A MINI PROJECT

PYSPARK AND AZURE
HOCKEY ANALYSIS PROJECT REPORT

DOCUMENTATION BY

IYSHWARYA RAMESH

RAMITHA R M

SANJANA B L

TABLE OF CONTENTS

1. INTRODUCTION

The purpose of this mini-project is to design, develop, and implement a comprehensive data analysis and reporting system aimed at delivering meaningful insights into hockey game statistics using PySpark. This project focuses on extracting operational data from multiple flat file sources, performing systematic data transformation and cleaning processes, and structuring it into an optimized dimensional data warehouse model following a star schema approach.

Through this framework, the system facilitates in-depth analysis of player and team performances across various leagues, seasons, and match types. By leveraging the power of distributed data processing with PySpark, the project efficiently handles large-scale datasets, performs complex transformations, and generates insightful reports that support strategic decision-making for coaches, analysts, and sports management teams.

The project not only enhances data handling and reporting capabilities but also serves as a practical exercise in applying real-world data engineering techniques using PySpark in a structured ETL (Extract, Transform, Load) pipeline, culminating in the generation of comprehensive reports in CSV format for further visualization and interpretation.

2. PROBLEM STATEMENT

2.1 OBJECTIVE

To develop an Analysis & Reporting System for Hockey Games using PySpark within an Azure-supported environment. The system aims to manage, analyze, and report operational hockey game data — covering player performance statistics, club records, and match outcomes — by transforming raw operational data from multiple flat files into structured, analytical reports that support effective decision-making.

2.2 ABSTRACT OF THE PROJECT

This project involves building an end-to-end data processing and reporting pipeline using PySpark to handle diverse operational hockey data, including player profiles, team performances, league records, and match results. The raw data, collected from heterogeneous flat file sources, undergoes a systematic Extract, Load, and Transform (ELT) process to convert it into a well-defined dimensional data warehouse model.

A star schema is designed, comprising various dimension tables (such as Player, Club, Country, League, Match Type, and Time) and fact tables for capturing key performance metrics. The transformed data is then used to generate a set of insightful, business-relevant reports, including top goal scorers, maximum appearances, disciplinary records, and club performance rankings.

The system facilitates efficient, scalable processing of large datasets and demonstrates practical application of PySpark's distributed computing capabilities in building a

real-time sports analytics reporting solution.

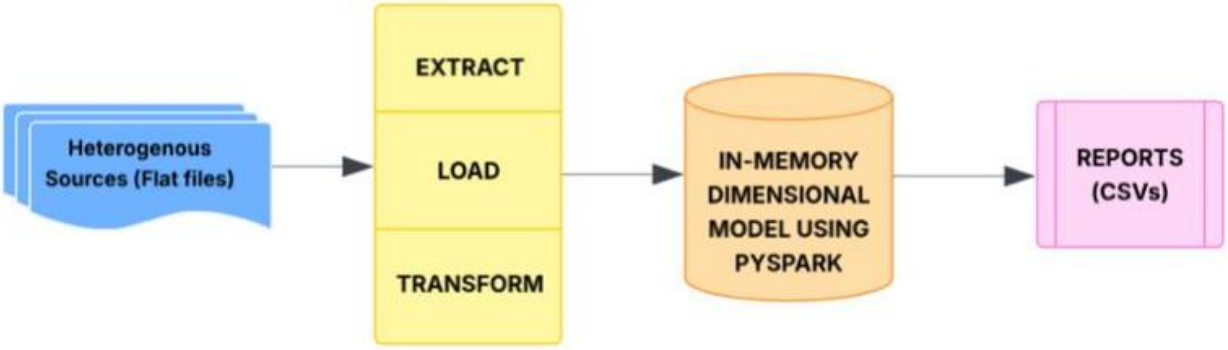
2.3 CSV FILES USED

Hockey_Src1.csv

Hockey_Src2.csv

DIM.Date.Table.csv

3.IMPLEMENTATION



3.1 WORK OVERVIEW:

1. Data Collection & Loading
2. Data Cleaning & Preprocessing
3. Data Analysis
4. Reporting & Visualization
5. Final Summary

The source files are:

SOURCE FILE 1: PLAYER STATISTICS

- Year
- Match
- Player Name
- Club Name
- Country Name
- Appearances, Goals scored, Goals assist, Shots, Fouls, Cards, Salary, etc.

SOURCE FILE 2: TEAM STATISTICS

- Year
- Match Name
- League Name
- Club Name
- Country Name
- Appearances, Wins, Losses, Drawn, Clean Sheets, Net Worth
- Date Dimension.

3.2 PROJECT FLOW & DATA WAREHOUSE SCHEMA

PROJECT WORKFLOW:

- Operational data from heterogeneous sources (CSV files)
- Extraction, Loading, and Transformation using PySpark
- In-memory dimensional model creation
- Report generation in CSV format

DATA WAREHOUSE MODEL:

DIMENSION TABLES

- **League:**

  League_ID

  League_Name.

- **Club:**

  Club_ID

  Club_Name

  League_ID

  Manager

  Owner

❖ **Country:**

Country_ID

Country_Name

Coach

❖ **Match Type:**

Match_Type_ID

Match_Name

❖ **Player:**

Player_ID

Player_Name

Club_ID

Country_ID

Position

Jersey No

DOB

Nationality

❖ **Time:**

Time_ID

Year

Quarter

Month

FACT TABLES

Fact_Player_Statistics:

- Fact_ID
- Time_ID
- Match_Type_ID
- Player_ID

- Club_ID
- Appearances
- Goals
- Shots
- Fouls
- Salary, etc.

Fact_Team_Statistics:

- Fact_ID
- Time_ID
- League_ID
- Club_ID
- Wins
- Losses
- Drawn
- Clean Sheets
- Net Worth, etc.

3.3 DATA TRANSFORMATION PROCESS

STEPS:

- ❖ Load source CSV files into PySpark DataFrames
- ❖ Clean and preprocess the data (null handling, trimming, datatype conversion)
- ❖ Create unique IDs for each dimension table using ranking functions
- ❖ Populate Dimension tables first
- ❖ Generate Fact tables by joining respective dimension IDs
- ❖ Create analytical queries to produce the required reports
- ❖ Export reports as CSV files

STEPS PERFORMED:

STEP 1: AZURE RESOURCE SETUP

- Create Azure Storage Account
- Create Containers

STEP 2: UPLOAD FILES TO BLOB STORAGE

STEP 3: INGEST SOURCE FILES USING AZURE DATA FACTORY (ADF)

- Create Azure Data factory
- Open ADF Studio
- Create Linked services
- Create datasets
- Create a pipeline to copy data
- Run the pipeline

STEP 4: CREATE AZURE SQL DATABASE

- Create SQL Database on Azure
- Configure firewall
- Connect to SQL database
- Create table structure

STEP 5: REGISTERED AN APP MOUNTED THE AZURE DATA LAKE STORAGE (ADLS) CONTAINER TO AZURE DATABRICKS

STEP 6: CREATE AZURE DATABRICKS WORKSPACE

STEP 7: TRANFORMATION IN DATABRICKS

- Read CSVs
- Create dimensions
- Create fact tables
- Write transformed tables to output

STEP 8: LOAD TO SQL USING AZURE DATA FCATORY (ADF)

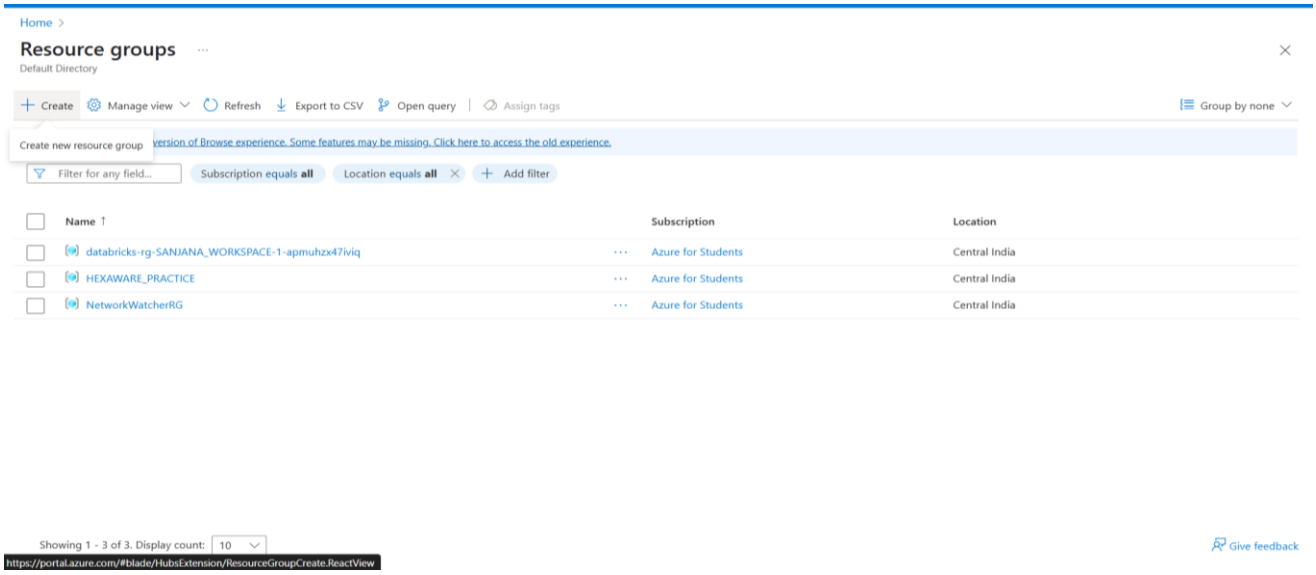
- Create linked services
- Create datasets (Input + Output)
- Build the copy data pipeline
- Run the pipeline
- Verify in Azure SQL

