PATIENT RECORD MANAGEMENT SYSTEM

- ADITYA SATISH SAPKAL.
BATCH- T341 / DS

Patient Record Management System

A **Patient Record Management System** is a project that simplifies and automates the tasks of managing a hospital's data and services. This system ensures the efficient handling of healthcare operations such as recording patient details, managing doctor assignments, and tracking appointment statuses.

This project demonstrates the implementation of a Patient Record Management System using **MySQL**. It includes creating and managing normalized tables, performing database operations, and executing advanced SQL queries. The goal is to showcase skills in **database design**, **data manipulation**, and **querying** in a real-world healthcare scenario.

PROJECT AIM

• **Patient Management**: Add, update, and remove patient records. Track patient details such as name, gender, age, and unique ID.
• **Doctor Management**: Maintain information about doctors, including their names, specializations, and assigned departments.
• **Appointment Management**: Record and monitor patient appointments, including date, disease, attending doctor, and appointment status (Attended, Missed, Cancelled).
• **Department Management**: Manage hospital departments, including department names and the doctors assigned to them.
• **Healthcare Analytics**: Analyze patient data, doctor workload, and department-wise performance using SQL queries for effective decision-making.

OBJECTIVES

Set up the Patient Record Management System Database:
Create and populate the database with tables for patients, doctors, departments, and appointments.

CRUD Operations:
Perform Create, Read, Update, and Delete operations on healthcare data to ensure accurate and up-to-date records.

Advanced SQL Queries:
Develop complex SQL queries to analyse patient visits, doctor workload, department performance, and appointment trends.

