**Project Documentation: Azure-Based Data Engineering Pipeline**

**1. Project Overview**

**Summary:**  
This project demonstrates an end-to-end data engineering pipeline using Azure services to ingest, clean, process, store, and visualize IPL-related data from raw CSVs to insightful Power BI dashboards.

**Objective:**  
To build a scalable, automated, and efficient data pipeline that:

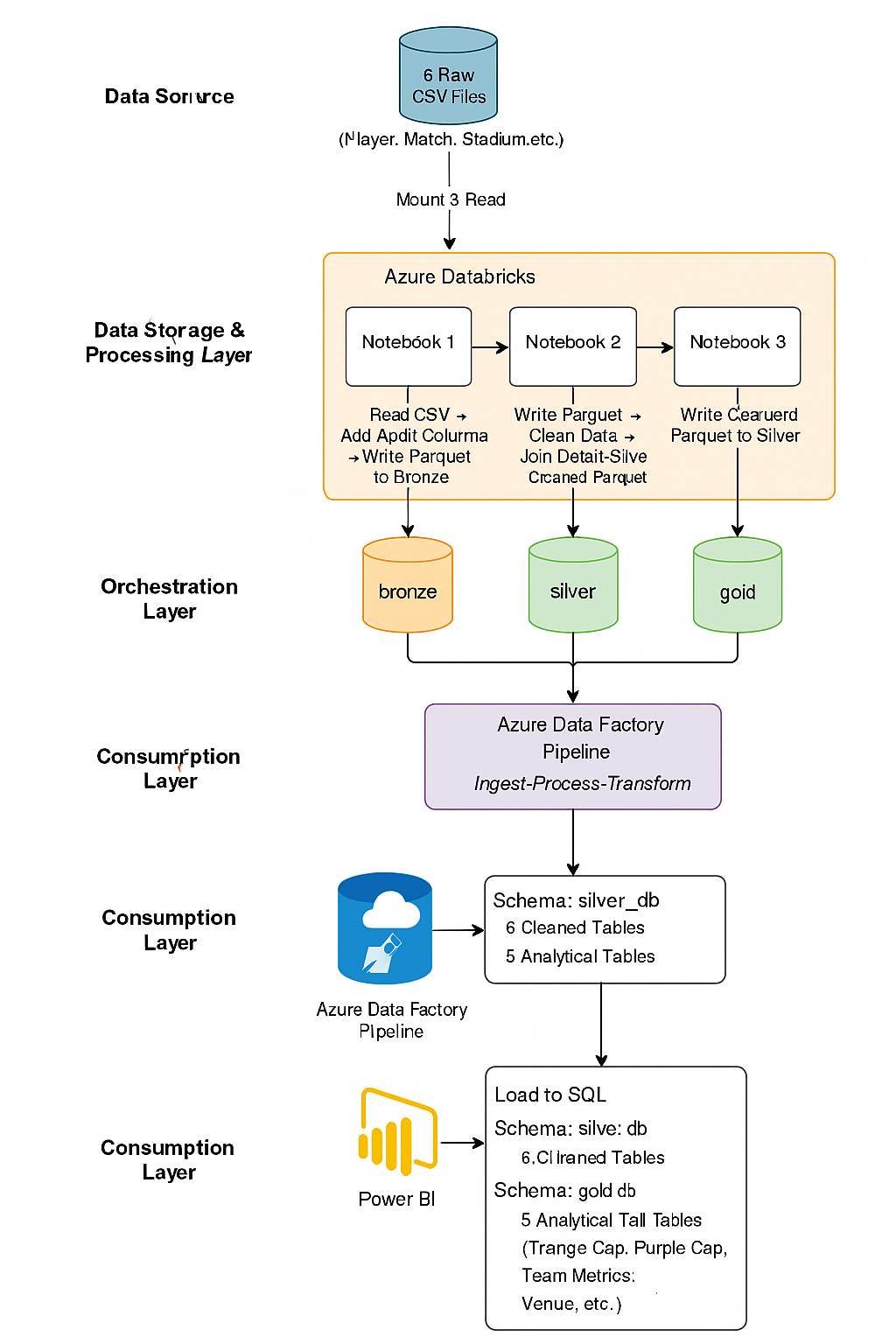
* Ingests raw CSV files into Azure Blob Storage.
* Transforms and cleans data using Azure Databricks (PySpark).
* Stores data at multiple stages (Bronze, Silver, Gold) in Azure Data Lake Storage Gen2.
* Loads data into Azure SQL Database for querying.
* Creates a final Power BI dashboard for analytics and KPIs.

