USA REAL ESTATE -POSTGRESQL ANALYTICAL

ISM6218 - ADVANCED DATABASE MANAGEMENT

FINAL PROJECT

MEMBERS

ROHIT DHAWALE- (U87947936)

PROFESSOR

CLINTON DANIEL

a. <u>A detailed description of the DBMS</u>**: POSTGRESQL OVERVIEW**

PostgreSQL, often simply referred to as Postgres, is an open-source relational database management system (RDBMS) known for its robustness, extensibility, and adherence to SQL standards.

1. **ACID Compliance**: To ensure the integrity of transactions, PostgreSQL is based upon ACID properties such as Atomicity, Consistency, Isolation and Durability.

2. **Indexing:** Provides various indexing methods, including B-tree, Hash, GIN (Generalized Inverted Index), and GiST (Generalized Search Tree), optimizing query performance.

3. **Security Features:** Implements role-based access control, SSL certificates for encrypted communication, and supports LDAP and Kerberos authentication.

4. **Foreign Data Wrappers:** Allows integration with other data sources through Foreign Data Wrappers (FDWs), enabling PostgreSQL to interact with external databases.

5. **Scalability:** Scales well both vertically and horizontally, making it suitable for small projects to large enterprise applications.

6. **Community and Support:** PostgreSQL has a vibrant and active community that contributes to its development. It also has extensive documentation and community support.

7. **Open Source:** Released under the PostgreSQL License, PostgreSQL is free and open-source software, allowing users to modify and distribute it.

In summary, PostgreSQL is a powerful and extensible RDBMS that combines the benefits of traditional relational databases with modern features, making it a popular choice for a wide range of applications.

PostgreSQL indeed supports a rich set of data types, including some unique ones not commonly found in other popular database management systems. Here are five notable data types, including geometrics:

1. **Geometry and Geography:**

    PostgreSQL has native support for geometric and geographic data types through the **geometry** and **geography** types. This is particularly useful for applications dealing with spatial data, such as mapping and geographic information systems (GIS).

2. **Array:**

    PostgreSQL allows you to define arrays for various data types, including numbers, text, and even custom types. This flexibility is handy when dealing with collections of values.

3. **Hstore:**

    The **hstore** data type in PostgreSQL allows you to store sets of key-value pairs within a single column. This is different from a JSON or JSONB column, as it is specifically designed for simple key-value pairs.

4. **UUID (Universally Unique Identifier):**

    While some other databases support UUIDs, PostgreSQL has a native **uuid** data type that simplifies the storage and manipulation of UUIDs. This is particularly useful for generating unique identifiers.

5. **Enumerated Types:**

    PostgreSQL supports enumerated types, allowing you to create a static, ordered set of values. This can improve data integrity by restricting the possible values a column can hold.

These data types showcase PostgreSQL's commitment to flexibility and extensibility, enabling developers to work with a diverse range of data in a structured manner.

b. **DATASET**

USA REAL ESTATE DATASET:

- This dataset contains Real Estate listings in the US broken by State and zip code.
- It has 10 columns with Property sale status as it is available for sale or ready to build.
- Nature of the property like Number of bedrooms, Number of bathrooms, Land size in acres and living space in square feet
- Address for the property such as city, state, and zip code
- Other columns such as Previously sold date and price.
- Link: USA Real Estate Dataset (kaggle.com)

RANDOM USA PEOPLE DATASET:

- This dataset contains data for people in USA
- It has 16 columns with contact details such as contact phone, Email, additional email.
- Address for people broken into Address, city, state, Zip code and country.
- Other personal data such as prefix, birthdate, and Name
- Additional columns such as Year, Time, link, Text and SSN
- Link: Random USA People Dataset (kaggle.com)

c. <u>A DETAILED DESCRIPTION OF THE PRODUCT</u>

The "Product" dataset in a PostgreSQL analytical database is a structured repository of information related to buyers and properties, encompassing attributes such as "Buyer ID," "Property ID," "Name," "Email," and various other relevant details. This dataset is designed for comprehensive data analysis, reporting, and business intelligence, supporting tasks like buyer behavior analysis, property portfolio management, and buyer-seller relationships assessment. The data is typically well-organized, normalized, and optimized for efficient querying, serving as a foundational resource for deriving insights, creating visualizations, and facilitating data-driven decision-making for businesses and organizations. Data quality, security, and access control are integral for safeguarding the dataset's integrity and confidentiality.