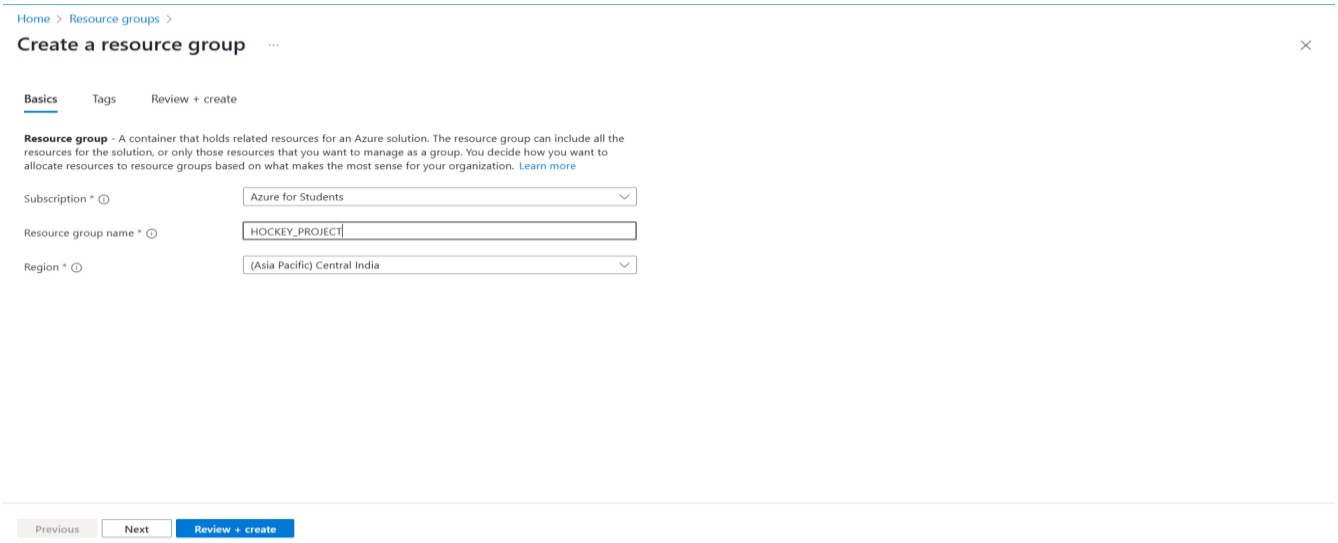
STEP 9: SQL REPORTING QUERIES

STEP 10: CSV OUTPUT

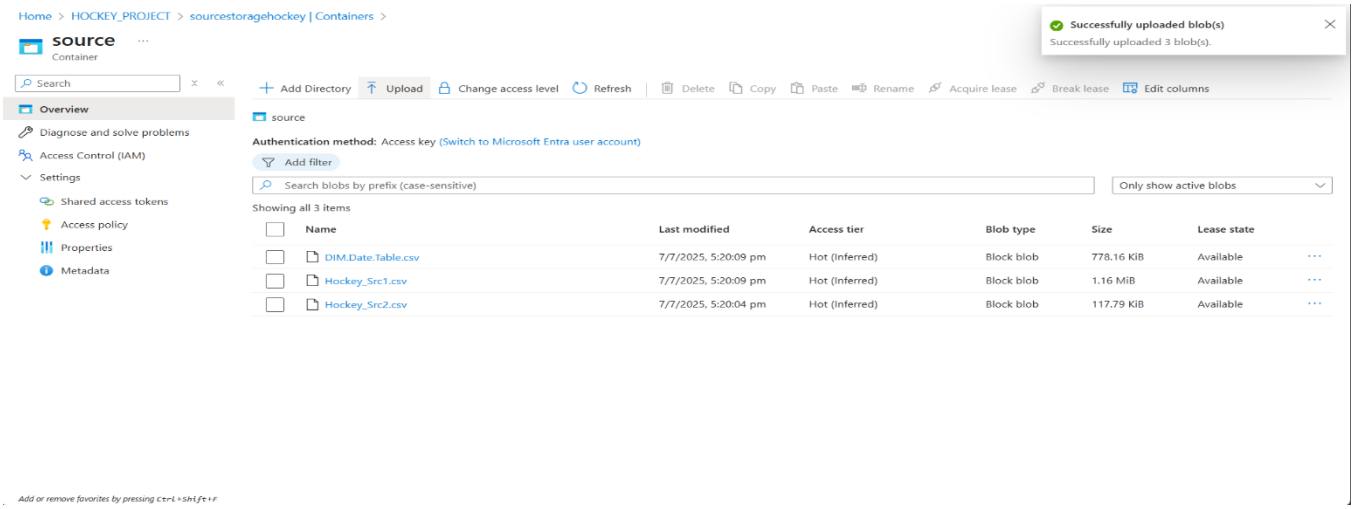
SCREENSHOTS

STEP 1: AZURE RESOURCE SETUP



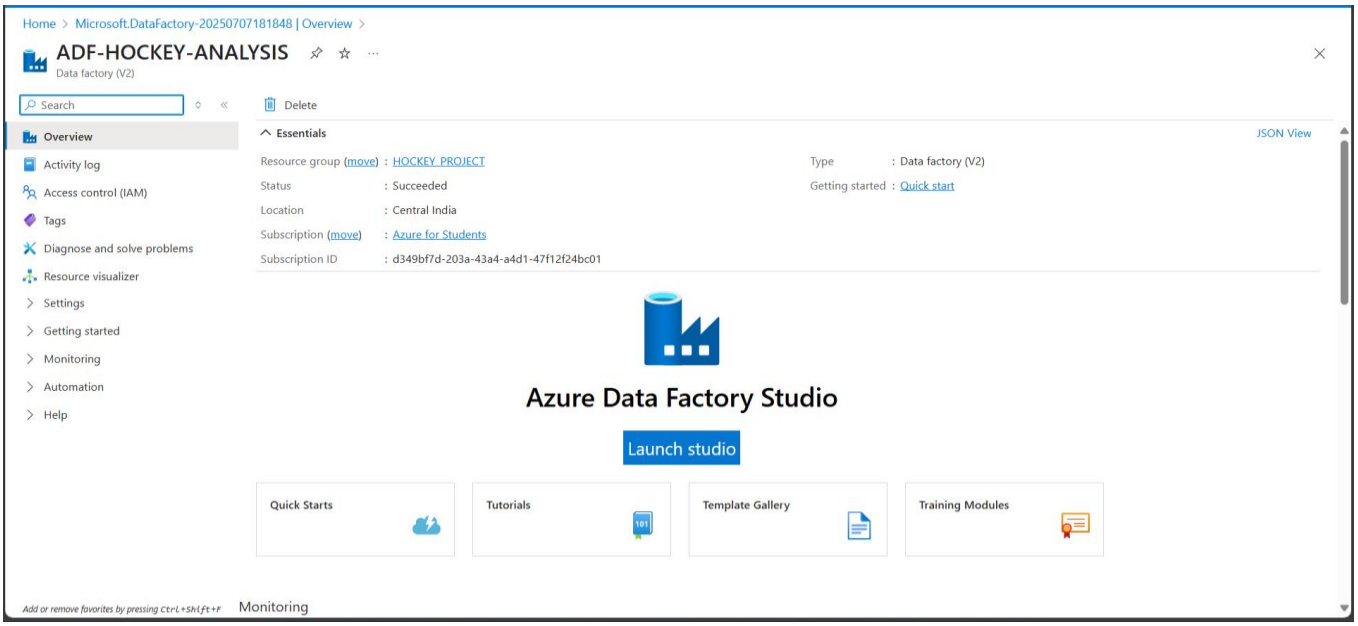


STEP 2: UPLOAD FILES TO BLOB STORAGE

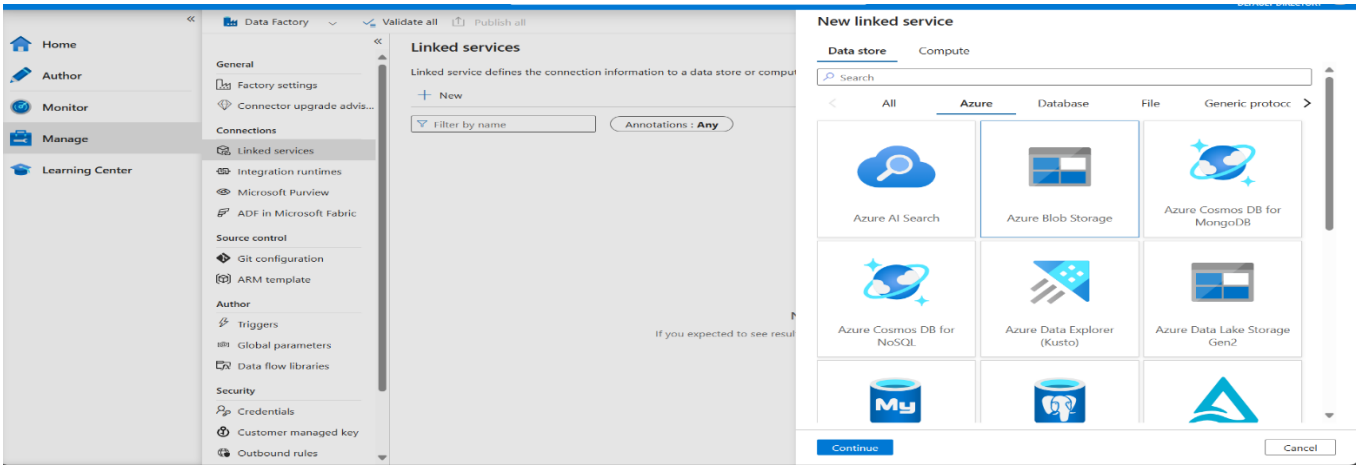


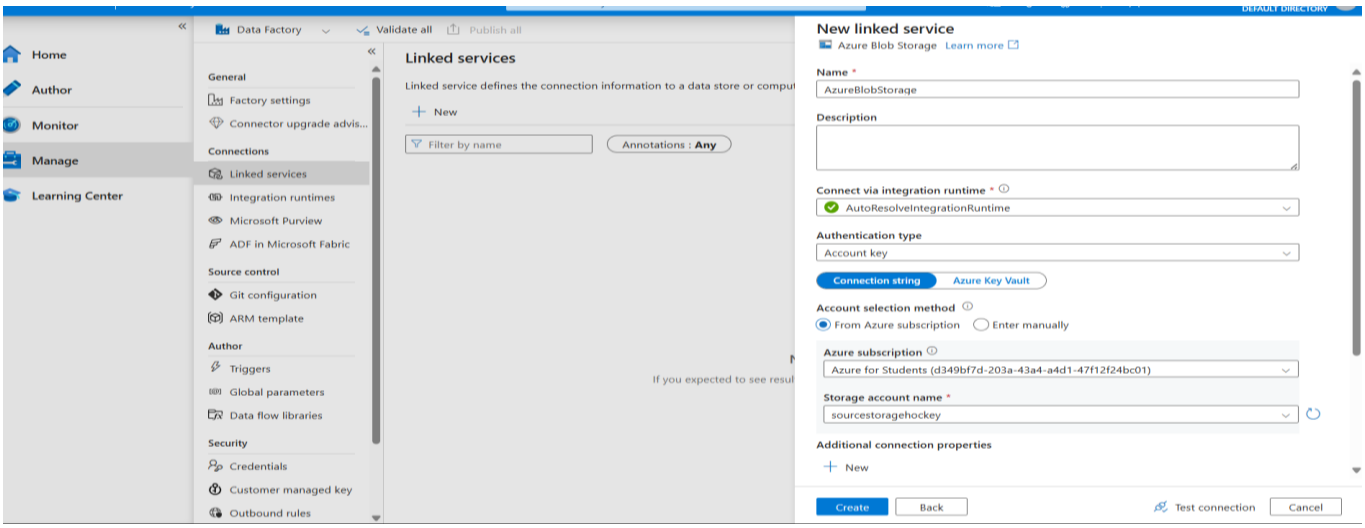
STEP 3: INGEST SOURCE FILES USING AZURE DATA FACTORY (ADF)

