



Pearson Higher Nationals in Computing (Network Engineering)

**Unit 04: Database Design & Development
Assignment**

Higher National Diploma in Computing

Course Title	Pearson BTEC Level 5 Higher National Diploma in Computing		
Unit Number and Title:	Unit 4: Database Design & Development		
Assignment Title	Database Design & Development - HEALTH IS WEALTH - Organic Products Pharmacy & Medical Supply		
Name of the Learner	KALAWILAGE VIHANGA PATHUM		
Centre Ref. No of The Learner		Pearson Regd. No.	PSM-17217
Assignment Number	1	Batch No & Semester	Part Time Batch 2A – Semester 2
Issue Date	13/10/2024	Submission Date	28/11/2024
Re-submission Date		Date Received 1st submission	
Unit Assessor:	Mr. Ajan Sri Sheheran	Academic Year	2024

Assessor Summative Feedback		
<p>Assessor Feedback:</p> <p>*Please note that constructive and useful feedback should allow students to understand:</p> <ul style="list-style-type: none"> a) Strengths of performance b) Limitations of performance c) Transferable skills and competencies developed d) Any improvements needed in future assessments <p>Filled by Lecturer</p>		
Grade: Filled by	Assessor Signature: Ajan Sri Sheheran	Date: Filled by Lecturer

Resubmission Feedback - Formative		
Grade:	Assessor Signature:	Date
Internal Verifier's Comments		
Filled by IV		
Signature of the IV Filled by IV		Date Filled by IV
Student Agreement: I understand the feedback given to me and agree to carry out the actions in future works as required and indicated.		Student Signature vihang Date 28/11/2024

Please note that grade decisions are provisional. They are only confirmed once internal and external moderation has taken place and grades decisions have been agreed at the assessment board.

STUDENT ASSESSMENT SUBMISSION AND DECLARATION

Student name: K.Vihanga pathum		Assessor name: Mr .Ajan Sri Sheheran
Issue date: 13/10/2024	Submission date: 13/11/2024	Submitted on: 28/11/2024
Programme: Pearson BTEC Level 5 Higher National Diploma in Computing (NE)		
Unit: Unit 4: Database Design & Development		
Assignment number and title: 01: Database Design & Development – HEALTH IS WEALTH – Organic Products Pharmacy & Medical Supply		

Plagiarism

Plagiarism is a particular form of cheating. Plagiarism must be avoided at all costs and students who break the rules, however innocently, may be penalised. It is your responsibility to ensure that you understand correct referencing practices. As a university level student, you are expected to use appropriate references throughout and keep carefully detailed notes of all your sources of materials for material you have used in your work, including any material downloaded from the Internet. Please consult the relevant unit lecturer or your course tutor if you need any further advice.

Student Declaration

Student declaration

I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice.

Student signature: vihanga

Date: 28/11/2024

SUBMISSION FORMAT

Assignment	Learning Outcome	Task	Submission Format	Page Number
1	P1, P2, P3, P4, P5, M1, M2, M3, M4,M5,D1, D2, D3	01	Report	
		02	Report	
		03	Report	
		04	Report	
		05	Report	

ASSESSMENT REVIEW PLAN

Assignment	Grading Criteria	Evidence	Formative Comments	Date
01	P1			
	P2			
	P3			
	P4			
	P5			
	M1			
	M2			
	M3			
	M4			
	M5			
	D1			
	D2			
	D3			

ASSESSMENT TRACKING

ASSESSMENT RECORD AND FEEDBACK SHEET					
Programme :	BTEC HN in Computing (NE)	Student Name:	Filled by Student	Assessment Date:	13/10/2024
Unit No. & Title:	Unit 4: Database Design & Development	Year	2024	Unit Grade:	
Assessor Name:	Mr. Ajan Sri Sheheran	Unit Completion Date:	17/11/2024	IV Signature:	Filled by IV

Assignment No and Title	Learning Objectives	Criteria Targeted	Date Issued	Hand In Date	Summative Feedback	Resubmission Date*
01	LO1 LO2 LO3 LO4	P1, P2, P3, P4, P5, M1, M2, M3, M4,M5, D1, D2, D3				

* Resubmissions must be approved by the Assessment Board

SUMMATIVE FEEDBACK

Assignmen t No.	Gradin g Criteria	Date Achieved	Comments	Assessor Signatur e
Assignment 1	P1			
	P2			
	P3			
	P4			
	P5			
	M1			
	M2			
	M3			

	M4			
	M5			
	D1			
	D2			
	D3			

Question 01

The existing database system design for “Health is Wealth” pharmacy aims to fulfill several core user and system requirements, including usability, scalability, performance, and security. The goal is to streamline pharmaceutical operations, manage relationships between patients, doctors, pharmacies, and pharmaceutical companies, and ensure smooth data handling.

- **Usability**

The system must allow users to easily navigate through information, such as viewing patient details, doctor prescriptions, and pharmaceutical contracts. By utilizing a relational database, users can access and retrieve data efficiently.

- **Scalability**

The database design supports scalability as it allows for the addition of new entities, such as new pharmacies or drugs, without disrupting existing data relationships.

- **Performance**

Indexing primary keys and relationships ensures fast data retrieval. Proper normalization eliminates redundancies, improving overall performance.

- **Security**

The design incorporates security mechanisms like primary keys, foreign keys, and user-level access control to ensure data integrity and restricted access to sensitive information.

1.1 Identifying Entities, Attributes, and ER Diagram Design

The following entities and attributes are derived from the scenario,

1. **Patient**

- o Attributes: SSN (Primary Key), Name, Address, Age, PrimaryDoctorID (Foreign Key)

2. **Doctor**

- o Attributes: SSN (Primary Key), Name, Specialty, YearsOfExperience

3. **Pharmacy**

- o Attributes: PharmacyID (Primary Key), Name, Address, PhoneNumber

4. **Drug**

- o Attributes: DrugID (Primary Key), TradeName, Formula, CompanyID (Foreign Key)

5. **Pharmaceutical Company**

- o Attributes: CompanyID (Primary Key), Name, PhoneNumber

6. **Contract**

- o Attributes: ContractID (Primary Key), PharmacyID (Foreign Key), CompanyID (Foreign Key), StartDate, EndDate, ContractText, Supervisor

7. **Prescription**

- o Attributes: PrescriptionID (Primary Key), PatientID (Foreign Key), DoctorID (Foreign Key), DrugID (Foreign Key)

ER Diagram

The ER diagram will include entities with their respective attributes, primary keys, and relationships. Relationships such as "Prescribes," "Supplies," "Sells," and "Contracts" will be depicted with cardinalities

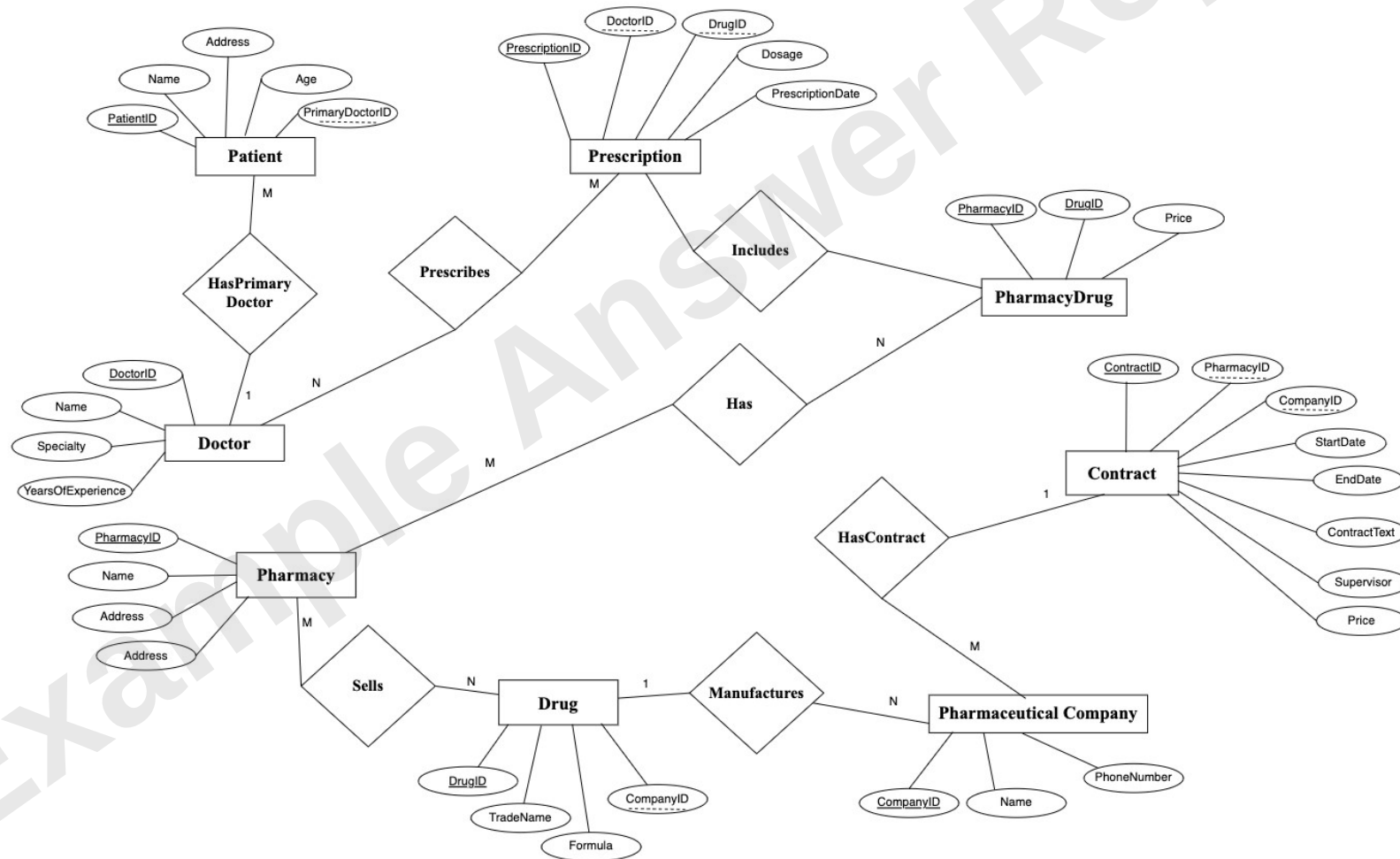


Figure 1 ER Diagram

1.2 Resolving Many-to-Many Relationships

The scenario includes several many-to-many relationships, such as,

- **Doctors prescribing drugs to multiple patients.**
- **Pharmacies selling multiple drugs at different prices.**
- **Pharmaceutical companies contracting with multiple pharmacies.**

To resolve these,

1. Create an associative entity, such as **Prescription**, to link Doctors, Patients, and Drugs.
2. Create another associative entity, **PharmacyDrug**, to link Pharmacies and Drugs with attributes like Price.
3. Establish **Contract** to manage the relationship between Pharmacies and Pharmaceutical Companies.

Four Inter-Related Tables:

1. **Patient:** PatientID, Name, Address, Age, PrimaryDoctorID
2. **Doctor:** DoctorID, Name, Specialty, YearsOfExperience
3. **PharmacyDrug:** PharmacyID, DrugID, Price
4. **Prescription:** PrescriptionID, PatientID, DoctorID, DrugID

1.3 Assumptions

1. Each patient has a unique SSN and is assigned a primary doctor.
2. Drugs are uniquely identified by their trade names and manufactured by specific pharmaceutical companies.
3. Pharmacies maintain unique drug prices, which may vary across locations.
4. All relationships in the database assume referential integrity through foreign keys.

1.4 Logical Database Design (ER Mapping)

ER mapping involves transforming the ER diagram into a relational schema. Each entity becomes a table, and attributes become columns. Primary keys and foreign keys enforce integrity.

