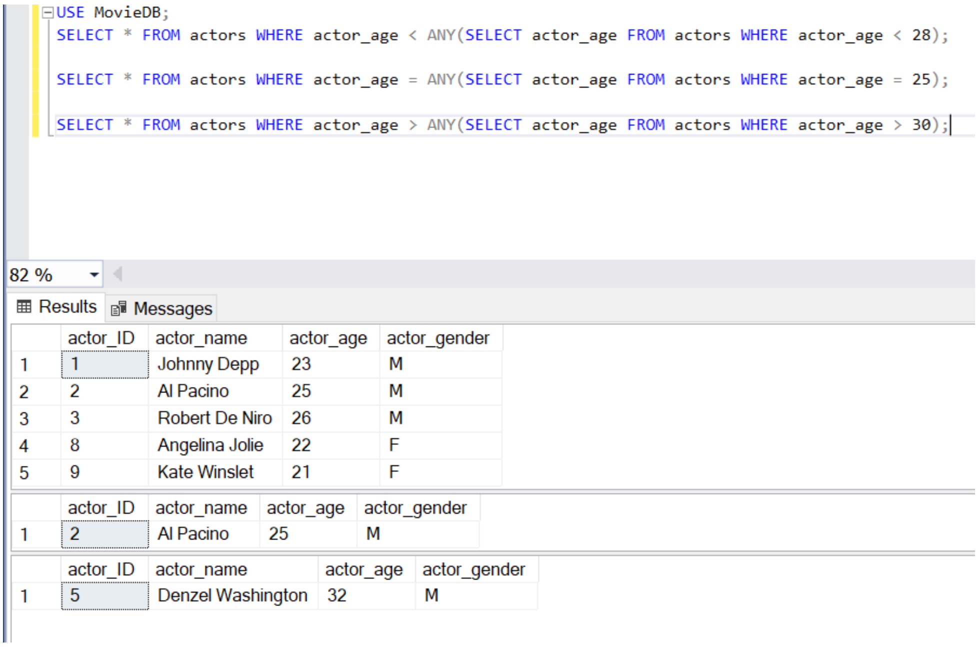
DBMS LAB ASSIGNMENT 5

V.KRISHNA SAI ROHITH
19BCS061
CSE

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"



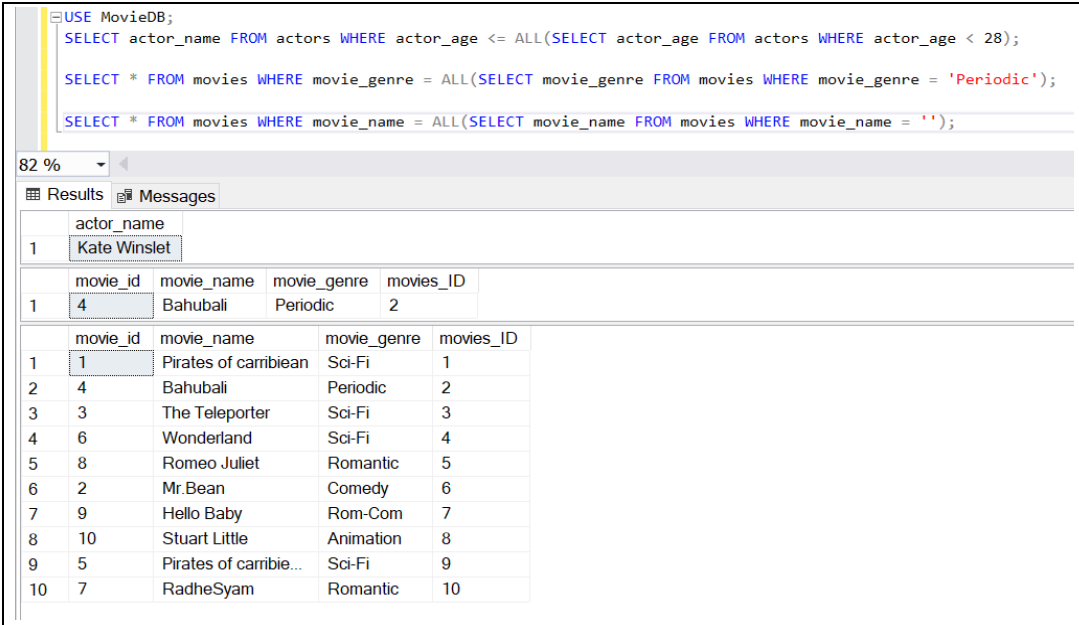
```
USE MovieDB;
SELECT * FROM actors WHERE actor_age < ANY(SELECT actor_age FROM actors WHERE actor_age < 28);

SELECT * FROM actors WHERE actor_age = ANY(SELECT actor_age FROM actors WHERE actor_age = 25);

SELECT * FROM actors WHERE actor_age > ANY(SELECT actor_age FROM actors WHERE actor_age > 30);
```

	actor_ID	actor_name	actor_age	actor_gender
1	1	Johnny Depp	23	M
2	2	Al Pacino	25	M
3	3	Robert De Niro	26	M
4	8	Angelina Jolie	22	F
5	9	Kate Winslet	21	F

	actor_ID	actor_name	actor_age	actor_gender
1	2	Al Pacino	25	M

	actor_ID	actor_name	actor_age	actor_gender
1	5	Denzel Washington	32	M

QUERIES FOR "ALL"



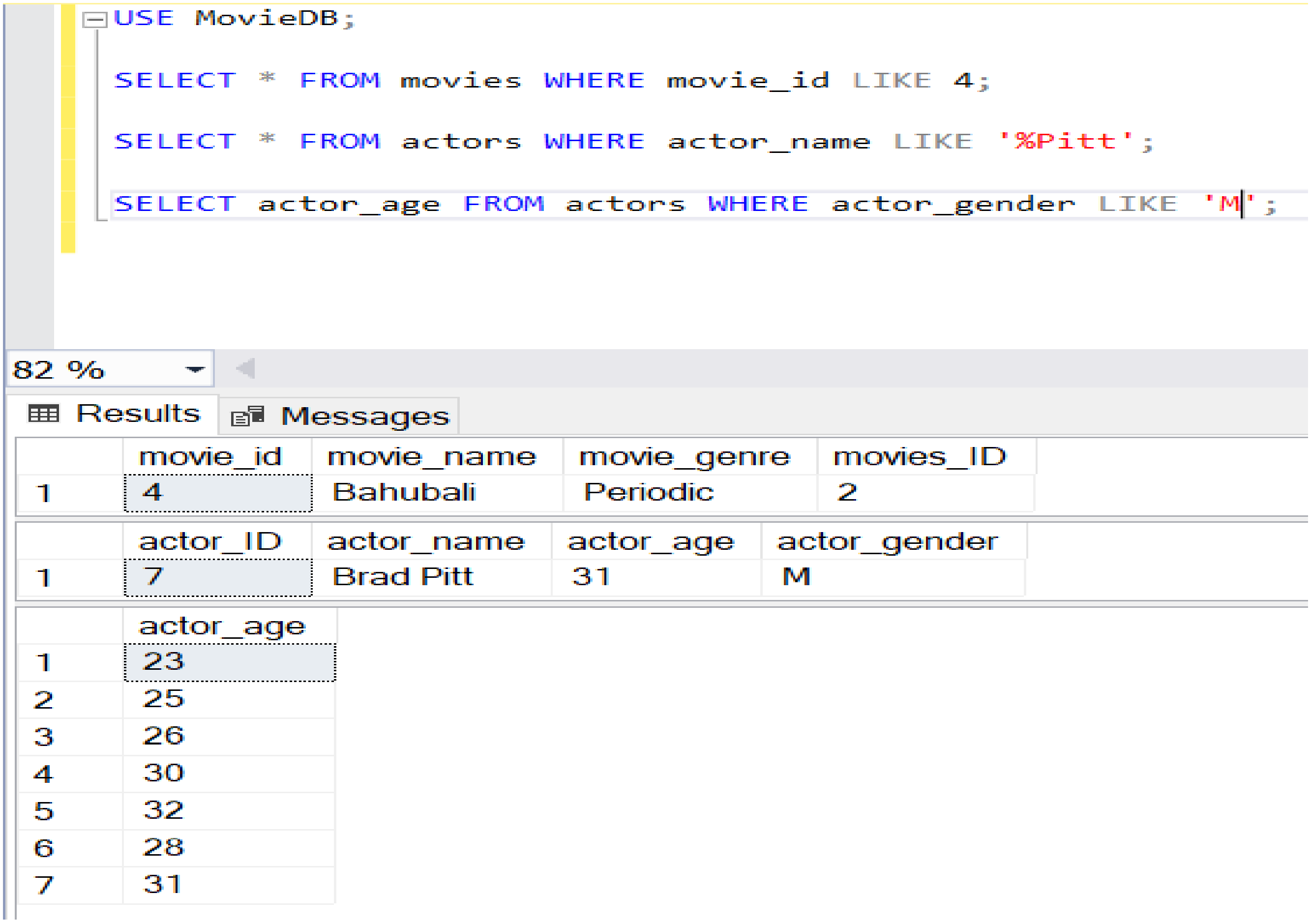
```sql
USE MovieDB;
SELECT actor_name FROM actors WHERE actor_age <= ALL(SELECT actor_age FROM actors WHERE actor_age < 28);

SELECT * FROM movies WHERE movie_genre = ALL(SELECT movie_genre FROM movies WHERE movie_genre = 'Periodic');

SELECT * FROM movies WHERE movie_name = ALL(SELECT movie_name FROM movies WHERE movie_name = '');
```

	actor_name
1	Kate Winslet

	movie_id	movie_name	movie_genre	movies_ID
1	4	Bahubali	Periodic	2

	movie_id	movie_name	movie_genre	movies_ID
1	1	Pirates of carribiean	Sci-Fi	1
2	4	Bahubali	Periodic	2
3	3	The Teleporter	Sci-Fi	3
4	6	Wonderland	Sci-Fi	4
5	8	Romeo Juliet	Romantic	5
6	2	Mr.Bean	Comedy	6
7	9	Hello Baby	Rom-Com	7
8	10	Stuart Little	Animation	8
9	5	Pirates of carribie...	Sci-Fi	9
10	7	RadheSyam	Romantic	10

