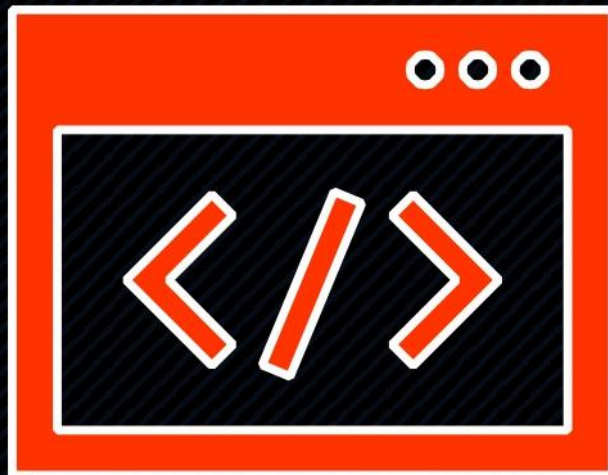


# Big Data Aplicado

---

## Consultas Modelo Relacional

{Andrei Alexandru Miu}



## Índice

Paso 1: Crear la BBDD, sin tablas .....	3
Paso 2: Generar las tablas de la BBDD .....	4
Paso 3: Insertar/modificar/borrar datos en las tablas .....	7
Paso 4: Generar consultas .....	15

## Paso 1: Crear la BBDD, sin tablas

```
C:\Users\Andrei>docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED    STATUS    PORTS    NAMES
C:\Users\Andrei>
```

Abriremos la cmd y escribiremos `docker ps a-`. Esto para ver los contenedores activos.

```
C:\Users\Andrei>docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED    STATUS    PORTS    NAMES
C:\Users\Andrei>docker run --name MysqIDBA -e MYSQL_ROOT_PASSWORD=my-secret-pw -e MYSQL_DATABASE=Instituto
-v MySQL_Volumen:/var/lib/mysql -d mysql
bc77b8a134fd4fd172ccaf515923584389c256d9bccdad4796ce81d5124b2eef
C:\Users\Andrei>
C:\Users\Andrei>
```

Luego pondremos el comando `docker run --name MysqIDBA -e MYSQL_ROOT_PASSWORD=my-secret-pw -e MYSQL_DATABASE=Instituto -v MySQL_Volumen:/var/lib/mysql -d mysql`

Explicación paso a paso:

- `docker run` -> para ejecutar un contenedor
- `--name` -> Nombre del contenedor en Docker.
- `-e [var]` -> Variables de entorno, en este caso hemos puesto la contraseña (`my-secret-pw`) y el nombre de la bbdd (`Instituto`). Para más información [click aquí](#).
- `-v [Nombre Volumen:/var/lib/mysql]` -> Para crear el volumen de Docker.
- `-d` -> daemon, para mantenerlo en segundo plano. El `:tag` es para las versiones, por default si no pones nada te pone la latest.

```
C:\Users\Andrei>docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED    STATUS    PORTS    NAMES
62f83f101fb9   mysql    "docker-entrypoint.s..." 10 minutes ago    Up 10 minutes    3306/tcp, 33060/tcp    MysqIDBA
C:\Users\Andrei>
```

Volveremos a escribir `docker ps a-` y veremos como ahora aparece el contenedor que acabamos de crear.

```
C:\Users\Andrei>docker exec -it MysqldbA bin/bash
bash-4.4# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.32 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Pondremos el comando `docker exec -it MysqldbA bin/bash` , y posteriormente en el bash escribiremos `mysql -u root -p`

Nos pedirá la contraseña que establecimos anteriormente en la variable de entorno, la pondremos y si esta bien deberíamos estar dentro de mysql.

Explicación paso a paso:

- `docker -exec -it ->` para ejecutar un contenedor existente (el `-it` es para iterativo).
- `MysqldbA`: Es el nombre puesto anteriormente o el id del contenedor.
- `bin/bash`: Para abrir una sesión en bash.

## Paso 2: Generar las tablas de la BBDD

