

ÍNDICE

1. SUBCONSULTAS	3
1.1. ¿QUÉ ES UNA SUBCONSULTA?	3
ACTIVIDAD. PRIMERA SUBCONSULTA.....	3
1.2. VALOR DEVUELTO POR UNA SUBCONSULTA	4
ACTIVIDAD. SUBCONSULTAS.....	5
1.3. ACCESO A TABLAS DISTINTAS	5
ACTIVIDAD. SUBCONSULTAS A DOS TABLAS	5
1.4. FORMATO SIMPLIFICADO	6
ACTIVIDAD. SUBCONSULTA CON FORMATO SIMPLIFICADO	6
1.5. SUBCONSULTAS CORRELACIONADAS.....	7
ACTIVIDAD. SUBCONSULTA CORRELACIONADA.....	8
2. COMBINACIÓN DE TABLAS	9
ACTIVIDAD. COMBINACIÓN DE DOS TABLAS.....	9
ACTIVIDAD. COMBINACIÓN DE TABLAS	10
3. JOINS	11
3.1. INNER JOIN (COMBINACIÓN DE TABLAS).....	11
3.2. OUTER JOIN (COMBINACIÓN DE TABLAS EXTERNA)	12
ACTIVIDAD. JOINS.....	13
3.3. FORMATOS ESPECÍFICOS JOIN	14
4. AGRUPACIONES (GROUP BY-HAVING).....	15
4.1. ¿QUÉ ES UN AGRUPACIÓN?	15
ACTIVIDAD. GROUP BY	16
ACTIVIDAD. GROUP BY MÚLTIPLE.....	17
4.2. CONDICIÓN HAVING	18
ACTIVIDAD. GROUP BY - HAVING	18
4.3. CLÁUSULAS YA ESTUDIADAS APLICADAS A AGRUPACIONES	19
▪ ORDENACIÓN EN AGRUPACIONES	19
▪ CONDICIÓN WHERE EN AGRUPACIONES	19
▪ FUNCIONES DE AGREGADO EN AGRUPACIONES	20
ACTIVIDAD. AGREGADOS EN AGRUPACIONES.....	20
▪ SUBCONSULTAS EN AGRUPACIONES	20
▪ COMBINACIÓN DE TABLAS EN AGRUPACIONES	21
ACTIVIDAD. AGRUPACIONES CON COMBINACIONES DE TABLAS Y SUBCONSULTAS	21
▪ OUTER JOIN EN AGRUPACIONES	22
ACTIVIDAD. AGRUPACIONES CON OUTER JOIN.....	23
5. OPERADORES DE CONJUNTOS	23

5.1. OPERADOR UNION	24
5.2. OPERADOR INTERSECT.....	24
5.3. OPERADOR MINUS	25
ACTIVIDAD. OPERADORES DE CONJUNTOS	25
5.4. COMBINACIÓN DE OPERADORES DE CONJUNTO	26
ACTIVIDAD. OPERADORES DE CONJUNTOS COMBINADOS	26

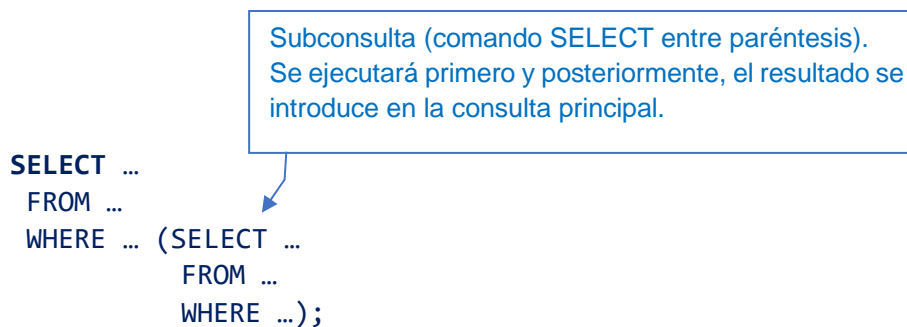
En este documento, a no ser que se indique lo contrario, se reflejan los comandos SQL referidos al Sistema Gestor de Base de Datos (SGBD) Oracle Database.

1. Subconsultas

1.1. ¿Qué es una subconsulta?

A menudo es necesario acceder a los datos de varias tablas en una misma consulta.

Una **subconsulta** es la realización de una consulta dentro de otra consulta (Un SELECT dentro de otro SELECT).



Ejemplo. Tabla EMPLE: visualizar los empleados con el mismo oficio que “GIL”:

Paso 1) Obtener el oficio de “GIL”:

```
select oficio from emple where apellido='GIL';
```

```
OFICIO
-----
ANALISTA
```

Paso 2) Obtener los empleados con el oficio “ANALISTA”:

```
select apellido from emple where oficio='ANALISTA';
```

Subconsulta) Obtención de la misma información en un solo paso:

```
select apellido from emple where oficio=(select oficio from emple
                                         where apellido='GIL');
```

Actividad. Primera subconsulta

A partir de la tabla EMPLE, obtén los empleados (apellido, oficio y fecha de alta) con salario mayor o igual que el empleado “FERNANDEZ”.

1.2. Valor devuelto por una subconsulta

Es fundamental tener en cuenta qué va a devolver una subconsulta para poder integrarla correctamente en una consulta.

Por ejemplo:



Esta consulta es incorrecta ya que no se puede comparar un capo (oficio) con una lista de valores

```
select apellido from emple where oficio = 'ANALISTA', 'DIRECTOR', 'DESARROLLADOR';
```

Correcto, ya que el operador IN permite comparar un campo (oficio) con una lista de valores

```
select apellido from emple where oficio in('ANALISTA','DIRECTOR','DESARROLLADOR');
```

De igual forma hay que prever qué valores puede devolver una subconsulta, para construir correctamente el comando:



Esta consulta es incorrecta ya que no se puede comparar un capo (oficio) con una lista de valores

```
select apellido from emple where oficio = (select oficio from emple  
                                         where dept_no=20);
```

Correcto, ya que el operador IN permite comparar un campo (oficio) con una lista de valores

```
select apellido from emple where oficio in (select oficio from emple  
                                           where dept_no=20);
```

De forma general:

- ▶ Se utilizan los operadores `>`, `<`, `>=`, `<=`, `=` cuando la subconsulta devuelva un valor único. Si el valor devuelto es de tipo texto podría emplearse el operador `LIKE`.
- ▶ Se utilizan los operadores `IN` cuando la subconsulta devuelva una lista de valores.

Actividad. Subconsultas

- a) En la tabla EMPLE, visualiza los apellidos de los empleados cuyo oficio es el mismo que el del empleado con número de empleado 7788 (GIL).
- b) Visualiza los empleados (todos sus datos) cuyo salario coincida con algún salario de los empleados del departamento 30.
- c) Visualiza los empleados (todos sus datos) que cobran por encima del salario medio de la empresa.

1.3. Acceso a tablas distintas

No es necesario que la consulta principal y la subconsulta hagan referencia a la misma tabla. Como siempre, la única condición es que se tenga en cuenta qué devuelve la subconsulta, para utilizar el operador adecuado (valor único, lista de valores...)

Ejemplo. Visualizar los datos de los empleados que trabajan en Madrid (tablas EMPLE y DEPART).

