**Architecture Design**

Application Idea

The goal of the application is to study quality of care for Medicare patients. This can be useful in understanding which players offer the highest quality of care, which procedures have high variability in quality, and how traditional quality of care metrics relate to patient survey outcomes. The data for the study comes from the CMS Hospital Compare project.

The application process the data from the CMS Hospital Compare project and loads the relevant elements into an HDFS data lake. It then creates Hive metadata, cleans the data, and transforms it into a set of easily accessed and understood tables. Finally, there are several queries that produce results answering several key questions.

Technologies used: Amazon EC2, HDFS, Postgres, Hive

Directory and File Structure

The code is stored in https://github.com/PageJewel/W205/tree/master/Exercise1. Inside this directory, there are three folders containing code:

- loading_and_modelling
    - 'load_data-Lake.sh' - this removes headers, renames files, and moves the data into HDFS
    - 'hive_base_ddl.sql' - this creates the Hive metadata for the underlying data
- transforming
    - 'hospitals.sql' - this creates the cleaned hospitals table
    - 'hospital_procedure_scores_temp' - this generates an intermediate cleaned table of scores by measure for each hospital
    - 'procedures.sql' - this create the cleaned procedures table
    - 'hospital_procedures_scores' - this creates the final cleaned table of scores by measure for each hospital
    - 'survey_results.sql' - this creates the cleaned table of survey responses
- investigations
    - 'best_hospitals/best_hospitals.sql' - returns the 10 hospitals with consistently high quality
    - 'best_states/best_states.sql' - returns the 10 states with consistently high quality

    - 'hospital_variability/hospital_variability.sql' - returns the 10 procedures with the most variation between hospitals
    - 'hospitals_and_patients/hospitals_and_patients.sql' - returns the correlations between hospital scores and variability with patient survey responses

There are also documentation and result files stored in this directory at:

- 'Readme.md' – instructions on how to set up and run the application
- 'Architecture.pdf' – this overview document
- 'loading_and_modelling/ER_diagram.png' – displays the entity relationship diagram for the transformed data
- 'investigations/best_hospitals/best_hospitals.txt' – process justification and results for top hospitals with consistently high quality
- 'investigations/best_states/best_states.sql' - process justification and results for top states with consistently high quality
- 'investigations/hospital_variability/hospital_variability.sql' - process justification and results for procedures with the most variation between hospitals
- 'investigations/hospitals_and_patients/hospitals_and_patients.sql' - process justification and results for the correlations between hospital scores and variability with patient survey responses

Data Cleaning Process

- Removed childrens and veterans hospitals because they did not have comparable populations to the other hospitals
- Removed measures which are not relevant for calculating a quality of care overall score (based on a consultation with a doctor for his medical opinion)
    - VTE_1 - subjective measure because it requires a good standard of who should be receiving prophylactic treatment to begin with
    - VTE_2 - subjective measure because it requires a good standard of who should be receiving prophylactic treatment to begin with
    - VTE_3 - irrelevant measure for quality of care
    - VTE_4 - irrelevant measure for quality of care
    - CAC_1 - measure of children's care
    - CAC_2-- measure of children's care
    - CAC_3 - measure of children's care
    - ED_1b - affects patient satisfaction, not quality of care with relation to medical outcomes
    - ED_2b - affects patient satisfaction, not quality of care with relation to medical outcomes
    - EDV - irrelevant measure for quality of care
    - OP_18b - affects patient satisfaction, not quality of care with relation to medical outcomes