ER Diagram for Patient Record Management System

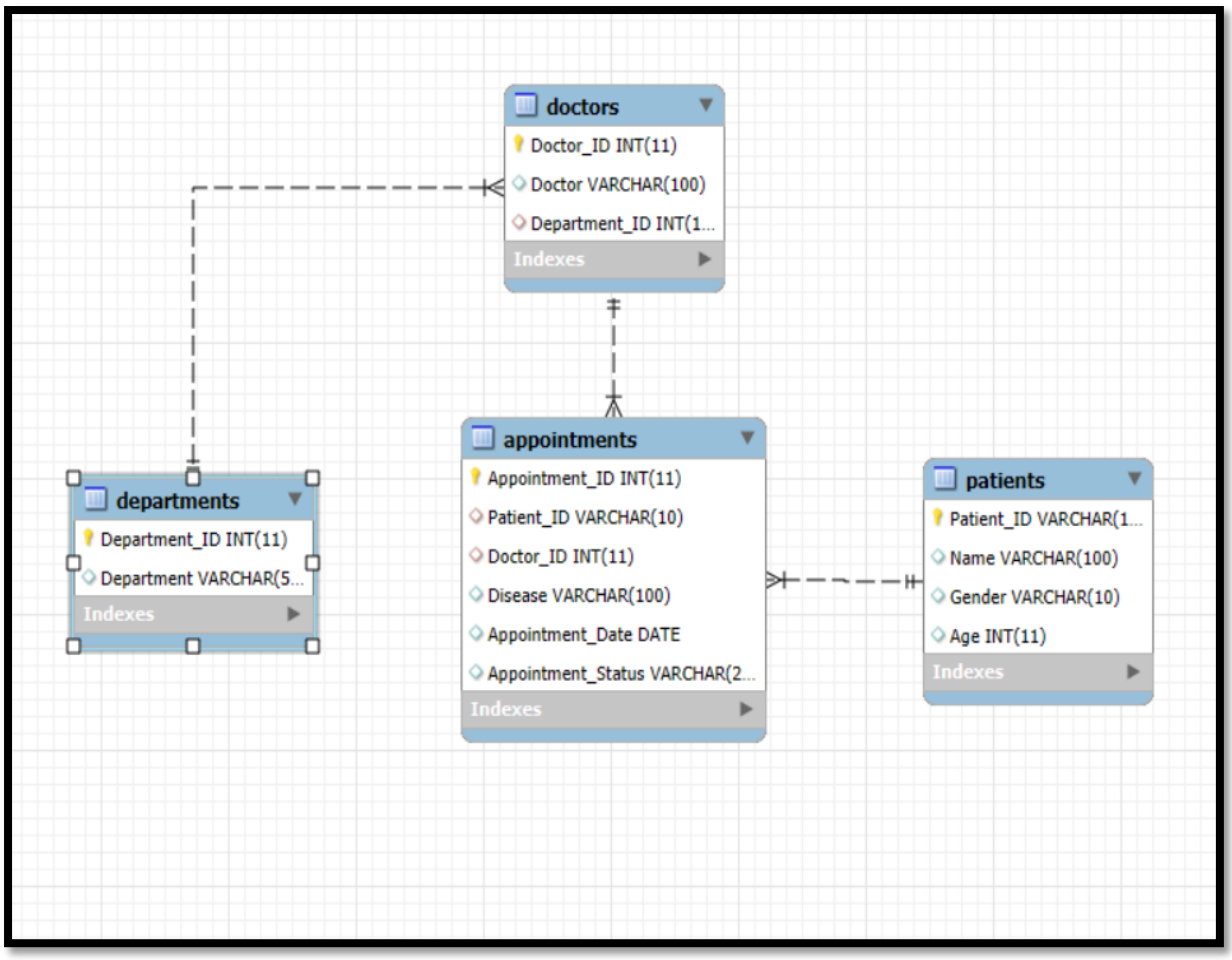


Table Description:

1) Patients

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

Field	Type	Null	Key	Default	Extra
Patient_ID	varchar(10)	NO	PRI	NULL	
Name	varchar(100)	YES		NULL	
Gender	varchar(10)	YES		NULL	
Age	int(11)	YES		NULL	

2) Departments

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

Field	Type	Null	Key	Default	Extra
Department_ID	int(11)	NO	PRI	NULL	
Department	varchar(50)	YES		NULL	

3) Doctors

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

Field	Type	Null	Key	Default	Extra
Doctor_ID	int(11)	NO	PRI	NULL	
Doctor	varchar(100)	YES		NULL	
Department_ID	int(11)	YES	MUL	NULL	

4) Appointments

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

Field	Type	Null	Key	Default	Extra
Appointment_ID	int(11)	NO	PRI	NULL	auto_increment
Patient_ID	varchar(10)	YES	MUL	NULL	
Doctor_ID	int(11)	YES	MUL	NULL	
Disease	varchar(100)	YES		NULL	
Appointment_Date	date	YES		NULL	
Appointment_Status	varchar(20)	YES		NULL	

CREATING DATABASE:

CREATE DATABASE healthcare_project;

USE healthcare_project;

Table Creation & Insertion Commands:

1) **Create Table Patients**

CREATE TABLE patients

(Patient_ID VARCHAR (10) PRIMARY KEY,

Name VARCHAR (100),

Gender VARCHAR (10),

Age INT);

Inserting Values into Patients:

INSERT INTO patients (Patient_ID, Name, Gender, Age) VALUES

('P0001', 'Brian Yang', 'Female', 32),

('P0002', 'Jonathan Johnson', 'Male', 76),

('P0003', 'Donald Booth', 'Male', 30), …..

Select*from patients;

OUTPUT:

Patient_ID	Name	Gender	Age
P0001	Brian Yang	Female	32
P0002	Jonathan Johnson	Male	76
P0003	Donald Booth	Male	30
P0004	Tyler Rogers	Male	84
P0005	Andrew Stevens	Female	76
P0006	Lisa Hensley	Female	44

2) **Create Table Departments**

CREATE TABLE departments

(Department_ID INT PRIMARY KEY,

Department VARCHAR (50));

Inserting Values into Departments:

INSERT INTO departments (Department_ID, Department) VALUES

(1, 'Cardiology'),

(2, 'Pediatrics'),

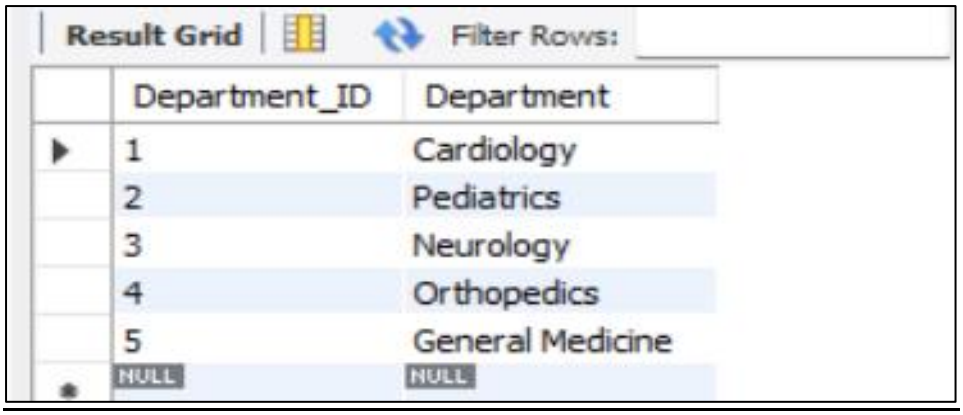
(3, 'Neurology'),

(4, 'Orthopedics'),

(5, 'General Medicine');

