

Modulo 12: Base de Datos

Actividad 22:

Ejercitación Script y DER

Evidencias con captura de pantalla

Presentado por:

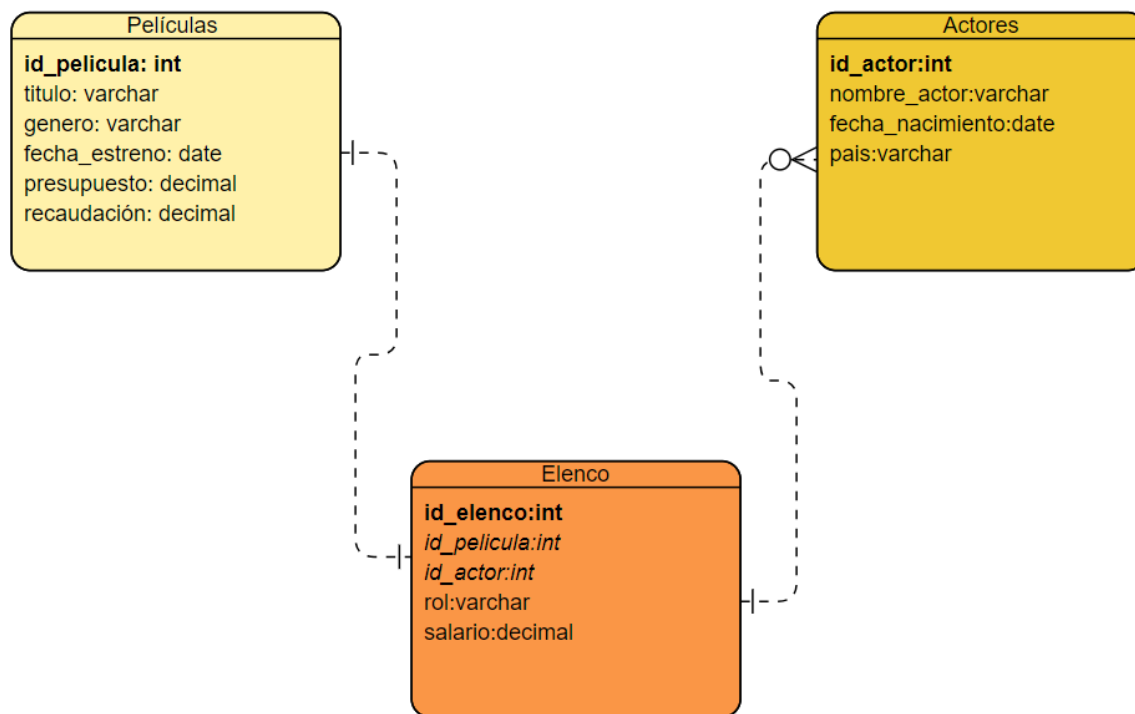
Leidy Dayana Tacuma Mina

Instructoras:

Blanca Vargas

Dalila Mistura

Diseño del DER de Base de Datos para un Estudio de Cine



En esta maqueta se puede observar el diseño de la base de datos para un estudio de cine, que permite la gestión de información de las Películas, Actores y su Elenco, como se puede observar en la imagen se proponen tres tablas principales para el diseño de la base de datos.

CAPTURA DE PANTALLA DE LA CREACIÓN DE TABLAS: Películas, Actores y Elenco.

Captura de pantalla de los Scripts de creación de tabla Películas

The screenshot shows the Programiz Online SQL Editor interface. The browser address bar displays 'programiz.com/sql/online-compiler/'. The editor has a top navigation bar with the Programiz logo, 'Online SQL Editor', and a button for 'Interactive SQL Course'. The main workspace is divided into three panels: 'Input', 'Output', and 'Available Tables'.

Input Panel: Contains the SQL script to create the 'Películas' table:

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

Output Panel: Displays the message: "SQL query successfully executed. However, the result set is empty."

Available Tables Panel: Shows a table named 'Películas' with the following columns: id_pelicula, titulo, genero, fecha_estreno, presupuesto, and an empty row below.

Left Sidebar: A tree view showing the structure of the 'Películas' table with the following fields and their data types:

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

Esta tabla almacena información de las películas.

Captura de pantalla de los Scripts de creación de tabla Actores

programiz.com/sql/online-compiler/

Programiz
Online SQL Editor

Interactive SQL Course

Actores [-]

- id_actor [int]
- nombre_actor [varchar(100)]
- fecha_nacimiento [date]
- pais [varchar(50)]

Películas [-]

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

Input

```
CREATE TABLE Actores (  
  id_actor INT ,  
  nombre_actor VARCHAR(100) NOT NULL,  
  fecha_nacimiento DATE,  
  pais VARCHAR(50),  
  PRIMARY KEY ("id_actor")  
);
```

Run SQL

Available Tables

Actores

id_actor	nombre_actor	fecha_nacimiento	pais
empty			

Películas

id_pelicula	titulo	genero	fecha_estreno	presupuesto	re
empty					

Output

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

Esta tabla almacena datos de los actores.

