

INTRODUCTION

In the era of **big data analytics**, the need to process and derive insights from massive datasets has led to the evolution of sophisticated tools and frameworks. This project revolves around leveraging Apache Hive, a powerful data warehousing and SQL-like query language, to analyze the **"demographic_data.csv" dataset**. By harnessing the distributed computing capabilities of Hive, we aim to efficiently uncover valuable insights from this extensive demographic dataset.

TOOLS AND TECHNOLOGIES

1. HADOOP

Hadoop, an open-source distributed computing framework, enables the processing of large datasets across clusters of computers. Its distributed storage (Hadoop Distributed File System - HDFS) and processing (MapReduce) capabilities make it an ideal solution for handling vast and complex datasets, as is the case with our demographic data.

2. HDFS

HDFS stands for **Hadoop Distributed File System**. HDFS operates as a distributed file system designed to run on commodity hardware. HDFS is fault-tolerant and designed to be deployed on low-cost, commodity hardware.

3. HIVE

Hive, built on top of Hadoop, provides a SQL-like interface, allowing users to query and analyze data stored in Hadoop Distributed File System (HDFS). This makes it an ideal tool for handling large-scale datasets and performing complex analyses.

4. HIVEQL

HiveQL simplifies the process of querying and analyzing large-scale datasets stored in Hadoop Distributed File System (HDFS). Its SQL-like syntax makes it accessible to data analysts and engineers familiar with relational databases, while its underlying distributed processing capabilities enable efficient handling of big data.

5. PYHIVE

PyHive is a Python package that provides a consistent interface for interacting with different Hive backends (such as Apache Hive and Apache Impala) using Python. It acts as a Python client for Hive, allowing users to execute Hive queries and manage Hive databases and tables from within a Python environment.

ABOUT DATASET

> The dataset is from kaggle, a well known online platform that provides users to share and discover datasets .Users can upload, explore, and download datasets for use in their own projects or analysis.

> <https://www.kaggle.com/datasets/anthonytherrien/synthetic-population-demographics-dataset>

> The dataset contains **15 columns** with the following column specifications:

Introducing the Synthetic Demographic Dataset, a large-scale, simulated dataset encompassing **5,000,000 rows**.The "demographic_data.csv" dataset encompasses a multitude of attributes such as **Name , Gender , Country , Age , Income , Educational level , Occupation , Marital Status , Number of Children , Location Type , Health Index , Exercise Frequency , Diet Quality S core , Credit Score , Car Ownership**.

ANALYSIS

Loading data into Hadoop Distributed File System (HDFS) involves several steps.

```
start-all.sh
```

> Create a Directory in HDFS

```
hadoop fs -mkdir /project
```

> Transfer Data to HDFS

```
hadoop fs -put /home/gayathri/data/PROJECT/3/synthetic/demographic_data.csv  
/project (local to hdfs)
```

> Load data to Hive

```
create database project;
```

```
use project;
```

> Create a Table

```
--> create table synthetic_demography(name string,gender string,country string,age  
int,income float,educational_lvl string,occupation string,marital_status  
string,no_of_children int,location_type string,health_index float,exercise_frequency  
float,diet_quality_score float,credit_score float,car_ownership string) row format  
delimited fields terminated by ',' tblproperties('skip.header.line.count'='1');
```

```
--> load data inpath '/project/demographic_data.csv' into table  
synthetic_demography;
```

```
--> select * from synthetic_demography limit 5;
```

Analysis 1

> How many individuals in the dataset are from a specific country, say "United States"? Provide the count

CODE :

```
select synthetic_demography.country,COUNT(*) as individual_count from  
synthetic_demography where country='United States' group by  
synthetic_demography.country;
```

RESULT :

synthetic_demography.country	individual_count
United States	920118

CONCLUSION :

The SQL query retrieves and summarizes demographic data from the "synthetic_demography" table, specifically focusing on the country 'United States.' By counting the number of individuals in the specified country, the query provides valuable insights into the population distribution within the dataset. This information can be crucial for various analytical purposes, such as understanding the representation of the United States in the synthetic demographic dataset. The result, presented in the form of a count of individuals for each unique country, allows for a quick and informative overview of the dataset's composition, particularly with regard to the targeted country of interest.

