

Bases de Datos

Unidad 5: Realización de consultas avanzadas Sesión 1

Hasta ahora únicamente hemos visto consultas realizadas sobre una única tabla.

Es muy frecuente y necesario tener que realizar consultas sobre **combinaciones de tablas** dado que necesitamos obtener datos en la consulta de varias tablas y/o tener condiciones aplicadas a datos de varias tablas.

Por ejemplo, en la base de datos de alquileres, para obtener el nombre y apellidos de los clientes que han alquilado coches en enero, necesitamos usar o combinar dos tablas en la consulta: **clientes y contratos**.

En MySQL podemos usar las siguientes operaciones de combinación de tablas:

- ☐ Producto cartesiano o CROSS JOIN
- ☐ Combinación INNER JOIN
- ☐ Combinación LEFT JOIN
- ☐ Combinación RIGHT JOIN

Combinación de tablas

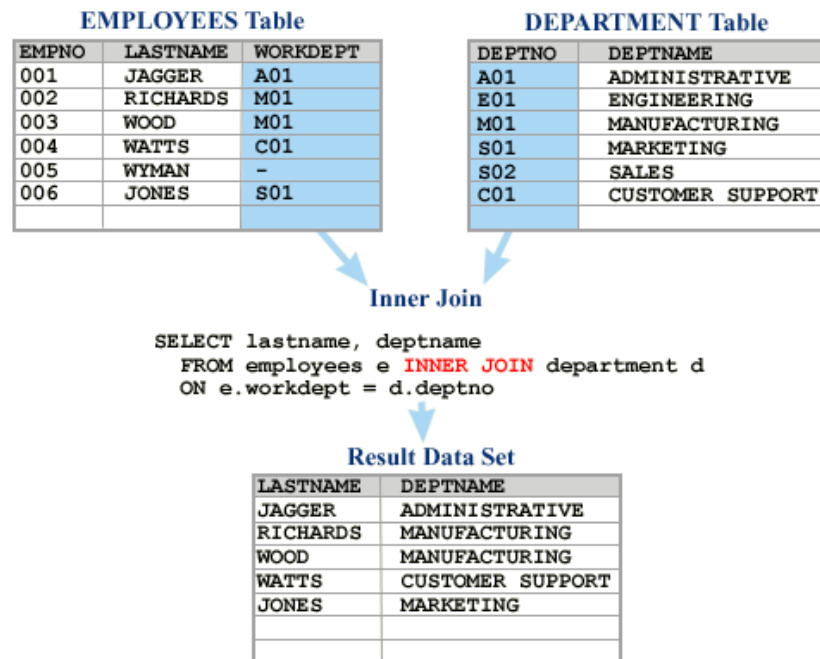
La reunión interna. INNER JOIN

Reunión Interna. INNER JOIN

Permite emparejar filas de dos tablas a través de una relación entre una columna de una tabla y otra columna de otra tabla.

Lo normal es que sean la clave principal de una tabla y la correspondiente clave ajena relacionada en la otra tabla, aunque pueden ser columnas que no tienen relación de clave ajena establecida.

En una consulta de este tipo, para cada fila de una de las tablas se busca en la otra tabla la fila o filas que cumplen la condición de relación que se quiera entre las dos columnas (normalmente se busca igualdad entre clave principal y clave ajena).



Reunión Interna. INNER JOIN

La sintaxis de esta operación dentro de una SELECT es:

```
SELECT ..... FROM tabla1 INNER JOIN tabla2 ON columna1  
condicion_relacion columna2
```

Tabla 1 y tabla 2 podrían ser incluso la misma tabla si hay alguna relación entre una columna de la tabla y la clave principal de la misma tabla. En este caso, al menos uno de los nombres de tabla tendría que ser un **alias**.

¿Por qué una de esos nombres debería ser un alias?

Columna1 y columna2 son las columnas que se emparejan o relacionan y deben tener el mismo tipo de datos o datos compatibles.

Condicion_relacion representa cualquier operación relacional, aunque normalmente se usa la igualdad. Se pueden combinar más de dos tablas usando varios INNER JOIN.

