**E**xtract **T**ransform **L**oad **Technical Final Report**

*Team Members:*

*Rachel Newberry, Madhusudan Tamrakar, and Donia Tarjam.*

**## Data Cleanup & Analysis**

​

Once you have identified your datasets, perform ETL on the data. Make sure to plan and document the following:

​

\* The sources of data that you will extract from.

​

\* The type of transformation needed for this data (cleaning, joining, filtering, aggregating, etc).

​

\* The type of final production database to load the data into (relational or non-relational).

​

\* The final tables or collections that will be used in the production database.

​

You will be required to submit a final technical report with the above information and steps required to reproduce your ETL process.

​

**## Project Report**

​

At the end of the week, your team will submit a Final Report that describes the following:

​

​

​

\* **\*\*L\*\***oad: the final database, tables/collections, and why this was chosen.

​

### Instructions

\* Create a `customer\_db` database in pgAdmin 4 then create the following two tables within:

\* A `premise` table that contains the columns `id`, `premise\_name` and `county\_id`.

\* A `county` table that contains the columns `id`, `county\_name`, `license\_count` and `county\_id`.

\* Be sure to assign a primary key, as Pandas will not be able to do so.

\* In Jupyter Notebook perform all ETL.

■ Datasets used and their sources. ■ Types of data wrangling performed - Data cleaning, joining, filtering and aggregating. ■ The schemata used in the final production database.

\* **\*\*E\*\***xtract: your original data sources and how the data was formatted (CSV, JSON, pgAdmin 4, etc).

**Extract** 9 CSV files JSON files HTML tables SQL databases Spreadsheets Extract Data may come from disparate sources, such as:

Extraction ◆ Put each CSV into a pandas DataFrame. ➔

The sources of data that you will extract from.

Extract: your original data sources and how the data was formatted (CSV, JSON, pgAdmin 4, etc)

\* \*\*Extraction\*\*

\* Put each CSV into a pandas DataFrame.

**Transform** 10 Data Cleaning Summarization Selection Joining Filtering Aggregating Transform the data to suit business needs. This may include:

\* **\*\*T\*\***ransform: what data cleaning or transformation was required.

Python and pandas for transformation, which can also be done with SQL or a specialized ETL tool.

Transform ◆ Copy only the columns needed into a new DataFrame. ◆ Rename columns to fit the tables created in the database. ◆ Handle any duplicates. HINT: some locations have the same name but each license number is unique. ◆ Set index to the previously created primary key.

● The type of transformation needed for this data (cleaning, joining, filtering, aggregating, etc).

Transform: what data cleaning or transformation was required.

\* \*\*Transform\*\*

\* Copy only the columns needed into a new DataFrame.

\* Rename columns to fit the tables created in the database.

\* Handle any duplicates. \*\*HINT:\*\* some locations have the same name but each license number is unique.

\* Set index to the previously created primary key.

**Load** 12 Can be a relational or non-relational database Load Can be local or in the cloud Can be a data lake or data warehouse Load the data into a final database that can be used for future analysis or business use.

Load ◆ Create a connection to database. ◆ Check for a successful connection to the database and confirm that the tables have been created. ◆ Append DataFrames to tables. Be sure to use the index set earlier.

The type of final production database to load the data into (relational or non-relational). ● The final tables or collections that will be used in the production database.

Load: the final database, tables/collections, and why this was chosen.

\* \*\*Load\*\*

\* Create a connection to database.

\* Check for a successful connection to the database and confirm that the tables have been created.

\* Append DataFrames to tables. Be sure to use the index set earlier.

\* Confirm successful \*\*Load\*\* by querying database.

\* Join the two tables and select the `id` and `premise\_name` from the `premise` table and `county\_name` from the `county` table.