DATA WAREHOUSE ON SALES DATA IN AMAZON REDSHIFT

Aditya Ghatol

1.INTRODUCTION

1.1 Amazon Web Services (AWS)

Amazon Web Services (AWS) is a comprehensive cloud computing platform offering a variety of services, including storage, computing, and analytics. AWS provides scalable solutions that allow businesses to process and analyze vast datasets efficiently, enabling faster and more informed decision-making.

1.2 Amazon S3 (Simple Storage Service)

Amazon S3 is a scalable, high-speed, web-based cloud storage service designed to store and retrieve large volumes of data such as files, images, backups, and database dumps. S3 ensures high durability and availability, making it ideal for storing both small and large-scale datasets. It serves as a reliable storage backend, supporting businesses by securely storing data for analytics, archiving, and disaster recovery.

1.3 Amazon Redshift

Amazon Redshift is a fully managed, petabyte-scale, cloud-based data warehouse service. It is optimized for running complex SQL queries and performing advanced analytics on large datasets. Redshift provides fast query execution through parallel processing, making it a powerful tool for data-driven decision-making. Its seamless integration with other AWS services, such as S3, makes it highly effective for managing large-scale data workflows.

1.4 Why Use Amazon S3 and Amazon Redshift

- Amazon S3 is used as a central storage solution for raw data because of its ability to handle large amounts of data with high reliability and scalability. Data stored in S3 is easy to access, secure, and cost-effective, making it ideal for long-term storage and backup.
- Amazon Redshift was chosen for this project to efficiently manage and analyze sales data at scale. Its performance-optimized architecture enables fast querying and reporting, even on large datasets, helping to uncover insights critical to business operations. Additionally, Redshift's ability to integrate with S3 makes it easy to import data for analysis.

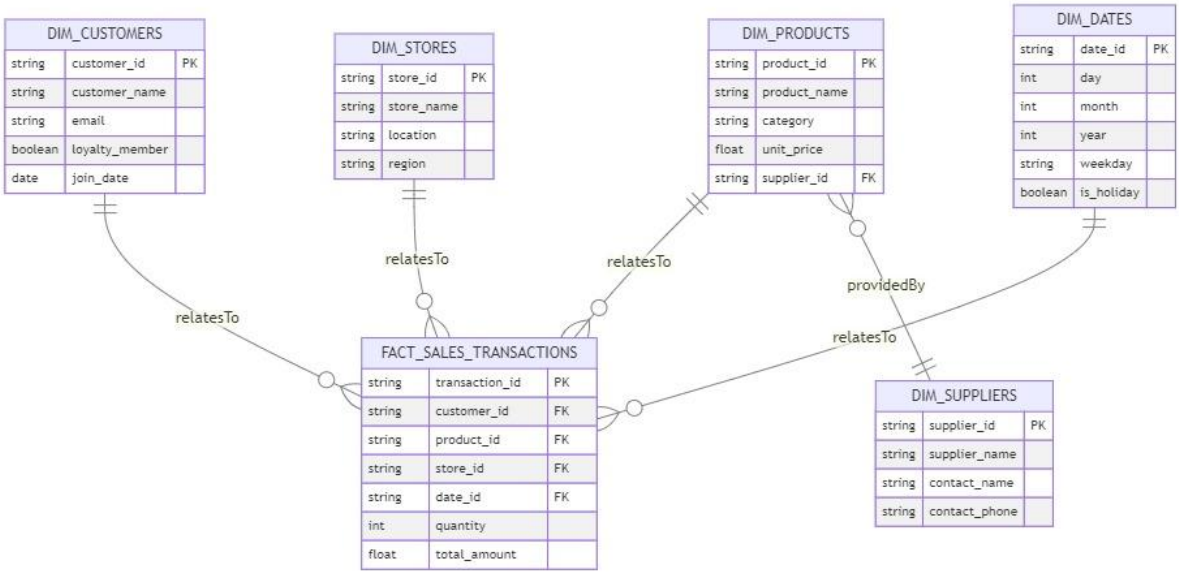
2. Problem Definition

The Sales Management System Data Warehouse project aims to efficiently manage and analyze large volumes of sales-related data by implementing a scalable data warehouse architecture using Amazon Redshift. As businesses generate vast amounts of data, including sales transactions, product performance metrics, store operations, and customer behaviors, managing and extracting meaningful insights from this data becomes challenging.