- **Patient (SSN, Name, Address, Age, PrimaryDoctorID):** Primary Key: SSN, Foreign Key: PrimaryDoctorID
- **Doctor (SSN, Name, Specialty, YearsOfExperience):** Primary Key: SSN
- **Pharmacy (PharmacyID, Name, Address, PhoneNumber):** Primary Key: PharmacyID
- **Drug (DrugID, TradeName, Formula, CompanyID):** Primary Key: DrugID, Foreign Key: CompanyID
- **Pharmaceutical Company (CompanyID, Name, PhoneNumber):** Primary Key: CompanyID
- **Contract (ContractID, PharmacyID, CompanyID, StartDate, EndDate, ContractText, Supervisor):** Primary Key: ContractID, Foreign Keys: PharmacyID, CompanyID
- **PharmacyDrug (PharmacyID, DrugID, Price):** Primary Key: PharmacyID, DrugID
- **Prescription (PrescriptionID, PatientID, DoctorID, DrugID):** Primary Key: PrescriptionID, Foreign Keys: PatientID, DoctorID, DrugID

(e) Data Normalization

Normalization reduces redundancy and ensures data consistency. Steps for achieving **Third Normal Form (3NF)**

1. First Normal Form (1NF)

To achieve **1NF**, we ensure that all attributes contain atomic values, meaning no attribute can hold multiple values in a single field. Additionally, each record must be unique, which is guaranteed by assigning a primary key to every table.

2. Second Normal Form (2NF):

The **2NF** focuses on eliminating partial dependencies. This is particularly relevant for tables with composite primary keys. Every non-key attribute should depend on the entire primary key, not just a part of it.

For example:

1. PharmacyDrug Table

- o Composite Key (PharmacyID, DrugID)
- o Non-Key Attribute: Price
- o Dependency Price depends on both PharmacyID and DrugID, meaning it is fully dependent on the composite key. This table is already in **2NF**.

2. Prescription Table

- o Composite Key (PatientID, DoctorID, DrugID) (if using composite keys).
- o Non-Key Attribute Dosage depends on all parts of the key.

3. Third Normal Form (3NF)

- Non-Normalized The CompanyPhoneNumber field exists in the Drug table. This creates a dependency between CompanyPhoneNumber and CompanyID, which violates 3NF.
- Normalized Move CompanyPhoneNumber to the **PharmaceuticalCompany** table.

Final Tables in Normalized Form

After applying normalization, the database structure includes well-structured tables such as:

1. Patient Table

Contains only patient-related attributes, with no redundant data.

2. Doctor Table

Stores doctor-specific information, including DoctorID as the primary key.

3. Drug Table

Stores drug details with references to the pharmaceutical company.

4. Prescription Table

Links doctors, patients, and drugs, storing only relevant prescription details.

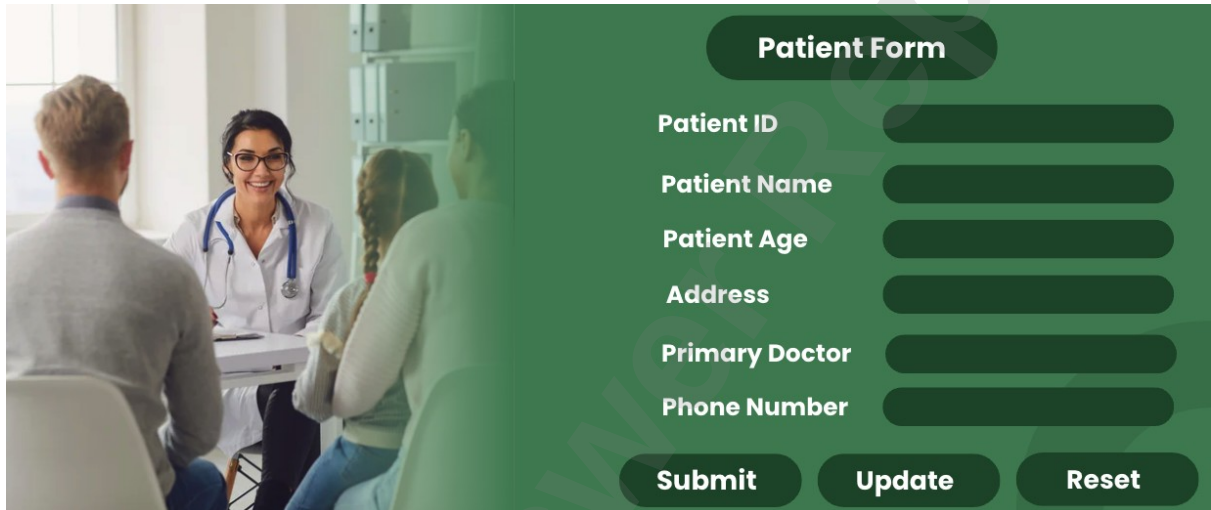
5. PharmacyDrug Table

Resolves the many-to-many relationship between pharmacies and drugs, including pricing.

1.5 Designing Interfaces

1. Patient Form

Input fields for SSN, Name, Address, Age, and Primary Doctor selection.



Patient Form

Patient ID

Patient Name

Patient Age

Address

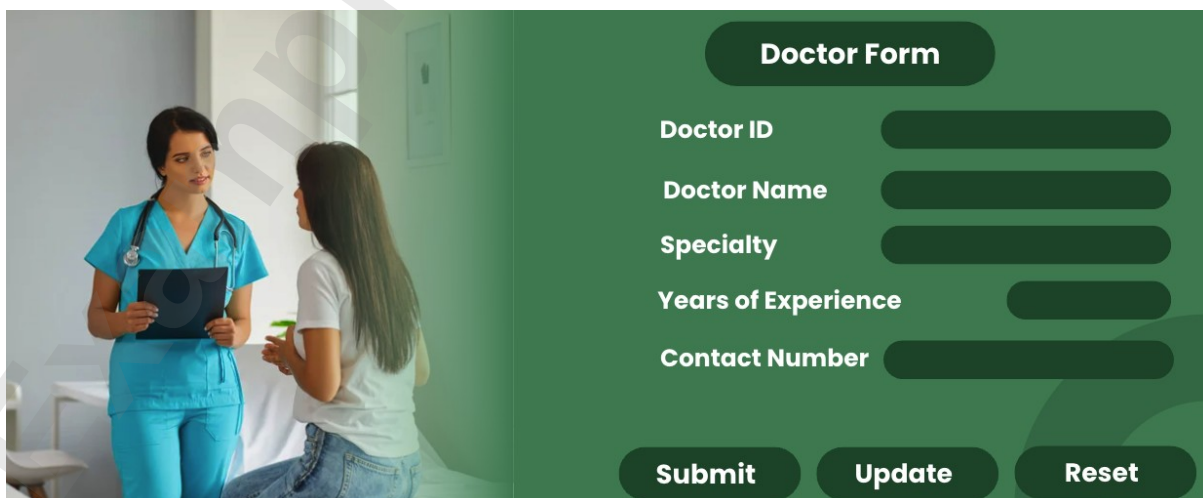
Primary Doctor

Phone Number

Figure 2 Patient Form

2. Doctor Form

Input fields for SSN, Name, Specialty, and Years of Experience.



Doctor Form

Doctor ID

Doctor Name

Specialty

Years of Experience

Contact Number

Figure 3 Doctor Form

3. Drug Form

Input fields for Trade Name, Formula, and Manufacturer.

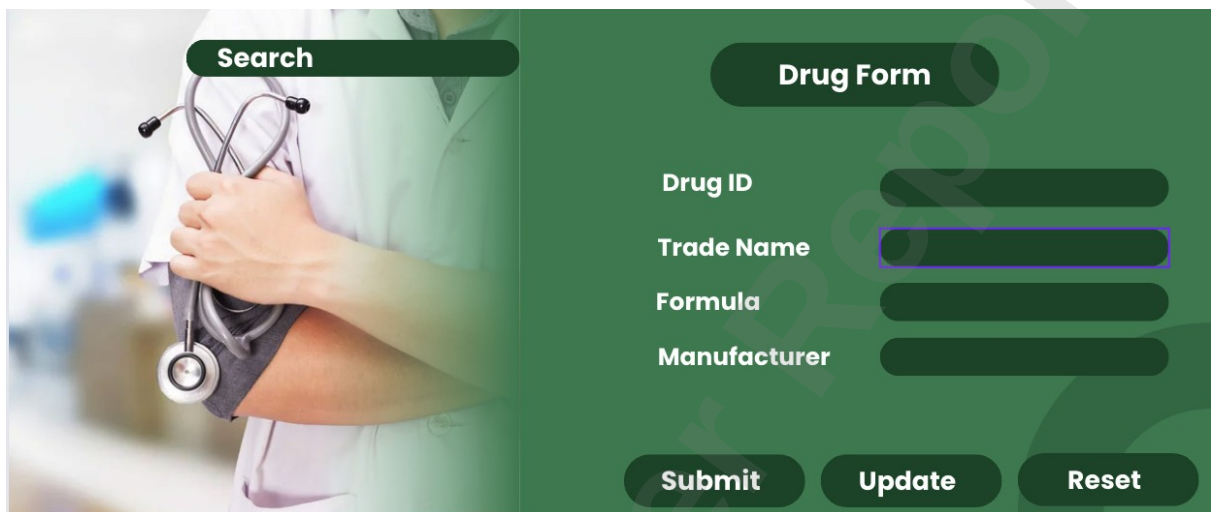
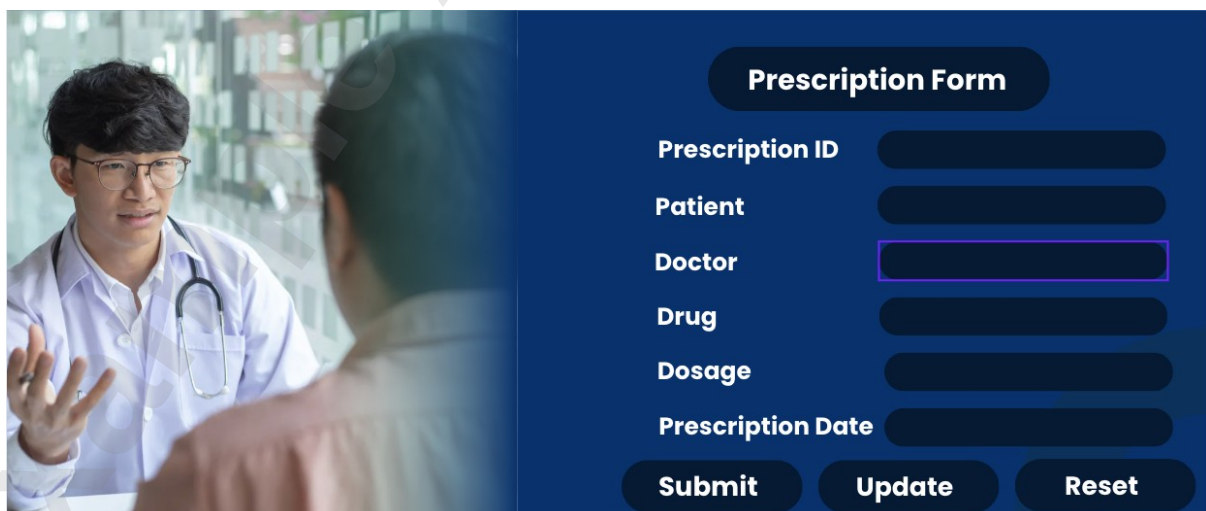
A UI mockup for a 'Drug Form' on a green background. On the left, there is a vertical image of a doctor's arm with a stethoscope. A dark green 'Search' button is overlaid on the image. To the right, the 'Drug Form' title is in a dark green rounded rectangle. Below it are four input fields: 'Drug ID', 'Trade Name' (highlighted with a purple border), 'Formula', and 'Manufacturer'. At the bottom are three buttons: 'Submit', 'Update', and 'Reset'.

