ASSIGNMENT 4 - VIEWS

1. Average rating of comedy genre movies.

```
CREATE VIEW `avg_rating_comedy` AS
SELECT AVG(m.rating) AS Average_Rating FROM Movie m
JOIN Movie_Genre mg ON m.movie_id=mg.movie_id
JOIN Genre g ON g.genre_id = mg.genre_id and g.genre_id=5;
```

2. The top 10 hit movies by Ajay Devgn.

```
CREATE VIEW `top_hit_Ajay` AS
SELECT m.name, m.rating, m.movie_id from movie m
INNER JOIN movie_stars ma on ma.movie_id = m.movie_id and
ma.star_id=(SELECT s.star_id from stars s where s.name='Ajay Devgn')
ORDER BY m.rating DESC
Limit 10;
```

3. The total number of theaters in Ahmedabad

```
CREATE VIEW `total_theaters_ahmedabad` AS
SELECT COUNT(theater_id) AS NoOfTheaters FROM Theaters t where
t.city_id=(Select c.city_id from city c where c.name='Ahmedabad') ;
```

4. Name the movies with minimum runtime.

```
CREATE VIEW `min_runtime` AS
SELECT N.NAME FROM MOVIE N WHERE N.RUNTIME=(SELECT
MIN(M.RUNTIME) FROM MOVIE M)
```

5. List the movie with the least number of tweets between 2022-11-1 between 2022-11-12.

```
CREATE VIEW `least_tweets_movie`
AS SELECT m.name, COUNT(tm.movie_id) AS count
FROM movie m
```

```
INNER JOIN
        twitter_tweet_movie tm ON tm.movie_id = m.movie_id
INNER JOIN
        twitter_tweets t ON t.tweet_id = tm.tweet_id
GROUP BY m.movie_id
ORDER BY count
LIMIT 1;
```

6. List actors of movies with highest positive reviews fetched from tweets between 2022-11-1 between 2022-11-12.

```
CREATE VIEW `highest_positive_tweet_movie` AS
SELECT s.name FROM stars s
INNER JOIN
movie_stars ms ON s.star_id = ms.star_id
AND ms.movie_id = (SELECT m.movie_id FROM movie m
INNER JOIN
twitter_tweet_movie tm ON tm.movie_id = m.movie_id
INNER JOIN
twitter_tweets t ON t.tweet_id = tm.tweet_id
GROUP BY m.name
ORDER BY SUM(t.sentiment)
LIMIT 1);
```

7. List the user with the maximum number of bookings.

```
CREATE VIEW `maximum_bookings` AS
SELECT u.name, b.user_id, SUM(b.tickets_booked) * s.ticket_price AS total
FROM bookings b
INNER JOIN screen_shows s ON b.screen_shows_id = s.id
INNER JOIN user u ON b.user_id = u.user_id
GROUP BY b.user_id
ORDER BY total
```

LIMIT 1;

8. List down top 3 theaters with least sales.

```
CREATE VIEW `least_sales_theaters` AS
SELECT t.name, sum(tt.amount) as total from theaters t
INNER JOIN screens s on t.theater_id = s.screen_id
INNER JOIN screen_shows ss on ss.screen_id = s.screen_id
INNER JOIN bookings b on ss.id = b.screen_shows_id
INNER JOIN transaction tt on tt.booking_id = b.booking_id
GROUP BY t.theater_id
order by total asc
limit 3;
```

9. Which was the most hit movie by “Priyanka Chopra Jonas” in 2013?

```
CREATE VIEW `most_hit_by_priyanka` AS
SELECT m.name, m.released_year, m.rating FROM stars s INNER JOIN
movie_stars ms on ms.star_id = s.star_id INNER JOIN movie m on ms.movie_id =
m.movie_id where s.name like "%Priyanka Chopra Jonas%" and m.released_year
= 2013 order by m.rating desc limit 1;
```

10. On which date most tickets were booked in 2020?

```
CREATE VIEW `most_booked_date` AS
SELECT b.created_at, b.tickets_booked
FROM bookings b
WHERE YEAR(b.created_at) = 2020
GROUP BY b.created_at
ORDER BY b.tickets_booked DESC
LIMIT 1
```

