Projekt arbetskollen

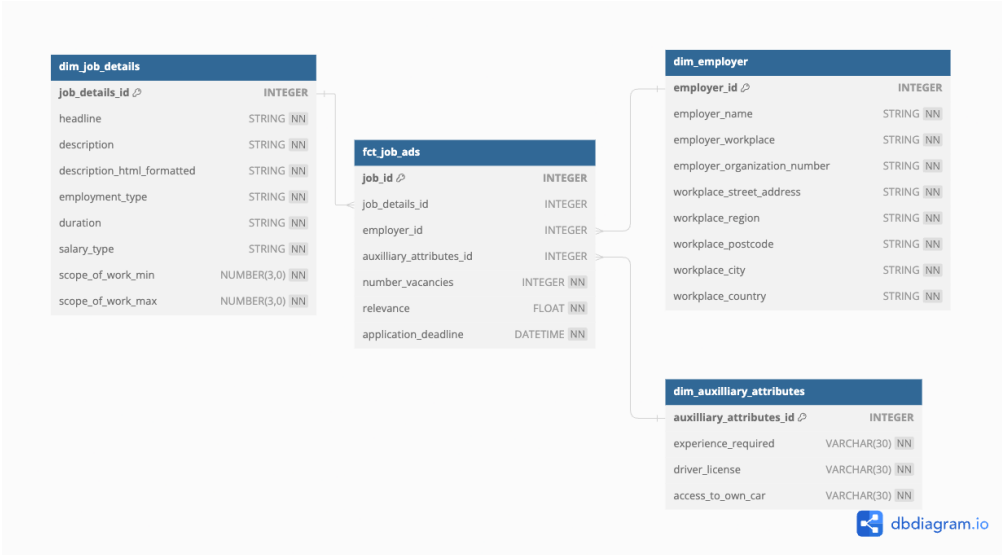
Syfte

Projektet syftar till att implementera den moderna data stacken för att lösa ett verklighetsnära problem. Förutom att använda ett visst antal tekniker finns ett fokus kring att jobba tillsammans i ett datateam. Detta innebär både agil utveckling och att kunna använda git och github i ett team.

Bakgrund

Företaget Arbetskollen AB vill bygga en data pipeline som servar en dashboard för att visualisera statistik för jobbannonser. Detta skulle kunna vara en dimension för att hjälpa framtida studenter att hitta rätt område att studera inom eller för arbetssökande att enklare navigera bland olika yrken.

Arbetskollen har haft en data engineer (DE) som tillsammans med product owner (PO) gjort en initial dimensionsmodellering som inte är fullständig. DE skulle egentligen velat göra färdigt dimensionsmodelleringen, men företagets VD ville snabbt se en prototyp för att kunna söka investering.



Dessvärre blev DE headhuntad och valde att säga upp sig. Innan han slutade implementerade han modellen till viss del och lyckades få en initial dashboard. Han lade också till ett par tester, och gjorde en simpel dokumentation.

VDn visade detta för sina investerare och fick in riskkapital från bland annat Yohanna. Yohanna som tidigare investerat i dataintensiva projekt förstår vikten av noggrannhet, då det oftast straffar sig av att ta genvägar. Därför har hon gett PO mer befogenheter att göra det här projektet noggrant.

Roller i organisationen

Person	Roll	Beskrivning
Yohanna	riskkapitalist	investerat i dataintensiva projekt, begränsad teknisk kompetens, bra businesskunskaper
Jacobi	VD	tuff förhandlare, duktig på business
Börje	product owner	begränsad teknisk kompetens, bra kontakt med både Yohanna och Jacobi
Feliz	frontendutvecklare	sitter för närvarande med annat projekt, men kan assistera med git och github

Scenario

Yohanna och Jacobi väljer nu att kontakta företaget DataRöris AB och ber dem om ett team av data engineers som kan hjälpa dem bygga dataplattformen tillsammans med Börje. Vilket sammanträffande att DataRöris råkade ha er som ska börja på LIA om ett par dagar och som läser data engineering.

Teamet i arbetskollen skissar hastigt en onboardingplan och en kravställning.

Onboarding

Jacobi berättar bakgrund

Feliz går igenom hur ni ska jobba med git och github i det här projektet

- bjuda in teamet
- git branches
- pull requests
- github projects
- tilldela personer till tasks

Börje går igenom agil projektmetodik i arbetskollen

- jobba enligt kanban
- backlog refinement
- definition of done
- 1 task åt gången

Uppgift 0 - uppvärmning

Det som beskrevs i bakgrunden av det som DE satte upp hittar ni i föreläsningarna 07, 09-15 och kursrepot. Se till så att varje person självständigt går igenom dessa föreläsningar och lär sig innehållet.

Detta kan göras asynkront, dvs att andra uppgifter kan göras utan att ha gått igenom alla lektionerna.

Uppgift 1 - setup

När DE lämnade företaget, har han glömt att lämna ut credentials till snowflakekontot eller så har företaget slarvat bort det. Så ni får skapa setupen själva.

Githubrepo

En person sätter upp ett publikt githubrepo och bjuder in övriga. Sätt även upp github projects för att ha en kanbanboard där ni kan se vilka tasks som behöver göras och vilka som jobbar med vilken task.

> **VIKTIGT**
>
> Kom ihåg att jobba i egna branches och kör pull från main branchen innan ni gör pull request till main. På så sätt löses eventuella merge conflicts i sin egna branch.

Gör många committs, vänta inte på att göra få stora committs. På så sätt versionshanteras koden bättre, ni får backup och kodbasen utvecklas över tid.

