**--HIGHEST GROSSING FILM**

SELECT film\_id, title, (revenue-budget) as profit

FROM movie

ORDER by profit DESC

A screenshot of a computer

Description automatically generated

**--SELECT all Movies with the name 'Avatar' or 'Avengers' in them**

SELECT film\_id, title

FROM movie

WHERE title LIKE '%Avatar%' OR title LIKE '%Avengers%'

A screenshot of a computer

Description automatically generated

**--Finding the count of movies under of a particular Movie collection**

SELECT m.collection\_id, c.name, COUNT(m.film\_id) as movie\_count

FROM movie as m

JOIN collection as c

ON m.collection\_id = c.collection\_id

WHERE c.name LIKE '%Star War%' or c.name LIKE '%Mission%'

GROUP BY m.collection\_id,c.name

A screenshot of a computer

Description automatically generated

**--List name of all companies together that made a film**

**--using string\_agg function and since its a Integer had to cast the column to a STRING(text)**

SELECT m.film\_id, m.title, string\_agg(mc.company\_id::text, ',') as company\_id,string\_agg(c.name, ',')

from movie as m

JOIN movie\_company as mc

ON m.film\_id = mc.film\_id

JOIN company as c

ON mc.company\_id = c.company\_id

GROUP BY m.film\_id

A screenshot of a computer program

Description automatically generated

In continuation to the above, now selectin all companies that made films where “PIXAR” company was involved

The Pseduo table ( CTE Table) gave us the table above and from there I am extracting companies who made films which involved PIXAR.

**--List all films where Pixar company was involved in creating the film.**

**----using WINDOWS function & string\_agg function and since its a Integer had to cast the column to a STRING(text)**

WITH X as (SELECT m.film\_id, m.title, string\_agg(mc.company\_id::text, ',') as company\_id,string\_agg(c.name, ',') as company\_names

from movie as m

JOIN movie\_company as mc

ON m.film\_id = mc.film\_id

JOIN company as c

ON mc.company\_id = c.company\_id

GROUP BY m.film\_id, m.title

)

SELECT X.film\_id, X.title, X.company\_names

FROM X

WHERE X.company\_names LIKE '%Pixar%'

A screenshot of a computer

Description automatically generated

**-- Find out all movies of TOM CRUISE where the rating of the movie is above 7**

*-- We first find Tom cruise ID number in the DB*

SELECT person\_id, name from crew\_credit

WHERE name LIKE '%Tom Cru%'

*--USING CTE we created a movies table having rating more than 7 and then an actor TABLE containing*

*--all TOM Cruise's movies.*

WITH movies as (

SELECT film\_id, title, rating

FROM movie

WHERE rating > 7.0

),

actor as (

SELECT film\_id, person\_id, name

FROM crew\_credit

WHERE person\_id = 500

)

SELECT movies.film\_id, actor.person\_id, actor.name, movies.title, movies.rating

FROM movies

JOIN actor

ON movies.film\_id = actor.film\_id

A screenshot of a computer

Description automatically generated

**-- Show all movies released in Oct 2020 where rating is greater than 6**

SELECT film\_id, title, release\_date, rating from movie

WHERE DATE\_PART('year', release\_date)=2020 and date\_part('month', release\_date)=10 and rating >6.0

A screenshot of a computer

Description automatically generated

-- FInd out all Directors of top rated Movies

WITH X as (

SELECT film\_id, title, rating, vote\_count

FROM movie

WHERE vote\_count >5000

ORDER BY rating DESC

),

Y as (

SELECT film\_id, department\_id, name

FROM crew\_credit

WHERE department\_id = 2

)

SELECT X.film\_id, X.title, X.rating, X.vote\_count, string\_agg(Y.name, ',') as director\_names

FROM X

JOIN Y

ON X.film\_id = Y.film\_id

GROUP BY X.film\_id, X.title, X.rating, X.vote\_count

ORDER BY X.rating DESC

A screenshot of a computer

Description automatically generated

**--SHOW the Directors who have the BEST RATED MOVIES on avg and votes received per movie is more than 5000**

**---clause : director should have directed 3 or more movies to be considered as top Director**

WITH X as (

SELECT cc.person\_id, cc.name as director, string\_agg(m.film\_id::text,',') as film\_ids, string\_agg(m.title,',') as film\_titles, ROUND(AVG(m.rating),2) as avg\_rating, SUM(m.vote\_count) as total\_votes

FROM crew\_credit as cc

JOIN movie as m

ON cc.film\_id = m.film\_id

WHERE cc.department\_id = 2 and m.vote\_count > 5000

GROUP BY CC.person\_id, cc.name

ORDER BY avg\_rating DESC

)

SELECT \*, SUM(length(x.film\_ids)-length(replace(X.film\_ids, ',',''))+1) as total\_films

FROM X

GROUP BY X.person\_id,X.director, X.film\_ids, X.film\_titles, X.avg\_rating, X.total\_votes

HAVING SUM(length(x.film\_ids)-length(replace(X.film\_ids, ',',''))+1) > 2

ORDER BY avg\_rating DESC

--SUM(length(x.film\_ids)-length(replace(X.film\_ids, ',',''))+1 counts the number of commas in the string field

A screenshot of a computer program

Description automatically generated

**--SHOW me the avg rating of the movies of the 4 actors along with the total count of movies.**

WITH X as (

SELECT cc.person\_id, cc.name as actor, string\_agg(m.film\_id::text,',') as film\_ids, string\_agg(m.title,',') as film\_titles, ROUND(AVG(m.rating),2) as avg\_rating, SUM(m.vote\_count) as total\_votes

FROM crew\_credit as cc

JOIN movie as m

ON cc.film\_id = m.film\_id

WHERE cc.name IN('Tom Cruise', 'Brad Pitt', 'Morgan Freeman', 'Leonardo DiCaprio')

GROUP BY CC.person\_id, cc.name

ORDER BY avg\_rating DESC

)

SELECT \*, SUM(length(x.film\_ids)-length(replace(X.film\_ids, ',',''))+1) as total\_films

FROM X

GROUP BY X.person\_id,X.actor, X.film\_ids, X.film\_titles, X.avg\_rating, X.total\_votes

HAVING SUM(length(x.film\_ids)-length(replace(X.film\_ids, ',',''))+1) > 4 -- SELECTING actors only who have acted in 5 movies and more

ORDER BY avg\_rating DESC

A screenshot of a computer

Description automatically generated