SQLAlchemy/database.py (Page 1 of 1)

```
#!/usr/bin/env python

#-----------------------------------------------------------------------
# database.py
# Author: Bob Dondero
#-----------------------------------------------------------------------

import sqlalchemy
import sqlalchemy.orm

class Base(sqlalchemy.orm.DeclarativeBase):
    pass

class Book (Base):
    __tablename__ = 'books'
    isbn = sqlalchemy.Column(sqlalchemy.String,
        primary_key=True, nullable=False)
    title = sqlalchemy.Column(sqlalchemy.String)
    quantity = sqlalchemy.Column(sqlalchemy.Integer)

class Author (Base):
    __tablename__ = 'authors'
    isbn = sqlalchemy.Column(sqlalchemy.String,
        sqlalchemy.ForeignKey('books.isbn'),
        primary_key=True, nullable=False)
    author = sqlalchemy.Column(sqlalchemy.String,
        primary_key=True, nullable=False)
    #books = sqlalchemy.orm.relationship('Book', backref='authors')

class Zipcode (Base):
    __tablename__ = 'zipcodes'
    zipcode = sqlalchemy.Column(sqlalchemy.String,
        primary_key=True, nullable=False)
    city = sqlalchemy.Column(sqlalchemy.String)
    state = sqlalchemy.Column(sqlalchemy.String)

class Customer (Base):
    __tablename__ = 'customers'
    custid = sqlalchemy.Column(sqlalchemy.String,
        primary_key=True, nullable=False)
    custname = sqlalchemy.Column(sqlalchemy.String)
    street = sqlalchemy.Column(sqlalchemy.String)
    zipcode = sqlalchemy.Column(sqlalchemy.String,
        sqlalchemy.ForeignKey('zipcodes.zipcode'))
    #zipcodes=sqlalchemy.orm.relationship('Zipcode',backref='customers')

class Order (Base):
    __tablename__ = 'orders'
    isbn = sqlalchemy.Column(sqlalchemy.String,
        sqlalchemy.ForeignKey('books.isbn'),
        primary_key=True, nullable=False)
    custid = sqlalchemy.Column(sqlalchemy.String,
        sqlalchemy.ForeignKey('customers.custid'),
        primary_key=True, nullable=False)
    quantity = sqlalchemy.Column(sqlalchemy.Integer)
```

blank (Page 1 of 1)

```
This page is intentionally blank.
```

SQLAlchemy/create.py (Page 1 of 2)

```
#!/usr/bin/env python

#-----------------------------------------------------------------------
# create.py
# Author: Bob Dondero
#-----------------------------------------------------------------------

import os
import sys
import sqlalchemy
import sqlalchemy.orm
import dotenv
import database

dotenv.load_dotenv()
DATABASE_URL = os.getenv('DATABASE_URL', 'sqlite:///bookstore.sqlite')
DATABASE_URL = DATABASE_URL.replace('postgres://', 'postgresql://')
_DATABASE_URL = _DATABASE_URL.replace(
    'postgresql://', 'postgresql+psycopg://')

engine = sqlalchemy.create_engine(DATABASE_URL)

#-----------------------------------------------------------------------

def main():

    if len(sys.argv) != 1:
        print(f'usage: python {sys.argv[0]}', file=sys.stderr)
        sys.exit(1)

    try:

        database.Base.metadata.drop_all(engine)
        database.Base.metadata.create_all(engine)

        with sqlalchemy.orm.Session(engine) as session:

            #-----------------------------------------------------------

            book = database.Book(isbn=123,
                title='The Practice of Programming', quantity=500)
            session.add(book)
            book = database.Book(isbn=234,
                title='The C Programming Language', quantity=800)
            session.add(book)
            book = database.Book(isbn=345,
                title='Algorithms in C', quantity=650)
            session.add(book)
            session.commit()

            #-----------------------------------------------------------

            author = database.Author(isbn=123, author='Kernighan')
            session.add(author)
            author = database.Author(isbn=123, author='Pike')
            session.add(author)
            author = database.Author(isbn=234, author='Kernighan')
            session.add(author)
            author = database.Author(isbn=234, author='Ritchie')
            session.add(author)
            author = database.Author(isbn=345, author='Sedgewick')
            session.add(author)
            session.commit()

            #-----------------------------------------------------------
```