En este caso es necesario acceder a dos tablas, ya que la localización no se encuentra directamente en la tabla EMPLE. Hay que construir un único comando. A modo didáctico, se desglosa aquí en dos pasos:

Paso 1) Obtener el número de departamento correspondiente a "MADRID":

```
select dept_no from depart where loc='MADRID';
```

DEPT	NO
20	

Paso 2) Obtener los empleados que trabajan en ese departamento:

```
select * from emple where dept_no=20;
```

Subconsulta) Obtención de la misma información en un solo paso:

```
select * from emple where dept_no=(select dept_no from depart  
                                   where loc='MADRID');
```

Actividad. Subconsultas a dos tablas

- a) Obtén los datos de los empleados de la empresa que trabajan en Madrid o Barcelona. (tablas EMPLE y DEPART)
- b) Obtén los datos de los empleados que trabajan en Madrid o Barcelona y cobran más de 2000 euros.

1.4. Formato simplificado

Cuando se quiere hacer una comparación de varios campos devueltos por la misma subconsulta, se puede emplear un formato de comando reducido.

Ejemplo: se quiere obtener a los empleados que tienen el mismo oficio que “GIL” y que cobran el mismo salario. La consulta sería:

```
select * from emple where
    oficio =(select oficio from emple where apellido='GIL') and
    salario=(select salario from emple where apellido='GIL');
```

De forma reducida podría hacerse:

```
select * from emple where (salario, oficio) =
    (select salario, oficio from emple where apellido='GIL');
```

Nota: si la subconsulta devolviese más de una fila, la consulta daría error.

Este formato reducido sólo es válido si se trata de una subconsulta. No sería posible (devolverá error):

```
select * from emple where (salario, oficio) = (3000, 'ANALISTA');
```

Actividad. Subconsulta con formato simplificado

- a) Obtén los datos de los empleados de la empresa que tienen el mismo trabajo que el empleado con número 7566 (JIMENEZ)
- b) Obtén el apellido, oficio y salario de los empleados del mismo departamento que FERNANDEZ y que tengan su mismo salario.

1.5. Subconsultas correlacionadas

Una **subconsulta correlacionada** es aquella que hace referencia a una columna o varias de la consulta externa. Si en la subconsulta el nombre de las columnas coincide con el nombre de las columnas de la consulta externa, deben emplearse **alias de tabla**.

Recordatorio:

- ▶ Alias de columna: se da un nombre para visualizarlo como título de la columna en el listado.

```
select salario "suelto en euros" from emple;
```

- ▶ Alias de tabla: se da un nombre alternativo a una tabla. En todo el resto del comando, ese nombre representa la tabla.

```
select * from emple EM;
```

Antes de estudiar las consultas correlacionadas, es interesante entender cómo funciona una consulta.

Ejemplo: obtener los datos de los empleados que cobran el salario máximo de la empresa.

```
select * from emple where salario=(select max(salario) from emple);
```

El resultado es:

EMP_NO	APELLIDO	OFICIO	DIR	FECHA_ALT	SALARIO	COMISION	DEPT_NO
7839	REY	PRESIDENTE	(null)	17/11/19	4100	(null)	10

¿Cómo funciona esta consulta? El SGBD recorre la tabla EMPLE, comprobando en cada registro si se cumple la condición. **En este caso el resultado de la subconsulta siempre es el mismo** (el salario máximo: 4100). Esta es la tabla EMPLE:

Paso 1 ¿Cumple el registro 1 la condición
"salario = (select max(salario) from emple)" ?
Se resuelve la subconsulta y se compara: ¿4100 = 4100?
La respuesta es TRUE: este registro sí se mostrará

Paso 2 ¿Cumple el registro 2 la condición
"salario = (select max(salario) from emple)" ?
Se resuelve la subconsulta y se compara: ¿2900 = 4100?
La respuesta es FALSE: este registro no se mostrará

Paso 3 ¿Cumple el registro 3 la condición
"salario = (select max(salario) from emple)" ?
Se resuelve la subconsulta y se compara: ¿3000 = 4100?
La respuesta es FALSE: este registro no se mostrará

EMP_NO	APELLIDO	OFICIO	DIR	FECHA_ALT	SALARIO	COMISION	DEPT_NO
1	7839 REY	PRESIDENTE	(null)	17/11/19	4100	(null)	10
2	7566 JIMENEZ	DIRECTOR	7839	02/04/20	2900	(null)	20
3	7902 FERNANDEZ	ANALISTA	7566	03/12/20	3000	(null)	20
4	7369 SANCHEZ	DESARROLLADOR	7902	17/12/19	1040	(null)	20
5	7698 NEGRO	DIRECTOR	7839	01/05/20	3005	(null)	30
6	7499 ARROYO	VENDEDOR	7698	20/02/21	1500	390	30
7	7521 SALA	VENDEDOR	7698	22/02/21	1625	650	30
8	7654 MARTIN	VENDEDOR	7698	29/09/21	1600	1020	30
9	7782 CEREZO	DIRECTOR	7839	09/06/21	2885	(null)	10
10	7788 GIL	ANALISTA	7566	09/11/20	3000	(null)	20
11	7844 TOVAR	DESARROLLADOR	7698	08/09/21	1350	0	30
12	7876 ALONSO	DESARROLLADOR	7788	23/09/20	1430	(null)	20
13	7900 JIMENO	DESARROLLADOR	7698	03/12/21	1335	(null)	30
14	7934 MUÑOZ	DESARROLLADOR	7782	23/01/20	1690	(null)	10

etc (se repite la misma comprobación para todos los registros)

Ejemplo de subconsulta correlacionada: obtener los datos de los empleados que cobran el salario máximo de su propio departamento.

```
select * from emple E where salario=(select max(salario) from emple
                                     where dept_no=E.dept_no);
```

El alias de tabla sirve para acceder a un dato de la consulta externa dentro de la subconsulta.
El formato es tabla.columna.

El resultado es:

EMP_NO	APELLIDO	OFICIO	DIR	FECHA_ALT	SALARIO	COMISION	DEPT_NO
7839	REY	PRESIDENTE	(null)	17/11/19	4100	(null)	10
7902	FERNANDEZ	ANALISTA	7566	03/12/20	3000	(null)	20
7698	NEGRO	DIRECTOR	7839	01/05/20	3005	(null)	30
7788	GIL	ANALISTA	7566	09/11/20	3000	(null)	20

¿Cómo funciona esta consulta? El SGBD recorre la tabla EMPLE, comprobando en cada registro si se cumple la condición. **En este caso el resultado de la subconsulta no es siempre es el mismo, depende de un valor de la consulta externa** (valor máximo del salario de un cierto departamento). Esta es la tabla EMPLE:

Paso 1 ¿Cumple el registro 1 la condición?

Primero se resuelve la subconsulta teniendo en cuenta que E.dept_no es el departamento de la consulta externa (el registro 1):

¿"salario = (select max(salario) from emple where dept_no=E.dept_no)? Es decir:

¿4100=(select max(salario) from emple where dept_no=10?

Es decir: ¿4100=4100?

La respuesta es TRUE: este registro sí se mostrará

Paso 2 ¿Cumple el registro 2 la condición?

Primero se resuelve la subconsulta teniendo en cuenta que E.dept_no es el departamento de la consulta externa (el registro 2):

¿"salario = (select max(salario) from emple where dept_no=E.dept_no)? Es decir:

¿2900=(select max(salario) from emple where dept_no=20?

Es decir: ¿2900=3000?

