

### Índice.

- 0. Datos
- 1. Consultas básicas
- 2. Consultas con filtros
  - 2.1. Operador de filtro =, >=, <=, <>, !=, <, >
  - 2.2. Operador de pertenencia a conjuntos IN
  - 2.3. Operador de rango BETWEEN
  - 2.4. Test de valor nulo NULL
  - 2.5. Test de patrón LIKE
  - 2.6. Límite en los registros LIMIT
  - 2.7. Test con expresiones regulares REGEXP
- 3. Consultas con ordenación ORDER BY
- 4. Consultas con resumen GROUP BY
- 5. Consultas multitable
  - 5.1. Composición interna INNER JOIN, NATURAL JOIN
  - 5.2. Composición externa RIGHT JOIN, LEFT JOIN
- 6. Subconsultas
  - 6.1. Subconsulta de comparación =, >=, <=, <>, !=, <, >
  - 6.2. Subconsulta de pertenencia a conjunto IN
  - 6.3. Subconsulta de cuantificación ALL, ANY
  - 6.4. Subconsulta de existencia EXISTS
- 7. Consultas reflexivas
- 8. Consultas derivadas

## Realización de consultas

### 0. Datos.

```
#-----
DROP DATABASE IF EXISTS Empresa;
CREATE DATABASE      Empresa;
USE                  Empresa;

#-----
#  CREACIÓN DE LA TABLA OFICINA
#-----
CREATE TABLE Oficina (
    Oficina    VARCHAR( 4 )    NOT NULL,
    Dimensión  INT              NOT NULL,
    Mesas       INT              NOT NULL,
    CONSTRAINT PK_Oficina      PRIMARY KEY( Oficina )
);

#-----
#  CREACIÓN DE LA TABLA EMPLEADO
#-----
CREATE TABLE Empleado (
    idEmpleado  INT,
    Nombre       VARCHAR( 40 )    NOT NULL,
    Apellidos    VARCHAR( 90 )    NOT NULL,
    Oficina      VARCHAR( 10 ),
    PuestoTrabajo VARCHAR( 50 ),
    Jefe         INT,
    CONSTRAINT PK_Empleado      PRIMARY KEY( idEmpleado ),
    CONSTRAINT FK_Empleado_Jefe FOREIGN KEY( Jefe )      REFERENCES Empleado( idEmpleado ),
    CONSTRAINT FK_Empleado_Oficina FOREIGN KEY( Oficina ) REFERENCES Oficina( Oficina )
);

#-----
#  AGREGACIÓN DE LOS DATOS DE LAS OFICINAS DE LA EMPRESA
#-----
INSERT INTO Oficina VALUES ( "OF01", 100, 5 ),
                             ( "OF02", 120, 30 ),
                             ( "OF03", 150, 40 ),
                             ( "OF04", 100, 20 ),
                             ( "OF05", 200, 23 ),
                             ( "OF06", 80, 15 ),
                             ( "OF07", 70, 16 ),
                             ( "OF08", 60, 18 );

#-----
#  AGREGACIÓN DE LOS DATOS DE LOS EMPLEADOS DE LA EMPRESA
#-----
INSERT INTO Empleado VALUES ( 1, "Fulgencio", "De la costa Feliz", "OF01", "Director General", NULL ),
                             ( 2, "Guillermino", "Feliz Amado", "OF02", "Director Ventas", 1 ),
                             ( 3, "Pancracio", "Gonzapota Alegre", "OF02", "Director Compras", 1 ),
                             ( 4, "Luzdivino", "Ponte Alegre", "OF03", "Asesoría", 1 ),
                             ( 5, "Bonifacio", "Viejobueno Llagaria", "OF04", "Oficinista", 2 ),
                             ( 6, "Laureano", "Nuevo perfume", "OF04", "Oficinista", 2 ),
                             ( 7, "Arnulfo", "Sin Chincurreta", "OF05", "Oficinista", 3 ),
                             ( 8, "Floripondio", "Gandul Piernabierta", "OF05", "Oficinista", 3 ),
                             ( 9, "Euquerio", "Marco Gol", NULL, "Oficinista", 3 ),
                             ( 10, "Tesifonte", "Susunaga Sorní", NULL, "Oficinista", 3 );
```

## Realización de consultas

### 1. Consultas básicas

Una consulta básica sirve para mostrar:

- Todos los elementos contenidos en una tabla.
- Todos los valores de algunos atributos de la tabla.
- Los valores distintos almacenados en la tabla.

```
SELECT * | {[DISTINCT] column|expression [alias],...}  
FROM      table;
```

- **SELECT** identifica las columnas que se van a mostrar.
- **FROM** identifica la tabla que contiene estas columnas.

```
#-----  
#  SELECT  *                -  Muestra todos los valores de los atributos de la tabla  
#  SELECT  atributo1, atributo2 -  Muestra valores de dos atributos de la tabla  
#  SELECT  DISTINCT  atributo1 -  Muestra los distintos valores del atributo atributo1  
#-----
```

*Si deseamos los datos de todos los trabajadores de la empresa.*

```
SELECT * FROM Empleado;
```

	idEmpleado	Nombre	Apellidos	Oficina	PuestoTrabajo	Jefe
▶	1	Fulgencio	De la costa Feliz	OF01	Director General	NULL
	2	Guillermino	Feliz Amado	OF02	Director Ventas	1
	3	Pancracio	Gonzapota Alegre	OF02	Director Compras	1
	4	Luzdivino	Ponte Alegre	OF03	Asesoría	1
	5	Bonifacio	Viejobueno Llagaria	OF04	Oficinista	2
	6	Laureano	Nuero perfume	OF04	Oficinista	2
	7	Arnulfo	Sin Chincurreta	OF05	Oficinista	3
	8	Floripondio	Gandul Piernabierta	OF05	Oficinista	3
	9	Euquerio	Marco Gol	NULL	Oficinista	3
	10	Tesifonte	Susunaga Sorní	NULL	Oficinista	3
*	NULL	NULL	NULL	NULL	NULL	NULL

## Realización de consultas

### 2. Consultas con filtros

Los filtros son condiciones que cualquier gestor de base de datos interpreta para realizar la selección de registros y mostrarlos como resultado de una consulta.

La palabra clave en SQL para realizar filtros es la cláusula WHERE.

```
SELECT [DISTINCT] expresión_select [,expresión_select]
[FROM tabla]
[WHERE filtro]
```

Un filtro es una expresión que indica la condición o condiciones que deben cumplir los registros para ser seleccionados.

#### 2.1. Operador de filtros (=, >=, <=, <>, !=, <, >).

El operador filtros (=, >=, <=, <>, !=, <, >) establece un requisito en el valor de un atributo que ha de ser satisfecho en la consulta.

Los operadores de filtros se pueden combinar a través de uniones con AND y OR.

Los operadores de filtros pueden negar su resultado con NOT.

```
#-----
#  = >= <= <>  -  Comprueba si el valor de un atributo cumple una restricción
#  != < >      especificada por un filtro
#  AND OR NOT   -  Une filtros
#-----
```

*Si deseamos los datos de los trabajadores de la empresa que trabajan en las oficinas 'OF01' y 'OF02' y cuyo Jefe sea el trabajador con el código 3.*

```
SELECT * FROM Empleado
WHERE PuestoTrabajo = "Oficinista" AND Jefe = 3;
```

	idEmpleado	Nombre	Apellidos	Oficina	PuestoTrabajo	Jefe
▶	7	Arnulfo	Sin Chincurreta	OF05	Oficinista	3
	8	Floripondio	Gandul Piernabierta	OF05	Oficinista	3
	9	Euquerio	Marco Gol	NULL	Oficinista	3
	10	Tesifonte	Susunaga Sorní	NULL	Oficinista	3
*	NULL	NULL	NULL	NULL	NULL	NULL

## Realización de consultas

### 2.2. Operador de pertenencia a conjuntos (IN).

El operador IN comprueba si los valores de un atributo se hallan incluidos en los especificados entre paréntesis.

```
SELECT [DISTINCT] expresión_select [,expresión_select]
[FROM tabla]
[WHERE nombre_columna IN (valor1, valor2, ...)]
```

```
#-----
#  IN          -   Comprueba si un atributo está incluido dentro de en unos
#                  valores determinados
#-----
```

*Si deseamos los datos de los trabajadores de la empresa que trabajan en las oficinas 'OF01' y 'OF02'.*

```
SELECT * FROM Empleado
WHERE Oficina IN ( 'OF01', 'OF02' );
```

	idEmpleado	Nombre	Apellidos	Oficina	PuestoTrabajo	Jefe
▶	1	Fulgencio	De la costa Feliz	OF01	Director General	NULL
	2	Guillermino	Feliz Amado	OF02	Director Ventas	1
	3	Pancracio	Gonzapota Alegre	OF02	Director Compras	1
*	NULL	NULL	NULL	NULL	NULL	NULL

