Optimizing Customer Satisfaction: A Deep Dive into Spar Nord Bank ATM Transactions

A Project Report

Presented to

DATA-228-11

Spring, 2023

By

Bhavana Prasad Kote (016044899)

Jay Dattoo Dale (016646279)

Rashmi Shree Veeraiah (016099395)

Sawan Shivanand Beli (016522662)

May 7, 2023

ABSTRACT

Optimizing Customer Satisfaction: A Deep Dive into Spar Nord Bank ATM Transactions

Bhavana Prasad Kote, Jay Dattoo Dale, Rashmi Shree Veeraiah, Sawan Shivanand Beli

Spar Nord Bank is a financial institution that aims to provide a unique blend of personal advice and service supported by active involvement and cutting-edge digital solutions. The bank's operations are based on a franchise-inspired model where locally based ownership is the primary driver of customer relations and business volume, backed by strong central support. In 2017, Spar Nord Bank made its ATM transaction data publicly available and encouraged analysis of the information to gain insights into ATM usage patterns. The goal is to comprehend the behavior of customer withdrawals and perform an analysis utilizing various AWS services like S3, construct a batch ETL pipeline that will extract transactional data from an RDS, transform it, and load it into target dimensions and facts on the Redshift Data Warehouse. The analysis of ATM usage patterns enables banks to enhance their operations, including ATM placement, network design, and cash management, which leads to decreased costs, heightened security, and an optimized ATM network. Furthermore, examining ATM transaction data assists in detecting fraud and minimizing financial losses. Ultimately, this analysis results in a more efficient, secure, and customer-oriented banking experience.

Acknowledgements

The creation of this study was greatly aided by Professor Andrew H. Bond and TA Venkatasai Rao Dheekonda, for whom the authors are extremely grateful.

Table of Contents

List of Figures

Chapter 1 Introduction

1.1 Project goals and objectives

Analyzing ATM transaction data from banks can yield insightful information that can be used by banks to increase security, enhance customer service, better utilize resources, and support business decisions.

This project analyzes Spar Nord Bank ATM transaction data using analytical queries and Visualization. The analysis will discover patterns, trends, and customer behaviors to optimize the bank's ATM network, decrease expenses, improve security, and improve customer service. This study extracts transactional data from an RDS using Sqoop and performs initial data exploration, cleaning and preprocessing using Python and PySpark. The data will then be extracted, transformed, and loaded into target dimensions and facts on Redshift Data Warehouse. ETL will be fast, precise, and scalable for big data. S3 will store and manage transactional data, Redshift will store target data, and other AWS services will be used to efficiently execute the ETL process and analyze data. By identifying high-traffic regions, peak usage hours, and cash management demands, Spar Nord Bank can optimize its ATM network. This optimization should reduce bank costs and improve user satisfaction. ATM transaction data helps detect fraud and other security concerns. The project discovers problematic patterns and makes security suggestions.

1.2 Problem and Motivation

The project's primary goal is to optimize ATM usage patterns and management in Spar Nord Bank, which is crucial to provide an efficient, secure, and customer-oriented banking experience while minimizing costs and mitigating risks. The project aims to gain insights into

customer behavior to enhance the bank's operations and optimize its ATM network. Failure to do so can lead to decreased customer satisfaction, loyalty, and financial losses due to fraudulent activities. The project proposes to address these challenges by utilizing AWS services and constructing a batch ETL pipeline to extract, transform, and load transactional data for analysis. The project's motivation is to better understand customer withdrawals and ATM usage patterns to improve the customer experience. The project has significant potential to enhance the efficiency, security, and customer-oriented approach of financial institutions while minimizing costs and mitigating risks, making it highly relevant and impactful.

The project's academic/technical contributions can extend to several areas, such as developing advanced machine learning algorithms to detect fraudulent activities by training models on historical transactional data. The project aims to construct a scalable and robust data pipeline architecture that can handle large volumes of transactional data efficiently by leveraging cloud-based services. Additionally, the project can contribute to the development of best practices for data governance, data quality, and data privacy to ensure accurate, consistent, and secure data, which can be leveraged to make informed business decisions.

1.3 Project Results and Expected Deliverables

The key outcome of this project is the creation of an enhanced ATM usage pattern and management system for Spar Nord Bank. Through analysis of customer behavior, this system will enable identification of patterns and trends, and facilitate optimization of the bank's operations including placement of ATMs, network design, and cash management.

The expected deliverables of this project include:

System prototype: A fully functional system prototype that demonstrates the optimized ATM usage pattern and management system's capabilities and functionalities.

Report: A comprehensive report that details the project's objectives, methodology, results, and academic/technical contributions.

Code: All code developed for the project will be made available on Github.

Documentation: Documentation on the sources, quality, and integrity of the data used in the project, report and visualizations.

Project presentation: A project presentation that summarizes the project's objectives, methodology, results, and impact. The presentation will be delivered to the intended audience and stakeholders.

Chapter 2 Background and Related Work

2.1 Background and Used Technologies

Background

Spar Nord Bank is a Danish financial institution that aims to provide personalized advice and services, supported by advanced digital solutions. The bank operates on a franchise-based model, where local ownership is the primary driver of customer relations and business volume, backed by strong central support. In 2017, Spar Nord Bank made its ATM transaction data publicly available to encourage analysis of the information to gain insights into ATM usage patterns. Analyzing this data can lead to more efficient and secure banking experiences for customers.

Amazon S3: S3 is an object storage service that offers industry-leading scalability, data durability, security, and performance. In this project, S3 will be used to store and manage raw transactional data from ATM transactions.

Amazon Glue: Glue is a fully managed ETL service that makes it easy to extract, transform, and load data for analytics. Glue will be used in this project to create a batch ETL pipeline that extracts transactional data from S3, transforms it, and loads it into target dimensions and facts on Redshift Data Warehouse.

Amazon Redshift: Redshift is a fully managed, petabyte-scale data warehouse that makes it simple and cost-effective to analyze all your data using standard SQL. Redshift will be used in this project to store the extracted and transformed data from ATM transactions and to perform analytics using queries on Redshift Query Editor.

The combination of these technologies in this project enable us to extract insights from ATM transaction data of Spar Nord bank to optimize their ATM network, reduce fraud, and improve customer service. The use of these technologies is intended to make the ETL process efficient, precise, and scalable for handling big data.

2.2 Literature Survey

ATM transaction data analysis has gained significant attention in recent years, with studies focusing on forecasting cash demands, optimizing ATM locations, and managing cash in ATMs. Teddy and Ng (2010) proposed a local learning model for forecasting ATM cash demands, while Rajwani et al. (2017) developed a regression model to predict cash demands using factors like day of the week and weather conditions. Genevois et al. (2015) optimized ATM locations and cash allocation to minimize costs and meet customer demands. These studies demonstrate the

potential of ATM transaction data analysis in improving the efficiency and effectiveness of ATM operations.

Chapter 3 Data Overview

3.1 Data Description

The Spar Nord Bank ATM transaction dataset is obtained from Kaggle. The dataset contains over 2.5 million records of ATM transactions conducted by customers of Spar Nord Bank.

Each transaction record includes information such as the date and time of the transaction, the amount withdrawn or deposited, the ATM location, the type of transaction, and whether the transaction was successful or not. The dataset covers a period of several years and contains data from multiple ATM locations throughout Denmark. This rich dataset will be used to uncover patterns and trends in customer behavior and ATM usage, which can then be used to optimize the bank's ATM network and improve customer service. The dataset was put into RDS and then extracted using Sqoop. The raw dataset is in CSV format and is stored in Amazon S3 after initial cleaning and preprocessing for further processing and analysis.

The dataset consists of two csv files ‘atm_data_part1.csv’ and ‘atm_data_part2.csv’. There are 2.5 million rows and 33 columns that consists of ATM transactions data and these columns in the dataset includes which is shown below:

month : The month the withdrawal was made.

year : The year the withdrawal was made

day: The day of month the withdrawal was made.

weekday : The day of the week the withdrawal was made.

hour: The hour of day the withdrawal was made.

atm_status: Is the atm still in service or has it been closed permanently

atm_id: An arbitrary unique ID number for each ATM

atm_manufacturer: The manufacturer of the ATM

atm_location: A text string describing the location of the atm, typically a city.

atm_streetname: The street name of the street where the atm is located

atm_street_number: The street number where the atm is located

atm_zipcode: The zipcode where the atm is located

atm_lat: Latitude of the atm location, coordinates are WGS84

atm_lon: Longitude of the atm location, coordinates are WGS84

currency: Currency of the withdrawal

card_type: The type of card that was used in the withdrawal.

service: The atm service used.

message_code: Error codes. Missing value means the withdrawal was made without error.

message_text: Error text message corresponding to an error code.

weather_lat:Longitude of the weather measurement location, coordinates are WGS84.

weather_lon: Latitude of the weather measurement location, coordinates are WGS84.

weather_city_id: The city id of the city where the measurement was taken

weather_city_name: The name of the city where the measurement was taken

temp: Temperature at the moment of measurement within the hour.

pressure: Atmospheric pressure (on the sea level), hPa

humidity: Humidity, %

wind_speed: Wind speed. Unit Default: meter/sec

wind_deg: Wind direction, degrees (meteorological)

rain_3h: Rain volume for the last 3 hours

clouds_all: Cloudiness

weather_id: Weather condition id

weather_main: Group of weather parameters (Rain, Snow, Extreme etc.)

weather_description: Weather condition within the group

3.2 Dataset Samples

Some of the samples of the raw ATM transactions data of Spar Nord bank are shown in below figures.

	year	month	day	weekday	hour	atm_status	atm_id	atm_manufacturer	atm_location	atm_streetname	...	temp	pressure	humidity	wind_speed	wind_
0	2017	January	1	Sunday	0	Active	1	NCR	Næstved	Farimagsvej	...	281.15	1014	87	7	
1	2017	January	1	Sunday	0	Inactive	2	NCR	Vejgaard	Hadsundvej	...	280.64	1020	93	9	
2	2017	January	1	Sunday	0	Inactive	2	NCR	Vejgaard	Hadsundvej	...	280.64	1020	93	9	
3	2017	January	1	Sunday	0	Inactive	3	NCR	Ikast	Rådhusstrædet	...	281.15	1011	100	6	
4	2017	January	1	Sunday	0	Active	4	NCR	Svogerslev	Brønsager	...	280.61	1014	87	7	
5	2017	January	1	Sunday	0	Active	5	NCR	Nibe	Torvet	...	280.64	1020	93	9	
6	2017	January	1	Sunday	0	Active	6	NCR	Fredericia	Sjællandsgade	...	281.15	1014	93	7	
7	2017	January	1	Sunday	0	Active	7	Diebold Nixdorf	Hjallerup	Hjallerup Centret	...	280.64	1020	93	9	
8	2017	January	1	Sunday	0	Active	8	NCR	Glyngøre	Færgevej	...	281.15	1011	100	6	
9	2017	January	1	Sunday	0	Active	9	Diebold Nixdorf	Hadsund	Storegade	...	280.64	1020	93	9	

10 rows × 33 columns

Fig. 1. Sample of the raw ATM transactions data

	year	month	day	weekday	hour	atm_status	atm_id	atm_manufacturer	atm_location	atm_streetname	...	temp	pressure	humidity	wind_spe
1249990	2017	July	2	Sunday	12	Active	41	Diebold Nixdorf	Skagen	Sct. Laurentiivej	...	287.974082	1009	77	
1249991	2017	July	2	Sunday	12	Active	10	NCR	Nørresundby	Torvet	...	288.756852	1020	71	
1249992	2017	July	2	Sunday	12	Active	33	NCR	Vadum	Ellehammersvej	...	288.756852	1020	71	
1249993	2017	July	2	Sunday	12	Active	71	NCR	Aalbæk	Centralvej	...	287.594031	1018	90	
1249994	2017	July	2	Sunday	12	Inactive	102	NCR	Aalborg Storcenter Afd	Hobrovej	...	288.756852	1020	71	
1249995	2017	July	2	Sunday	12	Active	31	NCR	Slagelse	Mariendals Alle	...	289.237276	1016	79	
1249996	2017	July	2	Sunday	12	Active	78	Diebold Nixdorf	Nyborg	Vestergade	...	289.390441	1017	88	
1249997	2017	July	2	Sunday	12	Inactive	2	NCR	Vejgaard	Hadsundvej	...	288.756852	1020	71	
1249998	2017	July	2	Sunday	12	Active	5	NCR	Nibe	Torvet	...	288.756852	1020	71	
1249999	2017	July	2	Sunday	12	Active	69	NCR	Taars	Bredgade	...	288.756852	1020	71	

