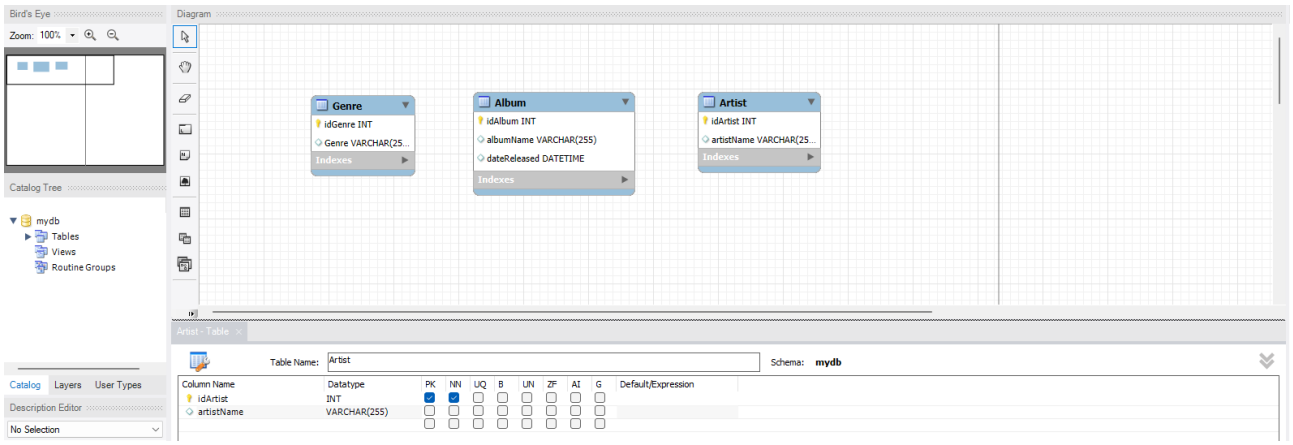
Crea el siguiente UML (pg. 29)

Ej. 1

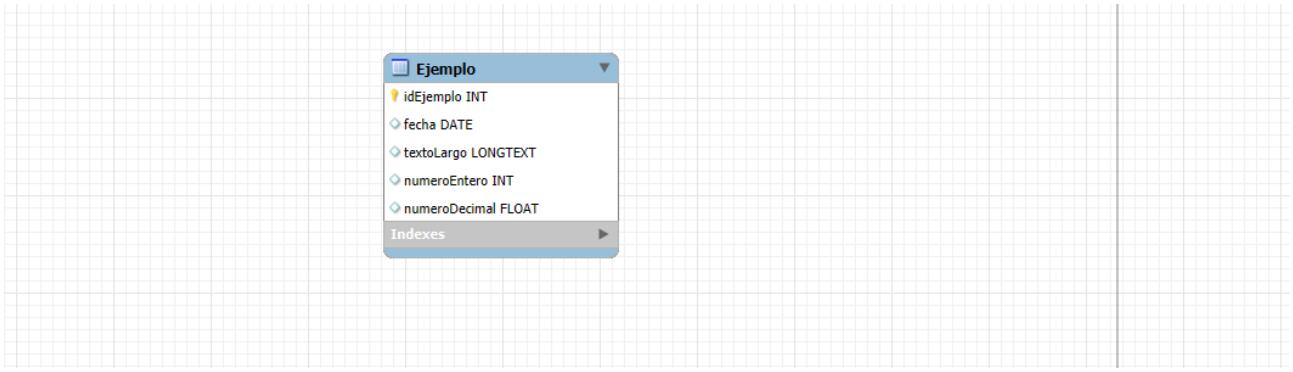


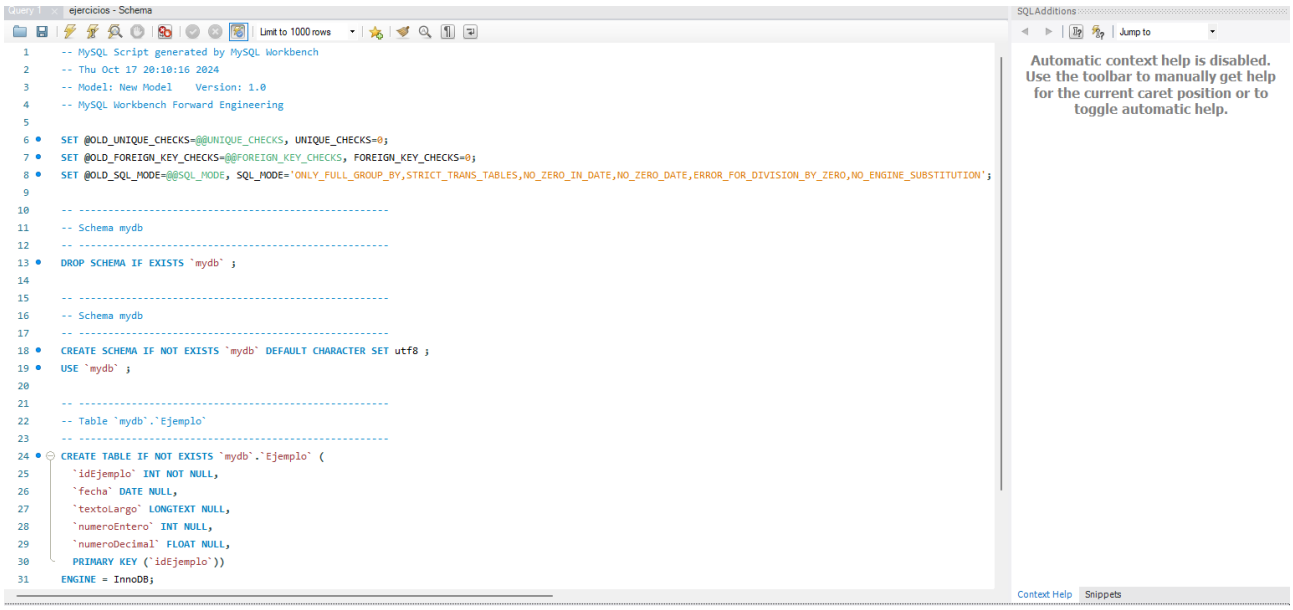
Ej. 2 (pg. 60)

```
16
17 •  DROP DATABASE IF EXISTS ejercicios;
18 •  CREATE DATABASE ejercicios;
```

```
12  16:58:12  DROP DATABASE IF EXISTS ejercicios   3 row(s) affected   0.015 sec
13  16:58:12  CREATE DATABASE ejercicios           1 row(s) affected   0.000 sec
```

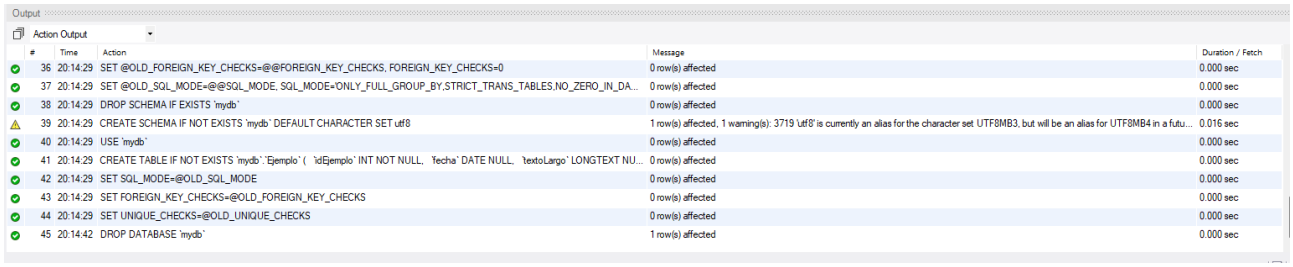
Ej. 3 (pg. 66)



