DBMS LAB ASSIGNMENT - 5

NAME : G.MOHITH KRISHNA
ROLL NO. : 19BCS043

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"

Run | Cancel | Disconnect | Change Connection | Movie_DB | Explain | Enable SQLCMD | Export as Notebook

```
1   select * from actors where actor_age < any(select actor_age from actors where actor_age<28);
2
3   select * from actors where actor_age = any(select actor_age from actors where actor_age=25);
4
5   select * from actors where actor_age > any(select actor_age from actors where actor_age>30);
```

Results | Messages

	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"

Run | Cancel | Disconnect | Change Connection | Movie_DB | Explain | Enable SQLCMD | Export as Notebook

```
1   select actor_name from actors where actor_age <= all(select actor_age from actors where actor_age < 28);
2
3   select * from movies where movie_genre = all(select movie_genre from movies where movie_genre='Periodic');
4
5   select * from movies where movie_name = all(select movie_name from movies where movie_name='');
```

Results Messages

1	Kate Winslet

	actor_ID	movie_name	movie_genre	movies_ID
1	4	Bahubali	Periodic	2

	actor_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 carribiean2	Sci-Fi	9
10	7	RadheSyam	Romantic	10

QUERIES FOR “LIKE”

Run Cancel Disconnect Change Connection Movie_DB

```
1   select * from actors where actor_name like '%Pitt';
2
3   select * from movies where movies_ID like 5;
4
5   select actor_age from actors where actor_gender like 'M';
```

Results Messages

	actor_ID	actor_name	actor_age	actor_gender
1	7	Brad Pitt	31	M

	actor_ID	movie_name	movie_genre	movies_ID
1	8	Romeo Juliet	Romantic	5

	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

Run | Cancel | Disconnect | Change Connection | Movie_DB | Explain | Enable SQL

```
1  select avg(actor_age) as average_age from actors;
2
3  select max(actor_age) as max_age from actors;
4
5  select min(actor_age) as min_age from actors;
6
7  select count(movies_ID) as count from movies where movie_name like '%Little';
8
9  select sum(actor_age) as total_age from actors;
```

Results Messages

	average_age
1	26

	max_age
1	32

	min_age
1	21

	count
1	1

	total_age
1	265

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

ORDER BY

Run | Cancel | Disconnect | Change Connection | Movie_DB

```
1  select * from movies order by movie_name;
2  select * from actors order by actor_age desc;
```

Results | Messages

	actor_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
9	3	The Teleporter	Sci-Fi	3
10	6	Wonderland	Sci-Fi	4

	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

GROUP BY

Run | Cancel | Disconnect | Change Connection | Movie_DB | Explain

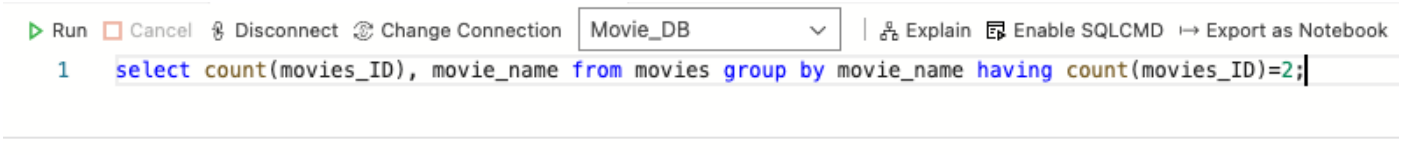
```
1  select actor_age,count(*) as actor_ID from actors group by actor_age;
2  select movie_name,count(*) from movies group by movie_name;
```

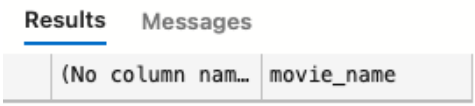
Results | Messages

	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
9	31	1
10	32	1

	movie_name	(No column name)
1	Bahubali	1
2	Hello Baby	1
3	Mr.Bean	1
4	Pirates of carribiean	1
5	Pirates of carribiean2	1
6	RadheSyam	1
7	Romeo Juliet	1
8	Stuart Little	1
9	The Teleporter	1
10	Wonderland	1