Fig. 2. Sample of the raw ATM transactions data

Chapter 4 Architecture and Project Flow

4.1 Architecture

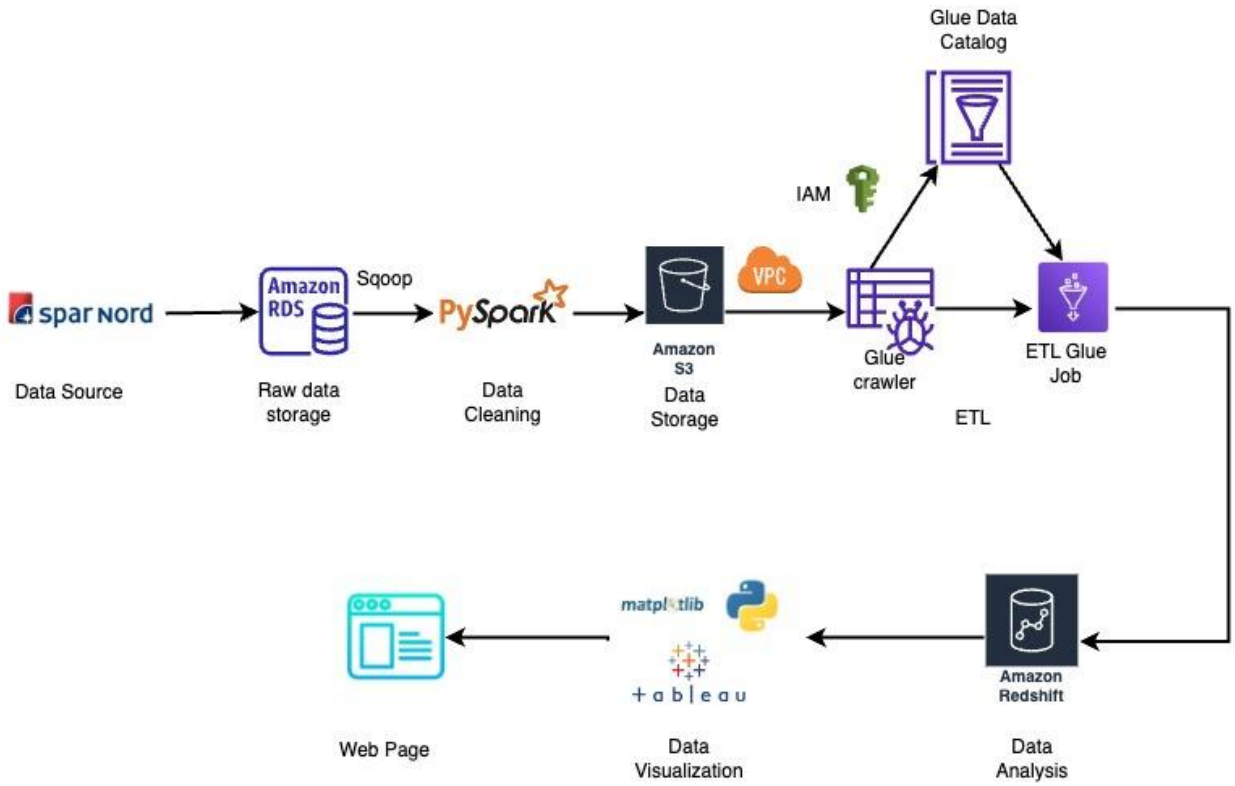


Fig. 3. Project Architecture Diagram

4.2 Data Modeling and ER diagram

The raw dataset consisting of two csv files are merged and cleaned using PySpark module on Jupyter Notebook. Further, designed a data model for the data warehouse using Star schema. The data model designed for ATM transactions data consists of four dimension tables and one fact table. The dimension tables include atm, date, location and card_type whereas the fact table is fact_. The dimension tables and fact table have one-to-many relationships where dimension tables are on one side and the fact table is on many sides. Each Dimension table, which is

relevant to each entity type defines the items in the fact table and is largely static. It has a single fact table with a composite primary key containing all the quantitative information and is dynamic which can accommodate quantitative data. Below figure shows the data model designed for this project.

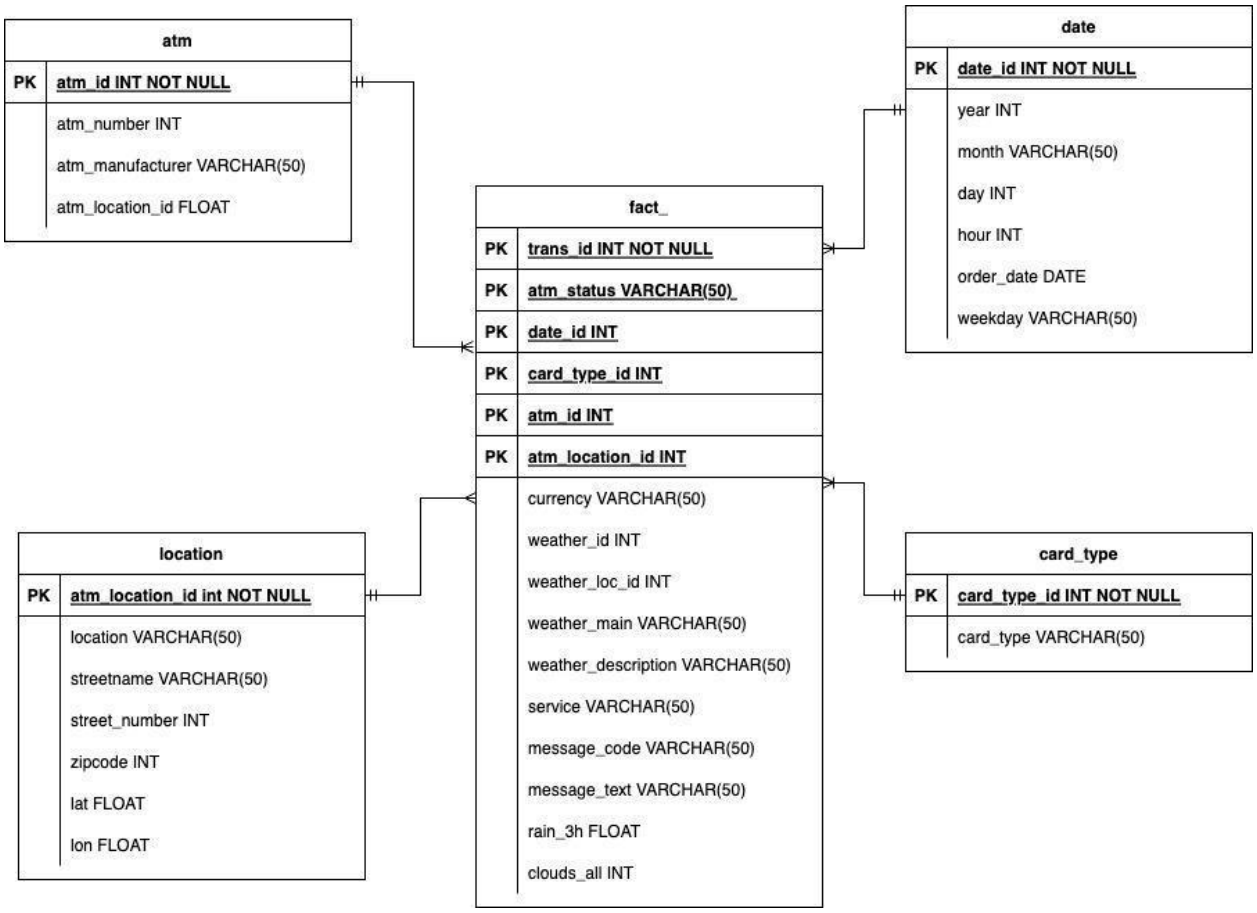


Fig. 4. Entity Relationship Diagram with Fact & Dimension Tables

STAR SCHEMA DIAGRAM

4.3 Project Flow

The table below depicts the technological stack employed during the project development and subsequent project flow

Tool	Purpose
Sqoop	Extract data from RDS
PySpark	Data preprocessing, transformation, analysis
AWS S3	Store transactional data from Spar Nord Bank ATMs.
AWS Glue	Extract, Transform, Load
AWS Redshift	Perform analysis using queries.
Tableau	Data Visualization

To improve the bank's ATM network, the project flow follows an organized plan. First, Sqoop is used to get transactional data from the RDS into Python's PySpark. After performing initial data cleaning and transformation, the data is stored in the S3 bucket. Second, AWS Glue is used to carry out extract, transform and load operations (ETL) so that it is consistent and ready for analysis using SQL queries. The data is subsequently loaded into the Redshift Data Warehouse after it has been cleaned and transformed. Furthermore, the data is looked at and analyzed with the help of Redshift Query Editor to find patterns, trends, and customer habits that will help Spar Nord Bank improve its ATM network.

Lastly, visualization tools like Tableau are used to show the insights and suggestions that came out of the analysis in a clear and concise way. This structured method makes sure that the project's goals are met and can be used in the future to improve the bank's ATM network based on data. The dashboard containing insightful visual analysis performed is hosted on a web page.

Chapter 5 Data Engineering

5.1 Importing data

First a schema is defined to specify column names, data types, and null value allowance. The schema includes transaction and weather details, ATM information, date and time, and ATM status. It ensures data in the DataFrame conforms to the expected format and data types, avoiding errors during data processing and analysis.

```
|-- year: integer (nullable = true)
|-- month: string (nullable = true)
|-- day: integer (nullable = true)
|-- weekday: string (nullable = true)
|-- hour: integer (nullable = true)
|-- atm_status: string (nullable = true)
|-- atm_id: string (nullable = true)
|-- atm_manufacturer: string (nullable = true)
|-- atm_location: string (nullable = true)
|-- atm_streetname: string (nullable = true)
|-- atm_street_number: integer (nullable = true)
|-- atm_zipcode: integer (nullable = true)
|-- atm_lat: float (nullable = true)
|-- atm_lon: float (nullable = true)
|-- currency: string (nullable = true)
|-- card_type: string (nullable = true)
|-- transaction_amount: integer (nullable = true)
|-- service: string (nullable = true)
|-- message_code: string (nullable = true)
|-- message_text: string (nullable = true)
|-- weather_lat: float (nullable = true)
|-- weather_lon: float (nullable = true)
|-- weather_city_id: integer (nullable = true)
|-- weather_city_name: string (nullable = true)
|-- temp: float (nullable = true)
|-- pressure: integer (nullable = true)
|-- humidity: integer (nullable = true)
|-- wind_speed: integer (nullable = true)
|-- wind_deg: integer (nullable = true)
|-- rain_3h: float (nullable = true)
|-- clouds_all: integer (nullable = true)
|-- weather_id: integer (nullable = true)
|-- weather_main: string (nullable = true)
|-- weather_description: string (nullable = true)
```

Fig. 5. Dataset Schema

Two separate CSV files, atm_data_part1.csv and atm_data_part2.csv, obtained from Kaggle are combined into a single data frame, spar_df which contains all the transactional data required for analysis.

```
df = spark.read.csv("/content/drive/MyDrive/BDT_dataset/atm_data_part1.csv", header=True, inferSchema = True, schema = dfschema)

df.head()

Row(year=2017, month='January', day=1, weekday='Sunday', hour=0, atm_status='Active', atm_id='1', atm_manufacturer='NCR', atm_location='Næstved', atm_streetname='Farimagsvej', atm_street_number=8, atm_zipcode=4700,
atm_lat=55.233421325683594, atm_lon=11.763270378112793, currency='DKK', card_type='MasterCard', transaction_amount=None, service=None, message_code=None, message_text='55.229919', weather_lat=11.760919570922852,
weather_lon=2616038.0, weather_city_id=None, weather_city_name='281.15', temp=1014.0, pressure=87, humidity=7, wind_speed=260, wind_deg=None, rain_3h=92.0, clouds_all=500, weather_id=None, weather_main='light
rain', weather_description=None)

df1 = spark.read.csv("/content/drive/MyDrive/BDT_dataset/atm_data_part2.csv", header=True, inferSchema = True, schema = dfschema)

df.head()

Row(year=2017, month='January', day=1, weekday='Sunday', hour=0, atm_status='Active', atm_id='1', atm_manufacturer='NCR', atm_location='Næstved', atm_streetname='Farimagsvej', atm_street_number=8, atm_zipcode=4700,
atm_lat=55.233421325683594, atm_lon=11.763270378112793, currency='DKK', card_type='MasterCard', transaction_amount=None, service=None, message_code=None, message_text='55.229919', weather_lat=11.760919570922852,
weather_lon=2616038.0, weather_city_id=None, weather_city_name='281.15', temp=1014.0, pressure=87, humidity=7, wind_speed=260, wind_deg=None, rain_3h=92.0, clouds_all=500, weather_id=None, weather_main='light
rain', weather_description=None)

spar_df = df.union(df1)
```

Fig. 6. Merging the two CSV files into a single dataframe for analysis

5.2 Data Cleaning

Data cleaning is an essential step in the data analysis process, and it involves identifying and correcting or removing errors and inconsistencies in the data. For analyzing Spar Nord Bank ATM transaction data, data cleaning involves identifying and correcting any errors or inconsistencies in the data related to transaction amounts, dates, and locations.

Data Cleaning involved few important steps mentioned below:

a. Data validation: We checked for missing values, null values, and data types that were not consistent with the schema.
b. Removing duplicates: We removed any duplicate rows in the dataset, ensuring that each transaction is unique.
c. Removing irrelevant columns: Empty columns that had only null values in the entire dataset were removed.
d. Removing outliers: We identified and removed any transactions that were outside of the expected range, such as transactions with unusually high or low amounts.
e. Resolving inconsistencies: We resolved any inconsistencies in the data related to transaction locations by standardizing the location names and correcting any misspellings or typos.

