GROUP – 6 PROJECT REPORT

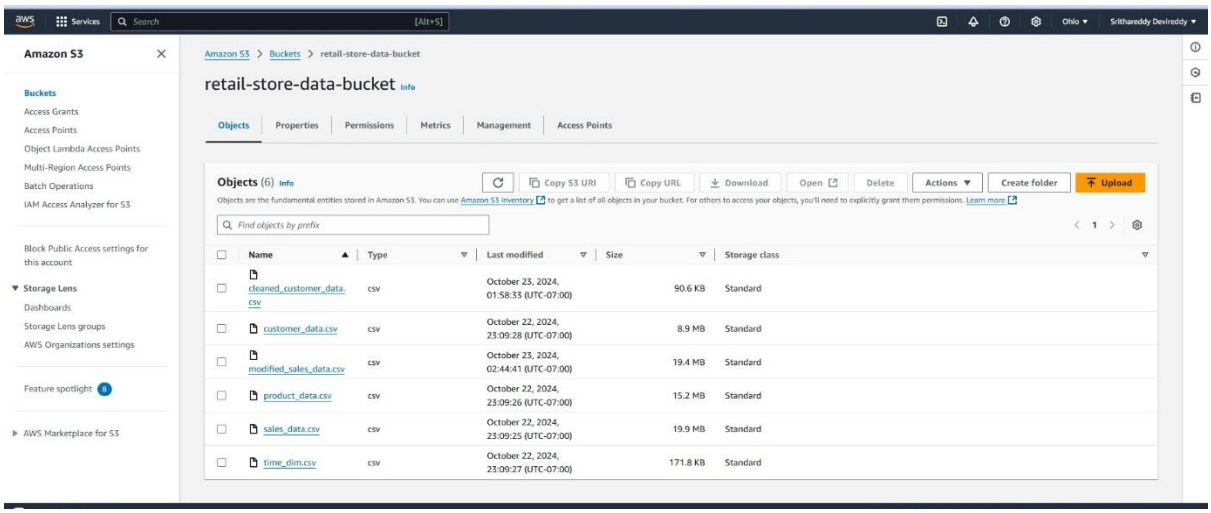
Optimizing Retail Operations through Data Warehousing

Anshu Reddy Dhamana - 018172141
Brendan Chao - 014142726
Sri Gopi Sarath Gode - 018191537
Srithareddy Devireddy - 018175456
Vinuthna Papana – 018176106

ETL

Amazon S3 Bucket:
An S3 bucket in Amazon S3 is a storage container used to hold various types of data, including CSV files. These CSV files are raw data that typically undergo Extract, Transform, and Load (ETL) operations within AWS Glue or similar services.

During ETL, the CSV files are extracted from the S3 bucket, transformed into a structured format suitable for analysis or other purposes, and then loaded back into another location, often another S3 bucket or a database, ready for use. This process helps in organizing, cleaning, and preparing the data for analytics, reporting, or further processing.



IAM Roles:
In Amazon Web Services (AWS), Identity and Access Management (IAM) roles serve as fundamental tools for securely delegating permissions within AWS resources. These roles define a set of permissions that dictate actions authorized for entities, be it an AWS service or a user, on AWS resources. They eliminate the necessity for long-term credentials like usernames and passwords when granting access to resources.

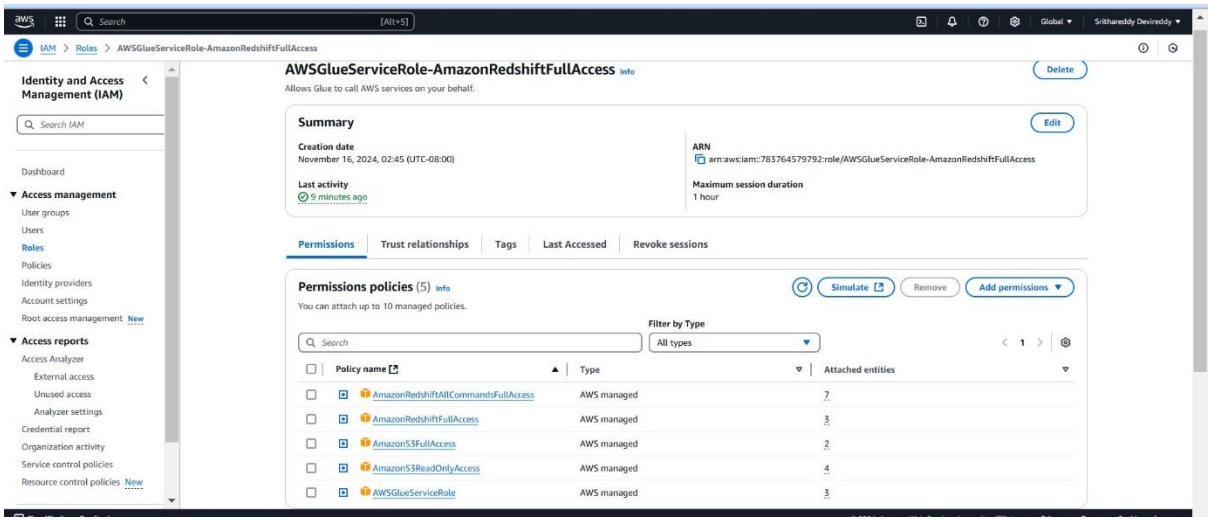
The IAM role in the image below is configured with specific permissions:-

AmazonS3FullAccess: Grants complete access to Amazon S3 (Simple Storage Service), enabling any action on S3 buckets within the AWS account.
AmazonRedshiftFullAccess: Provides full access to Amazon Redshift, empowering the IAM role to manage Redshift clusters comprehensively. This includes tasks such as creating, modifying, deleting clusters, and handling administrative functions.

AWSGlueServiceRole: Grants the essential permissions for executing actions within AWS Glue, a fully managed ETL service facilitating data preparation and loading for analytics.

By consolidating these permissions within a single IAM role, entities assuming this role gain the combined access and capabilities across Amazon S3, Amazon Redshift, and AWS Glue. This practice aligns with security best practices, adhering to the principle of least privilege by granting only the necessary permissions for specific tasks.

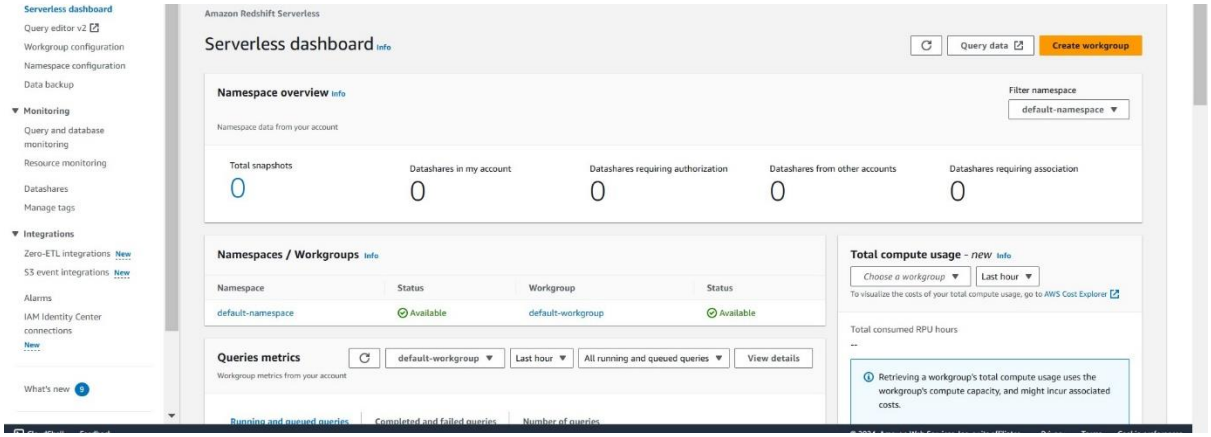
IAM roles offer remarkable flexibility, allowing different entities such as AWS services, applications, or other AWS accounts to assume these roles based on predefined trust policies. Assigning roles to entities enables centralized permission management, upholding security measures, and reducing reliance on persistent credentials, thereby enhancing the overall security posture of AWS resources.



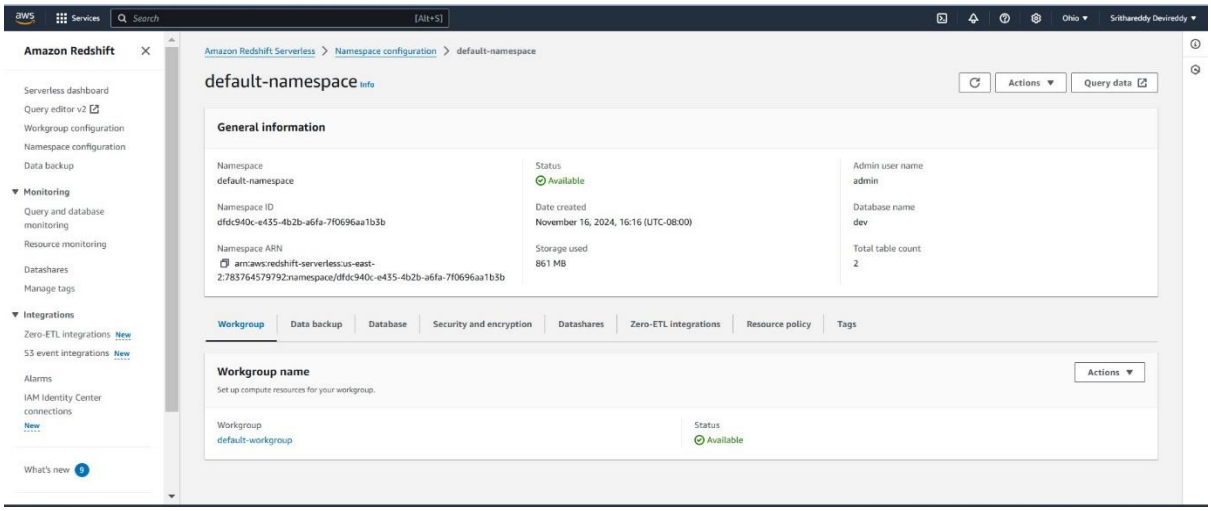
Red Shift Cluster:

Amazon Redshift is a data warehouse service in AWS used for analyzing large datasets. A Redshift cluster is a collection of nodes that work together to handle queries and data storage. After the ETL process, the transformed data from CSV files is often loaded into a Redshift cluster, typically organized into tables.

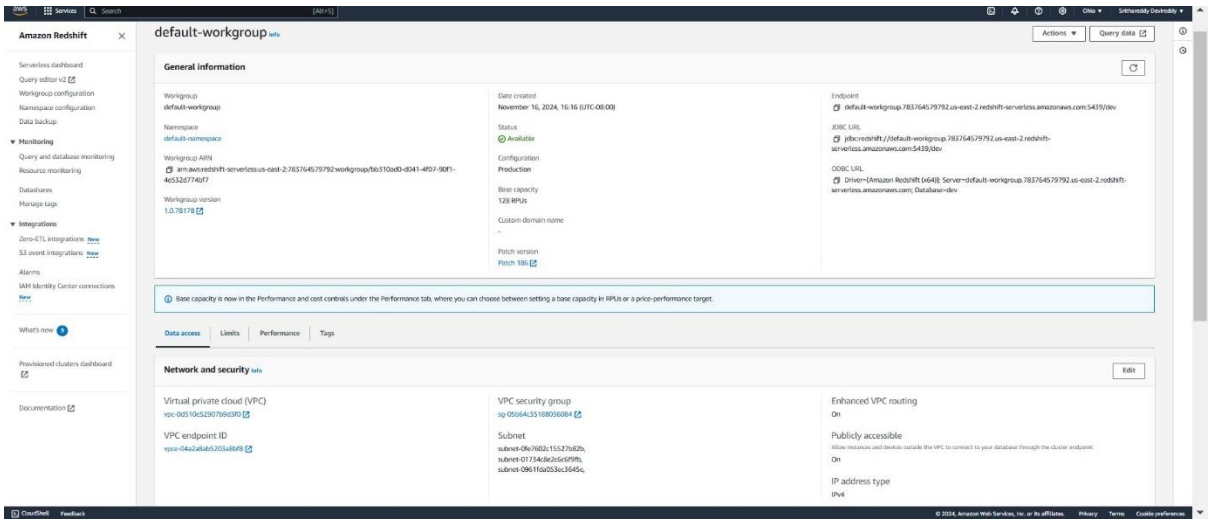
These tables store structured data and are designed to support efficient querying and analysis. In your case, there are four tables within the Redshift cluster, each containing specific datasets that have undergone transformation. These tables serve as organized repositories of data, allowing users to run complex queries and analytics to derive insights.



