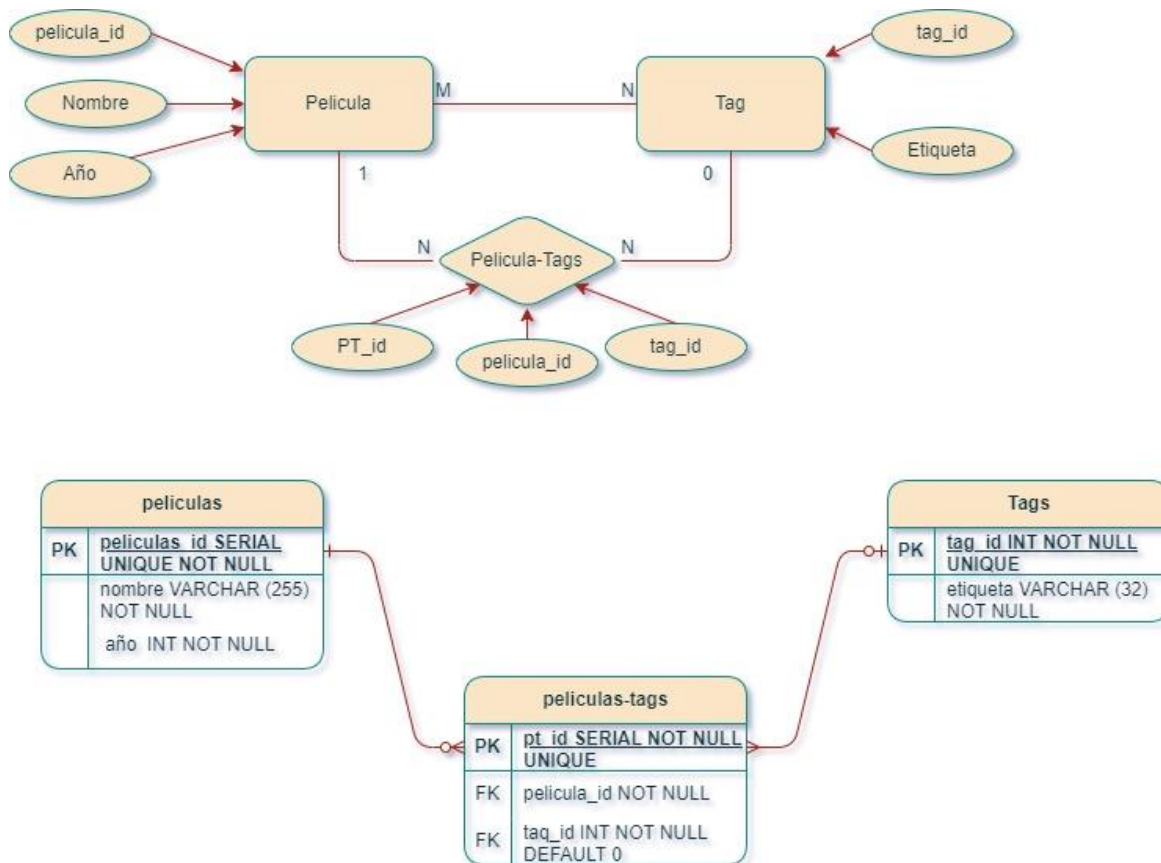


Prueba - Fundamentos de bases de datos relacionales

Nombre: Nathalie Moraga

1. Crea el modelo (revisa bien cuál es el tipo de relación antes de crearlo), respeta las claves primarias, foráneas y tipos de datos. (1 punto)



Se crea database.

```
prueba-Nathalie-Moraga-045=# \c prueba-Nathalie-Moraga-045
Ahora está conectado a la base de datos «prueba-Nathalie-Moraga-045» con el usuario «postgres».
prueba-Nathalie-Moraga-045=#
```

Se crea tablas

2. Inserta 5 películas y 5 tags, la primera película tiene que tener 3 tags asociados, la segunda película debe tener dos tags asociados. (1 punto)

```
prueba-Nathalie-Moraga-045=# SELECT * FROM peliculas;
 pelicula_id | nombre | año
-----+-----+-----
1 | the_Lord_of_the_Rings_The_Fellowship_of_the_Ring | 2001
2 | suzume_no_tojimari | 2022
3 | titanic | 1998
4 | Fight_Club | 2011
5 | The_Conjuring | 2013
(5 filas)
```

```
prueba-Nathalie-Moraga-045=# SELECT * FROM peliculas_tags;
```

pt_id	pelicula_id	tag_id
1	1	1
2	1	4
3	1	5
4	2	3
5	2	4
6	4	5
7	5	2
8	3	

(8 filas)

