DEVELOPING DATA PIPELINES WITH AZURE SYNAPSE

By Temi | November 29, 2023

REPLY
MACHINE LEARNING

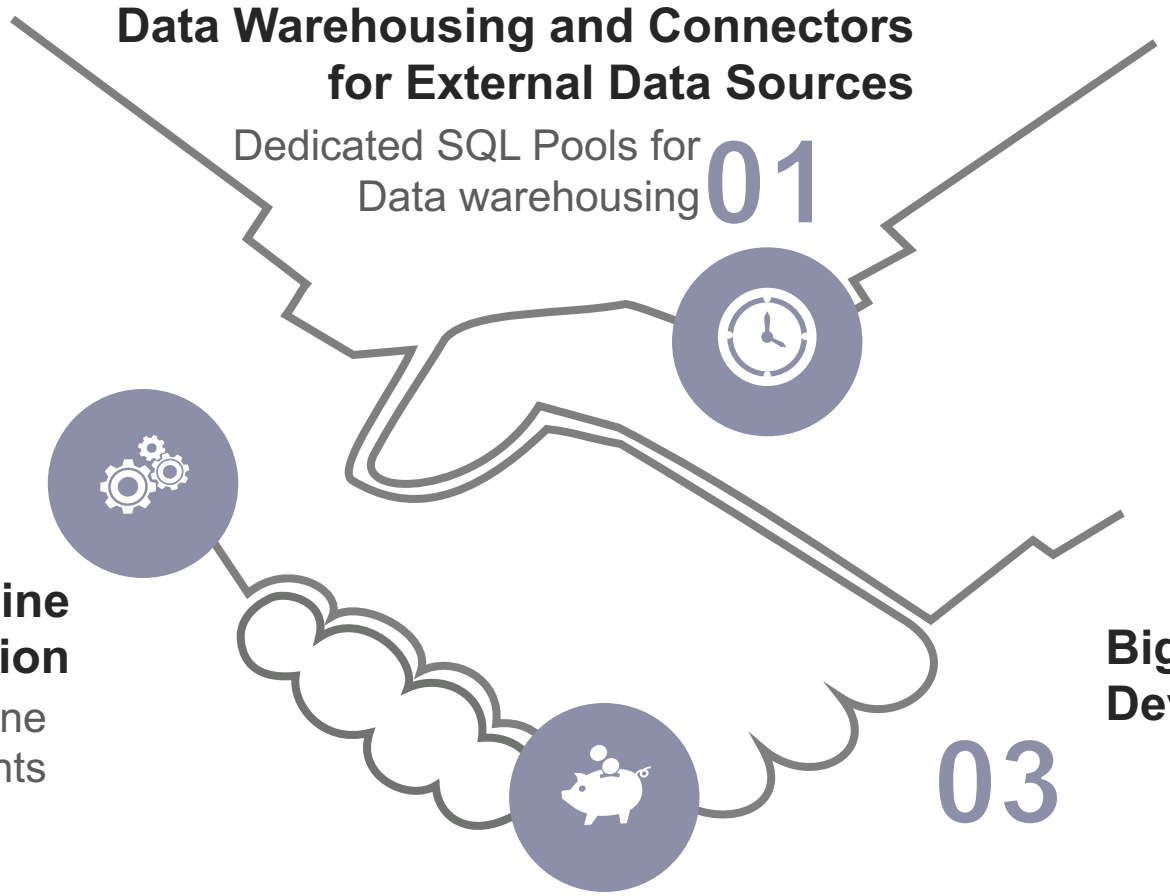
Agenda

REPLY
MACHINE LEARNING

Content **Page**

Big Data Integration with Azure Synapse

REPLY
MACHINE LEARNING



Data Warehousing and Connectors for External Data Sources

Dedicated SQL Pools for Data warehousing

01

02

Data Pipeline Orchestration

Built-in ETL pipeline Orchestration components

03

Big Data Spark Clusters and Development Environment

Spark Pools and Development Notebooks

Azure Synapse Features

REPLY MACHINE LEARNING

1 Linked Services

- Stores connection information to external datasources or compute like storage account, Azure SQL DB (postgres, mysql), Azure Keyvault

2 Integration Datasets

- Stores connection information to datasets themselves

3 Integration Runtime

- It is referenced by a linked service or a pipeline activity
- Provides compute where pipeline activity is dispatched
- **Types**: Azure or Selfhosted