11. Top 5 actors with maximum number of movie releases in the year 2016?

```
CREATE VIEW `_max_movie_releases` AS
SELECT count(stars.star_id) as top_actors ,stars.name, movie.released_year
FROM stars
INNER JOIN movie_stars ON movie_stars.star_id = stars.star_id
INNER JOIN movie ON movie.movie_id = movie_stars.movie_id and
movie.released_year = 2016
GROUP BY stars.star_id
ORDER BY top_actors DESC
LIMIT 5;
```

12. Total number of screens in theaters in the city of Mumbai?

```
CREATE VIEW `total_screens_mumbai` AS
SELECT c.name, COUNT(s.screen_id) FROM
theaters AS t
INNER JOIN city AS c ON t.city_id = c.city_id
INNER JOIN screens AS s ON s.theater_id = t.theater_id
AND c.name = 'Mumbai';
```

13. List the actors whose movies were released between 2018 and 2021.

```
CREATE VIEW `movie_actors` AS
SELECT s.name as actor, m.name, m.released_year
FROM movie AS m INNER JOIN movie_stars AS ms ON m.movie_id =
ms.movie_id AND m.released_year BETWEEN 2018 AND 2021 INNER JOIN
stars AS s ON s.star_id = ms.star_id;
```

14. List the actors of movies with highest number of retweets between "2022-11-01" and "2022-11-12"

```
CREATE VIEW `highest_retweets_movie` AS
SELECT s.name
FROM stars AS s INNER JOIN movie_stars AS ms ON s.star_id = ms.star_id
AND ms.movie_id = (SELECT m.movie_id
FROM movie AS m INNER JOIN twitter_tweet_movie AS tm ON tm.movie_id =
m.movie_id
INNER JOIN twitter_tweets AS t ON tm.tweet_id = t.tweet_id
```

```
ORDER BY t.retweet DESC
LIMIT 1);
```

15. List the movie with most negative reviews fetched from tweets between 2022-11-1 between 2022-11-12.

```
CREATE VIEW `most_negative_reviews` AS
SELECT m.name, SUM(t.sentiment) AS sentiment
FROM movie AS m
INNER JOIN twitter_tweet_movie AS tm ON tm.movie_id = m.movie_id
INNER JOIN twitter_tweets AS t ON tm.tweet_id = t.tweet_id
GROUP BY m.name
ORDER BY SUM(sentiment) ASC
LIMIT 1;
```

16. List movies with most and distinct tweet mentions fetched from tweets

```
CREATE VIEW `most_mentions` AS
SELECT m.name, COUNT(DISTINCT tt.target_user) as all_mentions FROM
movie AS m
INNER JOIN twitter_tweet_movie AS tm ON tm.movie_id = m.movie_id
INNER JOIN twitter_tweets AS t ON tm.tweet_id = t.tweet_id
INNER JOIN tweet_mentions AS tt ON t.tweet_id = tt.tweet_id
GROUP BY m.name
ORDER BY all_mentions DESC
LIMIT 1;
```

17. List the movie with the maximum number of bookings.

```
CREATE VIEW `movie_max_bookings` AS
SELECT m.name, m.released_year, SUM(b.tickets_booked) * s.ticket_price AS
total FROM bookings b
INNER JOIN screen_shows s ON b.screen_shows_id = s.id
INNER JOIN movie m ON s.movie_id = m.movie_id
GROUP BY m.movie_id
ORDER BY total DESC
```

```
LIMIT 1;
```

18. Which are the top 3 movies with the maximum number of screens?

```
CREATE VIEW `movie_max_screens` AS
SELECT m.name, m.released_year, (s.screen_id) AS totalScreens
FROM screen_shows s
INNER JOIN movie m ON m.movie_id = s.movie_id
GROUP BY m.movie_id
ORDER BY totalScreens DESC
LIMIT 3
```

19. What are the top 5 movies with good ratings in the drama genre in 2020?

```
CREATE VIEW `drama_good_ratings_movies` AS
SELECT m.name, m.released_year, m.rating, g.genre
FROM movie m
INNER JOIN movie_genre mg ON mg.movie_id = m.movie_id
INNER JOIN genre g ON g.genre_id = mg.genre_id
AND g.genre = 'Drama'
AND m.released_year = 2020
ORDER BY m.rating DESC
LIMIT 5;
```