Figure 4 Drug Form

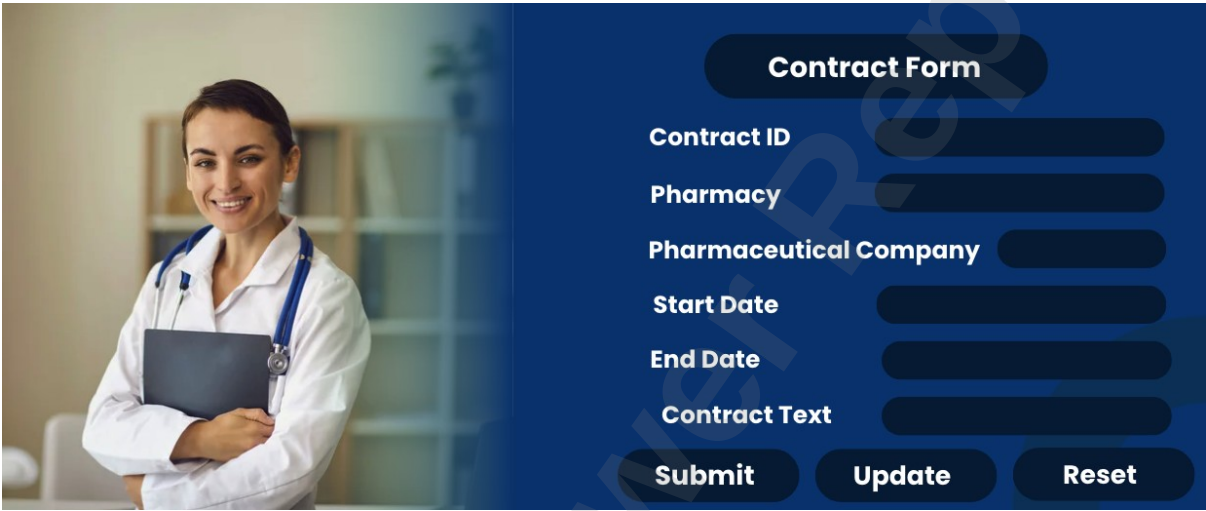
4. Prescription Form

Dropdowns for selecting Patient, Doctor, and Drug, and additional notes for dosage.

A UI mockup for a 'Prescription Form' on a dark blue background. On the left, there is a vertical image of a doctor talking to a patient. A dark blue rounded rectangle with the title 'Prescription Form' is on the right. Below the title are six input fields: 'Prescription ID', 'Patient', 'Doctor' (highlighted with a purple border), 'Drug', 'Dosage', and 'Prescription Date'. At the bottom are three buttons: 'Submit', 'Update', and 'Reset'.

5. **Contract Form**

Fields for Start Date, End Date, Contract Text, and Supervisor.



The image displays a user interface for a 'Contract Form'. On the left, a female doctor in a white coat and stethoscope holds a clipboard. On the right, a dark blue panel contains the form. The form has a title 'Contract Form' at the top. Below it are six input fields: 'Contract ID', 'Pharmacy', 'Pharmaceutical Company', 'Start Date', 'End Date', and 'Contract Text'. At the bottom of the form are three buttons: 'Submit', 'Update', and 'Reset'.

Figure 5 Contract Form

This design not only meets the system requirements but also ensures future scalability and ease of management. The database adheres to normalization principles, effectively resolves many-to-many relationships, and integrates interfaces for streamlined user interaction. Further improvements could include role-based access controls and real-time analytics for enhanced usability.

Question 02

a) Structured Query Language Health is Welth Pharmacy System

01. Create the Database

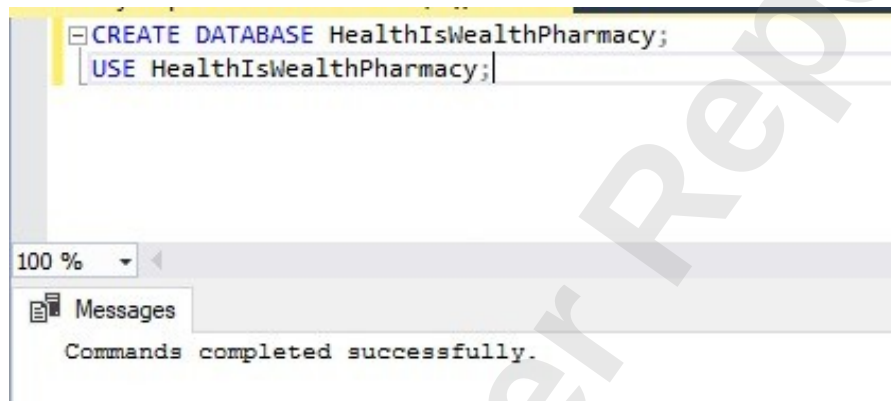


Figure 6 Create the Database

02. Create the Tables

Patient Table

```
CREATE TABLE Patient (  
    PatientID VARCHAR(10) PRIMARY KEY,  
    Name VARCHAR(50) NOT NULL,  
    Street VARCHAR(50) NOT NULL,  
    City VARCHAR(30) NOT NULL,  
    ZipCode VARCHAR(10) NOT NULL,  
    DateOfBirth DATE NOT NULL,  
    PrimaryDoctorID VARCHAR(10),  
    FOREIGN KEY (PrimaryDoctorID) REFERENCES Doctor(DoctorID)  
);
```

Doctor Table

```
CREATE TABLE Doctor (  
    DoctorID VARCHAR(10) PRIMARY KEY,  
    Name VARCHAR(50) NOT NULL,  
    Specialty VARCHAR(30) NOT NULL,  
    YearsOfExperience INT NOT NULL,  
    ContactNumber VARCHAR(15) NOT NULL  
);
```

Pharmacy Table

```
CREATE TABLE Pharmacy (  
    PharmacyID VARCHAR(10) PRIMARY KEY,  
    Name VARCHAR(50) NOT NULL,  
    Street VARCHAR(50) NOT NULL,  
    City VARCHAR(30) NOT NULL,  
    ZipCode VARCHAR(10) NOT NULL,  
    PhoneNumber VARCHAR(15) NOT NULL  
);
```

Pharmaceutical Company Table

```
CREATE TABLE PharmaceuticalCompany (  
    CompanyID VARCHAR(10) PRIMARY KEY,  
    Name VARCHAR(50) NOT NULL,  
    PhoneNumber VARCHAR(15) NOT NULL  
);
```

Drug Table

```
CREATE TABLE Drug (  
    DrugID VARCHAR(10) PRIMARY KEY,  
    TradeName VARCHAR(50) NOT NULL,  
    Formula VARCHAR(100) NOT NULL,  
    CompanyID VARCHAR(10),  
    FOREIGN KEY (CompanyID) REFERENCES PharmaceuticalCompany(CompanyID)  
);
```

PharmacyDrug Table

```
CREATE TABLE PharmacyDrug (  
    PharmacyID VARCHAR(10),  
    DrugID VARCHAR(10),  
    Price DECIMAL(10, 2) NOT NULL,  
    PRIMARY KEY (PharmacyID, DrugID),  
    FOREIGN KEY (PharmacyID) REFERENCES Pharmacy(PharmacyID),  
    FOREIGN KEY (DrugID) REFERENCES Drug(DrugID)  
);
```

Prescription Table

```
CREATE TABLE Prescription (  
    PrescriptionID VARCHAR(10) PRIMARY KEY,  
    PatientID VARCHAR(10),  
    DoctorID VARCHAR(10),  
    DrugID VARCHAR(10),  
    Dosage VARCHAR(100) NOT NULL,  
    PrescriptionDate DATE NOT NULL,  
    FOREIGN KEY (PatientID) REFERENCES Patient(PatientID),  
    FOREIGN KEY (DoctorID) REFERENCES Doctor(DoctorID),  
    FOREIGN KEY (DrugID) REFERENCES Drug(DrugID)  
);
```

Contract Table

```

CREATE TABLE Contract (
    ContractID VARCHAR(10) PRIMARY KEY,
    PharmacyID VARCHAR(10),
    CompanyID VARCHAR(10),
    StartDate DATE NOT NULL,
    EndDate DATE,
    ContractText TEXT NOT NULL,
    Supervisor VARCHAR(50) NOT NULL,
    FOREIGN KEY (PharmacyID) REFERENCES Pharmacy(PharmacyID),
    FOREIGN KEY (CompanyID) REFERENCES PharmaceuticalCompany(CompanyID)
);

```

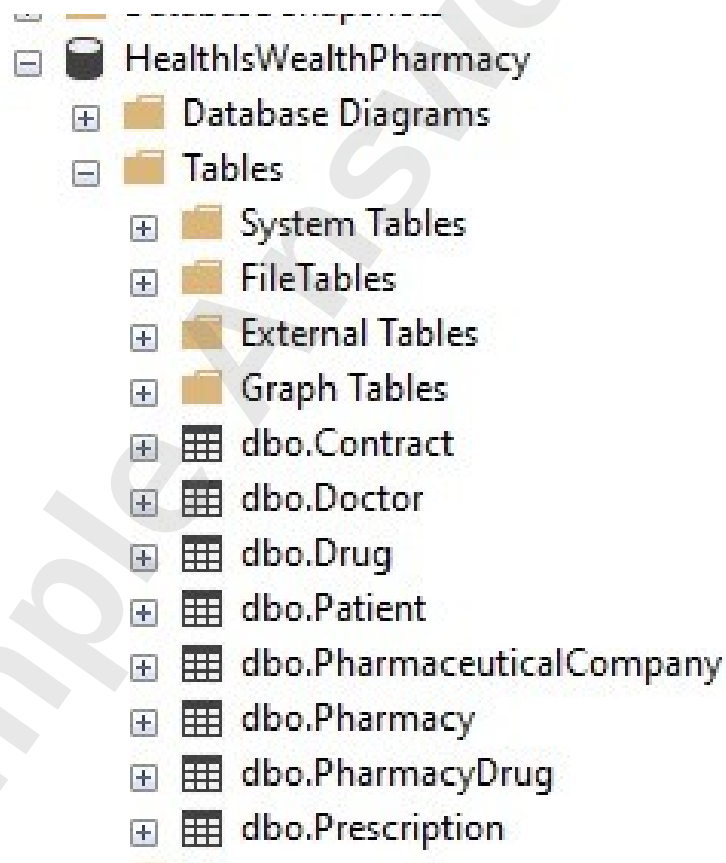


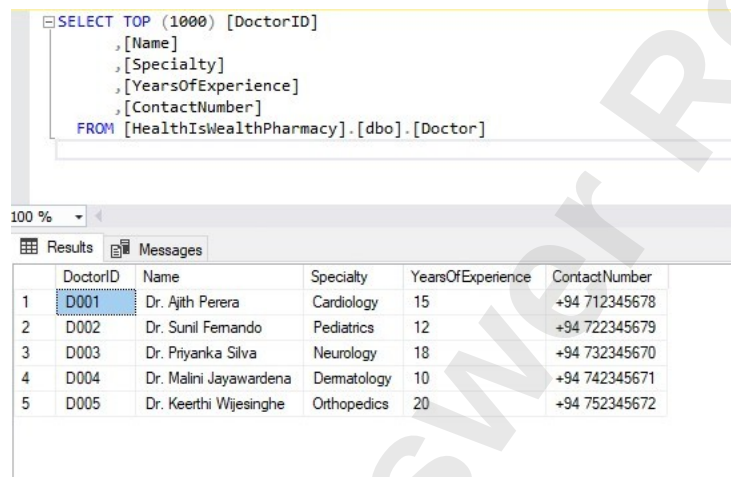
Figure 7 Table Structure

3. Populating Tables

Insert Data into Doctor Table

INSERT INTO Doctor VALUES

('D001', 'Dr. Ajith Perera', 'Cardiology', 15, '+94 712345678'),
('D002', 'Dr. Sunil Fernando', 'Pediatrics', 12, '+94 722345679'),
('D003', 'Dr. Priyanka Silva', 'Neurology', 18, '+94 732345670'),
('D004', 'Dr. Malini Jayawardena', 'Dermatology', 10, '+94 742345671'),
('D005', 'Dr. Keerthi Wijesinghe', 'Orthopedics', 20, '+94 752345672');



The screenshot shows a SQL query window with the following query:

```
SELECT TOP (1000) [DoctorID]
, [Name]
, [Specialty]
, [YearsOfExperience]
, [ContactNumber]
FROM [HealthIsWealthPharmacy].[dbo].[Doctor]
```

Below the query window, the 'Results' tab displays the following data:

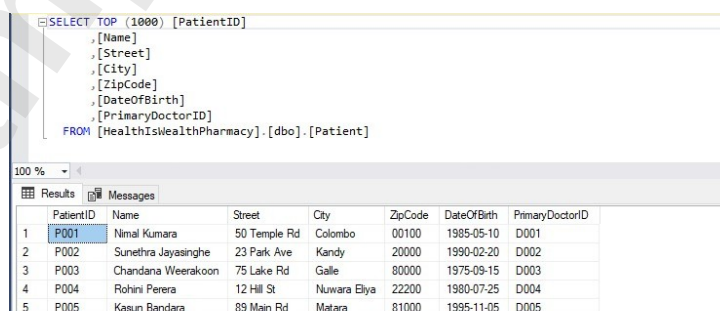
	DoctorID	Name	Specialty	YearsOfExperience	ContactNumber
1	D001	Dr. Ajith Perera	Cardiology	15	+94 712345678
2	D002	Dr. Sunil Fernando	Pediatrics	12	+94 722345679
3	D003	Dr. Priyanka Silva	Neurology	18	+94 732345670
4	D004	Dr. Malini Jayawardena	Dermatology	10	+94 742345671
5	D005	Dr. Keerthi Wijesinghe	Orthopedics	20	+94 752345672

Figure 8 Doctor Table

Insert Data into Patient Table

INSERT INTO Patient VALUES

('P001', 'Nimal Kumara', '50 Temple Rd', 'Colombo', '00100', '1985-05-10', 'D001'),
('P002', 'Sunethra Jayasinghe', '23 Park Ave', 'Kandy', '20000', '1990-02-20', 'D002'),
('P003', 'Chandana Weerakoon', '75 Lake Rd', 'Galle', '80000', '1975-09-15', 'D003'),
('P004', 'Rohini Perera', '12 Hill St', 'Nuwara Eliya', '22200', '1980-07-25', 'D004'),
('P005', 'Kasun Bandara', '89 Main Rd', 'Matara', '81000', '1995-11-05', 'D005');



The screenshot shows a SQL query window with the following query:

```
SELECT TOP (1000) [PatientID]
, [Name]
, [Street]
, [City]
, [ZipCode]
, [DateOfBirth]
, [PrimaryDoctorID]
FROM [HealthIsWealthPharmacy].[dbo].[Patient]
```

Below the query window, the 'Results' tab displays the following data:

	PatientID	Name	Street	City	ZipCode	DateOfBirth	PrimaryDoctorID
1	P001	Nimal Kumara	50 Temple Rd	Colombo	00100	1985-05-10	D001
2	P002	Sunethra Jayasinghe	23 Park Ave	Kandy	20000	1990-02-20	D002
3	P003	Chandana Weerakoon	75 Lake Rd	Galle	80000	1975-09-15	D003
4	P004	Rohini Perera	12 Hill St	Nuwara Eliya	22200	1980-07-25	D004
5	P005	Kasun Bandara	89 Main Rd	Matara	81000	1995-11-05	D005

Insert Data into Pharmacy Table

INSERT INTO Pharmacy VALUES

('PH001', 'City Pharmacy', '120 Main St', 'Colombo', '00100', '+94 114567890'),
 ('PH002', 'Hilltop Pharmacy', '45 Station Rd', 'Kandy', '20000', '+94 114567891'),
 ('PH003', 'Seaside Pharmacy', '90 Beach Rd', 'Galle', '80000', '+94 114567892'),
 ('PH004', 'Mountain Pharmacy', '34 Green Ln', 'Nuwara Eliya', '22200', '+94 114567893'),
 ('PH005', 'Southern Pharmacy', '18 Bay Rd', 'Matara', '81000', '+94 114567894');



```

SELECT TOP (1000) [PharmacyID]
      ,[Name]
      ,[Street]
      ,[City]
      ,[ZipCode]
      ,[PhoneNumber]
FROM [HealthIsWealthPharmacy].[dbo].[Pharmacy]
  
```

	PharmacyID	Name	Street	City	ZipCode	PhoneNumber
1	PH001	City Pharmacy	120 Main St	Colombo	00100	+94 114567890
2	PH002	Hilltop Pharmacy	45 Station Rd	Kandy	20000	+94 114567891
3	PH003	Seaside Pharmacy	90 Beach Rd	Galle	80000	+94 114567892
4	PH004	Mountain Pharmacy	34 Green Ln	Nuwara Eliya	22200	+94 114567893
5	PH005	Southern Pharmacy	18 Bay Rd	Matara	81000	+94 114567894

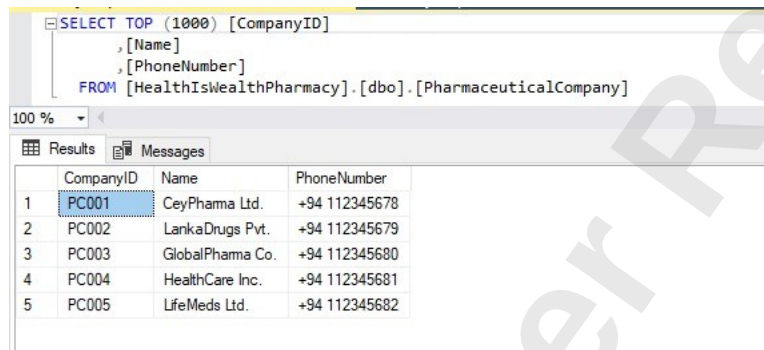
Figure 10 Pharmacy Table

Insert Data into PharmaceuticalCompany Table

```

INSERT INTO PharmaceuticalCompany VALUES
('PC001', 'CeyPharma Ltd.', '+94 112345678'),
('PC002', 'LankaDrugs Pvt.', '+94 112345679'),
('PC003', 'GlobalPharma Co.', '+94 112345680'),
('PC004', 'HealthCare Inc.', '+94 112345681'),
('PC005', 'LifeMeds Ltd.', '+94 112345682');

```



	CompanyID	Name	PhoneNumber
1	PC001	CeyPharma Ltd.	+94 112345678
2	PC002	LankaDrugs Pvt.	+94 112345679
3	PC003	GlobalPharma Co.	+94 112345680
4	PC004	HealthCare Inc.	+94 112345681
5	PC005	LifeMeds Ltd.	+94 112345682

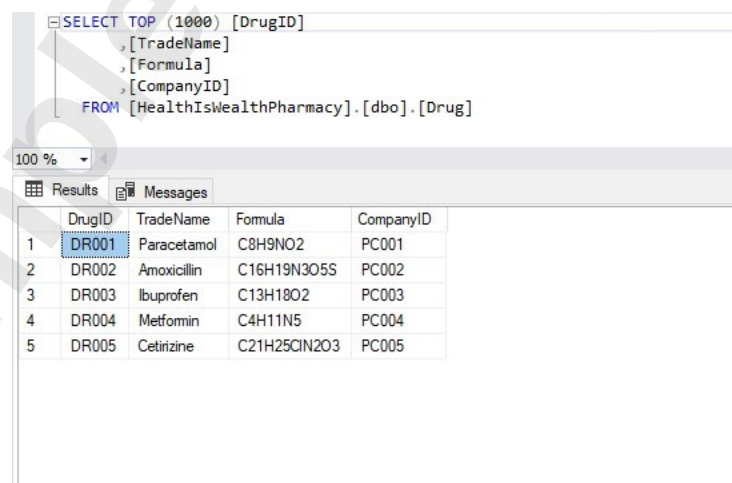
Figure 11 PharmaceuticalCompany Table

Insert Data into Drug Table

```

INSERT INTO Drug VALUES
('DR001', 'Paracetamol', 'C8H9NO2', 'PC001'),
('DR002', 'Amoxicillin', 'C16H19N3O5S', 'PC002'),
('DR003', 'Ibuprofen', 'C13H18O2', 'PC003'),
('DR004', 'Metformin', 'C4H11N5', 'PC004'),
('DR005', 'Cetirizine', 'C21H25ClN2O3', 'PC005');

```



	DrugID	TradeName	Formula	CompanyID
1	DR001	Paracetamol	C8H9NO2	PC001
2	DR002	Amoxicillin	C16H19N3O5S	PC002
3	DR003	Ibuprofen	C13H18O2	PC003
4	DR004	Metformin	C4H11N5	PC004
5	DR005	Cetirizine	C21H25ClN2O3	PC005

Figure 12 Drug Table

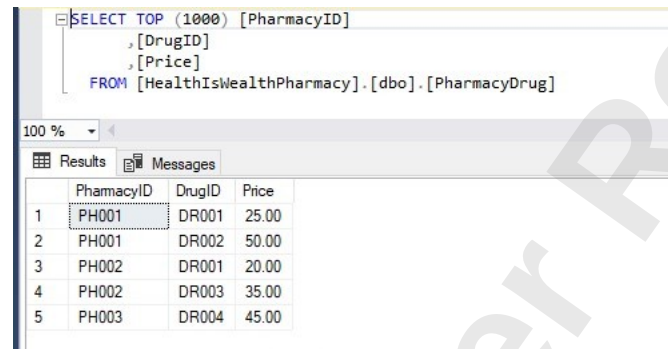
Insert Data into PharmacyDrug Table

```

INSERT INTO PharmacyDrug VALUES

```

('PH001', 'DR001', 25.00),
 ('PH001', 'DR002', 50.00),
 ('PH002', 'DR001', 20.00),
 ('PH002', 'DR003', 35.00),
 ('PH003', 'DR004', 45.00);



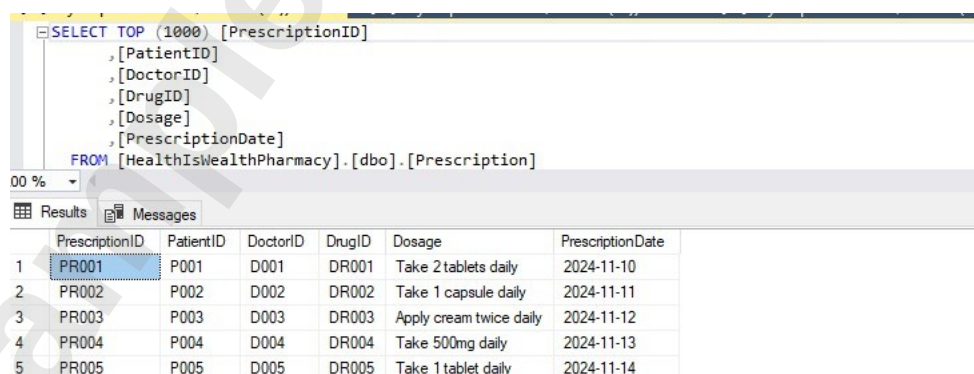
	PharmacyID	DrugID	Price
1	PH001	DR001	25.00
2	PH001	DR002	50.00
3	PH002	DR001	20.00
4	PH002	DR003	35.00
5	PH003	DR004	45.00

Figure 13 PharmacyDrug Table

Insert Data into Prescription Table

INSERT INTO Prescription VALUES

('PR001', 'P001', 'D001', 'DR001', 'Take 2 tablets daily', '2024-11-10'),
 ('PR002', 'P002', 'D002', 'DR002', 'Take 1 capsule daily', '2024-11-11'),
 ('PR003', 'P003', 'D003', 'DR003', 'Apply cream twice daily', '2024-11-12'),
 ('PR004', 'P004', 'D004', 'DR004', 'Take 500mg daily', '2024-11-13'),
 ('PR005', 'P005', 'D005', 'DR005', 'Take 1 tablet daily', '2024-11-14');



	PrescriptionID	PatientID	DoctorID	DrugID	Dosage	PrescriptionDate
1	PR001	P001	D001	DR001	Take 2 tablets daily	2024-11-10
2	PR002	P002	D002	DR002	Take 1 capsule daily	2024-11-11
3	PR003	P003	D003	DR003	Apply cream twice daily	2024-11-12
4	PR004	P004	D004	DR004	Take 500mg daily	2024-11-13
5	PR005	P005	D005	DR005	Take 1 tablet daily	2024-11-14

