas en una tabla.

```
DELETE FROM nombreTabla  
    [WHERE condición]
```



Si no indicamos la cláusula WHERE se borrarán todas las filas de la tabla.

Ejemplo:

```
DELETE FROM reserva;
```

Puedes encontrar más ejemplos en el fichero *oficina-deletes.sql*