This project utilizes a star schema design to organize data, ensuring faster query execution and streamlined reporting. By structuring the data warehouse effectively, the system enables businesses to gain insights into sales trends, product performance, customer behavior, and store performance, helping them make informed decisions about inventory management, promotions, and customer engagement strategies.

The report outlines the implementation of the data warehouse using Amazon Redshift, with the raw sales data stored in Amazon S3 in CSV format. The integration between S3 and Redshift ensures smooth data loading and management. The system supports complex queries on sales and customer behavior, providing actionable insights that enhance operational efficiency and boost profitability.

3. Star Schema



4. Implementation Steps

3.1 Amazon Redshift Cluster Setup

1. **Cluster Configuration**:

- A Redshift cluster was set up with nodes optimized for sales data analysis.
- The cluster configuration was based on the expected data volume and query complexity to ensure high performance.

2. **VPC and Security**:
   - The cluster was deployed within a Virtual Private Cloud (VPC) to restrict access to only trusted entities.
   - **Security groups** were configured to ensure safe connectivity with external tools.
3. **IAM Role Creation**:
   - An IAM role was created to grant Redshift access to data stored in Amazon S3**.**
   - This allowed for seamless data transfer between S3 and Redshift during data loading.

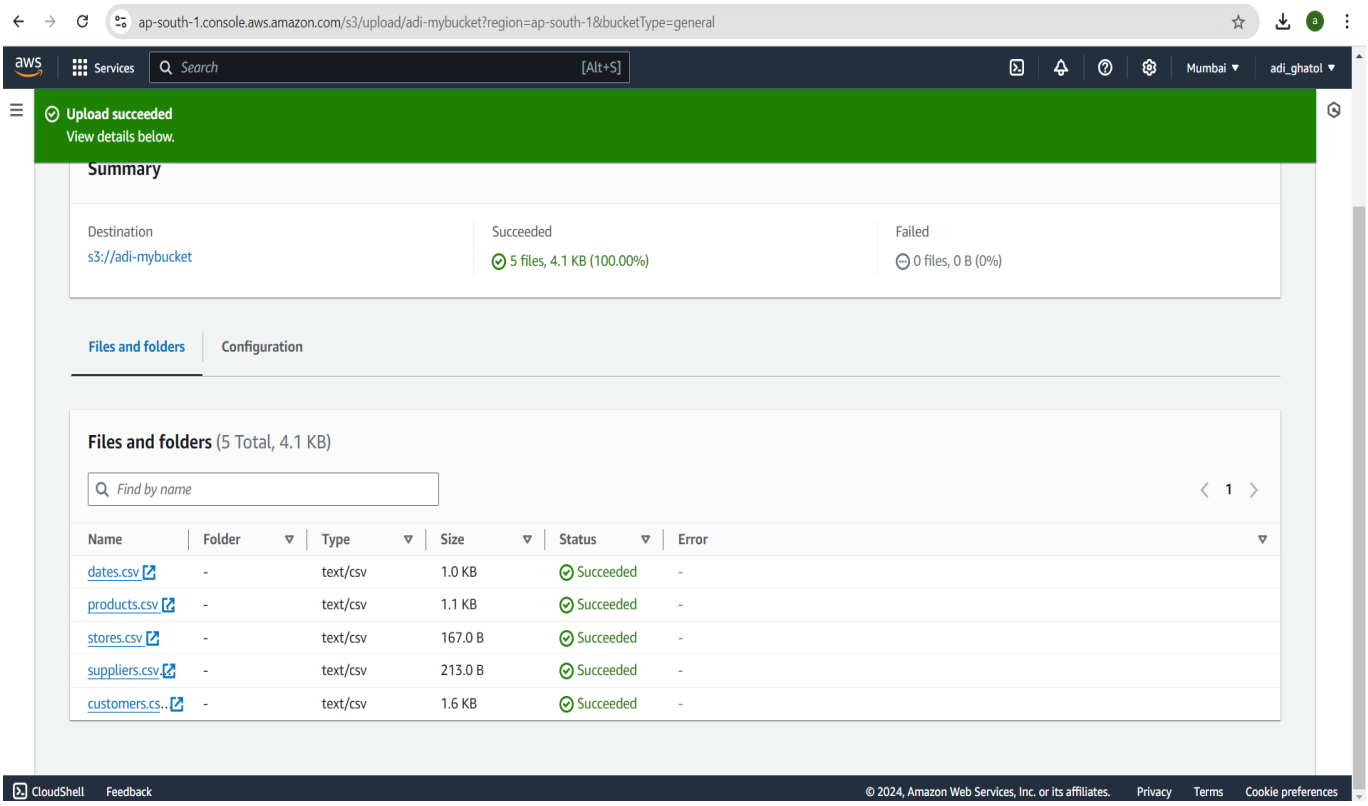
- **Fact Table**:
  - **sales_transactions**: Stores transactional data, including sales amount, store ID, product ID, and customer ID.
- **Dimension Tables**:
  - **customers**: Contains customer details (e.g., name, location, and demographics).
  - **products**: Describes products, including product name, category, and price.
  - **stores**: Stores details of each store, such as store name, location, and size.
  - **dates**: Holds the date and time information for each sales transaction.