La respuesta es FALSE: este registro no se mostrará

	EMP_NO	APELLIDO	OFICIO	DIR	FECHA_ALT	SALARIO	COMISION	DEPT_NO
1	7839	REY	PRESIDENTE	(null)	17/11/19	4100	(null)	10
2	7566	JIMENEZ	DIRECTOR	7839	02/04/20	2900	(null)	20
3	7902	FERNANDEZ	ANALISTA	7566	03/12/20	3000	(null)	20
4	7369	SANCHEZ	DESARROLLADOR	7902	17/12/19	1040	(null)	20
5	7698	NEGRO	DIRECTOR	7839	01/05/20	3005	(null)	30
6	7499	ARROYO	VENDEDOR	7698	20/02/21	1500	390	30
7	7521	SALA	VENDEDOR	7698	22/02/21	1625	650	30
8	7654	MARTIN	VENDEDOR	7698	29/09/21	1600	1020	30
9	7782	CEREZO	DIRECTOR	7839	09/06/21	2885	(null)	10
10	7788	GIL	ANALISTA	7566	09/11/20	3000	(null)	20
11	7844	TOVAR	DESARROLLADOR	7698	08/09/21	1350	0	30
12	7876	ALONSO	DESARROLLADOR	7788	23/09/20	1430	(null)	20
13	7900	JIMENO	DESARROLLADOR	7698	03/12/21	1335	(null)	30
14	7934	MUÑOZ	DESARROLLADOR	7782	23/01/20	1690	(null)	10

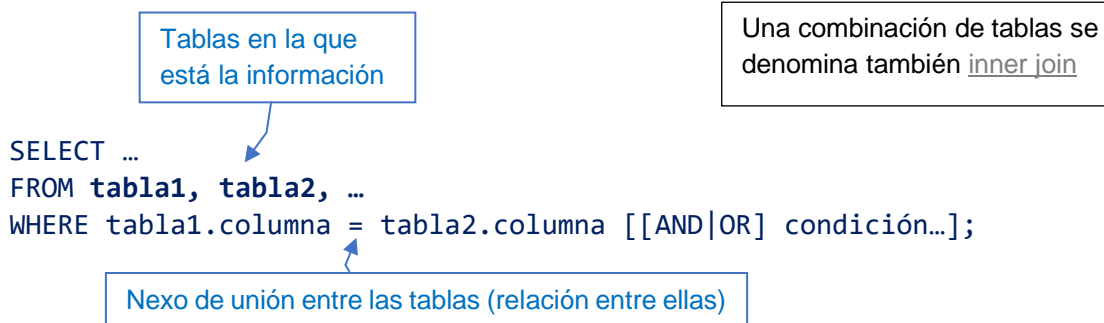
etc (se repite la misma comprobación para todos los registros)

Actividad. Subconsulta correlacionada

De la tabla EMPLE, obtén los datos de los empleados con mayor antigüedad en su oficio.
(Nota: la fecha más antigua puede calcularse con MIN(fecha alt)).

2. Combinación de tablas

Frecuentemente es necesario acceder a más de una tabla en una consulta, ya que la información que se busca está dividida en varias tablas. Para ello se deben indicar todas las tablas necesarias en la cláusula FROM:



Ejemplo: a partir de las tablas EMPLE y DEPART se quiere obtener un listado de todos los empleados de la empresa, visualizando para cada uno su apellido y localidad en la que trabajan.

```
select apellido, loc
  from emple, depart
 where emple.dept_no=depart.dept_no;
```

Si no se indicara la condición, aparecería un producto cartesiano de ambas tablas (todas las combinaciones posibles entre los registros), y la información mostrada no sería útil.

- ▶ Es posible unir tantas tablas como sea necesario, indicándolas en la cláusula FROM
- ▶ En la cláusula SELECT (campos a visualizar) pueden aparecer columnas de todas las tablas indicadas
- ▶ Si dos o más tablas tienen el mismo nombre de columna, debe identificarse de qué tabla tomar el campo con el formato: nombreTabla.nombreColumna
- ▶ **En la cláusula WHERE deben aparecer las condiciones que relacionan las tablas.** En caso contrario el resultado será un producto cartesiano de las tablas. Además pueden haber otras condiciones unidas con AND u OR.

Actividad. Combinación de dos tablas

- Tablas EMPLE y DEPART: para todos los empleados de la empresa, obtén los siguientes datos: número de empleado, apellido, oficio, nombre de su departamento y ciudad en la que trabaja.
- Para todos los empleados **analistas** de la empresa, obtén los siguientes datos: número de empleado, apellido, oficio, nombre de su departamento y ciudad en la que trabaja.

Cuando se combinan más de dos tablas el procedimiento es el mismo:

Ejemplo: carga las tablas ALUMNOS, ASIGNATURAS y NOTAS.

Se desea obtener un listado en el que aparezca el nombre completo del alumno, el nombre de la asignatura que curso y la nota que ha obtenido en ella.

```
select apenom, nombre, nota
  from alumnos, asignaturas, notas
 where alumnos.dni = notas.dni and notas.cod = asignaturas.cod;
```

Al ser tres tablas, hay dos condiciones que las relacionan.

APENOM	NOMBRE	NOTA
Alcalde García, Elena	Prog. Leng. Estr.	6
Alcalde García, Elena	Sist. Informáticos	5
Alcalde García, Elena	Análisis	6
Cerrato Vela, Luis	FOL	6
Díaz Fernández, María	FOL	8
Cerrato Vela, Luis	RET	8
Díaz Fernández, María	RET	7
Cerrato Vela, Luis	Entornos Gráficos	4
Díaz Fernández, María	Entornos Gráficos	8
Cerrato Vela, Luis	Aplic. Entornos 4ªGen	5
Díaz Fernández, María	Aplic. Entornos 4ªGen	9

Ejemplo: con las tablas ALUMNOS, ASIGNATURAS y NOTAS,

se desea obtener los nombres de los alumnos matriculados en la asignatura "FOL".

```
select apenom
  from alumnos, asignaturas, notas
 where alumnos.dni = notas.dni and notas.cod = asignaturas.cod and
        nombre='FOL';
```

APENOM
Cerrato Vela, Luis
Díaz Fernández, María

Esta consulta podría haberse realizado también a través de subconsultas, ya que los datos a mostrar solo están en una tabla (ALUMNOS):

```
select apenom
  from alumnos
 where alumnos.dni in (select dni from notas
                        where cod = (select cod from asignaturas
                                    where nombre='FOL'));
```

Actividad. Combinación de tablas

- Tablas ALUMNOS, ASIGNATURAS y NOTAS: visualiza los nombres de los alumnos que tengan una nota entre un 7 y un 8 en la asignatura "FOL".
- Visualiza los nombres de las asignaturas que no tengan suspensos. Si la consulta está bien realizada, no debería aparecer "Entornos Gráficos", ya que tiene un suspenso.

Ejercicios: Subconsultas y combinación de tablas. Ejercicios 6 a 10.

3. Joins

3.1. Inner join (combinación de tablas)

Observa y ejecuta el siguiente ejemplo de una combinación de tablas:

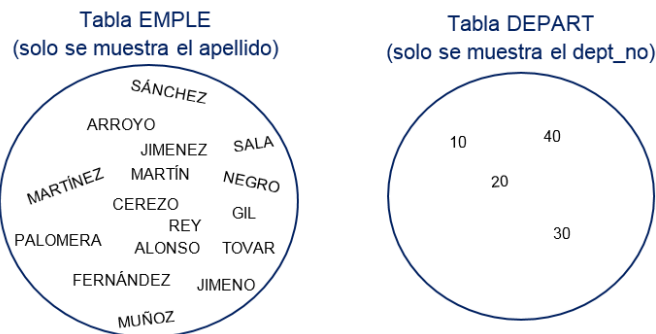
```
select emp_no, apellido, dnombre
  from emple, depart
 where emple.dept_no=depart.dept_no;
```

¿Qué ocurre con los empleados que no tienen departamento asignado?