Reunión Interna. INNER JOIN

Sintaxis:

```
SELECT ..... FROM tabla1 INNER JOIN tabla2 ON columna1  
condicion_relacion columna2
```

Cuando coincida el nombre de las dos columnas relacionadas, tendremos que escribir nombres cualificados, escribiendo el nombre de la tabla a la que pertenecen, un punto y el nombre de la columna. En ese caso, también se puede usar y es más adecuada esta sintaxis:

```
SELECT ..... FROM tabla1 INNER JOIN tabla2 USING (columna);
```

Ejemplo: Obtener el número de contrato y la matrícula, marca y modelo de todos los automóviles que están contratados actualmente por algún cliente.

```
SELECT numcontrato, automoviles.matricula, marca, modelo FROM contratos  
INNER JOIN automoviles ON contratos.matricula=automoviles.matricula WHERE  
ffin IS NULL;
```

¿Por qué hemos escrito automoviles.matricula pero no automoviles.marca?

numcontrato	matricula	marca	modelo
18	2058JGF	Seat	Leon
19	3273JGH	Audi	A4
20	2123JTB	Renault	Megane
21	8795JTK	Mercedes	GLA

Reunión Interna. INNER JOIN

Ejemplo: Obtener el número de contrato y el nombre y apellidos de todos los clientes que tienen actualmente contrato algún automóvil.

SELECT numcontrato, nombre, apellidos **FROM** clientes **INNER JOIN** contratos **ON** dnicliente=dni **WHERE** ffin **IS NULL**;

¿Por qué ponemos **WHERE ffin IS NULL**?

numcontrato	nombre	apellidos
18	Fanny	Cepeda
19	Carlos Javier	Lopez Carvajal
20	Ismael	Poza Rincón
21	Alicia	de la Hoz Gomez

Reunión Interna. INNER JOIN

Ejemplo: De todos los contratos finalizados, obtener la matricula, marca y modelo de cada coche contratado, el nombre y apellidos del cliente que hizo cada contrato y los kilómetros recorridos por el coche en el contrato.

SELECT numcontrato, automoviles.matricula, marca, modelo, nombre, apellidos, kfin-kini **FROM** (contratos **INNER JOIN** automoviles **ON** contratos.matricula = automoviles.matricula) **INNER JOIN** clientes **ON** dnicliente=dni **WHERE** ffin **IS NOT NULL**;

numcontrato	matricula	marca	modelo	nombre	apellidos	kfin-kini
1	1234JMY	Mercedes	Clase C Coupe 170CV	Mariano	Dorado	361
2	7856JLD	BMW	318 TDI	Fanny	Cepeda	1000
3	5573JFS	Seat	Leon SW	Reyes	Sanz Lopez	2250
4	3273JGH	Audi	A4	Ismael	Poza Rincón	870
5	3765JSD	Seat	Ibiza	Ana Belén	Fuentes Rojas	833
6	1678JCN	Ford	Fiesta	Antonio	Díaz Vera	3328

Reunión Interna. INNER JOIN

Ejemplo: En una base de datos nba tenemos una tabla equipos. En la tabla equipos, entre otros datos, se tiene el nombre del equipo y la división en la que participa. Obtener todos los enfrentamientos o partidos posibles entre equipos de la división central sin usar la tabla partidos, buscando los distintos cruces.

```
SELECT a.nombre AS local, b.nombre AS visitante FROM equipos AS a INNER JOIN  
equipos AS b ON a.nombre <> b.nombre WHERE a.division='central' AND  
b.division='central';
```

local	visitante
Bulls	Bucks
Cavaliers	Bucks
Pacers	Bucks
Pistons	Bucks
Bucks	Bulls
Cavaliers	Bulls
Pacers	Bulls

Explicación: Tenemos una INNER JOIN entre dos tablas que son la misma. A la primera la renombramos como tabla a y la segunda como tabla b, pero las dos son equipos.

Ejemplo: Obtener el nombre y apellidos de los clientes que han contratado automóviles de la marca Seat.

```
SELECT DISTINCT nombre, apellidos FROM (contratos INNER  
JOIN automoviles ON contratos.matricula = automoviles.matricula) INNER JOIN  
clientes ON dncliente=dni WHERE marca='seat';
```

```
select nombre, apellidos from clientes inner join contratos as con on  
clientes.dni=con.dncliente inner join automoviles on  
con.matricula=automoviles.matricula where marca='seat';
```

nombre	apellidos
Anais	Rodriguez
Fanny	Cepeda
Ana Belén	Fuentes Roias
Reves	Sanz Lopez

*Si el uso de las operaciones de combinación de tablas permiten relacionar dos o más tablas por medio de diferentes **columnas independientemente de que estas sean claves primarias o foráneas...***

¿Por qué al diseñar una base de datos e implementar su diseño físico utilizamos las claves foráneas?

