**Лабораторная работа 2**

**Работа с SQLAlchemy и alembic**

**Студент:** Лещев Иван Анатольевич

**Группа:** РИМ-150950

**Репозиторий:** <https://github.com/qqwq1/appDev>

**Цель работы**

Освоить принципы работы с библиотеками SQLAlchemy и Alembic для создания и управления реляционными базами данных на Python, изучить механизмы миграции базы данных.

**Задачи**:

1. Установить необходимые библиотеки

2. Создать ORM для пользователя и адреса

3. Инициализировать миграции

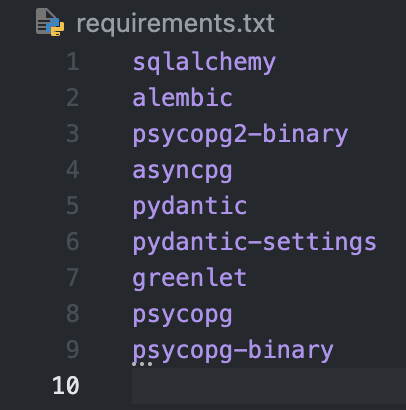
4. Наполнить БД данными

5. Запрос связанных подключений

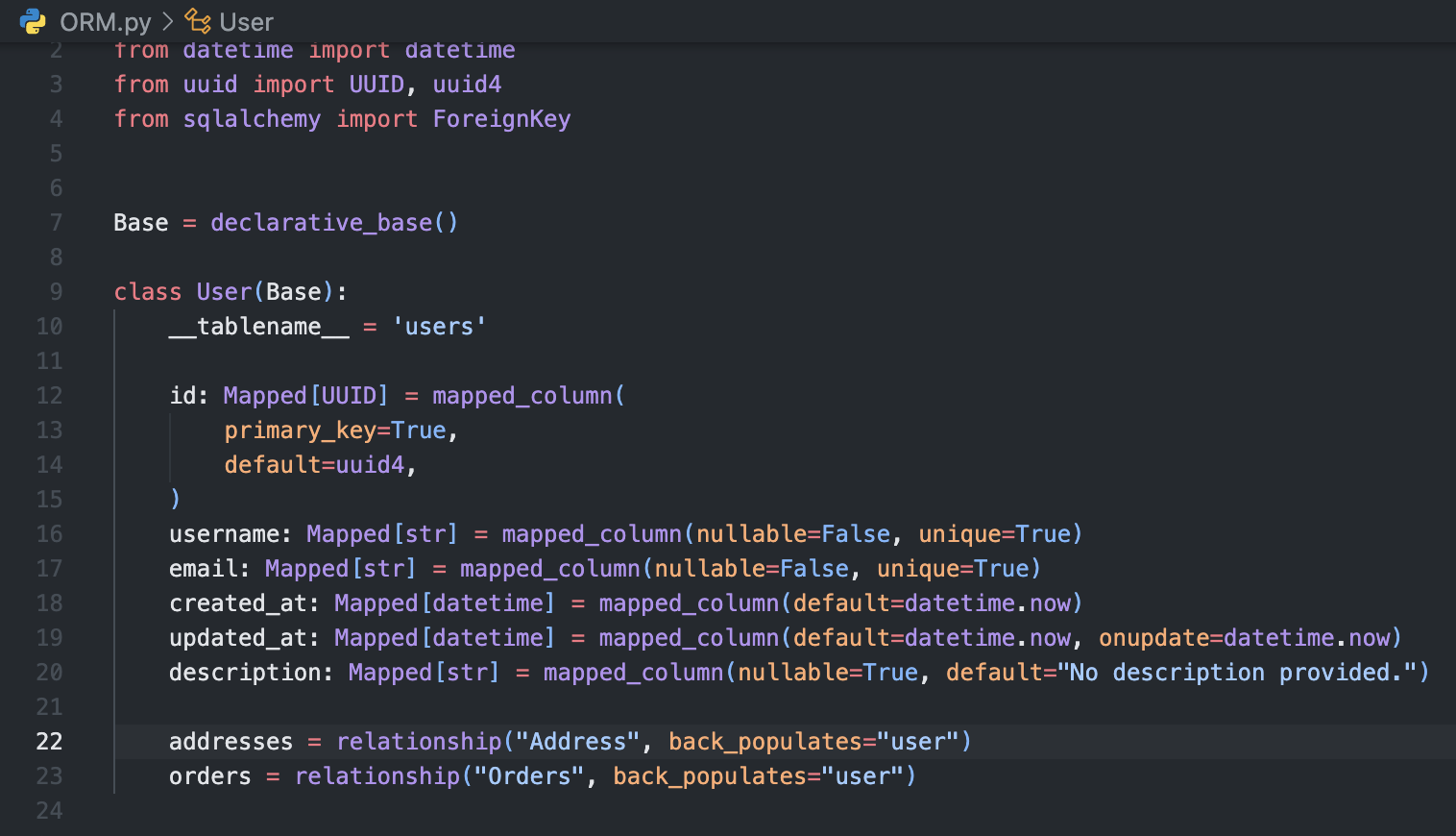
6. Добавить пользователю поле description

