

Bases de Datos

7.- Consultas multitable

IES La Encantá

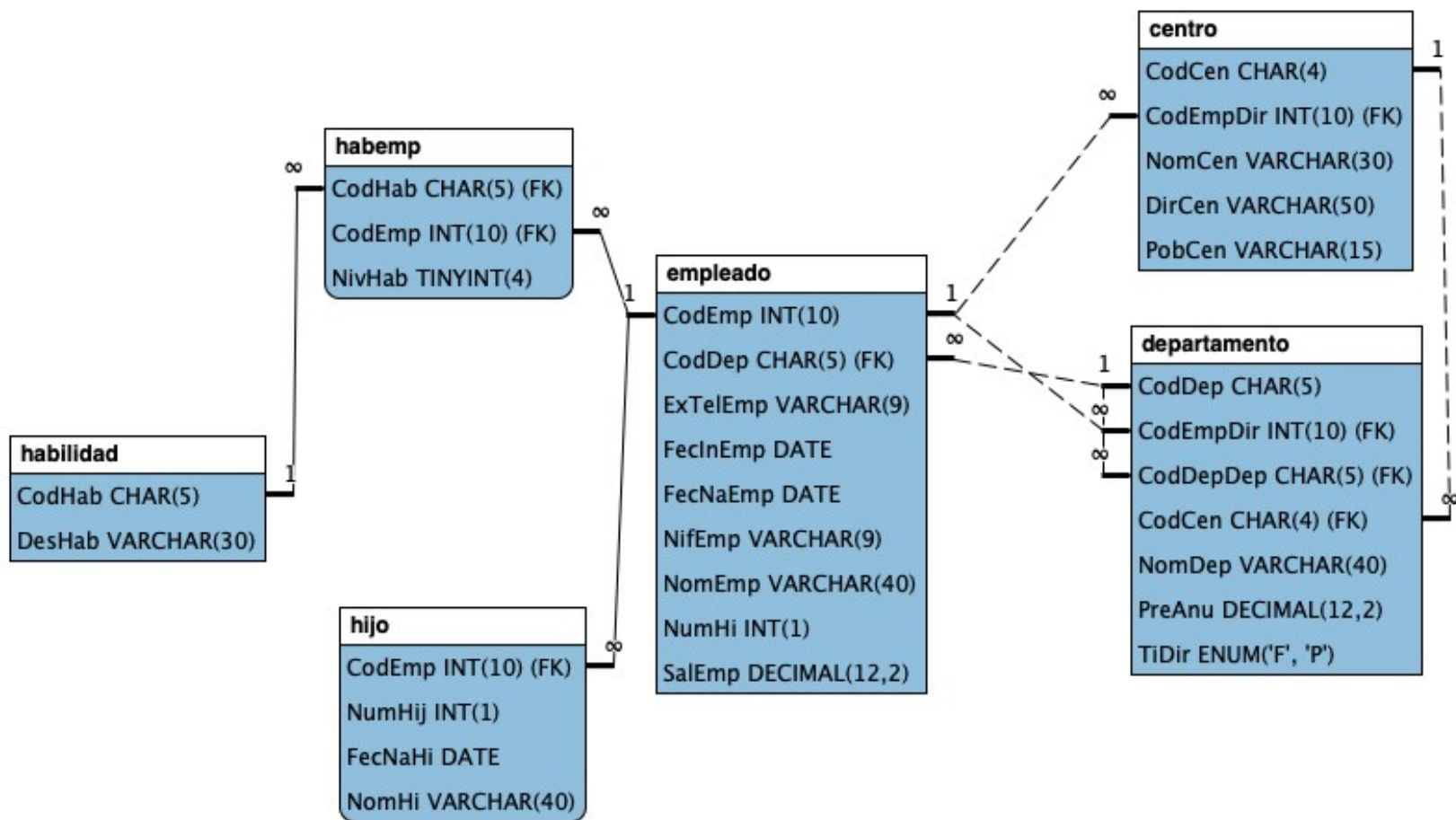


Índice



- Consultas multitas
● Producto cartesiano
● Inner Join
● Outer Join
- Operadores de conjuntos
- Funciones agregadas