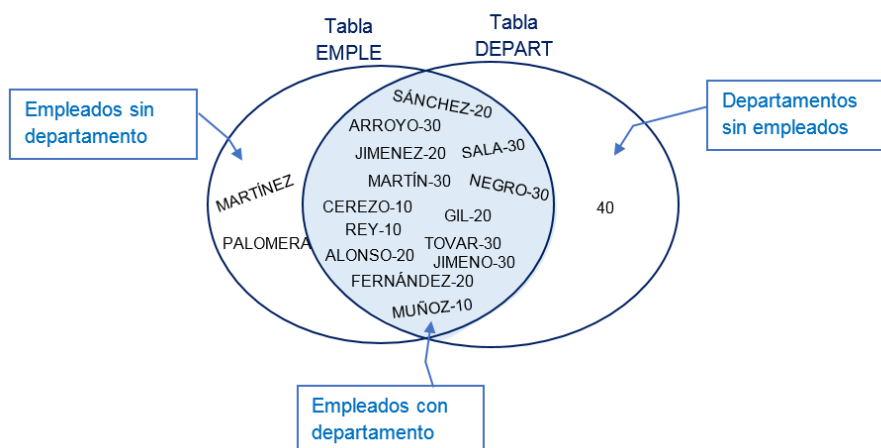
¿Qué ocurre con los departamentos que no tienen empleados?

APELLIDO	DNOMBRE
SANCHEZ	INVESTIGACION
ARROYO	VENTAS
SALA	VENTAS
JIMENEZ	INVESTIGACION
MARTIN	VENTAS
NEGRO	VENTAS
CEREZO	CONTABILIDAD
GIL	INVESTIGACION
REY	CONTABILIDAD
TOVAR	VENTAS
ALONSO	INVESTIGACION
JIMENO	VENTAS
FERNANDEZ	INVESTIGACION
MUÑOZ	CONTABILIDAD

Si representamos gráficamente las dos tablas:



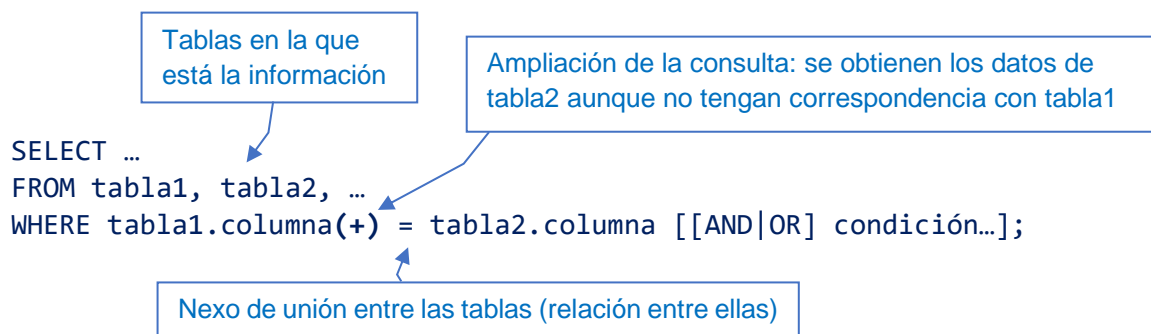
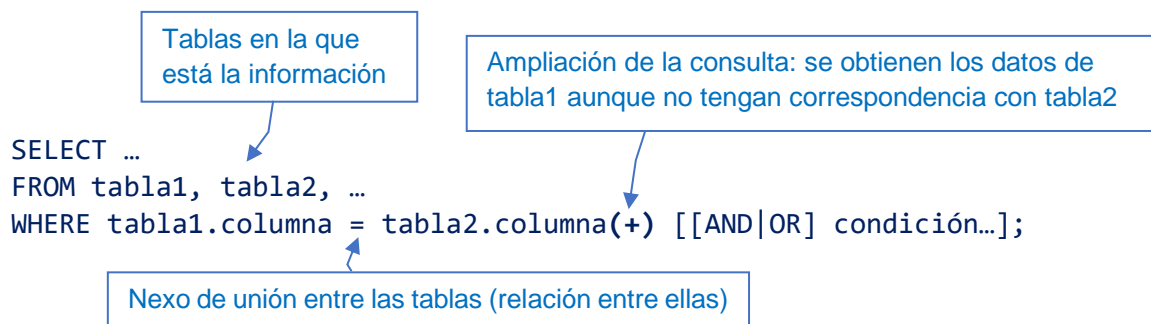
Podemos ver que hay una intersección:



Una **combinación de tablas** muestra los datos que están en la intersección de las dos tablas, es decir, los datos que están en las dos tablas. Esto es un **inner join**.

3.2. Outer join (combinación de tablas externa)

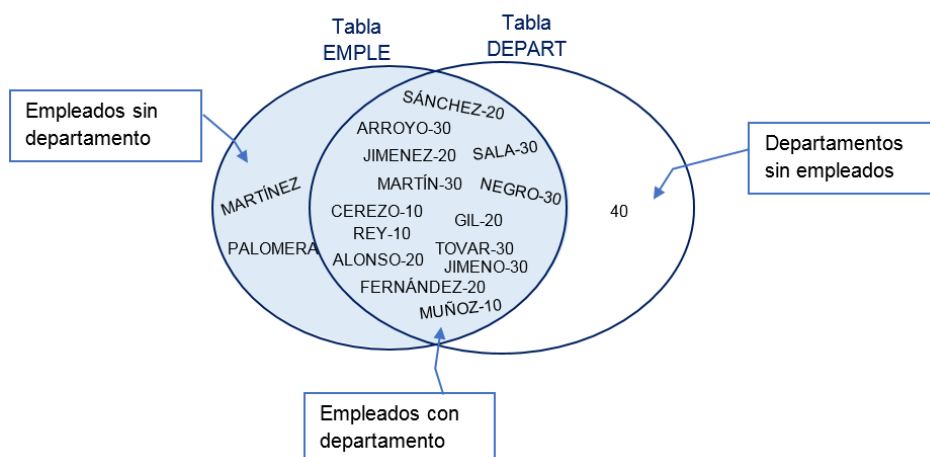
Con un **outer join** es posible seleccionar los datos que están en la tabla de la izquierda o derecha, incluida la intersección de las dos tablas. Formato:



Ejemplos: tablas EMPLE y DEPART.

```
select emp_no, apellido, dnombre
from emple, depart
where emple.dept_no=depart.dept_no(+);
```

Este (+) amplía la consulta a la combinación de tablas: se obtendrán los datos de EMPLE aunque no tengan correspondencia con DEPART

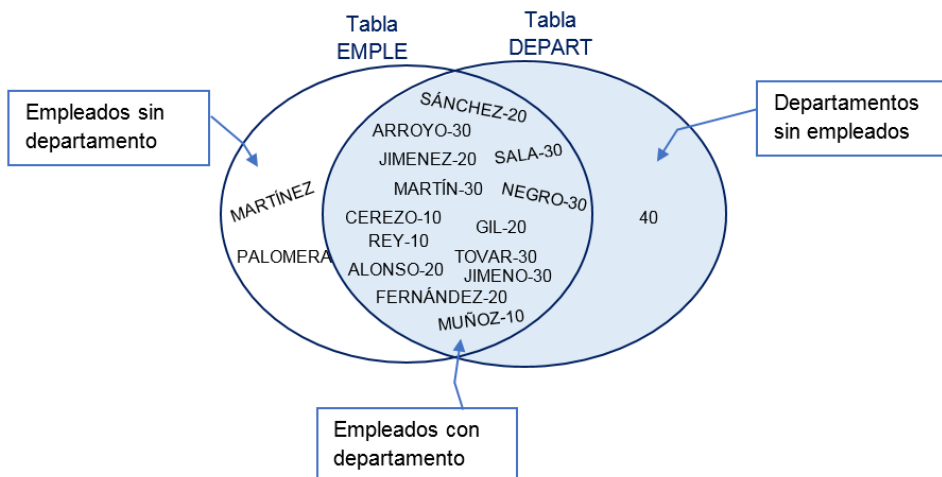


APELLIDO	DNOMBRE
CEREZO	CONTABILIDAD
REY	CONTABILIDAD
MUÑOZ	CONTABILIDAD
SANCHEZ	INVESTIGACION
JIMENEZ	INVESTIGACION
GIL	INVESTIGACION
ALONSO	INVESTIGACION
FERNANDEZ	INVESTIGACION
ARROYO	VENTAS
SALA	VENTAS
MARTIN	VENTAS
NEGRO	VENTAS
TOVAR	VENTAS
JIMENO	VENTAS
PALOMERA	(null)
MARTÍNEZ	(null)

```
select emp_no, apellido, dnombre
from emple, depart
where emple.dept_no(+) = depart.dept_no;
```

