

## DBMS ASSIGNMENT 5

**ROLL NO:-19BCS120**

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?

ANS)

QUERY:-

use iit

```
SELECT * FROM T1_Room
WHERE room_no < ANY (SELECT room_no FROM T1_Room
                     WHERE room_id < 4 );
```

use iit

```
SELECT * FROM T1_Doctor
WHERE s_id < ANY (SELECT s_id FROM T1_Doctor
                 WHERE department = 'AYURVEDIC');
```

use iit

```
SELECT * FROM T1_Room
WHERE room_id < ANY (SELECT room_id FROM T1_Room
                    WHERE room_type = 'special');
```

use iit

```
SELECT * FROM T1_Room
WHERE room_no > ALL (SELECT room_no FROM T1_Room
                    WHERE room_id < 4);
```

use iit

```
SELECT * FROM T1_Doctor
WHERE s_id <> ALL (SELECT s_id FROM T1_Doctor
                 WHERE department = 'AYURVEDIC');
```

use iit

```
SELECT * FROM T1_Room
WHERE room_id <> ALL (SELECT room_id FROM T1_Room
                    WHERE room_type = 'special' );
```

use iit

```
select p_name from T1_patient
where p_name like 'a%'
```

use iit

```
select p_name from T1_patient
where p_name like '%a'
```

use iit

```
select p_name from T1_patient
where p_name like '%no%'
```

## OUTPUT:-

SQLQuery3.sql - loc...J968QT0\jayni (56)

SQLQuery2.sql - loc...J968QT0\jayni (57))

SQLQuery1.sql - loc...J968QT0\jayni (59))\*

100 %

Results

Messages

room_id	patient_id	d_name	room_type	room_no
1	1	rajesh	deluxe	101
2	2	RAMESH	SPECIAL	101

d_name	s_id	department
JEETU	5	AYURVEDIC
MAHESH	4	AYURVEDIC
rajesh	1	AYURVEDIC
RAMESH	2	HOMEOPATHIC

room_id	patient_id	d_name	room_type	room_no
1	1	rajesh	deluxe	101
2	2	RAMESH	SPECIAL	101
3	3	RAMESH	SPECIAL	102
4	4	MAHESH	GENERAL	201
5	5	JEETU	SPECIAL	103
6	6	MAHESH	SPECIAL	104

room_id	patient_id	d_name	room_type	room_no
1	4	MAHESH	GENERAL	201
2	5	JEETU	SPECIAL	103
3	6	MAHESH	SPECIAL	104
4	7	JEETU	SPLIT S...	202
5	8	HITEN	3-SHARI...	301

d_name	s_id	department
RAMESH	2	HOMEOPATHIC

room_id	patient_id	d_name	room_type	room_no
1	1	rajesh	deluxe	101
2	4	MAHESH	GENERAL	201
3	7	JEETU	SPLIT S...	202
4	8	HITEN	3-SHARI...	301

p_name
AMAN

p_name

p_name
VINOD
BINOD

Query executed successfully.

localhost (15.0 RTM)

LAPTOP-0J968QT0\jayni ...

iiit

00:00:00

8 rows

## DIFFERENCE:-

SQLQuery7.sql - loc...J968QT0\jayni (63))

SQLQuery6.sql - loc...J968QT0\jayni (52))

SQLQuery1.sql - loc...J968QT0\jayni (59))\*

```

use iit
SELECT * FROM T1_Room
WHERE room_id < ANY (SELECT room_id FROM T1_Room
                     WHERE room_id < 4 and room_id>1);

SELECT * FROM T1_Room
WHERE room_id < All (SELECT room_id FROM T1_Room
                     WHERE room_id < 4 and room_id>1)

/* in the output we can see that any results in 2 rows as 1<2 and 2<3 but in the case
of all only one row is shown as 2<3 but 2>1 due to which it returns false condition*/

```

100 %

Results

Messages

room_id	patient_id	d_name	room_type	room_no
1	1	rajesh	deluxe	101
2	2	RAMESH	SPECIAL	101

room_id	patient_id	d_name	room_type	room_no
1	1	rajesh	deluxe	101

Query executed successfully.

localhost (15.0 RTM) | LAPTOP-0J968QT0\jayni ... | iit | 00:00:00 | 3 rows

Q2)One query for each Aggregate function?

