

AN MYSQL PROJECT REPORT

Healthcare Analytics with SQL

Project work done by
Thaslima Banu. S

Data Analytics

1. Write this query to create a database:

`CREATE database Healthcare_Dataset;`

`use Healthcare_Dataset;`

2. Right click on the tables – Select – Table data Import Wizard to import the data from excel one by one.
3. Assign Primary Key(PK) and Foreign Key(FK) after importing table – Right click the imported table – Alter Table.

Tasks:

-- 1. Inner and Equi Joins

```
select p.name as patient_name, d.name as doctor_name, d.specialization, a.appointment_date, a.status
from healthcare_dataset.appointments a
inner join healthcare_dataset.patients p on a.patient_id = p.patient_id
inner join healthcare_dataset.doctors d on a.doctor_id = d.doctor_id
where a.status = "Completed";
```

```
-- 1. Inner and Equi Joins
select p.name as patient_name, d.name as doctor_name, d.specialization, a.appointment_date, a.status
from healthcare_dataset.appointments a
inner join healthcare_dataset.patients p on a.patient_id = p.patient_id
inner join healthcare_dataset.doctors d on a.doctor_id = d.doctor_id
where a.status = "Completed";
```

-- 2. Left Join with Null Handling

```
select p.name, p.contact_number, p.address
from healthcare_dataset.patients p
left join healthcare_dataset.appointments a on p.patient_id = a.patient_id
where a.appointment_id is null;
```

```
-- 2. Left Join with Null Handling
select p.name, p.contact_number, p.address
from healthcare_dataset.patients p
left join healthcare_dataset.appointments a on p.patient_id = a.patient_id
where a.appointment_id is null;
```

-- 3. Right Join and Aggregate Functions

```
select d.name as Doctor_name, d.specialization,  
count(di.diagnosis_id) as Total_Diagnoses  
from healthcare_dataset.diagnoses di  
right join healthcare_dataset.doctors d on di.doctor_id = d.doctor_id  
group by d.name, d.specialization;
```

Doctor_name	specialization	Total_Diagnoses
Doctor_1	Orthopedics	48
Doctor_2	Pediatrics	49
Doctor_3	Cardiology	59
Doctor_4	Pediatrics	60
Doctor_5	Cardiology	58
Doctor_6	Orthopedics	57
Doctor_7	Cardiology	56
Doctor_8	General Medicine	46
Doctor_9	Pediatrics	47
Doctor_10	Orthopedics	53
Doctor_11	Orthopedics	53
Doctor_12	Pediatrics	46
Doctor_13	Orthopedics	39
Doctor_14	General Medicine	43

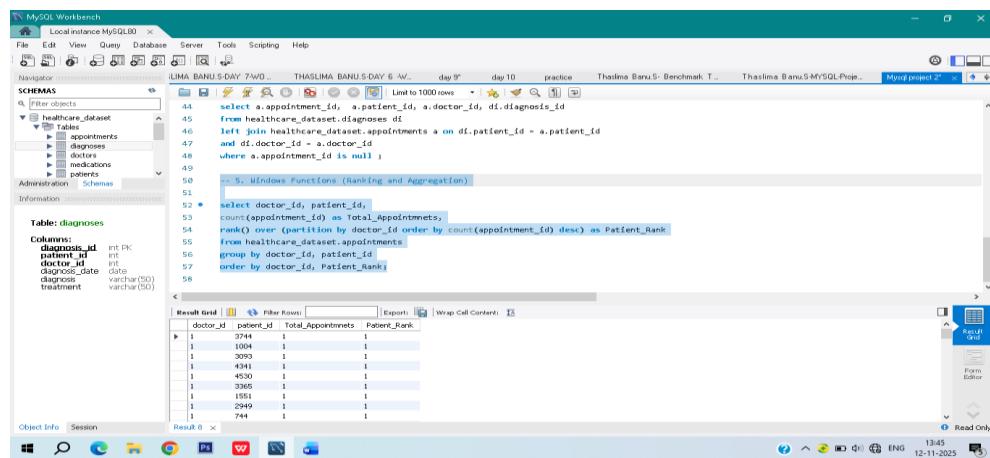
-- 4. Full Join for Overlapping Data

```
select a.appointment_id, a.patient_id, a.doctor_id, di.diagnosis_id  
from healthcare_dataset.appointments a  
left join healthcare_dataset.diagnoses di on a.patient_id = di.patient_id  
and a.doctor_id = di.doctor_id  
where di.diagnosis_id is null  
union  
select a.appointment_id, a.patient_id, a.doctor_id, di.diagnosis_id  
from healthcare_dataset.diagnoses di  
left join healthcare_dataset.appointments a on di.patient_id = a.patient_id  
and di.doctor_id = a.doctor_id  
where a.appointment_id is null ;
```

