

Lakehouse vs Data warehouse: Metadata Driven Pipelines Pattern

The logo for Data Saturdays, featuring the word "DATA" in large white letters and "SATURDAYS" in smaller white letters below it, with a blue stylized "D" icon to the left.

Sponsors



Chi sono

- Nel 2023 Fondatore e CTO start-up Regolo Farm società Partner Microsoft per analisi dati
- Nel 2020 CTO presso Cargo BI per insurance data analytics per le compagnie assicurative
- Dal 2017 mi occupo di architetture big Data e in generale di tutta la proposizione data platform di Microsoft.
- 2010 Consulente e formatore in ambito business intelligence
- Docente all'Università di Pordenone per i corsi IFTS di analisi Big Data
- Community Lead di 1nn0va (www.innovazionefvg.net)
- MCP,MCSA,MCSE dal 2017 MCT e dal 2014 MVP per SQL Server e relatore in diverse conferenze sul tema.
 - marco.pozzan@regolofarm.com
 - @marcopozzan.it
 - www.marcopozzan.it

DATA
SATURDAYS




Estrazione di 4 voucher con 100%
sconto su corso Power BI


Estrazione di 4 voucher con 70%
sconto su corso Power BI

Come utilizzare Microsoft Power BI

Scopri come utilizzare al meglio Microsoft Power BI, uno strumento potente e versatile che ti permetterà di analizzare, visualizzare e presentare dati in modo semplice e intuitivo.

★★★★★
© 1,746 visualizzazioni


**Marco Pozzan**
Esperto di Business Intelligence con oltre 10 anni di esperienza nel settore dell'analisi dei dati















€ 49,95 +IVA

Accedi a tutti i corsi — abbonamento

Accedi a questo corso — **€ 49,95** +IVA

 Regala questo corso

-  **Garanzia 100% soddisfatti** 
-  Accesso a vita 
-  Attestato di formazione 
-  Impara ovunque e in qualsiasi momento con le app mobile e l'accesso offline 
 
-  Rilascio crediti ECP 
-  29 video, 02h 57m 26s

Workshop Fabric Full Day

20%

di sconto

~~350~~ - > 280

Vieni a provare fabric con il tuo portatile!!!

WORKSHOP FABRIC FULL DAY



INTRODUZIONE A MICROSOFT FABRIC, LA NUOVA PIATTAFORMA DI ANALISI DATI DI MICROSOFT

8 MARZO 2024

**Intera giornata di
laboratori!!!**

Consorzio di Pordenone
via Prasecco 3/A | 33170
Pordenone | Italy

Per informazioni info@consortiumservice.it

Il workshop partirà con un minimo di 20 persone

AGENDA

Lakehouse vs Data Warehouse

Punto principale per decidere quale servizio utilizzare

- Come verranno consumati i dati
- Requisiti per il tuo ETL/ELT
- Competenze dello sviluppatore

Architetture possibili

- Demo Pattern 1
- Demo Pattern 2

<https://www.linkedin.com/pulse/diversi-endpoint-di-fabric-e-i-loro-scenari-utilizzo-marco-pozzan-4frge/>

Data Lakehouse (Synapse Data Engineering)

- È una piattaforma per archiviare, gestire e analizzare dati strutturati e non strutturati in un unico posto, in un formato file predefinito cioè delta parquet. Puoi archiviare i dati in 2 posizioni fisiche:
 - **Tables** (managed delta tables): Questa è un'area gestita per ospitare tabelle di tutti i formati in Spark (CSV, Parquet o Delta Parquet). Tutte le tabelle, create automaticamente o esplicitamente, vengono riconosciute come tabelle in Lakehouse.
 - **File** (unmanaged): Questa è un'area non gestita per l'archiviazione di dati in qualsiasi formato. Tutti i file Delta Parquet archiviati in quest'area non vengono riconosciuti automaticamente come tabelle.
- Lakehouse ti consente di eseguire trasformazioni tramite calcoli Spark o utilizzare endpoint SQL per l'analisi o l'esplorazione dei tuoi dati. Il formato file predefinito del Data Lakehouse è delta parquet, ottimale per le prestazioni del carico di lavoro analitico.

Data Warehouse (Synapse Data Warehousing)

- È incentrato su OneLake ed è costruito su un motore di calcolo distribuito. Uno dei principali vantaggi di Fabric Data Warehouse rispetto ad altre soluzioni di data warehousing è che non è necessario copiare i dati affinché possano essere utilizzati da altri motori o team di elaborazione poiché il warehouse utilizza OneLake come spazio di archiviazione quindi un'unica copia dei dati archiviati in formato delta parquet.
- Fondamentalmente, il data warehouse è un motore SQL MPP (motore di elaborazione parallela) con tabelle Delta ed endpoint TDS (Tabular Data Stream), che fornirà il supporto completo DDL e DML T-SQL. È un'infrastruttura serverless che consente una scalabilità infinita con allocazione dinamica delle risorse:
 - scalabilità verso l'alto istantanea senza alcun provisioning fisico
 - scalabilità verso il basso istantanea senza alcun provisioning fisicole risorse fisiche vengono allocate ai processi in pochi millisecondi.

Recap

| | Data warehouse | Lakehouse |
|--------------------------------------|--|--|
| Data volume | Unlimited | Unlimited |
| Type of data | Structured | Unstructured,semi-structured,structured |
| Primary developer persona | Data warehouse developer, SQL engineer | Data engineer, data scientist |
| Primary developer skill set | SQL | Spark(Scala, PySpark, Spark SQL, R) |
| Data organized by | Databases, schemas, and tables | Folders and files, databases, and tables |
| Read operations | Spark,T-SQL | Spark,T-SQL |
| Write operations | T-SQL | Spark(Scala, PySpark, Spark SQL, R) |
| Multi-table transactions | Yes | No |
| Primary development interface | SQL scripts | Spark notebooks,Spark job definitions |

| | Data warehouse | Lakehouse |
|--------------------------------------|--|--|
| Primary development interface | SQL scripts | Spark notebooks,Spark job definitions |
| Security | Object level (table, view, function, stored procedure, etc.), column level, row level, DDL/DML, dynamic data masking | Row level, table level (when using T-SQL), none for Spark |
| Access data via shortcuts | Yes (indirectly through the lakehouse) | Yes |
| Can be a source for shortcuts | Yes (tables) | Yes (files and tables) |
| Query across items | Yes, query across lakehouse and warehouse tables | Yes, query across lakehouse and warehouse tables;query across lakehouses (including shortcuts using Spark) |
| Advanced analytics | | |
| Advanced formatting support | | |
| Ingestion latency | | |

Punto principale per decidere quale servizio utilizzare

1. Come verranno consumati i dati
2. Requisiti per il tuo ETL/ELT
3. Competenze di sviluppatore tra cui data engineers, scienziati dei dati, developers

COME VERRANNO CONSUMATI I DATI

Lakehouse Endpoint per Spark runtimes/libraries

- Per interagire con i file/tabelle Lakehouse per eseguire analisi, trasformazioni o elaborazioni utilizzando Spark, ti connetti all'endpoint per Lakehouse. Proprio come i metodi standard esterni a Fabric per interagire con file/tabelle delta, ti connetti utilizzando l'URL, il percorso ABFS o connettendo Lakehouse direttamente in Explorer.
- L'utilizzo di Spark consente di eseguire operazioni di scrittura scegliendo tra Scala, PySpark, Spark SQL o R.
- Tuttavia, se desideri utilizzare T-SQL, dovrai utilizzare SQL Analytics Endpoint in cui puoi eseguire solo operazioni di lettura.