Select*from departments;

OUTPUT:



Department_ID	Department
1	Cardiology
2	Pediatrics
3	Neurology
4	Orthopedics
5	General Medicine
NULL	NULL

3) **Create Table Doctors**

CREATE TABLE doctors

(Doctor_ID INT PRIMARY KEY,

Doctor VARCHAR (100),

Department_ID INT,

FOREIGN KEY (Department_ID) REFERENCES

departments (Department_ID));

Inserting Values into doctors:

INSERT INTO (Doctor_ID, Doctor, Department_ID) VALUES

(1, 'Patrick Sanchez', 1),

(2, 'Javier Johnson', 1),

(3, 'Meredith Barnes', 1),

(4, 'Melissa Peterson', 1),

(5, 'Ian Cooper', 2),

(6, 'Jamie Arnold', 1), ….

Select*from doctors;

OUTPUT:

Result Grid | Filter Rows:

Doctor_ID	Doctor	Department_ID
1	Patrick Sanchez	1
2	Javier Johnson	1
3	Meredith Barnes	1
4	Melissa Peterson	1
5	Ian Cooper	2
6	Jamie Arnold	1

4) **Create Table Appointments**

CREATE TABLE appointments

(Appointment_ID INT AUTO_INCREMENT PRIMARY KEY,

Patient_ID VARCHAR (10),

Doctor_ID INT,

Disease VARCHAR (100),

Appointment_Date DATE,

Appointment_Status VARCHAR (20),

FOREIGN KEY (Patient_ID) REFERENCES patients (Patient_ID),

FOREIGN KEY (Doctor_ID) REFERENCES doctors (Doctor_ID));

Inserting Values into appointments:

INSERT INTO appointments (Patient_ID, Doctor_ID,

Disease, Appointment_Date, Appointment_Status) VALUES

('P0001', 1, 'Heart Attack', '2025-03-08', 'Attended'),

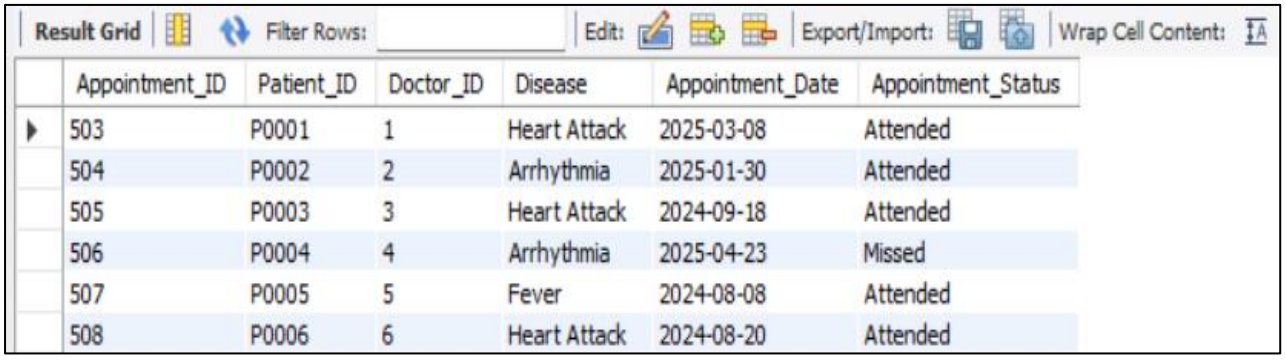
('P0002', 2, 'Arrhythmia', '2025-01-30', 'Attended'),

('P0003', 3, 'Heart Attack', '2024-09-18', 'Attended'),

('P0004', 4, 'Arrhythmia', '2025-04-23', 'Missed'), ….

Select*from appointments;

OUTPUT:



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

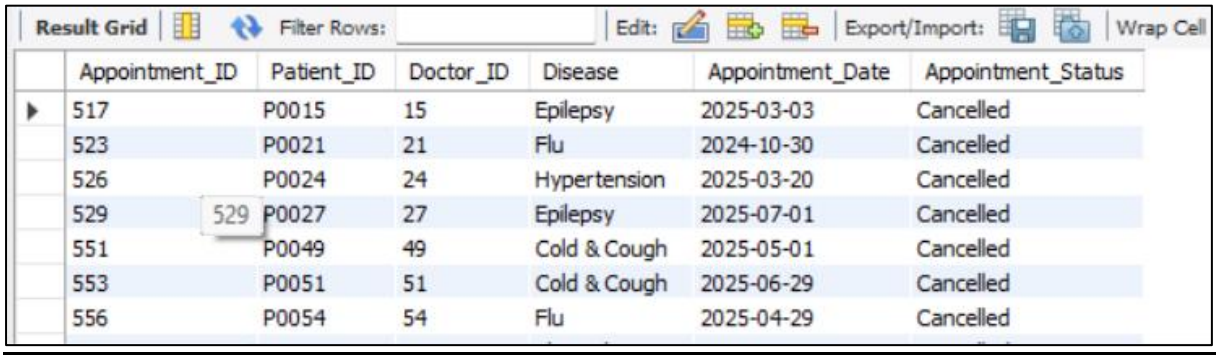
Appointment_ID	Patient_ID	Doctor_ID	Disease	Appointment_Date	Appointment_Status
503	P0001	1	Heart Attack	2025-03-08	Attended
504	P0002	2	Arrhythmia	2025-01-30	Attended
505	P0003	3	Heart Attack	2024-09-18	Attended
506	P0004	4	Arrhythmia	2025-04-23	Missed
507	P0005	5	Fever	2024-08-08	Attended
508	P0006	6	Heart Attack	2024-08-20	Attended

BASIC QUESTIONS

1) Find appointments with status ‘Cancelled’.

SELECT * FROM appointments **WHERE** Appointment_Status = 'Cancelled';

OUTPUT:



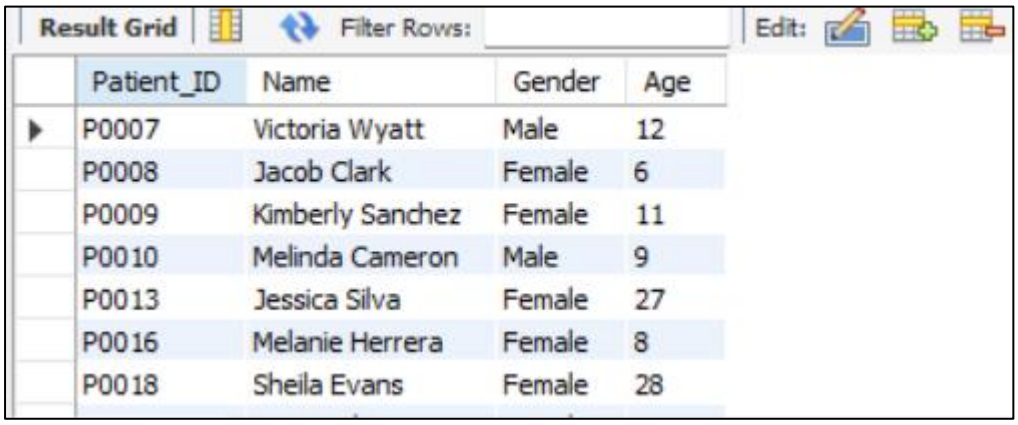
Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell

Appointment_ID	Patient_ID	Doctor_ID	Disease	Appointment_Date	Appointment_Status
517	P0015	15	Epilepsy	2025-03-03	Cancelled
523	P0021	21	Flu	2024-10-30	Cancelled
526	P0024	24	Hypertension	2025-03-20	Cancelled
529	P0027	27	Epilepsy	2025-07-01	Cancelled
551	P0049	49	Cold & Cough	2025-05-01	Cancelled
553	P0051	51	Cold & Cough	2025-06-29	Cancelled
556	P0054	54	Flu	2025-04-29	Cancelled

2) Show patients younger than 30.

SELECT * FROM patients **WHERE** Age < 30;

OUTPUT:



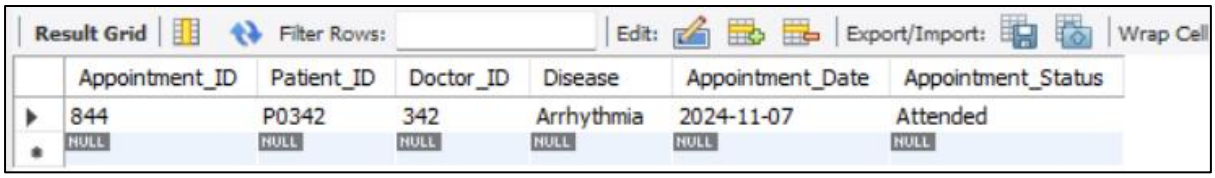
Result Grid | Filter Rows: | Edit:

Patient_ID	Name	Gender	Age
P0007	Victoria Wyatt	Male	12
P0008	Jacob Clark	Female	6
P0009	Kimberly Sanchez	Female	11
P0010	Melinda Cameron	Male	9
P0013	Jessica Silva	Female	27
P0016	Melanie Herrera	Female	8
P0018	Sheila Evans	Female	28

3) List appointments scheduled on a specific date.

