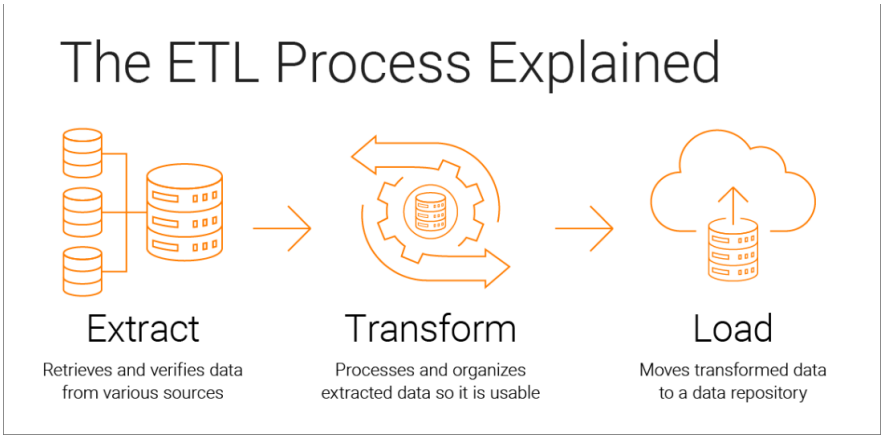
Data integration og ETL

Opgavebeskrivelse

I skal arbejde med at samle data fra forskellige kilder i en database vha. ETL-processer. ETL (Extract, Transform, Load**)** er en metode til at samle og bearbejde data fra forskellige kilder. Processen bruges til at udtrække data (Extract), rense og omforme dem (Transform) og derefter indlæse dem i en samlet database (Load). ETL kan ses som en struktureret måde at konsolidere og rense data på – lidt som det generelle dataforberedelsesarbejde, man laver før analyser, blot mere systematisk og automatiseret.

Se evt. denne model for en visuel fremstilling af, hvordan ETL-processen typisk er opbygget:



(Kilde: https://www.informatica.com/resources/articles/what-is-etl.html)

Case

Forestil dig, at du arbejder som dataanalytiker hos en fiktiv virksomhed, "BikeCorp Inc.", der ønskerat få bedre mulig for at tage data-drevne beslutninger. For at facilitere dette, har du fået til opgave at konsolidere data, der ligger spredt i forskellige kilder med forskellige formater. Data skal hentes fra de nuværende kilder, transformeres og samles i en database. Din chef har fortalt dig, at de gerne vil kigge anderledes på deres data og se om

der er kategorier i datasættene, som er relevante for virksomhedens forretningsanalyse. Derfor har han bedt dig om at lave en løsning for en centraliseret database i kan arbejde i fremover.

Hvis I bliver færdige og har tid, kan I bruge det I har lært i tidligere uger, til at

lave en måde at tilgå data i databasen. Det kunne fx være igennem Python eller PowerBI.

Selve ETL-processen og dokumentation er dog det vigtigste, så det skal være jeres fokus.

Hvis I bliver færdige og har tid, kan I bruge det I har lært i tidligere uger, til at lave en måde at tilgå data i databasen. Det kunne fx være igennem Python eller PowerBI. Selve ETL-processen og dokumentation er dog det vigtigste, så det skal være jeres fokus.

Denne opgave har 3 vigtige læringsmål:

1. **Lære om ETL-processer** – Få praktisk erfaring med at lave ETL i kode, og forstå hvordan man designer dem, fx hvordan man håndterer manglende data og sikrer ensartede dataformater.
2. **Dokumentere beslutninger** – Skriv ned hvilke valg I træffer i ETL-processen, fx ændringer i datatyper eller strukturen af data før og efter.
3. **Kommunikere klart** – Øv jer i at forklare ETL-processen til kollegaer og ledere, så de forstår datagrundlaget for analyser og beslutninger.

Alle tre mål er lige vigtige. Tænk på dem under hele arbejdet, og lav gerne dokumentation løbende.

Afleveringsformat

Opgaven skal ende ud i en præsentation på 10-15 min. I skal forestille jer, at I overleverer resultatet af jeres ETL-process til en leder eller en gruppe kollegaer, der skal bruge den i deres arbejde. Præsentationen skal indeholde:

- Et overblik over den datastruktur I er endt med. Dvs. jeres database-schema, samt eventuelt andre ting I finder relevante, afhængigt af hvilke valg I har taget undervejs.
- En introduktion til jeres dokumentation, så brugere af databasen kan tjekke forskellige ting der kunne være relevante for deres arbejde.
- I kan også komme ind på hvordan I har implementeret processen i kode. Er der ting I var i tvivl om, synes var udfordrende eller problemer I fandt en elegant løsning på? Det kan både være detaljer eller en overordnet struktur I synes fungerede godt/mindre godt. Men I skal ikke gå hele jeres kode igennem i detaljer, find et par nedslag.
- Hvis I har lavet et interface til databasen i Python, PowerBI eller noget andet, kan I også komme ind på det.

Igen skal selve ETL-processen og dokumentationen være i fokus.

Opgavevejledning og opmærksomhedspunkter

Data:

- Der er 3 datakilder som tilgås via en API. I kan tilgå databasen ved at kalde API'et gennem følgende links:
    - https://etl-server.fly.dev/orders
    - https://etl-server.fly.dev/order_items
    - https://etl-server.fly.dev/customers
- Derudover er der en mappe med CSV-filer: "*brands", "categories", "products", "staff", "stocks" og "stores"*

Her er et forslag til en måde at komme igang med opgaven:

- Start med at danne jer et overblik over datakilderne. Prøv at udtænke en plan for hvordan I vil hente og behandle data fra de 3 API'er.

- Prøv at udtænke en strategi for hvordan I vil lave dokumentation løbende. Man kan nemt blive distraheret af selve programmeringsdelen, så det kan være en god idé at have en eksplicit strategi og struktur for dokumentation fra starten.

- Lav et design til jeres endelige database som I selv opretter lokalt (I er også velkomne til at lave en SQL server til den). Tænk på hvilke tabeller I vil have, datatyper og relationer imellem tabeller. Det behøver ikke være det design I ender med, da man ofte vil blive klogere på problemet når man arbejder med det, men det kan være godt at have en overordnet struktur at sigte efter.

- Overvej at lave en skeletstruktur for jeres kode. Hvilke moduler, klasser og funktioner tænker I vil være smart at have. Det kan nogle gange være nemmere at definere nogle tomme moduler/klasser/funktioner med deskriptive navne og så fylde dem ud senere.

I må meget gerne være kreative med jeres dokumentation. Overvej om der er diagrammer, flowcharts og lign. der kan gøre det nemmere for andre at forstå jeres ETL-process.

Pensum og Ressourcer

- Extract Transform Load https://en.wikipedia.org/wiki/Extract%2C_transform%2C_load
- ETL Architecture Explained With Diagram https://airbyte.com/data-engineering-resources/etl-architecture
- https://www.informatica.com/resources/articles/what-is-etl.html

---