SQLAlchemy/create.py (Page 2 of 2)

```

            zipcode = database.Zipcode(zipcode='08540',
                city='Princeton', state='NJ')
            session.add(zipcode)
            zipcode = database.Zipcode(zipcode='02138',
                city='Cambridge', state='MA')
            session.add(zipcode)
            zipcode = database.Zipcode(zipcode='02142',
                city='Cambridge', state='MA')
            session.add(zipcode)
            session.commit()

            #-----------------------------------------------------------

            customer = database.Customer(custid='111',
                custname='Princeton', street='114 Nassau St',
                zipcode='08540')
            session.add(customer)
            customer = database.Customer(custid='222',
                custname='Harvard', street='1256 Mass Ave',
                zipcode='02138')
            session.add(customer)
            customer = database.Customer(custid='333',
                custname='MIT', street='292 Main St',
                zipcode='02142')
            session.add(customer)
            session.commit()

            #-----------------------------------------------------------

            order = database.Order(isbn='123', custid='222',
                quantity=20)
            session.add(order)
            order = database.Order(isbn='345', custid='222',
                quantity=100)
            session.add(order)
            order = database.Order(isbn='123', custid='111',
                quantity=30)
            session.add(order)
            session.commit()

    except Exception as ex:
        print(f'{sys.argv[0]}: {ex}', file=sys.stderr)
        sys.exit(1)

#-----------------------------------------------------------------------

if __name__ == '__main__':
    main()
```

SQLAlchemy/display.py (Page 1 of 2)

```
#!/usr/bin/env python

#-----------------------------------------------------------------------
# display.py
# Author: Bob Dondero
#-----------------------------------------------------------------------

import os
import sys
import sqlalchemy
import sqlalchemy.orm
import dotenv
import database

dotenv.load_dotenv()
DATABASE_URL = os.getenv('DATABASE_URL', 'sqlite:///bookstore.sqlite')
DATABASE_URL = DATABASE_URL.replace('postgres://', 'postgresql://')
_DATABASE_URL = _DATABASE_URL.replace(
    'postgresql://', 'postgresql+psycopg://')

engine = sqlalchemy.create_engine(DATABASE_URL)

#-----------------------------------------------------------------------

def main():

    if len(sys.argv) != 1:
        print(f'usage: python {sys.argv[0]}', file=sys.stderr)
        sys.exit(1)

    try:

        with sqlalchemy.orm.Session(engine) as session:

            print('-----------------------------------------')
            print('books')
            print('-----------------------------------------')
            query = session.query(database.Book)
            table = query.all()
            for row in table:
                print(row.isbn, row.title, row.quantity)

            print('-----------------------------------------')
            print('authors')
            print('-----------------------------------------')
            query = session.query(database.Author)
            table = query.all()
            for row in table:
                print(row.isbn, row.author)

            print('-----------------------------------------')
            print('zipcodes')
            print('-----------------------------------------')
            query = session.query(database.Zipcode)
            table = query.all()
            for row in table:
                print(row.zipcode, row.city, row.state)

            print('-----------------------------------------')
            print('customers')
            print('-----------------------------------------')
            query = session.query(database.Customer)
            table = query.all()
            for row in table:
                print(row.custid, row.custname, row.street, row.zipcode)
```

SQLAlchemy/display.py (Page 2 of 2)

```

            print('-----------------------------------------')
            print('orders')
            print('-----------------------------------------')
            query = session.query(database.Order)
            table = query.all()
            for row in table:
                print(row.isbn, row.custid, row.quantity)

        engine.dispose()

    except Exception as ex:
        print(f'{sys.argv[0]}: {ex}', file=sys.stderr)
        sys.exit(1)

#-----------------------------------------------------------------------

if __name__ == '__main__':
    main()
```

SQLAlchemy/authorsearch.py (Page 1 of 1)

```
#!/usr/bin/env python

#-----------------------------------------------------------------------
# authorsearch.py
# Author: Bob Dondero
#-----------------------------------------------------------------------

import os
import sys
import sqlalchemy
import sqlalchemy.orm
import dotenv
import database

dotenv.load_dotenv()
DATABASE_URL = os.getenv('DATABASE_URL', 'sqlite:///bookstore.sqlite')
DATABASE_URL = DATABASE_URL.replace('postgres://', 'postgresql://')
_DATABASE_URL = _DATABASE_URL.replace(
    'postgresql://', 'postgresql+psycopg://')

engine = sqlalchemy.create_engine(DATABASE_URL)

#-----------------------------------------------------------------------

def main():

    if len(sys.argv) != 2:
        print(f'usage: python {sys.argv[0]} author', file=sys.stderr)
        sys.exit(1)

    author = sys.argv[1]

    try:
        with sqlalchemy.orm.Session(engine) as session:

            query = (session.query(database.Book.isbn,
                database.Book.title, database.Book.quantity)
                .filter(database.Book.isbn == database.Author.isbn)
                .filter(database.Author.author == author))
            print(query)

            table = query.all()
            for row in table:
                print('ISBN:', row.isbn)
                print('Title:', row.title)
                print('Quantity:', row.quantity)
                print()

    except Exception as ex:
        print(f'{sys.argv[0]}: {ex}', file=sys.stderr)
        sys.exit(1)

#-----------------------------------------------------------------------

if __name__ == '__main__':
    main()
```