SQL Analytics Endpoint per Lakehouse

- Questo endpoint fornisce un'esperienza basata su SQL per le tabelle delta Lakehouse. Fornisce la possibilità di interagire, eseguire query e fornire dati all'interno di Lakehouse utilizzando SQL. Questa esperienza offre la sola lettura e si applica solo alle tabelle delta.
- Questo endpoint utilizza solo le "Tabelle" del Lakehouse mentre la sezione "File" non è individuabile tramite l'endpoint SQL. SQL Analytics Endpoint consente di analizzare tabelle delta utilizzando T-SQL, salvare funzioni, generare visualizzazioni e applicare la sicurezza a livello di oggetto SQL. Ti consente di creare un livello relazionale sopra i dati fisici in Lakehouse ed esporlo a strumenti di analisi e reporting utilizzando la stringa di connessione SQL.
- Un endpoint di SQL Analytics viene creato automaticamente quando si crea un Lakehouse, che punta all'archiviazione di tabelle delta.
- Ogni Lakehouse ha un solo endpoint SQL e ogni workspace può avere più di un Lakehouse. Ciò significa che il numero di endpoint SQL in uno workspace corrisponde al numero di Lakehouse.

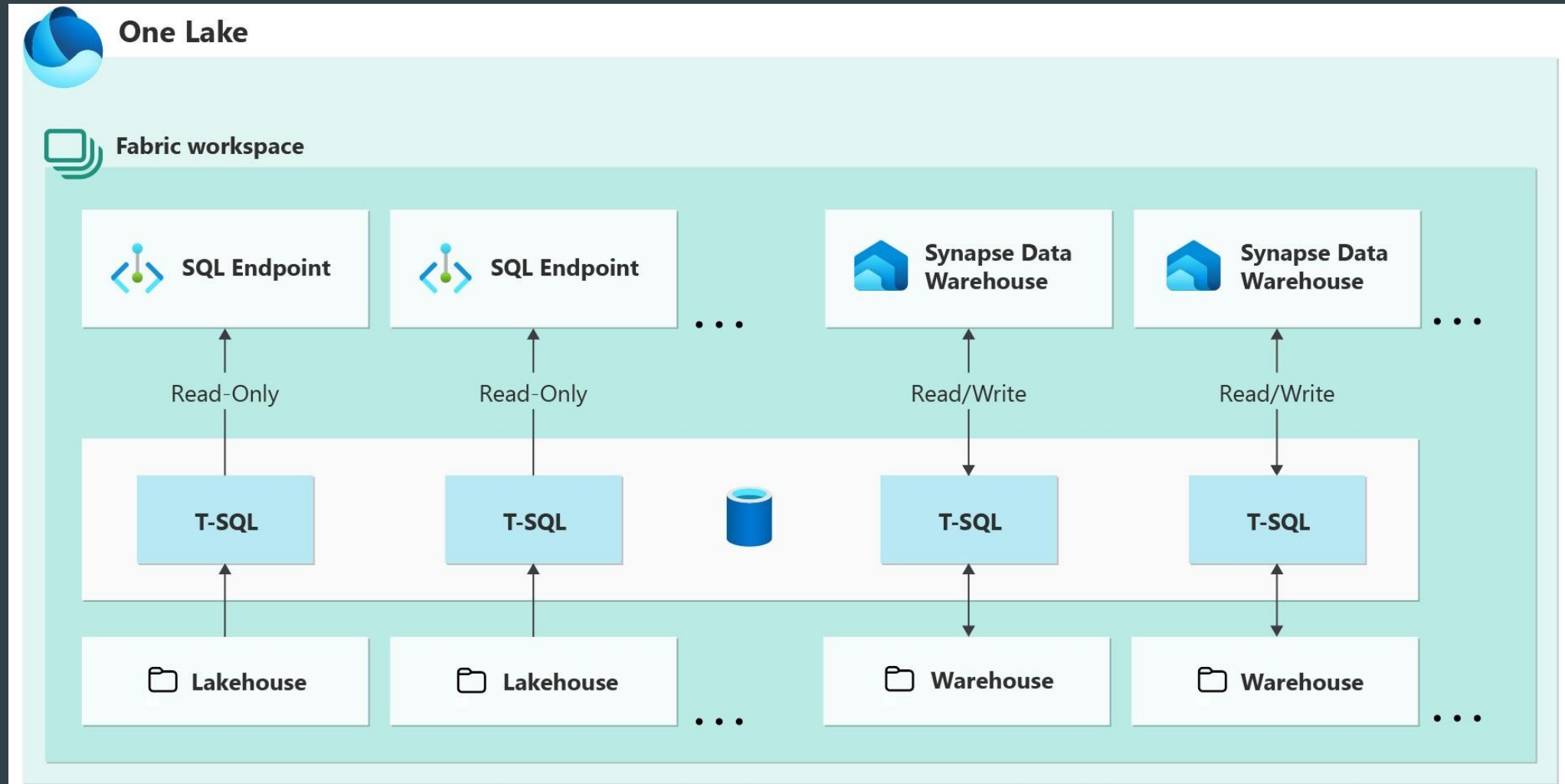
SQL Analytics Endpoint for Lakehouse

- È importante notare che la creazione/modifica delle tabelle delta deve essere fatta utilizzando Apache Spark, le tabelle Delta create tramite Spark all'interno di Lakehouse sono automaticamente rilevabili tramite l'endpoint.
- È possibile impostare RLO per l'accesso ai dati, ma le regole di sicurezza verranno applicate solo per l'accesso ai dati tramite SQL Analytics Endpoint. Ciò significa che se vuoi assicurarti che i tuoi dati non siano accessibili in altri modi, devi impostare ruoli e autorizzazioni nel workspace.
- Puoi anche connetterti a questo endpoint dall'esterno di Fabric con strumenti come SSMS o Azure Data Studio fornendo l'autenticazione e la stringa di connessione dell'endpoint. Dietro le quinte, il SQL endpoint utilizza lo stesso motore dell'endpoint Warehouse per fornire query SQL a prestazioni elevate e a bassa latenza. Ciò significa che è anche un endpoint TDS (Tabular Data Stream), solo con restrizioni sulla funzionalità DML/DDI e limitazioni sul T-SQL.

Data Warehouse Endpoint

- Questo endpoint funziona in modo tradizionale come il data warehouse SQL. Ciò significa che **supporta tutte le funzionalità T-SQL**.
- A differenza dell'endpoint SQL Analytics può leggere e scrivere tabelle delta parquet ma con Spark si può solo leggere con i comandi T-SQL possiamo anche scrivere.
- Disponiamo del supporto completo DML e DDL T-SQL, incluso l'inserimento, la modifica e lo sviluppo dei dati tramite TSQL o l'interfaccia utente. È possibile utilizzare COPY INTO, pipeline, flussi di dati o insert tra database utilizzando CREATE TABLE AS SELECT (CTAS), INSERT..SELECT o SELECT..INTO.
- **La cosa più importante è la completa conformità alle proprietà e alle transazioni ACID con il supporto per transazioni multitabella.**
- Quando inserisci i dati nel data warehouse, una tabella Delta verrà creata e archiviata in OneLake.

