



Northeastern University

Aly6030: Data Warehousing and SQL.

By: Rohit Raman

To: Prof. Shahram Sattar

Topic – Final Project.

Date: 26/10/2023

Introduction and learning outcomes from this project.

Summary of the Project:

The project aims to normalize raw data into relational tables adhering to the Third Normal Form (3NF) standards. It involves identifying entities (Members, Drugs, Drug Fill Information) and organizing data into dimension tables (Members and Drugs) and a fact table (Drug Fill Fact). The fact table captures detailed information about drug fill events. Primary and foreign keys are designated for each table to establish relationships between them, ensuring data integrity. Actions on deletion and update are specified for foreign keys to maintain referential integrity. Finally, an entity-relationship diagram (ERD) is constructed, illustrating the relationships between dimension and fact tables.

Skills Acquired:

- Understanding of database normalization principles, particularly the Third Normal Form (3NF).
- Identification of entities and relationships within a dataset.
- Designing dimension and fact tables to organize data effectively.
- Assigning primary and foreign keys to establish relationships between tables.
- Ensuring data integrity through proper constraint management, such as CASCADE and SET NULL actions.
- Construction and interpretation of entity-relationship diagrams (ERDs) to visualize database structures.
- Practical application of SQL queries for data retrieval, analysis, and reporting within a healthcare-related database context.

Aly 6030: Final Project.

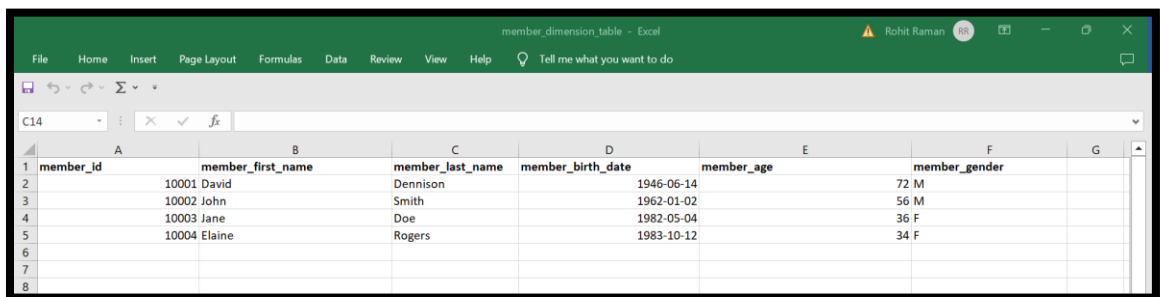
- Importance of maintaining data integrity and structured relationships for effective data organization and analysis across domains.

Part 1 Normalizations:

To normalize the provided raw data into a set of relational tables that meet 3NF standards, I need to identify the entities and their relationships. Based on the data provided, it seems like there are three main entities: Members, Drugs, and Drug Fill Information. Here's how we can organize the data into fact and dimension tables:

1. Members Dimension Table:

- This table contains information about members.
- Attributes: member_id (Primary Key), member_first_name, member_last_name, member_birth_date, member_age, and member_gender.



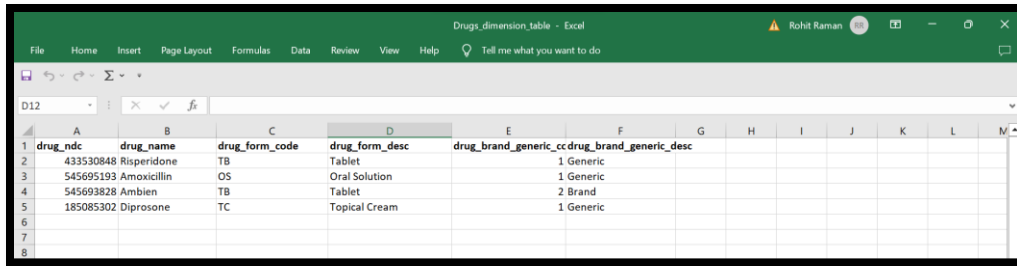
	A	B	C	D	E	F	G
	member_id	member_first_name	member_last_name	member_birth_date	member_age	member_gender	
1							
2	10001	David	Dennison	1946-06-14	72	M	
3	10002	John	Smith	1962-01-02	56	M	
4	10003	Jane	Doe	1982-05-04	36	F	
5	10004	Elaine	Rogers	1983-10-12	34	F	
6							
7							
8							

Figure 1: Member dimension table.

2. Drugs Dimension Table:

- This table contains information about drugs.
- Attributes: drug_ndc (Primary Key), drug_name, drug_form_code, drug_form_desc, drug_brand_generic_code, and drug_brand_generic_desc.

Aly 6030: Final Project.



