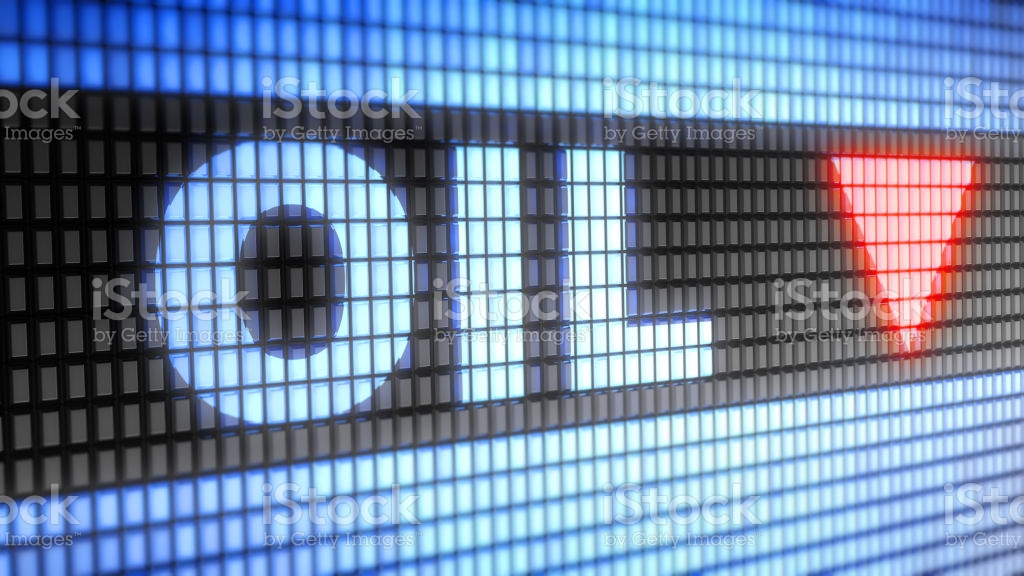
**ETL Final Report**

ETL Project

Nima Karimi and Kirstie McCown

Week 12: University of Western Australia Data Analytics Bootcamp



|  |  |
| --- | --- |
| **Project Title:** | ETL Project |
| **Class Instructor:** | Daniel Kasatchkow |
| **TA:** | Hazar Ayaz |
| **Project Due Date:** | Saturday 29th August 2020 |
| **Date of Report:** | Saturday 29th August 2020 |
| **Reporting Period:** | Tuesday 25th August 2020 – Saturday 29th August 2020 |

**Table of Contents**

Summary 1

Data Sources 1

Data Transformation 1

Database 2

Database Tables and Columns 2

Project Conclusion 3

**Annexures**

Annexure 1. Cleaned Oil Table Schema 4

Annexure 2. Cleaned Accident Table Schema 4

Annexure 3. Table Join Query 5

Annexure 4. Joined Tables from Database 5

Annexure 5. Screenshot of Database and Schemas 6

Annexure 6. Screenshot of Pandas Merge Code and DataFrame 6

**Summary**

Currently, the price of oil is ever changing, and sometimes for unknown cause.

We are carrying out this project to identify if there is any correlation between US Oil Pipeline Accidents and the Crude Oil Price around the same period (2010 to 2016).

**Data Sources**

We utilised two sets of data from Kaggle.com, one was in cvs. format and the other .xlsx:

Oil Prices

<https://www.kaggle.com/rockbottom73/crude-oil-prices>

US Oil Pipeline Accidents

<https://www.kaggle.com/usdot/pipeline-accidents>

**Data Transformation**

Our overall data transformation we wanted to look at the following elements, we will detail these further below for each individual data set.

