University of Maryland Global Campus

DBST 651

Technical Report

Database Design Document for the Pharmacy Management System

Student: Pang-Kuo Lo

Date: 12/11/2022

Table of Contents

1. Introduction

This technical report documents the entire process for designing and developing a simulated database intended for being integrated with a pharmacy management software system. Due to the need for the automation of routine pharmacy processes related to prescribed medication management, a well-designed database for storing the information data of prescribed medication, prescriptions, pharmacies, patients, and doctors is essential for the pharmacy management software system to input these data into and retrieve them from the database for powerful data analytics. Through the pharmacy management software system, three types of users including pharmacies, doctors, and pharmaceutical companies can interact with the pharmacy management database for entry and retrieval of data. The expected evolution of the developed database technology is to build a database managed by a RDBMS as the backend of a web-based pharmacy management system software implemented by HTML, CSS, Javascript, and PHP. Security and privacy are two key challenges that electronic health/medical systems need to face. The privacy considerations involve the privileges for users to have access to and modify specific entities/data in the database. Data security is the main focus in the security considerations for database development. Multiple strategies to implement data security are discussed in this technical report.

2. Overview

To automate the entire process of prescription and pharmacy management, the development and application of an electronic management software system for medical prescriptions and pharmacy operations is necessary. To make the electronic pharmacy management software system functional, its interaction and integration with a database that can store data related to patients, physicians, pharmacies and medications would be needed. Therefore, the design and the requirements from the pharmacy management software system would determine the database design.

There are two main dependencies for database design and development. The first dependency is the pharmacy management system that interacts with the database. The second dependency is the database management system (DBMS) that serves as an interface platform to communicate between the pharmacy management software system and the database. In the project, Oracle Relational Database Management System (RDBMS) was used to set up and manage the simulated database. To ensure that the pharmacy management system has full functionality and meets business requirements, a DBMS needs to be chosen first before designing and developing the pharmacy management system. Parallel development of both the pharmacy management software system and the database according to the chosen DBMS is an ideal strategy. Good communications and tight collaborations between two functional teams for development of respective management software and the database are also essential for the success of the goal in pharmacy management applications.

3. Literature Review

This project mirrors the electronic prescription system that has been developed for a long time. An electronic prescription system is a computerized, internet-dependent mechanism that has been practised

around the world (Tamblyn et al., 2006; Vejdani et al., 2022). The benefits of the electronic prescription system include:

- improving the accuracy and efficiency of prescribing and dispensing medications.
- providing the better quality of health care services and decreasing health care costs.
- improving patient safety by preventing medication errors and adverse drug reactions.
- monitoring whether prescription drugs are appropriately prescribed to prevent prescription abuse and overprescribing.
- saving time for patients, doctors, and pharmacists due to the effectiveness of the prescribing and drug-dispensing process.

Due to the importance of the electronic prescription system in health system modernization, Vejdani et al. conducted literature searching to review the standards, requirements, and specifications of an electronic prescription system (Vejdani et al., 2022). After searching multiple literature databases including Web of Science, PubMed, Scopus, and ProQuest with no time limit, 13 articles were selected and reviewed. The ideal electronic prescription system includes the following main steps:

- A user (e.g., a physician) signs on the electronic prescription management system
- The physician identifies the patient in the electronic prescription management system and reviews the available information data for the patient.
- The physician performs drug selection and medication parameter entry (e.g., drug quantity, drug dose), and the e-prescription is approved by e-signature of the physician.
- The approved e-prescription is sent directly to the pharmacy for distribution via internet connection.

The perspectives from this review provide the useful information and considerations for this project. This review summarizes requirements for the electronic prescription management system, including:

- A detection and correction system is required for reducing errors in the electronic prescription management system.
- Providing safety alerts and filtering user-selectable alerts are needed to minimize possible prescribing problems (e.g., drugs causing patient's allergy, adverse effects from drug interactions).
- Providing computer-assisted dose calculations can improve prescribing accuracy. This capacity needs the electronic prescription management system to access patient's medical records (e.g., age, weight, BMI, diagnostic results, etc.).
- The electronic prescription management system can securely and reliably transfer the e-prescription data to the database server that can be accessed by pharmacies to obtain e-prescriptions.
- Data security and confidentiality are important requirements for implementing the electronic prescription management system.
- The electronic prescription system can provide precise, clear drug lists for physicians to perform drug selection for patients.
- The electronic prescription management system enables physicians to access the medication history of the patient.

- The implementation of an electronic prescription management system needs to be transparency and accountability.

This review concludes that the functional and technical capabilities of the electronic prescription management system can significantly benefit stakeholders, service providers, drug distributors, patients, and insurance organizations if it is used correctly (Vejdani et al., 2022). These requirements identified by this review are critical for the design and development of both electronic prescription management systems and associated databases.

4. Assumptions/Constraints/Risks

4.1 Assumptions

The end-users use the pharmacy management software system to access the database system for the entry and retrieval of data related to the pharmacy chain process. Three main types of end-users for the electronic pharmacy management software system are doctors, pharmacists, and pharmaceutical companies. End-users use the electronic pharmacy management system installed in desktop or laptop computers with Windows OS (Windows 10 or 11) to perform electronic prescribing and drug-dispensing processes. Pharmaceutical companies or their wholesalers as well as contracted pharmacies can update the drug prices through this system. The whole system architecture is that end-users use the electronic pharmacy management software system installed in the local computers to interact with the database stored in the local or cloud-based server through the internet connection. To enable the electronic pharmacy management software system to interact with the database, a database managed by a database management system (e.g., the Oracle Relational Database Management System) needs to be the backend of a web-based pharmacy management software system implemented by HTML, CSS, Javascript, and PHP.

4.2 Constraints

As the project limits maximal six entities in the simulated database, this constraint restricts the functionality and usefulness of the database. Due to the complexity of real-world prescribing and pharmacy-chain processes, the six entities including doctor, patient, drug, pharmacy, prescription, and prescribed drug, in the database are not sufficient to meet the needs from the electronic pharmacy management software system. To achieve the goal for improving accuracy, safety, efficiency, and communication in transactions between pharmaceutical manufacturers, pharmacies, doctors, and patients, including more entities/tables in the database will be necessary.

4.3 Risks

As the database used by the pharmacy management software stores sensitive data (e.g., patient's SSN, DOB, address, and card number), the biggest risk associated with the database is its vulnerabilities in data security and protection. There are multiple threats causing database vulnerabilities, such as database injection attacks, malware, excessive privileges for end-users, legitimate privilege abuse, exposure of storage media for the database, unmanaged sensitive data, the human factor, etc. The defensive

approaches for best practices and internal controls are necessary for the protection of databases. These approaches (Maurer, 2015) include:

- Identifying any database vulnerabilities as well as potentially compromised endpoints and classifying sensitive data.
- Using anti-malware software and secure browsers to block malicious web requests.
- Properly managing user access rights and getting rid of excessive privileges.
- Automatically auditing database uses as well as transactions using a database auditing and protection platform to identify data leakage, unauthorized SQL and big data transactions.
- Encrypting data stored in databases and archiving external data.
- Real-time monitoring of all database access and usage activities to detect protocol and system attacks.
- Training end-users to recognize common cyber threats and to implement risk-mitigation techniques.

5. Design Decisions

5.1 Key Factors Influencing Design

The key factors and business requirements influencing the database design involve transactions that include: (1) Doctors prescribe medications for patients via the electronic pharmacy management system; (2) Pharmacists prepare prescribed medications for patients according to the electronic prescriptions; (3) Pharmacies stock and sell prescription drugs through the contracts with pharmaceutical drug manufacturers or their representative wholesalers; (4) Pharmaceutical companies manufacture medication drugs and distribute them through their wholesalers or directly through pharmacies. Based on these key factors, six entities have been designed for the database, including DOCTOR, PATIENT, DRUG, PHARMACY, PRESCRIPTION, and PRESC_DRUG. Database deliverables include the created database, the database documentation such as database requirements and their business rules, entity relationship diagram, relational model diagram and entity/table structures, and DDL, DML and SQL scripts to simulate the above transactions. In the real-world application, the contracted IT company develops the electronic pharmacy management software system to interact with the database for implementing the above transactions.

5.2 Functional Design Decisions

The functional design of the database is based on its interaction with the electronic pharmacy management system. The users use the secure login system built in the electronic pharmacy management system to log on the management system platform. Doctors use the pharmacy management software system to enter the prescription information, which is converted into SQL queries by the pharmacy management system to update and/or modify the database. Pharmacists can have access to prescriptions stored in the updated database through the pharmacy management system that converts pharmacists' requests into SQL queries for retrieving the prescription information. Pharmaceutical companies can also

add new drug information to the database and update the current drug information in the database through the pharmacy management system that converts transactions into SQL queries. The DBMS processes SQL queries from users to trigger the output data from the database. The outputs from the database via the DBMS are processed by the pharmacy management system to display designed results (e.g., prescription, prescribed medication) to users.

5.3 Database Management System Decisions

The Oracle Database 18c Expression Edition (Release 18.0.0.0.0 - Production Version 18.4.0.0.0) is selected as the RDBMS for establishing the simulated database due to my familiarity with this database management system and its free for the use. The enterprise edition of the Oracle database with more security and stability should be purchased and used in the real-world application. The Oracle SQL developer (Version 21.2.1.204.1703, Build 204.1703) is used to build a database by connecting to the Oracle Database 18c. The SQL syntax (e.g., ALTER) from DDL is used to add and modify fields of tables in the database. When the change in business requirements for database design occurs, the entity-relationship diagram (ERD) will be revised to include new entities in the database for capturing the needs from the new requirements.

5.4 Security and Privacy Design Decisions

There are two layers of security in the pharmacy management system designed to protect prescription and pharmacy data in the database. The first layer of security is the login system that checks passwords provided by end-users and grants them access to the pharmacy management system after the provided passwords are authenticated by the login system. After users log in the pharmacy management software system, the graphical user interface (GUI), the second layer of security in the system, converts the user-input information into SQL queries and sends these SQL statements to the back-end database server for performing transactions. This design prevents end-users from direct access to the database.

There are three types of end-users to use the pharmacy management system, including doctors, pharmacists, and pharmaceutical companies. To maintain data security and protect data privacy, the pharmacy management system grants different privileges to different types of users. Under the pharmacy management system, doctors can view, add and modify data in DOCTOR, PATIENT, PRESCRIPTION, and PRESC_DRUG tables, but can only view data in DRUG and PHARMACY tables; Pharmacists can view, add and modify data in PHARMACY, PRESCRIPTION, and PRESC_DRUG tables, but can only view data in DOCTOR, PATIENT, and DRUG tables; Pharmaceutical companies can view, add and modify data in the DRUG table, but can only view data in PHARMACY and PRESC_DRUG tables, and they are not allowed to have access to data in DOCTOR, PATIENT, and PRESCRIPTION tables. The contracted IT company is responsible for having appropriate firewalls in place to protect the application software and DBMS system.

5.5 Performance and Maintenance Design Decisions

To maintain the data accuracy and secure database performance, the maintenance of database consistency is critical. The current best practices to maintain database consistency are to make sure that the transactions between end-users through the pharmacy management system and the database comply with ACID (Atomicity, Consistency, Isolation, and Durability) properties. The ACID properties are measures to ensure the accuracy and consistency of a database by managing each transaction as a single unit, producing consistent and reliable output results, isolating each transaction from others, and making sure that updates to the database are durably stored.

Due to the multi-user database used by the pharmacy management system, the potential concurrency issues such as lost updates, access to uncommitted data, non-repeatable reads, and Phantom reads need to be prevented and resolved. Setting up the property of serializability for concurrent transactions is critical for preventing the concurrency issues. Concurrency control algorithms, such as locking or time-stamping methods, are used to establish a scheduler to execute concurrency control for enforcing the serializability of concurrent transactions. The concurrency control ensures that these concurrent transactions are executed in the appropriate order.

To ensure continuity of the electronic pharmacy management process, the restoration of the database from the valid backup is crucial when the database is corrupted. Different DBMS platforms have their ways for backup and database restoration. There are three types of backups for Oracle databases, including logical backups, physical offline backups, and physical online backups. Database administrators need to determine what components (e.g., parameter files, control files, data files, redo log files, network files, and password files) of the database and other software applications (e.g., OS, RDBMS) need backups, what media (e.g., disk, cloud storage) are used to store database backups, and an appropriate backup schedule and window. The database backup needs to be tested for backup restoration to ensure that it is a valid backup. To document the agreement between end-users and the service provider for the electronic pharmacy management system for the database backup and restoration, a Service Level Agreement (SLA) is created by the database administration team and approved by management to cover details of backup procedures and a timeline for recovery.

To resolve the growing data in a database, archiving data is a necessary strategy. The popular data archiving strategy is to transfer prior data to other databases to resolve accumulating data over the time. The second strategy is to use database partitions. Each database partition is specifically designed to store different time and/or geographic data. The second strategy for data archiving is suitable for the database used in the electronic pharmacy management system as a large number of end-users are expected to use prescription and pharmacy chain data and multiple database partitions are designed for different time (e.g., year) and different geographic areas (e.g., different states). The end-users can have access to the time-specific and geography-specific data through selection from the drop-down menu of the electronic pharmacy management system.

6. Statement of Work (SOW)

6.1 Overview/Executive Summary

The development and application of a pharmacy management software system, also called the pharmacy information system, has facilitated the automation of routine pharmacy processes related to prescribed medication management, trading medicines within pharmacy chains, relationships with medical suppliers, etc. To make the pharmacy management system functional and workable, integration of this management software system with a well-designed database is essential. As a well-designed database enables users to leverage retrieved data of patient information, prescription transactions, payment records, supply chain information, drug sales, and others for powerful business analytics. Many benefits from the database use include: 1) Pharmacy inventory reports generated from the data in the database can help to prevent drug shortages. 2) The database allows medical professionals to track prescription activity for the detection of at-risk medicine cases. 3) Business revenue reports and gap analysis derived from data in the database can propel the industrial development in pharmaceutical markets. Due to the importance of the pharmacy database, this project is intended to develop and implement this database for simulating this application.

6.2 Objectives of the Database Project

The database will interact with the pharmacy management system to track and store pharmacy-related information data such as drug manufacturers (pharmaceutical companies), pharmacies, prescriptions, prescribed drug medication, patients, doctors, etc. Three main types of users can have access to and interact with this database through the software platform of the pharmacy management system, including pharmacists, doctors, and pharmaceutical companies. Through the pharmacy management software, the user can provide data to the database and query data stored in the database based on their privileges for specific entities (e.g., doctors can only have access to the information data related to themselves, their patients and their written prescriptions, but not other information).

6.3 Project Scope

Although the planned database needs to be integrated with the pharmacy management system software suite for showing its database functionality and benefits, the scope of this project can only encompass the design and implementation of the database. The in-scope work of this project will cover business requirements and rules, conceptual and logic data modeling of the database using the entity relationship diagram (ERD) method, the conversion of the logic ERD into **D**ata **D**efinition **L**anguage (DDL) scripts for the definition and implementation the database, and the entry of example data into the database using **D**ata **M**anipulation **L**anguage (DML) and **S**tructured **Q**uery **L**anguage (SQL) scripts for the database demonstration. While the real-world implementation of this database needs to interact with the pharmacy management software, the purpose of this project only focuses on designing and implementing a database for simulating its application. How the database interacts with pharmacy management software is outside of the scope of this project.

6.3.1 In-Scope Work:

- Documentation of project requirements and rules.
- Modeling of the database usingERD.
- Conversion of ERD into DDL scripts.
- Entry of example data using DML scripts.
- Demonstration of the database using example SQL scripts.
- The preparation of the deliverable comprehensive report.

6.3.2 Out-of-Scope Work:

- Development of a pharmacy management system software suite directly interacting with the created database.
- Development of an interface software suite allowing a currently used pharmacy management system to interact with the created database.
- Demonstration of how the created database interacts with and support the pharmacy management system.

6.4 Database Goals, Expectations and Deliverables

The database goals and expectations from this project are intended for simulating the following real-world events/transactions occurring in the pharmacy chain system:

- Doctors prescribe medications for patients via the electronic pharmacy management system.
- Patients gets medications from pharmacies.
- Pharmacists prepare medications for patients according to the electronic prescriptions.
- Pharmacies stock and sell prescription drugs through the contracts with pharmaceutical drug manufacturers or their representative wholesalers.
- Pharmaceutical companies manufacture medication drugs and sell them through their wholesalers or directly through pharmacies.

Database deliverables encompass the created database, the database documentation such as database requirements and their business rules, entity relationship diagrams and entity/table structures, and DDL, DML and SQL scripts to simulate the above transactions.

6.5 Database Benefits

- Through the analytic assessments of the database, pharmaceutical companies can track selling of their manufactured drugs for assessing their annual revenues and the selling performance of contracted pharmacies.
- According to data stored in the database, pharmacies can track the amounts of prescribed drugs for medication to determine the need of drug stocking and monitor drug selling statistics for

determining what drugs to be introduced to pharmacy stores for selling to improve pharmacy revenues.

- Through integrating the database with the pharmacy management system software suite, pharmacists can avoid error-prone handwritten prescriptions and instantly get error-free electronic prescriptions from medical providers. This advantage can improve prescription management of refills, modifications, or cancellations.
- Through the database, doctors and pharmacists can track patients' previous prescriptions (patients' historical records) and whether there are any new drugs in use.