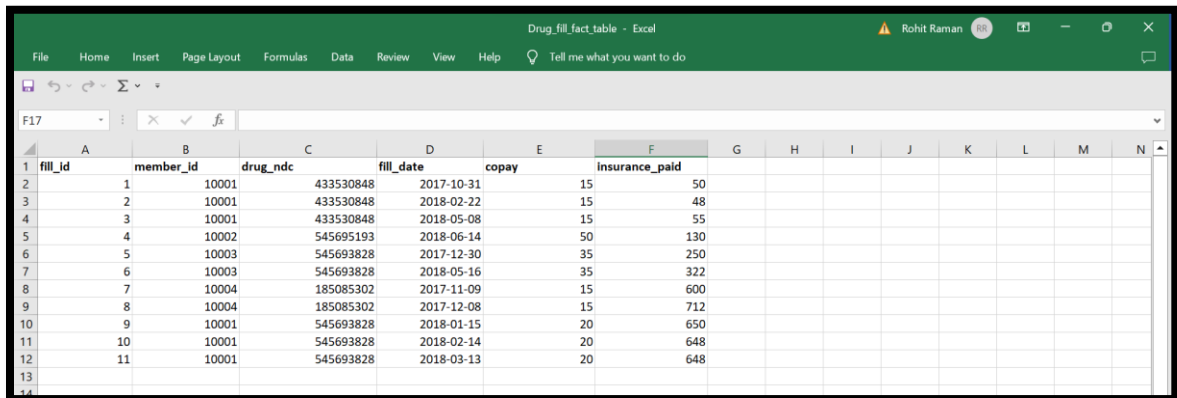
1	A	B	C	D	E	F	G	H	I	J	K	L	M
	drug_ndc	drug_name	drug_form_code	drug_form_desc	drug_brand_generic_cc	drug_brand_generic_desc							
2	433530848	Risperidone	TB	Tablet		1 Generic							
3	545695193	Amoxicillin	OS	Oral Solution		1 Generic							
4	545693828	Ambien	TB	Tablet		2 Brand							
5	185085302	Diprosone	TC	Topical Cream		1 Generic							
6													
7													
8													

Figure 2: Drug dimension table.

3. Drug Fill Fact Table:

This table contains information about drug fills.

Attributes: fill_id (Primary Key), member_id (Foreign Key), drug_ndc (Foreign Key), fill_date, copay, and insurance_paid.



1	A	B	C	D	E	F	G	H	I	J	K	L	M	N
	fill_id	member_id	drug_ndc	fill_date	copay	insurance_paid								
2	1	10001	433530848	2017-10-31	15	50								
3	2	10001	433530848	2018-02-22	15	48								
4	3	10001	433530848	2018-05-08	15	55								
5	4	10002	545695193	2018-06-14	50	130								
6	5	10003	545693828	2017-12-30	35	250								
7	6	10003	545693828	2018-05-16	35	322								
8	7	10004	185085302	2017-11-09	15	600								
9	8	10004	185085302	2017-12-08	15	712								
10	9	10001	545693828	2018-01-15	20	650								
11	10	10001	545693828	2018-02-14	20	648								
12	11	10001	545693828	2018-03-13	20	648								
13														
14														

Figure 3: Drug fill fact table.

Observations:

From figure 1, 2, 3 referring to 3NF-compliant schema, we have separated the data into dimension tables (Members and Drugs) and a fact table (Drug Fill Fact) to eliminate data redundancy and improve data integrity. The fact table uses foreign keys to reference the dimension tables. The dates are included as they appear in the data in the Drug Fill Fact table.

Aly 6030: Final Project.

Question 1:

The type of fact for each variable in the fact table are:

fill_id: This is a **non-additive** fact because each fill_id represents a unique identifier for a specific drug fill event, and aggregating or summing them does not provide meaningful information.

member_id: This is a **non-additive** fact as well, for the same reasons as fill_id.

drug_ndc: Again, this is a **non-additive** fact, as it also represents unique drug identifiers for specific drug fill events.

fill_date: This is also a **non-additive** fact because it represents the date of each drug fill event, and aggregating dates does not provide meaningful information.

The "**copay**" and "**insurance_paid**" are both **additive facts**. Here's the explanation for each:

Copay: This is an **additive fact**. Copay represents the amount that a member pays for a specific drug fill event. We can sum or aggregate copay amounts across dimensions (e.g., by member or drug) to calculate total copay expenses, and it remains meaningful.

Insurance Paid: Insurance paid is also an **additive fact**. It represents the amount that an insurance company pays for a specific drug fill event. We can aggregate insurance payments across dimensions to calculate total insurance payments for different groups or categories, and it maintains its meaning.

In summary:

Additive Fact: Copay and Insurance Paid

Non-additive Fact: Fill ID, Member ID, Drug NDC, Fill Date

Aly 6030: Final Project.

Question 2:

The grain of the fact table can be described as follows:

Grain: Each fact row in the Drug Fill Fact table represents a specific drug fill event, uniquely identified by fill_id. It captures detailed information about the individual drug fill events, including the member, drug, fill date, copay, and insurance payment for that particular event.

