

JOINS:

Casos complejos

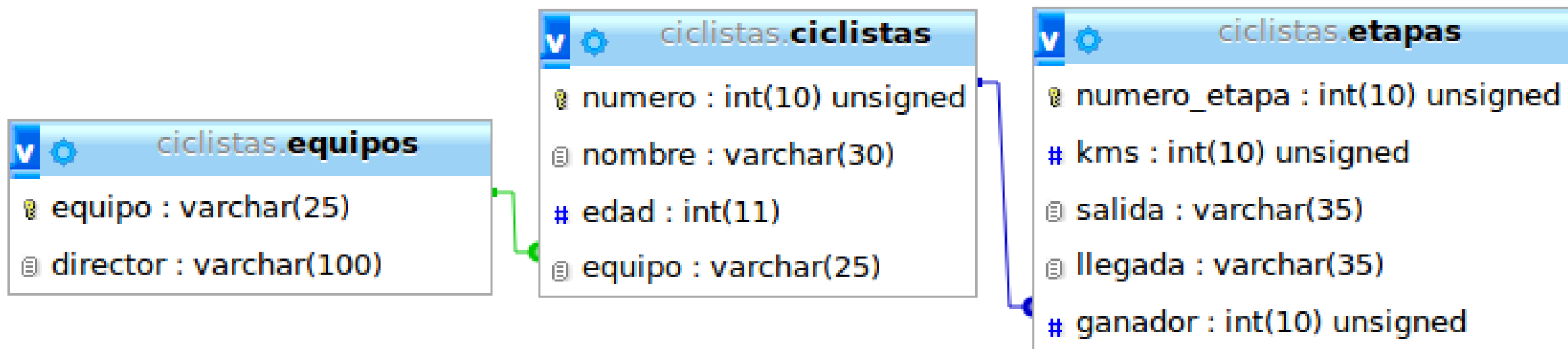
(SELECT a más de una tabla)

Bases de datos 2 - DS

Comisión 3º2ª

Prof. Ing. Juan S. Marquez

Tablas de ejemplo



SELF JOIN

- Es el join de una tabla **consigo misma**.
- Puede ser de cualquiera de los tipos vistos hasta aquí.
- Los **alias de tabla** son **obligatorios**
- **SELECT A.campo1, A.campo2, B.campo1**
FROM tabla1 A
[LEFT|RIGHT|INNER] JOIN tabla1 B
ON *condición*;

SELF JOIN

- Ejemplo: Seleccionar nombre y edad de cada ciclista, y la cantidad de ciclistas más jóvenes que él que participan en la carrera.
- **SELECT c.numero, c.nombre, c.edad,
COUNT(mas_jovenes.numero)
FROM ciclistas c
LEFT JOIN ciclistas mas_jovenes
ON c.edad > mas_jovenes.edad
GROUP BY c.numero
ORDER BY COUNT(mas_jovenes.numero);**

SELF JOIN

#	numero	nombre	edad	COUNT(mas_jovenes.numero)
1	49	Eugeni Berzin	23	0
2	66	Enrico Zaina	24	1
3	46	Agustin Sagasti	24	1
4	98	Eleuterio Anguita	25	3
5	38	Javier Palacin	25	3
6	41	Rolf Aldag	25	3
7	61	Walte Castignola	26	6
8	52	Vladislav Bobrik	26	6
9	45	Alberto Elli	26	6
10	3	Alex Zulle	27	9
11	50	Fernando Esc...	27	9
12	44	Luca Gelfi	27	9
13	90	Ivan Ivanov	27	9
14	23	Lance Armstrong	27	9
15	69	Eddy Seigneur	27	9
16	92	Federico García	27	9

*(Salida incompleta,
100 registros en total)*

SELF JOIN

- Otro ejemplo: Seleccionar nombre, edad y equipo de cada ciclista, junto con la cantidad de ciclistas que componen su equipo.
- **SELECT c.numero, c.nombre, c.edad, c.equipo,
COUNT(mismo_equipo.numero)
FROM ciclistas c
JOIN mismo_equipo
ON c.equipo = mismo_equipo.equipo
GROUP BY c.numero;**