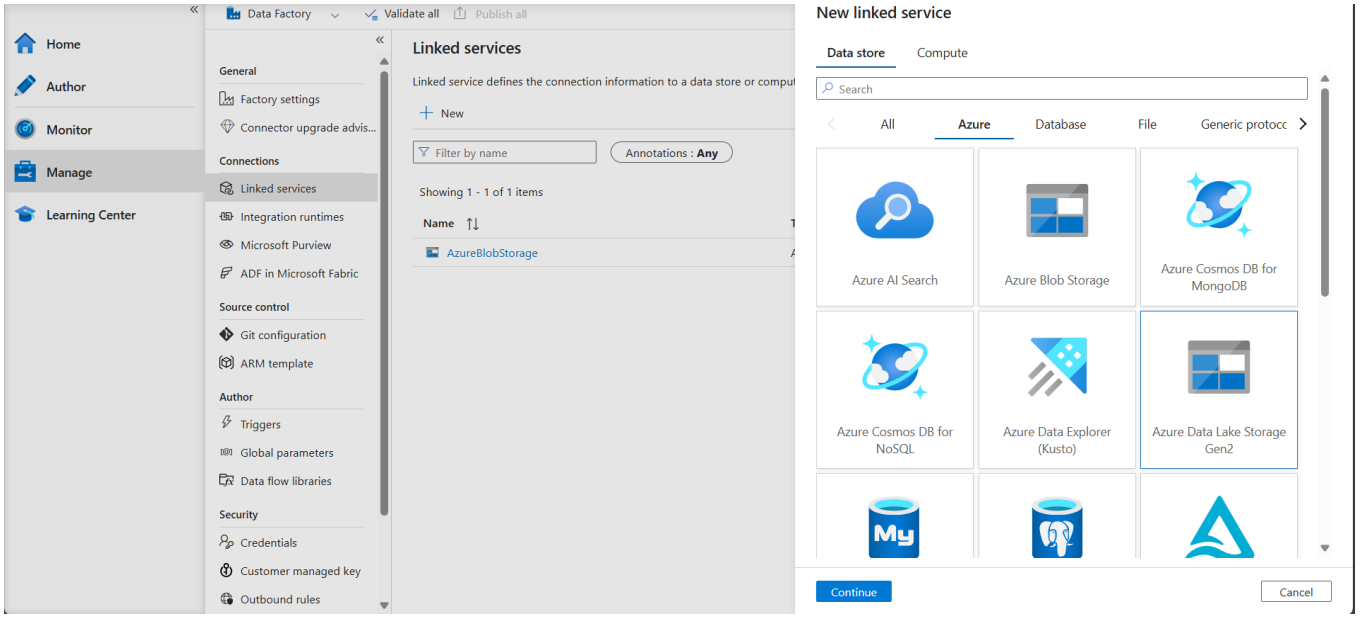
Create Azure Data factory

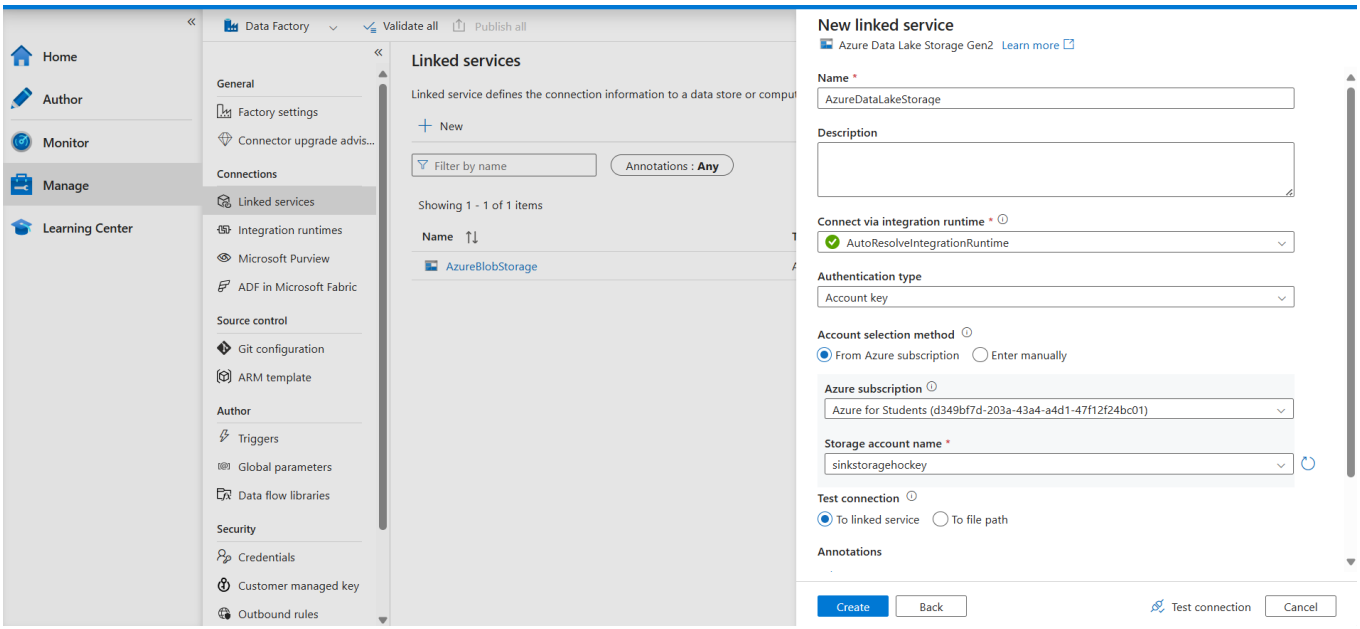


Create Linked Services



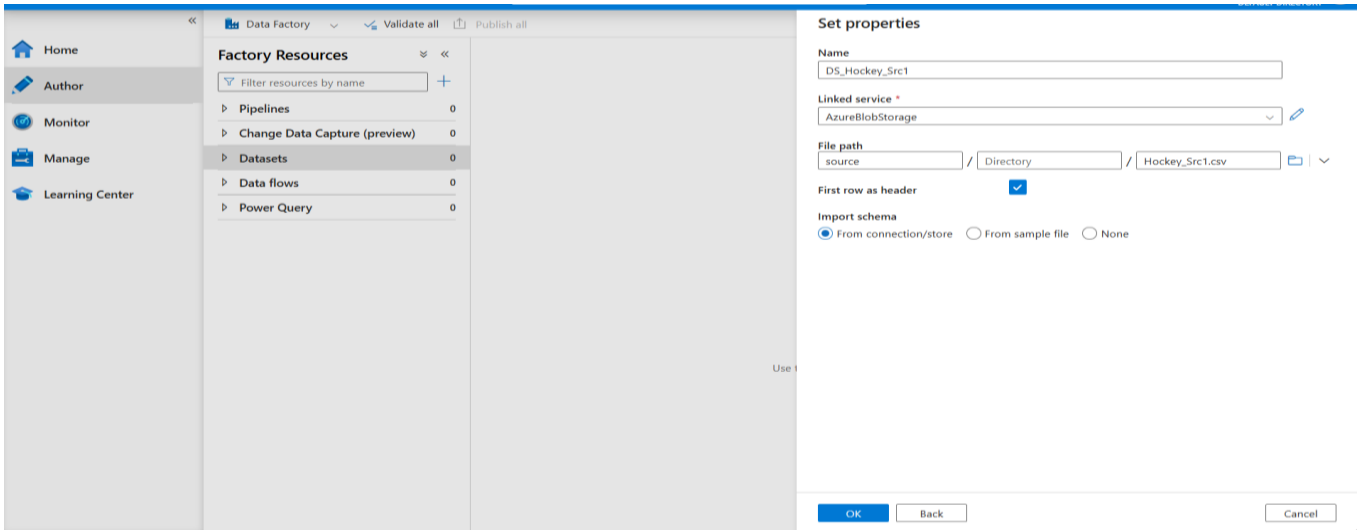






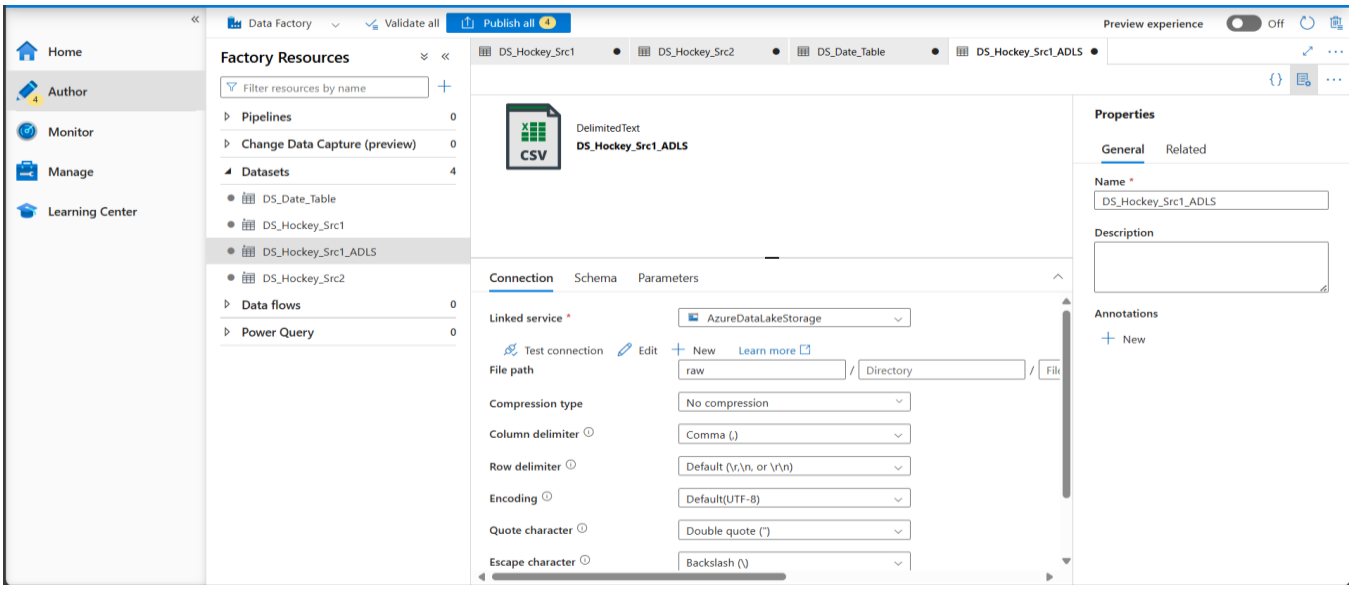
Create datasets

DS_Hockey_Src1 (Similarly For DS_Hockey_Src2, DS_Date_Table)

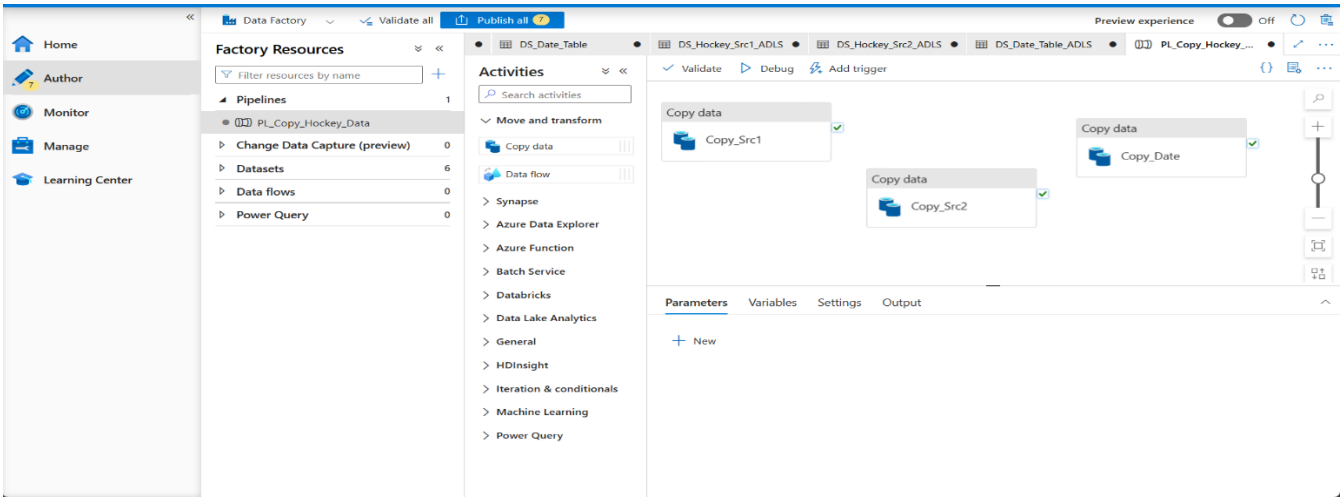


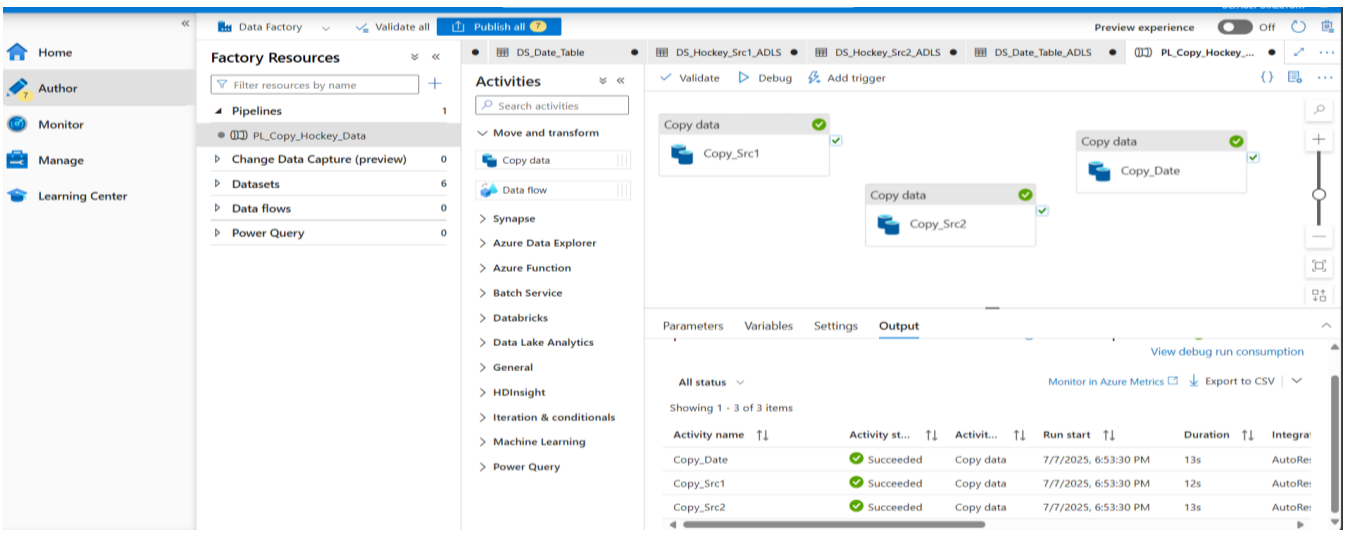
3 Destination Datasets (For Raw Container)