Motivos:

- 1. Integridad referencial. Al estar relacionadas las tuplas de dos tablas por medio de una FK, garantizamos la integridad de los datos. Ya que en caso de actualización o borrado estos serán tratados en consecuencia por medio de las restricciones correspondientes.***
- 2. Diseño estructural. Permiten una mayor claridad en la definición de la estructura de la base de datos. Facilitan conocer mejor las relaciones entre las tablas y ayudan a la hora de componer las consultas.***
- 3. Rendimiento. La rapidez al realizar las consultas aumenta, ya que una FK genera automáticamente un índice, por lo que las consultas serán más rápidas.***

Bases de Datos

Unidad 5: Realización de consultas Sesión 2

Producto cartesiano

El producto cartesiano de dos tablas permite obtener una tabla con las columnas de la primera tabla y las columnas de la segunda tabla (aunque tengan nombres iguales).

Las filas de la hoja de resultados resultante son todas las posibles combinaciones entre filas de la primera tabla y filas de la segunda tabla. Así, si una tabla tiene 6 filas y la otra tiene 8, el resultado del producto cartesiano es una tabla de 48 filas. Pero si una tabla tiene 6000 filas y otras 8000, se crea en memoria una tabla de 48 millones de filas, cada una de las cuales contiene varios bytes. Eso supone crear mucho espacio en memoria y puede ser un GRAVE PROBLEMA.

idfab	idproducto	fab	producto
bic	41672	aci	41004
imm	779c	aci	41002

COMPOSICION

idfab	idproducto	fab	producto
bic	41672	aci	41002
bic	41672	aci	41004
imm	779c	aci	41002
imm	779c	aci	41004

Producto cartesiano

Para obtener el producto cartesiano total entre dos tablas se escribe * (todas las columnas) después de SELECT y los nombres de las dos tablas separadas con coma después de FROM. No es muy normal tener que obtener el producto cartesiano total entre dos o más tablas. Lo normal es que sobre el resultado de un producto cartesiano apliquemos condiciones para extraer los datos combinados que queremos.

Por ejemplo, si ejecutamos:

SELECT * FROM automoviles, contratos;

matricula	marca	modelo	color	precio	kilometros	extras	alquilado	numcontrato	matricula	dnicliente	fini	ffin	kini	kfin
1234JMY	Mercedes	Clase C Coupe 170CV	Negro	165.78	22561	AUT,TS	0	1	1234JMY	08785691K	2016-12-12	2016-12-19	22200	22561
1678JCN	Ford	Fiesta	Verde	68.64	9500	AA,EE,CC,RC,ABS	0	1	1234JMY	08785691K	2016-12-12	2016-12-19	22200	22561
1732JBS	Seat	Leon	Negro	90.06	2500	TS,SN	0	1	1234JMY	08785691K	2016-12-12	2016-12-19	22200	22561
1978JNT	Opel	Corsa	Azul	42.7	45876		0	1	1234JMY	08785691K	2016-12-12	2016-12-19	22200	22561
2058JGF	Seat	Leon	Rojo	93.64	9736	GPS,SN	1	1	1234JMY	08785691K	2016-12-12	2016-12-19	22200	22561
2123JTB	Renault	Megane	Amarillo	92.65	34323	TS,SN	1	1	1234JMY	08785691K	2016-12-12	2016-12-19	22200	22561
3273JGH	Audi	A4	Rojo	124.2	17368	AUT,GPS,WIFI	1	1	1234JMY	08785691K	2016-12-12	2016-12-19	22200	22561
3765JSD	Seat	Ibiza	Rojo	70.56	7683	SN	0	1	1234JMY	08785691K	2016-12-12	2016-12-19	22200	22561
4387JDD	Citroen	C3	Verde	62.67	23057		0	1	1234JMY	08785691K	2016-12-12	2016-12-19	22200	22561
4738JBJ	Audi	A3	Amarillo	118.76	8008	GPS,WIFI,SN	0	1	1234JMY	08785691K	2016-12-12	2016-12-19	22200	22561
5031JHL	BMW	318 i	Azul	116.45	24796	GPS,WIFI,SN	0	1	1234JMY	08785691K	2016-12-12	2016-12-19	22200	22561
5573JFS	Seat	Leon SW	Gris	102.63	28500	AUT,GPS	1	1	1234JMY	08785691K	2016-12-12	2016-12-19	22200	22561
5678JRZ	Mercedes	Clase C	Blanco	123.65	7659	AUT,GPS,SN	0	1	1234JMY	08785691K	2016-12-12	2016-12-19	22200	22561
6761JYM	Renault	Clio	Blanco	53.62	25672	SN	0	1	1234JMY	08785691K	2016-12-12	2016-12-19	22200	22561
7839JDR	Ford	Focus	Blanco	87.62	15873	GPS,SN	1	1	1234JMY	08785691K	2016-12-12	2016-12-19	22200	22561
7856JLD	BMW	318 TDI	Azul	121.79	35978	TS,GPS,SN	0	1	1234JMY	08785691K	2016-12-12	2016-12-19	22200	22561