20. the total income of theater

```
CREATE VIEW `total_income` AS
SELECT SUM(tt.amount) AS totalSales, t.name
FROM transaction tt
INNER JOIN bookings b ON tt.booking_id = b.booking_id
INNER JOIN screen_shows ss ON b.screen_shows_id = ss.id
INNER JOIN screens s ON s.screen_id = ss.screen_id
INNER JOIN theaters t ON t.theater_id = s.theater_id
GROUP BY t.theater_id
```

```
ORDER BY totalSales DESC
LIMIT 1;
```

21. Best rated movie between 2015 to 2016.

```
CREATE VIEW `best_movie` AS
SELECT m.name, max(m.rating)
FROM movie m
WHERE m.released_year BETWEEN 2015 AND 2016
GROUP BY m.name
ORDER BY name ASC;
```

22. Which theater had the highest sales?

```
CREATE VIEW `theater_highest_sales` AS
SELECT t.name, MAX(s.ticket_price*b.tickets_booked)
FROM theaters t
INNER JOIN screens sc ON t.theater_id = sc.theater_id
INNER JOIN screen_shows s ON sc.screen_id = s.screen_id
INNER JOIN bookings b ON b.screen_shows_id = s.id;
```

23. Year with least number of movies

```
CREATE VIEW `year_least_no_movies` AS
SELECT m.released_year, COUNT(m.movie_id) as movie_count
FROM movie m
GROUP BY m.released_year
ORDER BY movie_count ASC
LIMIT 1;
```

24. List movies with most tweet tags fetched from tweets between 2022-11-1 between 2022-11-12.

```
CREATE VIEW `most_tweet_tags_movie` AS
SELECT m.name, COUNT(tt.tag) FROM movie AS m
INNER JOIN twitter_tweet_movie AS tm ON tm.movie_id = m.movie_id
INNER JOIN twitter_tweets AS t ON tm.tweet_id = t.tweet_id
```

```
INNER JOIN tweet_tags tt ON t.tweet_id = tt.tweet_id
GROUP BY m.name
ORDER BY COUNT(tt.tag) DESC
LIMIT 1;
```

25. List movies with most tweets fetched from tweets between 2022-11-1 between 2022-11-12.

```
CREATE VIEW `most_tweets` AS
SELECT m.name, COUNT(t.tweet_id)AS tweet_count
FROM movie  m
INNER JOIN twitter_tweet_movie AS tm ON tm.movie_id = m.movie_id
INNER JOIN twitter_tweets AS t ON tm.tweet_id = t.tweet_id
GROUP BY tm.movie_id
ORDER BY tweet_count DESC
LIMIT 1
```

26. List of all the movies and their ratings for the actor - Shah Rukh Khan?

```
CREATE VIEW `srk` AS
SELECT m.name AS SRK, m.rating
FROM movie_stars ms
INNER JOIN stars s ON ms.star_id = s.star_id
INNER JOIN movie m ON m.movie_id = ms.movie_id
WHERE s.name = "Shah Rukh Khan";
```

27. Movies of which genre have the highest rating

```
CREATE VIEW `highest_rating_genre` AS
SELECT g.genre, g.genre_id, MAX(m.rating)
FROM movie_genre mg
INNER JOIN genre g ON mg.genre_id = g.genre_id
INNER JOIN movie m ON m.movie_id = mg.genre_id
```

```
GROUP BY g.genre, g.genre_id
ORDER BY rating DESC
LIMIT 1
```

28. What are the different ticket prices across all screens and how many screen ids come under each ticket price?

```
CREATE VIEW `diff_ticket_prices` AS
SELECT COUNT(ss.screen_id), MAX(ss.ticket_price)
FROM screen_shows ss
GROUP BY ss.ticket_price
ORDER BY ss.ticket_price DESC
```

29. How many screens are there in every city?

```
CREATE VIEW `screens_city` AS
SELECT  c.name, COUNT(s.screen_id)
FROM city c
INNER JOIN theaters t ON t.city_id = c.city_id
INNER JOIN screens s ON s.theater_id = t.theater_id
GROUP BY c.name
ORDER BY COUNT(s.screen_id) DESC;
```

30. In which theaters were all the movies released in the year 2020?

```
CREATE VIEW `theater` AS
SELECT t.name
FROM theaters t
INNER JOIN screens s ON t.theater_id = s.theater_id
INNER JOIN screen_shows ss ON ss.screen_id = s.screen_id
INNER JOIN movie m ON m.movie_id = ss.movie_id
WHERE m.released_year = "2020" GROUP BY t.name
```