AI og data mini projekt

Indhold:

1. d
2. Lektion 2: Databaser
3. Lektion 5: Intro til Pandas og Datamanipulering
4. e

# Lektion 1:

# Lektion 2: Databaser

1. Skriv et Python script der opretter en SQLite database ‘school.db’ (se <https://realpython.com/python-sql-libraries/> )
2. Udvid scriptet til at oprette to tabeller ‘Students’ og ‘Courses’.
   1. Students skal indeholde student\_id, name, og major
   2. Courses skal indeholde course\_id, course\_name, og instructor
3. Indsæt mindst 5 records i hver tabel.
4. Opret en tredje tabel ‘Enrollments’ der opretter relationer mellem Students og Course.
   1. Enrollments skal indeholde enrollment\_id, student\_id, og course\_id
5. Lav forespørgsler (queries) der
   1. Vælger alle kurser som en specific studerende er tilmeldt
   2. Vælger alle studerende der er tilmeldt et specifikt kursus.
6. Opgave 1:

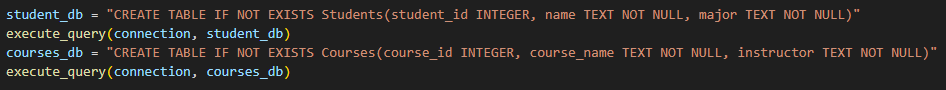
For at oprette en SQLite database, impoter jeg sqlite3. Herefter opretter jeg en connection object til en ”path” der kaldes school.db:  


1. Opgave 2:

Jeg executer ”CREATE TABEL IF NOT EXISTS” på cursor objectet.



Querierne laver 2 tabeller i school.db databasen, kaldet henholdsvis ”Students” og ”Courses”. Begge tabeller får deres egne id for at de nemmere kan samles senere.



1. Opgave 3:

Jeg executer INSERT INTO på tabellen og dens 3 columns og angiver 5 unikke studenter. Det samme bliver gjort for Courses.

Et billede, der indeholder tekst, Font/skrifttype, skærmbillede

Automatisk genereret beskrivelse

1. Opgave 4:

Jeg laver en ny tabel til Enrollments på samme måde som ved Students og Courses i opgave 2. Herefter tilføjer jeg 10 rækker til Enrollments hvor student\_id og course\_id kan relateres til hinanden.

Et billede, der indeholder tekst, skærmbillede, Font/skrifttype

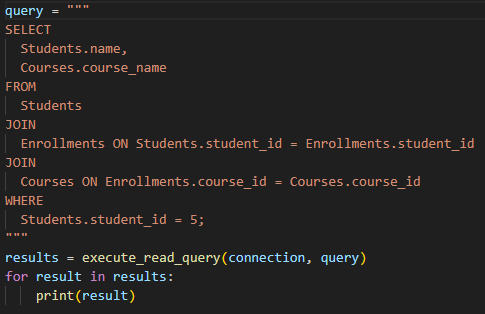
Automatisk genereret beskrivelse

1. Opgrave 5:

Her bruges .fetchall() funktionen til at returnere en liste af tuples der indeholder sammenhængen mellem de valgte records.



Jeg SELECTer først at jeg vil have student navnene og course navnene. Herefter JOINer jeg Students tabellen til Enrollments tabellen via student\_id. Så JOINer jeg det resulterende tabel sammen med courses tabellen via course\_id. Så bruger jeg WHERE til at specificere at jeg, i den resulterende tabel, kun vil have information fra den student med student\_id = 5. Til sidst printer jeg de tupler som .fetchall() returnerede.



Den samme fremgangsmåde blev brugt til at finde alle studenter der er i en specifik course. Herunder er de returnerede tupler:

Et billede, der indeholder tekst, Font/skrifttype, skærmbillede, typografi

Automatisk genereret beskrivelse

# Lektion 5: Intro til Pandas og Datamanipulering

1. Følg tutorial (under 'Forberedelse') grundigt på egen hånd. Opret en Google Colab notebook (https://colab.research.google.com/), og implementer og afprøv alle eksempler på egen hånd.
2. Anvend metoderne på datasættet fra kursusgang 4:
   1. indlæs datasættet i en DataFrame
   2. fjern manglende værdier
   3. plot numeriske variabler i datasættet (https://python.pages.doc.ic.ac.uk/lessons/pandas/06-methods/05-plot.html)

no

1. Opgave 1:

Jeg lavede en fil kaldet ”pandas\_intro.py” i lektion 5 mappen hvor jeg implementerede alle funktionerne jeg testede fra Pandas tutorialen.

1. Opgave 2:

Jeg indlæser datasættet i Pandas med følgende funktion:



Jeg printer .shape på dataframe før og efter jeg bruger .dropna funktionen:

Et billede, der indeholder Font/skrifttype, tekst, skærmbillede, Grafik

Automatisk genereret beskrivelse



Jeg bruger .info til at finde hvilke columns der har numeriske værdier:



Herefter plotter jeg for ”Size(L)” og ”Color”:

­­Et billede, der indeholder tekst, Font/skrifttype, skærmbillede, design

Automatisk genereret beskrivelse­

Et billede, der indeholder tekst, skærmbillede, display/skærm/fremvisning, Rektangel

Automatisk genereret beskrivelse

Et billede, der indeholder tekst, skærmbillede, diagram, display/skærm/fremvisning

Automatisk genereret beskrivelse