ANS)

Query:-

use iiit

```
SELECT COUNT(*) As total_patients
FROM T1_patient;
```

```
SELECT SUM(room_no)as sum_of_rooms
FROM T1_room;
```

```
SELECT AVG(room_no)as avg_of_rooms
FROM T1_room;
```

```
SELECT MIN(room_no)as min_of_rooms
FROM T1_room;
```

```
SELECT MAX(room_no)as max_of_rooms
FROM T1_room;
```

The screenshot shows a SQL IDE with five queries in the editor and their results in the Results pane. The queries are:

- `use iiit`
- `SELECT COUNT(*) As total_patients FROM T1_patient;`
- `SELECT SUM(room_no)as sum_of_rooms FROM T1_room;`
- `SELECT AVG(room_no)as avg_of_rooms FROM T1_room;`
- `SELECT MIN(room_no)as min_of_rooms FROM T1_room;`
- `SELECT MAX(room_no)as max_of_rooms FROM T1_room;`

The Results pane shows the following data:

total_patients
20

sum_of_rooms
1627

avg_of_rooms
147

min_of_rooms
101

max_of_rooms
301

At the bottom, a status bar indicates: "Query executed successfully. localhost (15.0 RTM) | LAPTOP-0J968QT0\jayni ... | iiit | 00:00:00 | 5 rows"

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

ANS)

QUERY:-

use iiit

```
select p_name from T1_patient  
where patient_id < 4 order by p_name ASC
```

use iiit

```
select * from T1_room  
where room_id < 5 order by d_name desc
```

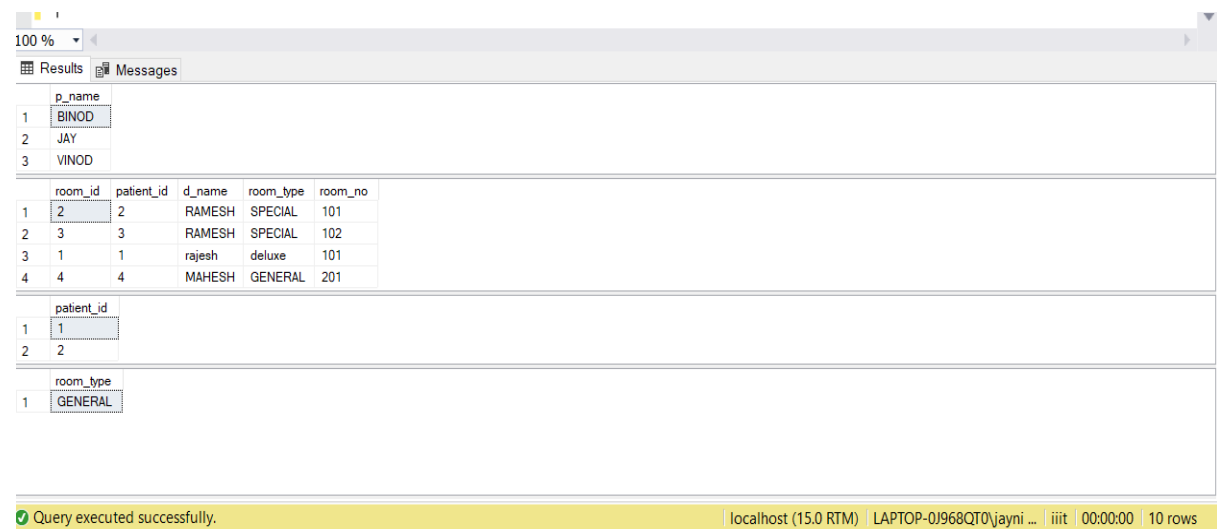
use iiit

```
select patient_id from T1_patient  
group by patient_id having patient_id < 3 ;
```

use iiit

```
select room_type from T1_room  
group by room_type having room_type = 'GENERAL' ;
```

