

BASES DE DATOS

# Consultas sobre una tabla

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



BASES DE DATOS

# Consultas sobre una tabla

## Introducción

Sintaxis básica

Cláusula DISTINCT

Cláusula WHERE

- Operador BETWEEN

- Operador IN

- Operador LIKE

- Operador IS NULL

Cláusula ORDER BY

Cláusula LIMIT

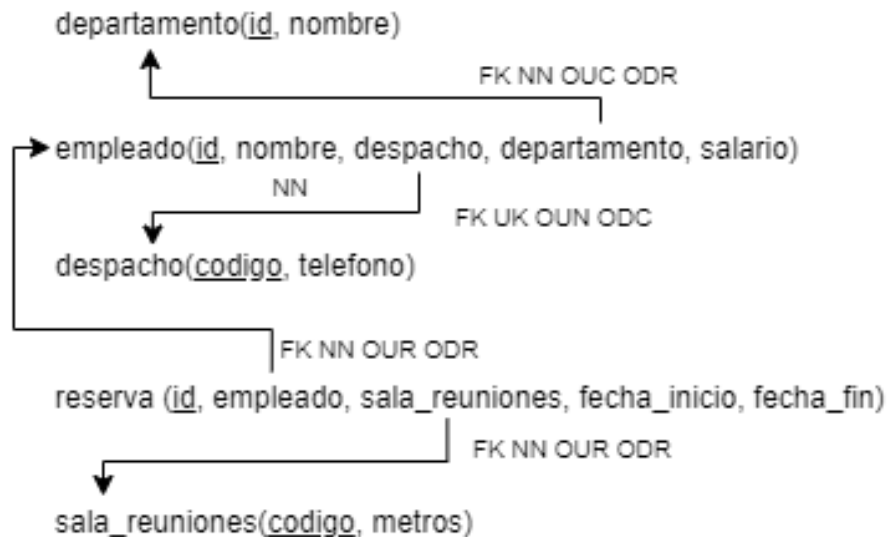
# Introducción

Para la consulta de los datos de la base de datos se utiliza el lenguaje de consulta de datos (*Data Query Language* o *DQL*).

El único comando que pertenece a este lenguaje es **SELECT**, comando muy versátil que permite la selección de filas de acuerdo a ciertos criterios, la realización de cálculo sobre los datos, la mezcla de diversas tablas, etc.

# 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

# Consultas sobre una tabla

Introducción

**Sintaxis básica**

Cláusula DISTINCT

Cláusula WHERE

Operador BETWEEN

Operador IN

Operador LIKE

Operador IS NULL

Cláusula ORDER BY

Cláusula LIMIT

# Sintaxis básica

Obtén toda la información almacenada en la tabla empleado.

```
SELECT * FROM empleado;
```

- El asterisco indica que se quieren obtener todas las columnas.
- Después de la cláusula *FROM* indicamos el nombre de la tabla de la que queremos obtener los datos.

# Sintaxis básica

Obtén el nombre y el salario de los empleados.

```
SELECT nombre, salario FROM empleado;
```

- En vez del asterisco podemos indicar el nombre de las columnas que queremos obtener separadas por comas.

# Sintaxis básica

Obtén el *id* de los empleados, su salario actual y el doble de su salario.

```
SELECT id, salario, salario * 2 FROM empleado;
```

- A la hora de seleccionar las columnas podemos realizar operaciones aritméticas como sumas, restas, multiplicaciones y divisiones.



# Alias

Los alias permite cambiar el nombre de una columna o expresión en el resultado de una sentencia SELECT.

```
SELECT id, salario, salario * 2 AS doble_salario  
FROM empleado;
```

BASES DE DATOS

# Consultas sobre una tabla

Introducción

Sintaxis básica

**Cláusula DISTINCT**

Cláusula WHERE

Operador BETWEEN

Operador IN

Operador LIKE

Operador IS NULL

Cláusula ORDER BY

Cláusula LIMIT

# Cláusula DISTINCT

La cláusula DISTINCT nos permite evitar que se muestren valores repetidos.

