SQL QUERIES

FOR

PATIENT MANAGEMENT SYSTEM

(NHS)

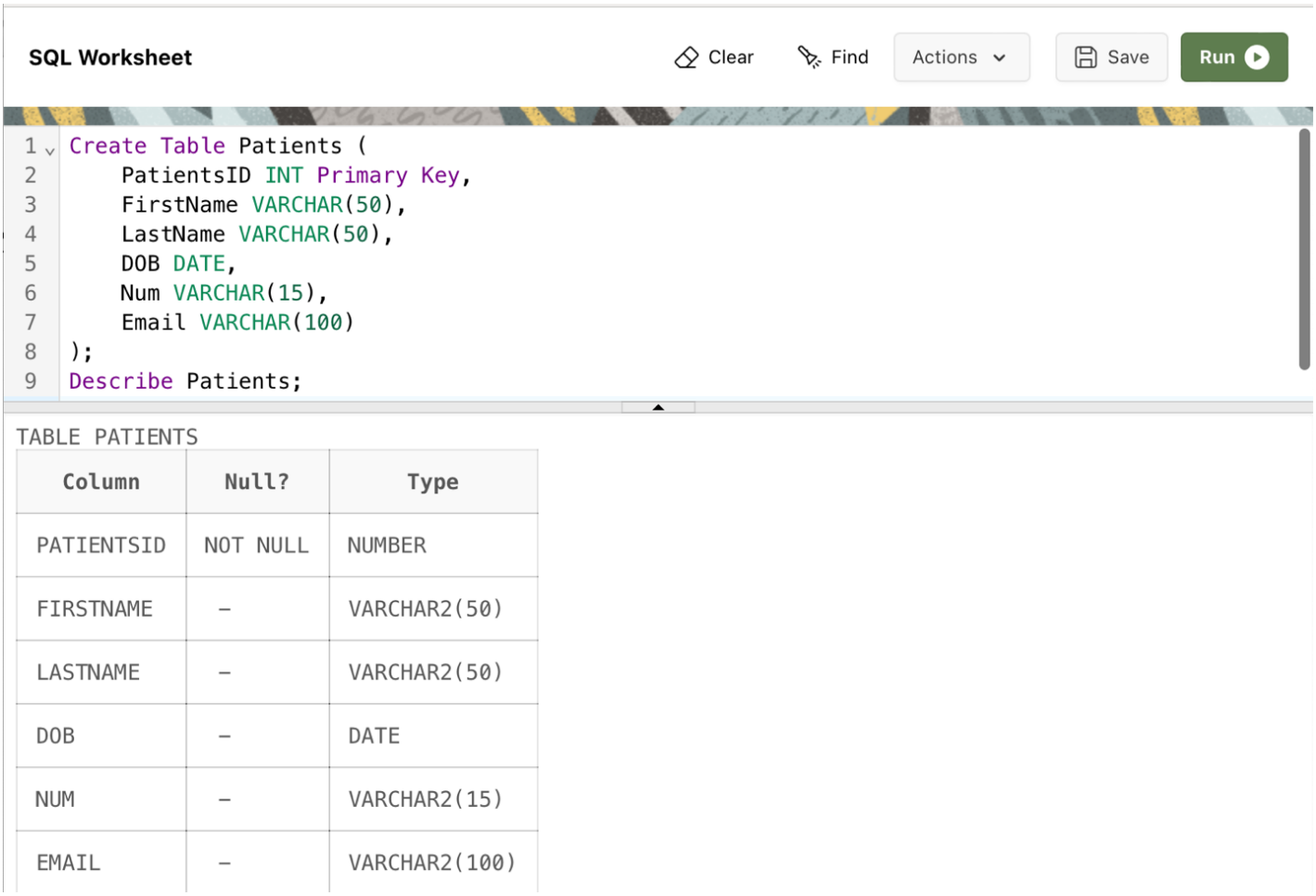
In this project, I developed a **Patient Management System** designed to streamline the management of patient information, billing, and healthcare services within an NHS context. The system enables healthcare providers to efficiently store, update, and retrieve patient data while ensuring secure handling of sensitive information.