BD empleados



Uniendo tablas



- Para unir varias tablas tenemos diferentes formas de hacerlo:
- <https://mariadb.com/kb/en/join-syntax/>
 - cross join
 - inner join
 - natural join
 - outer join

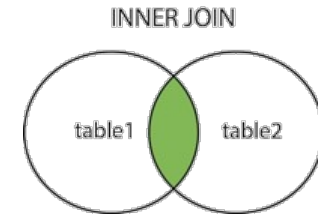
Producto cartesiano



- El producto cartesiano (*cross join*) de dos tablas son todas las combinaciones de las filas de una tabla unidas a las filas de la otra tabla.
 - Cada fila de A con todas las de B.
- `select nomemp, nomdep from empleado, departamento;`

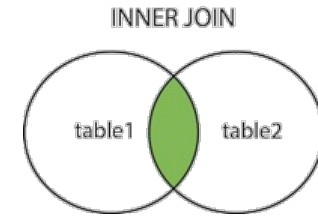
Empleado: 10 registros X Departamento: 6 registros
Resultado: 60 registros
- `select nomemp,nomdep from empleado cross join departamento;`

Composición interna I



- La operación [**INNER**] **JOIN** combina registros de dos tablas siempre que existan valores coincidentes en un campo común (clave ajena con clave primaria).
- ```
select nomemp,nomdep from empleado join departamento on empleado.coddep=departamento.coddep;
```
- ```
select nomemp,nomdep from empleado inner join departamento on empleado.coddep=departamento.coddep;
```
- 10 registros

Composición interna II



- Es conveniente poner un alias a las tablas.
- ```
select nomemp,nomdep from empleado e
join departamento d on
e.coddep=d.coddep;
```
- Este tipo de join es el que se realiza por defecto, con lo que se puede indicar dentro del WHERE
- ```
select nomemp,nomdep from empleado e,  
departamento d where e.coddep=d.coddep;
```

NATURAL JOIN



- Especialización de INNER JOIN.
- Se comparan todas las columnas que tengan el mismo nombre en ambas tablas.
 - La tabla resultante contiene sólo una columna por cada par de columnas con el mismo nombre
- ```
select CodDep, NomDep, NomEmp from
empleado natural join
departamento;
```



# Consultas sobre varias tablas



- Lo importante en las composiciones internas es emparejar los campos que han de tener valores iguales (FKs con PKs)
- Reglas:
  - Pueden combinarse tantas tablas como se desee.
  - El criterio de combinación puede estar formado por más de una pareja de columnas.
  - En la cláusula SELECT pueden citarse columnas de ambas tablas, condicionen o no, la combinación.
  - Si hay columnas con el mismo nombre en las distintas tablas, deben identificarse especificando la tabla de procedencia o utilizando un alias de tabla.

# Ejemplos join



-- Empleado y departamento en el que trabaja

```
select NomEmp, NomDep from empleado e, departamento d
where e.CodDep = d.CodDep;
select NomEmp, NomDep from empleado e
join departamento d on e.CodDep = d.CodDep;
```

-- Empleado, departamento y centro en el que trabaja

```
select NomEmp, NomDep, NomCen from empleado e, departamento d, centro c
where e.CodDep = d.CodDep and d.CodCen = c.CodCen;
select NomEmp, NomDep, NomCen from empleado e
join departamento d on e.CodDep = d.CodDep
join centro c on d.CodCen = c.CodCen;
```

