**QUESTIONS AND ANSWERS ON**

**Telugu Film Industry**

**1.Write a query to retrieve the names of all movies in which "Mahesh Babu" has acted?**

select M.MovieName from movies M

join cast C on M.MovieID=C.MovieID

join heroes on heroes.HeroID=C.HeroID

Where heroes.HeroName= "Mahesh Babu";

**2.Write a query to find the total box office collection of movies directed by "S.S. Rajamouli"?**

select Sum(C.BoxOfficeCollection) Total\_Collection\_rajamouli\_movies from collections C

Join movies on C.MovieID=movies.MovieID

WHERE movies.Director= "S.S. Rajamouli";

**3.Write a query to list the names of heroes who have acted in more than one movie?**

SELECT H.HeroName, COUNT(C.MovieID) AS MovieCount FROM Heroes H

JOIN Cast C ON H.HeroID = C.HeroID

GROUP BY H.HeroName

HAVING COUNT(C.MovieID) > 1;

**4. Write a query to find all movies released in 2018 along with the names of the heroes who acted in them?**

select M.MovieName,h.HeroName from movies M

join cast C on M.MovieID=C.MovieID

join heroes h on h.HeroID=C.HeroID

where year(M.ReleaseDate)= 2018;

**5.Write a query to retrieve the movie names and their total box office collections where "Prabhas" played a role?**

select M.MovieName,CC.BoxOfficeCollection from movies M

join cast C on M.MovieID=C.MovieID

join heroes h on h.HeroID=C.HeroID

join collections CC on CC.MovieID=C.MovieID

where h.HeroName= "Prabhas" ;

**6 Write a query to categorize movies into 'Low Budget', 'Medium Budget', and 'High Budget' based on the following criteria:**

**#Low Budget: Budget less than 400,000,000**

**#Medium Budget: Budget between 400,000,000 and 1,000,000,000**

**#High Budget: Budget greater than 1,000,000,000**

select MovieName,Budget,

Case When Budget < 400000000 then 'Low Budget'

WHEN Budget between 400000000 and 1000000000 then 'Medium Budget'

When Budget < 1000000000 then 'High Budget'

else

"More Than High Budget"

End As "BudgetCategory"

from movies;

**#7.Write a query to categorize heroes based on the number of awards won:**

**#'No Awards' if AwardsWon = 0**

**#'Few Awards' if AwardsWon between 1 and 5**

**#'Many Awards' if AwardsWon between 6 and 10**

**#'Legend' if AwardsWon > 10**

SELECT HeroName,AwardsWon,

CASE WHEN AwardsWon=0 THEN "NO AWARDS"

WHEN AwardsWon between 1 and 5 THEN "FEW AWARDS"

WHEN AwardsWon between 6 and 10 THEN "MANY AWARDS"

WHEN AwardsWon>10 THEN " LEGEND"

END AS AWARDLEVEL

FROM HEROES;

**8.Write a query to classify movies as 'Flop', 'Average', 'Hit', or 'Blockbuster' based on BoxOfficeCollection:**

**#Flop: BoxOfficeCollection < 500,000,000**

**#Average: BoxOfficeCollection between 500,000,000 and 1,000,000,000**

**#Hit: BoxOfficeCollection between 1,000,000,000 and 2,000,000,000**

**#Blockbuster: BoxOfficeCollection > 2,000,000,000**

select M.MovieName,C.BoxOfficeCollection,

CASE WHEN BoxOfficeCollection < 500000000 THEN "FLOP"

WHEN BoxOfficeCollection between 500000000 and 1000000000 THEN "AVERAGE"

WHEN BoxOfficeCollection between 1000000000 and 2000000000 THEN "HIT"

WHEN BoxOfficeCollection > 2000000000 THEN "BLOCKBUSTER"

END AS "SuccessCategory"

FROM collections C

join movies M ON C.MovieID=M.MovieID;