    - OP_21 - affects patient satisfaction, not quality of care with relation to medical outcomes
    - OP_22 - many reasons for leaving, not all of which are associated with quality of care
    - AMI_7a - irrelevant because majority of patients with heart attack treated with PCI instead
    - OP_1 - irrelevant because majority of patients with heart attack treated with PCI instead
    - OP_2 - irrelevant because majority of patients with heart attack treated with PCI instead
    - HF_2 - irrelevant measure for quality of care
    - STK_8 - affects patient satisfaction, not quality of care with relation to medical outcomes
    - OP_6 - irrelevant because there is not much evidence for prophylactic antibiotics affecting quality of care with relation to medical outcomes
    - OP_7 - irrelevant because there is not much evidence for prophylactic antibiotics affecting quality of care with relation to medical outcomes
    - SCIP_INF_1 - irrelevant because there is not much evidence for prophylactic antibiotics affecting quality of care with relation to medical outcomes
    - SCIP_INF_2 - irrelevant because there is not much evidence for prophylactic antibiotics affecting quality of care with relation to medical outcomes
    - SCIP_INF_3 - irrelevant because there is not much evidence for prophylactic antibiotics affecting quality of care with relation to medical outcomes
    - SCIP_INF_4 - no data for this measure
- Removed all data points with data quality issues
    - "Not available" scores for hospital-procedures as well as for survey responses
    - Hospital-procedures scores with footnotes saying "The number of cases/patients is too few to report" or "There were discrepancies in the data collection process"
- Assigned a parameter identifying whether it is better for a procedural score to be higher (1) or lower (-1)

Process to Run Application - note that these are also available in the Readme.txt file

# Inital Setup

1. Start up your AWS machine with the appropriate technology installed

[only needs to be done once] Follow steps 1-3 (through 'Download and Run the Setup Script') on https://github.com/UC-Berkeley-I-School/w205-fall-17-labs-exercises/blob/master/lab_2/Lab2.md to start an appropriate AWS instance.

[Already set up the AWS machine previously] Start the EC2 instance, ssh in to it, and mount your drive as /data.

2. Pull Exercise1 repository in EC2 instance (`git clone https://github.com/PageJewel/W205/Exercise1.git`).

3. Create a directory Exercise1/data - type `mkdir data` from Exercise1 folder.

4. Move the flat files into Exercise1/data if you already have them. Otherwise:

- In Exercise1/data, download the data by typing `wget -O Hospital_Revised_Flatfiles.zip "https://data.medicare.gov/views/bg9k-emty/files/Nqcy71p9Ss2RSBWDmP77H1DQXcyacr2khotGbDHHW_s?content_type=application%2Fzip%3B%20charset%3\Dbinary&filename=Hospital_Revised_Flatfiles.zip"`

- Unzip the downloaded data by typing `unzip Hospital_Revised_Flatfiles.zip`

5. Open permissions on these files so the w205 user can access them. Type `chmod 007 *`. Also grant access to the entire folder so w205 can create files - `chmod 007 ../data/`.

# Clean the data files and load into HDFS directory

6. Start Hadoop (type `/root/start-hadoop.sh`) and postgres (type `/data/start_postgres.sh`) in EC2 instance. Switch to w205 user by typing `su - w205`. Start Hive metastore (type `/data/start_metastore.sh`) in EC2 instance.

7. Go to Exercise1/loading_and_modelling, and type `. load_data_lake.sh`. This processes the data files and loads them into the hdfs file structure.

8. Type `hive -f hive_base_ddl.sql`. This creates the Hive metadata for the tables.

# Transform the data

9. Go to Exercise1/transforming.

10. Type `hive -f hospitals.sql`. This creates the hospitals table from the ER diagram.

11. Type `hive -f hospital_procedure_scores_temp.sql`. This generates an intermediate cleaned table of scores by measure for each hospital.

12. Type `hive -f procedures.sql`. This creates the procedures table from the ER diagram.

13. Type `hive -f hospital_procedure_scores.sql`. This creates the final hospital_procedure_scores table from the ER diagram.

14. Type `hive -f survey_results.sql`. This creates the survey_results table from the ER diagram.

# Return results

15. Go to Exercise1/investigations.

16. Type `hive -f best_hospitals/best_hospitals.sql` to return the top 10 hospitals with consistently high quality

17. Type `hive -f best_states/best_states.sql` to return the top 10 states with consistently high quality

18. Type `hive -f hospital_variability/hospital_variability.sql` to return the top 10 procedures with the most variation between hospitals

19. Type `hive -f hospitals_and_patients/hospitals_and_patients.sql` to return the results of whether average scores for hospital quality or procedural variability are correlated with patient survey responses.