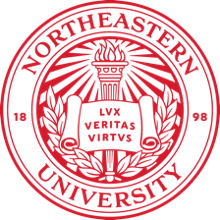
****

**Assignment**

**Final Project Assignment — Pharmacy Claims**

**Assignment by Dia Khosla**

**ALY6030: Data Warehousing & SQL**

**College of Professional Studies, Northeastern University**

**Instructor: Dr. Ghazal Tariri**

**Date: December 14, 2024**

### 

### 

### 

### 

### **Report on Normalization, Key Setup, ERD, and Analytics**

#### **Introduction**

In this report, I will walk through the process of normalizing the raw data, setting up the primary and foreign keys in MySQL, creating an Entity Relationship Diagram (ERD), and developing SQL queries to analyze the production data. The main goal is to ensure that the data is organized properly for a star schema and is ready for business analysis once the full production dataset is loaded.

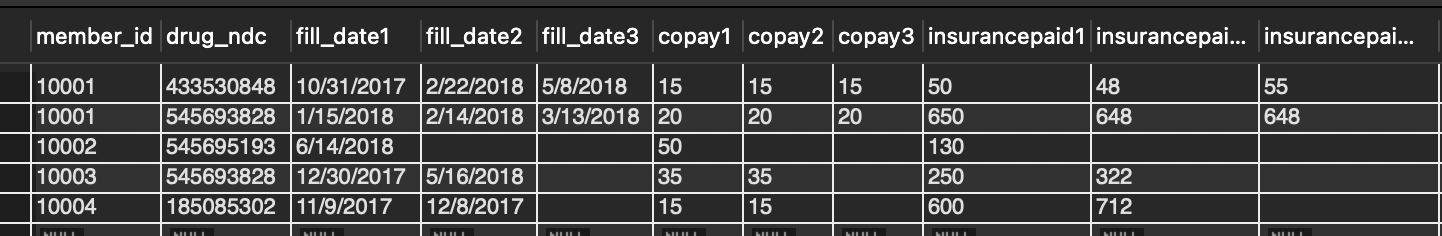
### **Part 1 – Normalization**

The first task was to convert the raw data into a set of relational tables that meet **Third Normal Form (3NF)** standards. This process involved identifying and separating out facts from dimension data and ensuring there were no redundant or partial dependencies.

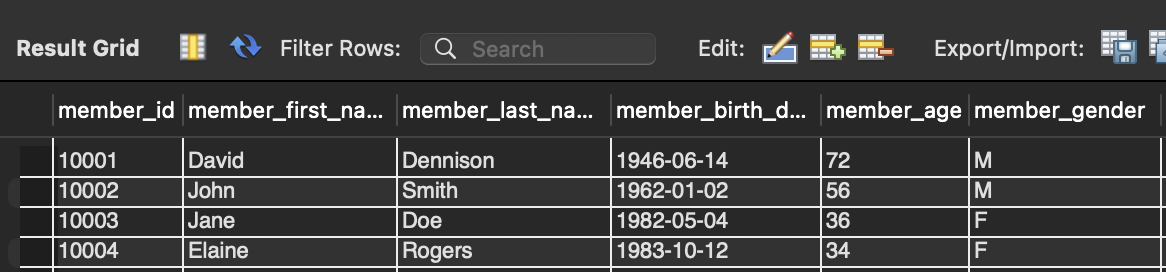
#### **Relational Tables**

I created the following tables:

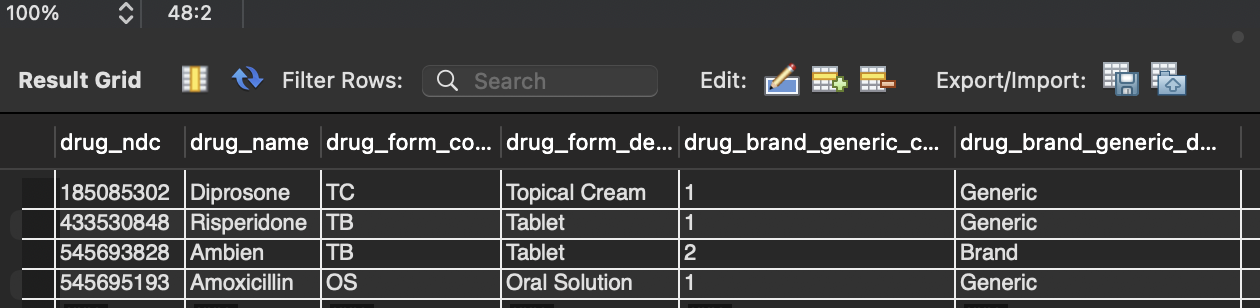
1. **Fact Table:**
   * **fact\_prescriptions.csv**: This table contains transactional data, specifically the prescriptions filled by members. Each row represents a prescription fill for a specific member with drug and financial data (copay, insurance payments, etc.).