Virtual environment

En person installerar de paket ni behöver och tar fram en requirementsfil som pushas till github. Övriga personer installerar de olika dependencies som finns i requirementsfilen.

Snowflake

Vi använder en persons snowflakekonto för att ha databas och användare. Den personen skapar users och roller som övriga personer får tillgång till. Ett exempel är att ha en user för dlt, en för dbt och en för streamlit med passwords som övriga kan få ta del av. Dessutom kan det vara bra att man skapar en user per person och respektive user får nödvändiga rollerna. Detta gör att de både kan logga in i snowsight och komma åt databasen, samt göra en connection via snowsql.

Notera att host/accountname då är den personens snowflakekonto (dvs det du får från account_locator_url). Detta är viktigt när varje person sätter upp sina connections i `secrets.toml`, `profiles.yml` och `.env`.

dbt

Här är också en person sätter upp dbt, och övriga gör `git pull`. Kom dock ihåg att ha samma innehåll på `profiles.yml` som pekar mot det här projektet. Kör `dbt deps` för att installera paket också för er som tagit ned kod med pull.

Uppgift 2 - ansvarsområde

Ni är uppdelade i olika teams som ska ha var sitt ansvarsområde. I parantes visas exempel, behöver inte vara exakt de jobben.

1. lärarjobb (gymnasielärare, grundskolelärare, förskollärare)
2. högre utbildning (yrkeshögskolejobb, högskolejobb, universitetsjobb)
3. datajobb (datarelaterade yrken ex data science, data engineer, data analyst, business intelligence)
4. ingenjörsjobb (högskoleingenjör, civilingenjör)
5. ekonomijobb (ekonom, redovisning, controller)

6. projektledarjobb
7. lagerjobb
8. vårdrelaterade jobb

Extrahera och ladda in data för ert arbetsområde från jobtechs API in till snowflake.

Börje berättar att DE en gång sagt att han bara kunda ladda in 100 annonser åt gången. Kör på 100 annonser först, så finns det en bonusuppgift om ni vill utforska vidare.

> **NOTE**
>
> Ni som har datajobb, ladda inte enbart in data engineer då DE redan gjort detta en gång.

Uppgift 3 - dimensionsmodellering

Dimensionsmodelleringen som DE påbörjat är inte fullständig och behöver byggas på. Börje ger er mycket frihet i att diskutera fram i ansvarsområdena vad ni vill modellera. Han har dock fått i uppdrag från Yohanna att se till så att dimensionsmodellen ska förbättras.

Finns dock några felaktigheter i dimensionsmodellen som inte stämmer med DEs implementation. Börje har hittat att fct_job_ads bör ha job_details_key och inte job_details_id som exempel.

Uppgift 4 - datatransformationer

Börje kommunicerar till er att Jacobi tryckt på att implementera i enlighet med dimensionsmodellen, då han vill få ut en produkt. Sen så har Yohanna som varit med i en del dataprojekt trycker på vikten med bra tester för att tåla eventuella felaktig data. Likaså här litar Börje på er att ni både implementerar dimensionsmodellen och har bra tester.

Uppgift 5 - dashboarden

Snygg och användarvänlig är ledorden här. Yohanna och Jacobi ser framemot att bli positivt förvånad över era coola dashboards.

Uppgift 6 - dokumentation

Yohanna vill ha mer dokumentation i dbt för att det enkelt ska gå att onboarda nya data engineers.

Presentera för stakeholders

Ni har 10 minuter på er att presentera per grupp ert projekt. Tänk på att er publik består av Börje, Feliz, Yohanna och Jacobi, så presentera

- presentera dashboarden
- presentera hur ni jobbat agilt
- kort presentera hur data flödat från upstream till downstream
- hur ni jobbat med git och github i teamet

> **VIKTIGT**
>
> ALLA ska presentera

Bonus (frivillig)

Gör dessa om ni har tid över efter er individuella inlämningsuppgift

- deploya dashboarden till streamlit cloud
- implementera paginering i dlt för att fylla på med jobbannonser där det finns fler än 100 annonser
- orkestrera dbt med dagster
- orkestrera dlt och dbt med dagster

Individuell inlämningsuppgift

Skriv en rapport på 1-2 sidor där ni beskriver tekniskt de olika stegen i projektet. Skriv på engelska och använd lämplig teknisk terminologi. Här är exempel på frågor att förhålla sig till

- hur har ni datamodellerat och varför?
- hur har datan flödat, beskriv data lineage här?
- beskriv hur ni har transformerat datan
- vad har ni gjort för avvägningar i dashboarden?
- hur ser uppdelningen av de olika datalagren ut, dvs de olika schemat ni skapat?
- vad är syftet bakom de olika datalagrena?
- vilken teknologi har ni använt och till vilka syften i det här projektet?
- beskriv gärna fler punkter, men håll dig till max 2 sidor för rapporten

Bedömning

Den individuella uppgiften tillsammans med det ni producerat i projektet bestämmer betyget. Notera att projektet bedöms individuellt, ni kommer skicka in er gemensamma kodbas i github och kanbanboardet. Där går det att följa vad respektive person gjort under projektet.

Godkänt

- gjort uppgifterna korrekt i grupp
- varit aktiv i projektet, utfört relevanta tasks
- gjort flera relevanta committs med pull requests mot main
- gjort individuella uppgiften på grundläggande nivå

Väl godkänt

- projektet är utfört på tillräcklig hög nivå på bland annat korrekta roller, modellering, transformationer, tester och professionell presentation
- gjort individuella uppgiften på fördjupande nivå med goda avvägningar och motivationer till olika designval