6.6 Project Hardware and Software Tools

As this project only involves the simulation environment for working on database design and implementation, the UMGC Virtual Lab Access Resources will be used for the entire process of this project. Given that only the UMGC Virtual Lab Access Resources are the sole source and computational environment, the client-side and server-side hardware tools and systems are the same.

6.6.1 Hardware Tools:

- **Processor:** Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz 2.59 GHz
- **Installed RAM:** 32.0 GB

6.6.2 Software Tools:

- **Windows OS:** Windows 11 Pro Version 21H2 (OS build 22000.493), the 64-bit operating system
- **Diagramming and Design Tool:** ER Assistant version 2.10 (2002-2004), running on Windows
- **Office Productivity:** Microsoft Office 365, running on Windows
- **Database Development Tool:** Oracle SQL Developer Data Modeler Version 21.2.0.183.1957 (Build 183.1957), 2008-2021
- **The DBMS system:** Oracle Database 18c Express Edition Release 18.0.0.0.0 - Production Version 18.4.0.0.0, Oracle SQL Developer Version 21.2.1.204.1703 (Build 204.1703), 2005-2020
- **Connectivity Tools and Access Method:** Microsoft Edge (Version 106.0.1370.42, 64-bit, 2022) will be the web browser to connect the UMGC Virtual Lab Access Resources.

6.7 DDL and DML for Database Creation, SQL Usage and Style Guide

According to the designed Entity Relationship Diagram (ERD), Data Definition Language (DDL) is used to define the schema (physical structure) of a database, such as the database structure, data types in tables, defined primary and foreign keys, constraints, etc. Therefore, DDL defines how the data is stored in the database. In contrast, Data Manipulation Language (DML) is used to manipulate data in the database (i.e.,retrieve, update, and delete the data in a database). Both DDL and DML are two different categories

of Structured Query Language (SQL). SQL usage and style guide for this project are summarized as follows:

6.7.1 SQL Statement Structure for Readability:

- Initiate a new line for each keyword to split SQL statements into multiple line for ensuring readability.
- When multiple columns are selected in the query, each column name starts on its own line instead of placing all column names on one line.
- Place a comma after each column name except the last one.
- Multiple join conditions are placed on separate lines.
- The subquery statements are indented.
- Complete each SQL statement with a semicolon.

6.7.2 SQL script format:

- Reserved keywords (e.g., SELECT, FROM, WHERE, etc.) are uppercase.
- Whenever possible, use BETWEEN instead of using AND to connect multiple statements.
- Use IN ( ) instead of multiple OR clauses to concise the query statement.
- Use ANSI SQL equivalent keywords instead of database management system-specific SQL keywords
- Spaces are always included around the equal sign (=) and quotes (' ' or " ") where not within parentheses.
- Place a space after a comma.
- Whenever possible, use AS to introduce meaningful table aliases in the query statement.

6.7.3 Comment Usage:

- Comments should start with "/*" and end with "*/".
- When a comment is intended to be added on a line with a SQL statement, it starts with "--".

6.7.4 Object Naming Conventions:

- For table names, singular nouns are preferable to plural nouns.
- Whenever possible, consistent and descriptive names should be used in SQL statements.
- Names used in the SQL statement should be unique and different from reserved keywords.
- When a name is composed of 2 or more words, use underscores between words instead of camel case (e.g., First Name --> First_Name).
- The object/table name should not be surrounded by quotes or square brackets.
- Names must start with a letter and should be composed of letters, numbers, underscores, or their combinations.
- Avoid using prefixes to name objects/tables.
- Collective nouns are preferable for naming objects/tables.
- Column names should not be the same as table names.
- A joining/intersecting table is named according to two table names it represents.

- Whenever possible, naming a column name with suffixes that can indicate the meaning/purpose of the column (e.g., _name, _age, _no, etc.).
- Avoid using abbreviations to name tables and columns unless they are commonly known.
- Avoid naming the relationship of two tables by concatenating the names of two tables.

6.8 Perspectives in Database Technology and Management

This project is aimed to improve accuracy, safety, efficiency, and communication in transactions between pharmaceutical manufacturers, pharmacies, doctors, and patients. To achieve this goal, a database managed by the oracle relational database management system (RDBMS) needs to be the backend of a web-based pharmacy management system software implemented by HTML, CSS, Javascript, and PHP. HTML is the HyperText Markup Language that is used to construct a web page structure and its content. CSS (Cascading Style Sheet) is a style sheet language used to stylize and display elements written by a Markup language like HTML on the web page. JavaScript (JS) is a dynamic computer programming language used to create the dynamic and interactive web contents. Hypertext Preprocessor (PHP) is a web scripting language that handles the communication and interaction between the web application and the database. Through the PHP-mediated connection between the web application(frontend) and the database (backend), users can add data to the database and retrieve stored data in the database for analytic and reporting purposes.

In addition to technology for database implementation, maintaining data security by the database management is also important. Multiple strategies can be adopted for database security, such as:

- **Deploy physical database security:** Selecting a web server hosted by a company with a good reputation for online security and including physical security measures (e.g., installation of security cameras and locks, and security monitoring by personnel) are recommended approaches for database security.
- **Separate database servers:** Using more than one server for a database is a good practice for preventing database corruption and a data breach.
- **Establishing an HTTPS proxy server:** As the HTTPS proxy server encrypts data, it provides additional security protection to the database.
- **Avoid using default network ports:** Given that default network ports are usually vulnerable to cyber attacks, they are not secure and should not be used for the sake of database security.
- **Employ real-time database monitoring measures:** Software systems such as Tripwire's real-time File Integrity Monitoring (FIM) and real-time security information and event monitoring (SIEM) can be used to promote database security.
- **Database and web application firewalls:** Installation and activation of a firewall system can be an effective measure to protect the database against various malicious cyber attacks.
- **Deploy data encryption protocols:** Establishment and implementation of data encryption protocols can protect data in the database to minimize the risk of a data breach.
- **Routine backups of the database:** To minimize the risk of losing sensitive, important data in the database due to database corruption or cyber attacks, regular encrypted backups of the database in the separate servers are critical.

- **Regularly update applications:** Besides using trusted, verified database management system software, it should be routinely updated to maintain its security.
- **Strengthen user authentication:** Adoptingthe multi-factor authentication process can strengthen database security.

7. Requirements Definition Document

7.1 Entity and Attribute Description

Entity Name: **PATIENT**

Entity Description: A patient is a person who is receiving medical treatment from a doctor.

Main Attributes of **PATIENT**:

- *Attribute Name*: **Patient_ID (*Primary Key*)**
  Attribute Description: The unique identification number for a patient.
- *Attribute Name*: **Pat_Last_Name**
  Attribute Description: The last name of the patient.
- *Attribute Name*: **Pat_First_Name**
  Attribute Description: The first name of the patient.
- *Attribute Name*: **Patient_DOB**
  Attribute Description: The date of birth of the patient.
- *Attribute Name*: **Patient_SSN**
  Attribute Description: The social security number of the patient.
- *Attribute Name*: **Patient_Gender**
  Attribute Description: The gender of the patient.
- *Attribute Name*: **Patient_Address**
  Attribute Description: The address of the patient.
- *Attribute Name*: **Patient_TEL**
  Attribute Description: The contact phone number for the patient.
- *Attribute Name*: **Patient_Email**
  Attribute Description: The contact email for the patient.

Entity Name: **DOCTOR**

Entity Description: A doctor is a medical professional who treats patients and writes prescriptions for patients.

Main Attributes of **DOCTOR**:

- *Attribute Name*: **Doctor_ID (*Primary Key*)**
  *Attribute Description***:** The unique identification number for the doctor.

- *Attribute Name*: **Dr_Last_Name**
  Attribute Description: The last name of the doctor.
- *Attribute Name*: **Dr_First_Name**
  Attribute Description: The first name of the doctor.
- *Attribute Name*: **Dr_Specialty**
  Attribute Description: A description for the professional specialty of the doctor.
- *Attribute Name*: **Dr_Address**
  Attribute Description: The address of the doctor.
- *Attribute Name*: **Doctor_TEL**
  Attribute Description: The contact phone number for the doctor.
- *Attribute Name*: **Doctor_Email**
  Attribute Description: The contact email for the doctor.

Entity Name: **PRESCRIPTION**

Entity Description: A prescription is written by a doctor for a patient.

Main Attributes of **PRESCRIPTION**:

- *Attribute Name*: **Presc_ID (*Primary Key*)**
  Attribute Description: The unique identification number for a prescription.
- *Attribute Name*: **Patient_ID (*Foreign Key*)**
  Attribute Description: It is derived from the primary key of the parent entity **PATIENT**.
- *Attribute Name*: **Doctor_ID (*Foreign Key*)**
  Attribute Description: It is derived from the primary key of the parent entity **DOCTOR**.
- *Attribute Name*: **Presc_Date**
  Attribute Description: The issued date for the prescription.
- *Attribute Name*: **Presc_Use**
  Attribute Description: The doctor's instruction for a patient to take the drug.
- *Attribute Name*: **Items_No**
  *Attribute Description***:** The number of prescribed drugs for the prescription.
- *Attribute Name*: **Pay_Method**
  Attribute Description: The method (e.g., cash, debt card, credit card, etc.) to pay the purchases of prescribed drugs.
- *Attribute Name*: **Card_No**
  Attribute Description: The card number of the debt or credit card.
- *Attribute Name*: **Presc_Note**
  Attribute Description: The additional information for the prescription.

Entity Name: **DRUG**

Entity Description: A drug is the medicine product manufactured by a pharmaceutical company.

Main Attributes of **DRUG**:

- *Attribute Name*: **Drug_Code (*Primary Key*)**
  Attribute Description: The unique identifier code for the drug.
- *Attribute Name*: **Drug_Desc**
  Attribute Description: The description for the drug.
- *Attribute Name*: **Drug_Trademark**
  Attribute Description: The trademark of the drug.
- *Attribute Name*: **Drug_Dose**
  Attribute Description: A description for the drug dose that can be taken by patients.
- *Attribute Name*: **Drug_Formula**
  Attribute Description: A description for the drug formula in a tablet, capsule, or syrup form.
- *Attribute Name*: **Ingredients**
  Attribute Description: The ingredients of the drug.
- *Attribute Name*: **Drug_Uses**
  Attribute Description: A description for the uses of the drug to treat what symptoms and diseases.
- *Attribute Name*: **Drug_Cost**
  Attribute Description: The unit cost for the drug.
- *Attribute Name*: **Manufacturer**
  Attribute Description: The name of the pharmaceutical company to manufacture the drug.

Entity Name: **PHARMACY**

Entity Description: A pharmacy is a store to sell drugs.

Main Attributes of **PHARMACY**:

- *Attribute Name*: **Pharm_ID (*Primary Key*)**
  Attribute Description: The unique identification number for a pharmacy store.
- *Attribute Name*: **Pharm_Name**
  Attribute Description: The store name of the pharmacy.
- *Attribute Name*: **Pharm_City**
  Attribute Description: The city of the pharmacy location.
- *Attribute Name*: **Pharm_State**
  Attribute Description: The state of the pharmacy location.
- *Attribute Name*: **Pharm_Zip**
  Attribute Description: The zip code of the pharmacy location.
- *Attribute Name*: **Pharm_TEL**
  Attribute Description: The contact phone number for the pharmacy.
- *Attribute Name*: **Pharm_FAX**
  Attribute Description: The FAX number for the pharmacy.
- *Attribute Name*: **Pharm_Email**

Attribute Description: The contact Email for the pharmacy.

Entity Name: **PRESC_DRUG**

Entity Description: The itemized drug prescribed on the prescription.

Main Attributes of **PRESC_DRUG**:

- *Attribute Name*: **Presc_Drug_ID (*Primary Key*)**
  Attribute Description: The unique identification number (a surrogate key) for the prescribed drug on the prescription.
- *Attribute Name*: **Presc_ID (*Foreign Key*)**
  Attribute Description: It is derived from the primary key of the parent entity **PRESCRIPTION**.
- *Attribute Name*: **Drug_Code (*Foreign Key*)**
  Attribute Description: It is derived from the primary key of the parent entity **DRUG**.
- *Attribute Name*: **Pharm_ID (*Foreign Key*)**
  Attribute Description: It is derived from the primary key of the parent entity **PHARMACY**.
- *Attribute Name*: **Quantity**
  Attribute Description: The quantity of the drug prescribed on the prescription.
- *Attribute Name*: **Total_Cost**
  Attribute Description: The cost for the quantity of the prescribed drug.
- *Attribute Name*: **Refill_No**
  Attribute Description: The number of refill times for the prescribed drug.
- *Attribute Name*: **Batch_No**
  Attribute Description: The batch number of the drug dispensed for the patient.
- *Attribute Name*: **Expiration_Date**
  Attribute Description: The expiration date for the drug dispensed for the patient.
- *Attribute Name*: **Dispense_Date**
  Attribute Description: The date to dispense the drug and to be sold to the patient.

7.2 Relationship and Cardinality Description

Writes:

- ***Relationship:*** "**Writes**" between DOCTOR and PRESCRIPTION.
- ***Cardinality:*** one-to-many (1:M) between DOCTOR and PRESCRIPTION.
- ***Business rule:*** A doctor can prescribe one or multiple drugs for a patient, but a prescription is written by one and only one doctor.

Has:

- ***Relationship:*** "**Has**" between PATIENT and PRESCRIPTION.

- ***Cardinality***: one-to-many (1:M) between PATIENT and PRESCRIPTION.
- ***Business rule:*** A patient can have one or multiple prescriptions, but a prescription is written for one and only one patient.

Contains:

- ***Relationship:*** "**Contains**" between PRESCRIPTION and PRESC_DRUG.
- ***Cardinality:*** one-to-many (1:M) between PRESCRIPTION and PRESC_DRUG.
- ***Business rule:*** A prescription includes one or multiple prescribed drugs. A prescribed drug is for one and only one prescription.

Sells:

- ***Relationship:*** "**Sells**" between PHARMACY and PRESC_DRUG.
- ***Cardinality:*** one-to-many (1:M) between PHARMACY and PRESC_DRUG.
- ***Business rule:*** A pharmacy sells one or multiple prescribed drugs, but may not sell every prescribed drug. Each prescribed drug is sold by one and only one pharmacy.

Exists:

- ***Relationship:*** "**Exists**" between DRUG and PRESC_DRUG.
- ***Cardinality:*** one-to-many (1:M) between DRUG and PRESC_DRUG.
- ***Business rule:*** A drug manufactured by a pharmaceutical company may be prescribed on zero, one or many prescriptions. A prescribed drug must be for one and only one drug manufactured by a pharmaceutical company.

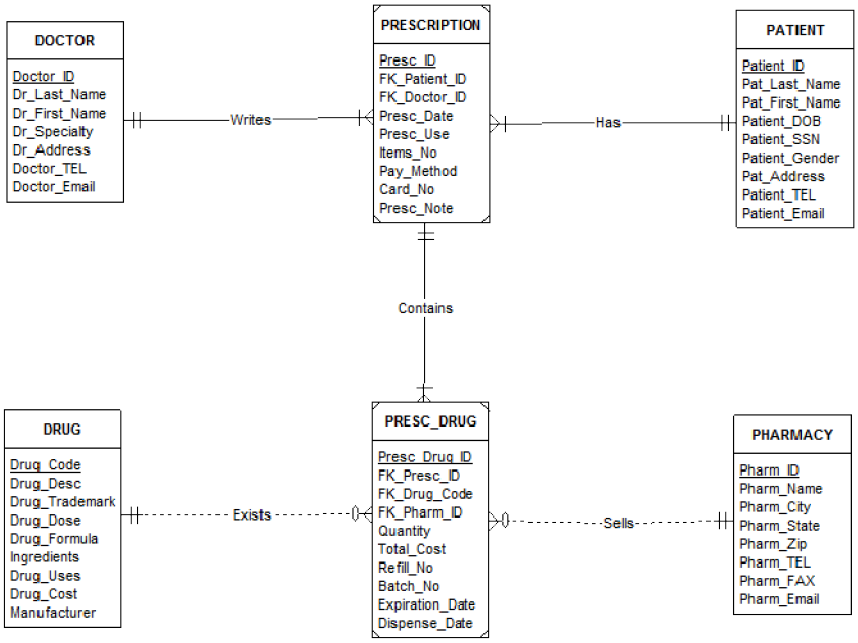
7.3 Assumptions and Special Considerations

The relationship between PRESCRIPTION and DRUG and the relationship between DRUG and PHARMACY are many-to-many (M:M). The entity PRESC_DRUG is the bridge/intersecting entity for resolving these two M:M relationships, which allows one-to-many (1:M) relationships with these three entities. PRESCRIPTION is a weak entity that cannot exist alone without two strong entities PATIENT and DOCTOR. PRESC_DRUG is also a weak entity that cannot exist alone and need to rely on PRESCRIPTION.

