DESIGN & DEVELOPMENT OF A GOOD DATABASE

AN ARTICLE ON THE IMPACT OF WELL-BUILT DATABASE DESIGNS IN HEALTH CARE BUSINESSES.

BY
ONYINYECHI MEREMIKWU
LinkedIn; https://www.linkedin.com/in/onyinyechi-meremikwu/
Github; https://github.com/omeremikwu/Onyinyechi-Meremikwu

TABLE OF CONTENT

- Introduction
- Mission
- Objectives
- Database Design and Development
  - Tables and Fields
  - Relationship between tables
- Entity Relationship Diagrams (ERD)
- Development
  - SQL Queries and testing screenshots
- Conclusion
- Appendix A
  - Data Dictionary

INTRODUCTION

In today's data-driven world, effective database design is essential for health care organizations aiming to leverage on data for optimized resource usage, efficient patient management and strategic decision-making.

A well-structured database design not only enhances performance and efficiency but also ensures data integrity, security, and scalability.

This article explores the importance of good database design and development in managing doctor schedules and patient wait times in the healthcare sector.

MISSION

The primary mission of database design is to create a robust framework for storing, retrieving, and managing data that meets the needs of an organization. This mission encompasses several key principles:

- Data Integrity: Ensuring that the data is accurate, consistent, and reliable over its lifecycle.
- Efficiency: Designing databases that optimize storage and retrieval operations, reducing the time and resources required to manage data.
- Scalability: Allowing for growth in data volume and user demand without compromising performance.
- Security: Implementing measures to protect sensitive information from unauthorized access and breaches.

OBJECTIVES

To fulfill its mission, effective database design should focus on several specific objectives:

1. Normalization

Normalization involves organizing data to minimize redundancy and dependency. A well-normalized database structure reduces the likelihood of data anomalies and ensures that changes are consistently applied throughout the database.

2. Relationship Management

Defining clear relationships between data entities is crucial. A well-designed database allows for easy navigation and access to related information, which is essential for generating insights and supporting decision-making.

3. Performance Optimization

Good database design considers indexing, query optimization, and efficient data storage techniques. These practices enhance the speed and performance of data retrieval, crucial for applications that require real-time access to information.

4. Security Protocols

A critical aspect of database design is implementing robust security measures. This includes user authentication, access controls, and data encryption, which are vital for protecting sensitive data from breaches and ensuring compliance with regulations.

5. Scalability Planning

As organizations grow, their data requirements change. A good database design anticipates future needs, allowing for easy expansion. This includes modular design and the ability to integrate with other systems and databases.

6. Backup and Recovery Strategies

Effective database design includes planning for data backup and recovery. This ensures that data can be restored in the event of loss or corruption, safeguarding against downtime and data loss.

DATABASE DESIGN & DEVELOPMENT

TABLES

S/N	Table Name	Description
1	Clinic Table	This table stores details about each clinic (or branch) of the hospital.
2	Department Table	The Department table manages the various medical departments available in each clinic.
3	Doctor Table	Stores information about doctors in the hospital.
4	Doctor Schedule Table	Tracks the availability of doctors across different clinics, helping to reduce patient wait times by optimizing doctor schedules.
5	Patient Table	Tracks patients, their medical conditions, and associated appointments.
6	Appointment Table	Handles patient appointments and their schedules, connecting patients, doctors, and clinics.

TABLES & FIELDS

1. Clinic Table		
Field Name	Data Type	Description
Clinicid	INTEGER	Primary key, unique identifier for clinic
Clinic_name	VARCHAR(100)	Name of the clinic
Address	VARCHAR(200)	Address of the clinic
Region	VARCHAR(100)	Region of the city where clinic is located
No_patient_perday	INTEGER	Average number of patients per day
No_appointment_request_perday	INTEGER	Average number of appointment requests per day

2. Department Table		
Field Name	Data Type	Description
Departmentid	INTEGER	Primary key, unique identifier for department
Clinicid	INTEGER	Foreign key, links to the clinic
Department_name	VARCHAR(100)	Name of the department (e.g., Surgery, Paediatrics)