QUERIES FOR "LIKE"



```sql
USE MovieDB;

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';
```

	movie_id	movie_name	movie_genre	movies_ID
1	4	Bahubali	Periodic	2

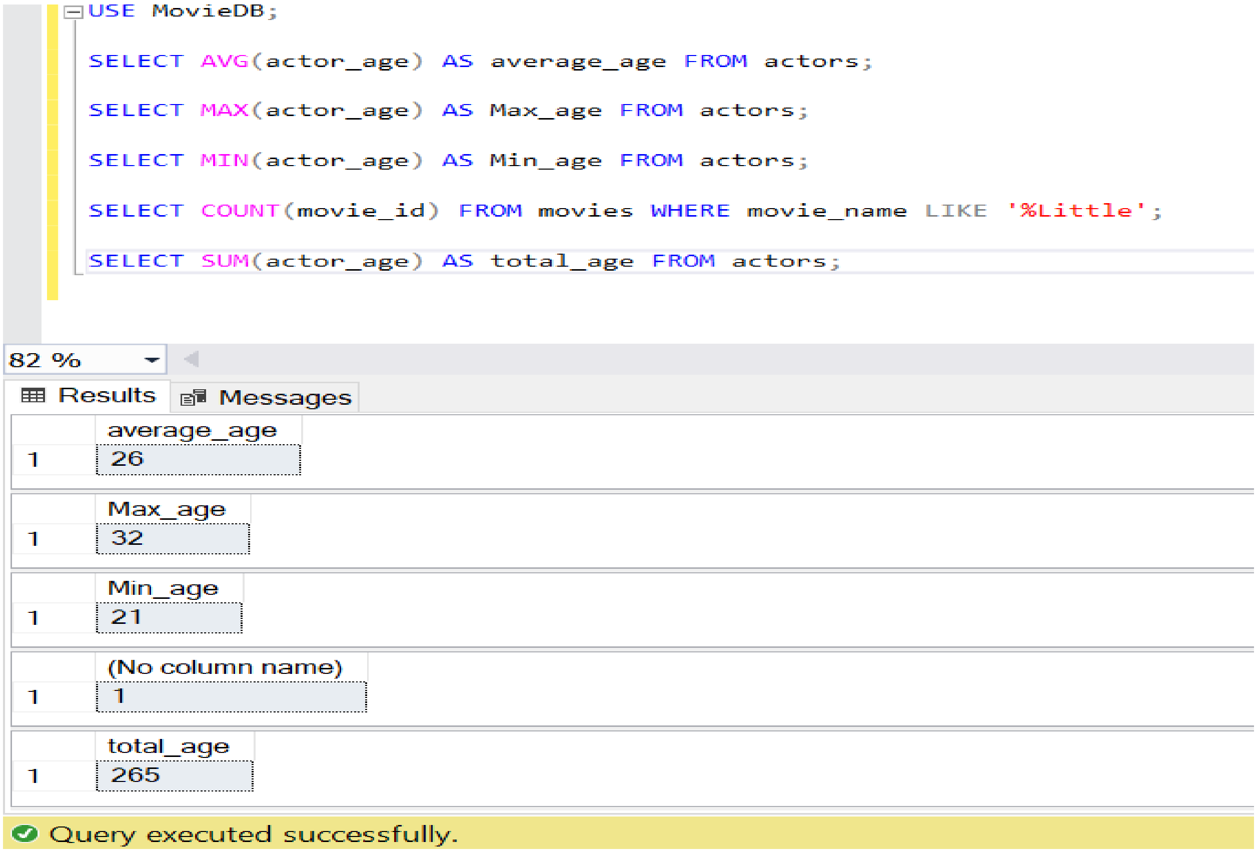
	actor_ID	actor_name	actor_age	actor_gender
1	7	Brad Pitt	31	M

	actor_age
1	23
2	25
3	26
4	30
5	32
6	28
7	31

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



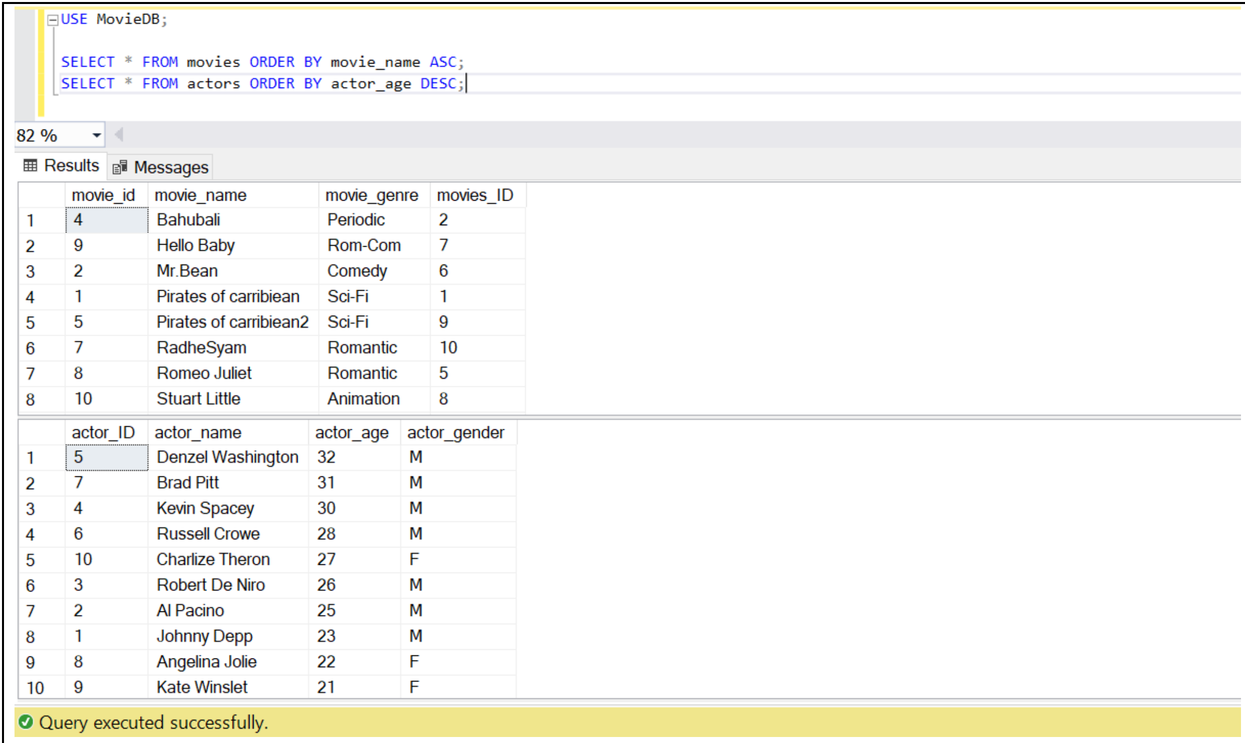
```
USE MovieDB;

SELECT AVG(actor_age) AS average_age FROM actors;

SELECT MAX(actor_age) AS Max_age FROM actors;

SELECT MIN(actor_age) AS Min_age FROM actors;

SELECT COUNT(movie_id) FROM movies WHERE movie_name LIKE '%Little';

SELECT SUM(actor_age) AS total_age FROM actors;
```

```
   average_age
1  26

   Max_age
1  32

   Min_age
1  21

   (No column name)
1  1

   total_age
1  265
```

Query executed successfully.

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

ORDER BY



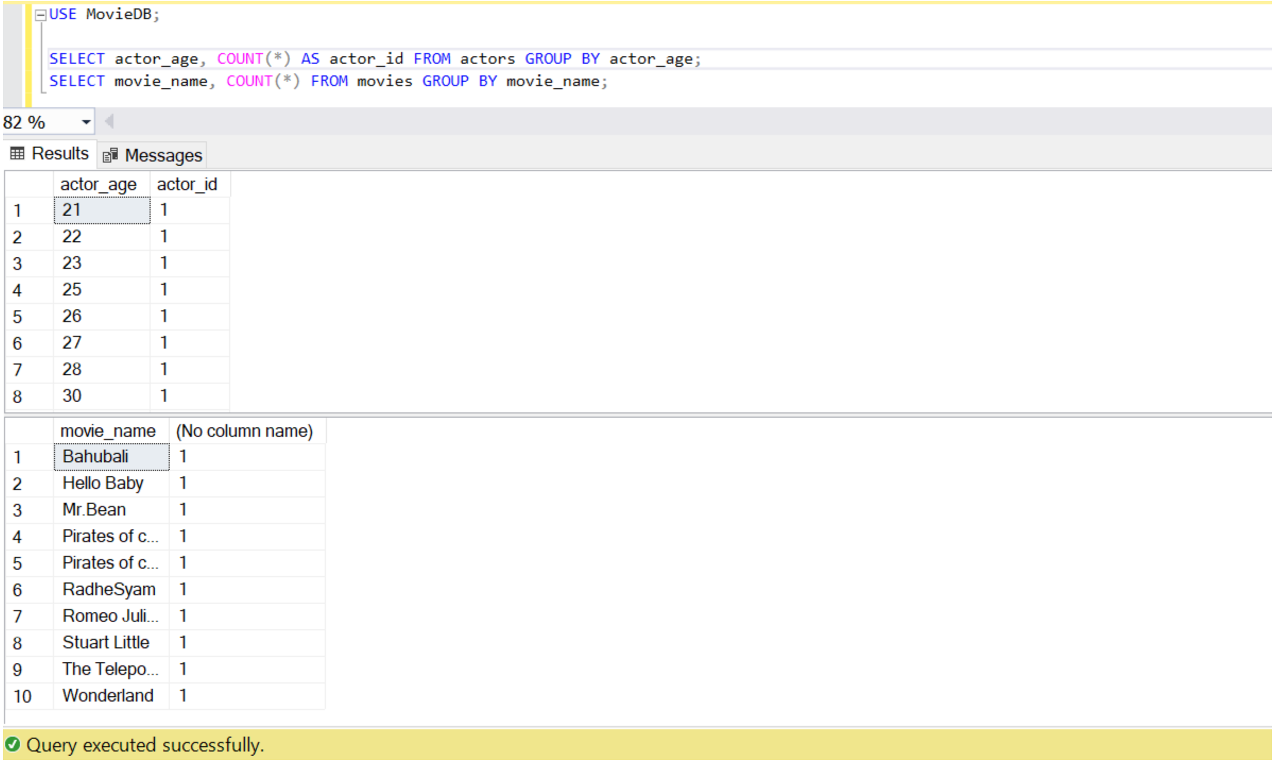
```
USE MovieDB;

SELECT * FROM movies ORDER BY movie_name ASC;
SELECT * FROM actors ORDER BY actor_age DESC;
```