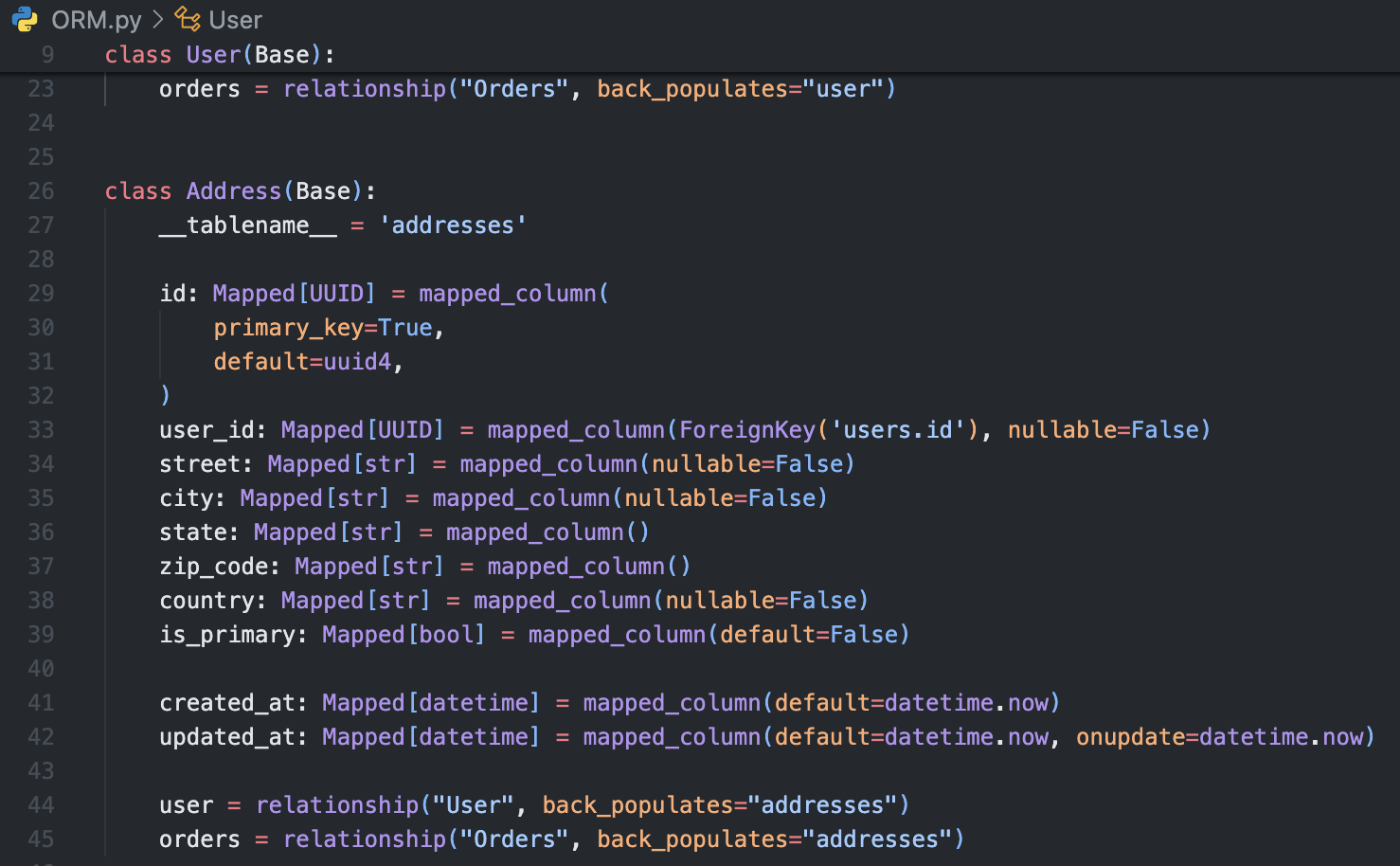
7. Создать таблицу с заказами и произвести миграцию