Recap



REQUISITI PER L'ETL o ELT

Requisiti per l' ETL o ELT

- ACID Transaction Compliance
- Multi-table transactions
- Dynamic data masking
- Real-time analytics

Requisiti per l' ETL o ELT

- Lakehouse ACID Compliance

- Le funzionalità di transazione ACID sono SOLO per le tabelle in formato Delta.
- Ciò significa che è necessario utilizzare le tabelle Delta gestite per avere la funzionalità ACID.
 - Le tabelle Delta estendono i file parquet con un registro delle transazioni basato su file per le transazioni ACID.
 - Puoi avere supporto ACID per le tabelle Delta con il motore Spark o SQL.
- Le tabelle non Delta e tutti gli altri file, non hanno il supporto ACID in Lakehouse.

- Data Warehouse ACID Compliance

- Le transazioni ACID sono completamente supportate nel Data Warehouse per tutte le tabelle.
- Tutte le tabelle sono tabelle Delta archiviate in OneLake come file delta parquet con un registro delle transazioni basato su file.

Requisiti per l' ETL o ELT

Le transazioni su più tabelle rappresentano un modo per raggruppare le modifiche apportate a più tabelle in un'unica transazione. Ciò consente di controllare il commit o il rollback delle query di lettura e scrittura e di modificare i dati archiviati nelle tabelle utilizzando le transazioni per raggruppare insieme le modifiche

- Lakehouse multi-table transaction:
 - Non supportate
- Data Warehouse multi-table transaction:
 - Supportate

Requisiti per l' ETL o ELT

- Data Warehouse data masking:
 - Completamente supportato..
 - Oltre al mascheramento dinamico dei dati, il data warehouse supporta autorizzazioni granulari di SQL, sicurezza a livello di colonna e riga e log di audit
- Lakehouse data masking:
 - Supportato solo tramite l'endpoint di analisi SQL del Lakehouse. File o motori diversi da SQL (Es. Spark) non saranno in grado di utilizzare il Dynamic Data Masking.
 - È inoltre possibile applicare la sicurezza a livello di oggetto e di riga, ma solo tramite SQL Analytics Endpoint.
 - I motori Spark non saranno in grado di utilizzare la sicurezza a livello di oggetto e riga.
 - Ciò potrebbe cambiare in futuro tramite OneLake Security!!!!.

Requisiti per l' ETL o ELT

- **Data Warehouse real-time:**
 - Il Data Warehouse non è una soluzione ideale per l'analisi in tempo reale.
 - I requisiti di dimensione, formato e latenza dei dati non rendono il data warehouse un'opzione per le soluzioni Real-Time Analytic.
 - Tuttavia, puoi importare i dati dalla tua soluzione di analisi in tempo reale nel data warehouse per ulteriori analisi o combinarli con i dati esistenti per ottenere ulteriori analisi.
- **Lakehouse real-time:**
 - Lakehouse potrebbe essere un'opzione per l'analisi in tempo reale, ma dipenderà dai requisiti della soluzione.
 - **Se hai milioni di transazioni al secondo con solo 100-500 righe di dati per transazione, Lakehouse potrebbe essere in grado di supportare** le esigenze della soluzione Real-Time Analytics. Tuttavia, se questi milioni di transazioni al secondo hanno centinaia di migliaia o milioni di righe per transazione, Lakehouse non sarà in grado di farlo.
 - Dipenderà anche dalla concorrenza e dalle dimensioni dei dati per determinare se Lakehouse sarà in grado di soddisfare i requisiti di bassa latenza e prestazioni elevate.

COMPARAZIONE SKILL DEGLI SVILUPPATORI

Comparazione degli skills degli sviluppatori

- Lakehouse

- Sviluppatori con conoscenza e preferenza Spark. Lavorare direttamente con file e/o tabelle Delta per ETL/ELT o altri carichi di lavoro. Ciò include l'utilizzo dei notebook Spark e delle definizioni dei processi Spark.
- Sviluppatori o utenti con solo conoscenze di T-SQL che possono leggere solo i dati selezionati per loro (tramite SQL Analytics Endpoint) per il consumo o l'analisi.

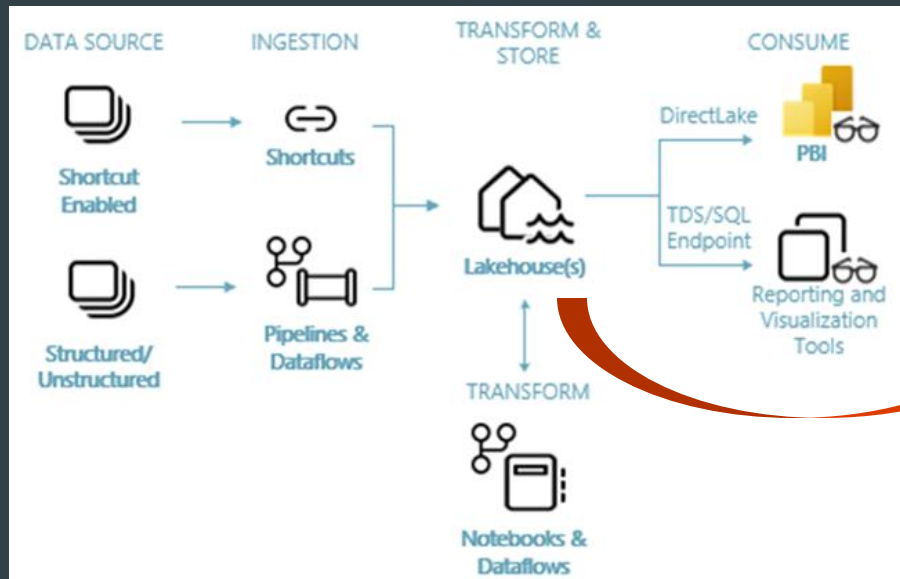
- Data Warehouse

- Sviluppatori con conoscenza e preferenza di T-SQL durante la creazione di un ETL/ELT. Esperti di data warehousing che lavorano con store procedure, funzioni e attività DBA.
- Sviluppatori o utenti con solo conoscenza di T-SQL possono leggere solo i dati selezionati per loro per il consumo o l'analisi.