	movie_id	movie_name	movie_genre	movies_ID
1	4	Bahubali	Periodic	2
2	9	Hello Baby	Rom-Com	7
3	2	Mr.Bean	Comedy	6
4	1	Pirates of carribiean	Sci-Fi	1
5	5	Pirates of carribiean2	Sci-Fi	9
6	7	RadheSyam	Romantic	10
7	8	Romeo Juliet	Romantic	5
8	10	Stuart Little	Animation	8

	actor_ID	actor_name	actor_age	actor_gender
1	5	Denzel Washington	32	M
2	7	Brad Pitt	31	M
3	4	Kevin Spacey	30	M
4	6	Russell Crowe	28	M
5	10	Charlize Theron	27	F
6	3	Robert De Niro	26	M
7	2	Al Pacino	25	M
8	1	Johnny Depp	23	M
9	8	Angelina Jolie	22	F
10	9	Kate Winslet	21	F

Query executed successfully.

GROUP BY



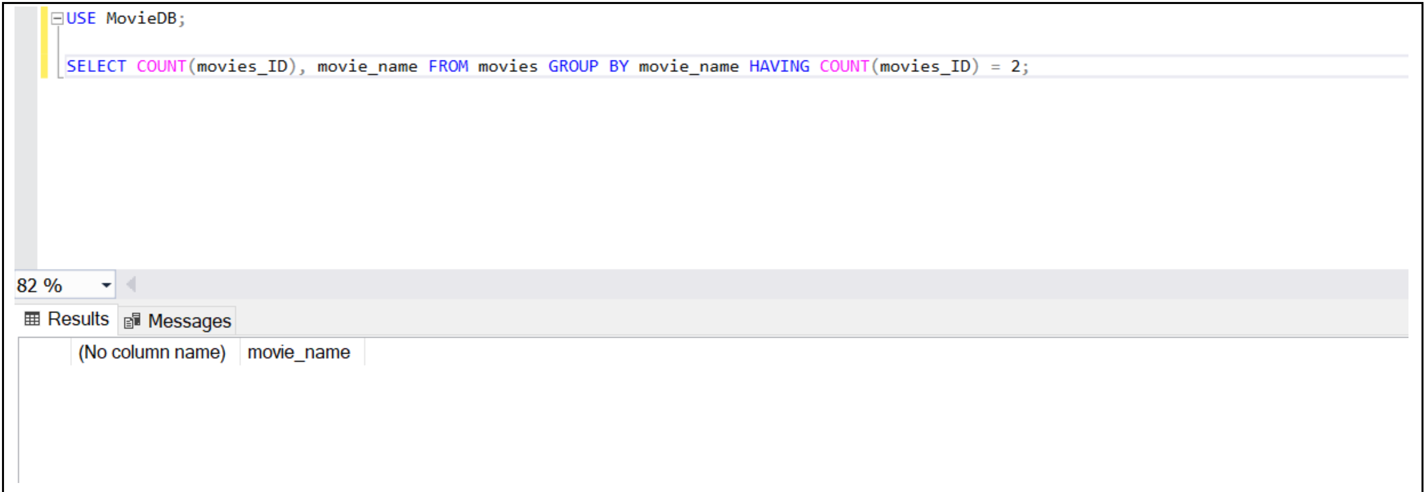
```
USE MovieDB;

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
1	21	1
2	22	1
3	23	1
4	25	1
5	26	1
6	27	1
7	28	1
8	30	1

	movie_name	(No column name)
1	Bahubali	1
2	Hello Baby	1
3	Mr.Bean	1
4	Pirates of c...	1
5	Pirates of c...	1
6	RadheSyam	1
7	Romeo Juli...	1
8	Stuart Little	1
9	The Telepo...	1
10	Wonderland	1

Query executed successfully.

HAVING CLAUSE

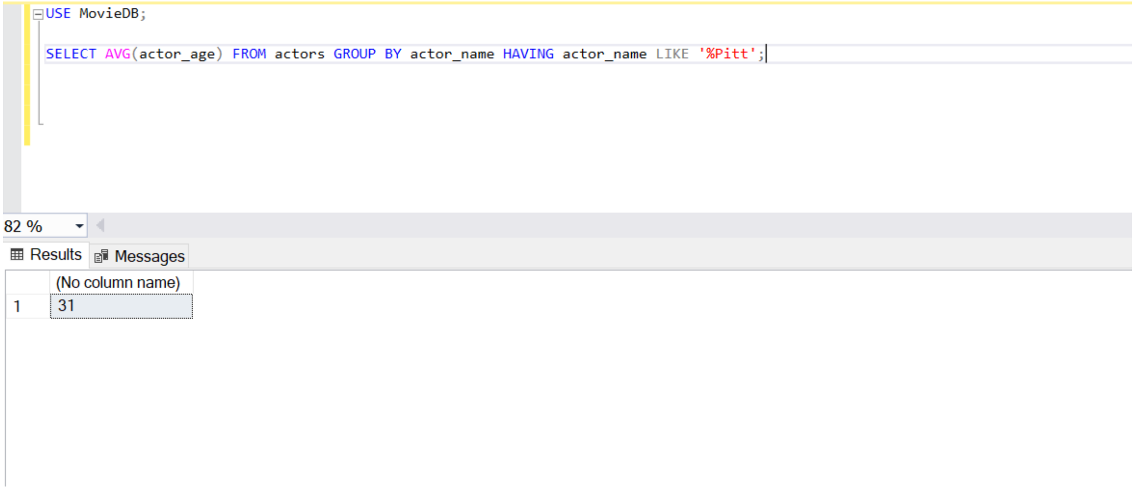


```
USE MovieDB;

SELECT COUNT(movies_ID), movie_name FROM movies GROUP BY movie_name HAVING COUNT(movies_ID) = 2;
```

```
   (No column name)   movie_name
```

Q4) Use Aggregate function with group by and having

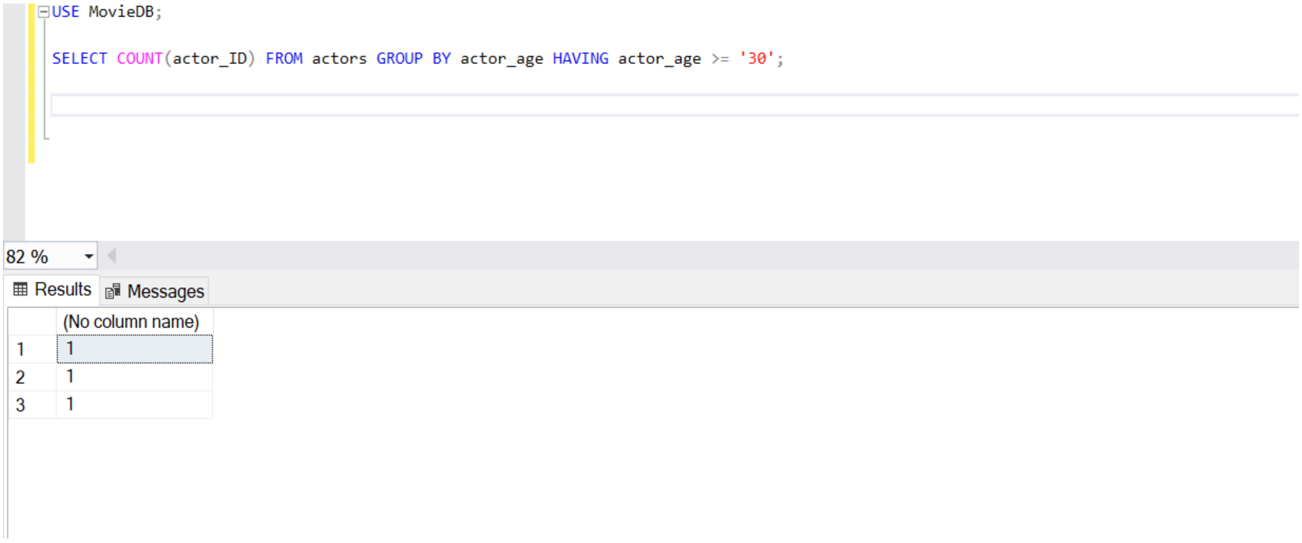
AVG():



```
USE MovieDB;

SELECT AVG(actor_age) FROM actors GROUP BY actor_name HAVING actor_name LIKE '%Pitt';
```

