SQL 3 - MOVIES DB

// Práctica Grupal

1. Explique el concepto de normalización y para que se utiliza.

Normalizar sería aplicar una serie de normas en las relaciones actuales para poder minimizar la redundancia de datos y a futuro evitar el uso de consultas innecesariamente complejas.

2. Agregue una película a la tabla movies.

```
INSERT INTO `movies_db`.`movies` (`id`, `title`, `rating`, `awards`, `release_date`, `length`) VALUES ('22', 'Una peli', '6.0', '3', '2011-06-04', '100');
```

3. Agregué un género a la tabla genres.

```
INSERT INTO `movies_db`.`genres` (`id`, `created_at`, `name`, `ranking`, `active`) VALUES ('13', '2000-06-03', 'Mistico', '13', '1');
```

4. Asocie a la película del Ej 2. con el género creado en el Ej. 3.

```
UPDATE `movies_db`.`movies` SET `genre_id` = '13' WHERE (`id` = '22');
```

5. Modifique la tabla actors para que al menos un actor tenga como favorita la película agregada en el Ej.2.

```
UPDATE `movies_db`.`actors` SET `favorite_movie_id` = '22' WHERE (`id` = '3');
```

6. Cree una tabla temporal copia de la tabla movies.

```
use movies_db;
CREATE TEMPORARY TABLE movies_copy SELECT * FROM movies;
```

Si quiero crear los propios campos sin copiar de otra tabla:

```
use movies_db;
CREATE TEMPORARY TABLE movies_copy
(`id` int unsigned NOT NULL AUTO_INCREMENT,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL,
  `title` varchar(500) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT
NULL,
  `rating` decimal(3,1) unsigned NOT NULL,
  `awards` int unsigned NOT NULL DEFAULT '0',
  `release_date` datetime NOT NULL,
  `length` int unsigned DEFAULT NULL,
  `genre_id` int unsigned DEFAULT NULL,
 PRIMARY KEY (`id`),
 KEY `movies_genre_id_foreign` (`genre_id`));
```

Para ver las tablas temporales

```
SELECT *
FROM information_schema.INNODB_TEMP_TABLE_INFO;
```

7. Elimine de esa tabla temporal todas las películas que hayan ganado menos de 5 awards.

```
use movies_db;
DELETE FROM `movies_db`.`movies_copy` AS movies
WHERE movies.awards < 5;
```

De forma manual

```
DELETE FROM `movies_db`.`movies_copy` WHERE (`id` = '19');
DELETE FROM `movies_db`.`movies_copy` WHERE (`id` = '20');
DELETE FROM `movies_db`.`movies_copy` WHERE (`id` = '21');
DELETE FROM `movies_db`.`movies_copy` WHERE (`id` = '22');
DELETE FROM `movies_db`.`movies_copy` WHERE (`id` = '16');
DELETE FROM `movies_db`.`movies_copy` WHERE (`id` = '17');
DELETE FROM `movies_db`.`movies_copy` WHERE (`id` = '6');
DELETE FROM `movies_db`.`movies_copy` WHERE (`id` = '7');
DELETE FROM `movies_db`.`movies_copy` WHERE (`id` = '8');
DELETE FROM `movies_db`.`movies_copy` WHERE (`id` = '9');
DELETE FROM `movies_db`.`movies_copy` WHERE (`id` = '10');
DELETE FROM `movies_db`.`movies_copy` WHERE (`id` = '11');
DELETE FROM `movies_db`.`movies_copy` WHERE (`id` = '12');
DELETE FROM `movies_db`.`movies_copy` WHERE (`id` = '13');
DELETE FROM `movies_db`.`movies_copy` WHERE (`id` = '14');
DELETE FROM `movies_db`.`movies_copy` WHERE (`id` = '1');
```

8. Obtenga la lista de todas los géneros que tengan al menos una película.

```
SELECT DISTINCT genres.*
FROM movies_db.genres AS genres
JOIN movies_db.movies AS movies
ON movies.genre_id=genres.id;
```

Si lo hacemos con movies_copy, faltarán géneros ya que hemos borrado varios registros.

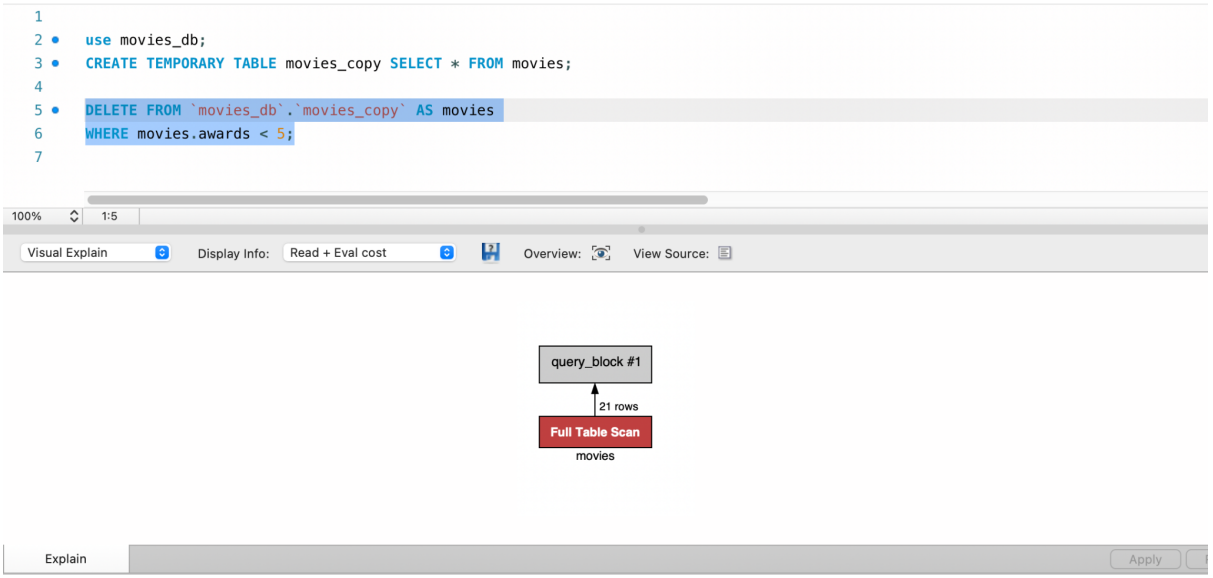
9. Obtenga la lista de actores cuya película favorita haya ganado más

de 3 awards.

```
SELECT DISTINCT actors.*
FROM movies_db.actors AS actors
WHERE actors.favorite_movie_id IN
(SELECT movies.id
FROM movies_db.movies AS movies
WHERE movies.awards>3);
```

10. Utilice el explain plan para analizar las consultas del Ej.6 y 7.

Del create temporary no puede generar info. Del delete si.



11. Qué son los índices? Para qué sirven?

Una estructura de datos asociada a una tabla que agiliza el acceso a la misma.

12. Cree un índice sobre el nombre en la tabla movies.

Se crea un índice sobre title.

Agregar Índice1:

```
CREATE [UNIQUE] INDEX indice_movies
ON movies_db.movies(title);
```

Agregar Índice 2:

ALTER TABLE movies_db.movies

ADD INDEX indice_movies(title)

Borrar Índice:

DROP INDEX idx_movies_title

ON movies_db.movies;

13. Chequee que el indice fue creado correctamente.

SHOW INDEX FROM movies_db.movies;

HELP INDEX movies_db.movies;