

Ejemplo de consultas Multitabla.

Creamos una base de datos llamada PersonaCoche de esta forma:

```
DROP DATABASE IF EXISTS PersonaCoche;
CREATE DATABASE PersonaCoche;
USE PersonaCoche;

CREATE TABLE Persona (
  IdPersona INTEGER NOT NULL,
  Nombre VARCHAR( 30 ) NOT NULL,
  PRIMARY KEY ( IdPersona )
);

CREATE TABLE Coche (
  IdCoche INTEGER NOT NULL,
  Modelo VARCHAR( 30 ) NOT NULL,
  IdPersona INTEGER,
  PRIMARY KEY ( IdCoche ),
  FOREIGN KEY ( IdPersona ) REFERENCES Persona( IdPersona )
);

INSERT INTO Persona VALUES ( 1, 'Sancho' ), ( 2, 'María' ), ( 3, 'Franciso' ), ( 4, 'Laura' );
INSERT INTO Coche VALUES ( 1, 'Audi', 1 ), ( 2, 'Toyota', NULL ), ( 3, 'Mercedes', 2 ), ( 4, 'BMW', NULL );
```

Unión por campo común: NATURAL JOIN

```
# -----
# Unión de dos tablas a través del campo común implícito (IdPersona) que
# debe existir en ambas tablas, pero NO aparecerá duplicado.
# -----
SELECT * FROM Persona NATURAL JOIN Coche;
```

	IdPersona	Nombre	IdCoche	Modelo
►	1	Sancho	1	Audi
	2	María	3	Mercedes

Unión por campo especificado: INNER JOIN

```
# -----
# Unión de dos tablas a través del campo común especificado (IdPersona)
# que aparece duplicado.
# -----
SELECT * FROM Persona INNER JOIN Coche
      ON Persona.IdPersona = Coche.IdPersona;
```

	IdPersona	Nombre	IdCoche	Modelo	IdPersona
►	1	Sancho	1	Audi	1
	2	María	3	Mercedes	2

Unión entre todas las tuplas: CROSS JOIN

```
# -----
# Producto cartesiano de dos tablas, tengan o no algún campo común: si NO existe
# relación entre alguna tupla de las dos relaciones, aparecerá NULL en la CLAVE FORÁNEA.
# -----
SELECT * FROM Persona CROSS JOIN Coche;
```

	IdPersona	Nombre	IdCoche	Modelo	IdPersona
▶	1	Sancho	1	Audi	1
	2	María	1	Audi	1
	3	Franciso	1	Audi	1
	4	Laura	1	Audi	1
	1	Sancho	2	Toyota	NULL
	2	María	2	Toyota	NULL
	3	Franciso	2	Toyota	NULL
	4	Laura	2	Toyota	NULL
	1	Sancho	3	Mercedes	2
	2	María	3	Mercedes	2
	3	Franciso	3	Mercedes	2
	4	Laura	3	Mercedes	2
	1	Sancho	4	BMW	NULL
	2	María	4	BMW	NULL
	3	Franciso	4	BMW	NULL
	4	Laura	4	BMW	NULL

Unión con tabla derecha preferente: RIGHT OUTER JOIN

```
# -----
# Las tuplas de la tabla de la derecha (Coche) se muestran asociadas a las tuplas de
# la izquierda (Persona) según la CLAVE FORÁNEA, si:
#   - Toma un valor de la otra tabla -> muestra la tupla correspondiente.
#   - Toma un valor NULO -> muestra NULL en los valores asociados a la tabla
# -----
SELECT * FROM Persona RIGHT OUTER JOIN Coche
ON Persona.IdPersona = Coche.IdPersona;
```

	IdPersona	Nombre	IdCoche	Modelo	IdPersona
▶	1	Sancho	1	Audi	1
	NULL	NULL	2	Toyota	NULL
	2	María	3	Mercedes	2
	NULL	NULL	4	BMW	NULL

Unión con tabla izquierda preferente: LEFT OUTER JOIN

```
# -----
# Las tuplas de la tabla de la izquierda (Persona) se muestran asociadas a las tuplas de
# la derecha (Coche) según la CLAVE FORÁNEA, si:
#   - Toma un valor de la otra tabla -> muestra la tupla correspondiente.
#   - Toma un valor NULO -> muestra NULL en los valores asociados a la tabla
# -----
SELECT * FROM Persona LEFT OUTER JOIN Coche
      ON Persona.IdPersona = Coche.IdPersona;
```

	IdPersona	Nombre	IdCoche	Modelo	IdPersona
▶	1	Sancho	1	Audi	1
	2	María	3	Mercedes	2
	3	Franciso	NULL	NULL	NULL
	4	Laura	NULL	NULL	NULL

Unión de las tablas con o sin campos en común: FULL OUTER JOIN

MySQL no implementa esta característica, pero sí se puede construir a través de una unión entre RIGHT OUTER JOIN y LEFT OUTER JOIN.

```
# -----
# Muestra las relaciones existentes entre las tuplas de cada tabla y si no tiene
# muestra NULL en los valores asociados a la otra tabla: en ambos sentidos.
# -----
SELECT * FROM Persona RIGHT OUTER JOIN Coche ON Persona.IdPersona = Coche.IdPersona
      UNION
SELECT * FROM Persona LEFT OUTER JOIN Coche ON Persona.IdPersona = Coche.IdPersona;
```

	IdPersona	Nombre	IdCoche	Modelo	IdPersona
▶	1	Sancho	1	Audi	1
	NULL	NULL	2	Toyota	NULL
	2	María	3	Mercedes	2
	NULL	NULL	4	BMW	NULL
	3	Franciso	NULL	NULL	NULL
	4	Laura	NULL	NULL	NULL