Obtén los diferentes salarios de los empleados

```
SELECT DISTINCT salario FROM empleado;
```

BASES DE DATOS

# Consultas sobre una tabla

Introducción

Sintaxis básica

Cláusula DISTINCT

**Cláusula WHERE**

Operador BETWEEN

Operador IN

Operador LIKE

Operador IS NULL

Cláusula ORDER BY

Cláusula LIMIT

# Cláusula WHERE

La cláusula WHERE permite indicar una condición que deben cumplir los registros de salida. Aquellos registros que no la cumplan no aparecerán en el resultado.

Obtén todos los empleados con un salario superior a 1500€.

```
SELECT * FROM empleado WHERE salario > 1500;
```

# Cláusula WHERE

En la condición podemos utilizar los operadores lógicos y relacionales:

>, <, >=, !=, <=, =  
AND, NOT, OR  
IS NULL, IS NOT NULL  
BETWEEN, IN, LIKE

Y referirnos a múltiples columnas de la tabla.

# Cláusula WHERE

Obtén los identificadores de departamento (evitando los resultados repetidos) en los que trabajan los empleados con un salario superior o igual a 1500€.

```
SELECT DISTINCT departamento FROM empleado  
WHERE salario >= 1500;
```

# Cláusula WHERE

Obtén el código de empleado y código de sala de aquellas reservas realizadas durante el mes de marzo de 2021.

```
SELECT empleado, sala_reuniones FROM reserva  
WHERE fecha_inicio >= '2021-03-01 00:00:00' AND  
       fecha_inicio <= '2021-03-31 23:59:59';
```



# Cláusula WHERE

Obtén el nombre de los empleados que trabajan en el departamento 1 y en el departamento 3.

```
SELECT nombre FROM empleado  
    WHERE departamento = 1 OR departamento = 3;
```

# Cláusula WHERE

Obtén el nombre de los empleados del departamento 1 que cobran más de 1500€.

```
SELECT nombre FROM empleado  
    WHERE departamento = 4 AND salario > 1500;
```

# Cláusula WHERE

Obtén el identificador y nombre de aquellos empleados que al incrementarles un 20% su salario superan los 1700€.

```
SELECT id, nombre FROM empleado  
      WHERE (salario * 1.2) > 1700;
```

- En la cláusula WHERE también podemos realizar operaciones aritméticas como sumas, restas, multiplicaciones y divisiones.

BASES DE DATOS

# Consultas sobre una tabla

Introducción

Sintaxis básica

Cláusula DISTINCT

Cláusula WHERE

**Operador BETWEEN**

Operador IN

Operador LIKE

Operador IS NULL

Cláusula ORDER BY

Cláusula LIMIT

# Operador BETWEEN

Permite obtener datos que se encuentren en un rango.

Obtén el nombre de los empleados cuyo identificador se encuentra entre el 3 y el 5 (inclusive).

```
SELECT nombre FROM empleado  
    WHERE id >= 3 AND id <=5;
```

## Operador BETWEEN

Obtén el nombre y el salario de los empleados cuyo salario se encuentra entre 1600€ y 1800€.

```
SELECT nombre, salario FROM empleado  
       WHERE salario BETWEEN 1600 AND 1800;
```

# Operador BETWEEN

También se puede utilizar el operador NOT BETWEEN.

Obtén el código de las salas de reuniones cuyos metros cuadrados no estén entre 15 y 20.

```
SELECT codigo FROM sala_reuniones  
    WHERE metros NOT BETWEEN 15 AND 20;
```

BASES DE DATOS

# Consultas sobre una tabla

Introducción

Sintaxis básica

Cláusula DISTINCT

Cláusula WHERE

Operador BETWEEN

**Operador IN**

Operador LIKE

Operador IS NULL

Cláusula ORDER BY

Cláusula LIMIT



# Operador IN

Permite obtener las filas en los que uno de sus valores está entre una lista de valores.