Producto cartesiano

matricula	marca	modelo	color	precio	kilometros	extras	alquilado	numcontrato	matricula	dncliente	fini	ffin	kini	kfin
1234JMY	Mercedes	Clase C Coupe 170CV	Negro	165.78	22561	AUT,TS	0	1	1234JMY	08785691K	2016-12-12	2016-12-19	22200	22561
1678JCN	Ford	Fiesta	Verde	68.64	9500	AA,EE,CC,RC,ABS	0	1	1234JMY	08785691K	2016-12-12	2016-12-19	22200	22561
1732JBS	Seat	Leon	Negro	90.06	2500	TS,SN	0	1	1234JMY	08785691K	2016-12-12	2016-12-19	22200	22561
1978JNT	Opel	Corsa	Azul	42.7	45876		0	1	1234JMY	08785691K	2016-12-12	2016-12-19	22200	22561
2058JGF	Seat	Leon	Rojo	93.64	9736	GPS,SN	1	1	1234JMY	08785691K	2016-12-12	2016-12-19	22200	22561
2123JTB	Renault	Megane	Amarillo	92.65	34323	TS,SN	1	1	1234JMY	08785691K	2016-12-12	2016-12-19	22200	22561
3273JGH	Audi	A4	Rojo	124.2	17368	AUT,GPS,WIFI	1	1	1234JMY	08785691K	2016-12-12	2016-12-19	22200	22561
3765JSD	Seat	Ibiza	Rojo	70.56	7683	SN	0	1	1234JMY	08785691K	2016-12-12	2016-12-19	22200	22561
4387JDD	Citroen	C3	Verde	62.67	23057		0	1	1234JMY	08785691K	2016-12-12	2016-12-19	22200	22561
4738JBJ	Audi	A3	Amarillo	118.76	8008	GPS,WIFI,SN	0	1	1234JMY	08785691K	2016-12-12	2016-12-19	22200	22561
5031JHL	BMW	318 i	Azul	116.45	24796	GPS,WIFI,SN	0	1	1234JMY	08785691K	2016-12-12	2016-12-19	22200	22561
5573JFS	Seat	Leon SW	Gris	102.63	28500	AUT,GPS	1	1	1234JMY	08785691K	2016-12-12	2016-12-19	22200	22561
5678JRZ	Mercedes	Clase C	Blanco	123.65	7659	AUT,GPS,SN	0	1	1234JMY	08785691K	2016-12-12	2016-12-19	22200	22561
6761JYM	Renault	Clio	Blanco	53.62	25672	SN	0	1	1234JMY	08785691K	2016-12-12	2016-12-19	22200	22561
7839JDR	Ford	Focus	Blanco	87.62	15873	GPS,SN	1	1	1234JMY	08785691K	2016-12-12	2016-12-19	22200	22561
7856JLD	BMW	318 TDI	Azul	121.79	35978	TS,GPS,SN	0	1	1234JMY	08785691K	2016-12-12	2016-12-19	22200	22561

Vemos que en el resultado vemos múltiples filas para el contrato 1. Vemos que ese contrato corresponde a la matrícula 12434JMY. Pero ese contrato se ha combinado con cada uno de los automóviles, es decir, se ha combinado con automóviles de matrículas distintas a las del contrato. Lo lógico es que, para cada contrato, relacionemos el contrato con los datos del automóvil que corresponda. Eso hay que hacerlo con una condición WHERE.

