

## Oracle SQL Capstone Project Exam (17/07/2024)

- Code is written in MySQL format due to technical issues in Oracle XE 21c Edition. So, I had provided this code. Below the output of this code, I had provided the equivalent code in Oracle SQL format. Please consider the code given below.

### MySQL Format: -

```
create database MHMS;  
use MHMS;
```

```
-- 1. Database Schema (DDL with Constraints)  
-- Create Tables patients, departments, staff, treatments  
create table patients(patientid int primary key,  
name varchar (100), dob date, email varchar (100) unique,  
phone varchar (20));
```

```
create table departments(  
departmentid int primary key,  
departmentname varchar(100));
```

```
create table staff(staffid int primary key,  
name varchar (100), role varchar (50), departmentid int,  
foreign key (departmentid) references departments  
(departmentid));
```

```
create table treatments(  
treatmentid int primary key,  
patientid int, staffid int,  
treatmentdate date, cost decimal(10, 2),  
foreign key (patientid) references patients(patientid),  
foreign key (staffid) references staff(staffid),  
check (cost>=0));
```

```
-- 2. Data Manipulation (DML)  
-- Insert values according to given parameters in particular table  
insert into patients values (1, 'john doe', '1985-05-15',  
'john.doe@example.com', '555-0101');
```

```
insert into patients values (2, 'jane smith', '1990-06-20',  
'jane.smith@example.com', '555-0102');  
insert into patients values (3, 'alice johnson', '1975-09-25',  
'alice.j@example.com', '555-0103');  
insert into patients values (4, 'bob brown', '1982-01-30',  
'bob.brown@example.com', '555-0104');  
insert into patients values (5, 'eve white', '1995-12-12',  
'eve.white@example.com', '555-0105');  
select * from patients;
```

```
insert into departments values (1, 'cardiology');  
insert into departments values (2, 'neurology');  
insert into departments values (3, 'orthopedics');  
select * from departments;
```

```
insert into staff values (1, 'dr. smith', 'cardiologist', 1);  
insert into staff values (2, 'dr. jones', 'neurologist', 2);  
insert into staff values (3, 'dr. brown', 'orthopedic surgeon', 3);  
insert into staff values (4, 'nurse kelly', 'nurse', 1);  
insert into staff values (5, 'nurse taylor', 'nurse', 2);  
select * from staff;
```

```
insert into treatments values (1, 1, 1, '2023-01-10', 500);  
insert into treatments values (2, 2, 2, '2023-02-15', 1500);  
insert into treatments values (3, 3, 3, '2023-03-20', 750);  
insert into treatments values (4, 4, 1, '2023-04-05', 200);  
insert into treatments values (5, 5, 2, '2023-05-30', 1200);  
insert into treatments values (6, 1, 3, '2023-06-10', 300);  
insert into treatments values (7, 2, 1, '2023-07-01', 800);  
insert into treatments values (8, 3, 2, '2023-07-15', 950);  
insert into treatments values (9, 4, 1, '2023-08-20', 1100);  
insert into treatments values (10, 5, 3, '2023-09-25', 650);  
select * from treatments;
```

```
-- Update the cost of a specific treatment.  
update treatments set cost=600 where treatmentid=6;  
select * from treatments;
```

```
-- Delete a patient and observe the cascading effect on associated  
treatments.  
delete from patients where patientid=4;  
-- Cannot delete or update a parent row due to failing of foreign key  
constraint
```

```

-- drop the existing foreign key constraint
alter table treatments drop foreign key treatments_ibfk_1;

-- recreate the foreign key constraint with ON DELETE CASCADE
alter table treatments
add constraint fk_treatments_patients
foreign key (patientid) references patients(patientid)
on delete cascade;

-- Delete a patient and observe the cascading effect on associated
treatments.
delete from patients where patientid=4;
select * from patients;

/*
Data Retrieval using different operations
Write a query to list all patients, their associated treatments, and the
staff who provided those treatments.
*/
select p.name as patient_name, t.treatmentdate, s.name as staff_name
from patients p
join treatments t on p.patientid = t.patientid
join staff s on t.staffid = s.staffid;

-- Create a query to display all treatments, including patient names, staff
names, and department names.
select p.name as patient_name, s.name as staff_name, d.departmentname,
t.treatmentdate, t.cost
from treatments t
join patients p on t.patientid = p.patientid
join staff s on t.staffid = s.staffid
join departments d on s.departmentid = d.departmentid;

/*
4. Subqueries and Correlated Subqueries
Use a subquery to find patients who have received treatments from a specific
department (provided DepartmentID).
*/
select distinct p.*
from patients p
join treatments t on p.patientid = t.patientid
join staff s on t.staffid = s.staffid

