2.1 Introduction to project:

In the modern era of data-driven decision-making, the ability to efficiently manage and analyze large volumes of data is crucial for businesses aiming to maintain a competitive edge. This project focuses on the development and implementation of an automated data engineering pipeline that seamlessly manages and analyzes orders data, leveraging the advanced capabilities of Amazon S3 and Microsoft Azure.

The project starts with raw orders data stored in Amazon S3, a scalable storage solution that provides high durability and availability. To streamline the data processing workflow, the pipeline employs an event-driven architecture. This means that as soon as new files are added to the S3 bucket, the system automatically triggers the data extraction process. This automation significantly reduces the need for manual intervention, ensuring that data processing is timely and efficient.

Once the data extraction is initiated, the pipeline securely transfers the data from Amazon S3 to Azure Data Lake Storage (ADLS). ADLS offers a highly scalable and secure data storage solution, optimized for big data analytics. This transfer process is designed to be efficient and reliable, maintaining data integrity throughout the process.

In Azure Data Lake Storage, the raw data undergoes a series of transformations using Azure Data Factory. Azure Data Factory is a cloud-based data integration service that orchestrates and automates the movement and transformation of data. The transformation logic applied in this stage includes data cleaning, aggregation, and structuring, which prepares the data for comprehensive analysis. This step ensures that the data is in a consistent and usable format, enhancing its value for analytical purposes.

After the data is transformed, it is loaded into an Azure SQL Database. Azure SQL Database is a fully managed relational database service that provides high availability,

performance, and security. By loading the data into this database, the project enables advanced querying and analysis capabilities. This setup allows for real-time insights into the orders data, facilitating data-driven decision-making and operational efficiency.

The project demonstrates the seamless integration of diverse cloud services, showcasing best practices in data engineering such as ensuring data integrity, optimizing performance, and maintaining scalability. It not only highlights the technical prowess required to integrate AWS and Azure platforms but also underscores the importance of automation in reducing manual workloads and improving processing times.

Furthermore, this project lays the foundation for future enhancements. Potential upgrades include incorporating real-time data processing to handle streaming data and integrating additional data sources to enrich the analysis. Advanced analytics using machine learning techniques can also be added to derive deeper insights and predictive capabilities, driving even greater business value.

In summary, this automated data engineering pipeline project exemplifies the effective use of cloud technologies to manage and analyze orders data efficiently. By automating data extraction, transfer, and transformation processes, it provides a robust solution for real-time data analysis, empowering businesses with the insights needed to make informed decisions and optimize their operations.

2.2 Background of study:

The rapid advancement of technology and the exponential growth of data have transformed the landscape of business operations and decision-making. In this context, efficient data management and analysis have become pivotal for organizations striving to harness the power

of big data. Companies generate vast amounts of data daily, which, if processed and analyzed effectively, can provide valuable insights and drive strategic decisions. However, managing such large volumes of data presents significant challenges, including data integration from multiple sources, ensuring data quality, and achieving real-time data processing.

Cloud computing platforms like Amazon Web Services (AWS) and Microsoft Azure have emerged as key enablers in addressing these challenges. AWS offers a suite of services for data storage, processing, and analytics, with Amazon S3 being one of the most widely used storage solutions due to its scalability, durability, and cost-effectiveness. Microsoft Azure, on the other hand, provides a robust set of tools for data engineering and analytics, including Azure Data Lake Storage (ADLS), Azure Data Factory, and Azure SQL Database. These platforms offer comprehensive solutions for building scalable and efficient data pipelines.

The integration of AWS and Azure services presents a powerful approach to creating an end-to-end data engineering pipeline. By leveraging the strengths of both platforms, organizations can achieve seamless data migration, transformation, and analysis.

In this project, we aim to design and implement an automated data engineering pipeline that extracts raw orders data from Amazon S3, transfers it to Microsoft Azure, and transforms it for comprehensive analysis in SQL. This pipeline will demonstrate best practices in data engineering, including data extraction, secure transfer, transformation, and loading into a SQL database. By focusing on automation and real-time processing, the project seeks to provide a scalable and efficient solution for managing and analyzing large volumes of orders data.

2.3 Novel features of Project:

This project introduces several innovative features that set it apart from conventional data engineering solutions. Leveraging the strengths of Amazon Web Services (AWS) and Microsoft Azure, the project integrates state-of-the-art technologies to create a robust, automated, and efficient data pipeline. The pipeline features an event-driven architecture,

which automatically triggers data processing upon detecting new files in Amazon S3, ensuring timely data handling with minimal manual intervention. It showcases seamless integration between AWS and Azure, utilizing AWS for scalable storage and Azure for powerful data processing and analytics. Automated data extraction and secure transfer to Azure Data Lake Storage maintain data integrity and privacy, while advanced data transformations using Azure Data Factory prepare the data for meaningful analysis. The system supports real-time data processing, providing up-to-date insights crucial for immediate decision-making. Scalability and performance optimization are achieved through best practices in data engineering, ensuring the pipeline can handle increasing data volumes and complexity. By loading transformed data into Azure SQL Database, the project enables comprehensive data analysis through advanced SQL queries. Designed with future enhancements in mind, the architecture can evolve to incorporate additional data sources, real-time streaming, and machine learning analytics. These features collectively ensure the system delivers timely, accurate insights, driving better decision-making and operational efficiency.

2.4 Problem Statement

In the current data-driven business environment, companies face significant challenges in efficiently managing and analyzing large volumes of data from diverse sources. Specifically, the process of extracting, transforming, and loading (ETL) orders data from storage systems like Amazon S3 into analytical platforms often involves manual intervention, which can lead to delays, errors, and inefficiencies. Additionally, the lack of seamless integration between different cloud services hampers the ability to perform real-time data processing and comprehensive analysis, limiting the potential for timely insights and informed decision-making.

Existing solutions often fall short in providing a fully automated, scalable, and secure pipeline

that can handle the dynamic nature of orders data. This inefficiency affects operational efficiency and hinders the ability to respond quickly to market changes and customer demands. Therefore, there is a critical need for a robust data engineering pipeline that can automate the ETL process, ensure seamless cloud integration, and support real-time data processing and advanced analytics.

This project aims to address these challenges by developing an automated data engineering pipeline that extracts raw orders data from Amazon S3, transfers it to Microsoft Azure, and transforms it for comprehensive analysis in SQL. The pipeline's event-driven architecture will automatically trigger data processing upon detecting new files, ensuring timely and efficient data handling. By leveraging the robust cloud capabilities of AWS and Azure, the project seeks to provide a reliable, scalable, and secure solution for real-time data analysis, ultimately enabling data-driven decision-making and enhancing operational efficiency.

2.5 Project Objective

The objective of this project is to develop and implement a fully automated data engineering pipeline that efficiently manages and analyzes orders data. The project aims to design an event-driven architecture that automatically triggers data extraction from Amazon S3 upon detecting new files, minimizing manual intervention and ensuring timely processing. It seeks to implement a seamless transfer of raw orders data from Amazon S3 to Azure Data Lake Storage, leveraging the strengths of both AWS and Microsoft Azure platforms. Utilizing Azure Data Factory, the project will apply sophisticated data transformation logic, including data cleaning, aggregation, and structuring, to prepare the data for comprehensive analysis. The transformed data will be loaded into Azure SQL Database, enabling advanced querying and analysis to derive meaningful business insights. The project will ensure that the data transfer and transformation processes are secure, maintaining data integrity and compliance with relevant data protection regulations. Additionally, the pipeline will be optimized for scalability

and performance, capable of handling increasing volumes of data and complex queries efficiently. Designed to support real-time data processing, the system will provide up-to-date insights for timely decision-making. Finally, the project aims to build a future-ready architecture that can be easily enhanced with additional data sources, real-time streaming capabilities, and advanced analytics using machine learning techniques. By achieving these objectives, the project will provide a robust, scalable, and efficient solution for managing and analyzing orders data, driving better decision-making and operational efficiency for businesses.

2.6 Scope of Study

The scope of this study encompasses the development and implementation of an automated data engineering pipeline designed to manage and analyze orders data effectively. It includes the process of automatically detecting and extracting new orders data files from Amazon S3 using an event-driven architecture, ensuring timely and efficient data handling. The study also covers the secure and efficient transfer of extracted data from Amazon S3 to Microsoft Azure Data Lake Storage, maintaining data integrity and security throughout the transfer process. Utilizing Azure Data Factory, the pipeline applies automated data transformation logic, including cleaning, aggregating, and structuring the raw data to prepare it for comprehensive analysis. The transformed data is then loaded into Azure SQL Database, optimized for performance and scalability to handle large volumes of data and complex queries efficiently. Additionally, the study involves implementing real-time data processing capabilities, providing up-to-date insights and minimizing latency. Ensuring data security and compliance with relevant data protection regulations is a critical aspect, including data encryption and access controls. The study also focuses on performance optimization techniques, scalable architecture design, and integration with analytical tools for advanced data analysis. Lastly, potential areas for future enhancements are identified, such as incorporating additional data

sources, integrating machine learning techniques for predictive analytics, and expanding real-time data processing capabilities. By addressing these areas, the study aims to create a comprehensive, efficient, and scalable data engineering pipeline that supports data-driven decision-making within organizations.

2.7 Feasibility Study

The feasibility study evaluates the technical, operational, and economic viability of developing an automated data engineering pipeline for managing and analyzing orders data.

Technical Feasibility

The project leverages reliable and scalable cloud platforms: Amazon Web Services (AWS) for data storage (Amazon S3) and Microsoft Azure for data processing and analysis (Azure Data Lake Storage, Azure Data Factory, and Azure SQL Database). The event-driven architecture ensures efficient and automated data processing, leveraging well-established technologies and best practices.

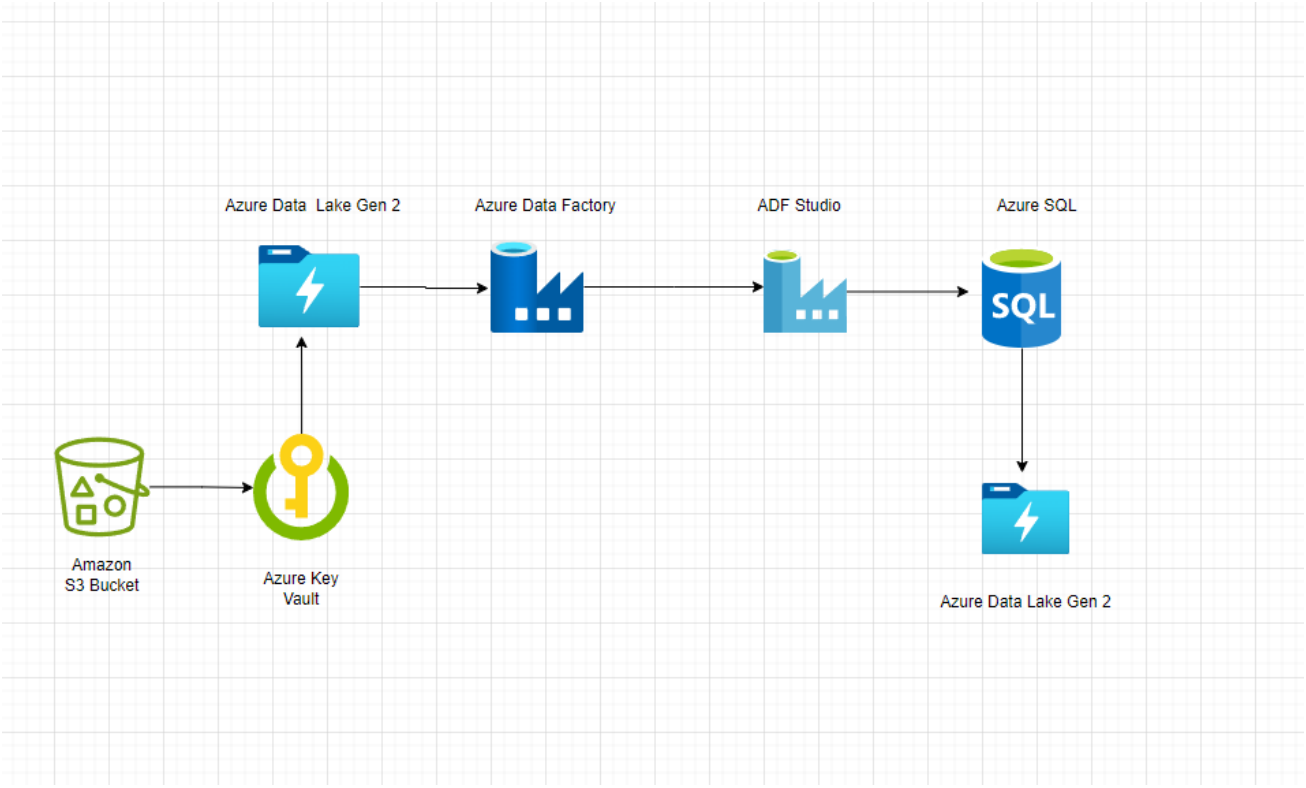
Operational Feasibility

Operationally, the project aims to reduce manual intervention through automation, enhancing efficiency and accuracy. The integration of AWS and Azure services can be managed with proper training and documentation. The scalable nature of cloud services ensures the pipeline can handle increasing data volumes, with comprehensive monitoring and logging for performance tracking.