1. **Dimension Tables:**
   * **dim\_member.csv**: This table contains information about the members, such as their member IDs, names, birth dates, and gender.



* + **dim\_drug.csv**: This table holds data on the drugs, including the drug NDC (National Drug Code), name, form, and brand/generic status.



Each dimension table is used to describe the facts in the **fact\_prescriptions** table.

#### **Fact Variables**

I categorized the fact variables in the **fact\_prescriptions** table as follows:

* **Additive**: Variables such as copay1, copay2, copay3, insurancepaid1, insurancepaid2, and insurancepaid3 are **additive** because they represent monetary values that can be summed.
* **Semi-Additive**: The variables fill\_date1, fill\_date2, and fill\_date3 are **semi-additive**, as they represent dates and cannot be summed but can be aggregated based on specific criteria.
* **Non-Additive**: The drug\_name is **non-additive** because it represents categorical data.

#### **Fact Table Grain**

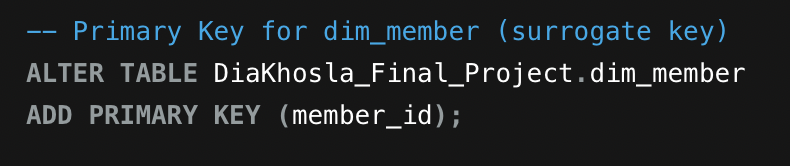
Each row in the **fact\_prescriptions** table represents a single prescription fill for a member, with associated financial data for up to three prescription fills.

### **Part 2 – Primary and Foreign Key Setup in MySQL**

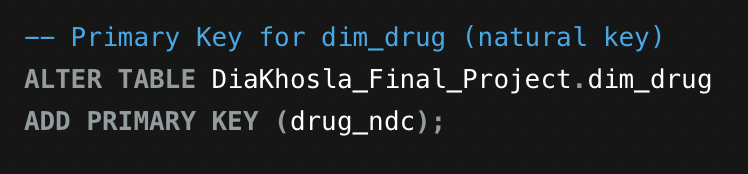
After normalizing the data and converting it into CSV files, the next step was to import the data into MySQL and set up the primary and foreign keys to create the star schema.

#### **Primary Keys (PKs):**

1. **dim\_member**: member\_id (Surrogate key)



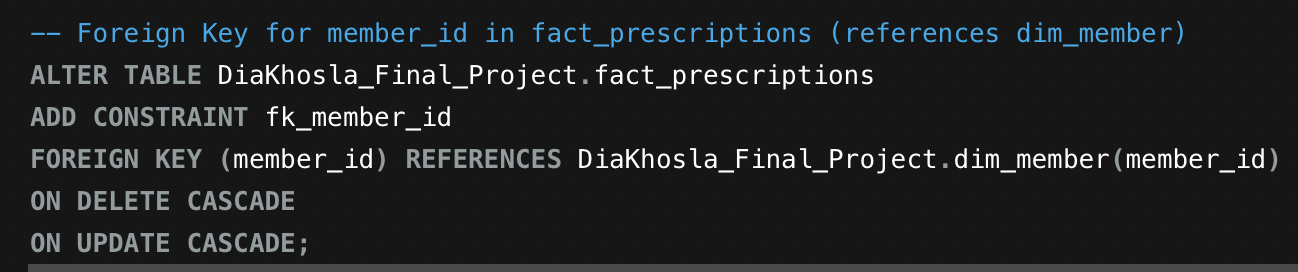
1. **dim\_drug**: drug\_ndc (Natural key)



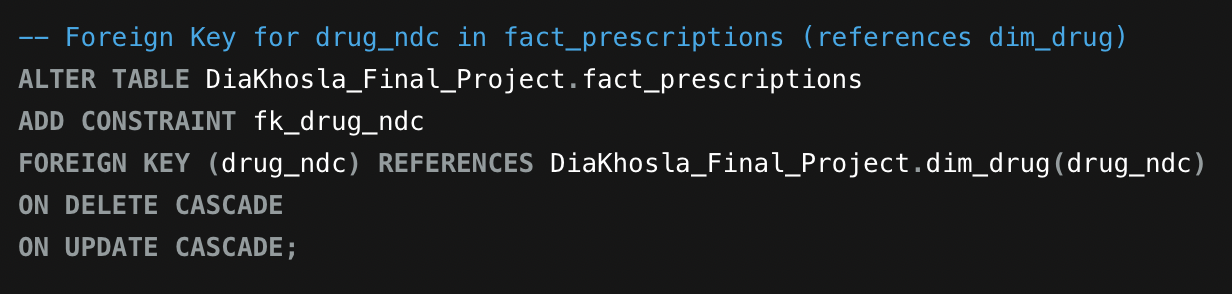
1. **fact\_prescriptions**: prescription\_id (Surrogate key, unique for each prescription fill)