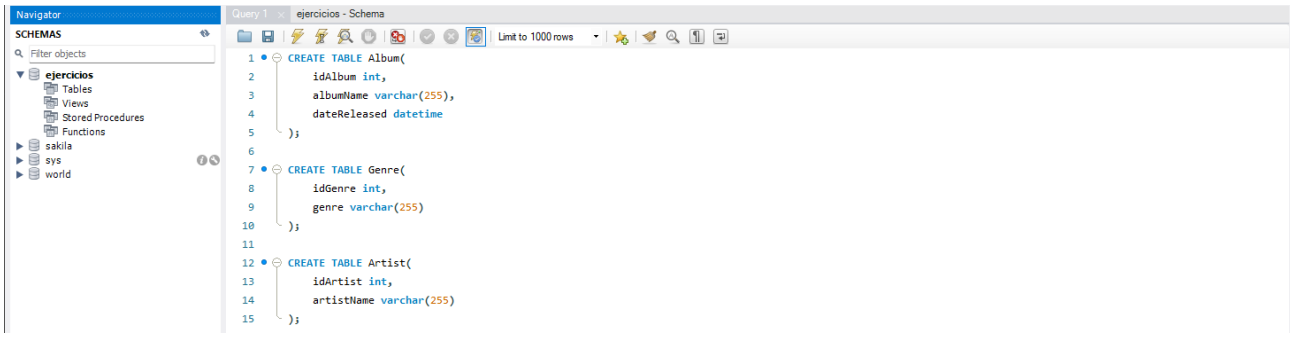


```sql
-- MySQL Script generated by MySQL Workbench
-- Thu Oct 17 20:10:16 2024
-- Model: New Model    Version: 1.0
-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

-- -----------------------------------------------------
-- Schema mydb
-- -----------------------------------------------------
DROP SCHEMA IF EXISTS `mydb` ;

-- -----------------------------------------------------
-- Schema mydb
-- -----------------------------------------------------
CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 ;
USE `mydb` ;

-- -----------------------------------------------------
-- Table `mydb`.`Ejemplo`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `mydb`.`Ejemplo` (
  `idEjemplo` INT NOT NULL,
  `fecha` DATE NULL,
  `textoLargo` LONGTEXT NULL,
  `numeroEntero` INT NULL,
  `numeroDecimal` FLOAT NULL,
  PRIMARY KEY (`idEjemplo`))
ENGINE = InnoDB;
```

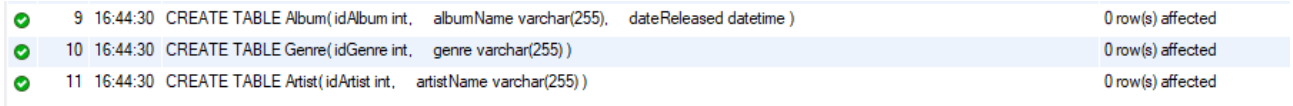
Ej. 4 (pg. 69)



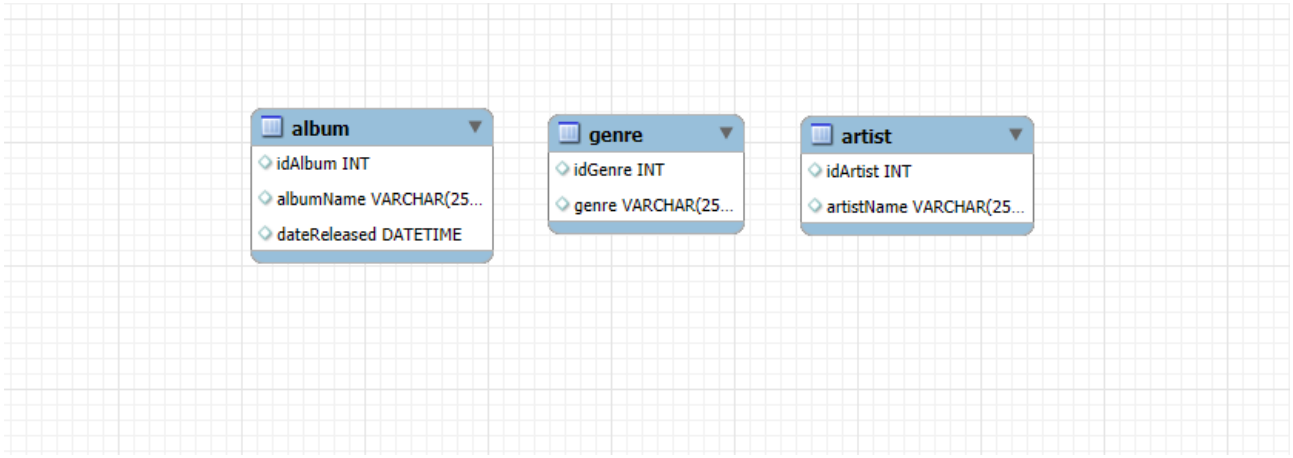
#	Time	Action	Message	Duration / Fetch
36	20:14:29	SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0	0 row(s) affected	0.000 sec
37	20:14:29	SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DA...	0 row(s) affected	0.000 sec
38	20:14:29	DROP SCHEMA IF EXISTS `mydb`	0 row(s) affected	0.000 sec
39	20:14:29	CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8	1 row(s) affected, 1 warning(s): 3719 'utf8' is currently an alias for the character set UTF8MB3, but will be an alias for UTF8MB4 in a futu...	0.016 sec
40	20:14:29	USE `mydb`	0 row(s) affected	0.000 sec
41	20:14:29	CREATE TABLE IF NOT EXISTS `mydb`.`Ejemplo` (`idEjemplo` INT NOT NULL, `fecha` DATE NULL, `textoLargo` LONGTEXT NU...	0 row(s) affected	0.000 sec
42	20:14:29	SET SQL_MODE=@OLD_SQL_MODE	0 row(s) affected	0.000 sec
43	20:14:29	SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS	0 row(s) affected	0.000 sec
44	20:14:29	SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS	0 row(s) affected	0.000 sec
45	20:14:42	DROP DATABASE `mydb`	1 row(s) affected	0.000 sec



```sql
CREATE TABLE Album(
    idAlbum int,
    albumName varchar(255),
    dateReleased datetime
);

CREATE TABLE Genre(
    idGenre int,
    genre varchar(255)
);

CREATE TABLE Artist(
    idArtist int,
    artistName varchar(255)
);
```



9	16:44:30	CREATE TABLE Album(idAlbum int, albumName varchar(255), dateReleased datetime)	0 row(s) affected
10	16:44:30	CREATE TABLE Genre(idGenre int, genre varchar(255))	0 row(s) affected
11	16:44:30	CREATE TABLE Artist(idArtist int, artistName varchar(255))	0 row(s) affected



Ej. 5 (pg. 74)

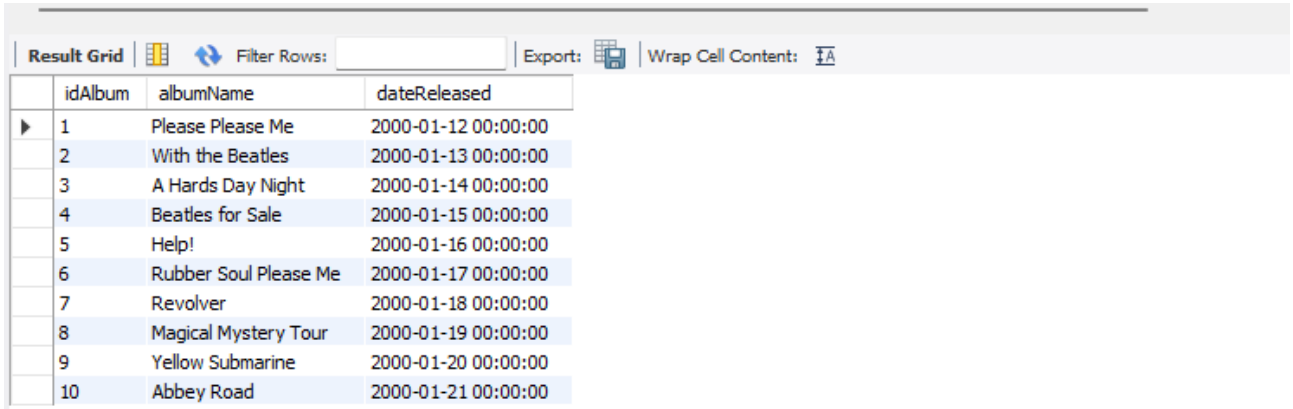
```sql
INSERT INTO Album(idAlbum, albumName, dateReleased) VALUES
(1, 'Please Please Me', '2000-01-12'),
(2, 'With the Beatles', '2000-01-13'),
(3, 'A Hards Day Night', '2000-01-14'),
(4, 'Beatles for Sale', '2000-01-15'),
(5, 'Help!', '2000-01-16'),
(6, 'Rubber Soul Please Me', '2000-01-17'),
(7, 'Revolver', '2000-01-18'),
(8, 'Magical Mystery Tour', '2000-01-19'),
(9, 'Yellow Submarine', '2000-01-20'),
(10, 'Abbey Road', '2000-01-21');

INSERT INTO Genre(idGenre, genre) VALUES
(1, 'Rock'),
(2, 'Pop'),
(3, 'Funk'),
(4, 'Gospel'),
(5, 'Jazz'),
(6, 'Rap'),
(7, 'Soul'),
(8, 'Blues'),
(9, 'Metal'),
(10, 'Punk');

INSERT INTO Artist(idArtist, artistName) VALUES
(1, 'Melendi'),
(2, 'Aitana'),
(3, 'Dani Martin'),
(4, 'Gospel'),
(5, 'Quevedo'),
(6, 'David Bisbal'),
```