Figure 14 Prescription Table

Insert Data into Contract Table

INSERT INTO Contract VALUES

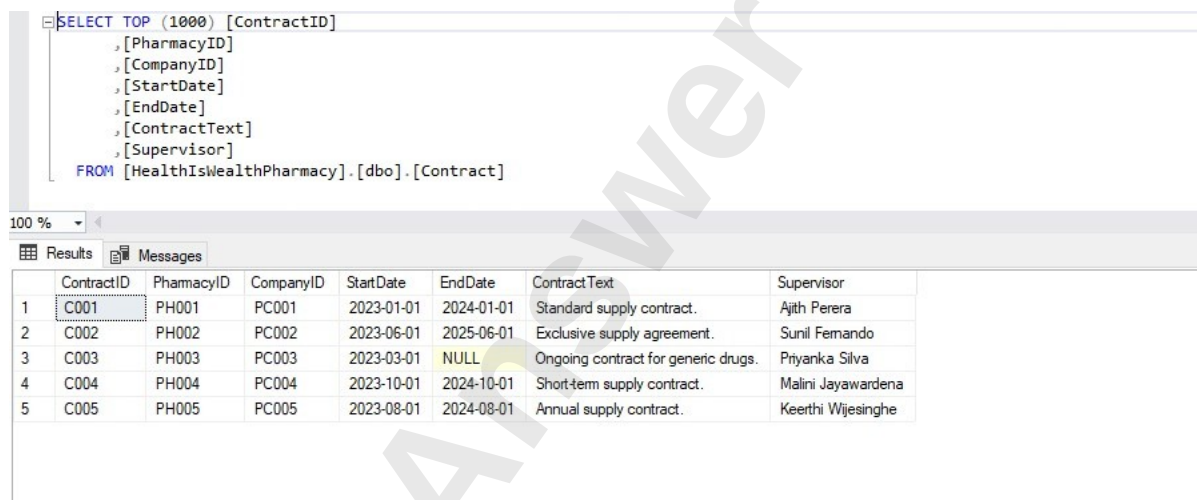
('C001', 'PH001', 'PC001', '2023-01-01', '2024-01-01', 'Standard supply contract.', 'Ajith Perera'),

('C002', 'PH002', 'PC002', '2023-06-01', '2025-06-01', 'Exclusive supply agreement.', 'Sunil Fernando'),

('C003', 'PH003', 'PC003', '2023-03-01', NULL, 'Ongoing contract for generic drugs.', 'Priyanka Silva'),

('C004', 'PH004', 'PC004', '2023-10-01', '2024-10-01', 'Short-term supply contract.', 'Malini Jayawardena'),

('C005', 'PH005', 'PC005', '2023-08-01', '2024-08-01', 'Annual supply contract.', 'Keerthi Wijesinghe');



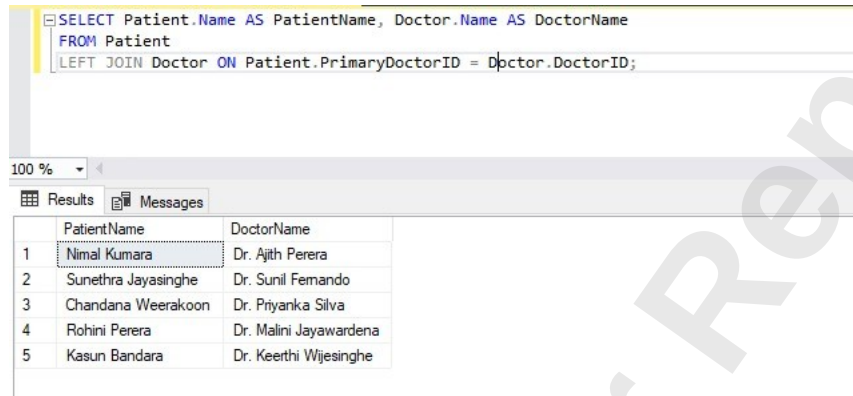
```
SELECT TOP (1000) [ContractID]
      ,[PharmacyID]
      ,[CompanyID]
      ,[StartDate]
      ,[EndDate]
      ,[ContractText]
      ,[Supervisor]
FROM [HealthIsWealthPharmacy].[dbo].[Contract]
```

	ContractID	PharmacyID	CompanyID	StartDate	EndDate	ContractText	Supervisor
1	C001	PH001	PC001	2023-01-01	2024-01-01	Standard supply contract.	Ajith Perera
2	C002	PH002	PC002	2023-06-01	2025-06-01	Exclusive supply agreement.	Sunil Fernando
3	C003	PH003	PC003	2023-03-01	NULL	Ongoing contract for generic drugs.	Priyanka Silva
4	C004	PH004	PC004	2023-10-01	2024-10-01	Short-term supply contract.	Malini Jayawardena
5	C005	PH005	PC005	2023-08-01	2024-08-01	Annual supply contract.	Keerthi Wijesinghe

Figure 15 Contract Table

b) SQL Queries and Command Prompt Execution

i) Display all patients and doctors in this pharmacy database

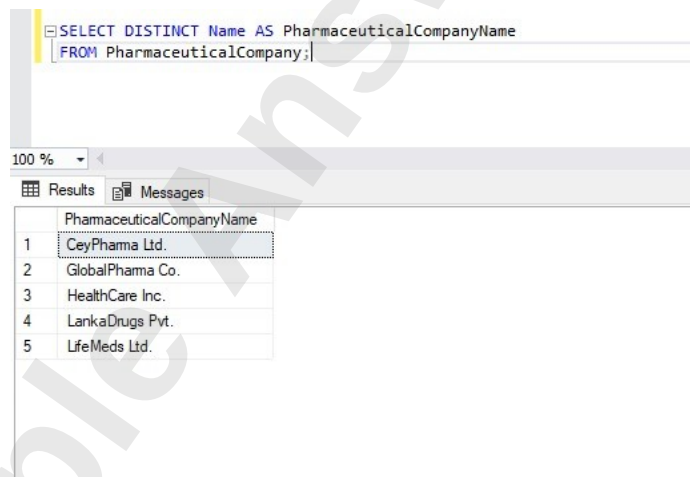


```
SELECT Patient.Name AS PatientName, Doctor.Name AS DoctorName
FROM Patient
LEFT JOIN Doctor ON Patient.PrimaryDoctorID = Doctor.DoctorID;
```

	PatientName	DoctorName
1	Nimal Kumara	Dr. Ajith Perera
2	Sunethra Jayasinghe	Dr. Sunil Fernando
3	Chandana Weerakoon	Dr. Priyanka Silva
4	Rohini Perera	Dr. Malini Jayawardena
5	Kasun Bandara	Dr. Keerthi Wijesinghe

Figure 16 Display all patients and doctors in this pharmacy database

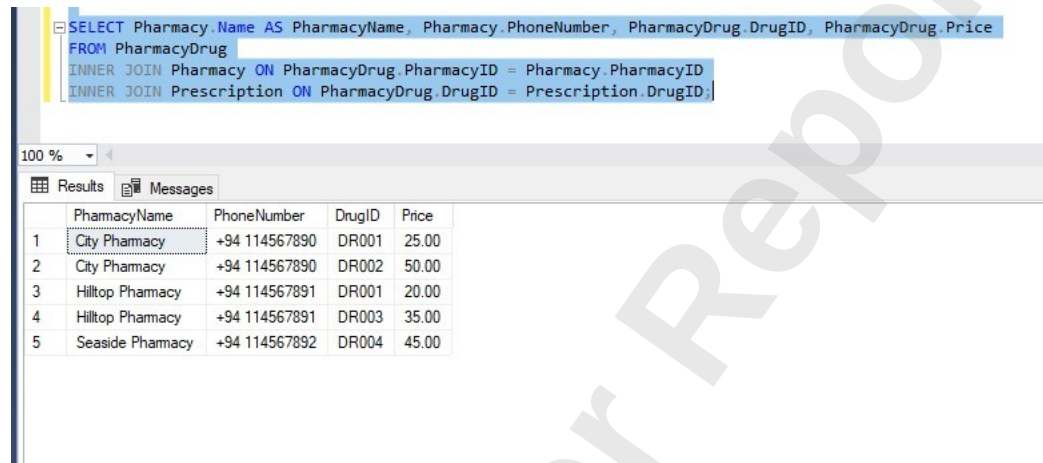
ii) List all unique pharmaceutical companies



```
SELECT DISTINCT Name AS PharmaceuticalCompanyName
FROM PharmaceuticalCompany;
```

	PharmaceuticalCompanyName
1	CeyPharma Ltd.
2	GlobalPharma Co.
3	HealthCare Inc.
4	LankaDrugs Pvt.
5	LifeMeds Ltd.

iii) Retrieve pharmacy contact numbers and drug prices of all drugs which were sold out for the patients

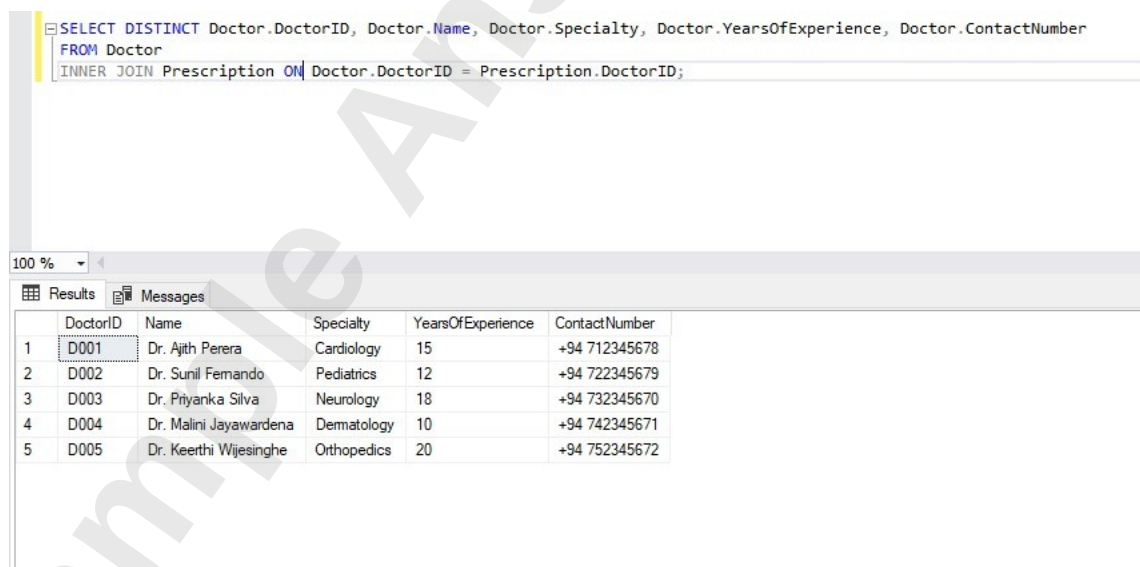


```
SELECT PharmacyName AS PharmacyName, Pharmacy.PhoneNumber, PharmacyDrug.DrugID, PharmacyDrug.Price
FROM PharmacyDrug
INNER JOIN Pharmacy ON PharmacyDrug.PharmacyID = Pharmacy.PharmacyID
INNER JOIN Prescription ON PharmacyDrug.DrugID = Prescription.DrugID;
```

	PharmacyName	PhoneNumber	DrugID	Price
1	City Pharmacy	+94 114567890	DR001	25.00
2	City Pharmacy	+94 114567890	DR002	50.00
3	Hilltop Pharmacy	+94 114567891	DR001	20.00
4	Hilltop Pharmacy	+94 114567891	DR003	35.00
5	Seaside Pharmacy	+94 114567892	DR004	45.00

Figure 17 Retrieve pharmacy contact numbers and drug prices of all drugs which were sold out for the patients

iv) Display doctors' details who prescribed at least a patient

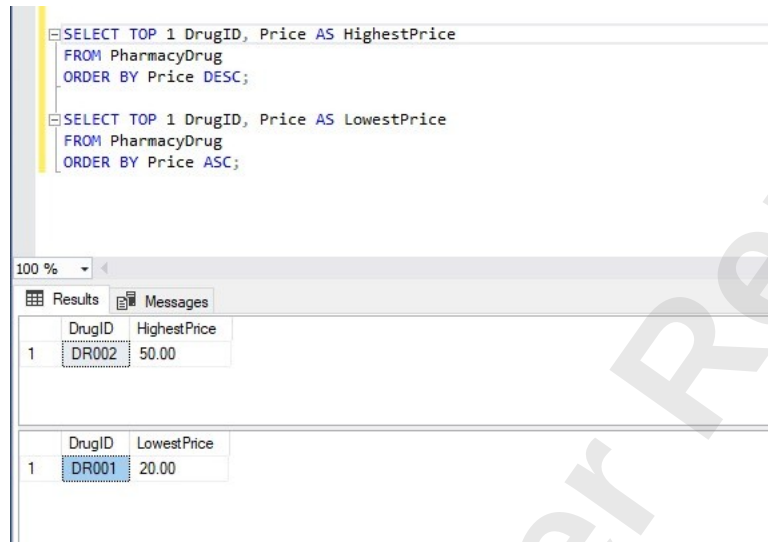


```
SELECT DISTINCT Doctor.DractorID, Doctor.Name, Doctor.Specialty, Doctor.YearsOfExperience, Doctor.ContactNumber
FROM Doctor
INNER JOIN Prescription ON Doctor.DractorID = Prescription.DractorID;
```

