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

Your task is to eventually load this sample data into your test database and create a star schema.

However before you do, you notice that the **raw data does not meet 3NF**. You make a note to let the PBM know they need to change their formats for a relational database. You also see **some of your sample members appear more than once** because they filled a different drug which further complicates the format.

For now you decide to fix this yourself directly in excel.

Tasks

- Convert the raw data into a set of relational tables that meet 3NF standards. Feel free to do this directly in excel.
- Each table you create should be either a **fact** table or a **dimension** table
- You do not need to create a separate date dimension joined on date keys. Just include the dates as they appear in the data.
- Save each table as its own .csv file. Indicate in your .csv filename whether your table is a fact table or a dimension table (e.g. name it dim_tablename.csv, fact_tablename.csv, etc.).
- At this time you do not need to create or designate primary or foreign keys, that will be done in Part 2 using MySQL
- Answer the following questions:

1. For each fact variable in your fact table, what type of fact is it? Additive, semi-additive, or non-additive?
2. In your fact table, describe the *grain* in one sentence. What does each fact row represent?

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 star 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.

Tasks

- Import the set of 3NF csv files into MySQL either into a new database or one you've worked with before, that's up to you.

- Designate the **primary keys (PKs)** in MySQL similar to what we did in Week 5 lab. Indicate whether you want to create a primary key as a **natural key** or a **surrogate key** using the SQL code from the lab. Either choice is valid to use so long as it meets the requirement of a PK that uniquely identifies the rows in your tables.

  A reminder that MySQL cannot designate a character variable as a PK unless you **first format it as VARCHAR** (use VARCHAR(100))

- Designate the **foreign keys (FKs)** using MySQL similar to Week 5 lab. Indicate what you want MySQL to do with the FKs in case of **DELETION** or **UPDATE**. Select either **CASCADE**, **SET NULL**, or **RESTRICT** for each of your FKs.

- Answer the following questions here in MS Word:

  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?
  2. 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?
  3. 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?

     Here I'm not necessarily looking for a right or wrong answer to this question, i.e. if you selected SET NULL but you should have selected CASCADE…that's not the goal; rather I want to make sure you understand the *differences* between these options enough to *justify* your selection. Again it matters less which one you choose vs. that you explain your choice to me in a way that demonstrates comprehension.

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.

Tasks

- Create an **ERD** of your star schema fact and dimension tables using MySQL (you can use ERDPlus instead if you want).
- Be sure to label your PKs and FKs as we did in lab 2. (For Mac users, I know the red key does not appear for FKs and that is ok, I will know that is your designated FK in that table even without the red key)
- Also, be sure that your tables are ***labeled*** as either facts or dimensions. This is easy to do if your table name also reflects the type of table that it is per Part 1.
- To be true to the star schema format, **the fact table should be placed in the middle of your diagram** and all dimension tables should fan out from there as we saw in class
- **Export your star schema** as a single page PDF. Include your PDF in your final folder submission.

  If your ERD is not fitting the page well on your PDF, along the top menu bar, first click on **Model** then **Diagram Properties and Size**. In the box that opens up make sure that it is set to one page like this:

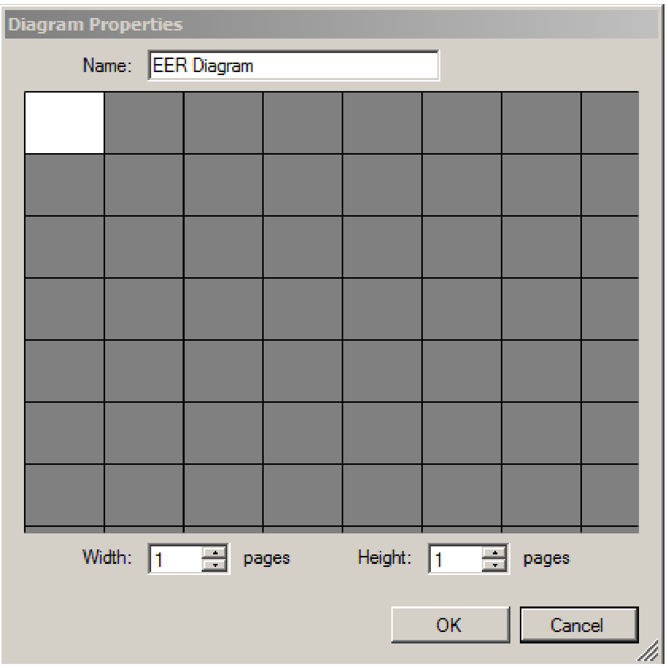

Diagram Properties

Name: EER Diagram

Width: 1 pages

Height: 1 pages

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?
- 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 50+' or ' < 50'. **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 50 years of age?
  - How many prescriptions did they fill?
- 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 10004**, what was the drug name listed on their most recent fill date?
  - How much did their insurance pay for that medication?

Submit one MS Word Document and one SQL program file into the Final project drop box in week 5.

Make sure your name is in both of your filenames

Also, submit all CSV files you generated in excel as part of your submission

Also, submit your ERD as a PDF export from MySQL Workbench

Example:

Final_firstname_lastname.doc, Final_firstname_lastname.sql, Final_firstname_lastname_ERD.pdf

All CSV Files you imported into MySQL Workbench submit those to the drop box as well.