**Technologies Used:**

* Azure Blob Storage
* Azure Data Lake Storage Gen2 (ADLS)
* Azure Databricks (PySpark)
* Azure SQL Database
* Azure Data Factory (ADF)
* Power BI

**2. Architecture Diagram**

**Summary:**  
The pipeline consists of multiple stages connected via Azure services. Each stage performs specific tasks, from raw ingestion to advanced analytics.



**3. Data Ingestion & Storage**

**Summary:**  
Set up cloud infrastructure to store raw and processed data in an organized manner.

* **Resource Group** created in Azure.
* **Blob Storage:**
  + Container: raw
  + Stores original CSV files.
* **Azure Data Lake Gen2 (ADLS):**
  + Containers: bronze, silver, gold
* **CSV Files Ingested:**
  + player.csv
  + match.csv
  + stadium.csv
  + player\_match.csv
  + team.csv
  + player\_team.csv

**4. Data Processing with Databricks**

**Summary:**  
Used three Databricks notebooks to transform and process data through different layers (Bronze, Silver, Gold).

**Notebook 1: Raw to Bronze**

* Mounted Blob storage to Databricks.
* Read all CSVs using Spark.
* Added audit columns: ingestion\_time, source\_file.
* Converted files to Parquet format.
* Wrote to bronze container.

**Notebook 2: Bronze to Silver**

* Mounted and read Parquet files from bronze.
* Data cleaning operations:
  + Drop nulls.
  + Rename columns.
* Performed joins to combine datasets into a unified master table.
* Wrote cleaned data to silver container.

**Notebook 3: Silver to Gold**

* Read cleaned data from silver.
* Created Temp Views in Spark.
* Performed SQL queries to generate insights:
  + Total Wins
  + Player Stats
  + Venue Analysis
* Stored analytical tables in the gold container.

**5. Automation with Azure Data Factory (ADF)**

**Summary:**  
Orchestrated the pipeline using ADF pipelines to trigger Databricks notebooks sequentially.

* Created **one ADF pipeline** with **3 notebook activities**:
  1. Raw → Bronze (Notebook 1)
  2. Bronze → Silver (Notebook 2)
  3. Silver → Gold (Notebook 3)
* Connected Databricks workspace and notebook activities.
* Achieved full end-to-end **automation**.

**6. Azure SQL Database Integration**

**Summary:**  
Used JDBC connections to transfer data from Databricks into Azure SQL DB for centralized storage and Power BI access.

* Created two schemas:
  + silver\_db – Stores cleaned tables
  + gold\_db – Stores analytical/aggregated tables