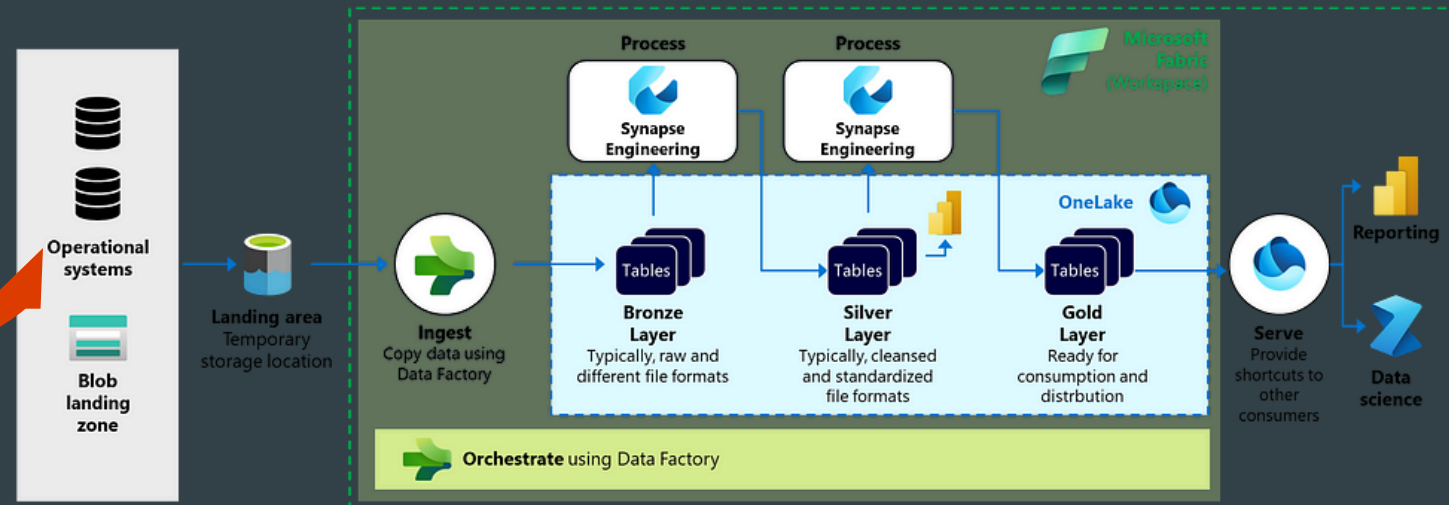
ARCHITETTURE

Meddalion for Lakehouse

- Il set di competenze del team di sviluppatori è principalmente Spark.
- Non sono necessarie funzionalità aggiuntive di data warehouse come transazioni multitabella e mascheramento dinamico dei dati.
- Non è necessario l'endpoint del data warehouse per le funzionalità richieste da alcuni strumenti di reporting di terze parti.
- La funzionalità DDL/DML T-SQL non è richiesta dagli sviluppatori.

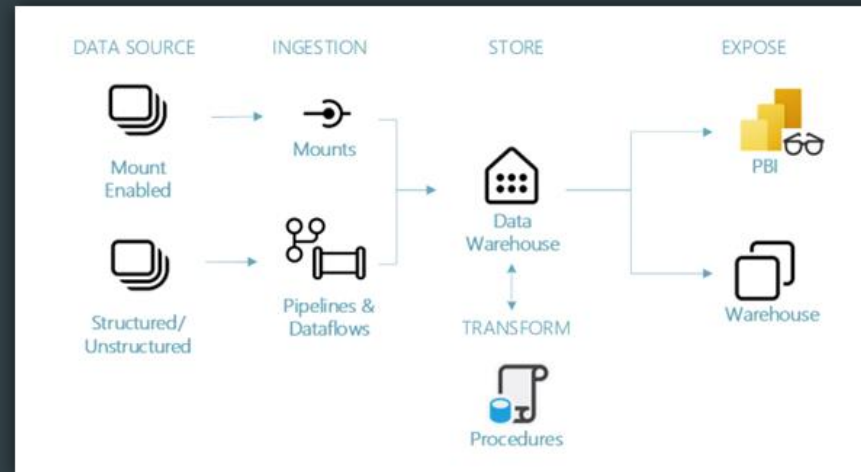


Reference architecture based on Lakehouse



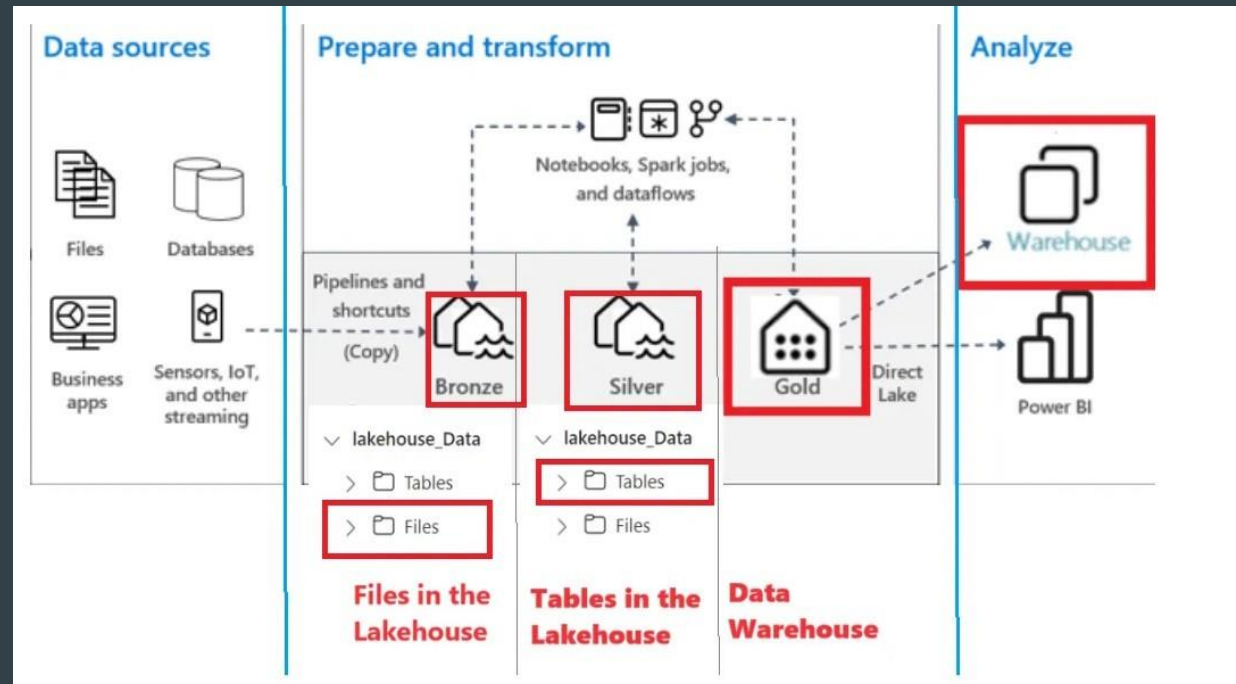
Data warehouse

- Il set di competenze del team di sviluppatori è principalmente T-SQL/Data Warehousing. Trasformazione dei dati con store procedure
- È necessario supportare funzionalità come le transazioni multitabella che sono supportate solo dal data warehouse.
- Il consumo richiede un endpoint di tipo data warehouse per funzionalità non disponibili con l'endpoint Lakehouse SQL richiesto per strumenti o processi di reporting di terze parti.
- E' richiesta dagli sviluppatori la funzionalità DDL/DML attraverso T-SQL. Il carico di lavoro richiede che gli utenti siano in grado di modificare i dati anche dopo che sono stati normalizzati o trasformati.

