INFORMATICS INSTITUTE OF TECHNOLOGY

In Collaboration with

ROBERT GORDON UNIVERSITY ABERDEEN

CM-2606

Data Engineering Coursework

Summary Report

Sathila Samarasinghe

IIT ID – 20210515

RGU ID – 2117535

Submitted in partial fulfillment of the requirements for the BSc (Hons) in Artificial Intelligence and Data Science degree at Robert Gordon University.

April 2023

Table of Contents

1. Introduction 3

2. Dataset Selection 3

3. Insight Generation Mechanism 5

4. Pipeline Design 9

4.1 Ingestion Flow 9

4.2 ETL Flow (Extract, Transform, Load) 9

4.3 Creating the Data Warehouse 20

5. Technical Implementation 26

6. Discussion and Conclusion 27

1. Introduction

Telecom churn refers to the phenomenon of customers switching to a different telecom service provider, and it is a critical challenge for telecom companies to address to retain their customer base. To effectively analyze and address telecom churn, a robust data pipeline is essential to process, analyze, and gain insights from large volumes of data.

The solution utilizes Hadoop, an open-source distributed data processing framework, to efficiently handle big data processing tasks. Hive, a data warehouse software built on top of Hadoop, is used for data storage and querying data cleaning and transformation. SQL Queries have been run through Data Analytics Studio (DAS), a web-based tool provided by Hadoop, and data visualization, analysis, and reporting have been done using DAS along with python and Excel.

Once the data is analyzed, an insights generation can be done for stakeholders for decision-making. The solution also incorporates machine learning algorithms, such as predictive modeling and clustering, to identify potential churn customers based on historical data patterns. The end-to-end solution aims to provide telecom companies with actionable insights to proactively address customer churn, improve customer retention strategies, and enhance overall business performance.

2. Dataset Selection

The **Orange Telecom’s Churn Dataset** found on Kaggle is the main source of data for this project. This dataset contains information about the customers of a telecom company and their churn status, i.e., whether they have left the company or not.

There were 2 csv files where data was split into 20% and 80% for the ease of a Machine Learning project. Those 2 have been connected as a single csv in this project. So, a total of 3335 rows and 21 columns have been created.

Each row in the dataset represents a customer, and the columns contain various features and attributes related to the customer. The attributes in the dataset are as follows:

1. State: A string representing the state where the customer is located.
2. Account length: An integer representing the length of the customer's account.
3. Area code: An integer representing the area code of the customer's phone number.
4. International plan: A string indicating whether the customer has an international plan (Yes/No).
5. Voice mail plan: A string indicating whether the customer has a voice mail plan (Yes/No).
6. Number vmail messages: An integer representing the number of voice mail messages the customer has.
7. Total day minutes: A double value representing the total number of minutes the customer used during the day.
8. Total day calls: An integer representing the total number of calls the customer made during the day.
9. Total day charge: A double value representing the total charge for the customer's day usage.
10. Total eve minutes: A double value representing the total number of minutes the customer used during the evening.
11. Total eve calls: An integer representing the total number of calls the customer made during the evening.
12. Total eve charge: A double value representing the total charge for the customer's evening usage.
13. Total night minutes: A double value representing the total number of minutes the customer used during the night.
14. Total night calls: An integer representing the total number of calls the customer made during the night.
15. Total night charge: A double value representing the total charge for the customer's night usage.
16. Total intl minutes: A double value representing the total number of international minutes

used by the customer.

17. Total intl calls: An integer representing the total number of international calls made by the customer.
18. Total intl charge: A double value representing the total charge for the customer's international usage.
19. Customer service calls: An integer representing the total number of customer service calls made by the customer.
20. Churn: A string indicating whether the customer has churned (TRUE/FALSE).
21. ID: An integer created to be used as primary key.

Scenario:

The Orange Telecom Company wants to reduce customer churn and retain more customers. To achieve this, they want to understand the factors that lead to customer churning and identify customers who are at high risk of churning.

3. Insight Generation Mechanism

The dataset contains various features and attributes related to customers, including their account details, phone usage patterns, international plans, voice mail plans, and customer service calls. By analyzing these features, the company can gain insights into the factors that may influence customer churn.

Following factors can be analyzed from this dataset:

1. Customer Demographics: The company can analyze the churn rate of customers from different states and areas to identify areas where they need to improve their services or marketing strategies. For example, if the churn rate is high in a particular state, they can analyze the reasons for the high churn rate and take steps to address the issues.
2. Usage patterns: The company can analyze the usage patterns of customers who churned compared to those who did not churn. For example, they can analyze the average usage minutes, the average number of calls made, and the average amount charged for customers who churned and compare it with those who did not churn. They can identify patterns in the data to identify customers who are at high risk of churning and take

proactive steps to retain them.

3. Plan and Pricing: The company can analyze the churn rate of customers who have international plans or voice mail plans to identify if there is a correlation between these plans and churn rate. They can also analyze the churn rate of customers who are on different pricing plans to identify if there is a correlation between pricing and churn rate.

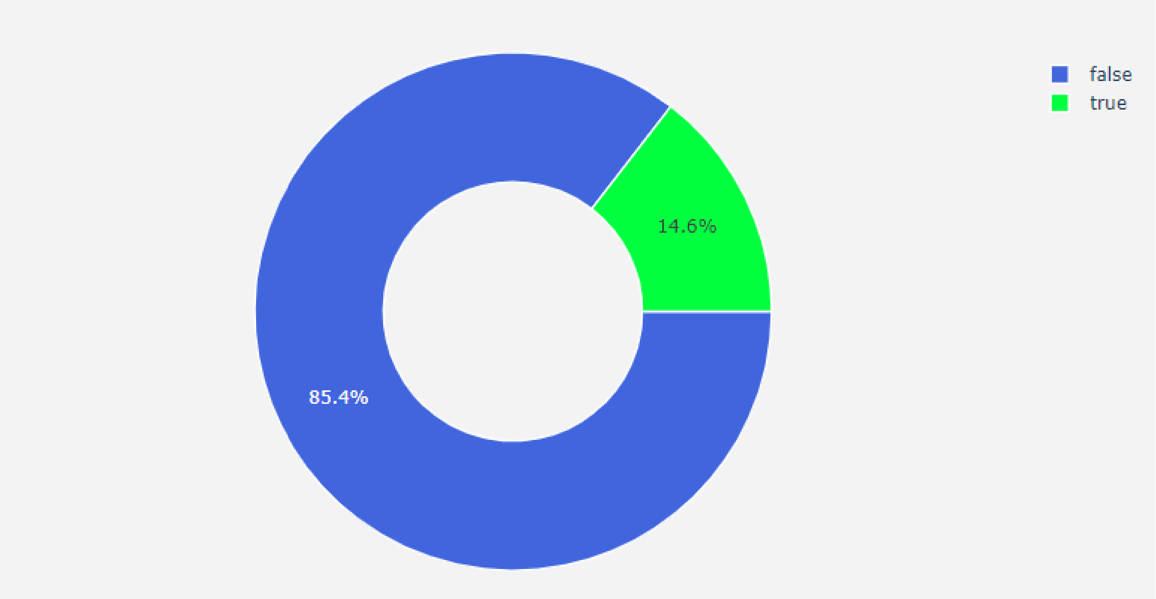
Exploratory Data Analysis

An exploratory data analysis (EDA) has been done to gain a better understanding of the dataset. This includes visualizing the distribution of each feature, identifying any missing or outlier values, and examining correlations between features. EDA provided valuable insights into the dataset and help in identifying patterns or trends that may be indicative of customer churn.

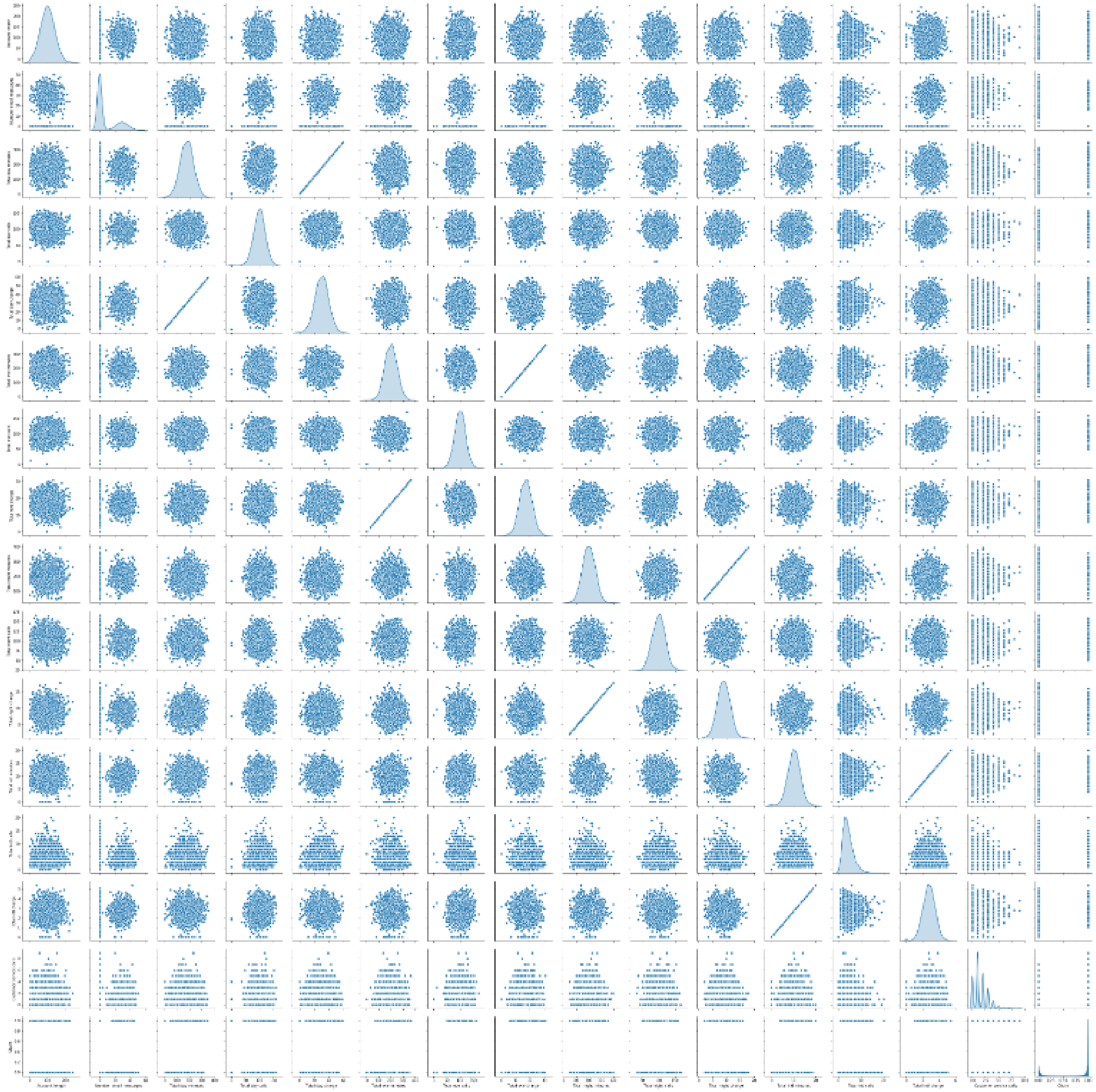
Customer churn:

False – Retained

True - Churned



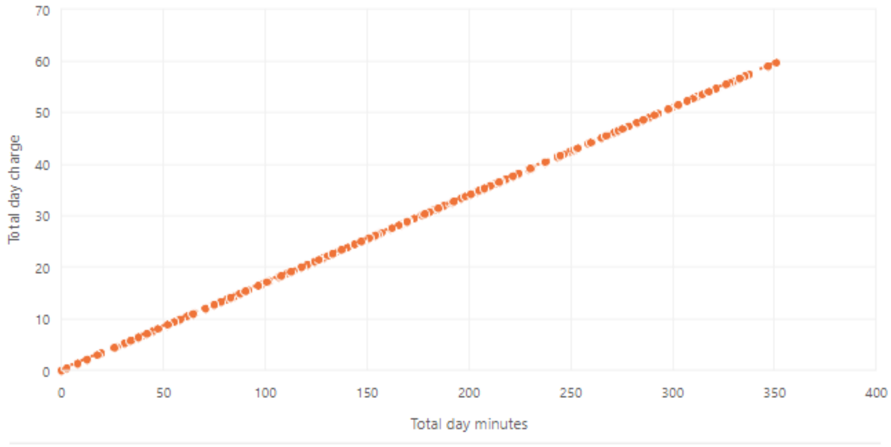
Variable Distributions:



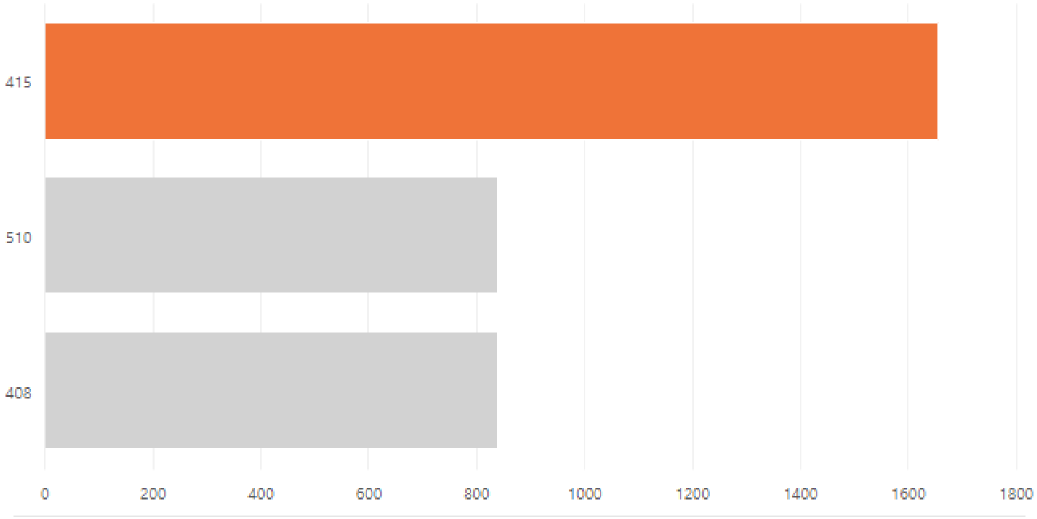
Several numerical data appear highly correlated:

1. Total day minutes and Total day charge.
2. Total eve minutes and Total eve charge.
3. Total night minutes and Total night charge.
4. Total intl minutes and Total intl charge.

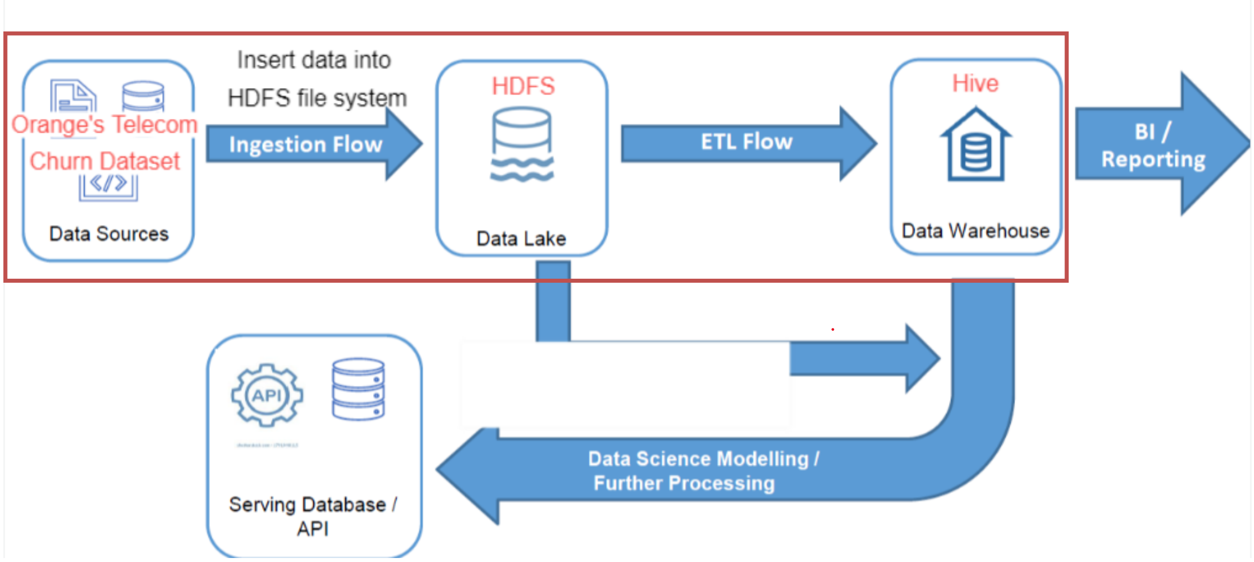
Total day minutes and Total day charge.



Area code



4. Pipeline Design


Orange's Telecom Churn Dataset
Data Sources
Insert data into HDFS file system
Ingestion Flow
HDFS
Data Lake
ETL Flow
Hive
Data Warehouse
BI / Reporting
API
Serving Database / API
Data Science Modelling / Further Processing

Note that the pipeline design has been done only up to the data warehousing part.

4.1 Ingestion Flow

Data ingested from data source (Kaggle dataset) into the data lake Hadoop Distributed File System (HDFS).

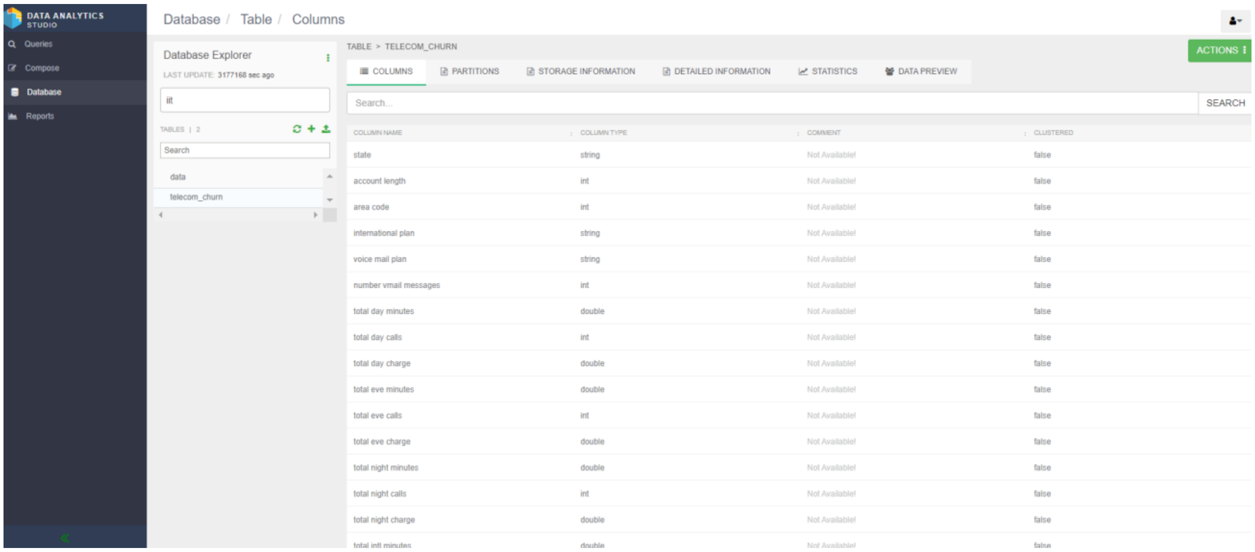
1. Download the dataset from the Kaggle to the local machine.
2. Upload the csv to the HDFS on the virtual machine.

