**School of Computing, Engineering and Built Environment**

**Big Data Platforms**

**Level: M**

**Module Code: MMI227050**

**Course work-2**

**Data Pipeline Design Report**

**Made by: Geeta Yadav**

**Student ID: S2120655**

**Student Declaration:** *This Course work is truly made by me.*

Contents

[**1. Overall Concept** 3](#_Toc102499850)

[**1.1 Data** 3](#_Toc102499851)

[**1.2 Transformation of Data** 4](#_Toc102499852)

[**2. Platforms** 7](#_Toc102499853)

[**2.1 Key components** 7](#_Toc102499854)

[**3. Bibliography** 9](#_Toc102499855)

[Figure 1 Designed Data Pipeline 6](#_Toc102499881)

# **1. Overall Concept**

The automated chain processing on data is called the data pipeline. It might be a flow that moves data from point A to point B, or it could be a flow that collects data from numerous sources and transmits it to a data warehouse, or it could be a flow that performs some type of analysis on the data retrieved. Data pipelines exist in a variety of shapes and sizes. They all have three things in common, though: they're automated, they're repeatable, and they're useful for breaking down large jobs into smaller, reusable components.

Extract, Transform and Load are the types of the data pipeline and this process is known as the ELT. Extract, load, and transform are the steps in the ELT pipeline. They're popular pipeline layouts utilized by a wide spectrum of data-driven businesses. Particularly when working with data from various sources that must be stored in a data warehouse. [1]

In this report to design the data pipeline followed the stages which are going to explain further step by step.

## **1.1 Data**

In this stage searched the raw data on the online platform and selected the LIHTC dataset from the website [ <https://lihtc.huduser.gov/> ]. The data downloaded in the CSV form.

HUD generated the LIHTC (Low-Income Housing Tax Credit) dataset (Housing Urban Development). Individual projects' size, mix of housing units, and location are all included in the dataset.

HUD ID (Unique Project ID), project details such as name, address, street number, zip code, city code, and state code, company details such as name, address, telephone number, and company owner name, total number of units and low-income units, number of bedrooms, year the credit was allocated, year the project was allocated, and whether the project was original construction or rehabilitation are all included in the database. These data are also geocoded and distributed globally. From 1987 until 2021, this database offers a list of all housing units covered by HUD's LIHTC program. [2]

**Data Schema**

There are the seventy columns and fifty thousand rows in the LIHTC data. Defining twelve important columns variables schema of the data on which further I will do analysis and visualization. [3]

|  |  |  |
| --- | --- | --- |
| **S.no** | **Variable Name** | **Variable Explanation** |
| 1. | HUD\_ID | It is a unique identity for the database which is the combination of the agency code and year of service placed. |
| 2. | PROJECT | Name of the project |
| 3. | PROJ\_CTY | Name of the city of project |
| 4. | PROJ\_ST | Name of the state of project |
| 5. | N\_UNITS | Total number of units |
| 6. | LI\_UNITS | Total number of Low-income units |
| 7. | N\_1BR | One-bedroom units |
| 8. | N\_2BR | Two number of bedrooms units |
| 9. | N\_3BR | Three number of bedrooms units |
| 10. | N\_4BR | Four Number of bedrooms units |
| 11. | YR\_PIS | Year of service placed |
| 12. | YR\_ALLOC | Year of allocation of the Unit |
| 13. | Type | Type 1 means new construction, type 2 means Acquisition and Rehab, type 3 and Both new construction and A/R 4 mean Existing in the database |

## **1.2 Transformation of Data**

To transformation of the LIHTC data I would consider the basic steps and rules following as:

**1.2.1 The data's reliability**

According to the first rule, I evaluated the authenticity of the data and discovered that many columns, such as project state, FIPS place code, nation codes, state code, and property type, unit digits, and years, have no value.

**1.2.2 Verify the data's accuracy**

I saw that data created with 75 columns and 50567 rows contains Project details, company details, geo detail of area, project unit details, and certain columns that are not mentioned in the schema. To adequately study the data, I decided to divide it into two tables.

**1.2.3 Verify the data's consistency**

There are duplicate names and IDs in the data that need to be sorted out, as well as mix digits and incorrect years of allocation according to the records.

**1.2.4 Check the data consistency**

The information must be presented in an instructive manner, avoiding the repetition of material in rows and columns.

After understanding and considering all the parameters perform the analysis and visualization on the data. As the data contain total seventy-five columns in notebook the below task performed for the transformation.

* **Import the libraries**

To perform any function in Pyspark (Python) it requires to import necessary libraries.

* **Check the number of rows and columns in the data**

In result, it displayed that the data has 75 columns and 50567 rows.

* **View the top 10 rows of the columns** to get the information contains in the column.
* **Drop unnecessary columns which are not required**

Dropping duplicates from the project\_uniqueID column if any. project\_uniqueID column should not have any duplicate in the column so if there is any duplicate in the column it will get drop.

* **Print the schema of the columns**

Print schema show the columns nature weather they are in string form or in Integer form.

* **Fill the null values of data columns**

Fill the null value fill the all columns will if they contain any blanks in the cell.

* **Rename the columns and give the clearer names.**

Giving new names to the columns for the better understanding of columns information.

* **Check and drop Duplicates in the column**
* **Remove the unnecessary rows from the data**

Removing all the rows containing fake data. 8888 number in serviceplaced\_yr giving a data of the units which are not confirmed so it is considering as a fake data.

After transformation visualize the data on the basis on creating some question after analysis of data.