```

```
where s.departmentid = 1;
```

```
-- Use a correlated subquery to find staff who have treated more than a  
specified number of patients (e.g., 3).
```

```
select *  
from staff s  
where (select count(*) from treatments t  
where t.staffid = s.staffid) > 2;
```

```
/*
```

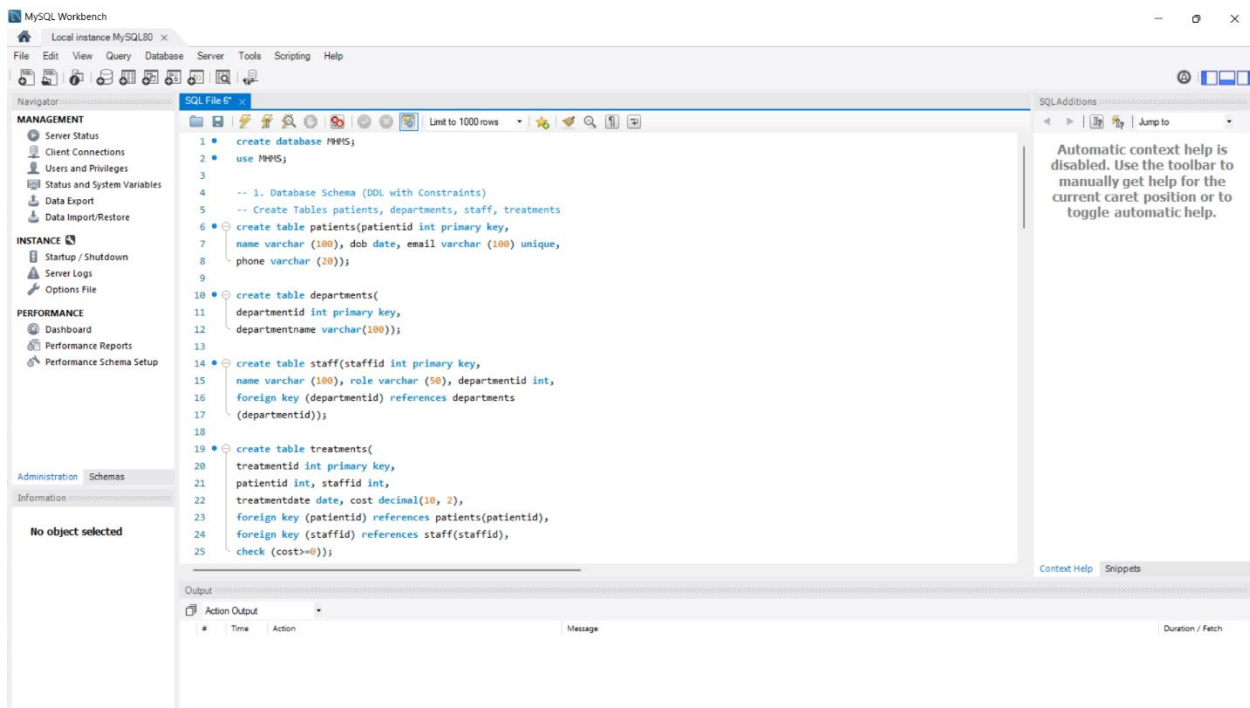
### 5. Aggregate Functions with GROUP BY and HAVING [5 Marks]

Calculate the total cost of treatments per patient and filter out patients  
with a total treatment cost of less than Rs.1000

```
*/
```

```
select p.name as patient_name, sum(t.cost) as total_cost  
from patients p  
join treatments t on p.patientid = t.patientid  
group by p.name  
having sum(t.cost) >= 1000;
```

## OUTPUT: -



MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Administration Schemas

Information

No object selected

SQL File 6

```
22 treatmentdate date, cost decimal(10, 2),
23 foreign key (patientid) references patients(patientid),
24 foreign key (staffid) references staff(staffid),
25 check (cost>=0));
26
27 -- 2. Data Manipulation (DML)
28 -- Insert values according to given parameters in particular table
29 • insert into patients values (1, 'john doe', '1985-05-15', 'john.doe@example.com', '555-0101');
30 • insert into patients values (2, 'jane smith', '1990-06-20', 'jane.smith@example.com', '555-0102');
31 • insert into patients values (3, 'alice johnson', '1975-09-25', 'alice.j@example.com', '555-0103');
32 • insert into patients values (4, 'bob brown', '1982-01-30', 'bob.brown@example.com', '555-0104');
33 • insert into patients values (5, 'eve white', '1995-12-12', 'eve.white@example.com', '555-0105');
34 • select * from patients;
35
36 • insert into departments values (1, 'cardiology');
37 • insert into departments values (2, 'neurology');
38 • insert into departments values (3, 'orthopedics');
39 • select * from departments;
40
41 • insert into staff values (1, 'dr. smith', 'cardiologist', 1);
42 • insert into staff values (2, 'dr. jones', 'neurologist', 2);
43 • insert into staff values (3, 'dr. brown', 'orthopedic surgeon', 3);
44 • insert into staff values (4, 'nurse kelly', 'nurse', 1);
45 • insert into staff values (5, 'nurse taylor', 'nurse', 2);
46 • select * from staff;
```