#### **Foreign Keys (FKs):**

1. **fact\_prescriptions** references **dim\_member** on member\_id



1. **fact\_prescriptions** references **dim\_drug** on drug\_ndc



In MySQL, I specified how to handle updates and deletions of foreign keys:

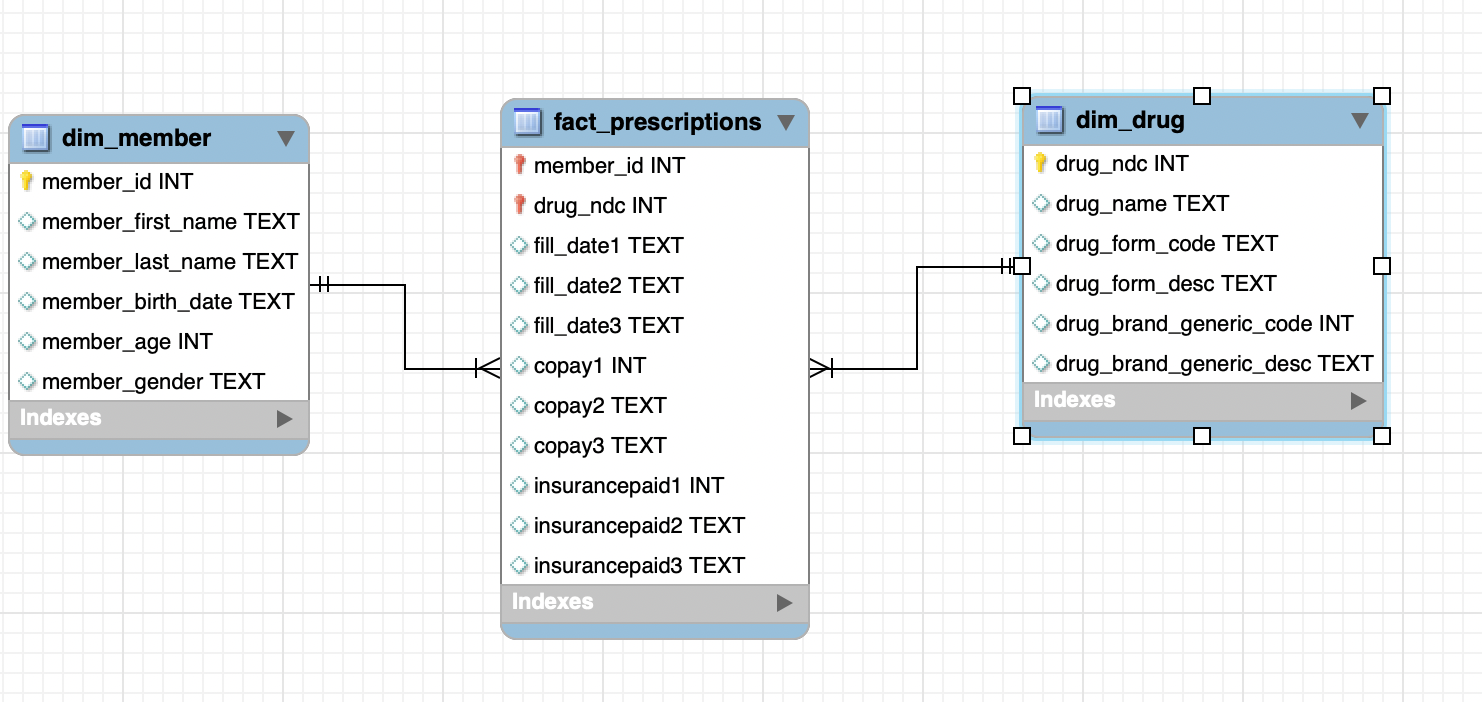
* **CASCADE**: This means that if a referenced record (such as a member) is deleted, the corresponding records in **fact\_prescriptions** will also be deleted.
* **SET NULL**: This option means that if a referenced record (such as a drug) is deleted, the foreign key column in **fact\_prescriptions** will be set to NULL.
* **RESTRICT**: This prevents the deletion of a referenced record if there are dependent records in **fact\_prescriptions**.

### 

### 

### **Part 3 – Entity Relationship Diagram (ERD)**

Once the primary and foreign keys were set up, I created an Entity Relationship Diagram (ERD) to visualize the relationships between the fact and dimension tables.

