Hive Assignment 2

Data Analysis with Apache Hive on Telecom Data

1. Data Loading (Beginner)
    a. Download the dataset and load it into a Hive table.

```
CREATE TABLE telecom_data (
customerID STRING,
gender STRING,
SeniorCitizen INT,
Partner STRING,
Dependents STRING,
tenure INT,
PhoneService STRING,
MultipleLines STRING,
InternetService STRING,
OnlineSecurity STRING,
OnlineBackup STRING,
DeviceProtection STRING,
TechSupport STRING,
StreamingTV STRING,
StreamingMovies STRING,
Contract STRING,
PaperlessBilling STRING,
PaymentMethod STRING,
MonthlyCharges FLOAT,
TotalCharges FLOAT,
Churn STRING,
Age INT,
Occupation STRING,
Income FLOAT
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;
```

    b. Write a query to display the top 10 rows of the table.

```
SELECT * FROM telecom_data LIMIT 10;
```

2. Data Exploration (Beginner)

a. Write a HiveQL query to find the total number of customers in the dataset.

```
SELECT COUNT(*) FROM telecom_data;
```

b. Write a HiveQL query to find the number of customers who have churned.

```
SELECT COUNT(*) FROM telecom_data WHERE Churn = 'Yes';
```

c. Analyze the distribution of customers based on gender and SeniorCitizen status.

```
SELECT gender, SeniorCitizen, COUNT(*)
FROM telecom_data
GROUP BY gender, SeniorCitizen;
```

d. Determine the total charge to the company due to churned customers.

```
SELECT SUM(TotalCharges)
FROM telecom_data
WHERE Churn = 'Yes';
```

3. Data Analysis (Intermediate)

a. Write a HiveQL query to find the number of customers who have churned, grouped by their Contract type.

```
SELECT Contract, COUNT(*)
FROM telecom_data
WHERE Churn = 'Yes'
GROUP BY Contract;
```

b. Write a HiveQL query to find the average MonthlyCharges for customers who have churned vs those who have not.

```
SELECT Churn, AVG(MonthlyCharges)
FROM telecom_data
GROUP BY Churn;
```

c. Determine the maximum, minimum, and average tenure of the customers.

```
SELECT MAX(tenure), MIN(tenure), AVG(tenure)
FROM telecom_data;
```

d. Find out which PaymentMethod is most popular among customers.

```
SELECT PaymentMethod, COUNT(*)
FROM telecom_data
GROUP BY PaymentMethod
ORDER BY COUNT(*) DESC LIMIT 1;
```

e. Analyze the relationship between PaperlessBilling and churn rate.

```
SELECT PaperlessBilling, COUNT(*)
FROM telecom_data
WHERE Churn = 'Yes'
GROUP BY PaperlessBilling;
```

4. Partitioning (Intermediate)

a. Create a partitioned table by Contract and load the data from the original table.

```
CREATE TABLE telecom_data_partitioned ( /* fields except
Contract */ )
PARTITIONED BY (Contract STRING);

INSERT OVERWRITE TABLE telecom_data_partitioned
PARTITION(Contract)
SELECT * FROM telecom_data;
```

b. Write a HiveQL query to find the number of customers who have churned in each Contract type using the partitioned table.

```
SELECT Contract, COUNT(*)
FROM telecom_data_partitioned
WHERE Churn = 'Yes'
GROUP BY Contract;
```

c. Find the average MonthlyCharges for each type of Contract using the partitioned table.

```
SELECT Contract, AVG(MonthlyCharges)
FROM telecom_data_partitioned
GROUP BY Contract;
```

d. Determine the maximum tenure in each Contract type partition.

```
SELECT Contract, MAX(tenure)
```

```
FROM telecom_data_partitioned
GROUP BY Contract;
```

5. Bucketing (Advanced)

a. Create a bucketed table by tenure into 6 buckets.

```
CREATE TABLE telecom_data_bucketed ( /* fields except tenure */ )
CLUSTERED BY (tenure) INTO 6 BUCKETS;
```

b. Load the data from the original table into the bucketed table.

```
INSERT OVERWRITE TABLE telecom_data_bucketed
SELECT * FROM telecom_data;
```

c. Write a HiveQL query to find the average MonthlyCharges for customers in each bucket.

```
SELECT AVG(MonthlyCharges)
FROM telecom_data_bucketed
WHERE tenure BETWEEN lower_range AND upper_range;
```