Details of Redshift cluster:



Connection details of Redshift Cluster:



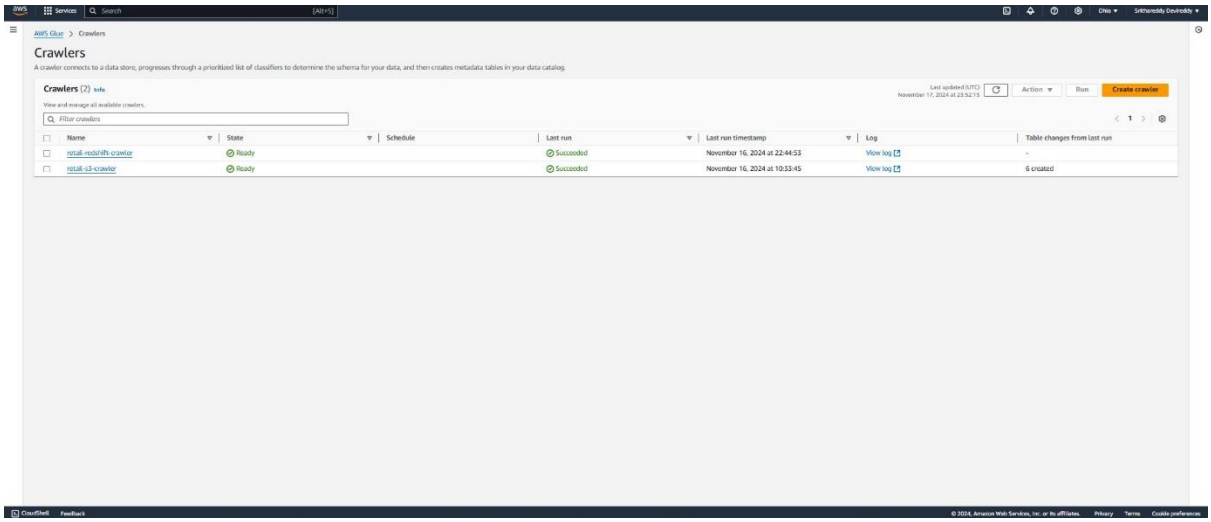
Crawlers:

In AWS Glue, crawlers serve as automated processes used for discovering and cataloging metadata from diverse data sources. These processes analyze data structure and schema, facilitating streamlined data processing within the ETL (Extract, Transform, Load) workflow.

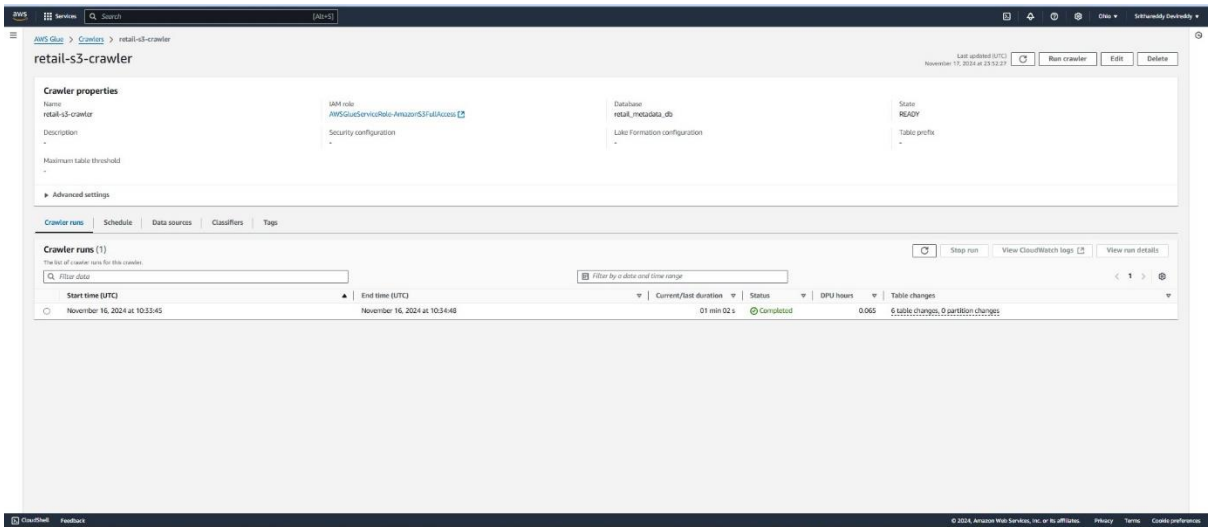
The S3 crawler within AWS Glue inspects data stored in Amazon S3 buckets. It identifies file formats and schema, organizing the data for subsequent use in the ETL process. By scanning S3 data, this crawler comprehends its structure, aiding in the preparation of data for transformation.

Similarly, the Redshift crawler in AWS Glue targets Redshift clusters. It assesses the data within Redshift tables, capturing their structure and schema. This acquired information becomes crucial for mapping and efficiently transforming data during ETL operations. It ensures consistency and accuracy in data processing within the Redshift environment.

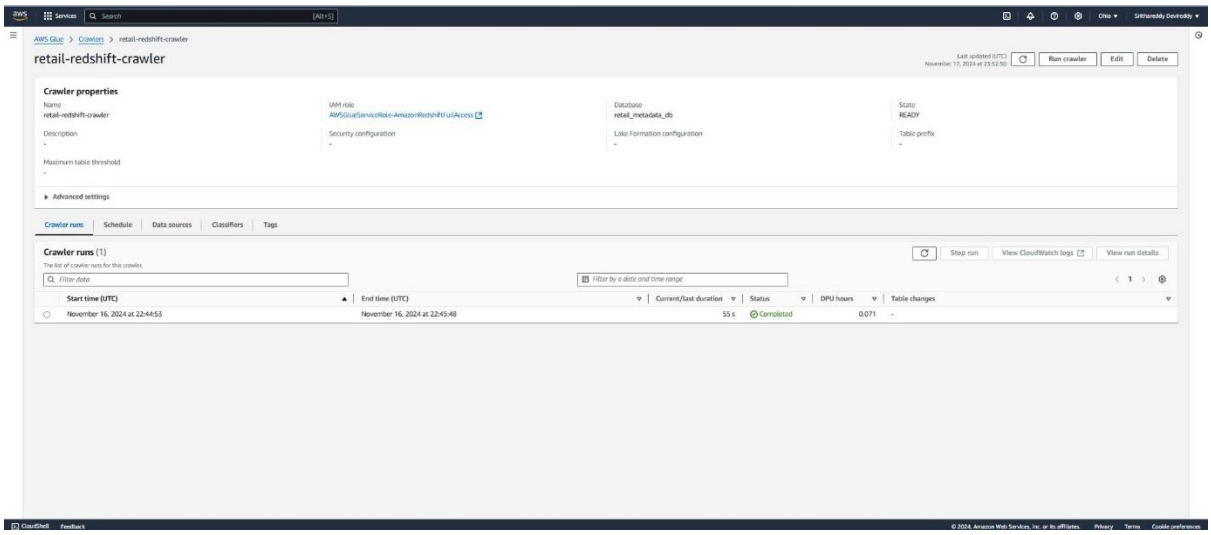
Both crawlers assume a pivotal role in the ETL process by automatically discovering and organizing metadata. Their function streamlines the transformation and loading of data from its raw state into structured datasets, primed for analysis or other intended uses.



Retail S3 Crawler:

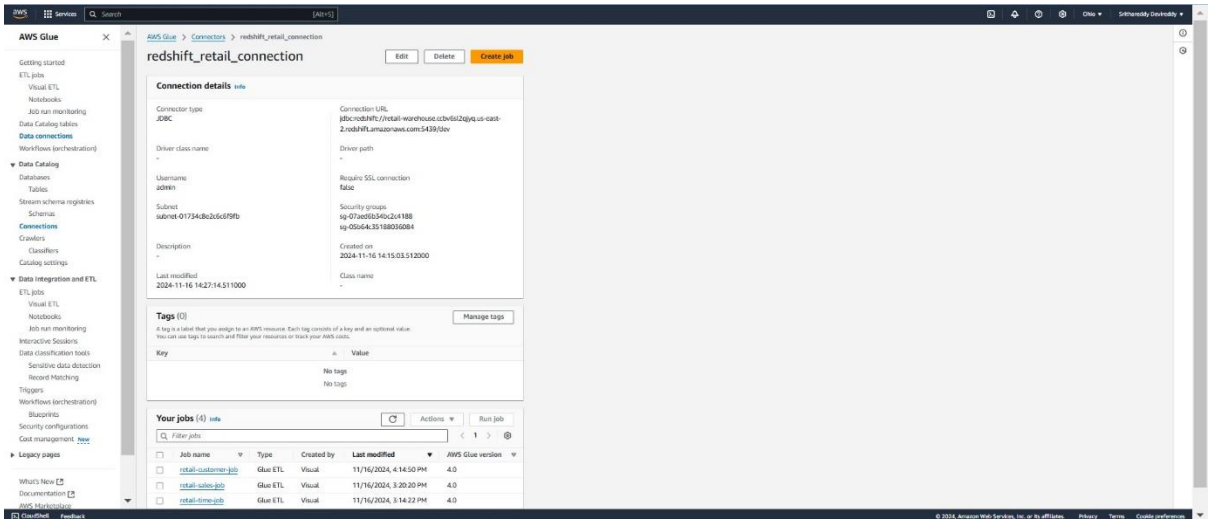


Redshift Crawler:

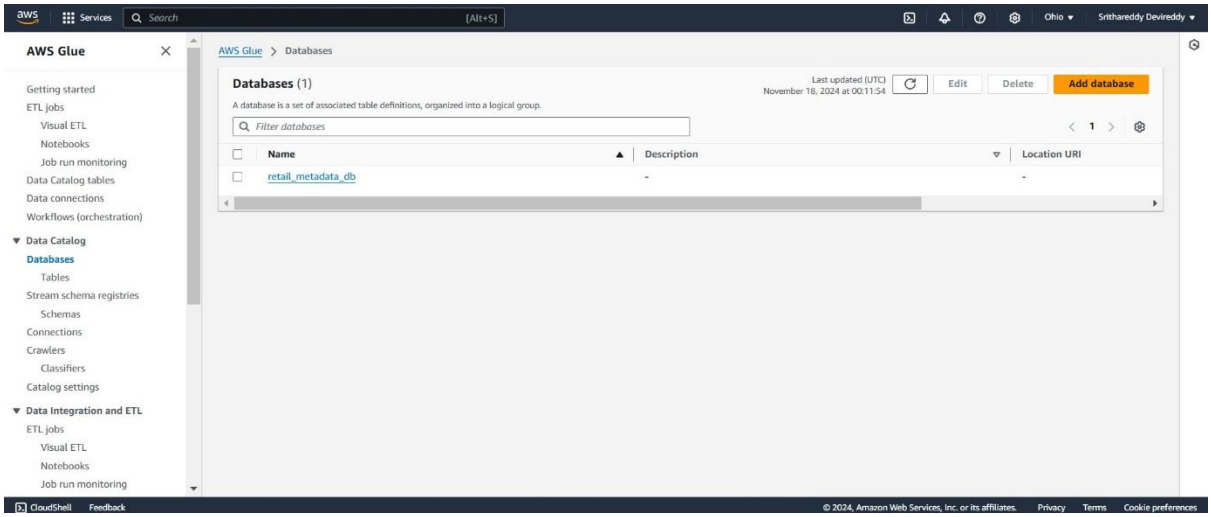


JDBC Connection to Temporary Database:

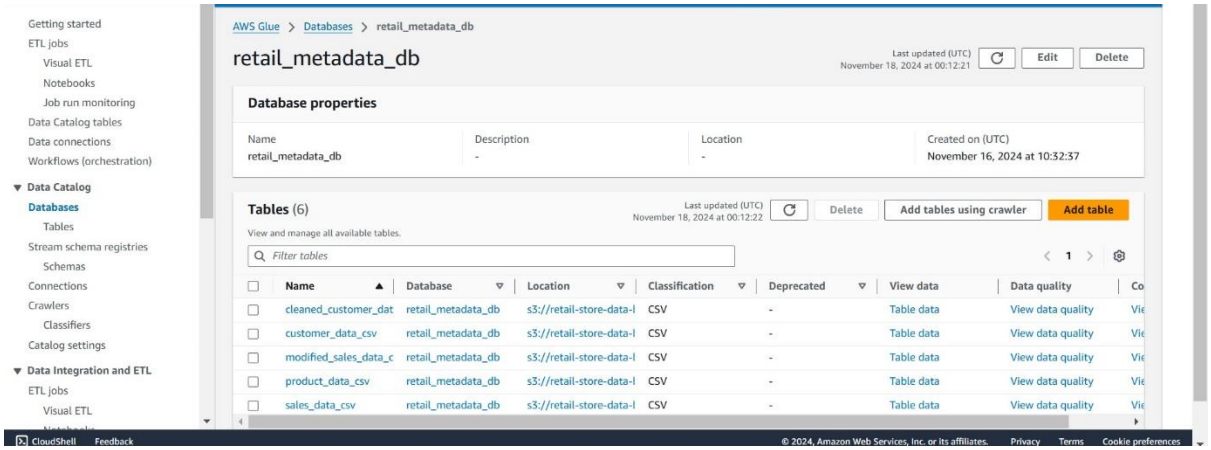
In AWS Glue, a JDBC connection establishes access to a temporary database, essential for the ETL process. JDBC, as a Java Database Connectivity tool, enables data interaction and transfer between the Glue environment and external databases. This connectivity supports AWS Glue in extracting, transforming, and loading data across diverse databases, ensuring smooth data processing throughout the ETL operations.



Temporary Database:



Contents of Temporary Database:



ETL Jobs:

In AWS Glue, an ETL job is created for each of the four tables to manage data flow within the ETL process.

These jobs define Extract, Transform, and Load operations tailored for each table. Here's the breakdown:

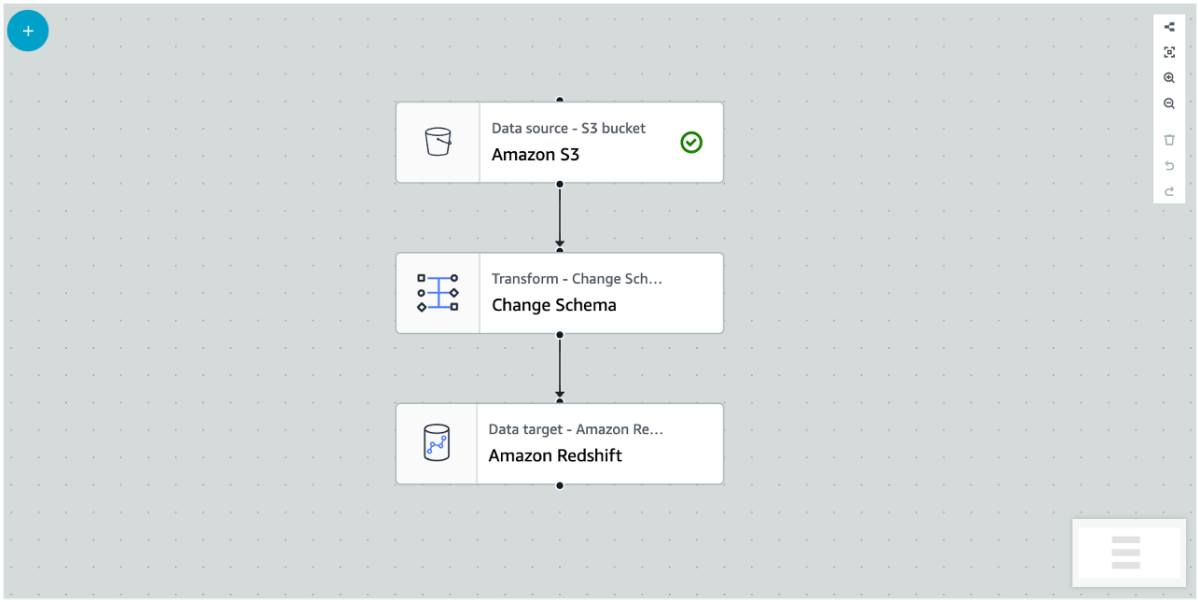
1. Extraction (Extract): Data is extracted from respective sources like Amazon S3 buckets or Redshift databases.

2. Transformation (Transform): Extracted data undergoes predefined logic-based transformations, such as cleaning, restructuring, aggregating, or enriching for analysis.

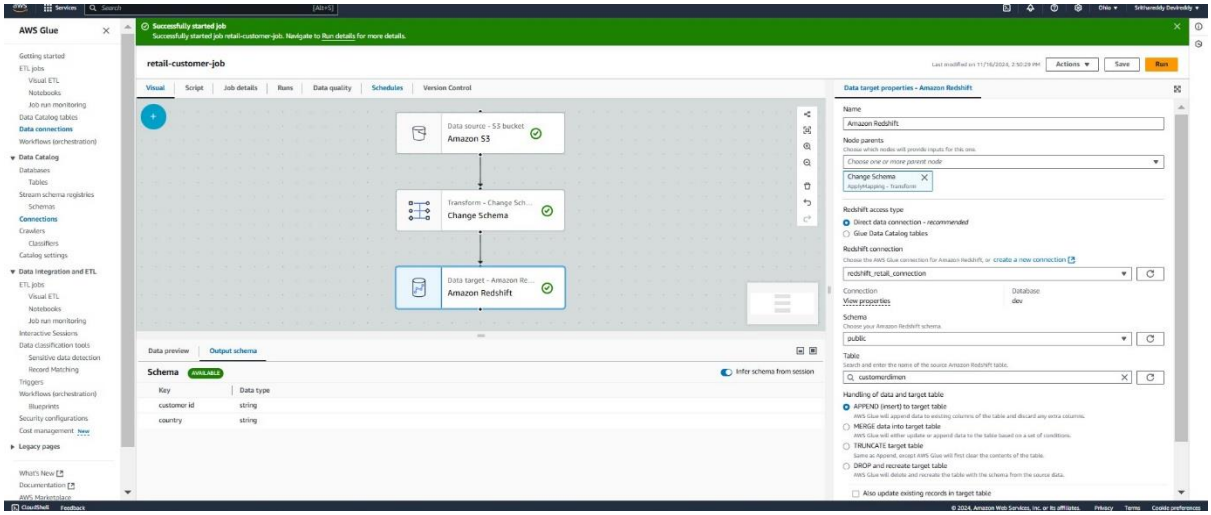
3. Loading (Load): Transformed data is loaded into the target destination, typically the respective Redshift table. This step ensures structured data storage, ready for efficient querying and analysis.

Each ETL job in AWS Glue addresses unique requirements and transformations specific to its table. Creating separate jobs for each table enables better organization, management, and optimization of the ETL process, aligning with the data characteristics within each table.

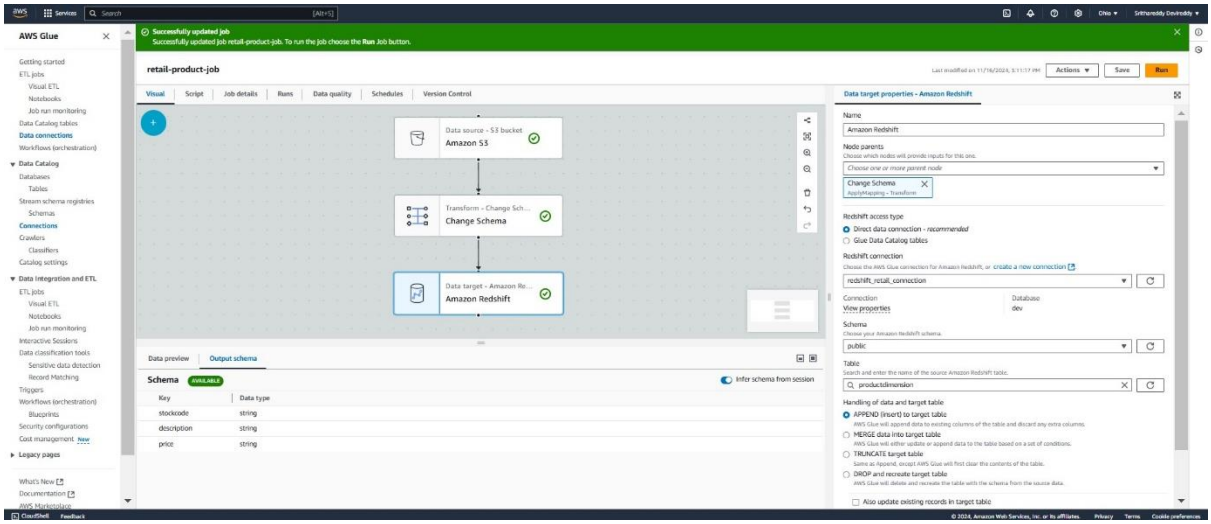
ETL Workflow:



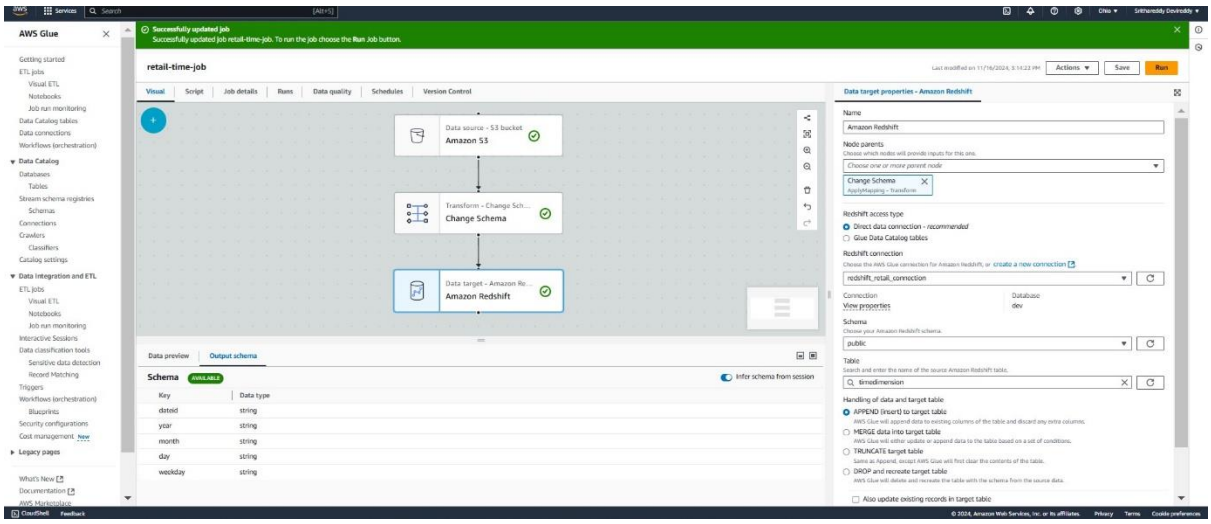
Customer ETL:



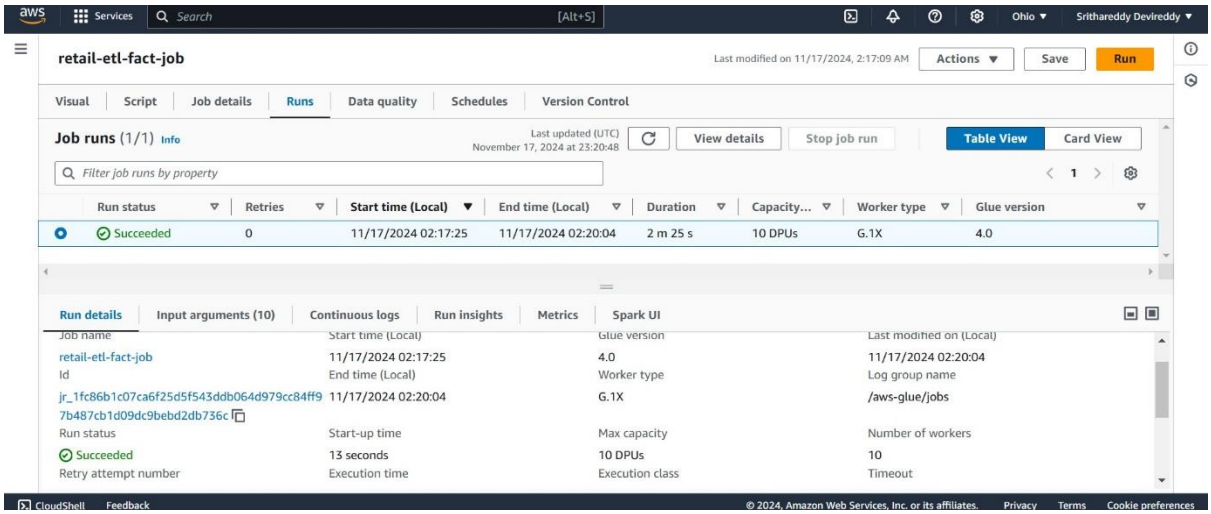
Product ETL:



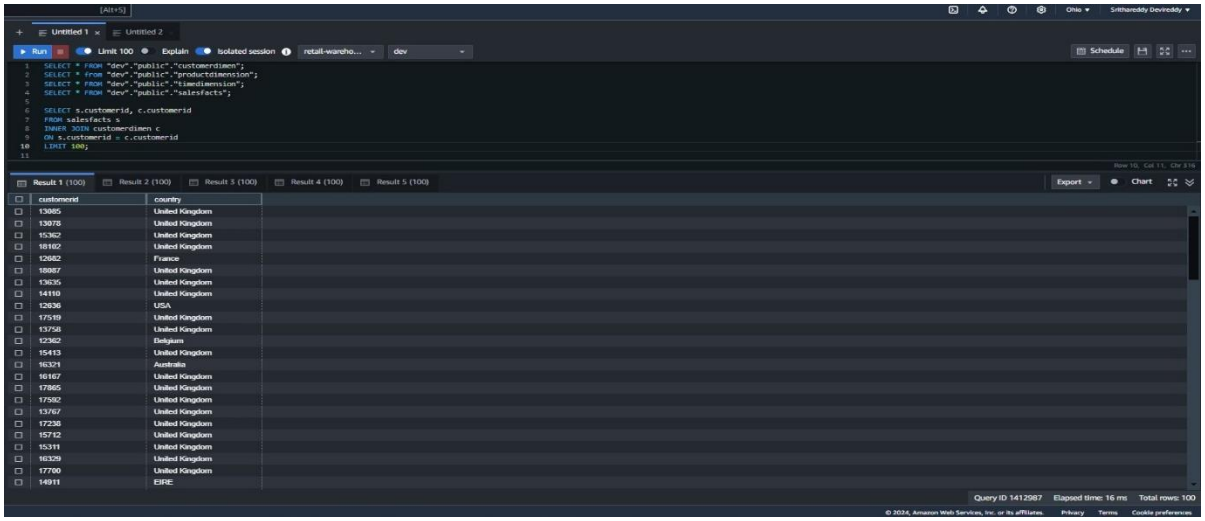
Time ETL:



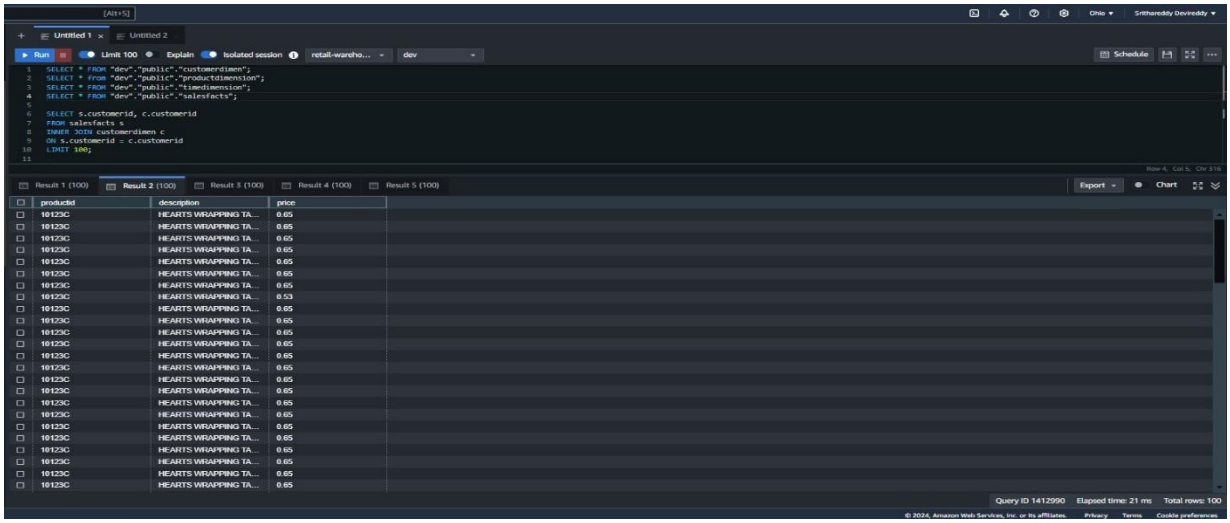
Execution of ETL Job:



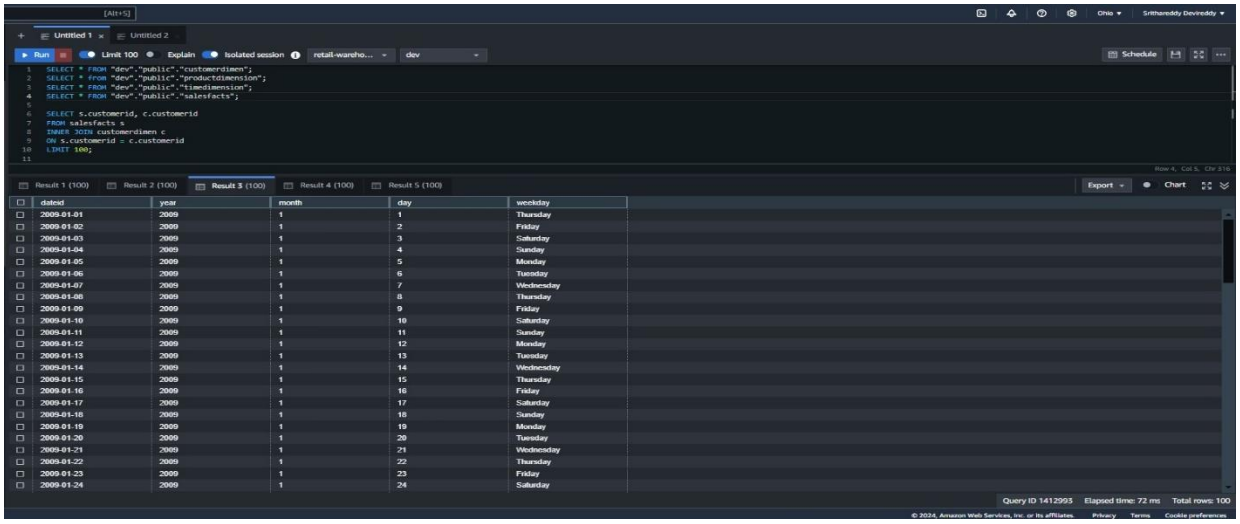
Customer Data Loaded into Redshift Cluster:



Product Data Loaded into Redshift Cluster:



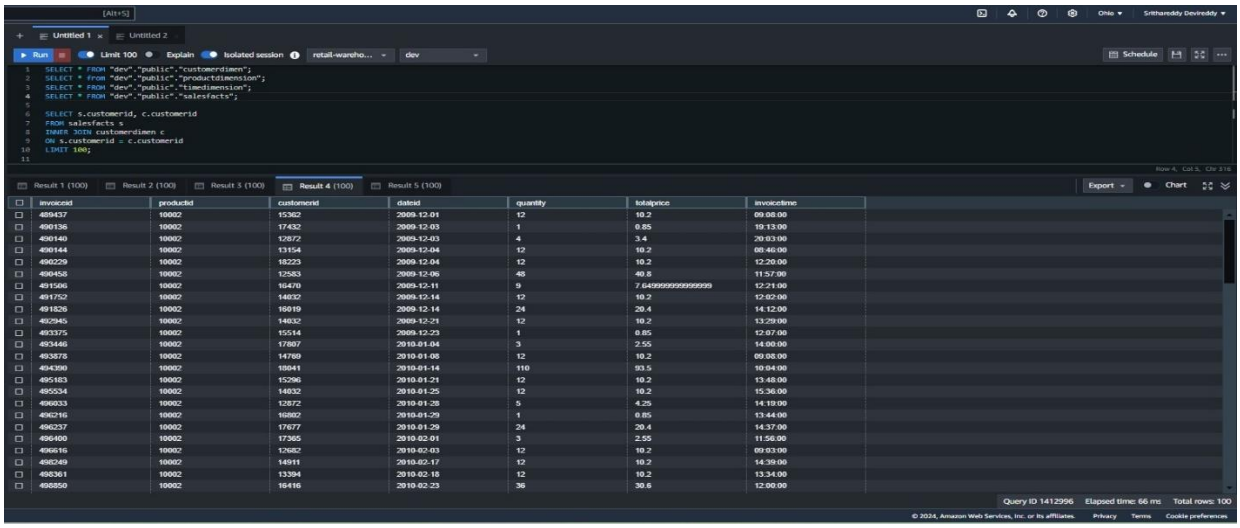
Time Data Loaded into Redshift Cluster:



```
SELECT * FROM "dev"."public"."customerdimen";
SELECT * FROM "dev"."public"."productdimension";
SELECT * FROM "dev"."public"."timedimension";
SELECT * FROM "dev"."public"."salesfacts";

SELECT s.customerid, c.customerid
FROM salesfacts s
INNER JOIN customerdimen c
ON s.customerid = c.customerid
LIMIT 100;
```

dateid	year	month	day	weekday
2009-01-01	2009	1	1	Thursday
2009-01-02	2009	1	2	Friday
2009-01-03	2009	1	3	Saturday
2009-01-04	2009	1	4	Sunday
2009-01-05	2009	1	5	Monday
2009-01-06	2009	1	6	Tuesday
2009-01-07	2009	1	7	Wednesday
2009-01-08	2009	1	8	Thursday
2009-01-09	2009	1	9	Friday
2009-01-10	2009	1	10	Saturday
2009-01-11	2009	1	11	Sunday
2009-01-12	2009	1	12	Monday
2009-01-13	2009	1	13	Tuesday
2009-01-14	2009	1	14	Wednesday
2009-01-15	2009	1	15	Thursday
2009-01-16	2009	1	16	Friday
2009-01-17	2009	1	17	Saturday
2009-01-18	2009	1	18	Sunday
2009-01-19	2009	1	19	Monday
2009-01-20	2009	1	20	Tuesday
2009-01-21	2009	1	21	Wednesday
2009-01-22	2009	1	22	Thursday
2009-01-23	2009	1	23	Friday
2009-01-24	2009	1	24	Saturday