Obtén los empleados cuyo despacho son DS01 y DS02.

```
SELECT * FROM empleado  
        WHERE despacho IN ('DS01', 'DS02');
```

BASES DE DATOS

# Consultas sobre una tabla

Introducción

Sintaxis básica

Cláusula DISTINCT

Cláusula WHERE

Operador BETWEEN

Operador IN

**Operador LIKE**

Operador IS NULL

Cláusula ORDER BY

Cláusula LIMIT

# Operador LIKE

Permite comprobar si una cadena de texto sigue un cierto patrón. Podemos utilizar los siguientes símbolos

Símbolo	Significado
%	Una secuencia cualquiera de caracteres.
_	Un carácter cualquiera

Obtén el nombre de los empleados que empiezan por 'S'

```
SELECT nombre FROM empleado WHERE nombre LIKE 'S%';
```

## Operador LIKE

Obtén el nombre de aquellos empleados cuya segunda letra es la 'e'.

```
SELECT nombre FROM empleado WHERE nombre LIKE '_e%';
```

BASES DE DATOS

# Consultas sobre una tabla

Introducción

Sintaxis básica

Cláusula DISTINCT

Cláusula WHERE

Operador BETWEEN

Operador IN

Operador LIKE

**Operador IS NULL**

Cláusula ORDER BY

Cláusula LIMIT

# Operador IS NULL

Devuelve verdadero si el valor que se examina es nulo.

Obtén el nombre de aquellos empleados que no tienen despacho.

```
SELECT nombre FROM empleado WHERE despacho IS NULL;
```

## Operador IS NULL

También se puede utilizar el operador IS NOT NULL, que devuelve verdadero cuando el valor no es nulo.

Obtén el nombre de aquellos empleados que tienen despacho.

```
SELECT nombre FROM empleado WHERE despacho IS NOT NULL;
```

BASES DE DATOS

# Consultas sobre una tabla

Introducción

Sintaxis básica

Cláusula DISTINCT

Cláusula WHERE

Operador BETWEEN

Operador IN

Operador LIKE

Operador IS NULL

**Cláusula ORDER BY**

Cláusula LIMIT



# Cláusula ORDER BY

Permite ordenar el resultado por alguna de las columnas.

Muestra el nombre de todos los empleados ordenados alfabéticamente.

```
SELECT nombre FROM empleado ORDER BY nombre ASC;
```

- Después de BY se escribe la o las columnas que se quieren utilizar para la ordenación.
- La ordenación puede ser ascendente (ASC) o descendente (DESC).

## Cláusula ORDER BY

Obtén las salas de reuniones cuya superficie sea superior a 10 metros cuadrados y muéstralas ordenadas según sus dimensiones de mayor a menor.

```
SELECT * FROM sala_reuniones WHERE metros >10  
      ORDER BY metros DESC;
```

BASES DE DATOS

# Consultas sobre una tabla

Introducción

Sintaxis básica

Cláusula DISTINCT

Cláusula WHERE

Operador BETWEEN

Operador IN

Operador LIKE

Operador IS NULL

Cláusula ORDER BY

**Cláusula LIMIT**

# Cláusula LIMIT

Limita el número de filas obtenidas por la consulta.

Obtén el empleado con mayor salario.

```
SELECT * FROM empleado ORDER BY salario DESC LIMIT 1;
```

Si especificamos sólo un número después de la cláusula LIMIT se obtendrá el número de filas indicado.

# Cláusula LIMIT

Obtén el empleado con el segundo mayor salario.

```
SELECT * FROM empleado ORDER BY salario DESC LIMIT 1,1;
```

Si especificamos dos números separados por coma después de la cláusula LIMIT, el primero hace referencia a cuántas filas se ignoran y el segundo cuántas filas se quieren seleccionar a partir de la primera fila no ignorada.