Once the dataset was cleaned, we performed feature engineering and created a derived attribute called ‘full_date_time’ that was created by combining Year, new month, day, hour and "00" value to create timestamp in YYYYMMDDMI24HHMI format.

Further, for creating the data model (ER Diagram), we used STAR schema design. The data is separated into denormalized dimension tables and a fact table by segregating attributes pertaining to each entity type as dimension tables and quantitative measures into the fact table. Each of the tables are explained further below.

Location Dimension

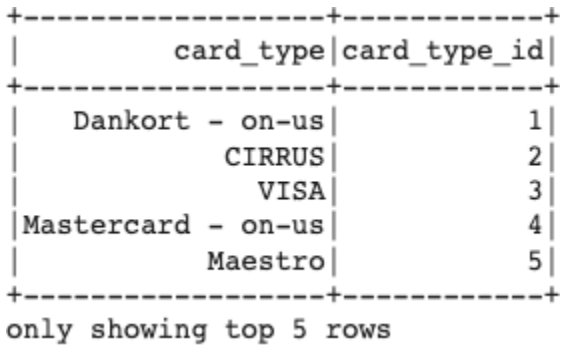
A dimension table for the location attribute was created by selecting relevant columns from transactional data including atm_location, atm_streetname, atm_street_number, atm_zipcode, atm_lat, and atm_lon, storing them in a new data frame named dim_location, and ensuring data uniqueness. This dimension table will serve as a critical data source for analysis.

location	streetname	street_number	zipcode	lat	lon
Daglig Brugsen Ø.Hurup	Kystvejen	51	9560	56.804127	10.27129
Hørning	Nørrealle	12	8362	56.086094	10.037351
Farsø	Torvet	8	9640	56.771267	9.340144
Storcenter indg. A	Hobrovej	452	9200	57.004795	9.875935
Fur	Stenøre	19	7884	56.804768	9.020021

Fig. 7. Location Dimension

Card_type Dimension

A dimension table is generated for card type with unique values of the card type column extracted from the transactional data. It also includes a primary key column labeled "card_type_id" that assigns a unique identifier to each distinct card type.



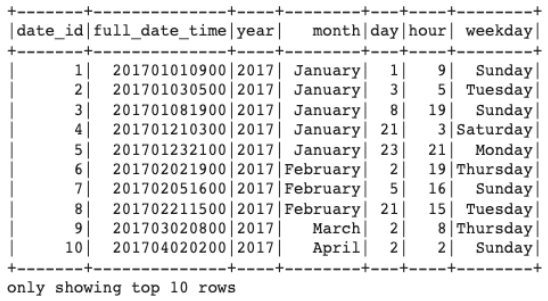
card_type	card_type_id
Dankort - on-us	1
CIRRUS	2
VISA	3
Mastercard - on-us	4
Maestro	5

only showing top 5 rows

Fig. 8. Card_type Dimension

Date Dimension

A dimension table for date is generated by selecting unique values of year, month, day, hour, and weekday from the transactional data. A new column 'month_new' is created by converting the month value to an integer. 'full_date_time' column is produced by concatenating year, month_new, day_new, hour_new, and '00', resulting in a timestamp format of "YYYYMMDD MI24HHMI". Additionally, a primary key column named 'date_id' is created for the dimension table.



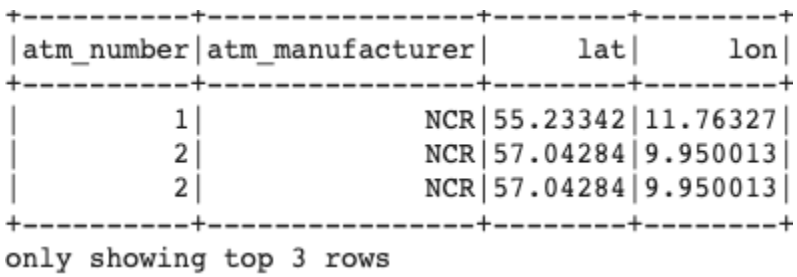
date_id	full_date_time	year	month	day	hour	weekday
1	201701010900	2017	January	1	9	Sunday
2	201701030500	2017	January	3	5	Tuesday
3	201701081900	2017	January	8	19	Sunday
4	201701210300	2017	January	21	3	Saturday
5	201701232100	2017	January	23	21	Monday
6	201702021900	2017	February	2	19	Thursday
7	201702051600	2017	February	5	16	Sunday
8	201702211500	2017	February	21	15	Tuesday
9	201703020800	2017	March	2	8	Thursday
10	201704020200	2017	April	2	2	Sunday

only showing top 10 rows

Fig. 9. Date Dimension

ATM Dimension

A dimension table for ATM data is created by selecting relevant columns such as ATM ID, manufacturer, latitude, and longitude from the transactional data.

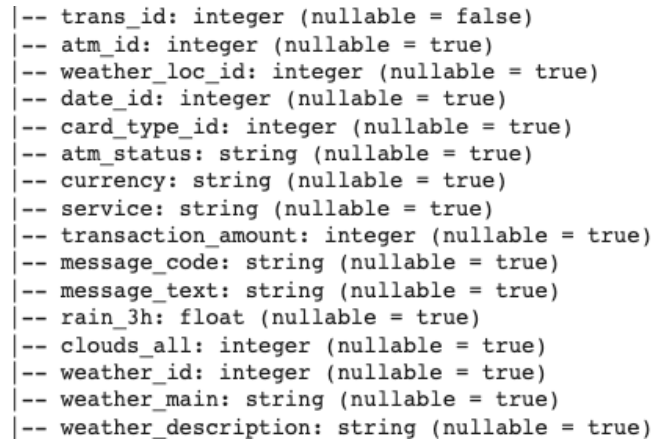


```
+----------+----------------+--------+--------+
|atm_number|atm_manufacturer|     lat|     lon|
+----------+----------------+--------+--------+
|         1|             NCR|55.23342|11.76327|
|         2|             NCR|57.04284|9.950013|
|         2|             NCR|57.04284|9.950013|
+----------+----------------+--------+--------+
only showing top 3 rows
```

Fig. 10. ATM Dimension

Fact Table

A fact table named 'fact_atm_trans' is created for the ATM transactions by joining several dimension tables with the input dataset on common columns, selecting necessary columns, and dropping duplicate columns. The dimensions include date, card type, location, and ATM. The resulting fact table conforms to a predetermined schema, which includes a primary key column named 'trans_id' and columns named 'atm_number', 'atm_manufacturer', and 'weather_loc_id'. Some columns from the input dataset, such as weather-related data, are dropped as they do not conform to the schema.



```
|-- trans_id: integer (nullable = false)
|-- atm_id: integer (nullable = true)
|-- weather_loc_id: integer (nullable = true)
|-- date_id: integer (nullable = true)
|-- card_type_id: integer (nullable = true)
|-- atm_status: string (nullable = true)
|-- currency: string (nullable = true)
|-- service: string (nullable = true)
|-- transaction_amount: integer (nullable = true)
|-- message_code: string (nullable = true)
|-- message_text: string (nullable = true)
|-- rain_3h: float (nullable = true)
|-- clouds_all: integer (nullable = true)
|-- weather_id: integer (nullable = true)
|-- weather_main: string (nullable = true)
|-- weather_description: string (nullable = true)
```

Fig. 11. Fact Table

The processed data is then exported as CSV files, with the fact table and dimension tables saved separately in distinct files, including fact.csv, atm.csv, card_type.csv, data.csv, and location.csv.

5.3 ETL process

a. Extract data

We build an ETL pipeline on AWS to perform extract, transform and load operations in order to carry out analysis of ATM transactions data for this study. This pipeline includes extracting the raw dataset stored in Amazon RDS using Sqoop to PySpark inorder to perform initial data cleaning and preprocessing. Amazon S3 which is a data lake for data storage, AWS Glue for performing ETL from S3 to Redshift and Amazon Redshift for data warehousing and to perform analysis using queries on analytical tool Redshift Query Editor provided by Redshift.

Once the initial cleaning and preprocessing of data is done, we export and load the dimension and fact tables as csv files into the S3 bucket on AWS. Same region replication of S3 bucket is enabled to ensure that the backup of the bucket exists if there is any issue with the existing bucket. S3 bucket with name ‘bigdata-technology’ is created and the csv files are loaded inside the input folder.

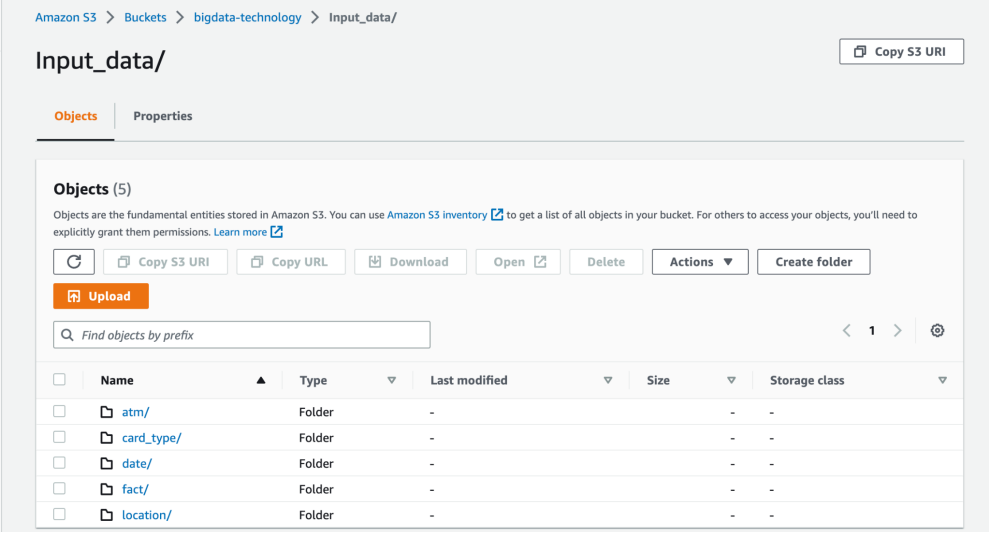


Fig. 12. S3 Bucket with data

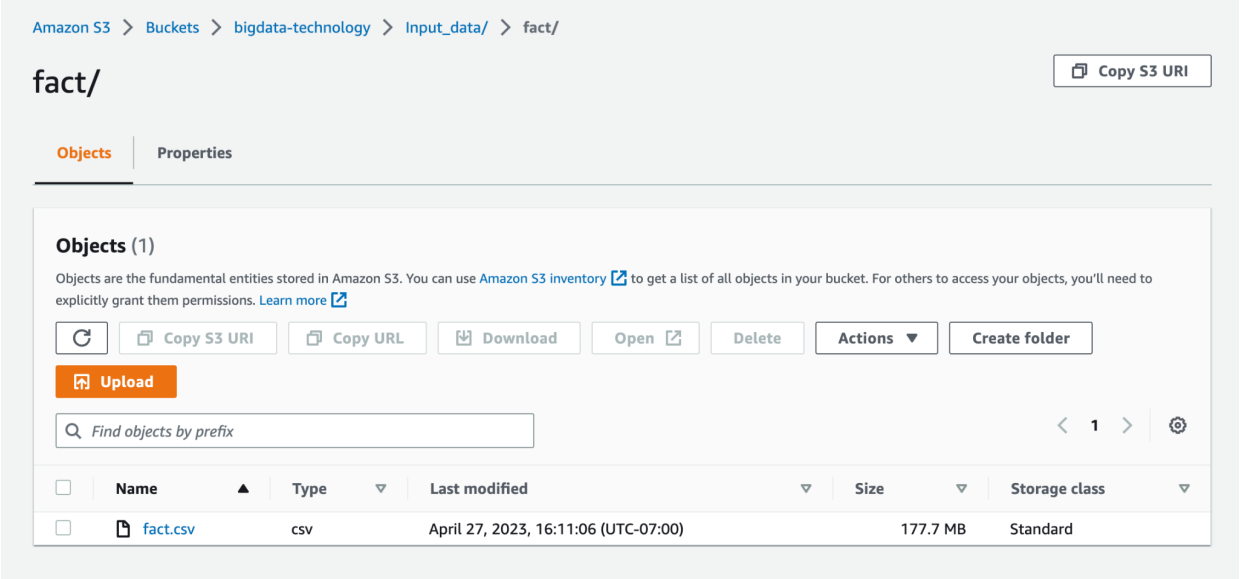


Fig. 13. Fact Table CSV

At this stage, we create a database on AWS Glue Data Catalog which is used to store the metadata and tables data after crawling from S3. A database with the name ‘bdt_proj’ is created which is as shown below.