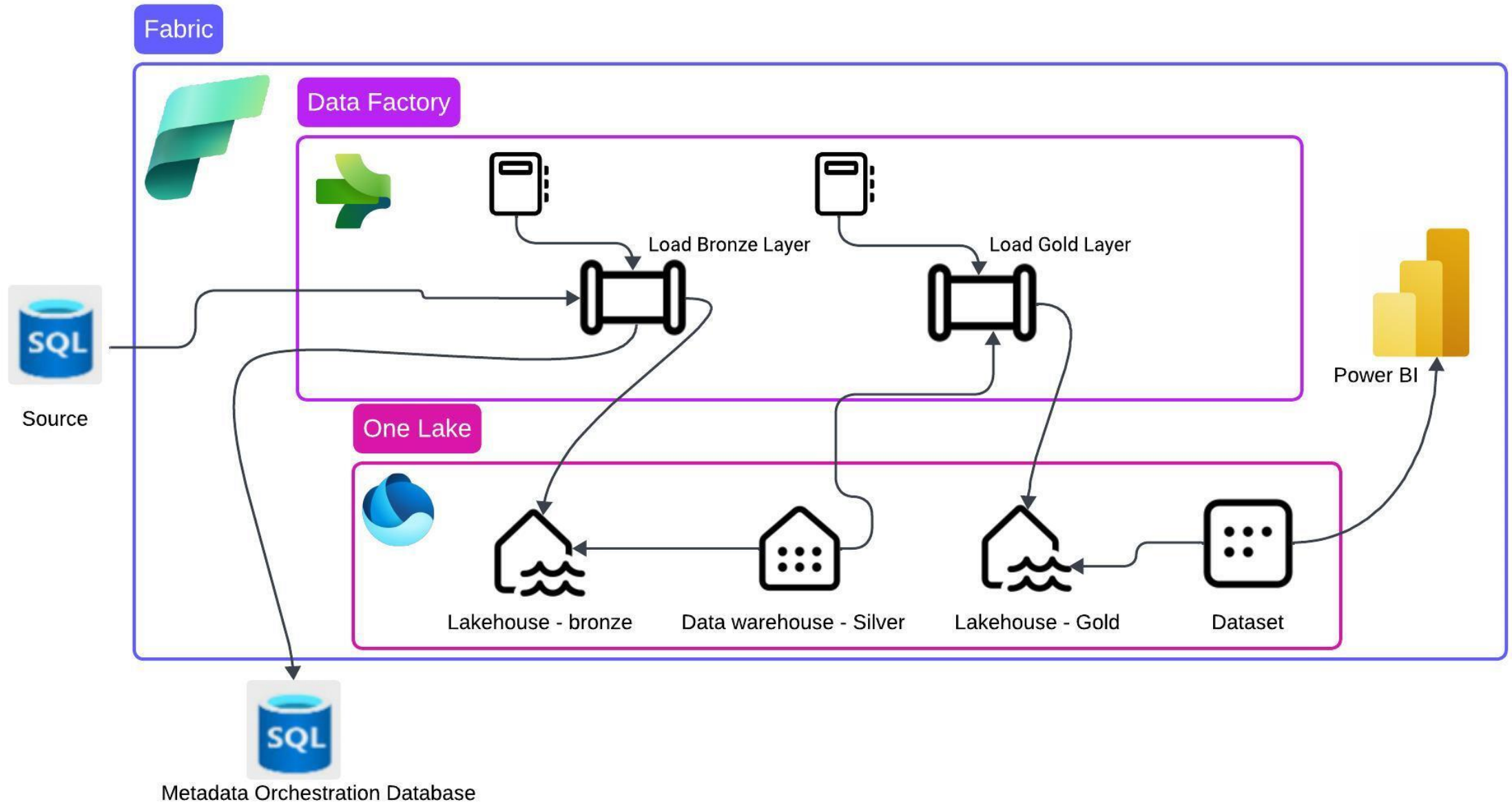


Data warehouse e Lakehouse

- Il set di competenze del team di sviluppatori è principalmente Spark. Trasformazione dei dati con i notebook Spark
- È necessario usare le funzionalità DDL/DML dell'utente finale tramite T-SQL, che è supportata solo dal data warehouse. **Ideale se il carico di lavoro richiede che gli utenti siano in grado di modificare i dati anche dopo che sono stati normalizzati o trasformati.**
- Il consumo richiede l'endpoint del data warehouse per alcuni strumenti o processi di reporting di terze parti.

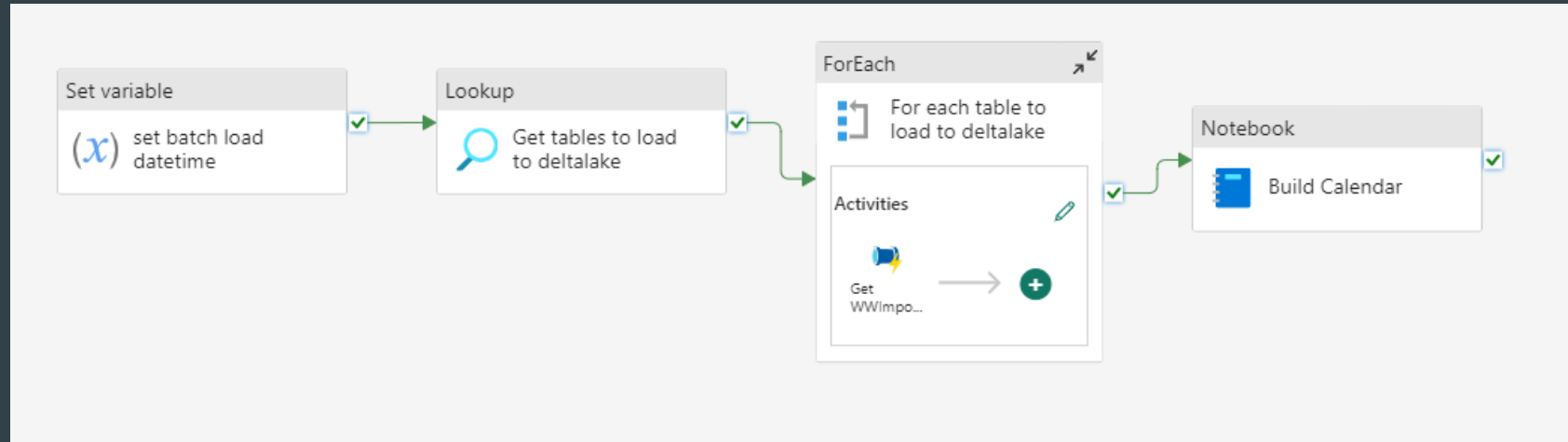


Pattern 1



ORCHESTRATOR

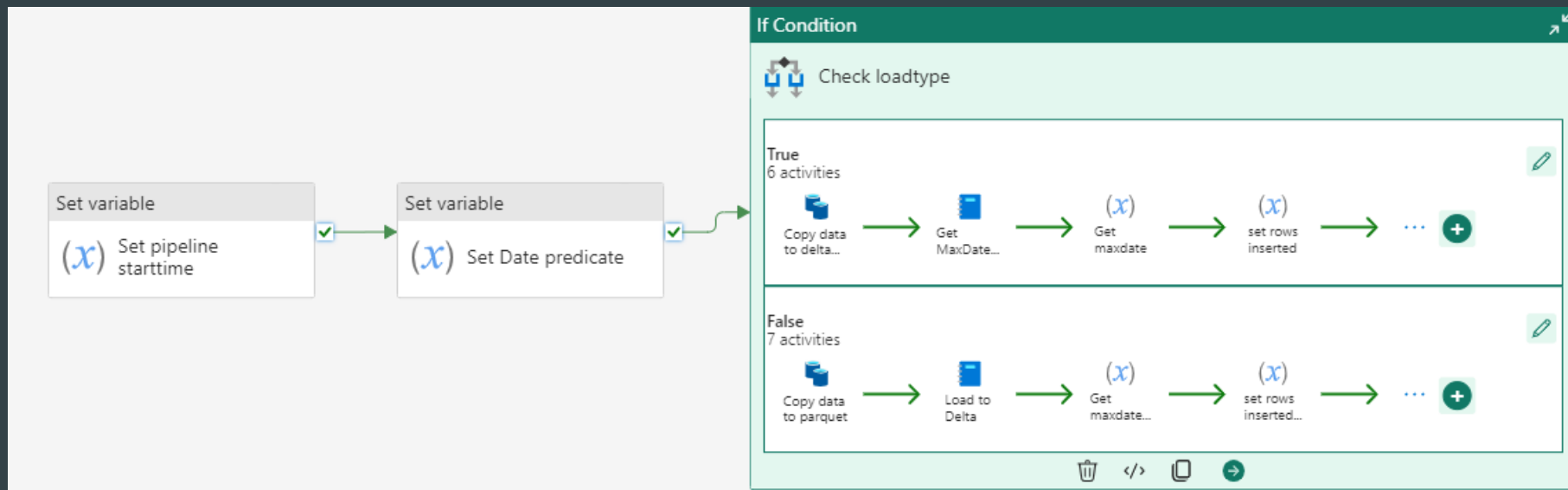
Orchestrator pipeline "Load Source to Fabric" (first part)



| Name | Type | Default | Description |
|------------|------|---------|--|
| startyear | int | 2013 | Year to start loading from Source |
| endyear | int | 2025 | Year to end loading from Source |
| loaddwh | int | 0 | Set to 1 if you want to load to Fabric Data Warehouse |
| loadgoldlh | int | 1 | Set to 1 if you want to load to Fabric Gold Lakehouse |
| loadbronze | int | 1 | Set to 1 if you want to load to Fabric Bronze Lakehouse |
| waittime | int | 300 | Delay needed for tables to materialize in Bronze Lakehouse before loading to DW or Gold LH. Can set to 1 if not loading Bronze or only loading to Bronze |