```
mysql> show databases;
+-----+
| Database |
+-----+
| Instituto |
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.01 sec)

mysql>
```

Haremos un `show databases` para ver las bases de datos existentes



```
mysql> use Instituto;  
Database changed  
mysql>
```

use [nombre bbdd] para usar esa base de datos.

## Entidades

- Estudiantes: StudentID, FirstName, LastName, DateOfBirth, ...
- Cursos: CourseID, CourseName, Credits, ...
- Profesores: ProfesorID, Nombre, Apellidos, ..
- Departamentos: DepartmentID, DepartmentName, ...

## Relaciones:

- Los estudiantes pueden matricularse en Cursos.
- Los profesores pueden impartir cursos.
- Los Cursos son ofrecidos por los Departamentos.
- Los Profesores están asociados a los Departamentos.

Imagen de las relaciones y las entidades solicitadas.

```
mysql> CREATE TABLE Departamento (  
  ->     DepartmentID int NOT NULL PRIMARY KEY,  
  ->     DepartmentName nvarchar(255)  
  -> );  
Query OK, 0 rows affected, 1 warning (0.04 sec)  
  
mysql>
```

Crearemos la tabla de Departamento

```
CREATE TABLE Departamento ( DepartmentID int NOT NULL PRIMARY KEY, DepartmentName  
nvarchar(255));
```

```
mysql> CREATE TABLE Profesor (  
  ->     ProfesorID int NOT NULL PRIMARY KEY,  
  ->     Nombre nvarchar(255),  
  ->     Apellido nvarchar(255),  
  ->     DepartmentID int,  
  ->     CONSTRAINT FK_Profesor_Departamento FOREIGN KEY (DepartmentID) REFERENCES Departamento(DepartmentID)  
  -> );  
Query OK, 0 rows affected, 2 warnings (0.04 sec)  
  
mysql>
```

Crearemos la tabla de Profesor

```
CREATE TABLE Profesor ( ProfesorID int NOT NULL PRIMARY KEY, Nombre nvarchar(255), Apellido  
nvarchar(255), DepartmentID int, CONSTRAINT FK_Profesor_Departamento FOREIGN KEY  
(DepartmentID) REFERENCES Departamento(DepartmentID) );
```

```
mysql> CREATE TABLE Curso (  
->   CourseID int NOT NULL PRIMARY KEY,  
->   CourseName nvarchar(255),  
->   Credits int,  
->   ProfesorID int,  
->   DepartmentID int,  
->   CONSTRAINT FK_Curso_Profesor FOREIGN KEY (ProfesorID) REFERENCES Profesor(ProfesorID),  
->   CONSTRAINT FK_Curso_Departamento FOREIGN KEY (DepartmentID) REFERENCES Departamento(DepartmentID)  
-> );  
Query OK, 0 rows affected, 1 warning (0.05 sec)  
mysql>
```

Creamos la tabla de Curso

```
CREATE TABLE Curso ( CourseID int NOT NULL PRIMARY KEY, CourseName nvarchar(255), Credits  
int, ProfesorID int, DepartmentID int, CONSTRAINT FK_Curso_Profesor FOREIGN KEY (ProfesorID)  
REFERENCES Profesor(ProfesorID), CONSTRAINT FK_Curso_Departamento FOREIGN KEY (DepartmentID)  
REFERENCES Departamento(DepartmentID) );
```

```
mysql> CREATE TABLE Estudiante (  
->   StudentID int NOT NULL PRIMARY KEY,  
->   FirstName nvarchar(255),  
->   LastName nvarchar(255),  
->   DateOfBirth DATETIME,  
->   CourseID int,  
->   CONSTRAINT FK_Estudiante_Curso FOREIGN KEY (CourseID) REFERENCES Curso(CourseID)  
-> );  
Query OK, 0 rows affected, 2 warnings (0.04 sec)  
mysql>
```

Creamos la tabla de Estudiante

```
CREATE TABLE Estudiante ( StudentID int NOT NULL PRIMARY KEY, FirstName nvarchar(255),  
LastName nvarchar(255), DateOfBirth DATETIME, CourseID int, CONSTRAINT FK_Estudiante_Curso  
FOREIGN KEY (CourseID) REFERENCES Curso(CourseID) );
```