Este (+) amplía la consulta: se obtendrán los datos de DEPART aunque no tengan correspondencia con EMPLE

APELLIDO	DNOMBRE
SANCHEZ	INVESTIGACION
ARROYO	VENTAS
SALA	VENTAS
JIMENEZ	INVESTIGACION
MARTIN	VENTAS
NEGRO	VENTAS
CEREZO	CONTABILIDAD
GIL	INVESTIGACION
REY	CONTABILIDAD
TOVAR	VENTAS
ALONSO	INVESTIGACION
JIMENO	VENTAS
FERNANDEZ	INVESTIGACION
MUÑOZ	CONTABILIDAD
(null)	PRODUCCION



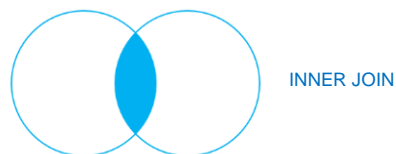
Actividad. Joins

- Tablas EMPLE y DEPART: visualiza, para cada empleado, su número, apellido y la localidad en la que trabaja. Deben aparecer todos los empleados de la empresa, aunque no trabajen en ninguna localidad.
- Tablas: DATOSPERSONALES y DP_TEL. En estas tablas se guardan los datos personales de unos usuarios, así como sus teléfonos. Lista los datos de los usuarios (login, idioma y teléfono). Deben aparecer todos los usuarios (aunque no tengan teléfono).
- Tablas: DATOSPERSONALES y DP_TEL. Visualiza, para todos los teléfonos, el idioma de su propietario. Si un teléfono no tiene propietario asociado, aparecerá en el listado con idioma "INDEFINIDO".

3.3. Formatos específicos Join

Existen formatos de comando específicos para los Joins, que se muestran en este apartado de forma ilustrativa:

```
select apellido, dnombre
  from emple
  inner join depart on emple.dept_no=depart.dept_no;
```



```
select apellido, dnombre
  from emple
  left join depart on emple.dept_no=depart.dept_no;
```



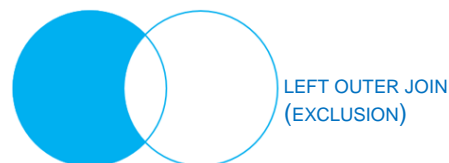
```
select apellido, dnombre
  from emple
  right join depart on emple.dept_no=depart.dept_no;
```



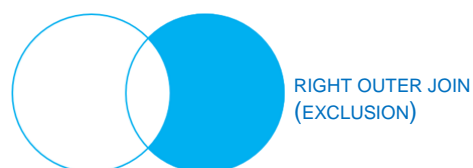
```
select apellido, dnombre
  from emple
  full join depart on emple.dept_no=depart.dept_no;
```



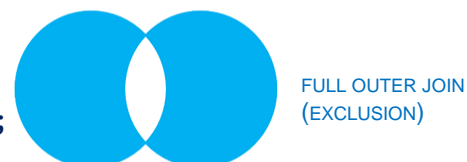
```
select apellido, dnombre
  from emple
  left join depart on emple.dept_no=depart.dept_no
  where depart.dept_no is null;
```



```
select apellido, dnombre
  from emple
  right join depart on emple.dept_no=depart.dept_no
  where emple.dept_no is null;
```



```
select apellido, dnombre
  from emple
  right join depart on emple.dept_no=depart.dept_no
  where emple.dept_no is null or depart.dept_no is null;
```



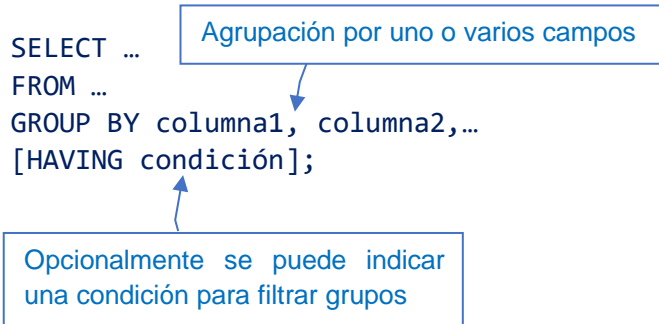
Ejercicios: Joins. Ejercicios 11 y 12.

4. Agrupaciones (GROUP BY-HAVING)

4.1. ¿Qué es un agrupación?

La cláusula **GROUP BY** permite agrupar registros en grupos. Es decir, crea grupos que tienen el mismo valor en un campo. A partir de ese momento, **los campos que se visualizan se calculan para cada grupo**, no para registros individuales.

La cláusula **HAVING** permite filtrar grupos. Es similar a **WHERE** (permite filtrar registros) pero aplicado a los grupos.



Los datos que se visualizan en una agrupación deben tener un valor único en el grupo. Solo pueden ser:

- ➡ Una constante
- ➡ Una función de grupo (SUM, COUNT, AVG...)
- ➡ Un campo dentro de la cláusula GROUP BY

Ejemplo: Tabla EMPLE

Select oficio from emple group by oficio;

¿Cómo funciona esta agrupación? Se toma la tabla EMPLE y se hacen grupos por el campo oficio. Es decir, cada valor distinto de oficio será un grupo.

EMP_NO	APELLIDO	OFICIO	DIR	FECHA_ALT	SALARIO	COMISION	DEPT_NO
7902 FERNANDEZ	ANALISTA	7566 03/12/20	3000	(null)	20		
7788 GIL	ANALISTA	7566 09/11/20	3000	(null)	20		
7844 TOVAR	DESARROLLADOR	7698 08/09/21	1350	0	30		
7876 ALONSO	DESARROLLADOR	7788 23/09/20	1430	(null)	20		
7900 JIMENO	DESARROLLADOR	7698 03/12/21	1335	(null)	30		
7369 SANCHEZ	DESARROLLADOR	7902 17/12/19	1040	(null)	20		
7934 MUÑOZ	DESARROLLADOR	7782 23/01/20	1690	(null)	10		
7782 CEREZO	DIRECTOR	7839 09/06/21	2885	(null)	10		
7698 NEGRO	DIRECTOR	7839 01/05/20	3005	(null)	30		
7566 JIMENEZ	DIRECTOR	7839 02/04/20	2900	(null)	20		
8000 MARTÍNEZ	EMPLEADOR	7902 27/01/15	1300	(null)	(null)		
7839 REY	PRESIDENTE	(null) 17/11/19	4100	(null)	10		
7654 MARTIN	VENDEDOR	7698 29/09/21	1600	1020	30		
7521 SALA	VENDEDOR	7698 22/02/21	1625	650	30		
7499 ARROYO	VENDEDOR	7698 20/02/21	1500	390	30		
8001 PALOMERA	VENDEDOR	7902 08/05/17	1505	(null)	(null)		

GRUPOS

¿Qué se puede visualizar? Para cada grupo aparecerá una única línea como resultado, por lo que solo es posible visualizar datos que tengan un valor único para el grupo.

