# Common Twitter Queries

#### **Q1.** What user posted this tweet?

**Query**

SELECT

u.name, t.twitter\_text

FROM

twitter\_user AS u

INNER JOIN

twitter\_tweets AS t ON u.twitter\_handle = t.twitter\_handle

WHERE

t.tweet\_id = 1590790945641230337

**Relational Algebra**

π u.name, t.twitter\_text

σ t.tweet\_id = 1590790945641230337

(ρ u twitter\_user ⋈ u.twitter\_handle = t.twitter\_handle

ρ t twitter\_tweets)

#### **Q2.** When did the user post this tweet?

**Query**

SELECT

u.name, t.twitter\_text, t.created\_at

FROM

twitter\_user as u

INNER JOIN

twitter\_tweets as t ON u.twitter\_handle = t.twitter\_handle

WHERE

t.tweet\_id = 1590790945641230337

**Relational Algebra**

π u.name, t.twitter\_text, t.created\_at

σ t.tweet\_id = 1590790945641230337

(ρ u twitter\_user ⋈ u.twitter\_handle = t.twitter\_handle

ρ t twitter\_tweets)

#### **Q3.** What tweets have this user posted in the past 24 hours?

**Query**

SELECT

u.name, t.twitter\_text, t.created\_at

FROM

twitter\_user AS u

INNER JOIN

twitter\_tweets AS t ON u.twitter\_handle = t.twitter\_handle

WHERE

u.twitter\_handle = 'gaugetherange' and t.created\_at > "2022-11-11 19:00:00.0000" and t.created\_at < "2022-11-12 18:59:59.0000"

**Relational Algebra**

π u . name, t . twitter\_text, t . created\_at

σ u . twitter\_handle = "gaugetherange" AND t . created\_at > "2022-11-11 19:00:00.0000" AND t . created\_at < "2022-11-12 18:59:59.0000"

(ρ u twitter\_user ⋈ u . twitter\_handle = t . twitter\_handle

ρ t twitter\_tweets)

#### **Q4.** How many tweets have this user posted in the past 24 hours?

**Query**

SELECT

u.name, count(t.twitter\_text), t.created\_at

FROM

twitter\_user AS u

INNER JOIN

twitter\_tweets AS t ON u.twitter\_handle = t.twitter\_handle

WHERE

u.twitter\_handle = 'gaugetherange' and t.created\_at > "2022-11-11 19:00:00.0000" and t.created\_at < "2022-11-12 18:59:59.0000"

**Relational Algebra**

π u . name, COUNT (twitter\_text), t . created\_at

γ COUNT (twitter\_text)

σ u . twitter\_handle = "gaugetherange" AND t . created\_at > "2022-11-11 19:00:00.0000" AND t . created\_at < "2022-11-12 18:59:59.0000"

(ρ u twitter\_user ⋈ u . twitter\_handle = t . twitter\_handle

ρ t twitter\_tweets)

#### **Q5.** When did this user join Twitter?

**Query**

SELECT

u.twitter\_handle, u.created\_at

FROM

twitter\_user AS u

WHERE

u.twitter\_handle = 'gaugetherange'

**Relational Algebra**

π u . twitter\_handle, u . created\_at

σ u . twitter\_handle = "gaugetherange"

ρ u twitter\_user

#### **Q6.** What keywords/ hashtags are popular?

**Query**

SELECT

t.tag, COUNT(t.tag)

FROM

tweet\_tags AS t

GROUP BY t.tag

ORDER BY COUNT(t.tag) DESC

LIMIT 10

**Relational Algebra**

τ COUNT (tag) ↓

γ tag, COUNT (tag)

ρ t tweet\_tags

#### **Q7.** What tweets are popular?

**Query**

SELECT

t.twitter\_text, t.retweet

FROM

twitter\_tweets AS t

ORDER BY t.retweet DESC

LIMIT 10;

**Relational Algebra**

τ t . retweet ↓

π t . twitter\_text, t . retweet

ρ t twitter\_tweets

# Manashree’s Queries

**ASSIGNMENT - 2**

#### **Q1.** What is the average rating of comedy genre movies?

**Query**

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;

**Relational Algebra**

π AVG (rating) → average\_rating

γ AVG (rating)

(ρ m movie ⋈ m . movie\_id = mg . movie\_id

ρ mg movie\_genre ⋈ g . genre\_id = mg . genre\_id AND g . genre\_id = 5

ρ g genre)

#### **Q2.** What are the top 10 hit movies by Ajay Devgn?

**Query**

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;

**Relational Algebra**

τ m . rating ↓

π m . name, m . rating, m . movie\_id

(ρ m movie ⋈ ma . movie\_id = m . movie\_id

ρ ma movie\_stars ⋈ ma . star\_id = 112

ρ s stars)

#### **Q3.** What is the total number of theaters in Ahmedabad?

**Query**

SELECT COUNT(theater\_id) AS NoOfTheaters FROM Theaters t where t.city\_id=(Select c.city\_id from city c where c.name='Ahmedabad') ;

**Relational Algebra**

π COUNT (theater\_id) → nooftheaters

γ COUNT (theater\_id)