1. Установка необходимых библиотек

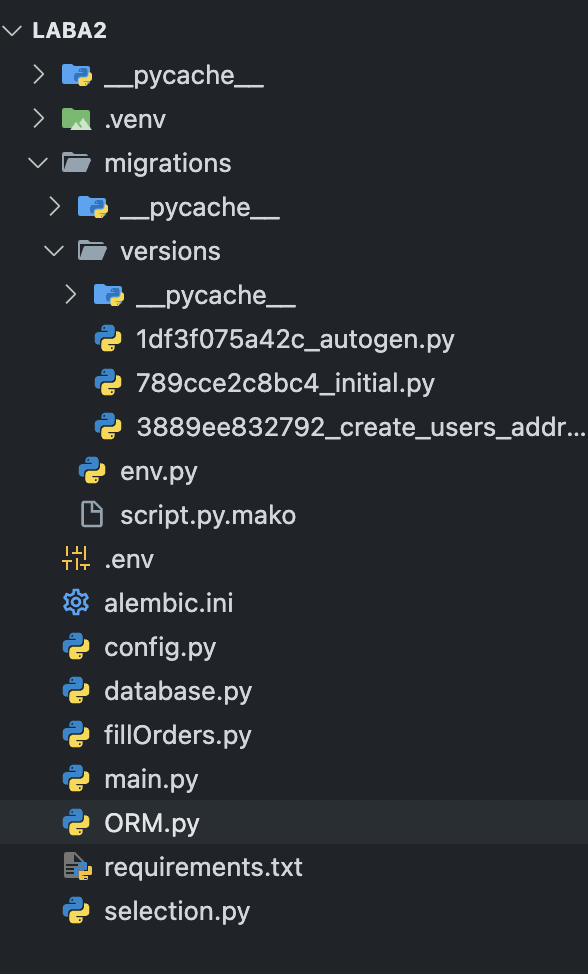


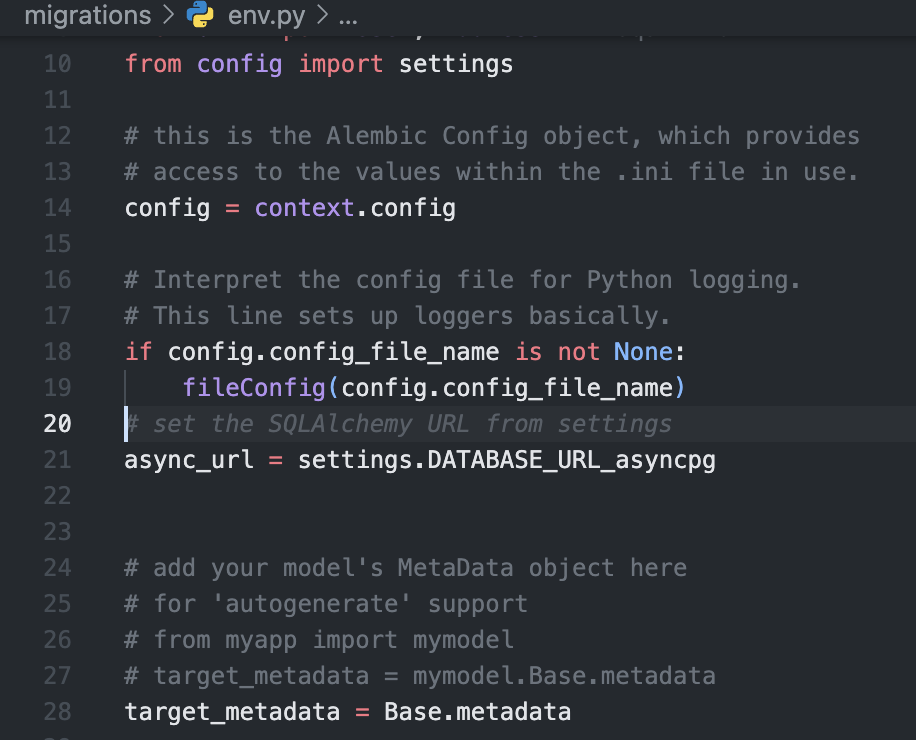
1. Создание ORM для пользователя и адреса

