

**Final Project Assignment — Pharmacy Claims**

ALY6030 Data Warehousing and SQL

**CRN:** 20361

Likhitha Varakala

**Email:** [**varakala.l@northeastern.edu**](mailto:varakala.l@northeastern.edu)

March 31, 2024

**Part 1 – Normalization**

**Create Dimension Tables:**

**Member Dimension Table:** Includes unique member information. Columns might include member\_id, member\_first\_name, member\_last\_name, member\_birth\_date, member\_age, member\_gender. Ensure each member\_id is unique.

**Drug Dimension Table:** Contains drug details. Columns might include drug\_ndc, drug\_name, drug\_form\_code, drug\_form\_desc, drug\_brand\_generic\_desc. Each drug\_ndc should be unique.

**Create Fact Table:**

**Transaction Fact Table:** This table captures the events of medication being filled. It includes member\_id, drug\_ndc, fill\_date, copay, and insurance\_paid. You'll create a row for each event where a medication is filled. To adhere to 3NF:

* Ensure there's no redundant information or unnecessary duplication of information.
* The member\_id and drug\_ndc in this table should exist in the Member and Drug dimension tables, respectively.

**Unpivot Fill Dates and Corresponding Values:**

If data includes multiple columns for fill dates and corresponding copay and insurance values (like fill\_date1, fill\_date2, copay1, copay2, etc.), you need to unpivot these columns so that each row represents a single transaction event. Each transaction event would include a fill date, the copay amount for that fill, and the insurance payment.

**Save as CSV:**

Save each table as a separate CSV file, naming them to indicate whether they are a fact or dimension table (e.g., dim\_member.csv, dim\_drug.csv, fact\_transaction.csv).

**For the provided fact variables in your fact table:**

**Copay:** This is an Additive fact. You can sum up the copay amounts across various dimensions like date, member, and drug without any loss of meaning or accuracy.

**Insurance Paid:** This is also an Additive fact for the same reasons as copay. It can be summed across different transactions, members, or drugs to get totals.

In the context of the grain of your fact table:

**Grain Description:** Each row in the fact table represents a single instance of a medication dispensing event, capturing the specific date a drug was filled, along with the copay amount and insurance payment associated with that particular dispensing event.

**Part 2 - Primary and Foreign Key Setup in MySQL**

**Member Table:** The member\_id column was chosen as the primary key, acting as a natural key. This decision was based on the unique identification of each member, which is intrinsic to the dataset.

**Drug Table:** Similarly, the drug\_ndc column served as the primary key, identified as a natural key due to the unique National Drug Code assigned to each drug.

**Fact Table**: For the fact table, a surrogate key named prescription\_id was introduced. This key does not hold any business meaning but provides a unique identifier for each transaction record.

**Foreign Key Designation**

Foreign keys were established to maintain referential integrity within the star schema:

**Fact Table to Member Table:** The member\_id in the fact table references the member\_id in the member table.

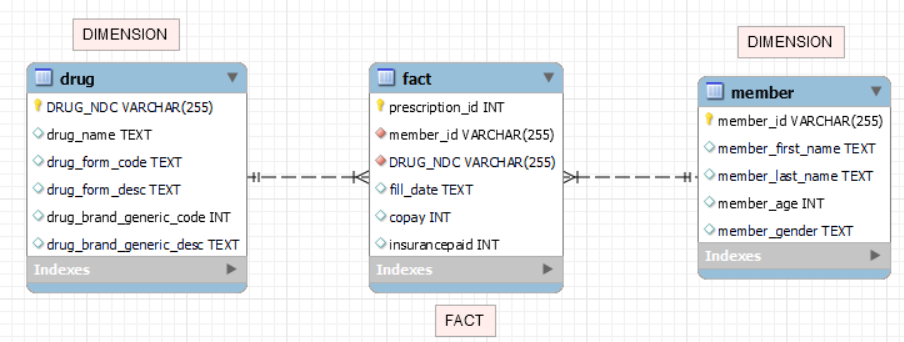
**Fact Table to Drug Table:** The drug\_ndc in the fact table references the drug\_ndc in the drug table.

**Foreign Key Actions**

**On Delete Cascade:** This option was chosen for both FKs to ensure that any deletion of a member or drug would automatically remove associated transaction records in the fact table, maintaining the cleanliness and relevance of the dataset.

**On Update Cascade:** Similarly, updating a member\_id or drug\_ndc would reflect the changes in the fact table, preserving data integrity and consistency across the schema.

**Part 3 — Entity Relationship Diagram (ERD)**

****

**Part 4 — Analytics and Reporting**

1. Write a SQL query that identifies the number of prescriptions grouped by drug name. Paste your output to this query in the space below here; your code should be included in your .sql file.

Also answer this question: How many prescriptions were filled for the drug Ambien?

**A screenshot of a computer

Description automatically generated**

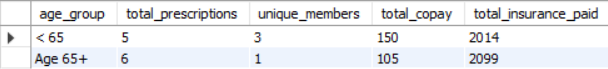
Based on the output, the drug "Ambien" has been filled 5 times**.**

1. Write a SQL query that counts total prescriptions, counts unique (i.e. distinct) members, sums copay, and suminsurancepaid for members grouped as either ‘age 65+’ or ’ < 65’. Use case statement logic to develop this query like lecture 3. Paste your output in the space below here; your code should be included in your .sql file.

Also answer these questions:

How many unique members are over 65 years of age?

How many prescriptions did they fill?

****

Based on the output, here is 1 unique member over 65 years of age, and they have filled a total of 6 prescriptions. The total copay for members under 65 is 150, and the total insurance paid for this group is 2014. For members 65 and older, the total copay is 105, and the total insurance paid is 2099.

1. Write a SQL query that identifies the amount paid by the insurance for the most recent prescription fill date. Use the format that we learned with SQL Window functions. Your output should be a table with member\_id, member\_first\_name, member\_last\_name, drug\_name, fill\_date (most recent), and most recent insurance paid. Paste your output in the space below here; your code should be included in your .sql file.

Also answer these questions:

For member ID 10003, what was the drug name listed on their most recent fill date?

How much did their insurance pay for that medication?

**A screenshot of a computer

Description automatically generated**

Based on the output, for member ID 10003, the drug name listed on their most recent fill date is "Ambien," and the insurance paid 322 for that medication.

**References**

GfG. (2023e, October 31). *SQL Join (Inner, left, right and Full Joins)*. GeeksforGeeks. <https://www.geeksforgeeks.org/sql-join-set-1-inner-left-right-and-full-joins/>

*SQL Window Functions | Advanced SQL - Mode*. (2016, May 23). Mode Resources. <https://mode.com/sql-tutorial/sql-window-functions>

Simplilearn. (2023b, June 6). *Fact Table vs. Dimension Table - Differences Between The Two*. Simplilearn.com. <https://www.simplilearn.com/fact-table-vs-dimension-table-article#:~:text=Fact%20tables%20contain%20numerical%20data,data%20analysis%20and%20decision%2Dmaking>.

Custer, C. (2023b, May 4). What is a foreign key? (with SQL examples). *Cockroach Labs*. <https://www.cockroachlabs.com/blog/what-is-a-foreign-key/#:~:text=Primary%20keys%20serve%20as%20unique,cross%2Dreferencing%20the%20two%20tables>.