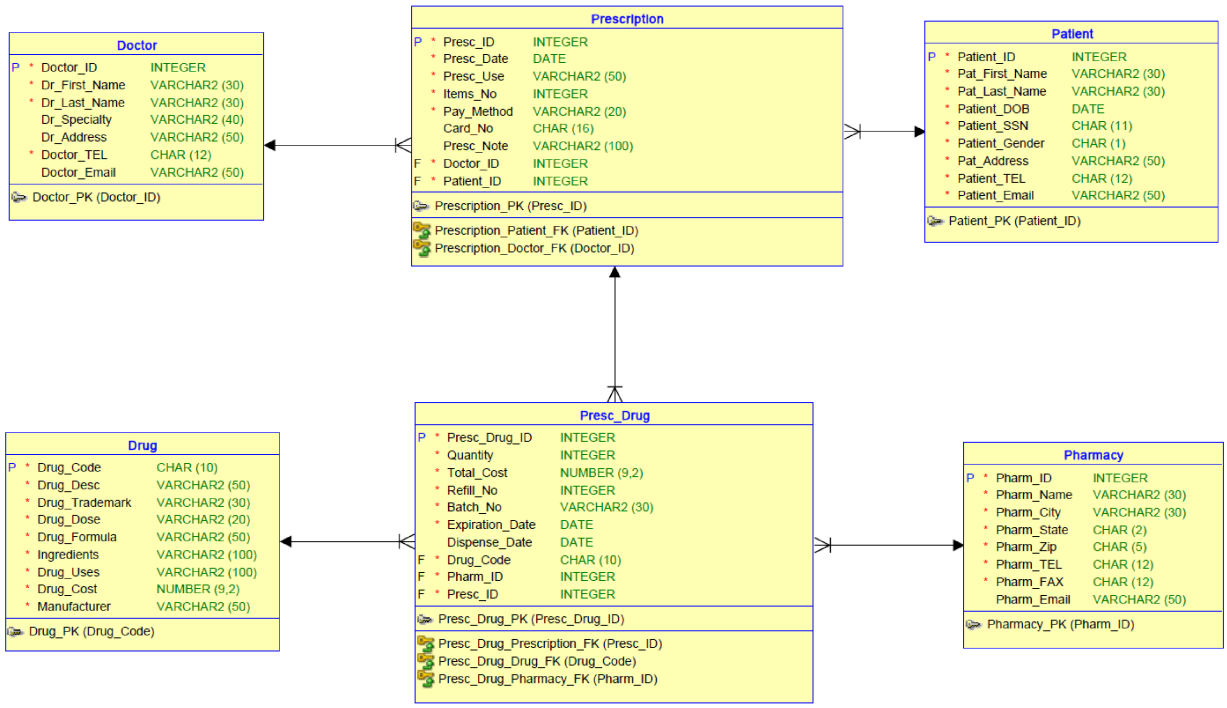
8. Detailed Database Design

8.1 Entity Relationship Diagram (ERD)

The following ERD was created by ER-Assistant. The foreign key is indicated by convention such as FK_parent_PK/col_name.



The relational data model was created by the Oracle SQL Developer Data Modeler based on the ERD.



8.2 DDL Source Code Embedded

Data Definition Language (DDL) SQL Script

```
--Drop Statements to clean up all objects from previous run

--Drop Triggers
DROP TRIGGER trg_presc_drug;
DROP TRIGGER trg_prescription;
DROP TRIGGER trg_doctor;
DROP TRIGGER trg_patient;
DROP TRIGGER trg_drug;
DROP TRIGGER trg_pharmacy;



--Drop Sequences
DROP SEQUENCE seq_doctor_id;
DROP SEQUENCE seq_patient_id;
DROP SEQUENCE seq_presc_id;
DROP SEQUENCE seq_pharm_id;
DROP SEQUENCE seq_presc_drug_id;



--Drop Views
DROP VIEW doctor_view;
DROP VIEW patient_view;
DROP VIEW drug_view;
DROP VIEW pharmacy_view;
DROP VIEW prescription_view;
DROP VIEW presc_drug_view;
DROP VIEW presc_pat_dr;
DROP VIEW cost_per_presc;



--Drop Indexes
DROP INDEX doctor_tel_idx;
DROP INDEX patient_ssn_idx;
DROP INDEX patient_tel_idx;
DROP INDEX pharm_tel_idx;

DROP INDEX presc_doctor_fk_idx;
DROP INDEX presc_patient_fk_idx;
DROP INDEX presc_drug_drug_fk_idx;
DROP INDEX presc_drug_pharmacy_fk_idx;
DROP INDEX presc_drug_prescription_fk_idx;

DROP INDEX dr_first_name_idx;
DROP INDEX dr_last_name_idx;
DROP INDEX pat_first_name_idx;
DROP INDEX pat_last_name_idx;
```

```
DROP INDEX drug_cost_idx;
DROP INDEX pharm_name_idx;
DROP INDEX pharm_city_idx;
DROP INDEX pharm_zip_idx;



--Drop Tables
DROP TABLE presc_drug CASCADE CONSTRAINTS;
DROP TABLE prescription CASCADE CONSTRAINTS;
DROP TABLE doctor CASCADE CONSTRAINTS;
DROP TABLE patient CASCADE CONSTRAINTS;
DROP TABLE drug CASCADE CONSTRAINTS;
DROP TABLE pharmacy CASCADE CONSTRAINTS;
```

--Create/Alter Statements for All Tables and Constraints

```
CREATE TABLE doctor (
    doctor_id     INTEGER NOT NULL,
    dr_first_name VARCHAR2(30) NOT NULL,
    dr_last_name  VARCHAR2(30) NOT NULL,
    dr_specialty  VARCHAR2(40),
    dr_address    VARCHAR2(50),
    doctor_tel    CHAR(12) NOT NULL,
    doctor_email  VARCHAR2(50)
);

ALTER TABLE doctor ADD CONSTRAINT doctor_pk PRIMARY KEY ( doctor_id );

CREATE TABLE drug (
    drug_code      CHAR(10) NOT NULL,
    drug_desc      VARCHAR2(50) NOT NULL,
    drug_trademark VARCHAR2(30) NOT NULL,
    drug_dose      VARCHAR2(20) NOT NULL,
    drug_formula   VARCHAR2(50) NOT NULL,
    ingredients    VARCHAR2(100) NOT NULL,
    drug_uses      VARCHAR2(100) NOT NULL,
    drug_cost      NUMBER(9, 2) NOT NULL,
    manufacturer   VARCHAR2(50) NOT NULL
);

ALTER TABLE drug ADD CONSTRAINT drug_pk PRIMARY KEY ( drug_code );

CREATE TABLE patient (
    patient_id     INTEGER NOT NULL,
    pat_first_name VARCHAR2(30) NOT NULL,
    pat_last_name  VARCHAR2(30) NOT NULL,
    patient_dob    DATE NOT NULL,
    patient_ssn    CHAR(11) NOT NULL,
    patient_gender CHAR(1) NOT NULL,
    pat_address    VARCHAR2(50) NOT NULL,
```

```
    patient_tel    CHAR(12) NOT NULL,
    patient_email  VARCHAR2(50) NOT NULL
);

ALTER TABLE patient ADD CONSTRAINT patient_pk PRIMARY KEY ( patient_id );

CREATE TABLE pharmacy (
    pharm_id    INTEGER NOT NULL,
    pharm_name  VARCHAR2(30) NOT NULL,
    pharm_city  VARCHAR2(30) NOT NULL,
    pharm_state CHAR(2) NOT NULL,
    pharm_zip   CHAR(5) NOT NULL,
    pharm_tel   CHAR(12) NOT NULL,
    pharm_fax   CHAR(12) NOT NULL,
    pharm_email VARCHAR2(50)
);

ALTER TABLE pharmacy ADD CONSTRAINT pharmacy_pk PRIMARY KEY ( pharm_id );


CREATE TABLE prescription (
    presc_id   INTEGER NOT NULL,
    presc_date DATE NOT NULL,
    presc_use  VARCHAR2(50) NOT NULL,
    items_no   INTEGER NOT NULL,
    pay_method VARCHAR2(20) NOT NULL,
    card_no    CHAR(16) NOT NULL,
    presc_note VARCHAR2(100),
    doctor_id  INTEGER NOT NULL,
    patient_id INTEGER NOT NULL
);

ALTER TABLE prescription ADD CONSTRAINT prescription_pk PRIMARY KEY
(presc_id);


CREATE TABLE presc_drug (
    presc_drug_id   INTEGER NOT NULL,
    quantity        INTEGER NOT NULL,
    total_cost      NUMBER(9, 2) NOT NULL,
    refill_no       INTEGER NOT NULL,
    batch_no        VARCHAR2(30) NOT NULL,
    expiration_date DATE NOT NULL,
    dispense_date   DATE,
    drug_code       CHAR(10) NOT NULL,
    pharm_id        INTEGER NOT NULL,
    presc_id        INTEGER NOT NULL
);

ALTER TABLE presc_drug ADD CONSTRAINT presc_drug_pk PRIMARY KEY
(presc_drug_id);
```

```
ALTER TABLE prescription
    ADD CONSTRAINT prescription_doctor_fk FOREIGN KEY ( doctor_id )
        REFERENCES doctor ( doctor_id );

ALTER TABLE prescription
    ADD CONSTRAINT prescription_patient_fk FOREIGN KEY ( patient_id )
        REFERENCES patient ( patient_id );

ALTER TABLE presc_drug
    ADD CONSTRAINT presc_drug_drug_fk FOREIGN KEY ( drug_code )
        REFERENCES drug ( drug_code );

ALTER TABLE presc_drug
    ADD CONSTRAINT presc_drug_pharmacy_fk FOREIGN KEY ( pharm_id )
        REFERENCES pharmacy ( pharm_id );

ALTER TABLE presc_drug
    ADD CONSTRAINT presc_drug_prescription_fk FOREIGN KEY ( presc_id )
        REFERENCES prescription ( presc_id );
```

--Create Indexes for Natural, Foreign Key, and Frequently Queried Columns

```
--Create unique index on natural key columns
CREATE UNIQUE INDEX doctor_tel_idx ON doctor ( doctor_tel );
CREATE UNIQUE INDEX patient_ssn_idx ON patient ( patient_ssn );
CREATE UNIQUE INDEX patient_tel_idx ON patient ( patient_tel );
CREATE UNIQUE INDEX pharm_tel_idx ON pharmacy ( pharm_tel );


--Create index on foreign key columns

CREATE INDEX presc_doctor_fk_idx ON prescription ( doctor_id );
CREATE INDEX presc_patient_fk_idx ON prescription ( patient_id );
CREATE INDEX presc_drug_drug_fk_idx ON presc_drug ( drug_code );
CREATE INDEX presc_drug_pharmacy_fk_idx ON presc_drug ( pharm_id );
CREATE INDEX presc_drug_prescription_fk_idx ON presc_drug ( presc_id );


--Create index on frequently queried columns

CREATE INDEX dr_first_name_idx ON doctor ( dr_first_name );
CREATE INDEX dr_last_name_idx ON doctor ( dr_last_name );
CREATE INDEX pat_first_name_idx ON patient ( pat_first_name );
CREATE INDEX pat_last_name_idx ON patient ( pat_last_name );
CREATE INDEX drug_cost_idx ON drug ( drug_cost );
CREATE INDEX pharm_name_idx ON pharmacy ( pharm_name );
CREATE INDEX pharm_city_idx ON pharmacy ( pharm_city );
CREATE INDEX pharm_zip_idx ON pharmacy ( pharm_zip );
```

```
--Alter Tables by adding Audit Columns

ALTER TABLE doctor ADD (
    created_by      VARCHAR2(30) NOT NULL,
    date_created  DATE NOT NULL,
    modified_by     VARCHAR2(30) NOT NULL,
    date_modified DATE NOT NULL
);


ALTER TABLE patient ADD (
    created_by      VARCHAR2(30) NOT NULL,
    date_created  DATE NOT NULL,
    modified_by     VARCHAR2(30) NOT NULL,
    date_modified DATE NOT NULL
);


ALTER TABLE drug ADD (
    created_by      VARCHAR2(30) NOT NULL,
    date_created  DATE NOT NULL,
    modified_by     VARCHAR2(30) NOT NULL,
    date_modified DATE NOT NULL
);


ALTER TABLE pharmacy ADD (
    created_by      VARCHAR2(30) NOT NULL,
    date_created  DATE NOT NULL,
    modified_by     VARCHAR2(30) NOT NULL,
    date_modified DATE NOT NULL
);


ALTER TABLE prescription ADD (
    created_by      VARCHAR2(30) NOT NULL,
    date_created  DATE NOT NULL,
    modified_by     VARCHAR2(30) NOT NULL,
    date_modified DATE NOT NULL
);


ALTER TABLE presc_drug ADD (
    created_by      VARCHAR2(30) NOT NULL,
    date_created  DATE NOT NULL,
    modified_by     VARCHAR2(30) NOT NULL,
    date_modified DATE NOT NULL
);
```

```
--Create Views
/*Only business columns, but not audit columns, are included in views*/


--The view for Doctor
/*Business purpose: The Doctor view will be used primarily for rapidly
fetching information about individual doctors who write prescriptions
for patients.*/

CREATE OR REPLACE VIEW doctor_view AS
SELECT doctor_id, dr_first_name, dr_last_name, doctor_tel
FROM doctor;


--The view for Patient
/*Business purpose: The Patient view will be used primarily for rapidly
fetching information about individual patients who has prescriptions
written from doctors.*/

CREATE OR REPLACE VIEW patient_view AS
SELECT patient_id, pat_first_name, pat_last_name, patient_dob, patient_ssn,
       patient_tel, patient_email
FROM patient;


--The view for Drug
/*Business purpose: The Drug view will be used primarily for rapidly
fetching information about drugs that can be prescribed by doctors.*/

CREATE OR REPLACE VIEW drug_view AS
SELECT drug_code, drug_desc, drug_trademark, drug_dose, drug_formula,
       ingredients, drug_uses, drug_cost
FROM drug;


--The view for Pharmacy
/*Business purpose: The Pharmacy view will be used primarily for rapidly
fetching information about pharmacies that sell prescribed drugs.*/

CREATE OR REPLACE VIEW pharmacy_view AS
SELECT pharm_id, pharm_name, pharm_city, pharm_state, pharm_zip,
       pharm_tel, pharm_email
FROM pharmacy;


--The view for Prescription
/*Business purpose: The Prescription view will be used primarily for rapidly
fetching information about prescriptions written by doctors for patients.*/

CREATE OR REPLACE VIEW prescription_view AS
SELECT presc_id, presc_date, presc_use, items_no, pay_method, card_no,
       doctor_id, patient_id
FROM prescription;
```

```
--The view for Presc_drug
/*Business purpose: The Presc_drug view will be used primarily for rapidly
fetching information about prescribed drugs shown on prescriptions.*/

CREATE OR REPLACE VIEW presc_drug_view AS
SELECT presc_drug_id, quantity, total_cost, refill_no, batch_no,
       expiration_date, dispense_date, drug_code, pharm_id, presc_id
FROM presc_drug;


--The view for Presc_Pat_Dr
/*Business purpose: The presc_pat_dr view will be used primarily for rapidly
fetching information about prescriptions that belong to which patients and
are written by which doctors.*/

CREATE OR REPLACE VIEW presc_pat_dr AS
SELECT presc_id, presc_date, presc_use, pat_first_name, pat_last_name,
       dr_first_name, dr_last_name
FROM prescription pr JOIN patient pa ON pr.patient_id = pa.patient_id JOIN
     doctor d ON pr.doctor_id = d.doctor_id;


--The view for cost_per_presc
/*Business purpose: The cost_per_presc view will be used primarily for
rapidly fetching information about the total cost of prescribed drugs from
a prescription and which patient is responsible for this prescription
cost.*/

CREATE OR REPLACE VIEW cost_per_presc AS
SELECT pr.presc_id, SUM(total_cost) AS total_cost, pat_first_name,
       pat_last_name, pay_method, card_no
FROM presc_drug pd
LEFT OUTER JOIN prescription pr ON pd.presc_id = pr.presc_id
LEFT OUTER JOIN patient pa ON pr.patient_id = pa.patient_id
Group BY pat_first_name, pat_last_name, pr.presc_id, pay_method, card_no
ORDER BY pr.presc_id;
```

--Create Sequences

```
CREATE SEQUENCE seq_doctor_id
    MINVALUE 1
    START WITH 1
    INCREMENT BY 1
    CACHE 10;


CREATE SEQUENCE seq_patient_id
    MINVALUE 1
    START WITH 1
```

```
    INCREMENT BY 1
    CACHE 10;


CREATE SEQUENCE seq_presc_id
    MINVALUE 1
    START WITH 1
    INCREMENT BY 1
    CACHE 10;


CREATE SEQUENCE seq_pharm_id
    MINVALUE 1
    START WITH 1
    INCREMENT BY 1
    CACHE 10;


CREATE SEQUENCE seq_presc_drug_id
    MINVALUE 1
    START WITH 1
    INCREMENT BY 1
    CACHE 10;
```