```
mysql> show tables;  
+-----+  
| Tables_in_Instituto |  
+-----+  
| Curso                |  
| Departamento         |  
| Estudiante           |  
| Profesor             |  
+-----+  
4 rows in set (0.00 sec)  
mysql>
```

Por último, haremos uso de **show tables** para visualizar las tablas creadas.

## Paso 3: Insertar/modificar/borrar datos en las tablas

```
mysql> insert into Departamento (DepartmentID, DepartmentName) VALUES
-> (1, 'Matemáticas'),
-> (2, 'Física y Química'),
-> (3, 'Astronomía'),
-> (4, 'Música'),
-> (5, 'Geografía e Historia'),
-> (6, 'Literatura'),
-> (7, 'Idiomas'),
-> (8, 'Informática'),
-> (9, 'Religión'),
-> (10, 'Latín');
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql>
```

Añadimos datos a la tabla Departamento

```
insert
into
  Departamento
  (DepartmentID, DepartmentName)
VALUES
  (1, 'Matemáticas'), (
    2, 'Física y Química'
  ), (
    3, 'Astronomía'
  ), (
    4, 'Música'
  ), (
    5, 'Geografía e Historia'
  ), (
    6, 'Literatura'
  ), (
    7, 'Idiomas'
  ), (
    8, 'Informática'
  ), (
    9, 'Religión'
  ), (
    10, 'Latín'
  );
```

```
mysql> insert into Profesor (ProfesorID, Nombre, Apellido, DepartmentID) VALUES
-> (1, 'Juan', 'Garcia', 1),
-> (2, 'Maria', 'Lopez', 2),
-> (3, 'Pedro', 'Martinez', 3),
-> (4, 'Ana', 'Snchez', 4),
-> (5, 'Luis', 'Gonzlez', 5),
-> (6, 'Sofia', 'Prez', 6),
-> (7, 'Carlos', 'Rodriguez', 7),
-> (8, 'Laura', 'Fernndez', 8),
-> (9, 'Miguel', 'Diaz', 9),
-> (10, 'Elena', 'Ruiz', 10),
-> (11, 'Javier', 'Hernndez', 1),
-> (12, 'Carmen', 'Torres', 3),
-> (13, 'Ricardo', 'Jimnez', 3),
-> (14, 'Isabel', 'Luna', 2),
-> (15, 'Antonio', 'Gomez', 3),
-> (16, 'Marta', 'Serrano', 7),
-> (17, 'Fernando', 'Vega', 7),
-> (18, 'Rosa', 'Molina', 7),
-> (19, 'Daniel', 'Ortega', 8),
-> (20, 'Sara', 'Castro', 9);
Query OK, 20 rows affected (0.01 sec)
Records: 20  Duplicates: 0  Warnings: 0

mysql>
```

Añadimos datos a la tabla Profesor

```
insert
into
  Profesor
  (ProfesorID, Nombre, Apellido, DepartmentID)
VALUES
  (1, 'Juan', 'Garcia', 1), (
  2, 'Maria', 'Lopez', 2
  ), (
  3, 'Pedro', 'Martinez', 3
  ), (
  4, 'Ana', 'Sánchez', 4
  ), (
  5, 'Luis', 'González', 5
  ), (
  6, 'Sofia', 'Pérez', 6
  ), (
  7, 'Carlos', 'Rodriguez', 7
  ), (
  8, 'Laura', 'Fernández', 8
  ), (
  9, 'Miguel', 'Diaz', 9
  ), (
  10, 'Elena', 'Ruiz', 10
  ), (
  11, 'Javier', 'Hernández', 1
  ), (
  12, 'Carmen', 'Torres', 3
  ), (
  13, 'Ricardo', 'Jiménez', 3
  ), (
  14, 'Isabel', 'Luna', 2
  ), (
  15, 'Antonio', 'Gomez', 3
```



```
), (  
  16, 'Marta', 'Serrano', 7  
) , (  
  17, 'Fernando', 'Vega', 7  
) , (  
  18, 'Rosa', 'Molina', 7  
) , (  
  19, 'Daniel', 'Ortega', 8  
) , (  
  20, 'Sara', 'Castro', 9  
) ;
```

