**A blue and white background

AI-generated content may be incorrect.**

A blue logo with a black background

AI-generated content may be incorrect.

**Azure Data Engineering**

**Project Report**

**Done by:**

Milan Tony

**Date:** 09-07-2025

**Azure Data Engineering Project**

Database used: [AdventureWorksLT2022](https://github.com/Microsoft/sql-server-samples/releases/download/adventureworks/AdventureWorksLT2022.bak)

Project contents: <https://github.com/milantony05/data-engineering-project.git>

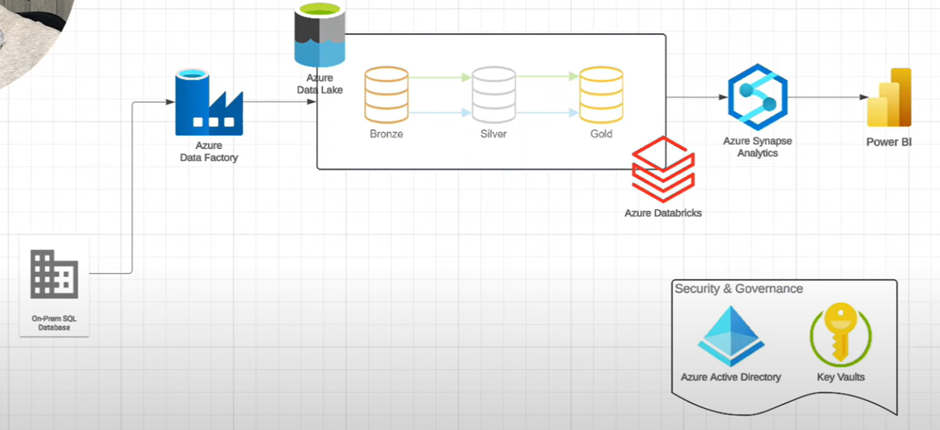
**Business Request**

In this project, your company has recognized a gap in understanding its customer demographics—specifically, the gender distribution within the customer base and how it might influence product purchases. With a significant amount of customer data stored in an on-premises SQL database, key stakeholders have requested a comprehensive KPI dashboard.

This dashboard should provide insights into sales by gender and product category, showing total products sold, total sales revenue, and a clear gender split among customers. Additionally, they need the ability to filter this data by product category and gender, with a user-friendly interface for date-based queries.

**Solution Overview**

To address this request, we'll build a robust data pipeline that extracts the on-premises data, loads it into Azure, and performs the necessary transformations to make the data more query-friendly. The transformed data will then feed into a custom-built report that meets all the specified requirements. This pipeline will be scheduled to run automatically every day, ensuring that stakeholders always have access to up-to-date and accurate data.



**CHAPTERS**

**Introduction**

* Setting Up the Azure Environment
* SQL Database Configuration

**SSMS (SQL Server Management Studio)**

* Configuring Azure Data Factory
* Copying Data from SQL to Data Lake

**Azure Data Factory**

* Testing the SQL-to-Bronze Pipeline
* Debugging the Pipeline
* Monitoring Pipeline Runs
* Verifying Data in Bronze Layer

**Databricks**

* Creating a Databricks Cluster
* Mounting Data Lake Storage in Databricks
* Transformation in Databricks
* Automating Data Transformations
* Integrating Databricks with Data Factory
* Pipeline Testing and Monitoring

**Synapse Analytics**

* Loading Data into Synapse Analytics
* Creating Views in Synapse
* Integrating Synapse Views into Data Factory Pipelines

**Power BI**

* Power BI Dashboard Setup
* Building Relationships in Power BI
* Dashboard Filters and Slicers

**Automation and Active Directory**

* Automating the Entire Pipeline
* Active Directory (Entra ID) Integration
* Triggering and Monitoring Automated Pipelines

**Introduction**

This project aims to bridge the knowledge gap in understanding customer demographics - particularly gender distribution - and its impact on product purchases. By building an automated Azure-based data pipeline we deliver a Power BI dashboard that surfaces Key Performance Indicators (KPIs) such as total products sold, total sales revenue and gender split, with flexible category and date filtering.

**1) Setting Up the Azure Environment**

**1.1 Resource Provisioning**

|  |  |  |
| --- | --- | --- |
| Resource | Purpose | Notes |
| Resource Group | Logical container for all artefacts | ‑ |
| Azure Data Factory (ADF) | Orchestrate data movement & transformation | Created inside RG |
| Storage Account (Data Lake Gen2) | Persist raw (bronze), cleaned (silver) & curated (gold) data | Bronze container created |
| Azure Databricks | Spark-based transformations | Workspace auto-creates its own RG |
| Azure Synapse Analytics | Serverless SQL pool for BI serving layer | Connected to Data Lake |
| Azure Key Vault | Secrets management (SQL pwd, Databricks token) | Access via RBAC |

**1.2 Security & IAM**

• Key Vault Administrator role assigned to RG to allow secret creation.  
• Managed Identities & Entra ID security group used instead of individual principals for least-privilege governance.