--Create Triggers

```
--The Trigger for Doctor
/*Business Purpose: This trigger populates surrogate key and audit columns
with appropriate values into the Doctor table.*/

CREATE OR REPLACE TRIGGER trg_doctor
  BEFORE INSERT OR UPDATE ON doctor
  FOR EACH ROW
BEGIN
  -- use the sequence object to generate the surrogate key
  IF :NEW.doctor_id IS NULL THEN
    :NEW.doctor_id := seq_doctor_id.NEXTVAL;
  END IF;

  -- automatically generate appropriate values for audit columns
  IF INSERTING THEN
    IF :NEW.created_by IS NULL THEN :NEW.created_by := USER; END IF;
    IF :NEW.date_created IS NULL THEN :NEW.date_created := SYSDATE; END IF;
  END IF;
  IF INSERTING OR UPDATING THEN
    IF :NEW.modified_by IS NULL THEN :NEW.modified_by := USER; END IF;
    IF :NEW.date_modified IS NULL THEN :NEW.date_modified := SYSDATE; END IF;
  END IF;
END;
```

```
/


--The Trigger for Patient
/*Business Purpose: This trigger populates surrogate key and audit columns
with appropriate values into the Patient table.*/

CREATE OR REPLACE TRIGGER trg_patient
  BEFORE INSERT OR UPDATE ON patient
  FOR EACH ROW
BEGIN
  -- use the sequence object to generate the surrogate key
  IF :NEW.patient_id IS NULL THEN
    :NEW.patient_id := seq_patient_id.NEXTVAL;
  END IF;

  -- automatically generate appropriate values for audit columns
  IF INSERTING THEN
    IF :NEW.created_by IS NULL THEN :NEW.created_by := USER; END IF;
    IF :NEW.date_created IS NULL THEN :NEW.date_created := SYSDATE; END IF;
  END IF;
  IF INSERTING OR UPDATING THEN
    IF :NEW.modified_by IS NULL THEN :NEW.modified_by := USER; END IF;
    IF :NEW.date_modified IS NULL THEN :NEW.date_modified := SYSDATE; END IF;
  END IF;
END;
/


--The Trigger for Prescription
/*Business Purpose: This trigger populates surrogate key and audit columns
with appropriate values into the Prescription table.*/

CREATE OR REPLACE TRIGGER trg_prescription
  BEFORE INSERT OR UPDATE ON prescription
  FOR EACH ROW
BEGIN
  -- use the sequence object to generate the surrogate key
  IF :NEW.presc_id IS NULL THEN
    :NEW.presc_id := seq_presc_id.NEXTVAL;
  END IF;

  -- automatically generate appropriate values for audit columns
  IF INSERTING THEN
    IF :NEW.created_by IS NULL THEN :NEW.created_by := USER; END IF;
    IF :NEW.date_created IS NULL THEN :NEW.date_created := SYSDATE; END IF;
  END IF;
  IF INSERTING OR UPDATING THEN
    IF :NEW.modified_by IS NULL THEN :NEW.modified_by := USER; END IF;
    IF :NEW.date_modified IS NULL THEN :NEW.date_modified := SYSDATE; END IF;
  END IF;
END;
/
```

```
--The Trigger for Drug
/*Business Purpose: This trigger populates audit columns with appropriate
values into the Drug table.*/

CREATE OR REPLACE TRIGGER trg_drug
  BEFORE INSERT OR UPDATE ON drug
  FOR EACH ROW
BEGIN
  -- automatically generate appropriate values for audit columns
  IF INSERTING THEN
    IF :NEW.created_by IS NULL THEN :NEW.created_by := USER; END IF;
    IF :NEW.date_created IS NULL THEN :NEW.date_created := SYSDATE; END IF;
  END IF;
  IF INSERTING OR UPDATING THEN
    IF :NEW.modified_by IS NULL THEN :NEW.modified_by := USER; END IF;
    IF :NEW.date_modified IS NULL THEN :NEW.date_modified := SYSDATE; END IF;
  END IF;
END;
/


--The Trigger for Pharmacy
/*Business Purpose: This trigger populates surrogate key and audit columns
with appropriate values into the Pharmacy table.*/

CREATE OR REPLACE TRIGGER trg_pharmacy
  BEFORE INSERT OR UPDATE ON pharmacy
  FOR EACH ROW
BEGIN
  -- use the sequence object to generate the surrogate key
  IF :NEW.pharm_id IS NULL THEN
    :NEW.pharm_id := seq_pharm_id.NEXTVAL;
  END IF;

  -- automatically generate appropriate values for audit columns
  IF INSERTING THEN
    IF :NEW.created_by IS NULL THEN :NEW.created_by := USER; END IF;
    IF :NEW.date_created IS NULL THEN :NEW.date_created := SYSDATE; END IF;
  END IF;
  IF INSERTING OR UPDATING THEN
    IF :NEW.modified_by IS NULL THEN :NEW.modified_by := USER; END IF;
    IF :NEW.date_modified IS NULL THEN :NEW.date_modified := SYSDATE; END IF;
  END IF;
END;
/


--The Trigger for Presc_Drug
/*Business Purpose: This trigger populates surrogate key and audit columns
with appropriate values into the Presc_Drug table.*/

CREATE OR REPLACE TRIGGER trg_presc_drug
  BEFORE INSERT OR UPDATE ON presc_drug
```

```
  FOR EACH ROW
BEGIN
  -- use the sequence object to generate the surrogate key
  IF :NEW.presc_drug_id IS NULL THEN
    :NEW.presc_drug_id := seq_presc_drug_id.NEXTVAL;
  END IF;

  -- automatically generate appropriate values for audit columns
  IF INSERTING THEN
    IF :NEW.created_by IS NULL THEN :NEW.created_by := USER; END IF;
    IF :NEW.date_created IS NULL THEN :NEW.date_created := SYSDATE; END IF;
  END IF;
  IF INSERTING OR UPDATING THEN
    IF :NEW.modified_by IS NULL THEN :NEW.modified_by := USER; END IF;
    IF :NEW.date_modified IS NULL THEN :NEW.date_modified := SYSDATE; END IF;
  END IF;
END;
/
```

--Database Catalog/Data Dictionary Queries

```
SELECT TABLE_NAME
FROM USER_TABLES;


SELECT OBJECT_NAME, STATUS, CREATED, LAST_DDL_TIME
FROM USER_OBJECTS;
```

