SQL og Python

Introduktion og formål

Der er to hovedformål med denne uges opgaver.

1. At lære hvordan man interagerer med en MySQL database fra Python.
2. At få erfaring med strukturering af python kode der har flere forskellige moduler og filer, der 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 med at komme igang.

Python setup

Første skridt er at lave en god opsætning af Python projekt mappen. Dokumentet *Python tips* indeholder et eksempel på en mappe struktur.
Herefter skal mappen initialiseres som et github repository og der skal laves en virtual environment. Dokumentet *Package Management Python* indeholder praktisk information om virtual environments og *Virtual Environments* indeholder mere baggrundsviden. I mappen Github kan I finde information om opsætning af git. Vær opmærksom på at inkludere en *.gitignore* fil og tilføj *.venv* mappen til den.
Overvej at bruge branches og pull-requests når I arbejder på de forskellige moduler. Det kan virke som overkill når man kun er én person der arbejder på koden, men det er en god vane at opbygge. Se *Guide til github opsætning* og *Git best practices*.

MySQL-Connector-Python

Start med at lave en python class, der håndterer forbindelsen til MySQL serveren. Lav den i sin egen fil, så du kan importere den som et modul andre steder i din kode.

Opret databasen

Indlæs filen *orders_combined.csv* og skriv indholdet til en database, med en enkelt tabel, der har de samme kolonner som csv filen. Det kan gøres ved at lave et python modul, der håndterer at oprette databasen og skrive indholdet til den. Evt kan man også implementere en funktion til at droppe databasen igen. Det kan gøre det nemmere at debugge senere, hvis man nemt har mulighed for at nulstille databasen.

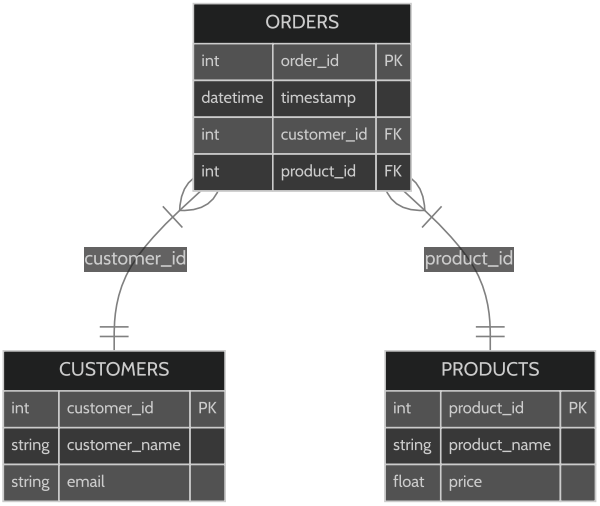
CRUD

Formålet med dette afsnit er at få en forståelse af CRUD queries i MySQL.
Lav python funktioner der håndterer create, read, update, delete operationer på den oprettede database. Der skal muligvis være flere funktioner til hver type operation. Enkapsulér dem i en class, så du har et enkelt objekt du kan bruge til at kommunikere med databasen.
Man kan evt. refaktorere modulet der skriver information til databasen, til at bruge det modul i laver her.

Relations og Join queries

Nu skal du oprette en database med primary-foreign key relations fra python. De tre filer *orders.csv, products.csv,* og *customers.csv*, indeholder samme data som *orders_combined.csv,* men organiseret med

primary-foreign key relations. 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