

Neoskillz Zeno Talent

Python Developer

Assignment-1

Using MySQL, create 2 tables of your choice

Perform "Where" with "AND", " OR" and "NOT"
conditional operations

Also perform "Order By" and "LIKE"

Dheeraj Kumar S

dheerajkumars9926@gmail.com

1. Create table

(i). patient:

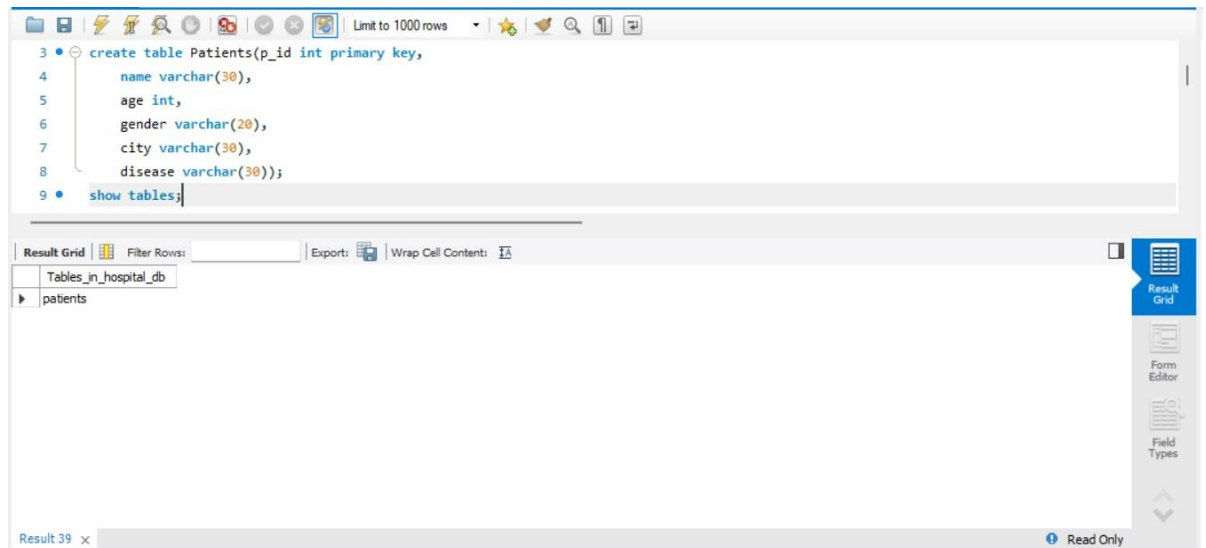
Code:

```
create table Patients(p_id int primary key,  
    name varchar(30),  
    age int,  
    gender varchar(20),  
    city varchar(30),  
    disease varchar(30));
```

Output:

Tables
patients

Screenshot:



(ii). Appointments

Code:

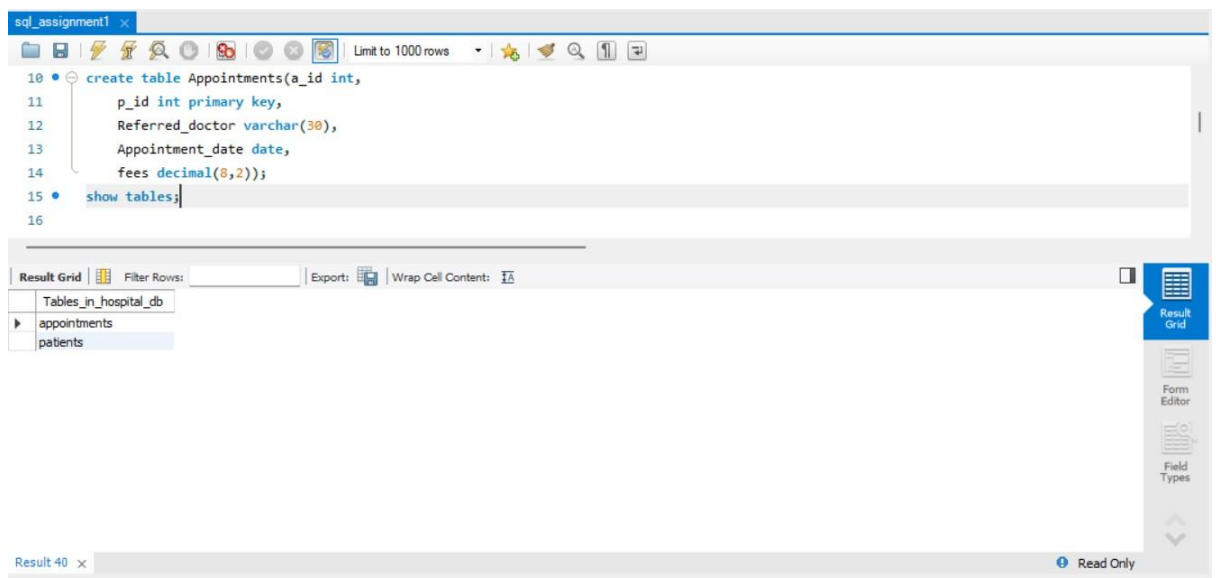
```
create table Appointments(a_id int primary key,  
    p_id int,  
    Referred_doctor varchar(30),
```