OUTPUT:-



The screenshot shows a database query results window with a toolbar at the top (100% zoom, Results, Messages) and a status bar at the bottom. The status bar indicates 'Query executed successfully.', 'localhost (15.0 RTM)', 'LAPTOP-0J968QT0\jayni ...', 'iiit', '00:00:00', and '10 rows'.

	p_name
1	BINOD
2	JAY
3	VINOD

	room_id	patient_id	d_name	room_type	room_no
1	2	2	RAMESH	SPECIAL	101
2	3	3	RAMESH	SPECIAL	102
3	1	1	rajesh	deluxe	101
4	4	4	MAHESH	GENERAL	201

	patient_id
1	1
2	2

	room_type
1	GENERAL

Q4)Use Aggregate function with group by and having?

ANS)

Query:-

use iit;

```
SELECT max(room_id) as room,room_type FROM T1_room  
GROUP BY room_id ,room_type  
HAVING room_type='general';
```

use iit;

```
SELECT min(room_id) as room,room_type FROM T1_room  
GROUP BY room_id ,room_type  
HAVING room_type='special';
```

use iit;

```
SELECT sum(room_id) as sum_room FROM T1_room  
GROUP BY room_type  
HAVING room_type='special';
```

use iit;

```
SELECT avg(room_id) as avg_room FROM T1_room  
GROUP BY room_type  
HAVING room_type='special';
```

use iit;

```
SELECT count(room_id) as special_rooms FROM T1_room  
GROUP BY room_type  
HAVING room_type='special';
```

The screenshot shows a SQL query editor with five queries and their results. The queries are:

```
HAVING room_type='special';  
use iit;  
SELECT sum(room_id) as sum_room FROM T1_room  
GROUP BY room_type  
HAVING room_type='special';  
use iit;  
SELECT avg(room_id) as avg_room FROM T1_room  
GROUP BY room_type  
HAVING room_type='special';  
use iit;
```

The results are displayed in a table format:

room	room_type
4	GENERAL
11	GENERAL

room	room_type
2	SPECIAL
3	SPECIAL
5	SPECIAL
6	SPECIAL
9	SPECIAL
10	SPECIAL

sum_room
35

avg_room
5

special_rooms
6

Query executed successfully.

localhost (15.0 RTM) | LAPTOP-0J968QT0\jayni ... | iit | 00:00:00 | 11 rows

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

QUERY:-

use iit

select p\_name,patient\_id from T1\_patient

group by p\_name,patient\_id having patient\_id<5 order by p\_name asc

use iit

select room\_id,room\_type,d\_name from T1\_room

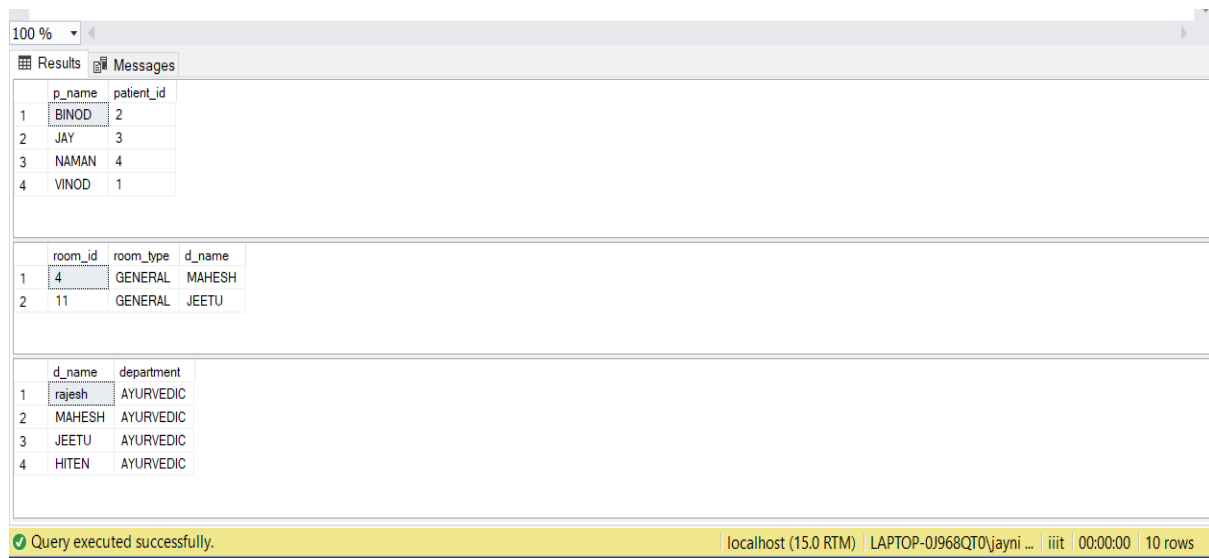
group by room\_id,room\_type,d\_name having room\_type='general' order by d\_name desc