SELECT * FROM automoviles, contratos WHERE automoviles.matricula = contratos.matricula;

Aunque ahora veamos en el resultado una fila por cada contrato, realmente se ha creado en memoria una tabla intermedia con el producto cartesiano completo. 17

Producto cartesiano

Vamos a ver varios ejemplos en los que se realizan consultas en las que se puede usar el producto cartesiano. Se aplica la combinación producto cartesiano y se establecen condiciones de selección WHERE y se indican las columnas o expresiones que se quieren consultar.

Ejemplo: Obtener la matricula, marca, modelo, fecha inicial y fecha final del contrato número 1 (habrá que establecer la condición WHERE para que el contrato sea el número 1 y para que la matrícula del automóvil coincida con la del contrato).

```
SELECT contratos.matricula, marca, modelo, fini, ffin FROM automoviles,  
contratos WHERE numcontrato=1 AND  
contratos.matricula=automoviles.matricula;
```

matricula	marca	modelo	fini	ffin
1234JMY	Mercedes	Clase C Coupe 170CV	2016-12-12	2016-12-19

Vemos que en la instrucción se deben usar **cualificadores de tabla** en columnas que tienen el mismo nombre en las dos tablas.

Producto cartesiano

Cuando se combinan tablas puede ser útil a veces renombrarlas. Por ejemplo, la instrucción usada en la anterior diapositiva:

```
SELECT contratos.matricula, marca, modelo, fini, ffin FROM automoviles,  
contratos WHERE numcontrato=1 AND  
contratos.matricula=automoviles.matricula;
```

Sería equivalente a esta que usa renombrado de tablas:

```
SELECT c.matricula, marca, modelo, fini, ffin FROM automoviles AS a, contratos AS  
c WHERE numcontrato=1 AND c.matricula=a.matricula;
```

Producto cartesiano

Ejemplo: Suponiendo que tenemos en una base de datos una tabla con los módulos de un curso y otra tabla con los alumnos del curso, realizar una consulta que obtiene todas las posibles combinaciones de códigos o números de alumnos con todos los códigos de módulos del curso DAM1.

numalumn	nombre	apellidos	direccion	dni
001	Ana	Alonso	Alonso	11111111A
002	Beatriz	Bueno	Bueno	22222222B
003	Carlos	Campos	Campos	33333333C

codmodulo	nommodulo	horassemana	transversal
BD	Bases de ...	6	0
ED	Entornos ...	2	0
FOL	Formación...	3	1
LMSG	Lenguajes...	4	0
PRG	Programac...	8	0
SI	Sistemas I...	7	0

SELECT numalumn, codmodulo FROM alumnos, modulos ORDER BY codmodulo;

numalumn	codmodulo
001	BD
002	BD
003	BD
001	ED
002	ED
003	ED
001	FOL
002	FOL
003	FOL
001	LMSG

Producto cartesiano

Ejemplo: En la base de datos alquileres obtener la marca y modelo (sin repetir) de todos los automóviles contratados alguna vez en diciembre de 2017.

SELECT DISTINCT marca, modelo FROM automoviles, contratos WHERE automoviles.matricula=contratos.matricula AND fini LIKE '2017-12%';

marca	modelo
Mercedes	Clase C Coupe 170CV
BMW	318 TDI
Seat	Leon SW
Audi	A4
Seat	Ibiza
Ford	Fiesta
Renault	Clio
Opel	Corsa
Seat	Leon

Producto cartesiano

Ejemplo: Del contrato de alquiler de coches número 10, obtener el cliente que hizo el contrato, la matrícula, marca y modelo del coche y la duración del contrato.

SELECT apellidos, nombre, contratos.matricula, marca, modelo, fini, ffin **FROM** automoviles, contratos, clientes **WHERE** automoviles.matricula = contratos.matricula **AND** contratos.dnicliente=clientes.dni **AND** numcontrato=10;

apellidos	nombre	matricula	marca	modelo	fini	ffin
de la Hoz Gomez	Alicia	3273JGH	Audi	A4	2016-12-27	2017-01-02

Producto cartesiano

Ejemplo: En una base de datos nba tenemos una tabla equipos. En la tabla equipos, entre otros datos, se tiene el nombre del equipo y la división en la que participa. Obtener todos los enfrentamientos o partidos posibles entre equipos de la división central. Habrá que combinar la tabla equipos consigo misma evitando que el equipo local y el visitante sea el mismo.