* Remove any unnecessary columns
* Drop all accidents not related to crude oil
* Drop select items which are N/A or have the value of NaN
* Split accident date/time field to show only dates

Oil Prices

* Read in xlsx to Panadas DataFrame to enable visualisation and cleaning
* Rename columns so they are easier to work with
* Drop everything which is N/A
* Drop all rows that are not the same dates as what is in the accidents DataFrame

US Oil Pipeline Accidents

* Read in csv to Panadas DataFrame to enable visualisation and cleaning
* Remove any unnecessary columns and rename columns so they are easier to work with
* Look at what non null values are in the DataFrame to see if values need to be removed
* Drop all accidents not related to crude oil
* Drop everything which is N/A in the following columns: city, facility\_name, country, shutdown
* Split the date/time column keeping the date in a newly created column, whist dropping the original date/time column
* Change the format of the date so that both DataFrame dates match format

**Database**

For our project we utilised a Postgres SQL Database, as part of our ETL process we conducted the following steps:

* Create a new Postgres Database called “oil\_db”
* Create two table schema’s called “cleaned\_oil” and “cleaned\_accidents”
* Connect to Postgres database via our Jupyter Notebook (.ipynb file)
* Check to ensure tables are available in Postgres database and able to be connected with via our Jupyter Notebook
* Load panda's DataFrame to postgres sql tables

**See: Annexure 1 and Annexure 2**

**Database Tables and Columns**

|  |  |
| --- | --- |
| **Table Name** | **Number of Columns** |
| **cleaned\_oil** | 2 |
| **cleaned\_accidents** | 17 |

**Columns – cleaned\_oil**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Column Type** | **Description** |
| **date** | date | Date oil price was recorded |
| **price** | decimal | End of day price for a given day |

**Columns – cleaned\_ accidents**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Column Type** | **Description** |
| **Report\_number** | Int | Accident Report Number |
| **Op\_id** | int | Operator ID in charge at time of Accident |
| **Op\_name** | varchar | Operator Name |
| **Facility\_name** | varchar | Facility Name where Accident occurred |
| **Location** | varchar | Location of accident (on/offshore) |
| **Pipeline\_type** | varchar | Under/Above Ground Pipeline |
| **Liquid\_type** | varchar | Type of Liquid in the Accident Area |
| **City** | varchar | City of Accident |
| **Country** | varchar | Country of Accident |
| **State** | varchar | State of Accident |
| **Cause\_cat** | varchar | Category of Reason of Accident |
| **Cause\_subcat** | varchar | Sub-Category of Reason of Accident |
| **Shutdown** | varchar | Was the plant shut down at the time of the accident (Yes/No) |
| **Shut\_date\_time** | date | Shutdown date and time if applicable |
| **Restart\_date\_time** | date | Restart date and time if applicable |
| **Date** | date | Date of Accident |

The above two tables were joined in both Panda’s and SQL to create one table for further analysis.

In our Panda’s DataFrame with all merged information there are some N/A’s still present because for each day there was a price reading, there was not necessarily an accident occur on that same date. These have been left in for clarity and can be removed once further investigation is required and commenced.

**See: Annexure 3, Annexure 4, Annexure 5 and Annexure 6**

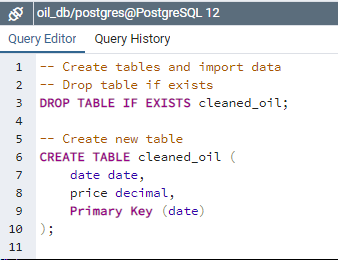
**Project Conclusion**

We feel that our ETL process has prepared the two datasets adequately in order to be able to further analyse and identify if there is any correlation between US Oil Pipeline Accidents and the fluctuation of Crude Oil Prices around the same period of time.

