

Paso 1: Crear la Base de Datos.

Empiezo el proceso generando el container de Docker en donde estaré trabajando.

```
PS C:\Users\Vespertino> docker run --name ejercicioSQL -e MYSQL_DATABASE=BaseDeSQL -e MYSQL_ROOT_PASSWORD=contra -v MySQL_volume:/var/lib/mysql_volumen -d mysql
```

Con el comando `docker run` ejecuto el nuevo contenedor, cuyo nombre será “ejercicioSQL” como indico con el `--name`. Con el comando `-e MYSQL_DATABASE` genero una nueva base de datos con el nombre “BaseDeSQL” y con el `-e MYSQL_ROOT_PASSWORD` establezco que la contraseña del usuario root será “contra”. Por último, con `-v MySQL_volume` indico el volumen donde lo monto, con `-d` ejecuto el contenedor en segundo plano (modo deamon) y con `mysql` especifico la imagen que emplearé.

```
PS C:\Users\Vespertino> docker exec -it ejercicioSQL /bin/bash
bash-5.1#
```

Una vez el contenedor está montado, accedo a él empleando `docker exec`. Con `-it` hago que la terminal sea interactiva y `/bin/bash` será el comando que se ejecutará dentro del contenedor.

```
PS C:\Users\Vespertino> docker exec -it ejercicioSQL /bin/bash
bash-5.1# mysql --user=root --password BaseDeSQL
Enter password:
```

Por último, con el comando `mysql` inicio el cliente de MySQL para interactuar con la base de datos, indico que el usuario es el root y con el `--password` solicito que al momento de entrar me exija la contraseña anteriormente establecida.

Paso 2: Generar las tablas de la Base de Datos.

Estas son las tablas que estaré empleando.

```
199
200 CREATE TABLE Estudiantes (
201     EstudiantesID INT AUTO_INCREMENT PRIMARY KEY,
202     Nombre VARCHAR(255) NOT NULL,
203     Apellidos VARCHAR(255),
204     FechaNacimiento DATETIME,
205     DNI VARCHAR(9),
206 );
207
208 CREATE TABLE Cursos (
209     CursoID INT AUTO_INCREMENT PRIMARY KEY,
210     NombreCurso VARCHAR(255) NOT NULL,
211     Creditos INT,
212     Duracion INT,
213     DepartamentoID INT,
214     ProfesorID INT,
215     FOREIGN KEY (DepartamentoID) REFERENCES Departamentos(DepartamentoID) ON DELETE CASCADE,
216     FOREIGN KEY (ProfesorID) REFERENCES Profesores(ProfesorID) ON DELETE CASCADE
217 );
218
219 CREATE TABLE Profesores (
220     CursoID INT AUTO_INCREMENT PRIMARY KEY,
221     Nombre VARCHAR(255) NOT NULL,
222     Apellidos VARCHAR(255),
223     FechaNacimiento DATETIME,
224     DNI VARCHAR(9),
225     DepartamentoID INT,
226     FOREIGN KEY (DepartamentoID) REFERENCES Departamentos(DepartamentoID) ON DELETE CASCADE,
227 );
228
229 CREATE TABLE Departamentos (
230     DepartamentoID INT AUTO_INCREMENT PRIMARY KEY,
231     DepartamentoName VARCHAR(255) NOT NULL,
232 );
233
234 CREATE TABLE Matriculacion (
235     MatriculacionID INT AUTO_INCREMENT PRIMARY KEY,
236     EstudianteID INT,
237     CursoID INT,
238     FOREIGN KEY (EstudianteID) REFERENCES Estudiantes(EstudianteID) ON DELETE CASCADE,
239     FOREIGN KEY (CursoID) REFERENCES Cursos(CursoID) ON DELETE CASCADE
240 );
241
```

El modelo está totalmente basado en el de la tarea anterior, las tabla **Cursos** tiene la Foreign Key del departamento al que le pertenece y el profesor que la imparte, la tabla **Profesores** tiene indicado a que departamento pertenece y la tabla **Matriculación** funciona como tabla pivote entre Alumnos y Cursos.

Paso 3: Insertar/Modificar/Borrar datos en las tablas.

A continuación, relleno con datos las tablas con contenido para poder realizar las consultas.