### 2.3. Operador de rango (BETWEEN).

El operador BETWEEN comprueba si los valores de un atributo se hallan incluidos en un rango de valores especificados por valor\_máximo AND valor\_mínimo.

```
SELECT [DISTINCT] expresión_select [,expresión_select]
[FROM tabla]
[WHERE nombre_columna BETWEEN valor_min AND valor_max]
```

```
#-----
#  BETWEEN      -  Comprueba que los valores de un atributo se hallan incluido
#                  en el rango especificado entre <valor> AND <valor>
#-----
```

*Si deseamos los datos de los trabajadores de la empresa cuyos jefes tengan identificadores comprendidos entre 1 y 3.*

```
SELECT * FROM Empleado
WHERE Jefe BETWEEN 1 and 3;
```

	idEmpleado	Nombre	Apellidos	Oficina	PuestoTrabajo	Jefe
▶	2	Guillermino	Feliz Amado	OF02	Director Ventas	1
	3	Pancraccio	Gonzapota Alegre	OF02	Director Compras	1
	4	Luzdivino	Ponte Alegre	OF03	Asesoría	1
	5	Bonifacio	Viejobuena Llagaria	OF04	Oficinista	2
	6	Laureano	Nuero perfume	OF04	Oficinista	2
	7	Arnulfo	Sin Chincurreta	OF05	Oficinista	3
	8	Floripondio	Gandul Piernabierna	OF05	Oficinista	3
	9	Euquerio	Marco Gol	NULL	Oficinista	3
	10	Tesifonte	Susunaga Sorní	NULL	Oficinista	3
*	NULL	NULL	NULL	NULL	NULL	NULL



### 2.4. Operador con test de valor nulo (NULL).

El operador NULL comprueba si el valor de un atributo es nulo (no está asignado).

```
SELECT [DISTINCT] expresión_select [,expresión_select]
[FROM tabla]
[WHERE nombre_columna IS [NOT] NULL
```

#-----  
# IS NULL - Comprueba si el atributo especificado toma un valor nulo  
#-----

*Si deseamos los datos de los trabajadores de la empresa cuyos jefes tengan identificadores comprendidos entre 1 y 3.*

```
SELECT * FROM Empleado
WHERE Jefe IS NULL;
```

	idEmpleado	Nombre	Apellidos	Oficina	PuestoTrabajo	Jefe
▶	1	Fulgencio	De la costa Feliz	OF01	Director General	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL

### 2.5. Operador con test de patrón (LIKE).

El operador LIKE comprueba si el valor de un atributo tiene un formato específico bajo estas restricciones:

- % busca coincidencias con cualquier número de caracteres, incluso cero caracteres
- \_ busca coincidencias con un único carácter

```
SELECT [DISTINCT] expresión_select [,expresión_select]
[FROM tabla]
[WHERE nombre_columna [NOT] LIKE pattern;
```

```
#-----
#  LIKE          -   Busca atributos que cumplan un patrón muy breve
#                  _   puede ser cualquier carácter
#                  %   puede ser cualquier número de caracteres
#-----
```

*Si deseamos los datos de los trabajadores de la empresa cuyos apellidos contengan una 'n' y acaben en 'a'.*

```
SELECT * FROM Empleado
WHERE Apellidos LIKE '%n%a';
```

	idEmpleado	Nombre	Apellidos	Oficina	PuestoTrabajo	Jefe
▶	5	Bonifacio	Viejobueno Llagaria	OF04	Oficinista	2
	7	Arnulfo	Sin Chincurreta	OF05	Oficinista	3
	8	Floripondio	Gandul Piernabierta	OF05	Oficinista	3
*	NULL	NULL	NULL	NULL	NULL	NULL



### 2.6. Operador con test de número de registros (LIMIT).

El operador LIMIT establece un valor fijo a la hora de mostrar datos, definiéndose el número de tuplas y pudiéndose definir la posición desde la cual mostrar esos datos.

```
SELECT [DISTINCT] expresión_select [,expresión_select]
[FROM tabla]
[LIMIT [desplazamiento,] filas]
```

```
#-----
#  LIMIT      -  Fija el número de atributos mostrados en origen (inicio) y cantidad
#               origen  -> primer valor, el registro siguiente al valor especificado
#               cantidad -> segundo valor, o valor especificado si es único
#-----
```