4.2 ETL Flow (Extract, Transform, Load)

After data is stored in the data lake, ETL flow has been applied through the data to transform to a suitable format for analysis.

4.2.1 Extract

The uploaded csv in the HDFS has to be further preprocessed. So, data is extracted into the hive.



COLUMN NAME	COLUMN TYPE	COMMENT	CLUSTERED
state	string	Not Available!	false
account length	int	Not Available!	false
area code	int	Not Available!	false
international plan	string	Not Available!	false
voice mail plan	string	Not Available!	false
number vmail messages	int	Not Available!	false
total day minutes	double	Not Available!	false
total day calls	int	Not Available!	false
total day charge	double	Not Available!	false
total eve minutes	double	Not Available!	false
total eve calls	int	Not Available!	false
total eve charge	double	Not Available!	false
total night minutes	double	Not Available!	false
total night calls	int	Not Available!	false
total night charge	double	Not Available!	false
total intl minutes	double	Not Available!	false

4.2.2 Transform

Data contained some duplicate rows and missing values. Those have been handled in the transform section.

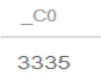
Duplication Handling

A new table called telecom_churn_new was created which has selected unique rows from the previously created telecom_churn table.

```
create table telecom_churn_new as select distinct * from telecom_churn;
```

```
SELECT count(*) FROM telecom_churn;
```

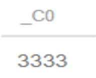
Number of rows before removing duplicates =



_C0
3335

```
SELECT count(*) FROM telecom_churn_new;
```

Number of rows after removing duplicates =



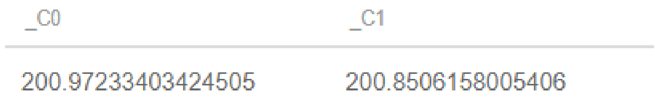
_C0
3333

Missing Value Handling

There were some null values in the dataset. Those were replaced by the mean value of the respective column.

1. Finding the average value of the columns which contain missing values:

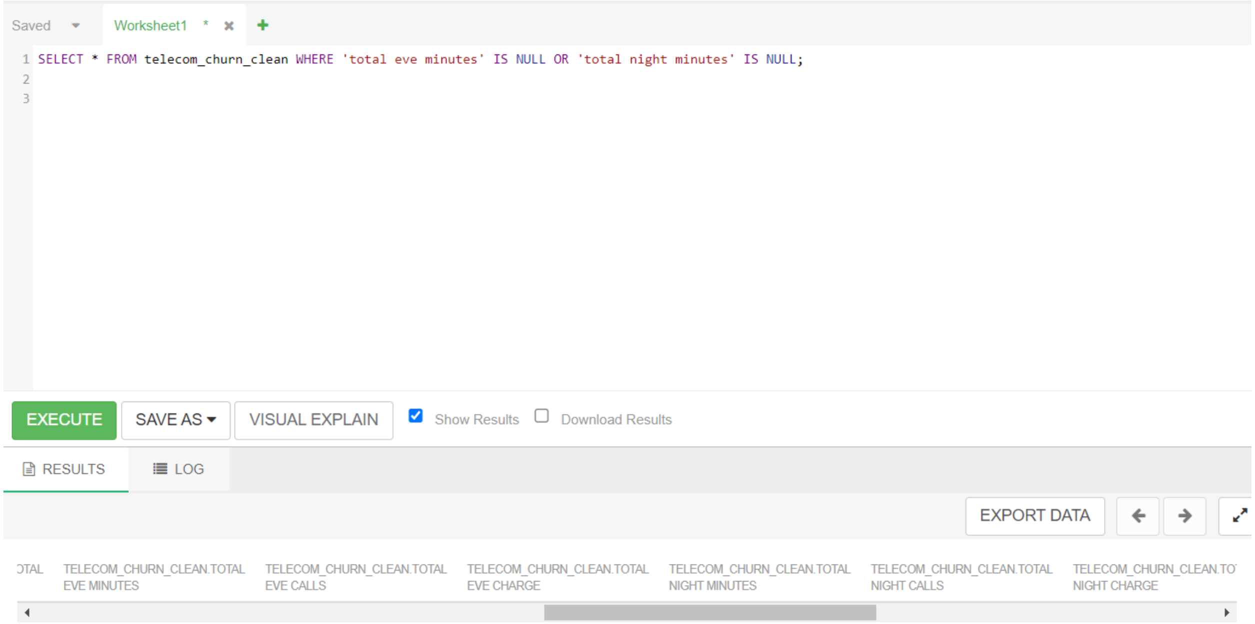
```
select avg(`Total eve minutes`), avg(`Total night minutes`) from telecom_churn_new;
```



_C0	_C1
200.97233403424505	200.8506158005406

2. Replace the missing value with the average value of that column:

```
create table telecom_churn_clean as select
State,
`Account length`,
`Area code`,
`International plan`,
`Voice mail plan`,
`Number vmail messages`,
`Total day minutes`,
`Total day calls`,
`Total day charge`,
coalesce(`Total eve minutes`, 200.97233403424505) as `Total eve minutes`,
`Total eve calls`,
`Total eve charge`,
coalesce(`Total night minutes`, 200.8506158005406) as `Total night minutes`,
`Total night calls`,
`Total night charge`,
`Total intl minutes`,
`Total intl calls`,
`Total intl charge`,
`Customer service calls`,
Churn,
ID
from telecom_churn_new;
```



```
SELECT * FROM telecom_churn_clean WHERE 'total eve minutes' IS NULL OR 'total night minutes' IS NULL;
```

```
OTAL  TELECOM_CHURN_CLEAN.TOTAL  TELECOM_CHURN_CLEAN.TOTAL  TELECOM_CHURN_CLEAN.TOTAL  TELECOM_CHURN_CLEAN.TOTAL  TELECOM_CHURN_CLEAN.TOTAL  TELECOM_CHURN_CLEAN.TO
      EVE MINUTES                EVE CALLS                  EVE CHARGE                 NIGHT MINUTES              NIGHT CALLS                NIGHT CHARGE
```

Missing values were successfully handled by replacing them with the mean.

4.2.3 Loading

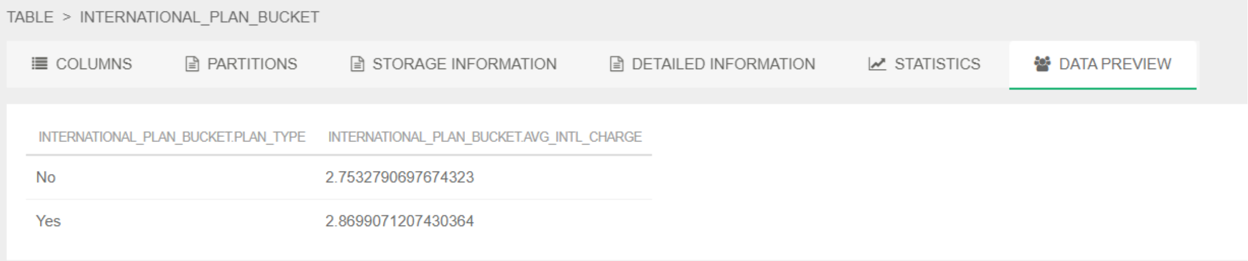
Bucketing

Bucketing is done to the International plan column by taking an average value for total intl charge. If the value in the International plan column is 'No', then it is labeled as 'No' in the plan_type column of the international_plan_bucket table. Otherwise, if the value is not 'No', it is labeled as 'Yes' in the plan_type column.

```
create table international_plan_bucket as SELECT
case
when `International plan` = 'No' then 'No'
else 'Yes'
end as plan_type,
avg(`Total intl charge`) as `avg_intl_charge`
from telecom_churn_clean
group by
case
```

```
when `International plan` ='No' then 'No'
else 'Yes'
end;
```

The resulting average is stored in the avg_intl_charge column of the international_plan_bucket table. Finally, the GROUP BY clause is used to group the data based on the bucket labels defined in the CASE statement, allowing for aggregation and summarization of data based on the 'No' or 'Yes' values in the international plan column.