SELECT * FROM appointments **WHERE** Appointment_Date = '2024-11-07';

OUTPUT:

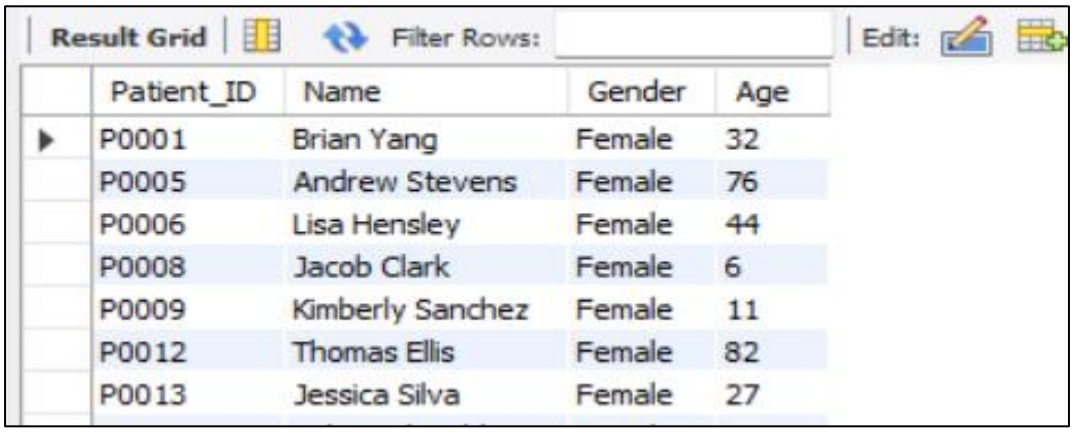


Appointment_ID	Patient_ID	Doctor_ID	Disease	Appointment_Date	Appointment_Status
844	P0342	342	Arrhythmia	2024-11-07	Attended
NULL	NULL	NULL	NULL	NULL	NULL

4) List all female patients.

SELECT * FROM patients **WHERE** Gender = "female" ;

OUTPUT:

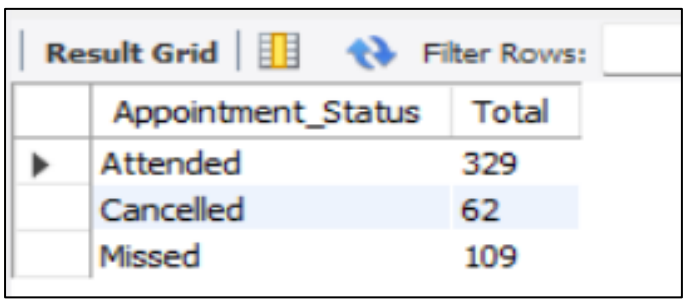


Patient_ID	Name	Gender	Age
P0001	Brian Yang	Female	32
P0005	Andrew Stevens	Female	76
P0006	Lisa Hensley	Female	44
P0008	Jacob Clark	Female	6
P0009	Kimberly Sanchez	Female	11
P0012	Thomas Ellis	Female	82
P0013	Jessica Silva	Female	27

5) Count of appointments by appointment status.

SELECT Appointment_Status, **COUNT** (*) AS Total **FROM** appointments **GROUP BY** Appointment_Status;

OUTPUT:

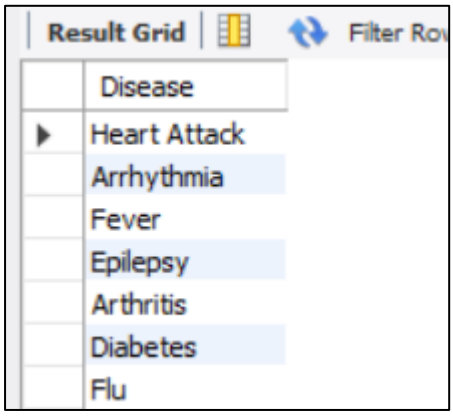


Appointment_Status	Total
Attended	329
Cancelled	62
Missed	109

6) List all diseases treated in the hospital.

