

# Gestión de bases de datos

Diseño físico de  
bases de datos

# Consultas

Habitualmente habrá que hacer peticiones a la base de datos para obtener información almacenada y relacionada en diferentes tablas.

Sintaxis: SELECT TABLA1.COL1, TABLA1.COL2, ...,  
TABLA2.COL1, TABLA2.COL2, ... FROM TABLA1  
[CROSS JOIN TABLA2] | [NATURAL JOIN TABLA2] |  
[JOIN TABLA2 USING (COL)] | [JOIN TABLA2 ON  
(TABLA1.COL=TABLA2.COL)] | [LEFT | RIGHT | FULL  
OUTER JOIN TABLA2 ON  
(TABLA1.COL=TABLA2.COL)]

# **Composiciones internas**

Se refieren a la combinación de dos tablas sin ninguna restricción.

El resultado son todas las combinaciones posibles entre las dos tablas.

# Ejercicio 1

Realiza una composición interna entre la tabla partidos y la tabla equipos.

# Solución Ejercicio 1

```
select * from partidos, equipos;
```

Como la tabla partidos tiene 6 registros y la tabla equipos 3 filas, el resultado de combinar ambas son **18 registros posibles**.

# **Emparejar campos**

En las composiciones internas hay que emparejar aquellos campos que deben tener un valor igual.

Si un campo tiene el mismo nombre en las distintas tablas, hay que especificarlo indicando el nombre de la tabla junto a un punto y el nombre del campo.

# **Emparejar campos**

En el filtrado (WHERE) se indicarán las condiciones del emparejamiento, siendo una de las columnas clave principal en su tabla.

Si se quiere combinar una tabla consigo misma hay que poner un alias a uno de los nombres de la tabla.

## **Ejercicio 2**

Muestra los datos de los nombres de los futbolistas junto al nombre de su equipo.



## **Solución Ejercicio 2**

```
select futbolistas.nombre, equipos.nombre from  
futbolistas, equipos where  
futbolistas.ID_EQUIPO = equipos.ID;
```

# Composiciones externas

A veces se requiere seleccionar filas de una tabla sin correspondencia con las filas de otra tabla.

Ej.: se ha creado un nuevo equipo de fútbol pero todavía no se han inscrito los jugadores en la base de datos. Si queremos un listado de jugadores por equipo, debería mostrarse ese nuevo equipo con 0 jugadores.

# Composiciones externas

Sintaxis: en la condición de filtrado WHERE, en la igualdad entre campos se incluye (+) detrás del nombre de la tabla de la que queremos aceptar nulos. (SOLO ORACLE)

## **Ejercicio 3**

Incluye un nuevo equipo en la tabla EQUIPOS.

Obtén un listado de los equipos y jugadores que pertenecen a estos (nombre de equipo, nombre de jugador y posición).

## **Solución Ejercicio 3**

```
insert into equipos VALUES (4, 'EQUIPO D',  
'ESTADIO D', 150000);
```

```
select equipos.nombre, futbolistas.nombre,  
apellidos from equipos, futbolistas where  
equipos.ID = futbolistas.ID_EQUIPO(+);
```

# JOIN

Supone una mejora en la sintaxis de creación de composiciones de tablas.

## Sintaxis:

```
SELECT TABLA1.COL1, ... TABLA1.COLN, TABLA2.COL1, ...  
TABLA2.COLN  
FROM TABLA1  
    [CROSS JOIN TABLA2] |  
    [NATURAL JOIN TABLA2] |  
    [JOIN TABLA2 USING (COL)] |  
    [JOIN TABLA2 ON (TABLA1.COL = TABLA2.COL)] |  
    [LEFT | RIGHT | FULL OUTER JOIN TABLA2 ON  
    (TABLA1.COL = TABLA2.COL)];
```

# CROSS JOIN

Crea un producto cartesiano de las filas de ambas tablas si no se utiliza WHERE.

Se comporta igual que el primer SELECT visto de varias tablas.