INTERNATIONAL_PLAN_BUCKET.PLAN_TYPE	INTERNATIONAL_PLAN_BUCKET.AVG_INTL_CHARGE
No	2.7532790697674323
Yes	2.8699071207430364

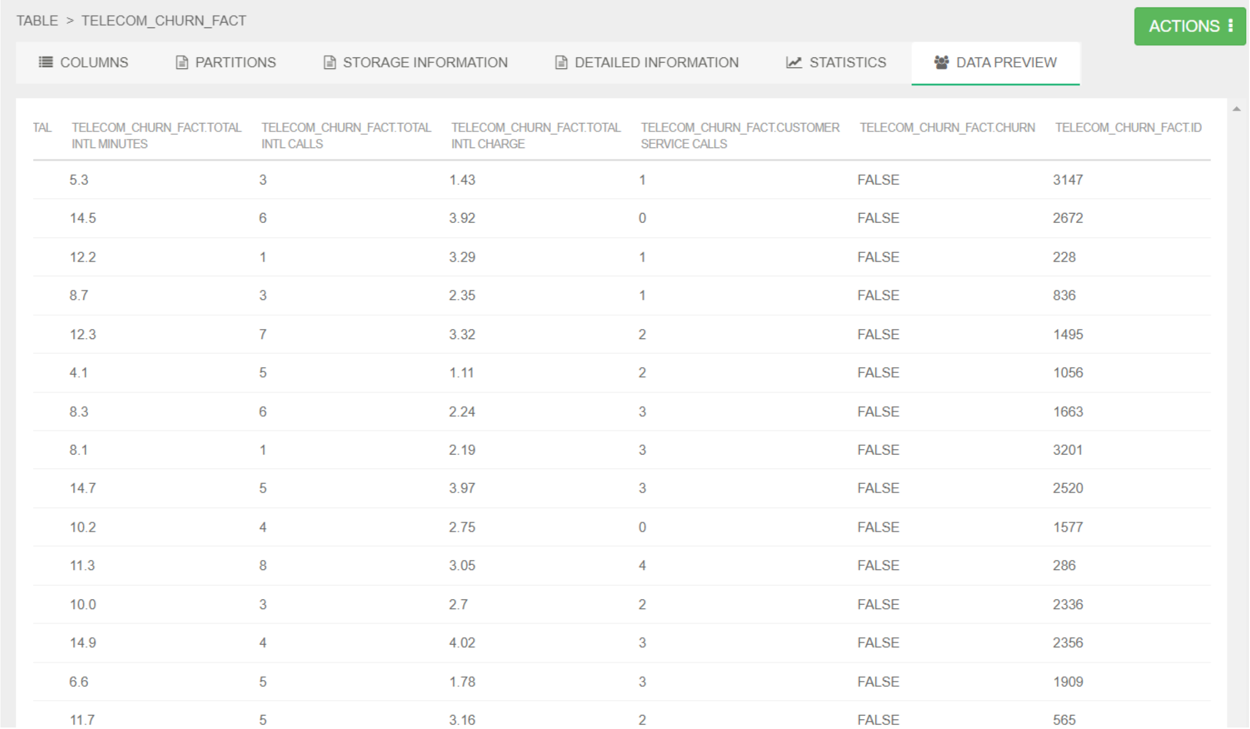
Creating the Fact Table

The fact table acts as the central repository of quantitative data that captures key performance indicators and metrics related to customer churn. It serves as a foundation for data analysis and reporting, enabling insights and actionable information for informed decision-making.

```
create table telecom_churn_fact (
  State STRING,
  `Account length` int,
  `Area code` int,
  `International plan` STRING,
  `Voice mail plan` STRING,
  `Number vmail messages` int,
  `Total day minutes` DOUBLE,
  `Total day calls` int,
  `Total day charge` DOUBLE,
  `Total eve minutes` DOUBLE,
  `Total eve calls` int,
  `Total eve charge` DOUBLE,
  `Total night minutes` DOUBLE,
  `Total night calls` int,
  `Total night charge` DOUBLE,
  `Total intl minutes` DOUBLE,
  `Total intl calls` int,
  `Total intl charge` DOUBLE,
  `Customer service calls` int,
  Churn STRING,
  ID int,
  CONSTRAINT telecom_churn_pk primary key (ID) disable novalidate);
```

Adding Data to Fact Table

```
INSERT INTO TABLE telecom_churn_fact
SELECT
  State,
 `Account length`,
 `Area code`,
 `International plan`,
 `Voice mail plan`,
 `Number vmail messages`,
 `Total day minutes`,
 `Total day calls`,
 `Total day charge`,
 `Total eve minutes`,
 `Total eve calls`,
 `Total eve charge`,
 `Total night minutes`,
 `Total night calls`,
 `Total night charge`,
 `Total intl minutes`,
 `Total intl calls`,
 `Total intl charge`,
 `Customer service calls`,
 Churn,
 ID
FROM telecom_churn_clean;
```

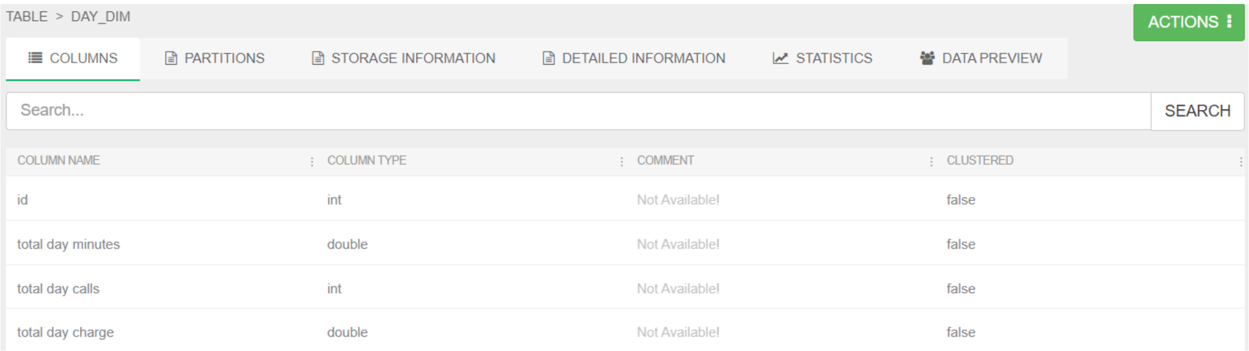


TAL	TELECOM_CHURN_FACT.TOTAL INTL MINUTES	TELECOM_CHURN_FACT.TOTAL INTL CALLS	TELECOM_CHURN_FACT.TOTAL INTL CHARGE	TELECOM_CHURN_FACT.CUSTOMER SERVICE CALLS	TELECOM_CHURN_FACT.CHURN	TELECOM_CHURN_FACT.ID
	5.3	3	1.43	1	FALSE	3147
	14.5	6	3.92	0	FALSE	2672
	12.2	1	3.29	1	FALSE	228
	8.7	3	2.35	1	FALSE	836
	12.3	7	3.32	2	FALSE	1495
	4.1	5	1.11	2	FALSE	1056
	8.3	6	2.24	3	FALSE	1663
	8.1	1	2.19	3	FALSE	3201
	14.7	5	3.97	3	FALSE	2520
	10.2	4	2.75	0	FALSE	1577
	11.3	8	3.05	4	FALSE	286
	10.0	3	2.7	2	FALSE	2336
	14.9	4	4.02	3	FALSE	2356
	6.6	5	1.78	3	FALSE	1909
	11.7	5	3.16	2	FALSE	565

Creating Dimension Tables

a. Creating Dimension Table day_dim

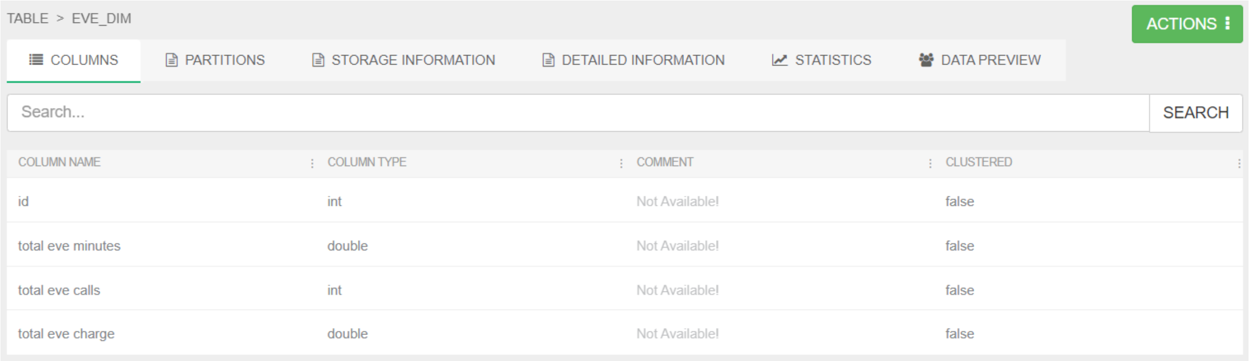
```
CREATE TABLE day_dim (
  ID int,
  `Total day minutes` DOUBLE,
  `Total day calls` int,
  `Total day charge` DOUBLE,
  CONSTRAINT day_dim_pk PRIMARY KEY (ID) DISABLE NOVALIDATE
);
```