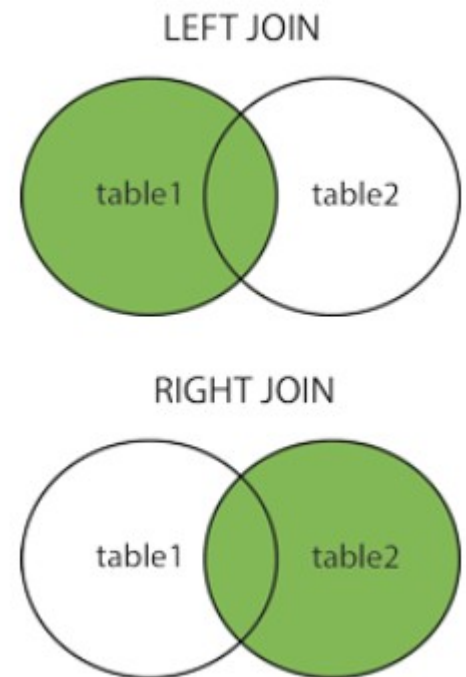
-- Empleado, departamento, centro y director del centro en el que trabaja

```
select e.NomEmp, NomDep, NomCen, e2.NomEmp as Director
from empleado e, departamento d, centro c, empleado e2
where e.CodDep = d.CodDep and d.CodCen = c.CodCen and c.CodEmpDir =
e2.CodEmp;
select e.NomEmp, NomDep, NomCen, e2.NomEmp as Director from empleado e
join departamento d on e.CodDep = d.CodDep
join centro c on d.CodCen = c.CodCen
join empleado e2 on c.CodEmpDir = e2.CodEmp;
```

# Composición externa



- Cuando efectuamos una combinación interna los registros que se quedan sin emparejar en la otra tabla no aparecen en el resultado.
- Mediante una combinación externa, aparecerán los datos emparejados más los que no están relacionados, ya sea por la:
  - **Izquierda:** `left [outer] join` – Todos los de la tabla de la izquierda, y de la derecha los relacionados.
  - **Derecha:** `right [outer] join` – Los de la izquierda los relacionados, y de la derecha todos los datos.



# Ejemplos *left/right join* I



-- OUTER JOIN CON DOS TABLAS

-- Empleado y código de la habilidad  
-- (empleados que tienen habilidades)

```
select nomemp, codhab
from empleado e inner join habemp h on e.codemp = h.codemp;
-- Todos los empleados y código de la habilidad si la tienen
-- (puede haber habilidades nulas)
```

```
select nomemp, codhab
from empleado e left outer join habemp h on e.codemp = h.codemp;
-- Igual que el join, ya que para todo habemp siempre hay un empleado
```

```
select nomemp, codhab
from empleado e right outer join habemp h on e.codemp = h.codemp;
-- Misma consulta que left join pero en orden inverso
```

```
select nomemp, codhab
from habemp h right join empleado e on e.codemp = h.codemp;
```

# Ejemplos left/right join II



-- OUTER JOIN CON TRES TABLAS

-- Empleado y nombre de la habilidad  
-- (empleados que tienen habilidades)

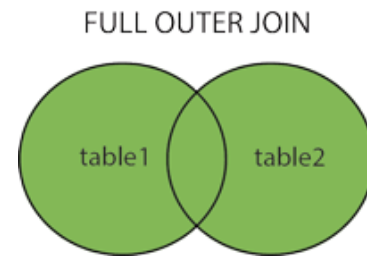
```
select nomemp, deshabs
from empleado e join habemp he on e.codemp = he.codemp
 join habilidad h on he.codhab = h.codhab;
-- Todos los empleados y nombre de la habilidad si la tienen
-- (puede haber habilidades nulas)
-- Mal: al hacer el segundo join, elimina los que tienen
-- la habilidad nula
```

```
select nomemp, deshabs
from empleado e left join habemp he on e.codemp = he.codemp
 join habilidad h on he.codhab = h.codhab;
-- Bien, primero unimos los códigos y habilidades,
-- y luego todos los empleados
```

```
select nomemp, deshabs
from habemp he join habilidad h on he.codhab = h.codhab
 right join empleado e on e.codemp = he.codemp;
-- Todas las habilidades, y si lo hay, el empleado que la tiene
```

```
select nomemp, deshabs
from empleado e join habemp he on e.codemp = he.codemp
 right join habilidad h on he.codhab = h.codhab;
```