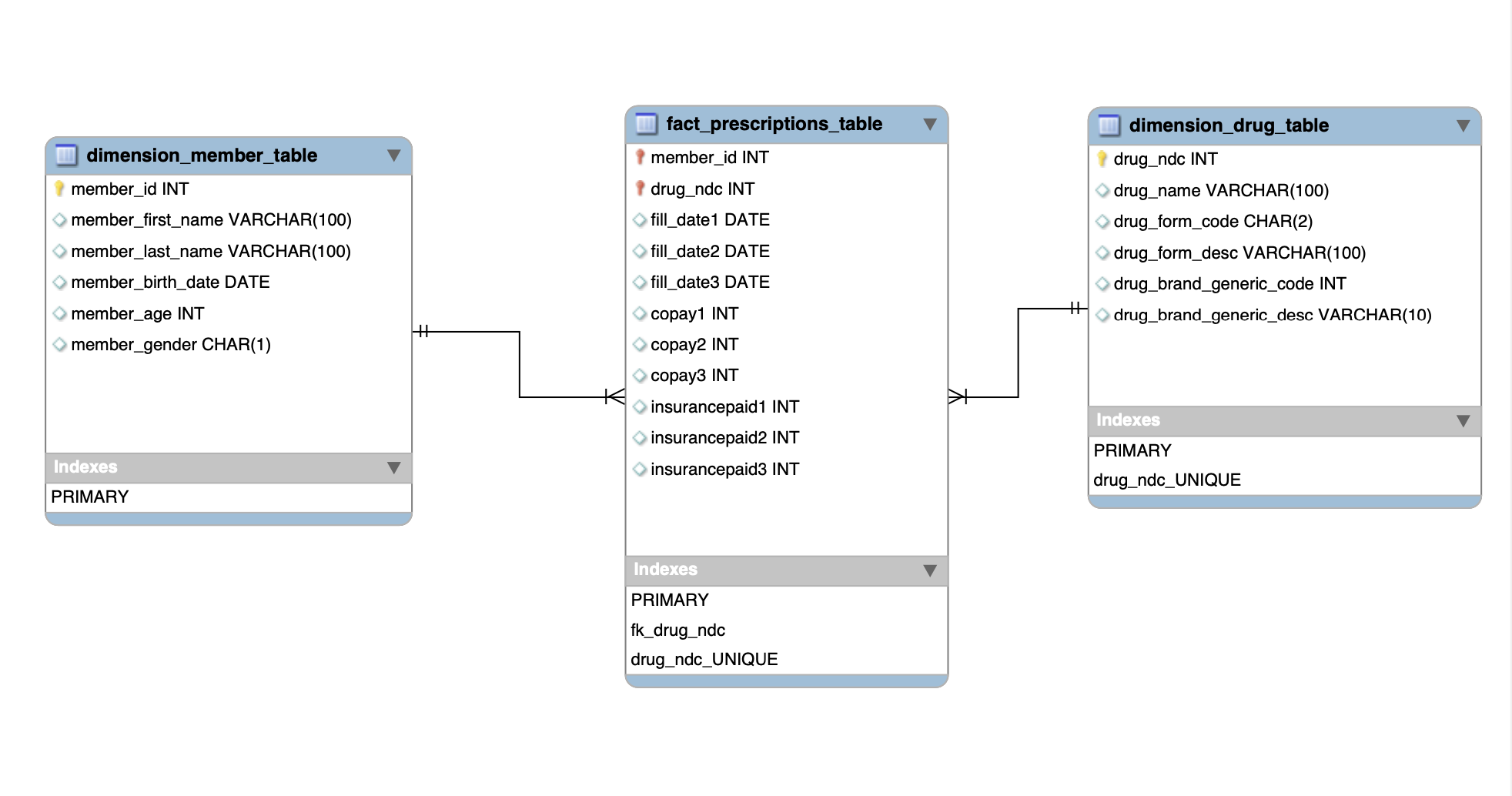


I have corrected the ERD diagram by identifying the appropriate variables and assigning suitable data types for each column. For example, integers are defined as INT, variable-length character strings as VARCHAR, and fixed-length character strings as CHAR. Text data types were removed where necessary to ensure compatibility within the database schema.

Additionally, table names have been updated to clearly identify fact and dimension tables. The changes include renaming:

* dim\_member to dimension\_member\_table,
* fact\_prescriptions to fact\_prescriptions\_table, and
* dim\_drug to dimension\_drug\_table.

All corrections and modifications have been summarized and presented in a one-page PDF format for ease of review and reference.



The ERD consists of:

* A **fact table** in the center.
* **Dimension tables** connected to the fact table by foreign key relationships.
* Each table was labeled with either **fact** or **dimension** as appropriate.
* Primary keys (PKs) and foreign keys (FKs) were clearly labeled to show how data is linked between tables.

