



SQL Introduktion

Opgavebeskrivelse

I denne uge skal I arbejde med Python og SQL, og hvordan de to teknologier kan bruges sammen til at hente, behandle og analysere data. Fokus vil være både på det tekniske samspil og på, hvordan man strukturerer sin kode i et større projekt.

Ved at integrere SQL og Python lærer man at bygge løsninger, der kan håndtere både datahentning- og analyse i én samlet arbejdsgang. Det er en værdifuld kompetence, når man f.eks. skal bygge backend-systemer, lave rapporteringsværktøjer eller udvikle datadrevne applikationer

Der er to hovedformål med denne uges opgaver:

- At lære, hvordan man interagerer med en MySQL-database fra Python.
- At få erfaring med strukturering af Python-kode, der består af flere forskellige moduler og filer, som skal fungere sammen. I den forbindelse vil vi også arbejde med refaktorering og iterere over den samme kodebase flere gange.

I dokumentet [SQL primer](#) findes forskellige referencer, der kan hjælpe jer med at komme i gang.

Afleveringsformat Præsentation

Præsentationen skal vare cirka 10 minutter og fungere som en mundtlig evaluering af jeres arbejde med Python og SQL. I skal kort præsentere jeres projekt, forklare hvordan Python og SQL arbejder sammen i jeres løsning, og vise eksempler på centrale dele af jeres kode, fx databaseforbindelse, datahentning og behandling. Derudover skal I beskrive, hvordan I har struktureret jeres projekt i moduler og filer, og hvordan I har arbejdet med refaktorering undervejs. Afslutningsvis bør I reflektere over, hvad I har lært, hvilke udfordringer I stødte på, og hvordan I løste dem.

Delopgaver og opmærksomhedspunkter

Brug gerne: mysql-connector-python, pandas, [Postgres](#)

Læringen for denne uge ligger i at kode SQL relationer. I må derfor ikke anvende ORM (object relational mapper) frameworks såsom SQLAlchemy



● Delopgave 1: Python opsætning

- Første skridt er at lave en god opsætning af Python-projektmappen. Dokumentet *Python Tips* indeholder et eksempel på en mappestruktur. Herefter skal mappen initialiseres som et GitHub-repository, og der skal oprettes et virtual environment.
- Dokumentet *Package Management Python* indeholder praktisk information om virtuelle miljøer, mens *Virtual Environments* giver mere baggrundsviden. I mappen *GitHub* kan I finde information om opsætning af Git.
- Vær opmærksom på at inkludere en `.gitignore`-fil og tilføje `.venv`-mappen til den. Overvej også at bruge *branches* og *pull requests*, når I arbejder på de forskellige moduler. Det kan måske virke som overkill, når man kun er én person, der arbejder på koden, men det er en god vane at opbygge.
- Se dokumenterne *Guide til GitHub-opsætning* og *Git Best Practices* for yderligere vejledning

● Delopgave 2: Opsætning af databaseforbindelse

- Opret en Python-klasse, der håndterer forbindelsen til MySQL-serveren.
- Placér klassen i en separat fil, så den kan importeres som et modul i resten af projektet.
- Brug `mysql-connector-python` til at etablere forbindelsen – husk at installere biblioteket.
- Opret en database og en tabel med samme struktur som `orders_combined.csv`.
- Lav et Python-modul, der indlæser CSV-filen og skriver indholdet til databasen.
- Overvej at implementere en funktion til at droppe databasen, så du nemt kan nulstille den under udvikling.
- Hold koden modulopdelt, så hver del har et klart ansvar og nemt kan genbruges andre steder i projektet.

● Delopgave 3: CRUD

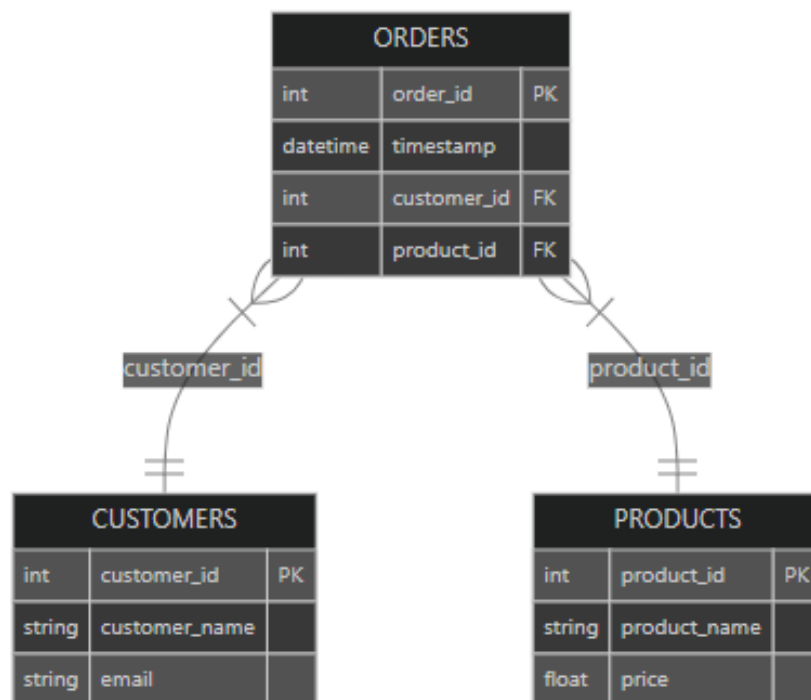
- Formålet med denne opgave er at opnå en forståelse for CRUD-forespørgsler i MySQL.
- Lav Python-funktioner, der håndterer Create, Read, Update og Delete-operationer på den oprettede database.
- Der vil muligvis være behov for flere funktioner til hver type operation.



- Indkapsl dem i en klasse, så du har ét samlet objekt, du kan bruge til at kommunikere med databasen.
- Overvej eventuelt at refaktorere det modul, der skriver information til databasen, så det benytter det modul, I laver her.

Delopgave 4: Relationer og join-queries

- opret en database med primary-foreign key-relationer fra Python. De tre filer *orders.csv*, *products.csv* og *customers.csv* indeholder de samme data som *orders_combined.csv*, men er organiseret med primary-foreign key-relationer. Refaktorér din kode, så den også kan håndtere data organiseret på denne måde.
- Nedenstående figur viser et skema over tabellerne, som de skal være i databasen:



PK: primary key, FK: foreign key