8.3 DML and Query Source Code Embedded

Data Manipulation Language (DML) and 20 Queries:

```
--Add data to each table using DML

--Insert 10 rows of data into the table DOCTOR
INSERT INTO doctor (dr_first_name, dr_last_name, Dr_specialty,
            dr_address, doctor_tel, doctor_email)
VALUES ('Fernand', 'Hanks', 'Internal Medicine',
        '254 Bleeker, New York, NY 10005',
        '212-545-1216', 'fhands@gmail.com');
INSERT INTO doctor (dr_first_name, dr_last_name, Dr_specialty,
            dr_address, doctor_tel, doctor_email)
VALUES ('Marilyn', 'Frantzen', 'Pediatrics',
        '100 East 87th, New York, NY 10015',
        '212-569-3762', 'mfrantzen@gmail.com');
INSERT INTO doctor (dr_first_name, dr_last_name, Dr_specialty,
            dr_address, doctor_tel, doctor_email)
VALUES ('Anita', 'Morris', 'Orthopedics',
        '34 Maiden Lane, New York, NY 10015',
        '212-388-3476', 'amorris@gmail.com');
INSERT INTO doctor (dr_first_name, dr_last_name, Dr_specialty,
            dr_address, doctor_tel, doctor_email)
VALUES ('Rick', 'Chow', 'Family Medicine',
        '56 10th Avenue, New York, NY 10015',
        '212-485-1762', 'rchow@gmail.com');
INSERT INTO doctor (dr_first_name, dr_last_name, Dr_specialty,
            dr_address, doctor_tel, doctor_email)
VALUES ('Tom', 'Wojick', 'Neurology',
        '518 West 120th, New York, NY 10025',
        '212-684-9832', 'twojick@gmail.com');
INSERT INTO doctor (dr_first_name, dr_last_name, Dr_specialty,
            dr_address, doctor_tel, doctor_email)
VALUES ('Nina', 'Schorin', 'Urology',
        '210 West 101st, New York, NY 10025',
        '212-774-2685', 'nschorin@gmail.com');
INSERT INTO doctor (dr_first_name, dr_last_name, Dr_specialty,
            dr_address, doctor_tel, doctor_email)
VALUES ('Todd', 'Smythe', 'Dermatology',
        '210 West 103st, New York, NY 10025',
        '212-775-9837', 'tsmythe@gmail.com');
INSERT INTO doctor (dr_first_name, dr_last_name, Dr_specialty,
            dr_address, doctor_tel, doctor_email)
VALUES ('Charles', 'Lowry', 'Ophthalmology',
        '518 West 120th, New York, NY 10025',
        '212-566-3644', 'clowry@gmail.com');
INSERT INTO doctor (dr_first_name, dr_last_name, Dr_specialty,
            dr_address, doctor_tel, doctor_email)
VALUES ('Gary', 'Pertez', 'Cardiology',
        '34 Sixth Ave, New York, NY 10035',
        '212-787-7543', 'gpertez@gmail.com');
INSERT INTO doctor (dr_first_name, dr_last_name, Dr_specialty,
```

```
            dr_address, doctor_tel, doctor_email)
VALUES ('Irene', 'Willig', 'Cardiology',
        '415 West 101st, New York, NY 10045',
        '212-845-2548', 'iwillig@gmail.com');


--Insert 10 rows of data into the table PATIENT
INSERT INTO patient (pat_first_name, pat_last_name, patient_dob,
            patient_ssn, patient_gender, pat_address, patient_tel,
            patient_email)
VALUES ('Fred', 'Crocitto', TO_DATE('02-JAN-2000','DD-MON-YYYY'),
        '218-51-3692', 'M', '101-09 120th St., Richmond Hill, NY 11419',
        '718-667-5692', 'fcrocitto@gmail.com');
INSERT INTO patient (pat_first_name, pat_last_name, patient_dob,
            patient_ssn, patient_gender, pat_address, patient_tel,
            patient_email)
VALUES ('Laetia', 'Enison', TO_DATE('16-MAR-1998','DD-MON-YYYY'),
        '217-61-9372', 'F', '144-61 87th Ave, Jamaica, NY 11435',
        '201-632-3857', 'lenison@gmail.com');
INSERT INTO patient (pat_first_name, pat_last_name, patient_dob,
            patient_ssn, patient_gender, pat_address, patient_tel,
            patient_email)
VALUES ('Angel', 'Moskowitz', TO_DATE('22-OCT-1994','DD-MON-YYYY'),
        '213-41-9281', 'F', '320 John St., Ft. Lee, NY 07024',
        '201-293-2846', 'amoskowitz@gmail.com');
INSERT INTO patient (pat_first_name, pat_last_name, patient_dob,
            patient_ssn, patient_gender, pat_address, patient_tel,
            patient_email)
VALUES ('Judith', 'Olvsade', TO_DATE('08-JUN-1992','DD-MON-YYYY'),
        '214-61-2847', 'F', '29 Elmwood Ave., Montclair, NY 07042',
        '214-826-1037', 'jolvsade@gmail.com');
INSERT INTO patient (pat_first_name, pat_last_name, patient_dob,
            patient_ssn, patient_gender, pat_address, patient_tel,
            patient_email)
VALUES ('Catherine', 'Mierzwa', TO_DATE('18-NOV-1996','DD-MON-YYYY'),
        '215-71-4923', 'F', '22-70 41st St., Astoria, NY 11105',
        '215-387-9027', 'cmierzwa@gmail.com');
INSERT INTO patient (pat_first_name, pat_last_name, patient_dob,
            patient_ssn, patient_gender, pat_address, patient_tel,
            patient_email)
VALUES ('Judy', 'Sethi', TO_DATE('12-APR-1999','DD-MON-YYYY'),
        '217-81-8237', 'F', '38 Bay 26th St., Brooklyn, NY 11214',
        '716-682-8263', 'jsethi@gmail.com');
INSERT INTO patient (pat_first_name, pat_last_name, patient_dob,
            patient_ssn, patient_gender, pat_address, patient_tel,
            patient_email)
VALUES ('Winsome', 'Laporte', TO_DATE('21-MAY-1995','DD-MON-YYYY'),
        '213-42-3648', 'M', '268 E. 3rd St, Brooklyn, NY 11226',
        '718-533-2947', 'wlaporte@gmail.com');
INSERT INTO patient (pat_first_name, pat_last_name, patient_dob,
            patient_ssn, patient_gender, pat_address, patient_tel,
            patient_email)
VALUES ('Sean', 'Pineda', TO_DATE('27-AUG-1991','DD-MON-YYYY'),
        '212-31-5493', 'M', '3 Salem Rd., New City, NY 10956',
```

```
        '212-573-2074', 'spineda@gmail.com');
INSERT INTO patient (pat_first_name, pat_last_name, patient_dob,
            patient_ssn, patient_gender, pat_address, patient_tel,
            patient_email)
VALUES ('Angela', 'Torres', TO_DATE('05-DEC-1992','DD-MON-YYYY'),
        '212-31-1038', 'F', '509 2nd St #4L, Brooklyn, NY 11215',
        '718-394-3877', 'atorres@gmail.com');
INSERT INTO patient (pat_first_name, pat_last_name, patient_dob,
            patient_ssn, patient_gender, pat_address, patient_tel,
            patient_email)
VALUES ('Monica', 'Waldman', TO_DATE('18-FEB-1998','DD-MON-YYYY'),
        '216-71-4037', 'F', '257 Depot Rd., Huntington, NY 11766',
        '718-875-9048', 'mwaldman@gmail.com');


--Insert 10 rows of data into the table DRUG
INSERT INTO drug (drug_code, drug_desc, drug_trademark, drug_dose,
            drug_formula, ingredients, drug_uses, drug_cost,
            manufacturer)
VALUES ('I10837332', 'Ibuprofen', 'Advil', 'three times a day',
        'C13H18O2', 'Ibuprofen 200 mg',
        'treat mild to moderate pain and arthritis',
        1.15, 'Pfizer');
INSERT INTO drug (drug_code, drug_desc, drug_trademark, drug_dose,
            drug_formula, ingredients, drug_uses, drug_cost,
            manufacturer)
VALUES ('O23547364', 'Olmesartan', 'Benicar', 'once a day',
        'C24H26N6O3', 'Olmesartan 20 mg',
        'treat high blood pressure (hypertension)',
        8.09, 'Lupin Limited');
INSERT INTO drug (drug_code, drug_desc, drug_trademark, drug_dose,
            drug_formula, ingredients, drug_uses, drug_cost,
            manufacturer)
VALUES ('D20847476', 'Dicyclomine', 'Bentyl', 'four times a day',
        'C19H35NO2', 'Dicyclomine 20 mg',
        'treat functional bowel or irritable bowel syndrome',
        0.87, 'Aptalis Pharma Canada');
INSERT INTO drug (drug_code, drug_desc, drug_trademark, drug_dose,
            drug_formula, ingredients, drug_uses, drug_cost,
            manufacturer)
VALUES ('C76488378', 'Cetirizine', 'Zyrtec', 'once a day',
        'C21H25ClN2O3', 'Cetirizine 10 mg',
        'treat cold or allergy symptoms',
        0.83, 'Johnson and Johnson');
INSERT INTO drug (drug_code, drug_desc, drug_trademark, drug_dose,
            drug_formula, ingredients, drug_uses, drug_cost,
            manufacturer)
VALUES ('D34843734', 'Dexamethasone', 'Decadron', 'once a day',
        'C22H29FO5', 'Dexamethasone 10 mg',
        'treat many different inflammatory conditions',
        3.29, 'Fera Pharmaceuticals');
INSERT INTO drug(drug_code, drug_desc, drug_trademark, drug_dose,
            drug_formula, ingredients, drug_uses, drug_cost,
            manufacturer)
```

```
VALUES ('H47637494', 'Hydrochlorothiazide', 'Esidrix', 'once a day',
        'C7H8ClN3O4S2', 'Hydrochlorothiazide 50 mg',
        'treat high blood pressure (hypertension)',
        2.49, 'Merck');
INSERT INTO drug(drug_code, drug_desc, drug_trademark, drug_dose,
            drug_formula, ingredients, drug_uses, drug_cost,
            manufacturer)
VALUES ('F68403835', 'Famotidine', 'Pepcid', 'twice a day',
        'C8H15N7O2S3', 'Famotidine 20 mg',
        'treat and prevent ulcers in the stomach and intestines',
        1.27, 'Johnson and Johnson');
INSERT INTO drug(drug_code, drug_desc, drug_trademark, drug_dose,
            drug_formula, ingredients, drug_uses, drug_cost,
            manufacturer)
VALUES ('C53274803', 'Clopidogrel', 'Plavix', 'once a day',
        'C16H16ClNO2S.H2SO4', 'Clopidogrel 75 mg',
        'treat stroke, blood clot, or serious heart problem',
        3.59, 'Bristol-Myers Squibb');
INSERT INTO drug(drug_code, drug_desc, drug_trademark, drug_dose,
            drug_formula, ingredients, drug_uses, drug_cost,
            manufacturer)
VALUES ('P39857223', 'Pseudoephedrine', 'Sudafed', 'twice a day',
        'C10H15NO', 'Pseudoephedrine 120 mg',
        'treat nasal and sinus congestion',
        1.31, 'Johnson and Johnson');
INSERT INTO drug(drug_code, drug_desc, drug_trademark, drug_dose,
            drug_formula, ingredients, drug_uses, drug_cost,
            manufacturer)
VALUES ('R87462748', 'Ranitidine', 'Zantac', 'once a day',
        'C13H22N4O3S', 'Ranitidine 150 mg',
        'treat and prevent ulcers in the stomach and intestines',
        0.87, 'GlaxoSmithKline');


--Insert 10 rows of data into the table PHARMACY
INSERT INTO pharmacy (pharm_name, pharm_city, pharm_state, pharm_zip,
            pharm_tel, pharm_fax, pharm_email)
VALUES ('Walgreens', 'Huntington', 'NY', '11766', '718-485-2984',
        '718-485-2992', 'customer@walgreens.com');
INSERT INTO pharmacy (pharm_name, pharm_city, pharm_state, pharm_zip,
            pharm_tel, pharm_fax, pharm_email)
VALUES ('CVS Health', 'Brooklyn', 'NY', '11232', '917-376-5849',
        '917-376-5852', 'customer@caremark.com');
INSERT INTO pharmacy (pharm_name, pharm_city, pharm_state, pharm_zip,
            pharm_tel, pharm_fax, pharm_email)
VALUES ('Walmart', 'New York', 'NY', '10048', '646-472-3027',
        '646-472-3028', 'customer@walmart.com');
INSERT INTO pharmacy (pharm_name, pharm_city, pharm_state, pharm_zip,
            pharm_tel, pharm_fax, pharm_email)
VALUES ('Rite Aid', 'Smithtown', 'NY', '11787', '212-873-9773',
        '212-873-9775', 'customer@riteaid.com');
INSERT INTO pharmacy (pharm_name, pharm_city, pharm_state, pharm_zip,
            pharm_tel, pharm_fax, pharm_email)
VALUES ('Kroger', 'Amherst', 'NY', '11373', '716-309-4720',
```

```
        '716-309-4722', 'customer@kroger.com');
INSERT INTO pharmacy (pharm_name, pharm_city, pharm_state, pharm_zip,
            pharm_tel, pharm_fax, pharm_email)
VALUES ('Albertsons', 'Brooklyn', 'NY', '11224', '718-472-7303',
        '718-472-7304', 'customer@albertsons.com');
INSERT INTO pharmacy (pharm_name, pharm_city, pharm_state, pharm_zip,
            pharm_tel, pharm_fax, pharm_email)
VALUES ('McKesson', 'Flushing', 'NY', '11366', '917-204-3936',
        '917-204-3938', 'customer@mcKesson.com');
INSERT INTO pharmacy (pharm_name, pharm_city, pharm_state, pharm_zip,
            pharm_tel, pharm_fax, pharm_email)
VALUES ('Costco', 'Roslyn', 'NY', '11576', '646-473-9028',
        '646-473-9029', 'customer@costco.com');
INSERT INTO pharmacy (pharm_name, pharm_city, pharm_state, pharm_zip,
            pharm_tel, pharm_fax, pharm_email)
VALUES ('Cardinal', 'Hollis', 'NY', '11412', '718-367-2094',
        '718-367-2095', 'customer@cardinal.com');
INSERT INTO pharmacy (pharm_name, pharm_city, pharm_state, pharm_zip,
            pharm_tel, pharm_fax, pharm_email)
VALUES ('AmerisourceBergen', 'Monbasset', 'NY', '11303', '716-734-8364',
        '716-734-8365', 'customer@amerisourceBergen.com');


--Insert 12 rows of data into the table PRESCRIPTION
INSERT INTO prescription (presc_date, presc_use, items_no, pay_method,
            card_no, presc_note, doctor_id, patient_id)
VALUES (TO_DATE('02-AUG-2022','DD-MON-YYYY'), 'treat inflammation',
        1, 'credit card', '2830174387483284', 'once a day for 14 days',
        1, 3);
INSERT INTO prescription (presc_date, presc_use, items_no, pay_method,
            card_no, presc_note, doctor_id, patient_id)
VALUES (TO_DATE('03-AUG-2022','DD-MON-YYYY'), 'treat cold symptoms',
        3, 'credit card', '4734537940355473', 'for 14 days',
        4, 5);
INSERT INTO prescription (presc_date, presc_use, items_no, pay_method,
            card_no, presc_note, doctor_id, patient_id)
VALUES (TO_DATE('03-AUG-2022','DD-MON-YYYY'), 'treat hypertension',
        1, 'credit card', '8732749229364927', 'once a day for 28 days',
        9, 1);
INSERT INTO prescription (presc_date, presc_use, items_no, pay_method,
            card_no, presc_note, doctor_id, patient_id)
VALUES (TO_DATE('04-AUG-2022','DD-MON-YYYY'), 'treat stroke',
        1, 'credit card', '4803849477937930', 'once a day for 28 days',
        10, 7);
INSERT INTO prescription (presc_date, presc_use, items_no, pay_method,
            card_no, presc_note, doctor_id, patient_id)
VALUES (TO_DATE('05-AUG-2022','DD-MON-YYYY'), 'treat stomach ulcers',
        1, 'credit card', '8748038774693077', 'four times a day for 28 days',
        1, 3);
INSERT INTO prescription (presc_date, presc_use, items_no, pay_method,
            card_no, presc_note, doctor_id, patient_id)
VALUES (TO_DATE('08-AUG-2022','DD-MON-YYYY'), 'treat stomach ulcers',
        1, 'debt card', '5467378937657976', 'twice a day for 14 days',
        1, 2);
```

```
INSERT INTO prescription (presc_date, presc_use, items_no, pay_method,
            card_no, presc_note, doctor_id, patient_id)
VALUES (TO_DATE('09-AUG-2022','DD-MON-YYYY'), 'treat stomach ulcers',
        1, 'credit card', '7382749203472027', 'once a day for 21 days',
        4, 4);
INSERT INTO prescription (presc_date, presc_use, items_no, pay_method,
            card_no, presc_note, doctor_id, patient_id)
VALUES (TO_DATE('10-AUG-2022','DD-MON-YYYY'), 'treat allergy',
        1, 'credit card', '68849933879035 79', 'once a day for 10 days',
        7, 6);
INSERT INTO prescription (presc_date, presc_use, items_no, pay_method,
            card_no, presc_note, doctor_id, patient_id)
VALUES (TO_DATE('11-AUG-2022','DD-MON-YYYY'), 'treat hypertension',
        1, 'credit card', '3467907575024528', 'once a day for 28 days',
        10, 8);
INSERT INTO prescription (presc_date, presc_use, items_no, pay_method,
            card_no, presc_note, doctor_id, patient_id)
VALUES (TO_DATE('12-AUG-2022','DD-MON-YYYY'), 'treat stomach pain and
ulcers',
        2, 'credit card', '9842704784262035', 'for 14 days',
        4, 9);
INSERT INTO prescription (presc_date, presc_use, items_no, pay_method,
            card_no, presc_note, doctor_id, patient_id)
VALUES (TO_DATE('15-AUG-2022','DD-MON-YYYY'), 'treat hypertension',
        1, 'debt card', '2983774665893734', 'once a day for 28 days',
        9, 10);
INSERT INTO prescription (presc_date, presc_use, items_no, pay_method,
            card_no, presc_note, doctor_id, patient_id)
VALUES (TO_DATE('16-AUG-2022','DD-MON-YYYY'), 'treat pain and inflammation',
        2, 'credit card', '5379927877466499', 'for 14 days',
        6, 5);


--Insert 16 rows of data into the table PRESC_DRUG
INSERT INTO presc_drug (Quantity, total_cost, refill_no, batch_no,
            expiration_date, dispense_date, drug_code, pharm_id,
            presc_id)
VALUES (14, 46.06, 1, 'D01242022', TO_DATE('24-JUL-2023','DD-MON-YYYY'),
        TO_DATE('02-AUG-2022','DD-MON-YYYY'), 'D34843734', 1, 1);
INSERT INTO presc_drug (Quantity, total_cost, refill_no, batch_no,
            expiration_date, dispense_date, drug_code, pharm_id,
            presc_id)
VALUES (42, 48.30, 2, 'I07152022', TO_DATE('16-JUL-2023','DD-MON-YYYY'),
        TO_DATE('03-AUG-2022','DD-MON-YYYY'), 'I10837332', 1, 2);
INSERT INTO presc_drug (Quantity, total_cost, refill_no, batch_no,
            expiration_date, dispense_date, drug_code, pharm_id,
            presc_id)
VALUES (14, 11.62, 2, 'C05212022', TO_DATE('22-MAY-2023','DD-MON-YYYY'),
        TO_DATE('03-AUG-2022','DD-MON-YYYY'), 'C76488378', 1, 2);
INSERT INTO presc_drug (Quantity, total_cost, refill_no, batch_no,
            expiration_date, dispense_date, drug_code, pharm_id,
            presc_id)
VALUES (28, 36.68, 2, 'P08142022', TO_DATE('15-AUG-2023','DD-MON-YYYY'),
        TO_DATE('03-AUG-2022','DD-MON-YYYY'), 'P39857223', 1, 2);
```

```
INSERT INTO presc_drug (Quantity, total_cost, refill_no, batch_no,
            expiration_date, dispense_date, drug_code, pharm_id,
            presc_id)
VALUES (28, 226.52, 3, 'O03172022', TO_DATE('18-MAR-2023','DD-MON-YYYY'),
        TO_DATE('03-AUG-2022','DD-MON-YYYY'), 'O23547364', 1, 3);
INSERT INTO presc_drug (Quantity, total_cost, refill_no, batch_no,
            expiration_date, dispense_date, drug_code, pharm_id,
            presc_id)
VALUES (28, 100.52, 3, 'C04082022', TO_DATE('09-APR-2023','DD-MON-YYYY'),
        TO_DATE('04-AUG-2022','DD-MON-YYYY'), 'C53274803', 3, 4);
INSERT INTO presc_drug (Quantity, total_cost, refill_no, batch_no,
            expiration_date, dispense_date, drug_code, pharm_id,
            presc_id)
VALUES (112, 97.44, 2, 'D06132022', TO_DATE('14-JUN-2023','DD-MON-YYYY'),
        TO_DATE('05-AUG-2022','DD-MON-YYYY'), 'D20847476', 3, 5);
INSERT INTO presc_drug (Quantity, total_cost, refill_no, batch_no,
            expiration_date, dispense_date, drug_code, pharm_id,
            presc_id)
VALUES (28, 35.56, 3, 'F09192022', TO_DATE('20-SEP-2023','DD-MON-YYYY'),
        TO_DATE('08-AUG-2022','DD-MON-YYYY'), 'F68403835', 4, 6);
INSERT INTO presc_drug (Quantity, total_cost, refill_no, batch_no,
            expiration_date, dispense_date, drug_code, pharm_id,
            presc_id)
VALUES (21, 18.27, 4, 'R02172022', TO_DATE('18-FEB-2023','DD-MON-YYYY'),
        TO_DATE('09-AUG-2022','DD-MON-YYYY'), 'R87462748', 4, 7);
INSERT INTO presc_drug (Quantity, total_cost, refill_no, batch_no,
            expiration_date, dispense_date, drug_code, pharm_id,
            presc_id)
VALUES (10, 32.90, 1, 'D05182022', TO_DATE('19-MAY-2023','DD-MON-YYYY'),
        TO_DATE('10-AUG-2022','DD-MON-YYYY'), 'D34843734', 5, 8);
INSERT INTO presc_drug (Quantity, total_cost, refill_no, batch_no,
            expiration_date, dispense_date, drug_code, pharm_id,
            presc_id)
VALUES (28, 69.72, 3, 'H08222022', TO_DATE('23-AUG-2023','DD-MON-YYYY'),
        TO_DATE('11-AUG-2022','DD-MON-YYYY'), 'H47637494', 7, 9);
INSERT INTO presc_drug (Quantity, total_cost, refill_no, batch_no,
            expiration_date, dispense_date, drug_code, pharm_id,
            presc_id)
VALUES (42, 48.30, 2, 'I10142022', TO_DATE('15-OCT-2023','DD-MON-YYYY'),
        TO_DATE('12-AUG-2022','DD-MON-YYYY'), 'I10837332', 2, 10);
INSERT INTO presc_drug (Quantity, total_cost, refill_no, batch_no,
            expiration_date, dispense_date, drug_code, pharm_id,
            presc_id)
VALUES (14, 12.18, 2, 'R07272022', TO_DATE('28-JUL-2023','DD-MON-YYYY'),
        TO_DATE('12-AUG-2022','DD-MON-YYYY'), 'R87462748', 2, 10);
INSERT INTO presc_drug (Quantity, total_cost, refill_no, batch_no,
            expiration_date, dispense_date, drug_code, pharm_id,
            presc_id)
VALUES (28, 226.52, 3, 'O03282022', TO_DATE('29-MAR-2023','DD-MON-YYYY'),
        TO_DATE('15-AUG-2022','DD-MON-YYYY'), 'O23547364', 8, 11);
INSERT INTO presc_drug (Quantity, total_cost, refill_no, batch_no,
            expiration_date, dispense_date, drug_code, pharm_id,
            presc_id)
VALUES (42, 48.30, 2, 'I08092022', TO_DATE('10-AUG-2023','DD-MON-YYYY'),
```

```
        TO_DATE('16-AUG-2022','DD-MON-YYYY'), 'I10837332', 2, 12);
INSERT INTO presc_drug (Quantity, total_cost, refill_no, batch_no,
            expiration_date, dispense_date, drug_code, pharm_id,
            presc_id)
VALUES (14, 46.06, 2, 'D04162022', TO_DATE('17-APR-2023','DD-MON-YYYY'),
        TO_DATE('16-AUG-2022','DD-MON-YYYY'), 'D34843734', 2, 12);

COMMIT;
```