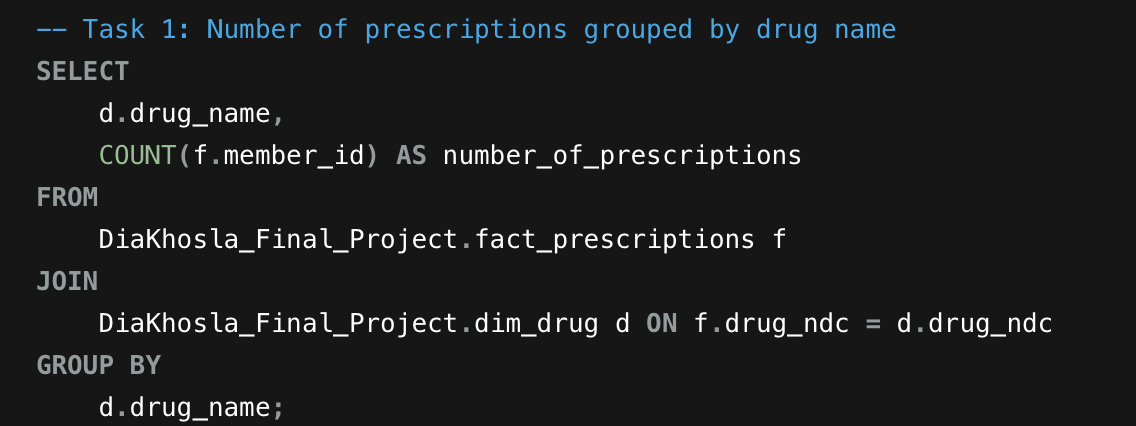
The diagram was created using MySQL Workbench, ensuring that all relationships were depicted correctly.

### **Part 4 – Analytics and Reporting SQL Queries**

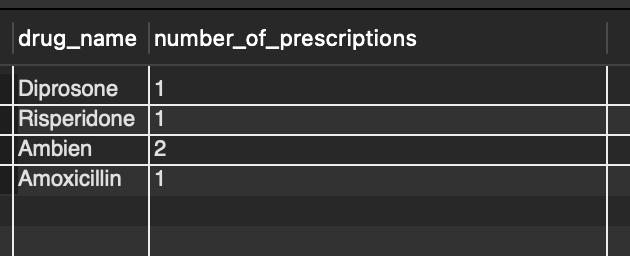
For the final task, I developed several SQL queries to analyze the production data. These queries will help business users assess the prescription data once it is fully loaded.

#### 

#### **Query 1: Number of prescriptions grouped by drug name**



**Results:**

****

* **How many prescriptions were filled for the drug Ambien?**
  + I query the result for drug\_name = 'Ambien' to get the total number of prescriptions filled for this drug which is 2.

