Class Exercise: Introduction to SQLAlchemy

Step 1: Installing SQLAlchemy

1. **Install SQLAlchemy** using `pip`:

   ```
   bash
   Copy code
   pip install sqlalchemy
   ```

Step 2: Basic Setup and Connecting to a Database

1. **Create a Python script** called `database_setup.py`.
2. **Import the necessary modules** and set up a connection to an SQLite database (or any other database):

   ```
   python
   Copy code
   from sqlalchemy import create_engine
   from sqlalchemy.ext.declarative import declarative_base
   from sqlalchemy import Column, Integer, String
   from sqlalchemy.orm import sessionmaker

   # Create the engine and connect to the database
   engine = create_engine('sqlite:///students.db', echo=True)

   # Base class for model classes
   Base = declarative_base()
   ```

Step 3: Define a Table (Model)

1. **Define the `Student` model** representing a table in the database:

   ```
   python
   Copy code
   class Student(Base):
       __tablename__ = 'students'

       id = Column(Integer, primary_key=True)
       name = Column(String)
       age = Column(Integer)
       grade = Column(String)

       def __repr__(self):
           return f"<Student(name='{self.name}', age={self.age},
   grade='{self.grade}')>"
   ```

2. **Create the table**:

   ```
   python
   Copy code
   # Create the students table in the database
   Base.metadata.create_all(engine)
   ```

Step 4: Creating a Session

1. **Set up a session** to interact with the database:

```
python
Copy code
Session = sessionmaker(bind=engine)
session = Session()
```

Step 5: Adding Data to the Database (Create)

1. **Add new student records** to the database:

```
python
Copy code
# Add students
student1 = Student(name='Alice', age=20, grade='A')
student2 = Student(name='Bob', age=22, grade='B')

# Add to the session and commit the changes
session.add(student1)
session.add(student2)
session.commit()

print("Students added successfully!")
```

Step 6: Querying Data from the Database (Read)

1. **Query all students** from the database:

```
python
Copy code
# Query all students
students = session.query(Student).all()
for student in students:
    print(student)
```

2. **Query a single student**:

```
python
Copy code
# Query a student by name
alice = session.query(Student).filter_by(name='Alice').first()
print(alice)
```

Step 7: Updating Data in the Database (Update)

1. **Update a student's grade**:

```
python
Copy code
# Update Bob's grade
bob = session.query(Student).filter_by(name='Bob').first()
bob.grade = 'A+'
session.commit()

print(f"Updated {bob.name}'s grade to {bob.grade}")
```

Step 8: Deleting Data from the Database (Delete)

1. **Delete a student record**:

```
python
Copy code
# Delete a student from the database
student_to_delete =
session.query(Student).filter_by(name='Alice').first()
session.delete(student_to_delete)
session.commit()

print(f"Deleted student {student_to_delete.name}")
```

Step 9: Closing the Session

1. **Close the session** when you're done:

```
python
Copy code
session.close()
```

Bonus Challenge: Add More Features

- **Add validation**: Ensure that the `age` and `grade` fields follow specific rules (e.g., age must be a positive number, grade must be between A-F).
- **Handle exceptions**: Add error handling for database operations.
- **Extend the model**: Add more fields to the `Student` model, such as email or enrollment date.