SELECT a.nombre AS local, b.nombre AS visitante FROM equipos AS a, equipos AS b WHERE a.division='central' AND b.division='central' AND a.nombre <> b.nombre;

local	visitante
Bulls	Bucks
Cavaliers	Bucks
Pacers	Bucks
Pistons	Bucks
Bucks	Bulls
Cavaliers	Bulls
Pacers	Bulls
Pistons	Bulls
Bucks	Cavaliers
Bulls	Cavaliers

IMPORTANTE:

El producto cartesiano debe evitarse, siempre y cuando la consulta se pueda realizar con otra operación de combinación, cuando las tablas que se combinan tienen muchas filas.

La combinación produce el producto del número de filas combinadas y eso pueden ser muchísimas filas (y muchas columnas también). Todo eso se almacena temporalmente en RAM y ocupa mucho espacio.

Las consultas de los ejemplos de las diapositivas 7 y 10 son casos de una buena utilización del producto cartesiano ya que ahí si que queremos combinar todas las filas de una tabla con todas las de la otra.

Sin embargo, los otros ejemplos de este apartado se podrían realizar más óptimamente con otras operaciones de combinación.

Bases de Datos

Unidad 5: Realización de consultas Sesión 3

Combinación de tablas

Las reuniones externas.

LEFT JOIN

RIGHT JOIN

Reunión externa por la izquierda. LEFT JOIN

Permite emparejar filas de dos tablas a través de una relación entre una columna de una tabla y otra columna de otra tabla. Hasta aquí todo igual que INNER JOIN.

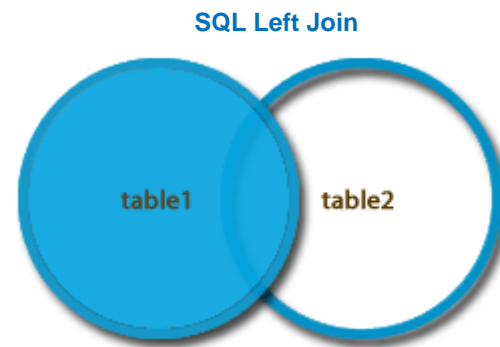
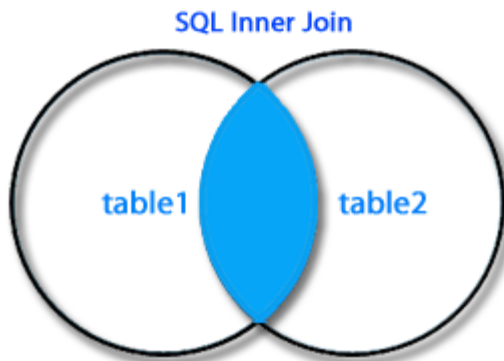
Además añade combinaciones de filas de la tabla de la izquierda con columnas vacías de la tabla de la derecha o a valores nulos para aquellas filas de la tabla de la izquierda que no tienen correspondencia con filas de la tabla de la derecha.

Por ejemplo, si se hace un **AUTOMOVILES LEFT JOIN CONTRATOS ON automoviles.matricula = contratos.matricula**

Para los automóviles que nunca han sido contratados, se generaría una fila con los datos del automóvil y todos los datos de contrato a valor NULL.

Sintaxis:

La misma que para INNER JOIN. Se pueden usar las cláusulas ON y USING.



Reunión externa por la izquierda. LEFT JOIN

*Ejemplo: Obtener la matrícula, marca y modelo de todos los automóviles junto con los datos de todos los contratos que se han realizado sobre esos automóviles. **Para los automóviles nunca contratados se debe obtener también una fila que no está relacionada con ningún contrato.***

SELECT automoviles.matricula, marca, modelo, contratos.* **FROM** automoviles **LEFT JOIN** contratos **USING** (matricula);

matricula	marca	modelo	numcontrato	matricula	dnidiente	fini	ffin	kini	kfin
1234JMY	Mercedes	Clase C Coupe 170CV	1	1234JMY	08785691K	2016-12-12	2016-12-19	22200	22561
1678JCN	Ford	Fiesta	6	1678JCN	24589635S	2016-12-22	2016-12-30	32650	35978
1732JBS	Seat	Leon	NULL	NULL	NULL	NULL	NULL	NULL	NULL
1978JNT	Opel	Corsa	8	1978JNT	13876715C	2016-12-25	2016-12-26	45650	45876
2058JGF	Seat	Leon	9	2058JGF	09856064L	2016-12-27	2016-12-30	8150	9736
2058JGF	Seat	Leon	18	2058JGf	07385709H	2017-01-08	NULL	9736	NULL
2123JTB	Renault	Megane	20	2123JTB	03549358G	2017-01-09	NULL	34323	NULL

Reunión externa por la izquierda. LEFT JOIN

Ejemplo: Obtener los datos de todos los automóviles que nunca han sido contratados.