d. Find the highest TotalCharges in each tenure bucket.

```
SELECT 'Bucket 1' AS Bucket, MAX(TotalCharges) as MaxCharge
FROM telecom_data_bucketed
WHERE tenure BETWEEN 1 AND 10
UNION ALL
SELECT 'Bucket 2' AS Bucket, MAX(TotalCharges) as MaxCharge
FROM telecom_data_bucketed
WHERE tenure BETWEEN 11 AND 20
UNION ALL
SELECT 'Bucket 3' AS Bucket, MAX(TotalCharges) as MaxCharge
FROM telecom_data_bucketed
WHERE tenure BETWEEN 21 AND 30
UNION ALL
```

```
        SELECT 'Bucket 4' AS Bucket, MAX(TotalCharges) as
MaxCharge
        FROM telecom_data_bucketed
        WHERE tenure BETWEEN 31 AND 40
        UNION ALL
        SELECT 'Bucket 5' AS Bucket, MAX(TotalCharges) as
MaxCharge
        FROM telecom_data_bucketed
        WHERE tenure BETWEEN 41 AND 50
        UNION ALL
        SELECT 'Bucket 6' AS Bucket, MAX(TotalCharges) as
MaxCharge
        FROM telecom_data_bucketed
        WHERE tenure BETWEEN 51 AND 60;
```

6. Performance Optimization with Joins (Advanced)

Assume another dataset, CustomerDemographics.csv, that contains the details of the demographic data of each customer.

a. Load the demographics dataset into another Hive table.

```
        CREATE TABLE demographic_data (
        /* Assume fields based on the actual demographic dataset you
have */
        );

        LOAD DATA INPATH 'path/to/demographic_data.csv'
        INTO TABLE demographic_data;
```

b. Write HiveQL queries to join the customer churn table and the demographics table on customerID using different types of joins - common join, map join, bucket map join, and sorted merge bucket join.

```
        SELECT /* select fields you are interested in */
        FROM telecom_data t
        JOIN demographic_data d ON t.customerID = d.customerID;

        SELECT /* select fields you are interested in */
        FROM telecom_data t
```

```
        JOIN /*+MAPJOIN(d)*/ demographic_data d ON
    t.customerID = d.customerID;
```

c. Observe and document the performance of each join type.

7. Advanced Analysis (Expert)

a. Find the distribution of PaymentMethod among churned customers.

```
SELECT PaymentMethod, COUNT(*)
FROM telecom_data
WHERE Churn = 'Yes'
GROUP BY PaymentMethod;
```

b. Calculate the churn rate (percentage of customers who left) for each InternetService category.

```
SELECT InternetService, COUNT(*)*100.0/(SELECT COUNT(*)
FROM telecom_data WHERE Churn = 'Yes') AS churn_rate
  FROM telecom_data
  WHERE Churn = 'Yes'
  GROUP BY InternetService;
```

c. Find the number of customers who have no dependents and have churned, grouped by Contract type.

```
SELECT Contract, COUNT(*)
FROM telecom_data
WHERE Churn = 'Yes' AND Dependents = 'No'
GROUP BY Contract;
```

d. Find the top 5 tenure lengths that have the highest churn rates.

```
SELECT tenure, COUNT(*)*100.0/(SELECT COUNT(*) FROM
telecom_data WHERE Churn = 'Yes') AS churn_rate
FROM telecom_data
WHERE Churn = 'Yes'
GROUP BY tenure
ORDER BY churn_rate DESC
LIMIT 5;
```

e. Calculate the average MonthlyCharges for customers who have PhoneService and have churned, grouped by Contract type.

```
SELECT AVG(MonthlyCharges)
FROM telecom_data
```

```
WHERE PhoneService = 'Yes' AND Churn = 'Yes';
```

f. Identify which InternetService type is most associated with churned customers.

```
SELECT InternetService, COUNT(*)
FROM telecom_data
WHERE Churn = 'Yes'
GROUP BY InternetService
ORDER BY COUNT(*) DESC
LIMIT 1;
```

g. Determine if customers with a partner have a lower churn rate compared to those without.

```
SELECT Partner, COUNT(*)*100.0/(SELECT COUNT(*) FROM
telecom_data WHERE Partner = 'Yes') AS
churn_rate_with_partner
FROM telecom_data
WHERE Churn = 'Yes' AND Partner = 'Yes'
UNION ALL
SELECT Partner, COUNT(*)*100.0/(SELECT COUNT(*) FROM
telecom_data WHERE Partner = 'No') AS
churn_rate_without_partner
FROM telecom_data
WHERE Churn = 'Yes' AND Partner = 'No';
```

h. Analyze the relationship between MultipleLines and churn rate.

```
SELECT MultipleLines, COUNT(*)*100.0/(SELECT COUNT(*)
FROM telecom_data WHERE MultipleLines = 'Yes') AS
churn_rate_with_multiplelines
FROM telecom_data
WHERE Churn = 'Yes' AND MultipleLines = 'Yes'
UNION ALL
SELECT MultipleLines, COUNT(*)*100.0/(SELECT COUNT(*)
FROM telecom_data WHERE MultipleLines = 'No') AS
churn_rate_without_multiplelines
FROM telecom_data
WHERE Churn = 'Yes' AND MultipleLines = 'No';
```