BRONZE LAYER

Load pipeline “Load Source to Bronze”

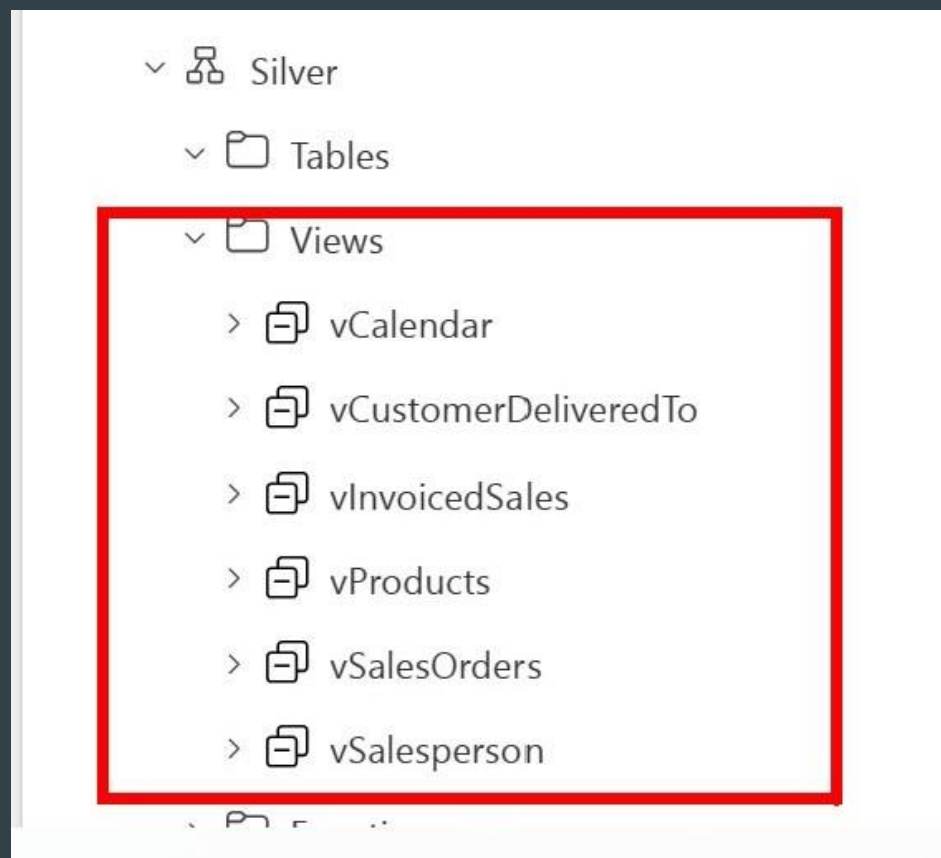


| Name | Type | Description |
|---------------------|--------|---|
| sqlsourcedatecolumn | String | Column name on which it will calculate the maximum |
| sqlstartdate | String | will be updated to the max date of each column indicated in the sqlsourcedatecolumn for each table. This means that if you run the pipeline again without resetting the sqlenddate , no new data will be added to the tables that are incrementally loaded |
| sqlenddate | String | if you run the pipeline again without resetting the sqlenddate , no new data will be added to the tables that are incrementally loaded |
| sqlsourceschema | String | Schema of table on which it will calculate the maximum |
| sqlsourcetable | String | Name of table on which it will calculate the maximum |
| sinktablename | String | |
| loadtype | String | Incremental / full |
| sourcekeycolumn | String | |
| batchloadaddatetime | String | |

SILVER LAYER

Creazione delle viste SQL per il livello silver

- Creeremo le viste SQL nel Fabric Data Warehouse "dwh"

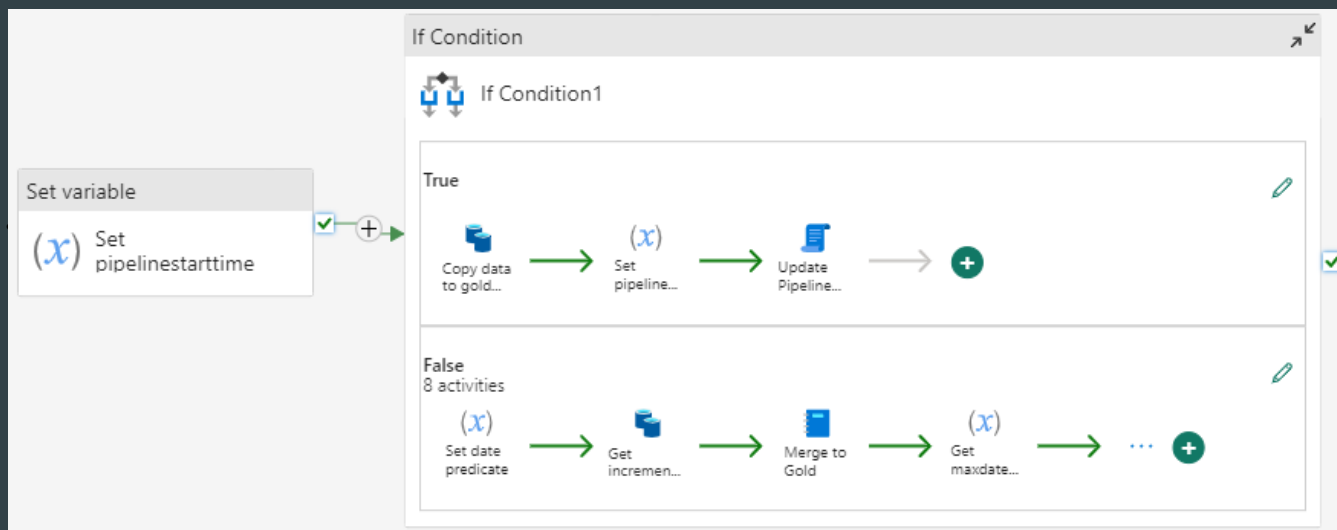


GOLD LAYER (Pattern 1)