appointment_id	patient_id	doctor_id	diagnosis_id
1	4219	5	_____
2	2182	202	_____
3	1643	202	_____
4	3089	179	_____
5	841	57	_____
6	2655	186	_____
7	4119	92	_____
8	566	292	_____
9	965	227	_____

-- 5. Windows Functions (Ranking and Aggregation)

```
select doctor_id, patient_id,  
count(appointment_id) as Total_Appointments,  
rank() over (partition by doctor_id order by count(appointment_id) desc) as Patient_Rank  
from healthcare_dataset.appointments  
group by doctor_id, patient_id  
order by doctor_id, Patient_Rank;
```

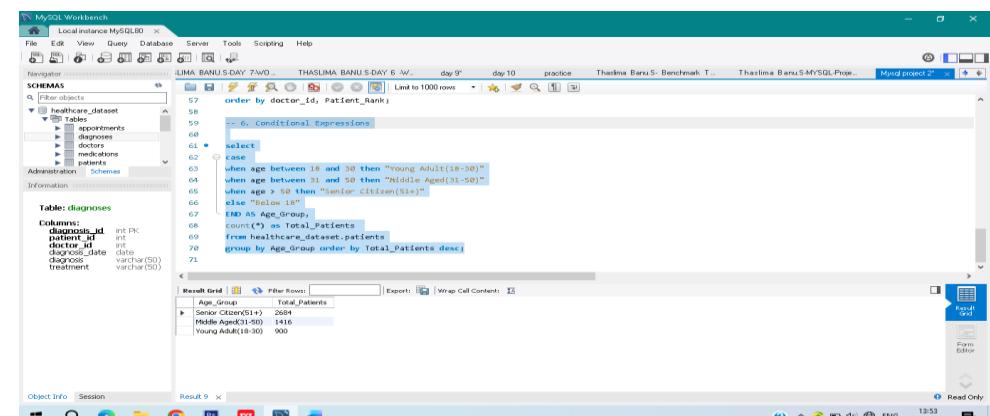


```
51 -- 5. Windows Functions (Ranking and Aggregation)  
52 select doctor_id, patient_id,  
53 count(appointment_id) as Total_Appointments,  
54 rank() over (partition by doctor_id order by count(appointment_id) desc) as Patient_Rank  
55 from healthcare_dataset.appointments  
56 group by doctor_id, patient_id  
57 order by doctor_id, Patient_Rank  
58
```

doctor_id	patient_id	Total_Appointments	Patient_Rank
1	3744	1	1
1	1004	1	1
1	3093	1	1
1	4941	1	1
1	4530	1	1
1	3595	1	1
1	1551	1	1
1	2949	1	1
1	744	1	1

-- 6. Conditional Expressions

```
select  
case  
when age between 18 and 30 then "Young Adult(18-30)"  
when age between 31 and 50 then "Middle Aged(31-50)"  
when age > 50 then "Senior Citizen(51+)"  
else "Below 18"  
END AS Age_Group,  
count(*) as Total_Patients  
from healthcare_dataset.patients  
group by Age_Group order by Total_Patients desc;
```



```
57 -- 6. Conditional Expressions  
58  
59  
60 select  
61 case  
62 when age between 18 and 30 then "Young Adult(18-30)"  
63 when age between 31 and 50 then "Middle Aged(31-50)"  
64 when age > 50 then "Senior Citizen(51+)"  
65 else "Below 18"  
66 END as Age_Group,  
67 count(*) as Total_Patients  
68 from healthcare_dataset.patients  
69 group by Age_Group order by Total_Patients desc  
70  
71
```