COLUMN NAME	COLUMN TYPE	COMMENT	CLUSTERED
id	int	Not Available!	false
total day minutes	double	Not Available!	false
total day calls	int	Not Available!	false
total day charge	double	Not Available!	false

b. Creating Dimension Table eve_dim

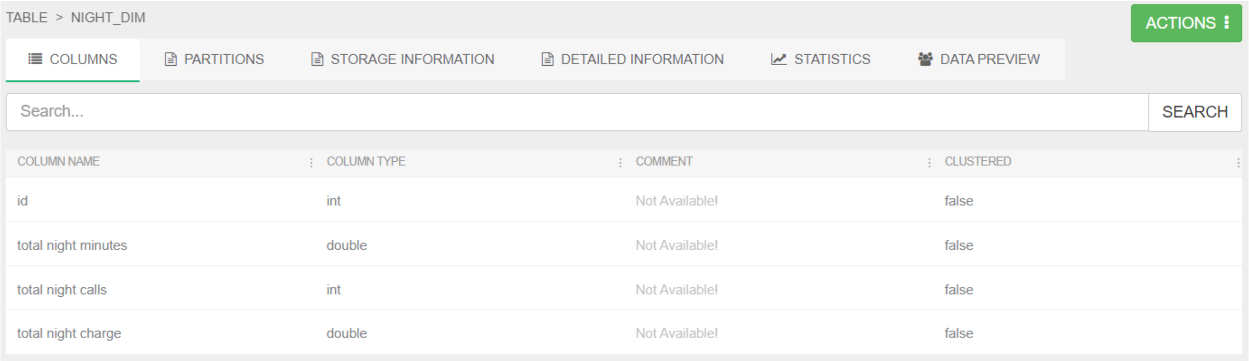
```
CREATE TABLE eve_dim (
  ID int,
  `Total eve minutes` DOUBLE,
  `Total eve calls` int,
  `Total eve charge` DOUBLE,
  CONSTRAINT eve_dim_pk PRIMARY KEY (ID) DISABLE NOVALIDATE
);
```



COLUMN NAME	COLUMN TYPE	COMMENT	CLUSTERED
id	int	Not Available!	false
total eve minutes	double	Not Available!	false
total eve calls	int	Not Available!	false
total eve charge	double	Not Available!	false

c. Creating Dimension Table night_dim

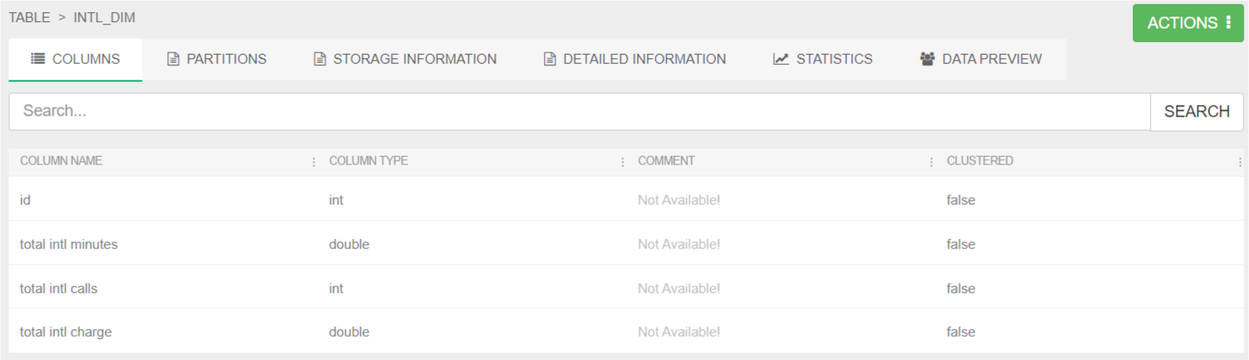
```
CREATE TABLE night_dim (
  ID int,
  `Total night minutes` DOUBLE,
  `Total night calls` int,
  `Total night charge` DOUBLE,
  CONSTRAINT night_dim_pk PRIMARY KEY (ID) DISABLE NOVALIDATE
);
```



COLUMN NAME	COLUMN TYPE	COMMENT	CLUSTERED
id	int	Not Available!	false
total night minutes	double	Not Available!	false
total night calls	int	Not Available!	false
total night charge	double	Not Available!	false

d. Creating Dimension Table intl_dim

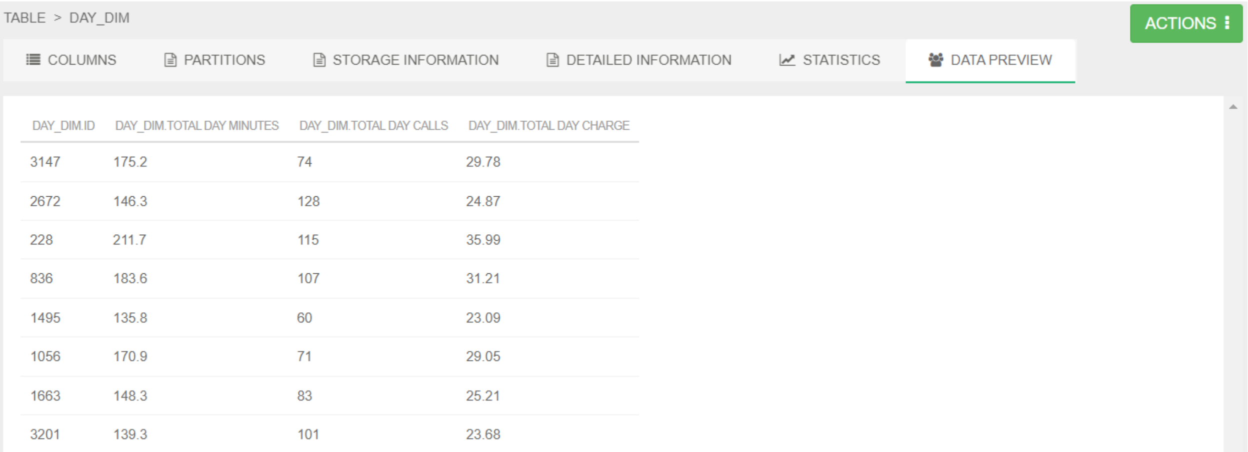
```
CREATE TABLE intl_dim (
  ID int,
  `Total intl minutes` DOUBLE,
  `Total intl calls` int,
  `Total intl charge` DOUBLE,
  CONSTRAINT intl_dim_pk PRIMARY KEY (ID) DISABLE NOVALIDATE
);
```



COLUMN NAME	COLUMN TYPE	COMMENT	CLUSTERED
id	int	Not Available!	false
total intl minutes	double	Not Available!	false
total intl calls	int	Not Available!	false
total intl charge	double	Not Available!	false

a. Adding Data to day_dim

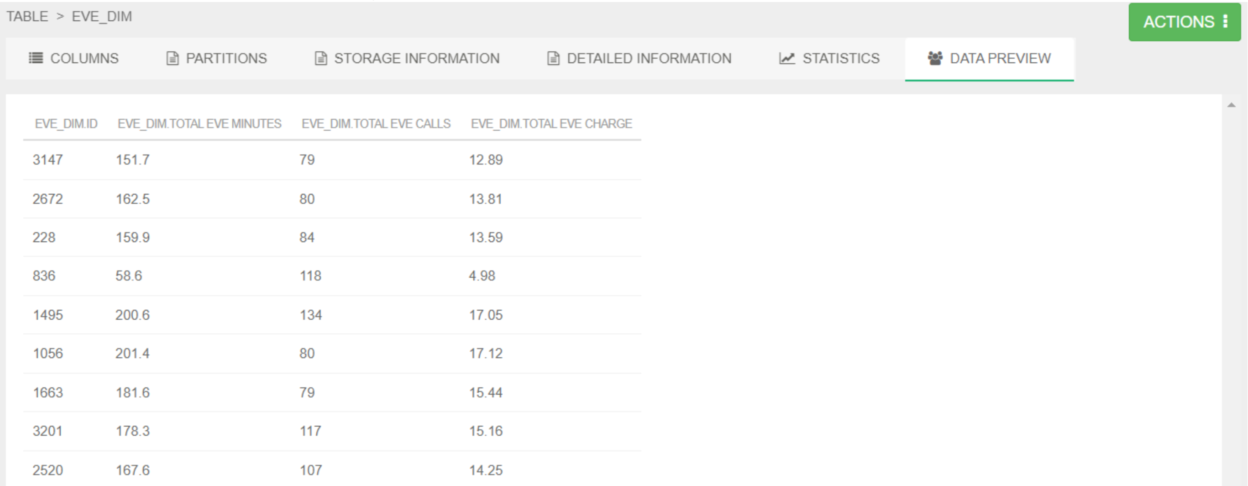
```
INSERT INTO TABLE day_dim
SELECT
  ID,
  `Total day minutes`,
  `Total day calls`,
  `Total day charge`
FROM telecom_churn_clean;
```



