

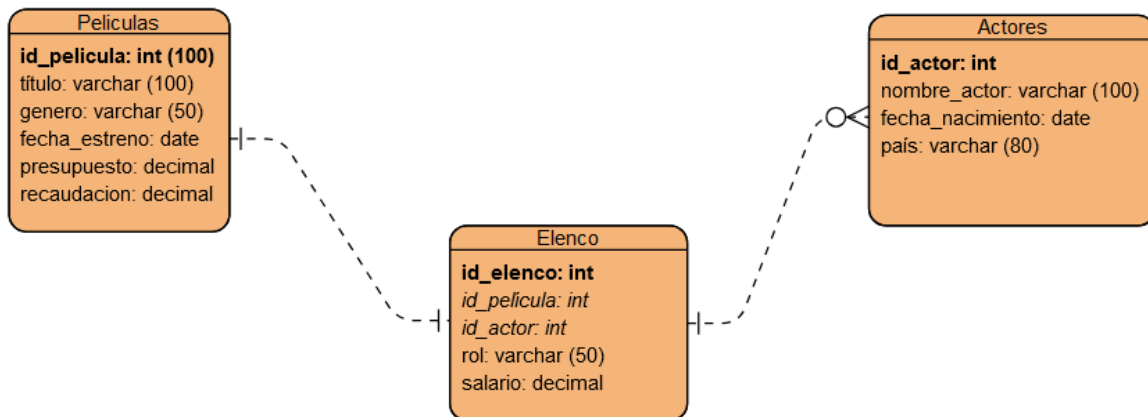
Ejercitación Script y DER

Antonella Burbano

Blanca Vargas
Dalila Mistura

Testing & Automatización
MindHub
2024

Diagrama Entidad - Relación (DER)



SCRIPT: Creación de tablas

- Tabla “Películas”

```
CREATE TABLE "Películas" (  
  "id_pelicula" INT(100) NOT NULL,  
  "titulo" VARCHAR(100) NOT NULL,  
  "genero" VARCHAR(50) NOT NULL,  
  "fecha_estreno" date,  
  "presupuesto" DECIMAL(15, 2),  
  "recaudacion" DECIMAL(15, 2),  
  PRIMARY KEY ("id_pelicula")  
)
```

- Tabla “Actores”

```
CREATE TABLE "Actores" (  
  "id_actor" INT(100) NOT NULL,  
  "nombre_actor" VARCHAR(100) NOT NULL,  
  "pais" VARCHAR(80) NOT NULL,  
  "fecha_nacimiento" date NOT NULL,  
  PRIMARY KEY ("id_actor")  
)
```

- Tabla “Elenco”

```
CREATE TABLE "Elenco" (  
  "id_elenco" INT(100) NOT NULL,  
  "id_pelicula" INT(100) NOT NULL,  
  "id_actor" INT(100) NOT NULL,  
  "rol" VARCHAR(50) NOT NULL,  
  "salario" DECIMAL(15, 2),  
  PRIMARY KEY ("id_elenco"),  
  FOREIGN KEY ("id_pelicula") REFERENCES "Películas"("id_pelicula"),  
  FOREIGN KEY ("id_actor") REFERENCES "Actores"("id_actor")  
)
```

- Captura de la creación de tablas en editor SQL

Programiz Online SQL Editor

Interactive SQL Course

Input

```
CREATE TABLE "Elenco" (
  "id_elenco" INT(100) NOT NULL,
  "id_pelicula" INT(100) NOT NULL,
  "id_actor" INT(100) NOT NULL,
  "rol" VARCHAR(50) NOT NULL,
  "salario" DECIMAL(15, 2),
  PRIMARY KEY ("id_elenco"),
  FOREIGN KEY ("id_pelicula") REFERENCES "Películas"("id_pelicula"),
  FOREIGN KEY ("id_actor") REFERENCES "Actores"("id_actor")
)
```

Output

SQL query successfully executed. However, the result set is empty.