SELF JOIN

#	numero	nombre	edad	equipo	COUNT(mismo_equipo.numero)
1	1	Miguel Indurain	32	Banesto	11
2	2	Pedro Delgado	35	Banesto	11
3	3	Alex Zulle	27	ONCE	5
4	4	Tony Rominger	30	Mapei-Clas	7
5	5	Gert-Jan Theunisse	32	TVM	6
6	6	Adriano Baffi	33	Mercatone Uno	8
7	7	Massimiliano Lelli	30	Mercatone Uno	8
8	8	Jean Van Poppel	33	Lotus Festina	3
9	9	Massimo Podenzana	34	Navigare	5
10	10	Mario Cipollini	28	Mercatone Uno	8
11	11	Flavio Giupponi	31	Bresciali-Refin	4
12	12	Alessio Di Basco	31	Amore Vita	3

*(Salida incompleta,
100 registros en total)*

SELF JOIN

- Otro ejemplo parecido: Seleccionar nombre, edad y equipo de cada ciclista, junto con la cantidad de compañeros de equipo que tiene.
- **SELECT** **c.numero,c.nombre,c.edad,c.equipo,**
COUNT(mismo_equipo.numero) **companeros**
FROM **ciclistas c**
LEFT JOIN **ciclistas mismo_equipo**
ON **c.equipo = mismo_equipo.equipo**
AND **c.numero <> mismo_equipo.numero**
GROUP BY **c.numero;**
- Se añade un alias de columna
- Se añade LEFT JOIN, para que figuren los ciclistas con 0 compañeros de equipo (en este caso, no hay ninguno).
- Se añade una segunda condición al JOIN, con AND.

SELF JOIN

#	numero	nombre	edad	equipo	companeros
1	1	Miguel Indurain	32	Banesto	10
2	2	Pedro Delgado	35	Banesto	10
3	3	Alex Zulle	27	ONCE	4
4	4	Tony Rominger	30	Mapei-Clas	6
5	5	Gert-Jan Theunisse	32	TVM	5
6	6	Adriano Baffi	33	Mercatone Uno	7
7	7	Massimiliano Lelli	30	Mercatone Uno	7
8	8	Jean Van Poppel	33	Lotus Festina	2
9	9	Massimo Podenzana	34	Navigare	4
10	10	Mario Cipollini	28	Mercatone Uno	7
11	11	Flavio Giupponi	31	Bresciali-Refin	3
12	12	Alessio Di Basco	31	Amore Vita	2

*(Salida incompleta,
100 registros en total)*

JOIN USING()

- Es una sintaxis alternativa cuando:
 - La condición del JOIN es de igualdad
 - Ambas columnas tienen el mismo nombre
- En lugar de escribir:
SELECT A.campo1, A.campo2, B.campo3
FROM tabla1 A
JOIN tabla2 B ON A.campo4 = B.campo4;
- ... podemos escribir:
SELECT A.campo1, A.campo2, B.campo3
FROM tabla1 A
JOIN tabla2 B USING(campo4);

JOIN USING()

- Ejemplo: Seleccionar ciclista, equipo y director.
- **SELECT** *c.nombre,c.equipo,q.director*
FROM *ciclistas c* **JOIN** *equipos q*
ON *c.equipo = q.equipo;*
- Equivale a:
- **SELECT** *c.nombre,equipo,q.director*
FROM *ciclistas c* **JOIN** *equipos q*
USING(*equipo*);

JOIN USING()

#	nombre	equipo	director	
1	Alessio Di Basco	Amore Vita	Ricardo Padacci	
2	Pascal Lino	Amore Vita	Ricardo Padacci	
3	Claudio Chioccioli	Amore Vita	Ricardo Padacci	
4	Alfonso Gutierrez	Artiach	Jose Perez	
5	Erwin Nijboer	Artiach	Jose Perez	
6	Alberto Elli	Artiach	Jose Perez	
7	Fernando Mota	Artiach	Jose Perez	
8	Ivan Ivanov	Artiach	Jose Perez	
9	Federico García	Artiach	Jose Perez	
10	Eleuterio Anguita	Artiach	Jose Perez	
11	Miguel Indurain	Banesto	Miguel Echev...	
12	Pedro Delgado	Banesto	Miguel Echev...	
13	Julian Gorospe	Banesto	Miguel Echev...	
14	Mikel Zarrabeitia	Banesto	Miguel Echev...	
15	Jesus Montoya	Banesto	Miguel Echev...	