DAY_DIM.ID	DAY_DIM.TOTAL DAY MINUTES	DAY_DIM.TOTAL DAY CALLS	DAY_DIM.TOTAL DAY CHARGE
3147	175.2	74	29.78
2672	146.3	128	24.87
228	211.7	115	35.99
836	183.6	107	31.21
1495	135.8	60	23.09
1056	170.9	71	29.05
1663	148.3	83	25.21
3201	139.3	101	23.68

b. Adding Data to eve_dim

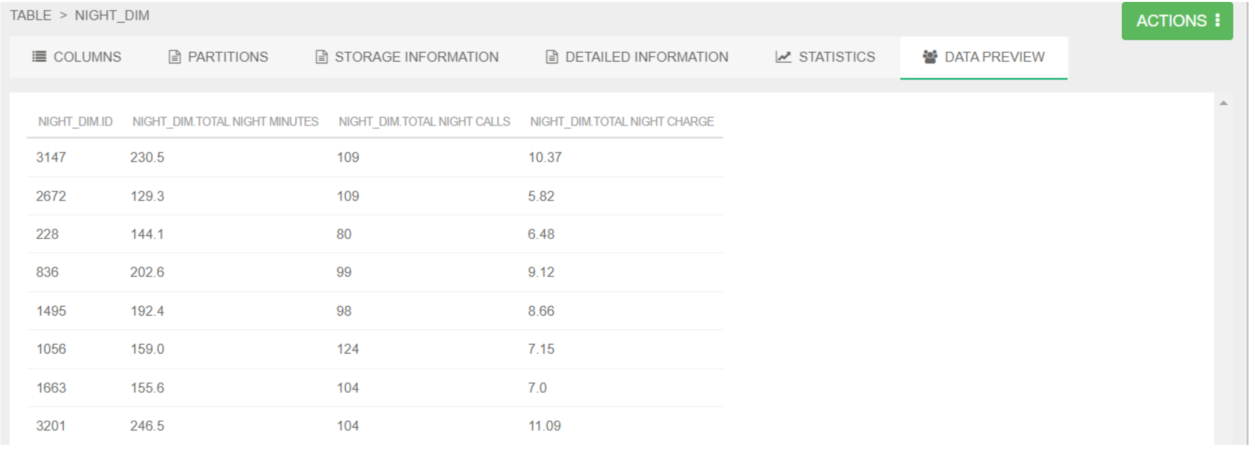
```
INSERT INTO TABLE eve_dim
SELECT
  ID,
  `Total eve minutes`,
  `Total eve calls`,
  `Total eve charge`
FROM telecom_churn_clean;
```



EVE_DIM.ID	EVE_DIM.TOTAL EVE MINUTES	EVE_DIM.TOTAL EVE CALLS	EVE_DIM.TOTAL EVE CHARGE
3147	151.7	79	12.89
2672	162.5	80	13.81
228	159.9	84	13.59
836	58.6	118	4.98
1495	200.6	134	17.05
1056	201.4	80	17.12
1663	181.6	79	15.44
3201	178.3	117	15.16
2520	167.6	107	14.25

c. Adding Data to night_dim

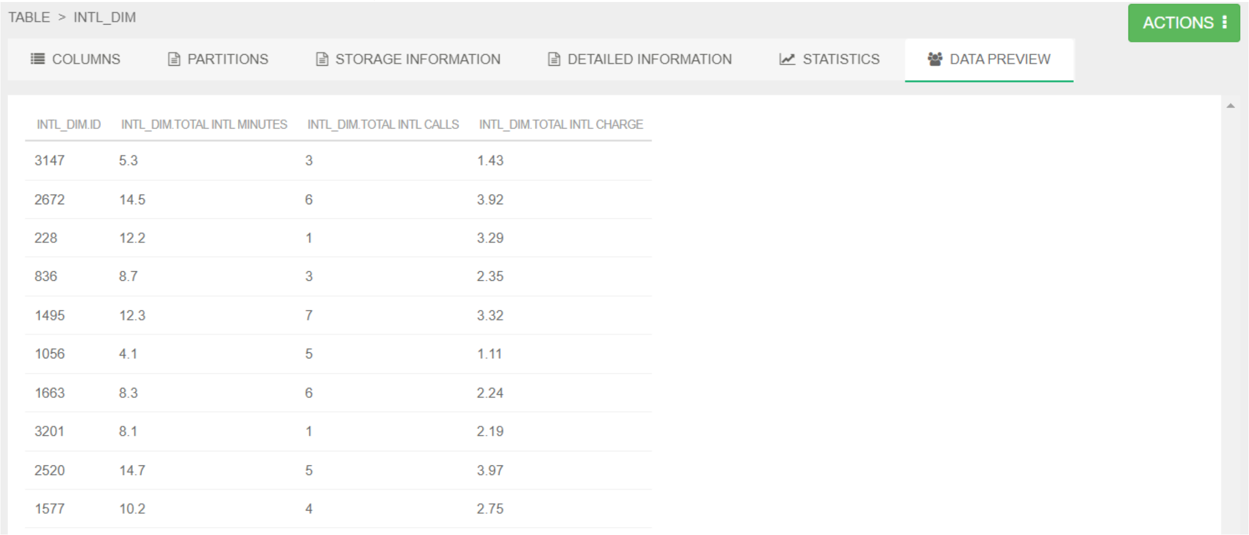
```
INSERT INTO TABLE night_dim
SELECT
  ID,
  `Total night minutes`,
  `Total night calls`,
  `Total night charge`
FROM telecom_churn_clean;
```



NIGHT_DIM.ID	NIGHT_DIM.TOTAL NIGHT MINUTES	NIGHT_DIM.TOTAL NIGHT CALLS	NIGHT_DIM.TOTAL NIGHT CHARGE
3147	230.5	109	10.37
2672	129.3	109	5.82
228	144.1	80	6.48
836	202.6	99	9.12
1495	192.4	98	8.66
1056	159.0	124	7.15
1663	155.6	104	7.0
3201	246.5	104	11.09

d. Adding Data to intl_dim

```
INSERT INTO TABLE intl_dim
SELECT
  ID,
  `Total intl minutes`,
  `Total intl calls`,
  `Total intl charge`
FROM telecom_churn_clean;
```



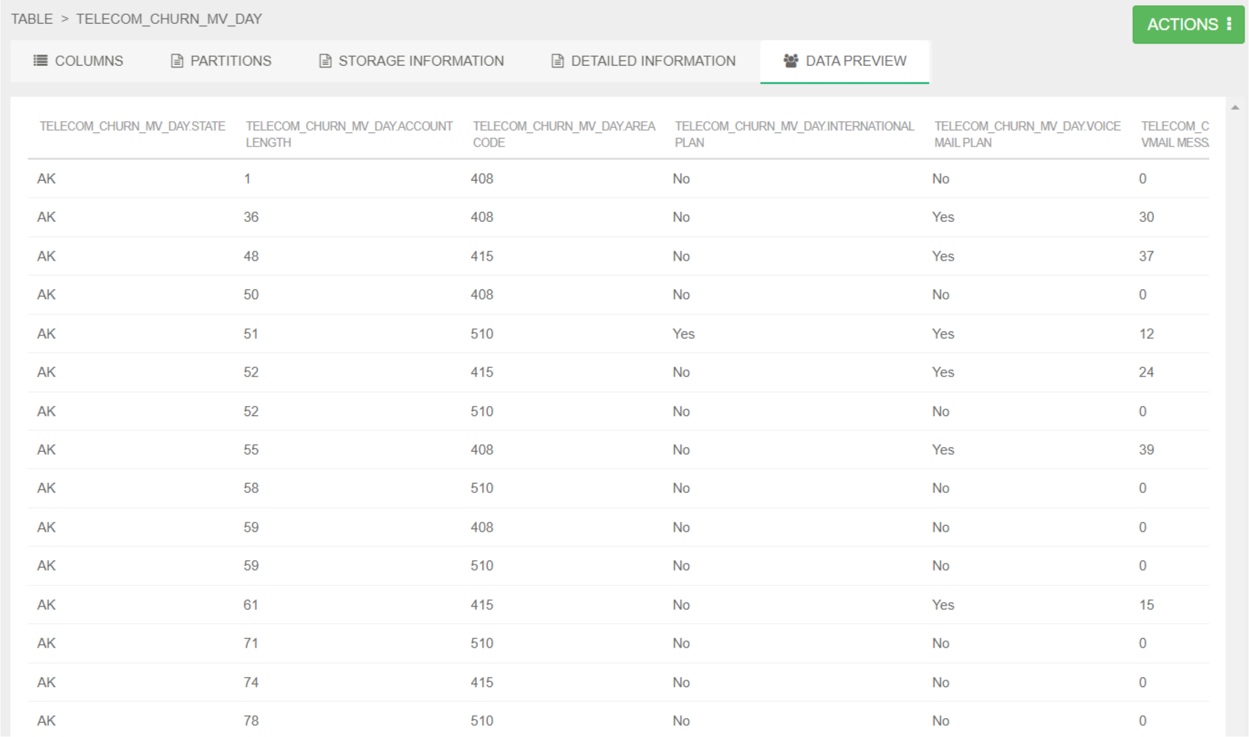
INTL_DIM.ID	INTL_DIM.TOTAL INTL MINUTES	INTL_DIM.TOTAL INTL CALLS	INTL_DIM.TOTAL INTL CHARGE
3147	5.3	3	1.43
2672	14.5	6	3.92
228	12.2	1	3.29
836	8.7	3	2.35
1495	12.3	7	3.32
1056	4.1	5	1.11
1663	8.3	6	2.24
3201	8.1	1	2.19
2520	14.7	5	3.97
1577	10.2	4	2.75

4.3 Creating the Data Warehouse

The data warehouse is designed to handle high volumes of data and is optimized for quick and efficient data retrieval and analysis. It is structured in a way that allows Orange Telecom to easily query and analyze customer data to identify patterns, trends, and correlations that can help uncover factors that lead to customer churning.