Available Tables

Actores

| id_actor | nombre_actor | pais | fecha_nacimiento |
|----------|--------------|------|------------------|
| empty | | | |

Elenco

| id_elenco | id_pelicula | id_actor | rol | salario |
|-----------|-------------|----------|-----|---------|
| empty | | | | |

Películas

| id_pelicula | titulo | genero | fecha_estreno | presupuesto |
|-------------|--------|--------|---------------|-------------|
| empty | | | | |

SCRIPT: Insertar datos

- Insertar datos a tabla "Películas"

```
INSERT INTO Películas VALUES
(1, 'Inception', 'Ciencia ficción', '2010-07-16', 160000000.00, 829895144.00),
(2, 'The Dark Knight', 'Acción', '2008-07-18', 185000000.00, 1004558444.00),
(3, 'Titanic', 'Romance', '1997-12-19', 200000000.00, 2187463944.00),
(4, 'Avengers: Endgame', 'Acción', '2019-04-26', 356000000.00, 2797800564.00),
(5, 'La La Land', 'Musical', '2016-12-09', 30000000.00, 446050884.00),
(6, 'The Godfather', 'Drama', '1972-03-24', 6000000.00, 246120986.00);
```

Programiz Online SQL Editor

Interactive SQL Course

Input

```
INSERT INTO Películas VALUES
(1, 'Inception', 'Ciencia ficción', '2010-07-16', 160000000.00, 829895144.00),
(2, 'The Dark Knight', 'Acción', '2008-07-18', 185000000.00, 1004558444.00),
(3, 'Titanic', 'Romance', '1997-12-19', 200000000.00, 2187463944.00),
(4, 'Avengers: Endgame', 'Acción', '2019-04-26', 356000000.00, 2797800564.00),
(5, 'La La Land', 'Musical', '2016-12-09', 30000000.00, 446050884.00),
(6, 'The Godfather', 'Drama', '1972-03-24', 6000000.00, 246120986.00);
```

Output

SQL query successfully executed. However, the result set is empty.

Available Tables

Actores

| id_actor | nombre_actor | pais | fecha_nacimiento |
|----------|--------------|------|------------------|
| empty | | | |

Elenco

| id_elenco | id_pelicula | id_actor | rol | salario |
|-----------|-------------|----------|-----|---------|
| empty | | | | |

Películas

| id_pelicula | titulo | genero | fecha_estreno | presupuesto |
|-------------|-------------------|-----------------|---------------|--------------|
| 1 | Inception | Ciencia ficción | 2010-07-16 | 160000000.00 |
| 2 | The Dark Knight | Acción | 2008-07-18 | 185000000.00 |
| 3 | Titanic | Romance | 1997-12-19 | 200000000.00 |
| 4 | Avengers: Endgame | Acción | 2019-04-26 | 356000000.00 |
| 5 | La La Land | Musical | 2016-12-09 | 30000000.00 |
| 6 | The Godfather | Drama | 1972-03-24 | 6000000.00 |

- Insertar datos a tabla “Actores”

```
INSERT INTO Actores VALUES
(1, 'Leonardo DiCaprio', 'Estados Unidos', '1974-11-11'),
(2, 'Joseph Gordon-Levitt', 'Estados Unidos', '1981-02-17'),
(3, 'Christian Bale', 'Reino Unido', '1974-01-30'),
(4, 'Robert Downey Jr.', 'Estados Unidos', '1965-04-04'),
(5, 'Ryan Gosling', 'Canadá', '1980-11-12'),
(6, 'Al Pacino', 'Estados Unidos', '1940-04-25');
```

Programiz Online SQL Editor

Interactive SQL Course

Actores [-]

- id_actor [int(100)]
- nombre_actor [varchar(100)]
- pais [varchar(80)]
- fecha_nacimiento [date]

Elenco [-]

- id_elenco [int(100)]
- id_pelicula [int(100)]
- id_actor [int(100)]
- rol [varchar(50)]
- salario [decimal(15, 2)]

Peliculas [-]

- id_pelicula [int(100)]
- titulo [varchar(100)]
- genero [varchar(50)]
- fecha_estreno [date]
- presupuesto [decimal(15, 2)]
- recaudacion [decimal(15, 2)]

Input

```
INSERT INTO Actores VALUES
(1, 'Leonardo DiCaprio', 'Estados Unidos', '1974-11-11'),
(2, 'Joseph Gordon-Levitt', 'Estados Unidos', '1981-02-17'),
(3, 'Christian Bale', 'Reino Unido', '1974-01-30'),
(4, 'Robert Downey Jr.', 'Estados Unidos', '1965-04-04'),
(5, 'Ryan Gosling', 'Canadá', '1980-11-12'),
(6, 'Al Pacino', 'Estados Unidos', '1940-04-25');
```

Run SQL

Output

SQL query successfully executed. However, the result set is empty.