use iit

select d\_name,department from T1\_doctor

group by d\_name,department having department='ayurvedic' order by d\_name desc

OUTPUT:-



The screenshot shows a database query results window with a zoom level of 100%. It displays three tables of data. The first table has columns p\_name and patient\_id. The second table has columns room\_id, room\_type, and d\_name. The third table has columns d\_name and department. A status bar at the bottom indicates the query was executed successfully.

	p_name	patient_id
1	BINOD	2
2	JAY	3
3	NAMAN	4
4	VINOD	1

	room_id	room_type	d_name
1	4	GENERAL	MAHESH
2	11	GENERAL	JEETU

	d_name	department
1	rajesh	AYURVEDIC
2	MAHESH	AYURVEDIC
3	JEETU	AYURVEDIC
4	HITEN	AYURVEDIC

Query executed successfully. localhost (15.0 RTM) | LAPTOP-0J968QT0\jayni ... | iit | 00:00:00 | 10 rows

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

QUERY:-

```
use iit
select patient_id from T1_patient
except
select patient_id from T1_room
```

```
use iit
select * from T1_room
where exists(select patient_id from T1_patient where patient_id <5 and
T1_room.patient_id=T1_patient.patient_id)
```

```
use iit
select * from T1_room
where not exists(select patient_id from T1_patient where patient_id <5 and
T1_room.patient_id=T1_patient.patient_id)
```

```
use iit
select d_name from T1_room
union
select d_name from T1_doctor
```

```
use iit
select d_name from T1_room
intersect
select d_name from T1_doctor
```

## OUTPUT:-

SQLQuery2.sql - loc...J968QT0\jayni (57)

SQLQuery5.sql - loc...J968QT0\jayni (68)

SQLQuery1.sql - loc...J968QT0\jayni (59)\*

100 %

Results Messages

	patient_id
1	12
2	13
3	14
4	15
5	16
6	17
7	18
8	19

	room_id	patient_id	d_name	room_type	room_no
1	1	1	rajesh	deluxe	101
2	2	2	RAMESH	SPECIAL	101
3	3	3	RAMESH	SPECIAL	102
4	4	4	MAHESH	GENERAL	201

	room_id	patient_id	d_name	room_type	room_no
1	5	5	JEETU	SPECIAL	103
2	6	6	MAHESH	SPECIAL	104
3	7	7	JEETU	SPLIT S...	202
4	8	8	HITEN	3-SHARI...	301
5	9	9	RAMESH	SPECIAL	105
6	10	10	MAHESH	SPECIAL	106
7	11	11	JEETU	GENERAL	201

	d_name
1	HITEN
2	JEETU
3	MAHE...
4	rajesh
5	RAME...

	d_name
1	HITEN
2	JEETU
3	MAHE...
4	rajesh
5	RAME...

