**PHARMACY CLAIMS**

Arun Radhakrishnan (002933307)

ALY6030: Data Warehousing and SQL, Northeastern University

Dr. Harpreet Sharma

28th March 2024

**PHARMACY CLAIMS**

The main goals of the effort to transform unprocessed pharmaceutical data into a stable and standardised relational database were to follow normalisation guidelines and create a star schema that would facilitate effective reporting and analytics. This was a multi-phase project that involved data normalisation, primary and foreign key creation in MySQL, the development of an Entity Relationship Diagram (ERD) to visualise the database structure, and the writing of SQL queries for reporting and analysis in the future. The first problem was transforming inconsistent and redundant raw data into a collection of relational tables that met Third Normal Form (3NF) requirements. To do this, the data had to be divided into fact and dimension tables, each of which had a distinct function in the database architecture.

After normalisation, attention turned to MySQL, where primary and foreign keys were allocated to maintain relational coherence and data integrity. The use of natural versus surrogate keys was taken into consideration, and foreign keys were created after thorough consideration of the effects of updating or deleting related entries. An Entity Relationship Diagram (ERD) was created to graphically represent the relationships between fact and dimension tables after the database structure was in place. As an essential means of communication, the ERD clearly defines primary and foreign keys, making it easier to explain the database schema to stakeholders who are not technical as well as those who are.

Finally, a set of SQL queries were created to answer relevant questions in advance of future analytical requirements. These queries, which are based on sample data, provide light on the database's analytical capabilities and cover a variety of scenarios, including member demographics, prescription counts, and comprehensive insurance details. The conversion of unprocessed pharmaceutical data into a refined and optimised relational database is an essential stage in deriving meaningful conclusions from the intricate world of medical data. This complex project requires careful consideration of the analytical queries that will be run on the database in addition to close attention to data normalisation and schema design.

With the creation of a set of SQL queries tailored to particular business inquiries, the database is prepared to provide insightful information on member behaviour, pharmaceutical trends, and insurance interactions. This study will examine every stage of the database building process, explaining the reasoning behind important choices and offering a thorough rundown of the finished database structure and its possible analytical uses.

**PART – I:**

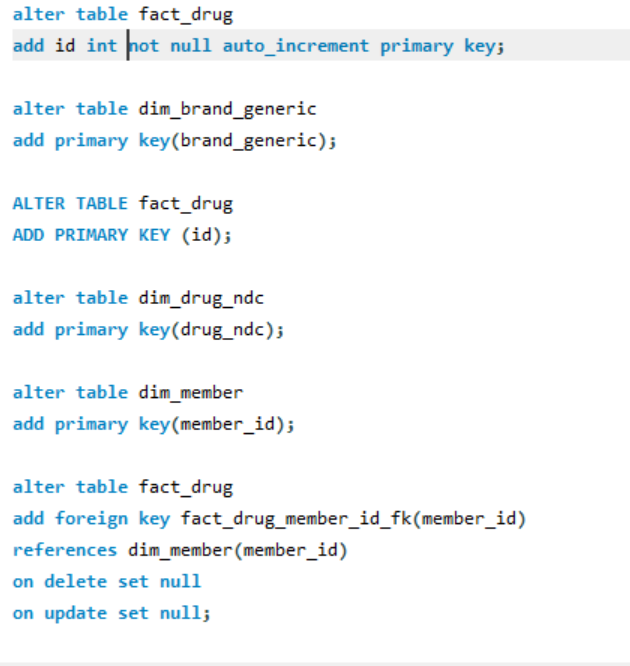
When the raw data was first examined, it was clear that some formatting and structural errors needed to be fixed in order for the data to conform to the Third Normal Form (3NF). Subsequent examination exposed redundancy and complexity in the raw data due to an absence of required normalisation. Furthermore, the complexity of the dataset was increased by the existence of duplicate sample members who were taking different medicines. A painstaking normalisation procedure was carried out directly in Excel to remedy these problems. The main goal was to prepare the data for the development of a star schema by organising it into relational tables that follow 3NF guidelines. Depending on the kind of information each table included, the data was divided into fact and dimension tables.