TABLES & FIELDS

3. Doctor Table		
Field Name	Data Type	Description
Doctorid	INTEGER	Primary key, unique identifier for doctor
Name	VARCHAR(100)	Name of the doctor
Specialization	VARCHAR(100)	Doctor's specialization (e.g., Surgeon)
Clinicid	INTEGER	Foreign key, links to the clinic
Departmentid	INTEGER	Foreign key, links to the department
Noof_patients_perday	INTEGER	Number of patients the doctor can see per day

4. Doctor Schedule Table		
Field Name	Data Type	Description
Doctorid	INTEGER	Foreign key, links to the doctor
Clinicid	INTEGER	Foreign key, links to the clinic
Monday	TIME	Doctor's available time on Monday
Tuesday	TIME	Doctor's available time on Tuesday
Wednesday	TIME	Doctor's available time on Wednesday
Thursday	TIME	Doctor's available time on Thursday
Friday	TIME	Doctor's available time on Friday
Saturday	TIME	Doctor's available time on Saturday
Sunday	TIME	Doctor's available time on Sunday

TABLES & FIELDS

5. Patient Table		
Field Name	Data Type	Description
Patientid	INTEGER	Primary key, unique identifier for patient
Name	VARCHAR(100)	Patient's name
Gender	VARCHAR(10)	Patient's gender
Age	INTEGER	Patient's age
Symptoms	VARCHAR(200)	Symptoms reported by the patient
Diagnosis	VARCHAR(2000)	Doctor's diagnosis for the patient

6. Appointment Table		
Field Name	Data Type	Description
Appointmentid	INTEGER	Primary key, unique identifier for appointment
Clinicid	INTEGER	Foreign key, links to the clinic
Patientid	INTEGER	Foreign key, links to the patient
Doctorid	INTEGER	Foreign key, links to the doctor
Appointment_type	VARCHAR(50)	Type of appointment (e.g., Consultation, Surgery)
Scheduled_time	TIME	Scheduled time of the appointment
Diagnosis	VARCHAR(2000)	Diagnosis or purpose of the appointment

Relationship between tables

Table Names	Relationship Type	Description
Clinic To Appointment	One To Many	One clinic can have multiple appointments
Clinic To Department	One to Many	One Clinic has multiple departments
Doctor To Clinic	One To Many	One doctor can be assigned to multiple clinics