**9.Write a query to classify movies as 'Recent' or 'Classic':**

**#'Recent' if the movie was released in or after 2015**

**#'Classic' if the movie was released before 2015**

SELECT MovieName,ReleaseDate,

CASE WHEN year(ReleaseDate) > 2015 THEN "RECENT"

WHEN YEAR(ReleaseDate)<= 2015 THEN "CLASSIC"

END AS "MovieAgeCategory"

FROM movies;

**10.Write a query to classify heroes based on the decade they debuted:**

**#'70s' for debut between 1970 and 1979**

**#'80s' for debut between 1980 and 1989**

**#'90s' for debut between 1990 and 1999**

**#'2000s' for debut between 2000 and 2009**

**#'2010s' for debut in or after 2010**

select HeroName,DebutYear,

CASE WHEN DebutYear between 1970 and 1979 THEN "70s"

WHEN DebutYear between 1980 and 1989 THEN "80s"

WHEN DebutYear between 1990 and 1999 THEN "90s"

WHEN DebutYear between 2000 and 2009 THEN "2000s"

WHEN DebutYear >=2010 THEN "2010s"

END AS " DEBUT DECADE"

FROM heroes;

**11.Find the total box office collections for each hero across all their movies and rank the heroes based on these collections in descending order. Use the RANK() window function**

SELECT h.HeroName,sum(CC.BoxOfficeCollection) AS TOATAL\_BOX\_OFFICE,

RANK() OVER(order by sum(CC.BoxOfficeCollection) desc)AS HEROS\_RANK\_ON\_COLLECTION from heroes h

join Cast C on h.HeroID=C.HeroID

join collections CC on CC.MovieID=C.MovieID

group by h.HeroName;

**12. Find the Top 3 Movies for Each Hero Based on Collections?**

**# For each hero, find the top 3 movies based on the box office collections using the ROW\_NUMBER() window function?**

select h.HeroName,M.MovieName,CC.BoxOfficeCollection,

row\_number() OVER(order by CC.BoxOfficeCollection DESC) AS Top\_THREE\_BoxOfficeCollection

from movies M

join cast C on M.MovieID=C.MovieID

join heroes h on h.HeroID=C.HeroID

join collections CC on CC.MovieID=C.MovieID

LIMIT 3;

**13.Calculate the Cumulative Box Office Collections for Each Hero?**

**#Calculate the cumulative box office collections for each hero over time, based on the release dates of their movies. Use the SUM() window function?**

select h.HeroName,h.HeroName,M.ReleaseDate,

sum(CC.BoxOfficeCollection) OVER (order by M.ReleaseDate DESC) AS CumulativeBoxOffice from movies M

join cast C on M.MovieID=C.MovieID

join heroes h on h.HeroID=C.HeroID join collections CC on CC.MovieID=C.MovieID;

**14.Determine the Difference in Collections Between Consecutive Movies for Each Hero?**

**#For each hero, calculate the difference in box office collections between consecutive movies using the LAG() window function?**

SELECT

h.HeroName,

m.MovieName,

c.BoxOfficeCollection,

LAG(c.BoxOfficeCollection,1,0) OVER(PARTITION BY h.HeroID ORDER BY m.ReleaseDate) AS PreviousMovieCollection,

( c.BoxOfficeCollection-LAG(c.BoxOfficeCollection,1,0) OVER(PARTITION BY h.HeroID ORDER BY m.ReleaseDate)) AS CollectionDifference

FROM

Heroes h

JOIN Cast ca ON h.HeroID = ca.HeroID

JOIN Movies m ON ca.MovieID = m.MovieID

JOIN Collections c ON m.MovieID = c.MovieID

ORDER BY

h.HeroName, m.ReleaseDate;

**15.Find the Average Box Office Collection for the Last 3 Movies of Each Hero?**

**#Calculate the average box office collection for the last 3 movies of each hero using the AVG() window function?**

SELECT h.HeroName,M.MovieName,M.ReleaseDate,CC.BoxOfficeCollection,

avg(CC.BoxOfficeCollection) over(partition by h.HeroName order by avg(CC.BoxOfficeCollection) desc) as AVRAGE\_COLLECTION