Por ejemplo, se podría visualizar el oficio o el número de empleados, pero no el apellido (no es único en todo el grupo).

select oficio, count(emp_no) from emple group by oficio;

Los datos a visualizar se calculan para cada grupo

OFICIO	COUNT(EMP_NO)
PRESIDENTE	1
EMPLEADOR	1
DIRECTOR	3
ANALISTA	2
DESARROLLADOR	5
VENDEDOR	4

select oficio, count(emp_no), apellido from emple group by oficio;



Esta consulta es incorrecta ya que no se puede visualizar el apellido en esta agrupación (no tiene un valor único)

ORA-00979: no es una expresión GROUP BY
00979. 00000 - "not a GROUP BY expression"

Actividad. Group by

- Tabla EMPLE: obtén el salario medio de la empresa.
- Obtén el salario medio del departamento 10.
- Obtén el salario medio de cada departamento.

Es posible realizar **agrupaciones por varios campos**. En ese caso, se harán grupos distintos para cada combinación distinta de los valores de los campos de agrupación.

Ejemplo: Tabla EMPLE. Visualizar el número de empleados que realizan **cada oficio en cada departamento**:

```
select dept_no, oficio, count(*) from emple group by dept_no, oficio;
```

EMP_NO	APELLIDO	OFICIO	DIR	FECHA_ALT	SALARIO	COMISION	DEPT_NO
7934	MUNOZ	DESARROLLADOR	7782	23/01/20	1690	(null)	10
7782	CEREZO	DIRECTOR	7839	09/06/21	2885	(null)	10
7839	REY	PRESIDENTE	(null)	17/11/19	4100	(null)	10
7788	GIL	ANALISTA	7566	09/11/20	3000	(null)	20
7902	FERNANDEZ	ANALISTA	7566	03/12/20	3000	(null)	20
7369	SANCHEZ	DESARROLLADOR	7902	17/12/19	1040	(null)	20
7876	ALONSO	DESARROLLADOR	7788	23/09/20	1430	(null)	20
7566	JIMENEZ	DIRECTOR	7839	02/04/20	2900	(null)	20
7844	TOVAR	DESARROLLADOR	7698	08/09/21	1350	0	30
7900	JIMENO	DESARROLLADOR	7698	03/12/21	1335	(null)	30
7698	NEGRO	DIRECTOR	7839	01/05/20	3005	(null)	30
7654	MARTIN	VENDEDOR	7698	29/09/21	1600	1020	30
7521	SALA	VENDEDOR	7698	22/02/21	1625	650	30
7499	ARROYO	VENDEDOR	7698	20/02/21	1500	390	30
8000	MARTINEZ	EMPLEADOR	7902	27/01/15	1300	(null)	(null)
8001	PALOMERA	VENDEDOR	7902	08/05/17	1505	(null)	(null)

EN TOTAL HAY 11 GRUPOS

Resultado:

DEPT_NO	OFICIO	COUNT(*)
30	DIRECTOR	1
30	VENDEDOR	3
30	DESARROLLADOR	2
(null)	VENDEDOR	1
10	DIRECTOR	1
10	PRESIDENTE	1
20	DESARROLLADOR	2
10	DESARROLLADOR	1
20	ANALISTA	2
(null)	EMPLEADOR	1
20	DIRECTOR	1

Actividad. Group by múltiple

- Tabla EJEMPLAR_ARBOL: muestra, para cada provincia, cuántos ejemplares de árbol hay de cada especie.
- Muestra, para cada provincia, cuándo se plantó el último árbol de cada especie.

4.2. Condición Having

La cláusula **HAVING** permite filtrar grupos: solo se mostrarán los datos de los grupos que cumplan las condiciones establecidas en Having.

Ejemplo: Tabla EMPLE. Visualizar el número de empleados que hay en cada departamento.

```
select dept_no, count(*) from emple group by dept_no;
```

EMP_NO	APELLIDO	OFICIO	DIR	FECHA_ALT	SALARIO	COMISION	DEPT_NO
7782	CEREZO	DIRECTOR	7839	09/06/21	2885	(null)	10
7839	REY	PRESIDENTE	(null)	17/11/19	4100	(null)	10
7934	MUÑOZ	DESARROLLADOR	7782	23/01/20	1690	(null)	10
7788	GIL	ANALISTA	7566	09/11/20	3000	(null)	20
7566	JIMENEZ	DIRECTOR	7839	02/04/20	2900	(null)	20
7902	FERNANDEZ	ANALISTA	7566	03/12/20	3000	(null)	20
7369	SANCHEZ	DESARROLLADOR	7902	17/12/19	1040	(null)	20
7876	ALONSO	DESARROLLADOR	7788	23/09/20	1430	(null)	20
7654	MARTIN	VENDEDOR	7698	29/09/21	1600	1020	30
7521	SALA	VENDEDOR	7698	22/02/21	1625	650	30
7698	NEGRO	DIRECTOR	7839	01/05/20	3005	(null)	30
7499	ARROYO	VENDEDOR	7698	20/02/21	1500	390	30
7900	JIMENO	DESARROLLADOR	7698	03/12/21	1335	(null)	30
7844	TOVAR	DESARROLLADOR	7698	08/09/21	1350	0	30
8000	MARTÍNEZ	EMPLEADOR	7902	27/01/15	1300	(null)	(null)
8001	PALOMERA	VENDEDOR	7902	08/05/17	1505	(null)	(null)

GRUPOS

DEPT_NO	COUNT(*)
30	6
(null)	2
10	3
20	5

Visualizar los departamentos que tienen más de 4 empleados:

```
select dept_no, count(*) from emple group by dept_no having count(*)>4;
```

DEPT_NO	COUNT(*)
30	6
20	5

Como es lógico, en las condiciones impuestas en HAVING se deben utilizar expresiones que tengan un valor único para el grupo. No sería posible por ejemplo: `having salario > 2000`, ya que hay distintos salarios en un mismo grupo.

Actividad. Group by - Having

- Tabla EMPLE: obtén los departamentos en los que el salario medio es mayor de 2500.
- Tabla EMPLE: obtén los departamentos en los que el salario medio es mayor o igual que la media de todos los salarios.

4.3. Cláusulas ya estudiadas aplicadas a agrupaciones

▪ Ordenación en agrupaciones

Cuando se emplean agrupaciones, la cláusula ORDER BY, al igual que HAVING, debe hacer referencia a valores que sean únicos en el grupo.

Ejemplo: Tabla EMPLE. Visualizar el número de empleados que hay en cada departamento, ordenados de departamentos más grandes a más pequeños.

```
select dept_no, count(*) "NUM.EMPLEADOS"
  from emple
  group by dept_no
  order by count(*) desc;
```

DEPT_NO	NUM.EMPLEADOS
30	6
20	5
10	3
(null)	2

▪ Condición WHERE en agrupaciones

En una agrupación puede utilizarse la cláusula WHERE. Hay que tener en cuenta que primero se ejecuta este WHERE, haciendo una selección de filas. Una vez se han filtrado las filas se procede a crear los grupos.

Ejemplo: Tabla EMPLE. Visualizar el número de empleados con oficio “desarrollador” que hay en cada departamento:

```
select dept_no,count(*) from emple
  where oficio='DESARROLLADOR'
  group by dept_no;
```

¿Cómo se ejecuta esta consulta?