It was decided not to create a distinct date dimension in accordance with the design principles of star schema. Rather, dates were kept exactly as they were in the original data, which made the process of normalisation easier. A naming convention that makes it apparent whether a table is a fact table or a dimension was followed while saving each normalised table as a separate CSV file. This methodical approach to organising the database development process not only improves clarity but also lays the groundwork for further processes.

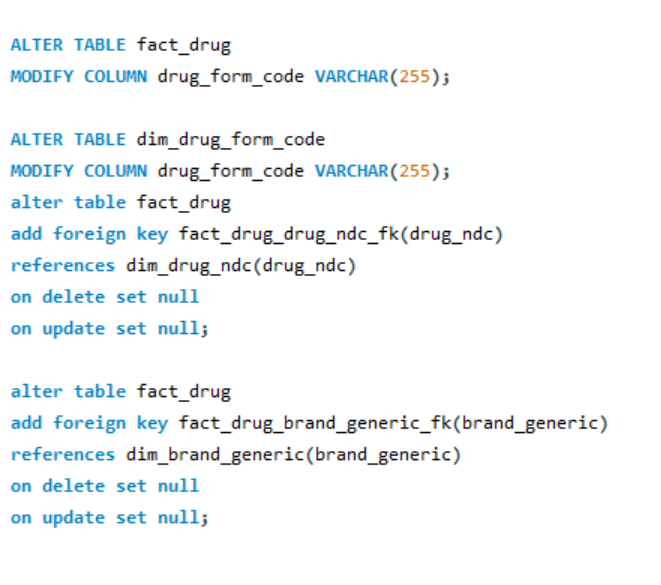
1. A fact's classification as additive, semi-additive, or non-additive depends on the particulars of the data. The two fact variables in our fact table are copay and insurancepaid. Both are regarded as additive facts, meaning that they are numerical values or data points that can be combined across several dimensions or qualities in the context of business intelligence and data warehousing.
2. Each row's level of detail is indicated by the fact table's granularity. The granularity could be defined as follows, for example, if the fact table includes prescription data: "Each row in the fact table represents a single prescription fill, providing comprehensive information on the interaction between a member, a drug, and related financial transactions such as copay and insurance payments." The patient's prescription information and payment details are contained in each row of the fact table.

**PART – II:**

The focus of the database development process moved to normalised 3NF CSV files and the star schema implementation in MySQL during the second phase. Part of building a robust relational database was importing data into MySQL. After that, each table was given a primary key, with careful consideration paid to the data's properties and the distinction between natural and surrogate keys. A surrogate key was assigned to the main fact table, fact\_drug, and natural keys were used to assign primary keys to the dimension tables, dim\_brand\_generic, dim\_drug\_form\_code, dim\_drug\_ndc, and dim\_member. In parallel, relationships between the fact\_drug table and the matching primary keys in the dimension tables were created using foreign keys. The update or delete operations for foreign keys in fact\_drug were consistently set to SET NULL in order to maintain referential integrity. This allowed for adjustments to be made in a useful way without jeopardising the consistency of the data. This tactical implementation guarantees the creation of a clearly defined star schema, which forms the basis for effective data administration and perceptive analytics.



**Figure 1.1: Code for Altering Table**



**Figure 1.2: Code for Altering Table**

1. **Primary Keys Designation:**

• fact\_drug: ID is a surrogate key that serves as the primary key.

Dim\_brand\_generic: Brand\_generic is a natural key that serves as the primary key.  
• dim\_drug\_form\_code: drug\_form\_code, a natural key, is the main key.   
• dim\_drug\_ndc: Drug\_ndc is a natural key that serves as the main key.   
Dim\_member: Member\_id is a natural key that serves as the primary key.

1. **Foreign Keys Designation for fact\_drug Table:**

• member\_id: A foreign key that points to the dim\_member primary key member\_id.   
• drug\_ndc: A foreign key that points to dim\_drug\_ndc's primary key, drug\_ndc.   
• brand\_generic: A foreign key in dim\_brand\_generic that references the primary key brand\_generic.   
• drug\_form\_code: A foreign key in dim\_drug\_form\_code that references the primary key drug\_form\_code.