Economic Feasibility

Economically, the pay-as-you-go pricing models of cloud services reduce upfront infrastructure investments. Automation lowers labor costs and minimizes downtime, leading to efficient resource use. The potential for real-time data insights can drive business growth and

provide a substantial return on investment (ROI). Long-term cost savings are expected from

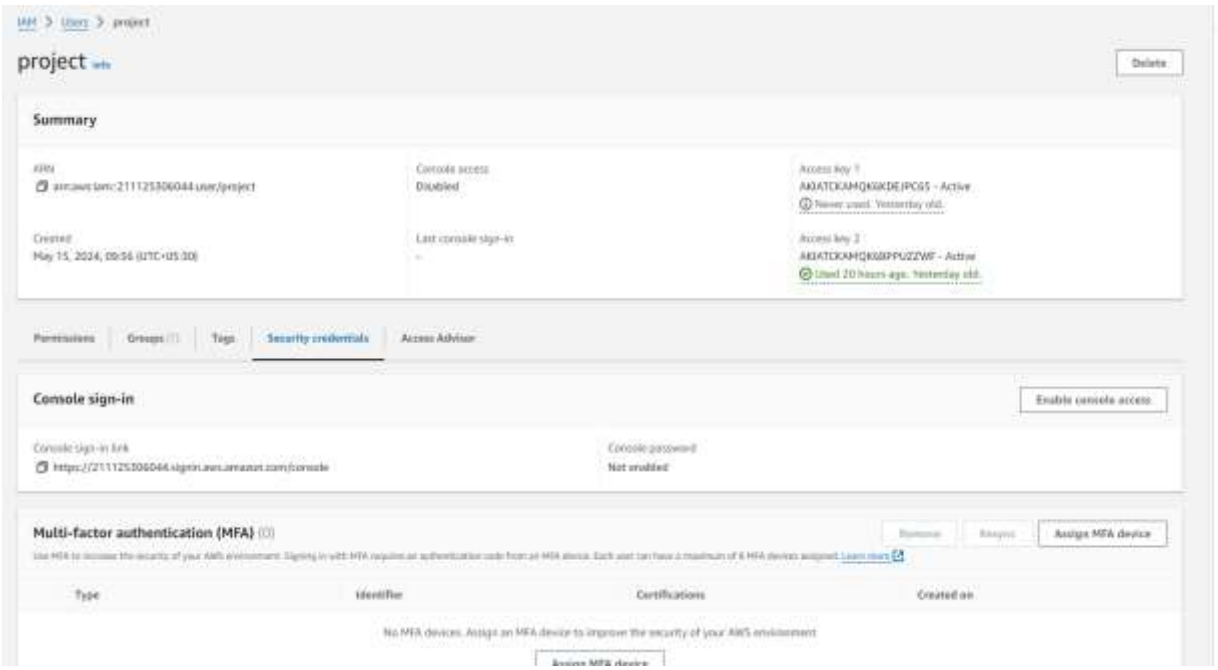


optimized data processing and storage.

5.1 Flow chart of the Project:

5.2 Amazon s3 Bucket Injestion

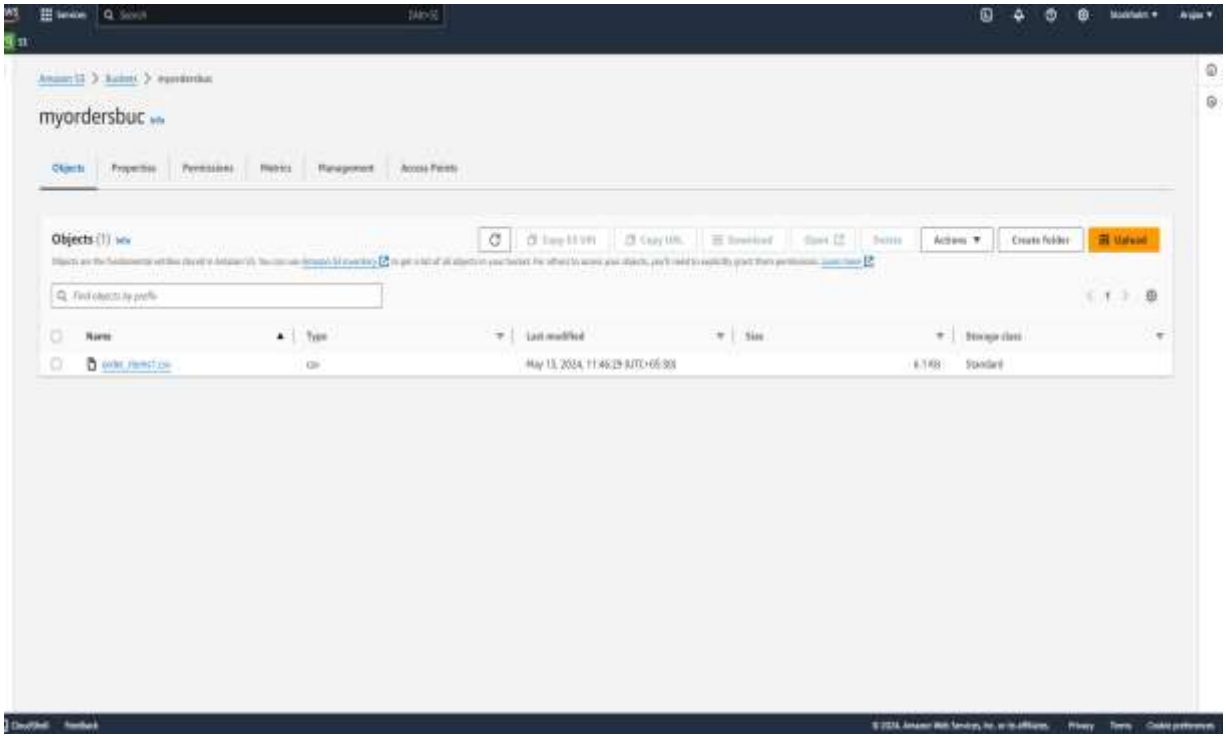
First step is to create a s3 bucket in aws where the Dataset is stored in