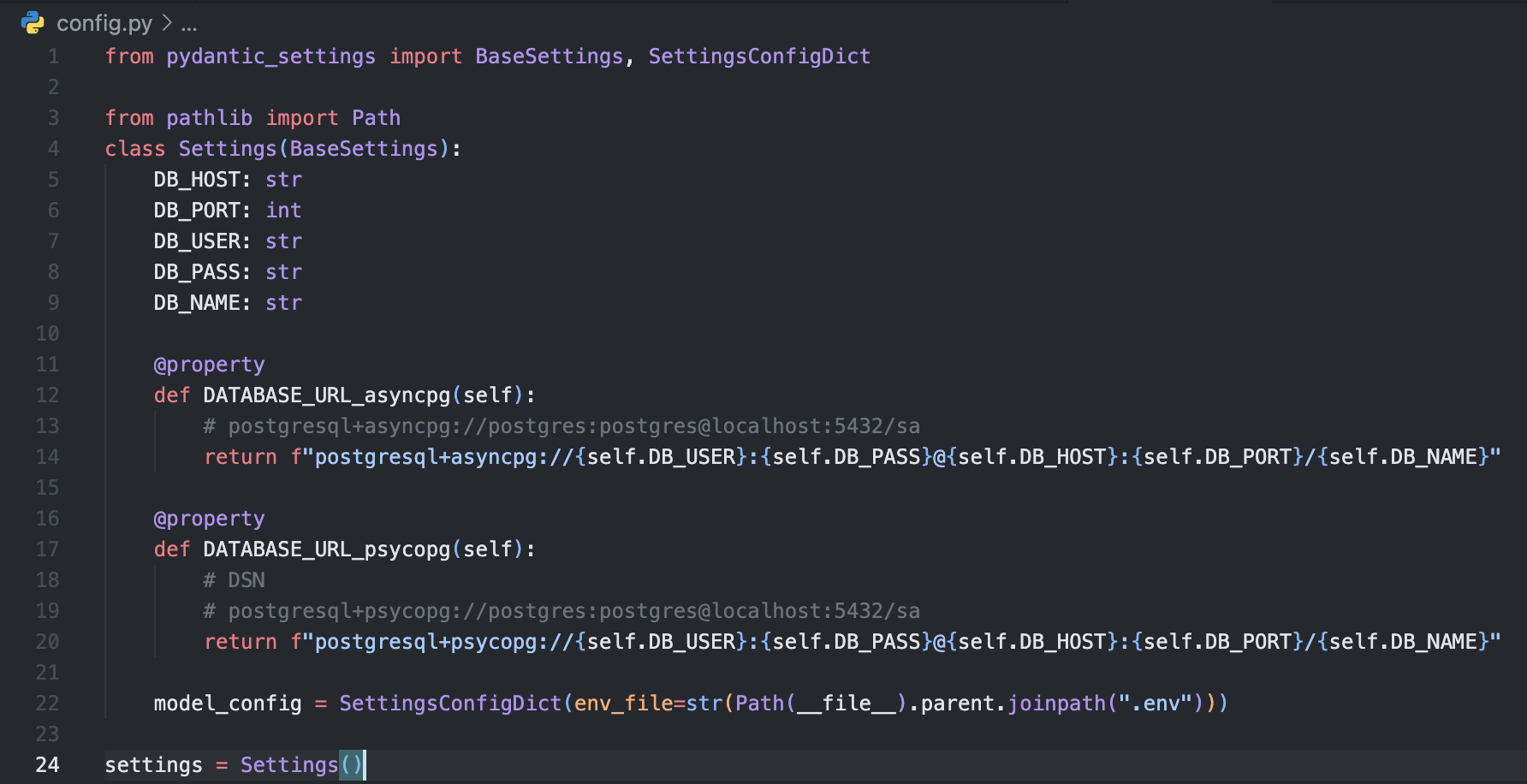


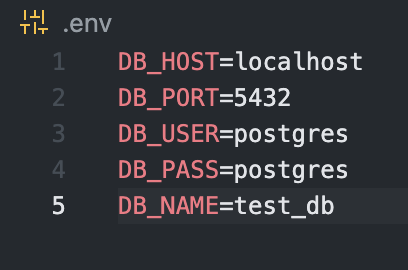


1. Инициализировать миграции







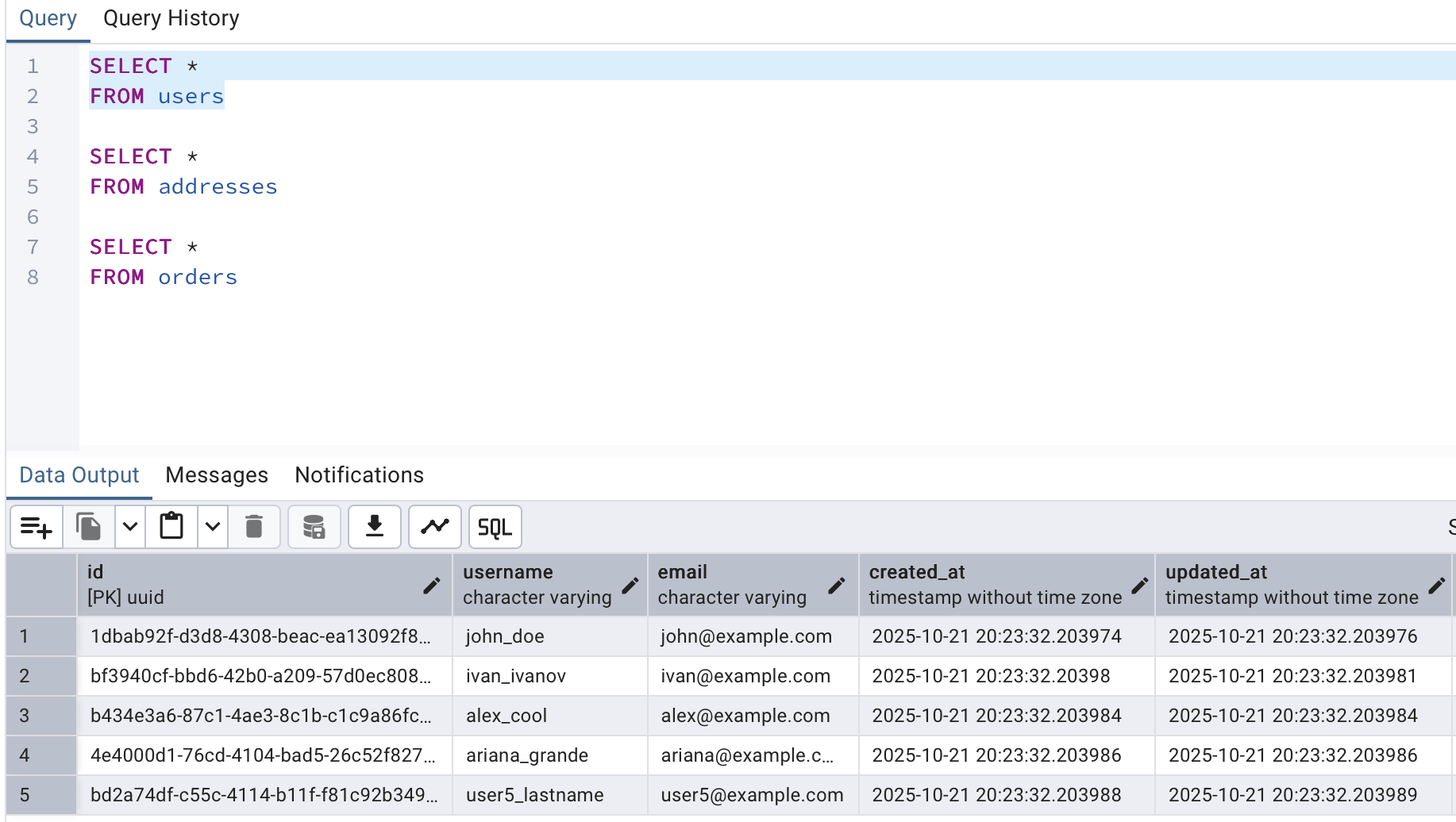


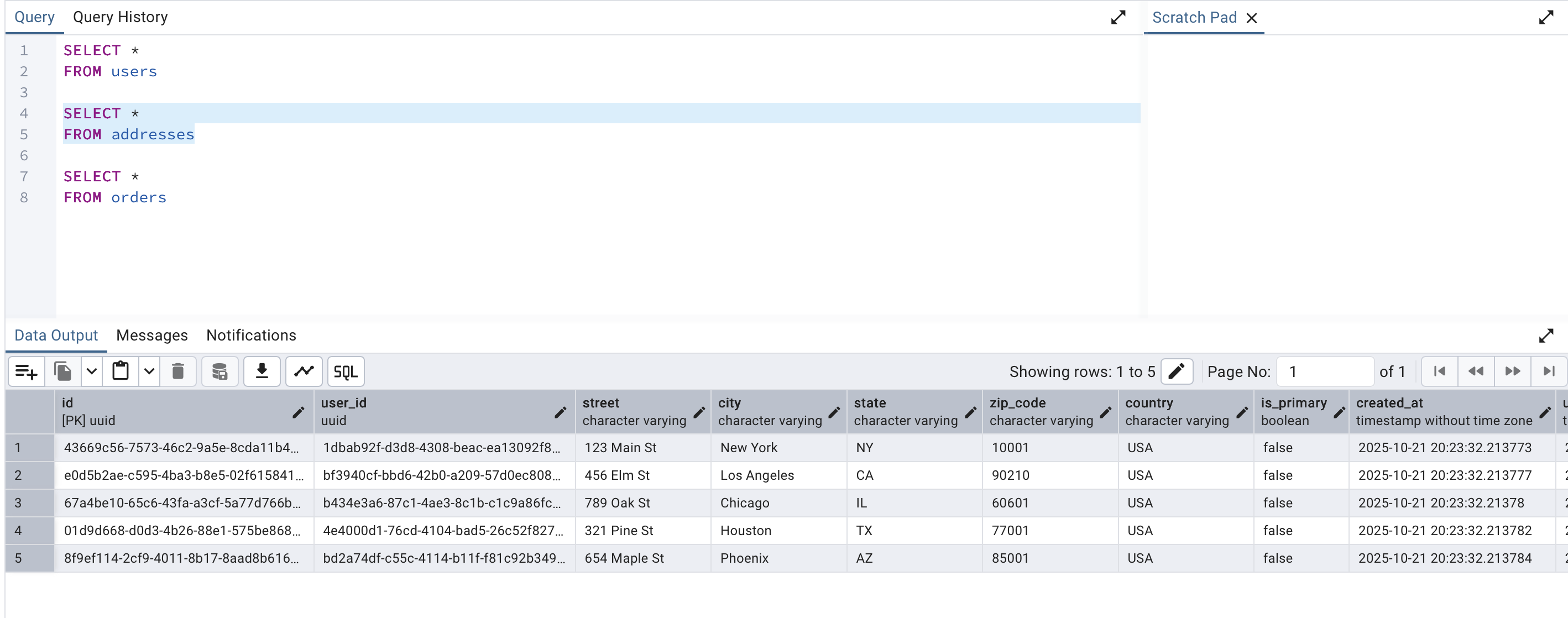
Изображение выглядит как текст, снимок экрана, Шрифт, число

Содержимое, созданное искусственным интеллектом, может быть неверным.