#### 

#### 

#### 

#### 

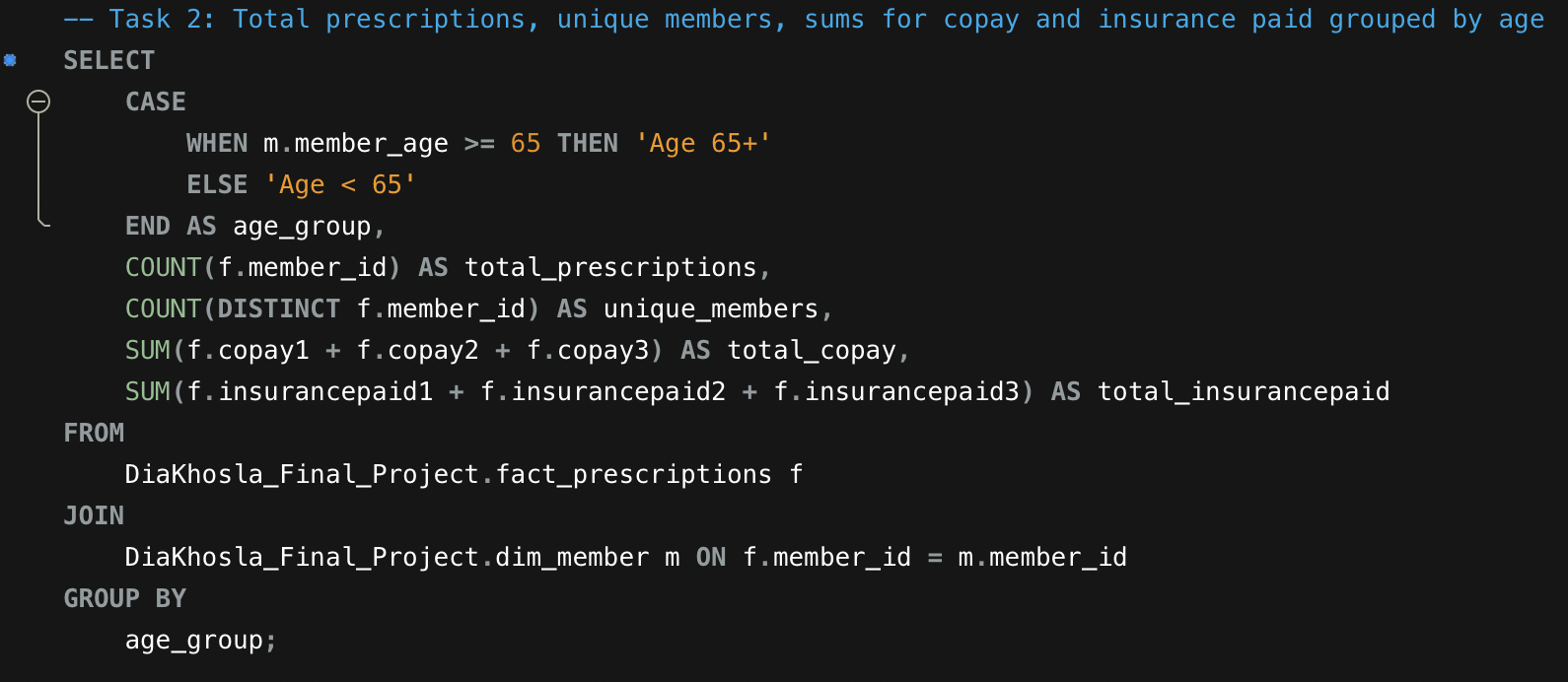
#### 

#### 

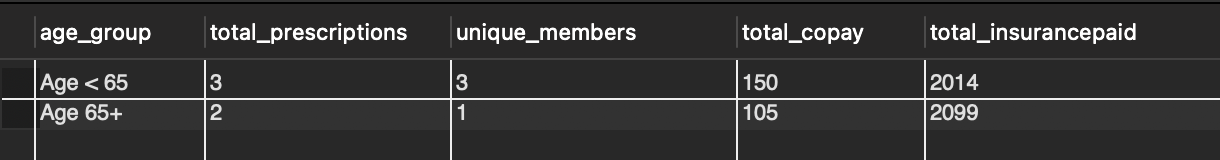
#### 

#### 

#### **Query 2: Count total prescriptions, unique members, and sum of copay for age groups**

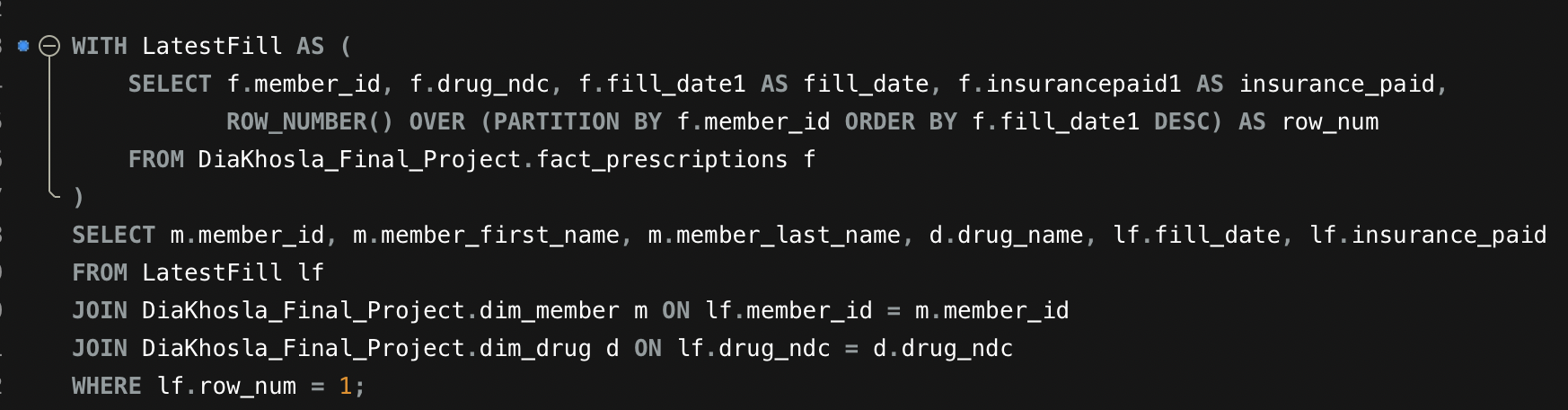


**Results:**

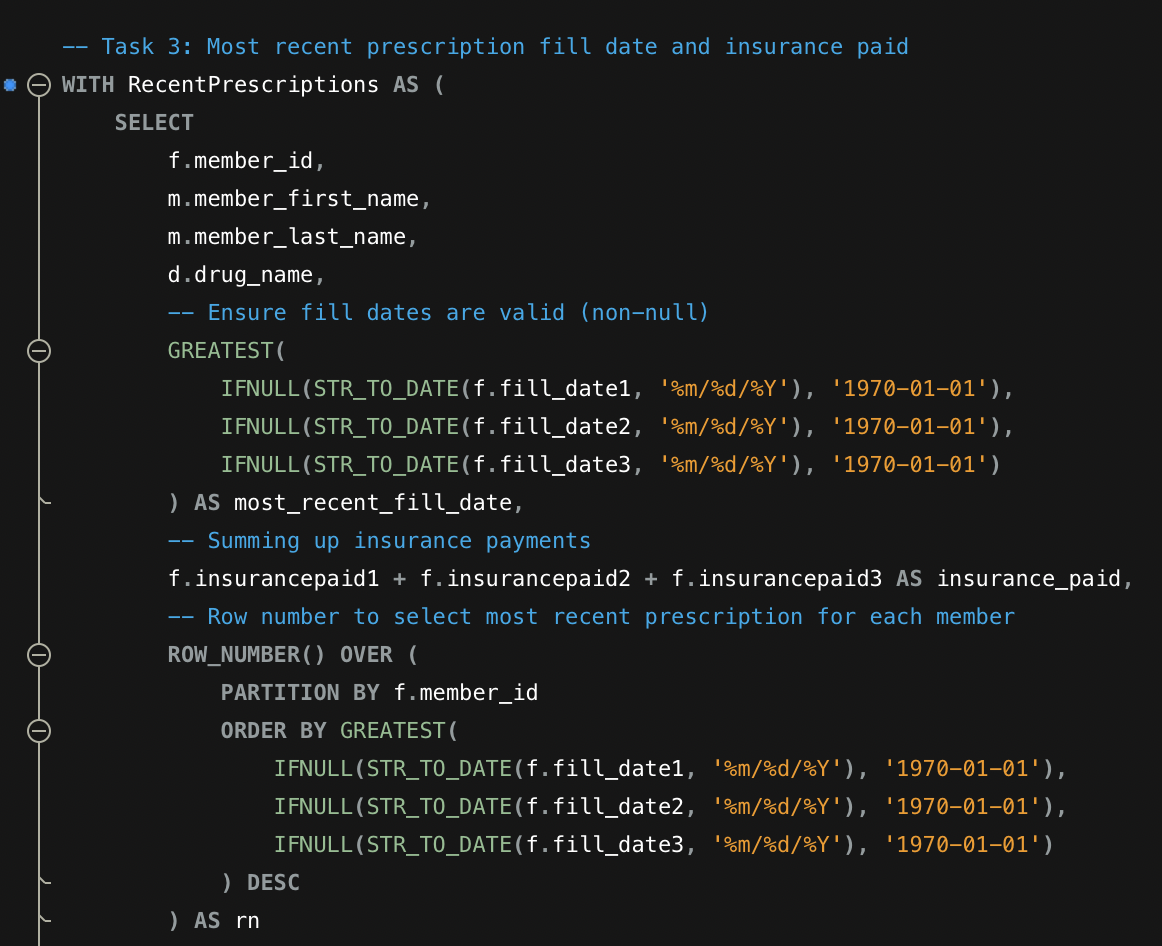