To achieve this, I implemented a series of **SQL queries** that perform essential database operations such as:

- **Creating tables** for storing patient, appointment, and billing information.
- **Inserting data** to add new patients, services, and transaction records.
- **Updating records** to reflect changes in patient details or billing status.
- **Deleting records** when necessary, such as removing outdated or incorrect entries.
- **Retrieving data** using queries to fetch patient information, appointments, and billing history for real-time access by healthcare staff.

This system ensures compliance with NHS data management protocols while enhancing the efficiency of patient care and administrative processes.

⇒ QUERY 1




```
Create Table Patients (
    PatientsID INT Primary Key,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    DOB DATE,
    Num VARCHAR(15),
    Email VARCHAR(100)
);
```

Explanation:

Purpose: This query creates a table named Patients to store essential patient information.

Columns:
PatientsID: Unique identifier for each patient (Primary Key)
FirstName: Patient's first name (up to 50 characters)
LastName: Patient's last name (up to 50 characters)
DOB: Patient's date of birth
Num: Patient's contact number (up to 15 characters)
Email: Patient's email address (up to 100 characters)

Describe Patients;

Purpose: This command displays the schema of the Patients table.

⇒ QUERY 2

SQL Worksheet | Clear | Find | Actions | Save | Run

```
10  Create Table Billing (
11      BillingID INT Primary Key,
12      PatientID INT,
13      ServiceDate DATE,
14      ServiceDescription VARCHAR (250),
15      Amount DECIMAL(10,2),
16      Status VARCHAR(20),
17      Foreign Key (PatientID) References Patients(PatientsID)
18  );
19  Describe Billing;
```

TABLE BILLING

Column	Null?	Type
BILLINGID	NOT NULL	NUMBER
PATIENTID	–	NUMBER
SERVICEDATE	–	DATE
SERVICEDESCRIPTION	–	VARCHAR2(250)
AMOUNT	–	NUMBER(10,2)
STATUS	–	VARCHAR2(20)

Create Table Billing (
BillingID INT Primary Key,
PatientID INT,
ServiceDate DATE,
ServiceDescription VARCHAR (250),
Amount DECIMAL(10,2),
Status VARCHAR(20),
Foreign Key (PatientID) References Patients(PatientsID)
);

Explanation:

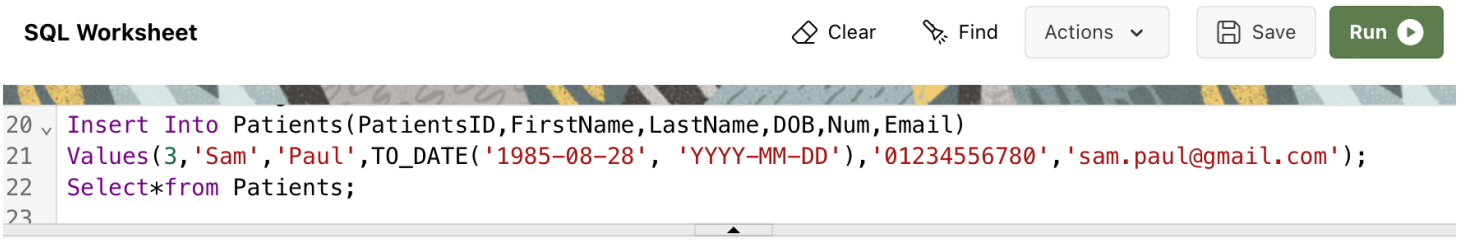
Purpose: This query creates a Billing table to manage billing records for services rendered to patients.

Columns:
BillingID: Unique identifier for each billing record (Primary Key)
PatientID: Identifier linking the billing record to a specific patient (Foreign Key)
ServiceDate: Date the service was provided.
ServiceDescription: Description of the service rendered (up to 250 characters)
Amount: Total charge for the service (up to 10 digits, 2 decimal places)
Status: Current status of the billing (e.g., paid, pending) (up to 20 characters)

Describe Billing;

Purpose: This command displays the schema of the Billing table.