**Total Tables:**

**Silver DB:**

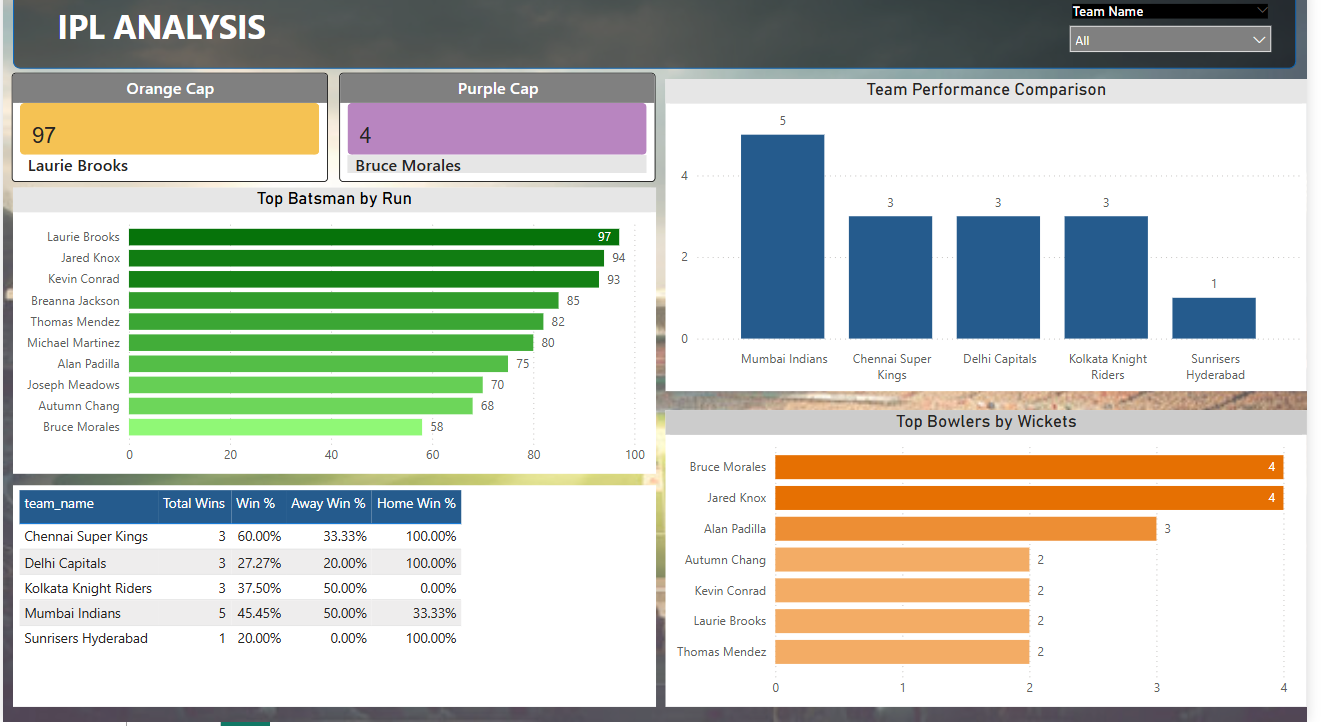
* player\_cleaned
* match\_cleaned
* player\_match\_cleaned
* team\_cleaned
* stadium\_cleaned
* player\_team\_cleaned

**Gold DB (Analytical Tables):**

* team\_performance\_metrics
* player\_contribution
* venue\_analysis
* player\_efficiency\_metrics
* match\_summary\_insights

**7. Power BI Dashboard**

**Summary:**  
Connected Power BI to Azure SQL Database to visualize insights, performance, and key metrics of the IPL dataset.

* **Connection:** Azure SQL (Gold DB tables)
* **KPIs Created:**
  + **Orange Cap** (Most Runs)
  + **Purple Cap** (Most Wickets)
* **Reports & Visuals:**
  + Team-wise Performance Metrics
  + Top Players by Runs & Wickets
  + Home vs Away Analysis
  + Average Strike Rate by Player
  + Match Results Summary

**8. Challenges & Learnings**

**Summary:**  
Real-world implementation involved handling multiple datasets, formats, and orchestrations.

**Challenges Faced:**

* **Small Dataset**  
  The IPL data volume was limited in size, which may not fully capture the complexities of large-scale, real-world sports analytics projects.
* **Local Environment Setup**  
  Setting up Power BI and SQL Server locally required careful attention to compatibility, especially with JDBC connections and port configurations.
* **Data Quality Issues**  
  The raw IPL files had missing or inconsistent entries, especially in player statistics like runs and wickets, which needed thorough data cleansing to ensure reliable analysis.
* **Inconsistent File Schemas**  
  Different CSV files had varying schema definitions, which made it necessary to perform schema alignment and column standardization during ingestion and transformation stages.

**Key Learnings:**

* Real-time ingestion and transformation
* PySpark optimizations and SQL querying
* Use of layered storage for scalability
* Understood the process of establishing JDBC connections between Azure Databricks and Azure SQL Database for reading and writing data.
* Learned how to automate multi-step ETL processes using ADF pipelines
* Power BI basics

**9. Conclusion**

**Summary:**  
The project successfully showcases how cloud-native tools can be combined to create a powerful, scalable, and automated data pipeline with meaningful analytics.

* All stages of data engineering lifecycle completed.
* Automation achieved using ADF + Databricks.