2. Cuenta la cantidad de tags que tiene cada película. Si una película no tiene tags debe mostrar 0. (1 punto)

pelicula_id	count
1	3
2	2
5	1
4	1
3	0

(5 filas)

3. Crea las tablas respetando los nombres, tipos, claves primarias y foráneas y tipos de datos. (1 punto)

```
CREATE TABLE
```

```
prueba-Nathalie-Moraga-045=# SELECT * FROM preguntas;
```

pregunta_id	pregunta	respuesta_correcta
-------------	----------	--------------------

(0 filas)

```
prueba-Nathalie-Moraga-045=# SELECT * FROM usuarios;
```

usuario_id	nombre	edad
------------	--------	------

(0 filas)

```
prueba-Nathalie-Moraga-045=# SELECT * FROM respuestas;
```

respuesta_id	respuesta	u_id	p_id
--------------	-----------	------	------

(0 filas)

5. Agrega datos, 5 usuarios y 5 preguntas, la primera pregunta debe estar contestada dos veces correctamente por distintos usuarios, la pregunta 2 debe estar contestada correctamente sólo por un usuario, y las otras 2 respuestas deben estar incorrectas. (1 punto)

a. Contestada correctamente significa que la respuesta indicada en la tabla respuestas es exactamente igual al texto indicado en la tabla de preguntas.

pregunta_id	pregunta	respuesta_correcta
1	¿cuanto es 2+2=	4
2	¿cual es el mes actual	Mayo
3	¿país en el que estas?	Chile
4	¿color del cielo?	azul
5	¿que día es hoy?	miercoles

(5 filas)

usuario_id	nombre	edad	respuesta_id	respuesta	u_id	p_id
1	Nathalie	35	1	4	1	1
2	Felipe	18	2	Mayo	2	2
3	Franco	20	3	4	3	1
4	Pabla	35	4	azul	5	4
5	Elena	60	5	jueves	4	5
			6	argentina	4	3

(5 filas) (6 filas)

u_id	respuesta_id	p_id
3	3	1
1	1	1
2	2	2
5	4	4

(4 filas)

6. Cuenta la cantidad de respuestas correctas totales por usuario (independiente de la pregunta). (1 punto)

```

--sql
u_id | respuesta_id | p_id
-----+-----
    3 |              |    1
    1 |              |    1
    2 |              |    2
    5 |              |    4
(4 filas)

```

Cada usuario respondió 1 pregunta correcta (usuario=u_id).

```

usuarios
-----
      4
(1 fila)

```

en total fueron 4 usuarios que respondieron 1 pregunta correcta.

7. Por cada pregunta, en la tabla preguntas, cuenta cuántos usuarios tuvieron la respuesta correcta. (1 punto)

```

--sql
respuesta_id | p_id
-----+-----
          3 |    1
          1 |    1
          2 |    2
          4 |    4
(4 filas)

```

Para la pregunta 1, 2 usuarios respondieron bien, para la pregunta 2 fue solo un usuario y 4 también un usuario.

Todas las demás respuestas estuvieron malas.

```

--sql
pregunta | u_id | p_id
-----+-----
¿cuanto es 2+2= |    3 |    1
¿cuanto es 2+2= |    1 |    1
¿cual es el mes actual |    2 |    2
¿color del cielo? |    5 |    4
(4 filas)

```

8. Implementa borrado en cascada de las respuestas al borrar un usuario y borrar el primer usuario para probar la implementación. (1 punto)

Truncate table usuarios cascade; --pero esta elimina toda la tabla y las tablas relacionadas con un FK

DELETE * from usuarios where usuario_id=1; no permite ya que esta restringido por la FK en la tabla respuesta.

9. Crea una restricción que impida insertar usuarios menores de 18 años en la base de datos. (1 punto)

En la columna edad, al crear la table se insertó la restricción de la siguiente forma "edad INT CHECK (edad >= 18);"

10. Altera la tabla existente de usuarios agregando el campo email con la restricción de único. (1 punto)

usuario_id	nombre	edad	email
1	Nathalie	35	
2	Felipe	18	
3	Franco	20	
4	Pabla	35	
5	Elena	60	

(5 filas)