Analysis 2

> Calculate the average age and income for each country in the dataset

CODE :

```
select synthetic_demography.country,avg(synthetic_demography.age) as  
avg_age,avg(synthetic_demography.income) as avg_income from  
synthetic_demography group by synthetic_demography.country;
```

RESULT :

synthetic_demography.country	avg_age	avg_income
China	37.11748338936813	54.627843838167514
France	37.112501293172365	54.62341575572445
Germany	37.110683687943265	54.554674500327415
Spain	37.174047569934785	54.6245683870477
Turkey	37.118365312733616	54.65459252499209
Canada	37.15003360111469	54.82080955729576
Japan	37.12351954567524	54.5011390557223
United Kingdom	37.11471789810349	54.255773677334155
India	37.10321784530723	54.628982093245405
Italy	37.08997525680438	54.72192005624235
Mexico	37.11756263245739	54.733812251099685
Russia	37.07687060999364	54.6479434435303
United States	37.09659195885745	54.597507566970556

CONCLUSION :

The query aims to provide insights into the demographic characteristics of different countries within the "synthetic_demography" dataset. It calculates and presents the average age (avg_age) and average income (avg_income) for each unique country. This summary allows for a comparative analysis of the demographic attributes, showcasing how the average age and income levels vary across different nations represented in the dataset. **This information can be useful for decision-making, strategic planning, and gaining a deeper understanding of the composition of the synthetic demographic data across various geographic regions.**

Analysis 3

> what are the most common names in the dataset

CODE :

```
select synthetic_demography.name, COUNT(*) as name_count from
synthetic_demography group by synthetic_demography.name order by name_count
DESC limit 10;
```

RESULT :

李秀荣	583
李利	582
王彬	580
王利	577
王桂芳	574
王涛	574
王冬梅	572
王阳	572
王秀华	570
王龙	570

CONCLUSION :

The SQL query is designed to identify and rank the top 10 most frequently occurring names in the "synthetic_demography" dataset. By grouping the data based on the 'name' column and counting the occurrences for each name, the query provides valuable insights into the distribution of names within the dataset. The result is then ordered in descending order by the count of occurrences ('name_count'), and only the top 10 names are displayed. Such analysis could be valuable for various purposes, including profiling, statistical summaries, or understanding the dataset's characteristics related to names.

Analysis 4

> Analyze the income distribution based on educational level and occupation

CODE :

```
select educational_lvl,occupation,avg(income)as average_income,min(income)as min_income,max(income)as max_income,count(income)as num_people from synthetic_demography group by educational_lvl,occupation limit 10;
```

RESULT :

educational_lvl	occupation	average_income	min_income	max_income	num_people
Bachelor's	Administration	54.57221194124922	0.5638	1652.7437	39748
Bachelor's	Arts	54.62971233665854	0.7776	1688.9933	29596
Bachelor's	Finance	54.6756971749841	0.395	1978.0906	60154
Bachelor's	Aviation	55.15629218181459	0.4397	1744.0952	28108
Bachelor's	Construction	54.74352465755888	0.4851	2432.013	53268
Bachelor's	Consulting	55.339350595496555	0.5082	2384.6335	31634
Bachelor's	Agriculture	54.82023740079488	0.5827	1722.9316	33210
Bachelor's	Education	54.282886433110086	0.5946	2956.1904	66581
Bachelor's	Energy	54.0697869026063	0.5047	2813.6833	43129
Bachelor's	Environment	55.0608518623778	0.5234	2275.838	25167

CONCLUSION :

This SQL query provides a summarized view of income statistics across different educational levels and occupations in the "synthetic_demography" dataset. The results include the average income (average_income), minimum income (min_income), maximum income (max_income), and the count of individuals (num_people) for each unique combination of educational level and occupation. By grouping the data based on these two dimensions, the query allows for a nuanced analysis of income variations within distinct educational and occupational categories. In conclusion, this query facilitates a better understanding of income distribution patterns in the synthetic demographic dataset, enabling insights into how income levels correlate with educational attainment and occupation.