FROM

Heroes h

JOIN Cast ca ON h.HeroID = ca.HeroID

JOIN Movies m ON ca.MovieID = m.MovieID

JOIN Collections CC ON m.MovieID = CC.MovieID

group by h.HeroName,M.MovieName,CC.BoxOfficeCollection,M.ReleaseDate;

**16.1. Find the Highest-Grossing Movie for Each Hero?**

**#Write a query to find the highest-grossing movie for each hero using a CTE?**

WITH Highest\_Grossing\_Movie AS

(SELECT h.HeroName,m.MovieName,CC.BoxOfficeCollection,

ROW\_NUMBER() OVER (PARTITION BY h.HeroID ORDER BY CC.BoxOfficeCollection DESC) AS Rank\_C

FROM Heroes h

JOIN Cast ca ON h.HeroID = ca.HeroID

JOIN Movies m ON ca.MovieID = m.MovieID

JOIN Collections CC ON m.MovieID = CC.MovieID)

SELECT HeroName, MovieName, BoxOfficeCollection

FROM Highest\_Grossing\_Movie

WHERE Rank\_C = 1;

**17.Calculate the Total Box Office Collection of Movies Released After 2015 for Each Hero?**

**#Use a CTE to calculate the total box office collection of movies released after 2015 for each hero?**

WITH MVAT AS(

SELECT h.HeroName,SUM(CC.BoxOfficeCollection) AS ToTAL\_COLLECTION

FROM Heroes h

JOIN Cast ca ON h.HeroID = ca.HeroID

JOIN movies M ON ca.MovieID = M.MovieID

JOIN Collections CC ON M.MovieID = CC.MovieID

WHERE

M.ReleaseDate > '2015-01-01'

GROUP BY

h.HeroName )

SELECT HeroName, ToTAL\_COLLECTION from MVAT

order by ToTAL\_COLLECTION desc;

18.Find Heroes Who Acted in More Than 3 Movies and Their Total Box Office Collection?

#Create a CTE to find heroes who have acted in more than 3 movies, along with their total box office collection?

with RR AS(

SELECT h.HeroName,Count(M.MovieID) AS TOTAL\_MOVIES,SUM(CC.BoxOfficeCollection) AS TOTAL\_COOLECTION

FROM Heroes h

JOIN Cast ca ON h.HeroID = ca.HeroID

JOIN movies M ON ca.MovieID = M.MovieID

JOIN Collections CC ON M.MovieID = CC.MovieID

GROUP BY h.HeroName

HAVING TOTAL\_MOVIES >=3)

SELECT HeroName,TOTAL\_MOVIES,TOTAL\_COOLECTION FROM RR;

**19.List Movies That Have Collected More Than the Average Collection**

**#Use a CTE to find movies that have collected more than the average box office collection across all movies?**

WITH AvgCollection AS

( SELECT avg(collections.BoxOfficeCollection) AS AverageCollection FROM collections)

SELECT

m.MovieName,

c.BoxOfficeCollection

FROM

Movies m

JOIN Collections c ON m.MovieID = c.MovieID

WHERE

c.BoxOfficeCollection > (SELECT AverageCollection FROM AvgCollection);

**20.Identify Heroes Whose Movies Have an Average Collection of Over 2 Billion?,2000000000**

**#Use a CTE to identify heroes whose movies have an average box office collection of over 2 billion?**

WITH AvgCollectionS AS(

SELECT h.HeroName,AVG(CC.BoxOfficeCollection) AS AVGG FROM

Heroes h

JOIN Cast ca ON h.HeroID = ca.HeroID

JOIN Movies M ON ca.MovieID = M.MovieID

JOIN Collections CC ON M.MovieID = CC.MovieID

group by h.HeroName)

SELECT HeroName,AVGG FROM AvgCollectionS

WHERE AVGG>=2000000000;

