

UNIVERSIDAD TECNOLÓGICA DE PANAMA
FACULTAD EN INGENIERÍA DE SISTEMAS COMPUTACIONALES
LICENCIATURA EN INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
BASE DE DATOS II

8
20

QUIZ (Recuperación Parcial)

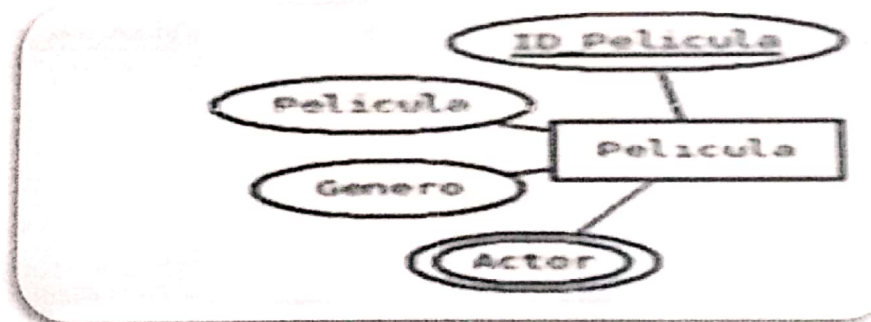
Nom/Cedula: Joel B. Grupo: ITF-B1

Nom/Cedula: Michael Sols Fecha: 11/11/2022

APLIQUE LAS TRES FORMAS NORMALES AL CASO DADO Y MUESTRE EL MODELO DE DATOS
NORMALIZADO AL FINAL DEL PROCESO, LUEGO IMPLEMENTE LA PROGRAMACIÓN
ALMACENA CORRESPONDIENTE PARA ADMINISTRAR LOS DATOS.

SEA CREATIVO EN SU IMPLEMENTACIÓN

Caso No. 1



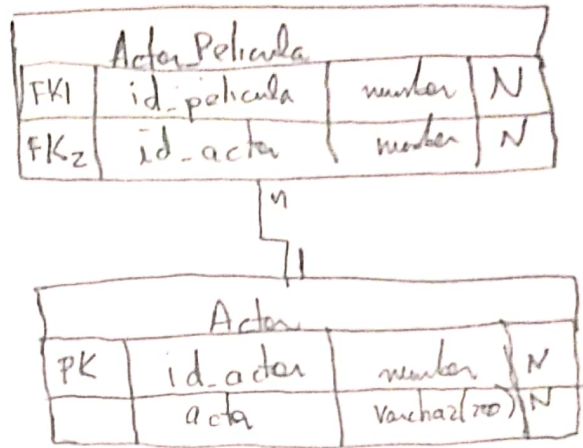
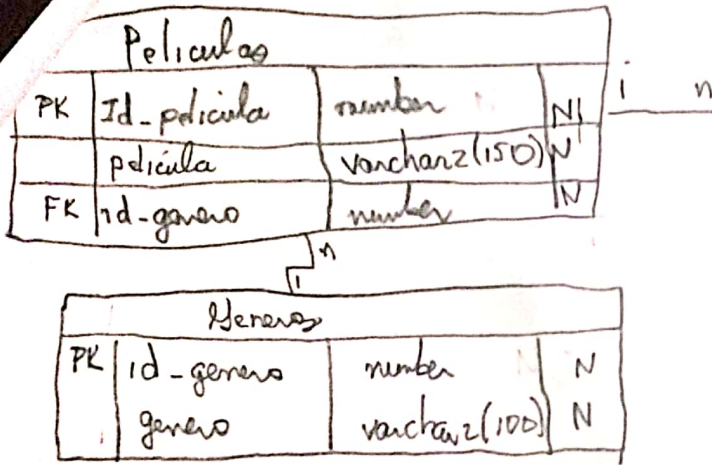
Alta funciones, trigger

