

**ALY6030 Final Project Assignment – Pharmacy Claims**

**Due End of Week 6**

**Background:**

The excel file “**ALY 6030 Final Project Data Set.xlsx”** contains sample pharmacy claims for made-up members of an insurance company that pays for pharmacy via a third party known as a *Pharmacy Benefit Manager* or PBM. The file also contains a data dictionary referring to the variables in the data along with their format for your reference.

The PBM has given you these sample records as a starting point so that you as the **Developer** can set up a **test database** and pre-program some common SQL query reports that will be expected from the reporting analysts and business users working at your company, once the *full* claims detail is made available in production in a few months from the PBM. Your work on the test warehouse will help ensure a smooth and successful rollout once the data warehouse goes live in production.

**Part 1) Normalization**

1. For each fact variable in your fact table, what type of fact is it? Additive, semi-additive, or non-additive?

**Answer:** As we know, a fact table is primarily used in the dimensional modeling while designing a Datawarehouse. The facts present in the fact table contain measurements and metrics. The fact table can store attributes with respect to the following measure types:

Additive: These measures can be added to all the dimensions

Non-additive: These measures cannot be added to all the dimensions

Semi-additive: These measures can be added to some dimensions and all

The following fact types were identified from the fact table:

id : Non-additive fact

copay: Additive fact

drug\_ndc: Non-additive fact

insurancepaid- Additive fact

fill\_date- Non-additive fact

1. In your fact table, describe the *grain* in one sentence. What does each fact row represent?

**Answer:** The grain determines the level of granularity that is present in every fact table. It shows the finest level of details that is associated with the data. The fact row in the fact table displays the charges and amount paid by every patient, on a specific date. The fact table consists of id, copay, drug\_ndc, insurancepaid and fill\_date.

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

After you’ve converted the raw table into your 3NF fact and dimension table csv files, you are now ready to upload the data and create the snowflake schema in MySQL as well as designate the primary and foreign keys. All of your output to the tasks below should be included in your .sql code file, except for the three questions below which you should answer directly in this word document.

* 1. What are the primary keys you designated for each of your tables? For each PK, is it a natural key or a surrogate key?

**Answer:** The primary key uniquely identifies every record within a database table. In addition, a natural key is a unique key that identifies attributes within a given database table and can be used outside the database. The table, dimension\_drug, has primary key as drug\_ndc, which is a natural key. Moreover, the table, dimension\_member, has primary key as id, which is a natural key. Also, the table, fact\_drug has id as the primary key, which is a natural key.

* 1. What are the foreign keys you designated for each of your tables? For each FK, which table did you reference where that FK is listed as the PK?

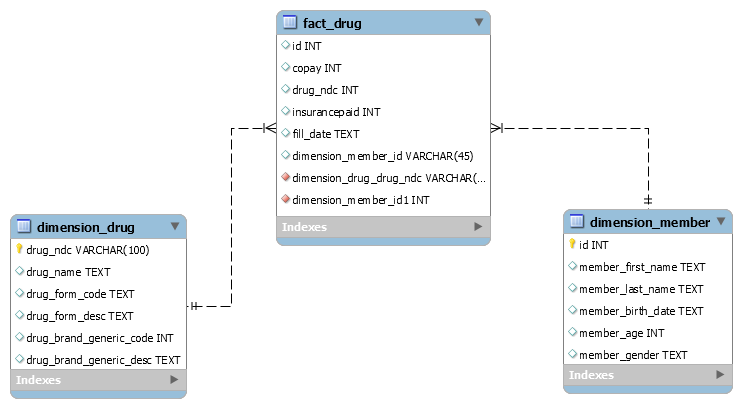
**Answer:** As we know, a foreign key is primarily used to link the two tables. It is a field in one table, whereas a primary key in the other table. The keys, ‘id’ and ‘drug\_ndc’ are used as foreign keys in tables, dimension\_drug and dimension\_member. The key, drug\_ndc acts as a primary key in dimension\_drug and the key, id is used as a primary key in the table, dimension\_member.

* 1. For each FK, what did you tell MySQL to in case of deletion or update (CASCADE, SET NULL, or RESTRICT)? Why did you select the option that you did for each FK?

**Answer:** As we know that, the CASCADE functionality will delete all the corresponding child records if the parent records are deleted, which may not be a feasible solution for a business problem, can lead to issues and loss of data. Whereas, the RESTRICT functionality will restrict the child nodes from being automatically deleted when the parent nodes are deleted, which is a safer option when we have large data. Hence, the RESTRICT clause is preferred.