**21.Create a view to display all hero details along with the total number of movies they acted in?**

CREATE VIEW FILMS\_ACTED AS

SELECT h.HeroName,count(M.MovieID) AS MOVIES\_ACTED from movies M

join cast C on M.MovieID=C.MovieID

join heroes h on h.HeroID=C.HeroID

GROUP BY h.HeroName;

select \* from FILMS\_ACTED

order by MOVIES\_ACTED desc;

**22. Create a view that shows the top 5 movies with the highest box office collections?**

create view top\_gross\_movies as

select M.MovieName,CC.BoxOfficeCollection FROM Movies M

JOIN Collections CC ON M.MovieID = CC.MovieID

ORDER BY CC.BoxOfficeCollection DESC

LIMIT 5;

SELECT \* FROM top\_gross\_movies;

**23.Find the names of movies released in the year 2018?**

SELECT MovieName,ReleaseDate FROM movies

WHERE YEAR(ReleaseDate)=2018;

**24.List all heroes who have their birthdays in the month of August?**

SELECT HeroName,BirthDate FROM heroes

WHERE MONTH(BirthDate)=08;

**25.Find the total box office collections of movies released after January 1, 2019?**

SELECT SUM(CC.BoxOfficeCollection) AS COLLECTION FROM

Movies M

JOIN Collections CC ON m.MovieID = CC.MovieID

WHERE DATE(M.ReleaseDate)>2019-01-01;

**26.Which movies celebrated their 5th anniversary in the year 2024?**

SELECT MovieName, ReleaseDate

FROM Movies

WHERE YEAR(ReleaseDate) = 2019;

**26.Calculate the difference in days between the release dates of 'Baahubali: The Beginning' and 'Baahubali: The Conclusion'?**

SELECT DATEDIFF(

(SELECT ReleaseDate FROM Movies WHERE MovieName = 'Baahubali: The Conclusion'),

(SELECT ReleaseDate FROM Movies WHERE MovieName = 'Baahubali: The Beginning')

) AS DaysBetweenReleases;

**27.Calculate the Difference in Box Office Collections Between Consecutive Movies?**

SELECT

MovieName,

ReleaseDate,

BoxOfficeCollection,

LAG(BoxOfficeCollection, 1) OVER (ORDER BY ReleaseDate) AS PreviousCollection,

(BoxOfficeCollection - LAG(BoxOfficeCollection, 1) OVER (ORDER BY ReleaseDate)) AS CollectionDifference

FROM

Movies M

JOIN

Collections C ON M.MovieID = C.MovieID;

**28.Find the Previous Movie's Box Office Collection for Each Movie?**

SELECT

MovieName,

ReleaseDate,

BoxOfficeCollection,

LAG(BoxOfficeCollection, 1) OVER (ORDER BY ReleaseDate) AS PreviousCollection

FROM

Movies M

JOIN

Collections C ON M.MovieID = C.MovieID;

**29.Find the Next Hero’s Movie After Each Movie?**

SELECT

h.HeroName,

M.MovieName,

M.ReleaseDate,

LEAD(M.MovieName,1) OVER(partition by h.HeroID order by M.ReleaseDate ) AS NXT\_MOVIE

FROM Heroes h

JOIN Cast ca ON h.HeroID = ca.HeroID

JOIN movies M ON ca.MovieID = M.MovieID;

**30.Compare the Opening Week Collection with the Previous Movie's Opening Week Collection for the Same Hero?**

select h.HeroName,M.MovieName,M.ReleaseDate,CC.OpeningWeekCollection,

LAG(CC.OpeningWeekCollection,1) OVER(ORDER BY ReleaseDate) AS PREVIWS\_WEEK\_OPEN\_COLLECTION,

(CC.OpeningWeekCollection-LAG(CC.OpeningWeekCollection,1) OVER(ORDER BY ReleaseDate)) AS OPEING\_WEEK\_COOLECTION

FROM Heroes h JOIN Cast ca ON h.HeroID = ca.HeroID

JOIN Movies M ON ca.MovieID = M.MovieID

JOIN Collections CC ON M.MovieID = CC.MovieID;