```
     (No column name)
1    31
```

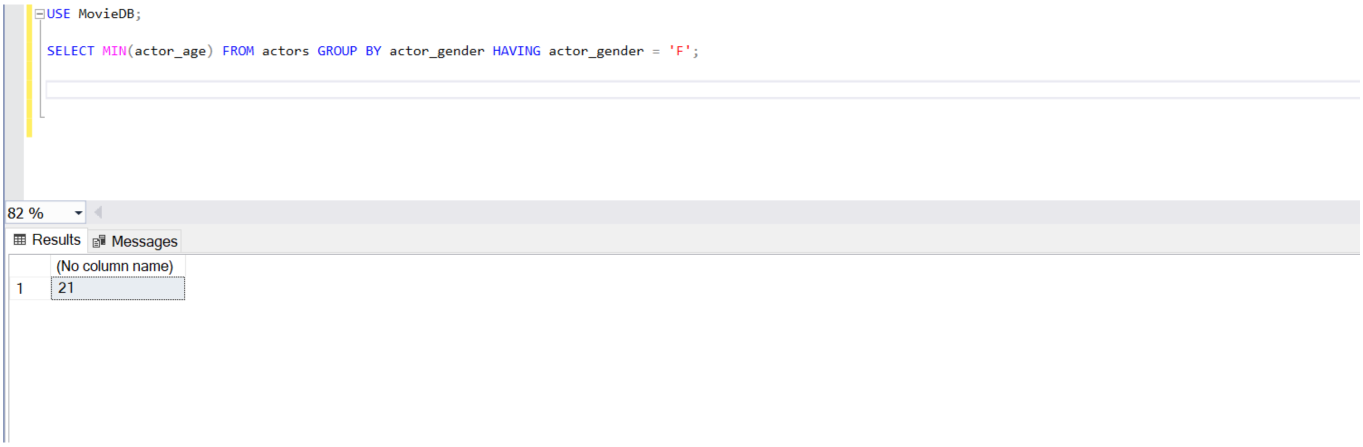
COUNT():



```
USE MovieDB;

SELECT COUNT(actor_ID) FROM actors GROUP BY actor_age HAVING actor_age >= '30';
```

```
     (No column name)
1    1
2    1
3    1
```

MIN():



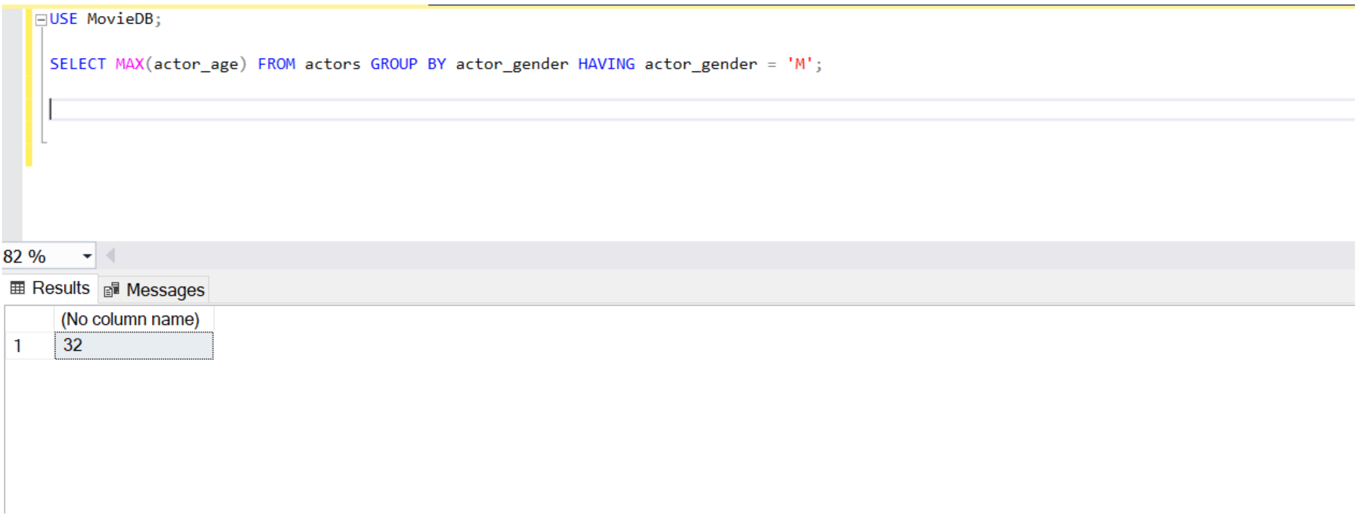
```sql
USE MovieDB;

SELECT MIN(actor_age) FROM actors GROUP BY actor_gender HAVING actor_gender = 'F';
```

82 %

Results | Messages

	(No column name)
1	21

MAX():



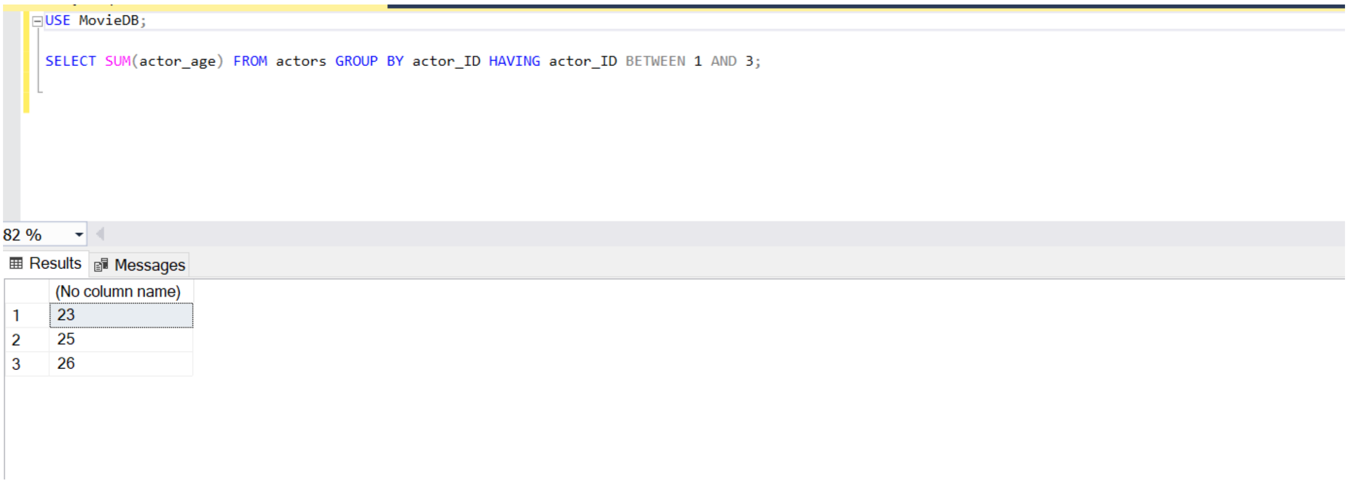
```sql
USE MovieDB;

SELECT MAX(actor_age) FROM actors GROUP BY actor_gender HAVING actor_gender = 'M';
```

82 %

Results | Messages

	(No column name)
1	32

SUM():



```sql
USE MovieDB;

SELECT SUM(actor_age) FROM actors GROUP BY actor_ID HAVING actor_ID BETWEEN 1 AND 3;
```

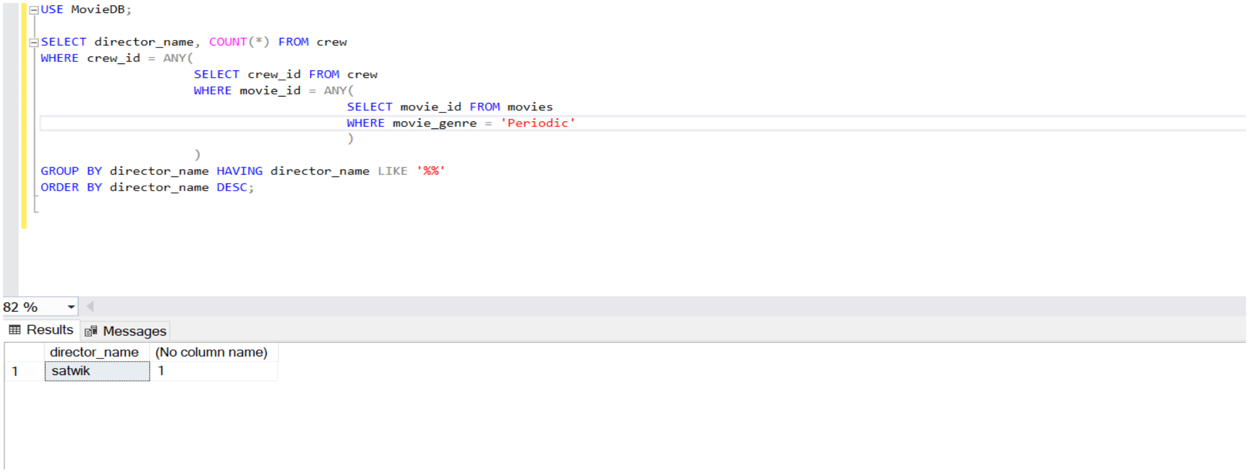
82 %

Results | Messages

	(No column name)
1	23
2	25
3	26

Q5) Write at least 3 nested queries using order by, group by and having clause.

QUERY:



```sql
USE MovieDB;