# Full Outer Join



- No está soportado por MariaDB / MySQL
- Combinación externa completa
- Presenta los resultados de la tabla de la izquierda y de la tabla de la derecha aunque no tengan correspondencia en la otra tabla.
- La tabla combinada contendrá entonces todos los registros de ambas tablas y presentará valores nulos para registros sin pareja.

```
-- full outer join sin repetidos
select * from t1
left join t2 on t1.id = t2.id
union
select * from t1
right join t2 on t1.id = t2.id
```

# Ejemplo *Full Outer Join*

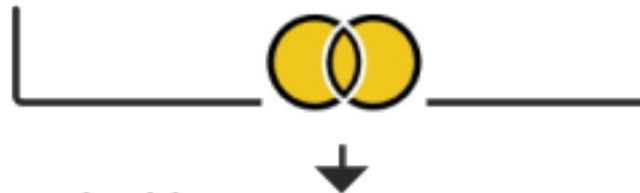


Left Table

| Date     | CountryID | Units |
|----------|-----------|-------|
| 1/1/2020 | 1         | 40    |
| 1/2/2020 | 1         | 25    |
| 1/3/2020 | 3         | 30    |
| 1/4/2020 | 2         | 35    |

Right Table

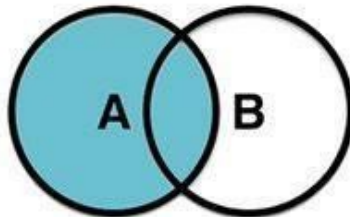
| ID | Country |
|----|---------|
| 1  | USA     |
| 2  | Canada  |
| 3  | Panama  |
| 4  | Spain   |



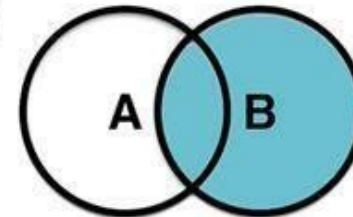
Merged Table

| Date     | CountryID | Units | Country |
|----------|-----------|-------|---------|
| 1/1/2020 | 1         | 40    | USA     |
| 1/2/2020 | 1         | 25    | USA     |
| 1/4/2020 | 2         | 35    | Canada  |
| 1/3/2020 | 3         | 30    | Panama  |
| null     | null      | null  | Spain   |

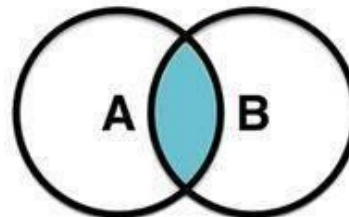
# GUIA VISUAL SQL JOINS



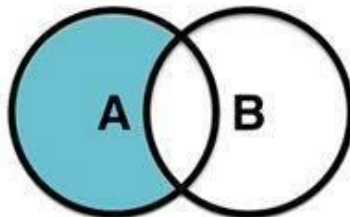
```
SELECT <fields list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```



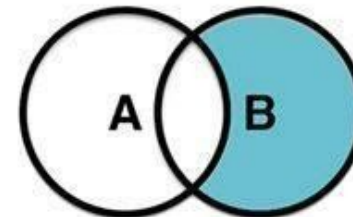
```
SELECT <fields list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```



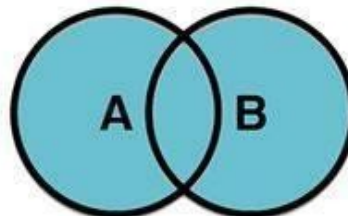
```
SELECT <fields list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```



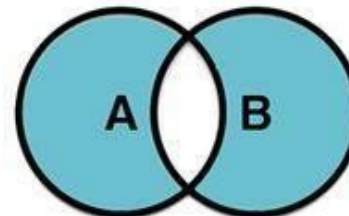
```
SELECT <fields list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```



```
SELECT <fields list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
```



```
SELECT <fields list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```