⇒ QUERY 3



PATIENTSID	FIRSTNAME	LASTNAME	DOB	NUM	EMAIL
1	John	Doey	29-SEP-80	01234567890	john.doey@gmail.com
3	Sam	Paul	28-AUG-85	01234556780	sam.paul@gmail.com
2	Samuel	Den	02-AUG-90	01234567780	samuel.den@gmail.com

Insert Into Patients (PatientsID,FirstName,LastName,DOB,Num,Email) Values (1,'John','Doey',TO_DATE('1980-09-29','YYYY-MM-DD'),'01234567890','john.doey@gmail.com')

Explanation:

Purpose: This query adds a new patient record to the Patients table.

Values Inserted:

PatientsID: 1 (Unique identifier for the patient)
FirstName: 'John' (First name of the patient)
LastName: 'Doey' (Last name of the patient)
DOB: Converts the string '1980-09-29' into a date format for the date of birth
Num: '01234567890' (Patient's contact number)
Email: 'john.doey@gmail.com' (Patient's email address)

This insert statement adds John Doey's record into the patient management system.

*Select * from Patients*

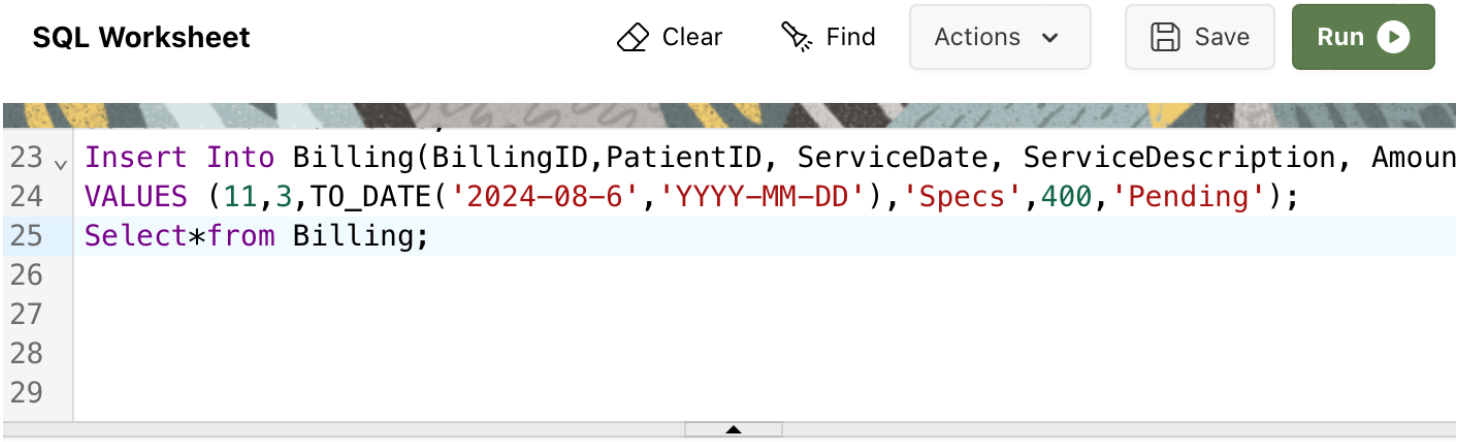
```
Select * from Patients
```

PATIENTSID	FIRSTNAME	LASTNAME	DOB	NUM	EMAIL
1	John	Doey	29-SEP-80	01234567890	john.doey@gmail.com
3	Sam	Paul	28-AUG-85	01234556780	sam.paul@gmail.com
2	Samuel	Den	02-AUG-90	01234567780	samuel.den@gmail.com

```
3 rows selected.
```

Purpose: This command selects all information from Patients table.