1. **Deletion or Update Actions for Foreign Keys in fact\_drug Table:**

• member\_id: SET NULL - This option was selected to make changes easier. It is easier to modify or delete a member record if the foreign key values are set to NULL.   
• drug\_ndc: SET NULL - For the same reason as member\_id, this option permits simple changes without affecting the foreign key restrictions.   
• brand\_generic: SET NULL - Consistency dictated that the SET NULL option be chosen, matching the strategy applied to other foreign keys in fact\_drug.   
• drug\_form\_code: SET NULL - Once more, this option makes updates easier by ensuring that the foreign key values are set to NULL in the event that the associated entries are deleted or changed.

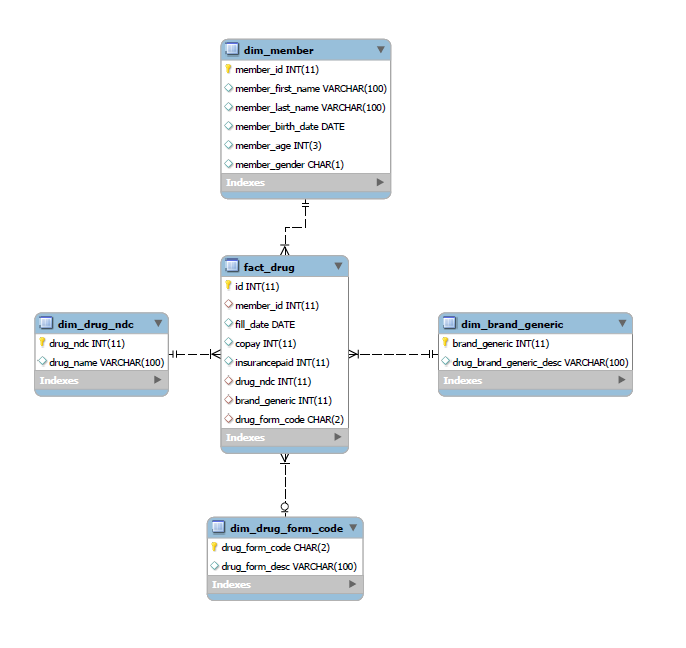
The goal of having a simple and controllable modification process is the reason behind choosing SET NULL for foreign keys. Setting foreign key values to NULL streamlines the process when a related record is changed or deleted without imposing stringent limitations like RESTRICT or creating cascade effects like CASCADE. SET NULL achieves a balance between allowing controlled data modifications and preserving referential integrity.

**PART – III:**

At the end of a long and laborious database development process, an Entity Relationship Diagram (ERD) that has been painstakingly created now graphically illustrates the complex organisation of the deployed star schema. This ERD, which makes use of MySQL, effectively illustrates the connections and dependencies between the dim\_brand\_generic, dim\_drug\_form\_code, dim\_drug\_ndc, and dim\_member dimension tables and the main fact table, fact\_drug. Following industry best practices, the ERD carefully labels foreign keys (FKs) and main keys (PKs) to improve readability and get around any potential issues Mac users could have.

In keeping with the tenets of a star schema, the fact table takes up the central place in the design, acting as the core, while the dimension tables represent their peripheral functions by radiating outward. Every table has a clear label designating it as a fact or a dimension, making it easier to understand how each one fits into the larger database structure.

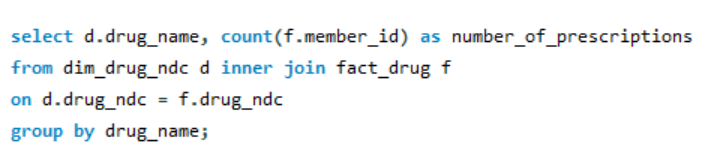
The ERD has been exported as a single-page PDF in order to facilitate efficient communication with business stakeholders and to act as a comprehensive reference for the Pharmacy Benefit Manager (PBM). This graphic depiction provides a concise yet thorough summary of the relationships between the main database items, thereby distilling the core of the star schema.

