Apartment and home repairs, infrastructure, & equipment issues  
 by  
 **HomeFront Solutions.org**  
 A logo of a house with tools

Description automatically generated  
 Data Engineer Members (Group7):

1. Reema Gattupalli  
 2. Venkata Siva ram pande Kalivela  
 3. Geetika Sai Manne  
 4. Swarna Neella  
 5. Pravallika Obulapuram  
 6. Keerthi Chandra Vemulamada

**Table of contents**  
  
 1. Abstract  
  
 2. Cloud provider  
  
 3. Data storage  
  
 4. Dataproc

5. Data Preprocessing. (OpenRefine)  
  
 6. Data Management.

**Abstract:**This project is done by the HomeFront Solutions.org data engineering team. They embarked on the journey to find the apartment, house repairs, infrastructure, and equipment issues. The solution that was found was regular inspections and maintenance programs, which help owners and renters keep track of and prevent issues in the house. The team, with the help of ETL transformations, has successfully created datasets that can help data analysts get the desired outputs through further analysis. So, the team used various functions of tools to build a robust infrastructure for data storage, preprocessing, and analysis. So, the team used Google Cloud Platform for the storage, OpenRefine for preprocessing, Dataproc for cluster creation, and Hadoop and Spark exploration. The last phase of the project involves data management. We utilized Spark, Hive, and Google BigQuery for analytics and querying of the data. With the team’s collaborative effort, we have completed the project and also addressed many challenges and ensured the data accuracy in the project. I hope that the data scientists and data analysts can use the data we have provided to create successful solutions for problems related to apartment and home repairs, and infrastructure and equipment issues. This project lays the groundwork for effective data analysis and informed decision-making.

**Cloud provider**

We have various options available for cloud services like AWS, Azure, and GCP. But we have opted Google Cloud Platform. Because Google Cloud Platform is a suite of cloud computing services offered by Google. GCP fits best for our project apartment and home repairs, infrastructure, and equipment issues. First, we have created a new project named “ADTA5240FINALPROJECT” and Below we have provided screenshot references.  
  
  
  
A screenshot of a web page

Description automatically generated  
  
  
  
  
 Reema

**Data storage**  
  
In the Google Cloud platform, we have created the bucket (s24finalproject) and uploaded three static data files. This data storage would be used throughout the project. So, for this project, we have three datasets Louisville, individual and property datasets. Below we have provided screenshots of the available data store location and data we uploaded.   
  
  
  
A screenshot of a web page

Description automatically generated  
  
  
  
 Reema

**Dataproc**We created a cluster named “dp-Hadoop 2-spark-finalproject”, and then we created VM instances in the compute engine. Further, we have explored Hadoop services through SSH-in-browser. We also monitored the cluster details. Below we have provided screenshots for references.

**A screen shot of a computer

Description automatically generated**

pande

**A screenshot of a computer

Description automatically generated  
  
  
   
A screenshot of a computer

Description automatically generated**

pande

We can check the cluster details in this project by clicking on the cluster in the dataproc. This is mainly to keep track of the costs.

**Data Preprocessing**we have used the tool OpenRefine as part of cleaning up the data. We have made several modifications and cleaned up data for further process of the data cycle. Below we have provided the screenshots of OpenRefine.  
  
We removed null values and changed the two columns PRC\_COUN\_DIST and TOTAL\_INSP to numbers from the property dataset.   
  
A screenshot of a computer

Description automatically generated

Geetika

A screenshot of a computer

Description automatically generated

This is the Louisville dataset and we deleted some rows and columns from this dataset and changed the zip code data type from string to number.

