



Q&A Webinar Datagedreven werken



Verschil ELT en ETL

Vraag

Wat is het verschil tussen ELT en ETL, in welke situatie toepassen. Ik ben benieuwd naar de standaard aanpak!

Antwoord

ELT staat voor Extract, Load en Transform. Dit betekent (bijna letterlijk): ik haal alle data op, zorg dat deze gestructureerd klaarstaat en transformeer de data daarna naar behoefte. Dit is tegenwoordig de absolute best practice voor data. Concreet creëer je hiermee een dataverwerkingsproces dat multipurpose ingezet kan worden. Je kunt bijvoorbeeld dezelfde data gebruiken voor rapportages, AI-agents of Robotic Process Automation (RPA).

ELT is in feite een doorontwikkeling van het vroegere ETL. Omdat data vroeger vrijwel alleen voor rapportages werd gebruikt (en vaak via OLAP-cubes; de techniek daarachter laten we voor nu buiten beschouwing), werden deze modellen single-purpose ontwikkeld. Bij ETL haal je de data op, verwerk je deze (semi)direct en stel je de resultaten beschikbaar voor een specifiek doel, zoals rapportages of draaitabellen. Hoewel we ETL nog wel tegenkomen, is het inmiddels een uitstervende oplossing. In een ETL-proces wordt ruwe data vaak slechts tijdelijk opgeslagen, waardoor je veel mogelijkheden verliest. Je houdt in dit geval alleen bewerkte data over, waardoor de opties voor hergebruik beperkt zijn. Toen opslagruimte nog duur was, was dit logisch. Met cloudopslag is het echter in de meeste gevallen een veel beter idee om raw data te bewaren.

Overigens worden de termen ETL en ELT in veel literatuur door elkaar gebruikt. Probeer daarom altijd te achterhalen wat er precies wordt bedoeld. Termen (en zeker afkortingen) zeggen namelijk niet alles.

Best practices

Vraag

Wat zijn de best practices voor het starten binnen MS Fabric?

Kan dit bijvoorbeeld met een heel klein team (3 mensen) snel zelf gedaan worden, om wat interne kennis op dit gebied te vergaren?

Antwoord

Tijdens het webinar hebben we een aantal best practices behandeld. We raden je aan om de opname zeker terug te kijken. In de presentatie kun je deze best practices ook vinden op slides 10 en 18.

Starten met een heel klein team is absoluut mogelijk! Bij Ekco zijn we zelfs begonnen met een team van slechts twee personen. In de basis kun je een Fabric-licentie activeren en beginnen met experimenteren. Het is echter belangrijk om te beseffen dat je meerdere componenten van Microsoft Fabric zult gebruiken (zoals Data Factory voor data-extractie, OneLake voor opslag en PySpark voor data-engineering). Hierdoor ontstaat er wel een steile leercurve, en het kan lastig zijn om de samenhang tussen de verschillende onderdelen te behouden.

Gevoel voor systeemdenken en affiniteit met coderen kunnen je hierbij enorm helpen. Begin vooral klein en probeer niet te veel tegelijk te bereiken. Microsoft Fabric heeft een relatief lage instapdrempel. Een eerste licentie kost 42 cent per uur (ongeveer €308 per maand). Door direct een jaarlicentie af te nemen, kan deze prijs dalen tot €183 per maand. Voor een eerste start is dit echter vaak een wat gedurfde keuze. Daarnaast zijn Power BI-licenties nodig voor het ontwikkelen van rapportages. Er is ook een proefversie van 60 dagen beschikbaar, inclusief één Power BI Premium-licentie. Er zijn daarnaast mogelijkheden voor een combinatie van Power BI Premium en Power BI Pro-licenties, maar dit gaat wat verder dan deze Q&A. Neem gerust contact met ons op om je situatie door te spreken.

Om zelf aan de slag te gaan, is het een goed idee om hier te kijken: [Browse all courses, learning paths, and modules - Training | Microsoft Learn](#). Uiteraard raden we ook aan om een training te volgen bij Kimura. Hiermee kun je sneller van start gaan, omdat je niet alleen ziet hoe alles werkt, maar ook onder begeleiding de eerste stappen zet. Zeker om de impact van je acties goed te begrijpen, is dit een absolute aanrader.