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Context Help Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
---	------	--------	---------	------------------

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Administration Schemas

Information

No object selected

SQL File 6

```
43 • insert into staff values (3, 'dr. brown', 'orthopedic surgeon', 3);
44 • insert into staff values (4, 'nurse kelly', 'nurse', 1);
45 • insert into staff values (5, 'nurse taylor', 'nurse', 2);
46 • select * from staff;
47
48 • insert into treatments values (1, 1, 1, '2023-01-10', 500);
49 • insert into treatments values (2, 2, 2, '2023-02-15', 1500);
50 • insert into treatments values (3, 3, 3, '2023-03-20', 750);
51 • insert into treatments values (4, 4, 1, '2023-04-05', 200);
52 • insert into treatments values (5, 5, 2, '2023-05-30', 1200);
53 • insert into treatments values (6, 1, 3, '2023-06-10', 300);
54 • insert into treatments values (7, 2, 1, '2023-07-01', 800);
55 • insert into treatments values (8, 3, 2, '2023-07-15', 950);
56 • insert into treatments values (9, 4, 1, '2023-08-20', 1100);
57 • insert into treatments values (10, 5, 3, '2023-09-25', 650);
58 • select * from treatments;
59
60 -- Update the cost of a specific treatment.
61 • update treatments set cost=600 where treatmentid=6;
62 • select * from treatments;
63
64 -- Delete a patient and observe the cascading effect on associated treatments.
65 • delete from patients where patientid=4;
66 -- Cannot delete or update a parent row due to falling of foreign key constraint
67
```

SQLAdditions

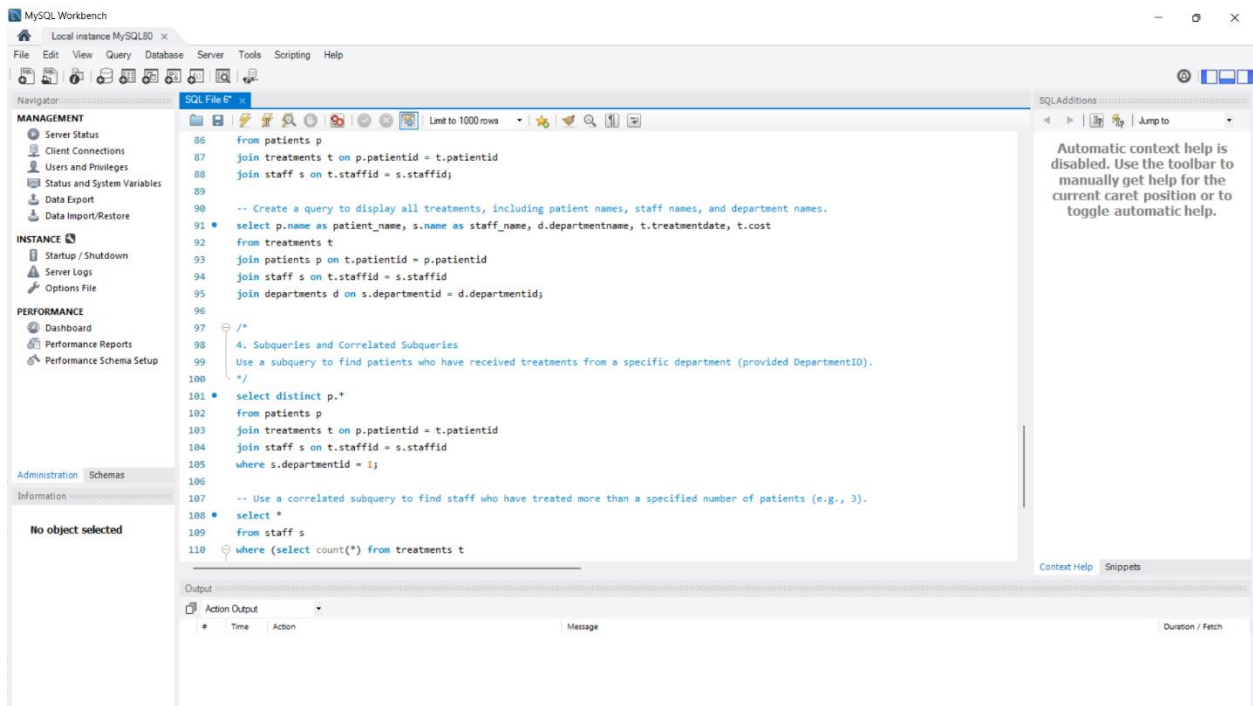
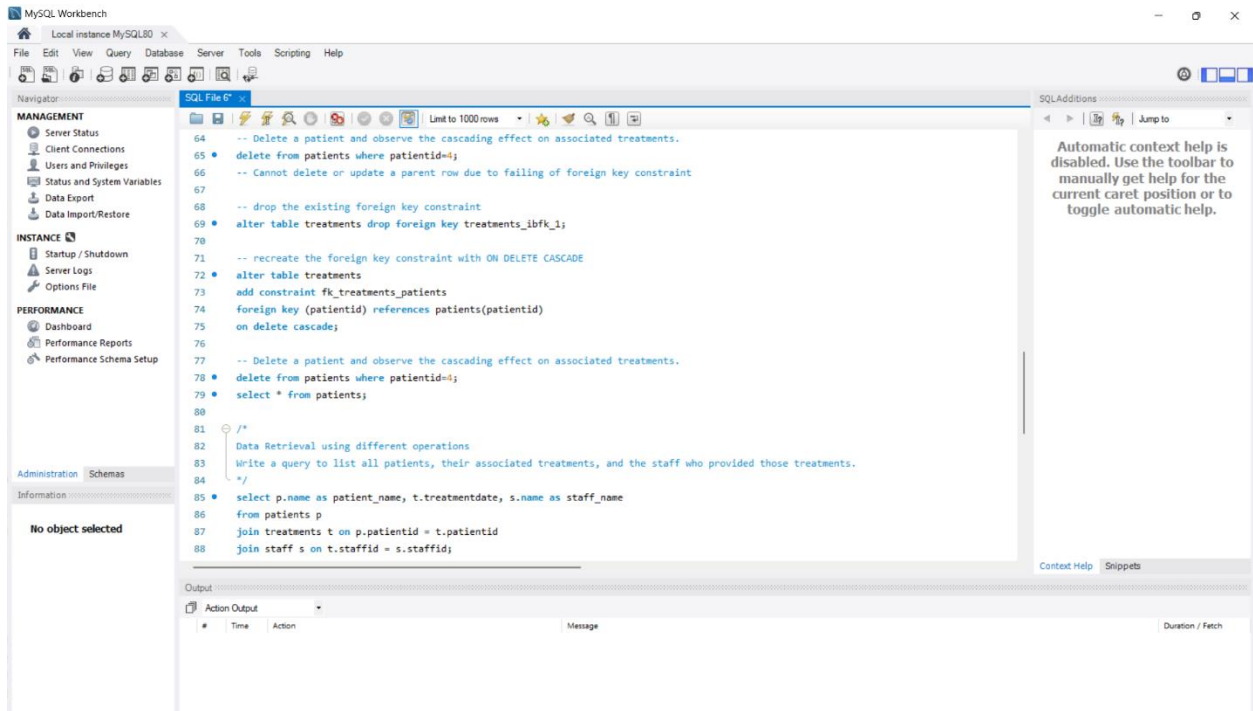
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Context Help Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
---	------	--------	---------	------------------



MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Administration Schemas

Information

No object selected

SQL File 0

```
4. Subqueries and Correlated Subqueries
Use a subquery to find patients who have received treatments from a specific department (provided DepartmentID).
*/
select distinct p.*
from patients p
join treatments t on p.patientid = t.patientid
join staff s on t.staffid = s.staffid
where s.departmentid = 1;

-- Use a correlated subquery to find staff who have treated more than a specified number of patients (e.g., 3).
select *
from staff s
where (select count(*) from treatments t
where t.staffid = s.staffid) > 3;

/*
5. Aggregate Functions with GROUP BY and HAVING [5 Marks]
Calculate the total cost of treatments per patient and filter out patients
with a total treatment cost of less than Rs.1000
*/
select p.name as patient_name, sum(t.cost) as total_cost
from patients p
join treatments t on p.patientid = t.patientid
group by p.name
having sum(t.cost) >= 1000;
```

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Context Help Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
---	------	--------	---------	------------------

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Administration Schemas

Information

No object selected

SQL File 0

```
check (cost>0));

-- 2. Data Manipulation (DML)
-- Insert values according to given parameters in particular table
insert into patients values (1, 'john doe', '1985-05-15', 'john.doe@example.com', '555-0101');
insert into patients values (2, 'jane smith', '1990-06-20', 'jane.smith@example.com', '555-0102');
insert into patients values (3, 'alice johnson', '1975-09-25', 'alice.j@example.com', '555-0103');
insert into patients values (4, 'bob brown', '1982-01-30', 'bob.brown@example.com', '555-0104');
insert into patients values (5, 'eve white', '1995-12-12', 'eve.white@example.com', '555-0105');
select * from patients;

insert into departments values (1, 'cardiology');
insert into departments values (2, 'neurology');
insert into departments values (3, 'orthopedics');
select * from departments;
```