Tabla Alumnos:

```
INSERT INTO Estudiantes (Nombre, Apellidos, FechaNacimiento, DNI) values ("Gustavo Adolfo","Becquer",NOW(),"14453213B");
INSERT INTO Estudiantes (Nombre, Apellidos, FechaNacimiento, DNI) values ("Luis","Cernuda",NOW(),"19482634F");
INSERT INTO Estudiantes (Nombre, Apellidos, FechaNacimiento, DNI) values ("Federico Garcia","Lorca",NOW(),"34542365R");
INSERT INTO Estudiantes (Nombre, Apellidos, FechaNacimiento, DNI) values ("José","de Espronceda",NOW(),"22749333P");
INSERT INTO Estudiantes (Nombre, Apellidos, FechaNacimiento, DNI) values ("Pio","Baroja",NOW(),"19482634F");
INSERT INTO Estudiantes (Nombre, FechaNacimiento, DNI) values ("Azorin",NOW(),"128745090");
INSERT INTO Estudiantes (Nombre, Apellidos, FechaNacimiento, DNI) values ("Antonio","Machado",NOW(),"12759384L");
INSERT INTO Estudiantes (Nombre, Apellidos, FechaNacimiento, DNI) values ("Miguel","de Unamuno",NOW(),"76451234U");
INSERT INTO Estudiantes (Nombre, Apellidos, FechaNacimiento, DNI) values ("Miguel","de Cervantes",NOW(),"97652938Y");
INSERT INTO Estudiantes (Nombre, Apellidos, FechaNacimiento, DNI) values ("Francisco","de Quevedo",NOW(),"12749502D");
INSERT INTO Estudiantes (Nombre, Apellidos, FechaNacimiento, DNI) values ("Lope","de Vega",NOW(),"34232276E");
```

Tabla Profesores:

```
INSERT INTO Profesores (Nombre, Apellidos, FechaNacimiento, DNI, DepartamentoID) values ("Ludwig","van Beethoven",NOW(),"33231298I", 1);
INSERT INTO Profesores (Nombre, Apellidos, FechaNacimiento, DNI, DepartamentoID) values ("Johann Sebastian","Mastropiero",NOW(),"76789345C", 2);
INSERT INTO Profesores (Nombre, FechaNacimiento, DNI, DepartamentoID) values ("Vivaldi",NOW(),"981243130", 3);
INSERT INTO Profesores (Nombre, Apellidos, FechaNacimiento, DNI, DepartamentoID) values ("Wolfgang Amadeus","Mozart",NOW(),"87462912D", 4);
INSERT INTO Profesores (Nombre, Apellidos, FechaNacimiento, DNI, DepartamentoID) values ("Paul","McCartney",NOW(),"1223634Y", 3);
INSERT INTO Profesores (Nombre, Apellidos, FechaNacimiento, DNI, DepartamentoID) values ("Claude","Debussy",NOW(),"77123129S", 7);
INSERT INTO Profesores (Nombre, Apellidos, FechaNacimiento, DNI, DepartamentoID) values ("Frederic","Chopin",NOW(),"14365488M", 8);
INSERT INTO Profesores (Nombre, Apellidos, FechaNacimiento, DNI, DepartamentoID) values ("Hans","Zimmer",NOW(),"34578323R", 8);
INSERT INTO Profesores (Nombre, Apellidos, FechaNacimiento, DNI, DepartamentoID) values ("Julio","Iglesias",NOW(),"23546598G", 9);
INSERT INTO Profesores (Nombre, FechaNacimiento, DNI, DepartamentoID) values ("Raphael",NOW(),"13525121E", 10);
```

Tabla Departamentos:

```
INSERT INTO Departamentos (DepartamentoName) values ("Departamento de Ciencias");
INSERT INTO Departamentos (DepartamentoName) values ("Departamento de Letras");
INSERT INTO Departamentos (DepartamentoName) values ("Departamento de Informatica");
INSERT INTO Departamentos (DepartamentoName) values ("Departamento de Sociales");
INSERT INTO Departamentos (DepartamentoName) values ("Departamento de Idiomas");
INSERT INTO Departamentos (DepartamentoName) values ("Departamento de Economía");
INSERT INTO Departamentos (DepartamentoName) values ("Departamento de Filosofía");
INSERT INTO Departamentos (DepartamentoName) values ("Departamento de Artes");
INSERT INTO Departamentos (DepartamentoName) values ("Departamento de Deportes");
INSERT INTO Departamentos (DepartamentoName) values ("Departamento de Extraescolares");
```