Part 2 — Primary and Foreign Key Setup in MySQL

The primary keys (PK) and foreign keys (FK) for each table are designated as follows:

Question 1:

1. Members Dimension Table:

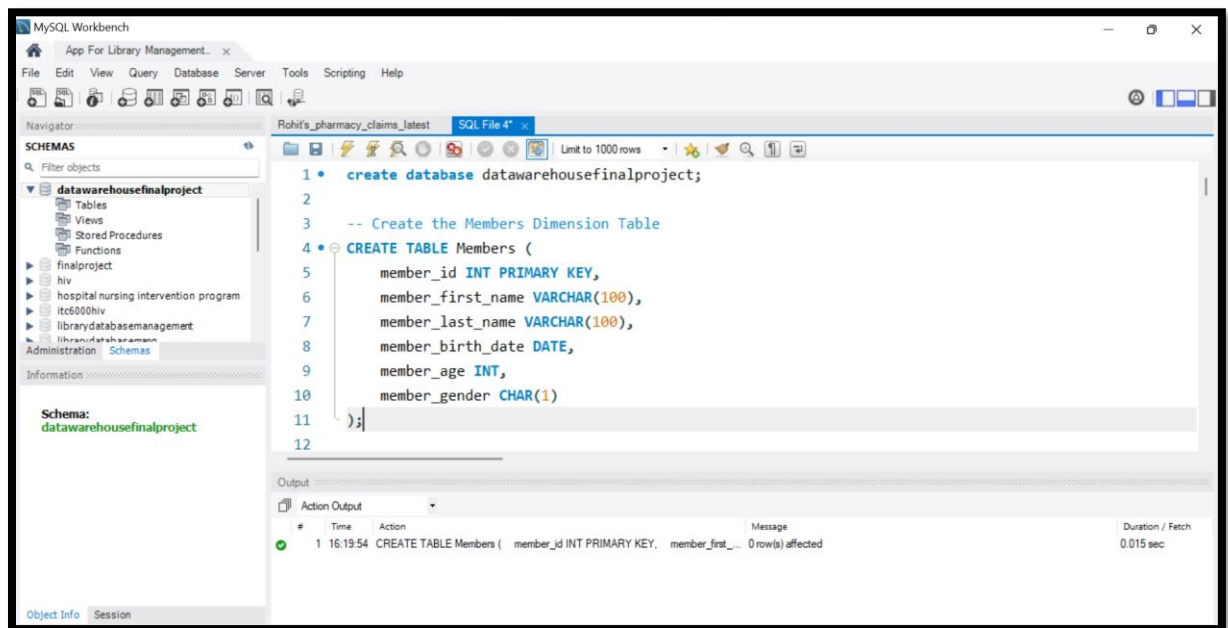


Figure 1: Code to create a member dimension table and designating primary key to the table

Primary Key: member_id (natural key, as it is an identifier for members)

Aly 6030: Final Project.

Foreign Keys: None

2. Drugs Dimension Table:

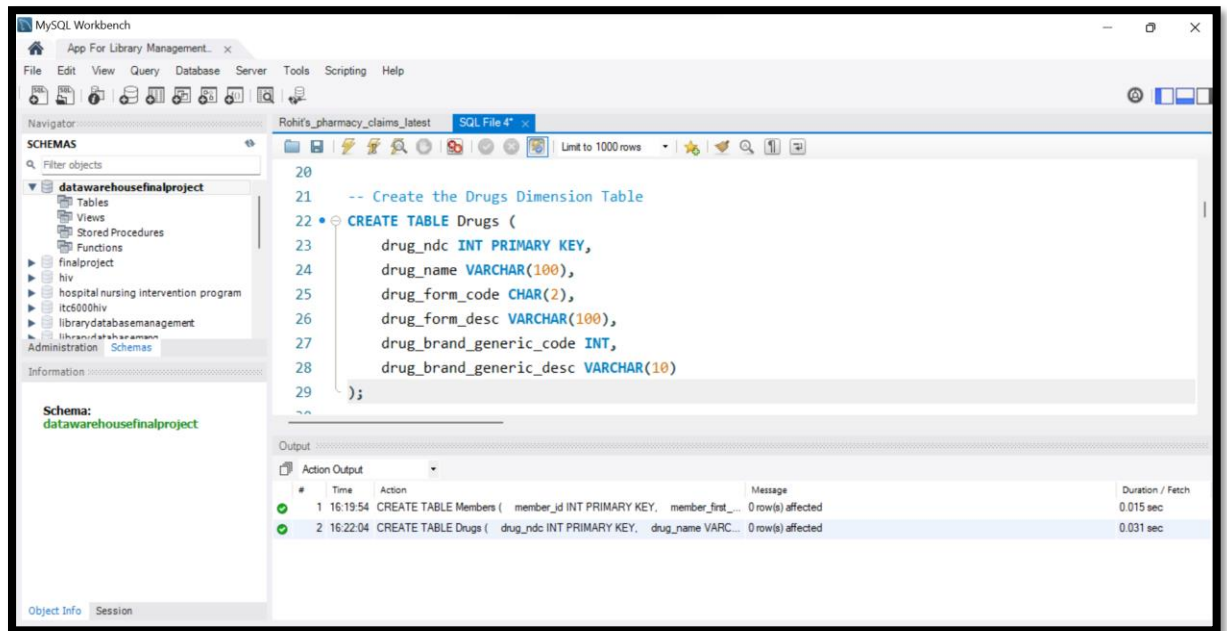


Figure 2: Code to create drug dimension table and designating key to it.

Primary Key: drug_ndc (natural key, as it is an identifier for drugs)

Foreign Keys: None

3. Drug Fill Fact Table:

Aly 6030: Final Project.