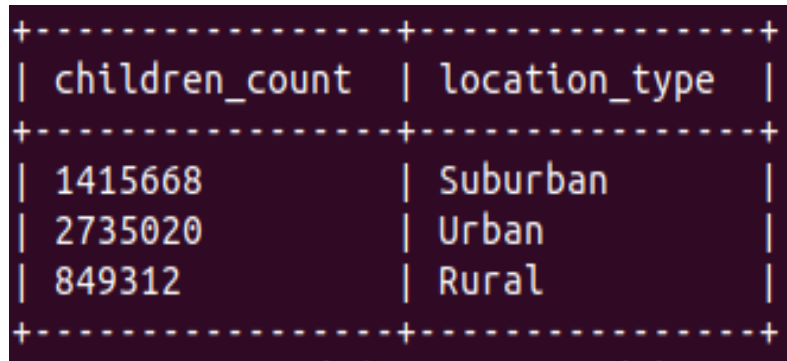
Analysis 5

> How does the location type influence the number of children in a family

CODE :

```
select count(synthetic_demography.no_of_children) as  
children_count,synthetic_demography.location_type as location_type from  
synthetic_demography group by synthetic_demography.location_type limit 10;
```

RESULT :



children_count	location_type
1415668	Suburban
2735020	Urban
849312	Rural

CONCLUSION :

The conclusion might involve interpreting the results to understand how the presence of children varies across different location types. For example, it could help identify if certain types of areas (urban, suburban, rural, etc.) have different patterns in terms of family size or the number of children. Analyzing and summarizing the data in this way can be valuable for demographic studies, urban planning, or any area where understanding the distribution of family sizes in different location types is important.

Analysis 6 – PARTITION

In Hive, partitions are a way to organize data in a table based on one or more columns, known as partition columns. Partitioning is particularly useful for improving query performance, as it allows Hive to skip unnecessary data when reading or querying specific partitions. It can significantly reduce the amount of data that needs to be processed, leading to faster query execution. And also here we used DYNAMIC PARTITION. Dynamic partitioning in Hive

allows you to automatically determine the partition values based on the data being inserted into a partitioned table. This feature is particularly useful when you want to insert data into a table with multiple partition columns, and you don't want to explicitly specify the partition values.

> Partition the dataset based on the country column

CODE :

```
create table dynamic_part(name string,gender string,age int,income
float,educational_lvl string,occupation string,marital_status string,no_of_children
int,location_type string,health_index float,exercise_frequency
float,diet_quality_score float,credit_score float,car_ownership string) partitioned by
(country string)row format delimited fields terminated by ',';
```

```
set hive.exec.dynamic.partition=true;
set hive.exec.dynamic.partition.mode=nonstrict;
```

```
insert into table dynamic_part partition(country) select
name,gender,age,income,educational_lvl,occupation,marital_status,no_of_children,
location_type,health_index,exercise_frequency,diet_quality_score,credit_score,car_
ownership,country from synthetic_demography;
```

RESULT :

dynamic_part.name	dynamic_part.gender	dynamic_part.age	dynamic_part.income	dynamic_part.educational_lvl	dynamic_part.occupation	dynamic_part.marital_status	dynamic_part.no_of_children	dynamic_part.location_type	dynamic_part.health_index	dynamic_part.exercise_frequency	dynamic_part.diet_quality_score	dynamic_part.credit_score	dynamic_part.car_ownership	dynamic_part.country
Michael Leonard	Male	21	10.5804	Not Finish Highschool	Government	Single	1	Urban	55.4899	1.2437	7.5188	609.2955	No	Canada
Chad Robinson PhD	Male	39	10.5148	High School	Finance	Married	1	Urban	33.5786	1.0446	8.3726	605.3831	Yes	Canada
Jessica Gordon	Female	32	11.8959	Bachelor's	Government	Married	2	Urban	34.0611	0.9067	7.3929	598.8132	Yes	Canada
Leslie Clark	Female	35	27.2345	Not Finish Highschool	Education	Divorced	1	Urban	48.9071	0.8317	4.5567	658.2346	Yes	Canada
Jennifer Wagner	Female	40	5.0629	Not Finish Highschool	HR	Single	1	Urban	58.3708	0.0057	8.3032	727.354	No	Canada

CONCLUSION :

It creates a partitioned Hive table named dynamic_part. Dynamic partitioning is configured to allow Hive to determine partition values automatically. Data from the "synthetic_demography" dataset is inserted into the dynamic_part table, with dynamic partitioning based on the country column. This setup is useful when you want to organize your data based on the country column, and dynamic partitioning streamlines the process by automatically determining the partition values during data insertion.