--20 SQL Queries (12 Basic, 8 Advanced)

```
--Query 1:
/*Select all columns and all rows from one table*/
SELECT *
FROM doctor;

--Query 2:
/*Select five columns and all rows from one table*/
SELECT patient_id, pat_first_name, pat_last_name,
       patient_dob, patient_ssn
FROM patient;

--Query 3:
/*Select all columns from all rows from one view*/
SELECT *
FROM cost_per_presc;

--Query 4:
/*Using a join on 2 tables, select all columns and all rows from
the tables without the use of a Cartesian product*/
SELECT *
FROM prescription pr
LEFT OUTER JOIN patient pa ON pr.patient_id = pa.patient_id;

--Query 5:
/*Select and order data retrieved from one table*/
SELECT drug_desc, drug_trademark, drug_cost
FROM drug
ORDER BY drug_cost;

--Query 6:
/*Using a join on 3 tables, select 5 columns from the 3 tables.
Use syntax that would limit the output to 10 rows*/
SELECT pr.presc_id, dr_first_name, dr_last_name,
       pat_first_name, pat_last_name
FROM prescription pr
LEFT OUTER JOIN doctor d ON pr.doctor_id = d.doctor_id
LEFT OUTER JOIN patient pa ON pr.patient_id = pa.patient_id
FETCH FIRST 10 ROWS ONLY;

--Query 7:
/*Select distinct rows using joins on 3 tables*/
```

```
SELECT DISTINCT drug_desc, manufacturer, pharm_name
FROM presc_drug pd
LEFT OUTER JOIN drug d ON pd.drug_code = d.drug_code
LEFT OUTER JOIN pharmacy ph ON pd.pharm_id = ph.pharm_id;

--Query 8:
/*Use GROUP BY and HAVING in a select statement using one or
more tables*/
SELECT pr.presc_id, SUM(total_cost) AS total_cost
FROM presc_drug pd
LEFT OUTER JOIN prescription pr ON pd.presc_id = pr.presc_id
GROUP BY pr.presc_id
HAVING SUM(total_cost) >= 100;

--Query 9:
/*Use IN clause to select data from one or more tables*/
SELECT pr.presc_id, sum(total_cost) AS total_cost, pharm_name
FROM presc_drug pd
LEFT OUTER JOIN prescription pr ON pd.presc_id = pr.presc_id
LEFT OUTER JOIN pharmacy ph ON pd.pharm_id = ph.pharm_id
WHERE pharm_name IN ('Walgreens', 'CVS Health', 'Rite Aid')
GROUP BY pharm_name, pr.presc_id;

--Query 10:
/*Select length of one column from one table (use LENGTH function)*/
SELECT LENGTH(drug_desc) AS "The length of drug name"
FROM drug;

--Query 11:
/*Delete one record from one table. Use select statements to
demonstrate the table contents before and after the DELETE
statement. Make sure you use ROLLBACK afterwards so that
the data will not be physically removed*/
SELECT * FROM presc_drug;

DELETE FROM presc_drug
WHERE presc_drug_id = 1;

SELECT * FROM presc_drug;

ROLLBACK;

--Query 12:
/*Update one record from one table. Use select statements to
demonstrate the table contents before and after the UPDATE
statement. Make sure you use ROLLBACK afterwards so that
the data will not be physically removed*/
SELECT * FROM drug;

UPDATE drug
SET drug_cost = 1.55
WHERE drug_code = 'I10837332';

SELECT * FROM drug;
```

```
ROLLBACK;

--Query 13:
/*List prescriptions prescribed on and after August 8, 2022 and
their information about who wrote these prescriptions and who
own them.*/
SELECT pr.presc_id AS presscription_ID,
       presc_date AS prescription_date,
       dr_first_name AS doctor_first_name,
       dr_last_name AS doctor_last_name,
       pat_first_name AS patient_first_name,
       pat_last_name AS patient_last_name
FROM prescription pr
LEFT OUTER JOIN doctor d ON pr.doctor_id = d.doctor_id
LEFT OUTER JOIN patient pa ON pr.patient_id = pa.patient_id
WHERE presc_date >= TO_DATE('August 08 2022', 'Month DD YYYY');

--Query 14:
/*Calculate the frequency of prescribed drugs among all prescriptions
and list frequencies in a descending order.*/
SELECT pd.drug_code, drug_desc,
       ROUND(COUNT(*) / (SELECT COUNT(*) FROM prescription), 3) AS
       "The frequency of prescribed drugs"
FROM presc_drug pd
LEFT OUTER JOIN drug d ON pd.drug_code = d.drug_code
GROUP BY pd.drug_code, drug_desc
ORDER BY "The frequency of prescribed drugs" DESC;

--Query 15:
/*List the top 3 of pharmacies based on their total sale prices.*/
SELECT pharm_name AS "Pharmacy Name",
       SUM(total_cost) AS "Total Sale Prices"
FROM presc_drug pd
LEFT OUTER JOIN pharmacy ph ON pd.pharm_id = ph.pharm_id
GROUP BY pharm_name
ORDER BY "Total Sale Prices" DESC
FETCH FIRST 3 ROWS ONLY;

--Query 16:
/*List doctors who prescribed Dicyclomine, Famotidine, Ranitidine for
treating gastric and intestinal ulcers.*/
SELECT dr_first_name AS doctor_first_name,
       dr_last_name AS doctor_last_name,
       drug_desc AS drug_generic_name, drug_uses
FROM presc_drug pd
LEFT OUTER JOIN drug d ON pd.drug_code = d.drug_code
LEFT OUTER JOIN prescription pr ON pd.presc_id = pr.presc_id
LEFT OUTER JOIN doctor dr ON pr.doctor_id = dr.doctor_id
WHERE drug_desc IN ('Dicyclomine', 'Famotidine', 'Ranitidine');

--Query 17:
/*List patients who obtained prescriptions from Dr.Fernand Hanks and
how much they paied for their prescriptions*/
```

```
SELECT pat_first_name AS patient_first_name,
       pat_last_name AS patient_first_name,
       COUNT(*) AS "The number of prescriptions",
       SUM(total_cost) AS "Prescription payment"
FROM presc_drug pd
LEFT OUTER JOIN prescription pr ON pd.presc_id = pr.presc_id
LEFT OUTER JOIN doctor d ON pr.doctor_id = d.doctor_id
LEFT OUTER JOIN patient pa ON pr.patient_id = pa.patient_id
WHERE dr_first_name = 'Fernand' AND dr_last_name = 'Hanks'
GROUP BY pat_first_name, pat_last_name;

--Query 18:
/*List which pharmaceutical manufacturer sells the drug with the
highest total sale pricess according to the PRESC_DRUG table.*/
SELECT manufacturer AS "Pharmaceutical manufacturer",
       drug_desc AS "Drug generic name",
       SUM(quantity) AS "The total prescribed quantity",
       SUM(total_cost) AS "Total sale prices"
FROM presc_drug pd
LEFT OUTER JOIN drug d ON pd.drug_code = d.drug_code
GROUP BY manufacturer, drug_desc
HAVING SUM(total_cost) = (
       SELECT MAX(SUM(total_cost))
       FROM presc_drug pd
       LEFT OUTER JOIN drug d ON pd.drug_code = d.drug_code
       GROUP BY drug_desc
       );

--Query 19:
/*List which pharmacy sold Dexamethasone, which doctor prescribed
this drug, and which patient received this drug.*/
SELECT pharm_name AS "Pharmacy Name",
       dr_first_name AS "Doctor First Name",
       dr_last_name AS "Doctor Last Name",
       pat_first_name AS "Patient First Name",
       pat_last_name AS "Patient Last Name"
FROM presc_drug pd
LEFT OUTER JOIN drug d ON pd.drug_code = d.drug_code
LEFT OUTER JOIN pharmacy ph ON pd.pharm_id = ph.pharm_id
LEFT OUTER JOIN prescription pr ON pd.presc_id = pr.presc_id
LEFT OUTER JOIN doctor dr ON pr.doctor_id = dr.doctor_id
LEFT OUTER JOIN patient pa ON pr.patient_id = pa.patient_id
WHERE drug_desc = 'Dexamethasone';

--Query 20:
/*Among drugs manufactured by Johnson and Johnson, which drug offers
the best profit to this pharmaceutical company according to the
PRESC_DRUG table.*/
SELECT drug_desc, SUM(total_cost)
FROM presc_drug pd
LEFT OUTER JOIN drug d ON pd.drug_code = d.drug_code
WHERE manufacturer = 'Johnson and Johnson'
GROUP BY drug_desc
HAVING SUM(total_cost) = (
```

```
SELECT MAX(SUM(total_cost))
FROM presc_drug pd
LEFT OUTER JOIN drug d ON pd.drug_code = d.drug_code
WHERE manufacturer = 'Johnson and Johnson'
GROUP BY drug_desc
);
```

8.4 DDL, DML, and Query Output

8.4.1 DDL Output:

```
Trigger TRG_PRESC_DRUG dropped.


Trigger TRG_PRESCRIPTION dropped.


Trigger TRG_DOCTOR dropped.


Trigger TRG_PATIENT dropped.


Trigger TRG_DRUG dropped.


Trigger TRG_PHARMACY dropped.


Sequence SEQ_DOCTOR_ID dropped.


Sequence SEQ_PATIENT_ID dropped.


Sequence SEQ_PRESC_ID dropped.


Sequence SEQ_PHARM_ID dropped.


Sequence SEQ_PRESC_DRUG_ID dropped.


View DOCTOR_VIEW dropped.


View PATIENT_VIEW dropped.
```

```
View DRUG_VIEW dropped.

View PHARMACY_VIEW dropped.

View PRESCRIPTION_VIEW dropped.

View PRESC_DRUG_VIEW dropped.

View PRESC_PAT_DR dropped.

View COST_PER_PRESC dropped.

Index DOCTOR_TEL_IDX dropped.

Index PATIENT_SSN_IDX dropped.

Index PATIENT_TEL_IDX dropped.

Index PHARM_TEL_IDX dropped.

Index PRESC_DOCTOR_FK_IDX dropped.

Index PRESC_PATIENT_FK_IDX dropped.

Index PRESC_DRUG_DRUG_FK_IDX dropped.

Index PRESC_DRUG_PHARMACY_FK_IDX dropped.

Index PRESC_DRUG_PRESCRIPTION_FK_IDX dropped.

Index DR_FIRST_NAME_IDX dropped.

Index DR_LAST_NAME_IDX dropped.

Index PAT_FIRST_NAME_IDX dropped.
```

```
Index PAT_LAST_NAME_IDX dropped.

Index DRUG_COST_IDX dropped.

Index PHARM_NAME_IDX dropped.

Index PHARM_CITY_IDX dropped.

Index PHARM_ZIP_IDX dropped.

Table PRESC_DRUG dropped.

Table PRESCRIPTION dropped.

Table DOCTOR dropped.

Table PATIENT dropped.

Table DRUG dropped.

Table PHARMACY dropped.

Table DOCTOR created.

Table DOCTOR altered.

Table DRUG created.

Table DRUG altered.

Table PATIENT created.

Table PATIENT altered.

Table PHARMACY created.
```

```
Table PHARMACY altered.

Table PRESCRIPTION created.

Table PRESCRIPTION altered.

Table PRESC_DRUG created.

Table PRESC_DRUG altered.

Table PRESCRIPTION altered.

Table PRESCRIPTION altered.

Table PRESC_DRUG altered.

Table PRESC_DRUG altered.

Table PRESC_DRUG altered.

INDEX DOCTOR_TEL_IDX created.

INDEX PATIENT_SSN_IDX created.

INDEX PATIENT_TEL_IDX created.

INDEX PHARM_TEL_IDX created.

Index PRESC_DOCTOR_FK_IDX created.

Index PRESC_PATIENT_FK_IDX created.

Index PRESC_DRUG_DRUG_FK_IDX created.

Index PRESC_DRUG_PHARMACY_FK_IDX created.
```

```
Index PRESC_DRUG_PRESCRIPTION_FK_IDX created.

Index DR_FIRST_NAME_IDX created.

Index DR_LAST_NAME_IDX created.

Index PAT_FIRST_NAME_IDX created.

Index PAT_LAST_NAME_IDX created.

Index DRUG_COST_IDX created.

Index PHARM_NAME_IDX created.

Index PHARM_CITY_IDX created.

Index PHARM_ZIP_IDX created.

Table DOCTOR altered.

Table PATIENT altered.

Table DRUG altered.

Table PHARMACY altered.

Table PRESCRIPTION altered.

Table PRESC_DRUG altered.

View DOCTOR_VIEW created.

View PATIENT_VIEW created.

View DRUG_VIEW created.
```

```
View PHARMACY_VIEW created.


View PRESCRIPTION_VIEW created.


View PRESC_DRUG_VIEW created.


View PRESC_PAT_DR created.


View COST_PER_PRESC created.


Sequence SEQ_DOCTOR_ID created.


Sequence SEQ_PATIENT_ID created.


Sequence SEQ_PRESC_ID created.


Sequence SEQ_PHARM_ID created.


Sequence SEQ_PRESC_DRUG_ID created.


Trigger TRG_DOCTOR compiled


Trigger TRG_PATIENT compiled


Trigger TRG_PRESCRIPTION compiled


Trigger TRG_DRUG compiled


Trigger TRG_PHARMACY compiled


Trigger TRG_PRESC_DRUG compiled


TABLE_NAME
----------------------------------
DOCTOR
DRUG
PATIENT
PHARMACY
```

```
PRESCRIPTION
PRESC_DRUG

6 rows selected.


OBJECT_NAME                                STATUS  CREATED   LAST_DDL_TIME
------------------------------------------ ------- --------- -------------
COST_PER_PRESC                             VALID   11-DEC-22 11-DEC-22
PRESC_PAT_DR                               VALID   11-DEC-22 11-DEC-22
DOCTOR                                     VALID   11-DEC-22 11-DEC-22
DOCTOR_PK                                  VALID   11-DEC-22 11-DEC-22
DRUG                                       VALID   11-DEC-22 11-DEC-22
DRUG_PK                                    VALID   11-DEC-22 11-DEC-22
PATIENT                                    VALID   11-DEC-22 11-DEC-22
PATIENT_PK                                 VALID   11-DEC-22 11-DEC-22
PHARMACY                                   VALID   11-DEC-22 11-DEC-22
PHARMACY_PK                                VALID   11-DEC-22 11-DEC-22
PRESCRIPTION                               VALID   11-DEC-22 11-DEC-22

OBJECT_NAME                                STATUS  CREATED   LAST_DDL_TIME
------------------------------------------ ------- --------- -------------
PRESCRIPTION_PK                            VALID   11-DEC-22 11-DEC-22
PRESC_DRUG                                 VALID   11-DEC-22 11-DEC-22
PRESC_DRUG_PK                              VALID   11-DEC-22 11-DEC-22
DOCTOR_TEL_IDX                             VALID   11-DEC-22 11-DEC-22
PATIENT_SSN_IDX                            VALID   11-DEC-22 11-DEC-22
PATIENT_TEL_IDX                            VALID   11-DEC-22 11-DEC-22
PHARM_TEL_IDX                              VALID   11-DEC-22 11-DEC-22
PRESC_DOCTOR_FK_IDX                        VALID   11-DEC-22 11-DEC-22
PRESC_PATIENT_FK_IDX                       VALID   11-DEC-22 11-DEC-22
PRESC_DRUG_DRUG_FK_IDX                     VALID   11-DEC-22 11-DEC-22
PRESC_DRUG_PHARMACY_FK_IDX                 VALID   11-DEC-22 11-DEC-22

OBJECT_NAME                                STATUS  CREATED   LAST_DDL_TIME
------------------------------------------ ------- --------- -------------
PRESC_DRUG_PRESCRIPTION_FK_IDX             VALID   11-DEC-22 11-DEC-22
DR_FIRST_NAME_IDX                          VALID   11-DEC-22 11-DEC-22
DR_LAST_NAME_IDX                           VALID   11-DEC-22 11-DEC-22
PAT_FIRST_NAME_IDX                         VALID   11-DEC-22 11-DEC-22
PAT_LAST_NAME_IDX                          VALID   11-DEC-22 11-DEC-22
DRUG_COST_IDX                              VALID   11-DEC-22 11-DEC-22
PHARM_NAME_IDX                             VALID   11-DEC-22 11-DEC-22
PHARM_CITY_IDX                             VALID   11-DEC-22 11-DEC-22
PHARM_ZIP_IDX                              VALID   11-DEC-22 11-DEC-22
SEQ_DOCTOR_ID                              VALID   11-DEC-22 11-DEC-22
SEQ_PATIENT_ID                             VALID   11-DEC-22 11-DEC-22

OBJECT_NAME                                STATUS  CREATED   LAST_DDL_TIME
------------------------------------------ ------- --------- -------------
SEQ_PRESC_ID                               VALID   11-DEC-22 11-DEC-22
SEQ_PHARM_ID                               VALID   11-DEC-22 11-DEC-22
SEQ_PRESC_DRUG_ID                          VALID   11-DEC-22 11-DEC-22
DOCTOR_VIEW                                VALID   11-DEC-22 11-DEC-22
```

```
PATIENT_VIEW                             VALID   11-DEC-22 11-DEC-22
DRUG_VIEW                                VALID   11-DEC-22 11-DEC-22
PHARMACY_VIEW                            VALID   11-DEC-22 11-DEC-22
PRESCRIPTION_VIEW                        VALID   11-DEC-22 11-DEC-22
PRESC_DRUG_VIEW                          VALID   11-DEC-22 11-DEC-22
TRG_DOCTOR                               VALID   11-DEC-22 11-DEC-22
TRG_PATIENT                              VALID   11-DEC-22 11-DEC-22

OBJECT_NAME                              STATUS  CREATED   LAST_DDL_TIME
---------------------------------------- ------- --------- -------------
TRG_PRESCRIPTION                         VALID   11-DEC-22 11-DEC-22
TRG_DRUG                                 VALID   11-DEC-22 11-DEC-22
TRG_PHARMACY                             VALID   11-DEC-22 11-DEC-22
TRG_PRESC_DRUG                           VALID   11-DEC-22 11-DEC-22

48 rows selected.
```

8.4.2 DML Output:

```
1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.
```

```
1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.
```

```
1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.
```

```
1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.
```

```
1 row inserted.


1 row inserted.


1 row inserted.


Commit complete.
```