⇒ QUERY 4



BILLINGID	PATIENTID	SERVICEDATE	SERVICEDESCRIPTION	AMOUNT	STATUS
2	2	16-SEP-23	Consultation	400	Pending
21	2	16-SEP-23	Consultation	400	Pending
1	1	02-AUG-24	Consultation	150	Paid
11	3	06-AUG-24	Specs	400	Pending

Insert Into Billing(BillingID,PatientID, ServiceDate, ServiceDescription, Amount, Status)
VALUES (2, 2, TO_DATE('2023-09-16', 'YYYY-MM-DD'),'Consultation', 400, 'Pending')

Explanation:

Purpose: This query inserts a new billing record into the Billing table.

Values Inserted:

BillingID: 2 (Unique identifier for this billing record).
PatientID: 2 (References the patient with PatientsID 2).
ServiceDate: Converts the string '2023-09-16' into a date format, representing the date of the service.
ServiceDescription: 'Consultation' (Describes the service provided).
Amount: 400 (Total charge for the consultation).
Status: 'Pending' (The current status of the billing, indicating payment is pending).

This insert statement logs a billing entry for a consultation service.

*Select*from Billing*

```
Select*from Billing
```

BILLINGID	PATIENTID	SERVICEDATE	SERVICEDESCRIPTION	AMOUNT	STATUS
2	2	16-SEP-23	Consultation	400	Pending
21	2	16-SEP-23	Consultation	400	Pending
1	1	02-AUG-24	Consultation	150	Paid
11	3	06-AUG-24	Specs	400	Pending

```
4 rows selected.
```

Purpose: This command selects all information from Billing table.

⇒ QUERY 5

```
Select*from Billing
```

BILLINGID	PATIENTID	SERVICEDATE	SERVICEDESCRIPTION	AMOUNT	STATUS
2	2	16-SEP-23	Consultation	400	Pending
21	2	16-SEP-23	Consultation	400	Pending
1	1	02-AUG-24	Consultation	150	Paid
11	3	06-AUG-24	Specs	400	Pending

4 rows selected.

SQL Worksheet Clear Find Actions Save Run

```
26  DELETE FROM Billing
27  WHERE BillingID = 2;
28
```

BILLINGID	PATIENTID	SERVICEDATE	SERVICEDESCRIPTION	AMOUNT	STATUS
21	2	16-SEP-23	Consultation	400	Pending
1	1	02-AUG-24	Consultation	150	Paid
11	3	06-AUG-24	Specs	400	Pending

Download CSV

3 rows selected.

DELETE FROM Billing WHERE BillingID = 2

Explanation:

Purpose: This query removes a specific billing record from the Billing table.

Condition:
BillingID = 2: The query deletes the record where the BillingID is 2.

This delete statement is used to remove a billing entry associated with BillingID 2 from the system.

⇒ QUERY 6

```
SELECT p.FirstName, p.LastName, b.ServiceDate, b.ServiceDescription, b.Amount, b.Status
FROM Patients p
JOIN Billing b ON p.PatientsID = b.PatientID
WHERE p.PatientsID = 1
```

FIRSTNAME	LASTNAME	SERVICEDATE	SERVICEDESCRIPTION	AMOUNT	STATUS
John	Doey	02-AUG-24	Consultation	150	Paid

SELECT p.FirstName, p.LastName, b.ServiceDate, b.ServiceDescription, b.Amount, b.Status
FROM Patients p
JOIN Billing b ON p.PatientsID = b.PatientID
WHERE p.PatientsID = 1

Explanation:

Purpose: This query retrieves detailed billing information for a specific patient from both the Patients and Billing tables.

Columns Selected:
p.FirstName and p.LastName: The patient's first and last name.
b.ServiceDate: Date of the service provided.
b.ServiceDescription: Description of the service.
b.Amount: Amount charged for the service.
b.Status: Current billing status (e.g., 'Pending' or 'Paid').

Join Condition:
p.PatientsID = b.PatientID: Matches the patient's record with their billing information based on the PatientsID.

Filter Condition:
WHERE p.PatientsID = 1: Retrieves the billing information only for the patient with PatientsID equal to 1.

This query is useful for getting a summary of the services and billing details for a particular patient.