*Si deseamos los datos de los tres trabajadores de la empresa siguientes al segundo.*

```
SELECT * FROM Empleado
LIMIT 2, 3;
```

	idEmpleado	Nombre	Apellidos	Oficina	PuestoTrabajo	Jefe
▶	3	Pancracio	Gonzapota Alegre	OF02	Director Compras	1
	4	Luzdivino	Ponte Alegre	OF03	Asesoría	1
	5	Bonifacio	Viejobueno Llagaria	OF04	Oficinista	2
*	NULL	NULL	NULL	NULL	NULL	NULL

### 2.7. Operador expresiones regulares de patrones (REGEXP).

El operador REGEXP (o RLIKE) comprueba si el valor de un atributo tiene un formato específico a partir de una expresión regular, que sirve para asegurar que se ajusta a un determinado patrón.

```
SELECT *
FROM persona
WHERE nombre REGEXP expresión_regular;
```

Los caracteres especiales de una expresión regular se pueden clasificar en:

- Comienzo y fin de cadena.
  - Comienzo de cadena → ^carácter
  - Fin de cadena → caracteres\$
- Coincidencia de caracteres.
  - Cualquier carácter → .
  - Secuencia de caracteres → carácter|carácter
- Número de repeticiones del carácter que precede.
  - No aparece o lo hace una sola vez → ?
  - Aparece una o más veces → +
  - No aparece, lo hace una o más veces → \*
  - Rango valores → {mín, max}
  - Rango de valores para varios caracteres → (caracteres){mín, max}
- Secuencia de caracteres.
  - Coincidencia con un carácter de un conjunto → [ ]
  - Coincidencia con la totalidad de los caracteres → ( )

#-----  
 # REGEXP - Busca atributos que cumplan un patrón específico  
 #-----

Si deseamos los datos de los trabajadores de la empresa cuyos nombres empiecen por 'a', 'b' o 'g' y acaben por 'o'.

```
SELECT * FROM Empleado
WHERE Nombre REGEXP '^[abg].*o$';
```

	idEmpleado	Nombre	Apellidos	Oficina	PuestoTrabajo	Jefe
▶	2	Guillermino	Feliz Amado	OF02	Director Ventas	1
	5	Bonifacio	Viejobueno Llagaria	OF04	Oficinista	2
	7	Arnulfo	Sin Chincurreta	OF05	Oficinista	3
*	NULL	NULL	NULL	NULL	NULL	NULL

## Realización de consultas

### 3. Consultas con ordenación (ORDER BY)

El operador ORDER BY ordena los datos mostrados en la consulta, de forma ascendente o descendente.

```
SELECT [DISTINCT] expresión_select [,expresión_select]
[FROM tabla]
[WHERE filtro]
[ORDER BY {nombre_columna | expresión | posición} [ASC | DESC] ], ...
```

```
#-----
# ORDER BY      - Clasifica los valores del (los) atributo(s) especificado(s)
#                de forma ascendente (ASC) o descendente (DESC)
#-----
```

Si deseamos los datos de los trabajadores de la empresa ordenados ~~ascendentemente~~ por apellido.

```
SELECT * FROM Empleado
ORDER BY Apellidos ASC;
```

	idEmpleado	Nombre	Apellidos	Oficina	PuestoTrabajo	Jefe
▶	1	Fulgencio	De la costa Feliz	OF01	Director General	NULL
	2	Guillermino	Feliz Amado	OF02	Director Ventas	1
	8	Floripondio	Gandul Piernabierta	OF05	Oficinista	3
	3	Pancracio	Gonzapota Alegre	OF02	Director Compras	1
	9	Euquerio	Marco Gol	NULL	Oficinista	3
	6	Laureano	Nuero perfume	OF04	Oficinista	2
	4	Luzdivino	Ponte Alegre	OF03	Asesoría	1
	7	Arnulfo	Sin Chincurreta	OF05	Oficinista	3
	10	Tesifonte	Susunaga Sorní	NULL	Oficinista	3
	5	Bonifacio	Viejobueno Llagaria	OF04	Oficinista	2
*	NULL	NULL	NULL	NULL	NULL	NULL