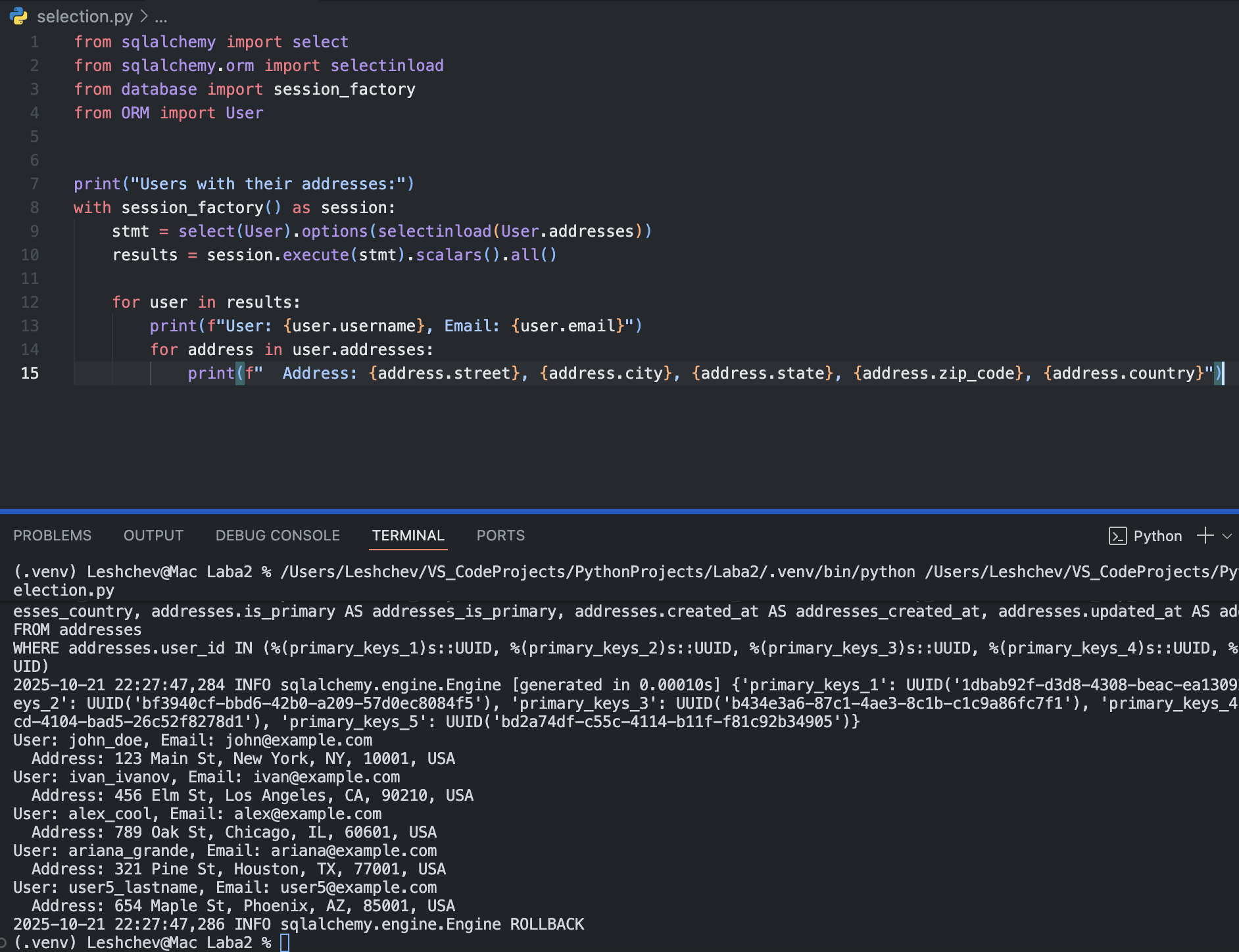
1. Наполнить БД данными
2. from database import async\_session\_factory, session\_factory
3. from ORM import User, Address
4. with session\_factory() as session:
5. user1 = User(username="john\_doe", email="john@example.com")
6. user2 = User(username="ivan\_ivanov", email="ivan@example.com")
7. user3 = User(username="alex\_cool", email="alex@example.com")
8. user4 = User(username="ariana\_grande", email="ariana@example.com")
9. user5 = User(username="user5\_lastname", email="user5@example.com")
11. session.add\_all([user1, user2, user3, user4, user5])
12. session.commit()