```
mysql> insert into Curso (CourseID, CourseName, Credits, ProfesorID, DepartmentID) VALUES  
-> (1, 'Matemáticas Avanzadas', 3, 1, 1),  
-> (2, 'Física Cuántica', 4, 2, 2),  
-> (3, 'Astronomía', 3, 3, 3),  
-> (4, 'Música Clásica', 2, 4, 4),  
-> (5, 'Historia Antigua', 4, 5, 5),  
-> (6, 'Literatura', 3, 6, 6),  
-> (7, 'Inglés', 2, 7, 7),  
-> (8, 'Programación 1', 4, 8, 8),  
-> (9, 'Religion', 1, 9, 9),  
-> (10, 'Latín', 3, 10, 10),  
-> (11, 'Matemáticas Avanzadas', 3, 11, 1),  
-> (12, 'Química', 4, 12, 2),  
-> (13, 'Astronomía', 3, 13, 3),  
-> (14, 'Teoría Musical', 2, 14, 4),  
-> (15, 'Geografía 2', 4, 15, 5),  
-> (16, 'Literatura', 3, 16, 6),  
-> (17, 'Inglés', 2, 17, 7),  
-> (18, 'Programación 2', 4, 18, 8),  
-> (19, 'Religion', 1, 19, 9),  
-> (20, 'Latín', 3, 20, 10);  
Query OK, 20 rows affected (0.01 sec)  
Records: 20  Duplicates: 0  Warnings: 0  
  
mysql>
```

Añadimos datos a la tabla Curso

```
insert  
into  
  Curso  
  (CourseID, CourseName, Credits, ProfesorID, DepartmentID)  
VALUES  
  (1, 'Matemáticas Avanzadas', 3, 1, 1), (  
    2, 'Física Cuántica', 4, 2, 2  
  ) , (  
    3, 'Astronomía', 3, 3, 3  
  ) , (  
    4, 'Música Clásica', 2, 4, 4  
  ) , (  
    5, 'Historia Antigua', 4, 5, 5  
  ) , (  
    6, 'Literatura', 3, 6, 6  
  ) , (  
    7, 'Inglés', 2, 7, 7  
  ) , (  
    8, 'Programación 1', 4, 8, 8  
  ) , (  
    9, 'Religion', 1, 9, 9  
  ) , (  
    10, 'Latín', 3, 10, 10  
  ) , (  
    11, 'Matemáticas Avanzadas', 3, 11, 1  
  ) , (  
    12, 'Química', 4, 12, 2  
  ) , (  
    13, 'Astronomía', 3, 13, 3  
  ) , (  
    14, 'Teoría Musical', 2, 14, 4  
  ) , (  
    15, 'Geografía 2', 4, 15, 5  
  ) , (  
    16, 'Literatura', 3, 16, 6  
  ) , (  
    17, 'Inglés', 2, 17, 7  
  ) , (  
    18, 'Programación 2', 4, 18, 8  
  ) , (  
    19, 'Religion', 1, 19, 9  
  ) , (  
    20, 'Latín', 3, 20, 10  
  ) ;
```

```
11, 'Matemáticas Avanzadas', 3, 11, 1
), (
12, 'Química', 4, 12, 2
), (
13, 'Astronomía', 3, 13, 3
), (
14, 'Teoría Musical', 2, 14, 4
), (
15, 'Geografía 2', 4, 15, 5
), (
16, 'Literatura', 3, 16, 6
), (
17, 'Inglés', 2, 17, 7
), (
18, 'Programación 2', 4, 18, 8
), (
19, 'Religion', 1, 19, 9
), (
20, 'Latín', 3, 20, 10
);
```