### 4. Consultas con resumen (GROUP BY)

El operador GROUP BY genera conjuntos de registros con los datos almacenados, generando información resumida a través de las funciones de columna, que operan con dichos conjuntos de registros y obtienen un resultado de cálculo.

```
SELECT [DISTINCT] expresión_select [,expresión_select] ...  
[FROM tabla]  
[WHERE filtro]  
[GROUP BY expresión [,expresión] ...
```

Las consultas de resumen permiten realizar:

- Funciones relacionadas con columnas.
- Agrupaciones de registros.
- Filtrado de resultados.

```
#-----  
# GROUP BY          -   Crea grupos bajo el mismo valor en el(los) atributo(s) especificado(s)  
#-----
```

*Si deseamos saber el número de empleados por oficina.*

```
SELECT Oficina, COUNT( * ) as 'Empleados'  
FROM Empleado  
GROUP BY Oficina;
```

	Oficina	Empleados
▶	NULL	2
	OF01	1
	OF02	2
	OF03	1
	OF04	2
	OF05	2

## Realización de consultas

### 5. Consultas multitabla

Una consulta entre varias tablas (o multitabla) es la que se produce al consultar información contenida en más de una tabla, aprovechándonos de los campos relacionados que haya entre las tablas para unirlos (join).

La única diferencia con las consultas sencillas se halla en la cláusula FROM, en la que se especifica las tablas que se van a usar y cómo se van a relacionar.

#### 5.1. Composición interna (INNER JOIN, NATURAL JOIN).

Una consulta multitabla de composición interna puede ser:

- NATURAL JOIN
- [INNER] JOIN

Si deseamos unir la información de las tablas Empleado y Oficina.

```
#-----  
# NATURAL JOIN - Unión de los datos de las tablas por campos comunes (con el mismo nombre)  
# sin duplicidad del campo común. En este ejemplo el campo 'Oficina'  
#-----
```

```
SELECT * FROM Empleado NATURAL JOIN Oficina;
```

	Oficina	idEmpleado	Nombre	Apellidos	PuestoTrabajo	Jefe	Dimensión	Mesas
►	OF01	1	Fulgencio	De la costa Feliz	Director General	NULL	100	5
	OF02	2	Guillermino	Feliz Amado	Director Ventas	1	120	30
	OF02	3	Pancracio	Gonzapota Alegre	Director Compras	1	120	30
	OF03	4	Luzdivino	Ponte Alegre	Asesoría	1	150	40
	OF04	5	Bonifacio	Viejobueno Llagaria	Oficinista	2	100	20
	OF04	6	Laureano	Nuero perfume	Oficinista	2	100	20
	OF05	7	Arnulfo	Sin Chincurreta	Oficinista	3	200	23
	OF05	8	Floripondio	Gandul Piernabierta	Oficinista	3	200	23

```
#-----  
# INNER JOIN - Unión de los datos de las tablas por campos especificados en ON  
# con duplicidad del campo común. En este ejemplo el campo 'Oficina'  
#-----
```

```
SELECT * FROM Empleado INNER JOIN Oficina ON Empleado.Oficina = Oficina.Oficina;
```

	idEmpleado	Nombre	Apellidos	Oficina	PuestoTrabajo	Jefe	Oficina	Dimensión	Mesas
►	1	Fulgencio	De la costa Feliz	OF01	Director General	NULL	OF01	100	5
	2	Guillermino	Feliz Amado	OF02	Director Ventas	1	OF02	120	30
	3	Pancracio	Gonzapota Alegre	OF02	Director Compras	1	OF02	120	30
	4	Luzdivino	Ponte Alegre	OF03	Asesoría	1	OF03	150	40
	5	Bonifacio	Viejobueno Llagaria	OF04	Oficinista	2	OF04	100	20
	6	Laureano	Nuero perfume	OF04	Oficinista	2	OF04	100	20
	7	Arnulfo	Sin Chincurreta	OF05	Oficinista	3	OF05	200	23
	8	Floripondio	Gandul Piernabierta	OF05	Oficinista	3	OF05	200	23



## Realización de consultas

### 5.2. Composición externa (RIGHT JOIN, LEFT JOIN).

Una consulta multitabla de composición externa puede ser:

- LEFT [OUTER] JOIN
- RIGHT [OUTER] JOIN
- FULL [OUTER] JOIN
- NATURAL LEFT [OUTER] JOIN
- NATURAL RIGHT [OUTER] JOIN

Si deseamos unir la información de las tablas Empleado y Oficina, asociando a cada Empleado con su Oficina.

```
#-----
#  LEFT OUTER JOIN      -  Unión de los datos de la tabla de la izquierda (LEFT) con
#                          los datos de la tabla de la derecha que cumplan la condición, SINO aparece NULL
#-----
```

```
SELECT *      FROM Empleado  LEFT OUTER JOIN      Oficina      ON Empleado.Oficina = Oficina.Oficina;
```

	idEmpleado	Nombre	Apellidos	Oficina	PuestoTrabajo	Jefe	Oficina	Dimensión	Mesas
▶	1	Fulgencio	De la costa Feliz	OF01	Director General	NULL	OF01	100	5
	2	Guillermino	Feliz Amado	OF02	Director Ventas	1	OF02	120	30
	3	Pancracio	Gonzapota Alegre	OF02	Director Compras	1	OF02	120	30
	4	Luzdivino	Ponte Alegre	OF03	Asesoría	1	OF03	150	40
	5	Bonifacio	Viejobueno Llagaria	OF04	Oficinista	2	OF04	100	20
	6	Laureano	Nuero perfume	OF04	Oficinista	2	OF04	100	20
	7	Arnulfo	Sin Chincurreta	OF05	Oficinista	3	OF05	200	23
	8	Floripondio	Gandul Piernabierta	OF05	Oficinista	3	OF05	200	23
	9	Euquerio	Marco Gol	NULL	Oficinista	3	NULL	NULL	NULL
	10	Tesifonte	Susunaga Sorní	NULL	Oficinista	3	NULL	NULL	NULL

```
#-----
#  RIGHT OUTER JOIN     -  Unión de los datos de la tabla de la derecha (RIGHT) con
#                          los datos de la tabla de la izquierda que cumplan la condición, SINO aparece NULL
#-----
```

```
SELECT *      FROM Empleado  RIGHT OUTER JOIN      Oficina      ON Empleado.Oficina = Oficina.Oficina;
```

	idEmpleado	Nombre	Apellidos	Oficina	PuestoTrabajo	Jefe	Oficina	Dimensión	Mesas
▶	1	Fulgencio	De la costa Feliz	OF01	Director General	NULL	OF01	100	5
	2	Guillermino	Feliz Amado	OF02	Director Ventas	1	OF02	120	30
	3	Pancracio	Gonzapota Alegre	OF02	Director Compras	1	OF02	120	30
	4	Luzdivino	Ponte Alegre	OF03	Asesoría	1	OF03	150	40
	5	Bonifacio	Viejobueno Llagaria	OF04	Oficinista	2	OF04	100	20
	6	Laureano	Nuero perfume	OF04	Oficinista	2	OF04	100	20
	7	Arnulfo	Sin Chincurreta	OF05	Oficinista	3	OF05	200	23
	8	Floripondio	Gandul Piernabierta	OF05	Oficinista	3	OF05	200	23
	NULL	NULL	NULL	NULL	NULL	NULL	OF06	80	15
	NULL	NULL	NULL	NULL	NULL	NULL	OF07	70	16
	NULL	NULL	NULL	NULL	NULL	NULL	OF08	60	18

## Realización de consultas

```
#-----
# FULL OUTER JOIN      -   Unión de los datos de la tabla de la derecha (RIGHT) con los datos de la tabla de
#                          la izquierda que cumplan la condición, SINO aparece NULL más la unión de los datos
#                          de la tabla de la izquierda (LEFT) con los datos de la tabla de la derecha que cumplan
#                          la condición, SINO aparece NULL
#-----
```

```
#SELECT * FROM Empleado FULL OUTER JOIN Oficina ON Empleado.Oficina = Oficina.Oficina;
SELECT * FROM Empleado RIGHT OUTER JOIN Oficina ON Empleado.Oficina = Oficina.Oficina
UNION
SELECT * FROM Empleado LEFT OUTER JOIN Oficina ON Empleado.Oficina = Oficina.Oficina;
```