	DoctorID	Name	Specialty	YearsOfExperience	ContactNumber
1	D001	Dr. Ajith Perera	Cardiology	15	+94 712345678
2	D002	Dr. Sunil Fernando	Pediatrics	12	+94 722345679
3	D003	Dr. Priyanka Silva	Neurology	18	+94 732345670
4	D004	Dr. Malini Jayawardena	Dermatology	10	+94 742345671
5	D005	Dr. Keerthi Wijesinghe	Orthopedics	20	+94 752345672

Figure 18 Display doctors' details who prescribed at least a patient

v) Find the drug with the highest price and the lowest price



```
SELECT TOP 1 DrugID, Price AS HighestPrice
FROM PharmacyDrug
ORDER BY Price DESC;

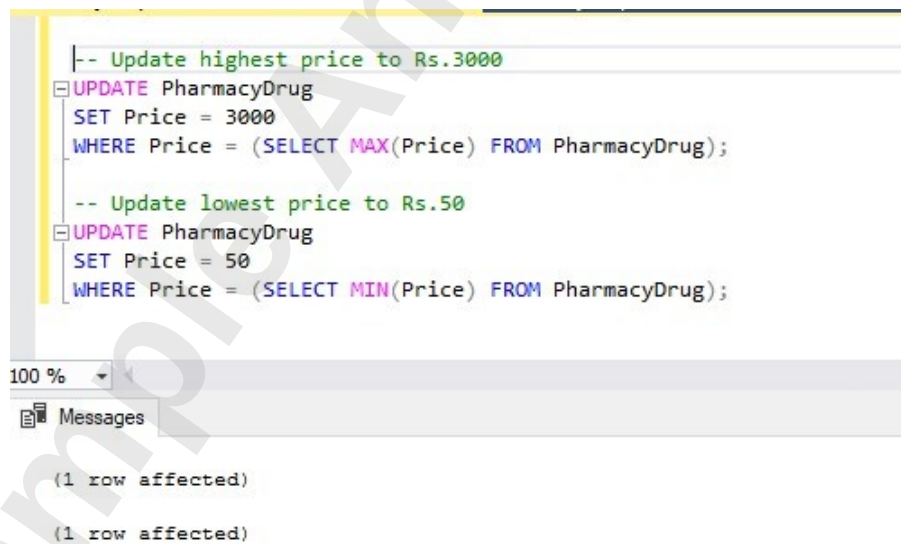
SELECT TOP 1 DrugID, Price AS LowestPrice
FROM PharmacyDrug
ORDER BY Price ASC;
```

	DrugID	HighestPrice
1	DR002	50.00

	DrugID	LowestPrice
1	DR001	20.00

Figure 19 Find the drug with the highest price and the lowest price

vi) Update the highest drug price into Rs.3000 and lowest price into Rs.50



```
-- Update highest price to Rs.3000
UPDATE PharmacyDrug
SET Price = 3000
WHERE Price = (SELECT MAX(Price) FROM PharmacyDrug);

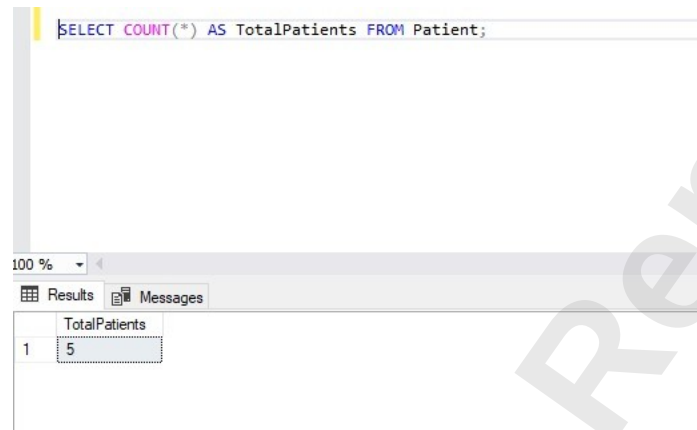
-- Update lowest price to Rs.50
UPDATE PharmacyDrug
SET Price = 50
WHERE Price = (SELECT MIN(Price) FROM PharmacyDrug);
```

{1 row affected}

{1 row affected}

Figure 20 Update the highest drug price into Rs.3000 and lowest price into Rs.50

vii) Count the number of patients who have registered in this pharmacy database



The screenshot shows a SQL query execution window. The query entered is `SELECT COUNT(*) AS TotalPatients FROM Patient;`. Below the query, there are tabs for 'Results' and 'Messages'. The 'Results' tab is active, displaying a single row with the column 'TotalPatients' and the value '5'.

	TotalPatients
1	5

Figure 21 Count the number of patients who have registered in this pharmacy database

(c) Security Mechanisms in the Database

1. User Authentication and Role-Based Access Control

```
CREATE USER 'pharmacist'@'localhost' IDENTIFIED BY 'securepassword';  
  
GRANT SELECT, INSERT, UPDATE ON HealthIsWealthPharmacy.* TO  
'pharmacist'@'localhost';
```

2. Data Encryption

```
INSERT INTO Patient (PatientID, Name, Street, City, ZipCode, DateOfBirth,  
PrimaryDoctorID)  
  
VALUES ('P006', AES_ENCRYPT('Anuradha Bandara', 'key'), '123', 'Colombo', '00100',  
'1980-01-01', 'D003');
```

3. Input Validation and SQL Injection Prevention

```
PREPARE stmt FROM 'SELECT * FROM Patient WHERE Name = ?';  
  
SET @name = 'Nimal Kumara';  
  
EXECUTE stmt USING @name;
```

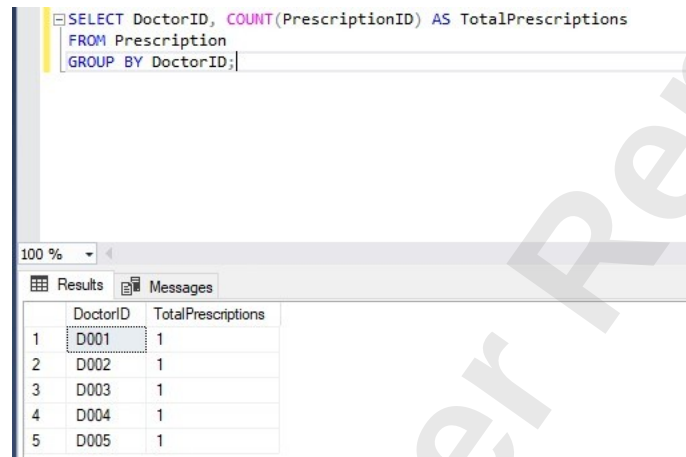
4. Backup and Recovery Mechanisms

Regular backups ensure that the data can be restored in case of corruption or unauthorized access.

(d) Usage of SQL Clauses

I. Group By

Groups rows with the same values in specified columns and aggregates results.



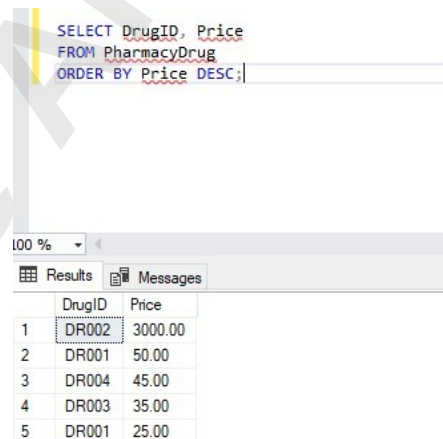
The screenshot shows a SQL query in a text editor and its results in a table. The query is: `SELECT DoctorID, COUNT(PrescriptionID) AS TotalPrescriptions FROM Prescription GROUP BY DoctorID;`. The results table has two columns: DoctorID and TotalPrescriptions. It contains five rows, each with a unique DoctorID and a count of 1.

	DoctorID	TotalPrescriptions
1	D001	1
2	D002	1
3	D003	1
4	D004	1
5	D005	1

Figure 22 Group By SQL Clauses

II. Order By

Sorts the result set in ascending (ASC) or descending (DESC) order.



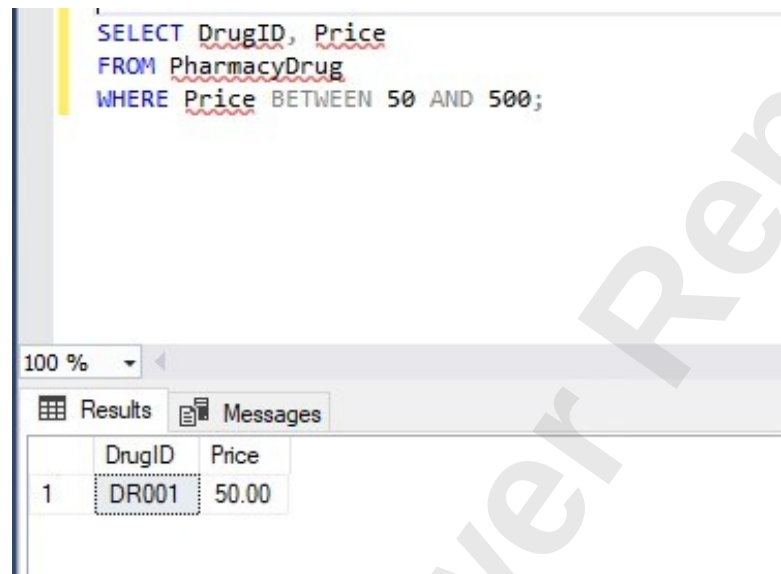
The screenshot shows a SQL query in a text editor and its results in a table. The query is: `SELECT DrugID, Price FROM PharmacyDrug ORDER BY Price DESC;`. The results table has two columns: DrugID and Price. It contains five rows, sorted by price in descending order.

	DrugID	Price
1	DR002	3000.00
2	DR001	50.00
3	DR004	45.00
4	DR003	35.00
5	DR001	25.00

Figure 23 Order By Clause

III. Between

- Filters results within a specified range.



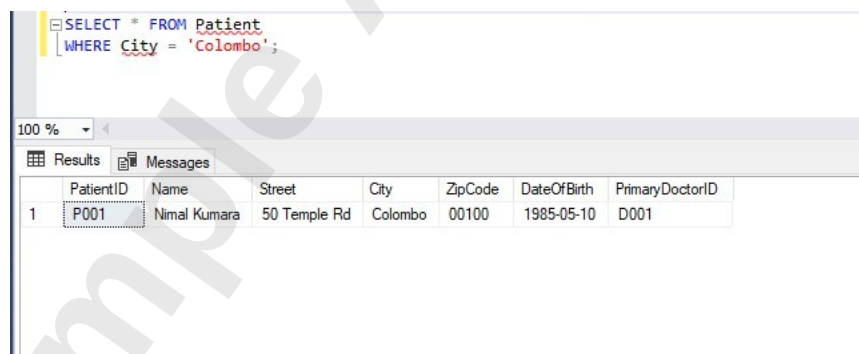
The screenshot shows a SQL query in a text editor: `SELECT DrugID, Price FROM PharmacyDrug WHERE Price BETWEEN 50 AND 500;`. Below the query, the 'Results' tab is active, displaying a table with two columns: 'DrugID' and 'Price'. The first row contains the values '1' and '50.00'.

	DrugID	Price
1	DR001	50.00

Figure 24 Between clause

IV. Where

- Filters rows based on specified conditions.



The screenshot shows a SQL query in a text editor: `SELECT * FROM Patient WHERE City = 'Colombo';`. Below the query, the 'Results' tab is active, displaying a table with seven columns: 'PatientID', 'Name', 'Street', 'City', 'ZipCode', 'DateOfBirth', and 'PrimaryDoctorID'. The first row contains the values '1', 'P001', 'Nimal Kumara', '50 Temple Rd', 'Colombo', '00100', and '1985-05-10'.