Tabla Cursos:

```
INSERT INTO Cursos (NombreCurso, Creditos, Duracion, DepartamentoID, ProfesorID) values ("Curso de Poesía Avanzada",30,250,2,1);
INSERT INTO Cursos (NombreCurso, Creditos, Duracion, DepartamentoID, ProfesorID) values ("Curso de Armónica y Flauta de Pan",120,300,8,2);
INSERT INTO Cursos (NombreCurso, Creditos, Duracion, DepartamentoID, ProfesorID) values ("Curso de Locucion",70,150,5,5);
INSERT INTO Cursos (NombreCurso, Creditos, Duracion, DepartamentoID, ProfesorID) values ("Curso de Matemáticas Analíticas",300,12,6,5);
INSERT INTO Cursos (NombreCurso, Creditos, Duracion, DepartamentoID, ProfesorID) values ("Curso de Gallego",300,50,6,5);
INSERT INTO Cursos (NombreCurso, Creditos, Duracion, DepartamentoID, ProfesorID) values ("Curso de Preparacion de Masas",50,15,10,6);
INSERT INTO Cursos (NombreCurso, Creditos, Duracion, DepartamentoID, ProfesorID) values ("Curso de Artes Marciales Aplicadas",100,40,9,7);
INSERT INTO Cursos (NombreCurso, Creditos, Duracion, DepartamentoID, ProfesorID) values ("Curso de MongoDB",60,45,3,7);
INSERT INTO Cursos (NombreCurso, Creditos, Duracion, DepartamentoID, ProfesorID) values ("Curso de Baile de Claqué",40,20,8,10);
INSERT INTO Cursos (NombreCurso, Creditos, Duracion, DepartamentoID, ProfesorID) values ("Curso de Debate",100,70,4,6);
INSERT INTO Cursos (NombreCurso, Creditos, Duracion, DepartamentoID, ProfesorID) values ("Curso de Globoflexia",20,25,8,2);
INSERT INTO Cursos (NombreCurso, Creditos, Duracion, DepartamentoID, ProfesorID) values ("Curso de Esgrima",90,55,9,9);
INSERT INTO Cursos (NombreCurso, Creditos, Duracion, DepartamentoID, ProfesorID) values ("Curso de Fabricacion de Titeres",30,200,8,8);
```

Tabla Matriculación:

```
INSERT INTO Matriculacion (EstudianteID, CursoID) values (1,3);
INSERT INTO Matriculacion (EstudianteID, CursoID) values (1,1);
INSERT INTO Matriculacion (EstudianteID, CursoID) values (2,2);
INSERT INTO Matriculacion (EstudianteID, CursoID) values (2,3);
INSERT INTO Matriculacion (EstudianteID, CursoID) values (2,5);
INSERT INTO Matriculacion (EstudianteID, CursoID) values (3,6);
INSERT INTO Matriculacion (EstudianteID, CursoID) values (4,7);
INSERT INTO Matriculacion (EstudianteID, CursoID) values (4,6);
INSERT INTO Matriculacion (EstudianteID, CursoID) values (4,9);
INSERT INTO Matriculacion (EstudianteID, CursoID) values (4,10);
INSERT INTO Matriculacion (EstudianteID, CursoID) values (6,1);
INSERT INTO Matriculacion (EstudianteID, CursoID) values (6,3);
INSERT INTO Matriculacion (EstudianteID, CursoID) values (7,2);
INSERT INTO Matriculacion (EstudianteID, CursoID) values (8,9);
INSERT INTO Matriculacion (EstudianteID, CursoID) values (8,8);
INSERT INTO Matriculacion (EstudianteID, CursoID) values (8,13);
INSERT INTO Matriculacion (EstudianteID, CursoID) values (8,11);
INSERT INTO Matriculacion (EstudianteID, CursoID) values (9,7);
INSERT INTO Matriculacion (EstudianteID, CursoID) values (9,6);
INSERT INTO Matriculacion (EstudianteID, CursoID) values (10,1);
INSERT INTO Matriculacion (EstudianteID, CursoID) values (10,6);
INSERT INTO Matriculacion (EstudianteID, CursoID) values (10,13);
INSERT INTO Matriculacion (EstudianteID, CursoID) values (10,12);
```