****

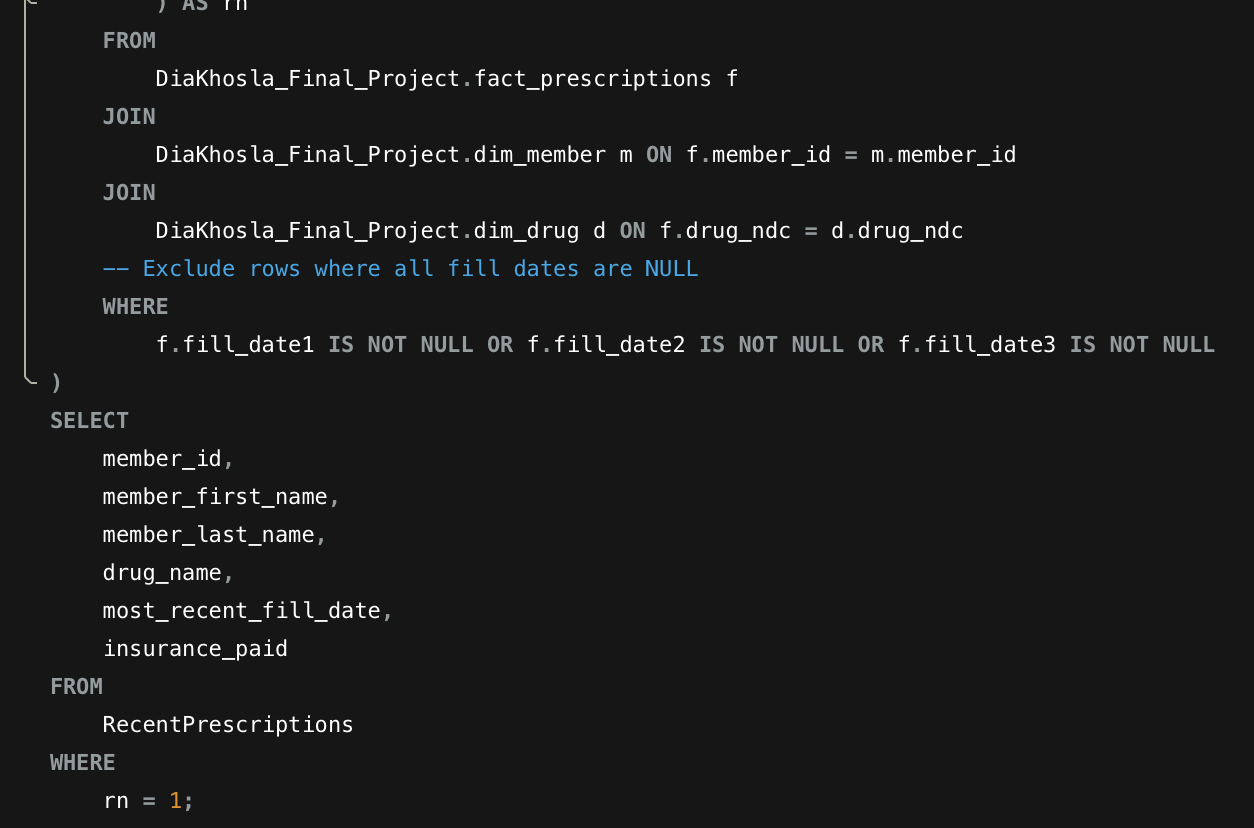
* **How many unique members are over 65 years of age?**
  + I checked the result where age\_group = '65+' and counted the unique\_members which turns out to be one only.
* **How many prescriptions did they fill?**
  + I reviewed the total\_prescriptions count for the age group '65+' and found that it is 2.