	PatientID	Name	Street	City	ZipCode	DateOfBirth	PrimaryDoctorID
1	P001	Nimal Kumara	50 Temple Rd	Colombo	00100	1985-05-10	D001

Figure 25 Where clause

Question 03

a)"Health is Wealth" database system

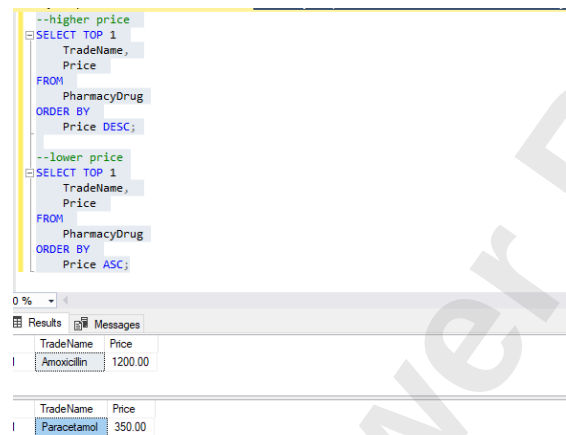
Designed by – Vihanga	Test Date – 1/04/2025
Test case-1 – Retrieve highest and lowest priced drugs	
Testing process Step 1 – Execute queries to retrieve the highest and lowest priced drugs from the PharmacyDrug table	
Test Data (code) -- Higher price SELECT TOP 1 TradeName, Price FROM PharmacyDrug ORDER BY Price DESC; -- Lower price SELECT TOP 1 TradeName, Price FROM PharmacyDrug ORDER BY Price ASC;	
Expected results Expected results Highest priced drug: Amoxicillin – 1200.00	

Lowest priced drug: Paracetamol – 350.00

Actual Results -

```
--higher price
SELECT TOP 1
  TradeName,
  Price
FROM
  PharmacyDrug
ORDER BY
  Price DESC;

--lower price
SELECT TOP 1
  TradeName,
  Price
FROM
  PharmacyDrug
ORDER BY
  Price ASC;
```



TradeName	Price
Amoxicillin	1200.00

TradeName	Price
Paracetamol	350.00

Test status -

Pass

Conclusion -

SQL queries correctly retrieve drugs with the highest and lowest prices from the database.

Designed by –

ihanga

Test Date –

1/04/2025

Test case-2 - Validate prescription data integrity

Testing process

Step 1 – Execute query to retrieve pharmacy names, drug trade names, and their prices

Test Data (code)

```
SELECT TOP (1000) [PharmacyName], [DrugTradeName], [Price]
FROM [HealthIsWealthPharmacy_].[dbo].[Pharmacy_Drug];
```

Expected results

PharmacyName | DrugTradeName | Price

Kurunegala Health Centre | Amoxil | 12.00

Kurunegala Health Centre | Panadol | 5.50

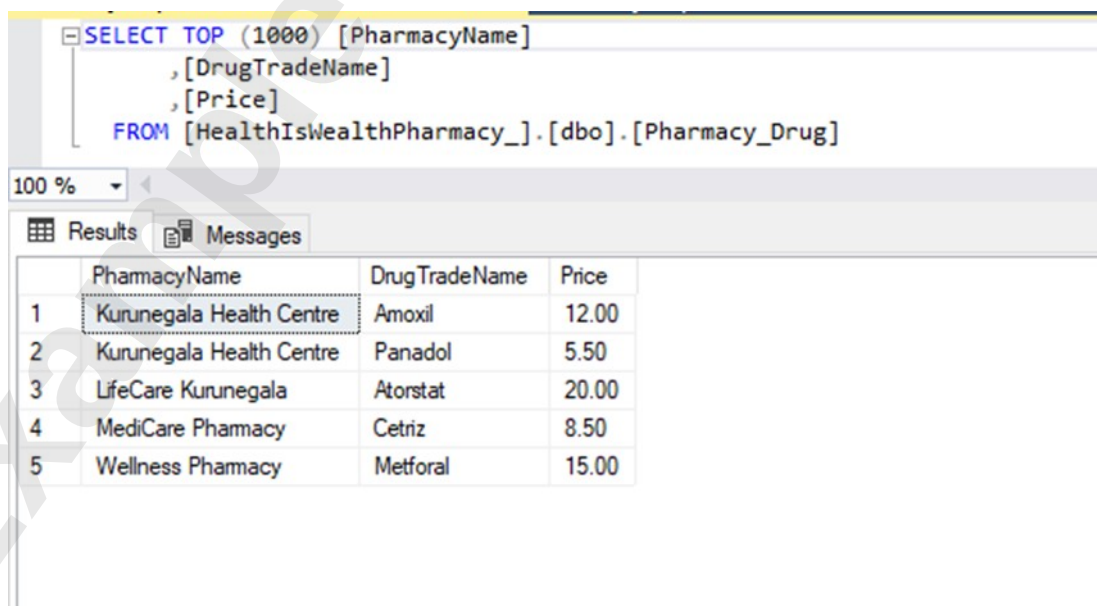
LifeCare Kurunegala | Atorstat | 20.00

MediCare Pharmacy | Cetriz | 8.50

Wellness Pharmacy | Metforal | 15.00

Actual Results -

Correct pharmacy drug details retrieved as expected.



The screenshot displays the SQL Server Enterprise Manager interface. At the top, a query window shows the following SQL statement:

```
SELECT TOP (1000) [PharmacyName]
, [DrugTradeName]
, [Price]
FROM [HealthIsWealthPharmacy_].[dbo].[Pharmacy_Drug]
```

Below the query window, the 'Results' tab is active, showing a table with the following data:

	PharmacyName	DrugTradeName	Price
1	Kurunegala Health Centre	Amoxil	12.00
2	Kurunegala Health Centre	Panadol	5.50
3	LifeCare Kurunegala	Atorstat	20.00
4	MediCare Pharmacy	Cetriz	8.50
5	Wellness Pharmacy	Metforal	15.00

Test status - Pass
Conclusion - SQL query successfully retrieves pharmacy, drug, and price details from the database.

Designed by – Vihanga	Test Date – 1/04/2025
Test case-3 - Test retrieval of doctor data with SQL query	
Testing process Step 1 – Execute query to retrieve distinct doctors who have issued prescriptions	
Test Data (code) SELECT DISTINCT Doctor.SSN, Doctor.Name, Doctor.Specialty, Doctor.YearsOfExperience FROM Doctor JOIN Prescription ON Doctor.SSN = Prescription.DoctorSSN;	
Expected results SSN Name Specialty YearsOfExperience 123456789V Dr. Nuwan Perera General Physician 10 345678901V Dr. Aruni Jayawardena Dermatologist 7	

567890123V | Dr. Kasun Wickramasinghe | Cardiologist | 12

876543219V | Dr. Thusitha Fernando | Orthopedic Surgeon | 15

987654321V | Dr. Chathur Silva | Pediatrician | 8

Actual Results -

Correct data retrieved

```
SELECT DISTINCT Doctor.SSN, Doctor.Name, Doctor.Specialty, Doctor.YearsOfExperience  
FROM Doctor  
JOIN Prescription ON Doctor.SSN = Prescription.DoctorSSN;
```

	SSN	Name	Specialty	YearsOfExperience
1	123456789V	Dr. Nuwan Perera	General Physician	10
2	345678901V	Dr. Aruni Jayawardena	Dermatologist	7
3	567890123V	Dr. Kasun Wickramasinghe	Cardiologist	12
4	876543219V	Dr. Thusitha Fernando	Orthopedic Surgeon	15
5	987654321V	Dr. Chathuri Silva	Pediatrician	8

Test status -

Pass

Conclusion -

SQL query correctly retrieves distinct doctor information associated with prescriptions from related tables.

Designed by – Vihanga	Test Date – 1/04/2025
Test case-4 – Test update operation on highest drug price & Test update operation on lowest drug price	
Testing process Step 1 – Update the highest drug price to 3000 in the Pharmacy_Drug table Step 2 – Update the lowest drug price to 50 in the Pharmacy_Drug table	
Test Data (code) UPDATE Pharmacy_Drug SET Price = 3000 WHERE Price = (SELECT MAX(Price) FROM Pharmacy_Drug);	
Expected results The drug with the maximum price is updated to 3000 in the Pharmacy_Drug table. The drug with the minimum price is updated to 50 in the Pharmacy_Drug table.	
Actual Results - Price updated successfully	

```

UPDATE Pharmacy_Drug
SET Price = 3000
WHERE Price = (SELECT MAX(Price) FROM Pharmacy_Drug);

UPDATE Pharmacy_Drug
SET Price = 50
WHERE Price = (SELECT MIN(Price) FROM Pharmacy_Drug);

```

100 %

Messages

(1 row affected)

(1 row affected)

Test status -

Pass

Conclusion -

Database correctly updates the maximum priced drug as intended

Database correctly updates the minimum priced drug as intended.

Designed by – Vihanga	Test Date – 1/04/2025
Test case-5 – Validate retrieval of pharmacy details by phone number	
Testing process Step 1 – Attempt to retrieve pharmacy details using a specific phone number	
Test Data (code)	

```
SELECT Name, Address
```

```
FROM Pharmacy
```

```
WHERE PhoneNumber = '+94-372-556677';
```

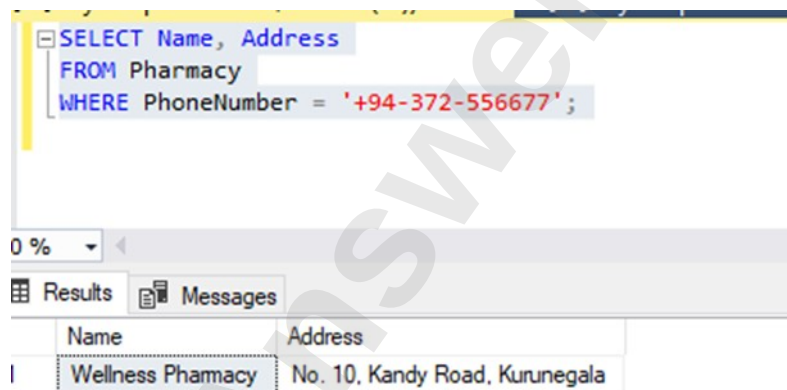
Expected results

NameAddress

Wellness Pharmacy No. 10, Kandy Road, Kurunegala

Actual Results -

Error received as expected

**Test status -**

Pass

Conclusion -

Pharmacy details are correctly retrieved based on the provided phone number.

b) Database Testing Techniques and Assessment

Database testing is a critical process to ensure the integrity, reliability, and performance of a database. It involves testing the backend of a database system to validate its structure, data, queries, and associated functionalities. The main goal is to identify and fix any errors, inconsistencies, or inefficiencies that may impact the database's functionality. Below, I will explain various database testing techniques and assess one technique implemented during the development of the "Health is Wealth" pharmacy database.

1. Types of Database Testing Techniques

1.1. Structural Database Testing

This technique focuses on testing the schema, tables, columns, keys, and relationships within the database. It ensures that:

- All tables and their fields are correctly defined.
- Primary keys and foreign keys are implemented and enforce referential integrity.
- Indexes are applied to optimize query performance.

For example, in the "Health is Wealth" database, structural testing verifies that foreign keys like PrimaryDoctorID in the Patient table reference the DoctorID in the Doctor table, and indexes are created for frequently queried fields like DrugID.

1.2. Functional Database Testing

Functional testing validates that the database behaves as expected for different operations, including data insertion, updating, deletion, and retrieval. This type of testing ensures that:

- Queries return the correct results.
- Constraints (e.g., NOT NULL, UNIQUE, FOREIGN KEY) are enforced.
- Stored procedures, triggers, and views function as intended.

In the pharmacy database, functional testing checks that a query for retrieving patients and their assigned doctors correctly returns the expected results, even when some patients have no assigned doctors.

1.3. Data Integrity Testing

Data integrity testing ensures that the data stored in the database is accurate, consistent, and valid. This includes checking:

- Consistency between related tables (e.g., Prescription references valid PatientID, DoctorID, and DrugID).
- Correct enforcement of constraints, such as ensuring that drug prices in the PharmacyDrug table are positive numbers.

For instance, testing verifies that a contract cannot be created without linking it to both a pharmacy and a pharmaceutical company.

