Practica 8 MYSQL

1.- Crea la base de datos **P8** y en ella la tabla **Usuarios**:

```
MariaDB [(none)]> CREATE DATABASE ejercicio8_p8;
Query OK, 1 row affected (0.00 sec)
```

```
CREATE TABLE usuarios(
        id Smallint (5) unsigned not NULL AUTO_INCREMENT,
        Nombre varchar (30) NOT NULL,
        curso Smallint (5) UNSIGNED DEFAULT NULL,
        PRIMARY KEY (id)
) ENGINE = InnoDB;
```

```
MariaDB [ejercicio8_p8]> CREATE TABLE usuarios(
    ->      id SMALLINT(5) UNSIGNED NOT NULL AUTO_INCREMENT,
    ->      nombre VARCHAR(30) NOT NULL,
    ->      curso SMALLINT(5) UNSIGNED DEFAULT NULL,
    ->      PRIMARY KEY(id)
    -> );
Query OK, 0 rows affected (0.03 sec)
```

Introduce los datos:

id	nombre	curso
1	Paula	1
2	Silvia	1
3	Carlos	2
4	Ruth	5
5	José Antonio	(NULL)

```
MariaDB [ejercicio8_p8]> INSERT INTO usuarios(id,nombre,curso)
    ->      VALUES(1,"Paula",1),
    ->          (2,"Silvia",1),
    ->          (3,"Carlos",2),
    ->          (4,"Ruth",5),
    ->          (5,"José Antonio",NULL);
Query OK, 5 rows affected (0.01 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

2.- Crea la tabla **Cursos**:

```
CREATE TABLE cursos (
        id smallint(5) unsigned NOT NULL AUTO_INCREMENT,
        nombre varchar(50) NOT NULL,
        PRIMARY KEY (id)
) ENGINE=InnoDB;
```

```
MariaDB [ejercicio8_p8]> CREATE TABLE cursos(
    ->     id SMALLINT(5) UNSIGNED NOT NULL AUTO_INCREMENT,
    ->  nombre VARCHAR(50) NOT NULL,
    ->  PRIMARY KEY (id)
    -> );
Query OK, 0 rows affected (0.03 sec)
```

Introduce los datos:

id	nombre
1	HTML5
2	CSS3
3	JavaScript
4	PHP
5	MySQL

```
MariaDB [ejercicio8_p8]> INSERT INTO cursos(id,nombre)
    ->     VALUES(1,"HTML5"),
    ->         (2,"CSS"),
    ->         (3,"JavaScript"),
    ->         (4,"PHP"),
    ->         (5,"MySQL");
Query OK, 5 rows affected (0.00 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

3.- Añadimos una clave ajena para relacionar los usuarios con los curos que realizan:

```
ALTER TABLE usuarios
ADD FOREIGN KEY(curso) REFERENCES cursos(id)
ON UPDATE CASCADE;
```

```
MariaDB [ejercicio8_p8]> ALTER TABLE usuarios ADD FOREIGN KEY(curso) REFERENCES cursos(id) ON UPDATE CASCADE;
Query OK, 5 rows affected (0.07 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

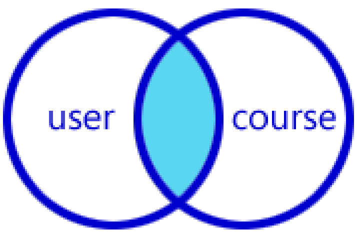
Comprueba que todo está correcto con **show create table**.