Ej.: `SELECT COL1, COL2, ... COLN FROM  
TABLA1 CROSS JOIN TABLA2;`

## **Ejercicio 4**

Muestra el nombre de los futbolistas seguido del nombre del equipo en el que juega utilizando CROSS JOIN.

Comenta el resultado.



## **Solución Ejercicio 4**

```
SELECT F.NOMBRE, E.NOMBRE FROM  
FUTBOLISTAS F CROSS JOIN EQUIPOS  
E;
```

Se obtiene el resultado de combinar todas las filas de ambas tablas.

## **NATURAL JOIN**

Detecta de forma automática las claves de unión basándose en el nombre de las columnas. Por tanto, el nombre de las columnas debe coincidir en ambas tablas sobre la columna de unión, además del tipo de dato.

EJ.: SELECT \* FROM TABLA1 NATURAL JOIN TABLA2;

NOMBRE CAMPO FK TABLA1 = NOMBRE CAMPO PK TABLA2

## **Ejercicio 5**

Muestra todos los datos de empleados y departamentos (EMP y DEPT) el nombre de los futbolistas seguido del nombre del equipo en el que juega utilizando CROSS JOIN.

Comenta el resultado.

## **Solución Ejercicio 5**

```
select * from dept natural join emp;
```

Se obtiene el resultado de combinar las filas de ambas tablas que coincidan en el campo DEPTNO que se repite en ambas tablas.

## **JOIN ... USING**

Permite establecer con qué campos se van a relacionar las tablas, de forma que, aunque haya más de un campo para relacionar, solo se tendrá en cuenta la relación con el campo que se indique.

## **Ejercicio 6**

Muestra el nombre de los futbolistas seguido del nombre del equipo en el que juega utilizando JOIN ... USING.

Comenta el resultado.

## **Solución Ejercicio 6**

```
select f.nombre, e.nombre from  
futbolistas f join equipos e using  
(id_equipo);
```

Al principio fallará porque el nombre de la columna no coincide en ambas tablas.

```
alter table equipos rename column id  
to id_equipo;
```

## **JOIN ... ON**

Une tablas en las que los nombres de las columnas no coinciden entre las tablas.

```
EJ.: SELECT TABLA1.COL, TABLA2. COL, ...  
FROM TABLA1 JOIN TABLA2 ON TABLA1.COL =  
TABLA2.COL;
```



## **Ejercicio 7**

Muestra el nombre de los futbolistas seguido del nombre del equipo en el que juega utilizando JOIN ... ON.

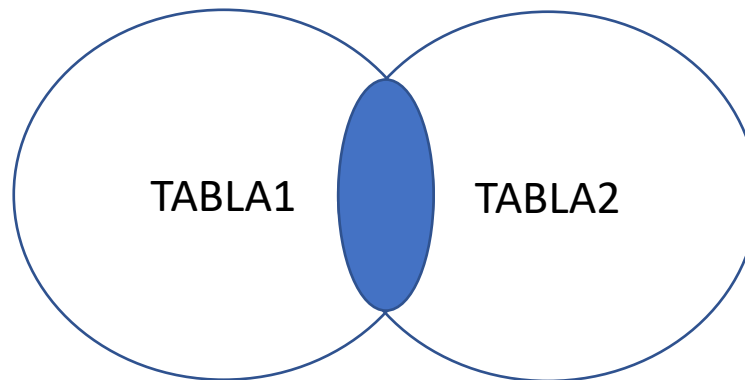
Comenta el resultado.

