MEDITRACK

A Pharmacy Database Management System

Team 5

Dhiraj Lahoti - dhiraj25@umd.edu

Harshitha Ramachandra - harshi07@umd.edu

Sakshi Patil - spatil1@umd.edu

Under the guidance of Professors Pamela Duffy and Vedat Diker,

TA Sagar Chand

Team 5

Dhiraj Manoj Lahoti, Harshitha Ramachandra, Sakshi B.R Patil

CONTENTS

I. Introduction **1**

II. Database Design and Implementation **1**

- **A. Logical Design** **1**
- **B. Physical Database** **2**
- **C. Sample data and views/queries** **3**
- **D. Queries/views requirements** **4**

III. Changes from the Original Design **4**

IV. Issues Experienced during development **5**

V. Strengths and Limitations **6**

V. Lessons Learned **6**

VI. Potential Future Work **7**

I. Introduction

The pharmaceutical industry is a vital component of healthcare, responsible for the research, development, and production of drugs and medicines. However, managing the vast amount of data related to drug information, clinical trials, patient data, and manufacturing information poses a significant challenge for the industry. To address these challenges, the industry requires a pharma database management system to streamline data management, enhance data security, and ensure compliance with regulations.

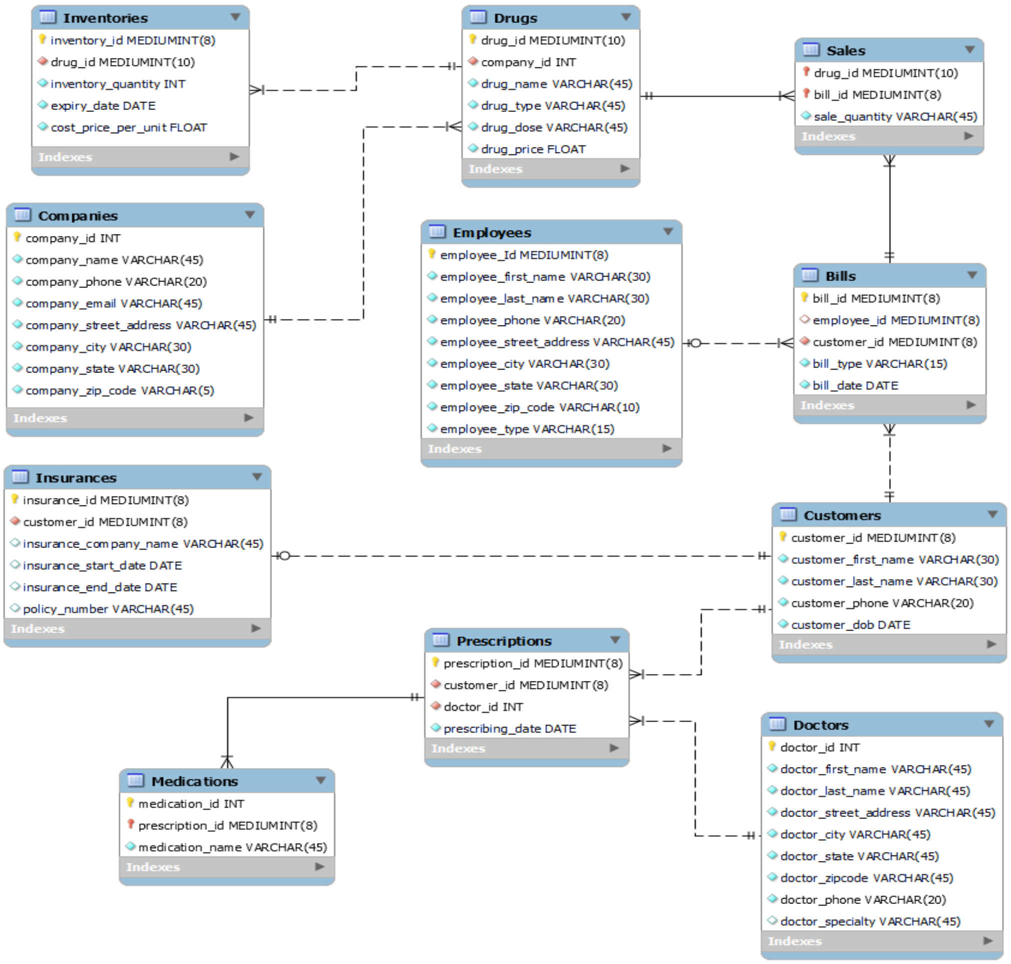
This project aims to develop a database management system called '**MediTrack**' that can store and retrieve information about drugs, patient prescriptions, and medication orders. MediTrack will serve as a central repository for all data related to the pharmacy, including customer information, inventory details, medication information, and sales transactions. The goal is to create a system that caters to the day-to-day requirements of the pharmacy, ensuring safety, accuracy, and efficiency and providing a competitive advantage to the pharmacy. This report outlines the process of designing and implementing the MediTrack database management system and its features, benefits, and potential applications in the pharmaceutical industry.