Appointment_date date,
fees decimal(8,2));

Output:

Tables
Appointments
Patients

Screenshot:



2. Insert into tables

(i). Patient

Code:

```
insert into Patients values
```

```
(1, 'Suresh', 35, 'male', 'Chennai', 'Asthma'),
```

```
(2, 'Lathika', 32, 'female', 'Chennai', 'fever'),
```

```
(3, 'Govind', 28, 'male', 'Chengalpattu', 'fever'),
```

```
(4, 'Hrithik', 27, 'male', 'Bengaluru', 'Fracture'),
```

```
(5, 'Prasanna', 37, 'female', 'Delhi', 'Meningitis');
```

Output:

1	Suresh	35	male	Chennai	Asthma
2	Lathika	32	female	Chennai	fever
3	Govind	28	male	Chengalpattu	fever
4	Hrithik	27	male	Bengaluru	Fracture
5	Prasanna	37	female	Delhi	Meningitis

Screenshot:

The screenshot shows a SQL IDE window titled 'sql_assignment1'. The SQL editor contains the following code:

```
16 • insert into Patients values
17 (1, 'Suresh', 35, 'male', 'Chennai', 'Asthma'),
18 (2, 'Lathika', 32, 'female', 'Chennai', 'fever'),
19 (3, 'Govind', 28, 'male', 'Chengalpattu', 'fever'),
20 (4, 'Hrithik', 27, 'male', 'Bengaluru', 'Fracture'),
21 (5, 'Prasanna', 37, 'female', 'Delhi', 'Meningitis');
22
```

Below the editor, the 'Result Grid' tab is active, displaying the following data:

p_id	name	age	gender	city	disease
1	Suresh	35	male	Chennai	Asthma
2	Lathika	32	female	Chennai	fever
3	Govind	28	male	Chengalpattu	fever
4	Hrithik	27	male	Bengaluru	Fracture
5	Prasanna	37	female	Delhi	Meningitis

The bottom status bar indicates 'Patients 7' and 'Read Only'.

(ii)Appointments

Code:

insert into appointments values

```
(101, 1, 'Dr. Avanthika', '2026-02-09', 10000),  
(102, 2, 'Dr. Ananya', '2026-02-09', 8000),  
(103, 3, 'Dr. Pavan', '2026-02-09', 11000),  
(104, 4, 'Dr. Anjali kumari', '2026-02-09', 13000),  
(105, 5, 'Dr. Pritam', '2026-02-09', 17000),  
(106, 4, 'Dr. Avanthika', '2026-02-10', 10000);
```

Output:

101	1	Dr. Avanthika	2026-02-09	10000.00
102	2	Dr. Ananya	2026-02-09	8000.00
103	3	Dr. Pavan	2026-02-09	11000.00
104	4	Dr. Anjali kumari	2026-02-09	13000.00
105	5	Dr. Pritam	2026-02-09	17000.00
106	4	Dr. Avanthika	2026-02-10	10000.00

Screenshot:

The screenshot shows a database management interface. At the top, there is a toolbar with various icons and a 'Limit to 1000 rows' dropdown. Below the toolbar, the SQL editor contains the following code:

```
25 • insert into appointments values  
26 (101, 1, 'Dr. Avanthika', '2026-02-09', 10000),  
27 (102, 2, 'Dr. Ananya', '2026-02-09', 8000),  
28 (103, 3, 'Dr. Pavan', '2026-02-09', 11000),  
29 (104, 4, 'Dr. Anjali kumari', '2026-02-09', 13000),  
30 (105, 5, 'Dr. Pritam', '2026-02-09', 17000),  
31 (106, 4, 'Dr. Avanthika', '2026-02-10', 10000);
```

Below the SQL editor, the 'Result Grid' tab is active, displaying the following data:

a_id	p_id	Referred_doctor	Appointment_date	fees
101	1	Dr. Avanthika	2026-02-09	10000.00
102	2	Dr. Ananya	2026-02-09	8000.00
103	3	Dr. Pavan	2026-02-09	11000.00
104	4	Dr. Anjali kumari	2026-02-09	13000.00
105	5	Dr. Pritam	2026-02-09	17000.00
106	4	Dr. Avanthika	2026-02-10	10000.00

At the bottom of the interface, there is a status bar showing 'appointments 47 x' and buttons for 'Apply' and 'Revert'.

3. Where, and, or , not:

(i), patients

(a) where:

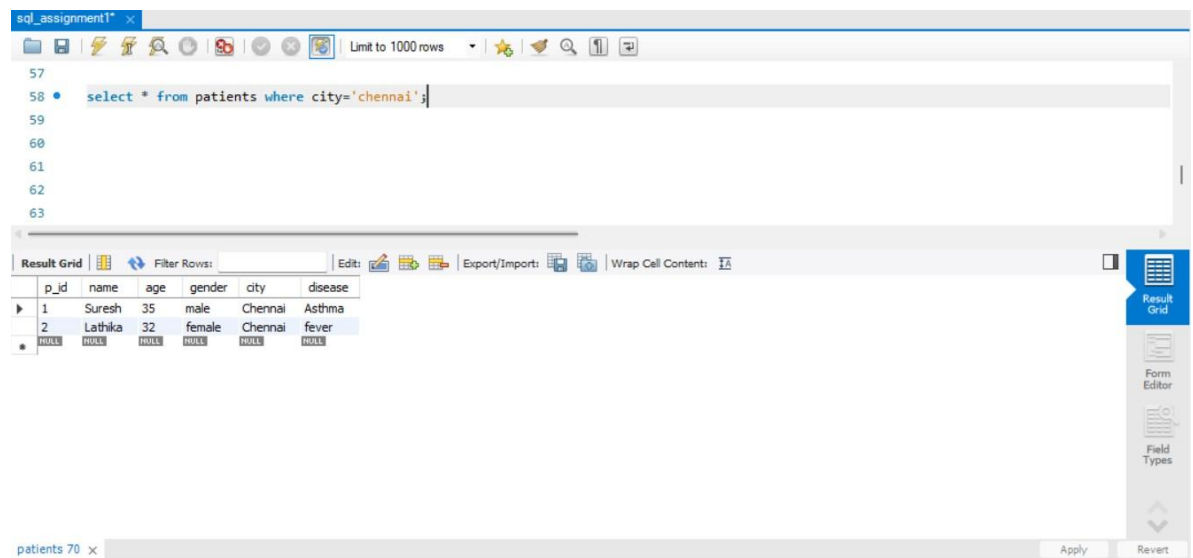
Code:

```
select * from patients where city='chennai';
```

Output:

1	Suresh	35	male	Chennai	Asthma
2	Lathika	32	female	Chennai	fever

Screenshot:



(b). and

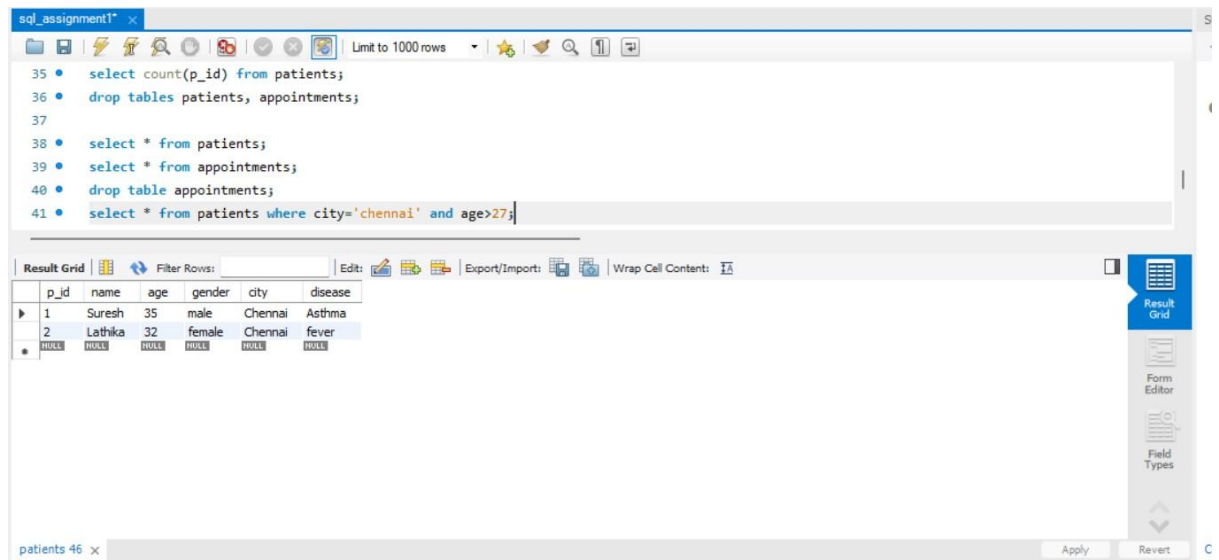
Code;

```
select * from patients where city='chennai' and age>27;
```

Output:

1	Suresh	35	male	Chennai	Asthma
2	Lathika	32	female	Chennai	fever

Screenshot:



(c). or

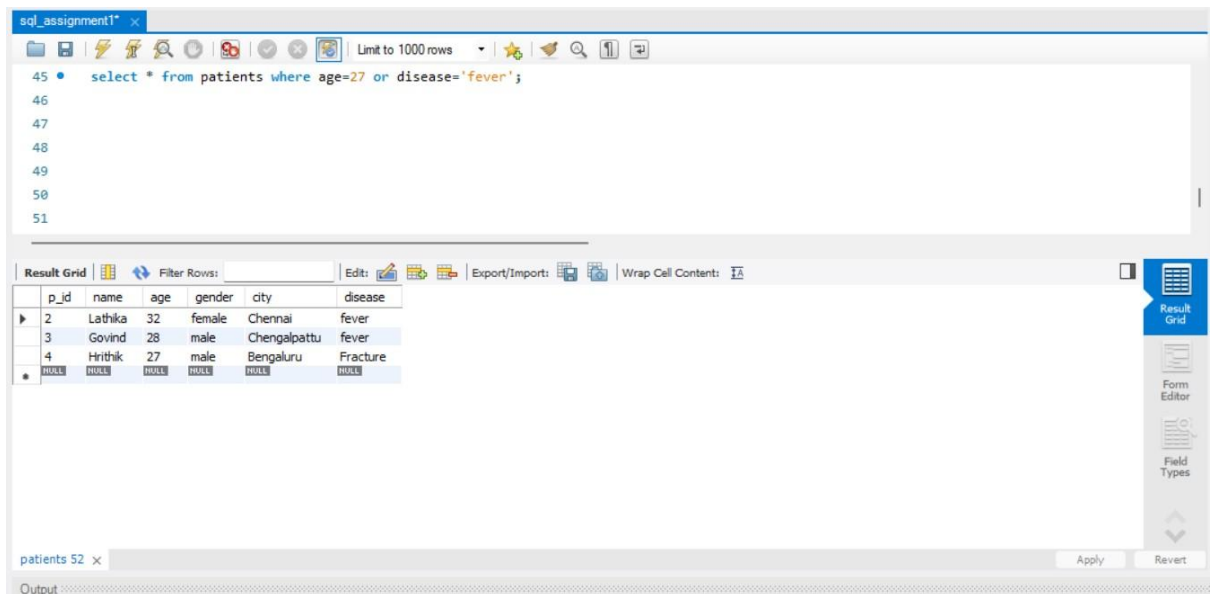
Code:

```
select * from patients where age=27 or disease='fever';
```