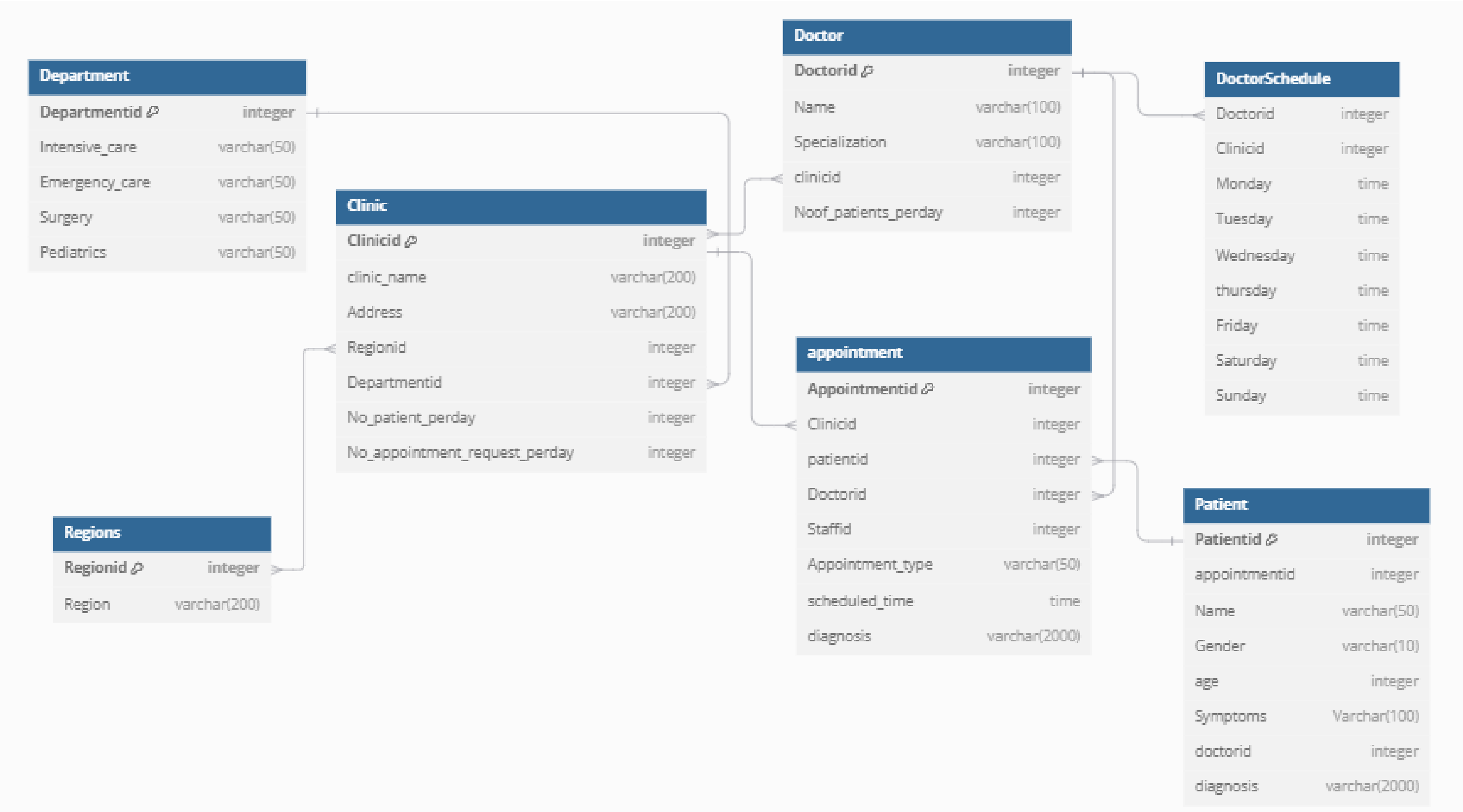
RELATIONSHIP BETWEEN TABLES

Table Names	Relationship Type	Description
Patient To Appointment	One To Many	One Patient can book multiple appointments

Tables Names	Relationship Type	Description
Appointment To Doctor Schedule	Many To One	Many appointments can be assigned as per doctor schedule
Appointment To Doctor	Many To One	Many appointments can be assigned to multiple doctors

ENTITY RELATIONSHIP DIAGRAM

The entity relationship diagrams, show the established relationships between tables (entities).



Department	
Departmentid	integer
Intensive_care	varchar(50)
Emergency_care	varchar(50)
Surgery	varchar(50)
Pediatrics	varchar(50)

Clinic	
Clinicid	integer
clinic_name	varchar(200)
Address	varchar(200)
Regionid	integer
Departmentid	integer
No_patient_perday	integer
No_appointment_request_perday	integer

Regions	
Regionid	integer
Region	varchar(200)

Doctor	
Doctorid	integer
Name	varchar(100)
Specialization	varchar(100)
clinicid	integer
Noof_patients_perday	integer