```
mysql> insert into Estudiante (StudentID, FirstName, LastName, DateOfBirth, CourseID) VALUES
-> (1, 'Juan', 'Garcia', '1995-05-15', 1),
-> (2, 'Maria', 'Lopez', '1996-08-20', 2),
-> (3, 'Pedro', 'Martinez', '1994-02-10', 2),
-> (4, 'Ana', 'Snchez', '1997-11-25', 7),
-> (5, 'Luis', 'Gonzalez', '1995-09-26', 1),
-> (6, 'Sofia', 'Prez', '1996-04-06', 18),
-> (7, 'Carlos', 'Rodriguez', '1994-07-27', 4),
-> (8, 'Laura', 'Fernandez', '1997-01-18', 4),
-> (9, 'Miguel', 'Diaz', '1995-03-22', 5),
-> (10, 'Elena', 'Ruiz', '1996-06-28', 9),
-> (11, 'Javier', 'Hernandez', '1994-10-07', 6),
-> (12, 'Carmen', 'Torres', '1997-12-14', 3),
-> (13, 'Ricardo', 'Jimnez', '1995-02-19', 7),
-> (14, 'Isabel', 'Luna', '1996-05-24', 17),
-> (15, 'Antonio', 'Gomez', '1994-08-29', 6),
-> (16, 'Marta', 'Serrano', '2001-11-05', 18),
-> (17, 'Fernando', 'Vega', '1995-01-10', 19),
-> (18, 'Rosa', 'Molina', '2001-11-01', 19),
-> (19, 'Daniel', 'Ortega', '1994-07-20', 20),
-> (20, 'Sara', 'Castro', '1997-10-25', 13),
-> (21, 'Qasim', 'Al-Ikbar', '1995-03-30', 1),
-> (22, 'Lope', 'De Vega', '1996-06-05', 3),
-> (23, 'Carmen', 'Medrano', '1994-09-10', 12),
-> (24, 'Pau', 'Cubarsi', '1997-12-30', 2),
-> (25, 'Brahim', 'Diaz', '2002-02-01', 3),
-> (26, 'Mustafa', 'Abqar', '1996-05-25', 3),
-> (27, 'Muhamad', 'El-Halil', '2000-08-30', 9),
-> (28, 'Mohamed', 'bin Zayed', '1961-03-11', 4),
-> (29, 'Salvador', 'Matamarios', '2002-01-10', 5),
-> (30, 'Marc', 'Marquez', '2004-04-13', 5),
-> (31, 'Valentin', 'Rossi', '1980-07-20', 6),
-> (32, 'Radu', 'Mitu', '1997-10-13', 6),
-> (33, 'Joao', 'Neves', '1995-11-30', 8),
-> (34, 'Junior', 'Neymar', '1996-06-05', 7),
-> (35, 'Cruz', 'Gomez', '1994-09-10', 8),
-> (36, 'Heraclio', 'Temandez', '1960-12-15', 5),
-> (37, 'Josefina', 'Perez', '1934-02-20', 5),
-> (38, 'Adonis', 'Martinez', '1944-05-17', 9),
-> (39, 'Jose', 'Chino', '1954-08-30', 3),
-> (40, 'Ismael', 'Sanchez', '1967-11-05', 5);
Query OK, 40 rows affected (0.01 sec)
Records: 40 Duplicates: 0 Warnings: 0

