Proposal

for

Design and implementation of health insurance trend analytics with PL/SQL

Category: Relational Application Development

INFO 606

term: Winter 2022

ZACH CARLSON, KATY MATULAY, JACOB STANK, JACOB WILLIAMSON

[I. Business Problem and Goals 3](#_Toc266428317)

[II. Scope and Functionality 3](#_Toc266428318)

[III. Assumptions and Requirements 4](#_Toc266428319)

[IV. Input, Output, and Knowledge Acquisition 5](#_Toc266428320)

[V. Deliverables 5](#_Toc266428321)

# **Business Problem and Goals**

Health Insurance Companies require the ability to analyze claim costs and utilization patterns over time. This is required to inform pricing decisions, create financial forecasts, inform product design, and enact cost saving programs. Within this database, health plan operational data will be enhanced to allow for more refined analysis to better target the areas driving per member per month claim costs. The adjustments to the data will enable more targeted actions to be taken and ensure effective use of staffing resources in producing solutions to drivers of trend and implementation of more sophisticated predictive analytics and business intelligence applications.

At a high level the enhancement to this data will be:

* Bucketing of clinical data elements to standardize reporting categories.
* Logic to assign a unique visit a member had to a provider within the defined reporting category.
* Integration of chronic conditions of members to provide a clinical view of trend.
* Integration of geographic buckets for views of trend by member location.

# **Scope and Functionality**

1. **IN-SCOPE:**

The scope of the project will be limited to trend analytics centering on members, providers, and claims using Python and SQL.

**Record Keeping**

* Manage (insert/delete/edit) the member and member data
* Manage the provider and provider data
* Manage member policy data
* Manage member claim data
* Manage the clinical conditions of members
* Manage claim categories and utilization
* Manage date mapping into trend periods

**Reporting**

* Create rolling average time series visualizations that highlight any seasonality or long-term trends.
* Visualize demographics of members.
* Identify members who are high dollar claimants and assess whether these outliers have an impact on overall trends.
* Separately group data by Claims, Member Condition, and Member Location
  + Analyze grouped data by utilization and cost over time.

**Search Functionality**

Search and retrieve Member records using the fully qualified criteria:

* Member ID
* Member last name
* Member Condition
* Member Location

**Processing Functionality**

* Application of business rules to data for reporting and analytics
* Ability to ingest new incremental data monthly
* Audit tracking of which user/process inserted, edited, or made changes
* Clinical view of trend
* Claims transaction view of trend
* Aggregation of data into easy-to-use summary views/tables to lessen the need for joins and simplify the experience for end-users and BI developers.

1. **OUT-SCOPE:**

This project will not include payment processing activities and business management activities such as: payroll, benefits, premium, pharmacy rebates, pharmacy claims, and office supply purchasing. All data will be de-identified and contain only the minimum necessary information needed to perform trend analytics defined in the reporting requirements.

# **Assumptions and Requirements**

Hardware and software requirements have been derived using the following assumptions.

**Assumptions**

* Users will run the analytical Python code within a Jupyter Notebook environment (i.e., Google Colab, Anaconda/Jupyter Lab).
* Users will have installed all required Python packages before running Notebook.
* Users will have created their own local MySQL database and entered their own connection string into the respective Notebook code cell.
* The company has an existing backup solution.
* If an employee is saving the Jupyter Notebook to a public repository on GitHub, the connection string is deleted before any commits are made, otherwise the MySQL connection string will be exposed, and the database compromised.

**Hardware/Software**

The hardware/software solution will work in both Windows 10 and mac OS as long as Google Colab is used to access the Jupyter Notebook and the database is created within MySQL Workbench and Heroku (or any other valid MySQL database set up).

**PL/SQL Implementation**

Each member will create at least a total of four program units with any combination of trigger, stored procedure, and stored function that implements certain processing functionality of this application.

# **Input, Output, and Knowledge Acquisition**

The project will take data stored in a MySQL Health Insurance database and save it as a DataFrame within Python. Each table within the database will be imported as its own DataFrame and joined to other tables when appropriate. The DataFrames will be analyzed and visualized using pandas, matplotlib.pyplot, and other Python libraries. In addition, company employees will need to acquire knowledge of certain components, to utilize and maintain the Jupyter Notebook.

**Input**

* Members
* Claims
* Provider
* Policies
* Clinical Conditions
* Claims Category Business Rules

**Output**

* Refined Claims Detail
* Reports to support Trend Analysis
* High Dollar Claimants
* Utilization measures

**Knowledge Acquisition**

* MySQL Workbench
* Heroku + ClearDB Set up
* Python
* Jupyter Notebook
* Connecting a MySQL database in Jupyter Notebook
* PL/SQL

# **Deliverables**

Upon completion of the project, a zipped file with all the diagram files, SQL and PL/SQL codes, presentation file, Jupyter Notebook, and the final report will be submitted. All files will be uploaded to a GitHub repository. For security purposes, the final Jupyter Notebook will be run entirely, output saved, and the cell containing the connection string for our database will be deleted. Anyone attempting to rerun the Jupyter Notebook will need to create their own database connection using the SQL scripts provided. The report will include the following components:

* + - 1. Overview
      2. Problem Statement
         1. Problem Description: context, goal, scope
         2. Assumptions
         3. Functional requirements
         4. Data requirements
         5. HW and SW
         6. Methodology
      3. Conceptual Design
         1. ERD
         2. ERD Explanation
      4. Logical Design
         1. Relational Schema
         2. Data dictionary
      5. Implementation
         1. CREATE TABLE commands
         2. INSERT commands and Data
         3. PL/SQL implementation and testing

Zach Carlson Programs

Katy Matulay Programs

Jacob Stank Programs

Jacob Williamson Programs

* + - 1. Visualizations and Analytics
      2. Conclusion
         1. Concluding Remarks
         2. Future work
         3. Lessons Learned
      3. References
      4. Appendix

# **References**