Creating a real estate database with customer analytics is a comprehensive project that involves designing a database schema, implementing data structures, and developing analytical queries. Here's an overview of the project.

Objective:

- Develop a robust real estate database to store information about properties,buyers and transactions.
- Implement analytical features to gain insights into customer behavior, sales trends, and property performance.

Database Schema:

Tables:

- Properties: Store details about each property, including address, type, size, features, etc.
- Buyers: Capture information about buyers, including demographics and contact details.

Detailed description of why the design of the product makes it Analytical.

The database described above is designed to be analytical because it goes beyond merely storing data; it provides the tools and structure needed to analyze and derive insights from that data. Let's break down the key features that make this database analytical:

Structured Data for Analysis:

The database is structured to store specific information about properties and buyers. This structured format makes it easy to perform analytical operations on the data.

Relational Database Design

The use of tables and relationships (as indicated by the foreign key) enables the database to handle complex queries that involve joining and aggregating data from multiple tables. This is crucial for analytical tasks.

Analytical Queries:

The inclusion of analytical features involves the creation of queries that go beyond simple data retrieval. These queries provide insights into patterns and trends within the data.

Conclusion: Hence it makes sense to make our design related to analytical.

d. A DETAILED DESCRIPTION OF THE PRODUCT DATA STRUCTURES

1. The database stores the data of properties and buyers with details of information of properties and buyers.
2. The database contains two tables.
   - Buyers table
   - Properties table

➢ **BUYERS TABLE**

1. BuyerID	Primary key and unique id for customer table,
2. Email_Id	Email of the customer,
3. Prefix	Prefix of the customer,
4. Name	Name of the customer
5. Birth_Date	Birthday of customer
6. Phone_Number	Phone number of customer
7. Additional_Email_Id	Email id of customer
8. Address	Address of customer
9. Zip_Code	zip code of customer
10. City	City of customer
11. State	State of customer
12. Country	Country of customer
13. Year	Year of sale
14. Time	Time of sale
15. Link	Link to the website
16. Text	Related Text
17. SSN	Social security number of buyer

➢ **PROPERTY TABLE**

1. Property id	Primary key and unique id for customer table
2. Status	Status of property sold or ready to build
3. Bed	Number of bedrooms
4. Bath	Number of Bathrooms
5. Acre_lot	Area in acre
6. City	The city in which property is located
7. State	The State in which property is located
8. zip_code	Zipcode of address of property
9. House_size	Size of property
10. Prev_sold_date	Date of the property sold
11. Price	Price of the property

- **RELATIONSHIP**

  A buyer can purchase more than one properties but one property can only be assigned to one buyer.

  The relationship between Customer and Property is a one-to-many relationship. This means one customer can buy multiple properties.

  The relationship between Customer and Property is a many-to-one relationship. one property can only be associated with one customer.

e. DETAILED DESCRIPTION OF THE ETL PROCESS

The process of extracting, transforming, and loading (ETL) data from staging tables into final dimension (DimBuyer and DimProp) and fact (Fact) tables. Below is a detailed description of the ETL process:

1. **Create Dimension Tables:**

   - Two dimension tables, `DimBuyer` and `DimProp`, are created to store information about buyers and properties, respectively.

   - Above tables have attributes such as email, name, address, city, state, etc., that will be used for analysis.

2. **Create Fact Table:**

   - The `Fact` table is created to store the main data where facts or measures are recorded. To store information related to properties (bed, bath, acre_lot, etc.) and their relationships with buyers and properties through foreign keys (`BuyerID` and `PropertyID`).

3. **Create Staging Table:**

   - A `Staging` table is created to temporarily hold data extracted from the source system. This is a common practice in ETL processes to perform data transformations before loading them into the final tables.

4. **Load Data into Staging:**

   - Data is loaded into the `Staging` table using an `INSERT INTO` statement with a `SELECT` query from the source table properties and buyers.

5. **Populate Dimension Tables:**

   - Data from the `Staging` table is transformed and inserted into the dimension tables (`DimProp` and `DimBuyer`). It is mapping and transforming data from the source to the destination tables.

6. **Update Staging Table with Foreign Keys:**

   - The `Staging` table is updated with foreign key values (`BuyerID` and `PropertyID`) that link the staging data to the corresponding dimension tables (`DimBuyer` and `DimProp`).

7. **<u>Update Staging Table with Buyer Details:</u>**

   - The `Staging` table is updated with additional attributes (`Year`, `Time`, `SSN`) obtained from the `Buyers` table, where `BuyerID` serves as the link between the tables.