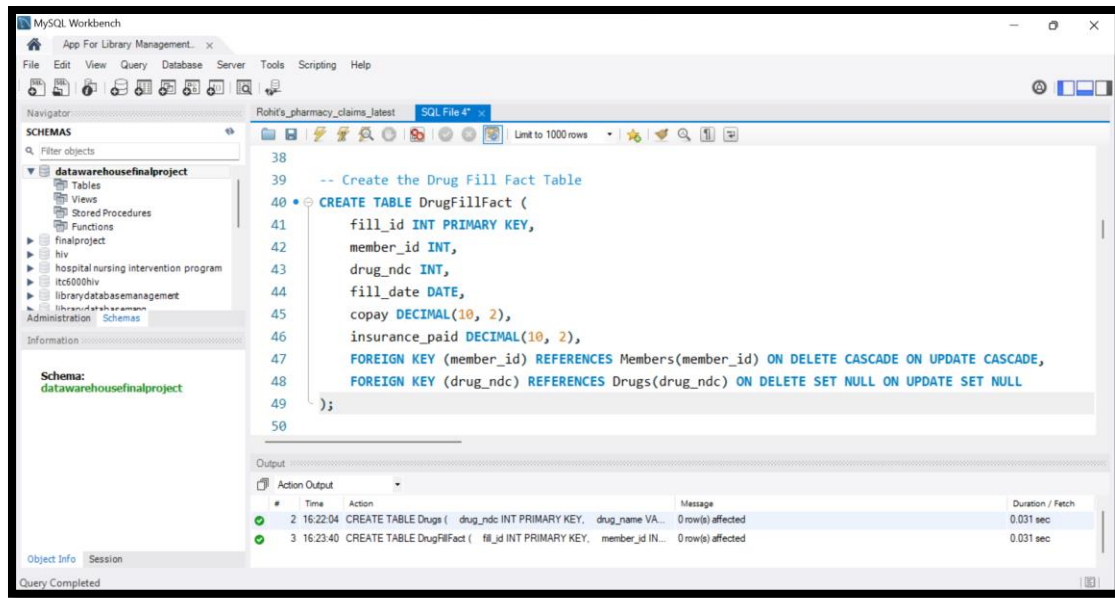


Figure 3: Code to create drug fill fact table and assigning primary and foreign key to it.

Primary Key: `fill_id` (surrogate key, as it is an auto-incremented identifier for the fact table)

Foreign Keys:

member_id (referencing `Members(member_id)` - natural key relationship, as it connects to the `member_id` in the `Members` table.

drug_ndc (referencing `Drugs(drug_ndc)` - natural key relationship, as it connects to the `drug_ndc` in the `Drugs` table.

Question 2:

From figure 3: The foreign keys (FK) designated for each table and the referenced primary key (PK) table are as follows:

Drug Fill Fact Table:

Foreign Key: `member_id` (referencing `Members(member_id)`)

Foreign Key: `drug_ndc` (referencing `Drugs(drug_ndc)`)

Aly 6030: Final Project.

For the Drug Fill Fact Table, the foreign keys member_id and drug_ndc are designated.

These foreign keys reference the primary keys in other tables:

member_id references the **member_id** primary key in the **Members** table.

drug_ndc references the **drug_ndc** primary key in the **Drugs** table.

These foreign keys establish relationships between the Drug Fill Fact Table and the Members and Drugs dimension tables, allowing data in the fact table to be linked to specific members and drugs.

Question 3:

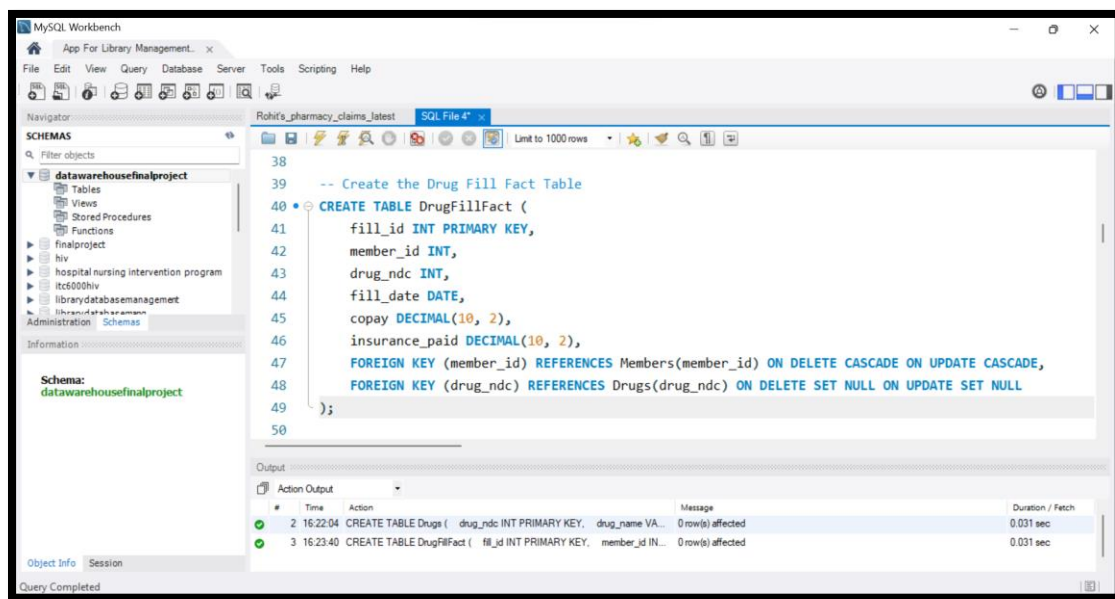


Figure 4: Assigning foreign key constraint i.e cascade and set null.