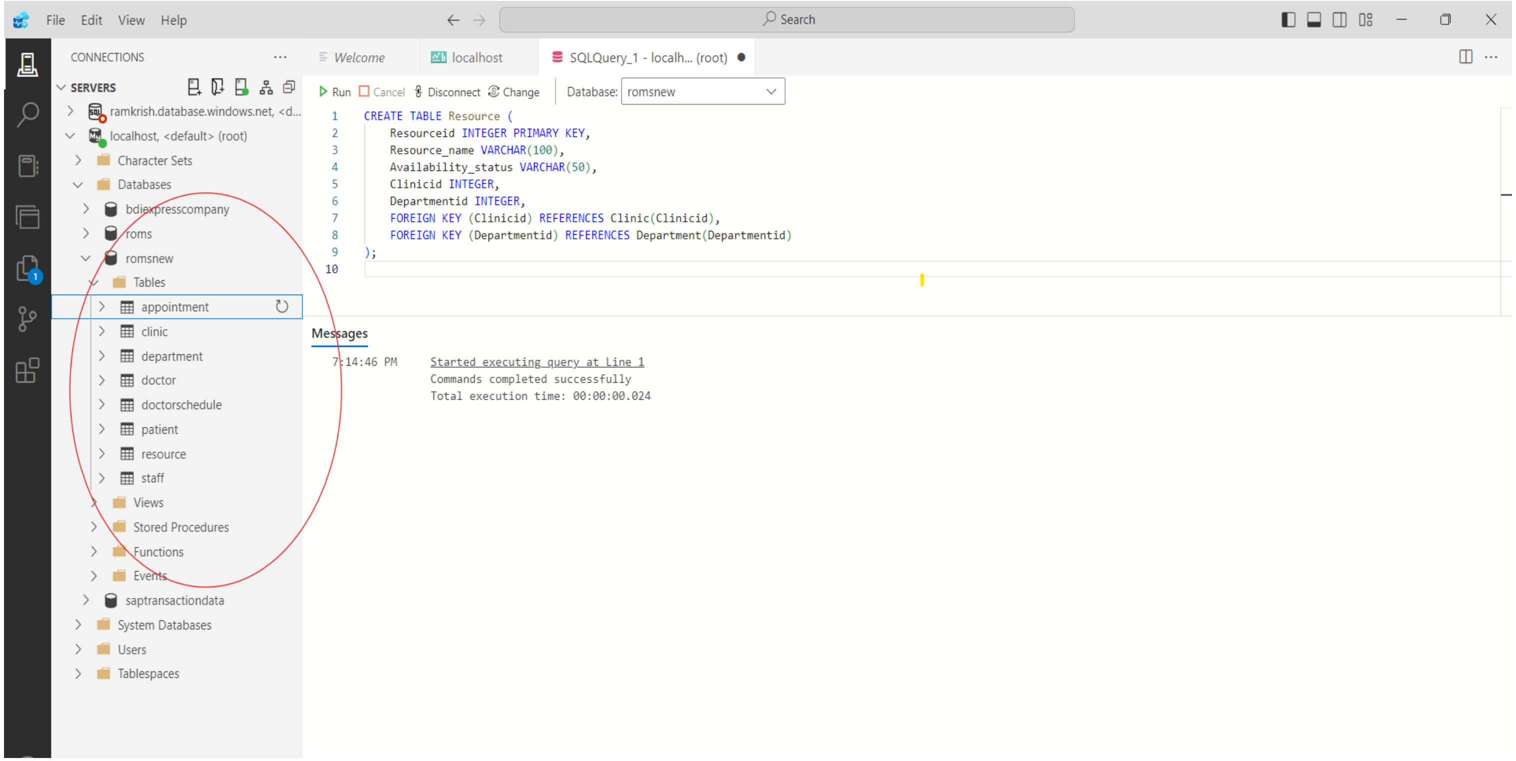
appointment	
Appointmentid	integer
Clinicid	integer
patientid	integer
Doctorid	integer
Staffid	integer
Appointment_type	varchar(50)
scheduled_time	time
diagnosis	varchar(2000)

DoctorSchedule	
Doctorid	integer
Clinicid	integer
Monday	time
Tuesday	time
Wednesday	time
thursday	time
Friday	time
Saturday	time
Sunday	time

Patient	
Patientid	integer
appointmentid	integer
Name	varchar(50)
Gender	varchar(10)
age	integer
Symptoms	Varchar(100)
doctorid	integer
diagnosis	varchar(2000)