**Visualization of data**

**Question -1 Find the average number of 2-bedrooms units placed in all years?**

Grouped the service placed year with the average of two bedrooms units to check minimum and maximum numbers of units placed in which year. Used the line graph and customized the plot to show the graph. In result, found that that average number of the 4-bedroom units. It can be seen that the minimum number of units placed in year 2022 and the highest number of the units placed in year 2004.

**Question-2. Find the average number of 4-bedrooms units placed in all years?**

The bar graph shows that average number of the 4-bedroom units. It can be seen that the minimum number of units placed in year 2022 and the highest number of the units placed in year 2004

**Question-3. Find the types of the 2-bedroom units.?**

The above Pie chart illustrates the percentage of the types of 2 bedroom units, it can be seen that the maximum fifty eight percent units are of new construction units, Acquisition and rehab types of units has thirty three percent whereas twenty eight percent of units are both new and rehab types of units. Lastly, there are six percent of 2 bedroom units which are not defined in the data.

# **2. Platforms**

**Pipeline Design**

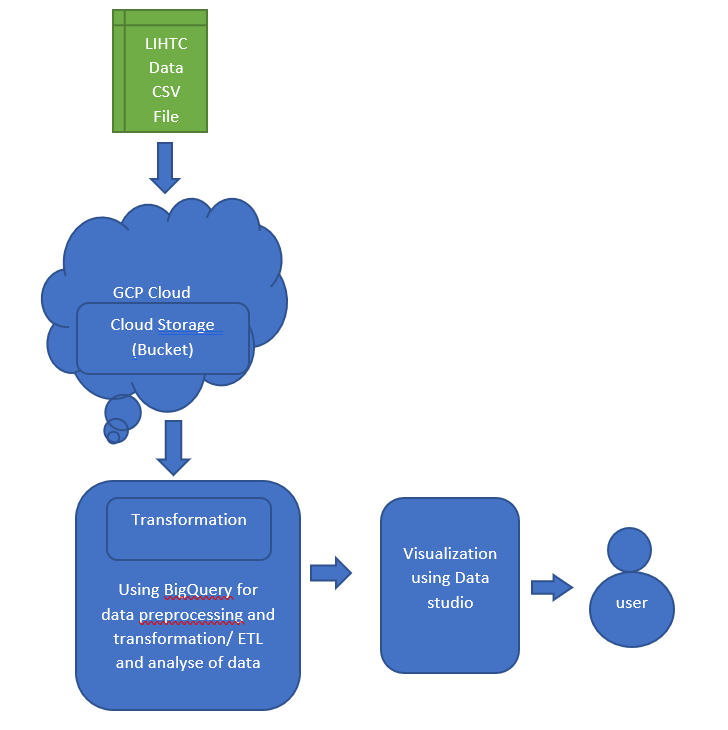


Figure 1 Designed Data Pipeline

## **2.1 Key components**

**2.1.1 Ingest**

LIHTC data csv file stored in local system and used uploaded on the Databricks cloud. Used Databricks Cloud service platform link [ <https://community.cloud.databricks.com/>] to upload the file. Created notebook, given the name to the notebook and select the python language (it is also a default language) and create the cluster.

**2.1.2 Storage**

There are some common types of storages provide by the Google cloud service such as file storage, object storage, archive storage, block storage, and other types for various purposes. Google Cloud storage, where the data is stored called “bucket.

DBFS is the default storage location in DBFS. Databricks use the DBFS is the Databricks File System. It provides a simple storage to store data in Databricks. All who have the access of the workspace of databricks they all can access the stored file. Also, using file semantics and directory you are allow to interact with the object storage. Files are saved to the object storage and that’s why no data loose when any cluster terminate. [4]

**2.1.3 Transformation and Analysis**

In cloud platforms there are different services available to upload and streaming the data. After uploading a file, it extracted for streamed and processed in cloud for the transformation.

In Google Cloud BigQuery ML feature in Google cloud help in preprocessing of data and with the help of simple SQl query functions transform the data. BigQuery automatically applies all the transformation at the time of prediction of data.

In Databricks after create the cluster and extract the csv file use python language which also set as default language to use for data analysis and transformation in cluster notebook. In ELT process in databricks Pyspark read the source file set in the directory and display the data. [4]

**2.1.4 Visualization**

Visualization gives the all required information to the customer related to the data or the information linked to access for the customer. Data visualization also aids in the migration of data from on-premises repositories to cloud storage while minimizing application impact.

The show and display HTML functions in Databricks allow a variety of visualizations out of the box. Databricks also includes native Python and R visualization libraries, as well as the ability to install and use third-party libraries. [4]

# **3. Bibliography**

|  |  |
| --- | --- |
| [1] | Snowflake, "BENEFITS OF A DATA PIPELINE," [Online]. Available: https://www.snowflake.com/guides/data-pipeline#:~:text=A%20data%20pipeline%20is%20a,used%20to%20develop%20business%20insights. [Accessed 02 05 2022]. |
| [2] | O. O. P. DEVELOPMENT. [Online]. Available: https://www.huduser.gov/portal/datasets/lihtc.html. [Accessed 02 05 2022]. |
| [3] | 1.-2. D. D. HUD National LIHTC Database, "HUD National Low Income Housing Tax Credit (LIHTC) Database," [Online]. Available: https://lihtc.huduser.gov/extract/datadict.pdf. [Accessed 02 05 2022]. |
| [4] | databricks. [Online]. Available: https://docs.databricks.com/notebooks/visualizations/index.html. [Accessed 02 05 2022]. |