SELECT automoviles.* FROM automoviles LEFT JOIN contratos USING (matricula) WHERE numcontrato IS NULL;

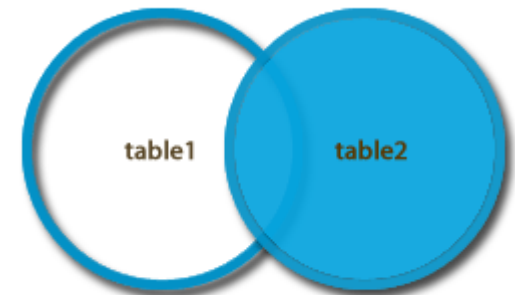
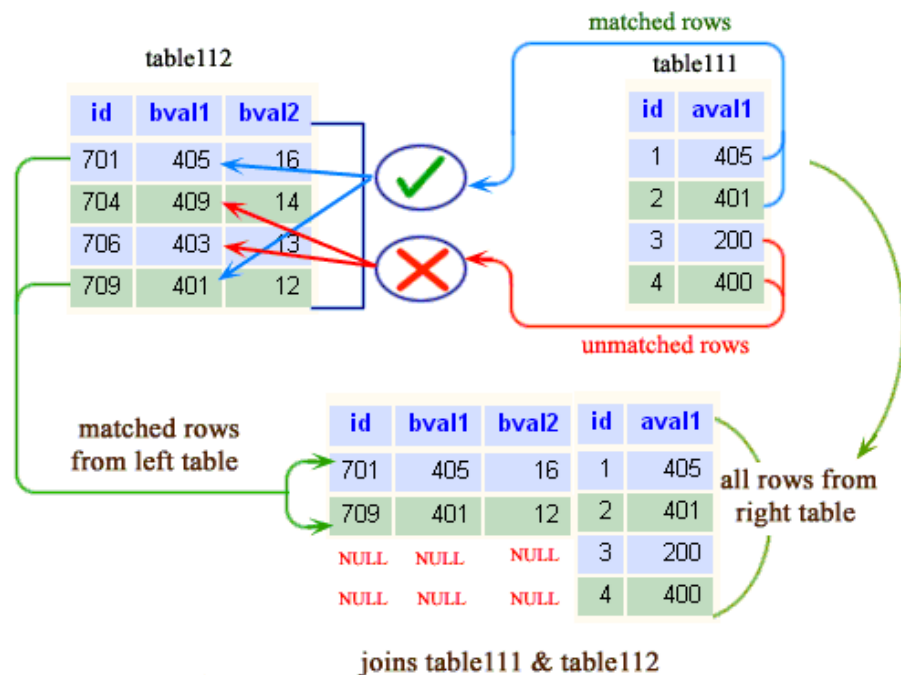
matricula	marca	modelo	color	precio	kilometros	extras	alquilado
1732JBS	Seat	Leon	Negro	90.06	2500	TS,SN	0
7839JDR	Ford	Focus	Blanco	87.62	15873	GPS,SN	1

Reunión externa por la derecha. RIGHT JOIN

Permite emparejar filas de dos tablas a través de una relación entre una columna de una tabla y otra columna de otra tabla. Hasta aquí todo igual que INNER JOIN.

Además añade combinaciones de filas de la tabla de la derecha con columnas vacías de la tabla de la izquierda o a valores nulos para aquellas filas de la tabla de la derecha que no tienen correspondencia con filas de la tabla de la izquierda.

MySQL RIGHT JOIN



Reunión externa por la derecha. RIGHT JOIN

Ejemplo: Obtener el DNI, nombre y apellidos de todos los clientes registrados junto con los datos de los contratos que han realizado. En la hoja de resultados se deben mostrar también los clientes que no han realizado ningún contrato.