f

```
111  20:49:36  INSERT INTO Album(idAlbum, albumName, dateReleased) VALUES (1, 'Please Please Me', '2000-01-12'), (2, 'With the Beatles', '2000...  10 row(s) affected Records: 10  Duplicates: 0  Warnings: 0
112  20:49:36  INSERT INTO Genre(idGenre, genre) VALUES (1, 'Rock'), (2, 'Pop'), (3, 'Funk'), (4, 'Gospel'), (5, 'Jazz'), (6, 'Rap'), (7, 'Soul'), (8, 'Blues')...  10 row(s) affected Records: 10  Duplicates: 0  Warnings: 0
113  20:49:36  INSERT INTO Artist(idArtist, artistName) VALUES (1, 'Melendi'), (2, 'Aitana'), (3, 'Dani Martin'), (4, 'Gospel'), (5, 'Quevedo'), (6, 'David Bi...  10 row(s) affected Records: 10  Duplicates: 0  Warnings: 0
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

idArtist	artistName
1	Melendi
2	Aitana
3	Dani Martin
4	Gospel
5	Quevedo
6	David Bisbal
7	Amaia Romero
8	Alvaro Soler
9	Monica Naranjo
10	Abraham Mateo



Generando datos para una BDD + Ejercicios (pg. 148)

1.
SELECT postalZip AS 'Codigo_Postal', region as 'Region', country AS 'Pais' FROM myTable;

2.
SELECT phone FROM myTable WHERE phone LIKE '(811)%';

3.
SELECT * FROM myTable WHERE country = 'Italy' or country = 'Spain';

4.
SELECT COUNT(*) FROM myTable;

5.
SELECT region, country, postalZip FROM myTable WHERE id IN (SELECT id FROM myTable WHERE country = 'Germany' or country = 'Turkey');

6.
SELECT MIN(id), MAX(id) FROM myTable;

7.
SELECT country, count(id) FROM myTable GROUP BY country;

8.
SELECT * FROM myTable ORDER BY postalZip LIMIT 10;

9.

```
DELETE FROM myTable WHERE country='Singapore';
```

10.

```
UPDATE myTable SET country='España' WHERE country='Spain';
```

Ejercicio MySQL Workbench (pg. 150)

```
CREATE SCHEMA IF NOT EXISTS db_geolocation;
USE db_geolocation;

DROP TABLE IF EXISTS `country`;
CREATE TABLE `country` (
  `country_id` SMALLINT unsigned NOT NULL auto_increment,
  `country` VARCHAR(50) default NULL,
  `last_update` TIMESTAMP default NULL,
  PRIMARY KEY (`country_id`)
) AUTO_INCREMENT=1;

DROP TABLE IF EXISTS `city`;
CREATE TABLE `city` (
  `city_id` SMALLINT unsigned NOT NULL auto_increment,
  `city` VARCHAR(50) default NULL,
  `last_update` TIMESTAMP default NULL,
  `country_id` SMALLINT unsigned,
  PRIMARY KEY (`city_id`),
  FOREIGN KEY (country_id) REFERENCES country(country_id)
) AUTO_INCREMENT=1;

DROP TABLE IF EXISTS `address`;
CREATE TABLE `address` (
  `address_id` INT unsigned NOT NULL auto_increment,
  `address` VARCHAR(50) default NULL,
  `address2` VARCHAR(50) default NULL,
  `district` VARCHAR(50) default NULL,
  `postal_code` VARCHAR(10) default NULL,
  `phone` VARCHAR(20) default NULL,
  `location` VARCHAR(20) default NULL,
  `last_update` TIMESTAMP default NULL,
  `city_id` SMALLINT unsigned,
  PRIMARY KEY (`address_id`),
  FOREIGN KEY (city_id) REFERENCES city(city_id)
) AUTO_INCREMENT=1;

INSERT INTO country (country, last_update)
VALUES
('Espania', NOW()),
('Francia', NOW()),
('Alemania', NOW()),
('Belgica', NOW()),
('Austria', NOW()),
('Bulgaria', NOW()),
('Noruega', NOW()),
('Rep. Checa', NOW()),
('Rumania', NOW()),
('Paises Bajos', NOW()),
('Canada', NOW()),
('Japon', NOW()),
('China', NOW()),
('Mexico', NOW()),
('Peru', NOW()),
```

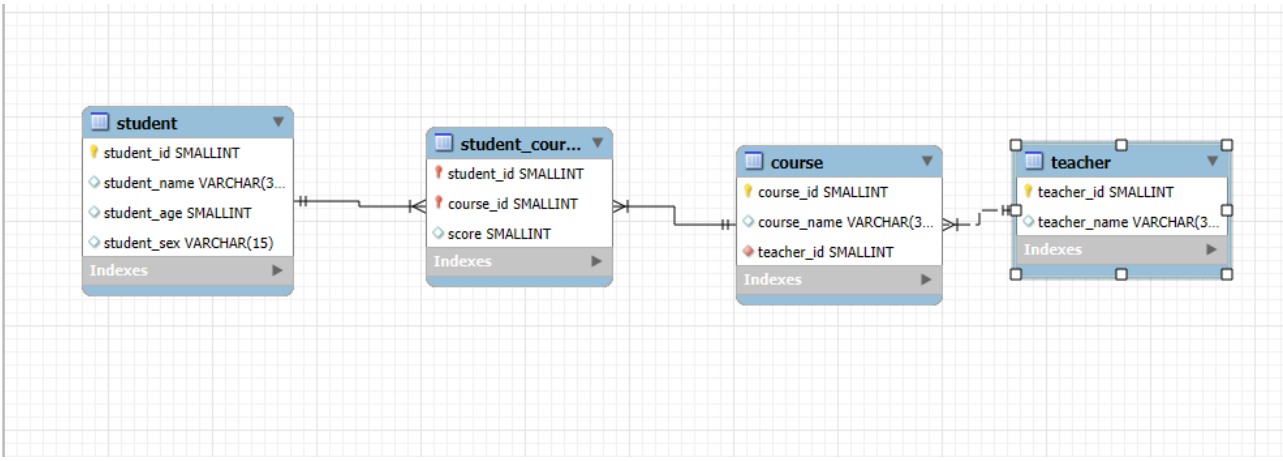
```
('Portugal', NOW()),
('Chile', NOW()),
('Puerto Rico', NOW()),
('Tailandia', NOW()),
('Corea', NOW());

INSERT INTO city(city, last_update, country_id)
VALUES
('Madrid', NOW(), 1),
('Barcelona', NOW(), 1),
('Berlin', NOW(), 3),
('Bruselas', NOW(), 4),
('Viena', NOW(), 5),
('Sofia', NOW(), 6),
('Oslo', NOW(), 7),
('Praga', NOW(), 8),
('Bucarest', NOW(), 9),
('Amsterdam', NOW(), 10),
('Toronto', NOW(), 11),
('Tokyo', NOW(), 12),
('Beijing', NOW(), 13),
('Ciudad de Mexico', NOW(), 14),
('Lima', NOW(), 15),
('Lisboa', NOW(), 16),
('Santiago', NOW(), 17),
('San Juan', NOW(), 18),
('Bangkok', NOW(), 19),
('Seoul', NOW(), 20);