Result Grid

patientid	name	dob	email	phone
1	john doe	1985-05-15	john.doe@example.com	555-0101
2	jane smith	1990-06-20	jane.smith@example.com	555-0102
3	alice johnson	1975-09-25	alice.j@example.com	555-0103
4	bob brown	1982-01-30	bob.brown@example.com	555-0104
5	eve white	1995-12-12	eve.white@example.com	555-0105

patients 15 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	20:41:45	select * from patients LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: SQL File 6 x

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Administration Schemas

Information

No object selected

```
25 check (cost>0));
26
27 -- 2. Data Manipulation (DML)
28 -- Insert values according to given parameters in particular table
29 insert into patients values (1, 'john doe', '1985-05-15', 'john.doe@example.com', '555-0101');
30 insert into patients values (2, 'jane smith', '1990-06-20', 'jane.smith@example.com', '555-0102');
31 insert into patients values (3, 'alice johnson', '1975-09-25', 'alice.j@example.com', '555-0103');
32 insert into patients values (4, 'bob brown', '1982-01-30', 'bob.brown@example.com', '555-0104');
33 insert into patients values (5, 'eve white', '1995-12-12', 'eve.white@example.com', '555-0105');
34 select * from patients;
35
36 insert into departments values (1, 'cardiology');
37 insert into departments values (2, 'neurology');
38 insert into departments values (3, 'orthopedics');
39 select * from departments;
```

Result Grid

departmentid	departmentname
1	cardiology
2	neurology
3	orthopedics

departments 16 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	20:41:45	select * from patients LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
2	20:42:08	select * from departments LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: SQL File 6 x

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Administration Schemas

Information

No object selected

```
40
41 insert into staff values (1, 'dr. smith', 'cardiologist', 1);
42 insert into staff values (2, 'dr. jones', 'neurologist', 2);
43 insert into staff values (3, 'dr. brown', 'orthopedic surgeon', 3);
44 insert into staff values (4, 'nurse kelly', 'nurse', 1);
45 insert into staff values (5, 'nurse taylor', 'nurse', 2);
46 select * from staff;
47
48 insert into treatments values (1, 1, 1, '2023-01-10', 500);
49 insert into treatments values (2, 2, 2, '2023-02-15', 1500);
50 insert into treatments values (3, 3, 3, '2023-03-20', 750);
51 insert into treatments values (4, 4, 1, '2023-04-05', 200);
52 insert into treatments values (5, 5, 2, '2023-05-30', 1200);
53 insert into treatments values (6, 1, 3, '2023-06-10', 300);
54 insert into treatments values (7, 2, 1, '2023-07-01', 800);
```

Result Grid

staffid	name	role	departmentid
1	dr. smith	cardiologist	1
2	dr. jones	neurologist	2
3	dr. brown	orthopedic surgeon	3
4	nurse kelly	nurse	1
5	nurse taylor	nurse	2

staff 17 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	20:41:45	select * from patients LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
2	20:42:08	select * from departments LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
3	20:42:22	select * from staff LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec



MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Administration Schemas

No object selected

SQL File 0

```
46 • select * from staff;
47
48 • insert into treatments values (1, 1, 1, '2023-01-10', 500);
49 • insert into treatments values (2, 2, 2, '2023-02-15', 1500);
50 • insert into treatments values (3, 3, 3, '2023-03-20', 750);
51 • insert into treatments values (4, 4, 1, '2023-04-05', 200);
52 • insert into treatments values (5, 5, 2, '2023-05-30', 1200);
53 • insert into treatments values (6, 1, 3, '2023-06-10', 300);
54 • insert into treatments values (7, 2, 1, '2023-07-01', 800);
55 • insert into treatments values (8, 3, 2, '2023-07-15', 950);
56 • insert into treatments values (9, 4, 1, '2023-08-20', 1100);
57 • insert into treatments values (10, 5, 3, '2023-09-25', 650);
58 • select * from treatments;
59
```

Result Grid

	treatmentid	patientid	staffid	treatmentdate	cost
1	1	1	1	2023-01-10	500.00
2	2	2	2	2023-02-15	1500.00
3	3	3	3	2023-03-20	750.00
5	5	2	2	2023-05-30	1200.00
6	1	3	3	2023-06-10	300.00
7	2	1	1	2023-07-01	800.00
8	3	2	2	2023-07-15	950.00
10	5	3	3	2023-09-25	650.00
•	max	max	max	max	max

treatments 20

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	20:43:30	select * from treatments LIMIT 0, 1000	8 row(s) returned	0.000 sec / 0.000 sec