Paso 1) Ejecutar la cláusula WHERE filtrando todos los empleados por su oficio:

EMP_NO	APELLIDO	OFICIO	DIR	FECHA_ALT	SALARIO	COMISION	DEPT_NO
7934	MUÑOZ	DESARROLLADOR	7782	23/01/20	1690	(null)	10
7369	SANCHEZ	DESARROLLADOR	7902	17/12/19	1040	(null)	20
7876	ALONSO	DESARROLLADOR	7788	23/09/20	1430	(null)	20
7900	JIMENO	DESARROLLADOR	7698	03/12/21	1335	(null)	30
7844	TOVAR	DESARROLLADOR	7698	08/09/21	1350	0	30

Paso 2) Realiza la agrupación:

EMP_NO	APELLIDO	OFICIO	DIR	FECHA_ALT	SALARIO	COMISION	DEPT_NO
7934	MUÑOZ	DESARROLLADOR	7782	23/01/20	1690	(null)	10
7369	SANCHEZ	DESARROLLADOR	7902	17/12/19	1040	(null)	20
7876	ALONSO	DESARROLLADOR	7788	23/09/20	1430	(null)	20
7900	JIMENO	DESARROLLADOR	7698	03/12/21	1335	(null)	30
7844	TOVAR	DESARROLLADOR	7698	08/09/21	1350	0	30

GRUPOS

Resultado:

DEPT_NO	COUNT(*)
10	1
20	2
30	2

■ Funciones de agregado en agrupaciones

Como ya hemos visto, es posible utilizar funciones de agregado (SUM, MIN, MAX, AVG, COUNT) en agrupaciones, y el valor se calculará para cada grupo. Observa los siguientes ejemplos:

```
select salario from emple where dept_no=20;
```

SALARIO
2900
3000
1040
3000
1430

Se selecciona el mayor de la lista
(Mayor salario)

```
select max(salario) from emple where dept_no=20;
```

MAX(SALARIO)
3000

```
select count(*) from emple group by dept_no;
```

COUNT(*)
6
2
3
5

Se selecciona el mayor de la lista

```
select max(count(*)) from emple group by dept_no;
```

MAX(COUNT(*))
6

(Mayor número de empleados en un dep.)

Actividad. Agregados en agrupaciones

- Tablas EMPLE y DEPART: obtén la suma de salarios, el salario máximo y el salario mínimo por cada departamento.
- Visualiza el número de empleados del departamento más pequeño (con menos empleados).

■ Subconsultas en agrupaciones

Ejemplo: Visualiza el número de departamento y el número de empleados del departamento con más empleados:

```
select dept_no, count(*) from emple
group by dept_no
having count(*)=(select max(count(*)) from emple group by dept_no );
```



El resultado es un número:
(Mayor número de empleados en un dep.)

MAX(COUNT(*))
6

Equivale a:

```
select dept_no, count(*) from emple
group by dept_no
having count(*)=( 6 );
```

▪ Combinación de tablas en agrupaciones

Ejemplo: Para cada departamento visualiza el número de departamento, el nombre de departamento y el número de empleados que tiene.

En este caso, es necesario acceder de alguna forma a las dos tablas, ya que la información solicitada se encuentra en ambas.

```
select emple.dept_no, dnombre, count(*)
  from emple, depart
 where emple.dept_no=depart.dept_no
 group by emple.dept_no, dnombre;
```

DEPT_NO	DNOMBRE	COUNT(*)
20	INVESTIGACION	5
10	CONTABILIDAD	3
30	VENTAS	6



Es necesario agrupar por el nombre de departamento para poder visualizarlo (ver [reglas de las agrupaciones](#)). Esto no cambia los grupos creados al agrupar por dept_no.

Es equivalente agrupar por emple.dept_no que por depart.dept_no:

```
select depart.dept_no, dnombre, count(*)
  from emple, depart
 where emple.dept_no=depart.dept_no
 group by depart.dept_no, dnombre;
```

DEPT_NO	DNOMBRE	COUNT(*)
20	INVESTIGACION	5
10	CONTABILIDAD	3
30	VENTAS	6

Actividad. Agrupaciones con combinaciones de tablas y subconsultas

- Tablas EMPLE y DEPART: Visualiza el mayor salario medio por departamento que haya en la empresa.
- Visualiza el número de departamento con el mayor salario medio de toda la empresa.
- Visualiza el número de departamento y el número de empleados que tiene el departamento con más empleados.
- Visualiza el número de departamento, el nombre del departamento y el número de empleados del departamento con más empleados.
- Obtén los nombres de departamentos con más de 4 personas trabajando en ellos

▪ Outer Join en agrupaciones

Ejemplo: Para cada departamento visualiza el número de departamento, el nombre de departamento y el número de empleados que tiene. Los departamentos sin empleados aparecerán en el listado con 0 empleados.

¿Por qué se cuenta con count(emp_no)?

Una combinación de tablas mezcla filas de las dos tablas, relacionando empleados y departamentos. Se creará una línea para cada empleado con su departamento. Al ser una consulta extendida (outer join), también aparecerán los departamentos sin empleados: aparecerá una línea con el departamento.

Si se utiliza count(*) se contará también esa línea. Para forzar a que la cuenta se produzca por empleado, debe emplearse count(emp_no).

```
select depart.dept_no, dnombre, count(emp_no)
from emple, depart
where emple.dept_no(+) = depart.dept_no
group by depart.dept_no, dnombre;
```

¿Por qué se agrupa por depart.dept_no?

Si se hacen grupos por emple.dept_no se toma la tabla EMPLE, en la que no están los departamentos con 0 empleados:

EMP_NO	APELLIDO	OFICIO	DIR	FECHA_ALT	SALARIO	COMISION	DEPT_NO
7782	CEREZO	DIRECTOR	7839	09/06/21	2885	(null)	10
7839	REY	PRESIDENTE	(null)	17/11/19	4100	(null)	10
7934	MUÑOZ	DESARROLLADOR	7782	23/01/20	1690	(null)	10
7788	GIL	ANALISTA	7566	09/11/20	3000	(null)	20
7566	JIMENEZ	DIRECTOR	7839	02/04/20	2900	(null)	20
7902	FERNANDEZ	ANALISTA	7566	03/12/20	3000	(null)	20
7369	SANCHEZ	DESARROLLADOR	7902	17/12/19	1040	(null)	20
7876	ALONSO	DESARROLLADOR	7788	23/09/20	1430	(null)	20
7654	MARTIN	VENDEDOR	7698	29/09/21	1600	1020	30
7521	SALA	VENDEDOR	7698	22/02/21	1625	650	30
7698	NEGRO	DIRECTOR	7839	01/05/20	3005	(null)	30
7499	ARROYO	VENDEDOR	7698	20/02/21	1500	390	30
7900	JIMENO	DESARROLLADOR	7698	03/12/21	1335	(null)	30
7844	TOVAR	DESARROLLADOR	7698	08/09/21	1350	0	30
8000	MARTÍNEZ	EMPLEADOR	7902	27/01/15	1300	(null)	(null)
8001	PALOMERA	VENDEDOR	7902	08/05/17	1505	(null)	(null)

GRUPOS

En cambio, si se hacen grupos por depart.dept_no se toma la tabla DEPART, en la que están todos los departamentos de la empresa, tengan o no empleados:

DEPT_NO	DNOMBRE	LOC
10	CONTABILIDAD	SEVILLA
20	INVESTIGACION	MADRID
30	VENTAS	BARCELONA
40	PRODUCCION	BILBAO

GRUPOS

Actividad. Agrupaciones con outer join

- a) Tablas FACTURA y CLIENT: Visualiza cuántas facturas tiene cada CIF de cliente. Si un cliente no tiene facturas, aparecerá en el listado con 0 facturas.

Ejercicios: Agrupaciones. Ejercicios 13 y 14.

5. Operadores de conjuntos

Los operadores relacionales **UNION**, **INTERSECT** y **MINUS** son operadores de conjuntos, entendiendo como conjunto una serie de filas resultantes de una sentencia **SELECT**.