INSERT INTO address (address, address2, district, postal_code, phone, location, last_update, city_id)
VALUES
('011 Calle de Aragón', 'Apt 1', 'stilf Bazaar', '10001', '123-456-7890', 'Location1', NOW(), 1),
('899 Calle de Alcalá de Guadaíra', 'Apt 2', 'coopab Hill', '90001', '987-654-3210', 'Location2', NOW(), 3),
('981 Calle de Avinyó', '', 'Central', 'Lower liv', '456-789-1234', 'Location3', NOW(), 3),
('199 Calle de Entenza', 'Suite 3', 'Waterside', 'V6E 1B4', '654-321-0987', 'Location4', NOW(), 4),
('911 Calle de Balmes', '', 'spramiok', '01000', '321-098-7654', 'Location5', NOW(), 5),
('972 Calle de la Jota', 'Floor 4', 'spagob', 'SW1A 1AA', '01234 567890', 'Location6', NOW(), 6),
('530 Calle de Muntaner', '', 'Lower nomd', '10115', '030 1234567', 'Location7', NOW(), 7),
('858 Calle de Pelayo', '', 'Altstadt', '80331', '089 12345678', 'Location8', NOW(), 8),
('011 Calle de Aragón', 'Apt 1', 'Uper sprem', '10001', '123-456-7890', 'Location1', NOW(), 9),
('899 Calle de Alcalá de Guadaíra', 'Apt 2', 'Lower Nor', '90001', '987-654-3210', 'Location2', NOW(), 8),
('981 Calle de Avinyó', '', 'Central', 'M5H 2N2', '456-789-1234', 'Location3', NOW(), 10),
('199 Calle de Entenza', 'Suite 3', 'West End', 'V6E 1B4', '654-321-0987', 'Location4', NOW(), 11),
('911 Calle de Balmes', '', 'Midtown proand', '01000', '321-098-7654', 'Location5', NOW(), 12),
('972 Calle de la Jota', 'Floor 4', 'Chelsea', 'SW1A 1AA', '01234 567890', 'Location6', NOW(), 13),
('530 Calle de Muntaner', '', 'Mitte', '10115', '030 1234567', 'Location7', NOW(), 14),
('858 Calle de Pelayo', '', 'Altstadt', '80331', '089 12345678', 'Location8', NOW(), 15),
('199 Calle de Entenza', 'Apt 1', 'Midtown proand', '10001', '123-456-7890', 'Location1', NOW(), 16),
('858 Calle de Pelayo', 'Apt 2', 'North sneepord', '90001', '987-654-3210', 'Location2', NOW(), 17),
('789 Maple St', '', 'stregaic Center', 'M5H 2N2', '456-789-1234', 'Location3', NOW(), 18),
('530 Calle de Muntaner', 'Suite 3', 'puriatlen Circle', 'V6E 1B4', '654-321-0987', 'Location4', NOW(), 19),
('972 Calle de la Jota', 'Suite 3', 'dossotop Row', 'V6E 1B4', '654-321-0987', 'Location4', NOW(), 20);


SELECT country, city, address, phone
FROM country AS pais
  INNER JOIN
  city AS ciudad
  ON pais.country_id = ciudad.city_id
  INNER JOIN
  address AS direccion
  ON ciudad.city_id = direccion.city_id;
```

Ejercicio MySQL Workbench (pg. 151)

```
CREATE SCHEMA IF NOT EXISTS db_sales_enterprise;
USE db_sales_enterprise;

DROP TABLE IF EXISTS `customer`;
CREATE TABLE `customer` (
  `customer_id` SMALLINT NOT NULL auto_increment,
  `cust_name` VARCHAR(30) default NULL,
  `city`  VARCHAR(15) default NULL,
  `grade`  SMALLINT default NULL,
  PRIMARY KEY (`customer_id`)
) AUTO_INCREMENT=1;

DROP TABLE IF EXISTS `salesman`;
CREATE TABLE `salesman` (
  `salesman_id` SMALLINT NOT NULL auto_increment,
  `name` VARCHAR(30) default NULL,
  `city`  VARCHAR(15) default NULL,
  `commission` REAL default NULL,
  PRIMARY KEY (`salesman_id`)
) AUTO_INCREMENT=1;

CREATE TABLE orders (
    order_number SMALLINT NOT NULL auto_increment,
    purchase_amt DECIMAL(8,2) DEFAULT NULL,
    order_date DATE DEFAULT NULL,
    customer_id SMALLINT NOT NULL,
    salesman_id SMALLINT NOT NULL,
    PRIMARY KEY (order_number, customer_id, salesman_id),
    FOREIGN KEY (customer_id) REFERENCES customer(customer_id),
    FOREIGN KEY (salesman_id) REFERENCES salesman(salesman_id)
) AUTO_INCREMENT=1;

INSERT INTO customer (cust_name, city, grade)
VALUES ('Alex Tintor', 'Nueva York', 100),
('Aitor Menta', 'Los Angeles', 200),
('Elena Morada', 'Chicago', 150),
('Benito Camelas', 'Florida', 180),
('Deborah Dora', 'Nueva York', 220),
('John Smith', 'Grecia', 190),
('Frank Stein', 'Malibu', 210),
('Gene McGee', 'St. Petersburg', 170),
('Elmo Skito', 'Brooklyn', 160),
('Elton Tito', 'Berlin', 120);

INSERT INTO salesman (name, city, commission)
VALUES
('Alex Johnson', 'New York', 0.15),
('Jose Smith', 'Los Angeles', 0.12),
('Michael Green', 'Chicago', 0.14),
('David Brown', 'Chicago', 0.16),
('Betty White', 'Metropolis', 0.13),
('Frank Gray', 'Grecia', 0.10),
('Grace Silver', 'Malibu', 0.11),
('Harry Gold', 'St. Petersburg', 0.09),
('Ivy Diamond', 'Brooklyn', 0.08),
('Jack Daemon', 'Boston', 0.17);

INSERT INTO orders (purchase_amt, order_date, customer_id, salesman_id)
VALUES
(500.50, '2023-01-15', 1, 1),
```

```
(300.00, '2023-02-12', 2, 2),
(450.75, '2023-03-10', 3, 3),
(600.25, '2023-04-05', 4, 4),
(700.80, '2023-05-20', 5, 5),
(250.40, '2023-06-30', 6, 6),
(550.90, '2023-07-14', 7, 7),
(450.60, '2023-08-19', 8, 8),
(350.70, '2023-09-02', 9, 9),
(650.30, '2023-10-10', 10, 10);

SELECT count(order_number) FROM orders;

SELECT DISTINCT cust_name FROM customer;

SELECT * FROM salesman ORDER BY commission DESC;

SELECT * FROM customer ORDER BY customer_id ASC LIMIT 5;

SELECT MIN(purchase_amt) FROM orders;

SELECT MAX(purchase_amt) FROM orders;
```

Ejercicio MySQL Workbench (pg. 151)




```
CREATE SCHEMA IF NOT EXISTS db_school;
USE db_school;

DROP TABLE IF EXISTS student;
CREATE TABLE student (
  student_id SMALLINT NOT NULL auto_increment,
  student_name VARCHAR(35) default NULL,
  student_age  SMALLINT default NULL,
  student_sex  VARCHAR(15) default NULL,
  PRIMARY KEY (student_id)
) AUTO_INCREMENT=1;

DROP TABLE IF EXISTS teacher;
CREATE TABLE teacher (
  teacher_id SMALLINT NOT NULL auto_increment,
  teacher_name VARCHAR(35) default NULL,
  PRIMARY KEY (teacher_id)
) AUTO_INCREMENT=1;
```

```
DROP TABLE IF EXISTS course;
CREATE TABLE course (
  course_id SMALLINT NOT NULL auto_increment,
  course_name VARCHAR(35) default NULL,
  teacher_id SMALLINT NOT NULL,
  PRIMARY KEY (course_id),
  FOREIGN KEY (teacher_id) REFERENCES teacher(teacher_id)
) AUTO_INCREMENT=1;

DROP TABLE IF EXISTS student_course;
CREATE TABLE student_course (
  student_id SMALLINT NOT NULL,
  course_id SMALLINT NOT NULL,
  score SMALLINT default NULL,
  PRIMARY KEY (student_id, course_id),
  FOREIGN KEY (student_id) REFERENCES student(student_id),
  FOREIGN KEY (course_id) REFERENCES course(course_id)
);
```

Ejercicio MySQL Workbench (pg. 152)

```
CREATE SCHEMA IF NOT EXISTS db_school;
USE db_school;

DROP TABLE IF EXISTS student;
CREATE TABLE student (
  student_id SMALLINT NOT NULL auto_increment,
  student_name VARCHAR(35) default NULL,
  student_age  SMALLINT default NULL,
  student_sex  VARCHAR(15) default NULL,
  PRIMARY KEY (student_id)
) AUTO_INCREMENT=1;

