



#>/<>

HACK A BOSS

<CODE YOUR TALENT>

CONSULTAS

#>/<>

HACK A BOSS

<CODE YOUR TALENT>

*"La tecnología, bien utilizada, es uno
de los mayores catalizadores sociales
que han existido nunca"*

2020 EDITION



ÍNDICE

01.

Intro

- Definición
- Modelos
- BBDD relacionales

02.

Diseño

- Proceso
- Modelo Entidad-Relación
- Diagramas ER

03.

Implementación

- Creación y modificación
- Consultas
- Transacciones

#>/<>

HACK A BOSS

<CODE YOUR TALENT>

CONSULTAS

Datos agrupados

Consultas en varias tablas

Consultas anidadas

#>/<>

HACK A BOSS

<CODE YOUR TALENT>

GROUP BY

GROUP BY

- Nos permite agrupar los resultados filtrados por una función de agrupamiento.

```
SELECT COUNT(*)  
FROM employee  
WHERE name LIKE '%A%'  
GROUP BY department_id;
```

- Todas las columnas deben devolver datos únicos por grupo

GROUP BY

GROUP BY

- Todas las columnas deben devolver datos únicos por grupo

```
SELECT MAX(salary), department_id  
FROM employee  
GROUP BY department_id;
```

- ¿Qué pasaría en este caso?

```
SELECT MAX(salary), department_id, name  
FROM employee  
GROUP BY department_id;
```

GROUP BY

HAVING

Nos permite aplicar condiciones sobre los grupos

```
SELECT MAX(salary), MIN(salary), COUNT(id), department_id  
FROM employee  
GROUP BY department_id  
HAVING MIN(salary) > 1000;
```

- Todas las condiciones se deben poder aplicar sobre cada grupo

CONSULTAS ANIDADAS

Las consultas anidadas o subconsultas son dos o más consultas donde el resultado de una consulta se utiliza como parámetro – valor – para realizar la siguiente consulta.

Consulta interior o subconsulta.

Es la primer consulta que se realiza. De esta consulta obtenemos el valor o los valores necesarios para que se realice la consulta exterior.

Consulta exterior.

Esta consulta utiliza el resultado de la consulta interior como parámetro y muestra el resultado que queremos obtener.

CONSULTAS ANIDADAS

Pueden existir cualquier número de consultas anidadas. Las consultas se realizan del interior al exterior.

```
select * from employee
where salary in (
    select salary from employee
    where name = 'FORD' or name = 'WARD'
);
```

```
select * from employee
where department_id IN (
    select id from department
    where location = 'NEW YORK'
);
```


JOINS

Permite obtener datos a partir de tablas relacionadas entre sí aplicando las condiciones indicadas.

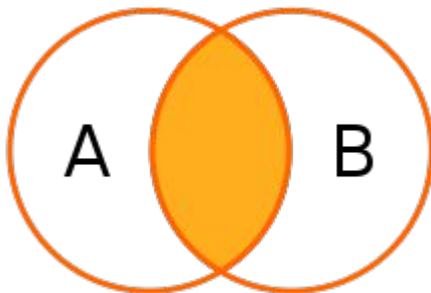
Varios tipos:

- **INNER JOIN**
- **LEFT JOIN**
- **RIGHT JOIN**
- **FULL OUTER JOIN**

INNER JOIN

Con esta operación cada registro en la tabla A es combinado con los correspondientes de la tabla B que satisfagan las condiciones que se especifiquen en el predicado del *JOIN*.

- Es el tipo de JOIN más utilizado, por lo que es considerado el tipo de combinación predeterminado.



INNER JOIN

Hay 2 formas de hacerlo.

EXPLÍCITO

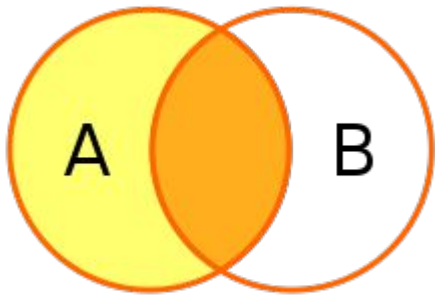
```
select * from employee e inner join department d  
on e.department_id = d.id;
```

IMPLÍCITO

```
select * from employee e, department d  
where e.department_id = d.id;
```