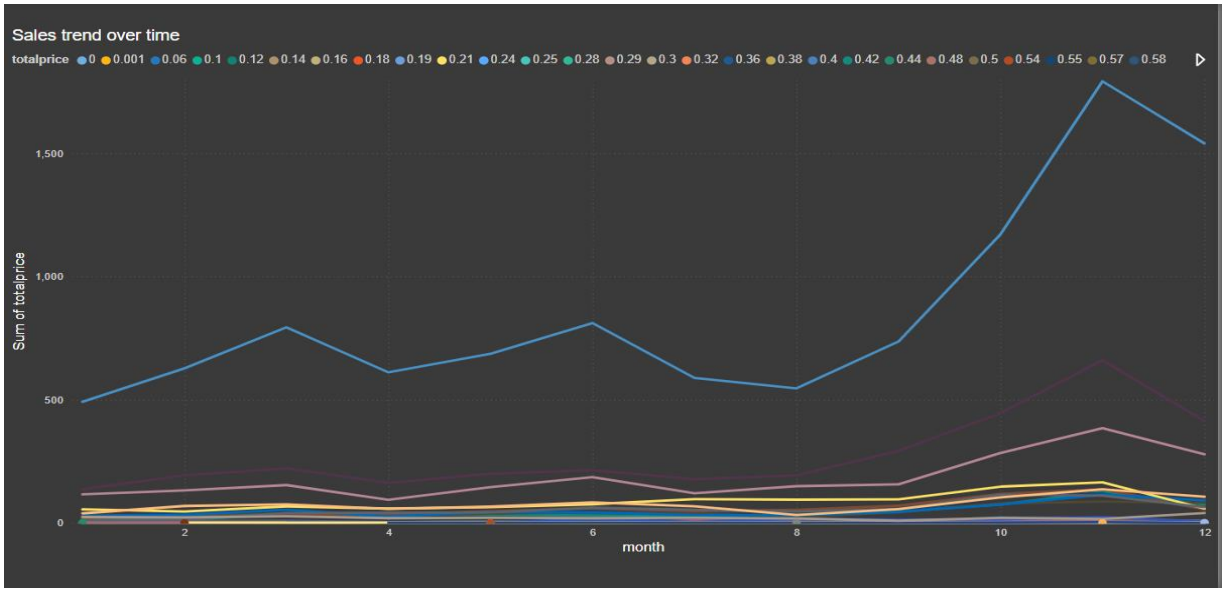
Sales Data Loaded into Redshift Cluster:

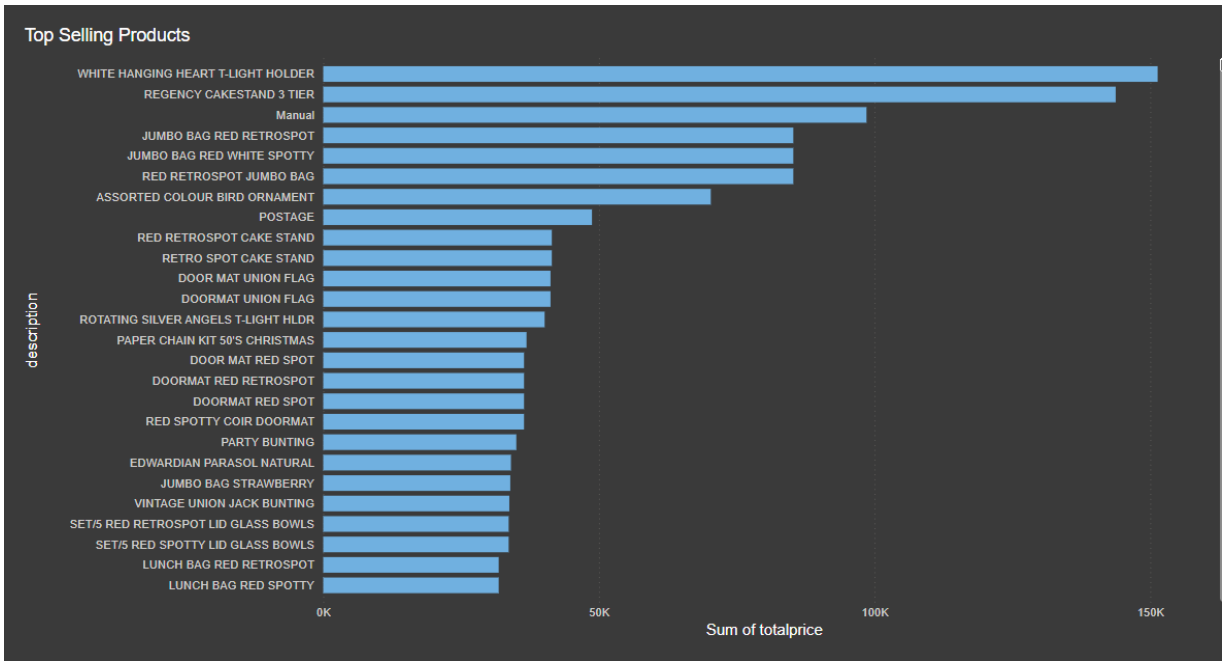


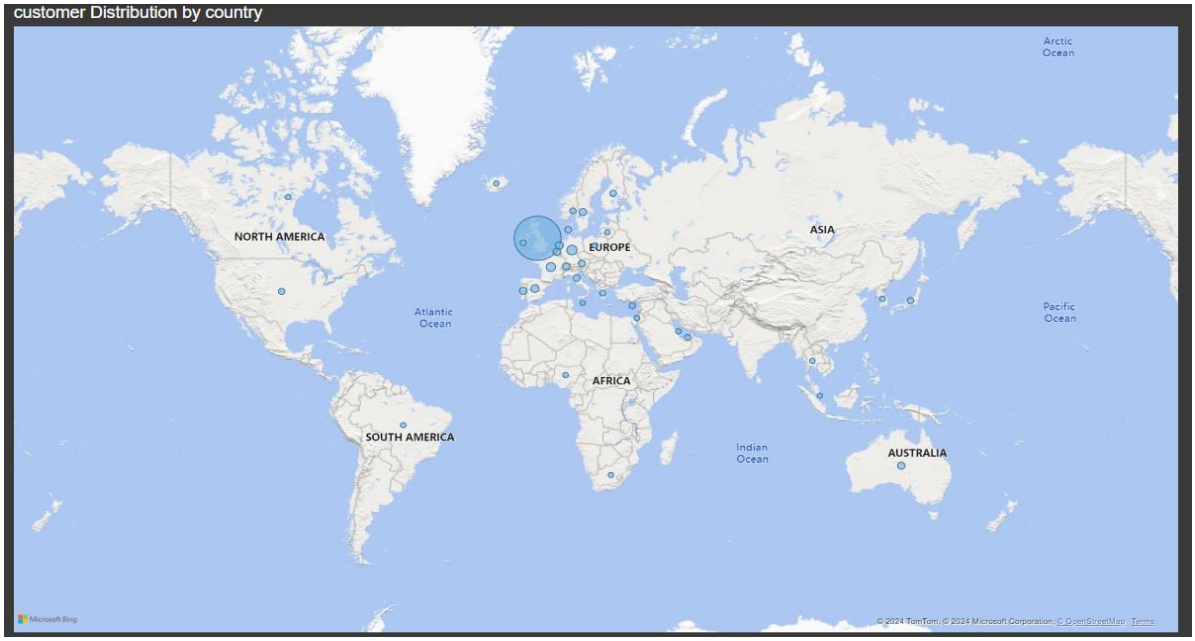
```
SELECT * FROM "dev"."public"."customerdimen";
SELECT * FROM "dev"."public"."productdimension";
SELECT * FROM "dev"."public"."timedimension";
SELECT * FROM "dev"."public"."salesfacts";

SELECT s.customerid, c.customerid
FROM salesfacts s
INNER JOIN customerdimen c
ON s.customerid = c.customerid
LIMIT 100;
```

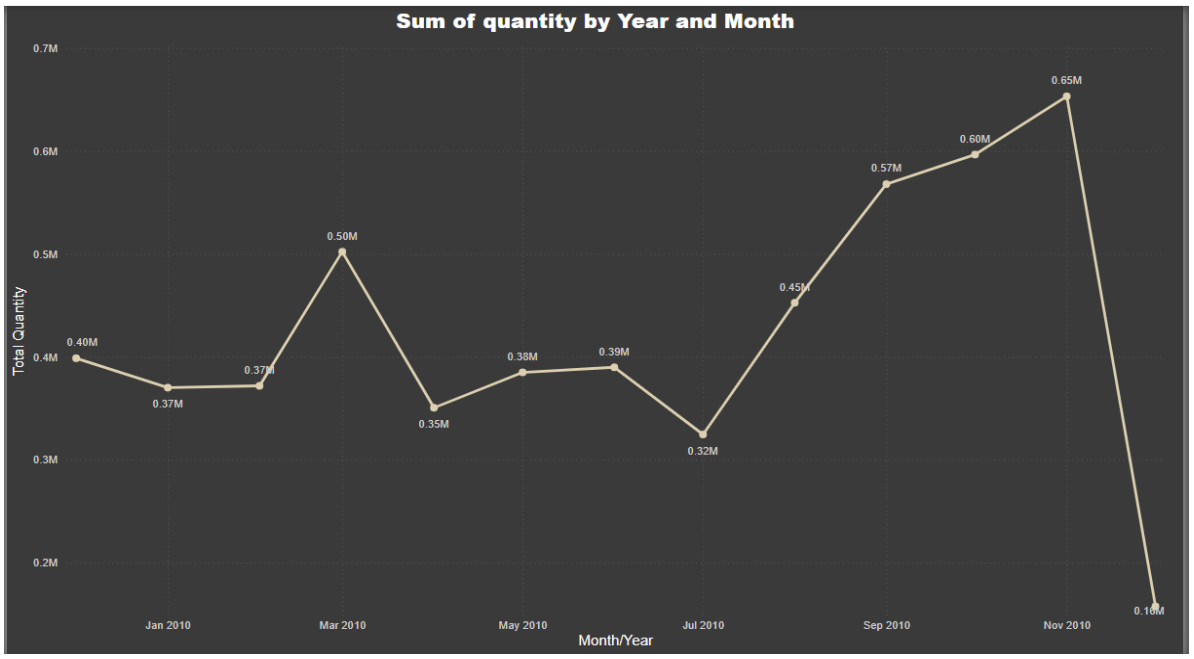
invoiceid	productid	customerid	dateid	quantity	totalprice	invoicetime
489437	10002	15362	2009-12-01	12	10.2	09:08:00
490136	10002	17432	2009-12-03	1	0.85	19:13:00
490140	10002	12872	2009-12-03	4	3.4	20:53:00
490144	10002	13154	2009-12-04	12	10.2	08:46:00
490229	10002	18223	2009-12-04	12	10.2	12:20:00
490458	10002	12583	2009-12-06	48	40.8	11:57:00
491506	10002	16470	2009-12-11	9	7.649999999999999	12:21:00
491752	10002	14832	2009-12-14	12	10.2	12:02:00
491826	10002	16819	2009-12-14	24	20.4	14:12:00
492945	10002	14832	2009-12-21	12	10.2	13:29:00
493375	10002	15514	2009-12-23	1	0.85	12:07:00
493446	10002	17807	2010-01-04	3	2.55	14:00:00
493878	10002	14769	2010-01-08	12	10.2	09:08:00
494398	10002	18941	2010-01-14	110	93.5	10:04:00
495183	10002	15296	2010-01-21	12	10.2	13:48:00
495534	10002	14832	2010-01-25	12	10.2	15:36:00
496033	10002	12872	2010-01-28	5	4.25	14:19:00
496216	10002	16802	2010-01-29	1	0.85	13:44:00
496237	10002	17677	2010-01-29	24	20.4	14:37:00
496400	10002	17365	2010-02-01	3	2.55	11:56:00
496616	10002	12682	2010-02-03	12	10.2	09:03:00
496249	10002	14911	2010-02-17	12	10.2	14:39:00
498361	10002	13394	2010-02-18	12	10.2	13:34:00
498850	10002	16416	2010-02-23	36	30.6	12:00:00

Visualisations:






customer Distribution by country
Arctic Ocean
NORTH AMERICA
EUROPE
ASIA
Atlantic Ocean
Pacific Ocean
AFRICA
SOUTH AMERICA
Indian Ocean
AUSTRALIA
Microsoft Bing
© 2024 TomTom, © 2024 Microsoft Corporation, © OpenStreetMap Terms


Sum of quantity by Year and Month
0.7M
0.6M
0.5M
0.4M
0.3M
0.2M
Total Quantity
0.40M
0.37M
0.37M
0.50M
0.35M
0.38M
0.39M
0.32M
0.45M
0.57M
0.60M
0.65M
0.10M
Jan 2010
Mar 2010
May 2010
Jul 2010
Sep 2010
Nov 2010
Month/Year

Significance to the Real World:

The ability to use data efficiently has become very essential for success in the fast-paced retail industry, where competition is very intense and client expectations are always changing. By helping companies transform unstructured data into insightful knowledge it leads to better decisions and results, this project offers a concrete answer to real-world problems.