DROP TABLE IF EXISTS teacher;
CREATE TABLE teacher (
  teacher_id SMALLINT NOT NULL auto_increment,
  teacher_name VARCHAR(35) default NULL,
  PRIMARY KEY (teacher_id)
) AUTO_INCREMENT=1;

DROP TABLE IF EXISTS course;
CREATE TABLE course (
  course_id SMALLINT NOT NULL auto_increment,
  course_name VARCHAR(35) default NULL,
  teacher_id SMALLINT NOT NULL,
  PRIMARY KEY (course_id),
  FOREIGN KEY (teacher_id) REFERENCES teacher(teacher_id)
) AUTO_INCREMENT=1;

DROP TABLE IF EXISTS student_course;
CREATE TABLE student_course (
  student_id SMALLINT NOT NULL,
  course_id SMALLINT NOT NULL,
  score SMALLINT default NULL,
  PRIMARY KEY (student_id, course_id),
  FOREIGN KEY (student_id) REFERENCES student(student_id),
  FOREIGN KEY (course_id) REFERENCES course(course_id)
);

INSERT INTO student (student_name, student_age, student_sex)
```

```
VALUES
('Aitor Menta', 20, 'Hombre'),
('Benito Camelas', 22, 'Hombre'),
('Deborah Dora', 19, 'Mujer'),
('Alicia Llaves', 21, 'Mujer'),
('Cristina Flores', 23, 'Hombre'),
('Alberto Abierto', 18, 'Mujer'),
('Victor Buenamigo', 24, 'Hombre'),
('Laura Dora', 22, 'Mujer'),
('Mike Thompson', 21, 'Hombre'),
('Chris Jackson', 20, 'Hombre');

INSERT INTO teacher (teacher_name)
VALUES
('Emmet Brown'),
('Rick Sanchez'),
('James Neutron'),
('Flink Stein'),
('Albert Einstein'),
('Bethany White'),
('Mekane Orange'),
('Sophia Violet'),
('George Washing'),
('Billy Jackson');

INSERT INTO course (course_name, teacher_id)
VALUES
('Matematicas', 1),
('Biologia', 2),
('Quimica', 3),
('Historia', 4),
('Fisica', 5),
('Filosofia', 6),
('Ingles', 7),
('Latin', 8),
('Frances', 9),
('Aleman', 10);

INSERT INTO student_course (student_id, course_id, score)
VALUES
(1, 1, 85),
(1, 2, 90),
(2, 3, 75),
(2, 4, 80),
(3, 1, 95),
(3, 5, 88),
(4, 2, 78),
(4, 6, 82),
(5, 3, 90),
(5, 7, 85),
(6, 4, 91),
(6, 8, 77),
(7, 1, 88),
(7, 9, 92),
(8, 2, 81),
(8, 10, 79),
(9, 3, 84),
(9, 5, 87),
(10, 4, 86),
(10, 6, 89);

SELECT stu.student_name, tea.teacher_name FROM student stu
LEFT JOIN student_course stu_cour ON stu.student_id = stu_cour.student_id
```

```
LEFT JOIN course cour ON stu_cour.course_id = cour.course_id
LEFT JOIN teacher tea ON tea.teacher_id = cour.teacher_id;

SELECT tea.teacher_name, COUNT(cour.course_id) FROM course cour
LEFT JOIN teacher tea ON tea.teacher_id = cour.teacher_id GROUP BY cour.teacher_id;

SELECT stu.student_name, COUNT(stuco.course_id) FROM student_course stuco
LEFT JOIN student stu ON stuco.student_id = stu.student_id GROUP BY stuco.student_id;
```

Ejercicio MySQL Workbench (pg. 152)

ROLLBACK

Para este ejercicio se realizarán las actividades en la tabla students creada para la actividad anterior:

```
SELECT * from students;
```

student_id	student_name	student_age	student_sex
1	Aitor Menta	20	Hombre
2	Benito Camelas	22	Hombre
3	Deborah Dora	19	Mujer
4	Alicia Llaves	21	Mujer
5	Cristina Flores	23	Hombre
6	Alberto Abierto	18	Mujer
7	Victor Buenamigo	24	Hombre
8	Laura Dora	22	Mujer
9	Mike Thompson	21	Hombre
10	Chris Jackson	20	Hombre
NULL	NULL	NULL	NULL

```
Begin;
DELETE FROM student_course WHERE student_id IN (SELECT student_id FROM student);
DELETE FROM student;
SELECT * from student;
Rollback;
```

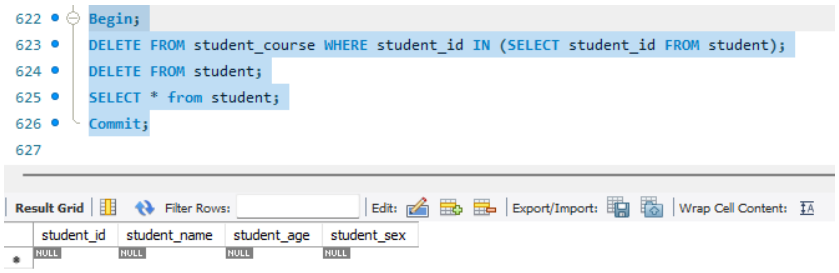
```
616  Begin;
617  DELETE FROM student_course WHERE student_id IN (SELECT student_id FROM student);
618  DELETE FROM student;
619  SELECT * from student;
620  Rollback;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

student_id	student_name	student_age	student_sex
1	Aitor Menta	20	Hombre
2	Benito Camelas	22	Hombre
3	Deborah Dora	19	Mujer
4	Alicia Llaves	21	Mujer
5	Cristina Flores	23	Hombre
6	Alberto Abierto	18	Mujer
7	Victor Buenamigo	24	Hombre
8	Laura Dora	22	Mujer
9	Mike Thompson	21	Hombre
10	Chris Jackson	20	Hombre

COMMIT

```
Begin;
DELETE FROM student_course WHERE student_id IN (SELECT student_id FROM student);
DELETE FROM student;
SELECT * from student;
Commit;
```



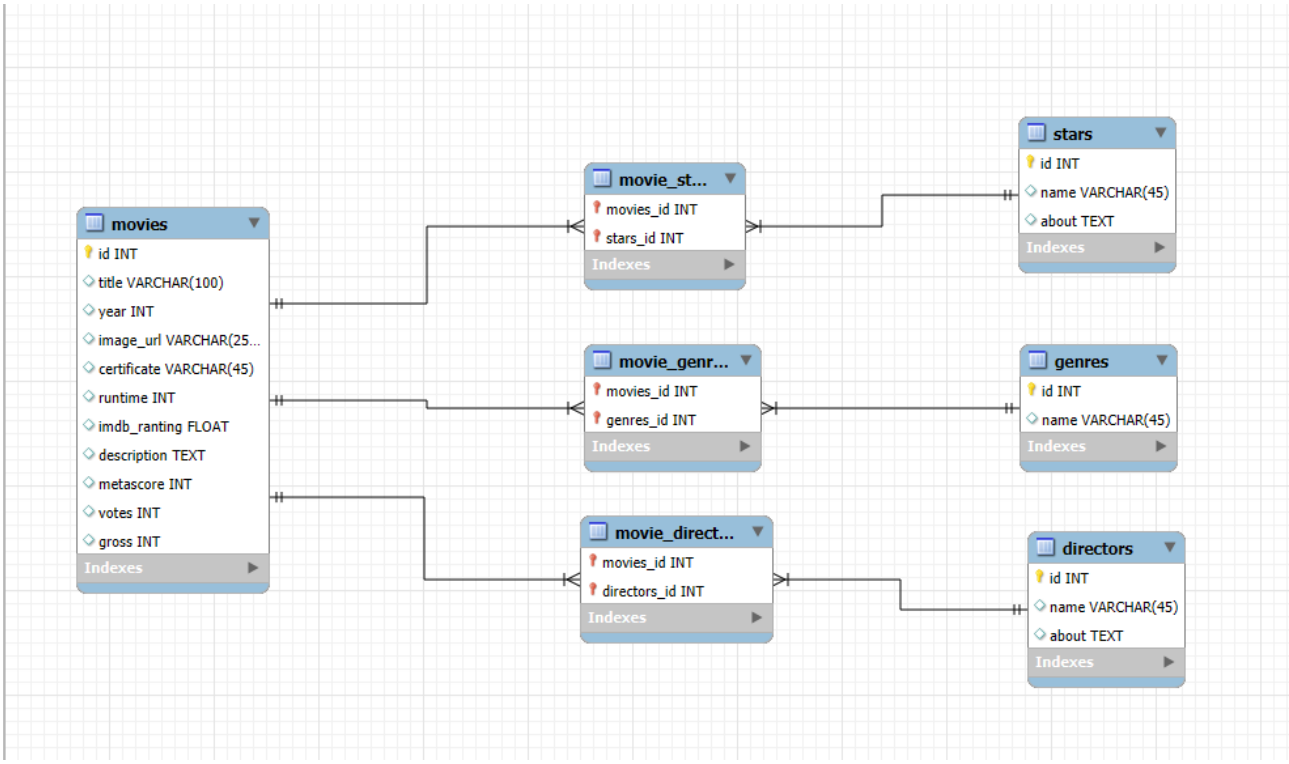
Ejercicio 1. creación de tablas y relaciones entre tablas (PDF)