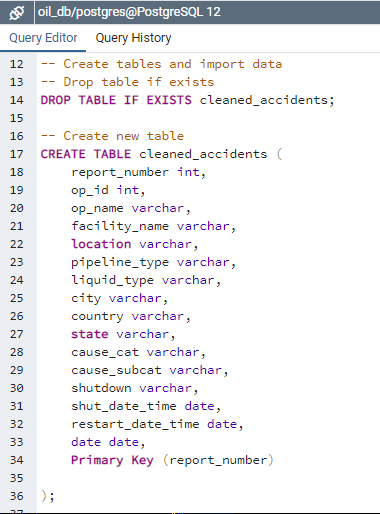
Our dataset has been prepared into two separate tables, which have then been joined to allow for further investigation and manipulation, while maintaining the integrity of each individual data set as a whole.

**Annexures/ Figures**

**Annexure 1 – Cleaned Oil Table Schema**



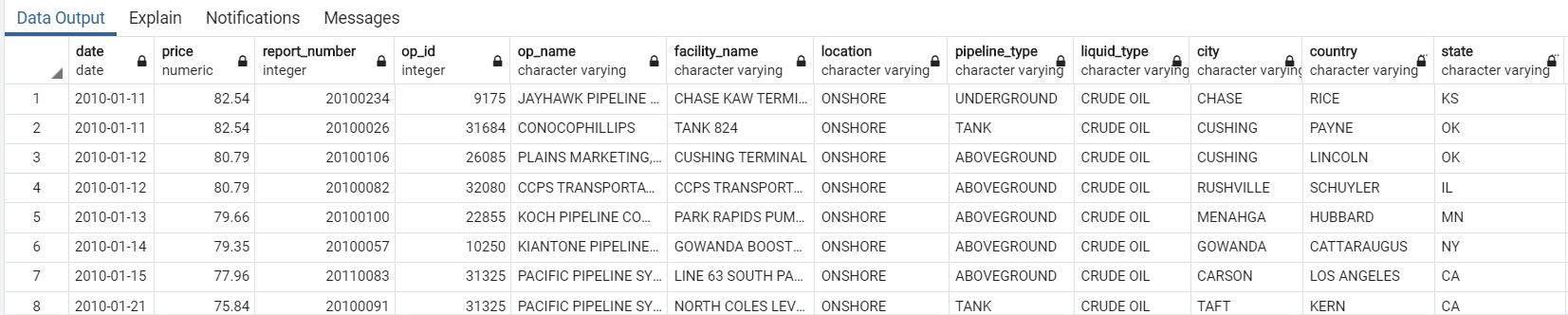
**Annexure 2 – Cleaned Accidents Table Schema**



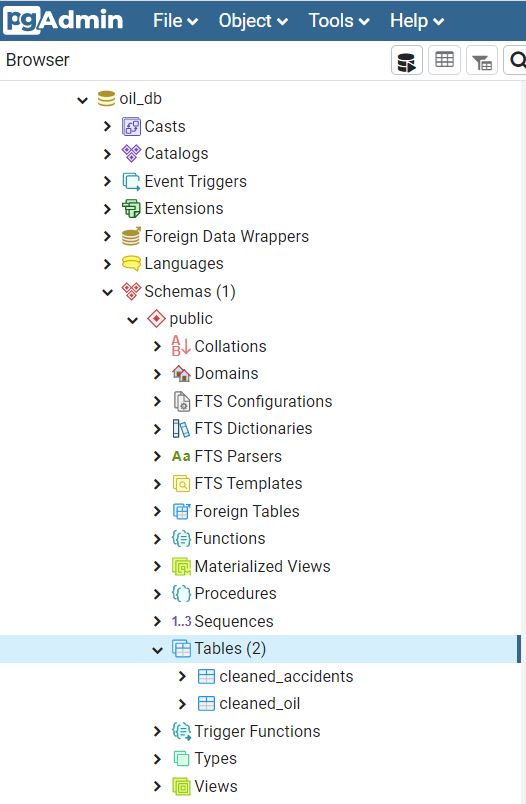
**Annexure 3 – Table Join Query**

****

**Annexure 4 - Joined Tables from Database**

****

**Annexure 5 - Screenshot of Database and Schemas**



**Annexure 6 - Screenshot of Panda’s Merge Code and DataFrame**