Query executed successfully.

localhost (15.0 RTM) | LAPTOP-0J968QT0\jayni ... | iiit | 00:00:00 | 30 rows



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

ANS)

INNER JOIN:-

The screenshot shows a SQL Developer window with three tabs: SQLQuery2.sql, SQLQuery5.sql, and SQLQuery1.sql. The active tab is SQLQuery1.sql, which contains three SQL queries. The first query is an INNER JOIN between T1\_room and T1\_patient on patient\_id. The second query is an INNER JOIN between T1\_room and T1\_patient on room\_type='general' AND patient\_id. The third query is an INNER JOIN between T1\_room and T1\_patient on room\_type='special' AND patient\_id. The Results pane shows three tables of results. The first table has 8 rows, the second has 2 rows, and the third has 6 rows. The status bar at the bottom indicates 'Query executed successfully.' and 'localhost (15.0 RTM) | LAPTOP-0J968QT0\jayni ... | iiit | 00:00:00 | 19 rows'.

```
use iiit
select T1_room.d_name, T1_patient.p_name
from T1_room
inner join T1_patient on T1_room.patient_id=T1_patient.patient_id

select T1_room.d_name, T1_patient.p_name
from T1_room
inner join T1_patient on T1_room.room_type='general' AND T1_room.patient_id=T1_patient.patient_id

select T1_room.d_name, T1_patient.p_name
from T1_room
inner join T1_patient on T1_room.room_type='special' AND T1_room.patient_id=T1_patient.patient_id
```

d_name	p_name
rajesh	VINOD
RAMESH	BINOD
RAMESH	JAY
MAHESH	NAMAN
JEETU	AMAN
MAHESH	VIRAT
JEETU	MANAN
HITEN	MANAV

d_name	p_name
MAHESH	NAMAN
JEETU	OM

d_name	p_name
RAMESH	BINOD
RAMESH	JAY
JEETU	AMAN
MAHESH	VIRAT
RAMESH	MANAV
MAHESH	JAYESH

LEFT OUTER JOIN:-

The screenshot shows a SQL Developer window with three tabs: SQLQuery2.sql, SQLQuery5.sql, and SQLQuery1.sql. The active tab is SQLQuery1.sql, which contains three SQL queries. The first query is a LEFT OUTER JOIN between T1\_room and T1\_patient on room\_type='general' AND patient\_id. The second query is a LEFT OUTER JOIN between T1\_room and T1\_patient on room\_type='special' AND patient\_id. The third query is a LEFT OUTER JOIN between T1\_room and T1\_patient on patient\_id. The Results pane shows three tables of results. The first table has 8 rows, the second has 8 rows, and the third has 8 rows. The status bar at the bottom indicates 'Query executed successfully.' and 'localhost (15.0 RTM) | LAPTOP-0J968QT0\jayni ... | iiit | 00:00:00 | 33 rows'.

```
select T1_room.d_name, T1_patient.p_name
from T1_room
left join T1_patient on T1_room.room_type='general' AND T1_room.patient_id=T1_patient.patient_id

select T1_room.d_name, T1_patient.p_name
from T1_room
left join T1_patient on T1_room.room_type='special' AND T1_room.patient_id=T1_patient.patient_id

select T1_room.d_name, T1_patient.p_name
from T1_room
left join T1_patient on T1_room.patient_id=T1_patient.patient_id
```

d_name	p_name
rajesh	NULL
RAMESH	NULL
RAMESH	NULL
MAHESH	NAMAN
JEETU	NULL
MAHESH	NULL
JEETU	NULL
HITEN	NULL

d_name	p_name
rajesh	NULL
RAMESH	BINOD
RAMESH	JAY
MAHESH	NULL
JEETU	AMAN
MAHESH	VIRAT
JEETU	NULL
HITEN	NULL

d_name	p_name
rajesh	VINOD
RAMESH	BINOD
RAMESH	JAY
MAHESH	NAMAN
JEETU	AMAN

## RIGHT OUTER JOIN:-

The screenshot displays a SQL IDE with three queries in the editor and their corresponding results in the Results pane.