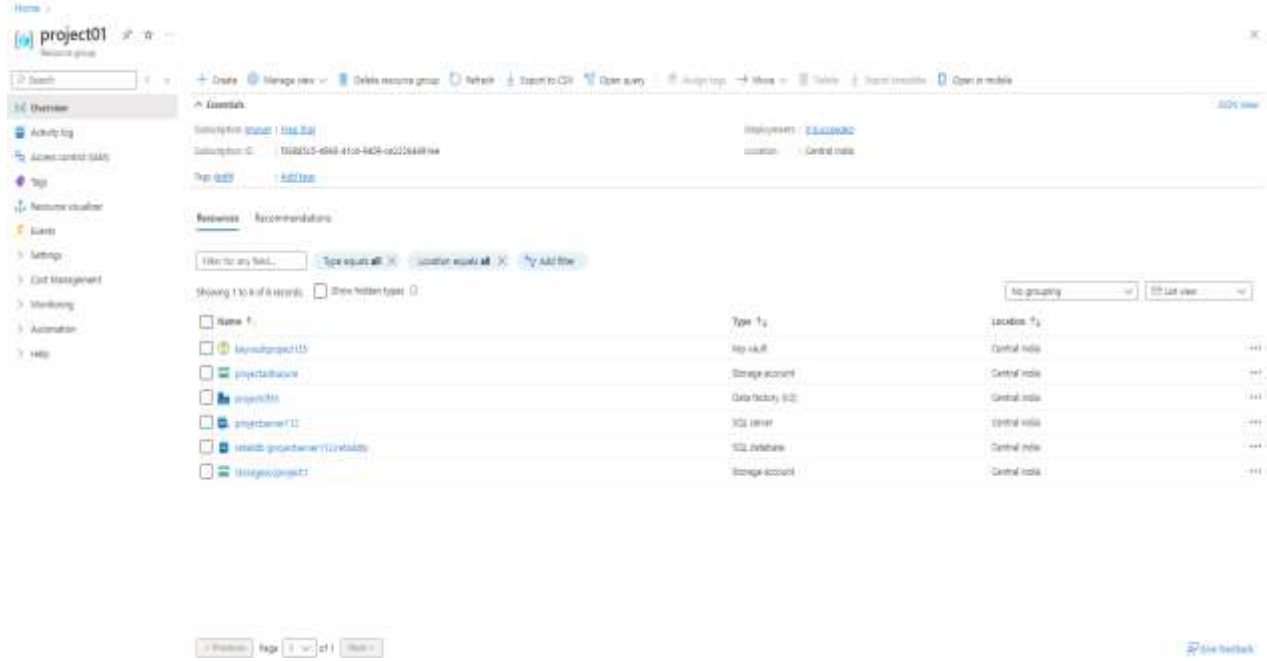


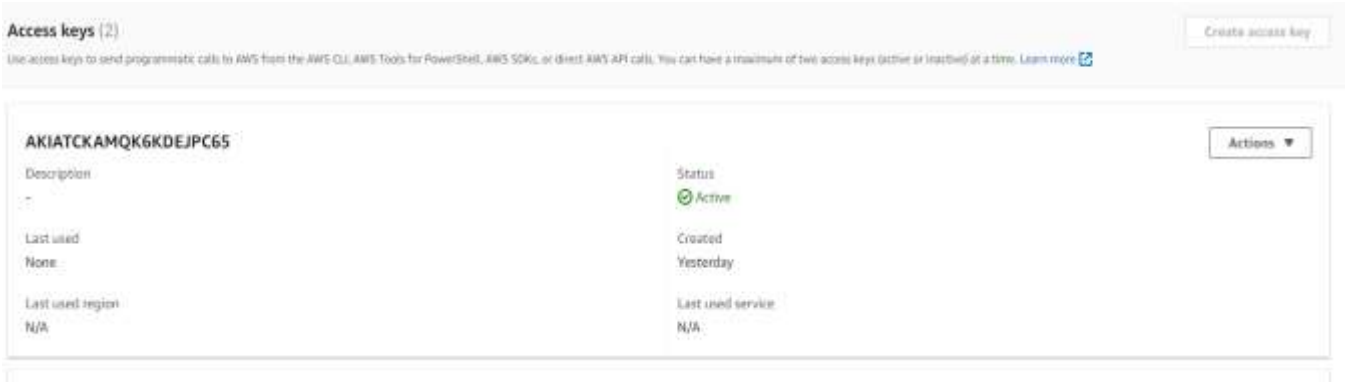
the Bucket

Second Step is to create a user in the IAM menu

Third Step is to create a Access Key







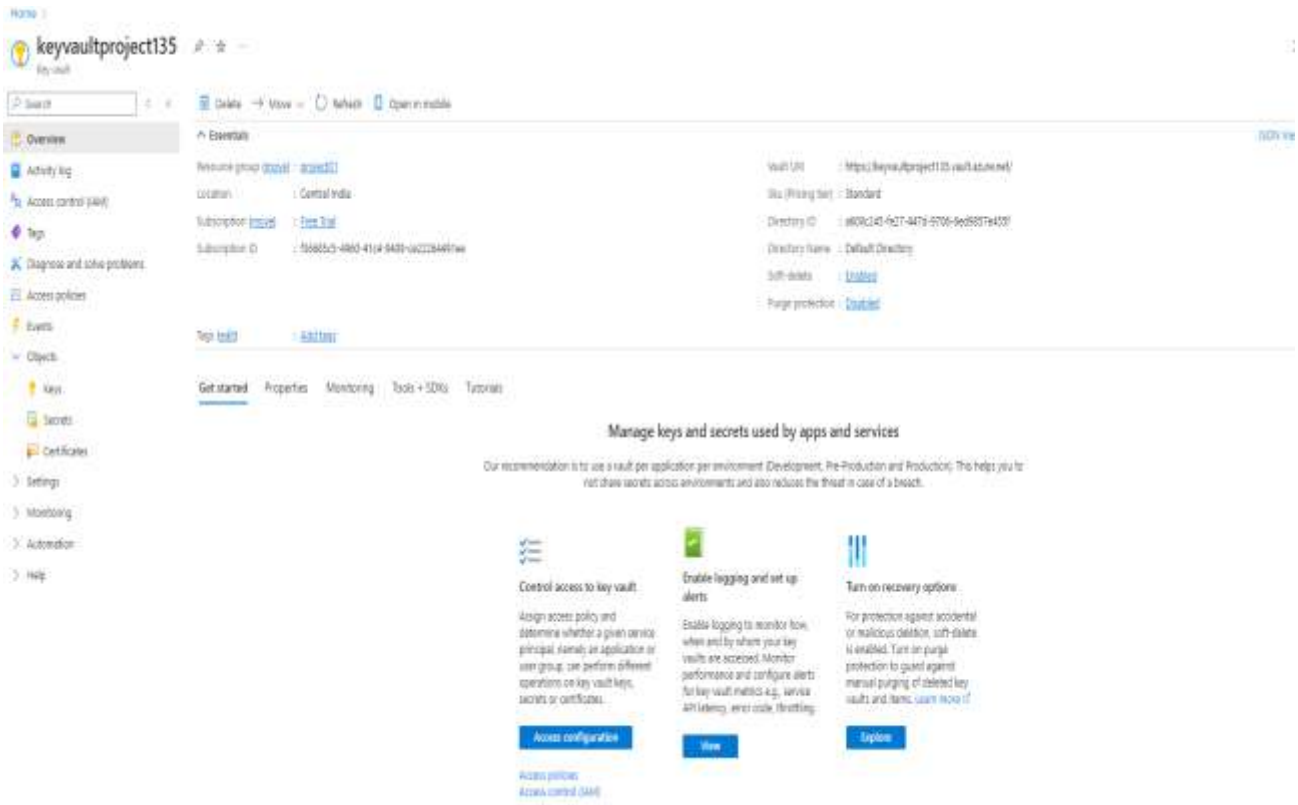
5.2 Creation of Resource Group

Under Resorce Group Creating the necessary Resources for the project

Resorce1: Azure Key Vault creation and Giving the access to Azure

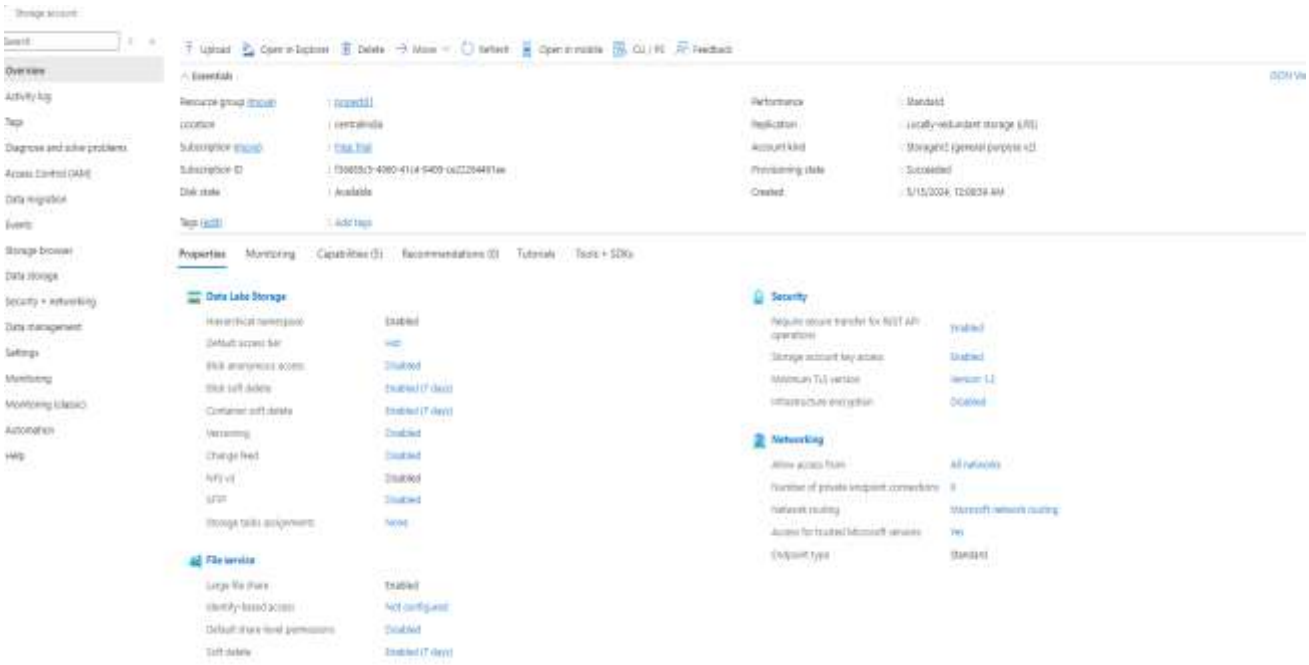
Step1:creating the key vault

Resorce1: Azure Key Vault creation and Giving the access to Azure

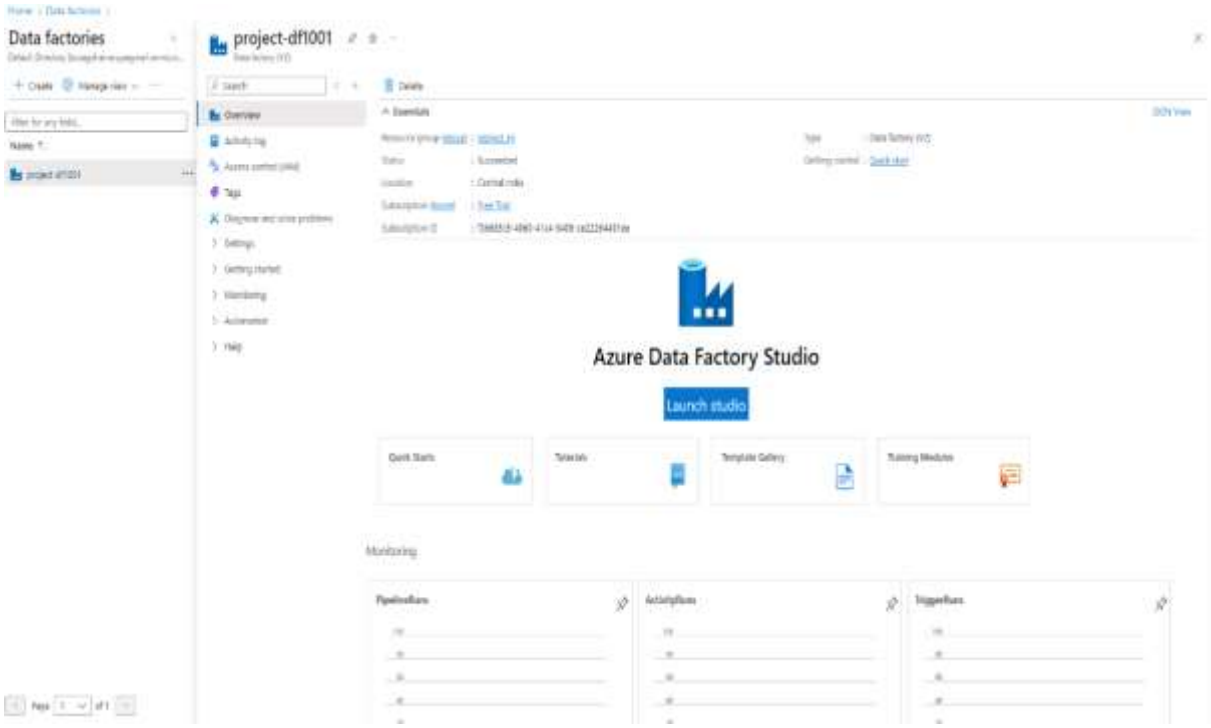


Step1:creating the secret keys

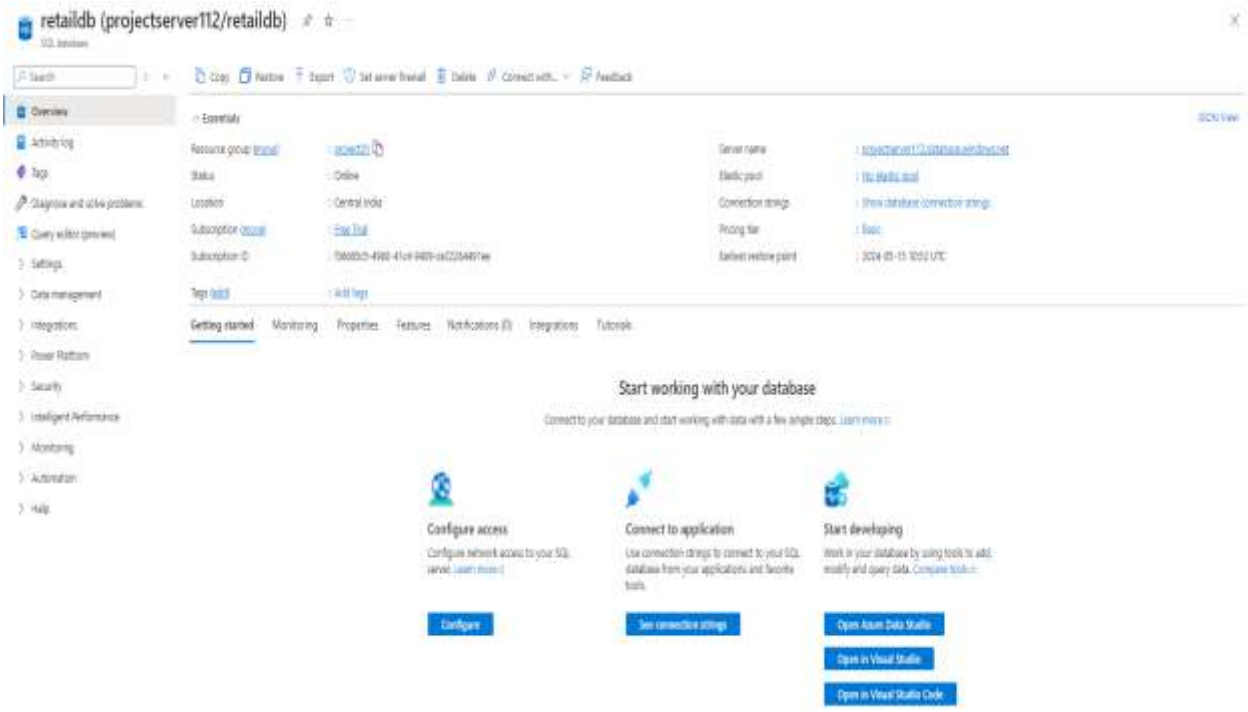
Resource 2: Creating Azure Data Lake Gen 2:



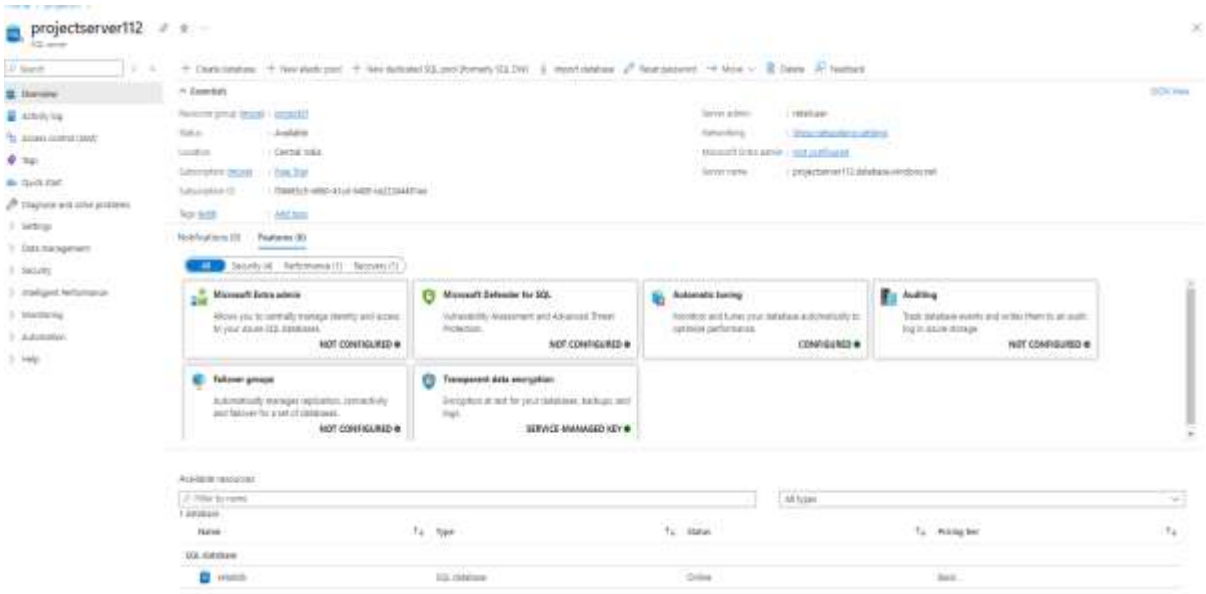
Resource 3:Creating Azure Data Factory



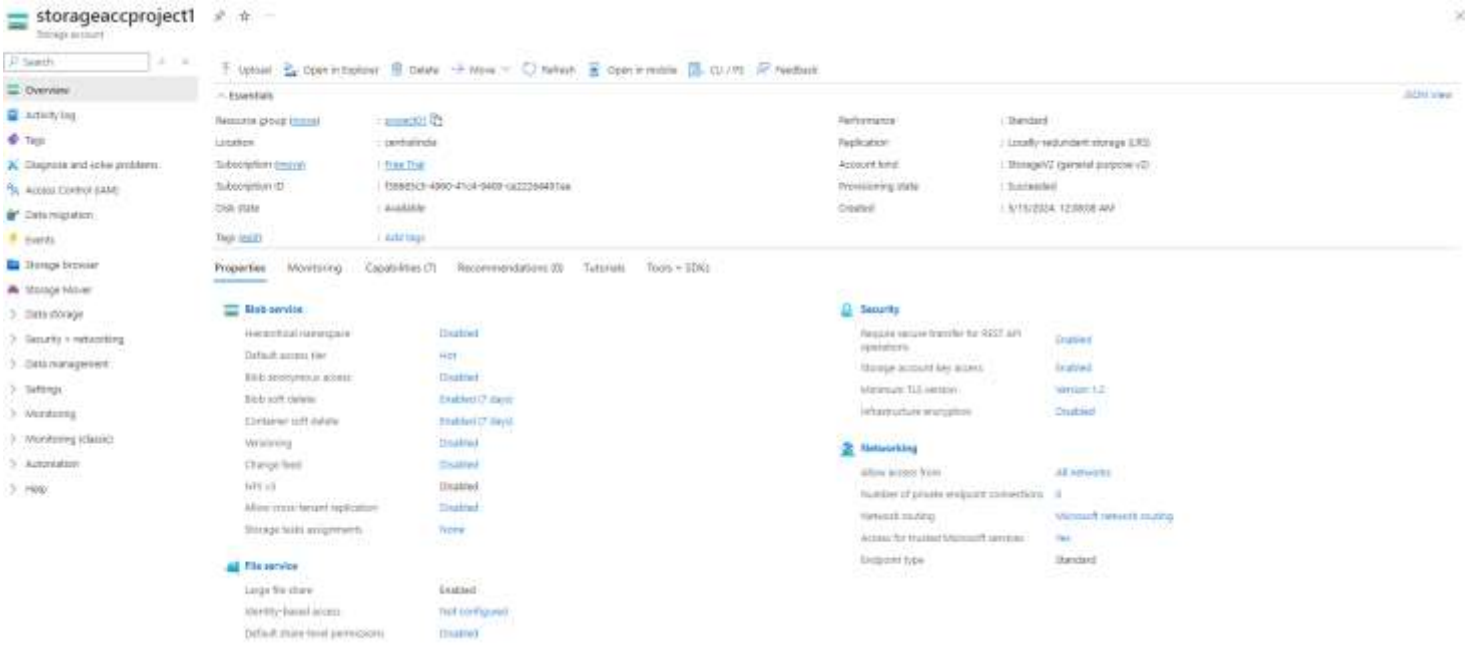
Resource 4:Creation of Server for SQL



Resource 5:SQL Storage

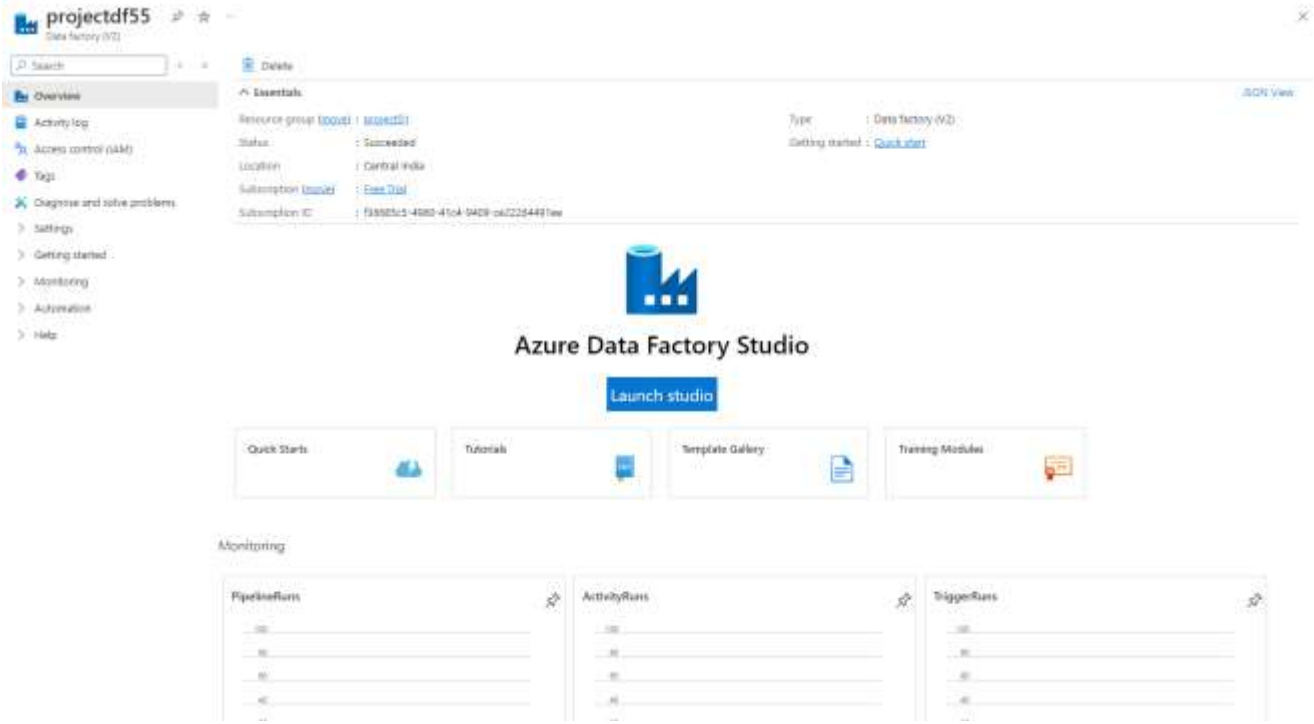


Resource 6:Storage account

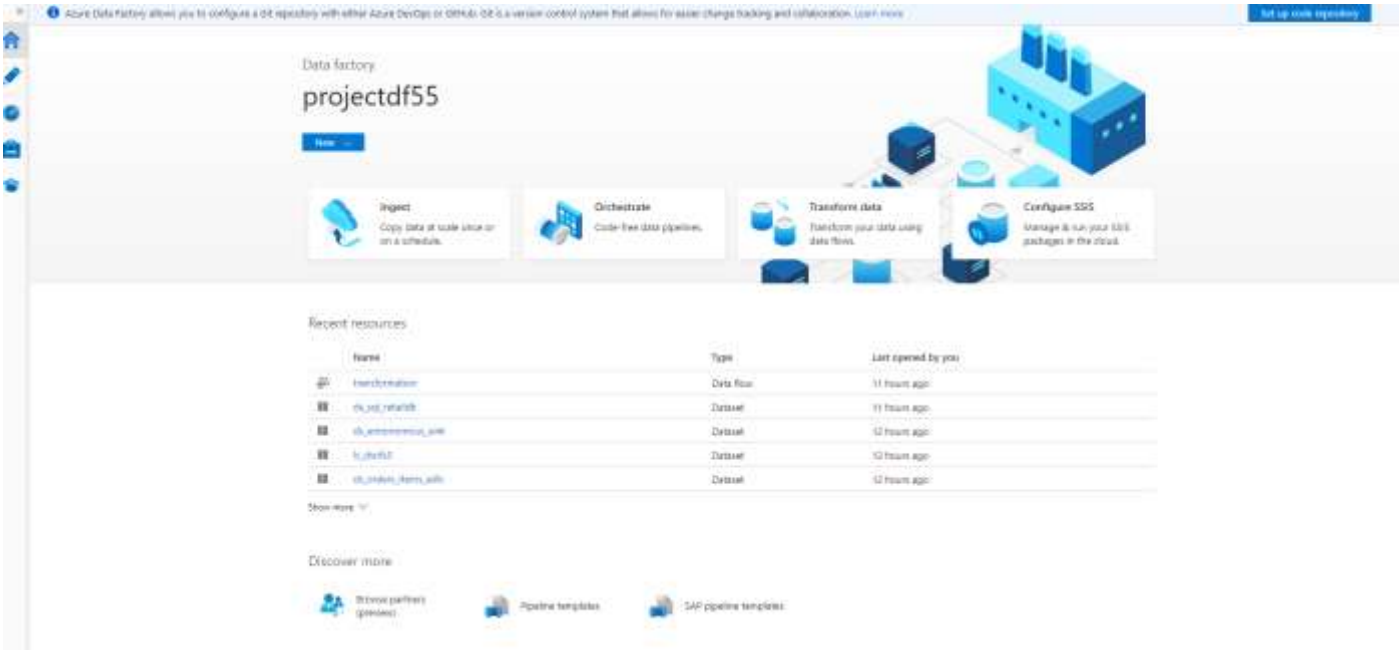


5.3 Go to Azure data Factory studio in Azure data factory

STEP1:OPEN ADLS

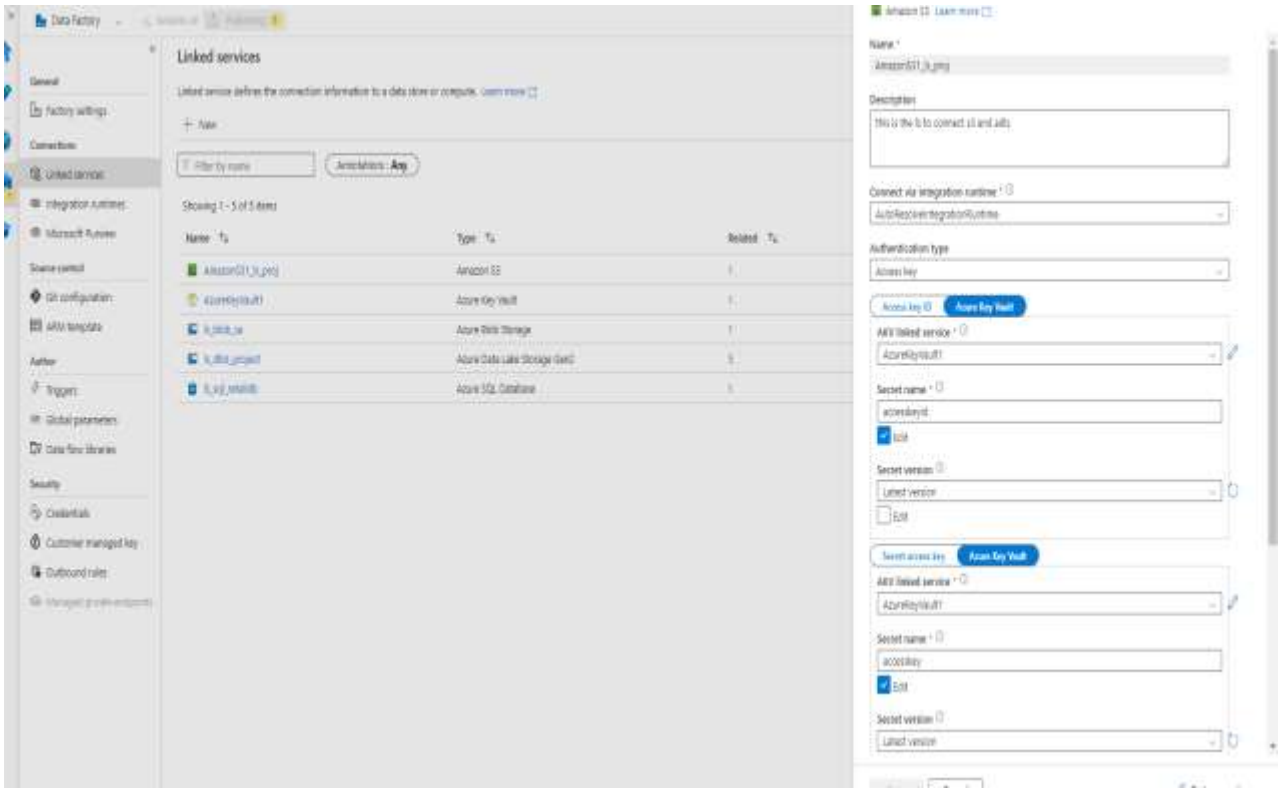


STEP2: LAUNCH THE ADLS STUDIO



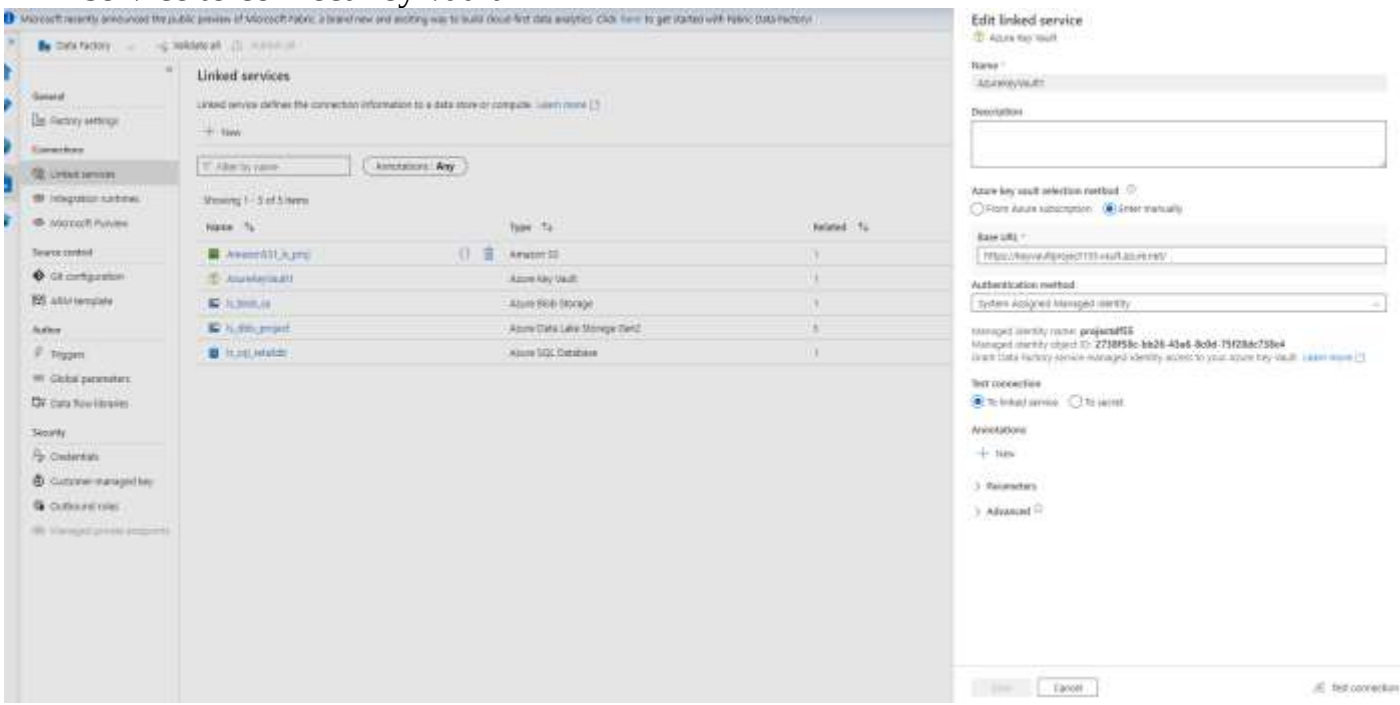
STEP 3: ADD THE LINK SERVICES

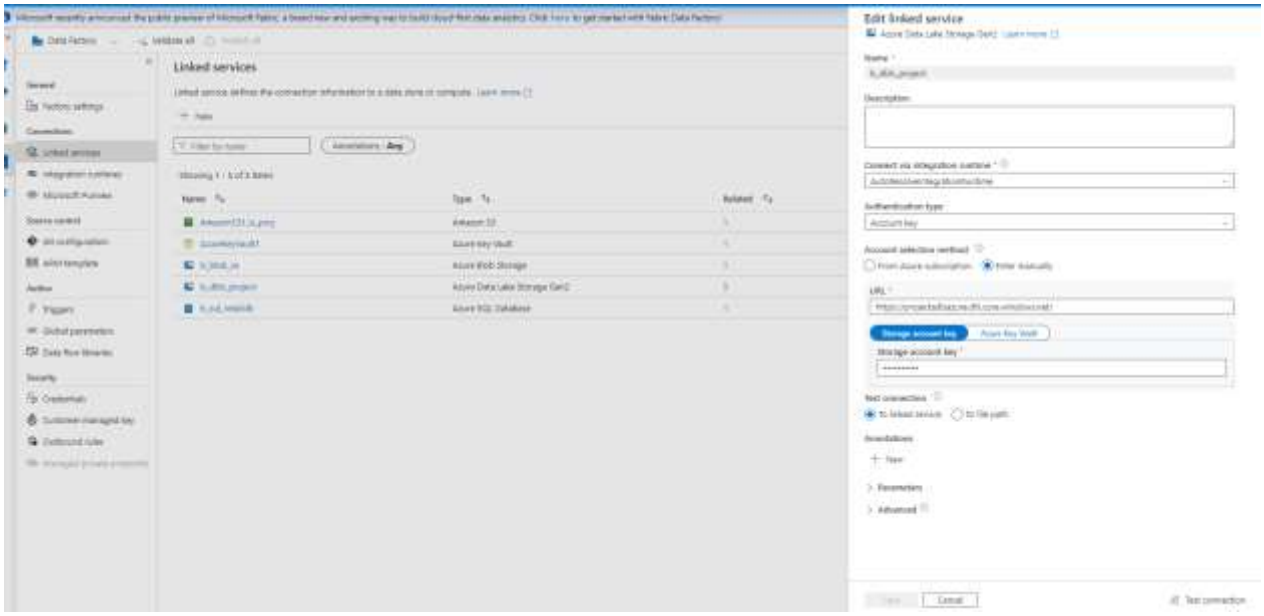
Linked service to connect amazon s3 bucket