Output:

2	Lathika	32	female	Chennai	fever
3	Govind	28	male	Chengalpattu	fever
4	Hrithik	27	male	Bengaluru	Fracture

Screenshot:



(d). not

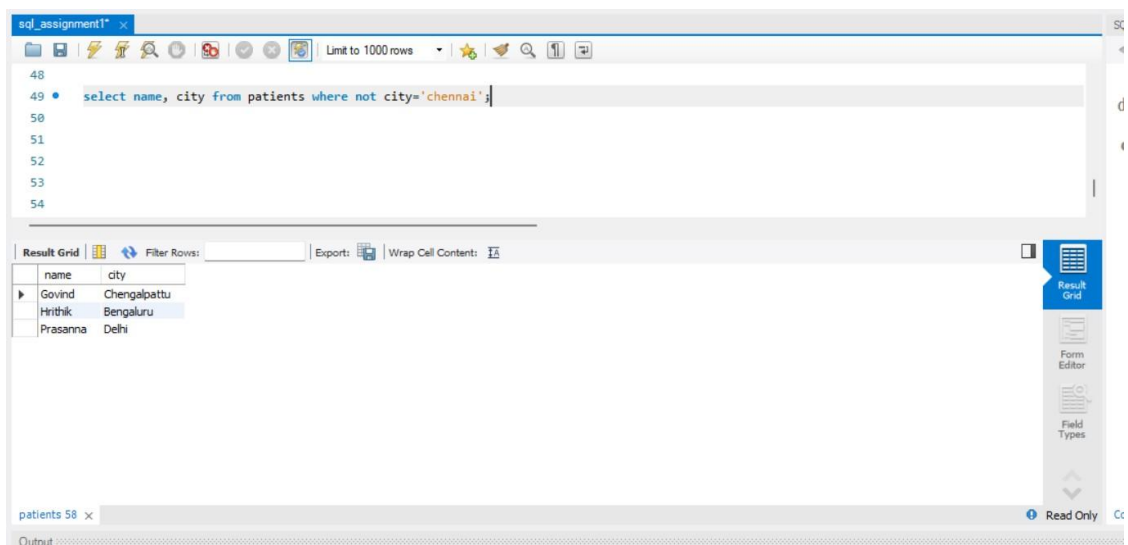
Code:

```
select name, city from patients where not city='chennai';
```

Output:

Govind	Chengalpattu
Hrithik	Bengaluru
Prasanna	Delhi

Screenshot:



(il). Appointments:

(a). where:

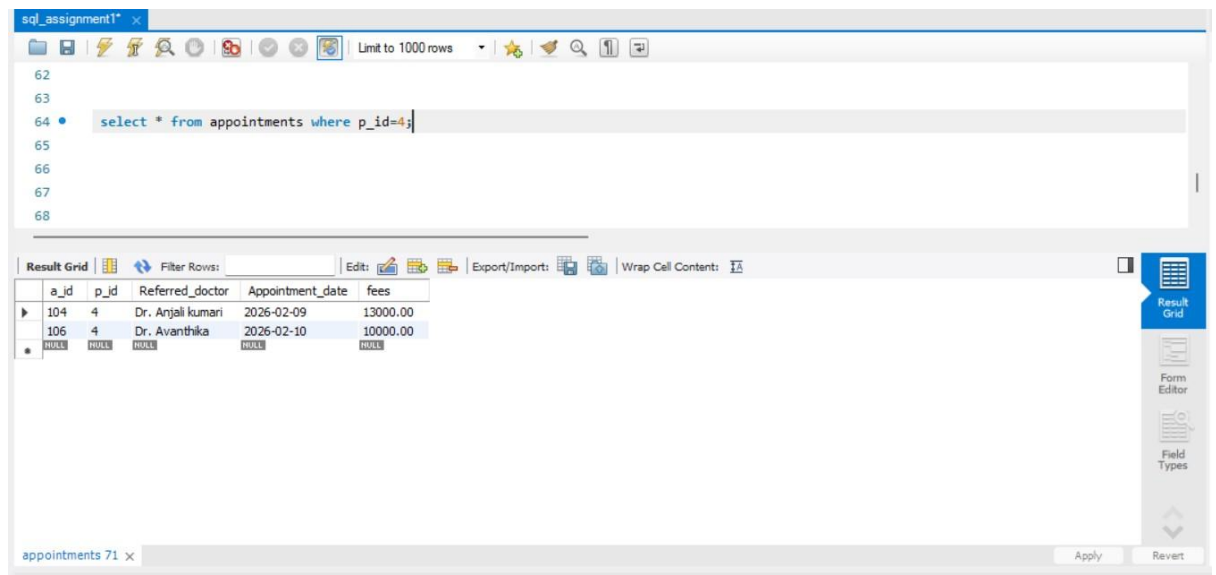
Code:

```
select * from appointments where p_id=4;
```

