

DBMS

ASSIGNMENT: 5

NAME: HENI PRAJAPATI
REG. NO.: 19BCS119

1. 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.

Answer:

a. ANY

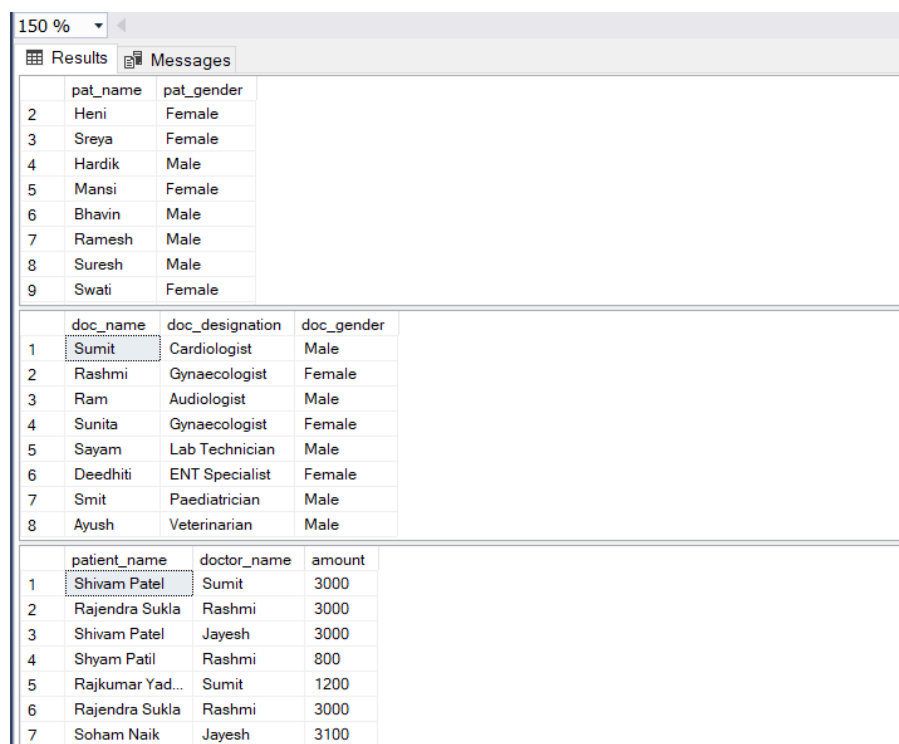
Query:

```
SELECT pat_name, pat_gender  
FROM t1_patients WHERE  
pat_id > ANY (SELECT doc_id FROM t1_doctor  
WHERE doc_id > 5);
```

```
SELECT doc_name, doc_designation, doc_gender  
FROM t1_doctor WHERE  
doc_id > ANY (SELECT pat_id FROM t1_patients  
WHERE pat_gender = 'Female');
```

```
SELECT patient_name, doctor_name, amount  
FROM t1_bill WHERE  
pat_id > ANY (SELECT doc_id FROM t1_doctor  
WHERE doc_id > 5);
```

Output:



The screenshot shows a database query results window with a zoom level of 150%. It displays three tables: t1_patients, t1_doctor, and t1_bill. The t1_patients table has 9 rows, t1_doctor has 8 rows, and t1_bill has 7 rows. The results are shown in a grid format with column headers and data rows.

pat_name	pat_gender
Heni	Female
Sreya	Female
Hardik	Male
Mansi	Female
Bhavin	Male
Ramesh	Male
Suresh	Male
Swati	Female

doc_name	doc_designation	doc_gender
Sumit	Cardiologist	Male
Rashmi	Gynaecologist	Female
Ram	Audiologist	Male
Sunita	Gynaecologist	Female
Sayam	Lab Technician	Male
Deedhiti	ENT Specialist	Female
Smit	Paediatrician	Male
Ayush	Veterinarian	Male