15. address1 = Address(
16. user\_id=user1.id, # Связь с user1
17. street="123 Main St",
18. city="New York",
19. state="NY",
20. zip\_code="10001",
21. country="USA"
22. )
23. address2 = Address(
24. user\_id=user2.id, # Связь с user2
25. street="456 Elm St",
26. city="Los Angeles",
27. state="CA",
28. zip\_code="90210",
29. country="USA"
30. )
32. address3 = Address(
33. user\_id=user3.id, # Связь с user3
34. street="789 Oak St",
35. city="Chicago",
36. state="IL",
37. zip\_code="60601",
38. country="USA"
39. )
40. address4 = Address(
41. user\_id=user4.id, # Связь с user4
42. street="321 Pine St",
43. city="Houston",
44. state="TX",
45. zip\_code="77001",
46. country="USA"
47. )
48. address5 = Address(
49. user\_id=user5.id, # Связь с user5
50. street="654 Maple St",
51. city="Phoenix",
52. state="AZ",
53. zip\_code="85001",
54. country="USA"
55. )
56. session.add\_all([address1, address2, address3, address4, address5]) # Добавьте все адреса
57. session.commit()

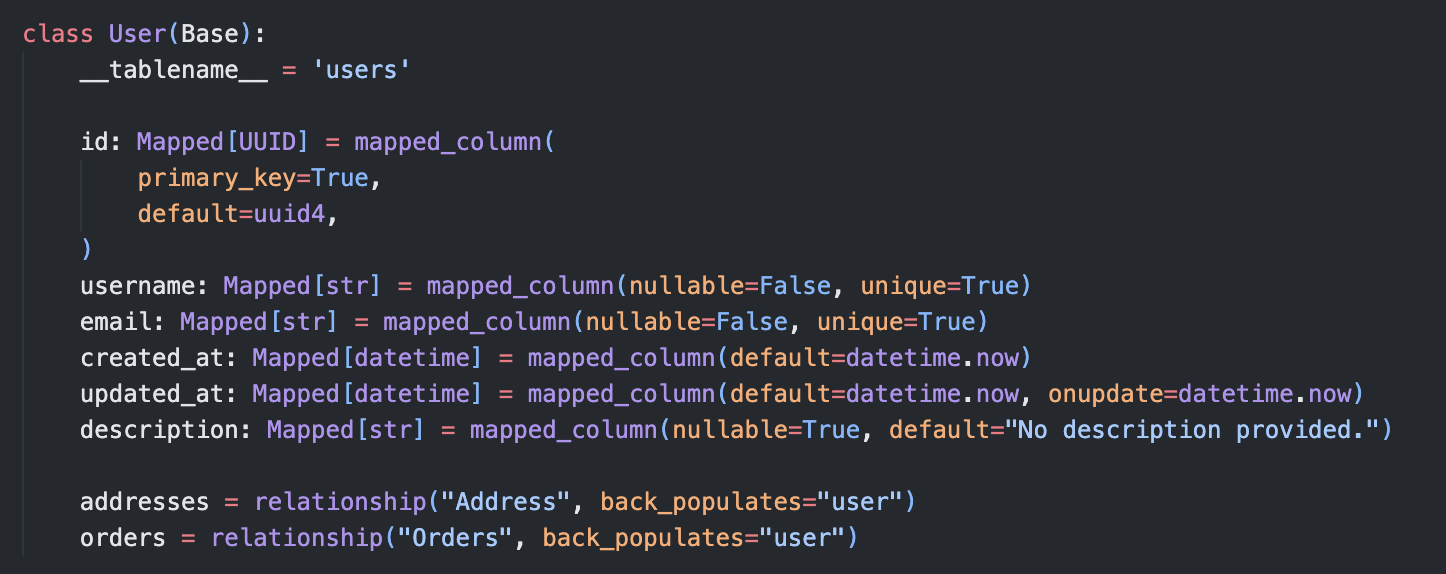




5) Запрос связанных подключений

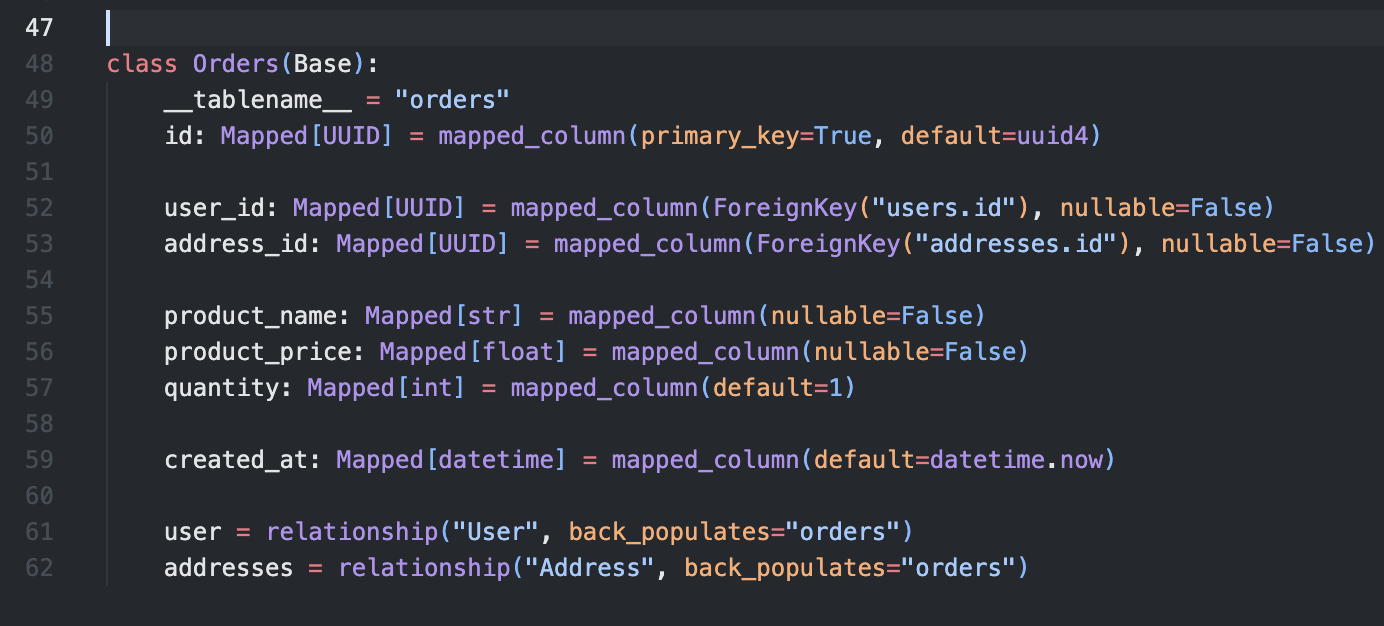


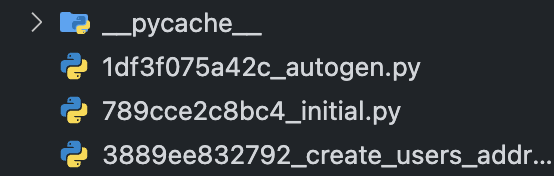
6) Добавить пользователю поле description