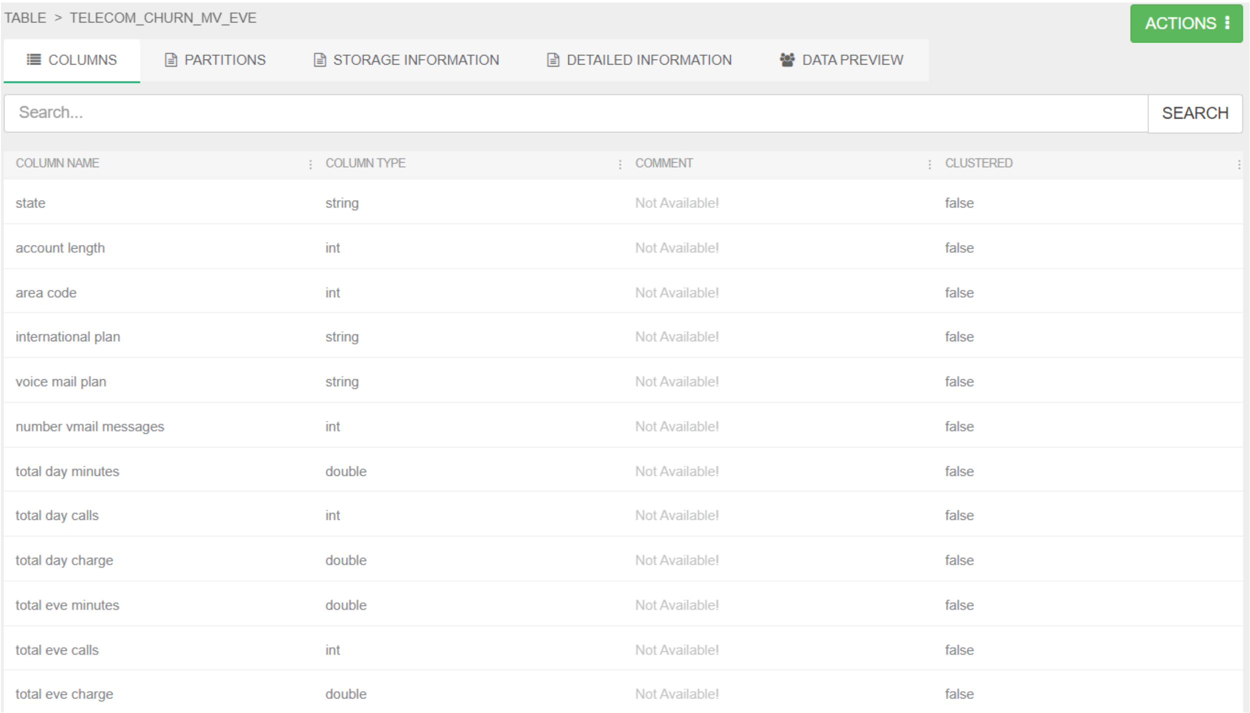
A materialized view is created by executing a query on one or more tables and storing the query results in a table-like structure. This allows for faster query performance, as the materialized view can be queried directly instead of running the query on the underlying tables every time.

```
CREATE MATERIALIZED VIEW telecom_churn_mv_day AS
SELECT
fact.State,
fact.`Account length`,
fact.`Area code`,
fact.`International plan`,
fact.`Voice mail plan`,
fact.`Number vmail messages`,
dim.`Total day minutes`,
dim.`Total day calls`,
dim.`Total day charge`,
fact.`Total eve minutes`,
fact.`Total eve calls`,
fact.`Total eve charge`,
fact.`Total night minutes`,
fact.`Total night calls`,
fact.`Total night charge`,
fact.`Total intl minutes`,
fact.`Total intl calls`,
fact.`Total intl charge`,
fact.`Customer service calls`,
fact.Churn,
fact.ID
FROM
telecom_churn_fact fact
INNER JOIN day_dim dim ON fact.`Total day minutes` = dim.`Total day minutes`
AND fact.`Total day calls` = dim.`Total day calls`
AND fact.`Total day charge` = dim.`Total day charge`;
```



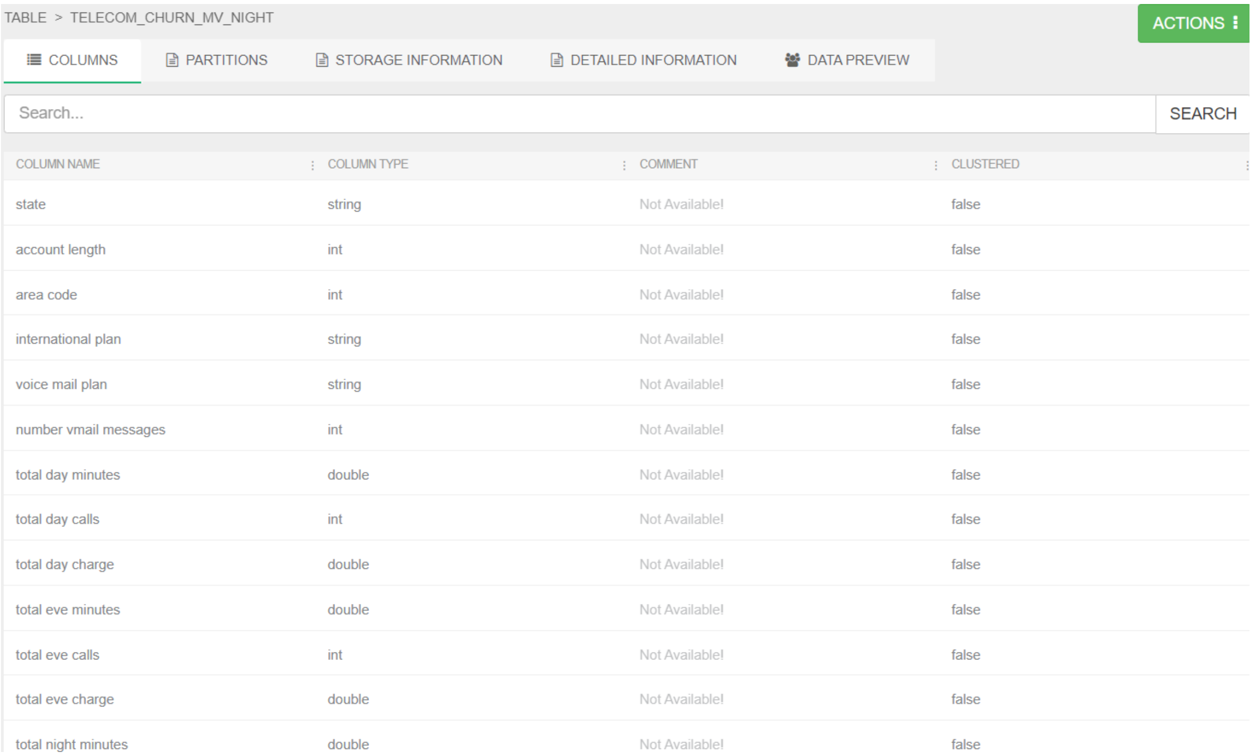
TELECOM_CHURN_MV_DAY.STATE	TELECOM_CHURN_MV_DAY.ACCOUNT LENGTH	TELECOM_CHURN_MV_DAY.AREA CODE	TELECOM_CHURN_MV_DAY.INTERNATIONAL PLAN	TELECOM_CHURN_MV_DAY.VOICE MAIL PLAN	TELECOM_C VMAIL MESS
AK	1	408	No	No	0
AK	36	408	No	Yes	30
AK	48	415	No	Yes	37
AK	50	408	No	No	0
AK	51	510	Yes	Yes	12
AK	52	415	No	Yes	24
AK	52	510	No	No	0
AK	55	408	No	Yes	39
AK	58	510	No	No	0
AK	59	408	No	No	0
AK	59	510	No	No	0
AK	61	415	No	Yes	15
AK	71	510	No	No	0
AK	74	415	No	No	0
AK	78	510	No	No	0

```
CREATE MATERIALIZED VIEW telecom_churn_mv_eve AS
SELECT
fact.State,
fact.`Account length`,
fact.`Area code`,
fact.`International plan`,
fact.`Voice mail plan`,
fact.`Number vmail messages`,
fact.`Total day minutes`,
fact.`Total day calls`,
fact.`Total day charge`,
dim.`Total eve minutes`,
dim.`Total eve calls`,
dim.`Total eve charge`,
fact.`Total night minutes`,
fact.`Total night calls`,
fact.`Total night charge`,
fact.`Total intl minutes`,
fact.`Total intl calls`,
fact.`Total intl charge`,
fact.`Customer service calls`,
fact.Churn,
fact.ID
FROM
telecom_churn_fact fact
INNER JOIN eve_dim dim ON fact.`Total eve minutes` = dim.`Total eve minutes`
AND fact.`Total eve calls` = dim.`Total eve calls`
AND fact.`Total eve charge` = dim.`Total eve charge`;
```