Available Tables

Actores

| id_actor | nombre_actor | pais | fecha_nacimiento |
|----------|----------------------|----------------|------------------|
| 1 | Leonardo DiCaprio | Estados Unidos | 1974-11-11 |
| 2 | Joseph Gordon-Levitt | Estados Unidos | 1981-02-17 |
| 3 | Christian Bale | Reino Unido | 1974-01-30 |
| 4 | Robert Downey Jr. | Estados Unidos | 1965-04-04 |
| 5 | Ryan Gosling | Canadá | 1980-11-12 |
| 6 | Al Pacino | Estados Unidos | 1940-04-25 |

Elenco

| id_elenco | id_pelicula | id_actor | rol | salario |
|-----------|-------------|----------|-----|---------|
| empty | | | | |

Peliculas

- Insertar datos a tabla “Actores”

```
INSERT INTO Elenco VALUES
(1, 1, 1, 'Dominick Cobb', 20000000.00),
(2, 1, 2, 'Arthur', 7500000.00),
(3, 2, 3, 'Bruce Wayne / Batman', 30000000.00),
(4, 4, 4, 'Tony Stark / Iron Man', 75000000.00),
(5, 5, 5, 'Sebastian Wilder', 8000000.00),
(6, 6, 6, 'Michael Corleone', 5000000.00);
```

Programiz Online SQL Editor

Interactive SQL Course

Actores [-]

- id_actor [int(100)]
- nombre_actor [varchar(100)]
- pais [varchar(80)]
- fecha_nacimiento [date]

Elenco [-]

- id_elenco [int(100)]
- id_pelicula [int(100)]
- id_actor [int(100)]
- rol [varchar(50)]
- salario [decimal(15, 2)]

Peliculas [-]

- id_pelicula [int(100)]
- titulo [varchar(100)]
- genero [varchar(50)]
- fecha_estreno [date]
- presupuesto [decimal(15, 2)]
- recaudacion [decimal(15, 2)]

Input

```
INSERT INTO Elenco VALUES
(1, 1, 1, 'Dominick Cobb', 20000000.00),
(2, 1, 2, 'Arthur', 7500000.00),
(3, 2, 3, 'Bruce Wayne / Batman', 30000000.00),
(4, 4, 4, 'Tony Stark / Iron Man', 75000000.00),
(5, 5, 5, 'Sebastian Wilder', 8000000.00),
(6, 6, 6, 'Michael Corleone', 5000000.00);
```

Run SQL

Output

SQL query successfully executed. However, the result set is empty.

Available Tables

Elenco

| id_elenco | id_pelicula | id_actor | rol | salario |
|-----------|-------------|----------|-----------------------|----------|
| 1 | 1 | 1 | Dominick Cobb | 20000000 |
| 2 | 1 | 2 | Arthur | 7500000 |
| 3 | 2 | 3 | Bruce Wayne / Batman | 30000000 |
| 4 | 4 | 4 | Tony Stark / Iron Man | 75000000 |
| 5 | 5 | 5 | Sebastian Wilder | 8000000 |
| 6 | 6 | 6 | Michael Corleone | 5000000 |

Peliculas

| id_pelicula | titulo | genero | fecha_estreno | presupuesto |
|-------------|-----------|-----------------|---------------|-------------|
| 1 | Inception | Ciencia ficción | 2010-07-16 | 160000 |

SCRIPT: Consultas SQL

- Consultar registros de la tabla “Actores” que los nombres terminen con la letra “o”

```
SELECT nombre_actor, pais
FROM Actores
WHERE nombre_actor LIKE "%o"
```

```
SELECT nombre_actor, pais
FROM Actores
WHERE nombre_actor LIKE "%o"
```