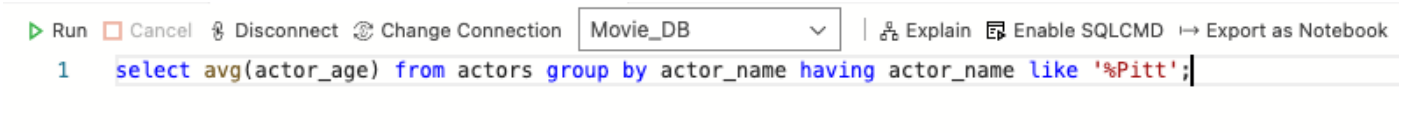
HAVING CLAUSE

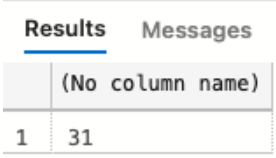




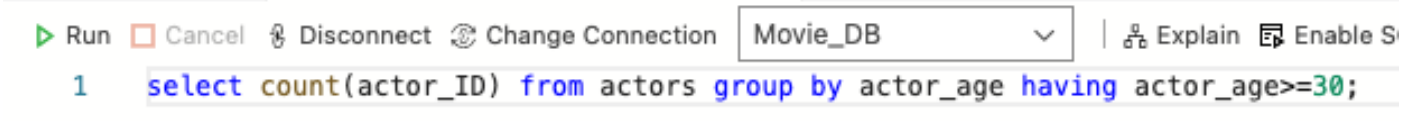
Q4) Use Aggregate function with group by and having

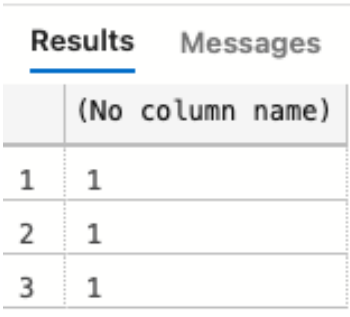
AVG():





COUNT():





MIN():

Run Cancel Disconnect Change Connection Movie_DB Explain Enable SQLCMD

```
1  select min(actor_age) from actors group by actor_gender having actor_gender='F';
```

Results Messages

	(No column name)
1	21

MAX():

Run Cancel Disconnect Change Connection Movie_DB Explain Enable SQLCMD

```
1  select max(actor_age) from actors group by actor_gender having actor_gender='M';
```

Results Messages

	(No column name)
1	32

SUM():

Run Cancel Disconnect Change Connection Movie_DB Explain Enable SQLCMD

```
1  select sum(actor_age) from actors group by actor_ID having actor_ID between 1 and 3;
```

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:

Run | Cancel | Disconnect | Change Connection | Movie_DB | Explain | Enable SQLCMD

```
1   select director_name , count(*) from crew
2   where crew_ID=any(
3                       select crew_ID from crew
4                       where movies_ID=any(
5                                           select movies_ID from movies
6                                           where movie_genre='Periodic'
7                                          )
8                     )
9   group by director_name having director_name like '%%'
10  order by director_name desc;
```

Results | Messages

	director_name	(No column name)
1	Dacia	1

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

EXCEPT():

Run Cancel Disconnect Change Connection Movie_DB

```
1   select director_name from crew
2   EXCEPT
3   select movie_name from movies;
```

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():

Run | Cancel | Disconnect | Change Connection | Movie_DB

```
1   select movies_ID from movies
2   where exists
3   (select movies_ID from crew)
4   order by movies_ID desc;
```

Results | Messages

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

NOT EXISTS():

Run | Cancel | Disconnect | Change Connection | Movie_DB | Explain | Enable SQLCM

```
1   select * from crew
2   where not exists
3   (select movies_ID from movies);
```

Results | Messages

	crew_ID	movies_ID	director_name	producer_name	musicdirector_…

UNION():