DS_Hockey_Src1_ADLS (Similarly For DS_Hockey_Src2_ADLS, DS_Date_Table_ADLS)

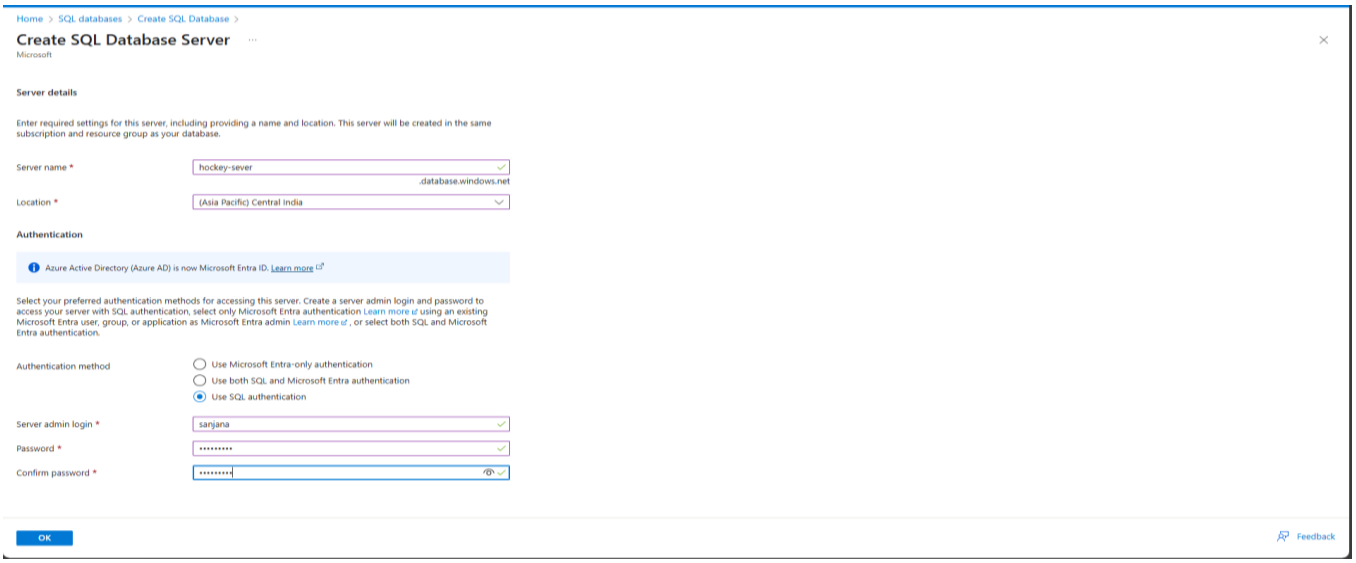


Create and run the Pipeline to Copy Data

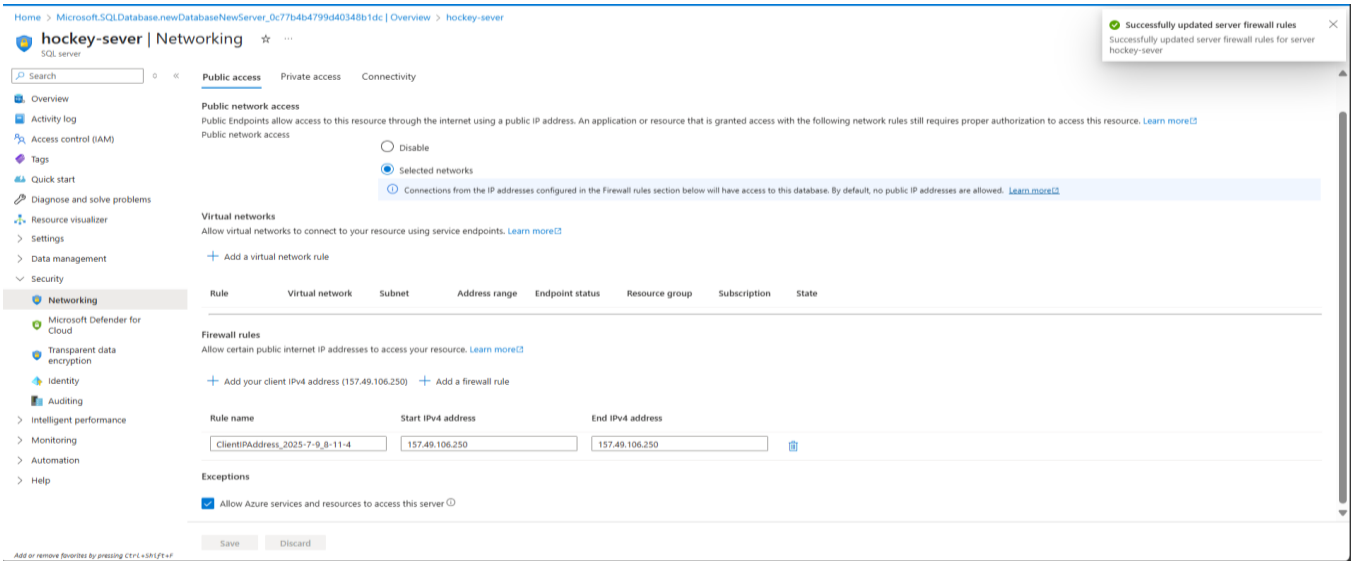




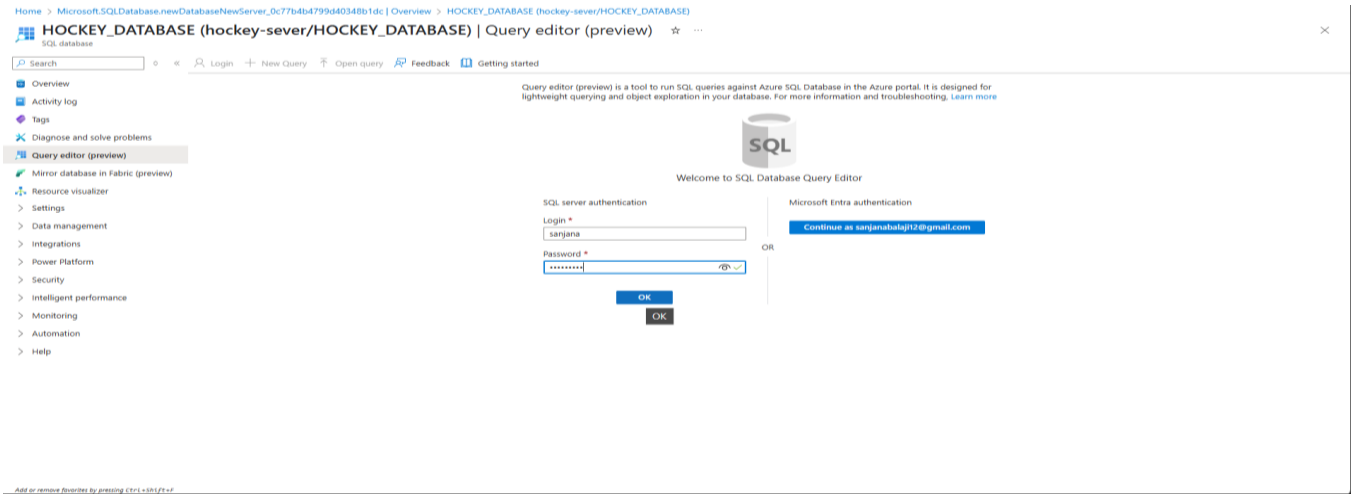
STEP 4: CREATE AZURE SQL DATABASE



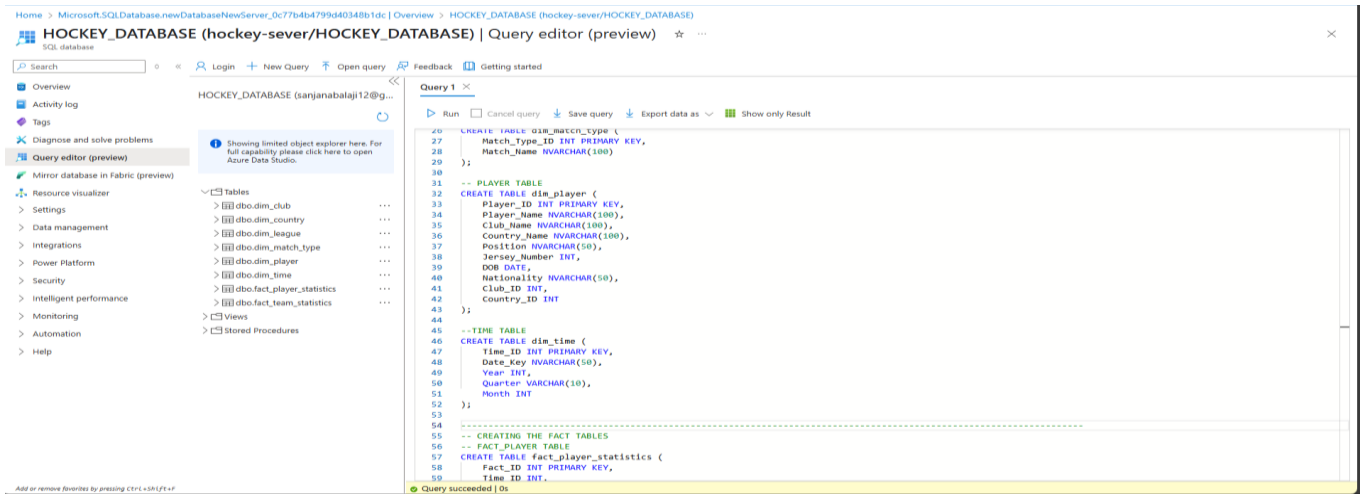
Configure firewall