8.4.3 Query Output:

```
DOCTOR_ID DR_FIRST_NAME                  DR_LAST_NAME
DR_SPECIALTY                             DR_ADDRESS
DOCTOR_TEL   DOCTOR_EMAIL                                        CREATED_BY
DATE_CREA MODIFIED_BY                    DATE_MODI
---------- ------------------------------ ------------------------------ ----
---------------------------------- ----------------------------------------
---------- ------------ -------------------------------------------------- --
--------------------------- --------- ------------------------------ -------
--
         1 Fernand                       Hanks
Internal Medicine                        254 Bleeker, New York, NY 10005
212-545-1216 fhands@gmail.com                                    SYSTEM
11-DEC-22 SYSTEM                         11-DEC-22
         2 Marilyn                       Frantzen
Pediatrics                               100 East 87th, New York, NY 10015
212-569-3762 mfrantzen@gmail.com                                 SYSTEM
11-DEC-22 SYSTEM                         11-DEC-22
         3 Anita                         Morris
Orthopedics                              34 Maiden Lane, New York, NY 10015
212-388-3476 amorris@gmail.com                                   SYSTEM
11-DEC-22 SYSTEM                         11-DEC-22
         4 Rick                          Chow
Family Medicine                          56 10th Avenue, New York, NY 10015
212-485-1762 rchow@gmail.com                                     SYSTEM
11-DEC-22 SYSTEM                         11-DEC-22
         5 Tom                           Wojick
Neurology                                518 West 120th, New York, NY 10025
212-684-9832 twojick@gmail.com                                   SYSTEM
11-DEC-22 SYSTEM                         11-DEC-22
         6 Nina                          Schorin
Urology                                  210 West 101st, New York, NY 10025
212-774-2685 nschorin@gmail.com                                  SYSTEM
11-DEC-22 SYSTEM                         11-DEC-22
```

```
         7 Todd                           Smythe
Dermatology                             210 West 103st, New York, NY 10025
212-775-9837 tsmythe@gmail.com                                  SYSTEM
11-DEC-22 SYSTEM                        11-DEC-22
         8 Charles                        Lowry
Ophthalmology                           518 West 120th, New York, NY 10025
212-566-3644 clowry@gmail.com                                   SYSTEM
11-DEC-22 SYSTEM                        11-DEC-22
         9 Gary                           Pertez
Cardiology                              34 Sixth Ave, New York, NY 10035
212-787-7543 gpertez@gmail.com                                  SYSTEM
11-DEC-22 SYSTEM                        11-DEC-22
        10 Irene                          Willig
Cardiology                              415 West 101st, New York, NY 10045
212-845-2548 iwillig@gmail.com                                  SYSTEM
11-DEC-22 SYSTEM                        11-DEC-22

10 rows selected.


PATIENT_ID PAT_FIRST_NAME                PAT_LAST_NAME
PATIENT_D PATIENT_SSN
---------- ------------------------------ ------------------------------ ----
----- -----------
         1 Fred                           Crocitto                       02-
JAN-00 218-51-3692
         2 Laetia                         Enison                         16-
MAR-98 217-61-9372
         3 Angel                          Moskowitz                      22-
OCT-94 213-41-9281
         4 Judith                         Olvsade                        08-
JUN-92 214-61-2847
         5 Catherine                      Mierzwa                        18-
NOV-96 215-71-4923
         6 Judy                           Sethi                          12-
APR-99 217-81-8237
         7 Winsome                        Laporte                        21-
MAY-95 213-42-3648
         8 Sean                           Pineda                         27-
AUG-91 212-31-5493
         9 Angela                         Torres                         05-
DEC-92 212-31-1038
        10 Monica                         Waldman                        18-
FEB-98 216-71-4037

10 rows selected.


  PRESC_ID TOTAL_COST PAT_FIRST_NAME                PAT_LAST_NAME
PAY_METHOD           CARD_NO
---------- ---------- ------------------------------ ------------------------
------ -------------------- ----------------
         1      46.06 Angel                          Moskowitz
credit card          2830174387483284
```

```
         2       96.6 Catherine                      Mierzwa
credit card          4734537940355473
         3     226.52 Fred                           Crocitto
credit card          8732749229364927
         4     100.52 Winsome                        Laporte
credit card          4803849477937930
         5      97.44 Angel                          Moskowitz
credit card          8748038774693077
         6      35.56 Laetia                         Enison
debt card            5467378937657976
         7      18.27 Judith                         Olvsade
credit card          7382749203472027
         8       32.9 Judy                           Sethi
credit card          6884993387903579
         9      69.72 Sean                           Pineda
credit card          3467907575024528
        10      60.48 Angela                         Torres
credit card          9842704784262035
        11     226.52 Monica                         Waldman
debt card            2983774665893734

  PRESC_ID TOTAL_COST PAT_FIRST_NAME                PAT_LAST_NAME
PAY_METHOD           CARD_NO
---------- ---------- ------------------------------ ------------------------
------ -------------------- ----------------
        12      94.36 Catherine                     Mierzwa
credit card          5379927877466499

12 rows selected.


  PRESC_ID PRESC_DAT PRESC_USE
ITEMS_NO PAY_METHOD           CARD_NO          PRESC_NOTE
DOCTOR_ID PATIENT_ID CREATED_BY                    DATE_CREA MODIFIED_BY
DATE_MODI PATIENT_ID PAT_FIRST_NAME                PAT_LAST_NAME
PATIENT_D PATIENT_SSN P PAT_ADDRESS
PATIENT_TEL  PATIENT_EMAIL                                      CREATED_BY
DATE_CREA MODIFIED_BY                    DATE_MODI
---------- --------- -------------------------------------------------- -----
----- -------------------- --------------- --------------------------------
---------------------------------------------------------------- ---------
- ---------- ------------------------------ --------- ----------------------
------- --------- ---------- ------------------------------ -----------------
------------- --------- ----------- - ------------------------------------
----------- ------------ -------------------------------------------------- -
----------------------------- --------- ------------------------------ ------
---
         1 02-AUG-22 treat inflammation
1 credit card          2830174387483284 once a day for 14 days
1          3 SYSTEM                          11-DEC-22 SYSTEM
11-DEC-22          3 Angel                           Moskowitz
22-OCT-94 213-41-9281 F 320 John St., Ft. Lee, NY 07024
201-293-2846 amoskowitz@gmail.com                               SYSTEM
11-DEC-22 SYSTEM                         11-DEC-22
```

```
         2 03-AUG-22 treat cold symptoms
3 credit card          4734537940355473 for 14 days
4          5 SYSTEM                          11-DEC-22 SYSTEM
11-DEC-22          5 Catherine                         Mierzwa
18-NOV-96 215-71-4923 F 22-70 41st St., Astoria, NY 11105
215-387-9027 cmierzwa@gmail.com                                    SYSTEM
11-DEC-22 SYSTEM                          11-DEC-22
         3 03-AUG-22 treat hypertension
1 credit card          8732749229364927 once a day for 28 days
9          1 SYSTEM                          11-DEC-22 SYSTEM
11-DEC-22          1 Fred                              Crocitto
02-JAN-00 218-51-3692 M 101-09 120th St., Richmond Hill, NY 11419
718-667-5692 fcrocitto@gmail.com                                   SYSTEM
11-DEC-22 SYSTEM                          11-DEC-22
         4 04-AUG-22 treat stroke
1 credit card          4803849477937930 once a day for 28 days
10          7 SYSTEM                           11-DEC-22 SYSTEM
11-DEC-22          7 Winsome                           Laporte
21-MAY-95 213-42-3648 M 268 E. 3rd St, Brooklyn, NY 11226
718-533-2947 wlaporte@gmail.com                                    SYSTEM
11-DEC-22 SYSTEM                          11-DEC-22
         5 05-AUG-22 treat stomach ulcers
1 credit card          8748038774693077 four times a day for 28 days
1          3 SYSTEM                          11-DEC-22 SYSTEM
11-DEC-22          3 Angel                             Moskowitz
22-OCT-94 213-41-9281 F 320 John St., Ft. Lee, NY 07024
201-293-2846 amoskowitz@gmail.com                                  SYSTEM
11-DEC-22 SYSTEM                          11-DEC-22
         6 08-AUG-22 treat stomach ulcers
1 debt card            5467378937657976 twice a day for 14 days
1          2 SYSTEM                          11-DEC-22 SYSTEM
11-DEC-22          2 Laetia                            Enison
16-MAR-98 217-61-9372 F 144-61 87th Ave, Jamaica, NY 11435
201-632-3857 lenison@gmail.com                                     SYSTEM
11-DEC-22 SYSTEM                          11-DEC-22
         7 09-AUG-22 treat stomach ulcers
1 credit card          7382749203472027 once a day for 21 days
4          4 SYSTEM                          11-DEC-22 SYSTEM
11-DEC-22          4 Judith                            Olvsade
08-JUN-92 214-61-2847 F 29 Elmwood Ave., Montclair, NY 07042
214-826-1037 jolvsade@gmail.com                                    SYSTEM
11-DEC-22 SYSTEM                          11-DEC-22
         8 10-AUG-22 treat allergy
1 credit card          6884993387903579 once a day for 10 days
7          6 SYSTEM                          11-DEC-22 SYSTEM
11-DEC-22          6 Judy                              Sethi
12-APR-99 217-81-8237 F 38 Bay 26th St., Brooklyn, NY 11214
716-682-8263 jsethi@gmail.com                                      SYSTEM
11-DEC-22 SYSTEM                          11-DEC-22
         9 11-AUG-22 treat hypertension
1 credit card          3467907575024528 once a day for 28 days
10          8 SYSTEM                           11-DEC-22 SYSTEM
11-DEC-22          8 Sean                              Pineda
27-AUG-91 212-31-5493 M 3 Salem Rd., New City, NY 10956
```

```
212-573-2074 spineda@gmail.com                                    SYSTEM
11-DEC-22 SYSTEM                        11-DEC-22
        10 12-AUG-22 treat stomach pain and ulcers
2 credit card          9842704784262035 for 14 days
4          9 SYSTEM                        11-DEC-22 SYSTEM
11-DEC-22          9 Angela                          Torres
05-DEC-92 212-31-1038 F 509 2nd St #4L, Brooklyn, NY 11215
718-394-3877 atorres@gmail.com                                    SYSTEM
11-DEC-22 SYSTEM                        11-DEC-22
        11 15-AUG-22 treat hypertension
1 debt card            2983774665893734 once a day for 28 days
9         10 SYSTEM                        11-DEC-22 SYSTEM
11-DEC-22         10 Monica                          Waldman
18-FEB-98 216-71-4037 F 257 Depot Rd., Huntington, NY 11766
718-875-9048 mwaldman@gmail.com                                   SYSTEM
11-DEC-22 SYSTEM                        11-DEC-22

  PRESC_ID PRESC_DAT PRESC_USE
ITEMS_NO PAY_METHOD           CARD_NO          PRESC_NOTE
DOCTOR_ID PATIENT_ID CREATED_BY                    DATE_CREA MODIFIED_BY
DATE_MODI PATIENT_ID PAT_FIRST_NAME                PAT_LAST_NAME
PATIENT_D PATIENT_SSN P PAT_ADDRESS
PATIENT_TEL  PATIENT_EMAIL                                        CREATED_BY
DATE_CREA MODIFIED_BY                   DATE_MODI
---------- --------- -------------------------------------------------- -----
----- ------------------- --------------- --------------------------------
-------------------------------------------------------------------- ---------
- ---------- ------------------------------ --------- ----------------------
------- --------- ---------- ------------------------------ -----------------
------------ --------- ----------- - --------------------------------------
----------- ------------ ------------------------------------------------- -
----------------------------- --------- ------------------------------ ------
---
        12 16-AUG-22 treat pain and inflammation
2 credit card          5379927877466499 for 14 days
6          5 SYSTEM                        11-DEC-22 SYSTEM
11-DEC-22          5 Catherine                        Mierzwa
18-NOV-96 215-71-4923 F 22-70 41st St., Astoria, NY 11105
215-387-9027 cmierzwa@gmail.com                                   SYSTEM
11-DEC-22 SYSTEM                        11-DEC-22

12 rows selected.


DRUG_DESC                                          DRUG_TRADEMARK
DRUG_COST
-------------------------------------------------- --------------------------
---- ----------
Cetirizine                                         Zyrtec
.83
Dicyclomine                                        Bentyl
.87
Ranitidine                                         Zantac
.87
```

```
Ibuprofen                                          Advil
1.15
Famotidine                                         Pepcid
1.27
Pseudoephedrine                                    Sudafed
1.31
Hydrochlorothiazide                                Esidrix
2.49
Dexamethasone                                      Decadron
3.29
Clopidogrel                                        Plavix
3.59
Olmesartan                                         Benicar
8.09

10 rows selected.


  PRESC_ID DR_FIRST_NAME                  DR_LAST_NAME
PAT_FIRST_NAME                 PAT_LAST_NAME
---------- ------------------------------ ------------------------------ ----
-------------------------- ------------------------------
         1 Fernand                        Hanks
Angel                          Moskowitz
         2 Rick                           Chow
Catherine                      Mierzwa
         3 Gary                           Pertez                         Fred
Crocitto
         4 Irene                          Willig
Winsome                        Laporte
         5 Fernand                        Hanks
Angel                          Moskowitz
         6 Fernand                        Hanks
Laetia                         Enison
         7 Rick                           Chow
Judith                         Olvsade
         8 Todd                           Smythe                         Judy
Sethi
         9 Irene                          Willig                         Sean
Pineda
        10 Rick                           Chow
Angela                         Torres

10 rows selected.


DRUG_DESC                                          MANUFACTURER
PHARM_NAME
-------------------------------------------------- --------------------------
------------------------ ------------------------------
Ibuprofen                                          Pfizer
CVS Health
Olmesartan                                         Lupin Limited
Costco
```

```
Clopidogrel                                        Bristol-Myers Squibb
Walmart
Hydrochlorothiazide                                Merck
McKesson
Dexamethasone                                      Fera Pharmaceuticals
Kroger
Ranitidine                                         GlaxoSmithKline
CVS Health
Dexamethasone                                      Fera Pharmaceuticals
CVS Health
Cetirizine                                         Johnson and Johnson
Walgreens
Ranitidine                                         GlaxoSmithKline
Rite Aid
Dicyclomine                                        Aptalis Pharma Canada
Walmart
Famotidine                                         Johnson and Johnson
Rite Aid

DRUG_DESC                                          MANUFACTURER
PHARM_NAME
-------------------------------------------------- --------------------------
---------------------- ------------------------------
Dexamethasone                                      Fera Pharmaceuticals
Walgreens
Ibuprofen                                          Pfizer
Walgreens
Pseudoephedrine                                    Johnson and Johnson
Walgreens
Olmesartan                                         Lupin Limited
Walgreens

15 rows selected.


  PRESC_ID TOTAL_COST
---------- ----------
        11     226.52
         4     100.52
         3     226.52


  PRESC_ID TOTAL_COST PHARM_NAME
---------- ---------- ------------------------------
         6      35.56 Rite Aid
        10      60.48 CVS Health
        12      94.36 CVS Health
         2       96.6 Walgreens
         3     226.52 Walgreens
         1      46.06 Walgreens
         7      18.27 Rite Aid

7 rows selected.
```

```
The length of drug name
-----------------------
                      9
                     10
                     11
                     10
                     13
                     19
                     10
                     11
                     15
                     10

10 rows selected.


PRESC_DRUG_ID   QUANTITY TOTAL_COST  REFILL_NO BATCH_NO
EXPIRATIO DISPENSE_ DRUG_CODE    PHARM_ID   PRESC_ID CREATED_BY
DATE_CREA MODIFIED_BY                    DATE_MODI
-------------- ---------- ---------- ---------- ------------------------------
--------- --------- ---------- ---------- ---------- ------------------------
------ --------- ------------------------------ ---------
             1         14      46.06          1 D01242022
24-JUL-23 02-AUG-22 D34843734           1          1 SYSTEM
11-DEC-22 SYSTEM                         11-DEC-22
             2         42       48.3          2 I07152022
16-JUL-23 03-AUG-22 I10837332           1          2 SYSTEM
11-DEC-22 SYSTEM                         11-DEC-22
             3         14      11.62          2 C05212022
22-MAY-23 03-AUG-22 C76488378           1          2 SYSTEM
11-DEC-22 SYSTEM                         11-DEC-22
             4         28      36.68          2 P08142022
15-AUG-23 03-AUG-22 P39857223           1          2 SYSTEM
11-DEC-22 SYSTEM                         11-DEC-22
             5         28     226.52          3 O03172022
18-MAR-23 03-AUG-22 O23547364           1          3 SYSTEM
11-DEC-22 SYSTEM                         11-DEC-22
             6         28     100.52          3 C04082022
09-APR-23 04-AUG-22 C53274803           3          4 SYSTEM
11-DEC-22 SYSTEM                         11-DEC-22
             7        112      97.44          2 D06132022
14-JUN-23 05-AUG-22 D20847476           3          5 SYSTEM
11-DEC-22 SYSTEM                         11-DEC-22
             8         28      35.56          3 F09192022
20-SEP-23 08-AUG-22 F68403835           4          6 SYSTEM
11-DEC-22 SYSTEM                         11-DEC-22
             9         21      18.27          4 R02172022
18-FEB-23 09-AUG-22 R87462748           4          7 SYSTEM
11-DEC-22 SYSTEM                         11-DEC-22
            10         10       32.9          1 D05182022
19-MAY-23 10-AUG-22 D34843734           5          8 SYSTEM
11-DEC-22 SYSTEM                         11-DEC-22
```

```
           11         28      69.72          3 H08222022
23-AUG-23 11-AUG-22 H47637494           7          9 SYSTEM
11-DEC-22 SYSTEM                         11-DEC-22

PRESC_DRUG_ID   QUANTITY TOTAL_COST  REFILL_NO BATCH_NO
EXPIRATIO DISPENSE_ DRUG_CODE    PHARM_ID   PRESC_ID CREATED_BY
DATE_CREA MODIFIED_BY                    DATE_MODI
------------- ---------- ---------- ---------- ------------------------------
--------- --------- ---------- ---------- ---------- ------------------------
------ --------- ------------------------------ ---------
           12         42       48.3          2 I10142022
15-OCT-23 12-AUG-22 I10837332           2         10 SYSTEM
11-DEC-22 SYSTEM                         11-DEC-22
           13         14      12.18          2 R07272022
28-JUL-23 12-AUG-22 R87462748           2         10 SYSTEM
11-DEC-22 SYSTEM                         11-DEC-22
           14         28     226.52          3 O03282022
29-MAR-23 15-AUG-22 O23547364           8         11 SYSTEM
11-DEC-22 SYSTEM                         11-DEC-22
           15         42       48.3          2 I08092022
10-AUG-23 16-AUG-22 I10837332           2         12 SYSTEM
11-DEC-22 SYSTEM                         11-DEC-22
           16         14      46.06          2 D04162022
17-APR-23 16-AUG-22 D34843734           2         12 SYSTEM
11-DEC-22 SYSTEM                         11-DEC-22

16 rows selected.


1 row deleted.


PRESC_DRUG_ID   QUANTITY TOTAL_COST  REFILL_NO BATCH_NO
EXPIRATIO DISPENSE_ DRUG_CODE    PHARM_ID   PRESC_ID CREATED_BY
DATE_CREA MODIFIED_BY                    DATE_MODI
------------- ---------- ---------- ---------- ------------------------------
--------- --------- ---------- ---------- ---------- ------------------------
------ --------- ------------------------------ ---------
            2         42       48.3          2 I07152022
16-JUL-23 03-AUG-22 I10837332           1          2 SYSTEM
11-DEC-22 SYSTEM                         11-DEC-22
            3         14      11.62          2 C05212022
22-MAY-23 03-AUG-22 C76488378           1          2 SYSTEM
11-DEC-22 SYSTEM                         11-DEC-22
            4         28      36.68          2 P08142022
15-AUG-23 03-AUG-22 P39857223           1          2 SYSTEM
11-DEC-22 SYSTEM                         11-DEC-22
            5         28     226.52          3 O03172022
18-MAR-23 03-AUG-22 O23547364           1          3 SYSTEM
11-DEC-22 SYSTEM                         11-DEC-22
            6         28     100.52          3 C04082022
09-APR-23 04-AUG-22 C53274803           3          4 SYSTEM
11-DEC-22 SYSTEM                         11-DEC-22
```

```
            7        112      97.44          2 D06132022
14-JUN-23 05-AUG-22 D20847476           3          5 SYSTEM
11-DEC-22 SYSTEM                         11-DEC-22
            8         28      35.56          3 F09192022
20-SEP-23 08-AUG-22 F68403835           4          6 SYSTEM
11-DEC-22 SYSTEM                         11-DEC-22
            9         21      18.27          4 R02172022
18-FEB-23 09-AUG-22 R87462748           4          7 SYSTEM
11-DEC-22 SYSTEM                         11-DEC-22
           10         10       32.9          1 D05182022
19-MAY-23 10-AUG-22 D34843734           5          8 SYSTEM
11-DEC-22 SYSTEM                         11-DEC-22
           11         28      69.72          3 H08222022
23-AUG-23 11-AUG-22 H47637494           7          9 SYSTEM
11-DEC-22 SYSTEM                         11-DEC-22
           12         42       48.3          2 I10142022
15-OCT-23 12-AUG-22 I10837332           2         10 SYSTEM
11-DEC-22 SYSTEM                         11-DEC-22

PRESC_DRUG_ID   QUANTITY TOTAL_COST  REFILL_NO BATCH_NO
EXPIRATIO DISPENSE_ DRUG_CODE    PHARM_ID   PRESC_ID CREATED_BY
DATE_CREA MODIFIED_BY                    DATE_MODI
------------- ---------- ---------- ---------- ------------------------------
--------- --------- ---------- ---------- ---------- -------------------------
------ --------- ------------------------------ ---------
           13         14      12.18          2 R07272022
28-JUL-23 12-AUG-22 R87462748           2         10 SYSTEM
11-DEC-22 SYSTEM                         11-DEC-22
           14         28     226.52          3 O03282022
29-MAR-23 15-AUG-22 O23547364           8         11 SYSTEM
11-DEC-22 SYSTEM                         11-DEC-22
           15         42       48.3          2 I08092022
10-AUG-23 16-AUG-22 I10837332           2         12 SYSTEM
11-DEC-22 SYSTEM                         11-DEC-22
           16         14      46.06          2 D04162022
17-APR-23 16-AUG-22 D34843734           2         12 SYSTEM
11-DEC-22 SYSTEM                         11-DEC-22

15 rows selected.


Rollback complete.


DRUG_CODE  DRUG_DESC                                                DRUG_TRADEMARK
DRUG_DOSE            DRUG_FORMULA
INGREDIENTS
DRUG_USES
DRUG_COST MANUFACTURER                                             CREATED_BY
DATE_CREA MODIFIED_BY                    DATE_MODI
---------- -------------------------------------------------- ---------------
-------------- -------------------- ----------------------------------------
---------- ----------------------------------------------------------------
-------------------------------- ----------------------------------------
```

