ORM

Brandstätter, Forsthuber, Jernej

Inhalt

[Datenbank aufsetzen 3](#_Toc42338862)

[Python Klassen anlegen 3](#_Toc42338863)

[Datenbankverbindung 3](#_Toc42338864)

# Datenbank aufsetzen

\*Klassendiagramm\*

\*SQL-Query zum Createn und aufsetzen\*

# Python Klassen anlegen

sqlalchemy muss installiert werden

Pip install sqlalchemy

Klassen müssen entsprechend den Tabellen in der DB angelegt werden und die Attribute müssen definiert werden. Die Klassen ?vererben? von declarative\_base und müssen alle dieselbe base verwenden, damit die Beziehungen beim instanziieren schon bekannt sind.

Beziehungen müssen mit relationship() definiert werden. Wir haben ausgehend vom Centertable n:1 Beziehungen auf Ordertimes, Producttable und Customertable. Folgendermaßen wird das definiert:

\*Bild/Code von Klassen\*

# Datenbankverbindung

In der dbConnection wird die Verbindung zur Datenbank hergestellt und eine Session aufgemacht:

engine = create\_engine(**'postgresql://postgres:Test@localhost:5432/postgres'**)  
  
*# create a configured "Session" class*Session = sessionmaker(engine)  
  
*# create a Session*session = Session()

über diese Session kann man die Daten der Datebank auslesen. Z.b mit

film = session.query(Ordertimes)

werden alle Zeilen der Tabelle Ordertimes in film geladen. Film ist eine Liste von Ordertimes objekten.

Die Klasse dbConnection soll grundsätzlich die Instanz der Datenbankverbindung halten und alle interaktionen mit der DB an einem Ort bündeln.

# UseCase

Als Benutzer möchte ich die Verkaufsdaten auslesen und neue einspeichern können. Dazu gibt es momentan ein primitives UI in der Konsole. Die Steuerung erfolgt über Textbefehle. Es können die verschiedenen Tabellen ausgelesen werden und der Inhalt wird ausgegeben. Es können neue Objekte angelegt und in der DB abgespeichert werden. Das Löschen und Bearbeiten von Objekten ist optional.