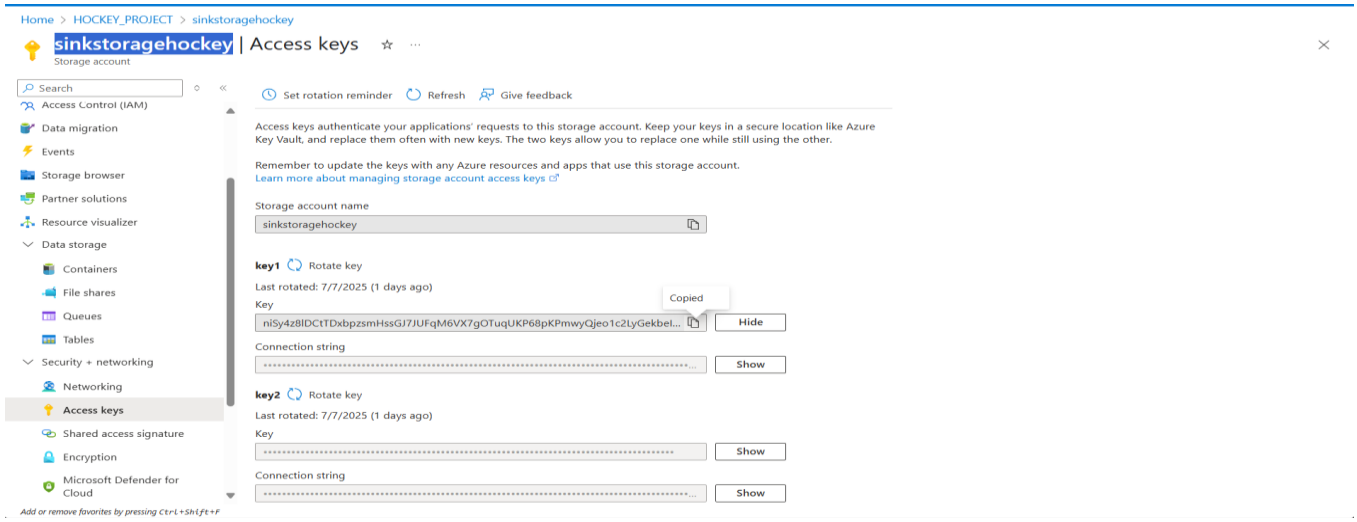
Connect to SQL Database



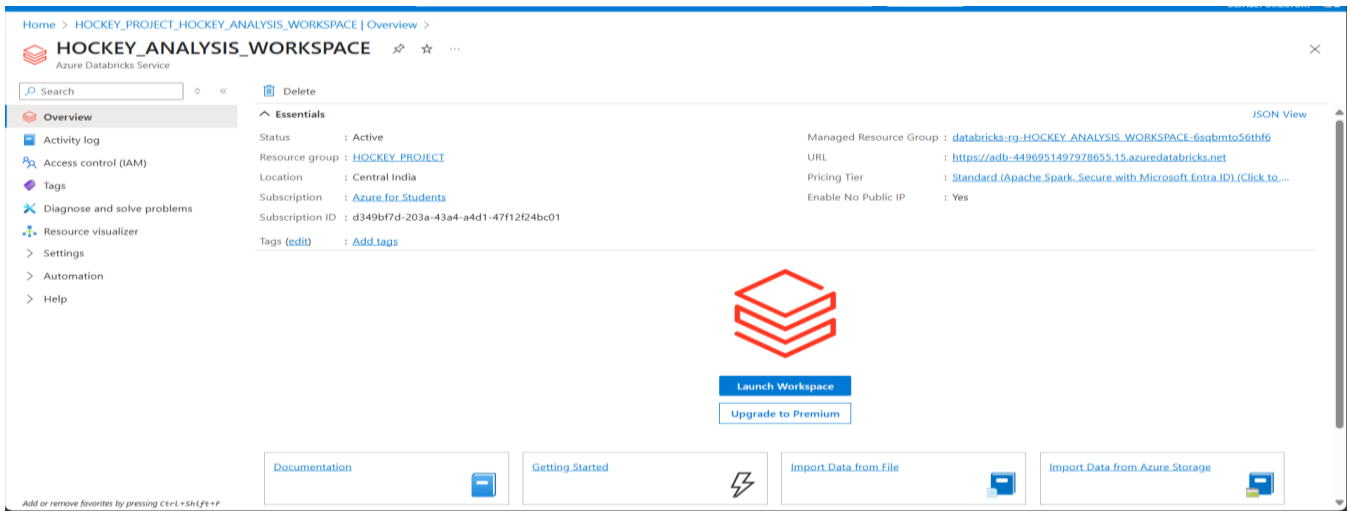
Create table structures

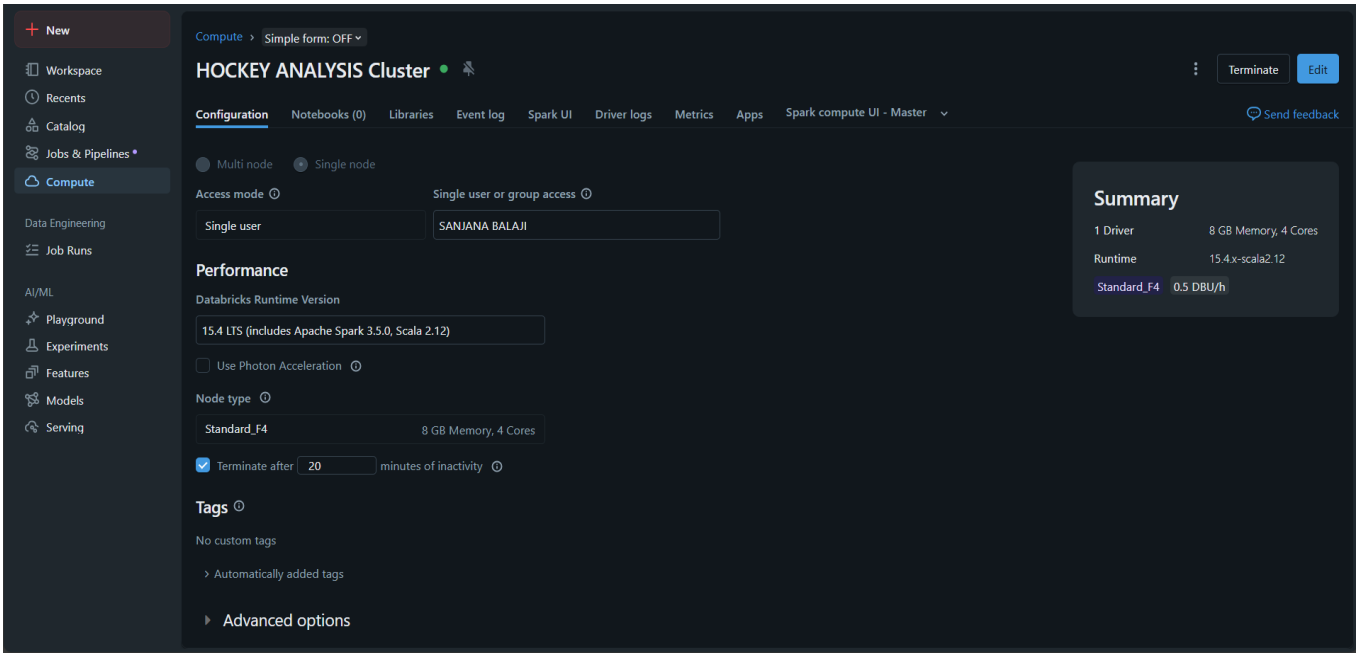


STEP 5: USING ACCESS KEY MOUNTED THE AZURE DATA LAKE STORAGE (ADLS) CONTAINER TO AZURE DATABRICKS



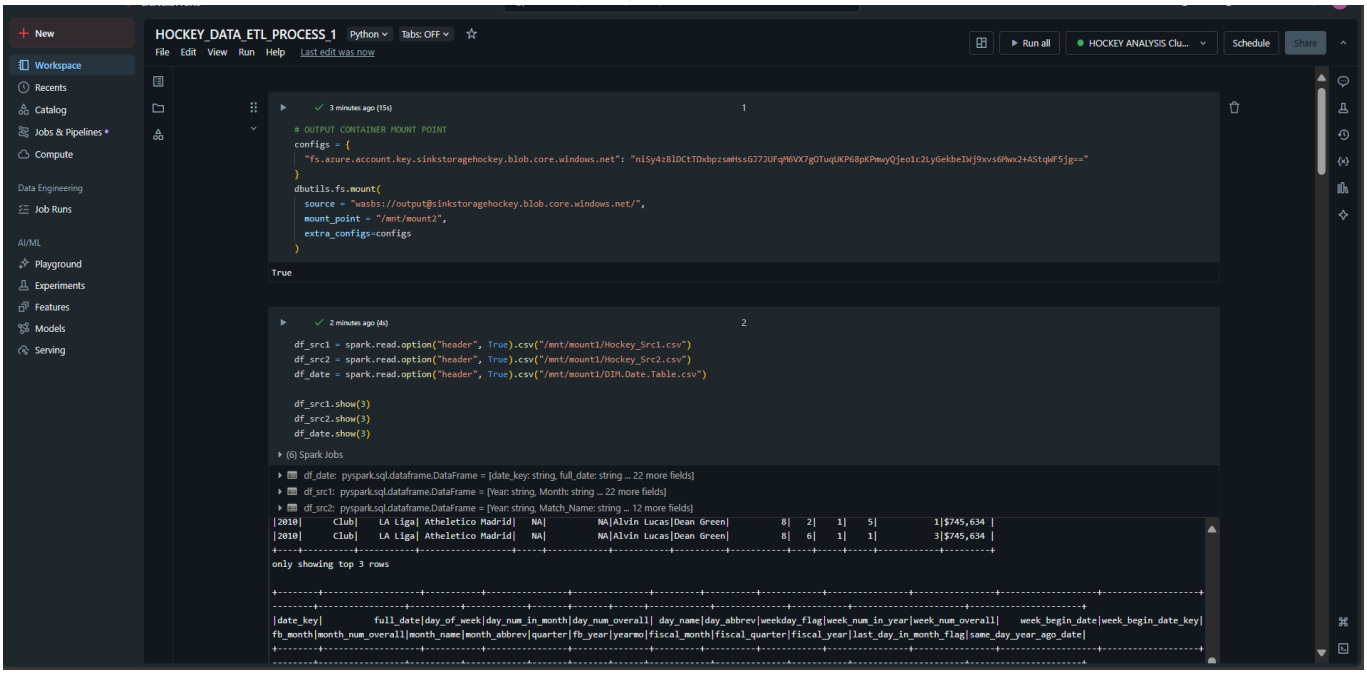
STEP 6: CREATE AZURE DATABRICKS WORKSPACE