II. Database Design and Implementation

The design of MediTrack consists of several phases, including requirements analysis, conceptual design, logical design, and physical design. The requirements analysis phase involved identifying the needs of the pharmacy and defining the scope of the system. The conceptual design phase involved creating a high-level overview of the system, including its entities, attributes, and relationships.

A. Logical Design

The logical design phase involved creating a detailed data model, including the database schema and relationships between tables. The image below depicts the Entity Relationship Diagram (ERD) for Meditrack.



B. Physical Database

The database was created by forward engineering the entity relationship diagram on the MySQL workbench. The following are details regarding the tables within our database.

1. Companies: Information about medication manufacturers/distributors. Includes company_id (PK), name, phone, email and address.
2. Employees: Information about pharmacy staff. Includes employee_id (PK), first/last name, phone, address, and employee_type.
3. Customers: Information about pharmacy clients. Includes customer_id (PK), first/last name, phone, DOB.
4. Doctors: Information about doctors who prescribe medicine for customers. Includes doctor_id (PK), first/last name, phone, address, and specialty.

5. Insurances: Information about customer insurance policies. Includes insurance_id (PK), customer_id (FK), company name, start/end date, and policy number.
6. Prescriptions: Information about prescribed medications. Includes prescription_id (PK), customer_id (FK), doctor_id (FK), and prescribing date.
7. Medications: Table to normalize the database. Includes medication_id and prescription_id (composite PK) and medication name.
8. Bills: Information about customer bills. Includes bill_id (PK), employee_id (FK), customer_id (FK), bill type, and bill date.
9. Sales: Link between drugs and bills table. Includes drug_id and bill_id (composite PK), sales quantity.
10. Inventories: Information about medication inventory levels. Includes inventory_id (PK), drug_id (FK), quantity, expiry date, and cost price per unit.
11. Drugs: Information about individual drug products. Includes drug_id (PK), company_id (FK), name, type, dose, and selling price.

C. Sample data and views/queries

We tabulated our data on separate Excel sheets for each table. The companies tables consist of 30 records, the employee table has 15 records, there are 80 entries for the customer's table, we have 15 records in the doctor's table, the insurances table has 50 records, the prescriptions table has 50 records, medications has 200 records of the number of medicines, we have generated 50 bills for the month of April, The sales table consists of 105 records, the inventories table has data stored for drugs, and lastly we have information of 200 drugs in our drugs table. We have generated the data using an artificial intelligence tool. We have tried to keep the data as realistic and relevant as possible. We have ensured that our data is accurate and consistent with their assigned data types as well.

Query 1: Analyze drug consumption patterns to identify top-selling drugs for better inventory management and sales forecasting.

Query 2: Determine the most profitable drugs to maximize profitability and secure long-term contracts.

Query 3: Track sales data to identify top-performing customers for tailored marketing and inventory strategies.

Query 4: Properly manage inventory expiry to minimize losses and maintain customer trust.

Query 5: Identify best-performing employees based on customer service for rewarding high-performers and identifying areas for improvement.

Query 6: Identify doctors with high patient demand to improve patient outcomes and business growth.

Query 7: Track insurance expiry dates to manage customer relationships, billing, and improve customer satisfaction and retention.

D. Queries/views requirements

View name	Req A	Req B	Req C	Req D	Req E
Query 1_remaining_stock	Satisfied		Satisfied		
Query 2_profitability	Satisfied		Satisfied		Satisfied
Query 3_top_performing_customers	Satisfied		Satisfied	Satisfied	
Query 4_inventory_expiry	Satisfied	Satisfied			
Query 5_best_performing_employees	Satisfied	Satisfied	Satisfied		Satisfied
Query 6_doctor_consultancy	Satisfied	Satisfied	Satisfied	Satisfied	
Query 7_insuarance_expiry	Satisfied	Satisfied	Satisfied		

III. Changes from the Original Design

- Drugs table: We identified and removed unwanted attributes from the drugs table, as they were not relevant to the requirements.
- Normalization: We performed normalization to ensure data consistency and eliminate data redundancy. We removed the “drugs_have_inventories” associative entity as we considered only one pharmacy store, hence single inventory medication.