SELECT director_name, COUNT(*) FROM crew
WHERE crew_id = ANY(
                    SELECT crew_id FROM crew
                    WHERE movie_id = ANY(
                                        SELECT movie_id FROM movies
                                        WHERE movie_genre = 'Periodic'
                                        )
                    )
GROUP BY director_name HAVING director_name LIKE '%%'
ORDER BY director_name DESC;
```

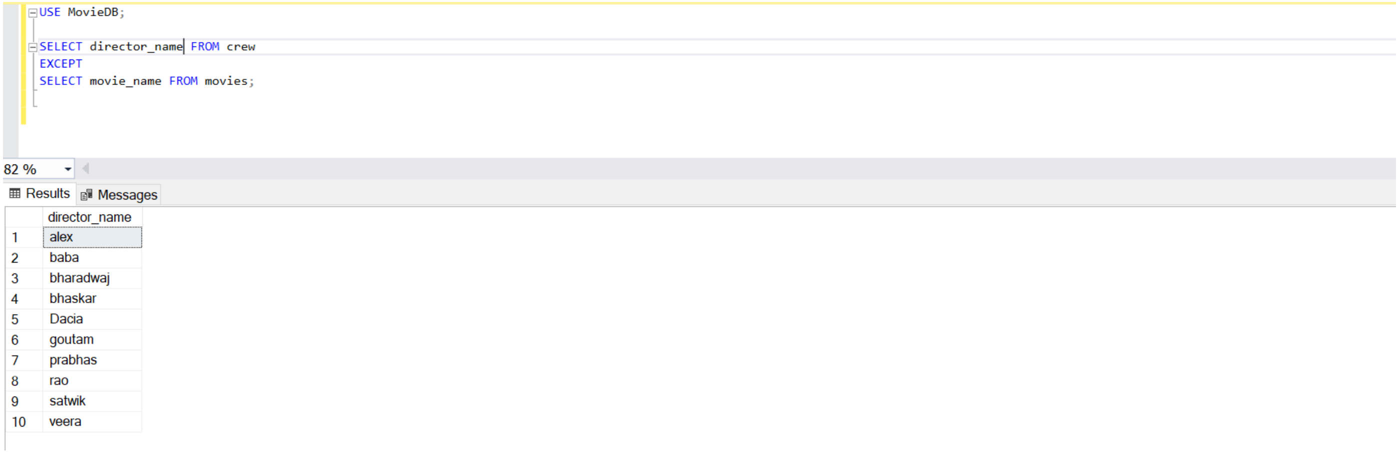
82 %

Results | Messages

	director_name	(No column name)
1	satwik	1

Q6) Illustrate the Usage of Except, Exists, Not Exists, Union, Intersection

EXCEPT():



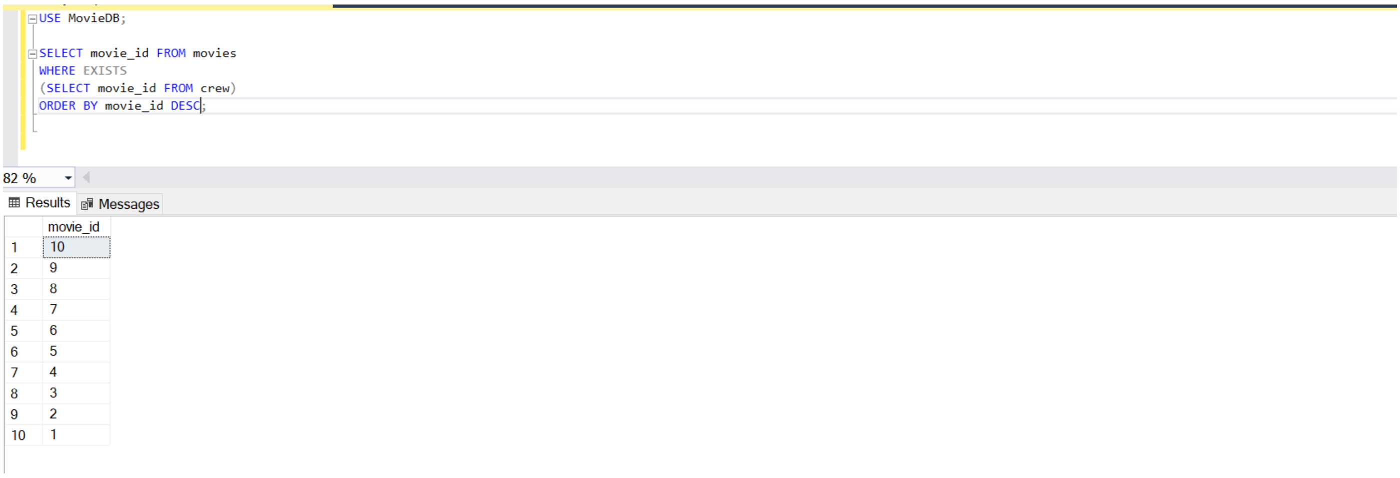
```sql
USE MovieDB;

SELECT director_name FROM crew
EXCEPT
SELECT movie_name FROM movies;
```

82 %

Results | Messages

	director_name
1	alex
2	baba
3	bharadwaj
4	bhaskar
5	Dacia
6	goutam
7	prabhas
8	rao
9	satwik
10	veera

EXISTS():



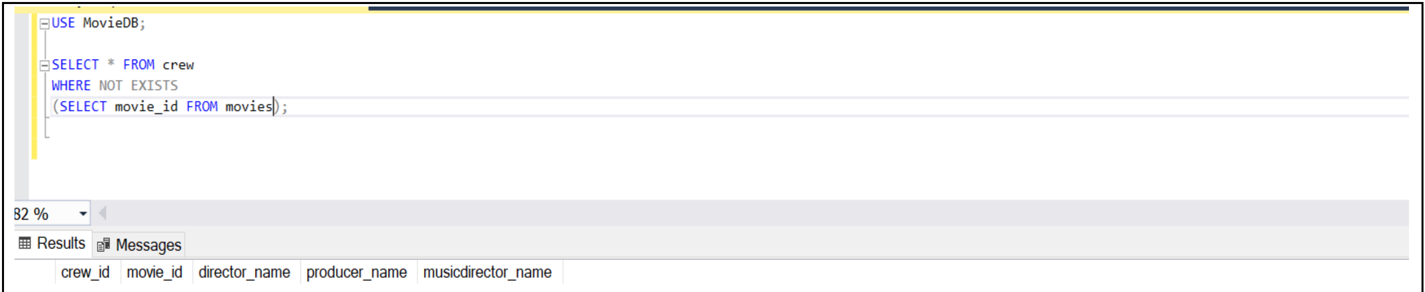
```sql
USE MovieDB;

SELECT movie_id FROM movies
WHERE EXISTS
(SELECT movie_id FROM crew)
ORDER BY movie_id DESC;
```

82 %

Results | Messages

	movie_id
1	10
2	9
3	8
4	7
5	6
6	5
7	4
8	3
9	2
10	1

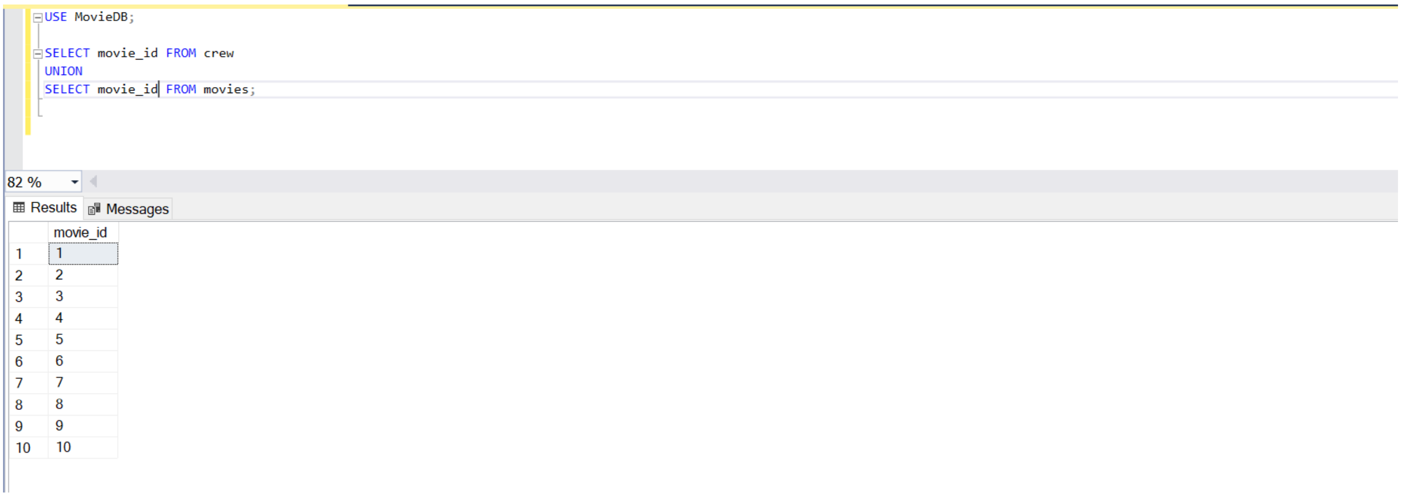
NOT EXISTS():



```
USE MovieDB;

SELECT * FROM crew
WHERE NOT EXISTS
(SELECT movie_id FROM movies);
```

```
crew_id  movie_id  director_name  producer_name  musicdirector_name
```

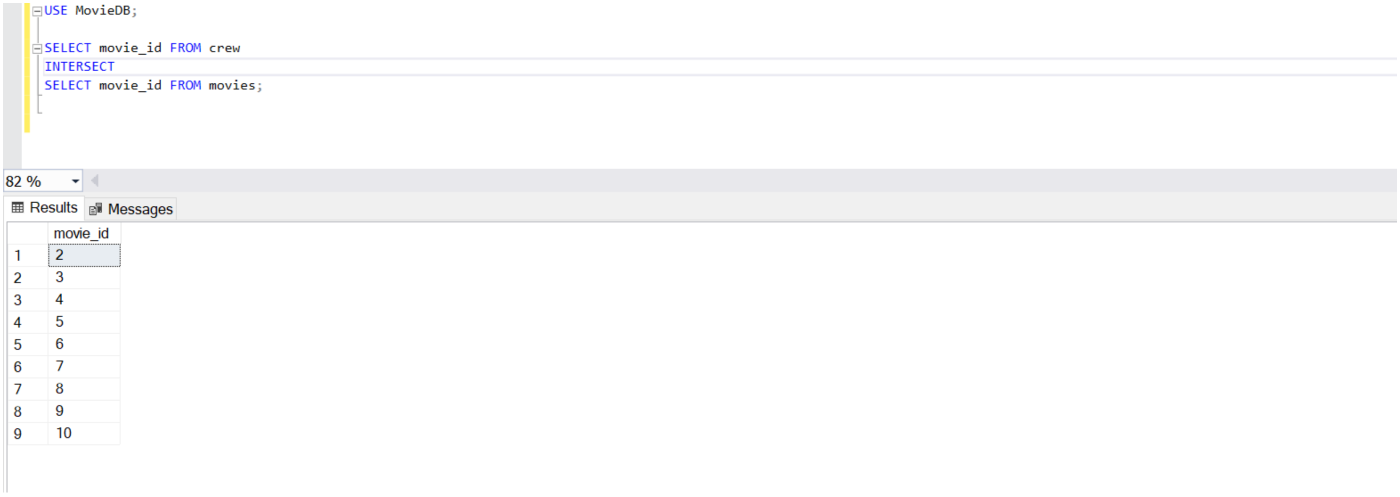
UNION():