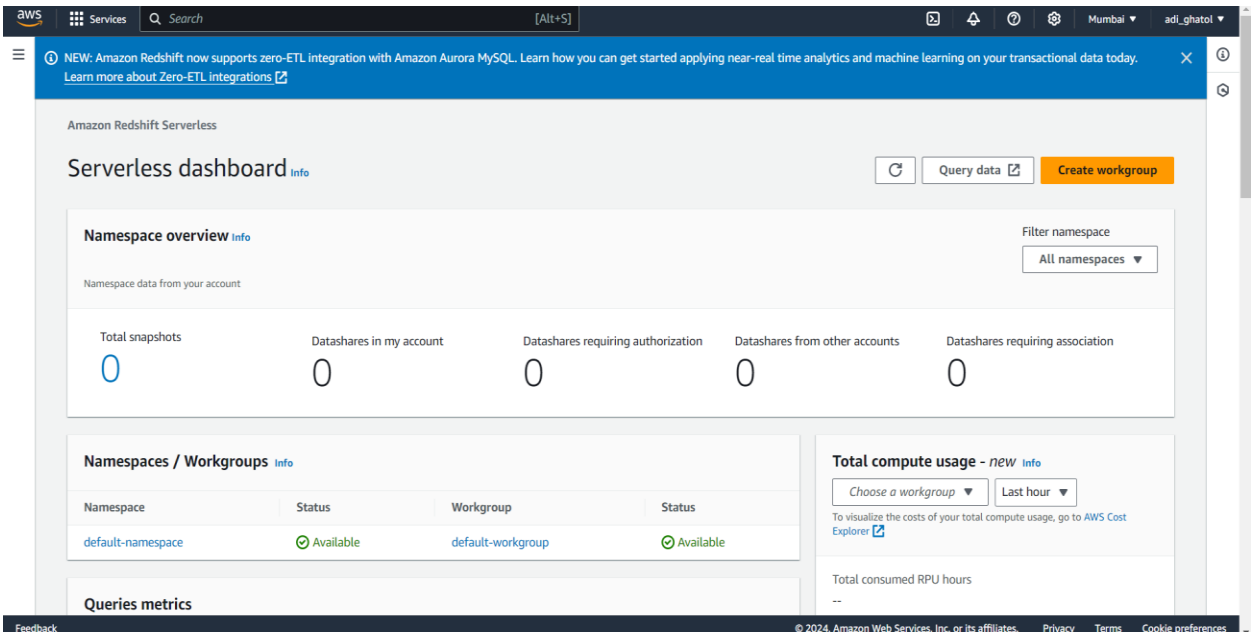
3.2 Setting Up Amazon Redshift Cluster

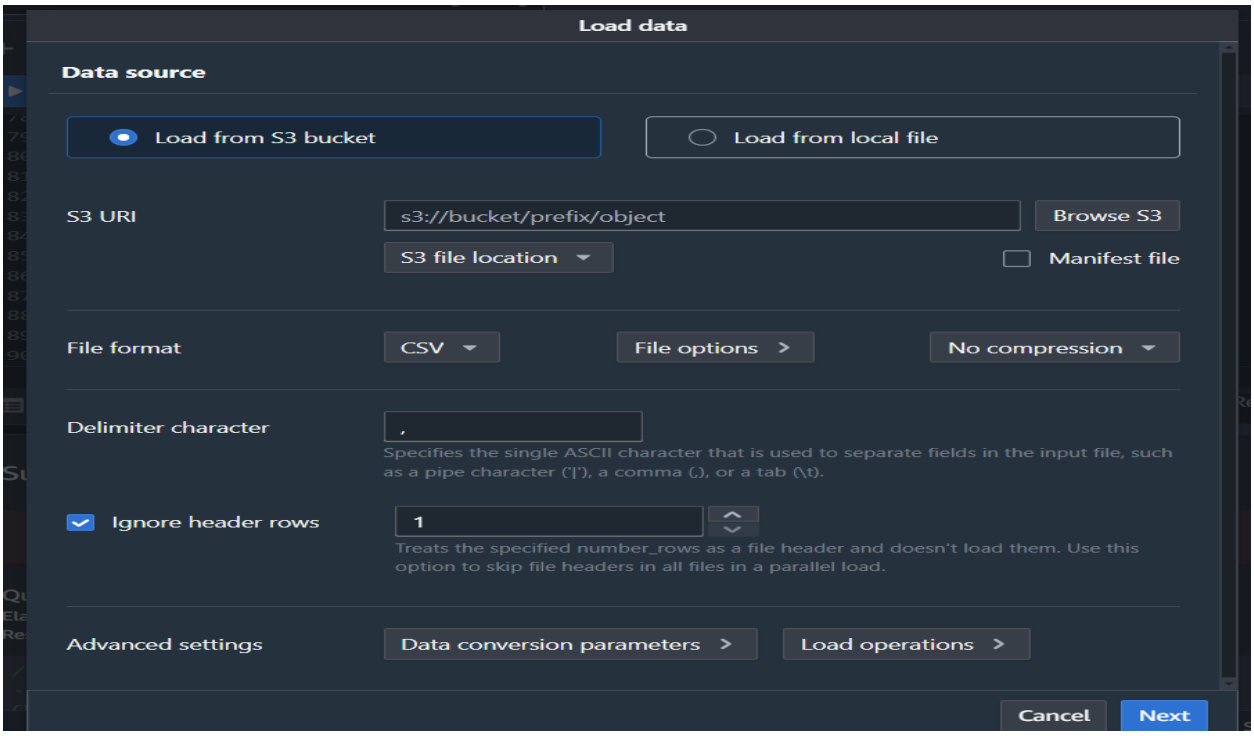
- **Cluster Creation**:
  - A Redshift cluster was created using the AWS Management Console**.**
- **Cluster Configuration**:
  - The cluster settings (e.g., node type, VPC, and security groups) were configured for optimal performance.
  - Appropriate scaling options were enabled to handle future data growth.