*(Salida incompleta,
100 registros en total)*

NATURAL JOIN

- Es una sintaxis alternativa cuando:
 - La condición del JOIN es de igualdad
 - Ambas columnas tienen el mismo nombre
 - No hay columnas con el mismo nombre que no formen parte de la condición.
- En lugar de escribir:
SELECT A.campo1, A.campo2, B.campo3
FROM tabla1 A
JOIN tabla2 B ON A.campo4 = B.campo4;
- ... podemos escribir:
SELECT A.campo1, A.campo2, B.campo3
FROM tabla1 A
NATURAL JOIN tabla2 B;
- siempre y cuando no existan campos con el mismo nombre en ambas tablas (ej: campo5) que no formen parte de la condición del join.

NATURAL JOIN

- Ejemplo: Seleccionar ciclista, equipo y director.
- **SELECT c.nombre, equipo, q.director**
FROM ciclistas c JOIN equipos q
ON c.equipo = q.equipo;
- Equivale a:
SELECT c.nombre, equipo, q.director
FROM ciclistas c JOIN equipos q
USING(equipo);
- Y también equivale a:
SELECT c.nombre, equipo, q.director
FROM ciclistas c NATURAL JOIN equipos q;
(porque no hay ningún campo con nombre repetido en ambas tablas salvo **equipo**, que forma parte de la condición del JOIN.)

NATURAL JOIN

- El uso del **NATURAL JOIN** **no se considera una buena** práctica, salvo para consultas eventuales que no se guardarán.
- Para código en producción (como parte de un sistema, procedimiento almacenado, etc.) es una mala práctica, porque:
 - Dificulta la lectura
 - No es obvio el criterio por el cual se unen las tablas.
 - Si alguna de las dos tablas agrega una columna, el `SELECT` podría comenzar a actuar de manera inesperada.

JOIN USING()

#	nombre	equipo	director	
1	Alessio Di Basco	Amore Vita	Ricardo Padacci	
2	Pascal Lino	Amore Vita	Ricardo Padacci	
3	Claudio Chioccioli	Amore Vita	Ricardo Padacci	
4	Alfonso Gutierrez	Artiach	Jose Perez	
5	Erwin Nijboer	Artiach	Jose Perez	
6	Alberto Elli	Artiach	Jose Perez	
7	Fernando Mota	Artiach	Jose Perez	
8	Ivan Ivanov	Artiach	Jose Perez	
9	Federico García	Artiach	Jose Perez	
10	Eleuterio Anguita	Artiach	Jose Perez	
11	Miguel Indurain	Banesto	Miguel Echev...	
12	Pedro Delgado	Banesto	Miguel Echev...	
13	Julian Gorospe	Banesto	Miguel Echev...	
14	Mikel Zarrabeitia	Banesto	Miguel Echev...	
15	Jesus Montoya	Banesto	Miguel Echev...	

*(Salida incompleta,
100 registros en total)*

INNER JOIN implícitos

- Es equivalente al INNER JOIN explícito, tanto en funcionamiento como en performance.
- **No se considera una buena práctica**, porque vuelve menos clara la sintaxis. MySQL, en sus primeras versiones, no permitía los JOINS explícitos, por lo que era la única forma. Mucho código que aún está funcionando está escrito con JOINS implícitos.
- No se pueden hacer ni LEFT ni RIGHT JOINS
- En lugar de escribir: **SELECT A.campo1, A.campo2, B.campo3**
FROM tabla1 A INNER JOIN tabla2 B ON condición;
- ...se escriben condiciones del JOIN directamente en el WHERE:
- **SELECT A.campo1, A.campo2, B.campo3**
FROM tabla1 A, tabla2 B WHERE condición;

UNION

- **No** es un JOIN (que sirven para unir dos o más tablas): **UNION sirve para unir dos o más SELECT.**
- Los SELECT unidos deben tener la misma cantidad de columnas.
- Las filas repetidas se eliminan automáticamente, salvo que se utilice UNION ALL.
- **SELECT campo1, campo2, campo3 FROM tabla1
UNION
SELECT campoA, campoB, campoC FROM tabla2;**

UNION

- Ejemplo: Mostrar el nombre de todas las personas que figuran en la BD (tanto ciclistas como directores de equipo):
- **SELECT nombre FROM ciclistas
UNION
SELECT director FROM equipos;**
- Si en caso de que dos personas tengan idéntico nombre, el nombre debe aparecer repetido:
- **SELECT nombre FROM ciclistas
UNION ALL
SELECT director FROM equipos;**