8. **<u>Insert Data into Fact Table:</u>**

   - Finally, data from the `Staging` table is inserted into the `Fact` table. This table hold detailed property-related information, including relationships with buyers and properties through foreign keys.

<u>Business Questions:</u>

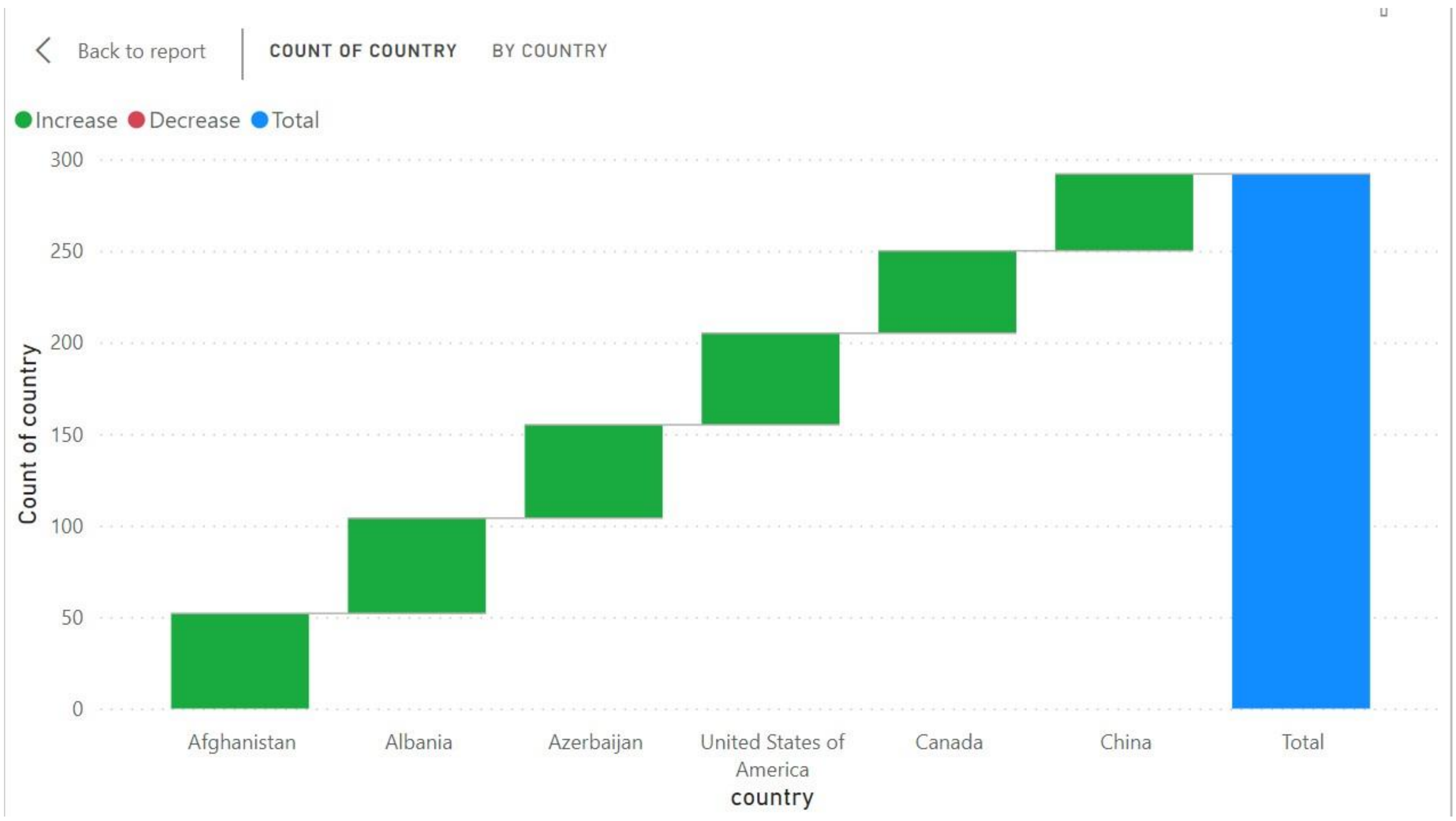
1.What is the count of countries in the Power BI visualization, including Afghanistan, Albania, United States of America, Canada, China, and Azerbaijan, and how do their counts compare?