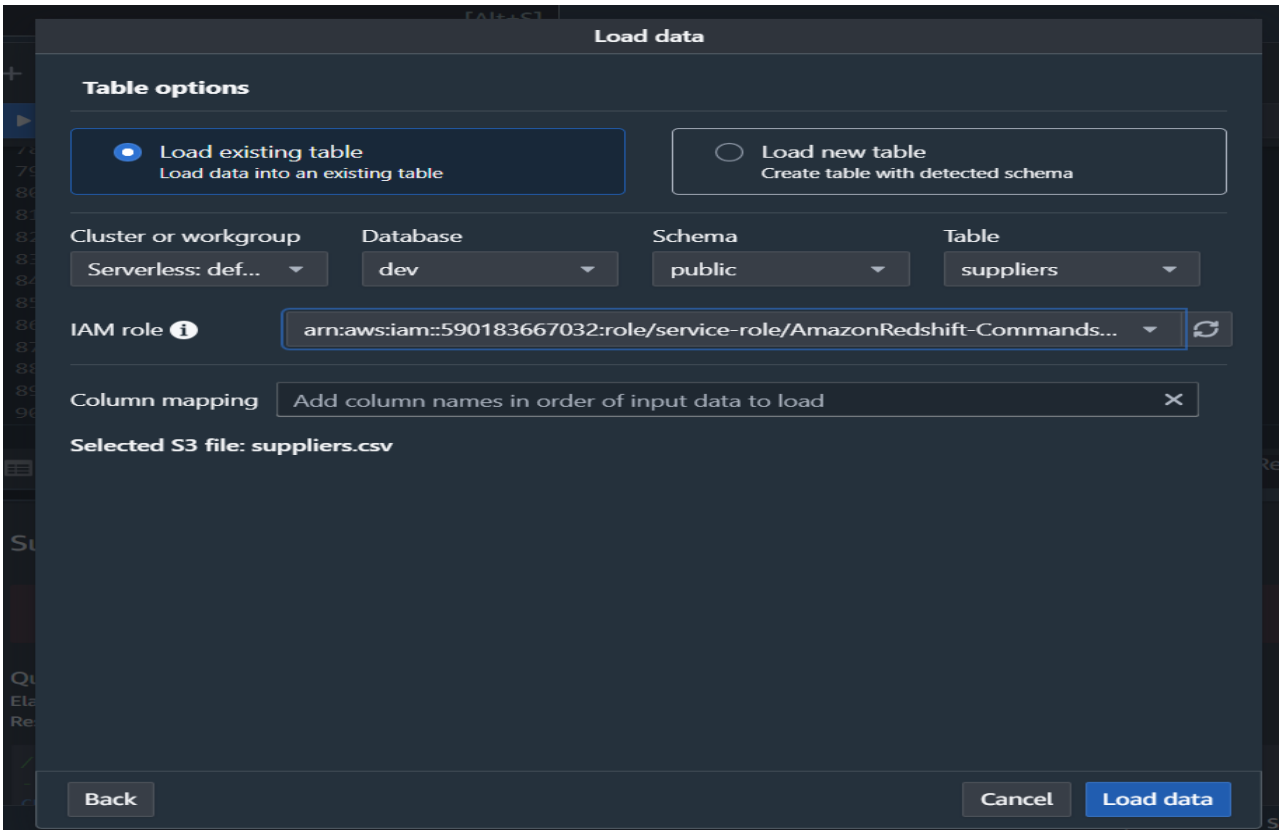
3.3 Creating Tables and Loading Data

- **Connection with SQL Workbench/J**:
  - SQL Workbench/J was used to connect to the Redshift cluster for database management and querying.
- **Table Creation**:
  - SQL scripts were executed to create the Fact Table and Dimension Tables based on the star schema design.
- **Data Loading**:
  - Data from Amazon S3 **in** CSV format was loaded into Redshift tables using SQL commands.
  - Additional data was inserted directly into the tables using the INSERT INTO statement for further testing and analysis.


ap-south-1.console.aws.amazon.com/s3/upload/adi-mybucket?region=ap-south-1&bucketType=general
Services
Search
[Alt+S]
Mumbai
adi_ghatol
Upload succeeded
View details below.
Summary
Destination
s3://adi-mybucket
Succeeded
5 files, 4.1 KB (100.00%)
Failed
0 files, 0 B (0%)
Files and folders
Configuration
Files and folders (5 Total, 4.1 KB)
Find by name
1
Name
Folder
Type
Size
Status
Error
dates.csv
products.csv
stores.csv
suppliers.csv
customers.cs..
text/csv
1.0 KB
1.1 KB
167.0 B
213.0 B
1.6 KB
Succeeded
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Services
Search
[Alt+S]
Mumbai
adi_ghatol
NEW: Amazon Redshift now supports zero-ETL integration with Amazon Aurora MySQL. Learn how you can get started applying near-real time analytics and machine learning on your transactional data today.
Learn more about Zero-ETL integrations
Amazon Redshift Serverless
Serverless dashboard Info
Query data
Create workgroup
Namespace overview Info
Filter namespace
All namespaces
Namespace data from your account
Total snapshots
0
Datashares in my account
0
Datashares requiring authorization
0
Datashares from other accounts
0
Datashares requiring association
0
Namespaces / Workgroups Info
Namespace
Status
Workgroup
Status
default-namespace
Available
default-workgroup
Available
Total compute usage - new Info
Choose a workgroup
Last hour
To visualize the costs of your total compute usage, go to AWS Cost Explorer
Total consumed RPU hours
--
Queries metrics
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