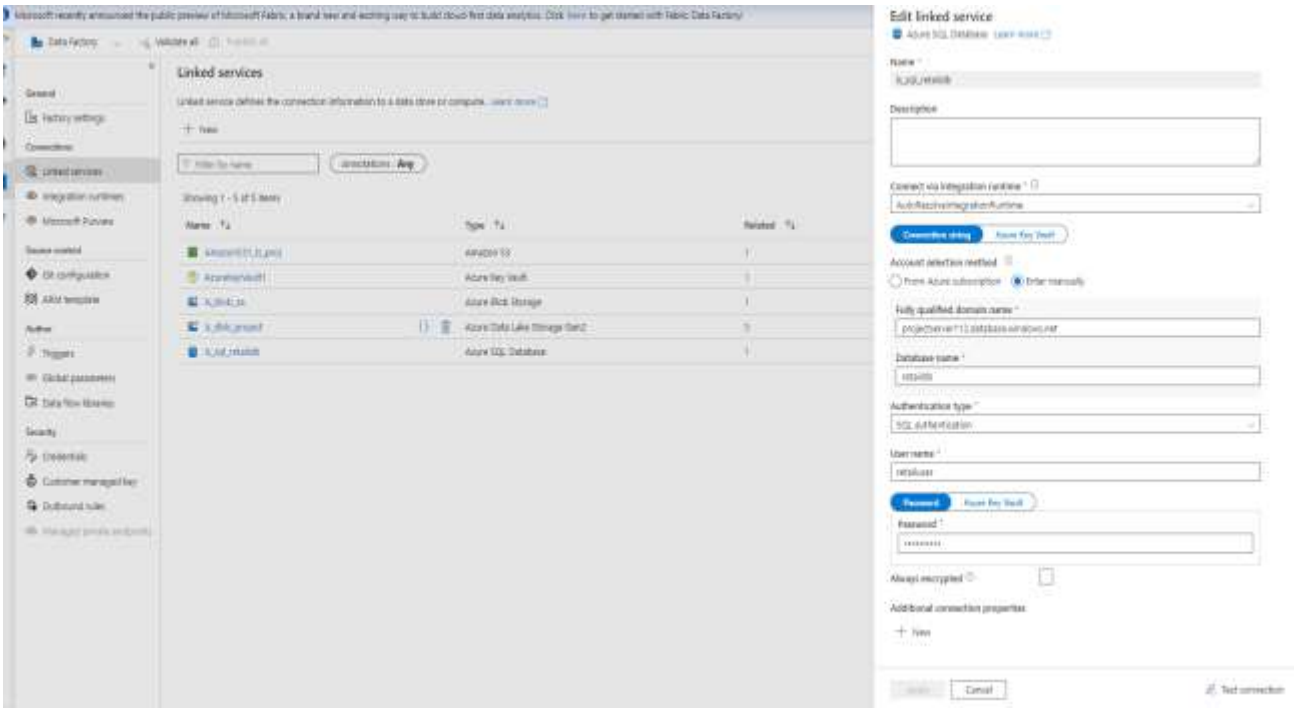
Link Service to connect to adls

Link service to connect key vault

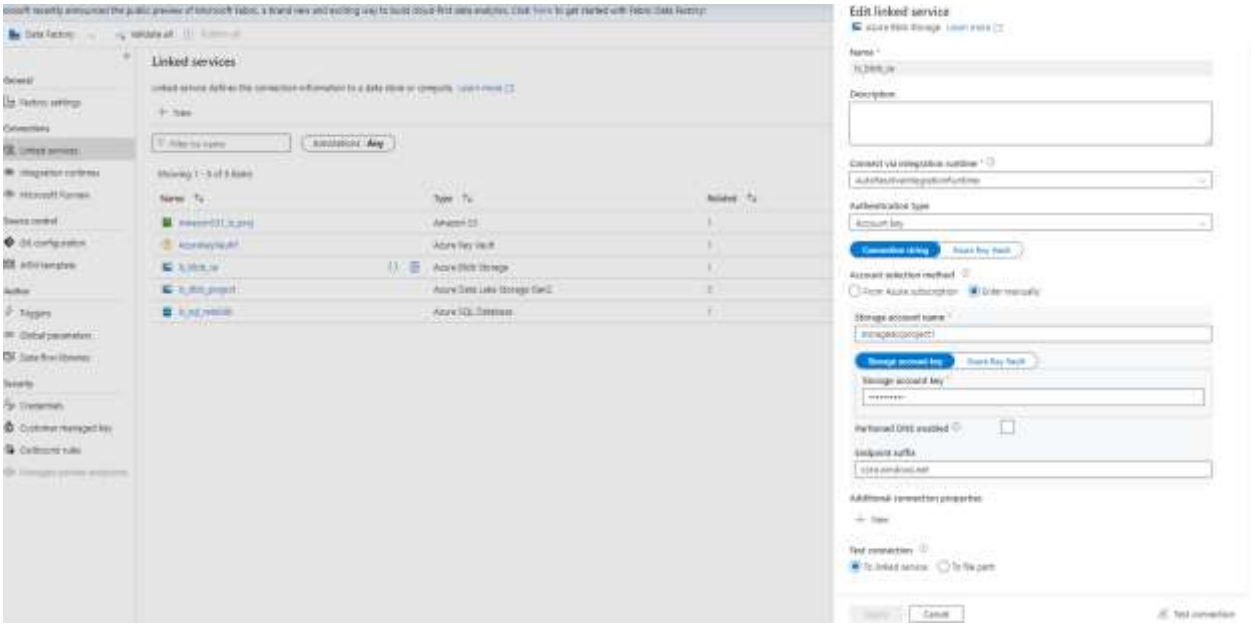



Linked services
Edit linked service

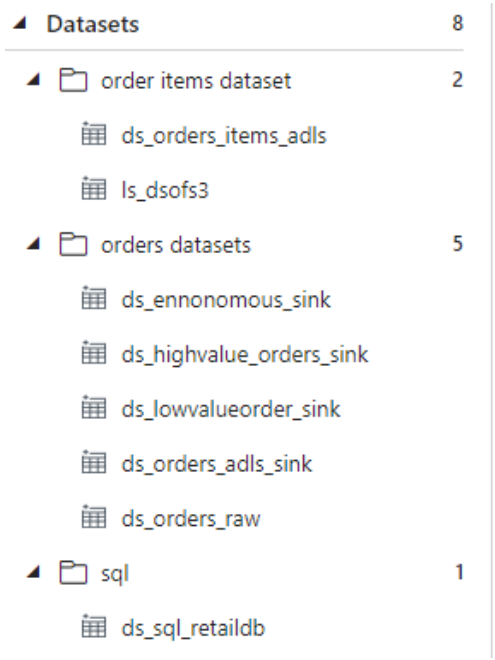
Link Service to connect blob storage



Link service to connect sql

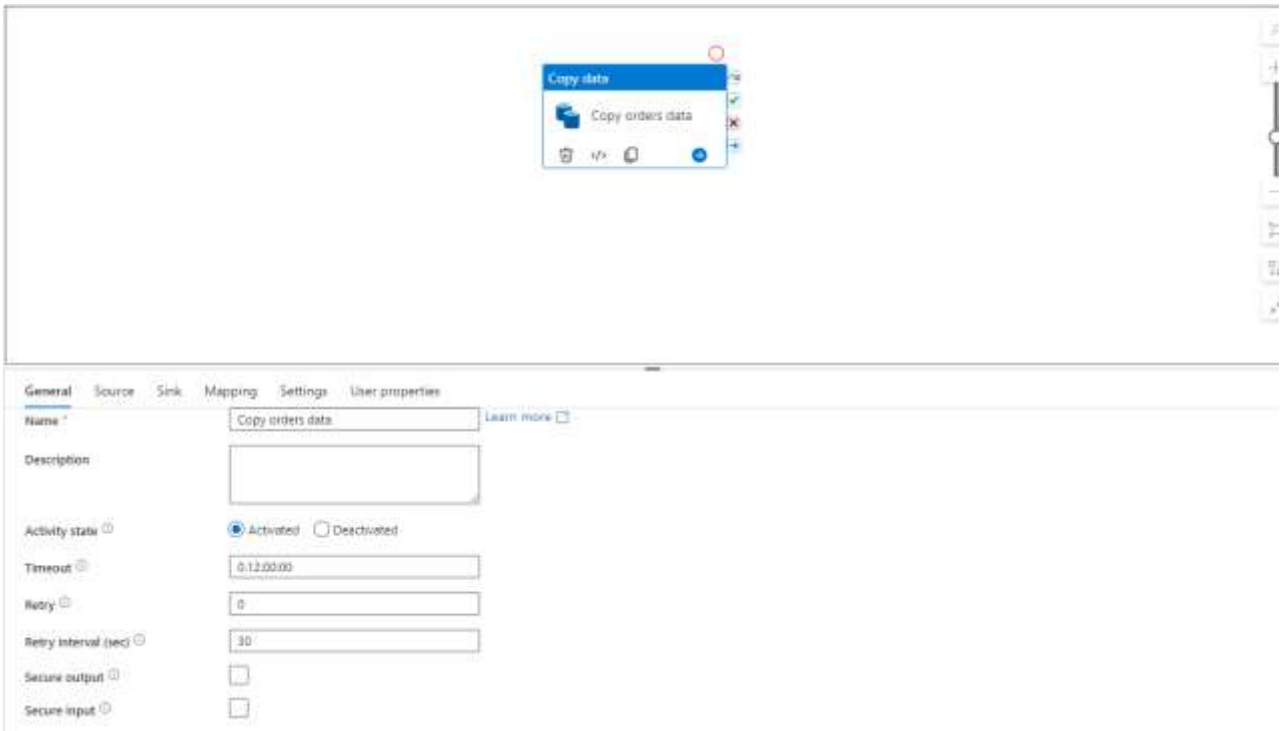


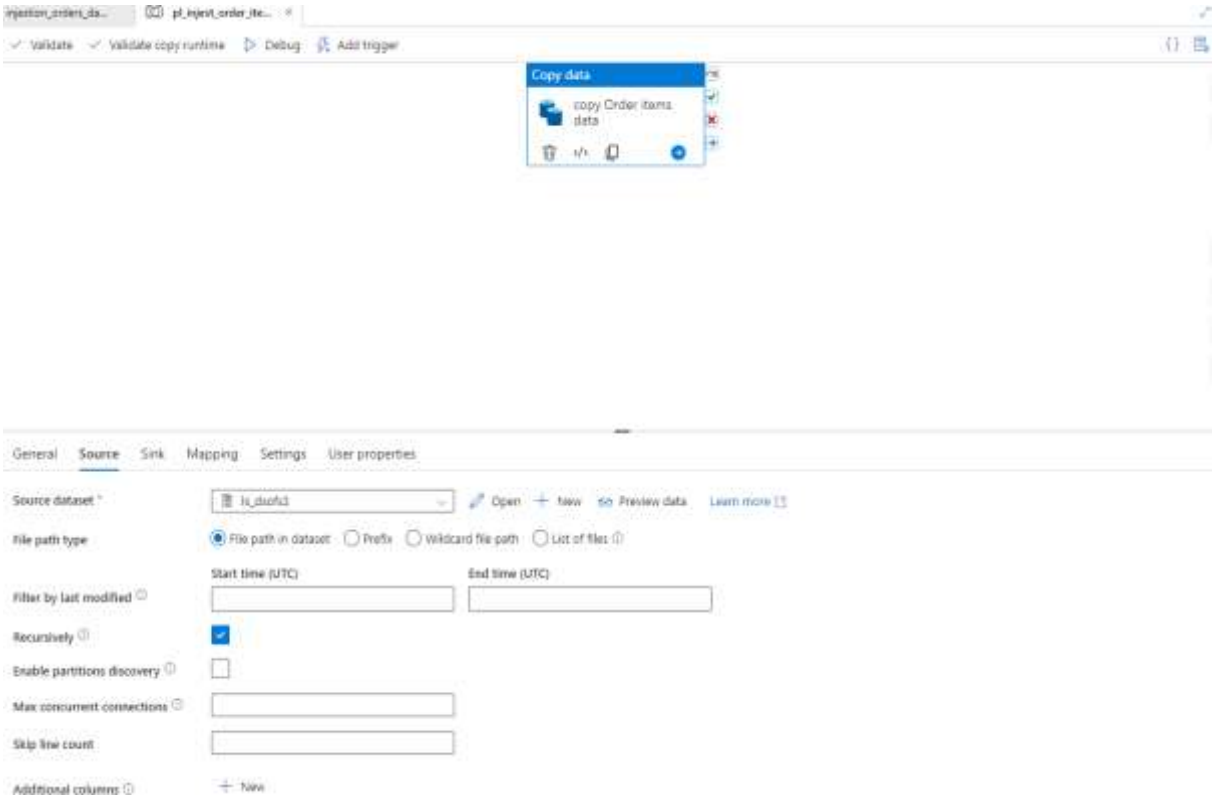
Step4: load the data set format



STEP4: Creating the Pipeline to ingest the data

Pipeline for injest the data from s3 to adls