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Administration Schemas

No object selected

SQL File 0

```
48 • insert into treatments values (1, 1, 1, '2023-01-10', 500);
49 • insert into treatments values (2, 2, 2, '2023-02-15', 1500);
50 • insert into treatments values (3, 3, 3, '2023-03-20', 750);
51 • insert into treatments values (4, 4, 1, '2023-04-05', 200);
52 • insert into treatments values (5, 5, 2, '2023-05-30', 1200);
53 • insert into treatments values (6, 1, 3, '2023-06-10', 300);
54 • insert into treatments values (7, 2, 1, '2023-07-01', 800);
55 • insert into treatments values (8, 3, 2, '2023-07-15', 950);
56 • insert into treatments values (9, 4, 1, '2023-08-20', 1100);
57 • insert into treatments values (10, 5, 3, '2023-09-25', 650);
58 • select * from treatments;
59
60 -- Update the cost of a specific treatment.
61 • update treatments set cost=600 where treatmentid=6;
62 • select * from treatments;
```

Result Grid

	treatmentid	patientid	staffid	treatmentdate	cost
1	1	1	1	2023-01-10	500.00
2	2	2	2	2023-02-15	1500.00
3	3	3	3	2023-03-20	750.00
5	5	2	2	2023-05-30	1200.00
6	1	3	3	2023-06-10	600.00
7	2	1	1	2023-07-01	800.00
8	3	2	2	2023-07-15	950.00
10	5	3	3	2023-09-25	650.00
•	max	max	max	max	max

treatments 21

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	20:43:30	select * from treatments LIMIT 0, 1000	8 row(s) returned	0.000 sec / 0.000 sec
2	20:43:55	update treatments set cost=600 where treatmentid=6;	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
3	20:43:59	select * from treatments LIMIT 0, 1000	8 row(s) returned	0.000 sec / 0.000 sec

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Administration Schemas

Information

No object selected

SQL File 6\*

```
65 delete from patients where patientid=4;
66 -- Cannot delete or update a parent row due to failing of foreign key constraint
67
68 -- drop the existing foreign key constraint
69 alter table treatments drop foreign key treatments_ibfk_1;
70
71 -- recreate the foreign key constraint with ON DELETE CASCADE
72 alter table treatments
73 add constraint fk_treatments_patients
74 foreign key (patientid) references patients(patientid)
75 on delete cascade;
76
77 -- Delete a patient and observe the cascading effect on associated treatments.
78 delete from patients where patientid=4;
79 select * from patients;
```

Result Grid

	patientid	name	dob	email	phone
1	john doe	1985-05-15	john.doe@example.com	555-0101	
2	jane smith	1990-06-20	jane.smith@example.com	555-0102	
3	alice johnson	1975-09-25	alice.j@example.com	555-0103	
5	eve white	1995-12-12	eve.white@example.com	555-0105	

patients 22 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	20:44:46	delete from patients where patientid=4	0 row(s) affected	0.000 sec
2	20:44:54	select * from patients LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Administration Schemas

Information

No object selected

SQL File 6\*

```
80
81 /*
82 Data Retrieval using different operations
83 Write a query to list all patients, their associated treatments, and the staff who provided those treatments.
84 */
85 select p.name as patient_name, t.treatmentdate, s.name as staff_name
86 from patients p
87 join treatments t on p.patientid = t.patientid
88 join staff s on t.staffid = s.staffid;
89
90 -- Create a query to display all treatments, including patient names, staff names, and department names.
91 select p.name as patient_name, s.name as staff_name, d.departmentname, t.treatmentdate, t.cost
92 from treatments t
93 join patients p on t.patientid = p.patientid
94 join staff s on t.staffid = s.staffid
```

Result Grid

	patient_name	treatmentdate	staff_name
1	john doe	2023-01-10	dr. smith
2	john doe	2023-06-10	dr. brown
3	jane smith	2023-02-15	dr. jones
4	jane smith	2023-07-01	dr. smith
5	alice johnson	2023-03-20	dr. brown
6	alice johnson	2023-07-15	dr. jones
7	eve white	2023-05-30	dr. jones
8	eve white	2023-09-25	dr. brown

Result 24 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	20:45:32	select p.name as patient_name, t.treatmentdate, s.name as staff_name fr...	8 row(s) returned	0.000 sec / 0.000 sec

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Administration Schemas

Information

No object selected

SQL File 6\*

```
86 from patients p
87 join treatments t on p.patientid = t.patientid
88 join staff s on t.staffid = s.staffid;
89
90 -- Create a query to display all treatments, including patient names, staff names, and department names.
91 select p.name as patient_name, s.name as staff_name, d.departmentname, t.treatmentdate, t.cost
92 from treatments t
93 join patients p on t.patientid = p.patientid
94 join staff s on t.staffid = s.staffid
95 join departments d on s.departmentid = d.departmentid;
96
97 /*
98 4. Subqueries and Correlated Subqueries
99 Use a subquery to find patients who have received treatments from a specific department (provided DepartmentID).
100 */
```

