

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


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

Output:

Tables
Appointments
Patients

Screenshot:

The screenshot shows the MySQL Workbench interface with a query editor window titled "sql_assignment1". The code entered is:

```
10 • ⚡ create table Appointments(a_id int,  
11     p_id int primary key,  
12     Referred_doctor varchar(30),  
13     Appointment_date date,  
14     fees decimal(8,2));  
15 •   show tables;
```

The results pane below shows the output of the "show tables" command:

Tables_in_hospital_db
appointments
patients

The interface includes various toolbars and a sidebar with icons for Result Grid, Form Editor, and Field Types.

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 interface with the following details:

- Query Tab:** The tab is titled "sql_assignment1". It contains the following SQL 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');
```
- Result Grid:** A table titled "Result Grid" displays the inserted data. The columns are P_Id, name, age, gender, city, and disease. The data matches the table shown above.
- Toolbar:** The toolbar includes various icons for file operations, search, and database management.
- Status Bar:** The status bar at the bottom shows "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 the MySQL Workbench interface. At the top, there is a toolbar with various icons. Below the toolbar, the SQL editor window displays 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 is the Result Grid window, which displays the data inserted into the 'appointments' table:

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
*	HULL	HULL	HULL	HULL

The bottom right corner of the interface shows a vertical toolbar with icons for Result Grid, Form Editor, and Field Types.

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:

The screenshot shows a database interface with a SQL editor and a result grid. The SQL query is:

```
57
58 • select * from patients where city='chennai';
59
60
61
62
63
```

The result grid displays the following data:

p_id	name	age	gender	city	disease
1	Suresh	35	male	Chennai	Asthma
2	Lathika	32	female	Chennai	fever
*	NULL	NULL	NULL	NULL	NULL

(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:

The screenshot shows a MySQL Workbench window titled "sql_assignment1". The query editor contains the following SQL code:

```

35 •    select count(p_id) from patients;
36 •    drop tables patients, appointments;
37
38 •    select * from patients;
39 •    select * from appointments;
40 •    drop table appointments;
41 •    select * from patients where city='chennai' and age>27;

```

The result grid displays the following data:

p_id	name	age	gender	city	disease
1	Suresh	35	male	Chennai	Asthma
2	Lathika	32	female	Chennai	fever
*		NULL	NULL	NULL	NULL

On the right side of the interface, there is a sidebar with icons for "Result Grid", "Form Editor", and "Field Types". At the bottom right, there are "Apply" and "Revert" buttons.

(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:

sql_assignment1* | Limit to 1000 rows | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

```

45 •  select * from patients where age=27 or disease='fever';
46
47
48
49
50
51

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

p_id	name	age	gender	city	disease
2	Lathika	32	female	Chennai	fever
3	Govind	28	male	Chengalpattu	fever
4	Hrithik	27	male	Bengaluru	Fracture
*	HULL	HULL	HULL	HULL	HULL

patients 52 | Output: | Apply | Revert |

(d). not

Code:

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

Output:

Govind	Chengalpattu
Hrithik	Bengaluru
Prasanna	Delhi

Screenshot:

sql_assignment1* | Limit to 1000 rows | Filter Rows: | Export: | Wrap Cell Content: |

```

48
49 •  select name, city from patients where not city='chennai';
50
51
52
53
54

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

name	city
Govind	Chengalpattu
Hrithik	Bengaluru
Prasanna	Delhi

patients 58 | Output: | Read Only | Co |

(ii). 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:

The screenshot shows a MySQL query editor window titled "sql_assignment1*". The code entered is:

```
62
63
64 • select * from appointments where p_id=4;
65
66
67
68
```

The results grid displays the following data:

a_id	p_id	Referred_doctor	Appointment_date	fees
104	4	Dr. Anjali kumari	2026-02-09	13000.00
106	4	Dr. Avanthika	2026-02-10	10000.00
*	HULL	HULL	HULL	HULL

The sidebar on the right includes icons for Result Grid, Form Editor, and Field Types.

(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 MySQL query editor window titled "sql_assignment1". The query entered is:

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

The result grid displays one row of data:

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

On the right side of the interface, there is a vertical toolbar with icons for "Result Grid", "Form Editor", and "Field Types".

(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 MySQL query editor window titled "sql_assignment1". The query entered is:

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

The result grid displays three rows of 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

On the right side of the interface, there is a vertical toolbar with icons for "Result Grid", "Form Editor", and "Field Types".

(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:

The screenshot shows a database query window titled "sql_assignment1*". The query entered is:

```
50 • select p_id, referred_doctor from appointments where not (p_id=4 or referred_doctor='dr. Pritam');  
51  
52  
53  
54  
55  
56
```

The results grid displays the following data:

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

The interface includes various toolbars and a sidebar with icons for Result Grid, Form Editor, and Field Types.

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:

The screenshot shows the MySQL Workbench interface with a query editor window titled "sql_assignment1*". The query is:

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

The results are displayed in a "Result Grid" table:

p_id	name	age	gender	city	disease
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
•	HULL	HULL	HULL	HULL	HULL

Below the grid, there is a message: "patients 63 x".

(b). like

Code:

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

Output:

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

Screenshot:

The screenshot shows a MySQL query editor window titled "sql_assignment1". The SQL code in the query pane is:

```
53
54 •  select * from patients where name like '%th%';
55
56
57
58
59
```

The results pane displays a grid of patient data:

p_id	name	age	gender	city	disease
2	Lathika	32	female	Chennai	fever
4	Hirthik	27	male	Bengaluru	Fracture
*	HULL	HULL	HULL	HULL	HULL

On the right side of the interface, there is a vertical toolbar with icons for "Result Grid", "Form Editor", and "Field Types".

(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 the SQL Management Studio interface with a query window titled 'sql_assignment1'. The query is:

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

The results grid displays the following data:

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

The output pane at the bottom shows 'Output'.

(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 SQL Management Studio interface with a query window titled 'sql_assignment1'. The query is:

```
55
56 •  select * from appointments where referred_doctor like '%a';
```

The results grid displays 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
106	4	Dr. Avanthika	2026-02-10	10000.00
*	NULL	NULL	NULL	NULL

The output pane at the bottom shows 'Output'.