σ t . city\_id = π c . city\_id

σ c . name = "Ahmedabad"

ρ c city

ρ t theaters

#### 

#### **Q4.** Name the movies with minimum runtime.

**Query**

SELECT N.NAME FROM MOVIE N WHERE N.RUNTIME=(SELECT MIN(M.RUNTIME) FROM MOVIE M)

**Relational Algebra**

π n . name

σ n . runtime = π MIN (runtime)

γ MIN (runtime)

ρ m movie

ρ n movie

#### **Q5.** List the movie with the least number of tweets between 2022-11-1 between 2022-11-12.

**Query**

SELECT m.name, count(t.movie\_id) as count from movie m

INNER JOIN twitter\_tweets t on t.movie\_id=m.movie\_id

GROUP BY m.name

ORDER BY count

LIMIT 1;

**Relational Algebra**

τ COUNT (movie\_id)

γ name, COUNT (movie\_id)

(ρ m movie ⋈ t . movie\_id = m . movie\_id

ρ t twitter\_tweets)

#### 

#### **ASSIGNMENT - 3**

#### **Q6.** List actors of movies with highest positive reviews fetched from tweets between 2022-11-1 between 2022-11-12. **Query**

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\_tweets t on t.movie\_id=m.movie\_id

GROUP BY m.name

ORDER BY sum(t.sentiment)

limit 1);

#### **Q7.** List the user with the maximum number of bookings. **Query**

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;

#### **Q8.** List down top 3 theaters with least sales. **Query**

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;

#### **Q9.** Which was the most hit movie by “Priyanka Chopra Jonas” in 2013? **Query**

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;

#### **Q10.** On which date most tickets were booked in 2020? **Query**

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

# 

# Anshul’s Queries

**ASSIGNMENT - 2**

#### **Q1.** Top 5 actors with maximum number of movie releases in the year 2016?

**Query**

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;

**Relational Algebra**

τ top\_actors ↓

π COUNT (star\_id) → top\_actors, stars . name, movie . released\_year

γ star\_id, COUNT (star\_id) (stars ⋈ movie\_stars . star\_id = stars . star\_id movie\_stars ⋈ movie . movie\_id = movie\_stars . movie\_id AND movie . released\_year = 2016 movie)

#### 

#### **Q2.** Total number of screens in theaters in the city of Mumbai?

**Query**

SELECT c.name, SUM(t.screens) FROM theaters AS t INNER JOIN city AS c ON t.city\_id = c.city\_id AND c.name = 'Mumbai';

**Relational Algebra**

π c . name, SUM (screens)

γ SUM (screens)

(ρ t theaters ⋈ t . city\_id = c . city\_id AND c . name = "Mumbai"

ρ c city)

#### **Q3.** List the actors whose movies were released between 2018 and 2021.

**Query**

SELECT s.name, 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;

**Relational Algebra**

π s . name, m . name, m . released\_year

(ρ m movie ⋈ m . movie\_id = ms . movie\_id AND (2018 <= m . released\_year AND m . released\_year <= 2021)

ρ ms movie\_stars ⋈ s . star\_id = ms . star\_id

ρ s stars)

#### **Q4.** List the actors of movies with highest number of retweets between "2022-11-01" and "2022-11-12"

**Query**

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\_tweets AS t ON m.movie\_id = t.movie\_id

ORDER BY t.retweet DESC

LIMIT 1);

**Relational Algebra**

π s . name

(ρ s stars ⋈ s . star\_id = ms . star\_id AND ms . movie\_id = (τ t . retweet ↓

π m . movie\_id

σ "2022-11-01" <= t . created\_at AND t . created\_at <= "2022-11-12"

(ρ m movie ⋈ m . movie\_id = t . movie\_id

ρ t twitter\_tweets))

ρ ms movie\_stars)

#### **Q5.** List the movie with most negative reviews fetched from tweets between 2022-11-1 between 2022-11-12.

**Query**

SELECT m.name, SUM(t.sentiment) AS sentiment

FROM movie AS m INNER JOIN twitter\_tweets AS t ON t.movie\_id = m.movie\_id

GROUP BY m.name

ORDER BY SUM(sentiment) ASC

LIMIT 1;

**Relational Algebra**

τ SUM (sentiment)

π m . name, SUM (sentiment) → sentiment

γ name, SUM (sentiment)

(ρ m movie ⋈ t . movie\_id = m . movie\_id

ρ t twitter\_tweets)

**ASSIGNMENT - 3**

#### 

#### **Q6.** List movies with most and distinct tweet mentions fetched from tweets **Query**

SELECT m.name, COUNT(DISTINCT tt.target\_user) as all\_mentions FROM movie AS m

INNER JOIN twitter\_tweets AS t ON m.movie\_id = t.movie\_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;

#### **Q7.** List the movie with the maximum number of bookings. **Query**

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

#### **Q8.** Which are the top 3 movies with the maximum number of screens?

#### **Query**

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

#### **Q9.** What are the top 5 movies with good ratings in the drama genre in 2020? **Query**

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 1

#### **Q10.** What was the total income of a theater in a particular month from a specific movie? **Query**

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

# Soham’s Queries

**ASSIGNMENT - 2**

#### **Q1.** Best rated movie between 2015 to 2016.

**Query -**

SELECT name, max(rating)

FROM movie

WHERE released\_year BETWEEN 2015 AND 2016

ORDER BY name ASC;

**Relational Algebra**

τ name

π name, MAX (rating)

γ MAX (rating)

σ 2015 <= released\_year AND released\_year <= 2016 movie

#### **Q2.** Which theater had the highest sales?

**Query -**

SELECT t.name, MAX(s.ticket\_price\*s.ticket\_sold)

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;

**Relational Algebra**

π t . name, MAX (?column?)