```
USE MovieDB;

SELECT movie_id FROM crew
UNION
SELECT movie_id FROM movies;
```

```
    movie_id
1   1
2   2
3   3
4   4
5   5
6   6
7   7
8   8
9   9
10  10
```

INTERSECT():

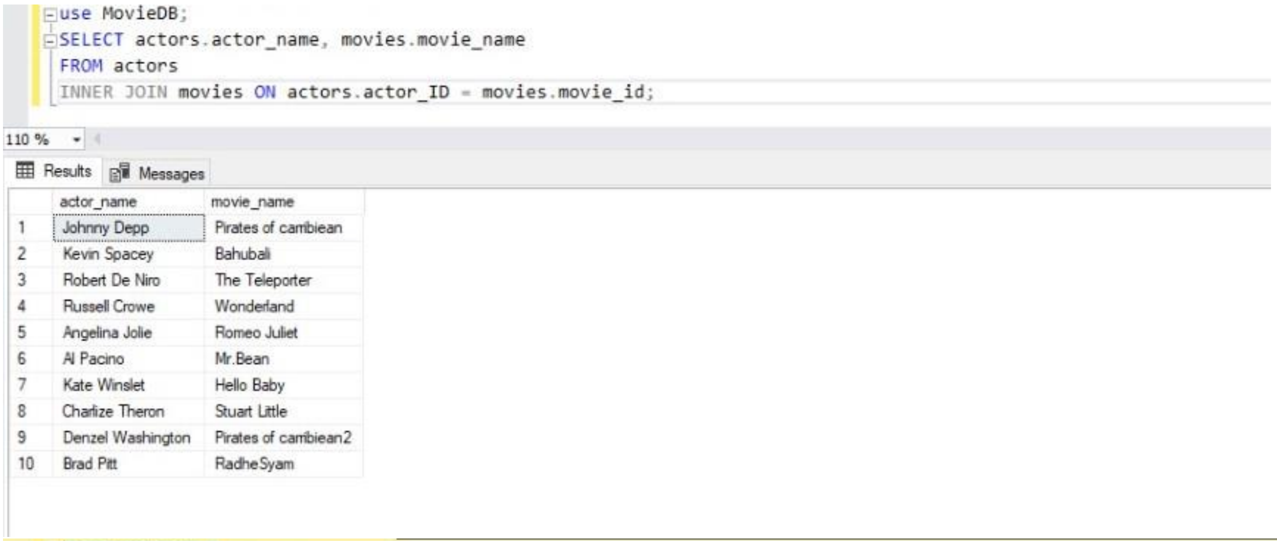


```
USE MovieDB;

SELECT movie_id FROM crew
INTERSECT
SELECT movie_id FROM movies;
```

```
    movie_id
1   2
2   3
3   4
4   5
5   6
6   7
7   8
8   9
9   10
```

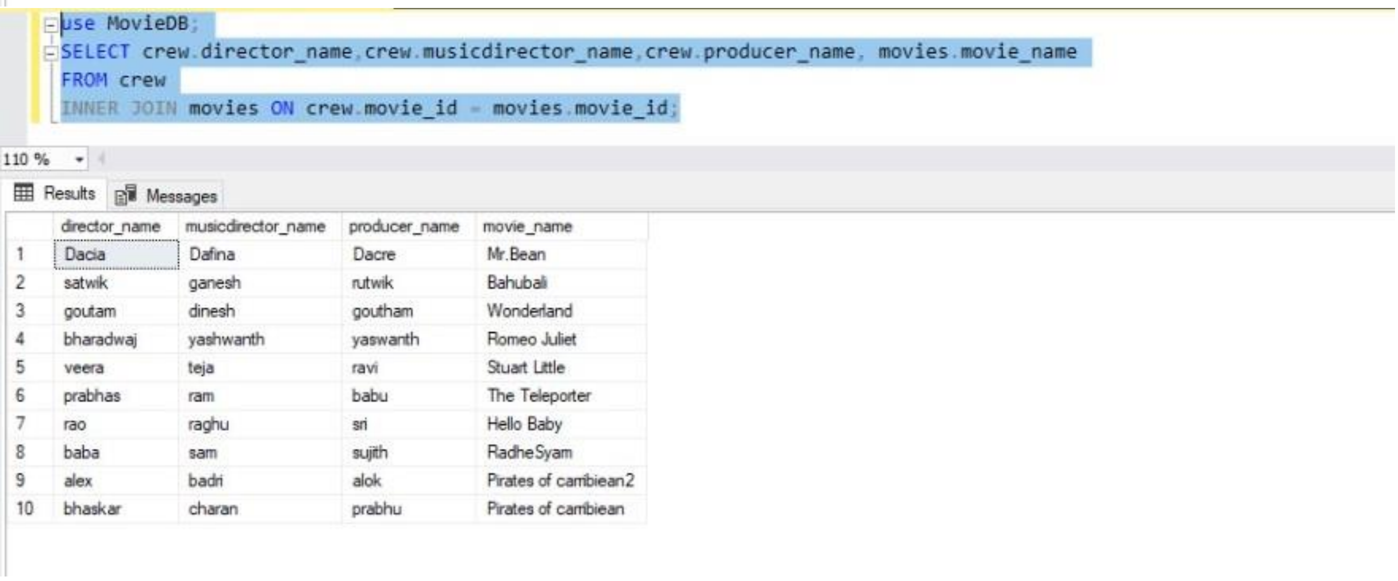
Q7) INNER JOIN, LEFT OUTER JOIN, RIGHT OUTER JOIN- 3 queries for each instance

INNER JOIN



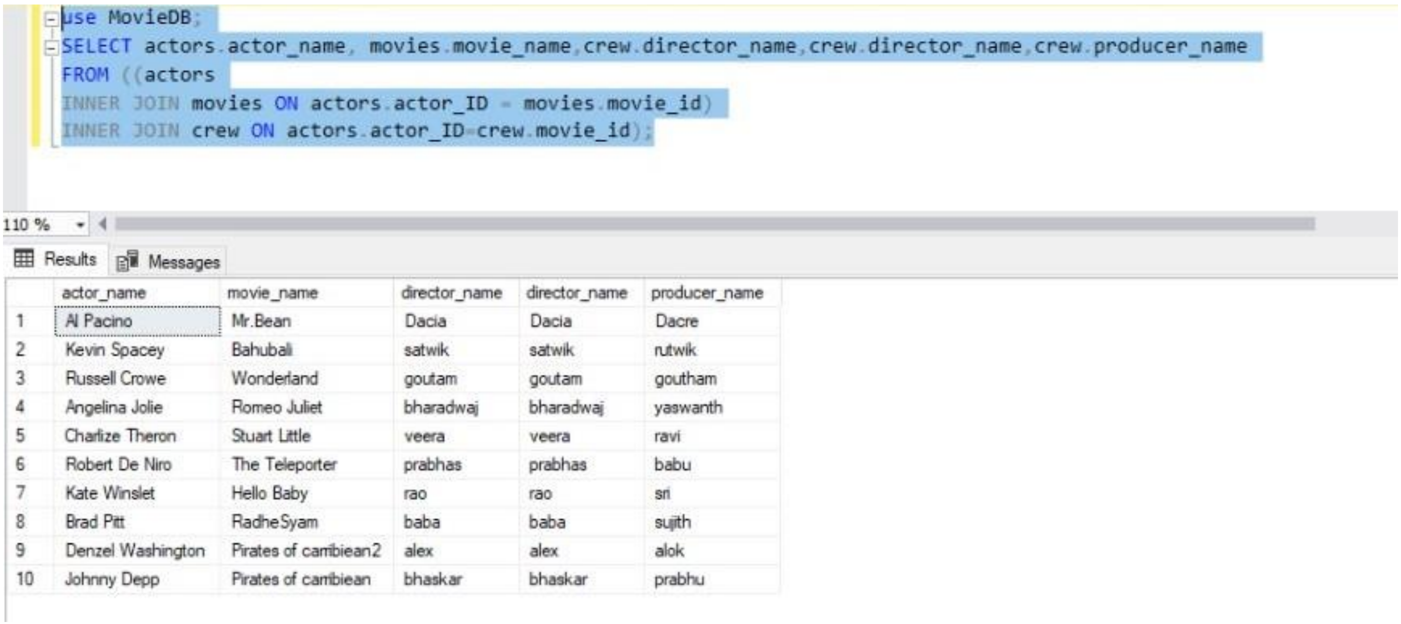
```
use MovieDB;
SELECT actors.actor_name, movies.movie_name
FROM actors
INNER JOIN movies ON actors.actor_ID = movies.movie_id;
```

	actor_name	movie_name
1	Johnny Depp	Pirates of carribiean
2	Kevin Spacey	Bahubali
3	Robert De Niro	The Teleporter
4	Russell Crowe	Wonderland
5	Angelina Jolie	Romeo Juliet
6	Al Pacino	Mr.Bean
7	Kate Winslet	Hello Baby
8	Charlize Theron	Stuart Little
9	Denzel Washington	Pirates of carribiean2
10	Brad Pitt	RadheSyam