#### **Query 3: Most recent insurance payment**

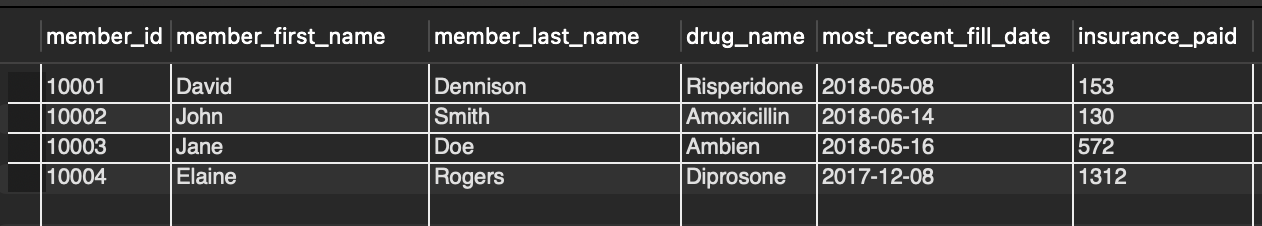


Or





**Results:**

****

* **For member ID 10003, what was the drug name listed on their most recent fill date?**
  + I queried for member\_id = 10003 to find the drug\_name is Ambien associated with their most recent prescription fill with a date of May 16, 2018.
* **How much did their insurance pay for that medication?**
  + I examined the insurance\_paid field for member\_id = 10003 to determine the insurance payment which amounts to a total 1312.

### **Conclusion**

In this report, I successfully demonstrated how to normalize raw data into relational tables that adhere to 3NF, set up primary and foreign keys in MySQL, and created an ERD to visualize the schema. Additionally, I developed SQL queries to provide business users with insights into prescription data, such as the number of prescriptions filled by drug name, prescription counts for different age groups, and the most recent insurance payments. This work sets the foundation for analyzing and reporting on the PBM's production data once it is expanded and fully implemented.

**References**:

1. Ramakrishnan, R., & Gehrke, J. (2003). *Database Management Systems* (3rd ed.). McGraw-Hill.  
   [Link to book](https://www.amazon.com/Database-Management-Systems-Raghu-Ramakrishnan/dp/0072465638)
2. MySQL Documentation. (n.d.). *MySQL Reference Manual*.  
   [Link to MySQL Docs](https://dev.mysql.com/doc/)
3. Elmasri, R., & Navathe, S. B. (2015). *Fundamentals of Database Systems* (7th ed.). Pearson.  
   [Link to book](https://www.amazon.com/Fundamentals-Database-Systems-Ramez-Elmasri/dp/0133970779)
4. Beaulieu, A. (2009). *Learning MySQL* (2nd ed.). O'Reilly Media.  
   Link to book
5. Koller, D., & Mongeau, M. (2019). *SQL Performance Explained*. Database Star.
6. Codd, E. F. (1990). *The Relational Model for Database Management*. Addison-Wesley.
7. Chen, P. P. (1976). *The Entity-Relationship Model: Toward a Unified View of Data*. ACM Transactions on Database Systems, 1(1), 9-36.
8. "What is 3NF (Third Normal Form)?". (2020). *Database Design Basics*.
9. *SQL Window Functions Tutorial* (n.d.). *Mode Analytics*.
10. "The Difference Between Foreign Keys and Primary Keys". (2021). *Learn SQL Online*.
11. Northeastern University. (2024). *Course module title*.<https://northeastern.instructure.com/courses/196426/modules>