```
SELECT <fields list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL OR B.Key IS NULL
```



# Condiciones y outer join



- Al hacer un OUTER JOIN, si el filtrado se debe realizar antes de unir las tablas, hay que indicarlo dentro del ON:

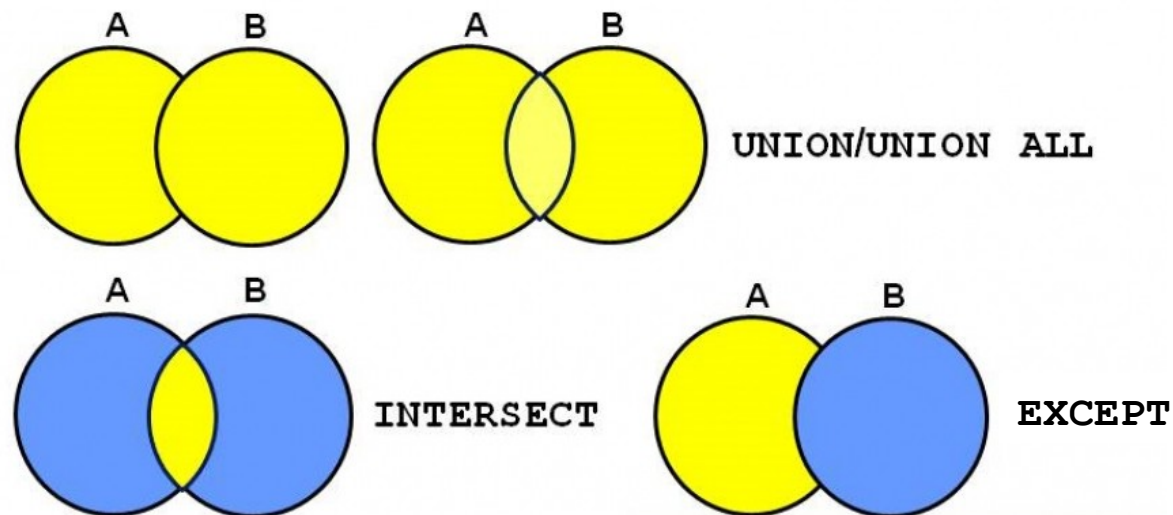
```
from empleado e join habemp he on e.codemp = he.codemp
and e.NumHi > 0
```

- En cambio, si ponemos la condición en el WHERE, el filtrado se realiza tras aplicarse el JOIN.
- En cambio, en los inner join esto no ocurre, y el resultado será siempre el mismo

# Operadores de conjuntos



- <https://mariadb.com/kb/en/joins-subqueries/>
- Se realizan dos o más consultas y se combinan.
- La cantidad y tipo de datos de las columnas debe ser el mismo



# Ejemplos conjuntos



```
-- directivos y empleados con hijos
select codempdir from departamento
union
select codemp from empleado where numhi > 0
```

```
-- directivos que tienen hijos
select codempdir from departamento
intersect
select codemp from empleado where numhi > 0
```

```
-- directivos que no tienen hijos
select codempdir from departamento
except
select codemp from empleado where numhi > 0
```

# Funciones agregadas / de resumen



- Se emplean para realizar cálculos sobre el total de elementos de la consulta (o agrupación)
- `count (*)`: cuenta las filas
- `sum(col)`: Suma de todos los valores de una columna
- `min(col)` / `max(col)`: Mínimo o máximo de una columna
- `avg(col)`: Valor medio de una columna

# Ejemplos funciones agregadas



-- Cantidad de empleados

```
select count(*) from empleado;
```

-- Cantidad de empleados con hijos

```
select count(*) from empleado where NumHi >= 0;
```

-- Cantidad de empleados que tienen habilidades

```
select count(distinct e.codemp)
from empleado e inner join habemp h on e.codemp = h.codemp;
```

-- Total de hijos de los empleados

```
select sum(NumHi) from empleado;
```

-- Salario más alto

```
select max(SalEmp) from empleado;
```

-- Salario medio

```
select avg(SalEmp) from empleado;
```



¿Alguna pregunta?