SELECT distinct Disease **FROM** appointments;

OUTPUT:



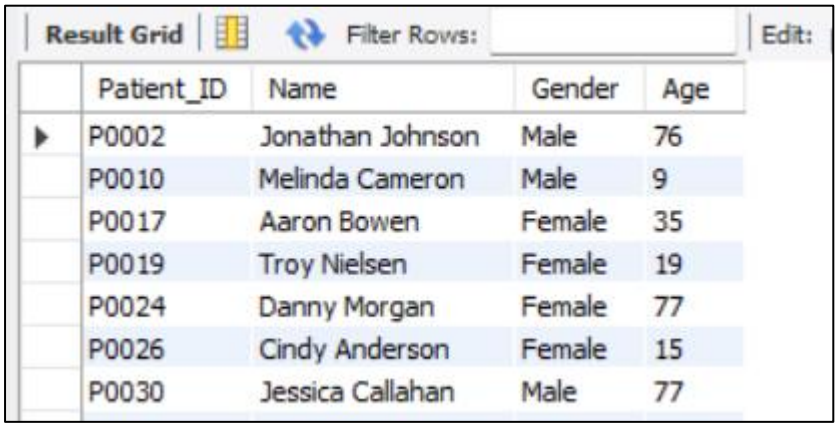
Result Grid | Filter Rov

Disease
Heart Attack
Arrhythmia
Fever
Epilepsy
Arthritis
Diabetes
Flu

7) Show all patients with names ending in 'n'.

SELECT * FROM patients **WHERE** Name **LIKE** '%n';

OUTPUT:



Result Grid | Filter Rows: | Edit:

Patient_ID	Name	Gender	Age
P0002	Jonathan Johnson	Male	76
P0010	Melinda Cameron	Male	9
P0017	Aaron Bowen	Female	35
P0019	Troy Nielsen	Female	19
P0024	Danny Morgan	Female	77
P0026	Cindy Anderson	Female	15
P0030	Jessica Callahan	Male	77

8) Find total appointments by status, but only include statuses starting with 'A' or 'M'.

SELECT Appointment_Status, **COUNT** (*) AS Total

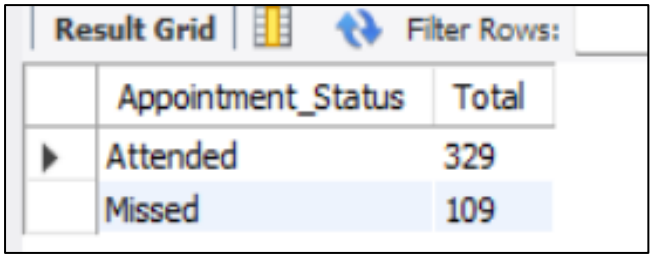
FROM appointments

WHERE Appointment_Status **LIKE** 'A%' OR Appointment_Status **LIKE** 'M%'

GROUP BY Appointment_Status

ORDER BY Total DESC;

OUTPUT:



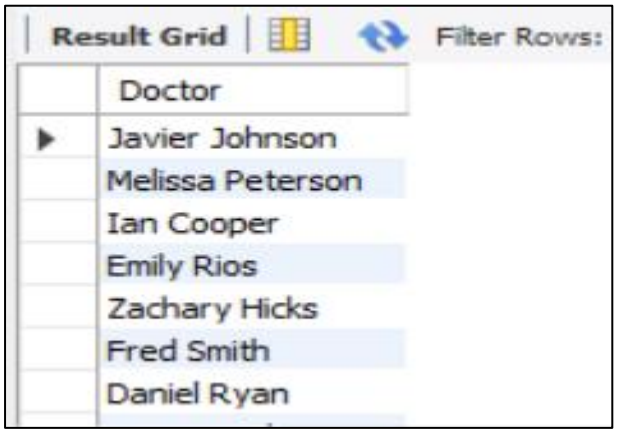
Appointment_Status	Total
Attended	329
Missed	109

SUB-QUERIES

1) List doctors who had appointments with patients over age 60.

```
SELECT Doctor FROM doctors WHERE Doctor_ID IN
(SELECT Doctor_ID FROM appointments a
 JOIN
 patients p ON  a.Patient_ID = p.Patient_ID
 WHERE p.Age > 60);
```

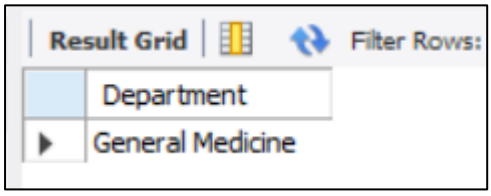
OUTPUT:



2) Find departments that have more than 100 doctors.

```
SELECT Department FROM departments WHERE Department_ID IN
(SELECT Department_ID FROM doctors
 GROUP BY Department_ID
 HAVING COUNT (*) > 100);
```

OUTPUT:



3) Show the most visited doctor with appointment count.