Se me ha escapado una tilde en la tabla de Alumnos, así que por limitaciones del sistema ASCII, donde debería poner “José” pone “Jos”. Con el siguiente comando lo corrijo.

```
UPDATE Estudiantes SET Nombre='Jose' WHERE Nombre='Jos';
mysql> UPDATE Estudiantes SET Nombre='Jose' WHERE Nombre='Jos';
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

Por último, elimino el registro del alumno número 11 para contemplar menos alumnos.

```
DELETE FROM Estudiantes WHERE EstudianteID=11;
mysql> DELETE FROM Estudiantes WHERE EstudianteID=11;
Query OK, 1 row affected (0.02 sec)
```

Paso 4: Generar Consultas.

Por último, el resultado de las consultas.

1. Muestra todos los estudiantes y sus nombres completos.

```
SELECT CONCAT_WS(' ', Nombre, Apellidos) AS "Nombre Completo"
FROM Estudiantes;
```

```
mysql> SELECT CONCAT_WS(' ', Nombre, Apellidos) AS "Nombre Completo" FROM Estudiantes;
+-----+
| Nombre Completo |
+-----+
| Gustavo Adolfo Becquer |
| Luis Cernuda |
| Federico García Lorca |
| Jose de Espronceda |
| Pio Baroja |
| Azorín |
| Antonio Machado |
| Miguel de Unamuno |
| Miguel de Cervantes |
| Francisco de Quevedo |
+-----+
10 rows in set (0.00 sec)
```

Por lo general, con el comando CONCAT() debería valer, pero como el campo "Apellido" es Nulleable, he decidido usar CONCAT_WS(), que contempla dicha posibilidad.

2. Lista los cursos ofrecidos por un departamento específico.

```
SELECT NombreCurso FROM Cursos WHERE DepartamentoID=8;
```

```
mysql> SELECT NombreCurso FROM Cursos WHERE DepartamentoID=8;
+-----+
| NombreCurso |
+-----+
| Curso de Armnica y Flauta de Pan |
| Curso de Baile de Claque |
| Curso de Globoflexia |
| Curso de Fabricacion de Titeres |
+-----+
4 rows in set (0.00 sec)
```

3. Obtén todos los profesores asociados a un departamento en particular.

```
SELECT CONCAT_WS(' ', Nombre, Apellidos) AS "Nombre Completo"
FROM Profesores WHERE DepartamentoID=8;
```

```
mysql> SELECT CONCAT_WS(' ', Nombre, Apellidos)AS "Nombre Completo" FROM Profesores WHERE DepartamentoID=8;
+-----+
| Nombre Completo |
+-----+
| Frederic Chopin |
| Hans Zimmer |
+-----+
2 rows in set (0.00 sec)
```

4. Muestra los estudiantes matriculados en un curso específico.

```
SELECT CONCAT_WS(' ', Nombre, Apellidos) AS "Nombre Completo"
FROM Estudiantes e LEFT JOIN Matriculacion m ON e.EstudianteID=m.EstudianteID
WHERE m.CursoID=6;

mysql> SELECT CONCAT_WS(' ', Nombre, Apellidos) AS "Nombre Completo" FROM Estudiantes e LEFT JOIN Matriculacion m ON e.EstudianteID=m.EstudianteID WHERE m.CursoID=6;
+-----+
| Nombre Completo |
+-----+
| Federico García Lorca |
| Jose de Espronceda |
| Miguel de Cervantes |
| Francisco de Quevedo |
+-----+
4 rows in set (0.00 sec)
```

5. Encuentra los cursos que no tienen ningún estudiante matriculado.

```
SELECT NombreCurso FROM Cursos c
LEFT JOIN Matriculacion m ON c.CursoID=m.CursoID
GROUP BY c.CursoID HAVING COUNT(m.EstudianteID)=0;

mysql> SELECT NombreCurso FROM Cursos c LEFT JOIN Matriculacion m ON c.CursoID=m.CursoID GROUP BY c.CursoID HAVING COUNT(m.EstudianteID)=0;
+-----+
| NombreCurso |
+-----+
| Curso de Matemáticas Analíticas |
+-----+
1 row in set (0.00 sec)
```

6. Identifica a todos los profesores que imparten más de un curso.