From figure 4; here for each foreign key (FK) in the DrugFillFact table is configured with a specific action for both deletion and update. Here's an explanation of the actions chosen for each FK and the reasoning behind those choices:

Aly 6030: Final Project.

member_id Foreign Key:

Action on Deletion: CASCADE

Action on Update: CASCADE

Justification:

CASCADE is chosen for both deletion and update actions on member_id. This means that if a member record is deleted or updated in the Members table, the corresponding records in the DrugFillFact table will also be deleted or updated automatically.

This choice ensures data consistency and reflects the real-world relationship where changes to member information should also affect their drug fill records. For example, if a member's ID changes due to a data entry error or a member leaves the database, this change should propagate to the associated drug fill records.

drug_ndc Foreign Key:

Action on Deletion: SET NULL

Action on Update: SET NULL

Justification:

SET NULL is chosen for both deletion and update actions on drug_ndc. This means that if a drug record is deleted or updated in the Drugs table, the foreign key value in the DrugFillFact table will be set to NULL.

This choice allows for flexibility in the drug information while maintaining referential integrity. For example, if a drug is no longer available or is replaced with another drug, you may want to preserve the drug fill records but make it clear that the drug information is no longer available.

Aly 6030: Final Project.

This choice is also aligned with the idea that drugs might be discontinued or their details could change over time, but the historical drug fill records still need to be retained for analysis or reporting.

These choices are made based on the understanding of the specific requirements of the database and the relationships between tables. The goal is to maintain data integrity while also allowing for practical scenarios that can occur in real-world data management. The selection of CASCADE for member_id ensures that member-related data is consistently updated, while SET NULL for drug_ndc allows for flexibility in drug management while retaining historical data.

Part 3 — Entity Relationship Diagram (ERD) Using star schema

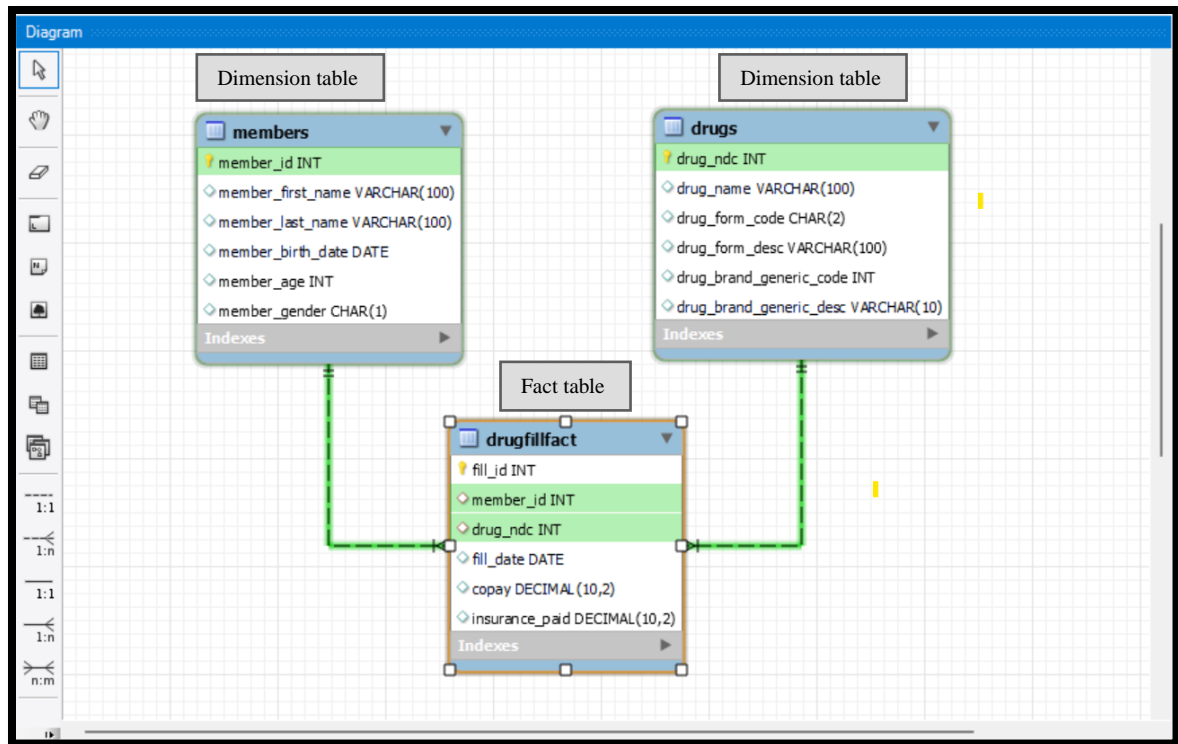


Figure 5: Entity relationship diagram.

Observations:

In the given entity relationship diagram, it can be observed that "Members" and "Drugs" are categorized as dimension tables, while "DrugFillFact" is categorized as a fact table.