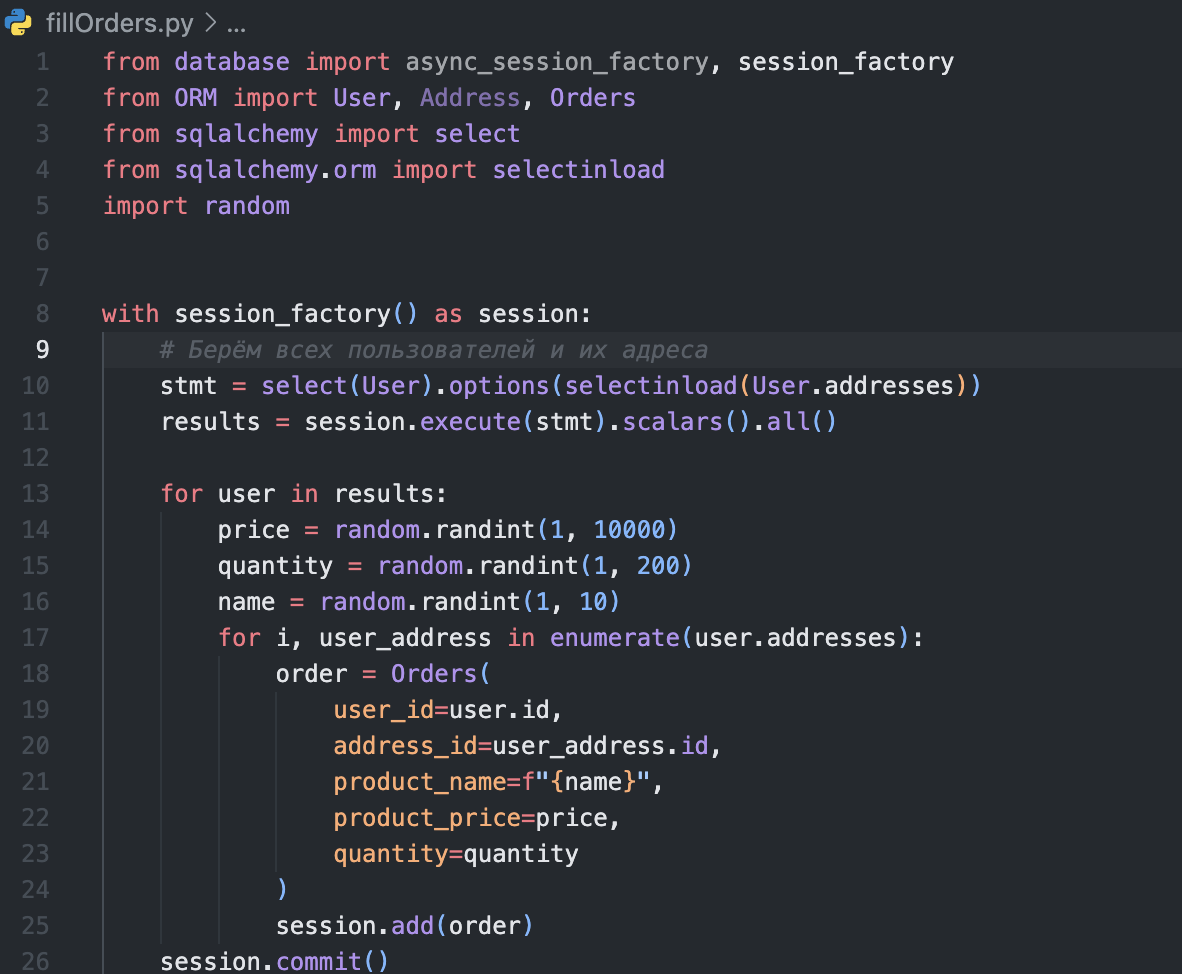


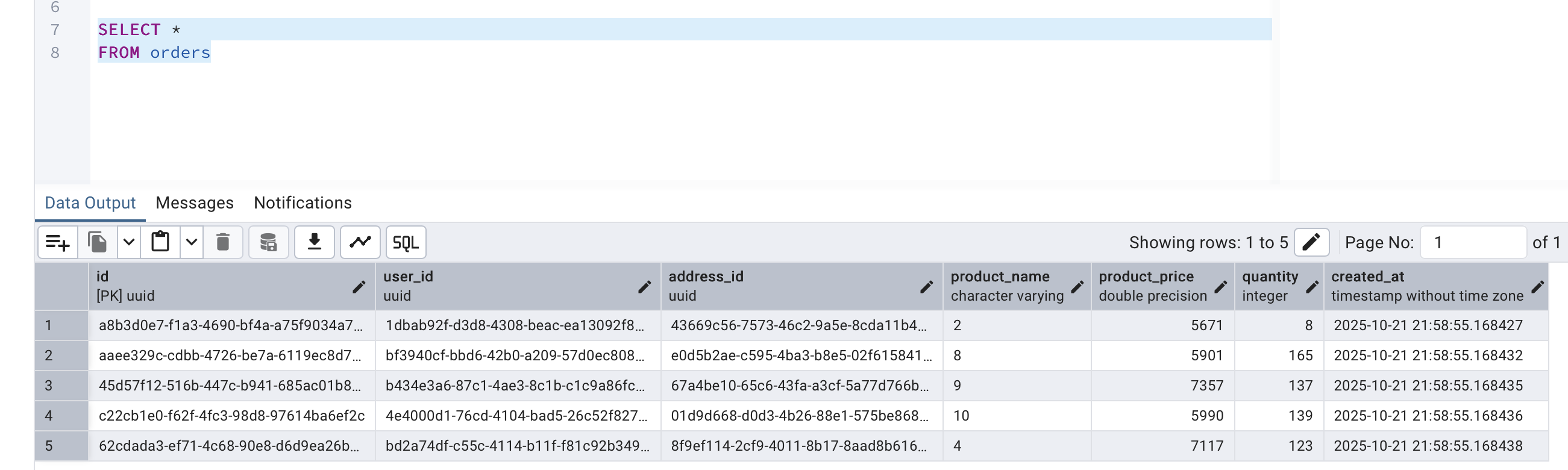
7) Создать таблицу с заказами и произвести миграцию











Изображение выглядит как текст, снимок экрана, Шрифт, число

Содержимое, созданное искусственным интеллектом, может быть неверным.

Ответы на вопросы

1. Какие есть подходы маппинга в SQLAlchemy? Когда следует использовать каждый подход?

Декларативный и классический.

1. Как Alembic отслеживает текущую версию базы данных?

При написании команды alembic revision… создается служебный файл с версией, который лежит в папке versions. В этом файле есть 2 функции upgrade и downgrade, которые указывают на то, что добавилось или ушло в бд.

1. Какие типы связей между таблицами вы реализовали в данной работе?

Один ко многим и многие к одному

1. Что такое миграция базы данных и почему она важна?

Изменение структуры БД. Нужна для контроля версий и облегчения развертывания, а так же командной работы над БД

1. Как обрабатываются отношения многие-ко-многим в SQLAlchemy?

Через промежуточную таблицу со связывающими ключами