1.

```
DROP DATABASE IF EXISTS cinema_db;
CREATE DATABASE IF NOT EXISTS cinema_db;

USE cinema_db;

CREATE TABLE IF NOT EXISTS movies(
        id int NOT NULL AUTO_INCREMENT,
    title varchar(100) UNIQUE,
    year int UNIQUE,
    image_url varchar(255) UNIQUE,
    certificate varchar(45),
    runtime int,
    imdb_ranting float,
    description TEXT,
        metascore int,
    votes int,
    gross int,
        PRIMARY KEY (id)
);

CREATE TABLE IF NOT EXISTS directors(
        id int NOT NULL AUTO_INCREMENT,
    name varchar(45) UNIQUE,
    about TEXT,
        PRIMARY KEY (id)
);

CREATE TABLE IF NOT EXISTS stars(
        id int NOT NULL AUTO_INCREMENT,
    name varchar(45) UNIQUE,
    about TEXT,
        PRIMARY KEY (id)
```

```
);

CREATE TABLE IF NOT EXISTS genres(
        id int NOT NULL AUTO_INCREMENT,
    name varchar(45) UNIQUE,
    PRIMARY KEY (id)
);

CREATE TABLE IF NOT EXISTS movie_directors(
        movies_id int,
        directors_id int,
    PRIMARY KEY (movies_id, directors_id),
        FOREIGN KEY (movies_id)
    REFERENCES cinema_db.movies(id),
        FOREIGN KEY (directors_id)
    REFERENCES cinema_db.directors(id)
);

CREATE TABLE IF NOT EXISTS movie_stars(
        movies_id int,
        stars_id int,
    PRIMARY KEY (movies_id, stars_id),
        FOREIGN KEY (movies_id)
    REFERENCES cinema_db.movies(id),
        FOREIGN KEY (stars_id)
    REFERENCES cinema_db.stars(id)
);

CREATE TABLE IF NOT EXISTS movie_genres(
        movies_id int,
        genres_id int,
    PRIMARY KEY (movies_id, genres_id),
        FOREIGN KEY (movies_id)
    REFERENCES cinema_db.movies(id),
        FOREIGN KEY (genres_id)
    REFERENCES cinema_db.genres(id)
);
```

2.



El esquema E/R coincide con el modelo propuesto en el ejercicio.

3.

INSERT INTO movies (title, year, image_url, certificate, runtime, imdb_ranting, description, metascore, votes, gross)
VALUES ('Back to the future', 1985, 'https://www.imdb.com/title/tt0088763/mediaviewer/rm3415438592/?ref_=tt_ov_i', 'PG', 116, 8.5,
'Marty McFly, a 17-year-old high school student, is accidentally sent 30 years into the past in a time-traveling DeLorean invented by his close friend, the maverick scientist Doc Brown.',
87, 999, 384577472),

('The Empire Strikes Back', 1980, 'https://www.imdb.com/title/tt0080684/mediaviewer/rm2002357760/?ref_=tt_ov_i', 'PG', 124, 8.7,
'After the Rebels are brutally overpowered by the Empire on the ice planet Hoth, Luke Skywalker begins Jedi training with Yoda, while his friends are pursued by Darth Vader.', 82, 1170000, 538375067),

('A New Hope', 1977, 'https://www.imdb.com/title/tt0076759/mediaviewer/rm3126483456/?ref_=tt_ov_i', 'PG', 121, 8.6,
'Luke Skywalker joins forces with a Jedi Knight, a cocky pilot, a Wookiee, and two droids to save the galaxy from the Empire\'s world-destroying battle station while also attempting to rescue Princess Leia from the mysterious Darth Vader.', 90, 1300000, 775398007),

('Revenge of the Sith', 2005, 'https://www.imdb.com/title/tt0121766/mediaviewer/rm1940222464/?ref_=tt_ov_i', 'PG-13', 140, 7.6,
'Three years into the Clone Wars, the Jedi rescue Palpatine from Count Dooku. As Obi-Wan pursues a new threat, Anakin acts as a double agent between the Jedi Council and Palpatine and is lured into a sinister plan to rule the galaxy.', 68, 750000, 848998877),

('The Lord of the Rings: The Two Towers', 2002,
'https://www.imdb.com/title/tt0167261/mediaviewer/rm3592950272/?ref_=tt_ov_i', 'PG-13', 179, 8.8,
'While Frodo and Sam edge closer to Mordor with the help of the shifty Gollum, the divided fellowship makes a stand against Sauron\'s new ally, Saruman, and his hordes of Isengard.', 87, 1600000, 947000000),

('How to Train Your Dragon', 2010, 'https://www.imdb.com/title/tt0892769/mediaviewer/rm1396368384/?

ref_=tt_ov_i', 'PG', 98, 8.1,
'A hapless young Viking who aspires to hunt dragons becomes the unlikely friend of a young dragon himself, and learns there may be more to the creatures than he assumed.', 75, 720000, 494878759),

('Shrek 2', 2004, 'https://www.imdb.com/title/tt0298148/mediaviewer/rm1959573504/?ref_=tt_ov_i', 'PG', 93, 7.3,
'Shrek and Fiona travel to the Kingdom of Far Far Away, where Fiona\'s parents are King and Queen, to celebrate their marriage. When they arrive, they find they are not as welcome as they thought they would be.', 75, 680000, 928760770),

('Toy Story 3', 2010, 'https://www.imdb.com/title/tt0435761/mediaviewer/rm3892618240/?ref_=tt_ov_i', 'G', 103, 8.3,
'The toys are mistakenly delivered to a day-care center instead of the attic right before Andy leaves for college, and it\'s up to Woody to convince the other toys that they weren\'t abandoned and to return home.', 92, 890000, 1066969703),

('Ratatouille', 2007, 'https://www.imdb.com/title/tt0382932/mediaviewer/rm1186899712/?ref_=tt_ov_i', 'G', 111, 8.1,
'A rat who can cook makes an unusual alliance with a young kitchen worker at a famous restaurant.', 96, 690000, 620702951),

('The Wizard of Oz', 1939, 'https://www.imdb.com/title/tt0032138/mediaviewer/rm3477868800/?ref_=tt_ov_i', 'G', 102, 8.1,
'Dorothy Gale is swept away from a farm in Kansas to a magical land of Oz in a tornado and embarks on a quest with her new friends to see the Wizard who can help her return home.', 92, 380000, 239000000),

('Jurassic Park', 1993, 'https://www.imdb.com/title/tt0107290/mediaviewer/rm552303360/?ref_=tt_ov_i', 'PG-13', 127, 8.1,
'A pragmatic paleontologist visiting an almost complete theme park is tasked with protecting a couple of kids after a power failure causes the park\'s cloned dinosaurs to run loose.', 68, 920000, 1040290000);