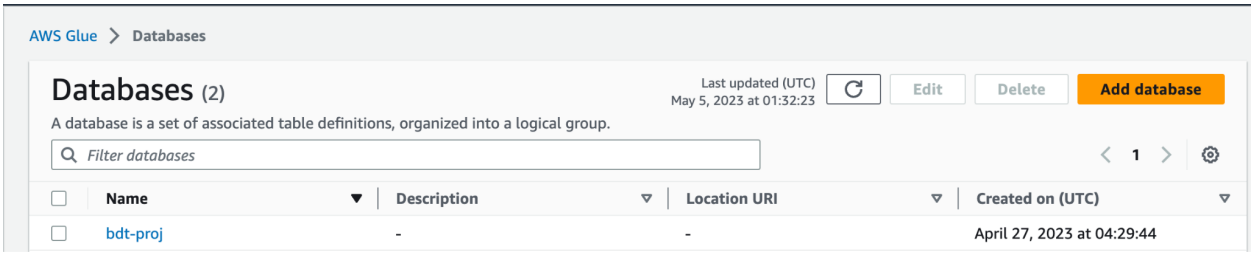


Fig. 14. Database Created on Glue Data Catalog

To crawl data from S3 bucket to Glue Data Catalog, an IAM role called 'AWSGlueServiceRole-bdt-again' is created which has sufficient permissions specific to perform this operation.

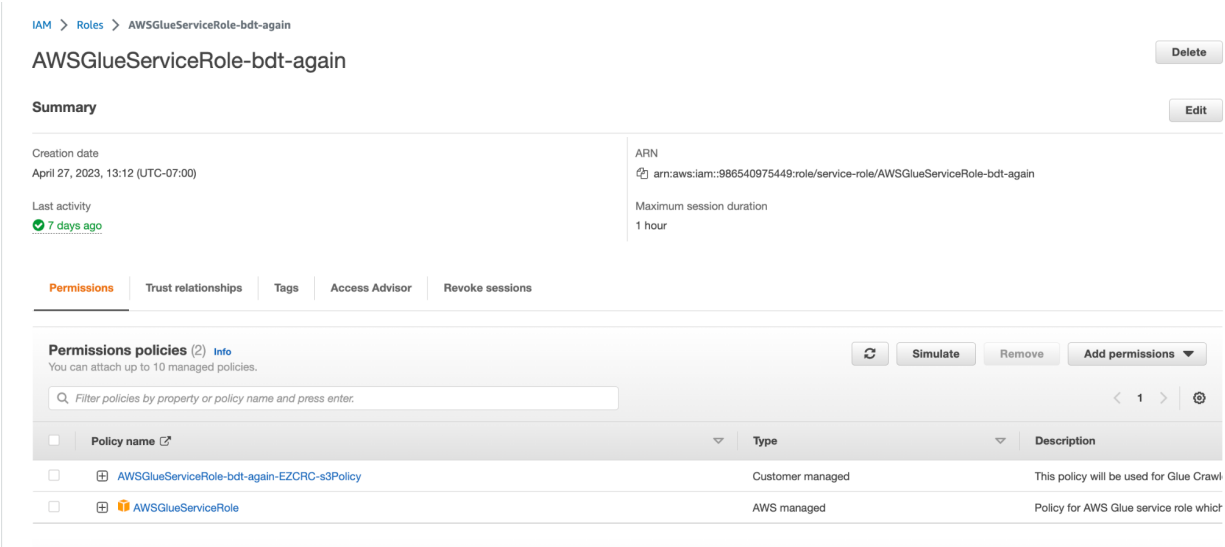


Fig. 15. IAM Role for AWS Glue

After this step, on the crawler section in Glue, a crawler is created by specifying the data store as S3 and S3 bucket location from where the data has to be crawled. Crawl all subfolders option has to be specified if all the files inside a given S3 location have to be crawled. Database

‘bdt_proj’ is specified as the crawler's output location. Below is the snapshot of the crawler created to pull the metadata and table data from S3 to Data Catalog tables.

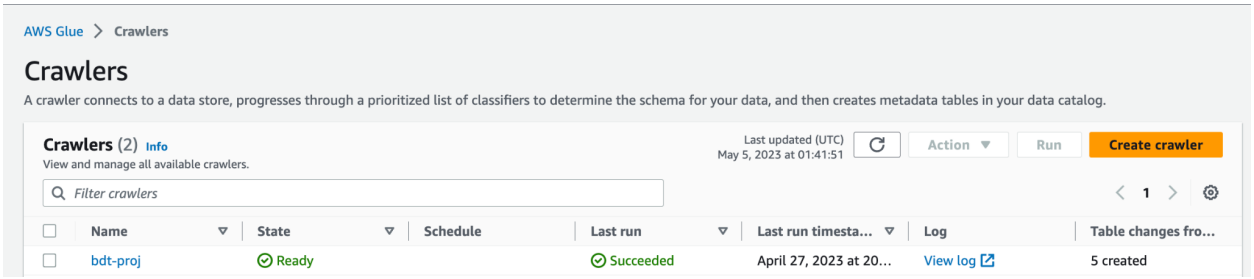


Fig. 16. Crawler after writing data to data store

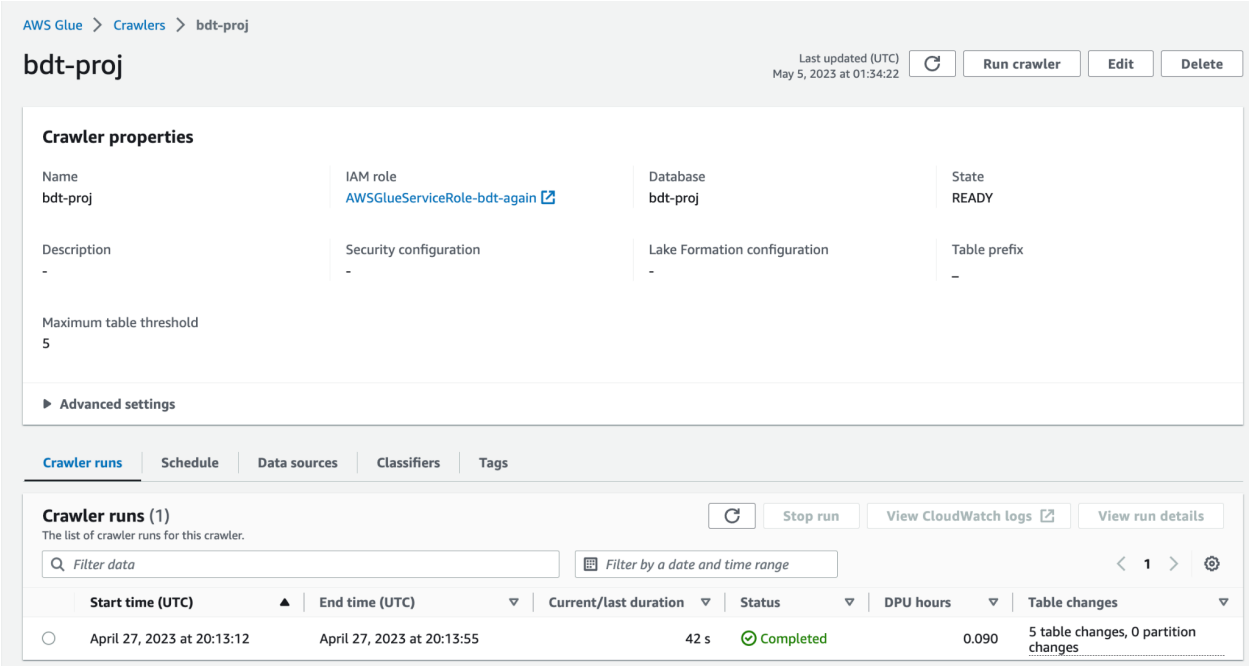


Fig. 17. Crawler run

Once the crawler successfully runs, it creates the tables and populates table data along with all other table information on Glue Data Catalog Tables. This is further verified by querying on athena to ensure that the data has been properly crawled.

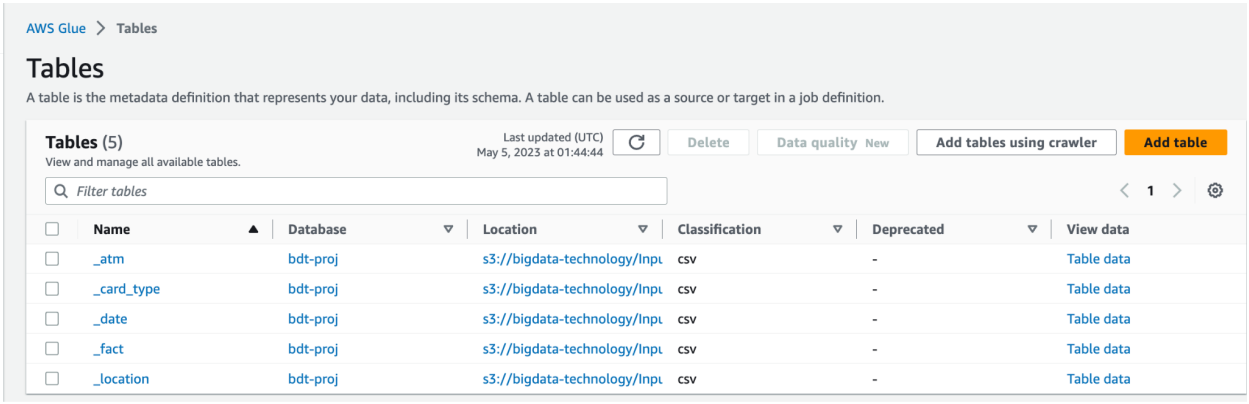


Fig. 18. Tables on Glue Data Catalog

Table metadata on Glue Data Catalog

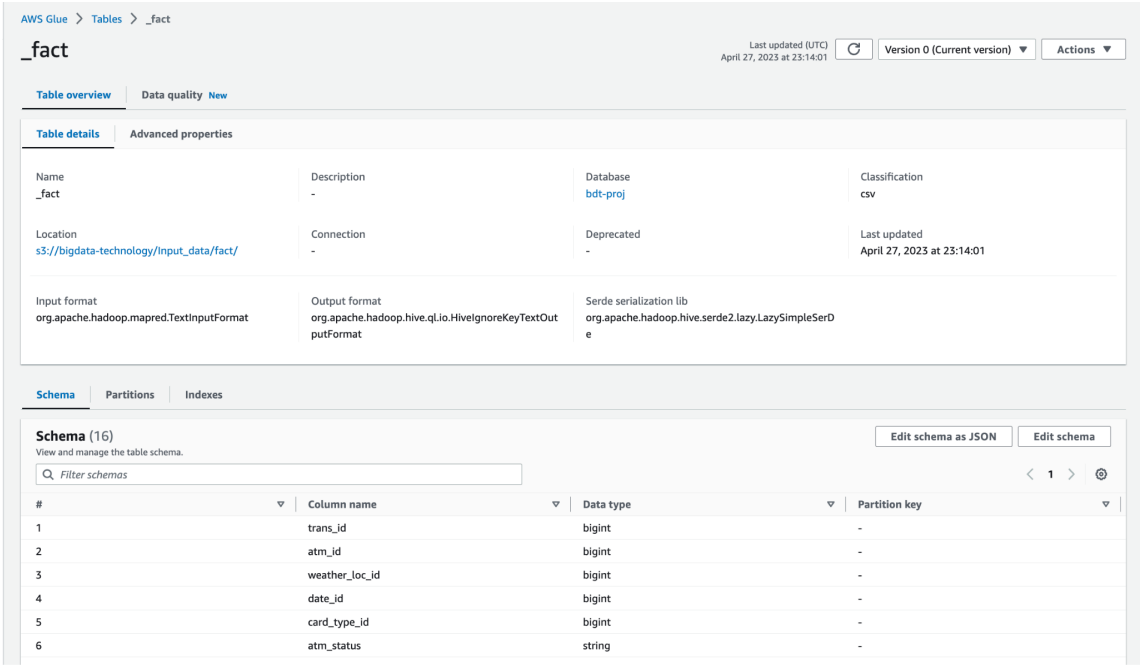


Fig. 19. Fact table metadata

Verifying the data on Athena

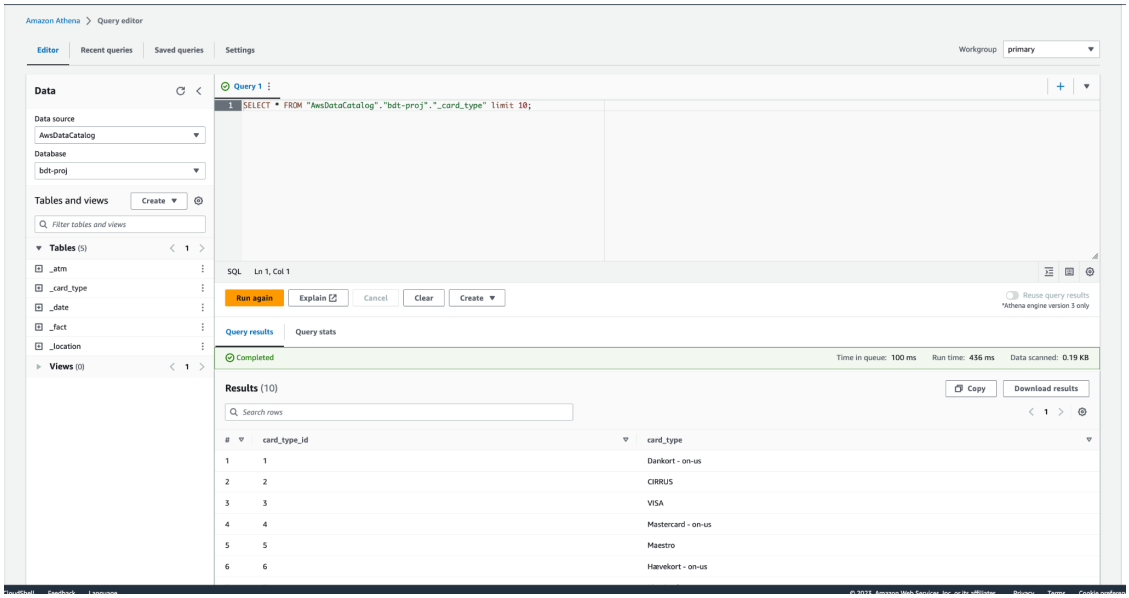


Fig. 20. Verifying data on Athena

b. Transfer data

Upon crawling the data from S3 to Glue Data Catalog tables, the data needs to be transformed into suitable format along with mapping and loaded into target tables on Amazon Redshift.