4 Version Control with Git

- Git integration with Github or Azure pipelines

5 Database Objects

- Schemas
- Tables and Views
- Store procedures

6 Synapse Pipelines and activities

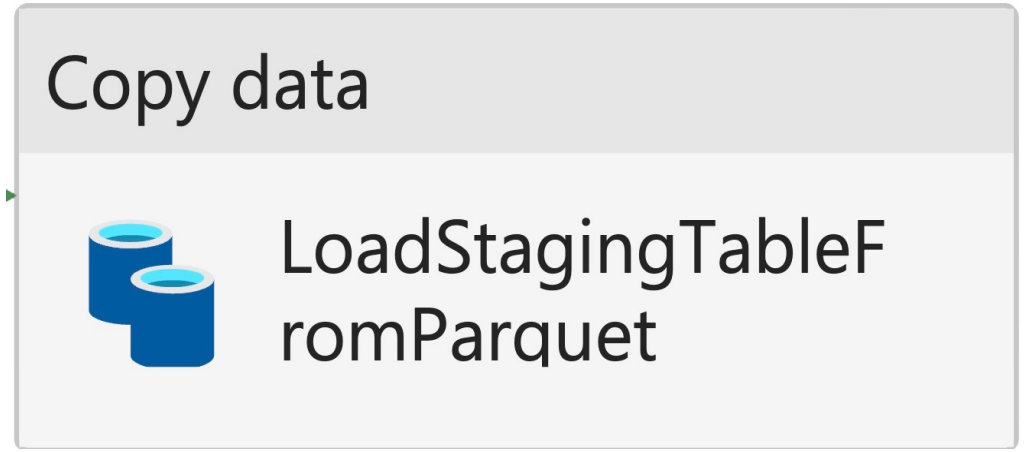
- Orchestrate ETL pipelines using components / activities

Pipeline Components – Copy Data Activity

REPLY MACHINE LEARNING

Copy data from one location to another

- From SQL Server database to Storage account in Parquet format.
- Copy text files (CSV) format from an on-premises file system and write Storage account in Avro format.
- Copy data from Storage account to SQL Pool



Pipeline Components – Notebook Activity

REPLY MACHINE LEARNING

Run a Notebook on a Spark Pool

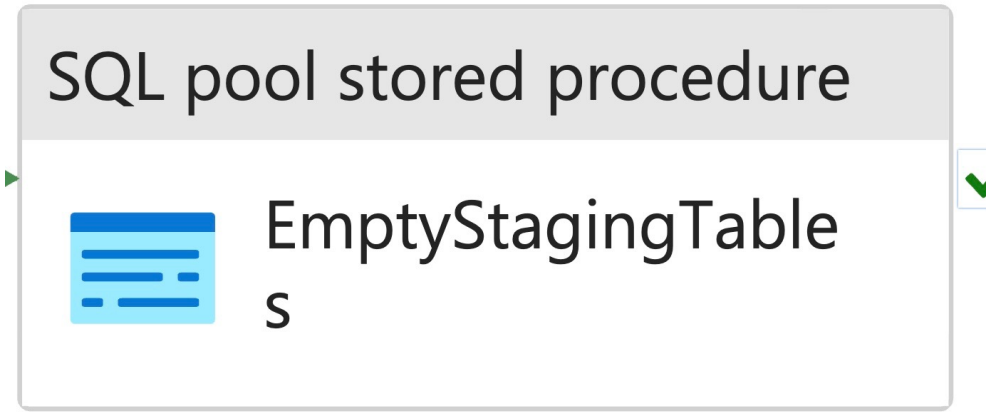
- Process data in Storage account using Pyspark and Pandas
- Save data to storage account using the abfss protocol (managed identity of synapse requires Storage Blob Contributor IAM permission)