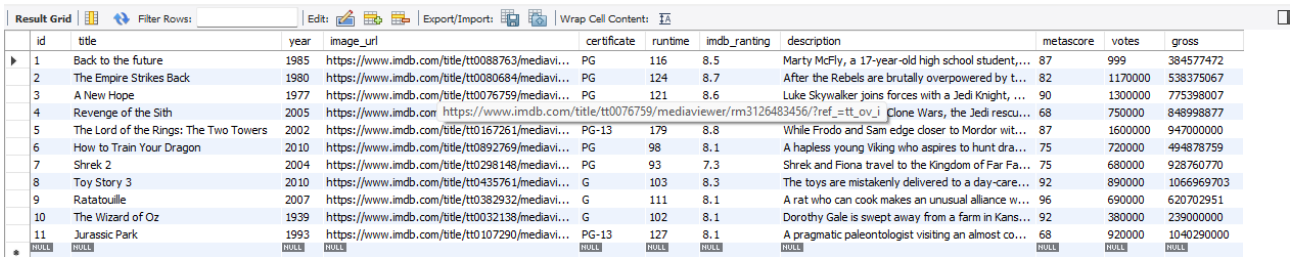
INSERT INTO directors (name, about)
VALUES ('Robert Zemeckis', ' director, productor y guionista estadounidense de cine. En 1984, alcanzó popularidad con su primera película destacada, Romancing the Stone, donde se narra una aventura de una pareja por Colombia.'),
('George Lucas', ' Es un cineasta estadounidense. Lucas es principalmente conocido por crear las franquicias de Star Wars e Indiana Jones y fundar Lucasfilm, LucasArts e Industrial Light & Magic. Cesó como presidente de Lucasfilm antes de venderlo a The Walt Disney Company en 2012.'),
('Peter Jackson', 'Es un director, guionista y productor de cine neozelandés, conocido especialmente por dirigir, producir y coescribir1 la trilogía cinematográfica de El Señor de los Anillos.'),
('Brad Bird', 'Es un director de animación, guionista y director de cine estadounidense. Ha dirigido, entre otras, Ratatouille, Los Increíbles y El gigante de hierro.'),
('Steven Spielberg', 'Es un director, guionista y productor de cine estadounidense.1 Se le considera uno de los pioneros de la era del Nuevo Hollywood y es también uno de los directores más reconocidos y populares de la industria cinematográfica mundial.'),
('Lee Unkrich', 'Es un director de cine y montador estadounidense.'),
('Chris Sanders', 'Es un cineasta, animador y actor de voz estadounidense.'),
('Victor Flemming', 'Fue un director de cine, director de fotografía y productor estadounidense.');

INSERT INTO genres (name)
VALUES ('Ciencia ficcion'),
('Comedia'),
('Familiar'),
('Fantasia'),
('Aventuras'),
('Accion'),
('Animacion'),
('Infantil'),
('Catastrofe'),
('Drama');

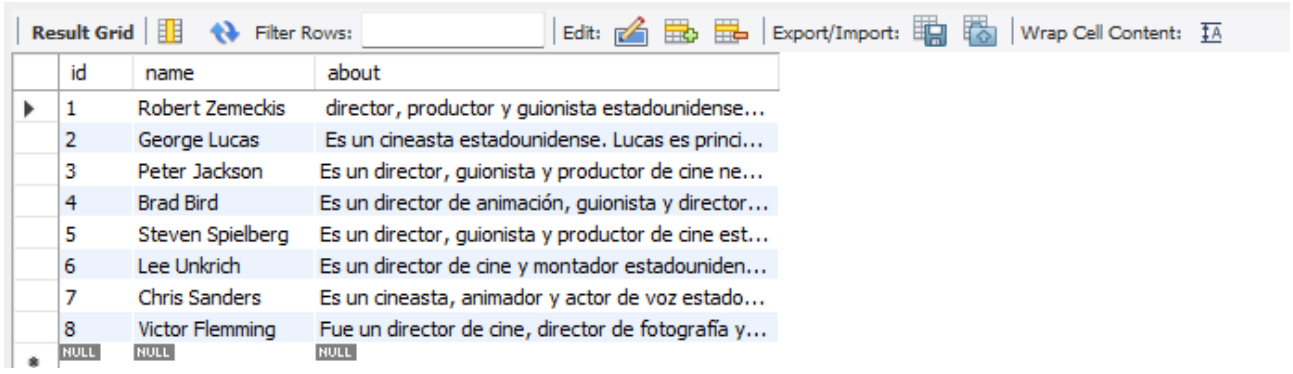
INSERT INTO stars (name, about)
VALUES ('Christopher Lloyd', 'Es un actor estadounidense. Ha interpretado a Doc Emmett Brown en la

```
trilogía de Back to the Future.'),
('Harrison Ford', 'Es un actor, productor, y actor de voz estadounidense de cine y televisión. Es recordado por haber interpretado al personaje de Indiana Jones, y también a Han Solo en la saga de Star Wars'),
('Mark Hamill', 'Es un actor de cine, televisión, voz, director, productor y escritor estadounidense. Es conocido por interpretar a Luke Skywalker en la serie de películas Star Wars.'),
('Ewan McGregor', 'Es un actor, cantante y director de cine británico nacionalizado estadounidense. Es famoso sobre todo por haber protagonizado la película de culto británica Trainspotting (1996).'),
('Elijah Wood', 'Es un actor de cine y televisión estadounidense. Debutó con un papel menor en Back to the Future Part II (1989), y después consiguió una serie de papeles cada vez más relevantes.'),
('Jay Baruchel', 'Es un actor, director, guionista y productor canadiense. Comenzó su carrera tempranamente, a mediados de la década de 1990, en programas de televisión de su país. Después de eso, trabajó en producciones estadounidenses tales como Casi famosos (2000).'),
('Michael John Myers', 'Es un actor, comediante, guionista, productor y director canadiense.1 Ha actuado en Saturday Night Live (1988-1995), Waynes World, en las tres películas de Austin Powers y ha prestado su voz en la serie de películas de Shrek.'),
('Tom Hanks', 's un actor, guionista, productor de cine y director de cine estadounidense.2 Es de los intérpretes más reconocidos de Hollywood. Varias de sus películas, sean dramas o comedias, han recibido el reconocimiento internacional.'),
('Patton Oswald', 'es un actor, humorista de comedia en vivo (stand-up) y guionista estadounidense. Es más conocido por sus papeles como Spencer Olchin en The King of Queens, por haber sido la voz de Remy en la película Ratatouille'),
('Judy Garland', 'Fue una actriz y cantante estadounidense. Si bien fue aclamada por la crítica por muchos papeles diferentes a lo largo de su carrera, es ampliamente conocida por interpretar el papel de Dorothy Gale en The Wizard of Oz (1939).'),
('Sam Neil','Es un actor neozelandés nacido en Reino Unido. Conocido por su papel de Damien Thorn en Omen III: The Final Conflict (La profecía III) y de Alan Grant en Parque Jurásico.');

SELECT * from movies;

INSERT INTO movie_directors (movies_id,        directors_id)
VALUES(1,1),
(2,2),
(3,2),
(4,2),
(5,3),
(6,7),
(7,6),
(8,4),
(9,4),
(10,8),
(11,5);

INSERT INTO movie_stars (movies_id, stars_id)
VALUES(1,1),
(2,2),
(3,3),
(4,4),
(5,5),
(6,6),
(7,7),
(8,8),
(9,9),
(10,10),
(11,11);

INSERT INTO movie_genres (movies_id, genres_id)
VALUES(1,1),
(1,2),
(1,6),
(2,1),
(2,4),
(2,6),
(2,10),
```