Virtual Private Cloud (VPC) and VPC Endpoint:

A secure, safe, private, and effective environment for data processing and transfer is provided by employing a VPC and a VPC Endpoint while conducting ETL from S3 to Redshift using Glue. The ETL work can be isolated from the public internet using a VPC, which also offers a safe environment for data processing. The Glue job can access resources within the VPC, such as S3 buckets and Redshift clusters, safely and privately by being launched within the VPC. Therefore, a VPC is created and a VPC Endpoint for S3 within the VPC is created. Further, inbound traffic rules and outbound traffic rules are also updated to allow the incoming and outgoing traffic from these trusted sources. This helps in minimizing unauthorized access.

In order to load the transformed data into target tables on Redshift, a Redshift Cluster of DC2 large node is created by specifying the username and password that will be used for connecting to the database.

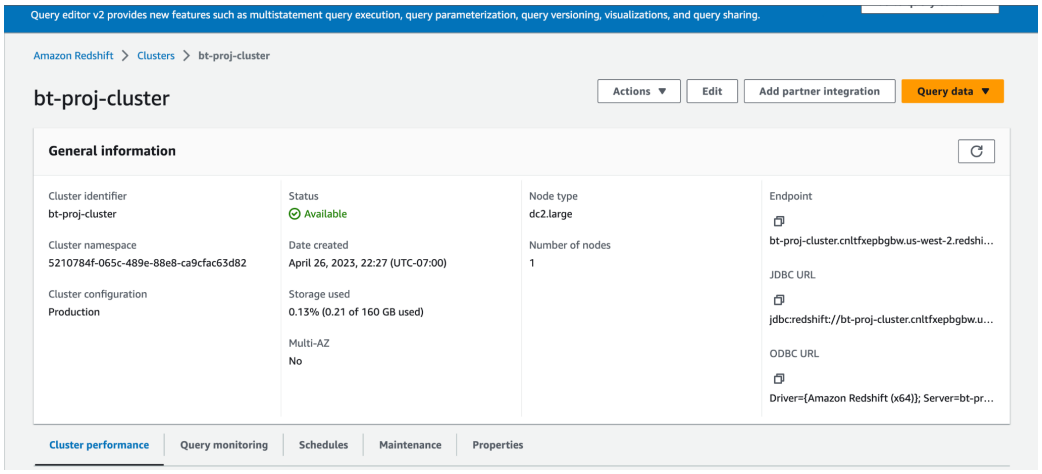


Fig. 21. Snapshot of Redshift cluster created

Further, a connection is created for connecting to Redshift by specifying the connection type as JDBC and connection parameters. Below screenshot shows the connection to redshift created. Connection to Redshift is also tested before proceeding.

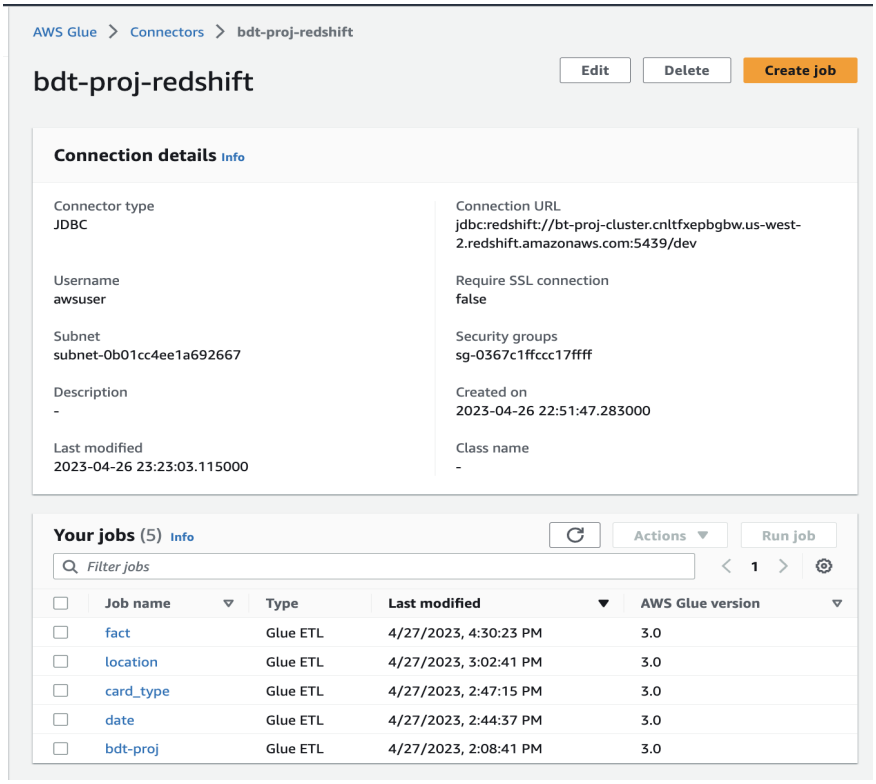


Fig. 22. JDBC Connection to Redshift

Once all the necessary steps are taken and the connection is established, ETL Glue jobs are created by providing all the parameters such as source, target, mappings that need to be applied. A glue job created for the fact table is shown below.

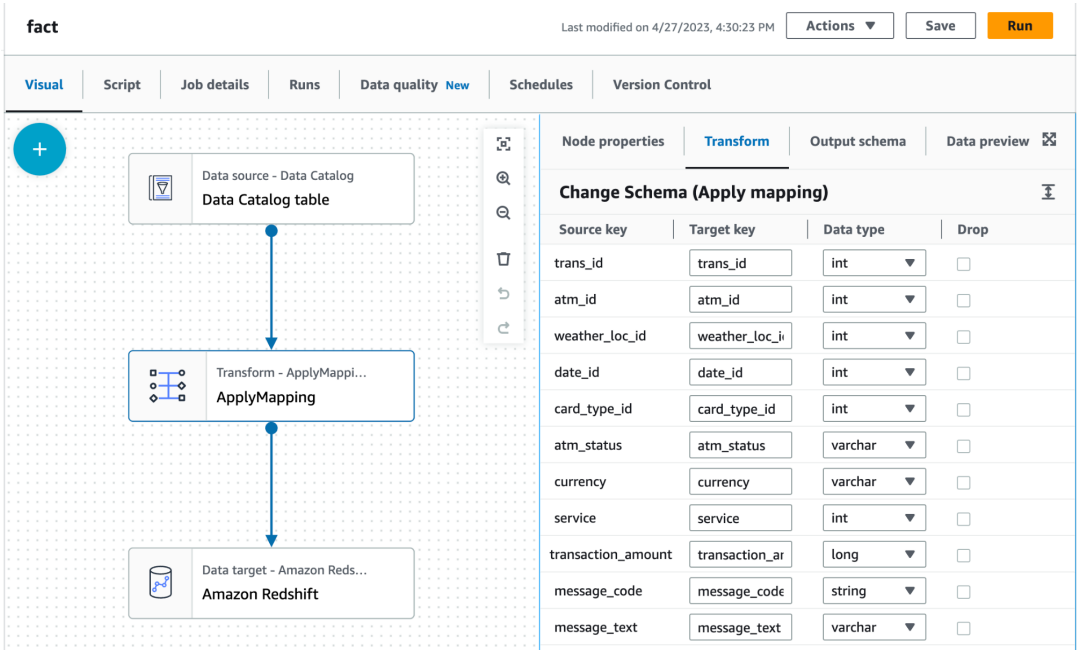


Fig. 23. Glue Job to extreme, transform and load data from S3 to Redshift

PySpark script for performing ETL - extract data from glue data catalog tables, apply necessary transformations and mappings and then load the transformed data to target tables on Redshift.

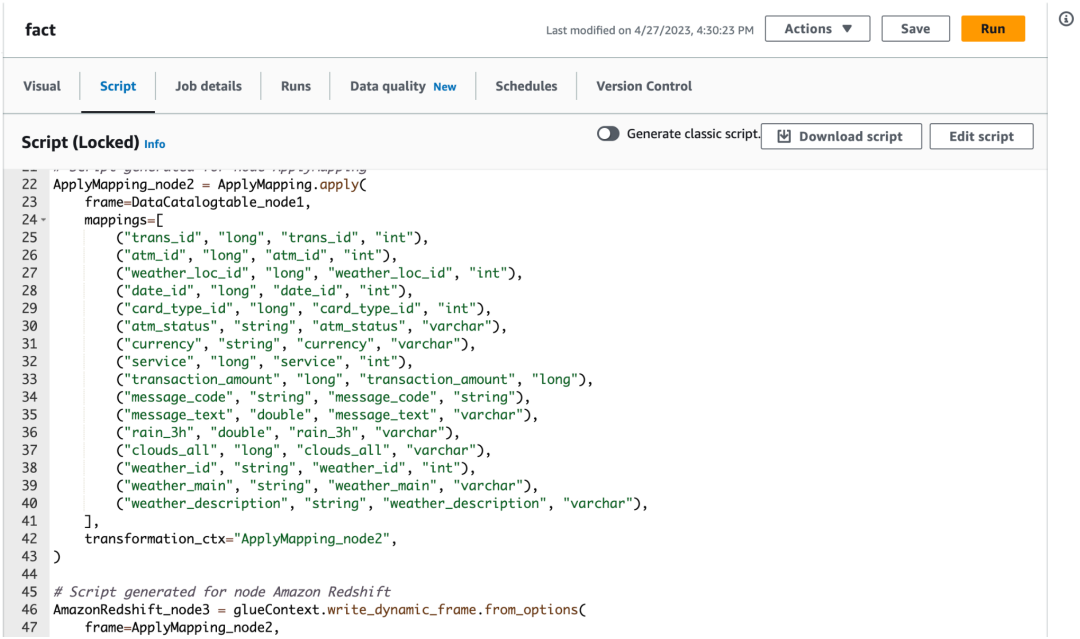


Fig. 24. PySpark Script for ETL

These glue jobs are run to carry out ETL.

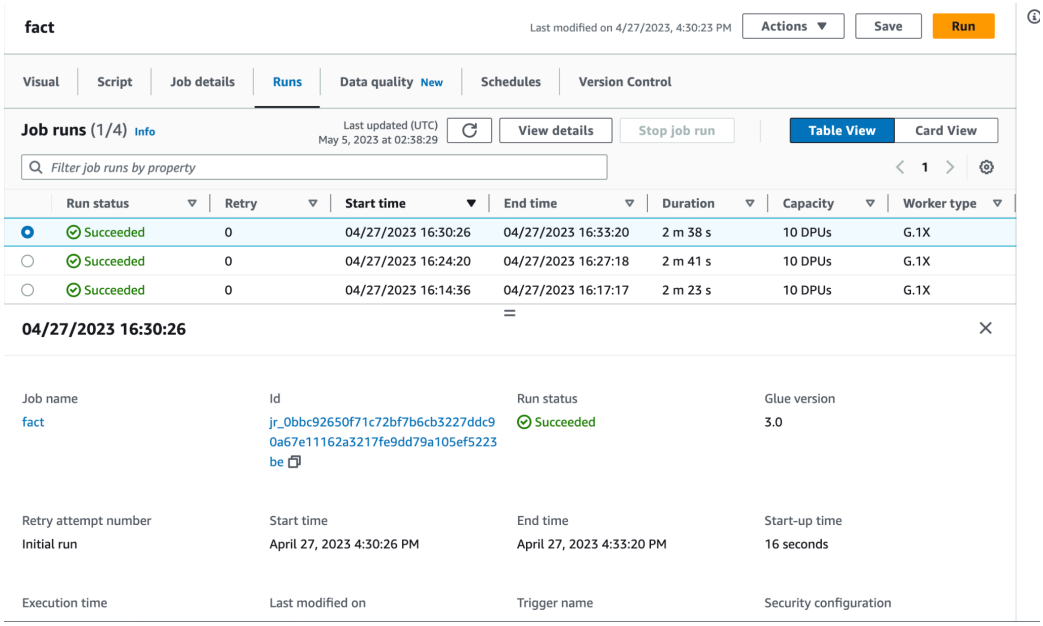


Fig. 25. Glue Jobs Run Status

c. Load data

After successfully running these glue jobs, the transformed data gets loaded into the target tables on Redshift. This can be verified on the Query Editor V2 after connecting to the database using aws username and password specified at the time of cluster creation.