**Part 3) Entity Relationship Diagram**

Now that you’ve created your database, imported your data, and designated your primary and foreign keys, you’ll need to create the official ERD to be able to communicate this table structure to all business users as well as send back to the PBM so that they have for reference.



*Fig.1: Entity Relationship Diagram*

**Part 4) Analytics and Reporting**

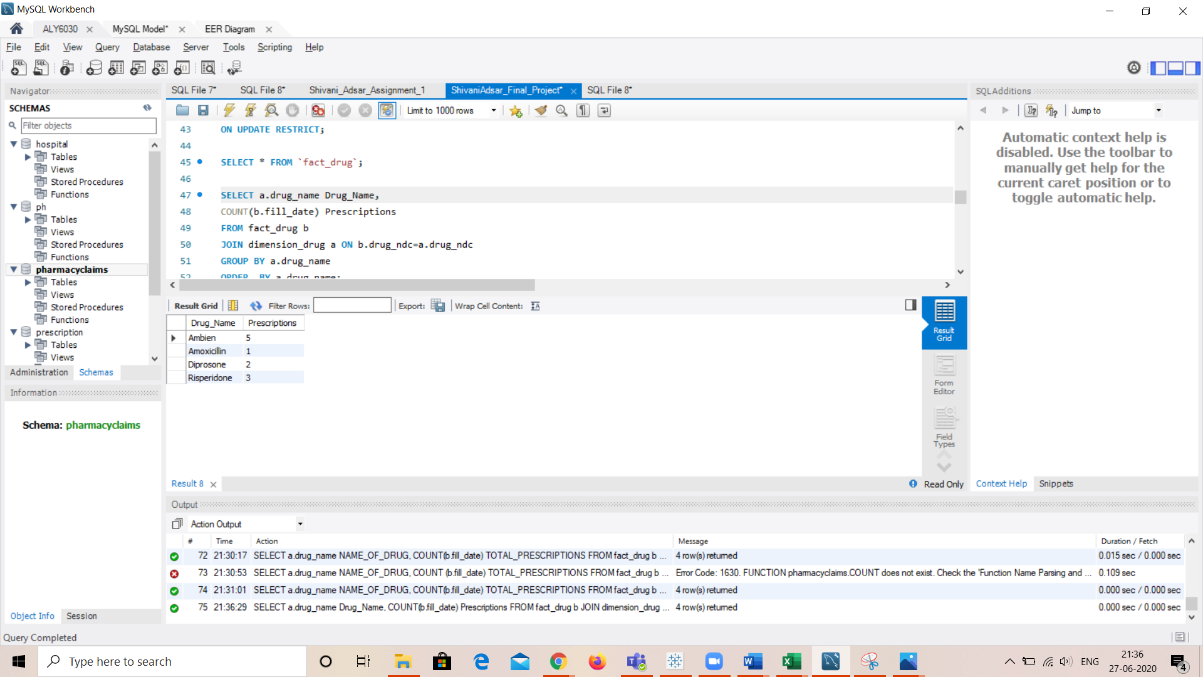
With your database set up and ERD drawn, you want to also provide the business users with some sample queries they will likely need to analyze the production data once it is sent from the PBM.

For now you develop the queries on the sample data even though you don’t really need them since it’s not very large, however in the future this will be expanded to several thousand per month so it’s good to plan ahead for the go-live date.

**Tasks**

* 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?

As we can see, a total of prescriptions for the drug named, Ambien is 5.