patient_name	doctor_name	amount
Shivam Patel	Sumit	3000
Rajendra Sukla	Rashmi	3000
Shivam Patel	Jayesh	3000
Shyam Patil	Rashmi	800
Rajkumar Yad...	Sumit	1200
Rajendra Sukla	Rashmi	3000
Soham Naik	Jayesh	3100

b. ALL

Query:

```
SELECT pat_name, pat_gender
FROM t1_patients WHERE
pat_id > ALL (SELECT doc_id FROM t1_doctor
WHERE doc_id < 5);
```

```
SELECT doc_name, doc_designation, doc_gender
FROM t1_doctor WHERE
doc_id <> ALL (SELECT pat_id FROM t1_patients
WHERE pat_gender = 'Female');
```

```
SELECT patient_name, doctor_name, amount
FROM t1_bill WHERE
pat_id > ALL (SELECT doc_id FROM t1_doctor
WHERE doc_id = 7);
```

Output:

150 %			
Results Messages			
	pat_name	pat_gender	
1	Raj	Male	
2	Ramesh	Male	
3	Zoya	Female	
4	Henri	Female	
5	Sreya	Female	
6	Hardik	Male	
7	Mansi	Female	
8	Bhavin	Male	

	doc_name	doc_designation	doc_gender
1	Sumit	Gynaecologist	Male
2	Rashmi	Cardiologist	Female
3	Rashmi	Gynaecologist	Female
4	Ram	Audiologist	Male
5	Smit	Paediatrician	Male

	patient_name	doctor_name	amount
1	Shivam Patel	Sumit	3000
2	Rajendra Sukla	Rashmi	3000
3	Shivam Patel	Jayesh	3000
4	Shyam Patil	Rashmi	800
5	Rajkumar Yad...	Sumit	1200
6	Rajendra Sukla	Rashmi	3000
7	Soham Naik	Jayesh	3100

c. LIKE

Query:

```
select pat_name from t1_patients
where pat_name like '%a%';
```

```
select doc_id, doc_name from t1_doctor
where doc_name like '%a';
```

```
select staff_name, staff_gender from t1_staff
where staff_gender like 'Fe%';
```

OUTPUT:

150 %	
Results	Messages
pat_name	
1	Ramesh
2	Swati
3	Pooja
4	Raj
5	Ramesh
6	Zoya
7	Sreya
8	Hardik
9	Mansi
doc_id	doc_name
1	7
	Sunita
staff_name	staff_gender
1	Himani
2	Suhani
3	Priya
	Female
	Female
	Female

2. One query for each Aggregate function.

a. Average

Query:

```
SELECT AVG(doc_id) AS 'Average'
FROM t1_doctor;
```

Output:

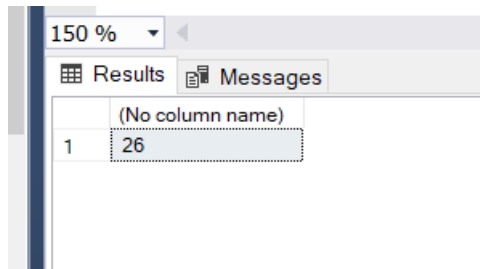
150 %	
Results	Messages
Average	
1	6

b. Count

Query:

```
SELECT COUNT(*)
FROM t1_patients
WHERE pat_id>10;
```

Output:



A screenshot of a database query result window. The window has a zoom level of 150% and two tabs: 'Results' and 'Messages'. The 'Results' tab is active, showing a table with one column labeled '(No column name)' and one row with the value 26.

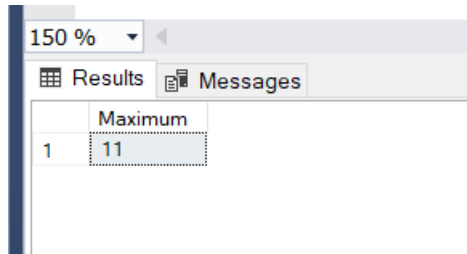
(No column name)
26

c. Max

Query:

```
SELECT MAX(doc_id)
AS 'Maximum'
From t1_doctor;
```

Output:



A screenshot of a database query result window. The window has a zoom level of 150% and two tabs: 'Results' and 'Messages'. The 'Results' tab is active, showing a table with one column labeled 'Maximum' and one row with the value 11.