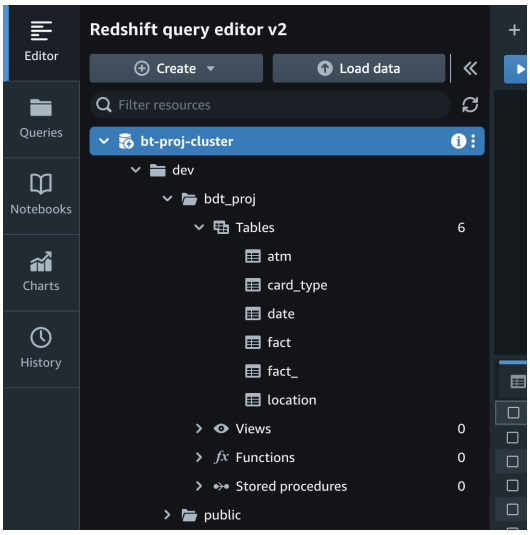


Fig. 26. Redshift Query Editor

The query editor is further used to run analytics using analytical queries on Redshift query editor which also provides an option to visualize the results as a graph or chart.

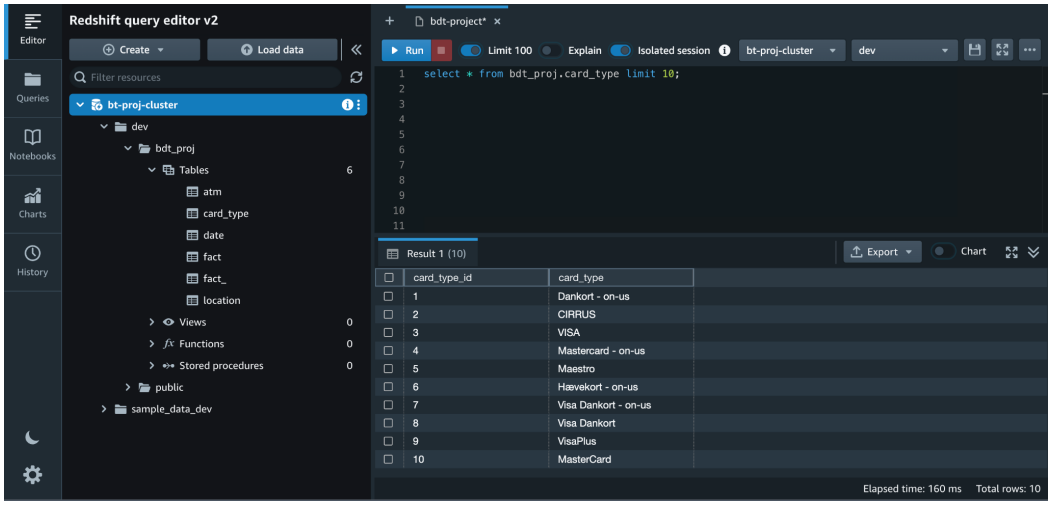


Fig. 27. Querying on Redshift Query Editor

Chapter 6 Data Analytics

6.1 Data Analytics Using Redshift

Once the data is loaded into Amazon Redshift using the Glue job, data analytics using analytical queries on Redshift Query Editor is carried out to analyze the ATM transactions data of Spar Nord bank and identify the ATM usage patterns, External climatic factors affecting the usage, trends analysis etc. Below are some of the analytics performed along with queries and their insights.

Query 1: Top 10 ATMs with the highest percentage of 'inactive' transactions

```
1  --Top 10 ATMs where most transactions are in the 'inactive' state
2  select a.atm_number, a.atm_manufacturer, l.location, count(a.atm_number) as total_transaction_count, sum(case when
   atm_status = 'Inactive' then 1 else 0 end) as inactive_count, round((inactive_count * 100.00 /
   total_transaction_count), 4) as inactive_count_percent from bdt_proj.atm a, bdt_proj.location l, bdt_proj.fact_ f
   where f.atm_id = a.atm_id and f.weather_loc_id=l.atm_location_id and f.atm_status = 'Inactive' group by a.
   atm_number, a.atm_manufacturer, l.location order by inactive_count DESC limit 10;
3
4
```

Result 1 (10) | Export | Chart

atm_number	atm_manufacturer	location	total_transaction_count	inactive_count	inactive_count_percent
16	NCR	Skive	44043	44043	100
12	NCR	Østerå Duus	33982	33982	100
2	NCR	Vejgaard	33725	33725	100
88	NCR	Storcenter indg. A	32183	32183	100
30	NCR	Nykøbing Mors	30883	30883	100
52	NCR	Farsø	27361	27361	100
50	NCR	Aarhus	23416	23416	100
29	NCR	Skelagervej 15	20773	20773	100
81	NCR	Spar Købmand Tornhøj	20148	20148	100
102	NCR	Aalborg Storcenter Afd	18297	18297	100

Fig.28. Top 10 ATMs with the highest percentage of 'inactive' transactions

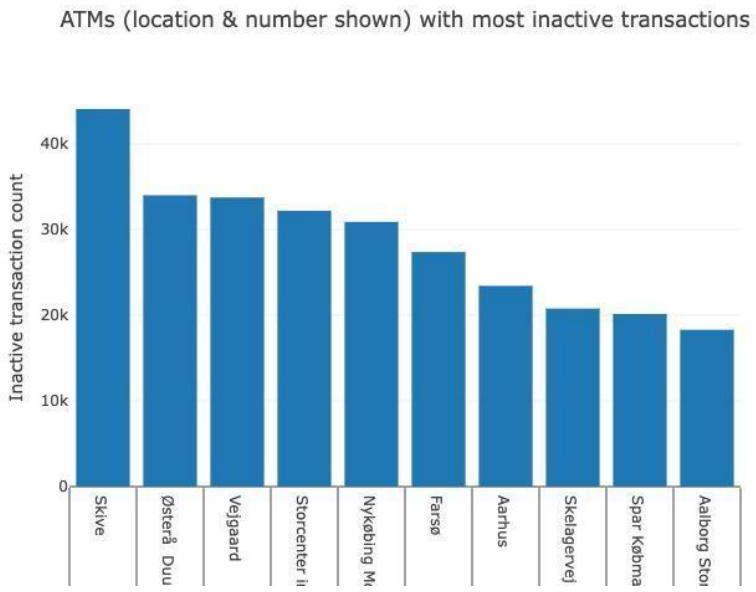


Fig.29. Bar chart of ATMs with inactive transactions

This query result shows the top 10 ATMs and their locations with the highest percentage of inactive transactions. Comparing the total number of transactions and failed ones, These ATMs shown are completely inactive which are not maintained at all since all of the transactions

occurred at these ATMs failed. This analysis is crucial for the Spar Nord bank to review the ATMs in non-working conditions that can impact the customer banking experience.

Query 2: The number of ATM failures according to the various weather conditions observed at the time of the transactions

```
--Number of ATM failures corresponding to the different weather conditions recorded at the time of the transactions
select weather_main, count(trans_id) as total_transaction_count, sum(case when atm_status = 'Inactive' then 1 else 0
end) as inactive_count, round((inactive_count * 100.00 / total_transaction_count), 4) as inactive_count_percent from
bdt_proj.fact_ where ascii(weather_main) != 0 group by weather_main order by inactive_count_percent DESC;
```

Result 1 (42) | Export | Chart

weather_main	total_transaction_count	inactive_count	inactive_count_percent
freezing rain	17	7	41.1765
light rain and snow	247	76	30.7692
sleet	268	75	27.9851
ragged shower rain	706	170	24.0793
light shower snow	4268	989	23.1724
fog	18324	3785	20.656
heavy intensity rain	2010	409	20.3483
light snow	14412	2919	20.254
light shower sleet	1011	194	19.1889
snow	3028	574	18.9564
heavy snow	49	9	18.3673
shower drizzle	413	75	18.1598
overcast clouds	130521	23569	18.0576

Elapsed time: 5674 ms Total rows: 42

Fig.30. The number of ATM failures according to the various weather conditions observed at the time of the transactions

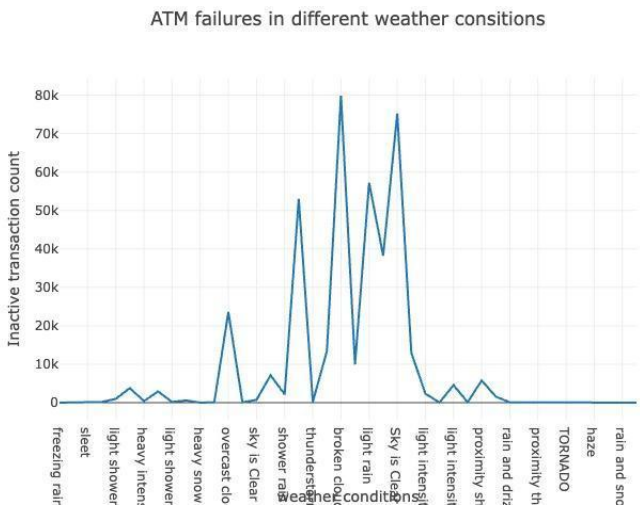


Fig.31. Graphical representation of ATM failures in different weather conditions

The above query result and the graphical representation of this analysis on Redshift Query Editor shows the trend in ATM transactions on different weather conditions. This can be used to understand the impact of weather conditions. The above graph shows that the impact is evident only in extreme weather conditions.

Query 3: Top 10 ATMs with the most transactions during the course of the year

```
9   --Top 10 ATMs with the most number of transactions throughout the year
10  select a.atm_number, a.atm_manufacturer, l.location, count(a.atm_number) as total_transaction_count from bdt_proj.
    atm a, bdt_proj.location l, bdt_proj.fact_ f where f.atm_id = a.atm_id and f.weather_loc_id=l.atm_location_id  group
    by a.atm_number, a.atm_manufacturer, l.location order by total_transaction_count DESC limit 10;
11
12
```

Result 1 (10) | Export | Chart

atm_number	atm_manufacturer	location	total_transaction_count
39	NCR	Svenstrup	55380
20	NCR	Bispensgade	54211
10	NCR	Nørresundby	53794
24	NCR	Hobro	53378
45	NCR	Abildgaard	53198
16	NCR	Skive	44043
40	Diebold Nixdorf	Frederikshavn	43767
1	NCR	Næstved	42787
41	Diebold Nixdorf	Skagen	42732
48	Diebold Nixdorf	Brønderslev	42493

Fig.32. Top 10 ATMs with the most transactions during the course of the year

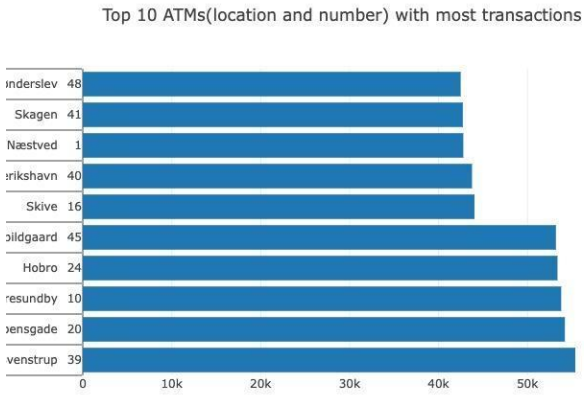


Fig.33. ATMs with most transactions

This depicts the top 10 ATMs with the most number of transactions occurring over the course of one year with ATM 39 in Svenstrup topping the list. This can be used to review and analyze if more ATM machines have to be installed and if there is a need to constantly check for maintenance and cash management of the ATMs.

Query 4: Number of overall ATM transactions going inactive per month for each month

```
13  --Number of overall ATM transactions going inactive per month for each month
14  select d.year,d.month,count(f.trans_id) as total_transaction_count,sum(case when f.atm_status = 'Inactive' then 1
    else 0 end) as inactive_count, round((inactive_count * 100.00 / total_transaction_count), 4) as
    inactive_count_percent from bdt_proj.fact_ f,bdt_proj.date d where f.date_id=d.date_id group by d.month,d.year order
    by d.month;
15
```

Result 1 (12) | Export | Chart

year	month	total_transaction_count	inactive_count	inactive_count_percent
2017	April	218865	41830	19.1122
2017	August	217218	36713	16.9015
2017	December	197048	20476	10.3914
2017	February	182659	36656	20.068
2017	January	180195	35953	19.9523
2017	July	227682	38139	16.751
2017	June	225166	36789	16.3386
2017	March	209586	41046	19.5843
2017	May	222418	37679	16.9406
2017	November	193967	21684	11.1792
2017	October	191667	21780	11.3635
2017	September	202101	28913	14.3062

Fig.34. Number of overall ATM transactions going inactive per month for each month

The above result shows the number of unsuccessful transactions in each month of 2017. This result can be utilized by the bank to analyze the root cause for these failures to better improve the customer experience by providing a fix.