γ MAX (?column?)

(ρ t theaters ⋈ t . theater\_id = sc . theater\_id

ρ sc screens ⋈ sc . screen\_id = s . screen\_id

ρ s screen\_shows)

#### **Q3.** Year with least number of movies

**Query -**

SELECT released\_year, COUNT(movie\_ID) as movie\_count

FROM movie

GROUP BY released\_year

ORDER BY movie\_count ASC

LIMIT 1;

**Relational Algebra**

τ movie\_count

π released\_year, COUNT (movie\_id) → movie\_count

γ released\_year, COUNT (movie\_id) movie

#### **Q4.** List movies with most tweet tags fetched from tweets between 2022-11-1 between 2022-11-12. **Query**

SELECT m.name, COUNT (tt.tag) FROM movie AS m INNER JOIN twitter\_tweets AS t ON m.movie\_id = t.movie\_id INNER JOIN tweet\_tags AS tt ON t.tweet\_id = tt.tweet\_id

WHERE t.created\_at BETWEEN '2022-11-1' AND '2022-11-12'

GROUP BY m.name

ORDER BY COUNT (tt.tag) DESC

LIMIT 1;

**Relational Algebra**

τ COUNT (tag) ↓

γ name, COUNT (tag)

σ "2022-11-1" <= t . created\_at AND t . created\_at <= "2022-11-12"

(ρ m movie ⋈ m . movie\_id = t . movie\_id

ρ t twitter\_tweets ⋈ t . tweet\_id = tt . tweet\_id

ρ tt tweet\_tags)

#### **Q5.** List movies with most tweets fetched from tweets between 2022-11-1 between 2022-11-12.

**Query**

SELECT m . movie\_id, COUNT (t . tweet\_id)AS tweet\_count

FROM movie AS m INNER JOIN twitter\_tweets AS t ON m . movie\_id = t . movie\_id

GROUP BY t . movie\_id

ORDER BY tweet\_count DESC

LIMIT 1;

**Relational Algebra**

τ tweet\_count ↓

π m . movie\_id, COUNT (tweet\_id) → tweet\_count

γ movie\_id, COUNT (tweet\_id)

(ρ m movie ⋈ m . movie\_id = t . movie\_id

ρ t twitter\_tweets)

**ASSIGNMENT - 3**

**Q6.** List of all the movies and their ratings for the actor - Shah Rukh Khan?

**Query:**

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";

**Relational Algebra:**

π m . name → srk, m . rating

σ s . name = "Shah Rukh Khan"

(ρ ms movie\_stars ⋈ ms . star\_id = s . star\_id

ρ s stars ⋈ m . movie\_id = ms . movie\_id

ρ m movie)

**Q7.** Movies of which genre have the highest rating?

**Query:**

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

**Relational Algebra:**

τ rating ↓

γ genre, genre\_id, MAX (rating)

(ρ mg movie\_genre ⋈ mg . genre\_id = g . genre\_id

ρ g genre ⋈ m . movie\_id = mg . genre\_id

ρ m movie)

**Q8.** What are the different ticket prices across all screens and how many screen ids come under each ticket price?

**Query:**

SELECT COUNT(screen\_id), MAX(ticket\_price)

FROM screenshows

GROUP BY ticket\_price

ORDER BY ticket\_price DESC

**Relational Algebra:**

τ ticket\_price ↓

π COUNT (screen\_id), MAX (ticket\_price)

γ ticket\_price, COUNT (screen\_id), MAX (ticket\_price) screenshows

**Q9.** How many screens are there in every city?

**Query:**

SELECT c.name, COUNT(t.screens)

FROM city c

INNER JOIN theaters t ON t.city\_id = c.city\_id

GROUP BY c.name

ORDER BY COUNT(t.screens) DESC

**Relational Algebra:**

τ COUNT (screens) ↓

γ name, COUNT (screens)

(ρ c city ⋈ t . city\_id = c . city\_id

ρ t theaters)

**Q10.** In which theaters were all the movies released in the year 2020?

**Query:**

SELECT t.name

FROM theaters t

INNER JOIN screens s ON t.theater\_id = s.theater\_id

INNER JOIN screenshows 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

**Relational Algebra:**

γ name,

σ m . released\_year = "2020"

(ρ t theaters ⋈ t . theater\_id = s . theater\_id

ρ s screens ⋈ ss . screen\_id = s . screen\_id

ρ ss screenshows ⋈ m . movie\_id = ss . movie\_id

ρ m movie)