**2) SQL Database Configuration**

On-premises SQL Server Express hosts AdventureWorksLT2022 sample DB. Steps:

1. Install SQL Server, SSMS and restore AdventureWorksLT2022 backup.
2. Create SQL login milan with SELECT rights on SalesLT schema.
3. Store credentials in Key Vault (sql-username, sql-password).

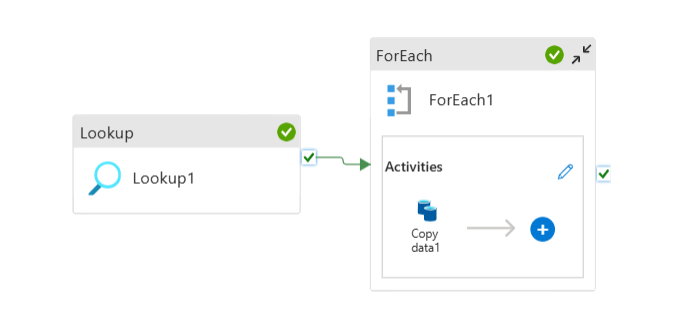
**3) Configuring Azure Data Factory**

**3.1 Self-Hosted Integration Runtime**

Installed on the on-prem machine to enable outbound copy from SQL Server to Azure.

**3.2 Bronze Ingestion Pipeline**

1. Lookup activity enumerates all tables under SalesLT schema using dynamic SQL.
2. ForEach + CopyData iterates through list, extracting each table into Parquet files at bronze/<schema>/<table>/<table>.parquet.
3. Trigger: scheduled daily; also manual reruns for schema/permission fixes.

****

**4) Data Transformation with Databricks**

**4.1 Cluster & Mount**

Single-node cluster (DBR 11.x), credential passthrough enabled. Data Lake containers mounted to /mnt/bronze, /mnt/silver, /mnt/gold.

**4.2 Bronze ➜ Silver (L1)**

• Convert ModifiedDate & any datetime columns to date.  
• Write each refined table to Delta format in silver layer.

**4.3 Silver ➜ Gold (L2)**

• Standardize column names to UPPER\_SNAKE\_CASE.  
• Perform joins/aggregations to build star-schema facts & dimensions.  
• Persist as Delta in gold layer.

**4.4 Operationalizing via ADF**

ADF notebook activities chain after ingestion pipeline: Bronze-to-Silver followed by Silver-to-Gold, using Databricks linked service authenticated via Key Vault-held PAT.

A screenshot of a computer

AI-generated content may be incorrect.

**5) Synapse Analytics Serving Layer**

1. Create gold\_db in serverless SQL pool.
2. External views (dbo.<table>) created automatically by stored procedure CreateSQLServerlessView\_gold, executed per table through Synapse pipeline with Get Metadata → ForEach → Stored Proc activities.
3. Views auto-reflect underlying Delta updates; pipeline rerun needed only for schema changes.

**A screenshot of a computer

AI-generated content may be incorrect.**

**6) Power BI Reporting**

• Power BI Desktop connects via DirectQuery to Synapse serverless endpoint.  
• Data model defines relationships between fact & dimension views (e.g., SalesOrderDetail ↔ Product).  
• Report visuals: KPI cards (Total Products, Total Sales), Donut chart (Gender distribution), slicers for Title (gender proxy) & Product Category.  
• Refresh triggered manually or via Power BI service after pipeline completes.

**A screenshot of a graph

AI-generated content may be incorrect.**

**7) Automation & Monitoring**

|  |  |  |
| --- | --- | --- |
| Layer | Mechanism | Frequency |
| Ingestion (SQL ➜ Bronze) | ADF Schedule Trigger | Daily @ 02:00 UTC |
| Transform (Bronze ➜ Gold) | Chained ADF activities | Follows ingestion |
| Synapse View Refresh | Not required (views are virtual) | N/A |
| Power BI Dataset Refresh | Pro licence scheduled | After pipeline |

ADF & Synapse provide run-level monitoring with retry & alert configuration. Delta time-travel in silver/gold offers data-quality rollback.

**8) Security & Governance**

• Entra ID security group controls contributor rights across RG.  
• Managed identities used for Synapse ↔ Data Lake & ADF ↔ Key Vault interactions.  
• RBAC roles: Storage Blob Data Contributor (Synapse), Key Vault Secrets User (ADF), etc.

**9) End-to-End Test**

Inserted new Product row in on-prem SQL. Scheduled trigger executed full pipeline; Power BI refresh displayed incremented product count, verifying data freshness.

**Conclusion**

The delivered Azure data platform ingests on-prem transactional data, applies scalable Spark transformations, serves analytics through serverless SQL and visualises insights in Power BI—all orchestrated and secured via Azure-native services. Stakeholders now possess a gender-aware sales dashboard refreshed daily, enabling data-driven marketing and inventory decisions.