SELECT Doctor, Total_Appointments **FROM** (SELECT d.Doctor, **COUNT**(a.Appointment_ID) AS Total_Appointments

FROM appointments a

JOIN doctors d

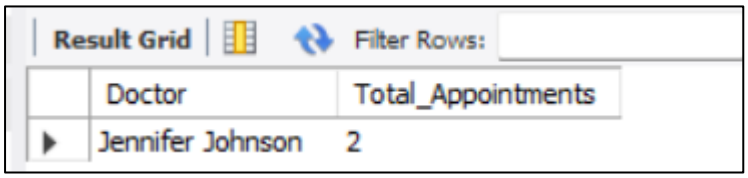
ON a.Doctor_ID = d.Doctor_ID

GROUP BY d.Doctor

ORDER BY Total_Appointments DESC

LIMIT 1) AS Most_Visited;

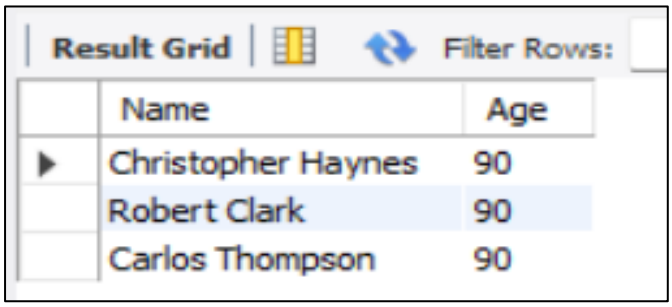
OUTPUT:



4) List the oldest patients in the system.

SELECT Name, Age **FROM** patients **WHERE** Age = (SELECT MAX(Age) **FROM** patients);

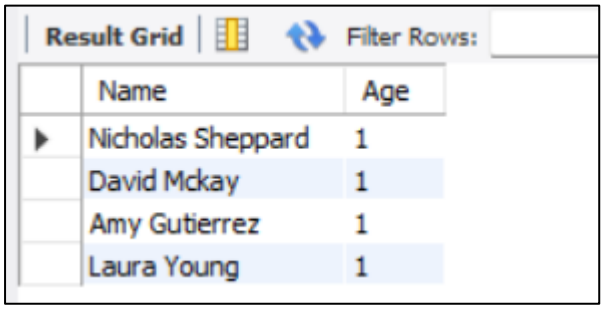
OUTPUT:



5) Show the youngest patient who missed an appointment.

```
SELECT Name, Age FROM patients WHERE Age =
 (SELECT MIN(Age) FROM patients
 WHERE Patient_ID IN
 (SELECT Patient_ID FROM appointments
  WHERE Appointment_Status = 'Missed'));
```

OUTPUT:


Result Grid | Filter Rows:

Name	Age
Nicholas Sheppard	1
David Mckay	1
Amy Gutierrez	1
Laura Young	1

JOINS

1) How many doctors are working in each department?

```
SELECT d.Department, COUNT (*) AS Total_Doctors
FROM doctors doc
JOIN departments d ON doc.Department_ID = d.Department_ID
GROUP BY d.Department;
```

OUTPUT:

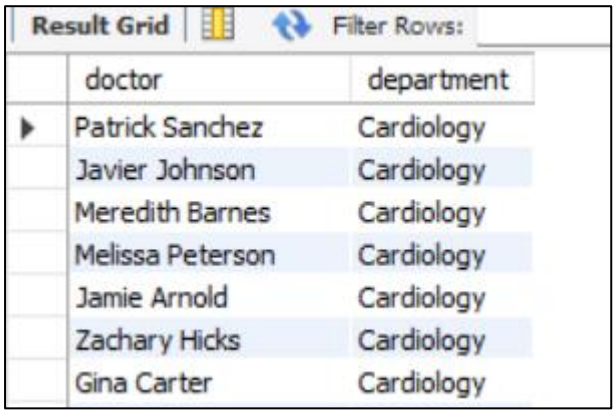
Result Grid | Filter Rows:

Department	Total_Doctors
Cardiology	99
General Medicine	117
Neurology	90
Orthopedics	96
Pediatrics	98

2) Find all doctors in the ‘Cardiology’ department.

```
SELECT doctor, department   FROM
doctors as doc
JOIN
departments as d ON
d.Department_ID = doc.Department_ID
WHERE d. Department = "Cardiology";
```

OUTPUT:



Result Grid | Filter Rows:

doctor	department
Patrick Sanchez	Cardiology
Javier Johnson	Cardiology
Meredith Barnes	Cardiology
Melissa Peterson	Cardiology
Jamie Arnold	Cardiology
Zachary Hicks	Cardiology
Gina Carter	Cardiology

3) List names of patients and the doctors they visited.

```
SELECT p.Name AS Patient_Name, doc.Doctor AS Doctor_Name
FROM
appointments a
JOIN
patients p ON
a.Patient_ID = p.Patient_ID
JOIN
 doctors doc ON
 a.Doctor_ID = doc.Doctor_ID;
```