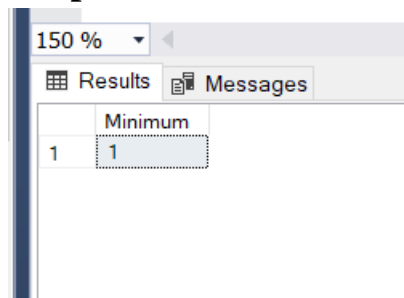
Maximum
11

d. Min

Query:

```
SELECT MIN(staff_id)
as 'Minimum'
From t1_staff;
```

Output:



A screenshot of a database query result window. The window has a zoom level of 150% and two tabs: 'Results' and 'Messages'. The 'Results' tab is active, showing a table with one column labeled 'Minimum' and one row with the value 1.

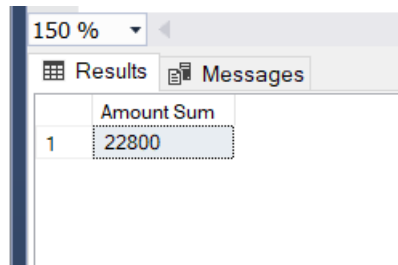
Minimum
1

e. Sum

Query:

```
SELECT SUM(amount)
AS 'Amount Sum'
From t1_bill;
```

Output:



Amount Sum	
1	22800

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

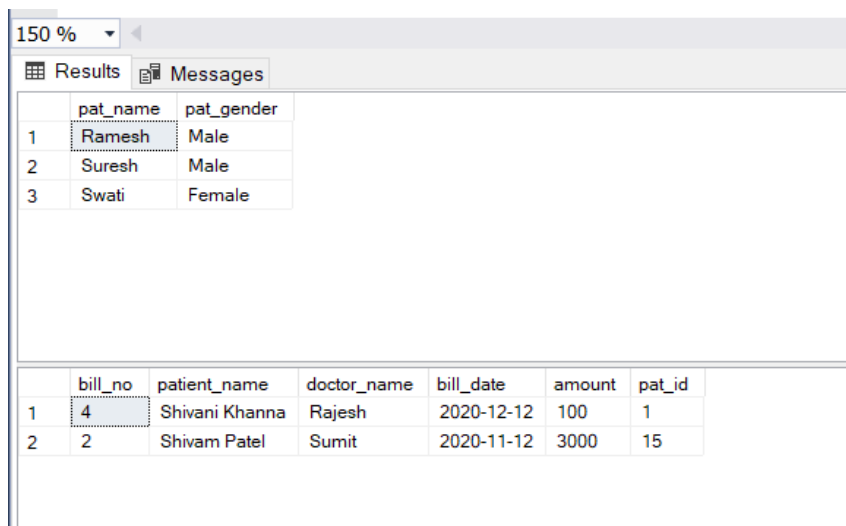
a. Order by

Query:

```
SELECT pat_name, pat_gender FROM t1_patients  
WHERE pat_id < 4 ORDER BY pat_name ASC
```

```
SELECT * FROM t1_bill  
WHERE bill_no < 5 ORDER BY patient_name DESC
```

Output:



pat_name	pat_gender
1 Ramesh	Male
2 Suresh	Male
3 Swati	Female

bill_no	patient_name	doctor_name	bill_date	amount	pat_id
1 4	Shivani Khanna	Rajesh	2020-12-12	100	1
2 2	Shivam Patel	Sumit	2020-11-12	3000	15

b. GROUP BY

Query:

```
SELECT doc_name  
FROM t1_doctor  
GROUP BY doc_name;
```

```
SELECT patient_name  
FROM t1_bill  
GROUP BY patient_name;
```