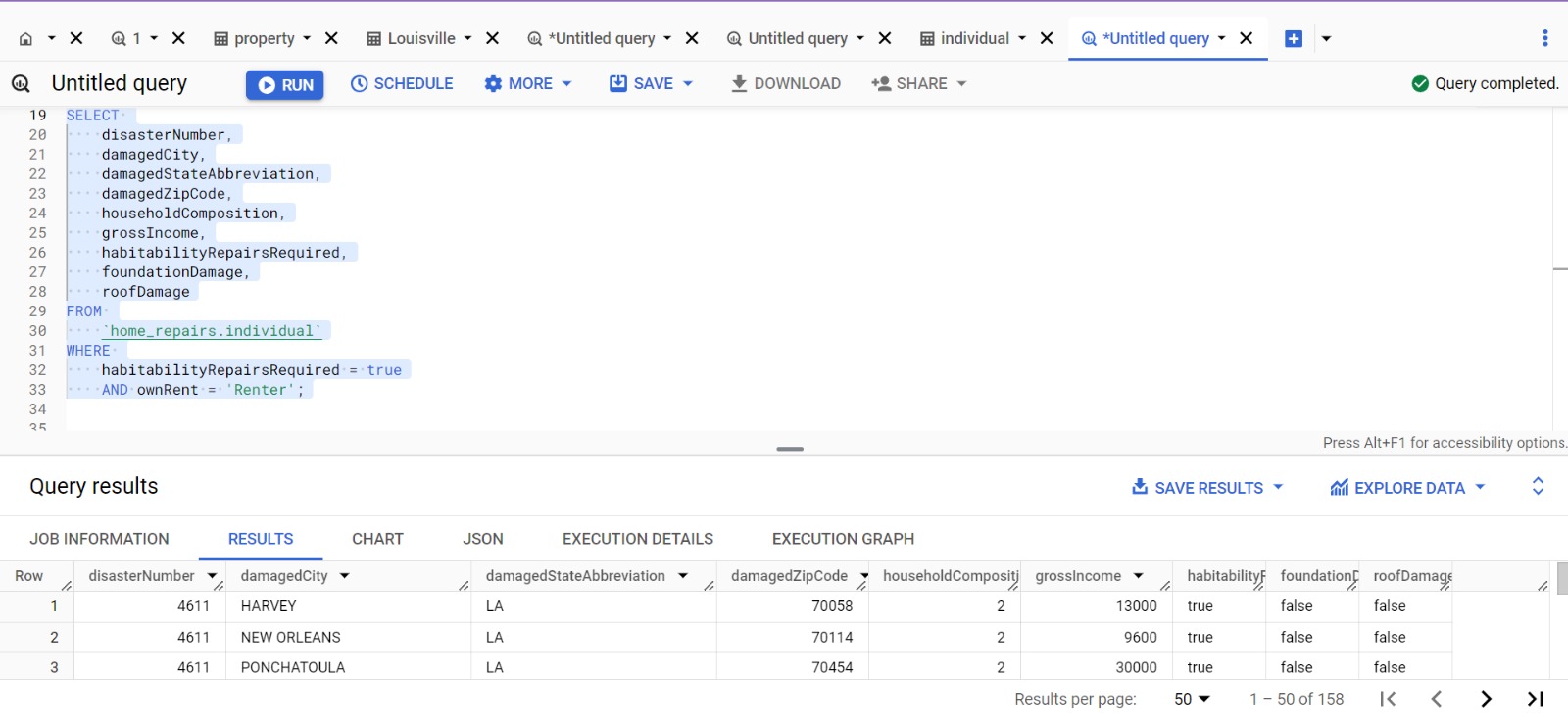
A screenshot of a computer

Description automatically generated

Geetika

In this dataset, considering the individual dataset we changed damagedZipCode and grossincome to number from string and we can see facet of zip codes in the above screenshot.

**Data management**   
  
The final step of our job, as data engineers is to maintain large data sets and do data processing. We used the spark, and hive for responsibilities like batch processing, stream processing, and querying. Apart from that, we used Google BigQuery to manage our massive data sets and processes in the most efficient way possible. Below we have provided the screenshots for references.  
  
This query selects some columns like damaged city, damaged state and zip code and uses the where clause to check if the house requires repairs habitually and if it is owned by the renter and it only selects data of the houses which are owned by the renter



Swarna

Next SQL query selects the first letter of csm\_address calculates an average number of inspections done to the home and records a value greater than 1. At last, we use group by and order by function to csm\_address. Another query selects some columns like damaged city, damaged state, and zip code and uses the where clause to check if the house requires repairs habitually and if it is owned by the renter and it only selects data that has a renter.

A screenshot of a computer

Description automatically generated

Keerthi

The SQL query calculates the average area median income ratio and identifies the maximum household size within each zip code group. It uses a where clause which filters the data that have case closed. We used the group by function to group the data with the zip code. At last, we used the Order Y function for the average AMI ratio in descending order.

A screenshot of a computer

Description automatically generated

Keerthi

We loaded data in the hdfs file system and used Hive SQL to create tables and query the tables. In the below-provided screenshot, we can see how many case IDs are provided in the data set. The time taken to complete this query is 5.79 seconds. Besides that, we also have performed a task in Spark, and the time taken to complete the query is 4.75 seconds, which is comparatively better than Hive.

A screenshot of a computer program

Description automatically generated

Pravallika

A screenshot of a computer screen

Description automatically generated

Pravallika

**Meeting Notes**

Meeting Notes – 04/11/2024  
  
We discussed the outline of the project and assigned tasks to all team members. We have searched for potential data sets.

Meeting Notes – 04/16/2024  
  
Finalized data sets and started with OpenRefine by Geetika. On the same day, we performed data proc and explored Hadoop by Pande & Swarna. Assigned work on BigQuery, Spark, and Hive to remaining teammates.

Meeting Notes – 04/19/2024  
  
Completed all technical work and started doing documentation and PPT presentation.  
  
Meeting Notes – 04/22/2024  
  
We did proofreading of documentation and PPT to make sure everything accurate and prepared for presentation.

**Data Sources and References:  
  
 1. Louisville Metro KY – Home Repair**

Website: <https://hub.arcgis.com/datasets/bee1d8b1ef1c44db82d696bc1d63e369_0/explore>  
2. **Property Maintenance Data**  
website:  
 https://www.phoenixopendata.com/dataset/property-maintenance-data.  
3. **Individual Assistance Housing Registrants - Large Disasters (NEMIS)**  
website:  
https://catalog.data.gov/dataset/individual-assistance-housing-registrants-large-disasters-nemis