Captura de pantalla de los Scripts de creación de tabla relacional Elenco

Programiz

Online SQL Editor

Interactive SQL Course

Actores [-]

id_actor [int]

nombre_actor [varchar(100)]

fecha_nacimiento [date]

pais [varchar(50)]

Elenco [-]

id_elenco [int]

id_pelicula [int]

id_actor [int]

rol [varchar(100)]

salario [decimal(10,2)]

Peliculas [-]

id_pelicula [int]

titulo [varchar(100)]

genero [varchar(50)]

fecha_estreno [date]

presupuesto [decimal(15, 2)]

recaudacion [decimal(15, 2)]

Input

```
CREATE TABLE "Elenco"(  
  id_elenco INT,  
  id_pelicula INT,  
  id_actor INT,  
  rol VARCHAR(100),  
  salario DECIMAL(10,2),  
  PRIMARY KEY ("id_elenco"),  
  FOREIGN KEY ("id_pelicula") REFERENCES Peliculas("id_pelicula"),  
  FOREIGN KEY ("id_actor") REFERENCES Actores(" id_actor")  
);
```

Run SQL

Available Tables

Actores

id_actor	nombre_actor	fecha_nacimiento	pais
empty			

Elenco

id_elenco	id_pelicula	id_actor	rol	salario
empty				

Peliculas

id_pelicula	titulo	genero	fecha_estreno	presupuesto	re
empty					

Output

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

La tabla Elenco contiene la tabla Películas y Actores, relaciona.

CAPTURA DE PANTALLA SOBRE AÑADIR INFORMACIÓN A LAS TABLAS: Películas, Actores y su Elenco

Captura de pantalla de agregar información a la tabla Películas

programiz.com/sql/online-compiler/

Programiz
Online SQL Editor

Interactive SQL Course

Actores [-]

id_actor [int]

nombre_actor [varchar(100)]

fecha_nacimiento [date]

pais [varchar(50)]

Elenco [-]

id_elenco [int]

id_pelicula [int]

id_actor [int]

rol [varchar(100)]

salario [decimal(10,2)]

Películas [-]

id_pelicula [int]

titulo [varchar(100)]

genero [varchar(50)]

fecha_estreno [date]

presupuesto [decimal(15, 2)]

recaudacion [decimal(15, 2)]

Input

```
INSERT INTO "Películas"(
  id_pelicula,titulo, genero, fecha_estreno, presupuesto, recaudacion)
VALUES(01, "el_conjuro", "terror", "19/07/2013", "20.000.000", "319.000.000"),
(02, "la_propuesta", "romance", "10/07/2009", "20.000.000", "340.000.000"),
(03, "Deadpool", "accion", "11/02/2016", "58.000.000", "1.300.000.000")
;
```

Run SQL

Output

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

Available Tables

Actores

id_actor	nombre_actor	fecha_nacimiento	pais
empty			

Elenco

id_elenco	id_pelicula	id_actor	rol	salario
empty				

Películas

id_pelicula	titulo	genero	fecha_estreno	presup
1	el_conjuro	terror	19/07/2013	20.000.
2	la_propuesta	romance	10/07/2009	20.000.
3	Deadpool	accion	11/02/2016	58.000.

presupuesto	recaudacion
20.000.000	319.000.000
20.000.000	340.000.000
58.000.000	1.300.000.000

Captura de pantalla de agregar información a la tabla Actores

Programiz

Online SQL Editor

Interactive SQL Course

Actores [-]

id_actor [int]

nombre_actor [varchar(100)]

fecha_nacimiento [date]

pais [varchar(50)]

Elenco [-]

id_elenco [int]

id_pelicula [int]

id_actor [int]

rol [varchar(100)]

salario [decimal(10, 2)]

Peliculas [-]

id_pelicula [int]

titulo [varchar(100)]

genero [varchar(50)]

fecha_estreno [date]

presupuesto [decimal(15, 2)]

recaudacion [decimal(15, 2)]

Input

```
INSERT INTO "Actores"(id_actor,nombre_actor, fecha_nacimiento,pais)
VALUES(1,"Vera_Farmiga","06/08/1973", "USA"),
(2,"Sandra_Bullock","26/07/1964","USA"),
(3,"Ryan_Reynolds","23/10/1976","Canada");
```

Run SQL

Available Tables

Output

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

Actores

id_actor	nombre_actor	fecha_nacimiento	pais
1	Vera_Farmiga	06/08/1973	USA
2	Sandra_Bullock	26/07/1964	USA
3	Ryan_Reynolds	23/10/1976	Canada

Elenco

id_elenco	id_pelicula	id_actor	rol	salario
empty				

Peliculas

id_pelicula	titulo	genero	fecha_estreno	presupuesto
1	el_conjuro	terror	19/07/2013	20.000.
2	la_propuesta	romance	10/07/2009	20.000.
3	Deadpool	accion	11/02/2016	58.000.