mysql>
```

Añadimos datos a la tabla Estudiante

```
insert
into
  Estudiante
(StudentID, FirstName, LastName, DateOfBirth, CourseID)
VALUES
(1, 'Juan', 'Garcia', '1995-05-15', 1), (
2, 'Maria', 'Lopez', '1996-08-20', 2
), (
3, 'Pedro', 'Martinez', '1994-02-10', 2
), (
4, 'Ana', 'Sánchez', '1997-11-25', 7
), (
5, 'Luis', 'González', '1995-09-26', 1
), (
6, 'Sofia', 'Pérez', '1996-04-06', 18
), (
7, 'Carlos', 'Rodriguez', '1994-07-27', 4
), (
8, 'Laura', 'Fernández', '1997-01-18', 4
), (
9, 'Miguel', 'Diaz', '1995-03-22', 5
), (
10, 'Elena', 'Ruiz', '1996-06-28', 9
), (
11, 'Javier', 'Hernández', '1994-10-07', 6
), (
12, 'Carmen', 'Torres', '1997-12-14', 3
), (
13, 'Ricardo', 'Jiménez', '1995-02-19', 7
), (
14, 'Isabel', 'Luna', '1996-05-24', 17
), (
15, 'Antonio', 'Gomez', '1994-08-29', 6
), (
16, 'Marta', 'Serrano', '2001-11-05', 18
), (
17, 'Fernando', 'Vega', '1995-01-10', 19
), (
18, 'Rosa', 'Molina', '2001-11-01', 19
), (
19, 'Daniel', 'Ortega', '1994-07-20', 20
), (
20, 'Sara', 'Castro', '1997-10-25', 13
), (
21, 'Qasim', 'Al-Ikbar', '1995-03-30', 1
), (
22, 'Lope', 'De Vega', '1996-06-05', 3
), (
23, 'Carmen', 'Medrano', '1994-09-10', 12
), (
24, 'Pau', 'Cubarsi', '1997-12-30', 2
), (
25, 'Brahim', 'Diaz', '2002-02-01', 3
), (
26, 'Mustafa', 'Abqar', '1996-05-25', 3
), (
27, 'Muhamad', 'El-Halil', '2000-08-30', 9
), (
28, 'Mohamed', 'bin Zayed', '1961-03-11', 4
), (
29, 'Salvador', 'Matamaros', '2002-01-10', 5
), (
30, 'Marc', 'Marquez', '2004-04-13', 5
), (
31, 'Valentin', 'Rossi', '1980-07-20', 6
), (
```

```
32, 'Radu', 'Mitu', '1997-10-13', 6
), (
33, 'Joao', 'Neves', '1995-11-30', 8
), (
34, 'Junior', 'Neymar', '1996-06-05', 7
), (
35, 'Cruz', 'Gomez', '1994-09-10', 8
), (
36, 'Heraclio', 'Temandez', '1960-12-15', 5
), (
37, 'Josefina', 'Perez', '1934-02-20', 5
), (
38, 'Adonis', 'Martinez', '1944-05-17', 9
), (
39, 'Jose', 'Chino', '1954-08-30', 3
), (
40, 'Ismael', 'Sanchez', '1967-11-05', 5
);
```

```
mysql> SELECT * FROM Estudiante WHERE StudentID=1;
+-----+-----+-----+-----+-----+
| StudentID | FirstName | LastName | DateOfBirth | CourseID |
+-----+-----+-----+-----+-----+
|          1 | Juan      | Garcia   | 1995-05-15 00:00:00 |          1 |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> UPDATE Estudiante SET CourseID = 13 WHERE StudentID = 1;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM Estudiante WHERE StudentID=1;
+-----+-----+-----+-----+-----+
| StudentID | FirstName | LastName | DateOfBirth | CourseID |
+-----+-----+-----+-----+-----+
|          1 | Juan      | Garcia   | 1995-05-15 00:00:00 |          13 |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

Modificamos el curso del alumno con id 1.

```
UPDATE
Estudiante
SET
  CourseID = 13
WHERE
  StudentID = 1;
```

```
mysql> SELECT * FROM Departamento WHERE DepartmentID=6;
+-----+-----+
| DepartmentID | DepartmentName |
+-----+-----+
|          6 | Literatura      |
+-----+-----+
1 row in set (0.00 sec)

mysql> UPDATE Departamento SET DepartmentName = 'Lengua y Literatura' WHERE DepartmentID = 6;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM Departamento WHERE DepartmentID=6;
+-----+-----+
| DepartmentID | DepartmentName |
+-----+-----+
|          6 | Lengua y Literatura |
+-----+-----+
1 row in set (0.00 sec)

mysql>
```

Cambiamos el nombre del departamento con id 6.

```
UPDATE
Departamento
SET
  DepartmentName = 'Lengua y Literatura'
WHERE
  DepartmentID = 6;
```

```
mysql> SELECT * FROM Curso WHERE CourseID=16;
+-----+-----+-----+-----+-----+
| CourseID | CourseName | Credits | ProfesorID | DepartmentID |
+-----+-----+-----+-----+-----+
|        16 | Literatura |       3 |          16 |             6 |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> UPDATE Curso SET CourseName = 'Lengua' WHERE CourseID = 16;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM Curso WHERE CourseID=16;
+-----+-----+-----+-----+-----+
| CourseID | CourseName | Credits | ProfesorID | DepartmentID |
+-----+-----+-----+-----+-----+
|        16 | Lengua     |       3 |          16 |             6 |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

Cambiamos el nombre del curso con id 16