DATABASE DEVELOPMENT

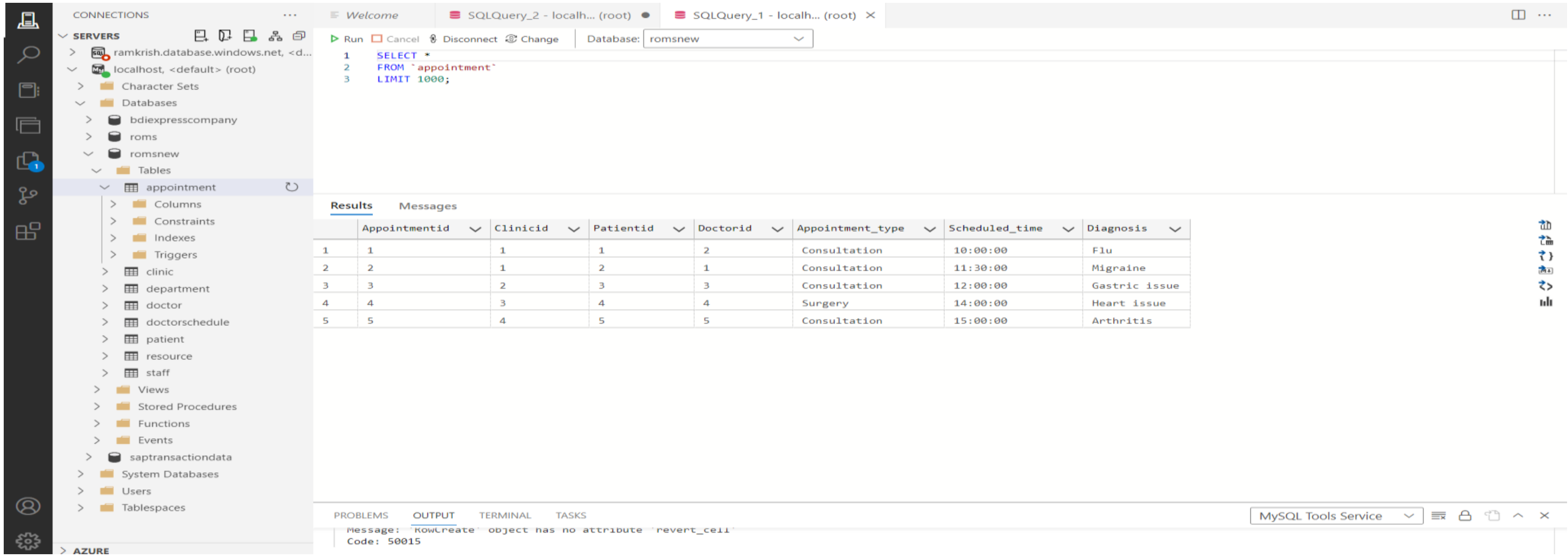
SQL QUERIES AND TESTING SCREENSHOTS



```sql
CREATE TABLE Resource (
    Resourceid INTEGER PRIMARY KEY,
    Resource_name VARCHAR(100),
    Availability_status VARCHAR(50),
    Clinicid INTEGER,
    Departmentid INTEGER,
    FOREIGN KEY (Clinicid) REFERENCES Clinic(Clinicid),
    FOREIGN KEY (Departmentid) REFERENCES Department(Departmentid)
);
```

```
7:14:46 PM    Started executing query at Line 1
              Commands completed successfully
              Total execution time: 00:00:00.024
```

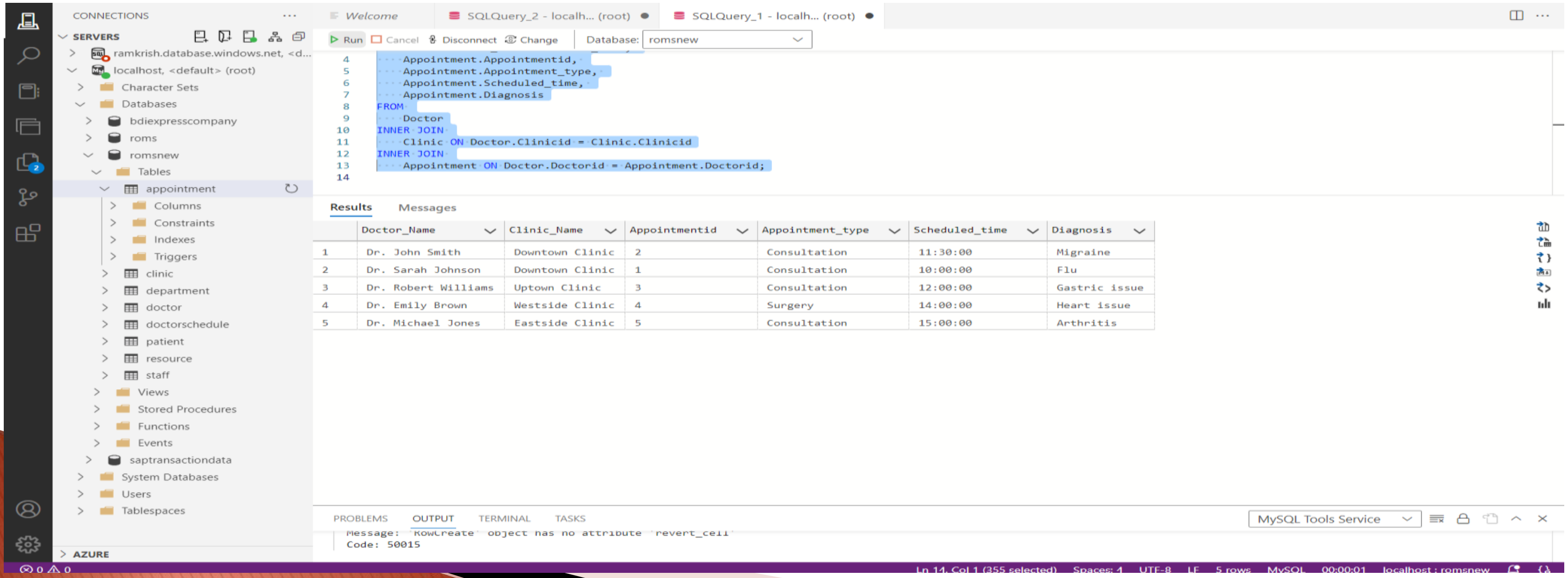
SQL QUERIES AND TESTING SCREENSHOTS



```sql
SELECT *
FROM `appointment`
LIMIT 1000;
```