```
MariaDB [ejercicio8_p8]> SHOW CREATE TABLE usuarios;
+----------+----------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------
-------------------------------------------------------+
| Table    | Create Table

                                                       |
+----------+----------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------
-------------------------------------------------------+
| usuarios | CREATE TABLE `usuarios` (
  `id` smallint(5) unsigned NOT NULL AUTO_INCREMENT,
  `nombre` varchar(30) NOT NULL,
  `curso` smallint(5) unsigned DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `curso` (`curso`),
  CONSTRAINT `usuarios_ibfk_1` FOREIGN KEY (`curso`) REFERENCES `cursos` (`id`) ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=latin1 |
+----------+----------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------
-------------------------------------------------------+
1 row in set (0.00 sec)
```

```
MariaDB [ejercicio8_p8]> SHOW CREATE TABLE cursos;
+--------+--------------------------------------------------
------------------------------------------+
| Table  | Create Table
                                          |
+--------+--------------------------------------------------
------------------------------------------+
| cursos | CREATE TABLE `cursos` (
  `id` smallint(5) unsigned NOT NULL AUTO_INCREMENT,
  `nombre` varchar(50) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=latin1 |
+--------+--------------------------------------------------
------------------------------------------+
1 row in set (0.00 sec)
```

INNER JOIN



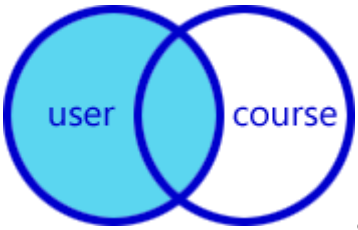
La cláusula se utiliza con más frecuencia es INNER JOIN. Esto produce un conjunto de registros que coinciden tanto en las tablas de usuario y cursos, es decir, todos los usuarios que están inscritos en un curso:

4.- Prueba y escribe el resultado:

```
SELECT usuarios.nombre,cursos.nombre
FROM usuarios INNER JOIN cursos on usuarios.curso=cursos.id;
```

```
MariaDB [ejercicio8_p8]> SELECT usuarios.nombre,cursos.nombre FROM usuarios INNER JOIN cursos on usuarios.curso=cursos.id;
+--------+--------+
| nombre | nombre |
+--------+--------+
| Paula  | HTML5  |
| Silvia | HTML5  |
| Carlos | CSS    |
| Ruth   | MySQL  |
+--------+--------+
4 rows in set (0.01 sec)
```

LEFT JOIN



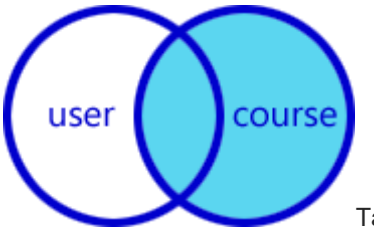
¿Qué pasa si se requiere una lista de todos los estudiantes y sus cursos, incluso si no están inscritos en uno? A LEFT JOIN produce un conjunto de registros que coincide con cada entrada en la tabla de la izquierda (por el usuario), independientemente de cualquier entrada coincidente en la tabla a la derecha.

5.- Prueba y escribe el resultado:

```
SELECT usuarios.nombre,cursos.nombre
FROM usuarios LEFT JOIN cursos on usuarios.curso=cursos.id;
```

```
MariaDB [ejercicio8_p8]> SELECT usuarios.nombre,cursos.nombre FROM usuarios LEFT JOIN cursos on usuarios.curso=cursos.id;
+--------------+--------+
| nombre       | nombre |
+--------------+--------+
| Paula        | HTML5  |
| Silvia       | HTML5  |
| Carlos       | CSS    |
| Ruth         | MySQL  |
| José Antonio | NULL   |
+--------------+--------+
5 rows in set (0.00 sec)
```

RIGHT JOIN



Tal vez necesitamos una lista de todos los cursos y los estudiantes, incluso si nadie se ha inscrito? A RIGHT JOIN produce un conjunto de registros que coincide con cada entrada en la tabla de la derecha (por supuesto), independientemente de cualquier entrada coincidente en la tabla de la izquierda.

6.- Prueba y escribe el resultado:

```
SELECT usuarios.nombre,cursos.nombre
FROM usuarios RIGHT JOIN cursos on usuarios.curso=cursos.id;
```