Entity relationship explanations:

The entity relationships are as follows:

Members and DrugFillFact: This is a one-to-many relationship. Each member can have multiple drug fill records in the DrugFillFact table, but each drug fill record is associated with a single member.

Drugs and DrugFillFact: This is also a one-to-many relationship. Each drug can be associated with multiple drug fill records in the DrugFillFact table, but each drug fill record is linked to a single drug.

Aly 6030: Final Project.

In summary, there are two one-to-many relationships in this database design: one between Members and DrugFillFact, and another between Drugs and DrugFillFact.

Part 4 — Analytics and Reporting

1.

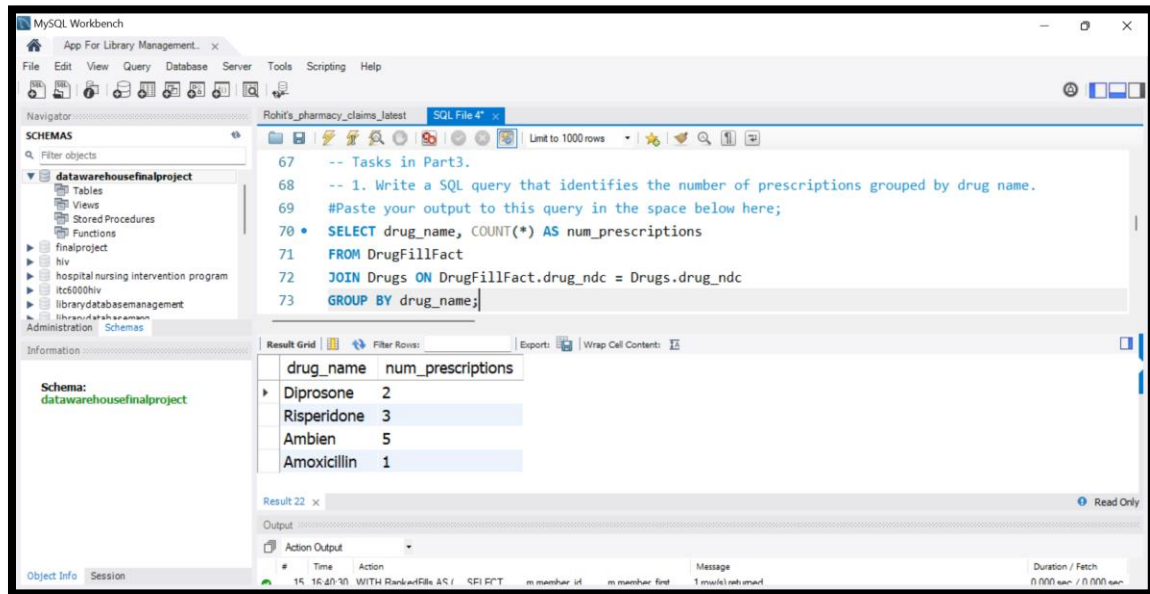


Figure 6: Query to identifies number of prescriptions.

Observations:

There were approximately 2 prescriptions for Diprosone and approximately 3 prescriptions for Risperidone, and so forth.

1.b

Aly 6030: Final Project.

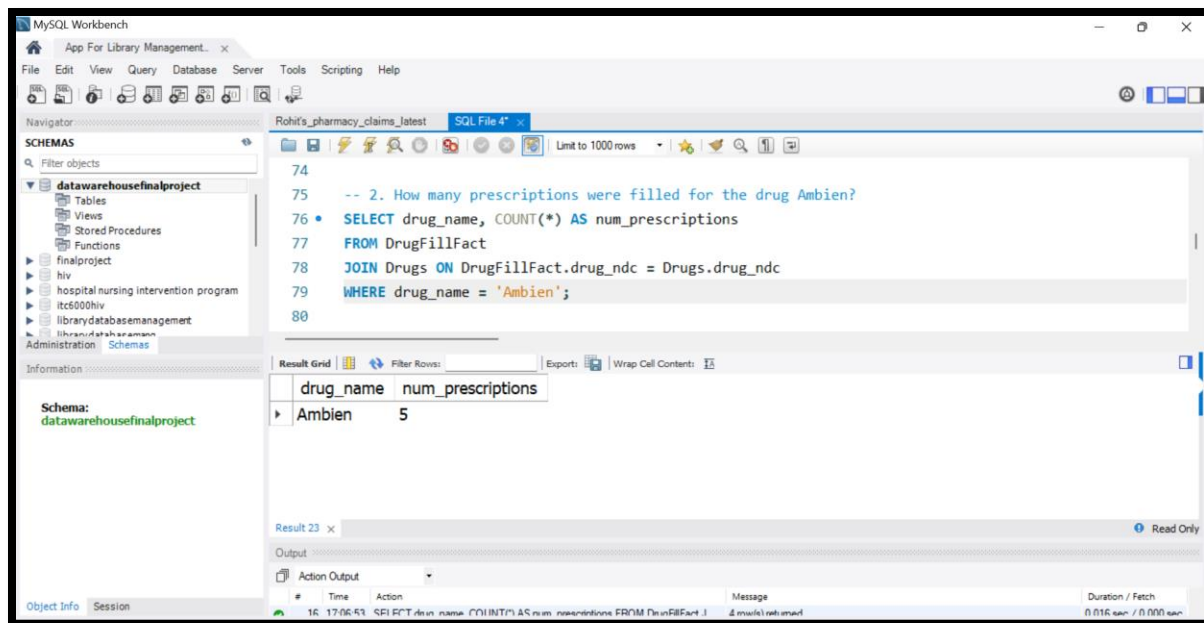


Figure 7: Query to retrieve prescription for Ambien

Observations:

For Ambien there were around 5 prescriptions

2.

