

Kathmandu University
Department of Computer Science and Engineering
Dhulikhel, Kavre



A Report on 'Lab Work 3' [COMP 232]

Submitted by:

Chandan Kumar Mahato (31)

II-year, II semester

Submitted to:

Dr. Rajani Chulyadyo

Department of Computer Science and Engineering

Submission Date: Feb 17, 2022

Qlab3

1. Download the given database dump from the following link
<https://drive.google.com/file/d/1AbZE3cvdNUXcSjli-LtX7DLbfFoxUgET/view?usp=sharing> (Links to an external site.)
2. Create a database named imdb in MySQL and import the downloaded database dump into it.
3. Update the tables to add primary and foreign key constraints.
4. Remove film_count from actors table.
5. Write SQL queries for the following:
 1. Find the number of actors, movies, genres, and directors.
 2. List the full name, and role of all actors who played in Titanic.
 3. Find the number of movies in all genres.
 4. Find the average number of movies played by the actors.
 5. Find the average number of actors in a movie.
 6. Find top 5 movies based on the rank.
 7. Find 20 directors who have directed at least 2 movies.
 8. Find all movies played by Kevin Bacon.
 9. Find all movies released from 1990 to 2000.

Solution:

Queries in relational algebra and SQL (structured query language) for the following questions:

3. Update the tables to add primary and foreign key constraints.

```
ALTER TABLE actors ADD CONSTRAINT PRIMARY KEY (id);
ALTER TABLE directors ADD CONSTRAINT PRIMARY KEY (id);
ALTER TABLE movies ADD CONSTRAINT PRIMARY KEY (id);

ALTER TABLE directors_genres ADD FOREIGN KEY(director_id) REFERENCES
directors(id);
ALTER TABLE movies_directors ADD FOREIGN KEY(director_id) REFERENCES movies(id);
ALTER TABLE movies_genres ADD FOREIGN KEY(movie_id) REFERENCES movies(id);
ALTER TABLE roles ADD FOREIGN KEY(actor_id) REFERENCES actors(id);
ALTER TABLE roles ADD FOREIGN KEY(movie_id) REFERENCES movies(id);
```

4. Remove film_count from actors table.

```
ALTER TABLE actors
DROP COLUMN film_count;
```

5. Write SQL queries for the following:

1. Find the number of actors, movies, genres, and directors.

```
SELECT
  (SELECT count(*) FROM movies) AS movies_count,
  (SELECT count(*) FROM actors) AS actors_count,
  (SELECT count(DISTINCT genre) FROM directors_genres dg ) AS
directors_genre_count,
  (SELECT count(DISTINCT genre) FROM movies_genres mg ) AS movies_genre_count,
  (SELECT count(*) FROM directors) AS directors_count;
```

2. List the full name, and role of all actors who played in Titanic.

```
SELECT
  movies.name AS movies_name,
  (CONCAT(actors.first_name , ' ', actors.last_name)) AS actors_name,
  roles.role AS actors_role
FROM
  roles
INNER JOIN actors
ON actors.id = roles.actor_id
INNER JOIN movies
ON movies.id = roles.movie_id
WHERE movies.name = 'Titanic';
```

3. Find the number of movies in all genres.

```
SELECT
  movies_genres.genre AS movies_genre,
  count(movies.name) AS movies_on_that_genre
FROM
  movies
INNER JOIN
  movies_genres
ON movies_genres.movie_id=movies.id
GROUP BY movies_genres.genre;
```

4. Find the average number of movies played by the actors.

```
SELECT (AVG(total.number_of_movies_played)) as avg_no_of_movies_by_the_actor
FROM (
    SELECT
        roles.actor_id,
        count(roles.movie_id) AS number_of_movies_played
    FROM
        roles
    GROUP BY
        roles.actor_id
) as total;
```

5. Find the average number of actors in a movie.

```
SELECT AVG(total.number_of_actors) AS avg_no_of_actor_in_a_movie
FROM (
    SELECT
        roles.movie_id,
        count(roles.actor_id) AS number_of_actors
    FROM
        roles
    GROUP BY roles.movie_id
) AS total;
```

6. Find top 5 movies based on the rank.

```
SELECT * FROM movies ORDER BY movies.rank DESC LIMIT 5;
```

7. Find 20 directors who have directed at least 2 movies.

```
SELECT
    CONCAT(directors.first_name, ' ', directors.last_name) as director_name,
    count(movies_directors.movie_id) AS number_of_movies_directed
FROM
    movies_directors
INNER JOIN
    directors
ON directors.id=movies_directors.director_id
GROUP BY movies_directors.director_id
ORDER BY number_of_movies_directed
DESC limit 20;
```

8. Find all the movies played by Kevin Bacon.

```
SELECT movies.name as movies_played_by_Kevin_Bacon, roles.role
FROM
roles
INNER JOIN
actors
ON roles.actor_id = actors.id
INNER JOIN
movies
ON roles.movie_id = movies.id
WHERE CONCAT(actors.first_name, ' ', actors.last_name) = 'Kevin Bacon';
```

9. Find all the movies released from 1990 to 2000

```
SELECT * FROM movies WHERE movies.year >= 1990 and movies.year < 2000;
```