Pipeline Components – Stored Procedure Activity

REPLY MACHINE LEARNING

Run a stored procedure in the Dedicated SQL pool

- Execute stored procedures written in TSQL in an ETL pipeline



ETL Pipeline Demo – Developer Productivity Tracking

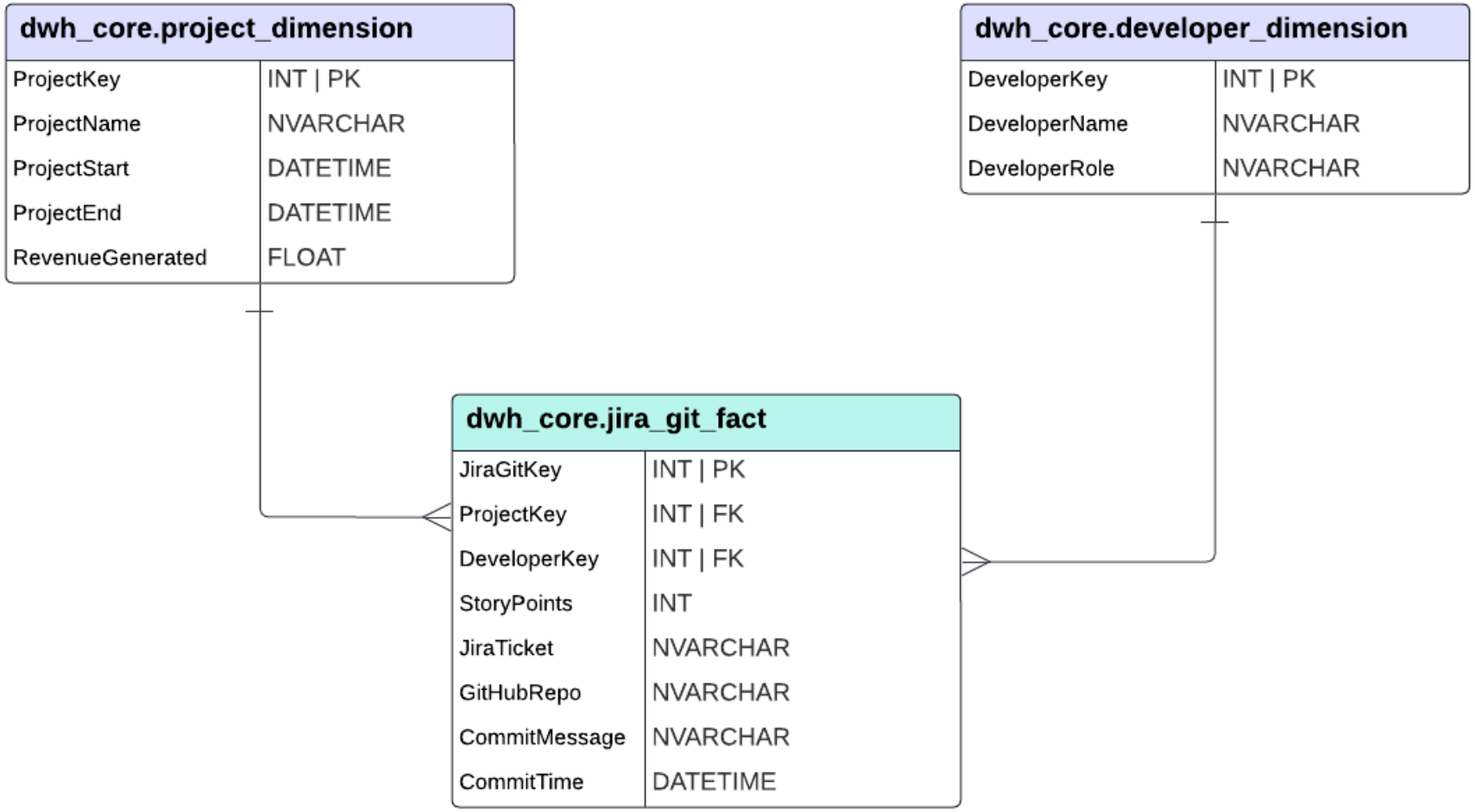
REPLY
MACHINE LEARNING

Scenario

External data from Git and Jira is integrated into our Datawarehouse in Synapse.

Given information on our projects, developers and their activity on Github and Jira, KPIs are created to measure their productivity.

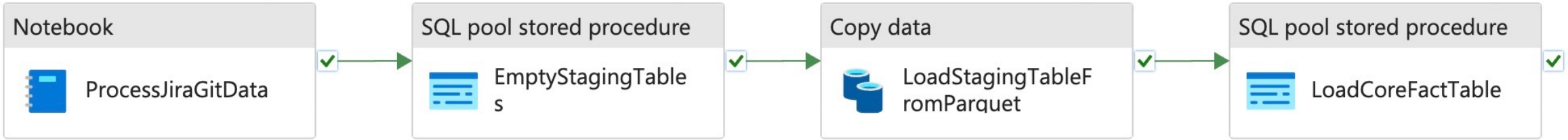
ERD Diagram


dwh_core.project_dimension
ProjectKey
INT | PK
ProjectName
NVARCHAR
ProjectStart
DATETIME
ProjectEnd
DATETIME
RevenueGenerated
FLOAT
dwh_core.developer_dimension
DeveloperKey
INT | PK
DeveloperName
NVARCHAR
DeveloperRole
NVARCHAR
dwh_core.jira_git_fact
JiraGitKey
INT | PK
ProjectKey
INT | FK
DeveloperKey
INT | FK
StoryPoints
INT
JiraTicket
NVARCHAR
GitHubRepo
NVARCHAR
CommitMessage
NVARCHAR
CommitTime
DATETIME

Developer Productivity Pipeline for Jira Git Table

REPLY MACHINE LEARNING

1. **Notebook:** process the **jira_git_fact.csv** into **jira_git_fact.parquet**
2. **CopyData:** copy from **jira_git_fact.parquet** into staging table – **dwh_staging.jira_git_fact**
3. **Stored Procedure:** copy from **dwh_staging.jira_git_fact** into core table - **dwh_core.jira_git_fact**



Load CSV Pipeline for Developer and Project Tables

REPLY MACHINE LEARNING

1. **CopyData:** copy from developer_dimenson.csv into staging table – dwh_staging.developer_dimension
2. **Stored Procedure:** copy from dwh_staging.developer_dimension into core table - dwh_core.developer_dimension

Load CSV Pipeline Parameters

REPLY
MACHINE LEARNING

PARAMETER NAME	DEVELOPER_DIM_VALUES	PROJECT DIM VALUES
CsvFolder	bronze/developer_dim	bronze/project_dim
CsvFileName	developer_dimension.csv	project_dimension.csv
StagingStoredProcedure	dwh.empty_staging_tables	dwh.empty_staging_tables
CoreStoredProcedure	dwh.load_core_developer	dwh.load_core_project
StagingTableName	developer_dimension	project_dimension

REPLY
MACHINE LEARNING

THANK YOU