---------- ---------- -------
---------- ---------- --
------- ---------- ---------
I10837332 Ibuprofen Advil
three times a day C13H18O2
Ibuprofen 200 mg
treat mild to moderate pain and arthritis
1.15 Pfizer SYSTEM
11-DEC-22 SYSTEM 11-DEC-22
O23547364 Olmesartan Benicar
once a day C24H26N6O3
Olmesartan 20 mg
treat high blood pressure (hypertension)
8.09 Lupin Limited SYSTEM
11-DEC-22 SYSTEM 11-DEC-22
D20847476 Dicyclomine Bentyl
four times a day C19H35NO2
Dicyclomine 20 mg
treat functional bowel or irritable bowel syndrome
.87 Aptalis Pharma Canada SYSTEM
11-DEC-22 SYSTEM 11-DEC-22
C76488378 Cetirizine Zyrtec
once a day C21H25ClN2O3
Cetirizine 10 mg
treat cold or allergy symptoms
.83 Johnson and Johnson SYSTEM
11-DEC-22 SYSTEM 11-DEC-22
D34843734 Dexamethasone Decadron
once a day C22H29FO5
Dexamethasone 10 mg
treat many different inflammatory conditions
3.29 Fera Pharmaceuticals SYSTEM
11-DEC-22 SYSTEM 11-DEC-22
H47637494 Hydrochlorothiazide Esidrix
once a day C7H8ClN3O4S2
Hydrochlorothiazide 50 mg
treat high blood pressure (hypertension)
2.49 Merck SYSTEM
11-DEC-22 SYSTEM 11-DEC-22
F68403835 Famotidine Pepcid
twice a day C8H15N7O2S3
Famotidine 20 mg
treat and prevent ulcers in the stomach and intestines
1.27 Johnson and Johnson SYSTEM
11-DEC-22 SYSTEM 11-DEC-22
C53274803 Clopidogrel Plavix
once a day C16H16ClNO2S.H2SO4
Clopidogrel 75 mg
treat stroke, blood clot, or serious heart problem
3.59 Bristol-Myers Squibb SYSTEM
11-DEC-22 SYSTEM 11-DEC-22
P39857223 Pseudoephedrine Sudafed
twice a day C10H15NO
Pseudoephedrine 120 mg

```
treat nasal and sinus congestion
1.31 Johnson and Johnson                                SYSTEM
11-DEC-22 SYSTEM                          11-DEC-22
R87462748  Ranitidine                                        Zantac
once a day          C13H22N4O3S
Ranitidine 150 mg
treat and prevent ulcers in the stomach and intestines
.87 GlaxoSmithKline                                     SYSTEM
11-DEC-22 SYSTEM                          11-DEC-22

10 rows selected.


1 row updated.


DRUG_CODE  DRUG_DESC                                            DRUG_TRADEMARK
DRUG_DOSE            DRUG_FORMULA
INGREDIENTS
DRUG_USES
DRUG_COST MANUFACTURER                                         CREATED_BY
DATE_CREA MODIFIED_BY                     DATE_MODI
---------- -------------------------------------------------- ---------------
-------------- -------------------- ----------------------------------------
---------- ----------------------------------------------------------------
-------------------------------- ----------------------------------------
------------------------------------------------------------ ---------- -------
---------------------------------------- ------------------------------ --
------- ------------------------------ ---------
I10837332  Ibuprofen                                            Advil
three times a day    C13H18O2
Ibuprofen 200 mg
treat mild to moderate pain and arthritis
1.55 Pfizer                                              SYSTEM
11-DEC-22 SYSTEM                          11-DEC-22
O23547364  Olmesartan                                         Benicar
once a day          C24H26N6O3
Olmesartan 20 mg
treat high blood pressure (hypertension)
8.09 Lupin Limited                                       SYSTEM
11-DEC-22 SYSTEM                          11-DEC-22
D20847476  Dicyclomine                                          Bentyl
four times a day     C19H35NO2
Dicyclomine 20 mg
treat functional bowel or irritable bowel syndrome
.87 Aptalis Pharma Canada                                SYSTEM
11-DEC-22 SYSTEM                          11-DEC-22
C76488378  Cetirizine                                          Zyrtec
once a day          C21H25ClN2O3
Cetirizine 10 mg
treat cold or allergy symptoms
.83 Johnson and Johnson                                  SYSTEM
11-DEC-22 SYSTEM                          11-DEC-22
```

```
D34843734  Dexamethasone                                    Decadron
once a day          C22H29FO5
Dexamethasone 10 mg
treat many different inflammatory conditions
3.29 Fera Pharmaceuticals                                SYSTEM
11-DEC-22 SYSTEM                        11-DEC-22
H47637494  Hydrochlorothiazide                             Esidrix
once a day          C7H8ClN3O4S2
Hydrochlorothiazide 50 mg
treat high blood pressure (hypertension)
2.49 Merck                                               SYSTEM
11-DEC-22 SYSTEM                        11-DEC-22
F68403835  Famotidine                                    Pepcid
twice a day         C8H15N7O2S3
Famotidine 20 mg
treat and prevent ulcers in the stomach and intestines
1.27 Johnson and Johnson                                 SYSTEM
11-DEC-22 SYSTEM                        11-DEC-22
C53274803  Clopidogrel                                   Plavix
once a day          C16H16ClNO2S.H2SO4
Clopidogrel 75 mg
treat stroke, blood clot, or serious heart problem
3.59 Bristol-Myers Squibb                                SYSTEM
11-DEC-22 SYSTEM                        11-DEC-22
P39857223  Pseudoephedrine                                Sudafed
twice a day         C10H15NO
Pseudoephedrine 120 mg
treat nasal and sinus congestion
1.31 Johnson and Johnson                                 SYSTEM
11-DEC-22 SYSTEM                        11-DEC-22
R87462748  Ranitidine                                    Zantac
once a day          C13H22N4O3S
Ranitidine 150 mg
treat and prevent ulcers in the stomach and intestines
.87 GlaxoSmithKline                                     SYSTEM
11-DEC-22 SYSTEM                        11-DEC-22

10 rows selected.


Rollback complete.


PRESSCRIPTION_ID PRESCRIPT DOCTOR_FIRST_NAME              DOCTOR_LAST_NAME
PATIENT_FIRST_NAME             PATIENT_LAST_NAME
---------------- --------- ------------------------------ --------------------
---------- ------------------------------ ------------------------------
               6 08-AUG-22 Fernand                        Hanks
Laetia                         Enison
               7 09-AUG-22 Rick                           Chow
Judith                         Olvsade
               8 10-AUG-22 Todd                           Smythe
Judy                           Sethi
```

```
              9 11-AUG-22 Irene                          Willig
Sean                          Pineda
             10 12-AUG-22 Rick                           Chow
Angela                        Torres
             11 15-AUG-22 Gary                           Pertez
Monica                        Waldman
             12 16-AUG-22 Nina                           Schorin
Catherine                     Mierzwa

7 rows selected.


DRUG_CODE  DRUG_DESC                                          The frequency
of prescribed drugs
---------- -------------------------------------------------- ---------------
-----------------
I10837332  Ibuprofen
.25
D34843734  Dexamethasone
.25
O23547364  Olmesartan
.167
R87462748  Ranitidine
.167
C76488378  Cetirizine
.083
H47637494  Hydrochlorothiazide
.083
D20847476  Dicyclomine
.083
F68403835  Famotidine
.083
C53274803  Clopidogrel
.083
P39857223  Pseudoephedrine
.083

10 rows selected.


Pharmacy Name                  Total Sale Prices
------------------------------ -----------------
Walgreens                                 369.18
Costco                                    226.52
Walmart                                   197.96


DOCTOR_FIRST_NAME              DOCTOR_LAST_NAME
DRUG_GENERIC_NAME                                DRUG_USES
------------------------------ ------------------------------ ---------------
---------------------------------- ----------------------------------------
------------------------------------------------------------
Fernand                        Hanks                          Dicyclomine
treat functional bowel or irritable bowel syndrome
```

```
Fernand                         Hanks                           Famotidine
treat and prevent ulcers in the stomach and intestines
Rick                            Chow                            Ranitidine
treat and prevent ulcers in the stomach and intestines
Rick                            Chow                            Ranitidine
treat and prevent ulcers in the stomach and intestines


PATIENT_FIRST_NAME              PATIENT_FIRST_NAME              The number of
prescriptions Prescription payment
------------------------------- ------------------------------- ---------------
------------ --------------------
Angel                           Moskowitz
2                143.5
Laetia                          Enison
1                35.56


Pharmaceutical manufacturer                             Drug generic name
The total prescribed quantity Total sale prices
------------------------------------------------- --------------------------
----------------------- ---------------------------- -----------------
Lupin Limited                                           Olmesartan
56            453.04


Pharmacy Name                   Doctor First Name               Doctor Last
Name                Patient First Name              Patient Last Name
------------------------------- ------------------------------- ---------------
--------------- ------------------------------ ------------------------------
Walgreens                       Fernand                         Hanks
Angel                           Moskowitz
Kroger                          Todd                            Smythe
Judy                            Sethi
CVS Health                      Nina                            Schorin
Catherine                       Mierzwa


DRUG_DESC                                           SUM(TOTAL_COST)
------------------------------------------------- ---------------
Pseudoephedrine                                             36.68
```

9. Database Administration and Monitoring

9.1 Roles and Responsibilities

In the real-world scenario, the electronic pharmacy management software system is managed by the IT company that creates this system or by the contracted IT company. The system administration team is responsible for the daily maintenance, operation, and debugging of the pharmacy management system. The database administration team is responsible for the daily maintenance and routine backups of the databases. The security administration team is responsible for the management of security and privacy of the pharmacy management software and associated databases.

9.2 System Information

As this project is to create a database that simulates the database used by the electronic pharmacy management system, the system information documented here is related to the database developed by the project. The system information for the real-world pharmacy management system and associated databases is beyond the scope of the project and won't be documented here. As the UMGC Virtual Lab Access Resources were used as the sole source and computational environment for the entire process of this project, the client-side and server-side hardware tools and systems are the same.

9.2.1 Database Management System Configuration

The Relational Database Management System (RDBMS) for the database is Oracle Database 18c Express Edition (Release 18.0.0.0.0 - Production Version 18.4.0.0.0). The RDBMS is run on Windows 11 Pro Version 21H2 (OS build 22000.493), the 64-bit operating system. The hardware system to run the OS system is Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz with 32 GB RAM. For DBMS installation and database configuration, minimum 100 GB free hard disk space is required for each server.

9.2.2 Database Support Software

Oracle SQL Developer software (Version 21.2.1.204.1703, Build 204.1703, 2005-2020) is used to connect the Oracle Database 18c Express Edition for access to the database. The transactions between end-users and the database are implemented by SQL queries. Microsoft Edge (Version 106.0.1370.42, 64-bit, 2022) is the web browser to connect the UMGC Virtual Lab Access Resources.

9.2.3 Security and Privacy

For the secure use of the database, a login system is required to check passwords provided by end-users and grant them access to the pharmacy management system after the provided passwords are authenticated by the login system. Through the pharmacy management software, the user can provide data to the database and query data stored in the database based on their privileges for specific entities (e.g., doctors can only have access to the information data related to themselves, their patients and their written prescriptions, but not other information).

9.3 Performance Monitoring and Database Efficiency

For the real-world scenario, both database and system administration teams should collaborate together to monitor the performance of the whole pharmacy management system, maintenance as well as efficiency of the database, the DBMS, and servers. The database administration team is mainly responsible for monitoring database performance, maintaining database consistency, improving database efficiency. The system administration team is mainly responsible for monitoring and maintaining the performance of the electronic pharmacy management software system and associated servers as well as supporting software systems. Both teams are responsible for monitoring and maintenance of the DBMS.

9.3.1 Operational Implications

The database management is critical for the operation of the pharmacy management system. In this project, the integrated application of both Oracle Database 18c Express Edition (Release 18.0.0.0.0 - Production Version 18.4.0.0.0) and Oracle SQL Developer software (Version 21.2.1.204.1703, Build 204.1703, 2005-2020) is used to manage the database to control the inputs and outputs of data.

9.3.2 Data Transfer Requirements

The electronic pharmacy management software system interacts with the database server through the internet. The data transfer between the pharmacy management software and the database server uses Transmission Control Protocol/Internet Protocol (TCP/IP).

9.3.3 Data Formats

End-users provide data to the database through the pharmacy management software system. End-users enter the data (e.g., patient data, prescription data) into the pharmacy management system and the system converts the data into SQL queries to update and modify the database. The prescription, pharmacy and drug data are stored as the raw binary data in the database.

9.4 Backup and Recovery

To recover the data from a variety of failures (e.g., media failure, user errors, hardware failures, natural disasters, etc.), backing up databases is the essential procedure. The differential backup of the database is performed daily (in the midnight) to secure newly updated data and the full database backup is performed once weekly. The full database backup needs to be tested for database recovery once monthly.

10. References

Brumm, B. (2022, October 6). *SQL Best Practices and Style Guide.* Database Star. Retrieved October 16, 2022, from https://www.databasestar.com/sql-best-practices/

Martynenko, M. (2020, September 22). *Pharmacy Management Software Development: Features, Steps, Costs.*Aimprosoft. Retrieved October 16, 2022, from https://www.aimprosoft.com/blog/blog-how-to-develop-a-pharmacy-management-software/

Maurer, R. (2015, July 30). *Top Database Security Threats and How to Mitigate Them.* SHRM. https://www.shrm.org/resourcesandtools/hr-topics/risk-management/pages/top-database-security-threats.aspx

Tamblyn, R., Huang, A., Kawasumi, Y., Bartlett, G., Grad, R., Jacques, A., Dawes, M., Abrahamowicz, M., Perreault, R., Taylor, L., Winslade, N., Poissant, L., & Pinsonneault, A. (2006). The Development and Evaluation of an Integrated Electronic Prescribing and Drug Management System for Primary Care. *Journal of the American Medical Informatics Association*, *13*(2), 148–159. https://doi.org/10.1197/jamia.m1887

Tuvikene, K. (2021, February 24). *10 Database Security Best Practices You Should Know.* Tripwire. Retrieved October 17, 2022, from https://www.tripwire.com/state-of-security/database-security-best-practices-you-should-know

Vejdani, M., Varmaghani, M., Meraji, M., Jamali, J., Hooshmand, E., & Vafaee-Najar, A. (2022). Electronic prescription system requirements: a scoping review. *BMC Medical Informatics and Decision Making*, *22*(1). https://doi.org/10.1186/s12911-022-01948-w