Query 5: Top 10 ATMs with the highest amount of money withdrawn overall over the course of the year

```
17  --Top 10 ATMs with the highest total withdrawn amount throughout the year
18  select a.atm_number, a.atm_manufacturer, l.location, sum(f.transaction_amount) as total_transaction_amount from
    bdt_proj.atm a, bdt_proj.location l, bdt_proj.fact_ f where f.atm_id = a.atm_id and f.weather_loc_id=l.
    atm_location_id  group by a.atm_number, a.atm_manufacturer, l.location order by total_transaction_amount DESC limit
    10;
19
```

Result 1 (10) | Export | Chart

atm_number	atm_manufacturer	location	total_transaction_amount
39	NCR	Svenstrup	1684563853
20	NCR	Bispensgade	1649891475
10	NCR	Nørresundby	1638870646
24	NCR	Hobro	1635599807
45	NCR	Abildgaard	1618189048
16	NCR	Skive	1344342981
40	Diebold Nixdorf	Frederikshavn	1337510534
1	NCR	Næstved	1305315094
41	Diebold Nixdorf	Skagen	1304478636
48	Diebold Nixdorf	Brønderslev	1299715426

Fig.35. Top 10 ATMs with the highest amount of money withdrawn overall over the course of the year

The above result displays the top 10 ATMs with the highest amount of money withdrawn in 2017. ATM number 39 in Svenstrup tops the list followed by ATM number 20 in Bispensgade.

Query 6: Number of unsuccessful ATM transactions using different card types

```
21  --Number of failed ATM transactions across various card types
22  select c.card_type,count(f.trans_id) as total_transaction_count,sum(case when f.atm_status = 'Inactive' then 1 else
    0 end) as inactive_count, round((inactive_count * 100.00 / total_transaction_count), 4) as inactive_count_percent
    from bdt_proj.fact_ f,bdt_proj.card_type c where f.card_type_id=c.card_type_id group by c.card_type order by
    inactive_count_percent desc;
```

Result 1 (12) | Export | Chart

card_type	total_transaction_count	inactive_count	inactive_count_percent
Mastercard - on-us	458226	86000	18.768
VISA	170828	30713	17.9789
Dankort - on-us	143813	24680	17.1612
CIRRUS	17362	2953	17.0084
Hævekort - on-us	62487	10331	16.533
Dankort	28581	4557	15.9442
MasterCard	400507	63482	15.8504
Visa Dankort - on-us	748805	112972	15.087
Hævekort	8459	1208	14.2806
Visa Dankort	427840	60547	14.1518
VisaPlus	1134	150	13.2275

Fig.36. Number of unsuccessful ATM transactions using different card types

This query result demonstrates the number of unsuccessful ATM transactions (inactive count) that occurred from using each card type. It is evident that the transactions made using Mastercard - on-us have the most unsuccessful transactions given that the total number of transactions using this card type is also higher than the other card types. Overall looking at the failed transactions count percentage, this has a higher rate about 18.768 percent.

Query 7: Amount of transactions made annually on weekdays and weekends using an ATM

```
25  --Number of transactions happening on an ATM on weekdays and on weekends throughout the year. Order this by the
    ATM_number, ATM_manufacturer, location, weekend_flag and then total_transaction_count
26  select a.atm_number, a.atm_manufacturer, l.location,case when d.weekday='Thursday' then 1 when d.weekday='Sunday'
    then 1 else 0 end as weekend_flag, count(a.atm_number) as total_transaction_count from bdt_proj.atm a, bdt_proj.
    location l, bdt_proj.fact_ f, bdt_proj.date d where f.atm_id = a.atm_id and f.weather_loc_id=l.atm_location_id and f.
    date_id=d.date_id  group by a.atm_number, a.atm_manufacturer, l.location,weekend_flag order by a.atm_number;
27
```

Result 1 (100) — Export — Chart

atm_number	atm_manufacturer	location	weekend_flag	total_transaction_count
1	NCR	Næstved	1	11154
1	NCR	Næstved	0	31633
2	NCR	Vejgaard	1	8641
2	NCR	Vejgaard	0	25084
3	NCR	Ikast	0	10019
3	NCR	Ikast	1	3621
4	NCR	Svogerslev	0	25151
4	NCR	Svogerslev	1	8953
5	NCR	Nibe	1	4683
5	NCR	Nibe	0	14058
6	NCR	Fredericia	0	16840

Elapsed time: 293 ms Total rows: 100

Fig.37. Amount of transactions made annually on weekdays and weekends using an ATM

The above query result shows the total number of transactions made at each ATM during a weekday and a weekend. It is clearly seen that the amount of transactions occurring during a weekend are more than double the amount of transactions occurring on a weekday.

Query 8: Most active day (in terms of transactions) in each ATM in Vejgaard location

```
29  --Most active day in each ATMs from location "Vejgaard"
30  select a.atm_number, a.atm_manufacturer, l.location,d.weekday, count(a.atm_number) as total_transaction_count from
    bdt_proj.atm a, bdt_proj.location l, bdt_proj.fact_ f, bdt_proj.date d where f.atm_id = a.atm_id and f.
    weather_loc_id=l.atm_location_id and f.date_id=d.date_id and l.location = 'Vejgaard' group by a.atm_number, a.
    atm_manufacturer, l.location, d.weekday order by d.weekday, total_transaction_count limit 2;
```

Result 1 (2) — Export — Chart

atm_number	atm_manufacturer	location	total_transaction_count	weekday
103	Diebold Nixdorf	Vejgaard	4757	Friday
2	NCR	Vejgaard	6290	Friday

Fig.38. Most active day (in terms of transactions) in each ATM in Vejgaard location

The above result shows that Friday is the most active weekday at the ATMs located in Vejgaard. There are two ATMs with number 103 and 2 in this particular location.

Query 9: During what time frame most ATM transactions occurred

```
32  -- Amount of transactions at ATMs by each hour sorted in descending order
33  select distinct d.hour, count(*) as no_of_transactions from bdt_proj.date d, bdt_proj.fact_ f where d.date_id = f.
    date_id group by d.hour order by no_of_transactions desc;
```

Result 1 (24) | Export | Chart

hour	no_of_transactions
11	256542
10	249928
12	237107
13	233322
14	230270
15	223323
16	196004
17	153816
9	150943

Fig.39. During what time frame most ATM transactions occurred

In the above query results, we can see that the highest number of transactions occurred within the time frame of 10 AM to 12 PM which starts to decline gradually towards evening. ATM maintenance and refilling can be done by the bank during evening times when the number of ATM transactions occurring reduces.

Chapter 7 Visualization

7.1 Data Visualization

The data visualization process was crucial in helping us understand the data patterns and trends in the ATM transactions. Using Tableau, we were able to create visualizations that provided valuable insights into customer behavior and preferences.

Figure 40 shows that the type of card used at ATMs was used to do the research. The graph shows that the most popular card was Visa Dankort, which is a well-known debit card in Denmark. Maestro cards, on the other hand, were used the least. This information can help the

bank figure out how different kinds of cards are used so they can improve their services.

No of transactions for each type of card used at ATMs

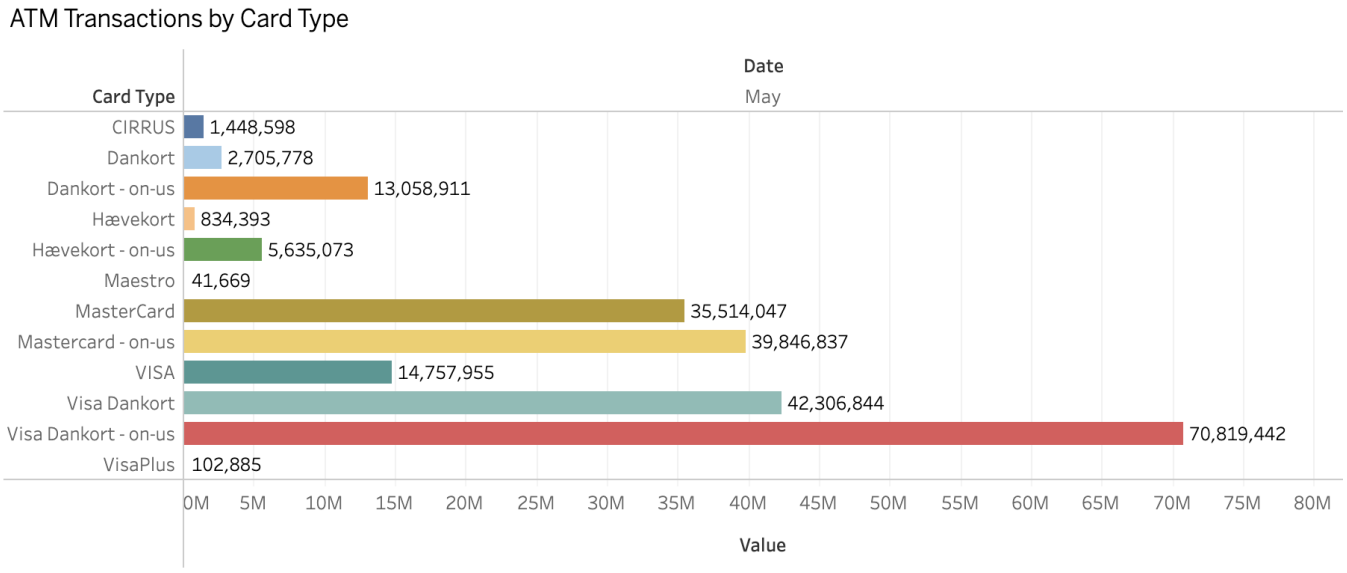


Fig.40. No of transactions for each type of card used at ATMs

Figure 41 shows that NCR has the most ATM activities, and Diebold Nixdorf is in the following position. This visualization is helpful because it shows the bank which manufacturers are used most often by customers and which ones may need more upkeep or updates to work better and have a higher success rate. It can also help the bank figure out if there are any patterns or trends in the state of ATMs. This can help improve the customer experience by making it less likely that transactions will be canceled or time out.

ATM Transactions by Manufacturer and their Status

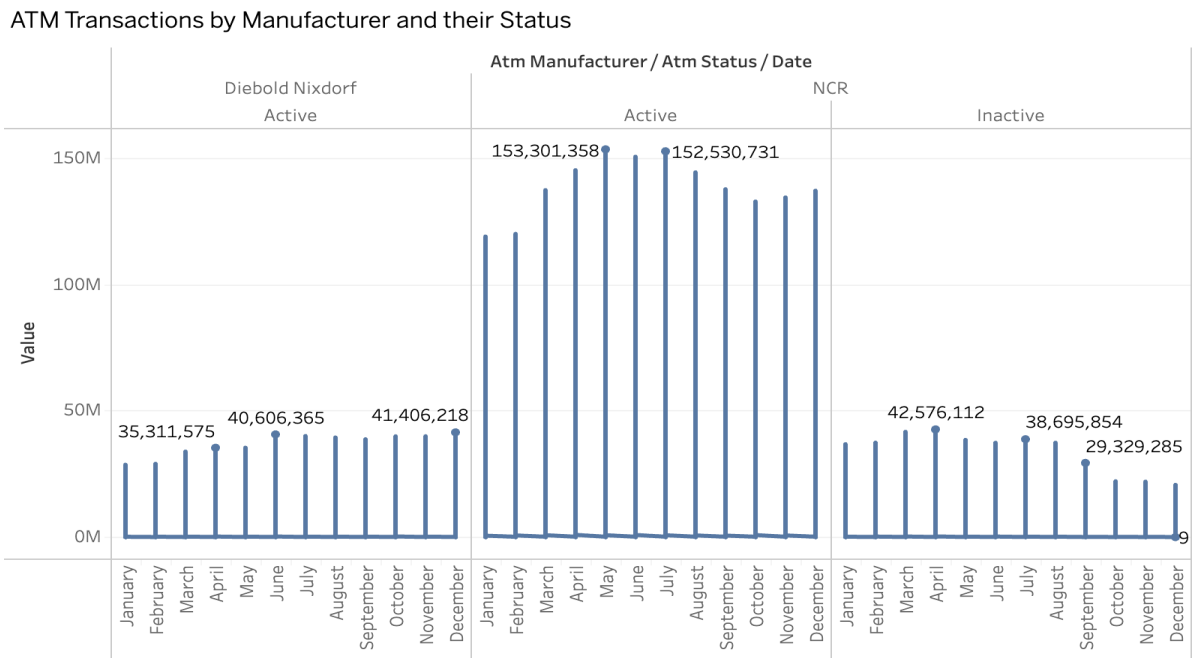


Fig.41. ATM Transactions by Manufacturer and their Status

Figure 42 is a line chart that shows how the number of deals per day has changed over time. With this representation, we could see patterns and trends in how customers act. From the picture, we learned that the busiest time for ATM transactions was between 10 am and 12 pm. As the day went on, the number of transactions slowly went down.

The bank can use this information to make the most of their resources by scheduling ATM refills outside of busy hours. This cuts down on the time customers have to wait and makes the bank more efficient overall. It also helps the bank make better use of their resources during busy times so that their customers can use ATMs without any problems.

Figure 42

A line chart showing the total number of transactions per day over time.