```
use MovieDB;
SELECT crew.director_name,crew.musicdirector_name,crew.producer_name, movies.movie_name
FROM crew
INNER JOIN movies ON crew.movie_id = movies.movie_id;
```

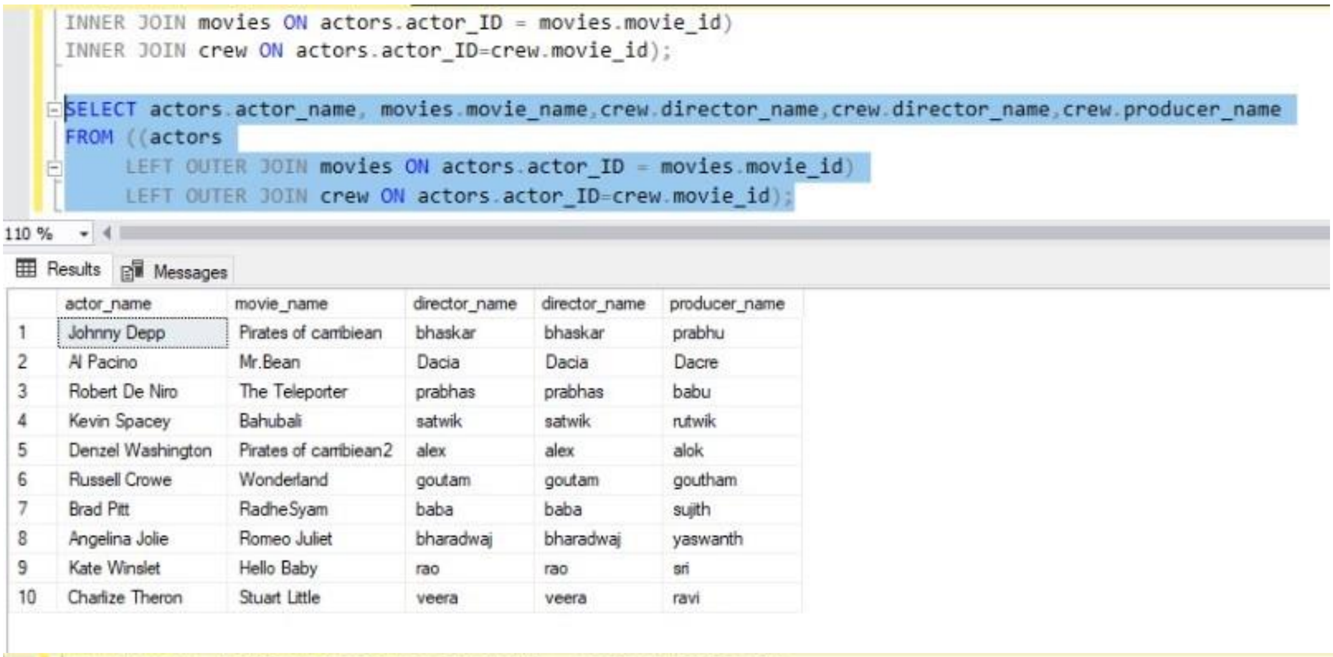
	director_name	musicdirector_name	producer_name	movie_name
1	Dacia	Dafina	Dacre	Mr.Bean
2	satwik	ganesh	rutwik	Bahubali
3	goutam	dinesh	goutham	Wonderland
4	bharadwaj	yashwanth	yaswanth	Romeo Juliet
5	veera	teja	ravi	Stuart Little
6	prabhas	ram	babu	The Teleporter
7	rao	raghu	sri	Hello Baby
8	baba	sam	sujith	RadheSyam
9	alex	badri	alok	Pirates of carribiean2
10	bhaskar	charan	prabhu	Pirates of carribiean



```
use MovieDB;
SELECT actors.actor_name, movies.movie_name,crew.director_name,crew.director_name,crew.producer_name
FROM ((actors
INNER JOIN movies ON actors.actor_ID = movies.movie_id)
INNER JOIN crew ON actors.actor_ID=crew.movie_id);
```

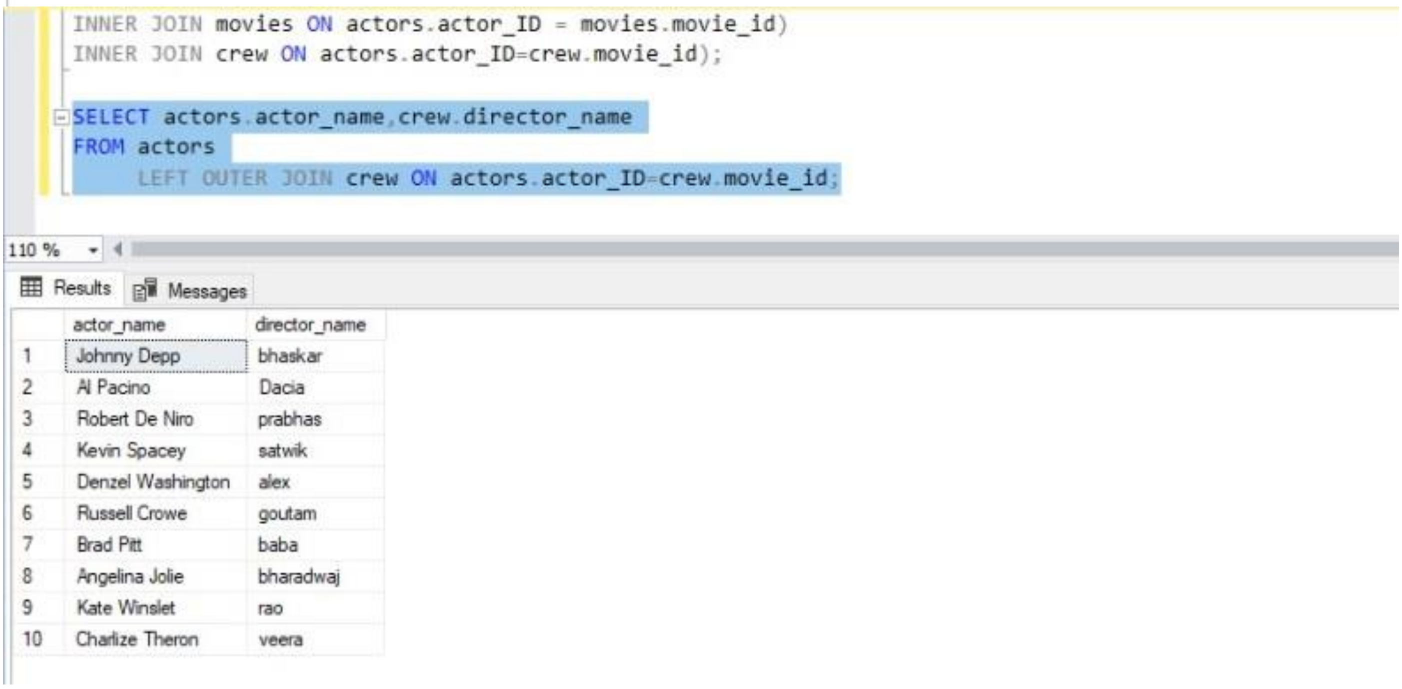
	actor_name	movie_name	director_name	director_name	producer_name
1	Al Pacino	Mr.Bean	Dacia	Dacia	Dacre
2	Kevin Spacey	Bahubali	satwik	satwik	rutwik
3	Russell Crowe	Wonderland	goutam	goutam	goutham
4	Angelina Jolie	Romeo Juliet	bharadwaj	bharadwaj	yaswanth
5	Charlize Theron	Stuart Little	veera	veera	ravi
6	Robert De Niro	The Teleporter	prabhas	prabhas	babu
7	Kate Winslet	Hello Baby	rao	rao	sri
8	Brad Pitt	RadheSyam	baba	baba	sujith
9	Denzel Washington	Pirates of carribiean2	alex	alex	alok
10	Johnny Depp	Pirates of carribiean	bhaskar	bhaskar	prabhu

LEFT OUTER JOIN



```
INNER JOIN movies ON actors.actor_ID = movies.movie_id)
INNER JOIN crew ON actors.actor_ID=crew.movie_id);

SELECT actors.actor_name, movies.movie_name,crew.director_name,crew.director_name,crew.producer_name
FROM ((actors
      LEFT OUTER JOIN movies ON actors.actor_ID = movies.movie_id)
      LEFT OUTER JOIN crew ON actors.actor_ID=crew.movie_id);
```

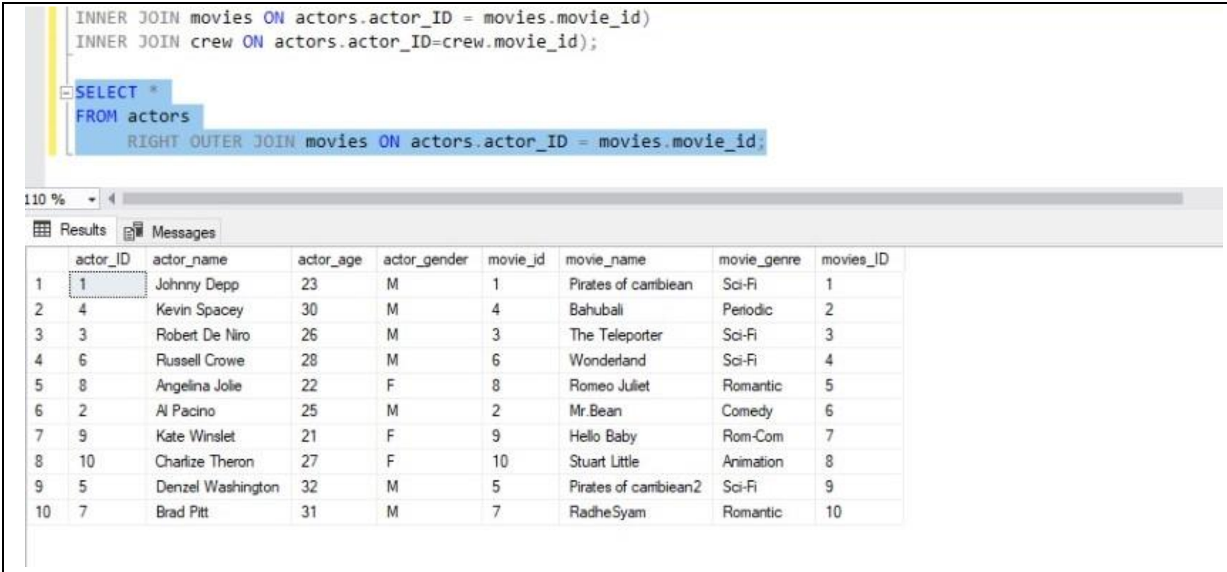
	actor_name	movie_name	director_name	director_name	producer_name
1	Johnny Depp	Pirates of carribiean	bhaskar	bhaskar	prabhu
2	Al Pacino	Mr.Bean	Dacia	Dacia	Dacre
3	Robert De Niro	The Teleporter	prabhas	prabhas	babu
4	Kevin Spacey	Bahubali	satwik	satwik	rutwik
5	Denzel Washington	Pirates of carribiean2	alex	alex	alok
6	Russell Crowe	Wonderland	goutam	goutam	goutham
7	Brad Pitt	RadheSyam	baba	baba	sujith
8	Angelina Jolie	Romeo Juliet	bharadwaj	bharadwaj	yaswanth
9	Kate Winslet	Hello Baby	rao	rao	sri
10	Charlize Theron	Stuart Little	veera	veera	ravi