Result Grid

patient_name	staff_name	departmentname	treatmentdate	cost
John doe	Dr. Smith	cardiology	2023-01-10	500.00
Jane Smith	Dr. Smith	cardiology	2023-07-01	800.00
Jane Smith	Dr. Jones	neurology	2023-02-15	1500.00
Eve White	Dr. Jones	neurology	2023-05-30	1200.00
Alice Johnson	Dr. Jones	neurology	2023-07-15	950.00
Alice Johnson	Dr. Brown	orthopedics	2023-03-20	750.00
John doe	Dr. Brown	orthopedics	2023-06-10	600.00
Eve White	Dr. Brown	orthopedics	2023-09-25	650.00

Result 26

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	20:45:53	select p.name as patient_name, s.name as staff_name, d.departmentname...	8 row(s) returned	0.000 sec / 0.000 sec

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Administration Schemas

Information

No object selected

SQL File 6\*

```
95 join departments d on s.departmentid = d.departmentid;
96
97 /*
98 4. Subqueries and Correlated Subqueries
99 Use a subquery to find patients who have received treatments from a specific department (provided DepartmentID).
100 */
101 select distinct p.*
102 from patients p
103 join treatments t on p.patientid = t.patientid
104 join staff s on t.staffid = s.staffid
105 where s.departmentid = 1;
106
107 -- Use a correlated subquery to find staff who have treated more than a specified number of patients (e.g., 3).
108 select *
109 from staff s
```

Result Grid

patient_name	staff_name	departmentname	treatmentdate	cost
John doe	Dr. Smith	cardiology	2023-01-10	500.00
Jane Smith	Dr. Smith	cardiology	2023-02-15	1500.00
Jane Smith	Dr. Jones	neurology	2023-05-30	1200.00
Eve White	Dr. Jones	neurology	2023-07-15	950.00
Alice Johnson	Dr. Jones	neurology	2023-03-20	750.00
John doe	Dr. Brown	orthopedics	2023-06-10	600.00
Eve White	Dr. Brown	orthopedics	2023-09-25	650.00

Result 26

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	20:45:53	select p.name as patient_name, s.name as staff_name, d.departmentname...	8 row(s) returned	0.000 sec / 0.000 sec

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Administration Schemas

Information

No object selected

SQL File 6' x

```
97 /*
98 4. Subqueries and Correlated Subqueries
99 Use a subquery to find patients who have received treatments from a specific department (provided DepartmentID).
100 */
101 select distinct p.*
102 from patients p
103 join treatments t on p.patientid = t.patientid
104 join staff s on t.staffid = s.staffid
105 where s.departmentid = 1;
106
107 -- Use a correlated subquery to find staff who have treated more than a specified number of patients (e.g., 3).
108 select *
109 from staff s
110 where (select count(*) from treatments t
111 where t.staffid = s.staffid) > 2;
```

Result Grid

staffid	name	role	departmentid
2	dr. Jones	neurologist	2
3	dr. Brown	orthopedic surgeon	3

staff 28 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	20:46:24	select * from staff s where (select count(*) from treatments t where t.staffid = s.staffid) > 2;	2 row(s) returned	0.000 sec / 0.000 sec

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Administration Schemas

Information

No object selected

SQL File 6' x

```
108 select *
109 from staff s
110 where (select count(*) from treatments t
111 where t.staffid = s.staffid) > 2;
112
113 /*
114 5. Aggregate Functions with GROUP BY and HAVING [5 Marks]
115 Calculate the total cost of treatments per patient and filter out patients
116 with a total treatment cost of less than Rs.1000
117 */
118 select p.name as patient_name, sum(t.cost) as total_cost
119 from patients p
120 join treatments t on p.patientid = t.patientid
121 group by p.name
122 having sum(t.cost) >= 1000;
```

Result Grid

patient_name	total_cost
john doe	1100.00
jane smith	2300.00
alice johnson	1700.00
eve white	1850.00

Result 30 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	20:46:53	select p.name as patient_name, sum(t.cost) as total_cost from patients p join treatments t on p.patientid = t.patientid group by p.name having sum(t.cost) >= 1000;	4 row(s) returned	0.000 sec / 0.000 sec

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

## Oracle Database XE 21c Format: -

-- Create Tables: patients, departments, staff, treatments

-- Patients table

```
create table patients (  
    patientid number primary key,  
    name varchar2(100),  
    dob date,  
    email varchar2(100) unique,  
    phone varchar2(20)  
);
```

-- Departments table

```
create table departments (  
    departmentid number primary key,  
    departmentname varchar2(100)  
);
```

-- Staff table

```
create table staff (  
    staffid number primary key,  
    name varchar2(100),  
    role varchar2(50),  
    departmentid number,  
    constraint fk_staff_department foreign key (departmentid) references  
departments(departmentid)  
);
```