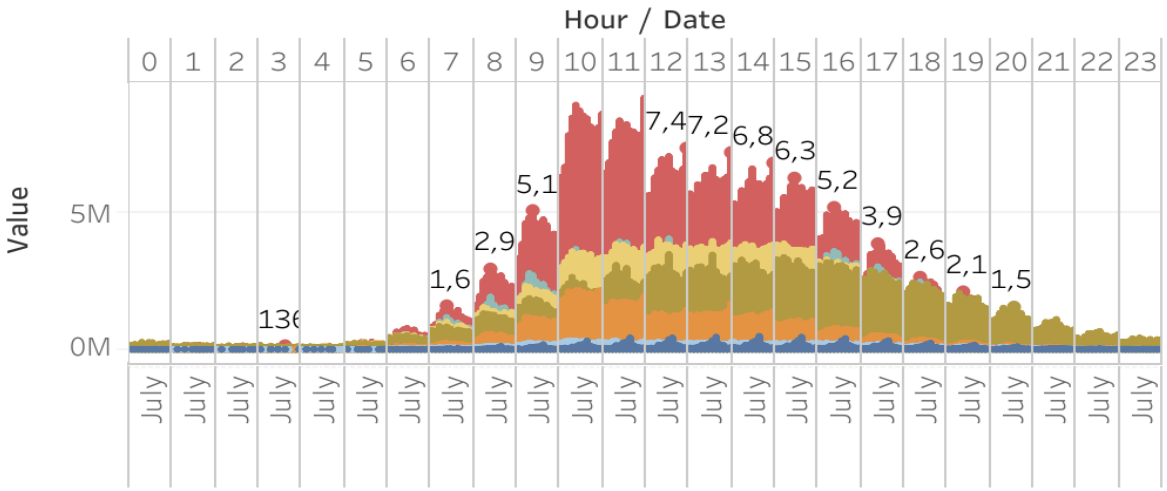


Fig.42. Daily ATM Transactions Over Time

Figure 43 shows the impact of weather conditions on ATM usage. To visualize this, we created a chart showing the average temperature and humidity over time, overlaid with the total number of transactions per day.

This insight can help the bank in making decisions about ATM placement and maintenance, taking into account weather patterns and their impact on customer behavior. For example, during hot summer months, the bank may consider adding additional ATMs or increasing the frequency of ATM refills to meet increased demand.

Figure 43

Weather Conditions Over Time

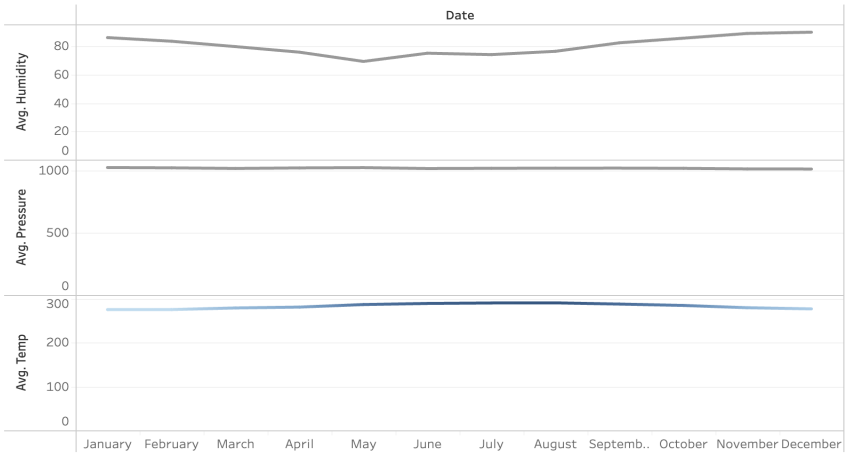


Fig.43. Weather Patterns and Their Impact on ATM Usage

7.2 Web Application to host dashboard

This document outlines the necessary steps for developing a user interface with React and Firebase. React is a widely used front-end framework, while Firebase is a Google platform that provides various services for mobile and online app development. By utilizing JavaScript functions called components to generate HTML elements, code maintenance is simplified, and component reuse is made possible. We will also demonstrate how to utilize Firebase Authentication for Single Sign-On management and install the Firebase library through the npm package manager for your course project. All pages of the app and necessary styling were created utilizing React components.

The application uses Firebase Authentication to manage user login. Users can either log in with their email address and password or create a new account. After logging into the Firebase dashboard, select the appropriate service from the list to integrate into the project as shown in the figure 44. To integrate Firebase, install npm package manager to get access to the Firebase library. This library offers a wide range of tools and services, one used in this project being the Firebase Authentication service for the user login.

Figure 44

Firebase Project Dashboard

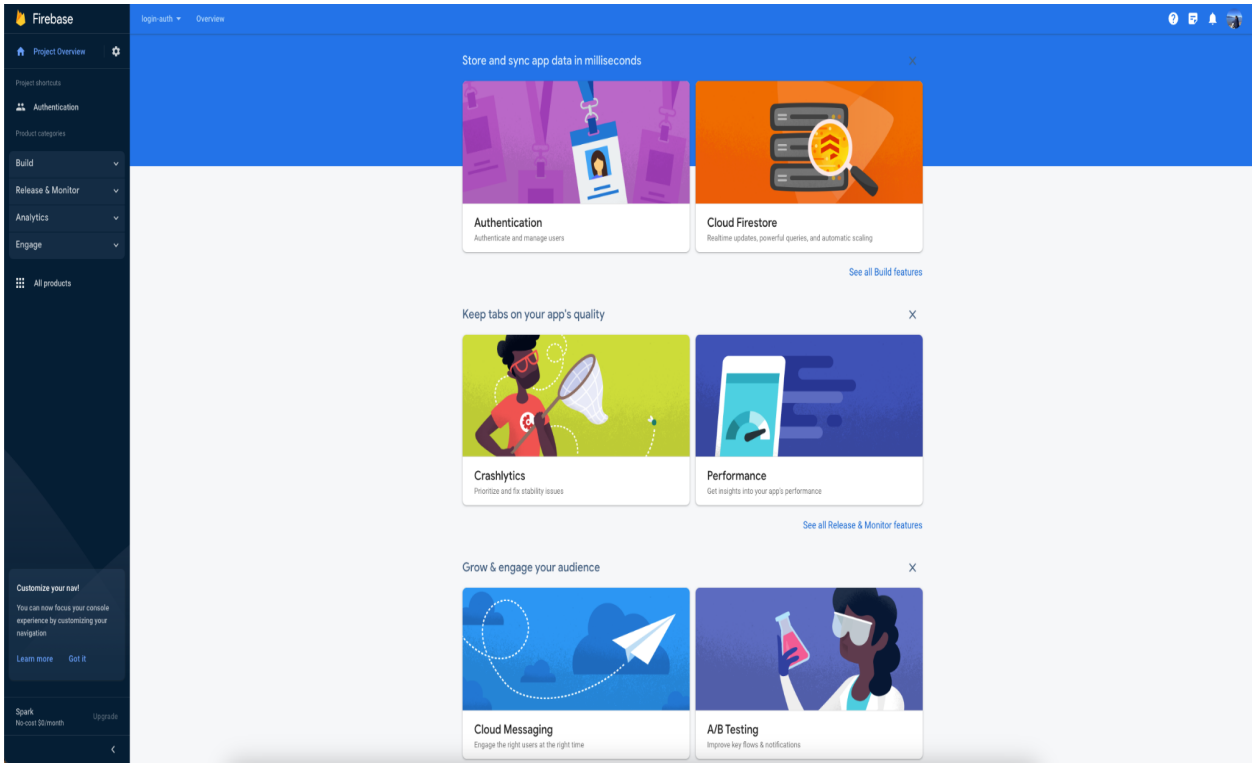


Fig.44. The Firebase project dashboard

This application includes only a few of the many services that can be added to React apps to improve their functionality. These services can provide features such as data storage and retrieval, notifications, and cloud processing. To speed up the development process, various libraries and tools can be used. The Firebase SDK for React enabled communication between the Firebase and React components. Additionally, the React Firebase Hooks package simplifies access to Firebase services in functional components. To ensure worldwide access, a content delivery network (CDN) with fast load times is employed to launch the React application.

A content delivery network (CDN) with quick load times is employed to launch the React application, enabling worldwide access. Users from all around the world may easily set up and

utilize this service. Web application development may be significantly streamlined by combining Firebase with React.

Figure 45 shows the authentication section of the application. The columns represent:

Identifier - The unique identifier for the user. This is typically the user's email address.

Providers - A list of the providers that the user has signed in with. This could include Google, Facebook, Twitter, or any other supported provider.

Created - The date and time that the user account was created.

Signed In - The date and time that the user last signed in.

User UID - The unique identifier for the user in the Firebase database. This is a 128-bit hexadecimal string

Figure 45

Application Authentication Settings

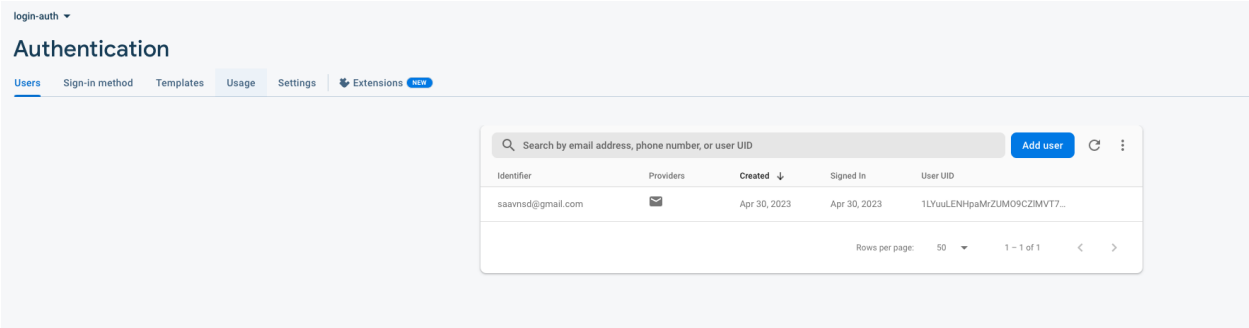


Fig.45. The application's authentication settings

Chapter 8 Collaboration

8.1 GitHub Repository

Source code, detailed project documentation (report), analytical queries used, project presentation ppt and detailed project plan are uploaded to the GitHub repository below.

GitHub Repository URL: https://github.com/rasho330/DATA-228-Big-Data-Tech-App

8.2 Team Members and IDs

Name	Student ID
Bhavana Prasad Kote	016044899
Jay Dattoo Dale	016646279
Rashmi Shree Veeraiah	016099395
Sawan Shivanand Beli	016522662

Chapter 9 Conclusion and Future Work

9.1 Conclusion

This project aims to improve how Spar Nord Bank manages its ATMs and how people use them by analyzing its transactional data. The project uses different AWS services to extract, transform, and load data. The data is then analyzed to learn about customer behavior, find patterns and trends, and make suggestions for improving the bank's ATM network.

Analysis of Spar Nord Bank's ATM transactional data has provided insightful information that can be utilized to optimize the bank's ATM network. Our study revealed that the hours between 10 AM - 12 PM have the highest transaction volumes, indicating that the bank can utilize other time periods to refill the ATMs.

Additionally, we found that Visa cards were the most used, followed by Mastercard, which can inform the bank's partnership strategies and marketing campaigns. Additionally, we found that the temperature outside had a significant impact on ATM usage, indicating that the bank can

take steps to improve customer comfort at the ATM locations.

9.2 Future Work

Even though analysis gives Spar Nord Bank's ATM network valuable data, there are still a number of places that could be looked into further. One possible direction for future study is to look at how features of ATMs like the ability to make deposits, language options, and accessibility options affect how people use them. By finding out which ATM features are most important to customers, the bank can decide where to put its money to best meet customers' needs.

Another thing that could be done in the future is to add demographic information about ATM users, such as their age, gender, and income, to the study. This could give us more information about which types of customers are most likely to use ATMs, when, and for what. By making ATM services fit the needs of different types of customers, the bank can improve the customer experience and possibly bring in new ones.

References

Teddy, S. D., & Ng, S. K. (2010). Forecasting ATM cash demands using a local learning model of cerebellar associative memory network. International Journal of Forecasting, 27, 760-776. https://www.sciencedirect.com/science/article/abs/pii/S0169207010000828

https://www.sparnord.com/

Rajwani, A., Syed, T., Khan, B., & Behlim, S. (2017). Regression analysis for ATM cash flow prediction. In Proceedings of the International Conference on Frontiers of Information Technology (pp.212-217). https://www.semanticscholar.org/paper/Regression-Analysis-for-ATM-Cash-Flow-Prediction-Rajwani-Syed/a06f4f1468563d99312300f4728a64d1e84d2aa0

Genevois, M. E., Celik, D., & Ulukan, H. Z. (2015). ATM location problem and cash management in automated teller machines. International Journal of Social, Behavioral, Educational, Economic, Business and Industrial Engineering, 9(7), 2399-2403. https://www.semanticscholar.org/paper/ATM-Location-Problem-and-Cash-Management-in-Teller-Genevois-Celik/118d37406827831011620c2648400ceabd89d371

2.5M Danish ATM Transactions from 2017. (2019, January 16). Kaggle.https://www.kaggle.com/datasets/sparnord/danish-atm-transactions