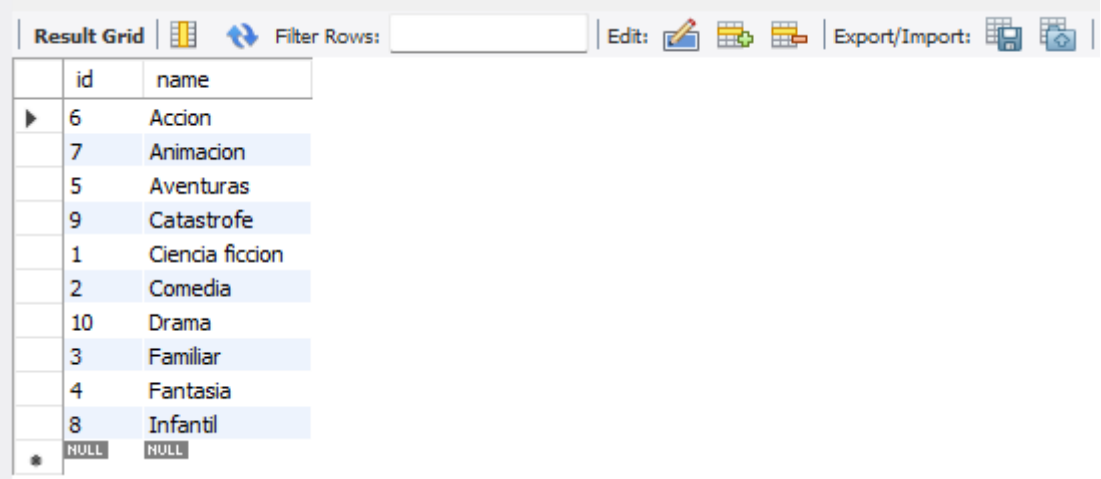
(3,4),
(3,6),
(3,10),
(4,4),
(4,6),
(4,10),
(5,4),
(5,5),
(5,6),
(6,2),
(6,3),
(6,4),
(7,2),
(7,3),
(7,8),
(8,2),
(8,3),
(8,8),
(9,2),
(9,3),
(9,8),
(10,3),
(10,4),
(10,8),
(11,3),
(11,9),
(11,10);



id	title	year	image_url	certificate	runtime	imdb_ranting	description	metascore	votes	gross
1	Back to the future	1985	https://www.imdb.com/title/tt0088763/mediavi...	PG	116	8.5	Marty McFly, a 17-year-old high school student,...	87	999	384577472
2	The Empire Strikes Back	1980	https://www.imdb.com/title/tt0080684/mediavi...	PG	124	8.7	After the Rebels are brutally overpowered by t...	82	1170000	538375067
3	A New Hope	1977	https://www.imdb.com/title/tt0076759/mediavi...	PG	121	8.6	Luke Skywalker joins forces with a Jedi Knight, ...	90	1300000	775398007
4	Revenge of the Sith	2005	https://www.imdb.com/title/tt0076759/mediaviewer/rm3126483456/?ref_=tt_ov_i	Clone Wars, the Jedi rescu...	68	750000	848998877			
5	The Lord of the Rings: The Two Towers	2002	https://www.imdb.com/title/tt0167261/mediavi...	PG-13	179	8.8	While Frodo and Sam edge closer to Mordor wit...	87	1600000	947000000
6	How to Train Your Dragon	2010	https://www.imdb.com/title/tt0892769/mediavi...	PG	98	8.1	A hapless young Viking who aspires to hunt dra...	75	720000	494878759
7	Shrek 2	2004	https://www.imdb.com/title/tt0298148/mediavi...	PG	93	7.3	Shrek and Fiona travel to the Kingdom of Far Fa...	75	680000	928760770
8	Toy Story 3	2010	https://www.imdb.com/title/tt0435761/mediavi...	G	103	8.3	The toys are mistakenly delivered to a day-care...	92	890000	1066969703
9	Ratatouille	2007	https://www.imdb.com/title/tt0382932/mediavi...	G	111	8.1	A rat who can cook makes an unusual alliance w...	96	690000	620702951
10	The Wizard of Oz	1939	https://www.imdb.com/title/tt0032138/mediavi...	G	102	8.1	Dorothy Gale is swept away from a farm in Kans...	92	380000	239000000
11	Jurassic Park	1993	https://www.imdb.com/title/tt0107290/mediavi...	PG-13	127	8.1	A pragmatic paleontologist visiting an almost co...	68	920000	1040290000

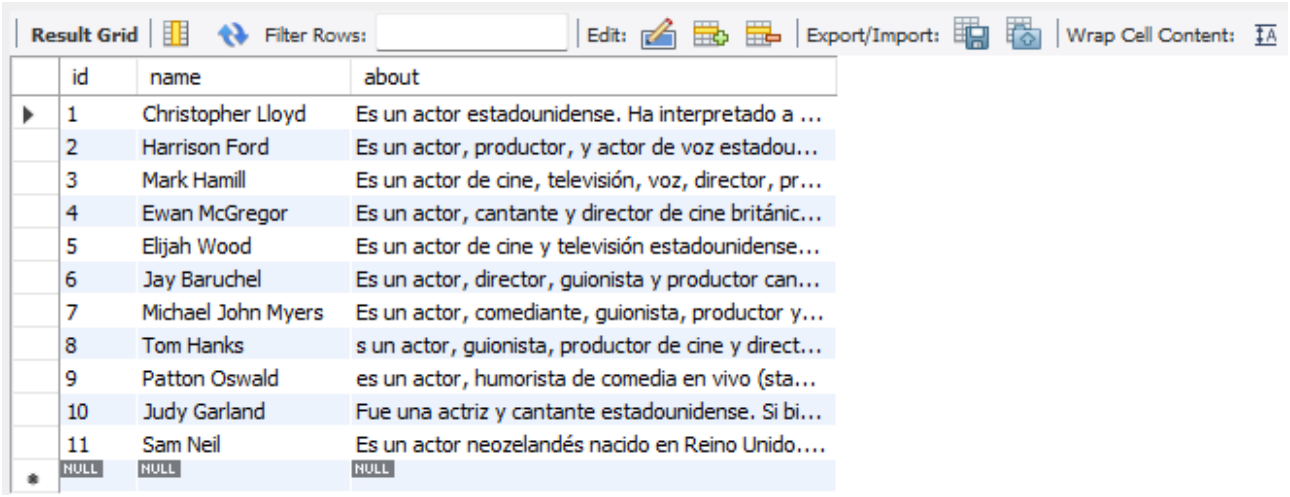


id	name	about
1	Robert Zemeckis	director, productor y guionista estadounidense...
2	George Lucas	Es un cineasta estadounidense. Lucas es princi...
3	Peter Jackson	Es un director, guionista y productor de cine ne...
4	Brad Bird	Es un director de animación, guionista y director...
5	Steven Spielberg	Es un director, guionista y productor de cine est...
6	Lee Unkrich	Es un director de cine y montador estadouniden...
7	Chris Sanders	Es un cineasta, animador y actor de voz estado...
8	Victor Flemming	Fue un director de cine, director de fotografía y...



Result Grid | Filter Rows: | Edit: | Export/Import:

id	name
6	Accion
7	Animacion
5	Aventuras
9	Catastrofe
1	Ciencia ficcion
2	Comedia
10	Drama
3	Familiar
4	Fantasia
8	Infantil
NULL	NULL



Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

id	name	about
1	Christopher Lloyd	Es un actor estadounidense. Ha interpretado a ...
2	Harrison Ford	Es un actor, productor, y actor de voz estadou...
3	Mark Hamill	Es un actor de cine, televisión, voz, director, pr...
4	Ewan McGregor	Es un actor, cantante y director de cine británic...
5	Elijah Wood	Es un actor de cine y televisión estadounidense...
6	Jay Baruchel	Es un actor, director, guionista y productor can...
7	Michael John Myers	Es un actor, comediante, guionista, productor y...
8	Tom Hanks	s un actor, guionista, productor de cine y direct...
9	Patton Oswald	es un actor, humorista de comedia en vivo (sta...
10	Judy Garland	Fue una actriz y cantante estadounidense. Si bi...
11	Sam Neil	Es un actor neozelandés nacido en Reino Unido....
NULL	NULL	NULL

4.

```
CREATE TABLE IF NOT EXISTS movie_directors(
        movies_id int,
        directors_id int,
   PRIMARY KEY (movies_id, directors_id),
        FOREIGN KEY (movies_id)
   REFERENCES cinema_db.movies(id)
   ON DELETE CASCADE,
        FOREIGN KEY (directors_id)
   REFERENCES cinema_db.directors(id)
        ON DELETE CASCADE
);

CREATE TABLE IF NOT EXISTS movie_stars(
        movies_id int,
        stars_id int,
   PRIMARY KEY (movies_id, stars_id),
        FOREIGN KEY (movies_id)
   REFERENCES cinema_db.movies(id)
        ON DELETE CASCADE,
        FOREIGN KEY (stars_id)
   REFERENCES cinema_db.stars(id)
        ON DELETE CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS movie_genres(
        movies_id int,
        genres_id int,
    PRIMARY KEY (movies_id, genres_id),
        FOREIGN KEY (movies_id)
    REFERENCES cinema_db.movies(id)
        ON DELETE CASCADE,
        FOREIGN KEY (genres_id)
    REFERENCES cinema_db.genres(id)
        ON DELETE CASCADE
);

-------------------------------------------------------

DELETE FROM movies;
DELETE FROM stars;
DELETE FROM directors;
DELETE FROM genres;
```