OUTPUT:

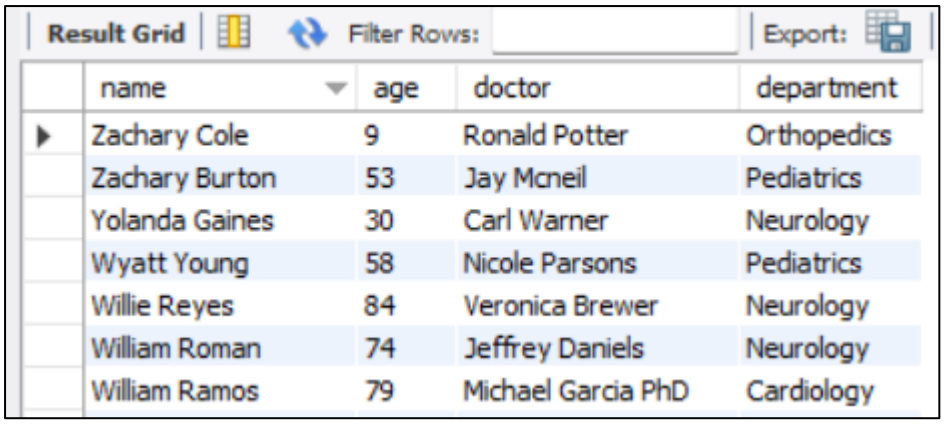
Result Grid | Filter Rows:

Patient_Name	Doctor_Name
Brian Yang	Patrick Sanchez
Jonathan Johnson	Javier Johnson
Donald Booth	Meredith Barnes
Tyler Rogers	Melissa Peterson
Andrew Stevens	Ian Cooper
Lisa Hensley	Jamie Arnold
Victoria Wyatt	Benjamin Stanley

4) Show patient name, age, department, and doctor they visited.

```
SELECT p.name , p.age , doc.doctor ,d.department
FROM
appointments A
JOIN
patients as p ON
p.Patient_ID = A.Patient_ID
JOIN
doctors as doc ON
doc.Doctor_ID = A.Doctor_ID
JOIN
departments as d ON
d.Department_ID = doc.Department_ID;
```

OUTPUT:



name	age	doctor	department
Zachary Cole	9	Ronald Potter	Orthopedics
Zachary Burton	53	Jay Mcneil	Pediatrics
Yolanda Gaines	30	Carl Warner	Neurology
Wyatt Young	58	Nicole Parsons	Pediatrics
Willie Reyes	84	Veronica Brewer	Neurology
William Roman	74	Jeffrey Daniels	Neurology
William Ramos	79	Michael Garcia PhD	Cardiology

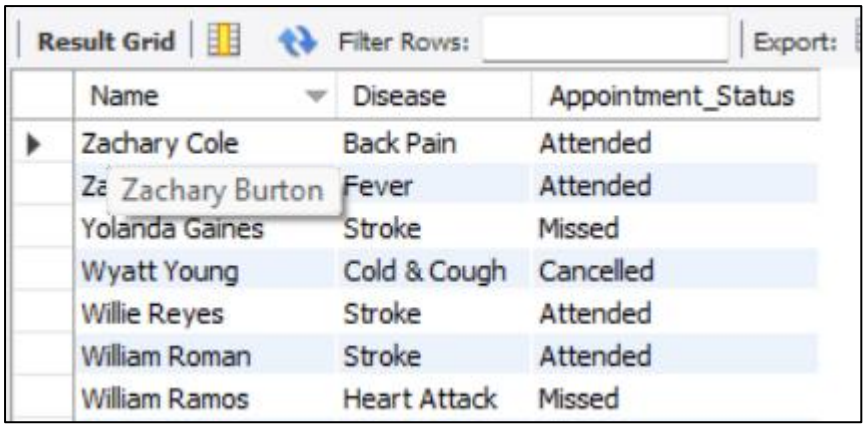
5) List patients with their appointment status and disease

SELECT p.Name, a.Disease, a.Appointment_Status

FROM appointments a

JOIN patients p **ON** a.Patient_ID = p.Patient_ID;

OUTPUT:



CONCLUSION

The Patient Record Management System database project has successfully designed and implemented a comprehensive database system that effectively manages and stores information related to hospital operations.

The system has achieved its objectives by organizing patient, doctor, department, and appointment data, improving data accuracy and accessibility. With its robust and scalable design, the system is well-positioned to support future growth and integration in healthcare record management.

This project demonstrates the application of SQL skills in creating and managing a healthcare management system. It includes database setup, data manipulation, and advanced querying, providing a solid foundation for real-world data handling and medical data analysis.