SELECT Country, COUNT(*) AS Count

FROM public.buyers

WHERE Country IN ('Afghanistan', 'Albania', 'United States of America', 'Canada', 'China', 'Azerbaijan')

GROUP BY Country;



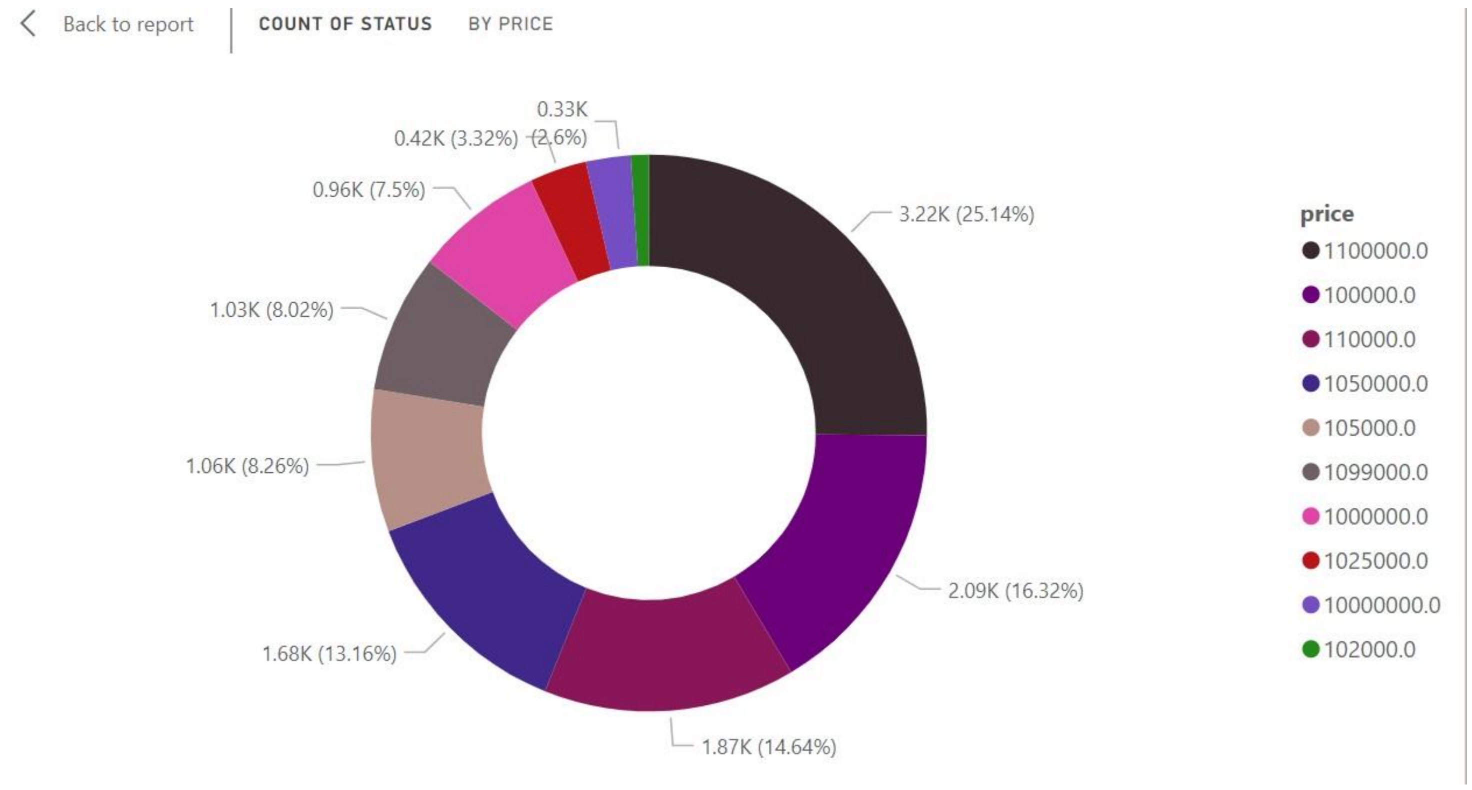
2. What is the count of different property prices in the 'properties' dataset, and what are the counts for the specific property prices: $1,100,000.0, $100,000.0, $110,000.0, $1,050,000.0, $105,000.0, $1,099,000.0, $1,000,000.0, $1,025,000.0, $10,000,000.0, and $102,000.0?