```
MariaDB [ejercicio8_p8]> SELECT usuarios.nombre,cursos.nombre FROM usuarios RIGHT JOIN cursos on usuarios.curso=cursos.id;
+--------+------------+
| nombre | nombre     |
+--------+------------+
| Paula  | HTML5      |
| Silvia | HTML5      |
| Carlos | CSS        |
| Ruth   | MySQL      |
| NULL   | JavaScript |
| NULL   | PHP        |
+--------+------------+
6 rows in set (0.00 sec)
```

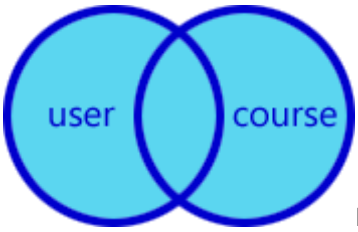
RIGHT JOIN rara vez se utilizan ya que se puede expresar el mismo resultado utilizando un LEFT JOIN. Esto puede ser más eficiente y más rápido para la base de datos para analizar:

```
SELECT usuarios.nombre,cursos.nombre
FROM cursos LEFT JOIN usuarios on usuarios.curso=cursos.id;
```

```
MariaDB [ejercicio8_p8]> SELECT usuarios.nombre,cursos.nombre FROM cursos LEFT JOIN usuarios on usuarios.curso=cursos.id;
+--------+------------+
| nombre | nombre     |
+--------+------------+
| Paula  | HTML5      |
| Silvia | HTML5      |
| Carlos | CSS        |
| Ruth   | MySQL      |
| NULL   | JavaScript |
| NULL   | PHP        |
+--------+------------+
6 rows in set (0.00 sec)
```

7.- Contar el número de alumnos matriculados en cada curso, usando LEFT JOIN.:

OUTER JOIN (o FULL OUTER JOIN)



Nuestra última opción es la OUTER JOIN que devuelve todos los registros en ambas tablas, independientemente de cualquier partido. Cuando no exista una coincidencia, el lado que falta contendrá NULL.

OUTER JOIN es menos útil que INNER, LEFT o RIGHT y no se implementó en MySQL. Sin embargo, puede evitar esta limitación mediante la unión de un LEFT y RIGHT JOIN, por ejemplo:

```
SELECT usuarios.nombre, cursos.nombre
FROM usuarios
LEFT JOIN cursos on usuarios.curso = cursos.id
UNION
SELECT usuarios.nombre, cursos.nombre
FROM usuarios
RIGHT JOIN cursos on usuarios.curso = cursos.id;
```

```
MariaDB [ejercicio8_p8]> SELECT usuarios.nombre,cursos.nombre FROM usuarios LEFT JOIN cursos on usuarios.curso=cursos.id
    ->     UNION SELECT usuarios.nombre,cursos.nombre FROM usuarios RIGHT JOIN cursos on usuarios.curso=cursos.id;
+--------------+------------+
| nombre       | nombre     |
+--------------+------------+
| Paula        | HTML5      |
| Silvia       | HTML5      |
| Carlos       | CSS        |
| Ruth         | MySQL      |
| José Antonio | NULL       |
| NULL         | JavaScript |
| NULL         | PHP        |
+--------------+------------+
7 rows in set (0.00 sec)
```

8.- Haz una copia de seguridad de la base de datos **P8**. Luego Borra la base de datos y restaura la copia efectuada.

```
Victor@V22 ~ > mysqldump -u root -p ejercicio8_p8 > "C:\\Users\\Usuario\\DAM.git\\DAM\\Bases de datos\\Unidad2-SQL\\Ejercicios\\Ejercicio8_p8_backup.sql"
Enter password:
Victor@V22 ~ >
```

```
MariaDB [(none)]> DROP DATABASE ejercicio8_p8;
Query OK, 2 rows affected (0.08 sec)
```

```
MariaDB [ejercicio8_p8]> SOURCE C:/Users/Usuario/DAM.git/DAM/Bases de datos/Unidad2-SQL/Ejercicios/Ejercicio8_p8_backup.sql
Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)
```