	idEmpleado	Nombre	Apellidos	Oficina	PuestoTrabajo	Jefe	Oficina	Dimensión	Mesas
▶	1	Fulgencio	De la costa Feliz	OF01	Director General	NULL	OF01	100	5
	2	Guillermino	Feliz Amado	OF02	Director Ventas	1	OF02	120	30
	3	Pancracio	Gonzapota Alegre	OF02	Director Compras	1	OF02	120	30
	4	Luzdivino	Ponte Alegre	OF03	Asesoría	1	OF03	150	40
	5	Bonifacio	Viejobueno Llagaria	OF04	Oficinista	2	OF04	100	20
	6	Laureano	Nuero perfume	OF04	Oficinista	2	OF04	100	20
	7	Arnulfo	Sin Chincurreta	OF05	Oficinista	3	OF05	200	23
	8	Floripondio	Gandul Piernabiarta	OF05	Oficinista	3	OF05	200	23
	NULL	NULL	NULL	NULL	NULL	NULL	OF06	80	15
	NULL	NULL	NULL	NULL	NULL	NULL	OF07	70	16
	NULL	NULL	NULL	NULL	NULL	NULL	OF08	60	18
	9	Euquerio	Marco Gol	NULL	Oficinista	3	NULL	NULL	NULL
	10	Tesifonte	Susunaga Sorní	NULL	Oficinista	3	NULL	NULL	NULL



## Realización de consultas

#-----  
# NATURAL LEFT JOIN - Unión de los datos de la tabla de la izquierda (LEFT) con los datos de la tabla  
# de la derecha que cumplan la condición, SINO aparece NULL  
#-----

SELECT \* FROM Empleado NATURAL LEFT JOIN Oficina;

	Oficina	idEmpleado	Nombre	Apellidos	PuestoTrabajo	Jefe	Dimensión	Mesas
►	OF01	1	Fulgencio	De la costa Feliz	Director General	NULL	100	5
	OF02	2	Guillermino	Feliz Amado	Director Ventas	1	120	30
	OF02	3	Pancracio	Gonzapota Alegre	Director Compras	1	120	30
	OF03	4	Luzdivino	Ponte Alegre	Asesoría	1	150	40
	OF04	5	Bonifacio	Viejobueno Llagaria	Oficinista	2	100	20
	OF04	6	Laureano	Nuero perfume	Oficinista	2	100	20
	OF05	7	Arnulfo	Sin Chincurreta	Oficinista	3	200	23
	OF05	8	Floripondio	Gandul Piernabiarta	Oficinista	3	200	23
	NULL	9	Euquerio	Marco Gol	Oficinista	3	NULL	NULL
	NULL	10	Tesifonte	Susunaga Sorní	Oficinista	3	NULL	NULL

#-----  
# NATURAL RIGHT JOIN - Unión de los datos de la tabla de la derecha (RIGHT) con los datos de la tabla  
# de la izquierda que cumplan la condición, SINO aparece NULL  
#-----

SELECT \* FROM Empleado NATURAL RIGHT JOIN Oficina;

	Oficina	Dimensión	Mesas	idEmpleado	Nombre	Apellidos	PuestoTrabajo	Jefe
►	OF01	100	5	1	Fulgencio	De la costa Feliz	Director General	NULL
	OF02	120	30	2	Guillermino	Feliz Amado	Director Ventas	1
	OF02	120	30	3	Pancracio	Gonzapota Alegre	Director Compras	1
	OF03	150	40	4	Luzdivino	Ponte Alegre	Asesoría	1
	OF04	100	20	5	Bonifacio	Viejobueno Llagaria	Oficinista	2
	OF04	100	20	6	Laureano	Nuero perfume	Oficinista	2
	OF05	200	23	7	Arnulfo	Sin Chincurreta	Oficinista	3
	OF05	200	23	8	Floripondio	Gandul Piernabiarta	Oficinista	3
	OF06	80	15	NULL	NULL	NULL	NULL	NULL
	OF07	70	16	NULL	NULL	NULL	NULL	NULL
	OF08	60	18	NULL	NULL	NULL	NULL	NULL

### 6. Subconsultas

Una subconsulta es una consulta anidada dentro de otra consulta y se suelen utilizar para el filtrado de datos a través de los datos de otra consulta.

La subconsulta siempre va entre paréntesis.

Las subconsultas pueden aparecer en las cláusulas SELECT, FROM, WHERE o HAVING.