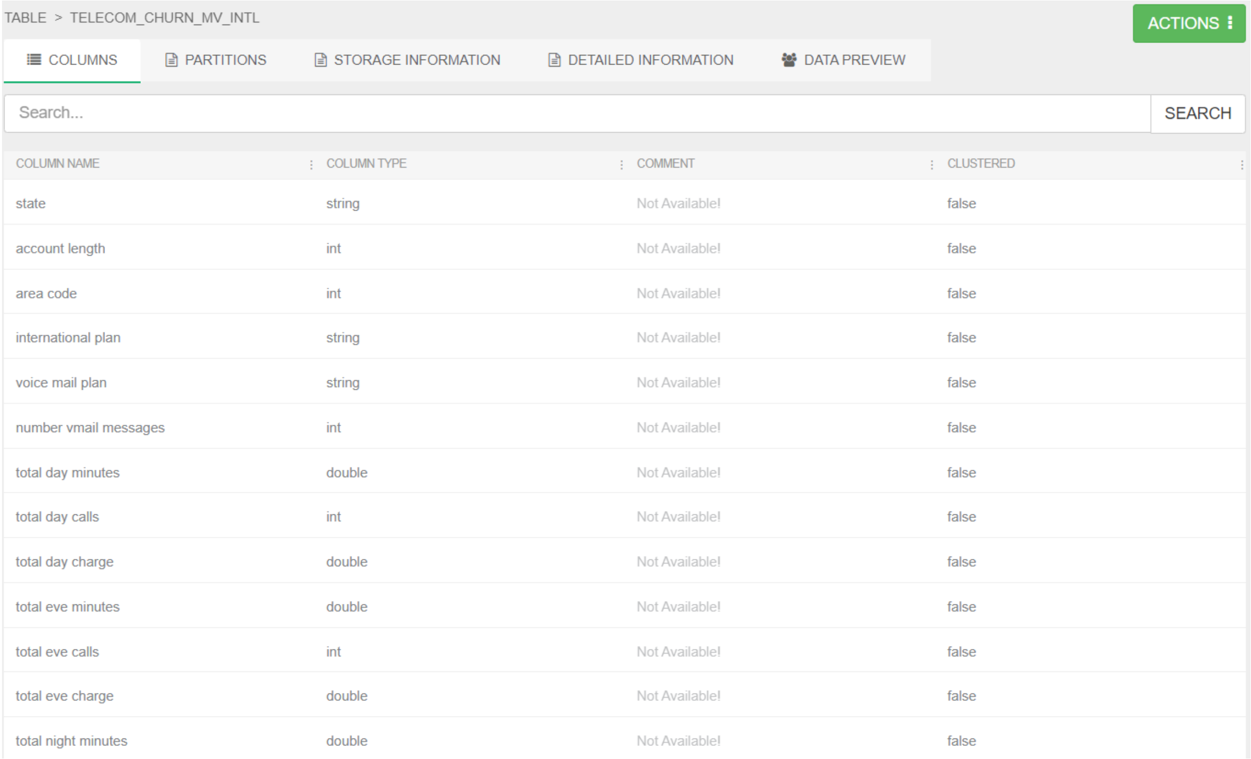
COLUMN NAME	COLUMN TYPE	COMMENT	CLUSTERED
state	string	Not Available!	false
account length	int	Not Available!	false
area code	int	Not Available!	false
international plan	string	Not Available!	false
voice mail plan	string	Not Available!	false
number vmail messages	int	Not Available!	false
total day minutes	double	Not Available!	false
total day calls	int	Not Available!	false
total day charge	double	Not Available!	false
total eve minutes	double	Not Available!	false
total eve calls	int	Not Available!	false
total eve charge	double	Not Available!	false

```
CREATE MATERIALIZED VIEW telecom_churn_mv_night AS
SELECT
fact.State,
fact.`Account length`,
fact.`Area code`,
fact.`International plan`,
fact.`Voice mail plan`,
fact.`Number vmail messages`,
fact.`Total day minutes`,
fact.`Total day calls`,
fact.`Total day charge`,
fact.`Total eve minutes`,
fact.`Total eve calls`,
fact.`Total eve charge`,
dim.`Total night minutes`,
dim.`Total night calls`,
dim.`Total night charge`,
fact.`Total intl minutes`,
fact.`Total intl calls`,
fact.`Total intl charge`,
fact.`Customer service calls`,
fact.Churn,
fact.ID
FROM
telecom_churn_fact fact
INNER JOIN night_dim dim ON fact.`Total night minutes` = dim.`Total night minutes`
AND fact.`Total night calls` = dim.`Total night calls`
AND fact.`Total night charge` = dim.`Total night charge`;
```



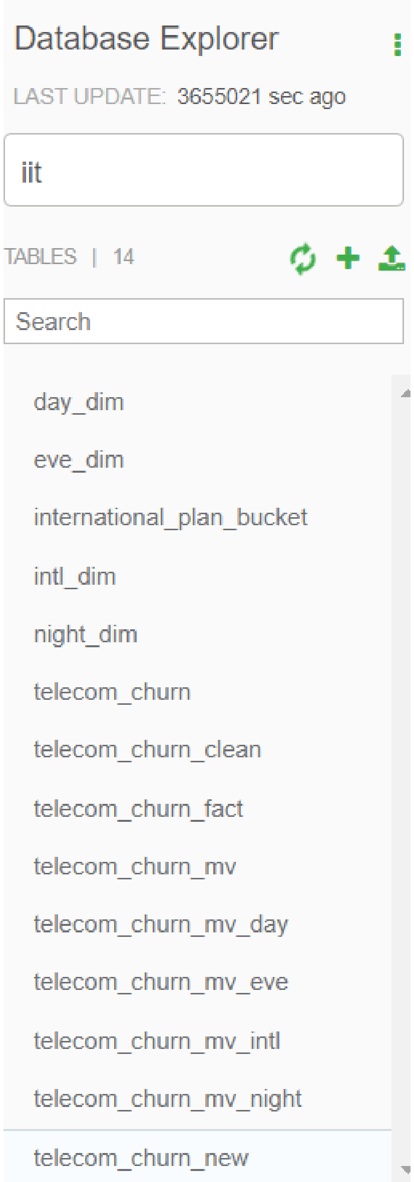
COLUMN NAME	COLUMN TYPE	COMMENT	CLUSTERED
state	string	Not Available!	false
account length	int	Not Available!	false
area code	int	Not Available!	false
international plan	string	Not Available!	false
voice mail plan	string	Not Available!	false
number vmail messages	int	Not Available!	false
total day minutes	double	Not Available!	false
total day calls	int	Not Available!	false
total day charge	double	Not Available!	false
total eve minutes	double	Not Available!	false
total eve calls	int	Not Available!	false
total eve charge	double	Not Available!	false
total night minutes	double	Not Available!	false

```
CREATE MATERIALIZED VIEW telecom_churn_mv_intl AS
SELECT
fact.State,
fact.`Account length`,
fact.`Area code`,
fact.`International plan`,
fact.`Voice mail plan`,
fact.`Number vmail messages`,
fact.`Total day minutes`,
fact.`Total day calls`,
fact.`Total day charge`,
fact.`Total eve minutes`,
fact.`Total eve calls`,
fact.`Total eve charge`,
fact.`Total night minutes`,
fact.`Total night calls`,
fact.`Total night charge`,
dim.`Total intl minutes`,
dim.`Total intl calls`,
dim.`Total intl charge`,
fact.`Customer service calls`,
fact.Churn,
fact.ID
FROM
telecom_churn_fact fact
INNER JOIN intl_dim dim ON fact.`Total intl minutes` = dim.`Total intl minutes`
AND fact.`Total intl calls` = dim.`Total intl calls`
AND fact.`Total intl charge` = dim.`Total intl charge`;
```



COLUMN NAME	COLUMN TYPE	COMMENT	CLUSTERED
state	string	Not Available!	false
account length	int	Not Available!	false
area code	int	Not Available!	false
international plan	string	Not Available!	false
voice mail plan	string	Not Available!	false
number vmail messages	int	Not Available!	false
total day minutes	double	Not Available!	false
total day calls	int	Not Available!	false
total day charge	double	Not Available!	false
total eve minutes	double	Not Available!	false
total eve calls	int	Not Available!	false
total eve charge	double	Not Available!	false
total night minutes	double	Not Available!	false

Summary of Tables:



Database Explorer

LAST UPDATE: 3655021 sec ago

iit

TABLES | 14

Search

- day_dim
- eve_dim
- international_plan_bucket
- intl_dim
- night_dim
- telecom_churn
- telecom_churn_clean
- telecom_churn_fact
- telecom_churn_mv
- telecom_churn_mv_day
- telecom_churn_mv_eve
- telecom_churn_mv_intl
- telecom_churn_mv_night
- telecom_churn_new

5. Technical Implementation

The technical implementation involved several tools and technologies. A pre-configured virtual machine (Virtual Box) was used to provide a complete environment for big data processing.

Hadoop, a widely used distributed storage and processing framework, was utilized to store and manage the raw data of the telecom churn dataset. Hive, a data warehouse solution built on top of Hadoop, was used with the Data Analytics Studio (DAS) to store the transformed data in a structured manner for efficient querying and analysis. MS excel and python were used for EDA and data visualizations. Hadoop queries and SQL were used to create the ETL pipeline.

These tools and technologies provided a robust and efficient solution for storing, processing, and analyzing the large and complex dataset, enabling effective churn prediction and customer retention strategies.

6. Discussion and Conclusion

Challenges:

- Hardware requirements – It was needed a minimum of 16 GB RAM and sufficient disk space to run Hadoop on the local machine.
- Quality of the dataset – Since the dataset contained missing values and duplicates, data needed to be preprocessed.
- Data ingestion and extraction - Extracting data from various sources other than downloading a Kaggle dataset was found difficult.

Possible Enhancements:

- Real-time Data Processing: Use web scraping and APIs as data sources and enhance the data pipeline to process real-time data, allowing telecom companies to monitor customer behavior and patterns in real-time. This can provide timely insights and enable proactive actions to prevent churn.
- Workflow Automation (Orchestration): Orchestration allows for the automation of complex data processing workflows, which can involve multiple tasks, tools, and dependencies. It helps streamline and automate repetitive tasks, reducing manual errors and increasing overall efficiency.
- Cloud-based Deployment: Consider deploying the solution on a cloud-based platform such as AWS or Azure to leverage the scalability and flexibility. This can enable faster processing of large volumes of data and provide easier access to data analytics tools and services.
- Automated Data Cleaning and Transformation: Implement automated data cleaning and transformation techniques to improve data quality and consistency. This can include data validation, outlier detection, and data imputation methods to ensure that the data used for analysis is accurate and reliable.