Analytics in Azure – week 2

In de tweede week van het leertraject Analytics in Azure gaan we dieper in op het verwerken van batch-data in de cloud. We hebben het gehad over onderwerpen als:

* Delta Lake architecturen
* Orchestratie
* ETL, ELT en moderne data-laadvarianten

Tooling-wise maken we kennis met diverse tools in het Azure-platform die kunnen helpen bij de batch-verwerking van data:

* Azure Data Factory (ADF) voor orchestratie
* Azure Synapse voor een “totaaloplossing” met:
  + Pipelines als geïntegreerde ADF
  + Serverless SQL Pools als SQL-queries die je direct op je Data Lake kunt uitvoeren
  + SQL Pools als een horizontaal schaalbaar SQL Data Warehouse
  + Spark Pools als geïntegreerde Big Data-oplossing voor alles waar Spark voor gebruikt kan worden
    - Interactieve queries (Spark SQL)
    - Machine Learning (Spark MLib)
    - Graph computation (Spark GraphX)
    - .. en de Python- en Scala- interfaces
* Azure Databricks als “Managed Spark” met heel veel handige verbeteringen en integraties
* Azure SQL Database wanneer je een klein of voordelig Data Warehouse in de cloud nodig hebt (hier hebben we het vorige week over gehad)

# Oefeningen

In deze week kun je de volgende oefeningen doen om meer ervaring op te doen. Aan te raden is om één kant te kiezen en die uit te diepen. Houd daarbij rekening met je beschikbare tijd, startniveau en wat voor jouw organisatie / klant / PoC de meest interessante techniek is.

In alle gevallen geldt: zorg ervoor dat je een stuk ETL ontwikkelt, waarbij data getransformeerd en/of geïntegreerd wordt. Waar het kan: neem een eenvoudige dataset uit je organisatie als voorbeeld. Als dat niet lukt: neem bijvoorbeeld de AdventureWorks-database als basis, en zorg dat deze in twee stappen verder ingeladen wordt:

1. Ingest (laden “as-is” naar Data Lake, bijvoorbeeld in Parquet-formaat)
2. Process (verwerken van data in je Data Lake naar een andere “laag”)

Over de aanpak:

* De “ingest” is het eenvoudigst met Azure Data Factory te doen – het *copy* statement is een heel krachtige manier
* Wanneer je voor je PoC data van on-prem wilt laden (dit kan met een self-hosted integration runtime in ADF), voeg dit dan later toe: zorg dat eerst de structuur staat
* De “Process” kun je op diverse manieren aanpakken:
  + Grafisch: Met ADF (of Synapse Pipelines) een Data Flow inrichten. Niet vergeten uit te zetten!
  + In DataBricks: gebruik DataBricks SQL of DataBricks Python-codes om de data te laden en transformeren
  + In Synapse Serverless SQL: SQL-code schrijven om data te transformeren en terug te schrijven naar het Data Lake
  + In Synapse SQL Pools: SQL-code schrijven om data te transformeren en te schrijven naar een SQL-cluster
* Wat betreft de locatie: Zowel Databricks als ADF kunnen de data zowel naar het Data Lake (tip: delta formaat) als naar Synapse SQL Pools efficiënt wegschrijven. Voor kleinere workloads is Azure SQL Database hier ook erg geschikt.

Hieronder vind je per tool nog wat extra materiaal voor naslag of verder leren:

## Azure Data Factory

Voor Azure Data Factory hebben we twee tutorials waarmee je meer kennis maakt ADF:

[Create a data factory pipeline](https://docs.microsoft.com/en-us/azure/data-factory/tutorial-copy-data-portal) helpt je om de basics nog eens te doorlopen: wat is er allemaal nodig om ADF simpelweg te gebruiken om data te verplaatsen? Prettig is dat hier alle diensten nog eens de revue passeren.

[Delta lake ETL with data flows](https://docs.microsoft.com/en-us/azure/data-factory/tutorial-data-flow-delta-lake) gaat een stapje verder, door het gebruiken van het Delta Lake formaat.

Voor alle genoemde formaten kun je ten slotte kijken hoe Power BI hiermee verbindt:

* Kun je Parquet-bestanden inladen in Power BI?
* Kun je Delta Lake bestanden inladen in Power BI?

Als je wat meer “los” bent op Power Query kun je ook eens nadenken over / kijken naar de impact van deze formaten: is *query folding* bijvoorbeeld nog steeds mogelijk?

## Databricks

In de video [Getting Started with Databricks SQL - YouTube](https://www.youtube.com/watch?v=OMjxlqIqSqs) laat Simon Whiteley zien hoe je Data Warehousing in Databricks kunt “regelen”. In de [Tutorial - Perform ETL operations using Azure Databricks | Microsoft Docs](https://docs.microsoft.com/en-us/azure/databricks/scenarios/databricks-extract-load-sql-data-warehouse) wordt dit verder uitgewerkt op een praktijkgerichte basis.