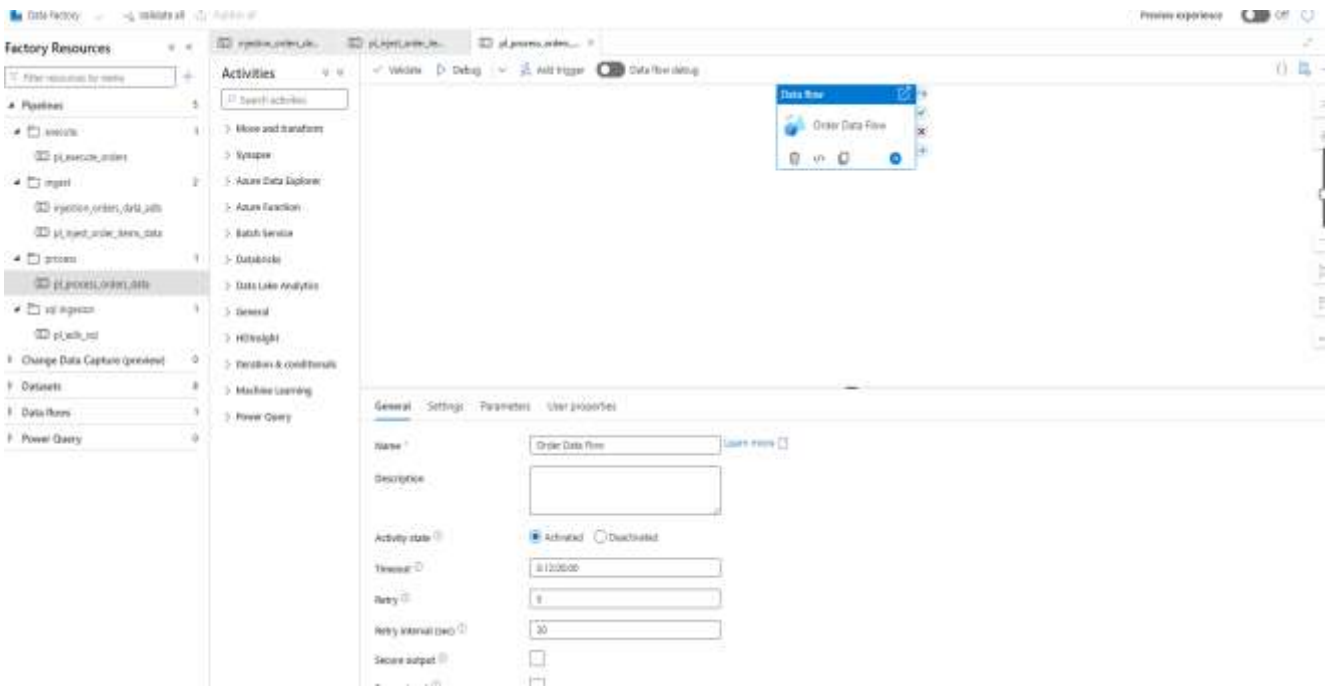


Pipeleline to injest data from blob storage to adls

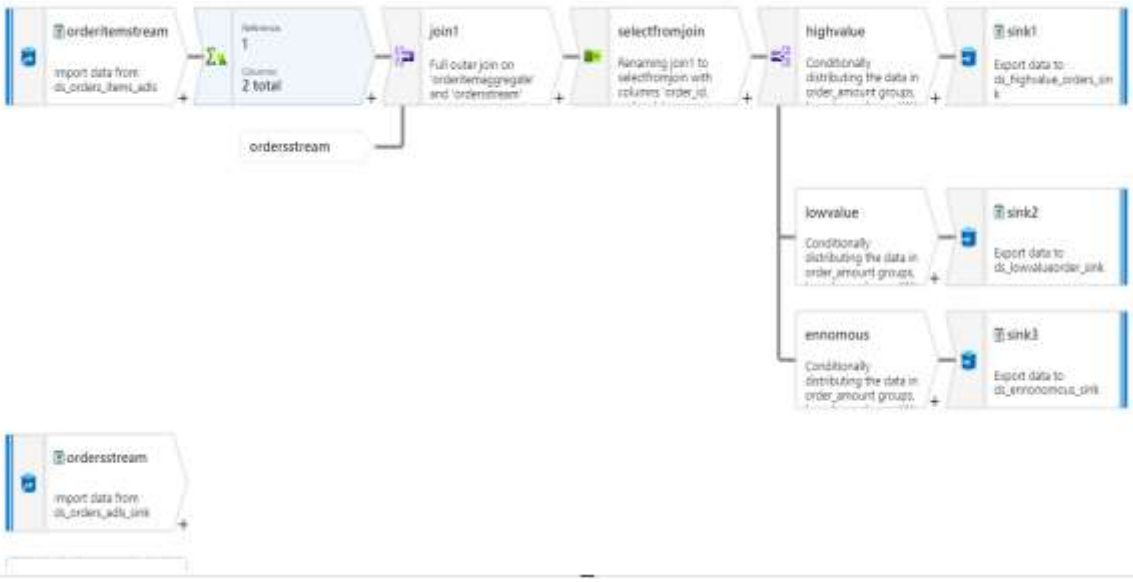
Pipeline to process orders data

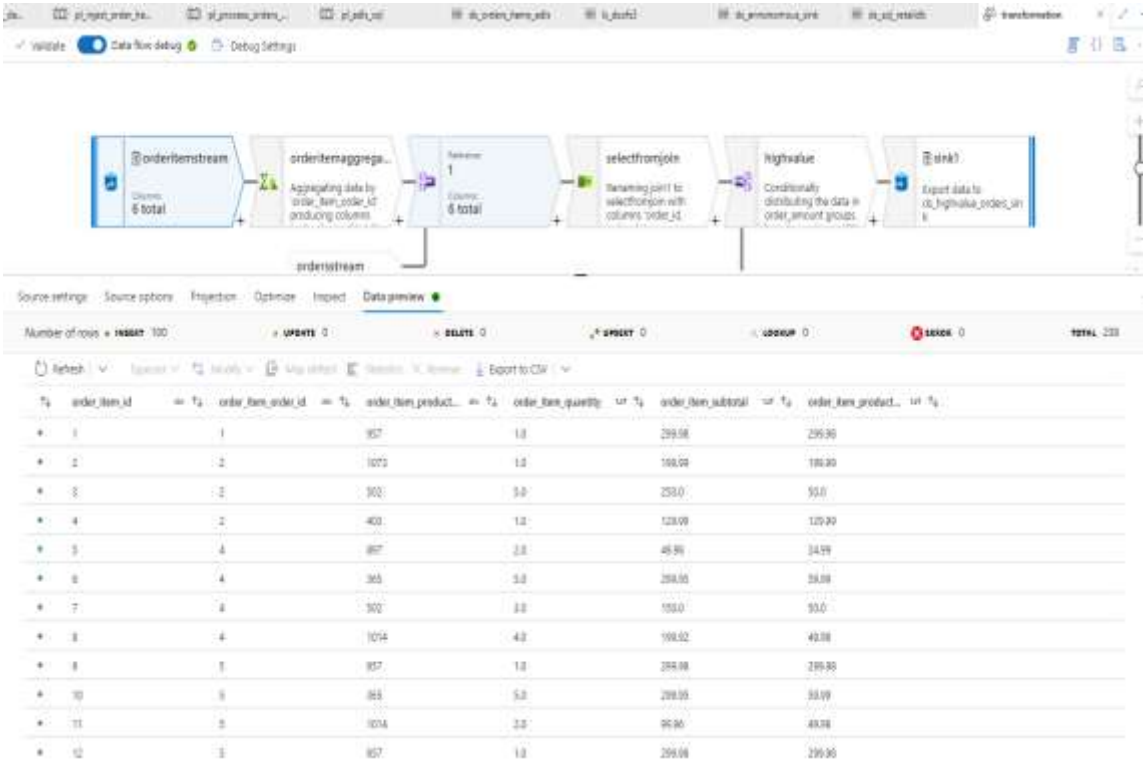
Step 5: Transform the data using data flow

Data Flow Diagram



Load the data from the adls





ggregate settings | Optimize | Inspect | Data preview

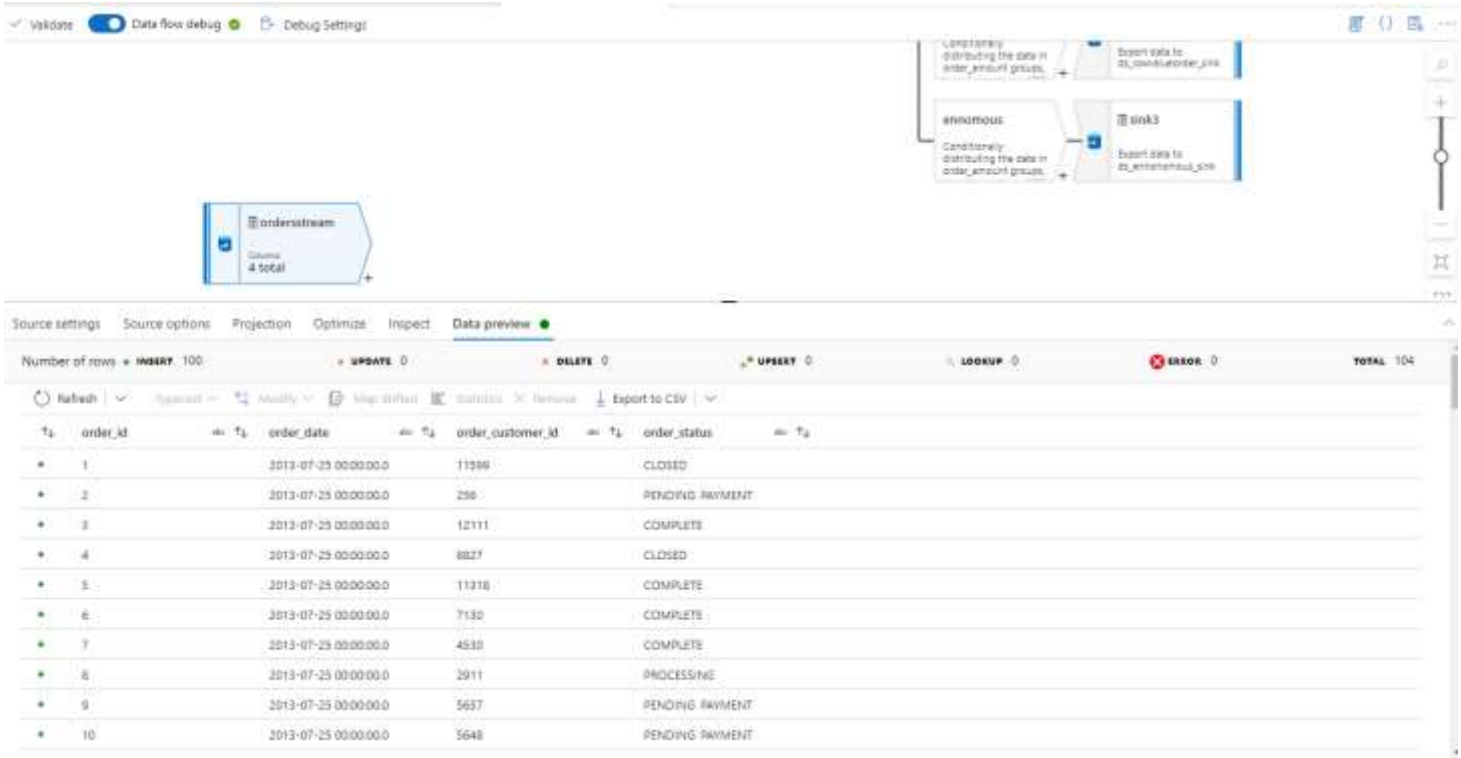
Number of rows: INSERT 83 | UPDATE 0 | DELETE 0 | UPSERT 0 | LOOKUP 0 | ERROR 0 | TOTAL