SELECT price, COUNT(*) AS Count

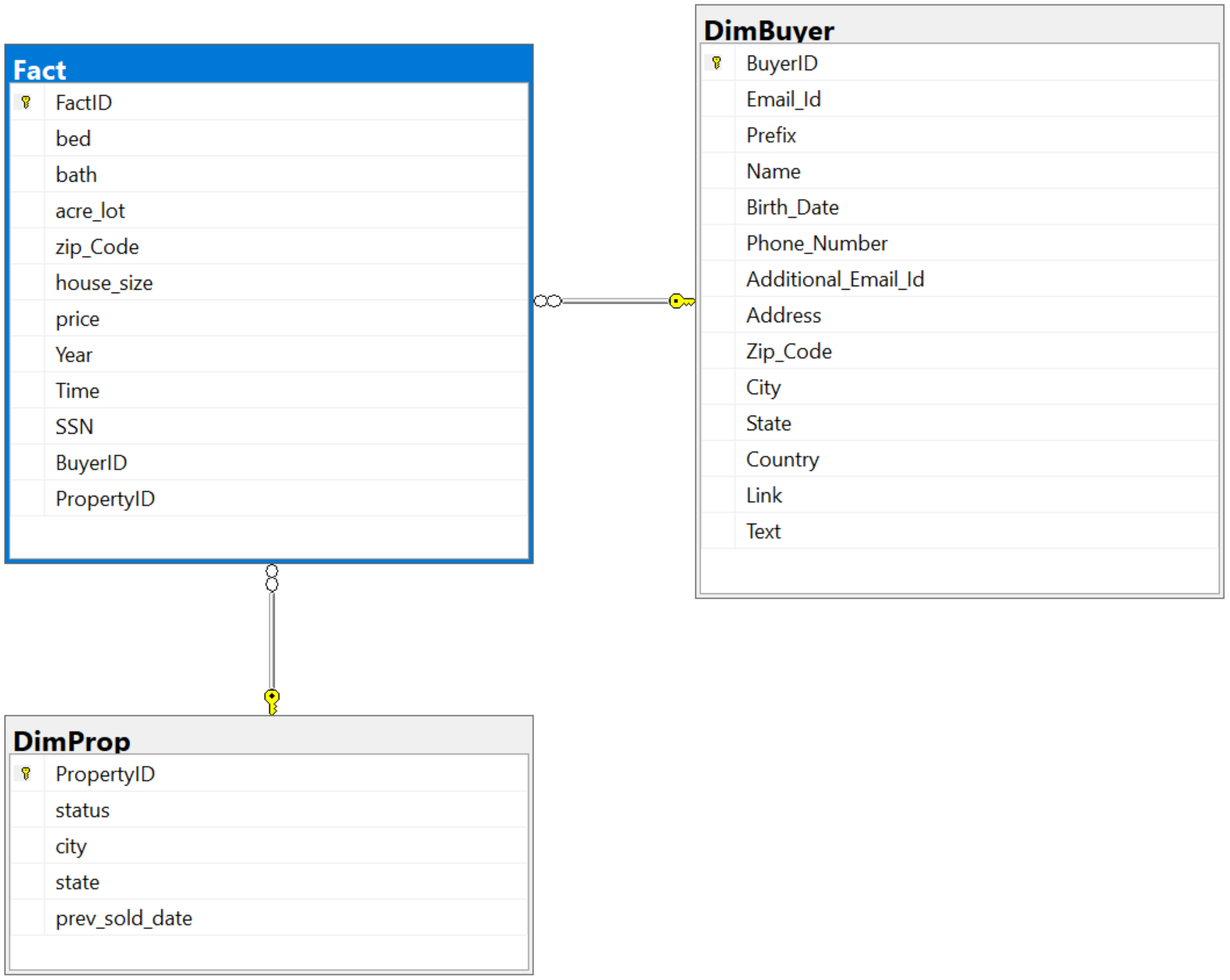
FROM properties

WHERE price::NUMERIC IN (1100000.0, 100000.0, 110000.0, 1050000.0, 105000.0, 1099000.0, 1000000.0, 1025000.0, 10000000.0, 102000.0)

GROUP BY price;



ERD DIAGRAM:


Fact
FactID
bed
bath
acre_lot
zip_Code
house_size
price
Year
Time
SSN
BuyerID
PropertyID
DimBuyer
BuyerID
Email_Id
Prefix
Name
Birth_Date
Phone_Number
Additional_Email_Id
Address
Zip_Code
City
State
Country
Link
Text
DimProp
PropertyID
status
city
state
prev_sold_date