Output:

104	4	Dr. Anjali kumari	2026-02-09	13000.00
106	4	Dr. Avanthika	2026-02-10	10000.00

Screenshot:



(b). and

Code:

```
select * from appointments where referred_doctor='dr. avanthika' and  
Appointment_date='2026-02-09';
```

Output:

101	1	Dr. Avanthika	2026-02-09	10000.00
-----	---	---------------	------------	----------

Screenshot:

The screenshot shows a SQL query editor with the following query:

```
select * from appointments where referred_doctor='dr. avanthika' and Appointment_date='2026-02-09';
```

The result grid displays the following data:

a_id	p_id	Referred_doctor	Appointment_date	fees
101	1	Dr. Avanthika	2026-02-09	10000.00

(c). or

Code:

```
select * from appointments where referred_doctor='Dr. Pavan' or  
p_id=4;
```

Output:

103	3	Dr. Pavan	2026-02-09	11000.00
104	4	Dr. Anjali kumari	2026-02-09	13000.00
106	4	Dr. Avanthika	2026-02-10	10000.00

Screenshot:

The screenshot shows a SQL query editor with the following query:

```
select * from appointments where referred_doctor='Dr. Pavan' or p_id=4;
```

The result grid displays the following data:

a_id	p_id	Referred_doctor	Appointment_date	fees
103	3	Dr. Pavan	2026-02-09	11000.00
104	4	Dr. Anjali kumari	2026-02-09	13000.00
106	4	Dr. Avanthika	2026-02-10	10000.00

(d). not

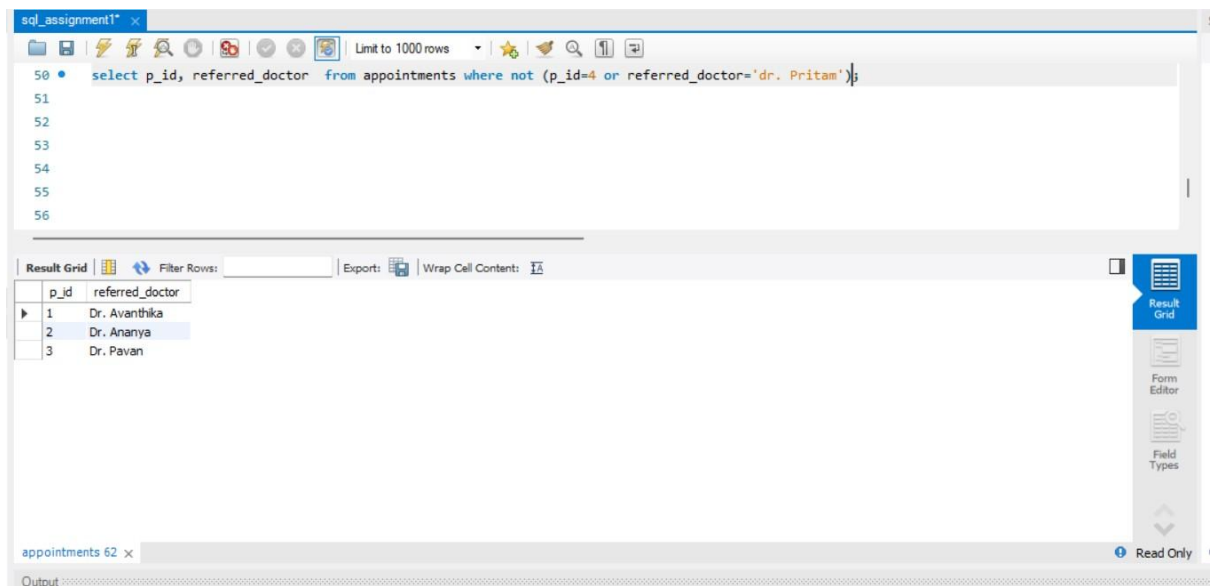
Code:

```
select p_id, referred_doctor from appointments where not (p_id=4 or referred_doctor='dr. Pritam');
```