Output

| nombre_actor | país |
|-------------------|----------------|
| Leonardo DiCaprio | Estados Unidos |
| Al Pacino | Estados Unidos |

- Consultar la información de la tabla “Películas” de género Acción y musical.

```
SELECT *
FROM Peliculas
WHERE genero IN ("Acción", "Musical")
```

```
SELECT *
FROM Peliculas
WHERE genero IN ("Acción", "Musical")
```

Output

| id_pelicula | título | genero | fecha_estreno | presupuesto | recaudacion |
|-------------|-------------------|---------|---------------|-------------|-------------|
| 2 | The Dark Knight | Acción | 2008-07-18 | 185000000 | 1004558444 |
| 4 | Avengers: Endgame | Acción | 2019-04-26 | 356000000 | 2797800564 |
| 5 | La La Land | Musical | 2016-12-09 | 30000000 | 446050884 |

- Consultar la información de la tabla “Elenco” filtrado por ID_Pelicula.

```
SELECT id_pelicula, rol, salario
FROM Elenco
WHERE id_pelicula
BETWEEN 2 AND 5
```

```
SELECT id_pelicula, rol, salario
FROM Elenco
WHERE id_pelicula
BETWEEN 2 AND 5
```

Output

| id_pelicula | rol | salario |
|-------------|-----------------------|----------|
| 2 | Bruce Wayne / Batman | 30000000 |
| 4 | Tony Stark / Iron Man | 75000000 |
| 5 | Sebastian Wilder | 8000000 |

- Consultar la información de las 3 tablas (Películas, Actores, Elenco) utilizando el JOIN para unir las

```
SELECT Actores.nombre_actor AS Actor, rol, Peliculas.titulo AS Pelicula
FROM Elenco JOIN Actores ON Elenco.id_actor = Actores.id_actor
JOIN Peliculas ON Elenco.id_pelicula = Peliculas.id_pelicula
```

```
SELECT Actores.nombre_actor AS Actor, rol, Peliculas.titulo AS Pelicula
FROM Elenco JOIN Actores ON Elenco.id_actor = Actores.id_actor
JOIN Peliculas ON Elenco.id_pelicula = Peliculas.id_pelicula
```

Output

| Actor | rol | Pelicula |
|----------------------|-----------------------|-------------------|
| Leonardo DiCaprio | Dominick Cobb | Inception |
| Joseph Gordon-Levitt | Arthur | Inception |
| Christian Bale | Bruce Wayne / Batman | The Dark Knight |
| Robert Downey Jr. | Tony Stark / Iron Man | Avengers: Endgame |
| Ryan Gosling | Sebastian Wilder | La La Land |

- Consultar la información de las 3 tablas (Películas, Actores, Elenco) utilizando el JOIN para unir las, filtrando por género.

```
SELECT Actores.nombre_actor AS Actor, rol, Peliculas.titulo AS Pelicula
FROM Elenco JOIN Actores ON Elenco.id_actor = Actores.id_actor
JOIN Peliculas ON Elenco.id_pelicula = Peliculas.id_pelicula
WHERE Peliculas.genero IN ("Acción", "Ciencia ficción", "Musical")
```

```
SELECT Actores.nombre_actor AS Actor, rol, Peliculas.titulo AS Pelicula
FROM Elenco JOIN Actores ON Elenco.id_actor = Actores.id_actor
JOIN Peliculas ON Elenco.id_pelicula = Peliculas.id_pelicula
WHERE Peliculas.genero IN ("Acción", "Ciencia ficción", "Musical")
```

Output

| Actor | rol | Pelicula |
|----------------------|-----------------------|-------------------|
| Leonardo DiCaprio | Dominick Cobb | Inception |
| Joseph Gordon-Levitt | Arthur | Inception |
| Christian Bale | Bruce Wayne / Batman | The Dark Knight |
| Robert Downey Jr. | Tony Stark / Iron Man | Avengers: Endgame |
| Ryan Gosling | Sebastian Wilder | La La Land |