Output:

150 %

Results Messages

	doc_name
1	Ayush
2	Deedhiti
3	Jayesh
4	Ram
5	Rashmi
6	Sayam
7	Smit

	patient_name
1	Heni Prajapati
2	Rajendra Sukla
3	Rajkumar Yadav
4	Shivam Patel
5	Shivani Khanna
6	Shyam Patil
7	Sima Patel
8	Soham Naik

✓ Query executed successfully.

c. HAVING

Query:

```
SELECT COUNT(diag_no) AS 'Count', diag_details
FROM t1_patient_diagnosis
GROUP BY diag_details
HAVING COUNT(pat_id) > 0;
```

```
SELECT staff_gender
FROM t1_staff
GROUP BY staff_gender
HAVING staff_gender = 'Male';
```

Output:

150 %

Results Messages Execution plan

	Count	diag_details
1	1	Covid-19
2	1	Food Infection
3	1	Indigestion
4	1	Thyroid fever

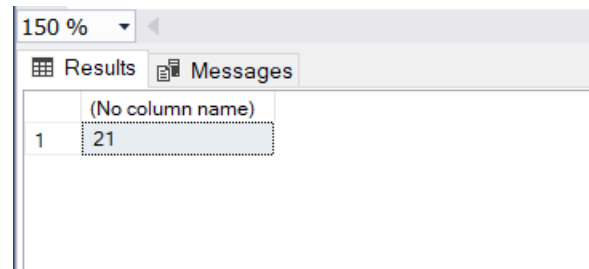
	staff_gender
1	Male

4. Use Aggregate function with group by and having

a. Query:

```
SELECT AVG(pat_id) FROM t1_patients  
GROUP BY pat_gender  
HAVING pat_gender='Female';
```

Output:



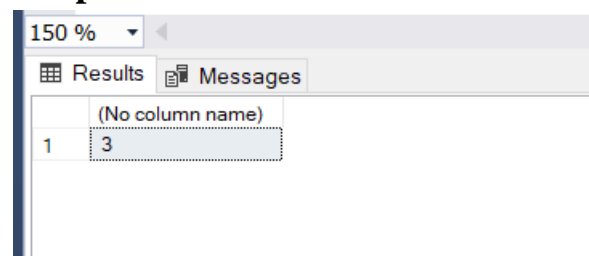
The screenshot shows a SQL query results window with a zoom level of 150%. The window has two tabs: 'Results' and 'Messages'. The 'Results' tab is active, displaying a single row of data. The column header is '(No column name)' and the value is 21.

	(No column name)
1	21

b. Query:

```
SELECT count(doc_id) FROM t1_doctor  
GROUP BY doc_designation  
HAVING doc_designation='Gynaecologist';
```

Output:



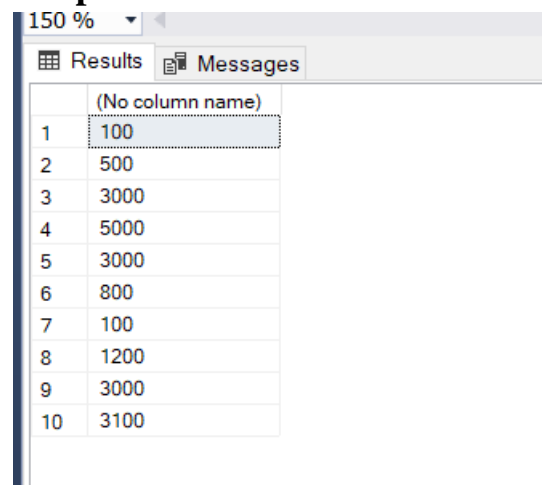
The screenshot shows a SQL query results window with a zoom level of 150%. The window has two tabs: 'Results' and 'Messages'. The 'Results' tab is active, displaying a single row of data. The column header is '(No column name)' and the value is 3.