Analysis 7 – BUCKETING

In Apache Hive, bucketing is a technique used to improve the performance of certain queries by organizing data into a set number of buckets or partitions based on a specific column or columns. This helps in distributing the data more evenly and can be particularly beneficial for join operations. bucketing in Hive is a

strategy to organize and distribute data across a specified number of buckets based on certain columns. When used wisely, it can improve the performance of certain queries, especially those involving joins, by minimizing data scans and reducing the overall processing load.

> Bucketing the dataset based on the location_type column

CODE :

1. When creating a table in Hive, you can specify the bucketing column(s) and the number of buckets using the CLUSTERED BY clause. 'location_type' is the column based on which the data is bucketed, and the table is divided into 3 buckets.

```
>> create table bkt_part(name string,gender string,country string,age int,income float,educational_lvl string,occupation string,marital_status string,no_of_children int,location_type string,health_index float,exercise_frequency float,diet_quality_score float,credit_score float,car_ownership string) clustered by (location_type) into 3 buckets row format delimited fields terminated by ',';
```

2. After creating the table, you can load data into it using the 'INSERT INTO TABLE' statement. Hive will automatically distribute the data into the specified number of buckets based on the clustering column.

```
>> insert into bkt_part select * from synthetic_demography;
```

3. DESCRIBE FORMATTED : In Hive, the DESCRIBE FORMATTED command is used to retrieve detailed information about a table, including its metadata, storage properties, and other related details. When applied to a table with bucketing, this command provides information about the bucketing configuration and statistics.

Here are some key sections related to bucketing that you might find in the output:

Table Information:

- **Table type (MANAGED_TABLE, EXTERNAL_TABLE)**
- **Table parameters, including bucketing-related parameters**

Bucketing Information

- **Bucket columns:** Specifies the columns based on which the data is bucketed.
- **Num Buckets:** Indicates the number of buckets configured for the table.

Storage Information:

- **Storage descriptor properties, including bucketing-related properties.**

Table Statistics:









- **The output may include statistics about the table, including the number of files, the total size, and the average file size for each bucket.**

Location:

- **The location where the table data is stored, including the directories for each bucket.**

>> describe formatted bkt_part;

RESULT :

	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name	
	-rwxr-xr-x	gayathri	supergroup	453.62 MB	Feb 05 16:24	1	128 MB	000000_0	
	-rwxr-xr-x	gayathri	supergroup	0 B	Feb 05 16:24	1	128 MB	000001_0	
	-rwxr-xr-x	gayathri	supergroup	91.98 MB	Feb 05 16:24	1	128 MB	000002_0	

Showing 1 to 3 of 3 entries

Previous 1 Next

Analysis 8 – PLOTTING

MATPLOTLIB : Matplotlib is a powerful and widely-used 2D plotting library for the Python programming language. It provides a flexible and comprehensive set of tools for creating static, animated, and interactive visualizations in Python. Matplotlib is often used for creating a wide range of plots and charts, from simple line plots to complex heatmaps and 3D visualizations.

CODE :

```
from pyhive import hive

c=hive.connect(host='localhost',database='projects').cursor()

c.execute('select count(synthetic_demography.no_of_children) as
children_count,synthetic_demography.location_type as location_type from
synthetic_demography group by synthetic_demography.location_type order by
children_count DESC')

a=c.fetchall()

print(a)

children_count=[]

loc_type=[]

for i in a:

    children_count.append(i[0])
```

```
loc_type.append(i[1])
```

```
print(children_count)
```

```
print(loc_type)
```

```
import matplotlib.pyplot as plt
```

```
plt.xlabel('loc_typ')
```

```
plt.ylabel('child_cnt')
```

```
plt.title('type_influence')
```

```
plt.bar(loc_type,children_count)
```