SELECT clientes.dni, nombre, apellidos, contratos.* FROM contratos RIGHT JOIN clientes ON dni=dnicliente;

dni	nombre	apellidos	numcontrato	matricula	dnicliente	fini	ffin	kini	kfin
00371569B	Beatriz	Garcia Martin	7	6761JYM	00371569B	2016-12-24	2016-12-30	21500	25672
00445760C	Sandra	Flores Jorje	NULL	NULL	NULL	NULL	NULL	NULL	NULL
00740365D	Carlos Javier	Lopez Carvajal	12	8795JTK	00740365D	2017-01-06	2017-01-10	44850	46980
00740365D	Carlos Javier	Lopez Carvajal	19	3273JGH	00740365D	2017-01-09	NULL	17368	NULL
02748375F	Vanessa	Rodriguez	15	4738JBJ	02748375F	2017-01-08	2017-01-12	7965	8008
03549358G	Ismael	Poza Rincón	4	3273JGH	03549358G	2016-12-15	2016-12-22	15380	16250
03549358G	Ismael	Poza Rincón	20	2123JTB	03549358G	2017-01-09	NULL	34323	NULL

Reunión externa por la izquierda. RIGHT JOIN

Ejemplo: Obtener los datos de todos los clientes que nunca han hecho contratos.

SELECT clientes.* FROM contratos RIGHT JOIN clientes ON dni=dnicliente WHERE numcontrato IS NULL;

dni	apellidos	nombre	direccion	localidad	carnet	fcarnet	fnac
00445760C	Flores Jorge	Sandra	C/ La Cañada 28	Madrid	B	1995-10-04	1974-06-02
11223344M	Garcia Garcia	Noelia	C/ Talavera 19	Toledo	B	2001-10-30	1982-03-13
28759595T	Ruiz Alonso	Ricardo	C/ Hortaleza, 56	Madrid	B	1999-10-13	1977-11-13
43809540X	Montoya	Natalia	C/ Verona, 3	Toledo	B	2005-10-14	1986-09-15

Otros tipos de reuniones

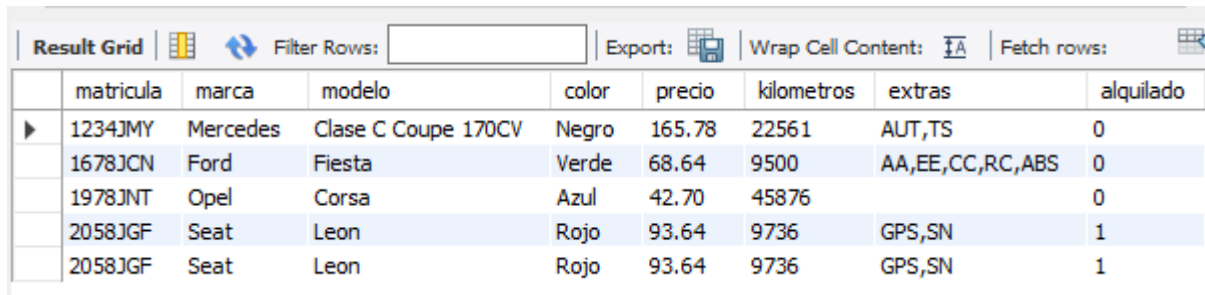
NATURAL JOIN

Permite combinar filas de dos tablas relacionadas por igualdad entre dos campos. Las condiciones son:

- Las tablas asociadas tienen uno o más pares de columnas con nombres idénticos.
- Las columnas deben ser del mismo tipo de datos.
- No se usa la cláusula ON.

Ejemplo: Obtener los datos de los automóviles que han realizado los cinco primeros contratos.

SELECT automoviles.* FROM automoviles NATURAL JOIN contratos LIMIT 5;



	matricula	marca	modelo	color	precio	kilometros	extras	alquilado
▶	1234JMY	Mercedes	Clase C Coupe 170CV	Negro	165.78	22561	AUT,TS	0
	1678JCN	Ford	Fiesta	Verde	68.64	9500	AA,EE,CC,RC,ABS	0
	1978JNT	Opel	Corsa	Azul	42.70	45876		0
	2058JGF	Seat	Leon	Rojo	93.64	9736	GPS,SN	1
	2058JGF	Seat	Leon	Rojo	93.64	9736	GPS,SN	1

¿Por medio de qué columna se están cruzando las tablas?

STRAIGHT_JOIN

Puede usarse con las cláusulas ON y USING para realizar lo mismo que INNER JOIN. Sin esas cláusulas realiza lo mismo que NATURAL JOIN.