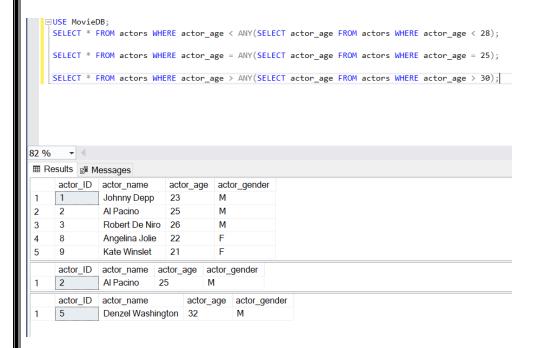
DBMS LAB ASSIGNMENT - 5

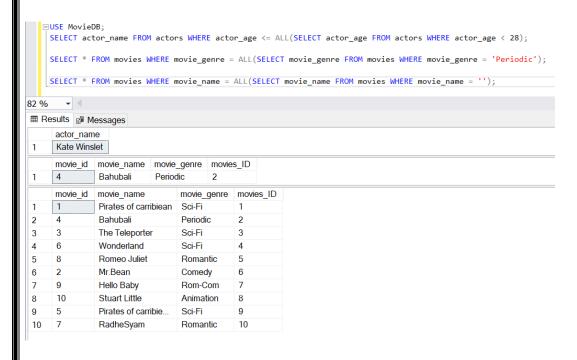
NAME: P. SATWIK ROLL NO.: 19BCS083

Q1) Illustrate logical ANY, ALL and LIKE operator- the queries should be relevant to your respective databases 3 queries for each operator. One query explaining the difference between ANY and ALL.

QUERIES FOR "ANY"



QUERIES FOR "ALL"

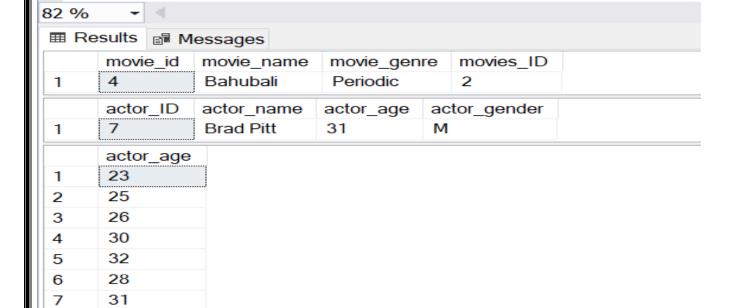


QUERIES FOR "LIKE"

```
SELECT * FROM movies WHERE movie_id LIKE 4;

SELECT * FROM actors WHERE actor_name LIKE '%Pitt';

SELECT actor_age FROM actors WHERE actor_gender LIKE 'M';
```



Q2) One query for each Aggregate function.

The aggregate functions are MIN(), MAX(), COUNT(), AVG(), SUM()

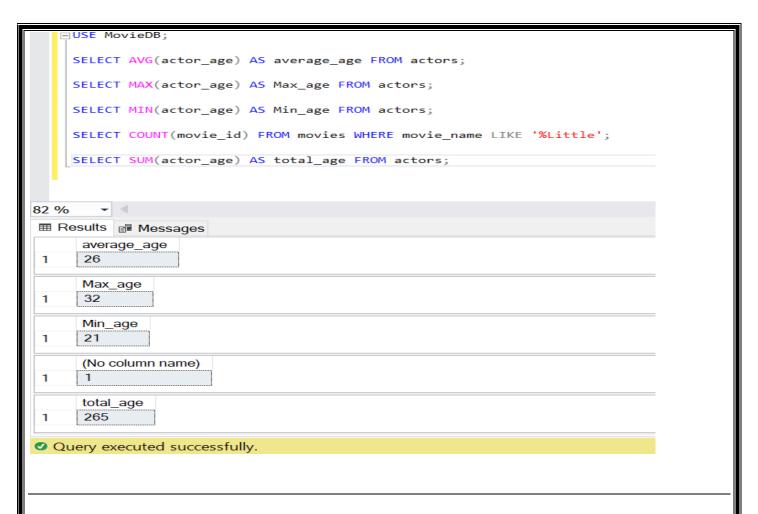
AVG() – return the average of the set

MIN() – returns the minimum value in a set

MAX() – returns the maximum value in set

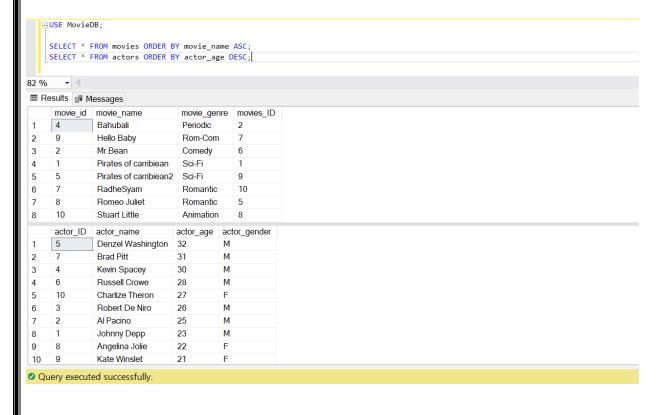
SUM() – returns the sum of all distinct values of a set

COUNT() – returns the number of items in a set



Q3) Illustrate the usage of order by, group by and having clause (2 queries for each case)

ORDER BY



GROUP BY □USE MovieDB; SELECT actor_age, COUNT(*) AS actor_id FROM actors GROUP BY actor_age; SELECT movie_name, COUNT(*) FROM movies GROUP BY movie_name; actor_age actor_id 2 22 3 23 25 4 26 6 27 7 28 8 30 movie_name (No column name) Bahubali 1 Hello Baby 2 3 Mr.Bean 4 Pirates of c... 1 5 Pirates of c... 6 RadheSyam 1 Romeo Juli... 1 8 Stuart Little 9 The Telepo... Wonderland 1 10 Query executed successfully. **HAVING CLAUSE** □USE MovieDB; SELECT COUNT(movies_ID), movie_name FROM movies GROUP BY movie_name HAVING COUNT(movies_ID) = 2; (No column name) movie_name