1.Empowering Smarter Decisions:
Every day, retailers produce vast volumes of data from client interactions and sales transactions. However, because it is unstructured, this data is frequently underutilized. Our project helps organizations understand what their customers want, when they want it, and how to give it most effectively by cleaning, organizing, and analyzing this data. These insights result in quicker and more intelligent decision-making, whether it's spotting development prospects or figuring out which goods are the most popular.

2.Building for Growth:
Businesses' data needs grow along with them. Our solution uses cloud-based technologies like Amazon Redshift to manage growing data volumes without slowing down operations, so it can expand with the company. This scalability guarantees that, regardless of the size of the company, the system will always be economical and effective.

3.Personalizing Customer Experiences:
Today's consumers demand individualized service. Businesses can create specialized marketing efforts and modify product offers to suit customer needs by having comprehensive information about purchasing trends and preferences. This increases client satisfaction and loyalty in addition to increasing sales.

4.Uncovering Trends and Patterns:
Staying ahead of the competition requires knowing when and why customers buy. Our solution enables companies to identify peak sales times and seasonal trends by incorporating time-based variables. This helps them maximize stock levels, prevent missed opportunities, and better plan for periods of strong demand.

5.Making Operations More Efficient:
Inefficiencies and lost opportunities are frequently the result of disorganized data. Our technology provides a clear picture of important parameters including inventory levels, product performance, and sales trends, which helps to improve retail operations. Businesses can efficiently allocate the resources, cut waste, and increase revenues thanks to this transparency.

6.Staying Competitive in a Data-Driven World:
Data is becoming a possible competitive advantage, and this project provides retailers with the tools they need to stay ahead. Converting unusable data into actionable insights can help businesses remain ahead of the competition and establish market trends.

LESSONS LEARNED

We ran into a number of obstacles during this endeavour, but we also learned a lot that helped us better grasp data warehousing and analytics. In addition to helping the project succeed, these lessons provided valuable insights that might be used to future projects of a similar nature.

1. The Importance of Data Quality
Cleaning and organizing the raw data was one of the project's most time-consuming tasks. We found that ensuring the accuracy, completeness, and consistency of the data is essential to the success of any analytics effort. High-quality data reduces errors and increases the reliability of insights.

2. ETL Complexity and Best Practices
Designing an efficient ETL pipeline was a technically challenging task. We discovered that breaking the pipeline into modular steps (Extract, Transform, Load) makes it easier to manage and debug. Automation and documentation of these processes proved essential for scalability and reproducibility.

3. Effective Data Modelling
The star schema design allowed us to organize data effectively for querying and analysis. However, we learned that tailoring the schema to specific business questions (e.g., sales trends, customer behaviour) significantly enhances the usability and relevance of the data warehouse.

4. Cloud Scalability and Cost Management
Using Amazon Redshift highlighted the trade-offs between performance and cost in cloud infrastructure. We learned how to optimize resource usage, such as through query optimization and compression techniques, to balance speed and cost efficiency.

5. Visualization as a Communication Tool
Creating dashboards in Tableau or Quick Sight was a key step in making insights accessible and actionable. We learned that simple, intuitive visualizations can convey complex data findings effectively, ensuring they drive decision-making.

6. Team Collaboration and Role Clarity
Collaboration among team members with clearly defined roles was crucial for meeting project milestones. Effective communication and alignment on shared goals ensured the project stayed on track, despite technical and logistical hurdles.

7. Anticipating Real-World Challenges
Simulating real-world business scenarios, such as handling incomplete or inconsistent datasets, provided invaluable experience. This prepared us to adapt and problem-solve when dealing with unpredictable data environments.

Uniqueness of Lessons
The lessons we learned are substantial because they go beyond theoretical knowledge, offering practical insights into real-world applications of data warehousing and analytics. What sets them apart is their emphasis on integrating technical expertise with business relevance, ensuring that our solution is not just functional but impactful.

By including these lessons in the report and presentation, we aim to showcase our journey, the hurdles we overcame, and the knowledge we gained, ensuring that others can benefit from our experience. These lessons reflect not only the technical growth of the team but also the strategic thinking required to align analytics with business objectives.

INNOVATION

This project is notable for taking a novel approach to solving the problems with handling and evaluating massive amounts of data that the retail sector faces. We have developed a solution that is both scalable and flexible enough to accommodate changing business requirements by fusing cloud-based infrastructure with contemporary data warehousing methodologies. The main features of the innovation are listed below.

1. Scalable Cloud-Based Architecture

What's New: In contrast to conventional on-premises solutions, this project makes use of Amazon Redshift's cloud capabilities to effectively manage huge and expanding datasets. Businesses may extend their data operations without worrying about capacity constraints or expensive infrastructure expenditures thanks to the utilization of cloud infrastructure.
Impact: Without having to make large additional investments, businesses may easily adjust to growing data quantities and maintain performance.

2. Star Schema with Temporal Dimensions

What's New: The integration of a star schema with temporal dimensions provides a unique way to analyze sales data over time. By breaking down data into manageable and interconnected fact and dimension tables, we enabled more precise queries and granular insights.
Impact: This design allows businesses to identify seasonal trends, peak sales periods, and year-over-year performance with minimal query complexity.

3. Streamlined ETL Pipeline

What's New: The ETL pipeline was designed to not only clean and transform raw data but also optimize its structure for downstream analytics. Using Python and SQL, the pipeline automates labor-intensive tasks such as handling missing values, creating time-series data, and calculating key metrics.
Impact: This approach reduces manual intervention, increases efficiency, and ensures that the processed data is ready for real-time analysis.

4. Real-Time Insights with Advanced Dashboards

What's New: To provide real-time, actionable insights, the project combines Amazon Redshift with visualization tools such as Tableau or QuickSight. With an emphasis on KPIs like customer behavior, sales trends, and product performance, the dashboards were made to be user-friendly.
Impact: Retailers' responsiveness to market dynamics is enhanced by their ability to swiftly recognize opportunities, resolve obstacles, and make data-driven decisions.

5. Focus on Business and Technical Alignment

What's New: The project serves as a link between corporate goals and technological implementation. The solution guarantees relevance and usability by coordinating data

processing with certain business requirements (such as inventory management and customer targeting).
Impact: By concentrating on this area, merchants are better able to connect data insights to observable results like higher customer satisfaction and profitability.

6. Cost-Effective Data Management
What's New: The system was designed to optimize resource usage in a cloud environment, balancing performance and cost. Techniques like query optimization and efficient data storage ensure high performance without unnecessary expenses.
Impact: Businesses, especially small and medium-sized enterprises, can adopt advanced analytics without prohibitive costs.

7. Industry-Ready Framework
What's New: The project offers a repeatable framework that can be used in sectors other than retail, such as logistics, healthcare, and finance. The star schema and ETL design's adaptability allows it to be used with a variety of data types and use scenarios.
Impact: The project's value is increased and it is positioned as a paradigm for data-driven transformation due to its cross-industry applicability.

Why This is Unique
The innovation lies in how the project seamlessly integrates technical excellence with business practicality. By emphasizing scalability, real-time insights, and cost-effectiveness, this solution goes beyond standard analytics frameworks, offering a forward-thinking approach that addresses current and future retail challenges. These innovative features make the project a powerful example of how modern technology can drive meaningful change in an industry as dynamic as retail.

TEAMWORK

The success of this project was a direct result of our team's strong collaboration, clear role definition, and commitment to shared goals. Each member contributed uniquely, leveraging their expertise and collaborating effectively to ensure the project was completed on time and to a high standard. Below is an overview of how teamwork played a pivotal role in achieving our objectives.

1. Defined Roles and Responsibilities
Each team member had clearly assigned roles, ensuring tasks were distributed efficiently:

- **Anshu Reddy Dhamana**: Took charge of data collection, cleaning, and ETL operations, ensuring raw data was prepared for analysis. Anshu also contributed to exploratory data analysis (EDA), uncovering valuable trends and patterns in the data.
- **Brendan Chao**: Focused on data modeling, creating ER diagrams, and setting up the data warehouse, ensuring a strong structural foundation for the project. Brendan also led efforts in data visualization, presenting insights in an impactful way.
- **Sri Gopi Sarath Gode**: Played a key role in data collection and EDA to uncover initial insights. Sri also supported ETL operations and data visualization, bridging raw data and actionable insights.

- **Srithareddy Devireddy**: Took responsibility for data cleaning and setting up the data warehouse, ensuring the processed data was structured and stored efficiently. Srithareddy also contributed to data modeling and ER diagram design, critical for system organization.
- **Vinuthna Papana**: Balanced a diverse set of responsibilities, including data collection, cleaning, and ETL operations, ensuring data quality and transformation. Vinuthna also worked on EDA and visualizations, contributing significantly to uncovering and presenting insights.

2. Collaboration and Coordination

- Weekly team meetings ensured alignment on milestones and tasks. These discussions created an open platform for sharing progress, identifying bottlenecks, and resolving challenges collaboratively.
- Team members worked in parallel streams (e.g., data cleaning and modeling happening simultaneously), ensuring efficiency and reducing delays.

3. Effective Communication

- Clear communication channels, including group chats and scheduled calls, ensured everyone was updated in real time.
- Constructive feedback and brainstorming sessions allowed the team to refine workflows and improve outcomes.

4. Problem-Solving Together

- Challenges such as managing large datasets and optimizing ETL pipelines were tackled as a team. For instance, members collaborated to resolve issues related to missing or inconsistent data, drawing on their combined expertise.

5. Supporting Each Other

- Mutual support was a key element of the teamwork. For example, team members readily stepped in to assist others with complex tasks like data modeling or visualization, ensuring the project stayed on track.

6. Leveraging Individual Strengths

Each team member's strengths were strategically utilized. For example:

- Anshu's expertise in data cleaning ensured high data quality.
- Brendan's skills in modeling and visualization streamlined the transition from raw data to actionable insights.
- Sri Gopi's balance of analysis and ETL tasks bridged technical and analytical components.
- Srithareddy's focus on data organization created a robust warehouse structure.
- Vinuthna's multitasking ability ensured timely delivery of both ETL and visualization tasks.

How Teamwork Enhanced Project Outcomes

- **Efficiency**: Role clarity and parallel workstreams minimized delays and maximized output.
- **Quality**: Collaboration ensured thorough reviews and high-quality deliverables.
- **Innovation**: The diverse skill sets of the team sparked creative solutions to challenges, such as optimizing query performance and creating intuitive dashboards.