Refresh | Export to CSV

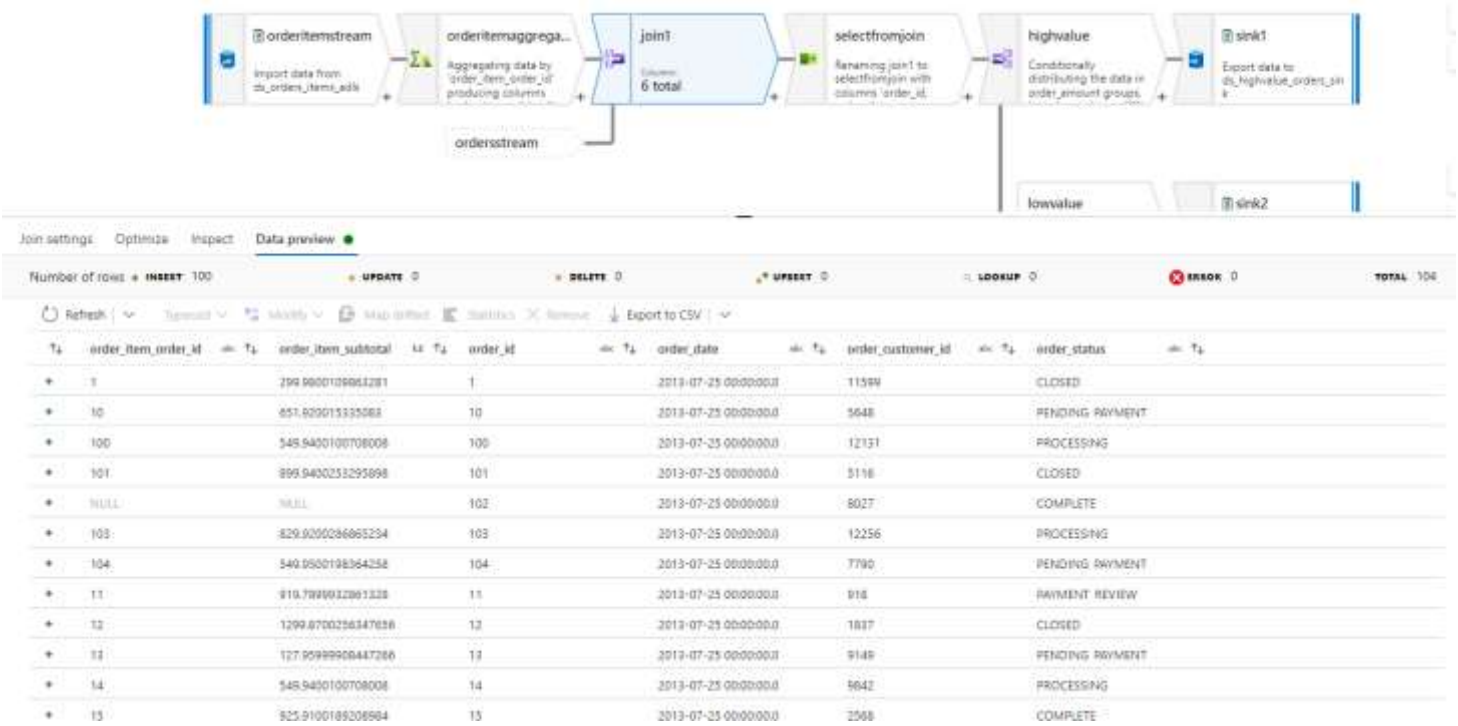
order_item_order_id	order_item_subtotal
1	299.9800109863281
2	579.9800109863281
4	699.8500099182129
5	1129.8600387573242
7	579.9200134277344
8	729.8400115966797
9	599.9600067138672

Aggregate the data by Order id for table order_items

Load the Orders table

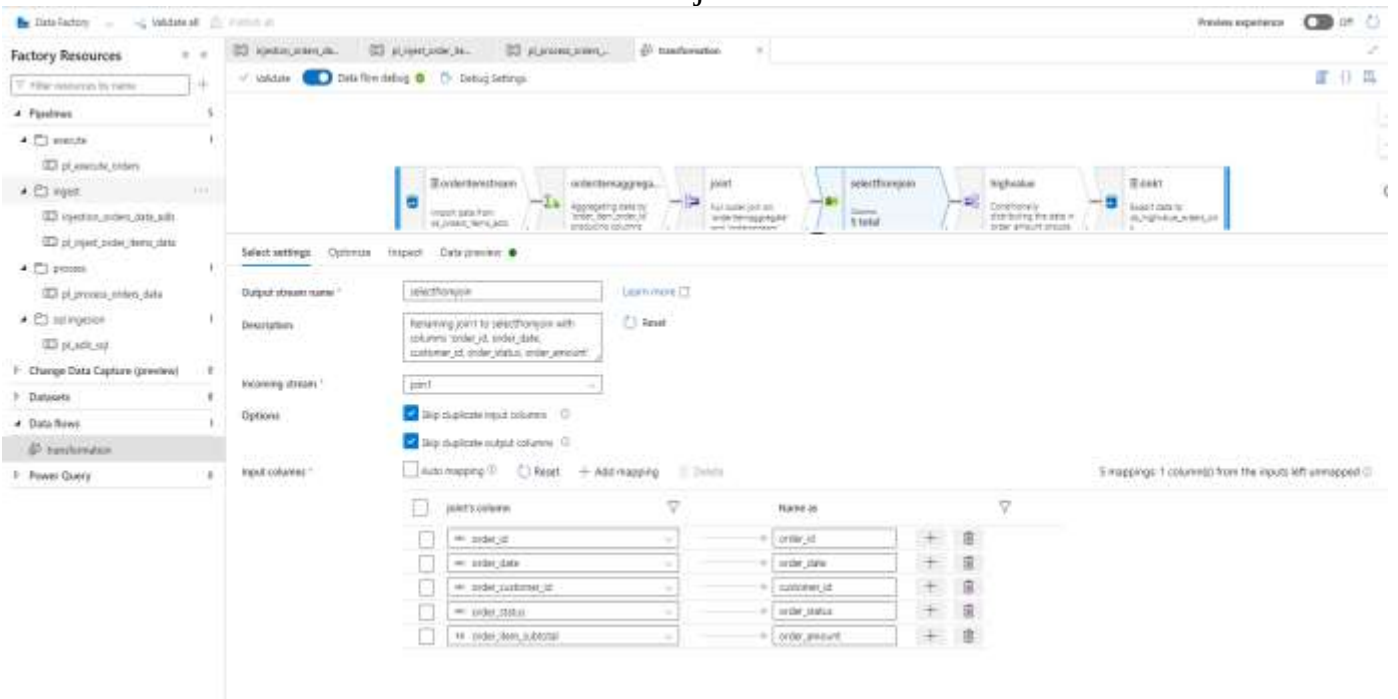


Join Orders and Order item table

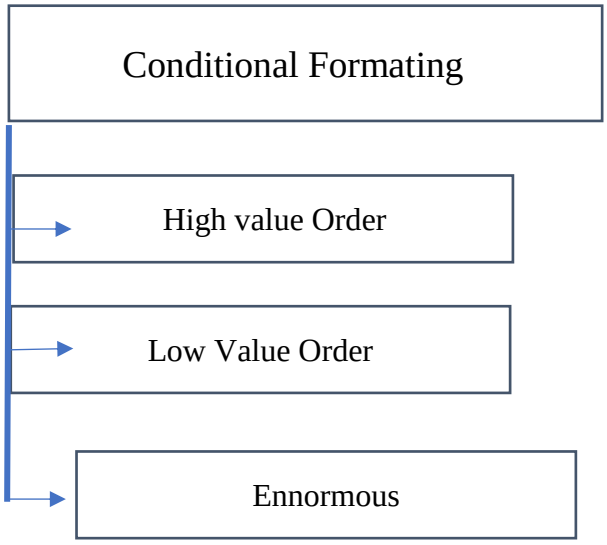


order_item_order_id	order_item_subtotal	order_id	order_date	order_customer_id	order_status
1	299.98001099853281	1	2013-07-25 00:00:00.0	11599	CLOSED
10	651.920015335083	10	2013-07-25 00:00:00.0	5648	PENDING_PAYMENT
100	549.9400100708008	100	2013-07-25 00:00:00.0	12131	PROCESSING
101	899.9400253295898	101	2013-07-25 00:00:00.0	5116	CLOSED
NULL	NULL	102	2013-07-25 00:00:00.0	8027	COMPLETE
103	829.9200286865234	103	2013-07-25 00:00:00.0	12256	PROCESSING
104	549.9500198364258	104	2013-07-25 00:00:00.0	7790	PENDING_PAYMENT
11	919.7999932861328	11	2013-07-25 00:00:00.0	918	PAYMENT_REVIEW
12	1299.8700256347656	12	2013-07-25 00:00:00.0	1837	CLOSED
13	127.99999904447266	13	2013-07-25 00:00:00.0	9149	PENDING_PAYMENT
14	549.9400100708008	14	2013-07-25 00:00:00.0	9642	PROCESSING
15	929.9100189208984	15	2013-07-25 00:00:00.0	2568	COMPLETE

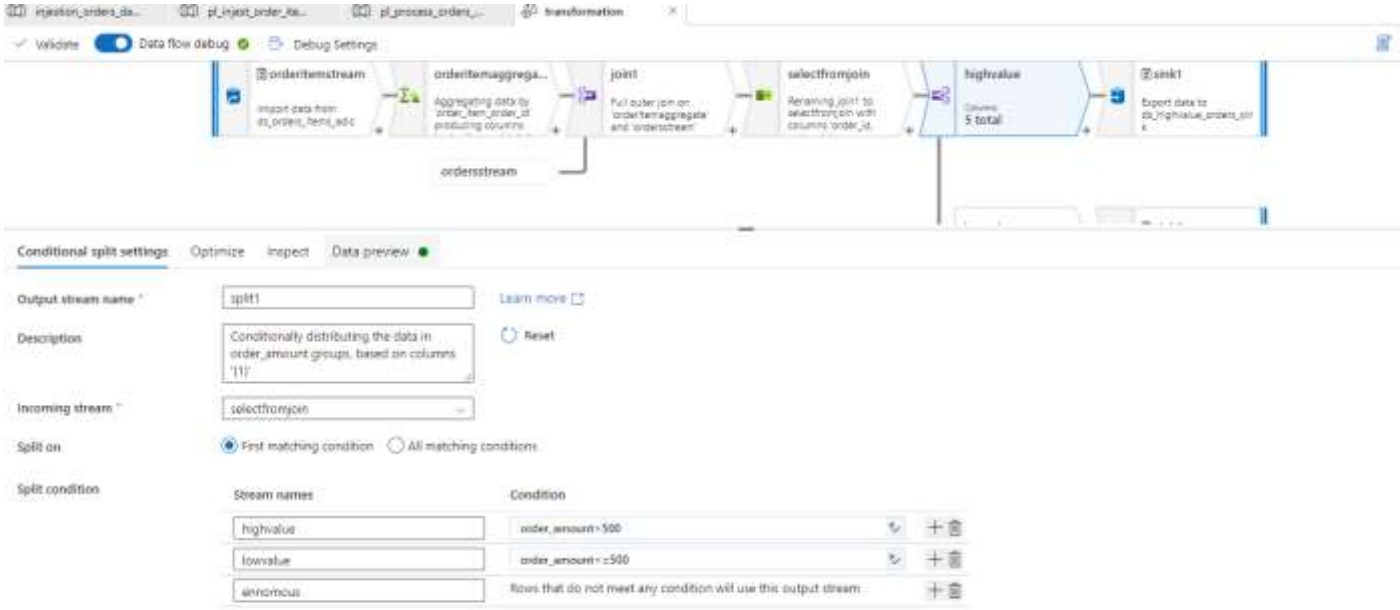
Select the desired attributes and data from the join