**Query 1:**

```
select T1_room.d_name,T1_patient.p_name
from T1_room
right join T1_patient on T1_room.room_type='special' AND T1_room.patient_id=T1_patient.patient_id
```

**Query 2:**

```
select T1_room.d_name,T1_patient.p_name
from T1_room
right join T1_patient on T1_room.patient_id=T1_patient.patient_id
```

**Query 3:**

```
select T1_room.d_name,T1_patient.p_name
from T1_room
right join T1_patient on T1_room.room_type='general' AND T1_room.patient_id=T1_patient.patient_id
```

**Results:**

The Results pane shows three tables of results, each with columns `d_name` and `p_name`.

**Table 1 (Query 1):**

d_name	p_name
NULL	VINOD
RAMESH	BINOD
RAMESH	JAY
NULL	NAMAN
JEETU	AMAN
MAHESH	VIRAT
NULL	MANAN
NULL	MANAV

**Table 2 (Query 2):**

d_name	p_name
rajesh	VINOD
RAMESH	BINOD
RAMESH	JAY
MAHESH	NAMAN
JEETU	AMAN
MAHESH	VIRAT
JEETU	MANAN
HITEN	MANAV

**Table 3 (Query 3):**

d_name	p_name
NULL	VINOD
NULL	BINOD
NULL	JAY
MAHE...	NAMAN
NULL	MANAN

Query executed successfully. | localhost (15.0 RTM) | LAPTOP-0J968QT0\jayni ... | iiit | 00:00:00 | 60 rows

Q8) Use all the above condition in JOIN as well?

ANS)

QUERY:-

```
select T1_room.d_name, T1_patient.p_name
from T1_room
join T1_patient on T1_room.room_type='special' AND
T1_room.patient_id=T1_patient.patient_id
```

```
select T1_room.d_name, T1_patient.p_name
from T1_room
join T1_patient on T1_room.patient_id=T1_patient.patient_id
```

```
select T1_room.d_name, T1_patient.p_name
from T1_room
join T1_patient on T1_room.room_type='general' AND
T1_room.patient_id=T1_patient.patient_id
```

JOIN:-

The screenshot shows the SQL Developer interface with three tabs: SQLQuery2.sql, SQLQuery5.sql, and SQLQuery1.sql. The SQLQuery1.sql tab is active, displaying three SQL queries. The first query is a JOIN query filtering for 'special' room type. The second query is a simple JOIN query. The third query is a JOIN query filtering for 'general' room type. Below the queries, the Results pane shows three result sets. The first result set has 6 rows. The second result set has 9 rows. The third result set has 2 rows. The status bar at the bottom indicates 'Query executed successfully.' and 'localhost (15.0 RTM) | LAPTOP-0J968QT0\jayni ... | 00:00:00 | 19 rows'.

```
from T1_room
join T1_patient on T1_room.room_type='special' AND T1_room.patient_id=T1_patient.patient_id

select T1_room.d_name, T1_patient.p_name
from T1_room
join T1_patient on T1_room.patient_id=T1_patient.patient_id

select T1_room.d_name, T1_patient.p_name
from T1_room
join T1_patient on T1_room.room_type='general' AND T1_room.patient_id=T1_patient.patient_id
```

d_name	p_name
1	RAMESH BINOD
2	RAMESH JAY
3	JEETU AMAN
4	MAHESH VIRAT
5	RAMESH MANAV
6	MAHESH JAYESH

d_name	p_name
1	rajesh VINOD
2	RAMESH BINOD
3	RAMESH JAY
4	MAHESH NAMAN
5	JEETU AMAN
6	MAHESH VIRAT
7	JEETU MANAN
8	HITEN MANAV
9	RAMESH MANAV

d_name	p_name
1	MAHESH NAMAN
2	JEETU OM

Query executed successfully. | localhost (15.0 RTM) | LAPTOP-0J968QT0\jayni ... | 00:00:00 | 19 rows