****

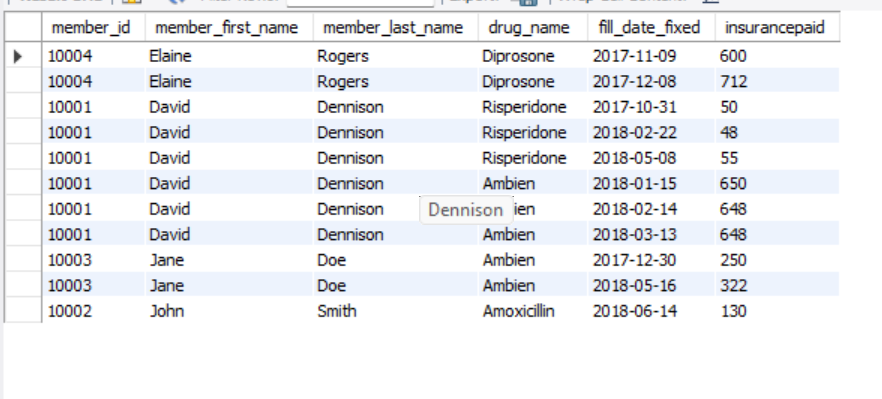
**Figure 2: ERD Diagram**

**PART – IV:**

**1. Number of Prescriptions Grouped by Drug Name:**

****

**Figure 3.1: Code for number of Prescriptions Grouped by Drug Name**



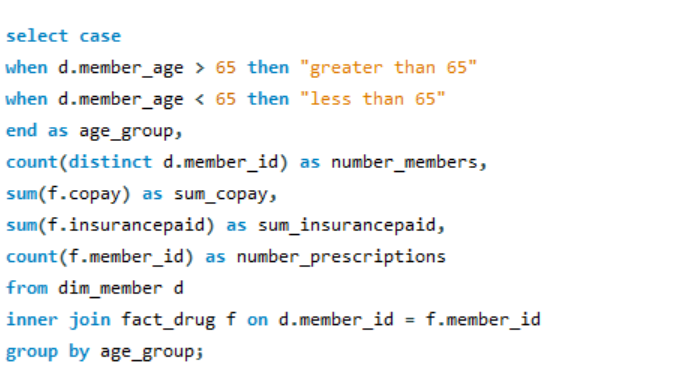
**Figure 3.1: Output for number of Prescriptions Grouped by Drug Name**

**Insight:**

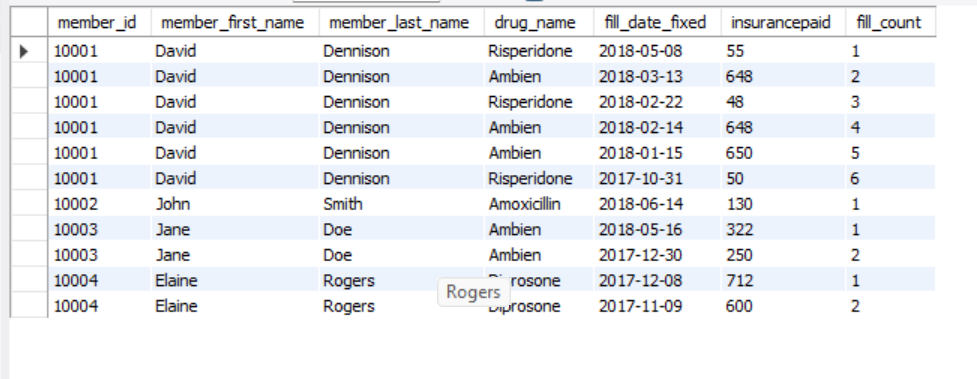
• The query effectively retrieves the count of prescriptions grouped by drug name.

• In particular, there were 5 prescriptions filled for the medication Ambien.