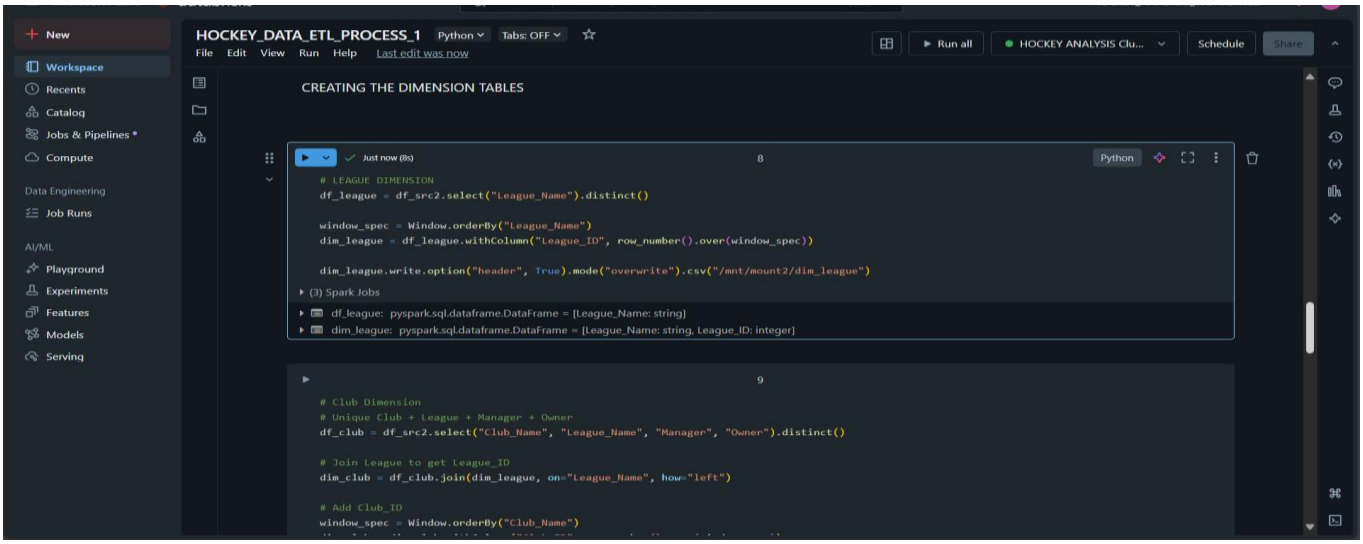


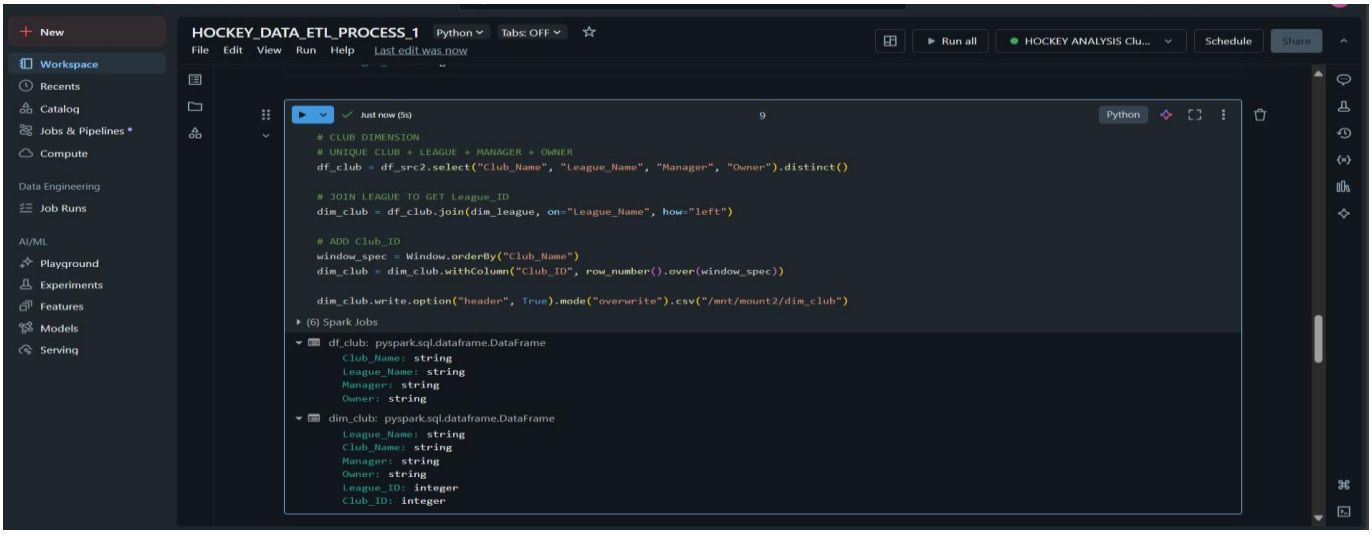
STEP 7: TRANFORMATION IN DATABRICKS

Read CSVs

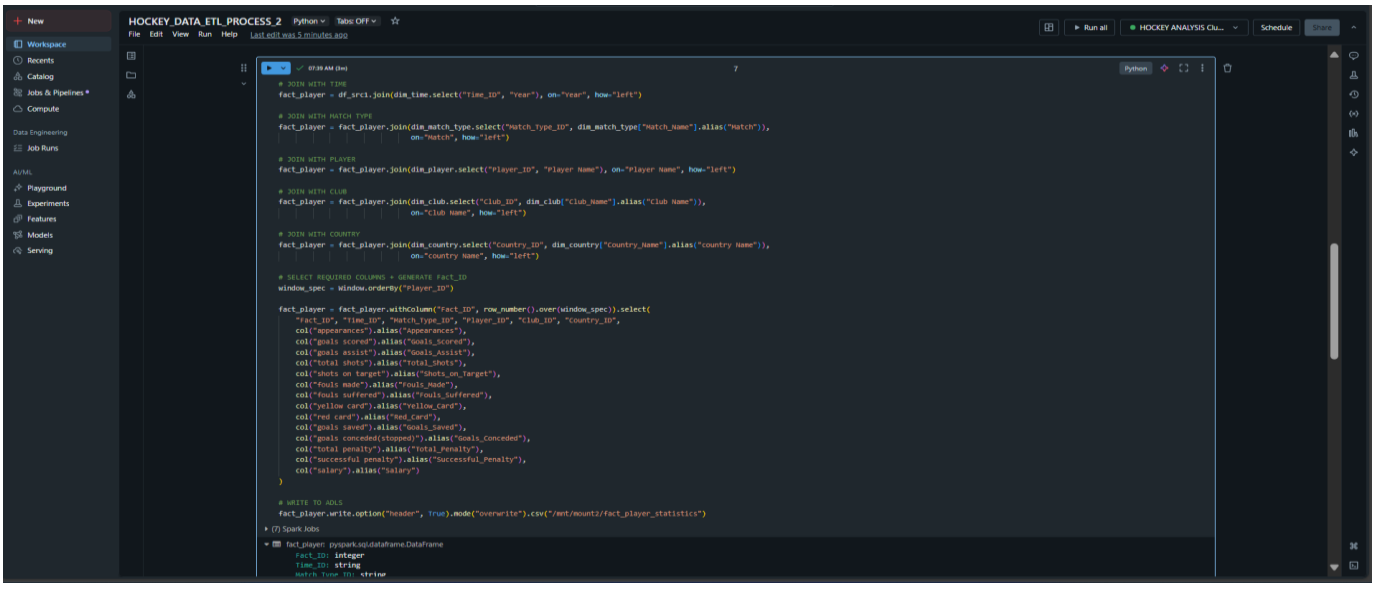


Create dimension tables

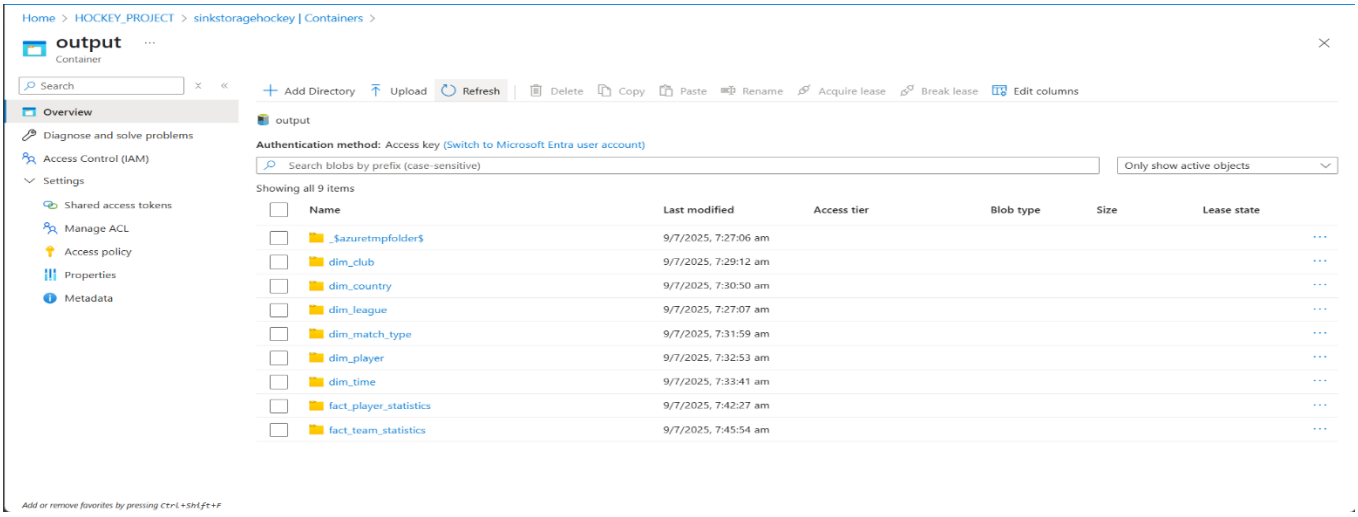




Create fact tables

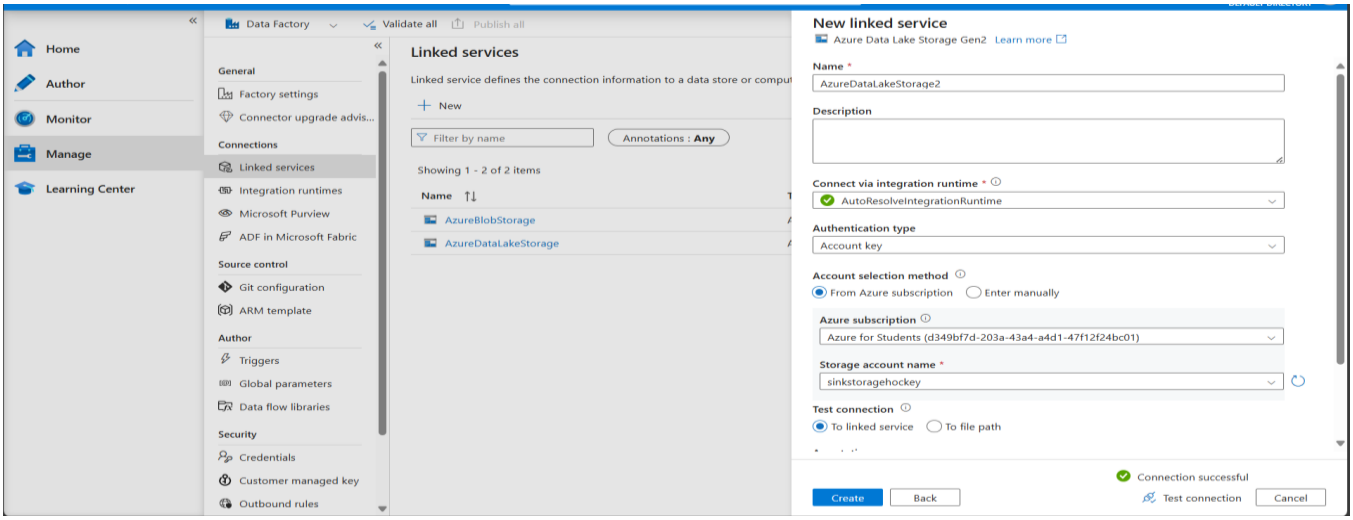