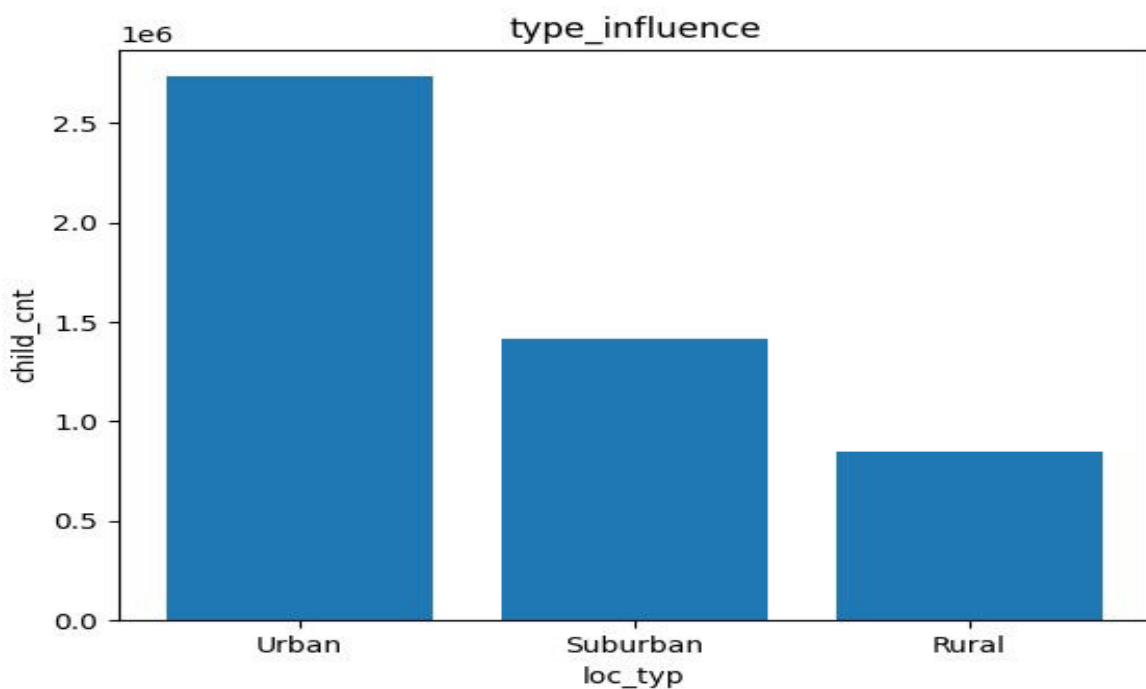
```
plt.show()
```

RESULT :

```
[(2735020, 'Urban'), (1415668, 'Suburban'), (849312, 'Rural')]
```

```
[2735020, 1415668, 849312]
```

```
['Urban', 'Suburban', 'Rural']
```



CONCLUSION

The analysis provides insights into the synthetic demographic data using Hadoop, Hive, HiveQL, HDFS, and PyHive. The conclusions drawn from the analysis can be summarized as follows:

1. **Geographic Distribution:** The dataset contains a specific count of individuals from the United States.
2. **Country-level Insights:** The average age and income for each country in the dataset have been calculated, providing a country-wise overview.
3. **Name Popularity:** The analysis reveals the most common names in the dataset based on the name count.
4. **Income Distribution:** The income distribution has been analyzed based on educational level and occupation, offering insights into earning patterns.
5. **Family Dynamics:** The influence of location type on the number of children in a family has been explored, providing insights into family demographics.
6. **Data Partitioning:** The dataset has been partitioned based on the country column, enabling efficient data retrieval for specific countries.
7. **Data Bucketing:** The dataset has been bucketed based on the location_type column, which can enhance certain types of query performance.
8. **Visualization:** A bar plot using PyHive demonstrates the average number of children by location type, providing a visual representation of the relationship.

This analysis leverages Hadoop ecosystem tools to gain valuable insights into synthetic demographic data, showcasing the power of distributed data processing and analytics.