1.4. Performance Testing

Performance testing measures how well the database handles large volumes of data and concurrent user operations. It includes:

- Query execution times.
- Load testing to evaluate performance under heavy traffic.
- Stress testing to determine the breaking point of the database.

For example, testing the "Health is Wealth" database with thousands of records ensures that queries, such as retrieving the drug with the highest price, execute within an acceptable timeframe.

1.5. Security Testing

Security testing ensures that sensitive data is protected and unauthorized access is prevented.

Key aspects include:

- User authentication and role-based access control.
- Prevention of SQL injection attacks.
- Encryption of sensitive data, such as patient details.

In the pharmacy database, testing ensures that only authorized users can access patient and prescription information.

1.6. Regression Testing

Regression testing ensures that changes to the database, such as schema updates or new queries, do not break existing functionalities. It involves re-executing previous test cases to validate the system's behavior.

2. Implementation of Functional Database Testing

During the development of the "Health is Wealth" database, **functional database testing** was implemented to validate the accuracy and reliability of queries and data manipulation operations. Below is a detailed assessment of the process:

Objective

The objective of functional testing was to verify that queries, constraints, and relationships worked correctly and returned accurate results based on user and system requirements.

Steps Taken

1. Testing Queries

- o A query was written to retrieve all patients and their assigned doctors:

SELECT Patient.Name AS PatientName, Doctor.Name AS DoctorName

FROM Patient

LEFT JOIN Doctor ON Patient.PrimaryDoctorID = Doctor.DoctorID;

- o This query was executed to ensure it returned the correct results, even when some patients did not have an assigned doctor.

2. Testing Constraints

- o The database was tested to ensure constraints, such as foreign keys and NOT NULL, were properly enforced.
- o For example, an attempt to add a new patient without assigning a primary doctor resulted in an error, as expected:

ERROR: Cannot add a record to the Patient table because PrimaryDoctorID is missing.

3. Testing Data Manipulation

- o Data insertion, updates, and deletions were tested to ensure the integrity of the data.
- o For instance, updating a drug price in the PharmacyDrug table was verified:

UPDATE PharmacyDrug

SET Price = 3000

WHERE DrugID = 'DR001';

- o This test confirmed that only the intended record was updated.

4. Testing Relationships

- o Relationships between tables were tested to confirm referential integrity.
- o For example, deleting a record in the Doctor table was tested to ensure that it did not violate constraints for patients referencing that doctor:

ERROR - Cannot delete record in Doctor because it is referenced in Patient.

Results

The functional testing process confirmed the following:

- Queries returned accurate and complete results.
- Constraints prevented invalid data entry and ensured referential integrity.
- Data manipulation operations (insert, update, delete) worked as intended without introducing errors.

Question 04

a) Produce a fully technical

The “Health is Wealth” pharmacy database system is effective solution which is helpful in solving operational issues, managing relationships with patients, doctors, pharmacies, drugs, and pharmaceutical companies and also helps in accurate management of information related to patients, doctors, available pharmacy, drugs, and companies related to it. Though the current system enables satisfying existing requirements, there are several future improvements to increase the system’s performance and adaptability to changes in operational demands. To these enhancements, there is the need to ensure that technical and user documentation enables provision of clear instructions relating to the management of the system and its use.

In more detail, from a technical point of view, the database must be designed with complex additional features for accommodating advancing data size and complexity. It is beneficial to optimize indexing for more often searched fields that include, DrugID, PatientID and DoctorID can greatly improve the performance of the queries. Moreover, enhancing features like forecast modeling for the demand of a new drug or patients can be offered to the key-decision makers. Security measures must also adapt to new threats for example, identification mechanisms must also have to go through changes. Encryption for the field containing records like patient information and strong user authentication: the possible implementation of two-level authentication will meet global standards of data protection .

It could also be a factor of efficiency improvement; for instance through automation. The reminder notification for contacting a client about the expiration of their contract or fill a prescription can be carried by the system instead of draining the employees’ resources. An example of triggers includes the use of notifications that alert pharmacies that stock of a certain drug is low, to prevent disruptions. In addition, extending the ability of the system to interface with external systems, including supplier systems or electronic health record systems, can improve data sharing and work process efficiency.

End-user documentation is equally important for the technical staff are not the only ones to be using the system. It was established that the input screens for tasks comprising entering patients records, prescription, and reports generation will be clear and with clear instructions in a sequence. Blogging tools, whether they are screenshots, flowcharts, graphic organizers,

diagrams or drawings make distilled poor graphics easy to comprehend. For example, a user guide that will determine how to change drug price or how to obtain contracts from the system will assist staff perform their duties in the ACE system without Technical IT skills. The technical documentation should contain specific information concerning the structure and architecture of a database, dependencies and restrictions. It also means that developers will be able to sustain and enhance the system in future due to better and more proper documentation. It should also capture issues to do with backup and recovery of the system as well as data validation during system crash. Technical staff will refer to descriptions of queries, stored procedures and performance tuning that will enable them to adapt to changing needs in the future.

With such future improvements as well as better documentation, the “Health is Wealth” pharmacy database will therefore continue to serve the increasing needs of a constantly evolving health care practices as a functional, efficient and secure pharmacy information system.

b) Technical and User Documentation

The Health is Wealth pharmacy database system ease the organization's data handling and operational functionality such as inserting data, modifying data and logging off. For the documentation, it recommends system processes' description along with flowcharts to support their function.

Technical Documentation

The technical documentation includes the description of the system architecture, the structure of the database, data transfer. These include the associations between the Patient, Doctor, Pharmacy, Drug and Pharmaceutical Company. From the various data flow diagrams, considerable details are given on how data moves between different user interfaces and the database. For instance,

1. **Update Function-** Users access records via a primary key (e.g., DrugID) and modify specific fields. The system validates inputs, checks constraints, and executes an UPDATE query.
2. **Insert Data** - Users input details for new records. The system validates constraints (e.g., unique keys and foreign key relationships) and executes an INSERT query to add data to the respective table.
3. **Logout-** The logout function ensures user sessions are securely terminated by clearing session variables and redirecting users to the login screen.

User Documentation

User documentation simplifies tasks for non-technical users. Step-by-step instructions with screenshots guide users on:

- **Data Entry-** How to use forms for entering patient details, prescriptions, or contracts.
- **Updates-** Procedures for modifying existing records, such as drug prices.
- **Logout-** Instructions to securely exit the system.

Flowchart Details

a. Update Function

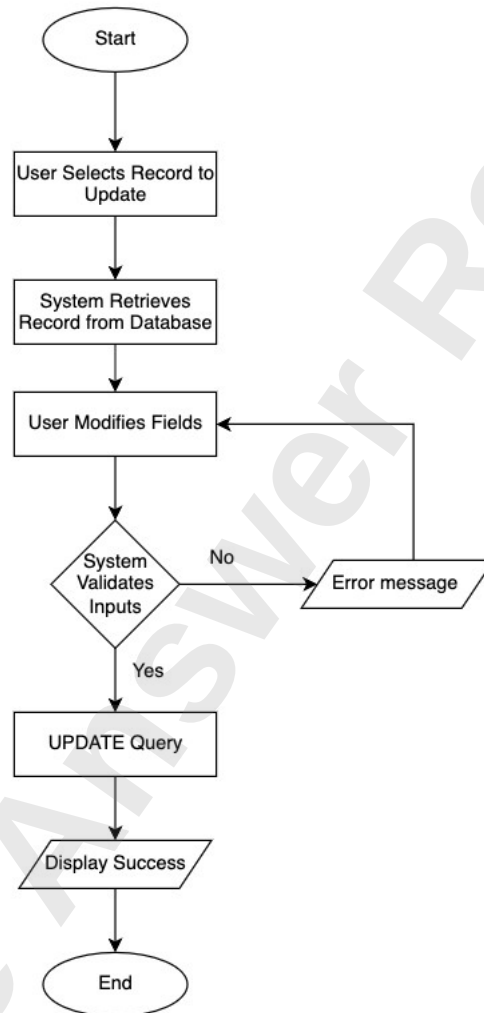


Figure 26 Flow Chart for Update Function

b. Insert Data

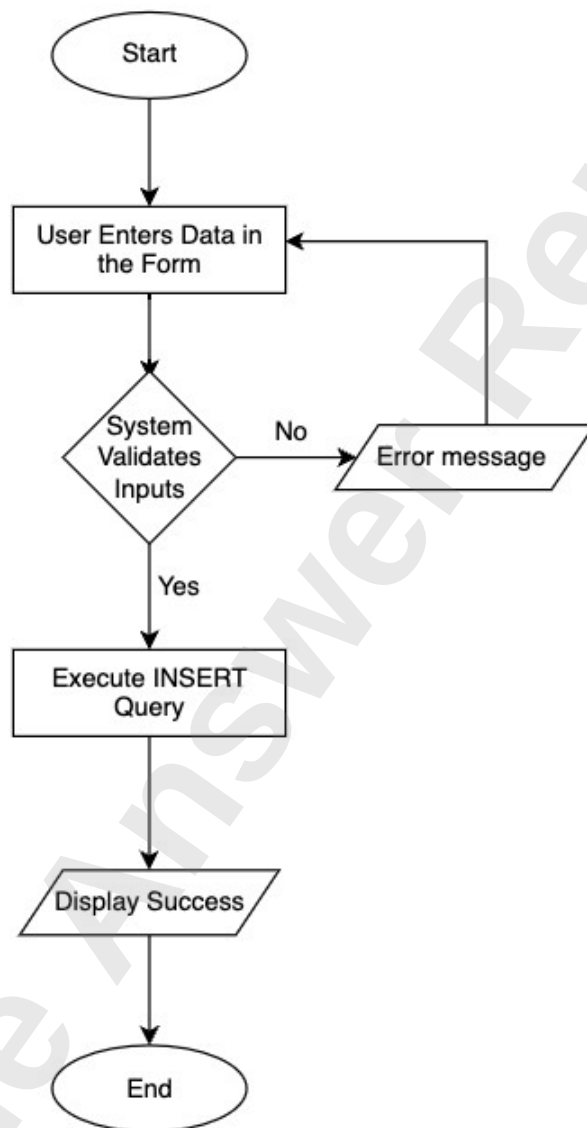


Figure 27 Insert Data Flowchart

c. Logout Function

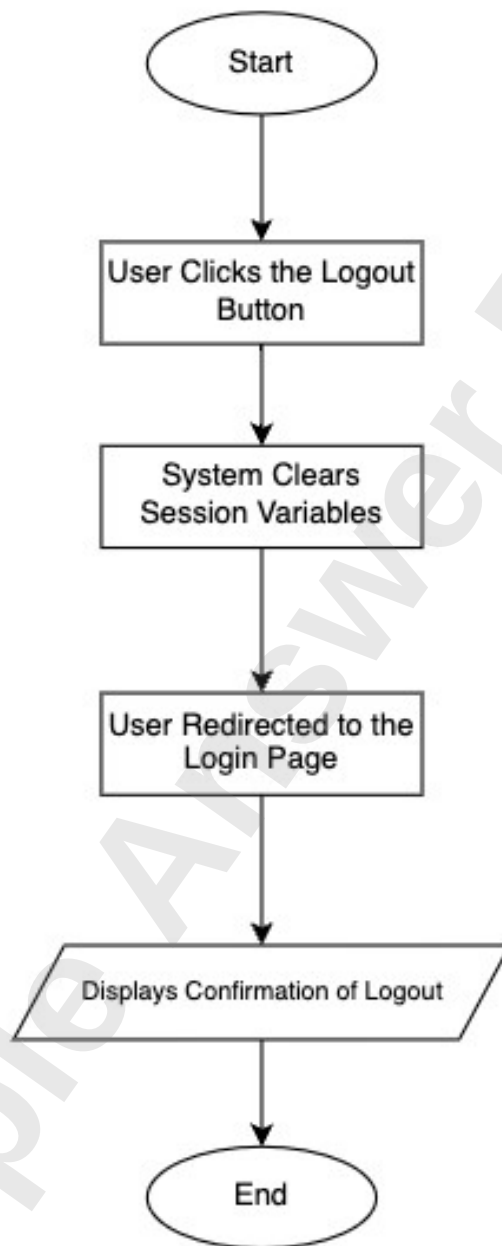


Figure 28 Flowchart for logout function