SQLAlchemy/order.py (Page 1 of 1)

```
#!/usr/bin/env python

#-----------------------------------------------------------------------
# order.py
# Author: Bob Dondero
#-----------------------------------------------------------------------

import os
import sys
import sqlalchemy
import sqlalchemy.orm
import dotenv
import database

dotenv.load_dotenv()
DATABASE_URL = os.getenv('DATABASE_URL', 'sqlite:///bookstore.sqlite')
DATABASE_URL = DATABASE_URL.replace('postgres://', 'postgresql://')
_DATABASE_URL = _DATABASE_URL.replace(
    'postgresql://', 'postgresql+psycopg://')

engine = sqlalchemy.create_engine(DATABASE_URL)

#-----------------------------------------------------------------------

def main():

    if len(sys.argv) != 3:
        print(f'usage: python {sys.argv[0]} isbn custid',
            file=sys.stderr)
        sys.exit(1)

    isbn = sys.argv[1]
    custid = sys.argv[2]

    try:
        with sqlalchemy.orm.Session(engine) as session:

            query = (session.query(database.Order)
                .filter(database.Order.isbn == isbn)
                .filter(database.Order.custid == custid))

            row = query.one()
            row.quantity += 1

            session.commit()

    except Exception as ex:
        print(f'{sys.argv[0]}: {ex}', file=sys.stderr)
        sys.exit(1)

#-----------------------------------------------------------------------

if __name__ == '__main__':
    main()
```

SQLAlchemy/purchase.py (Page 1 of 1)

```
#!/usr/bin/env python

#-----------------------------------------------------------------------
# purchase.py
# Author: Bob Dondero
#-----------------------------------------------------------------------

import os
import sys
import sqlalchemy
import sqlalchemy.orm
import dotenv
import database

dotenv.load_dotenv()
DATABASE_URL = os.getenv('DATABASE_URL', 'sqlite:///bookstore.sqlite')
DATABASE_URL = DATABASE_URL.replace('postgres://', 'postgresql://')
_DATABASE_URL = _DATABASE_URL.replace(
    'postgresql://', 'postgresql+psycopg://')

engine = sqlalchemy.create_engine(DATABASE_URL)

#-----------------------------------------------------------------------

def main():

    if len(sys.argv) != 3:
        print(f'usage: python {sys.argv[0]} isbn custid',
              file=sys.stderr)
        sys.exit(1)

    isbn = sys.argv[1]
    custid = sys.argv[2]

    try:
        with sqlalchemy.orm.Session(engine) as session:

            query = (session.query(database.Order)
                .filter(database.Order.isbn == isbn)
                .filter(database.Order.custid == custid))
            row = query.one()
            row.quantity += 1

            query = (session.query(database.Book)
                .filter(database.Book.isbn == isbn))
            row = query.one()
            row.quantity -= 1

            session.commit()
            print('Transaction committed.')

    except Exception as ex:
        print(f'{sys.argv[0]}: {ex}', file=sys.stderr)
        sys.exit(1)

#-----------------------------------------------------------------------

if __name__ == '__main__':
    main()
```

SQLAlchemy/recovery.py (Page 1 of 1)

```
#!/usr/bin/env python

#-----------------------------------------------------------------------
# recovery.py
# Author: Bob Dondero
#-----------------------------------------------------------------------

import os
import sys
import random
import sqlalchemy
import sqlalchemy.orm
import dotenv
import database

dotenv.load_dotenv()
DATABASE_URL = os.getenv('DATABASE_URL', 'sqlite:///bookstore.sqlite')
DATABASE_URL = DATABASE_URL.replace('postgres://', 'postgresql://')
_DATABASE_URL = _DATABASE_URL.replace(
    'postgresql://', 'postgresql+psycopg://')

engine = sqlalchemy.create_engine(DATABASE_URL)

#-----------------------------------------------------------------------

def main():

    if len(sys.argv) != 1:
        print(f'usage: python {sys.argv[0]}', file=sys.stderr)
        sys.exit(1)

    try:
        with sqlalchemy.orm.Session(engine) as session:

            for i in range(20):

                query = (session.query(database.Order)
                    .filter(database.Order.isbn == '123')
                    .filter(database.Order.custid == '222'))
                row = query.one()
                row.quantity += 1

                # Simulate a HW/SW failure occurring randomly.
                if random.randrange(5) == 0:
                    print(f'Simulated failure with i = {i}')
                    sys.exit(1)

                query = (session.query(database.Book)
                    .filter(database.Book.isbn == '123'))
                row = query.one()
                row.quantity -= 1

                session.commit()
                print('Transaction %d committed.' % i)

    except Exception as ex:
        print(f'{sys.argv[0]}: {ex}', file=sys.stderr)
        sys.exit(1)

#-----------------------------------------------------------------------

if __name__ == '__main__':
    main()
```