Mediante estos operadores es **posible combinar los resultados de distintos SELECT**. Al contrario que en una combinación de tablas, en este caso las tablas implicadas no tienen que estar relacionadas.



Ejemplo. Visualmente se pueden representar estos conjuntos de esta forma:

TABLA1

NOMBRE	APELLIDO	EDAD
Nerea	Pérez	20
Elena	García	14
Julia	García	11

*select nombre, edad
from tabla1;*

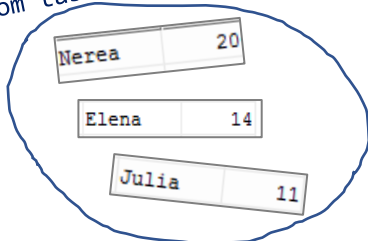
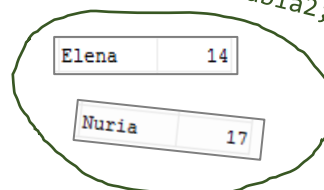


TABLA2

NOMBRE	EDAD	IDIOMA
Elena	14	ESP
Nuria	17	ESP

*select nombre, edad
from tabla2;*



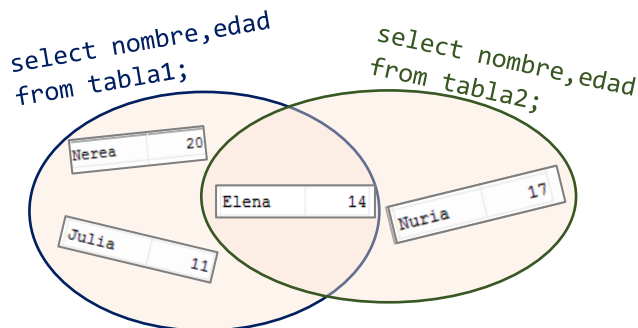
Reglas de los operadores de conjuntos:

- Los resultados de los **SELECT** deben tener el mismo número de columnas.
- No es necesario que las columnas se llamen igual, pero sí deben contener el mismo tipo de dato

5.1. Operador UNION

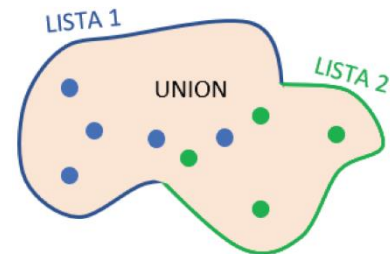
El operador **UNION** combina el resultado de dos consultas. Las filas duplicadas aparecen solo una vez, a no ser que se emplee el operador **UNION ALL**.

Ejemplo:



```
select nombre, edad from tabla1
union
select nombre, edad from tabla2;
```

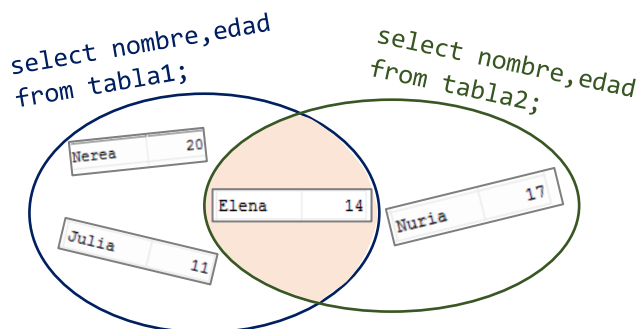
NOMBRE	EDAD
Elena	14
Julia	11
Nerea	20
Nuria	17



5.2. Operador INTERSECT

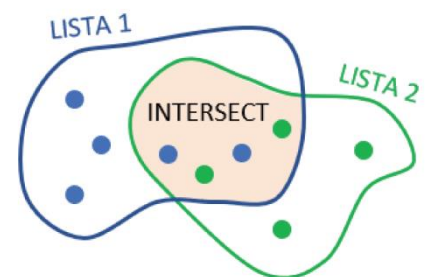
El operador **INTERSECT** devuelve las filas que son iguales en ambas consultas. Las filas duplicadas aparecen solo una vez.

Ejemplo:



```
select nombre, edad from tabla1
intersect
select nombre, edad from tabla2;
```

NOMBRE	EDAD
Elena	14

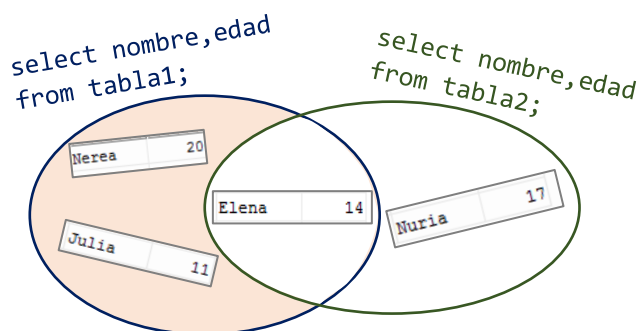


5.3. Operador MINUS

El operador **MINUS** devuelve las filas que están en la primera consulta pero no en la segunda. Las filas duplicadas en la primera consulta aparecen solo una vez.



Ejemplo:



```
select nombre, edad from tabla1
minus
select nombre, edad from tabla2;
```

NOMBRE	EDAD
Julia	11
Nerea	20

Actividad. Operadores de conjuntos

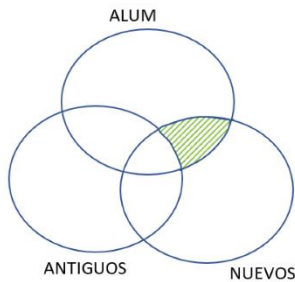
Tablas ALUM, NUEVOS y ANTIGUOS. Estas tablas guardan información sobre alumnos de un centro: los actuales (ALUM), los que han reservado plaza (NUEVOS) y los que terminaron sus estudios (ANTIGUOS). Puede ser que un mismo alumno esté estudiando actualmente y haya terminado otro estudio en el mismo centro (está en las tablas ALUM y ANTIGUOS simultáneamente)

- Visualiza los nombres (solo nombres) de los alumnos actuales y futuros, en un único listado. No deben aparecer nombres repetidos
- Visualiza los nombres (solo nombres) de los alumnos actuales y futuros, en un único listado. Si algún nombre está más de una vez, aparecerá varias veces.
- Visualiza los nombres (solo nombres) de alumnos antiguos que están estudiando actualmente en el centro. Escribe dos formas de realizar esta consulta: con operador de conjunto y sin él.
- Visualiza los alumnos actuales (nombre, edad y localidad) que nunca estudiaron antes en el centro (no están en ANTIGUOS). Se considera que es un mismo alumno si coincide su nombre, edad y localidad. Escribe dos formas de realizar esta consulta: con operador de conjunto y sin él.

5.4. Combinación de operadores de conjunto

Es posible combinar los operadores de conjunto para obtener distintos resultados. Los conjuntos se evalúan de izquierda a derecha, y se pueden usar paréntesis para forzar precedencia.

Ejemplo:

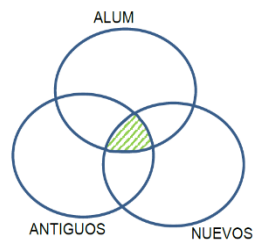


Seleccionar los alumnos actuales (ALUM) que estén también NUEVOS pero no estén en ANTIGUOS:

```
select nombre from alum
intersect
select nombre nuevos
minus
select nombre from antiguos;
```

Actividad. Operadores de conjuntos combinados

Visualiza los nombres (solo nombres) de alumnos que estudian en el centro, ya estudiaron otros estudios, y han reservado alguna plaza nueva.



Ejercicios: Operadores de conjuntos. Ejercicio 15.

Ejercicios complementarios