Run Cancel Disconnect Change Connection Movie_DB

```
1   select movies_ID from crew
2   union
3   select movies_ID from movies;
```

Results Messages

	movies_ID
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
1…	10

INTERSECT():

Run Cancel Disconnect Change Connection Movie_DB

```
1   select movies_ID from crew
2   intersect
3   select movies_ID from movies;
```

Results Messages

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

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

INNER JOIN

Run | Cancel | Disconnect | Change Connection | Movie_DB | Explain | Enable SQLCMD | Export as Notebook

```
1  select crew.director_name,crew.musicdirector_name,crew.producer_name,movies.movie_name from crew
2  inner join movies on crew.movies_ID=movies.movies_ID;
```

Results | Messages

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

Run | Cancel | Disconnect | Change Connection | Movie_DB

```
1  select actors.actor_name,movies.movie_name from actors
2  inner join movies on actors.actor_ID=movies.movies_ID;
```

Results | Messages

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

Run | Cancel | Disconnect | Change Connection | Movie_DB | Explain | Enable SQLCMD | Export as Notebook

```
1  select actors.actor_name,movies.movie_name,crew.director_name,crew.musicdirector_name,crew.producer_name
2  from ((actors
3  inner join movies on actors.actor_ID=movies.movies_ID)
4  inner join crew on actors.actor_ID=crew.movies_ID);
```

Results | Messages

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

LEFT OUTER JOIN

Run | Cancel | Disconnect | Change Connection | Movie_DB | Explain | Enable SQLCMD | Export as Notebook

```
1  select actors.actor_name,movies.movie_name,crew.director_name,crew.musicdirector_name,crew.producer_name
2  from ((actors
3  left outer join movies on actors.actor_ID=movies.movies_ID)
4  left outer join crew on actors.actor_ID=crew.movies_ID);
```

Results | Messages

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

Run Cancel Disconnect Change Connection Movie_DB

```
1  select actors.actor_name,crew.director_name from actors
2  left outer join crew on actors.actor_ID=crew.movies_ID;
```

Results Messages

	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

Run Cancel Disconnect Change Connection Movie_DB Explain Enable SQLCMD Export as Notebook

```
1  select * from actors
2  right outer join movies on actors.actor_ID=movies.movies_ID;
```

Results Messages

	actor_ID	actor_name	actor_age	actor_gender	actor_ID	movie_name	movie_genre	movies_ID
1	1	Johnny Depp	23	M	1	Pirates of carribiean	Sci-Fi	1
2	2	Al Pacino	25	M	4	Bahubali	Periodic	2
3	3	Robert De Niro	26	M	3	The Teleporter	Sci-Fi	3
4	4	Kevin Spacey	30	M	6	Wonderland	Sci-Fi	4
5	5	Denzel Washington	32	M	8	Romeo Juliet	Romantic	5
6	6	Russell Crowe	28	M	2	Mr.Bean	Comedy	6
7	7	Brad Pitt	31	M	9	Hello Baby	Rom-Com	7
8	8	Angelina Jolie	22	F	10	Stuart Little	Animation	8
9	9	Kate Winslet	21	F	5	Pirates of carribiean2	Sci-Fi	9
10	10	Charlize Theron	27	F	7	RadheSyam	Romantic	10

Run | Cancel | Disconnect | Change Connection | Movie_DB

```
1  select actors.actor_name,crew.director_name from actors
2  right outer join crew on actors.actor_ID=crew.movies_ID;
```

Results | Messages

	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

Run | Cancel | Disconnect | Change Connection | Movie_DB | Explain | Enable SQLCMD | Export as Notebook

```
1  select actors.actor_name,movies.movie_name,crew.director_name,crew.musicdirector_name,crew.producer_name
2  from ((actors
3  right outer join movies on actors.actor_ID=movies.movies_ID)
4  right outer join crew on actors.actor_ID=crew.movies_ID);
5
```

Results | Messages

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