	(No column name)
1	3

c. Query:

```
SELECT max(amount) FROM t1_bill  
GROUP BY bill_no  
HAVING bill_no > 3;
```

Output:



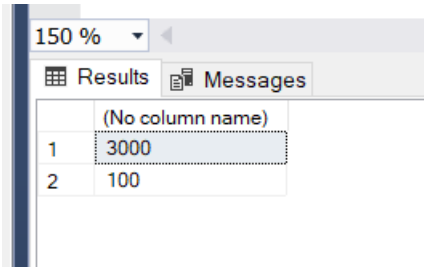
The screenshot shows a SQL query results window with a zoom level of 150%. The window has two tabs: 'Results' and 'Messages'. The 'Results' tab is active, displaying 10 rows of data. The column header is '(No column name)'. The values are 100, 500, 3000, 5000, 3000, 800, 100, 1200, 3000, and 3100.

	(No column name)
1	100
2	500
3	3000
4	5000
5	3000
6	800
7	100
8	1200
9	3000
10	3100

d. Query:

```
SELECT min(amount) FROM t1_bill  
GROUP BY bill_no  
HAVING bill_no < 10;
```

Output:

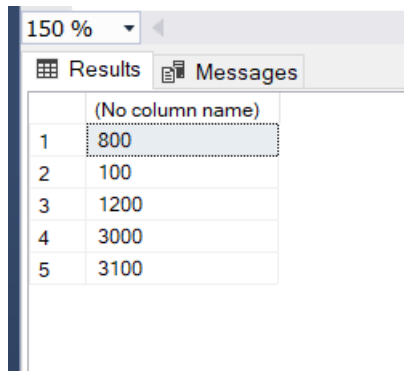


	(No column name)
1	3000
2	100

e. Query:

```
SELECT sum(amount) FROM t1_bill  
GROUP BY bill_no  
HAVING bill_no > 15;
```

Output:



	(No column name)
1	800
2	100
3	1200
4	3000
5	3100

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

Query:

```
select pat_name,pat_id from t1_patients  
group by pat_name,pat_id having pat_id < 5  
order by pat_name asc ;
```

```
select staff_id, staff_name from t1_staff  
group by staff_id, staff_name having staff_id > 2  
order by staff_id desc;
```

```
select doc_name,doc_designation from t1_doctor  
group by doc_name,doc_designation having doc_designation='Gynaecologist'  
order by doc_name desc;
```

Output:

150 %

Results Messages

	pat_name	pat_id
1	Pooja	4
2	Ramesh	1
3	Suresh	2
4	Swati	3

	staff_id	staff_name
1	7	Piyush
2	6	Rajesh
3	5	Priya
4	4	Suhani
5	3	Raja

	doc_name	doc_designation
1	Sunita	Gynaecologist
2	Sumit	Gynaecologist
3	Rashmi	Gynaecologist

6. Illustrate the Usage of Except, Exists, Not Exists, Union, Intersection

Query:

```
select doc_id from t1_doctor
except select pat_id
from t1_patient_diagnosis
```

```
select * from t1_doctor
where exists(select doc_id
from t1_patients
where doc_id <5 and t1_doctor.doc_id=t1_patients.doc_id)
```

```
select * from t1_doctor
where not exists(select doc_id
from t1_patients
where doc_id <5 and t1_doctor.doc_id=t1_patients.doc_id)
```

```
select staff_name from t1_staff
union select doc_name
from t1_doctor
```

```
select pat_name from t1_patients
intersect select staff_name
from t1_staff;
```

Output:

150 %	Results	Messages
doc_id		
1	1	
2	3	
3	4	
4	6	
5	7	
6	8	
7	9	
8	10	
doc_id	doc_name	doc_gender
1	Sumit	Male
2	Rashmi	Female
3	Jayesh	Male
doc_designation	doc_phone	doc_address
Gynaecologist	9876567821	13 jusddx 2swhv
Cardiologist	8751537485	67 dhjs dsdsj
Dentist	9898760912	jeak aukh wdj
doc_id	doc_name	doc_gender
4	Sumit	Male
5	Rashmi	Female
6	Ram	Male
7	Sunita	Female
8	Sayam	Male
9	Deedhiti	Female
10	Smit	Male
11	Ayush	Male
doc_designation	doc_phone	doc_address
Cardiologist	9876567821	13 jusddx 2swhv
Gynaecologist	8751537485	67 dhjs dsdsj
Audiologist	9898760912	jeak aukh wdj
Gynaecologist	9966715243	Surat
Lab Technician	9966715243	Ahmedabad
ENT Specialist	8810237512	Rajkot
Paediatrician	8809761256	Mumbai
Veterinarian	9887760987	Pune
staff_name		
9	Rajesh	
10	Ram	
11	Rashmi	
12	Sayam	
13	Smit	
14	Suhani	
15	Sumit	
16	Sunita	
pat_name		

7. INNER JOIN, LEFT OUTER JOIN, RIGHT OUTER JOIN- 3 queries for each instance

a. INNER JOIN

Query:

```
select t1_doctor.doc_name,t1_patients.pat_name
from t1_patients
inner join t1_doctor on t1_doctor.doc_id=t1_patients.doc_id;
```

```
select t1_doctor.doc_name,t1_staff.staff_name
from t1_staff
inner join t1_doctor on t1_doctor.doc_id < 10 AND
t1_doctor.doc_id=t1_staff.doc_id;
```

```
select t1_bill.doctor_name,t1_patients.pat_name
from t1_bill
```

inner join t1_patients on t1_bill.amount > 500 AND
t1_bill.pat_id=t1_patients.pat_id;

Output:

Results		Messages	
	doc_name	pat_name	
1	Rashmi	Ramesh	
2	Jayesh	Suresh	
3	Sumit	Swati	
4	Rashmi	Pooja	
5	Rashmi	Raj	
6	Rashmi	Ramesh	
7	Rashmi	Zoya	
8	Rashmi	Henri	
9	Rashmi	Sreya	
10	Rashmi	Hardik	
11	Rashmi	Mansi	
12	Rashmi	Bhavin	
13	Sumit	Ramesh	
14	Jayesh	Suresh	
15	Sumit	Swati	

	doc_name	staff_name	
1	Jayesh	Himani	
2	Rashmi	Mohan	
3	Jayesh	Raja	
4	Sumit	Suhani	
5	Jayesh	Priya	
6	Sumit	Rajesh	
7	Rashmi	Piyush	

	doctor_name	pat_name	
1	Sumit	Swati	
2	Rashmi	Ramesh	
3	Jayesh	Suresh	
4	Jayesh	Swati	
5	Rashmi	Mansi	
6	Sumit	Ramesh	
7	Rashmi	Ramesh	
8	Jayesh	Hardik	

b. LEFT OUTER JOIN

Query:

```
select t1_doctor.doc_name,t1_patients.pat_name
from t1_patients
left outer join t1_doctor on t1_doctor.doc_id=t1_patients.doc_id;
```

```
select t1_doctor.doc_name,t1_staff.staff_name
from t1_staff
left outer join t1_doctor on t1_doctor.doc_id < 10 AND
t1_doctor.doc_id=t1_staff.doc_id;
```

```
select t1_bill.doctor_name,t1_patients.pat_name
from t1_bill
```

left outer join t1_patients on t1_bill.amount > 500 AND
t1_bill.pat_id=t1_patients.pat_id;

Output:

150 %		
Results Messages		
	doc_name	pat_name
2	Jayesh	Suresh
3	Sumit	Swati
4	Rashmi	Pooja
5	Rashmi	Raj
6	Rashmi	Ramesh
7	Rashmi	Zoya
8	Rashmi	Heni
9	Rashmi	Sreya
10	Rashmi	Hardik
11	Rashmi	Mansi
12	Rashmi	Bhavin
	doc_name	staff_name
1	Jayesh	Himani
2	Rashmi	Mohan
3	Jayesh	Raja
4	Sumit	Suhani
5	Jayesh	Priya
6	Sumit	Rajesh
7	Rashmi	Piyush
	doctor_name	pat_name
1	Sumit	Swati
2	Rajesh	NULL
3	Sumit	NULL
4	Rashmi	Ramesh
5	Jayesh	Suresh
6	Jayesh	Swati
7	Rashmi	Mansi
8	Rajesh	NULL
9	Sumit	Ramesh
10	Rashmi	Ramesh
11	Jayesh	Hardik

c. RIGHT OUTER JOIN

Query:

```
select t1_doctor.doc_name,t1_patients.pat_name
from t1_patients
right outer join t1_doctor on t1_doctor.doc_id=t1_patients.doc_id;
```

```
select t1_doctor.doc_name,t1_staff.staff_name
from t1_staff
right outer join t1_doctor on t1_doctor.doc_id < 10 AND
t1_doctor.doc_id=t1_staff.doc_id;
```

```
select t1_bill.doctor_name,t1_patients.pat_name
from t1_bill
```

right outer join t1_patients on t1_bill.amount > 500 AND
t1_bill.pat_id=t1_patients.pat_id;

Output:

150 %

Results Messages

	doc_name	pat_name
1	Sumit	Swati
2	Sumit	Ramesh
3	Sumit	Swati
4	Sumit	Ramesh
5	Sumit	Swati
6	Rashmi	Ramesh
7	Rashmi	Pooja
8	Rashmi	Raj

	doc_name	staff_name
1	Sumit	Suhani
2	Sumit	Rajesh
3	Rashmi	Mohan
4	Rashmi	Piyush
5	Jayesh	Himani
6	Jayesh	Raja
7	Jayesh	Priya
8	Sumit	NULL

	doctor_name	pat_name
1	NULL	Ramesh
2	Jayesh	Suresh
3	NULL	Swati
4	NULL	Pooja
5	NULL	Raj
6	NULL	Ramesh
7	NULL	Zoya
8	NULL	Heni
9	NULL	Sreya
10	Jayesh	Hardik
11	NULL	Mansi
12	NULL	Bhavin
13	NULL	Ramesh
14	NULL	Suresh
15	Sumit	Swati
16	Jayesh	Swati
17	NULL	Pooja

Query executed successfully.

8. Use all the above condition in JOIN as well.

Query:

```
select t1_doctor.doc_name,t1_patients.pat_name
from t1_patients
join t1_doctor on t1_doctor.doc_id=t1_patients.doc_id
```

```
select t1_doctor.doc_name,t1_staff.staff_name
from t1_staff
join t1_doctor on t1_doctor.doc_id < 10 AND t1_doctor.doc_id=t1_staff.doc_id
```

```
select t1_bill.doctor_name,t1_patients.pat_name
from t1_bill
join t1_patients on t1_bill.amount > 500 AND t1_bill.pat_id=t1_patients.pat_id
```

Output:

Results		
Messages		
	doc_name	pat_name
1	Rashmi	Ramesh
2	Jayesh	Suresh
3	Sumit	Swati
4	Rashmi	Pooja
5	Rashmi	Raj
6	Rashmi	Ramesh
7	Rashmi	Zoya
8	Rashmi	Heni
	doc_name	staff_name
1	Jayesh	Himani
2	Rashmi	Mohan
3	Jayesh	Raja
4	Sumit	Suhani
5	Jayesh	Priya
6	Sumit	Rajesh
7	Rashmi	Piyush
	doctor_name	pat_name
1	Sumit	Swati
2	Rashmi	Ramesh
3	Jayesh	Suresh
4	Jayesh	Swati
5	Rashmi	Mansi
6	Sumit	Ramesh
7	Rashmi	Ramesh
8	Jayesh	Hardik