Output:

1	Dr. Avanthika
2	Dr. Ananya
3	Dr. Pavan

Screenshot:



4. Order by and like:

(i), Patient:

(a). Order by:

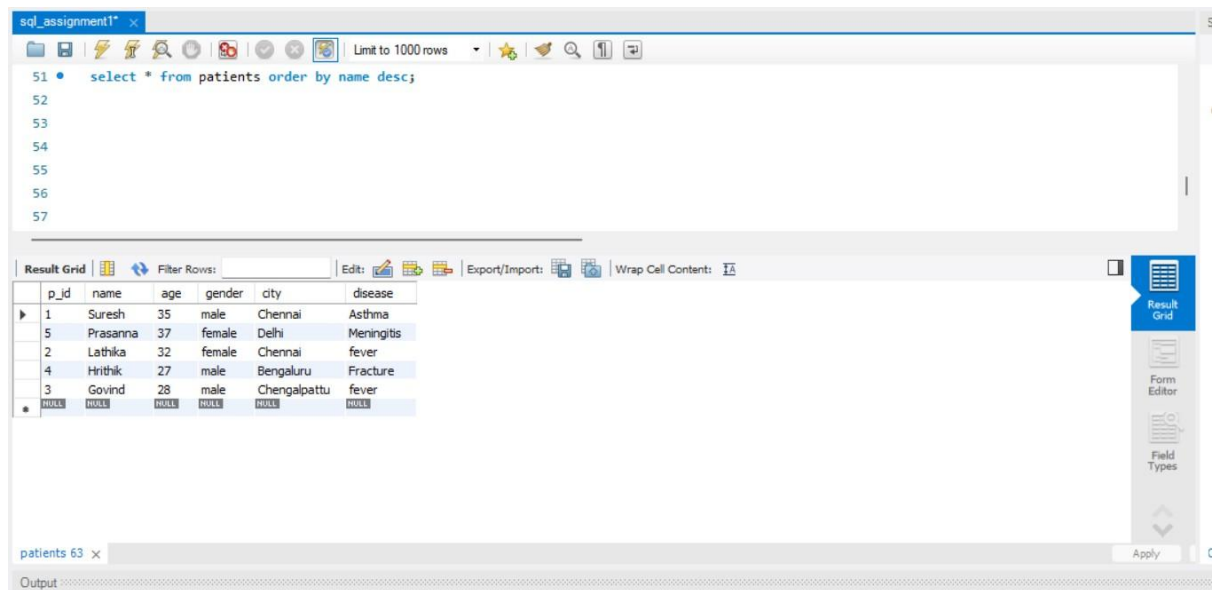
Code:

```
select * from patients order by name desc;
```

Output:

1	Suresh	35	male	Chennai	Asthma
5	Prasanna	37	female	Delhi	Meningitis
2	Lathika	32	female	Chennai	fever
4	Hrithik	27	male	Bengaluru	Fracture
3	Govind	28	male	Chengalpattu	fever

Screenshot:



(b). like

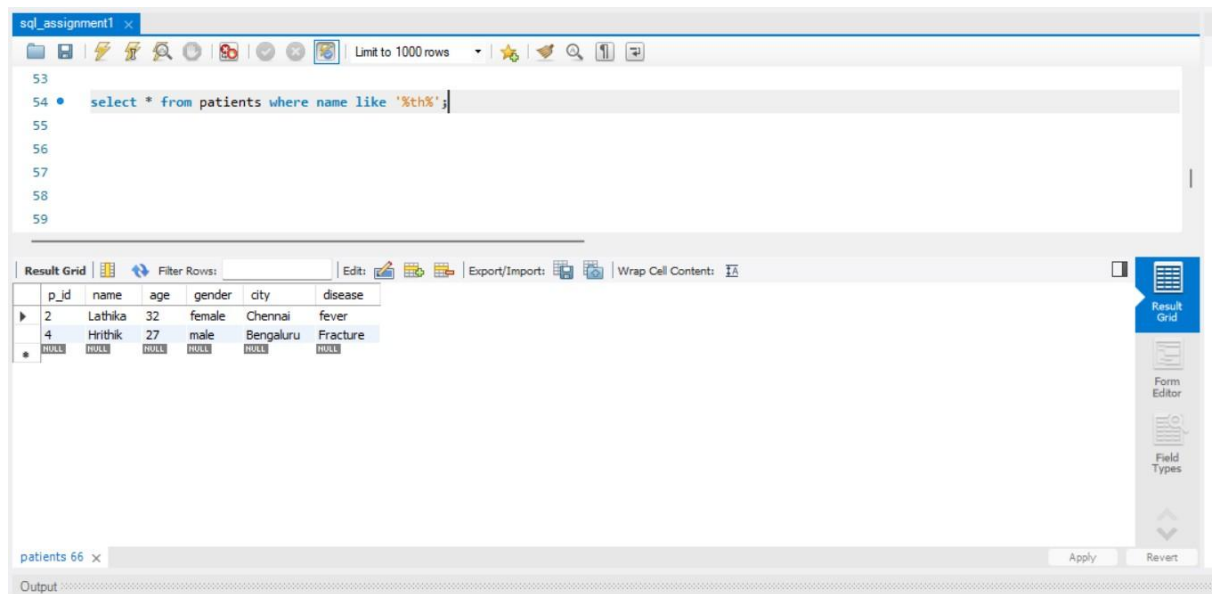
Code:

```
select * from patients where name like '%th%';
```

Output:

2	Lathika	32	female	Chennai	fever
4	Hrithik	27	male	Bengaluru	Fracture

Screenshot:



(ii) Appointments:

(a). Order by

Code:

```
select * from appointments order by referred_doctor desc;
```

Output:

105	5	Dr. Pritam	2026-02-09	17000.00
103	3	Dr. Pavan	2026-02-09	11000.00
101	1	Dr. Avanthika	2026-02-09	10000.00
106	4	Dr. Avanthika	2026-02-10	10000.00
104	4	Dr. Anjali kumari	2026-02-09	13000.00
102	2	Dr. Ananya	2026-02-09	8000.00

Screenshot:

The screenshot shows a database management tool interface. The top pane contains a SQL query: `select * from appointments order by referred_doctor desc;`. The bottom pane displays the results in a table format. The table has five columns: `a_id`, `p_id`, `Referred_doctor`, `Appointment_date`, and `fees`. The results are sorted by `Referred_doctor` in descending order.

a_id	p_id	Referred_doctor	Appointment_date	fees
105	5	Dr. Pritam	2026-02-09	17000.00
103	3	Dr. Pavan	2026-02-09	11000.00
101	1	Dr. Avanthika	2026-02-09	10000.00
106	4	Dr. Avanthika	2026-02-10	10000.00
104	4	Dr. Anjali kumari	2026-02-09	13000.00
102	2	Dr. Ananya	2026-02-09	8000.00
NULL	NULL	NULL	NULL	NULL

(b). like

Code:

```
select * from appointments where referred_doctor like '%a';
```

Output:

101	1	Dr. Avanthika	2026-02-09	10000.00
102	2	Dr. Ananya	2026-02-09	8000.00
106	4	Dr. Avanthika	2026-02-10	10000.00

Screenshot:

The screenshot shows the same database management tool interface. The top pane contains a SQL query: `select * from appointments where referred_doctor like '%a';`. The bottom pane displays the results in a table format. The table has five columns: `a_id`, `p_id`, `Referred_doctor`, `Appointment_date`, and `fees`. The results are filtered to show only appointments where the `Referred_doctor` contains the letter 'a'.

a_id	p_id	Referred_doctor	Appointment_date	fees
101	1	Dr. Avanthika	2026-02-09	10000.00
102	2	Dr. Ananya	2026-02-09	8000.00
106	4	Dr. Avanthika	2026-02-10	10000.00
NULL	NULL	NULL	NULL	NULL