```
UPDATE
Curso
SET
  CourseName = 'Lengua'
WHERE
  CourseID = 16;
```

```
mysql> SELECT * FROM Profesor WHERE ProfesorID=2;
+-----+-----+-----+-----+
| ProfesorID | Nombre | Apellido | DepartmentID |
+-----+-----+-----+-----+
|          2 | Maria  | Lopez    |             2 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> UPDATE Profesor SET Apellido = 'de la Fuente' WHERE ProfesorID = 2;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM Profesor WHERE ProfesorID=2;
+-----+-----+-----+-----+
| ProfesorID | Nombre | Apellido      | DepartmentID |
+-----+-----+-----+-----+
|          2 | Maria  | de la Fuente  |             2 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

Modificamos el apellido del profesor con id 2

```
UPDATE
Profesor
SET
  Apellido = 'de la Fuente'
WHERE
  ProfesorID = 2;
```

```
mysql> SELECT * FROM Estudiante WHERE StudentID=33;
+-----+-----+-----+-----+-----+
| StudentID | FirstName | LastName | DateOfBirth      | CourseID |
+-----+-----+-----+-----+-----+
|          33 | Joao      | Neves   | 1995-11-30 00:00:00 |          8 |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> DELETE FROM Estudiante WHERE StudentID = 33;
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM Estudiante WHERE StudentID=33;
Empty set (0.00 sec)

mysql>
```

Borramos un registro, en este caso el estudiante con id 33

```
DELETE
FROM
  Estudiante
WHERE
  StudentID = 33;
```

## Paso 4: Generar consultas

### 1. Muestra todos los estudiantes y sus nombres completos.

```
mysql> SELECT CONCAT(FirstName, ' ', LastName) AS Nombre_Completo FROM Estudiante;
```

Nombre_Completo
Juan Garcia
Maria Lopez
Pedro Martinez
Ana Snchez
Luis Gonzlez
Sofia Prez
Carlos Rodriguez
Laura Fernndez
Miguel Diaz
Elena Ruiz
Javier Hernndez
Carmen Torres
Ricardo Jimnez
Isabel Luna
Antonio Gomez
Marta Serrano
Fernando Vega
Rosa Molina
Daniel Ortega
Sara Castro
Qasim Al-Ikbar
Lope De Vega
Carmen Medrano
Pau Cubarsi
Brahim Diaz
Mustafa Abqar
Muhamad El-Halil
Mohamed bin Zayed
Salvador Matamaros
Marc Marquez
Valentin Rossi
Radu Mitu
Junior Neymar
Cruz Gomez
Heraclio Temandez
Josefina Perez
Adonis Martinez
Jose Chino
Ismael Sanchez

```
39 rows in set (0.00 sec)

mysql>
```

```
SELECT
CONCAT(FirstName,
',
LastName) AS Nombre Completo
FROM
Estudiante;
```

2. Lista los cursos ofrecidos por un departamento específico, por ejemplo, el Departamento de Matemáticas.

```
mysql> SELECT c.CourseName FROM Curso c
-> JOIN Departamento d ON c.DepartmentID = d.DepartmentID
-> WHERE d.DepartmentName = 'Matemáticas';
+-----+
| CourseName |
+-----+
| Matemáticas Avanzadas |
| Matemáticas Avanzadas |
+-----+
2 rows in set (0.00 sec)

mysql>
```

```
SELECT
c.CourseName
FROM
Curso c
JOIN
Departamento d
ON c.DepartmentID = d.DepartmentID
WHERE
d.DepartmentName = 'Matemáticas';
```

3. Obtén todos los profesores asociados a un departamento en particular, como el Departamento de Informática.

```
mysql> SELECT p.Nombre, p.Apellido FROM Profesor p
-> JOIN Departamento d ON p.DepartmentID = d.DepartmentID
-> WHERE d.DepartmentName = 'Informática';
+-----+-----+
| Nombre | Apellido |
+-----+-----+
| Laura | Fernández |
| Daniel | Ortega |
+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