Modelo de Datos Normalizado a3 FN

Grupo: IIF-131

1 8-914-582

alés 8-958-1219



Implementación de Quiz de Recuperación I Parcial BDII

ta
olis

8-914-587

8-958-1219

Grupo: 11F-131

```
CREATE TABLE genero (
  id_genero number PRIMARY KEY NOT NULL,
  genero varchar(100) NOT NULL,
);
```

```
CREATE TABLE pelicula (
  id_pelicula number PRIMARY KEY NOT NULL,
  pelicula varchar(100) NOT NULL,
  id_genero number,
  CONSTRAINT id_genero_fk FOREIGN KEY (id_genero) REFERENCES genero (id_genero)
);
```

```
CREATE TABLE actor (
  id_actor number PRIMARY KEY NOT NULL,
  actor varchar(100) NOT NULL,
);
```

```
CREATE TABLE actor_pelicula (
  id_pelicula number NOT NULL,
  id_actor number NOT NULL,
  CONSTRAINT actor_pelicula_fk PRIMARY KEY (id_pelicula, id_actor),
  CONSTRAINT id_pelicula_fk FOREIGN KEY (id_pelicula) REFERENCES pelicula (id_pelicula),
  CONSTRAINT id_actor_fk FOREIGN KEY (id_actor) REFERENCES actor (id_actor)
);
```

-- Secuencia para la Tabla Genero

```
CREATE SEQUENCE id_genero
START WITH 1
INCREMENT BY 1;
```

-- Secuencia para la Tabla Película

```
CREATE SEQUENCE id_pelicula
START WITH 1
INCREMENT BY 1;
```

-- Secuencia para la Tabla actor

```
CREATE SEQUENCE id_actor
START WITH 1
INCREMENT
```

-- Procedimientos para Tabla genero

```
CREATE OR REPLACE PROCEDURE pr_genero (
  p_genero genero.genero%TYPE,
)
```

```
AS
BEGIN
```

```
INSERT INTO genero (id_genero, genero) VALUES (id_genero.nextval, p_genero);
```

```
EXCEPTIONS
```

```
WHEN dup_val_or_index THEN
```

```
  dbms_output.put_line('Datos Repetido');
```

```
WHEN OTHERS THEN
```

```
  dbms_output.put_line('Error causado por el tamaño de los datos ingresados');
```

```
end pr_pelicula;
```

-- Continúa en la siguiente página

-- Procedimiento de almacenamiento para la Tabla película

```
CREATE OR REPLACE PROCEDURE pr_pelicula(
```

```
p_pelicula película, película % TYPE
```

```
);
```

```
AS
```

```
BEGIN
```

```
INSERT INTO película(id_pelicula, película) VALUES (id_pelicula.next val, p_pelicula);
```

```
EXCEPTIONS
```

```
WHEN DUP_VAL_ON_INDEX THEN
```

```
dbms_output.put_line('Datos repetidos');
```

```
WHEN OTHERS THEN
```

```
dbms_output.put_line('Error causado por el tamaño de los datos ingresados');
```

```
end pr_pelicula;
```

```
/
```

-- Procedimiento de almacenamiento para la Tabla actor

```
CREATE OR REPLACE PROCEDURE pr_actor(
```

```
p_actor actor, actor % TYPE
```

```
);
```

```
AS
```

```
BEGIN
```

```
INSERT INTO actor(id_actor, actor) VALUES (id_actor.next val, p_actor);
```

```
EXCEPTIONS
```

```
WHEN DUP_VAL_ON_INDEX THEN
```

```
dbms_output.put_line('Datos repetidos');
```

```
WHEN OTHERS THEN
```

```
dbms_output.put_line('Error causado por el tamaño de los datos ingresados');
```

```
end pr_actor;
```

```
/
```

-- Bloque anónimo de Inserción de Datos de película

```
DECLARE
```

```
v_actor actor.actor % TYPE;
```

```
BEGIN
```

```
v_actor := '&pelicula';
```

```
pr_pelicula(v_pelicula);
```

```
END;
```

-- Bloque anónimo de Inserción de Datos de género

```
DECLARE
```

```
v_genero genero.genero % TYPE;
```

```
BEGIN
```

```
v_genero := '&genero';
```

```
pr_genero(v_genero);
```

```
END;
```

-- Bloque anónimo de Inserción de Datos de actor

```
DECLARE
```

```
v_actor actor.actor % TYPE;
```

```
BEGIN
```

```
v_actor := '&actor';
```

```
pr_actor(v_actor);
```

```
END;
```

-- CREACIÓN de Vista de Datos Almacenados e Ingresados

```
CREATE VIEW ver_pelicula
```

```
AS
```

```
Select p.id_pelicula as numero_de_pelicula, p.pelicula, g.genero, a.actor as actor from película p
```

```
Inner join genero g on p.id_pelicula = g.id_genero
```

```
Inner join actor a on a.id_actor = p.id_pelicula;
```

-- Muestra de la Vista

```
{ Select * from ver_pelicula; }
```