Write transformed tables to output

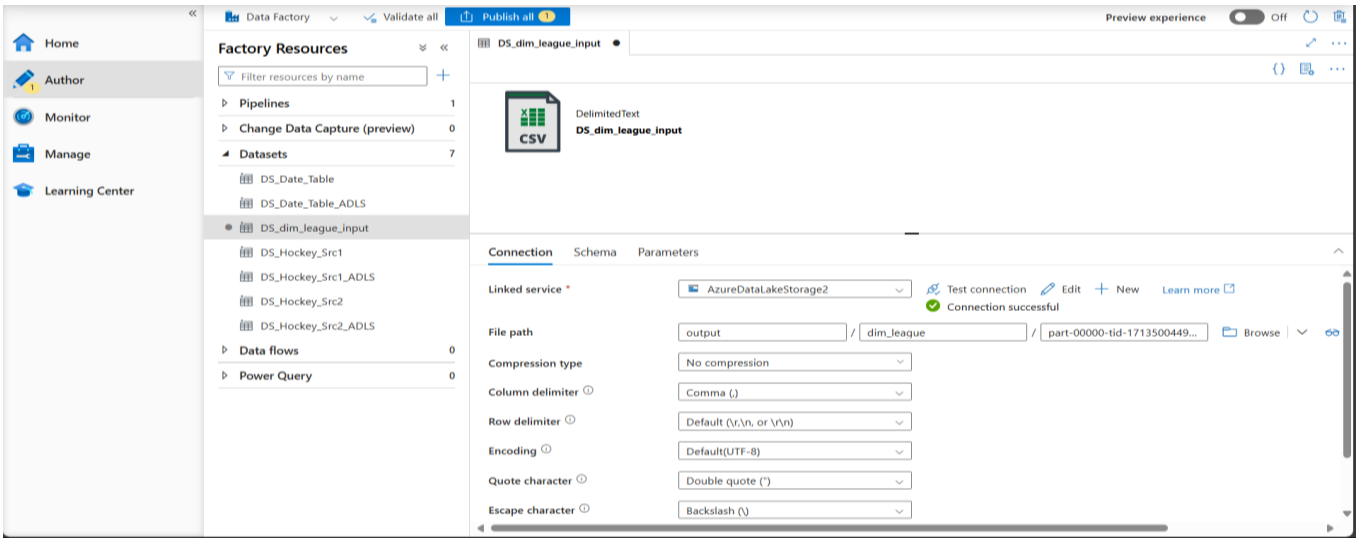


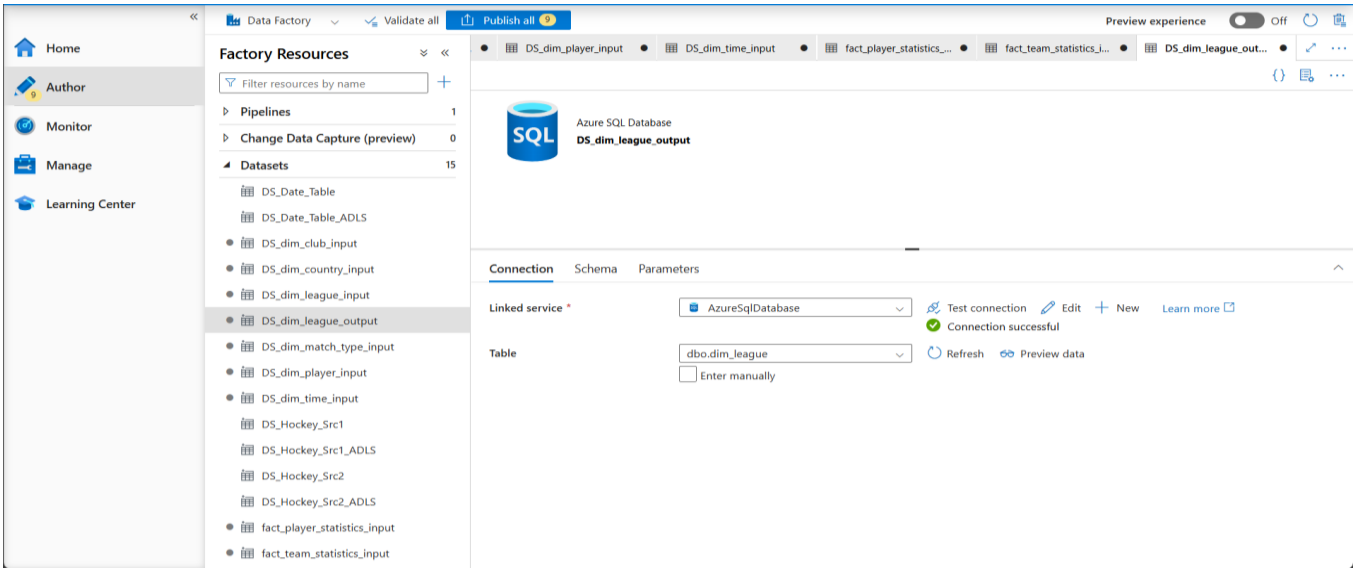
STEP 8: LOAD TO SQL USING AZURE DATA FCATORY (ADF)

Create linked services



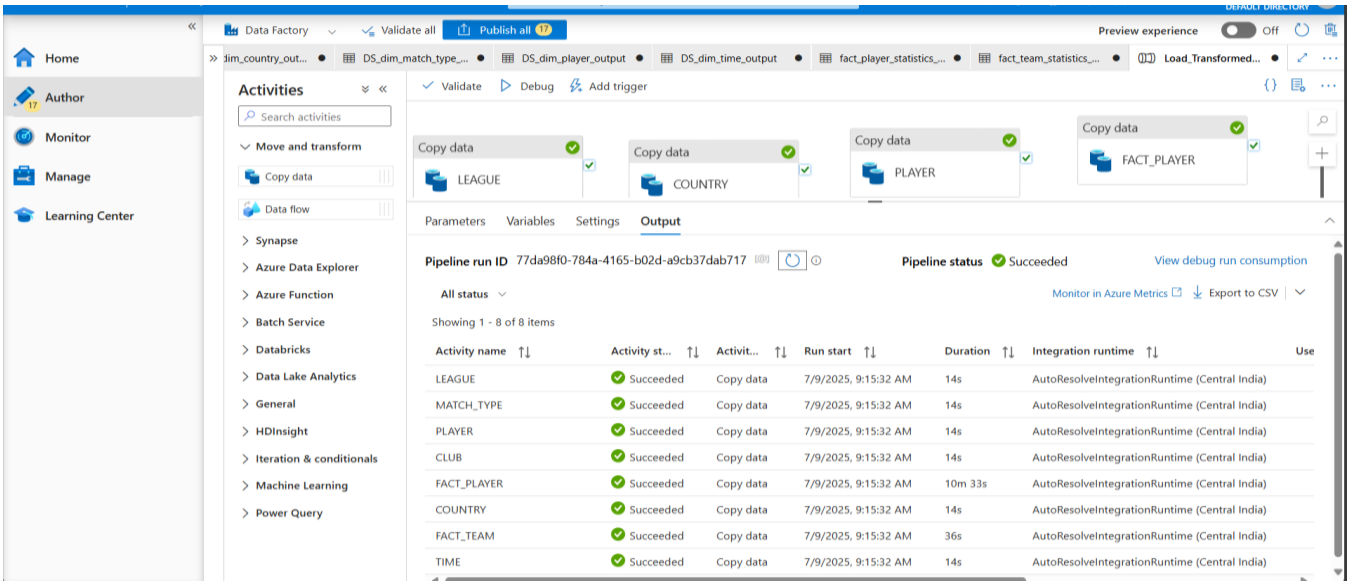
Create datasets (Input + Output)