-- Treatments table

```
create table treatments (  
    treatmentid number primary key,  
    patientid number,  
    staffid number,  
    treatmentdate date,  
    cost number(10, 2),  
    constraint fk_treatments_patients foreign key (patientid) references  
patients(patientid) on delete cascade,  
    constraint fk_treatments_staff foreign key (staffid) references  
staff(staffid),  
    constraint chk_cost_nonnegative check (cost >= 0)  
);
```

```
-- Insert sample data into patients table
insert into patients values (1, 'john doe', to_date('1985-05-15', 'yyyy-mm-dd'), 'john.doe@example.com', '555-0101');
insert into patients values (2, 'jane smith', to_date('1990-06-20', 'yyyy-mm-dd'), 'jane.smith@example.com', '555-0102');
insert into patients values (3, 'alice johnson', to_date('1975-09-25', 'yyyy-mm-dd'), 'alice.j@example.com', '555-0103');
insert into patients values (4, 'bob brown', to_date('1982-01-30', 'yyyy-mm-dd'), 'bob.brown@example.com', '555-0104');
insert into patients values (5, 'eve white', to_date('1995-12-12', 'yyyy-mm-dd'), 'eve.white@example.com', '555-0105');

-- Insert sample data into departments table
insert into departments values (1, 'cardiology');
insert into departments values (2, 'neurology');
insert into departments values (3, 'orthopedics');

-- Insert sample data into staff table
insert into staff values (1, 'dr. smith', 'cardiologist', 1);
insert into staff values (2, 'dr. jones', 'neurologist', 2);
insert into staff values (3, 'dr. brown', 'orthopedic surgeon', 3);
insert into staff values (4, 'nurse kelly', 'nurse', 1);
insert into staff values (5, 'nurse taylor', 'nurse', 2);

-- Insert sample data into treatments table
insert into treatments values (1, 1, 1, to_date('2023-01-10', 'yyyy-mm-dd'), 500);
insert into treatments values (2, 2, 2, to_date('2023-02-15', 'yyyy-mm-dd'), 1500);
insert into treatments values (3, 3, 3, to_date('2023-03-20', 'yyyy-mm-dd'), 750);
insert into treatments values (4, 4, 1, to_date('2023-04-05', 'yyyy-mm-dd'), 200);
insert into treatments values (5, 5, 2, to_date('2023-05-30', 'yyyy-mm-dd'), 1200);
insert into treatments values (6, 1, 3, to_date('2023-06-10', 'yyyy-mm-dd'), 300);
insert into treatments values (7, 2, 1, to_date('2023-07-01', 'yyyy-mm-dd'), 800);
insert into treatments values (8, 3, 2, to_date('2023-07-15', 'yyyy-mm-dd'), 950);
insert into treatments values (9, 4, 1, to_date('2023-08-20', 'yyyy-mm-dd'), 1100);
```

```
insert into treatments values (10, 5, 3, to_date('2023-09-25', 'yyyy-mm-dd'),
650);
```

```
-- Update the cost of a specific treatment
update treatments set cost = 600 where treatmentid = 6;
```

```
-- Delete a patient and observe the cascading effect on associated treatments
-- Note: Ensure ON DELETE CASCADE is enabled on the foreign key constraint
(already defined above)
```

```
-- Delete patient with patientid = 4
delete from patients where patientid = 4;
```

```
-- Verify the effect on treatments
select * from treatments;
```

```
/*
```

Data Retrieval using different operations

Write a query to list all patients, their associated treatments, and the staff who provided those treatments.

```
*/
```

```
select p.name as patient_name, t.treatmentdate, s.name as staff_name
from patients p
join treatments t on p.patientid = t.patientid
join staff s on t.staffid = s.staffid;
```

```
-- Create a query to display all treatments, including patient names, staff
names, and department names
```

```
select p.name as patient_name, s.name as staff_name, d.departmentname,
t.treatmentdate, t.cost
from treatments t
join patients p on t.patientid = p.patientid
join staff s on t.staffid = s.staffid
join departments d on s.departmentid = d.departmentid;
```

```
/*
```

Subqueries and Correlated Subqueries

Use a subquery to find patients who have received treatments from a specific department (provided DepartmentID).

```
*/
```

```
select distinct p.*
from patients p
join treatments t on p.patientid = t.patientid
```

```
join staff s on t.staffid = s.staffid  
where s.departmentid = 1;
```

```
-- Use a correlated subquery to find staff who have treated more than a  
specified number of patients (e.g., 3)
```

```
select *  
from staff s  
where (select count(*) from treatments t where t.staffid = s.staffid) > 2;
```

```
/*
```

Aggregate Functions with GROUP BY and HAVING

Calculate the total cost of treatments per patient and filter out patients  
with a total treatment cost less than 1000

```
*/
```

```
select p.name as patient_name, sum(t.cost) as total_cost  
from patients p  
join treatments t on p.patientid = t.patientid  
group by p.name  
having sum(t.cost) >= 1000;
```