Captura de pantalla de agregar información a la tabla Elenco

Programiz

Online SQL Editor

Interactive SQL Course

Actores [-]

id_actor [int]

nombre_actor [varchar(100)]

fecha_nacimiento [date]

pais [varchar(50)]

Elenco [-]

id_elenco [int]

id_pelicula [int]

id_actor [int]

rol [varchar(100)]

salario [decimal(10, 2)]

Peliculas [-]

id_pelicula [int]

titulo [varchar(100)]

genero [varchar(50)]

fecha_estreno [date]

presupuesto [decimal(15, 2)]

recaudacion [decimal(15, 2)]

Input

```
1 INSERT INTO "Elenco" (
  id_pelicula, id_actor, rol, salario)
  VALUES (1,1,1,"protagonista","1.000.000"),
  (2,2,2,"protagonista","900.000"),
  (3,3,3,"protagonista","7.000.000);
```

Run SQL

Available Tables

Output

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

Actores

id_actor	nombre_actor	fecha_nacimiento	pais
1	Vera_Farmiga	06/08/1973	USA
2	Sandra_Bullock	26/07/1964	USA
3	Ryan_Reynolds	23/10/1976	Canada

Elenco

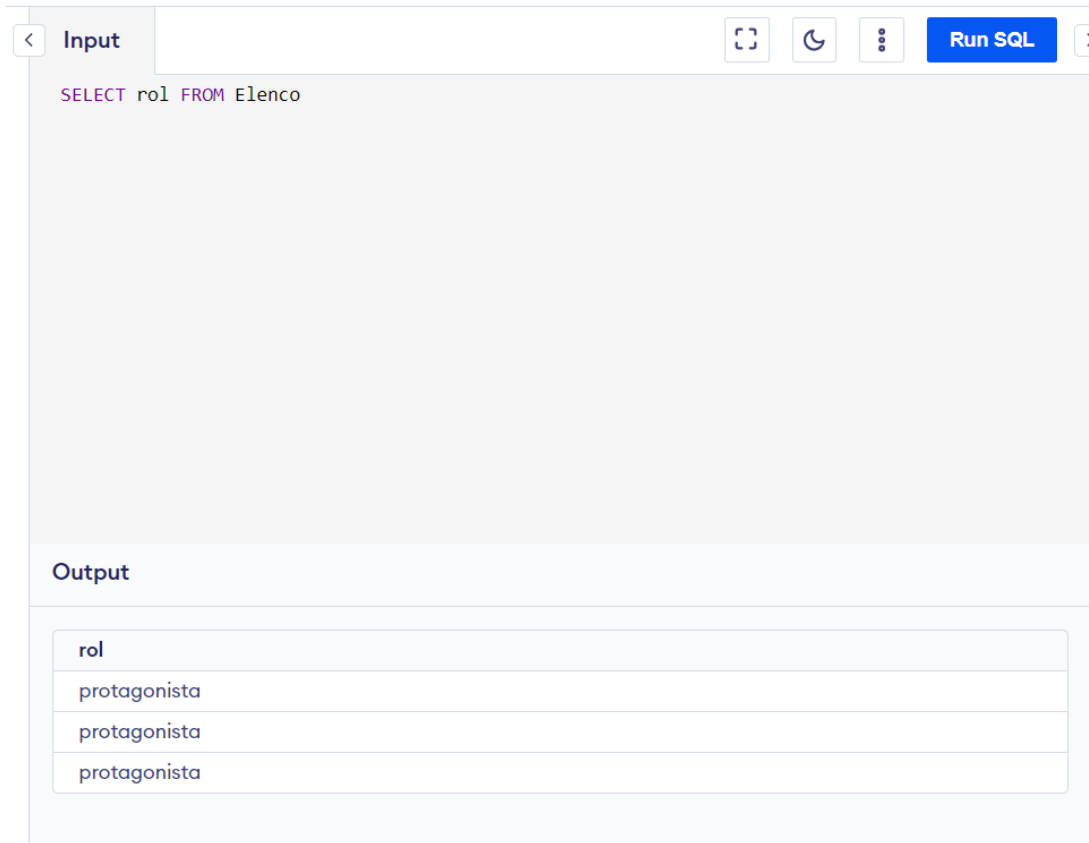
id_elenco	id_pelicula	id_actor	rol	salario
1	1	1	protagonista	1.000.000
2	2	2	protagonista	900
3	3	3	protagonista	7.000.000

Peliculas

id_pelicula	titulo	genero	fecha_estreno	presupuesto
1	el_conjuro	terror	19/07/2013	20.000.
2	la_propuesta	romance	10/07/2009	20.000.
3	Deadpool	accion	11/02/2016	58.000.

