Retrieving Data from the Database

1. Query all records

1.1 Select all columns

```
Use the all() method on the query attribute to get all the records in a table
>>> from app.models import Student
>>> q=Student.query.all()
```

1.2 Select specific columns:

`with_entities` method is used to specify which columns you want to select from a table

```
>>> s=Student.query.with_entities(Student.id, Student.first_name, Student.last_name).all()
```

1.3 Sorting the result

```
Use order_by() method to sort the result
>>> Student.query.order_by(Student.id.desc()).all()
>>> Student.query.order_by(Student.id.asc()).all()
```

2. Query a single record

2.1 Query the first record

```
Use the first() method to get the first record
>>> Student.query.order_by(Student.id.desc()).first()
```

2.1 Query a single record by its primary key (ID)

```
When records are identified with a unique ID, use get() method to fetch a record by its ID
>>> Student.query.get(1) #legacy as of the 1.x series of SQLAlchemy
>>> db.session.get(Student,1)
```

3. Query records based on a filter

- The `filter()` method is more flexible and versatile. It allows you to build complex filtering conditions by using SQLAlchemy's built-in filtering expressions. You can use comparison operators `(like ==, !=, <, >, etc.)`, logical operators `(like and, or)`, and even functions to construct more sophisticated queries.

  ```
  >>> s1 = Student.query.filter(Student.age>18,Student.gender_id==1)
  ```

- The `filter_by()` method is simpler and more concise, particularly when you're filtering based on equality conditions. It takes keyword arguments where the keys are the column names and the values are the values you're filtering for:

  ```
  >>> Student.query.filter_by(first_name='Prom').all()
  ```

4. Using join

4.1 Two tables

```
>>> q=db.session.query(Student, Gender).join \
    (Gender, Student.gender_id==Gender.id) \
    .with_entities(Student.id, Student.first_name, \
        Student.last_name, Gender.acronym, \
        Student.email).all()
```

4.2 Multiples Tables

```
>>> q1 = db.session.query(User,Student,Gender)\
    .join(Student, User.student_id==Student.id)\
    .join(Gender, Student.gender_id==Gender.id)\

.with_entities(Student.first_name,Student.last_name,User.login_name,Gender
.acronym).all()
```