PRACTICED PAIR PROGRAMMING

Throughout the project, pair programming was an essential practice that improved the quality of our work and strengthened collaboration within the team. Pair programming allowed two members to work together on the same task—one as the "driver" (writing the code) and the other as the "navigator" (reviewing, providing feedback, and guiding)—resulting in efficient problem-solving and better outcomes.

ETL Pipeline Development

- Anshu Reddy Dhamana and Vinuthna Papana worked together on developing the ETL pipeline.
- Anshu focused on writing Python scripts to handle data transformations.
- Vinuthna reviewed the scripts, suggested optimizations, and ensured that the pipeline adhered to the project's goals.

Data Cleaning

- Srithareddy Devireddy and Sri Gopi Sarath Gode paired up to clean and preprocess the raw data.
- Srithareddy implemented cleaning scripts to handle missing values and duplicates.
- Sri Gopi Sarath verified the data quality and ensured consistency before moving to the next stage.

Data Modeling and Schema Design

- Brendan Chao and Srithareddy Devireddy collaborated on designing the star schema and creating ER diagrams.
- Brendan drafted the relationships between fact and dimension tables.
- Srithareddy reviewed and ensured alignment with project requirements, suggesting refinements to improve query performance.

SQL Query Optimization

- Sri Gopi Sarath Gode and Brendan Chao worked together on optimizing SQL queries for Amazon Redshift.
- Sri Gopi Sarath wrote the initial queries to analyze sales trends.
- Brendan tested the queries, optimized them for performance, and validated the results.

Dashboard Development

- Vinuthna Papana and Brendan Chao paired up to create visualizations and dashboards.
- Vinuthna developed initial dashboard designs in Tableau.
- Brendan reviewed the dashboards for accuracy, refined visual elements, and ensured they effectively conveyed the insights.

Benefits of Pair Programming

- **Enhanced Code Quality**
  Pairing ensured that every piece of code was reviewed in real-time, leading to fewer errors and more maintainable scripts.
- **Faster Problem-Solving**
  Collaboration between members, such as Anshu and Vinuthna on the ETL pipeline, helped solve technical challenges more efficiently.
- **Knowledge Sharing**
  Team members learned from each other's expertise. For instance, Sri Gopi Sarath gained deeper insights into SQL optimization techniques from Brendan, while Srithareddy contributed his experience in data modeling.
- **Stronger Collaboration**
  Working closely together, such as Anshu and Vinuthna or Brendan and Srithareddy, fostered trust and better communication within the team.
- **Reduced Errors**
  Real-time feedback, such as Sri Gopi Sarath reviewing Srithareddy's cleaning scripts, ensured that errors were identified and corrected early.

Lessons Learned from Pair Programming

- **Time Efficiency**: Pair programming required careful scheduling, as tasks involving two people took longer than solo work but delivered higher quality.
- **Balancing Solo and Collaborative Efforts:** While pair programming was effective for complex tasks, simpler tasks benefited from independent efforts.
- **Team Synergy**: Pair programming helped team members understand each other's working styles, leading to stronger teamwork overall.

PRACTICED AGILE/SCRUM (1-WEEK SPRINTS)

For this project, we implemented Agile/Scrum methodology with 1-week sprints to manage tasks effectively and ensure timely completion of deliverables. Using the timeline of deliverables, we aligned our work into structured sprints, focusing on iterative progress and continuous improvement. Below is a detailed explanation of how Agile practices were used and the artifacts we maintained to document the process.

Sprint Structure and Deliverables

Sprint Number	Sprint Goals	Dates	Deliverables
Sprint 1	Collect and preprocess raw data.	09-23-2024 to 09-30-2024	- Data Collection from UCI Machine Learning Repository. - Initial assessment of data structure.
Sprint 2	Clean and organize the data.	10-01-2024 to 10-07-2024	- Handle missing values, remove duplicates, correct inconsistencies. - Validate data quality.
Sprint 3	Design data model and star schema.	10-08-2024 to 10-14-2024	- Create a star schema with fact and dimension tables. - Prepare ER diagrams.
Sprint 4	Implement the ETL pipeline.	10-15-2024 to 10-21-2024	- Develop and test the ETL pipeline using Python. - Verify data loading into Amazon Redshift.
Sprint 5	Set up data warehouse and conduct EDA.	10-22-2024 to 10-28-2024	- Set up tables in Amazon Redshift. - Conduct EDA to uncover trends and validate data integrity.
Sprint 6	Execute advanced data queries and generate insights.	10-29-2024 to 11-04-2024	- Write complex SQL queries for sales trends and customer behavior. - Aggregate actionable insights.
Sprint 7	Create visualizations and dashboards.	11-05-2024 to 11-09-2024	- Develop Tableau dashboards to present findings. - Validate dashboards for data accuracy.
Sprint 8	Finalize project and compile the report.	11-10-2024 to 11-11-2024	- Summarize outcomes and prepare the final report. - Submit deliverables.

Artifacts for Agile Evidence

We maintained the following artifacts to track our Agile process and document progress:

Meeting Minutes:

- For Sprint Planning at the start of each sprint.
- Daily stand-ups to discuss progress, blockers, and next steps.
- Retrospectives at the end of each sprint to reflect on successes and areas for improvement.

Sprint Backlog:

- A list of tasks and deliverables planned for each sprint.
- Tasks included data cleaning, modeling, ETL setup, SQL query development, and visualization creation.

Kanban Board:

- A visual representation of task progress categorized into To Do, In Progress, and Done.
- Updated in real-time to reflect ongoing sprint activities.

Burndown Charts:

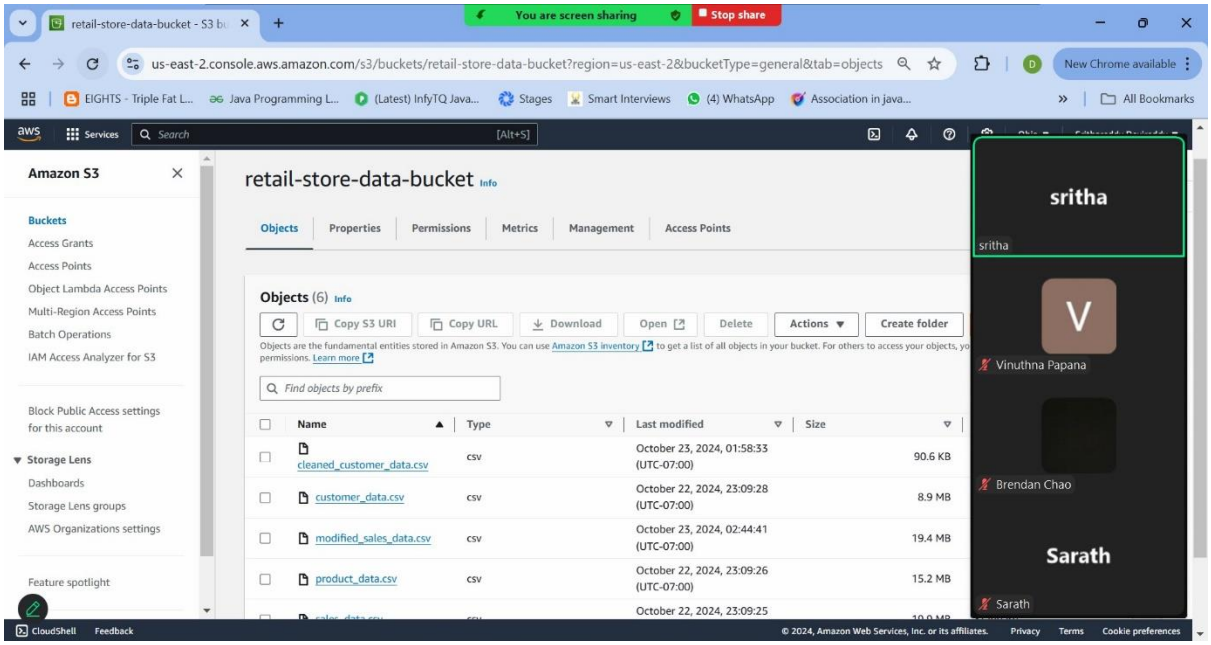
- Charts to monitor sprint progress by tracking completed tasks against the sprint timeline.

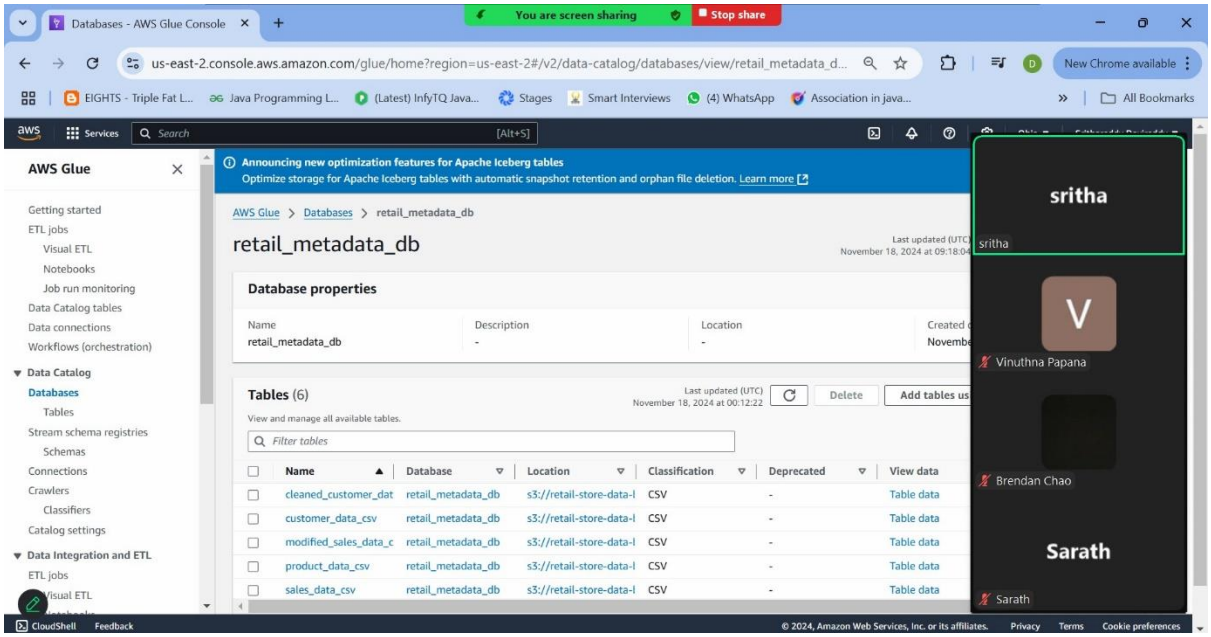
Deliverable Tracker:

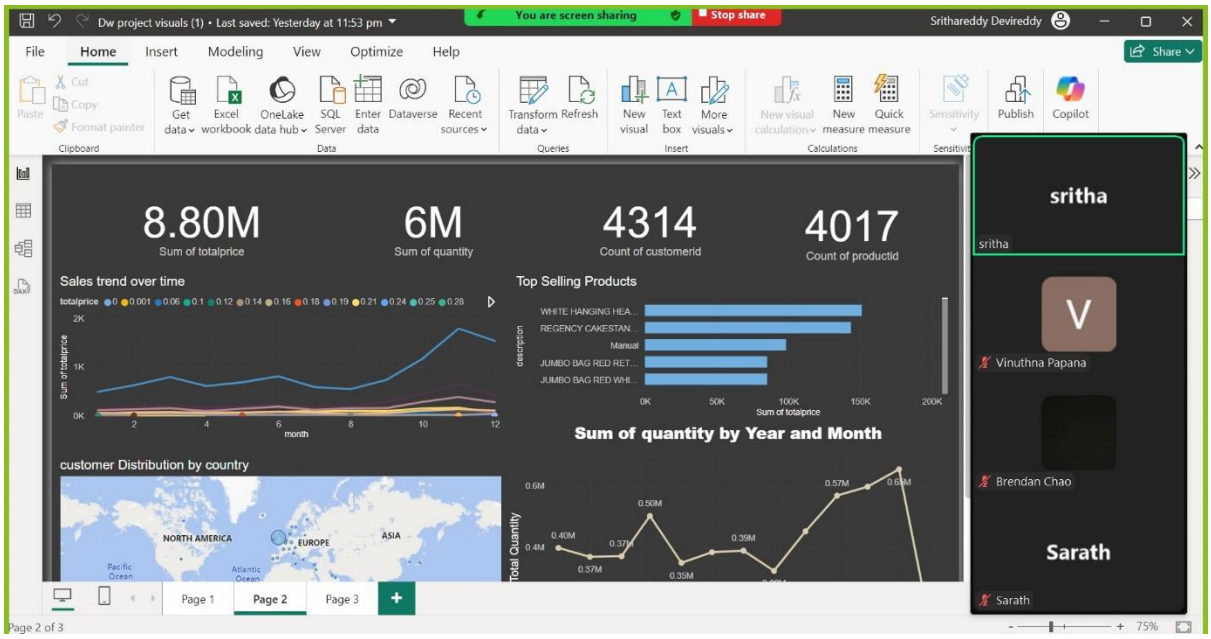
- A timeline matching the project's deliverables (e.g., Data Cleaning completed on 10-02-2024).