# **Solución Ejercicio 7**

```
select f.nombre, e.nombre from  
futbolistas f join equipos e on  
f.id_equipo = e.id;
```

# INNER JOIN

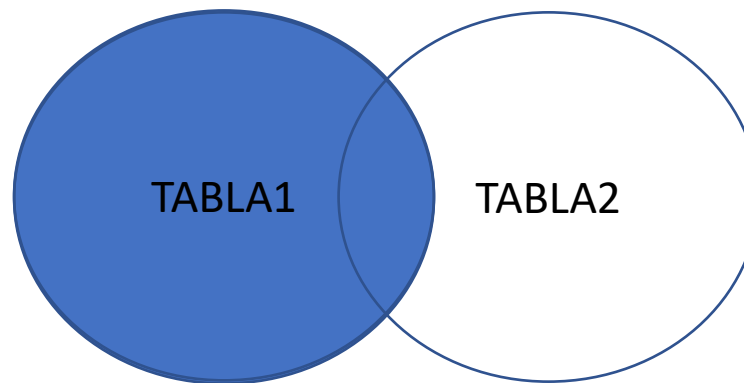
Devuelve los resultados que coinciden en ambas tablas.



```
SELECT * FROM TABLA1  
INNER JOIN TABLA2 ON CAMPO  
RELACIONADO ENTRE AMBAS
```

## LEFT JOIN

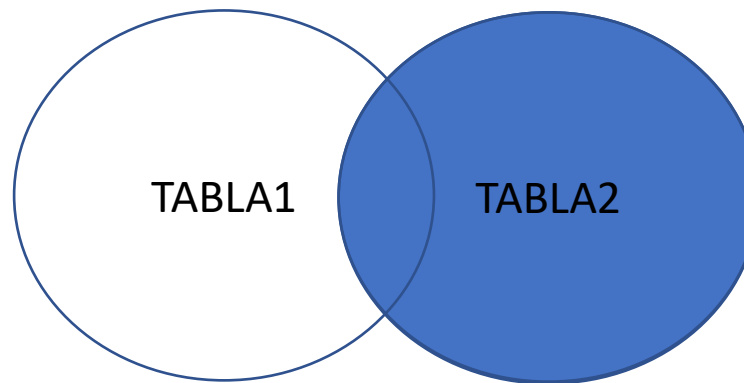
Es una composición izquierda. Esto quiere decir que todas las filas de la tabla de la izquierda se devuelven, aunque no haya ninguna columna correspondiente en las tablas combinadas.



```
SELECT * FROM TABLA1  
LEFT JOIN TABLA2 ON CAMPO  
RELACIONADO ENTRE AMBAS
```

## RIGHT JOIN

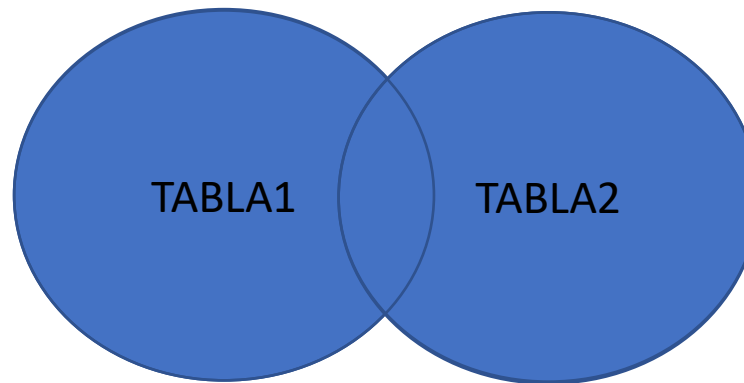
Es una composición derecha. Esto quiere decir que todas las filas de la tabla de la derecha se devuelven, aunque no haya ninguna columna correspondiente en las tablas combinadas.



```
SELECT * FROM TABLA1  
RIGHT JOIN TABLA2 ON CAMPO  
RELACIONADO ENTRE AMBAS
```

# **FULL OUTER JOIN**

Es una composición externa en la que se devolverán todas las filas de ambas tablas, y de los campos no relacionados de ambas tablas (donde pondrá null).



```
SELECT * FROM TABLA1  
FULL OUTER JOIN TABLA2 ON CAMPO  
RELACIONADO ENTRE AMBAS
```

## **Ejercicio 8**

Obtén el nombre, apellidos y nombre del equipo de los futbolistas que sean defensas.

