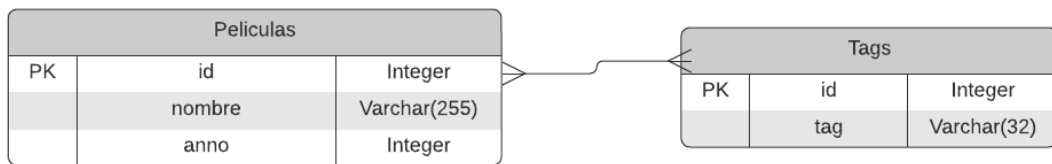


Solución Prueba SQL

Requerimientos

Dado el siguiente modelo:



1. Revisa el tipo de relación y crea el modelo correspondiente. Respeta las claves primarias, foráneas y tipos de datos. (1 punto)

Relación N:N, se debe crear tabla intermedia

```
CREATE TABLE peliculas (  
  id INTEGER,  
  nombre VARCHAR(255),  
  anno INTEGER,  
  PRIMARY KEY (id)  
);
```

```
CREATE TABLE tags (  
  id INTEGER,  
  tag VARCHAR(32),  
  PRIMARY KEY (id)  
);
```

```
CREATE TABLE pelicula_tag (  
  pelicula_id INT REFERENCES peliculas (id),  
  tag_id INT REFERENCES tags(id)  
);
```

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

```
INSERT INTO peliculas (id, nombre, anno) VALUES (1, 'Batman Begins', 2005);
INSERT INTO peliculas (id, nombre, anno) VALUES (2, 'Ghost', 1990);
INSERT INTO peliculas (id, nombre, anno) VALUES (3, 'Titanic', 1997);
INSERT INTO peliculas (id, nombre, anno) VALUES (4, 'Shrek', 2001);
INSERT INTO peliculas (id, nombre, anno) VALUES (5, 'The Revenant', 2015);
```

id	nombre	anno
1	Batman Begins	2005
2	Ghost	1990
3	Titanic	1997
4	Shrek	2001
5	The Revenant	2015

(5 filas)

```
INSERT INTO tags (id, tag) VALUES (1, 'Aventura');
INSERT INTO tags (id, tag) VALUES (2, 'Drama');
INSERT INTO tags (id, tag) VALUES (3, 'Fantasia');
INSERT INTO tags (id, tag) VALUES (4, 'Acción');
INSERT INTO tags (id, tag) VALUES (5, 'Romance');
```

id	tag
1	Aventura
2	Drama
3	Fantasia
4	Acción
5	Romance

(5 filas)

```
INSERT INTO pelicula_tag (pelicula_id, tag_id) VALUES (1, 1);
INSERT INTO pelicula_tag (pelicula_id, tag_id) VALUES (1, 2);
INSERT INTO pelicula_tag (pelicula_id, tag_id) VALUES (1, 4);
INSERT INTO pelicula_tag (pelicula_id, tag_id) VALUES (2, 2);
INSERT INTO pelicula_tag (pelicula_id, tag_id) VALUES (2, 5);
```

pelicula_id	tag_id
1	1
1	2
1	4
2	2
2	5

(5 filas)

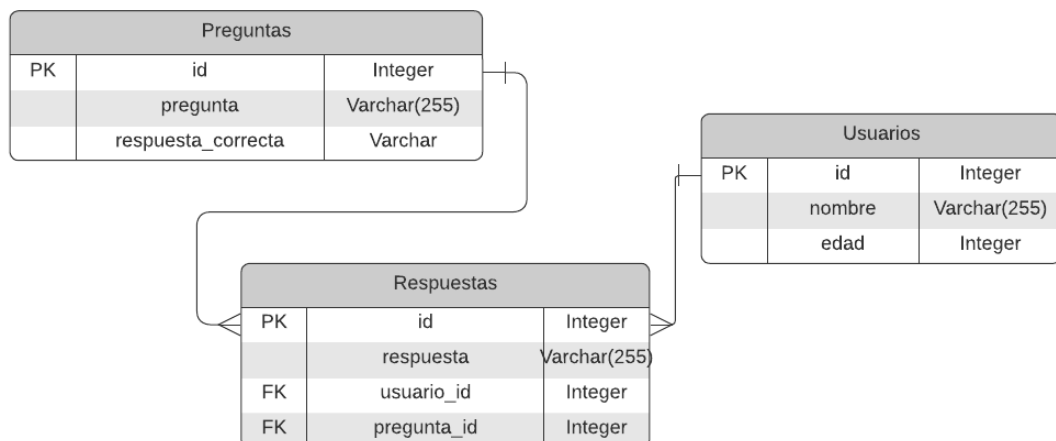
3. Cuenta la cantidad de tags que tiene cada película. Si una película no tiene tags debe mostrar 0. (1 punto)

```
SELECT peliculas.nombre, COUNT(pelicula_tag.tag_id)
FROM peliculas
LEFT JOIN pelicula_tag ON peliculas.id = pelicula_tag.pelicula_id
GROUP BY peliculas.nombre;
```

nombre	count
Titanic	0
The Revenant	0
Shrek	0
Ghost	2
Batman Begins	3

(5 filas)

Dado el siguiente modelo:



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

```
CREATE TABLE preguntas (  
  id INTEGER,  
  pregunta VARCHAR(255),  
  respuesta_correcta VARCHAR,  
  PRIMARY KEY (id)  
);
```

```
CREATE TABLE usuarios(  
  id INTEGER,  
  nombre VARCHAR(255),  
  edad INTEGER,  
  PRIMARY KEY (id)  
);
```

```
CREATE TABLE respuestas (  
  id INTEGER,  
  respuesta VARCHAR(255),  
  usuario_id INT REFERENCES usuarios (id),  
  pregunta_id INT REFERENCES preguntas (id),  
  PRIMARY KEY (id)  
);
```

5. Agrega 5 usuarios y 5 preguntas.
- La primera pregunta debe estar respondida correctamente dos veces, por dos usuarios diferentes.
 - La segunda pregunta debe estar contestada correctamente solo por un usuario.
 - Las otras dos preguntas deben tener respuestas incorrectas

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

```
INSERT INTO preguntas (id, pregunta, respuesta_correcta) VALUES (1,  
'Cuántas patas tiene un gato?', 'Tiene 4 patas');  
INSERT INTO preguntas (id, pregunta, respuesta_correcta) VALUES (2,  
'Como hacen los perros?', 'Guau');  
INSERT INTO preguntas (id, pregunta, respuesta_correcta) VALUES (3, 'De
```

```
que color es el pasto?', 'Verde');  
INSERT INTO preguntas (id, pregunta, respuesta_correcta) VALUES (4,  
'Cual es la capital de Chile?', 'Santiago');  
INSERT INTO preguntas (id, pregunta, respuesta_correcta) VALUES (5, 'De  
que pais son The Beatles?', 'Inglaterra');
```

id	pregunta	respuesta_correcta
1	Cuantas patas tiene un gato?	Tiene 4 patas
2	Como hacen los perros?	Guau
3	De que color es el pasto?	Verde
4	Cual es la capital de Chile?	Santiago
5	De que pais son The Beatles?	Inglaterra

(5 filas)

```
INSERT INTO usuarios (id, nombre, edad) VALUES (1, 'Juan Perez', '25');  
INSERT INTO usuarios (id, nombre, edad) VALUES (2, 'Maria Joaquina',  
'30');  
INSERT INTO usuarios (id, nombre, edad) VALUES (3, 'Pedro Diaz', '60');  
INSERT INTO usuarios (id, nombre, edad) VALUES (4, 'Camila Gonzalez',  
'35');  
INSERT INTO usuarios (id, nombre, edad) VALUES (5, 'Rocio Sanchez',  
'15');
```

id	nombre	edad
1	Juan Perez	25
2	Maria Joaquina	30
3	Pedro Diaz	60
4	Camila Gonzalez	35
5	Rocio Sanchez	15

(5 filas)

```
INSERT INTO respuestas (id, respuesta, usuario_id, pregunta_id) VALUES  
(1, 'Tiene 4 patas', 1, 1);  
INSERT INTO respuestas (id, respuesta, usuario_id, pregunta_id) VALUES  
(2, 'Tiene 4 patas', 2, 1);  
INSERT INTO respuestas (id, respuesta, usuario_id, pregunta_id) VALUES  
(3, 'Guau', 3, 2);
```

```
INSERT INTO respuestas (id, respuesta, usuario_id, pregunta_id) VALUES  
(4, 'Azul', 4,3);  
INSERT INTO respuestas (id, respuesta, usuario_id, pregunta_id) VALUES  
(5, 'Lota', 5, 4);
```

id	respuesta	usuario_id	pregunta_id
1	Tiene 4 patas	1	1
2	Tiene 4 patas	2	1
3	Guau	3	2
4	Azul	4	3
5	Lota	5	4

(5 filas)

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

```
SELECT usuarios.nombre, (  
    SELECT COUNT(*)  
    FROM respuestas  
    JOIN preguntas ON respuestas.pregunta_id = preguntas.id  
    WHERE respuestas.usuario_id = usuarios.id  
    AND respuestas.respuesta = preguntas.respuesta_correcta  
) AS respuestas_correctas  
FROM usuarios;
```

nombre	count
Pedro Diaz	1
Maria Joaquina	1
Rocio Sanchez	0
Juan Perez	1
Camila Gonzalez	0

7. Por cada pregunta, en la tabla preguntas, cuenta cuántos usuarios respondieron correctamente. (1 punto)

```
SELECT preguntas.pregunta, COUNT(respuestas.usuario_id)  
AS usuarios_con_respuesta_correcta FROM preguntas  
LEFT JOIN respuestas ON respuestas.pregunta_id = preguntas.id
```

```
AND respuestas.respuesta = preguntas.respuesta_correcta
GROUP BY preguntas.pregunta;
```

pregunta	count
De que color es el pasto?	0
Como hacen los perros?	1
Cual es la capital de Chile?	0
De que pais son The Beatles?	0
Cuantas patas tiene un gato?	2

(5 filas)

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

Alumno busca nombre de clave con `[\d nombre_tabla]`, en este caso `[\d respuestas]`

```
ALTER TABLE respuestas DROP CONSTRAINT respuestas_usuario_id_fkey, ADD
FOREIGN KEY (usuario_id) REFERENCES usuarios(id) ON DELETE CASCADE;
```

```
DELETE FROM usuarios WHERE id = 1;
```

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

```
ALTER TABLE usuarios ADD CONSTRAINT edad CHECK (edad > 18);
```

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

```
ALTER TABLE usuarios ADD COLUMN email VARCHAR;
ALTER TABLE usuarios ADD UNIQUE (email);
```