Wat we vaak zien, is dat de eerste stappen relatief snel te zetten zijn zodra je de basisprincipes begrijpt. Wanneer je model groter wordt, je organisatie het intensiever gaat gebruiken, er meer databronnen worden toegevoegd en de afhankelijkheid groeit, wordt een kwaliteitsframework echter cruciaal. Houd er rekening mee dat wat je nu zelf bouwt (zonder framework), vaak niet aansluit op een later geïmplementeerd framework en mogelijk opnieuw moet worden opgebouwd. Wacht daarom niet te lang met het nadenken over een degelijk framework. Dit bespaart je in de toekomst een hoop werk.

OTAP

Vraag

Is het verstandig om voor het ELT proces gebruik te maken van een OTAP omgeving?

Antwoord

OTAP staat voor Ontwikkel, Test, Acceptatie en Productie en wordt gebruikt om ervoor te zorgen dat wijzigingen volledig getest en geaccepteerd zijn voordat deze daadwerkelijk in de praktijk worden toegepast. Dit verhoogt de betrouwbaarheid van je systeem aanzienlijk.

En ja, voor ELT is dit een zeer verstandig idee. Dit geldt eigenlijk voor alle stappen binnen dit proces. Een fout in de extractor, waardoor je data mist of anders binnenkomt dan verwacht, is vaak lastig te herstellen. Dit is vooral problematisch wanneer je afhankelijk bent van data "in het moment". Het testen van wijzigingen voordat je een extractor aanpast, is daarom cruciaal.

Voor de load-stap hangt het meer af van de use case. Vaak zien we dat raw data (de brondata) niet per se meerdere keren wordt opgeslagen, zolang je hierop geen bewerkingen uitvoert. Maar zodra je data gaat transformeren, is een OTAP-werkwijze een absolute best practice. Er zijn verschillende manieren om dit in te richten, zoals het gebruik van aparte zones (of folders), maar je kunt ook werken met Fabric-workspaces. Dit hangt af van de werkwijze binnen je organisatie.

Uiteraard moet je nagaan of je echt een volledige OTAP-structuur nodig hebt of dat je bijvoorbeeld 1 of 2 van deze stappen kunt overslaan. Dit is vooral afhankelijk van je ontwikkelproces. Test- en productieomgevingen zijn wat ons betreft minimale vereisten, maar zeker als je in een team werkt, biedt een ontwikkelomgeving vaak veel extra waarde.

Houd er echter rekening mee dat werken met een OTAP-werkwijze veel discipline en onderhoud vereist. Je test-, ontwikkel- en acceptatieomgeving vragen minimaal evenveel (en vaak meer) beheer dan je productieomgeving. Het opschonen van deze omgevingen na elke test is cruciaal om het werk schoon te houden. Dit gaat helaas vaker fout dan je denkt. Testen op een vervuilde omgeving kan in zeldzame gevallen tot grotere problemen leiden dan het direct uitrollen van wijzigingen in productie. Onderschat dit dus niet.

Connectoren

Vraag

Kunnen wij een lijst krijgen van de reeds 50 kant en klare connectoren?

Antwoord

- | | |
|----------------------------------|------------------------------------|
| 1. NAV 2009 | 18. Yuki |
| 2. AX 2009 | 19. Numbrs |
| 3. AX 2012 | 20. Employes |
| 4. BaanIV | 21. TimeChimp |
| 5. BaanV | 22. AdminPulse |
| 6. Infor ERPLN | 23. Odoo API (SaaS) |
| 7. Dynamics 365 F&O | 24. Odoo lokaal (Postgres) |
| 8. Dynamics 365 CE | 25. Salesforce |
| 9. Dynamics 365 Business Central | 26. Autotask |
| 10. Slimstock / Slim4 | 27. Faster Forwards Elements (CRM) |
| 11. AII Solutions | 28. PowerApps |
| 12. HR2Day | 29. DataVerse |
| 13. Moneybird | 30. SAP (S4 HANA) |
| 14. Exact Online | 31. Netsuite |
| 15. AFAS | 32. SharePoint |
| 16. Simplicite | 33. Yoobi |
| 17. Teamleader | 34. Hubspot |

- 35. Jira
- 36. Excel
- 37. CSV
- 38. SQL Server
- 39. Azure SQL DB
- 40. MySQL
- 41. PostgreSQL
- 42. Oracle database
- 43. Pipedrive CRM
- 44. KNMI weerdata
- 45. Oracle ERP API
- 46. Informer
- 47. JDEdwards
- 48. DataChimp
- 49. Azure Cost Management API
- 50. Power BI Scanner API
- 51. Imprss / Legasi
- 52. CBS Open data StatLine