- Consultar la información de las 3 tablas (Películas, Actores, Elenco) utilizando el JOIN para unir las, filtrando por fecha de estreno.

```
SELECT Actores.nombre_actor AS Actor, Peliculas.fecha_estreno, Peliculas.titulo AS Pelicula
FROM Elenco JOIN Actores ON Elenco.id_actor = Actores.id_actor
JOIN Peliculas ON Elenco.id_pelicula = Peliculas.id_pelicula
WHERE Peliculas.fecha_estreno > '2008-01-01'
```

```
SELECT Actores.nombre_actor AS Actor, Peliculas.fecha_estreno, Peliculas.titulo AS Pelicula
FROM Elenco JOIN Actores ON Elenco.id_actor = Actores.id_actor
JOIN Peliculas ON Elenco.id_pelicula = Peliculas.id_pelicula
WHERE Peliculas.fecha_estreno > '2008-01-01'
```

Output

| Actor | fecha_estreno | Pelicula |
|----------------------|---------------|-------------------|
| Leonardo DiCaprio | 2010-07-16 | Inception |
| Joseph Gordon-Levitt | 2010-07-16 | Inception |
| Christian Bale | 2008-07-18 | The Dark Knight |
| Robert Downey Jr. | 2019-04-26 | Avengers: Endgame |
| Ryan Gosling | 2016-12-09 | La La Land |

- Actualizar la información de la tabla “Películas”, filtrando por id_Pelicula.

```
UPDATE Peliculas
SET genero = "Romance"
WHERE id_pelicula = 3;
```

```
UPDATE Peliculas
SET genero = "Romance"
WHERE id_pelicula = 3;
```

| | | | | |
|---|---|---|-----------------------|------|
| 4 | 4 | 4 | Tony Stark / Iron Man | 7500 |
| 5 | 5 | 5 | Sebastian Wilder | 8000 |
| 6 | 6 | 6 | Michael Corleone | 5000 |

Películas

| id_pelicula | titulo | genero | fecha_estreno |
|-------------|-------------------|-----------------|---------------|
| 1 | Inception | Ciencia ficción | 2010-07-16 |
| 2 | The Dark Knight | Acción | 2008-07-18 |
| 3 | Titanic | Romance | 1997-12-19 |
| 4 | Avengers: Endgame | Acción | 2019-04-26 |
| 5 | La La Land | Musical | 2016-12-09 |

Output

SQL query successfully executed. However, the result set is empty.

- Ordenar registros en orden ascendente

```
SELECT nombre_actor, pais, fecha_nacimiento
FROM Actores
ORDER BY pais
```

```
SELECT nombre_actor, pais, fecha_nacimiento
FROM Actores
ORDER BY pais
```

Output

| nombre_actor | pais | fecha_nacimiento |
|----------------------|----------------|------------------|
| Ryan Gosling | Canadá | 1980-11-12 |
| Leonardo DiCaprio | Estados Unidos | 1974-11-11 |
| Joseph Gordon-Levitt | Estados Unidos | 1981-02-17 |
| Robert Downey Jr. | Estados Unidos | 1965-04-04 |
| Al Pacino | Estados Unidos | 1940-04-25 |
| Christian Bale | Reino Unido | 1974-01-30 |

- Eliminar Elenco por Id_elenco.

DELETE FROM Elenco WHERE id_elenco = 6

```
DELETE FROM Elenco WHERE id_elenco = 6
```

Output

SQL query successfully executed. However, the result set is empty.

| | | | |
|---|--------------|----------------|------------|
| 5 | Ryan Gosling | Canadá | 1980-11-12 |
| 6 | Al Pacino | Estados Unidos | 1940-04-25 |

Elenco

| id_elenco | id_pelicula | id_actor | rol | salario |
|-----------|-------------|----------|-----------------------|----------|
| 1 | 1 | 1 | Dominick Cobb | 20000000 |
| 2 | 1 | 2 | Arthur | 7500000 |
| 3 | 2 | 3 | Bruce Wayne / Batman | 30000000 |
| 4 | 4 | 4 | Tony Stark / Iron Man | 75000000 |
| 5 | 5 | 5 | Sebastian Wilder | 8000000 |