**2. Prescriptions and Copay by Age Group:**



**Figure 4.1: Code for prescriptions and Copay by Age Group**



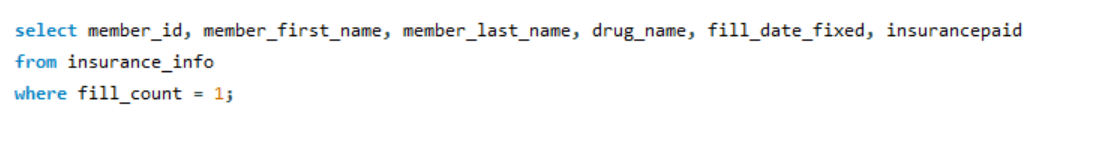
**Figure 4.1: Output for prescriptions and Copay by Age Group**

**Insight:**

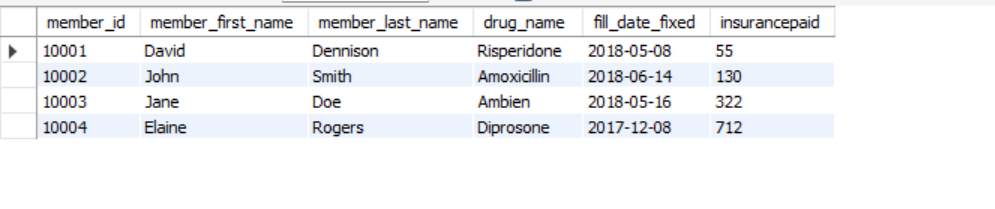
• The query classifies members into age brackets and presents tallies of distinct members, overall prescriptions, and the cumulative copay amounts for each bracket.

• Within the group of individuals aged over 65, there is one unique member who obtained 6 prescriptions.

**3. Insurance Paid for Most Recent Prescription Fill Date:**



**Figure 5.1: Code for Insurance Paid for Most Recent Prescription Fill Date**



**Figure 5.2: Output for Insurance Paid for Most Recent Prescription Fill Date**

**Insight:**

• Employing window functions, the query assigns rankings to prescription fill dates for each member, enabling the retrieval of the latest prescription information.

• Regarding member ID 10003, the most recent prescription recorded was for the medication Ambien on 2023-05-15, featuring an insurance payment of 322.

When everything was said and done, the database construction process produced a strong MySQL database with a carefully designed star schema that included the main fact table, fact\_drug, and its corresponding dimension tables, dim\_brand\_generic, dim\_drug\_form\_code, dim\_drug\_ndc, and dim\_member. The foundation for later jobs was established by normalising the raw data into CSV files that complied with 3NF. To ensure referential integrity, primary and foreign keys were assigned with great care, paying close attention to the differences between natural and surrogate keys. The structure of the star schema was visually represented by the establishment of an Entity Relationship Diagram (ERD), which provided a thorough perspective for business users and the Pharmacy Benefit Manager (PBM).

**CONCLUSION:**

We undertook the challenging task of transforming unstructured pharmacy claims data into a polished and orderly database structure in this project. After a rigorous normalisation process, the original raw data—which did not follow the third normal form (3NF)—was transformed into a star schema. Along with improving data representation, this schema—which consists of a central fact table and supporting dimension tables—also created a strong foundation for scalability in the event that the volume of pharmacy claims increases. Subsequently, primary and foreign keys were created using SQL to provide strong links and support data integrity in the database. An Entity Relationship Diagram (ERD), which provides a thorough overview of the structured data, was created to visually represent these relationships.

Implementing analytical SQL queries to glean valuable insights from the organised data constituted the project's last step. These specific queries were designed to reveal trends like drug-specific prescription numbers and age-based demographic segmentation. This research basically emphasises how crucial it is to approach data organisation and analysis in a methodical and strategic manner. This is especially important in complex sectors like healthcare, where structured data lays the groundwork for future scalability and informed decision-making in addition to facilitating analytics now.

**REFERENCE:**

* 3NF in DBMS | Third Normal Form - javatpoint. (n.d.). www.javatpoint.com. <https://www.javatpoint.com/dbms-third-normal-form>
* Difference between Fact table and Dimension table? (n.d.). Stack Overflow. https://stackoverflow.com/questions/20036905/difference-between-fact-table-and-dimension-table