ETL Project:

The Process

Team member:

Dikshant Dhall

Imanuel C. Gemilang

Allison Hsu

The project that we decided to do is about medical financial, focusing on procedure of coronary artery bypass grafting (CABG). We found three datasets that are related to each other. The first data (IPPS-all.csv) is download from CMS website and it contains that the DRG, financial information, and the provider ID (Medicare provide ID) for year of 2017. The following data (OSHPD.csv) is found from OSHPD website. It contains OSHPD ID, CABG mortality for year of 2017. Finally, the last data (cover ca.csv) is extracted via health insurance of covered California regarding to hospital information. It has both provider ID number and OSHPD ID number, which are used to join the three tables together.

For the IPPS data, we decided to focus only on the state of California to get more meaningful information to match with California State Data from OSHPD. We also filter the DRG that contains 233, 234, 235, and 236 which is related to the CABG. The following steps that we did were renaming some of the columns that would be proper for our database.

Next, for the cover ca dataset, we chose for columns: entity name (hospital name), provider ID (Medicare provide ID), and OSHPD ID. We dropped all the rows that contain null. We also converted the Medicare provider number and OSHPD to an integer type. Next, we rename the column to the appropriate columns name.

Finally, our last data set, OSHPD. To open the file, we needed to use an encoding, which in this case, we used Unicode\_escape. We filter out the table to the column that we want and look for a specific year, which is 2017. We dropped rows that contain no value, although it doesn’t say null. We also convert OSHPD ID to float and rename all the columns to fit better with other data sets.

The next step that we do after cleaning all the data, we decided to use PostgreSQL and export our data frame and create a table. Some of the challenges that we face were not being able to query any of the tables that we exported. We found out that PostgreSQL cannot read the upper-case letter table, so we had to convert all the tables into a lowercase table to work.