```
SELECT
  p.Nombre,
  p.Apellido
FROM
  Profesor p
JOIN
  Departamento d
  ON p.DepartmentID = d.DepartmentID
WHERE
  d.DepartmentName = 'Informatica';
```

**4. Muestra los estudiantes matriculados en un curso específico, por ejemplo, el curso de Matemáticas Avanzadas.**

```
mysql> SELECT e.FirstName, e.LastName FROM Estudiante e
-> JOIN Curso c ON e.CourseID = c.CourseID
-> WHERE c.CourseName = 'Matemáticas Avanzadas';
+-----+-----+
| FirstName | LastName |
+-----+-----+
| Luis      | Gonzlez  |
| Qasim     | Al-Ikbar |
+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

```
SELECT
  e.FirstName,
  e.LastName
FROM
  Estudiante e
JOIN
  Curso c
  ON e.CourseID = c.CourseID
WHERE
  c.CourseName = 'Matemáticas Avanzadas';
```

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

```
mysql> SELECT c.CourseName FROM Curso c
-> LEFT JOIN Estudiante e ON c.CourseID = e.CourseID
-> WHERE e.StudentID IS NULL;
+-----+
| CourseName |
+-----+
| Latn       |
| Matemáticas Avanzadas |
| Teoría Musical |
| Geografía 2 |
| Lengua     |
+-----+
5 rows in set (0.00 sec)

mysql>
```

```
SELECT
  c.CourseName
FROM
  Curso c
LEFT JOIN
  Estudiante e
  ON c.CourseID = e.CourseID
WHERE
  e.StudentID IS NULL;
```

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

```
mysql> SELECT p.Nombre, p.Apellido FROM Profesor p
      -> JOIN Curso c ON p.ProfesorID = c.ProfesorID
      -> GROUP BY p.ProfesorID HAVING COUNT(c.CourseID) > 1;
Empty set (0.00 sec)

mysql>
```

```
SELECT
  p.Nombre,
  p.Apellido
FROM
  Profesor p
JOIN
  Curso c
  ON p.ProfesorID = c.ProfesorID
GROUP BY
  p.ProfesorID
HAVING
  COUNT(c.CourseID) > 1;
```

## 7. Muestra todos los estudiantes que nacieron en un mes determinado, como en enero.

```
mysql> SELECT FirstName, LastName FROM Estudiante
      -> WHERE MONTH(DateOfBirth) = 1;
+-----+-----+
| FirstName | LastName |
+-----+-----+
| Laura    | Fernndez |
| Fernando | Vega     |
| Salvador | Matamaros|
+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

```
SELECT
  FirstName,
  LastName
FROM
  Estudiante
WHERE
  MONTH(DateOfBirth) = 1;
```



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

```
mysql> SELECT CourseName FROM Curso
      -> WHERE Credits = (SELECT MAX(Credits) FROM Curso);
+-----+
| CourseName |
+-----+
| Fsica Cuntica |
| Historia Antigua |
| Programacin 1 |
| Quimica |
| Geografa 2 |
| Programacin 2 |
+-----+
6 rows in set (0.00 sec)

mysql>
```

```
SELECT
  CourseName
FROM
  Curso
WHERE
  Credits = (
    SELECT
      MAX(Credits)
    FROM
      Curso
  );
```

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

```
mysql> SELECT FirstName, LastName FROM Estudiante
      -> WHERE StudentID NOT IN (SELECT StudentID FROM Estudiante);
Empty set (0.00 sec)

mysql>
```

```
SELECT
  FirstName,
  LastName
FROM
  Estudiante
WHERE
  StudentID NOT IN (
    SELECT
      StudentID
    FROM
      Estudiante
  );
```

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

```
mysql> SELECT DepartmentName FROM Departamento  
      -> WHERE DepartmentID NOT IN (SELECT DepartmentID FROM Curso);  
Empty set (0.00 sec)
```

```
mysql>
```

```
SELECT  
  DepartmentName  
FROM  
  Departamento  
WHERE  
  DepartmentID NOT IN (  
    SELECT  
      DepartmentID  
    FROM  
      Curso  
  );
```