- Data types: We changed the data types of phone numbers and zip codes to ensure consistency and better data management.
- New table: We added a new table for doctors to show how doctors are relevant to the project. The table includes the doctor's name, specialization, and contact details.
- Cardinality: We changed the cardinality of some relationships to better represent the data.
- Employee attributes: We removed the employee password attribute from the employee table to ensure data privacy and security.
- Attribute name change: We changed the attribute name from drug_barcode to drug_id to better reflect its purpose.
- Companies table: We changed the primary key of the companies table from ‘company_name’ to ‘company_id’ to ensure data consistency and to avoid duplication of company names.
- Inventories table: To better manage inventory data, we added new attributes, including cost_price_per_unit and expiry_date.
- Queries: We modified all queries based on the new ERD to ensure that they reflect the updated data model and provide accurate results.

IV. Issues Experienced during development

Issues Encountered	Solutions considered	Solutions chosen
Generating unbiased and internally consistent data based on the ERD	- Explored all possible scenarios for each table and relation in the pharmacy database report's ERD to create a comprehensive dataset. - Searched for an existing dataset from various sources, including Kaggle, to save time and effort in data creation.	Created a realistic dataset considering all the requirements and potential scenarios.
Addressing the challenge of incomplete historical data and finding ways to work around it	Attempted to include data for more than one month but encountered problems, due to data dependencies or limitations in the database	Selected data for the month of April, as it was necessary to consider data dependencies on other table attributes

Managing the process of importing and inserting large datasets into the database to maintain data accuracy and integrity	Identified Excel as a potential tool for managing the dataset but also considered alternative solutions to ensure optimal data accuracy and consistency	Chose to use Excel to manage the dataset, as it was an appropriate tool due to its flexibility and ease of use, and it could be integrated with other data management systems.

V. <u>Strengths and Limitations</u>

- Our database can provide information on sales data for drugs which will be useful for analyzing consumption patterns and identifying the most selling drugs. This information can help the pharmacy make decisions for inventory management such as bulk purchasing and contract negotiations with drug-selling companies.
- The database can help estimate the profitability of drugs sold by the Pharmacy. This information can be useful for Sales Managers and Relationship Managers to enter into long-term contracts with companies whose drugs are sold in the highest numbers by the Pharmacy.
- One of the limitations of our database is that it only contains data for the month of April 2023, which could hinder our ability to analyze long-term trends or make informed predictions and decisions.
- Our database has an appropriate schema for in-store purchases, a lot of changes will have to be made if we try to integrate online sales.
- Our inventory table can manage inventory only for one store, it cannot take data for multiple stores, in order to incorporate that we will have to create a linking table between inventories and drugs.

V. <u>Lessons Learned</u>

One of the most significant learning experiences in our project was designing the entity relationship diagram. Since the rest of the project depended on it, being able to conceptualize our problem statement and translate it into an accurate and consistent diagram was crucial. We found it to be an excellent learning experience. Another valuable lesson we learned was that collecting and entering data is a time-consuming process. We challenged ourselves to think of more efficient ways to import data into the tables, which helped us explore alternative approaches. Finally, identifying key business requirements and questions that our schema and queries could address was a fascinating process. We learned how to create effective queries and business questions, which will be useful in future projects. Overall, this project provided us with many valuable learning opportunities.

VI. Potential Future Work

A. Extensions within the current technology and design (MySQL/MariaDB)

- Our current database has bill dates only for the month of April 2023. We would like to expand our dataset and include at least one year's worth of data entries.
- Currently we have assumed that our pharmacy is a stand-alone pharmacy (has no branches). We can modify our database for multiple branches having multiple inventories.
- Having learnt about new technologies in our course, we can implement a reporting system for easy access to important data and statistics.

B. Feasibility of alternative implementations (other relational considerations and/or the non-relational solution, such as NoSQL approaches)

When considering the implementation of a database for a pharmacy that will eventually be used across multiple branches, scalability and flexibility become critical factors to consider. In this scenario, a NoSQL database, such as MongoDB, can be a good alternative to a relational database. One of the primary benefits of using MongoDB is its flexibility. MongoDB does not require a predefined schema, which makes it easier to modify and adapt to changing data requirements. This is particularly important when dealing with large amounts of data generated from multiple sources, as it allows for more efficient data storage and retrieval. In addition, MongoDB's scalability is a key advantage for large-scale applications like pharmacy databases. MongoDB uses a distributed architecture that allows for horizontal scaling, meaning that additional nodes can be added to the database as needed to handle increased data volumes or traffic.