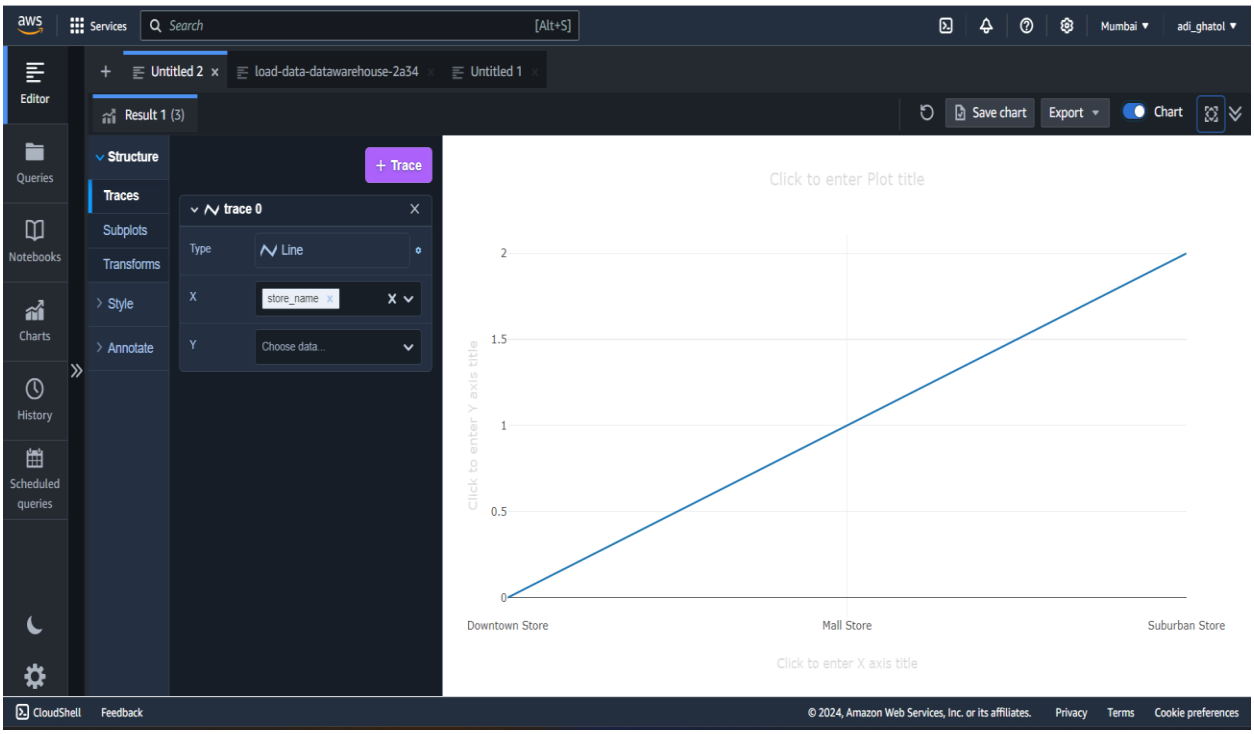

Load data
Data source
Load from S3 bucket
Load from local file
S3 URI
s3://bucket/prefix/object
Browse S3
S3 file location
Manifest file
File format
CSV
File options
No compression
Delimiter character
,
Specifies the single ASCII character that is used to separate fields in the input file, such as a pipe character ('|'), a comma (,), or a tab (\t).
Ignore header rows
1
Treats the specified number_rows as a file header and doesn't load them. Use this option to skip file headers in all files in a parallel load.
Advanced settings
Data conversion parameters
Load operations
Cancel
Next


Load data
Table options
Load existing table
Load data into an existing table
Load new table
Create table with detected schema
Cluster or workgroup
Serverless: def...
Database
dev
Schema
public
Table
suppliers
IAM role
arn:aws:iam::590183667032:role/service-role/AmazonRedshift-Commands...
Column mapping
Add column names in order of input data to load
Selected S3 file: suppliers.csv
Back
Cancel
Load data

Query performed :-

```sql
SELECT s.store_name, SUM(st.total_amount) AS total_sales
FROM sales_transactions st
JOIN stores s ON st.store_id = s.store_id
GROUP BY s.store_name
ORDER BY total_sales DESC;
```

Output :



5. Result

- **Sales by Store**: By querying the sales transactions table, real-time insights can be generated to show which stores are currently achieving the highest sales. Aggregating sales by store helps identify the top-performing locations.
- **Product Performance**: Analyzing sales trends for different products provides insights into which products are in high demand. This helps businesses focus on their best-selling items and plan promotions.
- **Customer Insights**: Queries analyzing customer purchase history reveal customer preferences, frequent buyers, and loyalty patterns, enabling businesses to improve customer retention strategies.
- **Seasonal Trends**: By analyzing sales across different time periods (e.g., holidays or seasons), businesses can identify seasonal trends and adjust inventory or marketing strategies accordingly.
- **Revenue Breakdown**: Aggregating sales by product category or store provides a clearer view of revenue streams, helping to identify the most profitable products and regions.

These insights allow for better decision-making, helping the business optimize operations, improve customer satisfaction, and boost profitability.

6. Conclusion

The star schema design implemented in Amazon Redshift provided an optimized structure for fast querying and analytics on sales data. The use of a Redshift cluster ensured scalability and performance, even with large datasets. Leveraging tools like SQL Workbench for data management and Tableau or Amazon QuickSight for visualization allowed for seamless interaction with the data.