Load pipeline “Load Silver to Gold” con lakehouse

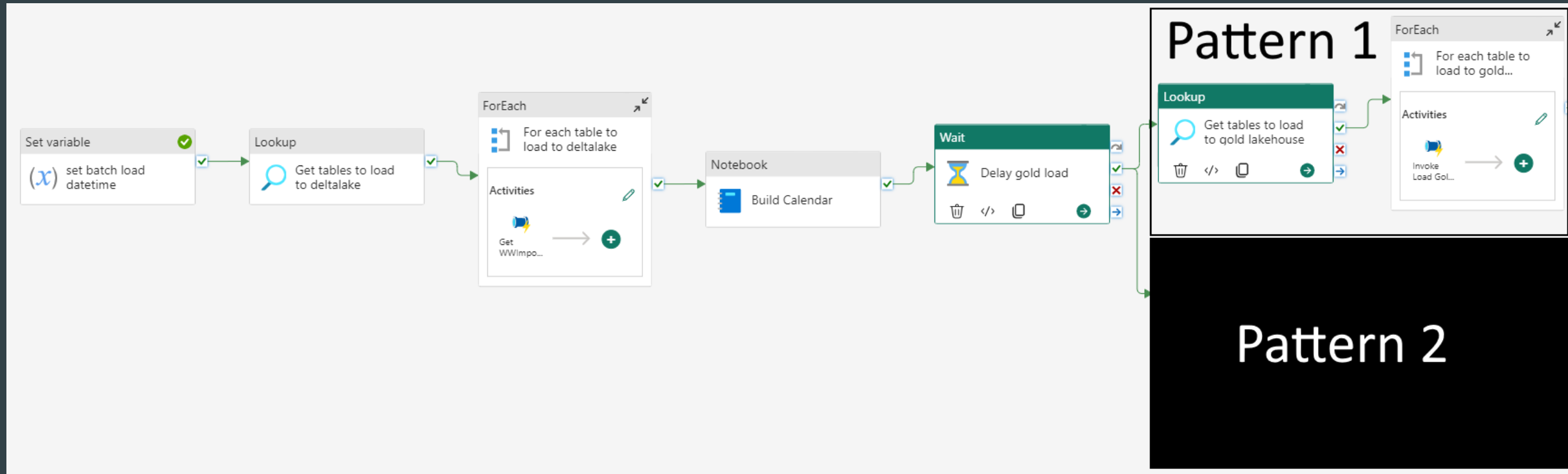


Load Lakehouse Table - Pattern 1



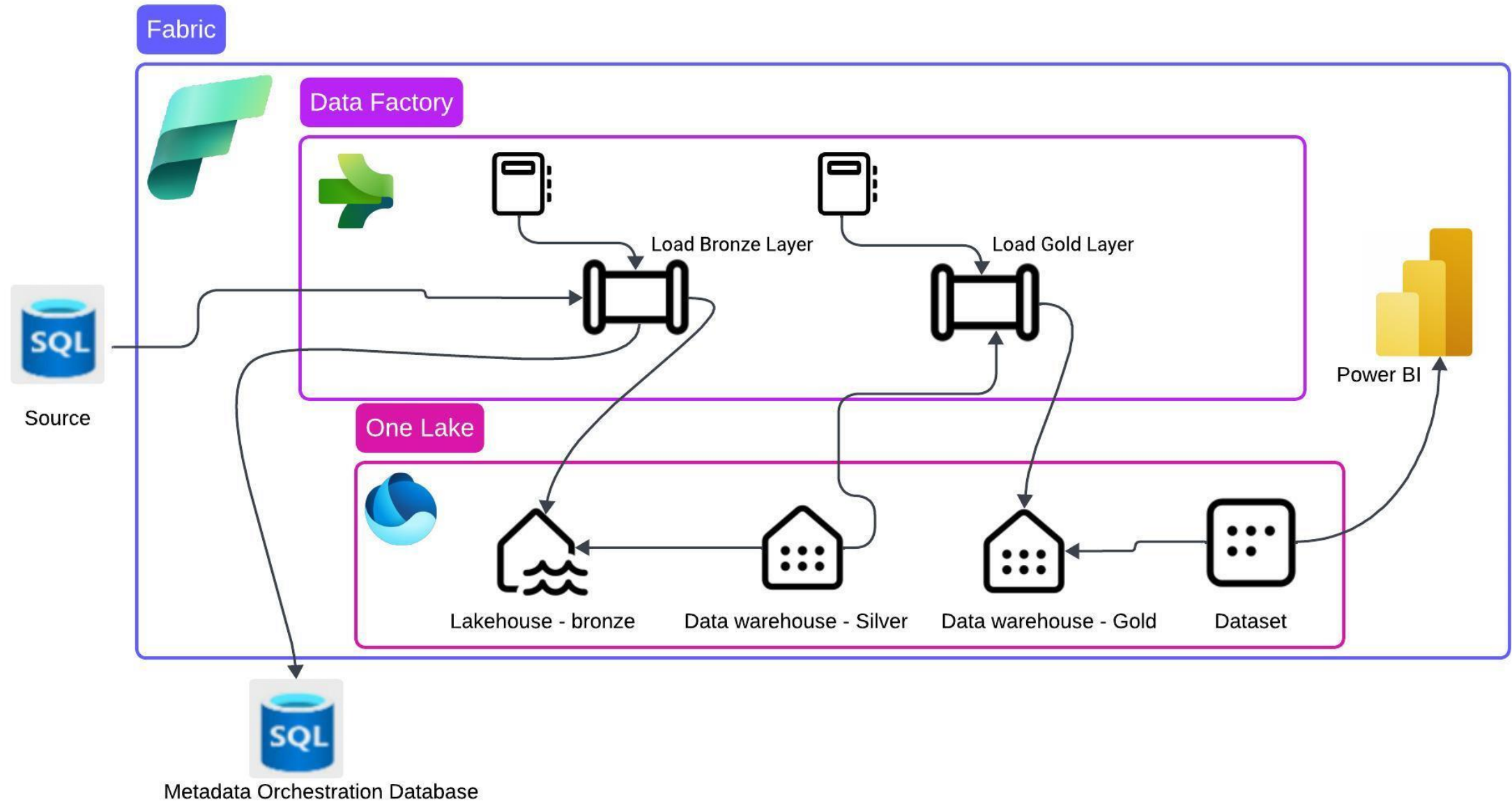
| Parameters | | | | Variables | Settings | Output |
|--------------------------|-------------------|--------|---------------|-----------|----------|--------|
| <input type="checkbox"/> | Name | Type | Default value | | | |
| <input type="checkbox"/> | sourcestartdate | String | Value | | | |
| <input type="checkbox"/> | sourceenddate | String | Value | | | |
| <input type="checkbox"/> | sourceschema | String | Value | | | |
| <input type="checkbox"/> | sourcetable | String | Value | | | |
| <input type="checkbox"/> | loadtype | String | Value | | | |
| <input type="checkbox"/> | batchloaddatetime | String | Value | | | |
| <input type="checkbox"/> | sinktable | String | Value | | | |
| <input type="checkbox"/> | tablekey | String | Value | | | |
| <input type="checkbox"/> | tablekey2 | String | Value | | | |