## Solución Ejercicio 8

```
SELECT F.NOMBRE, F.APELLIDOS, E.NOMBRE  
FROM FUTBOLISTAS F JOIN EQUIPOS E ON  
F.ID_EQUIPO = E.ID WHERE F.POSICION =  
'DEFENSA';
```

	NOMBRE	APELLIDOS	NOMBRE_1
1	PEDRO LUIS	GOMEZ	EQUIPO A
2	PABLO	CARO	EQUIPO C
3	ESTEBAN	BENITEZ	EQUIPO C



## **Ejercicio 9**

Obtén el nombre, apellidos y nombre del equipo de los futbolistas con un LEFT JOIN y un RIGHT JOIN. ¿Cuál es la diferencia?

## **Solución Ejercicio 9**

```
SELECT F.NOMBRE, F.APELLIDOS, E.NOMBRE  
FROM FUTBOLISTAS F LEFT JOIN EQUIPOS E  
ON F.ID_EQUIPO = E.ID;
```

```
SELECT F.NOMBRE, F.APELLIDOS, E.NOMBRE  
FROM FUTBOLISTAS F RIGHT JOIN EQUIPOS E  
ON F.ID_EQUIPO = E.ID;
```

El LEFT JOIN devuelve 7 registros mientras que el RIGHT JOIN devuelve 8 (el equipo D que no tiene futbolistas).

# **OTRAS CONSULTAS MULTITABLAS**

Existen tres consultas multitablas muy interesantes como son: UNION, INTERSECT y MINUS.

Estas operaciones se realizan entre dos sentencias SELECT.

Los dos SELECT deben tener el mismo número de columnas, mismos tipos de datos y mismo orden.

# UNION

Combina las filas de un SELECT con las filas de otro SELECT, sin que se muestren las filas duplicadas.

Es útil para combinar dos tablas idénticas con diferentes registros (backups antiguos, datos separados por años en tablas diferentes, etc.).

Sintaxis: SELECT ... UNION SELECT ...

# Ejercicio 10

Combina con UNION los salarios y presupuestos de futbolistas y equipos para tener en una única columna de resultados los valores monetarios de la base de datos (llámale a esa columna euros).

# **Solución Ejercicio 10**

```
SELECT SALARIO "EUROS" FROM  
FUTBOLISTAS UNION SELECT PRESUPUESTO  
FROM EQUIPOS;
```

# INTERSECT

Examina las filas de dos SELECT y devolverá aquellas que aparezcan en ambos conjuntos, sin mostrar filas duplicadas.

Sintaxis: SELECT ... INTERSECT SELECT ...

# Ejercicio 11

Combina con INTERSECT los estadios de los equipos y los partidos.



## Solución Ejercicio 1

select estadio from equipos INTERSECT select  
estadio from partidos;

	ESTADIO
1	ESTADIO A
2	ESTADIO B
3	ESTADIO C

# MINUS

Devuelve las filas que están en un primer SELECT pero no en el segundo SELECT. Las filas que estén duplicadas del primer SELECT se reducirán a una antes de comenzar la comparación.

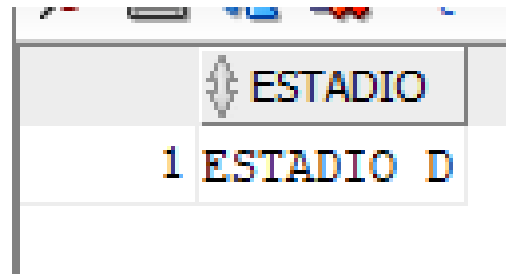
Sintaxis: SELECT ... MINUS SELECT ...

# Ejercicio 12

Combina con MINUS los estadios de los equipos y de los partidos (en ese orden).

## Solución Ejercicio 12

select estadio from equipos minus select estadio  
from partidos;



	ESTADIO
1	ESTADIO D



# CEU

*Centro de Estudios  
Profesionales*

---

*Fundación San Pablo Andalucía*