```
SELECT CONCAT_WS(' ', Nombre, Apellidos) AS "Nombre Completo"
FROM Profesores p LEFT JOIN Cursos c ON p.ProfesorID=c.ProfesorID
GROUP BY p.ProfesorID having Count(CursoID)>1;

mysql> SELECT CONCAT_WS(' ', Nombre, Apellidos) AS "Nombre Completo" FROM Profesores p LEFT JOIN Cursos c ON p.ProfesorID=c.ProfesorID GROUP BY p.ProfesorID having Count(CursoID)>1;
+-----+
| Nombre Completo |
+-----+
| Johann Sebastian Mastropiero |
| Paul McCartney |
| Claude Debussy |
| Frederic Chopin |
+-----+
4 rows in set (0.00 sec)
```

7. Muestra todos los estudiantes que nacieron en un mes determinado.

```
SELECT CONCAT_WS(' ', Nombre, Apellidos) AS "Nombre Completo"
FROM Estudiantes WHERE Month(FechaNacimiento)=1;

mysql> SELECT CONCAT_WS(' ', Nombre, Apellidos) AS "Nombre Completo" FROM Estudiantes WHERE Month(FechaNacimiento)=1;
Empty set (0.00 sec)
```

Como todos los Alumnos tienen el campo FechaNacimiento generado por NOW(), todos nacieron en noviembre.

```
mysql> SELECT CONCAT_WS(' ', Nombre, Apellidos) AS "Nombre Completo" FROM Estudiantes WHERE Month(FechaNacimiento)=11;
+-----+
| Nombre Completo |
+-----+
| Gustavo Adolfo Becquer |
| Luis Cernuda |
| Federico García Lorca |
| Jose de Espronceda |
| Pío Baroja |
| Azorín |
| Antonio Machado |
| Miguel de Unamuno |
| Miguel de Cervantes |
| Francisco de Quevedo |
+-----+
10 rows in set (0.00 sec)
```

8. Encuentra los cursos con el mayor número de créditos.

```
SELECT NombreCurso, Creditos FROM Cursos ORDER BY Creditos DESC LIMIT 3;
mysql> SELECT NombreCurso, Creditos FROM Cursos ORDER BY Creditos DESC LIMIT 3;
+-----+-----+
| NombreCurso | Creditos |
+-----+-----+
| Curso de Gallego | 300 |
| Curso de Matemáticas Analíticas | 300 |
| Curso de Armonía y Flauta de Pan | 120 |
+-----+-----+
3 rows in set (0.00 sec)
```

9. Encuentra a los estudiantes que no se hayan matriculado en ningún curso.

```
SELECT CONCAT_WS(' ', Nombre, Apellidos) AS "Nombre Completo"
FROM Estudiantes e LEFT JOIN Matriculacion m
ON e.EstudianteID=m.EstudianteID GROUP BY e.EstudianteID
Having Count(CursoID)=0;
mysql> SELECT CONCAT_WS(' ', Nombre, Apellidos) AS "Nombre Completo" FROM Estudiantes e LEFT JOIN Matriculacion m ON e.EstudianteID=m.EstudianteID GROUP BY e.EstudianteID Having Count(CursoID)=0;
+-----+
| Nombre Completo |
+-----+
| Pío Baroja |
+-----+
1 row in set (0.00 sec)
```

10. Identifica los departamentos que no tienen ningún curso asociado.

```
SELECT DepartamentoName FROM Departamentos d
LEFT JOIN Cursos c ON d.DepartamentoID=c.DepartamentoID
GROUP BY d.DepartamentoID Having Count(CursoID)=0;
mysql> SELECT DepartamentoName FROM Departamentos d LEFT JOIN Cursos c ON d.DepartamentoID=c.DepartamentoID GROUP BY d.DepartamentoID Having Count(CursoID)=0;
+-----+
| DepartamentoName |
+-----+
| Departamento de Ciencias |
| Departamento de Filosofía |
+-----+
2 rows in set (0.00 sec)
```