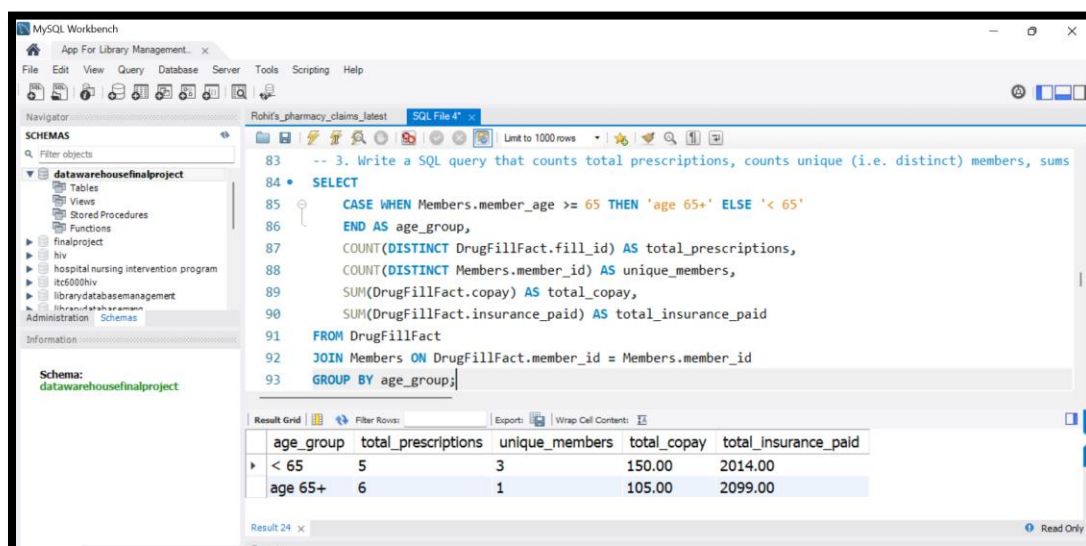
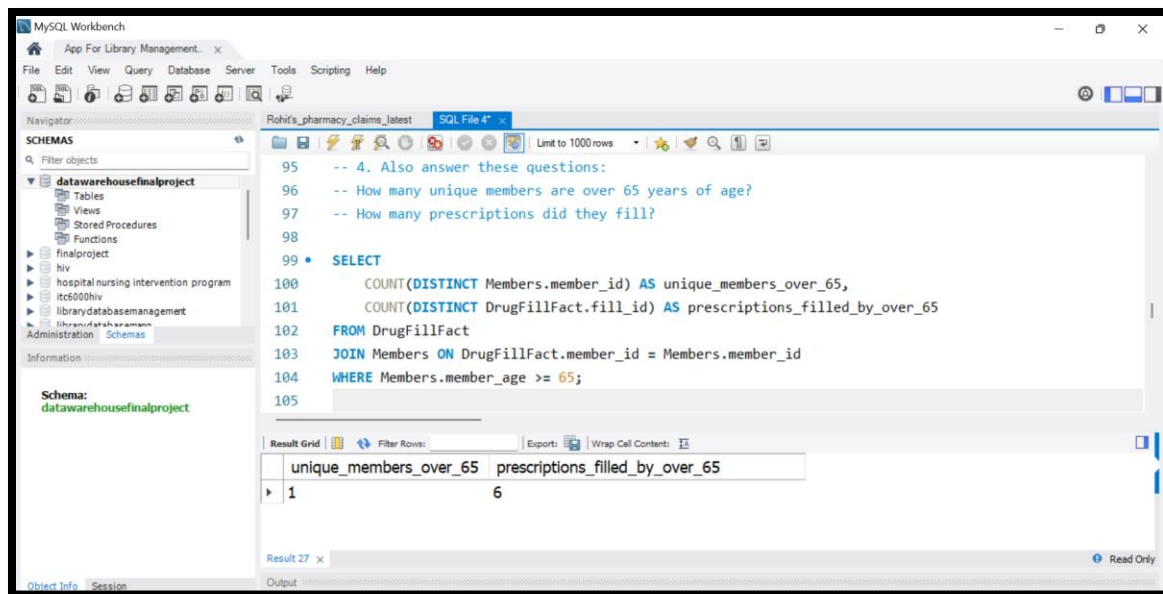


Figure 8: Sql Query to retrieve total prescription, unique member etc.

2.b

Aly 6030: Final Project.

**Observations:**

There was approximately 1 distinct member aged over 65 years who filled approximately 6 prescriptions.