⇒ QUERY 7

SQL Worksheet Clear Find Actions Save Run

```
32  Update Billing
33  Set Status ='Unpaid'
34  Where BillingID = 1;
35  Select*from Billing;
36
```

BILLINGID	PATIENTID	SERVICEDATE	SERVICEDESCRIPTION	AMOUNT	STATUS
21	2	16-SEP-23	Consultation	400	Pending
1	1	02-AUG-24	Consultation	150	Unpaid
11	3	06-AUG-24	Specs	400	Pending

Download CSV

3 rows selected.

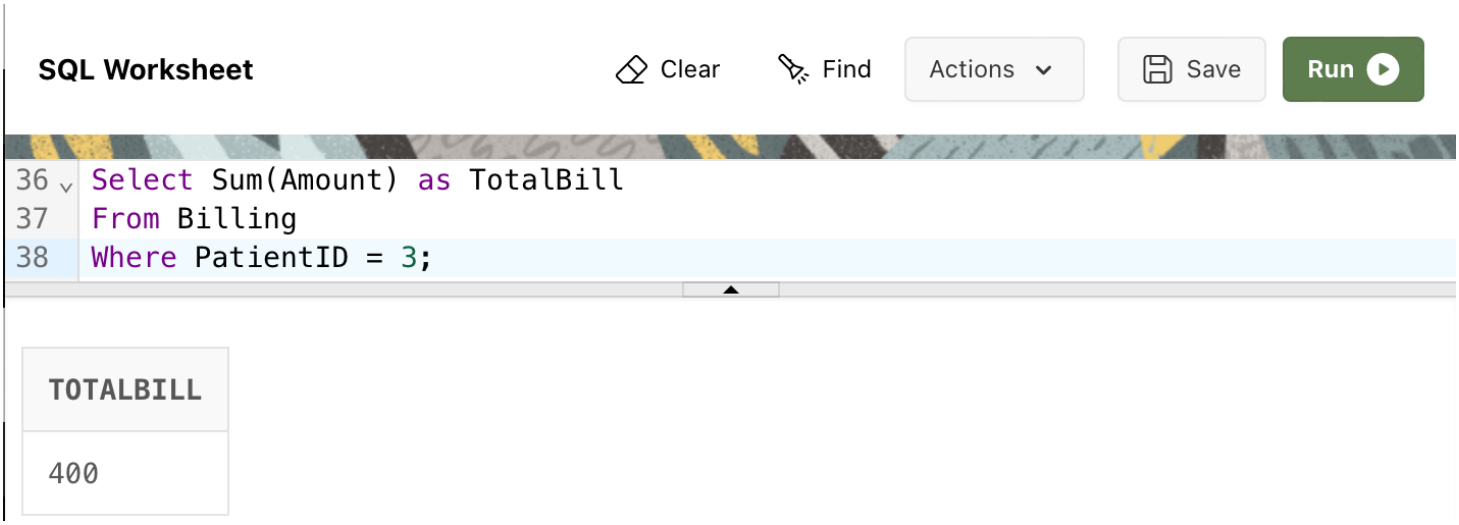
Update Billing Set Status ='Unpaid' Where BillingID = 1

Explanation:

Purpose: This query updates the status of a billing record in the Billing table.
Changes Made:
Sets the Status to 'Unpaid' for the specific billing record.
Condition:
WHERE BillingID = 1
The update only affects the record where the BillingID is 1.

This statement modifies the billing status for the record with BillingID 1 to 'Unpaid'.

⇒ QUERY 8



Select Sum(Amount) as TotalBill From Billing Where PatientID = 3

Explanation:

Purpose: This query calculates the total bill amount for a specific patient.

Columns Selected:
SUM(Amount): Calculates the total of all amounts in the Billing table for the specified patient.
AS TotalBill: Renames the result as TotalBill for clarity.

Condition:
WHERE PatientID = 3
Filters the results to calculate the sum only for the patient with PatientID 3.

This query provides the total amount billed to patient 3.