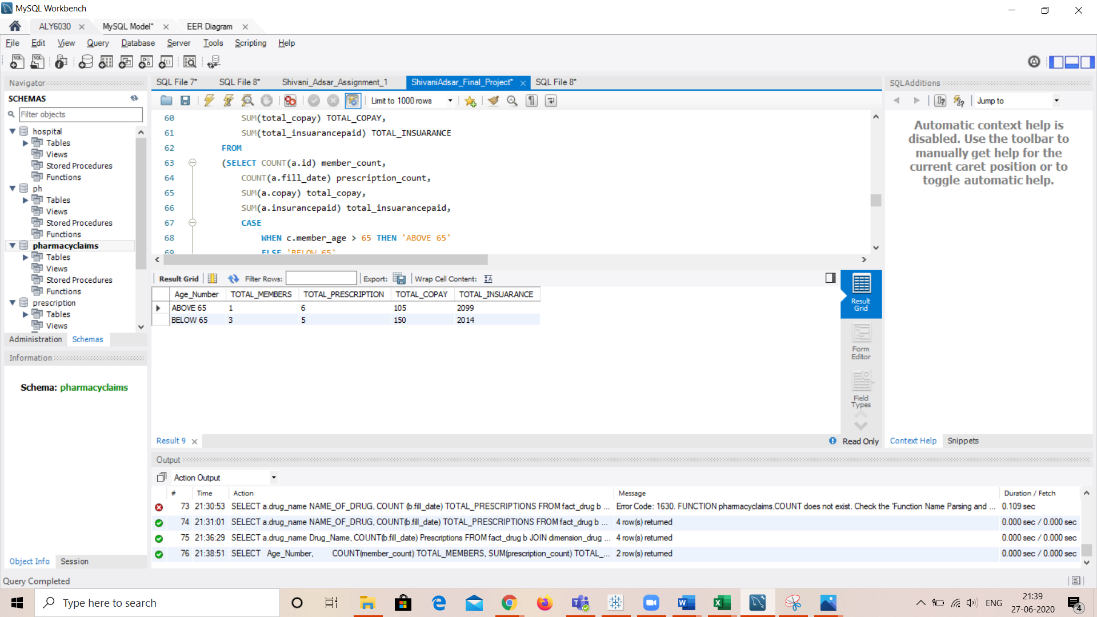
*Fig.1: Total number of prescriptions*

* Write a SQL query that counts total prescriptions, counts unique (i.e. *distinct*) members, sums copay $$, and sums insurance paid $$, for members grouped as either ‘age 65+’ or ’ < 65’. **Use case statement logic** to develop this query similar to 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?

As we can see, there is one patient above 65 years of age.

* + How many prescriptions did they fill?

The patient has filled up 6 prescriptions



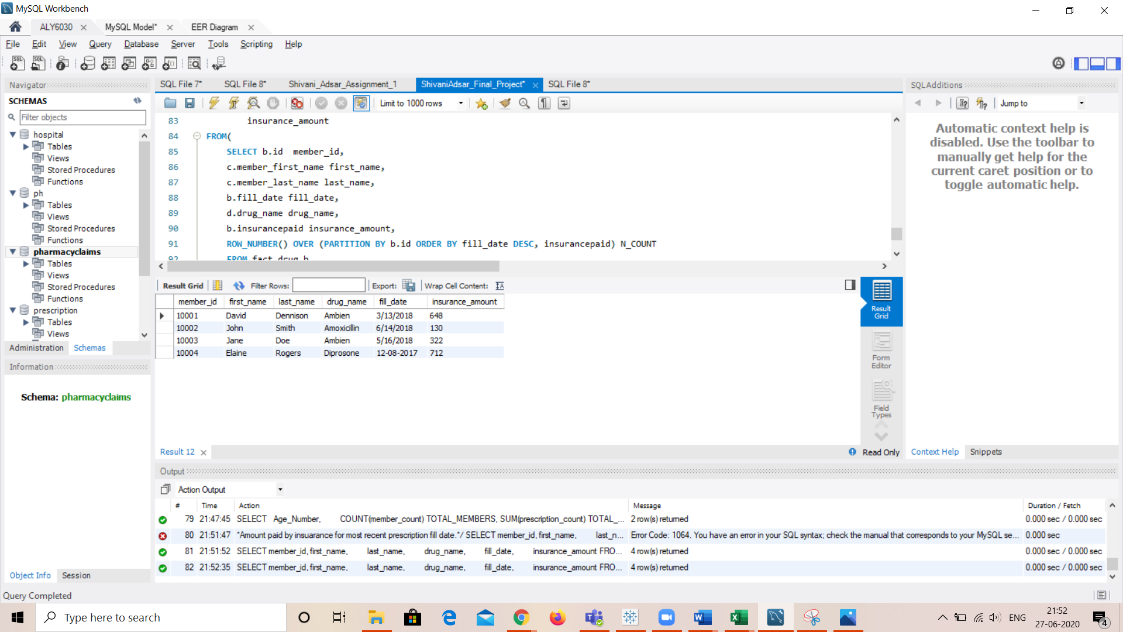
*Fig.3: Prescriptions as per age of patient*

* 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?

The drug name for member id 10003, was Ambien as of 5/16/2018.

* + How much did their insurance pay for that medication?

The insurance amount for the member was $322



*Fig.4: Insurance Details*