Age_Group	Total_Patients
Senior Citizen(51+)	260
Middle Aged(31-50)	1416
Young Adult(18-30)	900

-- 7. Numeric and String Functions

select

```
upper(NAME) as Patient_Name_Upper, contact_number  
from healthcare_dataset.patients  
where contact_number like "%1234";
```

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

```
upper(NAME) as Patient_Name_Upper, contact_number  
from healthcare_dataset.patients  
where contact_number like "%1234";
```

The results grid shows one row:

Patient_Name_Upper	contact_number
PATIENT_1234	98765431234

-- 8. Subqueries for Filtering

```
select p.patient_id, p.name  
from healthcare_dataset.patients p  
where p.patient_id in(  
select d.patient_id  
from healthcare_dataset.diagnoses d  
where d.treatment = "Insulin"  
group by d.patient_id having count(distinct d.treatment) = 1);
```

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

```
select p.patient_id, p.name  
from healthcare_dataset.patients p  
where p.patient_id in(  
select d.patient_id  
from healthcare_dataset.diagnoses d  
where d.treatment = "Insulin"  
group by d.patient_id having count(distinct d.treatment) = 1);
```

The results grid shows no rows.

Based on the Dataset none of the patients received “Insulin” as the sole treatment.

-- 9. Date and Time Functions

```
select diagnosis_id,  
avg(datediff(end_date, start_date)) as Avg_Medication_Duration  
from healthcare_dataset.medications group by diagnosis_id;
```

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

LIMA_BANU-S-7WD... THASLIMA_BANU-S-6... day 9* day 10 practice Thaslima Banu.S-Benchmark T... Thaslima Banu.S-MYSQL-Proje... MySQL project

Tables

healthcare_dataset

appointments

diagnoses

doctors

medications

patients

Information Schema Schemas

Table: appointments

Columns:

appointment_id	int	PK
patient_id	int	FK
doctor_id	int	FK
appointment_date	date	
status	varchar(50)	

101 select p.patient_id, p.name
102 from healthcare_dataset.patients p
103 where patient_id in
104 select d.patient_id
105 from healthcare_dataset.diagnoses d
106 where d.treatment = "Insulin"
107 group by d.patient_id having count(distinct d.treatment) > 1);
108
109 -- 9. Date and Time Functions
110
111 select diagnosis_id,
112 avg(datediff(end_date, start_date)) as Avg_Medication_Duration
113 from healthcare_dataset.medications
114 group by diagnosis_id;

Result Grid Filter Rows! Export! Wrap Call Content! Patch Rows!

diagnosis_id	Avg_Medication_Duration
1	-246.3333
4	372.0000
6	632.0000
8	439.0000
9	-77.0000
11	-34.0000
12	-134.0000
13	-216.0000
14	-199.0000

Object Info Session Result 15 x Read Only

-- 10. Complex Joins and Aggregation

```
select d.doctor_id, d.name, d.specialization,  
count(distinct a.patient_id) as Unique_Patients  
from healthcare_dataset.doctors d  
join healthcare_dataset.appointments a on d.doctor_id = a.doctor_id  
group by d.doctor_id, d.name, d.specialization  
order by Unique_Patients desc limit 1;
```

The screenshot shows the MySQL Workbench interface with a query editor displaying a complex SQL script. The script includes joins, aggregation, and ordering. The results of the query are shown in a table below, with one row returned:

doctor_id	name	specialization	Unique_Patients
37	Doctor_37	General Medicine	51

The MySQL Workbench interface includes a Navigator pane on the left showing the schema structure, and a Results pane on the right for viewing the output.