Final Presentation Slides:

- Slides showcasing how Agile/Scrum helped us complete each sprint goal on time.







DATABASE TECHNIQUES USED

- **Schema Design**: Implemented a star schema with a fact table and multiple dimension tables for optimized querying.
- **Data Transformation**: Utilized SQL and Python for cleaning, aggregating, and transforming data into analytical formats.
- **Advanced Querying**: Designed SQL queries to extract insights on customer behaviors, product performance, and seasonal trends.
- **Performance Optimization**: Indexed critical columns and optimized queries to enhance Redshift performance.

Key Analysis Performed:

- Trend Analysis: Identified long-term sales patterns and customer purchase behaviors.
- Sales Metrics: Calculated key metrics like total sales, average order value, and best-selling products.
- Customer Insights: Analyzed customer segmentation and loyalty through purchase frequency and spend levels.
- Seasonal Patterns: Evaluated sales fluctuations across time dimensions to identify peak season

NEW DATABASE TOOL OR DATA WAREHOUSE TOOL

Tool Adopted: Because of its strong performance, capacity to manage intricate analytical workloads, and easy connection with the AWS ecosystem, Amazon Redshift was chosen as the data warehouse option.

Key Features Utilized:

- **Scalability and Performance**: Redshift's architecture allowed the system to efficiently scale up to accommodate large datasets, ensuring fast query execution and analysis even for high-volume data.
- **Integration with AWS Ecosystem**: Tight integration with AWS Glue for ETL processes and Amazon S3 for raw data storage made it easy to ingest, transform, and load data into the warehouse.
- **Data Modeling**: The project utilized a star schema design in Redshift, with a central fact table and multiple dimension tables, to enable intuitive and efficient querying of retail data.
- **Advanced Querying**: SQL capabilities in Redshift were leveraged to perform advanced analytics, including customer segmentation, sales trend analysis, and product performance evaluation.
- **Enhanced Security:** Leveraged IAM roles to control data access and VPC configurations to ensure secure communication between services within the AWS network.
- **Outcome**: The project's capacity to process and analyze data was greatly enhanced by Amazon Redshift, which made it possible to produce useful insights like determining best-selling items, comprehending consumer purchasing tendencies, and spotting seasonal sales patterns. Its implementation was essential to providing the project with a scalable and secure data warehousing solution.

DATA MODELING TECHNIQUES

Data Modeling Approach:

- The project adopted a star schema as the core data modeling technique, designed for analytical querying, scalability, and performance. This schema allowed for a highly optimized data warehouse structure that supports intuitive and efficient data retrieval.

Fact Table Design:

- A sales transactions fact table was created as the central hub, containing key metrics such as total sales amount, transaction quantity, and invoice ID.
- Each transaction was linked to relevant dimensions through foreign keys, ensuring referential integrity and simplifying query execution.
- The fact table was structured to allow rapid aggregation and computation of critical metrics, such as revenue trends, top-selling products, and customer purchase patterns.

Dimension Tables:

- Customer Dimension: Contained detailed customer data, such as customer ID, country, and demographics, enabling geographic and customer behavior analysis.
- Product Dimension: Stored product-related attributes, including product ID, descriptions, and pricing details, which were crucial for product performance insights.
- Time Dimension: A meticulously designed time-based table that facilitated seasonal analysis by including columns for year, quarter, month, week, and day.

Design Principles:

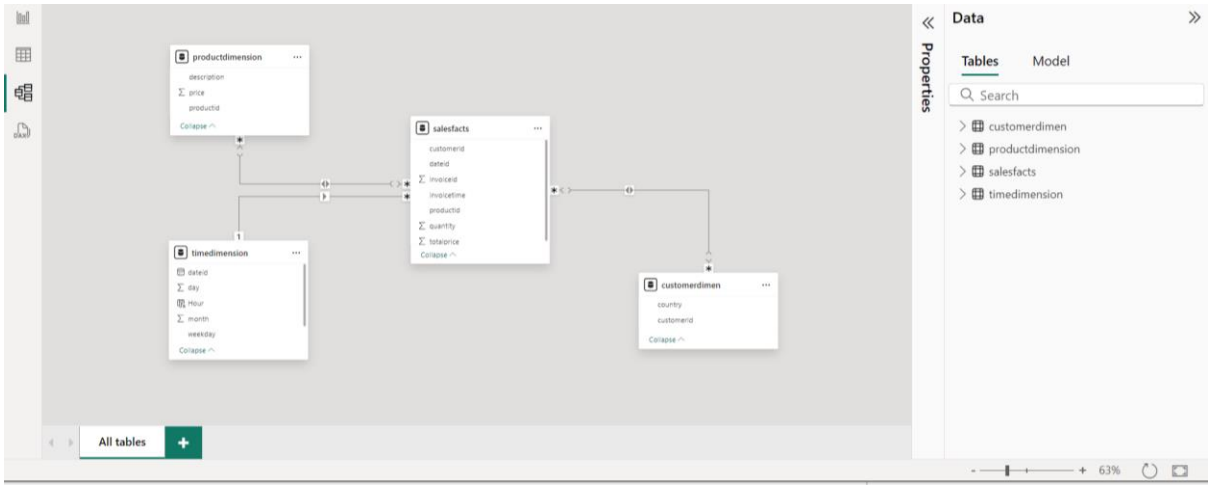
- Normalization for Dimensions: The dimension tables were normalized to eliminate redundancy, ensuring data consistency and maintainability.
- Denormalization for Facts: The fact table was deliberately denormalized to optimize query performance by reducing the need for complex joins.
- Efficient Indexing: Appropriate indexes were applied to foreign keys in the fact table and primary keys in dimension tables to enhance query performance.
- Hierarchical Relationships: The time dimension supported hierarchical navigation (e.g., Year → Quarter → Month), which facilitated drill-down capabilities in visualizations.

Integration with Amazon Redshift:

- Amazon Redshift was used to store the star schema, leveraging its scalability and distributed architecture to handle large volumes of transactional data.
- The schema design was compatible with Redshift's columnar storage and parallel processing, ensuring high-speed queries and seamless integration with analytics tools.

Outcome and Benefits:

- The star schema provided a user-friendly structure for creating dashboards and visualizations in Power BI.
- Optimized Query Performance: The design ensured that even complex analytical queries, such as identifying top-performing products or understanding long-term customer trends, were executed efficiently.
- Scalable Design: The model was scalable to accommodate future data expansions, making it a robust foundation for ongoing business analytics.
- Actionable Insights: Enabled in-depth examination of seasonal patterns, consumer behavior, and sales performance, resulting in well-informed business decision-making.
- These data modeling strategies were used in the project to successfully convert unstructured retail sales data into an analytics-ready format that facilitates strategic planning and in-depth insights.



HOW ANALYTICS SUPPORTS BUSINESS DECISIONS

Purpose:

- Data-driven decision-making for enhanced business performance was made possible by the utilization of analytics to transform unprocessed retail sales data into actionable insights.

Key Analytics Insights:

- Customer Insights: Identified top-performing customer segments based on purchase behavior, enabling targeted marketing strategies.
- Product Performance: Highlighted best-selling products and underperforming items, helping optimize inventory management and product promotions.
- Seasonal Trends: Analyzed time-based patterns in sales, such as peak seasons, aiding in resource allocation and demand forecasting.
- Revenue Trends: Tracked total sales and revenue growth over time, helping evaluate business health and set strategic goals.

Business Benefits:

- Enhanced Decision-Making: Stakeholders were able to make well-informed decisions because to data visualization dashboards (such as Power BI/Tableau), which offered an interactive, clear view of important KPIs.
- Better Customer Engagement: Personalized campaigns were made possible by insights into consumer behavior, which raised customer happiness and retention.
- Cost Optimization: Identified inefficiencies in operations, such as low-performing regions or products, guiding cost-saving measures.
- Strategic Planning: Data-driven insights on market trends and customer preferences enabled forward-thinking strategies for long-term growth.

Implementation Highlights:

- Amazon Redshift was used to create a data warehouse for consolidated data storage and speedy retrieval.
- ETL pipelines were implemented to keep current, high-quality data for analysis.
- improved communication with non-technical stakeholders by presenting analytics in an approachable manner using visualization tools.

The study demonstrated how data analytics is essential to corporate performance by enabling improved forecasting, resource management, and strategic planning.

RDBMS:

For this, Amazon Redshift is used as the RDBMS to store and manage the processed data. With DBeaver, we were able to connect to the Redshift cluster in order to validate schema designs, execute queries, and debug data loading issues. DBeaver provided an easy interface for looking into tables, testing relationships in the star schema, and fast execution of SQL commands. This tool was crucial for ensuring the integrity of the fact and dimension tables and optimizing query performance during the project.

DataWarehouse:

The project used Amazon Redshift as a primary data warehouse to store and analyze structured data. It supported fast query performance through columnar storage and optimization techniques such as distribution and sort keys. Redshift integrated well with AWS Glue for ETL and Power BI for visualization, hence making it a very strong solution for handling large-scale retail data while supporting complex analytical queries and business insights.

Includes DB Connectivity / API Calls:

The project utilized **Python** to establish database connectivity with **Amazon Redshift** using the psycopg2 library. Python scripts automated the ETL process, including data extraction from S3, transformation, and loading into Redshift. API calls to AWS services, such as S3 and Glue, were facilitated using the **Boto3** library, ensuring seamless integration and efficient data handling throughout the pipeline.

Already mentioned about roles and responsibilities in the above pages.

CRediT Statement for the Project

- **Conceptualization**: Conceptualized the project by identifying the goals of analyzing retail sales data using a scalable data warehousing solution in Amazon Redshift.
- **Data Curation**: Performed to clean, preprocess, and catalog raw data from the "Online Retail II" dataset into S3, enabling its integration with Redshift.
- **Formal Analysis**: Utilized Python and SQL to calculate key metrics such as total sales, seasonal trends, and customer purchasing patterns.
- **Methodology**: Designed and implemented the ETL pipeline, including schema design, data transformations, and automation for loading data into Redshift.
- **Project Administration**: Managed to oversee the setup of AWS resources, database configurations, and the orchestration of Glue jobs.
- **Resources**: Provisioned and configured AWS services, including Redshift, S3, Glue, and IAM roles, for secure and efficient data handling.

- **Software**: Developed Python scripts for ETL processes, API calls, and automated workflows to ensure smooth execution of data pipelines.
- **Validation**: Ensured data accuracy and query performance by testing the star schema implementation and validating data transformations.
- **Visualization**: Created detailed visualizations in Power BI and Tableau to showcase sales trends, best-selling products, and customer insights.
- **Writing - Original Draft**: Drafted technical reports, including methodology, ETL processes, and outcomes of the project.
- **Writing - Review & Editing**: Reviewed and refined the project report and visualizations for clarity and coherence.