LEFT JOIN

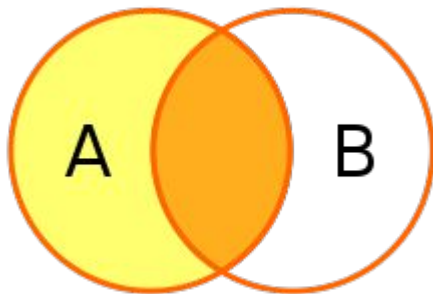
Devuelve todos los valores de la tabla **izquierda** relacionados con los valores de la tabla de la derecha correspondientes, si los hay. En caso que no haya relación devuelve un valor nulo en los campos de la tabla de la derecha.



```
select * from employee e left join department d
on e.department_id = d.id;
```

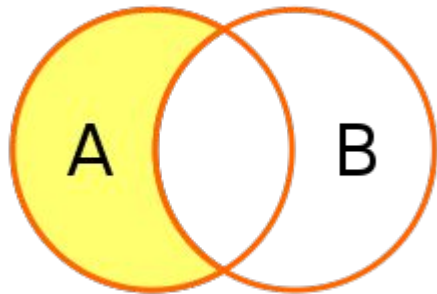
RIGHT JOIN

Devuelve todos los valores de la tabla **derecha** relacionados con los valores de la tabla de la izquierda correspondientes, si los hay. En caso de que no haya relación devuelve un valor nulo en los campos de la tabla de la izquierda.



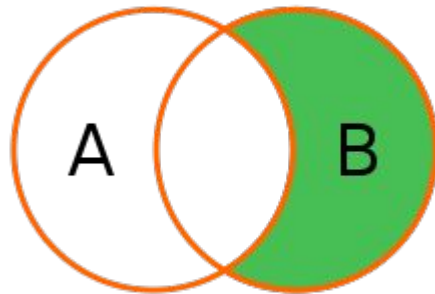
```
select * from employee e right join department d
on e.department_id = d.id;
```

SIN INTERSECCIONES



```
select * from employee e  
left join department d  
on e.department_id = d.id  
Where e.department_id is null;
```

Esto nos dará los empleados que no estén asignados a ningún departamento

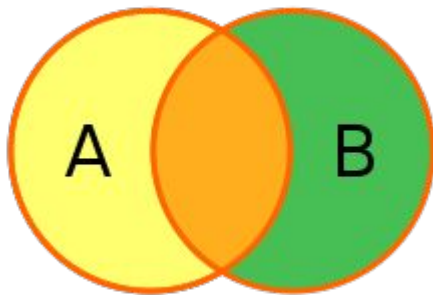


```
select * from employee e  
right join department d  
on e.department_id = d.id  
Where e.department_id is null;
```

Esto nos dará los departamentos que no tengan asignados ningún empleado.

FULL OUTER JOIN

Esta operación presenta los resultados de tabla izquierda y tabla derecha aunque alguna no tengan correspondencia en la otra tabla. La tabla combinada contendrá, entonces, todos los registros de ambas tablas y presentará valores NULL para registros sin pareja.



FULL OUTER JOIN

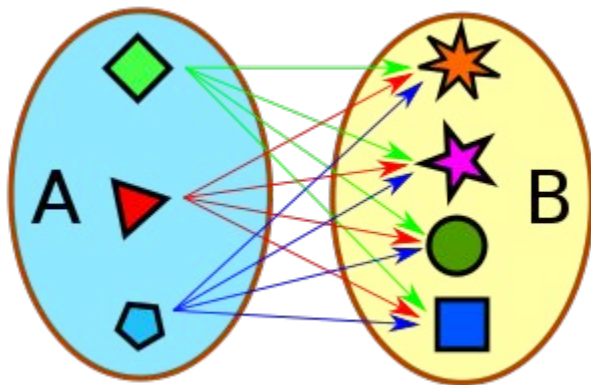
MySQL no soporta esta operación de forma nativa, por lo que hay que unir los resultados que producen las consultas LEFT JOIN y RIGHT JOIN.

```
SELECT *  
FROM employee e  
LEFT JOIN department d ON d.id = e.department_id  
UNION  
SELECT *  
FROM employee e  
RIGHT JOIN department d ON d.id = e.department_id;
```

- NOTA: El resultado de ambas consultas deberán tener las misma columnas.

CROSS JOIN

Presenta el producto cartesiano de los registros de las dos tablas. La tabla resultante tendrá todos los registros de la tabla izquierda combinados con cada uno de los registros de la tabla derecha. El número de filas en los resultado es por lo tanto $\text{COUNT}() \text{ de } A \times \text{COUNT}() \text{ de } B$.



```
select * from department cross join employee;
```



#>/<>

HACK A BOSS

<CODE YOUR TALENT>

#<THANX!>

#>/<>

HACK A BOSS

<CODE YOUR TALENT>

+34 919 04 23 63

www.hackaboss.com

Av.Linares Rivas 50-51, 15005, A Coruña

2020 EDITION