	Appointmentid	Clinicid	Patientid	Doctorid	Appointment_type	Scheduled_time	Diagnosis
1	1	1	1	2	Consultation	10:00:00	Flu
2	2	1	2	1	Consultation	11:30:00	Migraine
3	3	2	3	3	Consultation	12:00:00	Gastric issue
4	4	3	4	4	Surgery	14:00:00	Heart issue
5	5	4	5	5	Consultation	15:00:00	Arthritis

```
Message: 'RowCreate' object has no attribute 'revert_cell'
Code: 50015
```



```sql
    Appointment.Appointmentid,
    Appointment.Appointment_type,
    Appointment.Scheduled_time,
    Appointment.Diagnosis
FROM
    Doctor
INNER JOIN
    Clinic ON Doctor.Clinicid = Clinic.Clinicid
INNER JOIN
    Appointment ON Doctor.Doctorid = Appointment.Doctorid;
```

	Doctor_Name	Clinic_Name	Appointmentid	Appointment_type	Scheduled_time	Diagnosis
1	Dr. John Smith	Downtown Clinic	2	Consultation	11:30:00	Migraine
2	Dr. Sarah Johnson	Downtown Clinic	1	Consultation	10:00:00	Flu
3	Dr. Robert Williams	Uptown Clinic	3	Consultation	12:00:00	Gastric issue
4	Dr. Emily Brown	Westside Clinic	4	Surgery	14:00:00	Heart issue
5	Dr. Michael Jones	Eastside Clinic	5	Consultation	15:00:00	Arthritis

```
Message: 'RowCreate' object has no attribute 'revert_cell'
Code: 50015
```

Ln 14, Col 1 (355 selected)   Spaces: 4   UTF-8   LF   5 rows   MySQL   00:00:01   localhost : romsnew

CONCLUSION

Good database design is a cornerstone of effective data management. By focusing on integrity, efficiency, scalability, and security, healthcare organizations can create a strong foundation for leveraging their data assets.

In this digital age and in the future, the importance of a well-designed database cannot be overstated, and as businesses continue to evolve, it is an essential component that drives success, fosters innovation, and supports sustainable growth.

APPENDIX A

DATA DICTIONARY

- Primary Key: A field that uniquely identifies each record in the table.

- Auto-increment: Automatically generates a new value for each new record.

- Unique: Ensures no duplicate values for this field in the table.

- Nullable: Indicates that the field can contain null values.

- Integer: An integer is a data type that represents whole numbers, which can be positive, negative, or zero. In programming and databases, integers are commonly used for counting, indexing, and performing mathematical operations.

- String: A string is a data type used to represent text. It consists of a sequence of characters, which can include letters, numbers, symbols, and whitespace. Strings are commonly used for storing and manipulating textual data in programming and databases.

- Field: A field refers to a single piece of data within a record. Fields are the smallest unit of data in a database table and correspond to a specific attribute or characteristic of the data being stored.

THANK YOU