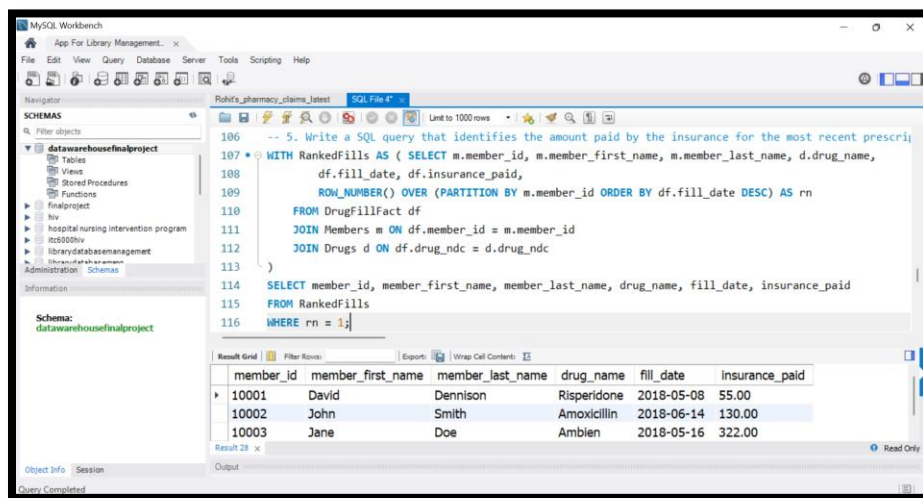
3.

Figure 9: Query to retrieve amount paid by the insurance for the most recent prescription.

Aly 6030: Final Project.

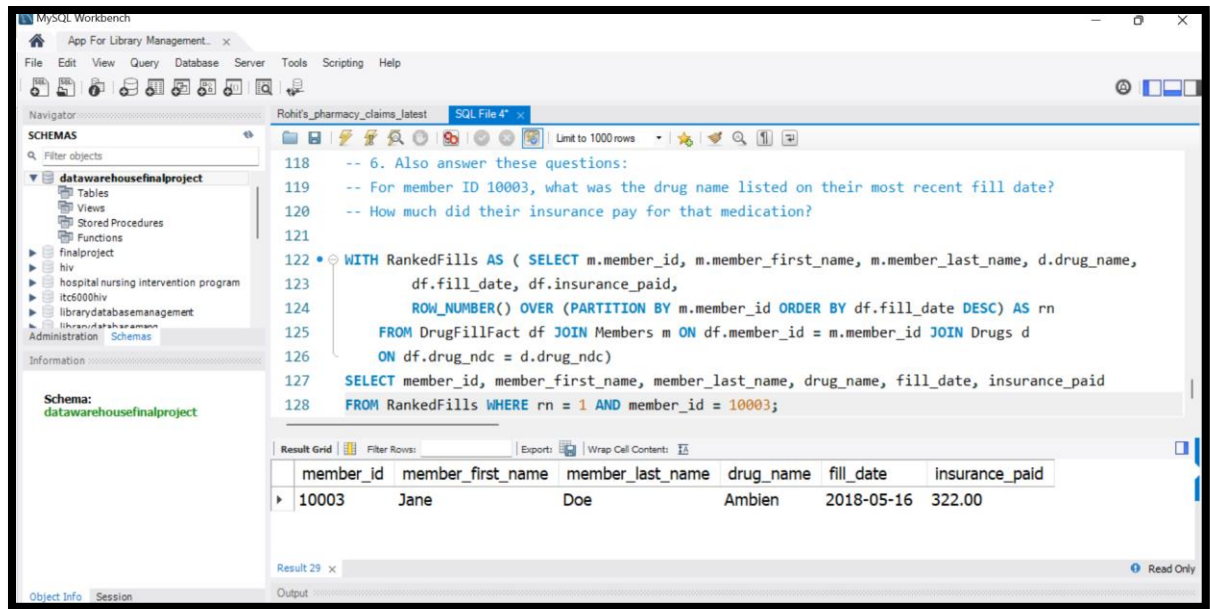


Figure 10: Query to retrieve most recent fill date about drug and insurance paid by id 10003

Observations:

On May 16, 2018, Jane Doe acquired Ambien and covered \$322 in insurance costs.

Summary:

In the presented SQL code and our observations, we have gleaned essential insights into the creation and management of a healthcare-related database. The code begins by establishing the "PHARMACY_CLAIMS" database and creating three crucial tables: "Members," "Drugs," and "DrugFillFact." The "Members" and "Drugs" tables serve as dimension tables, containing member and drug-related attributes, respectively. The "DrugFillFact" table, acting as a fact table, bridges members and drugs, recording specific drug fill events. Notably, the entity relationships are one-to-many, both between "Members" and "DrugFillFact" and between "Drugs" and "DrugFillFact." The code also demonstrates the use of foreign key constraints to maintain referential integrity,

Aly 6030: Final Project.

particularly in the "DrugFillFact" table. Furthermore, it allows for specifying actions in response to deletions or updates in referenced tables, providing flexibility in data management strategies. The project exemplifies the practical application of SQL in healthcare data management, showcasing the importance of data integrity and structured relationships between tables for effective data organization and analysis across various domains.

References:

1. Data based design and normalizations: <https://www.ibm.com/docs/en/db2-for-zos/11?topic=modeling-normalization-in-database-design>
2. SQL: https://www.w3schools.com/sql/sql_intro.asp
3. Entity Relationship Diagram: <https://www.lucidchart.com/pages/er-diagrams>