Ook kun je het volgende Databricks *notebook* importeren:

* Klik binnen Databricks op Workspace -> Users -> jouw username
* Kies hier het pijltje dat naar beneden wijst naast je naam, kies *import*
* Geef aan dat je het Notebook wilt downloaden van het volgende adres:  
  <https://github.com/MicrosoftDocs/mslearn-perform-basic-data-transformation-in-azure-databricks/blob/master/DBC/05.1-Basic-ETL.dbc?raw=true>

Zin om nóg een tandje dieper te gaan? Er is ook een “advanced transformations” notebook beschikbaar dat je kunt importeren:

<https://github.com/MicrosoftDocs/mslearn-perform-advanced-data-transformation-in-azure-databricks/blob/master/DBC/05.2-Advanced-ETL.dbc?raw=true>

## Synapse Serverless SQL

Microsoft zet Synapse Serverless SQL momenteel vooral in de markt voor “Exploratory Data Analysis”. Het is echter ook mogelijk om logische Data Warehouses te maken met behulp van Synapse.

Srini Velamakantie laat in zijn blog [Data Lakehouse Architecture — Azure Synapse Serverless SQL Pools | by Srini Velamakanti | Analytics Vidhya | Medium](https://medium.com/analytics-vidhya/data-lakehouse-architecture-azure-synapse-serverless-sql-pools-78a7a142ca37) zien hoe dat er qua ontwerp uitziet.

Wanneer je voor jouw PoC hier een stuk van wilt gebruiken, dan heeft hij ook een tweede artikel geschreven met daarin de technische “tussenstappen” uitgeschreven: [Build a new-age Cost-Effective Business Intelligence System using Azure Data Lakehouse and Synapse Serverless SQL Pool | by Srini Velamakanti | Analytics Vidhya | Medium](https://medium.com/analytics-vidhya/build-a-new-age-cost-effective-business-intelligence-system-using-azure-data-lakehouse-and-synapse-46d361ba33e6)

## Synapse Spark

Synapse Spark biedt met name voordelen wanneer je Machine Learning-modellen wilt integreren in je dataverwerking. Het operationaliseren van ML modellen is in diverse (grotere) organisaties een belangrijke rol voor Data Engineers. Wanneer dit in jouw organisatie niet het geval is, is een PoC in een andere techniek waarschijnlijk nuttiger.

Als je hiermee aan de slag wilt, heeft Microsoft een tutorial die zich specifiek richt op *data wrangling* met Spark: [Data wrangling with Apache Spark pools (preview) - Azure Machine Learning | Microsoft Docs](https://docs.microsoft.com/en-us/azure/machine-learning/how-to-data-prep-synapse-spark-pool). Deze tutorial staat sterk in de context van Machine Learning Operations (MLOps).

## Synapse SQL Pools

Voor grote DW-oplossingen is SQL Pools de aangewezen Microsoft-tool binnen het Azure-platform. Je kunt de data op diverse manieren in je SQL Pool krijgen. In drie artikelen wordt hier meer over uitgelegd:

* [Ingest data into a dedicated SQL pool - Azure Synapse Analytics | Microsoft Docs](https://docs.microsoft.com/en-us/azure/synapse-analytics/data-integration/data-integration-sql-pool) legt uit hoe je data de SQL Pool in krijgt met behulp van Synapse Pipelines
* [Quickstart: Bulk load data using a single T-SQL statement - Azure Synapse Analytics | Microsoft Docs](https://docs.microsoft.com/en-us/azure/synapse-analytics/sql-data-warehouse/quickstart-bulk-load-copy-tsql) geeft je een quick start om met behulp van het COPY statement (Synapse-specifiek) data eenvoudig in te laden
* [Quickstart: Bulk load data with a dedicated SQL pool - Azure Synapse Analytics | Microsoft Docs](https://docs.microsoft.com/en-us/azure/synapse-analytics/quickstart-load-studio-sql-pool) geeft je een tip over de wizard die je kunt gebruiken om het COPY-statement voor je te laten genereren

Wat betreft de snelste manier om data te laden is er momenteel wat onduidelijkheid. In het verleden waren er “best practices” over het laden met COPY of CREATE TABLE AS SELECT – manieren die logischerwijs aansloten bij een parallelle manier van data verwerken. Hier wordt in de huidige documentatie nauwelijks meer over gesproken. En het zou goed te verklaren zijn als Microsoft deze logica over het slim aanpakken van dataloads inmiddels verwerkt heeft in een “regulier” INSERT..SELECT statement.

Wanneer je hier op schaal mee wilt gaan werken, is het dus zaak om dit te testen. Gelukkig kan dat vrij eenvoudig in de cloud.

# Verdieping en verbreding

Voor een stukje verdieping en verbreding zou je ook kunnen kijken naar de zogenaamde *wrangling* activity. Hiermee krijg je feitelijk Power Query beschikbaar op grote schaal (let wel: het is nog vrij beperkt in vergelijking met wat je van Power Query gewend bent!)

[Getting started with wrangling data flow in Azure Data Factory - Azure Data Factory | Microsoft Docs](https://docs.microsoft.com/en-us/azure/data-factory/wrangling-tutorial)