```
INNER JOIN movies ON actors.actor_ID = movies.movie_id)
INNER JOIN crew ON actors.actor_ID=crew.movie_id);

SELECT actors.actor_name,crew.director_name
FROM actors
      LEFT OUTER JOIN crew ON actors.actor_ID=crew.movie_id;
```

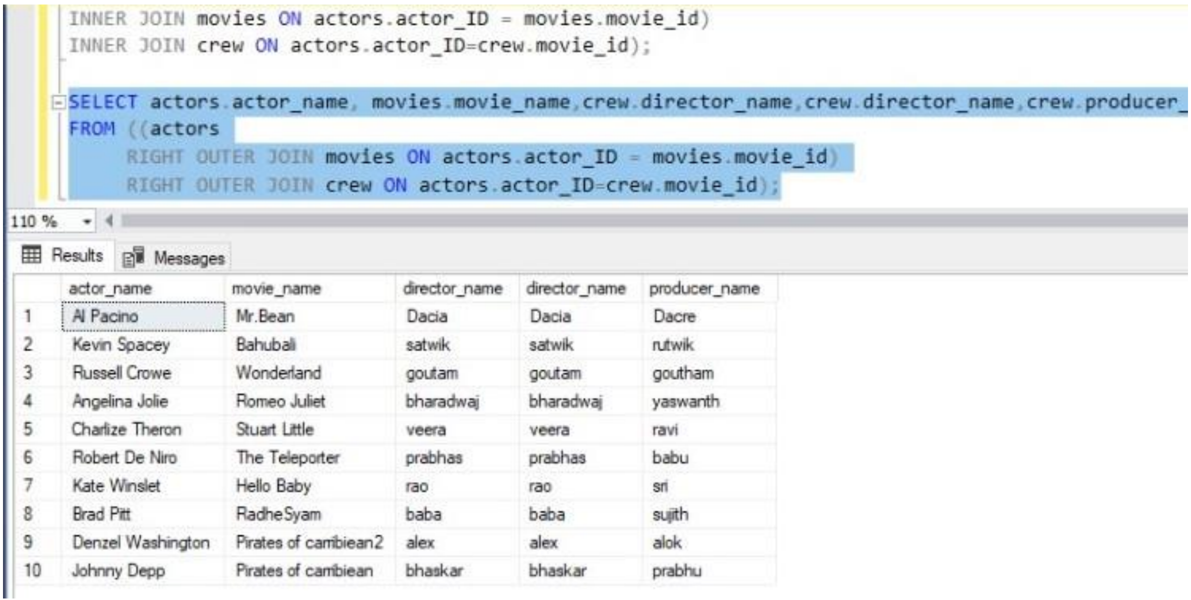
	actor_name	director_name
1	Johnny Depp	bhaskar
2	Al Pacino	Dacia
3	Robert De Niro	prabhas
4	Kevin Spacey	satwik
5	Denzel Washington	alex
6	Russell Crowe	goutam
7	Brad Pitt	baba
8	Angelina Jolie	bharadwaj
9	Kate Winslet	rao
10	Charlize Theron	veera

RIGHT OUTER JOIN



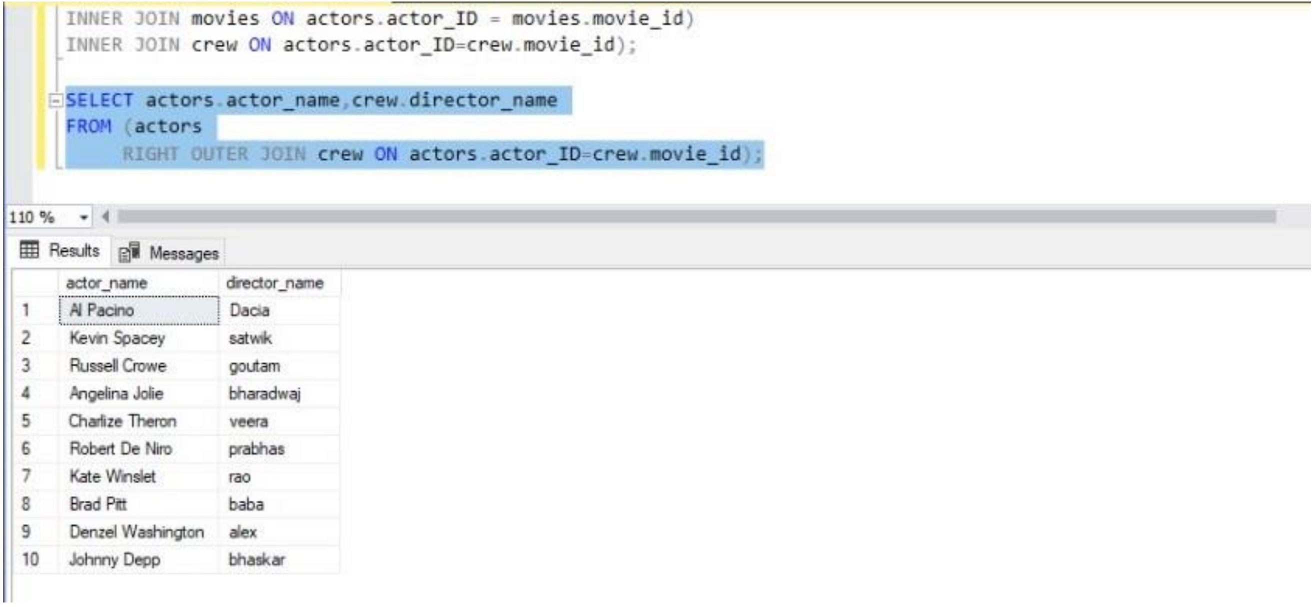
```
INNER JOIN movies ON actors.actor_ID = movies.movie_id)
INNER JOIN crew ON actors.actor_ID=crew.movie_id);

SELECT *
FROM actors
    RIGHT OUTER JOIN movies ON actors.actor_ID = movies.movie_id;
```

	actor_ID	actor_name	actor_age	actor_gender	movie_id	movie_name	movie_genre	movies_ID
1	1	Johnny Depp	23	M	1	Pirates of carribean	Sci-Fi	1
2	4	Kevin Spacey	30	M	4	Bahubali	Periodic	2
3	3	Robert De Niro	26	M	3	The Teleporter	Sci-Fi	3
4	6	Russell Crowe	28	M	6	Wonderland	Sci-Fi	4
5	8	Angelina Jolie	22	F	8	Romeo Juliet	Romantic	5
6	2	Al Pacino	25	M	2	Mr.Bean	Comedy	6
7	9	Kate Winslet	21	F	9	Hello Baby	Rom-Com	7
8	10	Charlize Theron	27	F	10	Stuart Little	Animation	8
9	5	Denzel Washington	32	M	5	Pirates of carribean2	Sci-Fi	9
10	7	Brad Pitt	31	M	7	RadheSyam	Romantic	10



```
INNER JOIN movies ON actors.actor_ID = movies.movie_id)
INNER JOIN crew ON actors.actor_ID=crew.movie_id);

SELECT actors.actor_name, movies.movie_name,crew.director_name,crew.director_name,crew.producer_
FROM ((actors
    RIGHT OUTER JOIN movies ON actors.actor_ID = movies.movie_id)
    RIGHT OUTER JOIN crew ON actors.actor_ID=crew.movie_id);
```

	actor_name	movie_name	director_name	director_name	producer_name
1	Al Pacino	Mr.Bean	Dacia	Dacia	Dacre
2	Kevin Spacey	Bahubali	satwik	satwik	rutwik
3	Russell Crowe	Wonderland	goutam	goutam	goutham
4	Angelina Jolie	Romeo Juliet	bharadwaj	bharadwaj	yaswanth
5	Charlize Theron	Stuart Little	veera	veera	ravi
6	Robert De Niro	The Teleporter	prabhas	prabhas	babu
7	Kate Winslet	Hello Baby	rao	rao	sri
8	Brad Pitt	RadheSyam	baba	baba	sujith
9	Denzel Washington	Pirates of carribean2	alex	alex	alok
10	Johnny Depp	Pirates of carribean	bhaskar	bhaskar	prabhu



```
INNER JOIN movies ON actors.actor_ID = movies.movie_id)
INNER JOIN crew ON actors.actor_ID=crew.movie_id);

SELECT actors.actor_name,crew.director_name
FROM (actors
    RIGHT OUTER JOIN crew ON actors.actor_ID=crew.movie_id);
```

	actor_name	director_name
1	Al Pacino	Dacia
2	Kevin Spacey	satwik
3	Russell Crowe	goutam
4	Angelina Jolie	bharadwaj
5	Charlize Theron	veera
6	Robert De Niro	prabhas
7	Kate Winslet	rao
8	Brad Pitt	baba
9	Denzel Washington	alex
10	Johnny Depp	bhaskar