Orchestrator pipeline integration

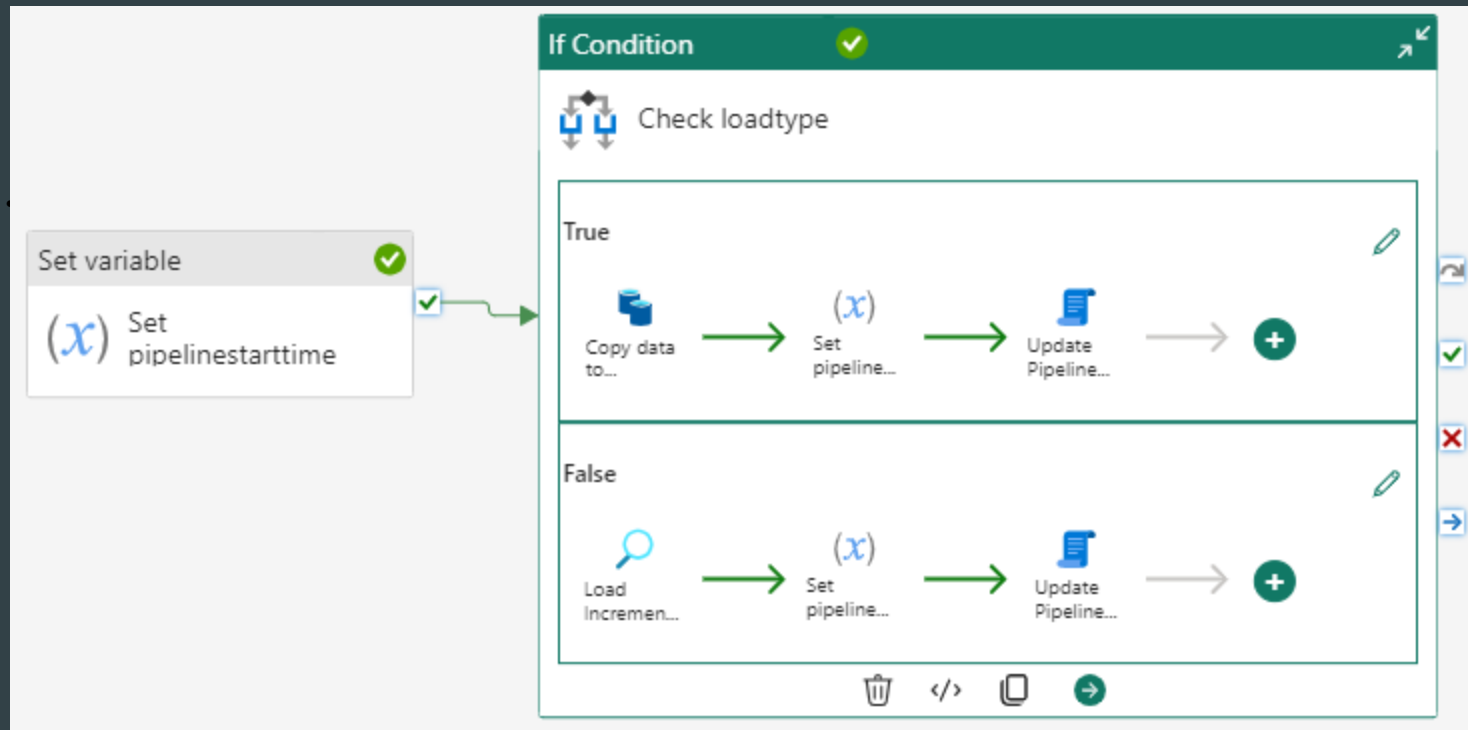


GOLD LAYER (Pattern 2)

Pattern 2














Load pipeline “Load Silver to Gold” data warehouse



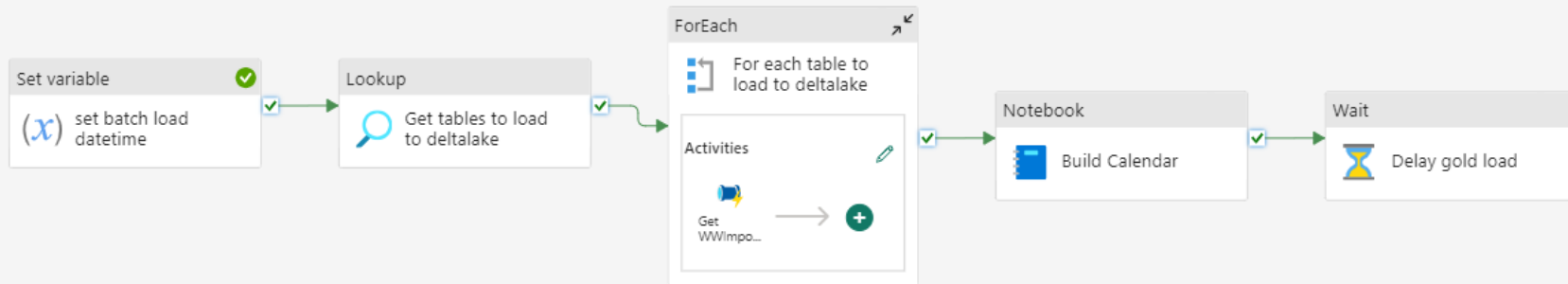
Parameters Variables Settings Output

+ New

 Delete

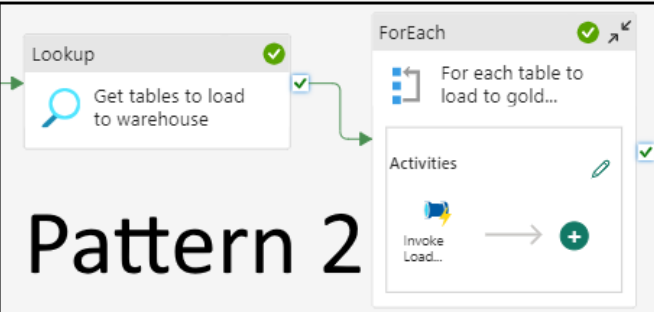
| <input type="checkbox"/> | Name | Type | Default value | |
|--------------------------|--|-------------------------------------|------------------------------------|---|
| <input type="checkbox"/> | <input type="text" value="sourcestartdate"/> | <input type="text" value="String"/> | <input type="text" value="Value"/> |  |
| <input type="checkbox"/> | <input type="text" value="sourceenddate"/> | <input type="text" value="String"/> | <input type="text" value="Value"/> |  |
| <input type="checkbox"/> | <input type="text" value="sourceschema"/> | <input type="text" value="String"/> | <input type="text" value="Value"/> |  |
| <input type="checkbox"/> | <input type="text" value="sourcetable"/> | <input type="text" value="String"/> | <input type="text" value="Value"/> |  |
| <input type="checkbox"/> | <input type="text" value="sinkschema"/> | <input type="text" value="String"/> | <input type="text" value="Value"/> |  |
| <input type="checkbox"/> | <input type="text" value="loadtype"/> | <input type="text" value="String"/> | <input type="text" value="Value"/> |  |
| <input type="checkbox"/> | <input type="text" value="batchloaddatetime"/> | <input type="text" value="String"/> | <input type="text" value="Value"/> |  |
| <input type="checkbox"/> | <input type="text" value="sinktable"/> | <input type="text" value="String"/> | <input type="text" value="Value"/> |  |
| <input type="checkbox"/> | <input type="text" value="storedprocschema"/> | <input type="text" value="String"/> | <input type="text" value="Value"/> |  |
| <input type="checkbox"/> | <input type="text" value="storedprocname"/> | <input type="text" value="String"/> | <input type="text" value="Value"/> |  |

Orchestrator pipeline integration



Pattern 1

Pattern 2



Domande?

Ringraziamenti

Gruppo FTA

Key Contacts

Samarendra Panda

Neeraj Jhaveri

Andrés Padilla

Ben Harding

Jean Hayes

Thiago Rotta

GitHub ID

@Sam-Panda

@neerajjhaveri

@AndresPad

@BennyHarding

@jehayesms

@rottathiago

Email

sapa@microsoft.com

neeraj.jhaveri@microsoft.com

andres.padilla@microsoft.com

ben.harding@microsoft.com

jean.hayes@microsoft.com

thiago.rotta@microsoft.com