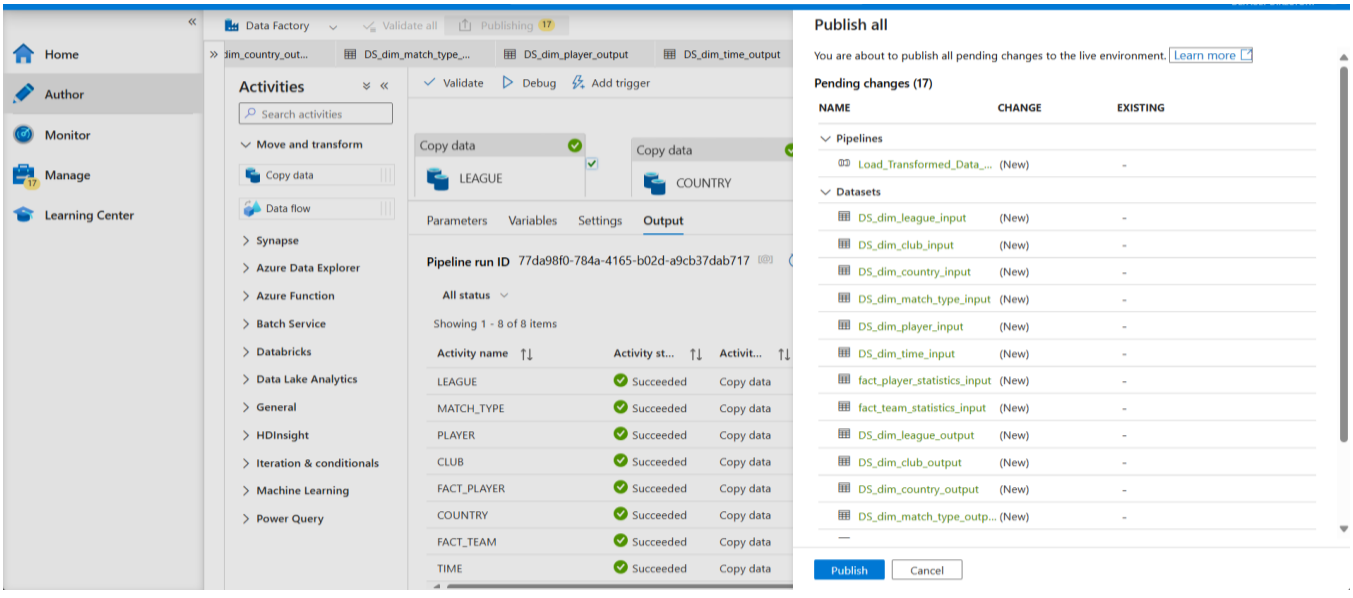




NOTE: Similarly created for other files

Build the copy data pipeline and Run the pipeline

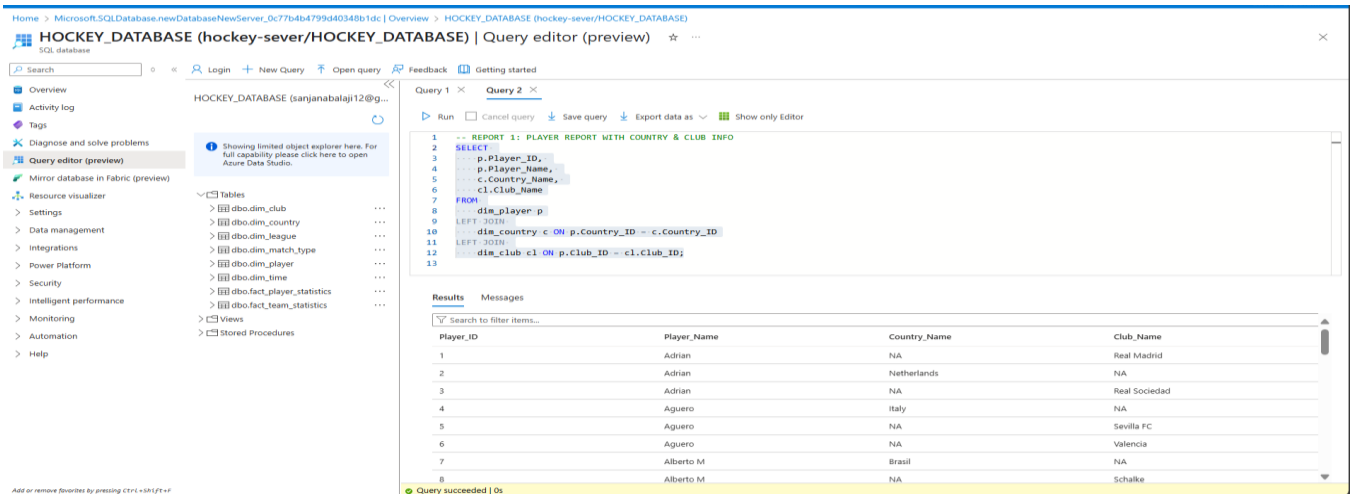




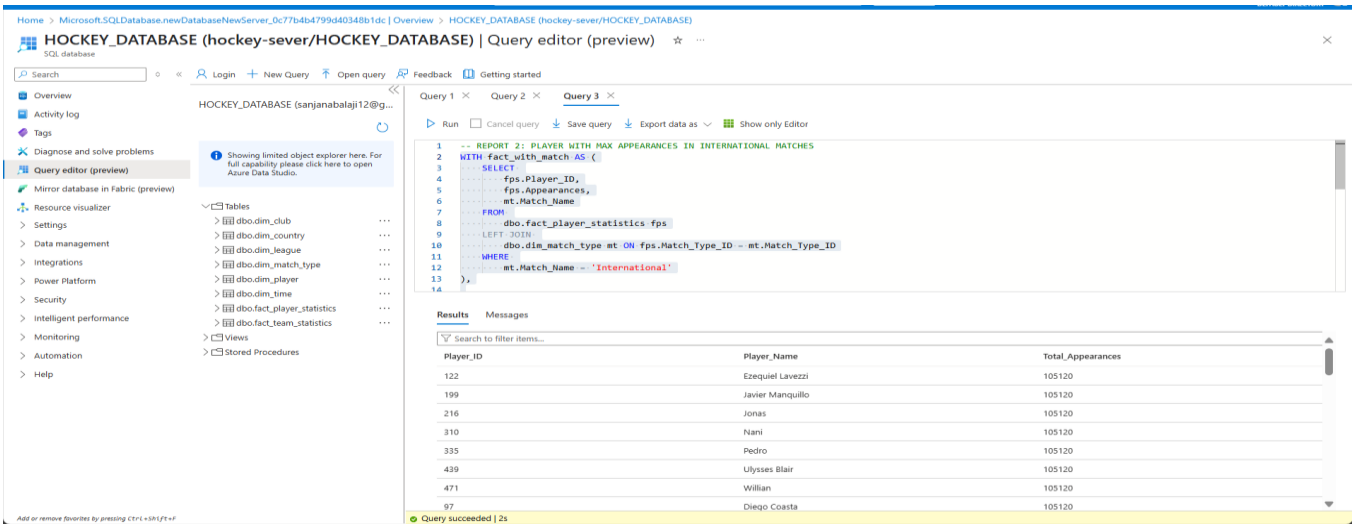
STEP 9: SQL REPORTING QUERIES

STEP 10: CSV OUTPUT

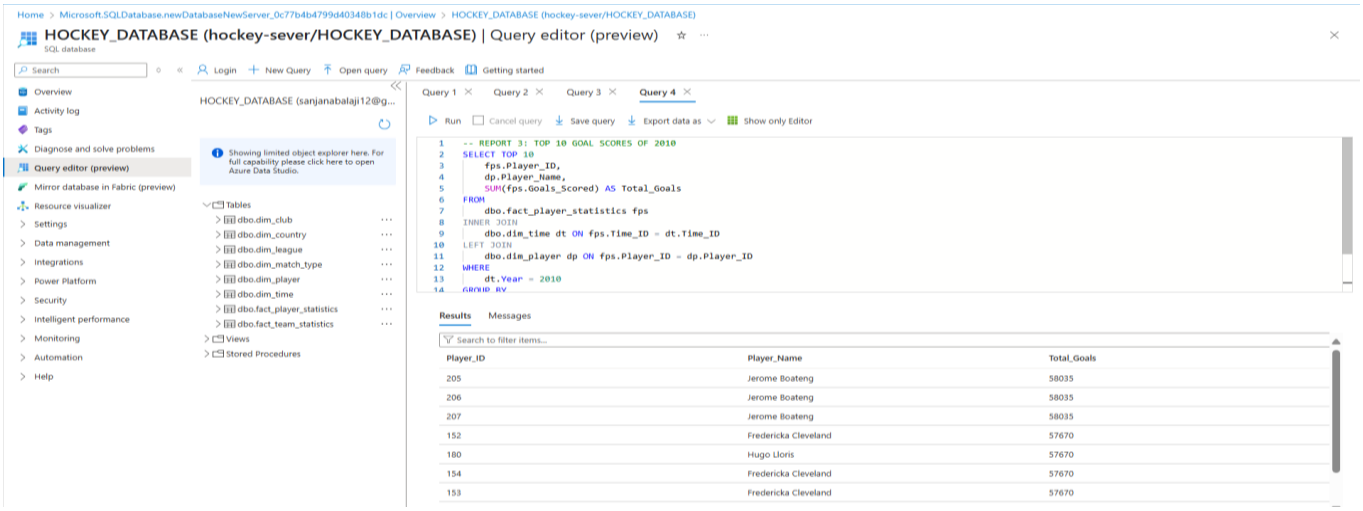
QUERY-1: Player report with country and club info



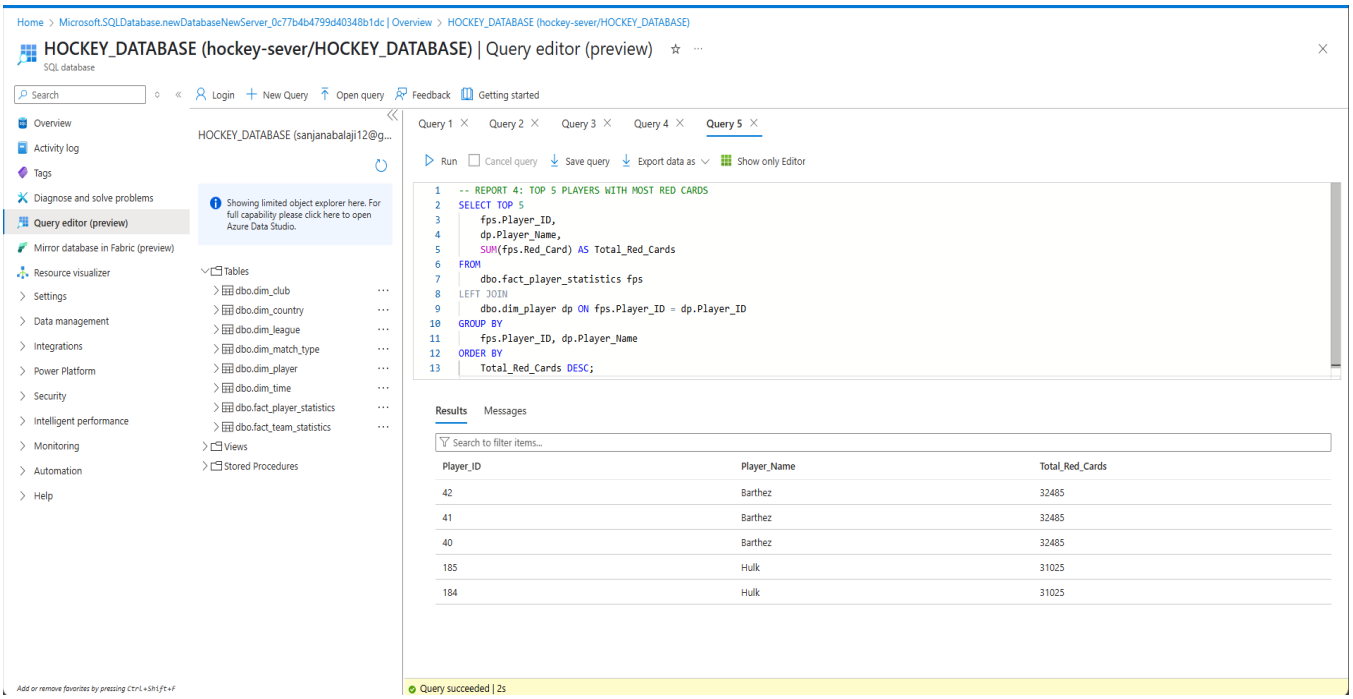
QUERY-2: Player with max Appearances in international matches



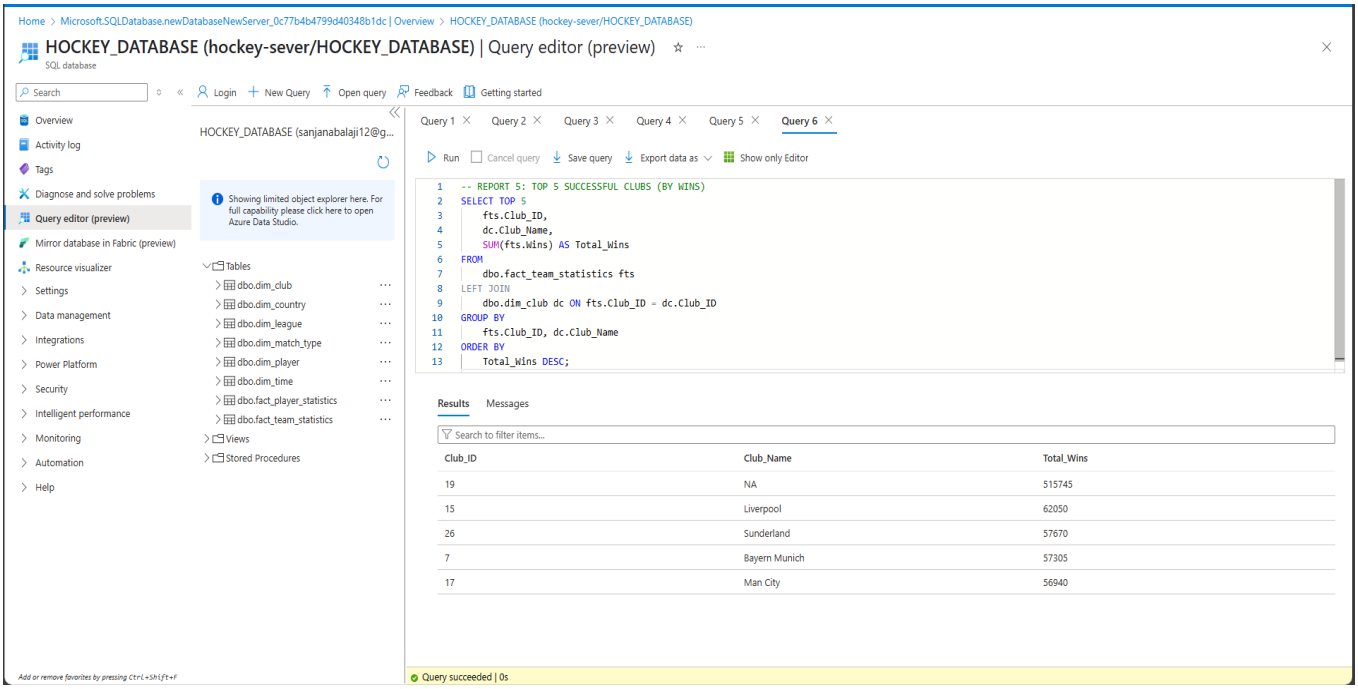
QUERY-3: Top 10 goal sources of 2010



QUERY-4: Top 5 players with most red cards



QUERY-5: Top 5 Successful clubs(by wins)



4. REPORTS

REPORT-1: Player report with country and club info

REPORT-1.csv

REPORT-2: Player with max Appearances in international matches

REPORT-2.csv

REPORT-3: Top 10 goal sources of 2010

REPORT-3.csv

REPORT-4: Top 5 players with most red cards

REPORT-4.csv

REPORT-5: Top 5 Successful clubs(by wins)

REPORT-5.csv

5.CONCLUSION

This mini-project successfully demonstrates the design and implementation of a scalable, end-to-end data analysis and reporting system for hockey game statistics using PySpark within the Azure ecosystem. By efficiently handling large volumes of flat file data, the system transforms raw operational datasets into a well-structured dimensional data warehouse using a star schema model. This model provides a clear, analytical view of various aspects of the game, including player performance, club statistics, and match outcomes.

The use of PySpark's distributed processing capabilities significantly enhanced the speed and scalability of data transformation, allowing for complex ETL operations to be performed seamlessly. The implementation of an ELT pipeline ensured that the data was cleaned, enriched, and integrated before being used for reporting. The dimensional model supported the generation of multiple performance-based reports such as top scorers, most appearances, and club rankings, offering valuable insights for coaches, analysts, and management teams.

In conclusion, this project not only enhanced our understanding of PySpark and big data processing but also reinforced the significance of well-designed data pipelines in modern analytics. The solution built through this project is flexible, scalable, and adaptable, capable of supporting future enhancements such as real-time data integration, machine learning-based performance predictions, and integration with cloud-based reporting tools.