CAPTURA DE PANTALLA SOBRE CONSULTAS

Captura de consultas con SELECT



The screenshot shows a web-based SQL interface. At the top, there is a toolbar with icons for undo, redo, and a 'Run SQL' button. Below the toolbar is a large text area for the SQL query. The query entered is 'SELECT rol FROM Elenco'. Below the query area is a section labeled 'Output' which displays the results of the query in a table format. The table has one column named 'rol' and three rows, all containing the value 'protagonista'.



```
SELECT rol FROM Elenco
```

rol
protagonista
protagonista
protagonista

Esta consulta selecciona todos los registros de la tabla Elenco, mostrando información del rol.

Captura de consultas con SELECT cuando el id_actor usando el BETWEEN, AND para determinar un intervalo de valores en específico.

Input



Run SQL

```
SELECT rol,salario
FROM Elenco
WHERE id_actor BETWEEN 1 AND 3
```

Output



Available Tables

rol	salario
protagonista	1.000.000
protagonista	900
protagonista	7.000.000

Aquí se selecciona todos los id_actor cuando este entre 1 y 10, se utiliza la cláusula BETWEEN para definir su rango.

Captura de la consulta JOIN para combinar tablas

Input



Run SQL

```
SELECT titulo, genero, fecha_estreno
FROM Peliculas
WHERE id_pelicula
BETWEEN 1 AND 3
ORDER BY fecha_estreno DESC
```



Output

Available Tables

titulo	genero	fecha_estreno
el_conjuro	terror	19/07/2013
Deadpool	accion	11/02/2016
la_propuesta	romance	10/07/2009

Esa consulta utiliza JOIN para combinar tabla de Películas, mostrando título, genero y fecha con un orden descendente con respecto a su fecha_estreno.

Input



Run SQL

```
SELECT genero
FROM Elenco JOIN Peliculas ON Elenco.id_pelicula=Peliculas.id_pelicula
WHERE Peliculas.id_pelicula>=2
```

Output



Available Tables

genero
romance
accion

Esa consulta utiliza JOIN para combinar tabla de Películas, mostrando el género de películas que tengan id_peliculas >=2.

Captura de pantalla de consulta básica con SELECT*

Input



Run SQL

```
SELECT * FROM Actores
WHERE id_actor=1
```

Output



Available Tables

id_actor	nombre_actor	fecha_nacimiento	pais
1	Vera_Farmiga	06/08/1973	USA

Esta consulta selecciona todos los registros de la tabla Actores, cuando el id_actor=1 mostrando su información completa.

Captura de pantalla usando DELETE, para eliminar registro en la tabla Elenco

Input



Run SQL

```
DELETE FROM Elenco
WHERE id_elenco=2
```

Output

Available Tables

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

Se elimina el registro específico del Elenco con id_elenco=2.

Antes de eliminar registro específico

Elenco

id_elenco	id_pelicula	id_actor	rol	salario
1	1	1	protagonista	1.000.000
2	2	2	protagonista	900
3	3	3	protagonista	7.000.000


Después de eliminar registro específico

Elenco

id_elenco	id_pelicula	id_actor	rol	salario
1	1	1	protagonista	1.000.000
3	3	3	protagonista	7.000.000

Captura de pantalla con UPDATE para actualizar información de la tabla actores seleccionado la columna país.

Input



Run SQL

```
UPDATE Actores SET pais="EE.UU"  
WHERE pais="USA"
```

Output

Available Tables

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

Se actualiza país de la tabla Actores, modificando USA por EE.UU.

Antes de actualizar la tabla Actores en la columna país.

Actores			
id_actor	nombre_actor	fecha_nacimiento	pais
1	Vera_Farmiga	06/08/1973	USA
2	Sandra_Bullock	26/07/1964	USA
3	Ryan_Reynolds	23/10/1976	Canada

Después de actualizar la tabla Actores columna país.

Actores			
id_actor	nombre_actor	fecha_nacimiento	pais
1	Vera_Farmiga	06/08/1973	EE.UU
2	Sandra_Bullock	26/07/1964	EE.UU
3	Ryan_Reynolds	23/10/1976	Canada