Las subconsultas pueden ser de los siguientes tipos:

- Subconsultas de comparación                      =, <=, >=, <>, !=, <, >
- Subconsultas de pertenencia a un conjunto    IN
- Subconsultas de cuantificación                ALL, ANY
- Subconsultas de existencia                    EXISTS

*Si deseamos la información de las Oficinas cuyo número de mesas sea menor o igual a la media de mesas por Oficina.*

```
#-----
# SUBCONSULTA DE COMPARACIÓN - Filtra el valor de un atributo según el tipo de
#                               comparación con el valor generado en la subconsulta
#-----
SELECT * FROM Oficina
      WHERE Mesas <= ( SELECT AVG( Mesas ) FROM Oficina );

#-----
# SUBCONSULTA DE PERTENENCIA A UN CONJUNTO - filtrar los valores de un atributo
#                                               según si están incluidos en los devueltos en la subconsulta
#-----
SELECT * FROM Oficina
      WHERE Mesas IN ( SELECT Mesas FROM Oficina WHERE Mesas <= ( SELECT AVG( Mesas ) FROM Oficina ) );

#-----
# SUBCONSULTA DE CUANTIFICACIÓN - filtra los valores de un atributo con respecto
#                                 a todos (ALL) o algunos (ANY) de los devueltos en la subconsulta
#-----
SELECT * FROM Oficina
      WHERE Mesas <= ANY ( SELECT AVG( Mesas ) FROM Oficina );

#-----
# SUBCONSULTA DE EXISTENCIA - filtra los valores de un atributo en función de si
#                             la subconsulta genera algún tipo de respuesta
#-----
SELECT * FROM Oficina Of1
      WHERE EXISTS ( SELECT * FROM Oficina Of2 WHERE Mesas <= ( SELECT AVG( Mesas ) FROM Oficina ) AND Of1.oficina = Of2.oficina );
```

	Oficina	Dimensión	Mesas
►	OF01	100	5
	OF04	100	20
	OF06	80	15
	OF07	70	16
	OF08	60	18
✱	NULL	NULL	NULL

## Realización de consultas

### 7. Consultas reflexivas

Una consulta reflexiva es una consulta que se realiza sobre los datos de una misma tabla, siendo necesario hacer un JOIN entre registros de la misma tabla.

```
#-----  
# CONSULTA REFLEXIVA - Consulta realizada dentro de una tabla a datos de esa  
# misma tabla y requiere dos nombres distintos de la  
# misma tabla  
#-----
```

*Si deseamos el nombre de los empleados de una empresa junto con el de sus jefes.*

```
SELECT CONCAT( Trabajador.Nombre, ' ', Trabajador.Apellidos ) AS 'Trabajador',  
        CONCAT( Jefe.Nombre, ' ', Jefe.Apellidos ) AS 'Jefe'  
FROM Empleado Trabajador INNER JOIN Empleado Jefe  
ON Trabajador.Jefe = Jefe.idEmpleado;
```

	Trabajador	Jefe
►	Guillermino Feliz Amado	Fulgencio De la costa Feliz
	Pancracio Gonzapota Alegre	Fulgencio De la costa Feliz
	Luzdivino Ponte Alegre	Fulgencio De la costa Feliz
	Bonifacio Viejobueno Llagaria	Guillermino Feliz Amado
	Laureano Nuero perfume	Guillermino Feliz Amado
	Arnulfo Sin Chincurreta	Pancracio Gonzapota Alegre
	Floripondio Gandul Piernabierta	Pancracio Gonzapota Alegre
	Euquerio Marco Gol	Pancracio Gonzapota Alegre
	Tesifonte Susunaga Sorní	Pancracio Gonzapota Alegre

### 8. Consultas derivadas

Una tabla derivada es un conjunto de datos que representan el origen de una tabla.

Las subconsultas con tablas derivadas (inline views) se utilizan dentro de las sentencias SELECT y dentro de la cláusula FROM en vez de emplear nombres de tablas.

Este tipo de subconsultas ayudan en la obtención de información de una forma más avanzada.

Este tipo de subconsultas NECESITAN SIEMPRE tener un nombre asociado.

```
#-----  
# CONSULTA DERIVADA      -   Consulta realizada a una tabla creada en el FROM de una  
#                           consulta y que requiere un nombre  
#-----
```

*Si deseamos el nombre de los empleados de una empresa cuya oficina sea 'OF03'.*

```
SELECT Tabla1.idEmpleado, Tabla1.Nombre  
FROM    ( SELECT idEmpleado, Nombre FROM    Empleado  
          WHERE Oficina = 'OF03' ) AS Tabla1;
```

	idEmpleado	Nombre
►	4	Luzdivino