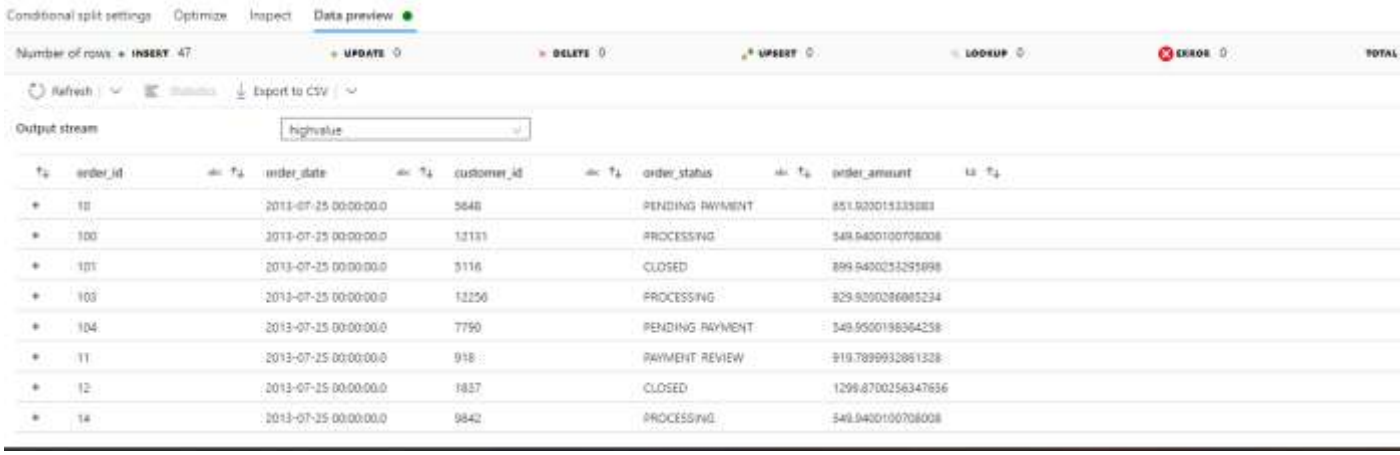
Conditional Formating



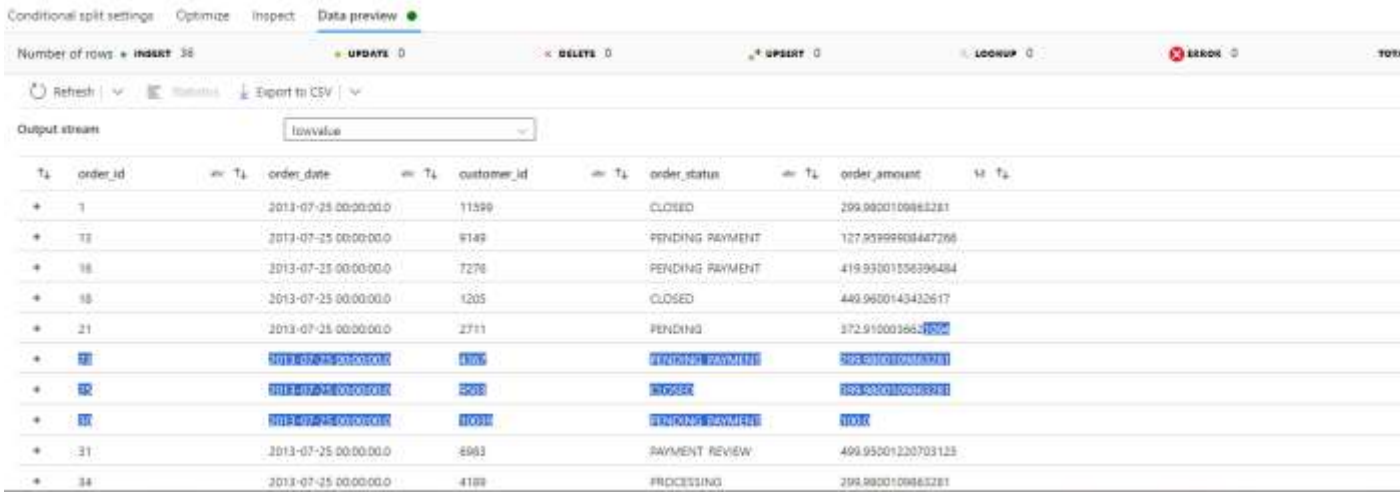
Screen shot of the Conditions



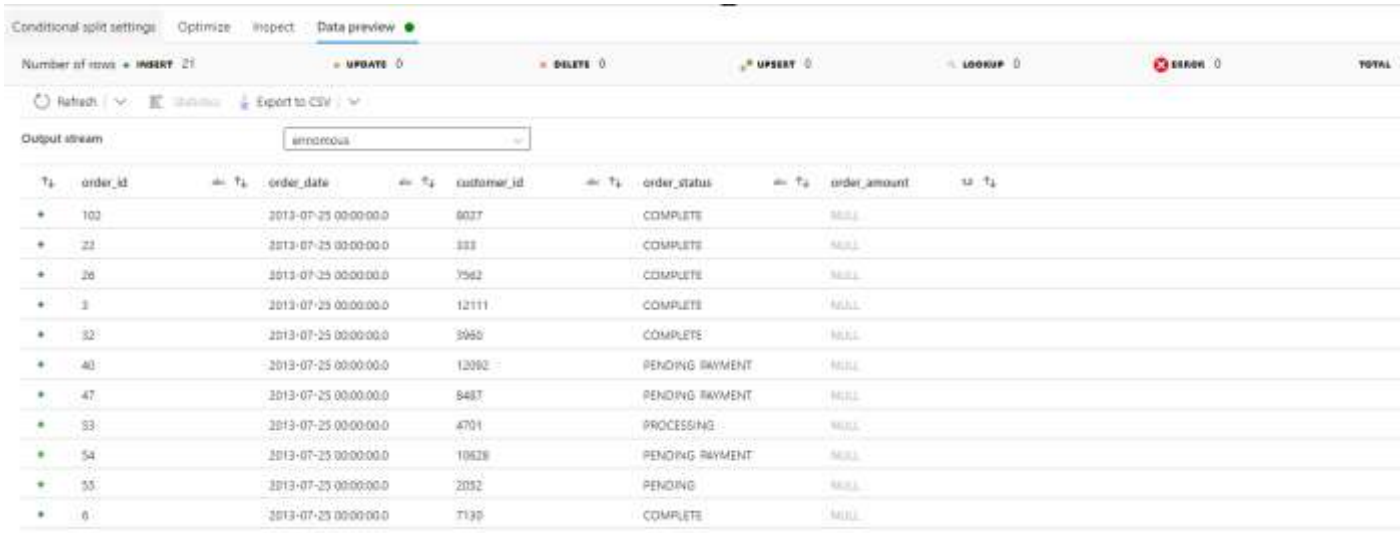
High value orders



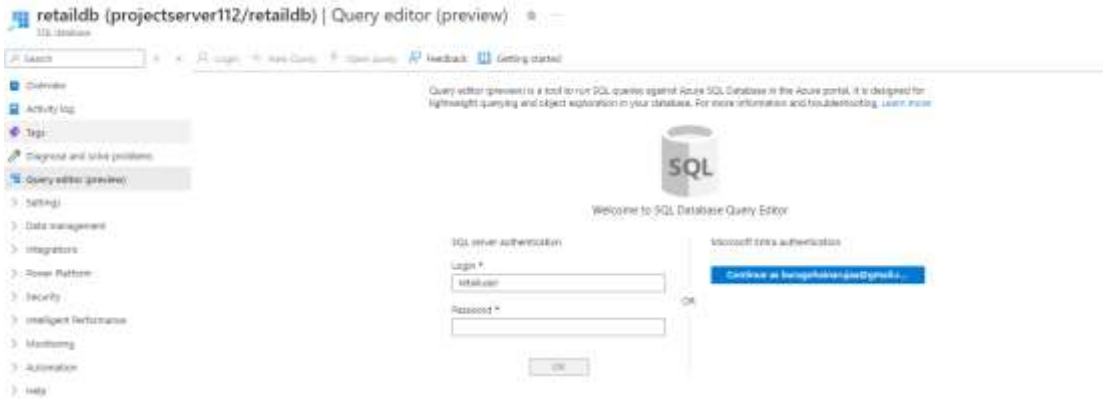
Low value orders



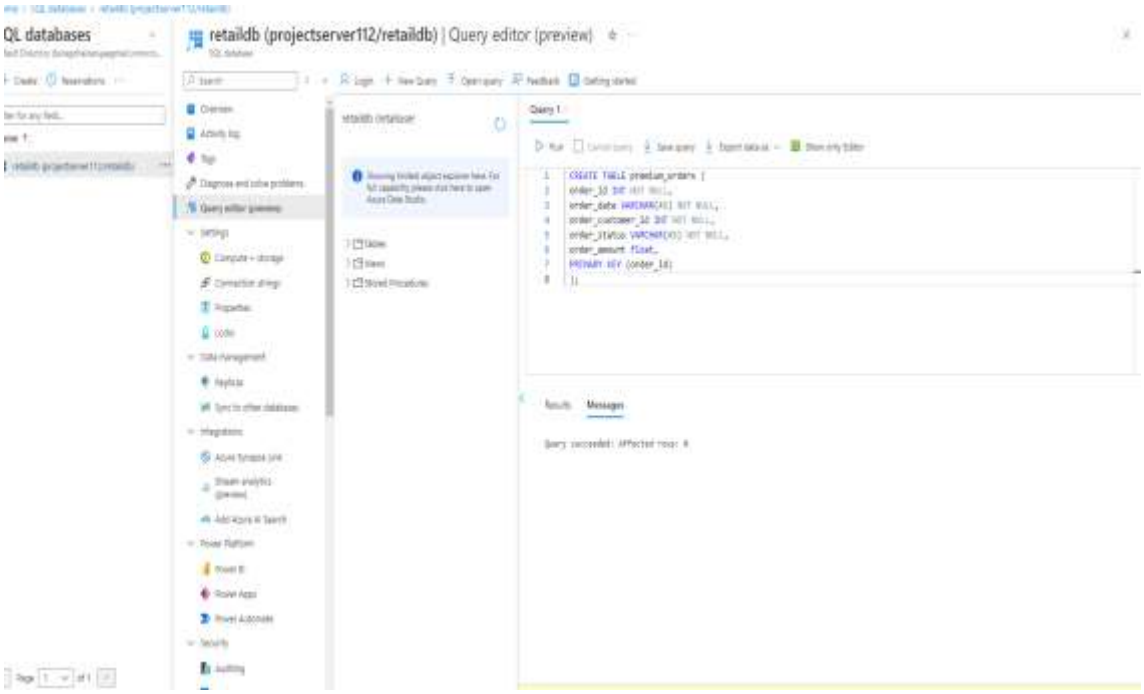
Ennormous



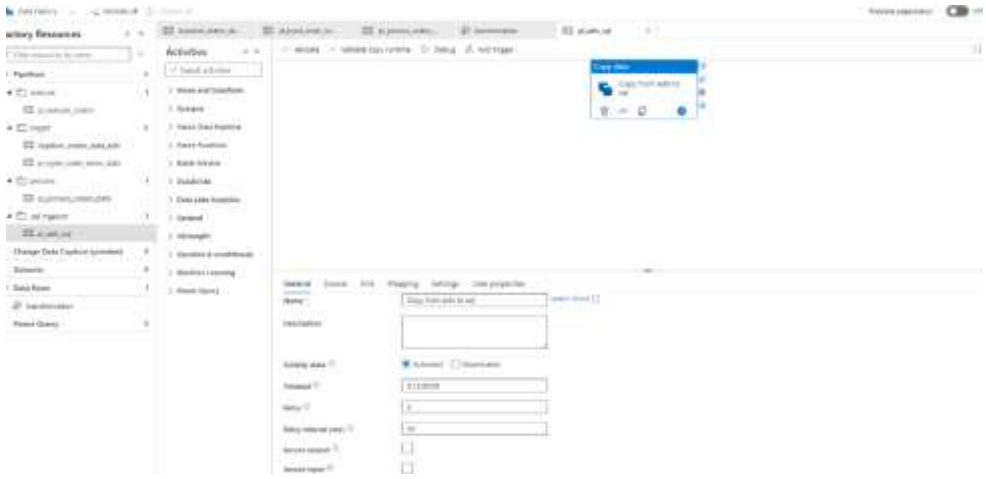
5.4 OPEN SQL SERVER



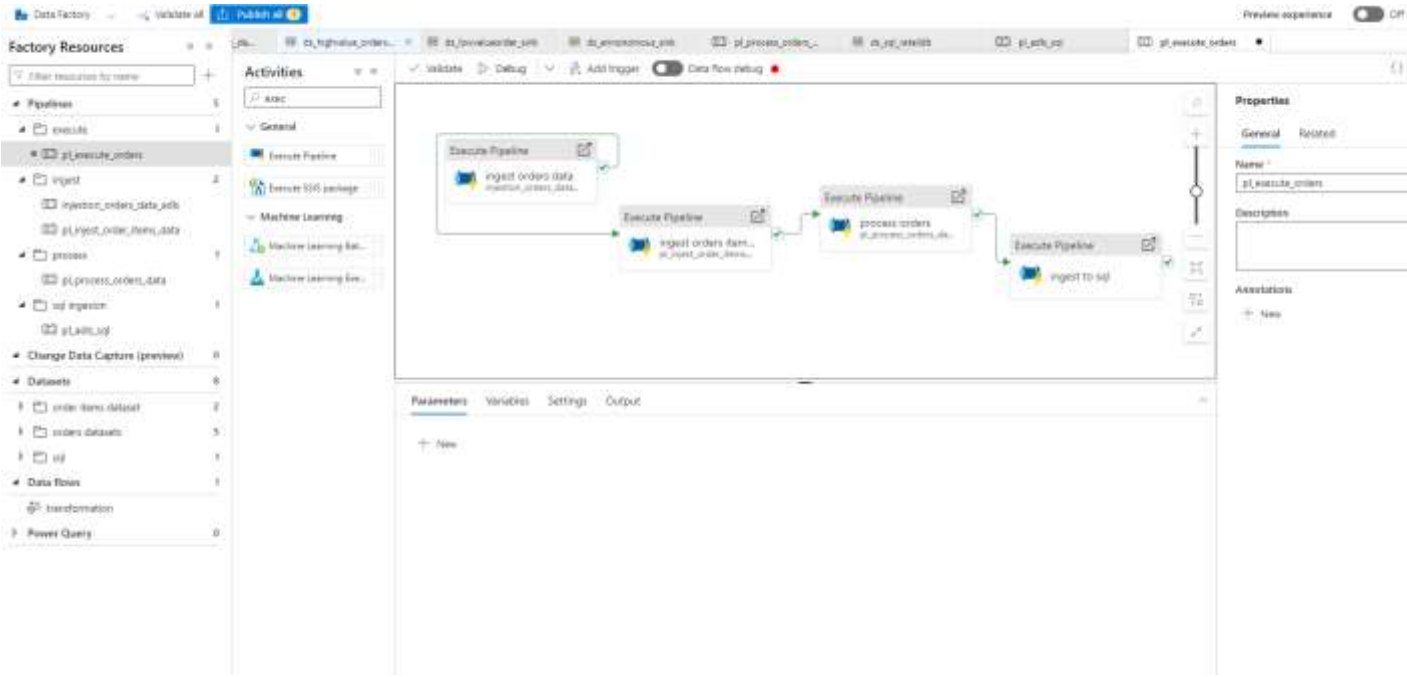
Creation of Premium table in Sql


QL databases
retaildb (projectserver112/retaildb) | Query editor (preview)
Query 1
CREATE TABLE preedam_orders (
order_id INT NOT NULL,
order_date VARCHAR(40) NOT NULL,
order_customer_id INT NOT NULL,
order_status VARCHAR(45) NOT NULL,
order_amount float,
PRIMARY KEY (order_id)
)
Results
Messages
Query succeeded: Affected rows: 0

Creating the Pipeline for Execution



Creating the Whole pipeline to execute the data



Add Triggers for the Pipeline