# **Лабораторная работа №2: Работа с SQLAlchemy и alembic**

**Студент:** Дрягин Максим Андреевич

**Группа:** Прикладной анализ данных, РИМ-150950

## **Цель работы**

## Целью работы было научиться работать с SQLAlchemy и Alembic: создавать таблицы через ORM, управлять структурой базы данных с помощью миграций и наполнять её данными, а также получать связанные данные между таблицами.

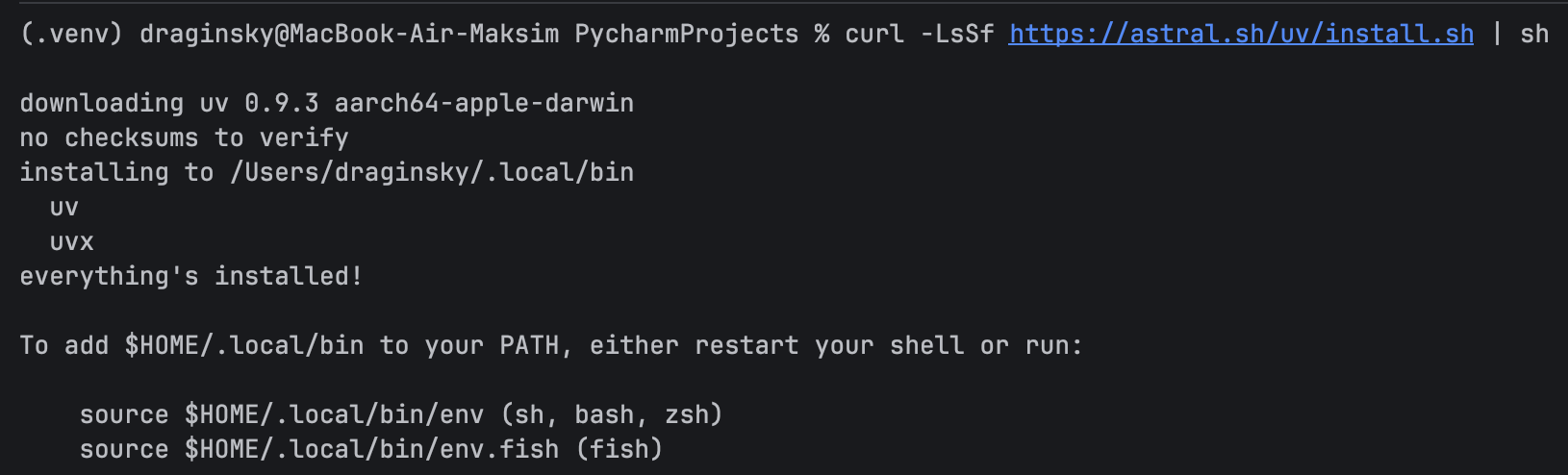
## **Описание задачи**

В ходе работы я создал модели пользователей и адресов, настроил Alembic для миграции, добавил поле description для пользователей, создал таблицы продуктов и заказов, настроил связи между пользователями, адресами, продуктами и заказами. После этого наполнил базу данных реальными объектами — пятью пользователями с адресами, пятью продуктами и пятью заказами.

## **Ход работы**

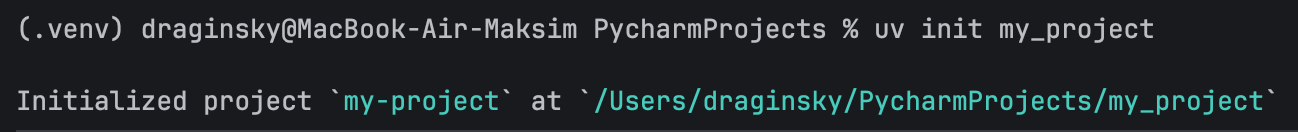
## **Инициализация БД**

Установил менеджер пакетов uv





Создал питон проект



Установил необходимые библиотеки

Изображение выглядит как текст, программное обеспечение, Мультимедийное программное обеспечение, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

Создал ОРМ для пользователя и адреса

Изображение выглядит как текст, снимок экрана, программное обеспечение

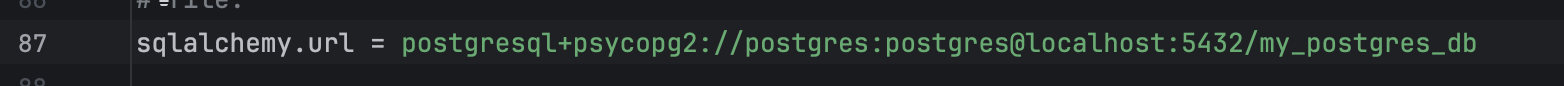
Контент, сгенерированный ИИ, может содержать ошибки. Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Во время инициализации миграций терминал пожаловался, что alembic не установился, установил его ещё раз с флагом --active.  
Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Поменял в alembic.ini строку подключения к нашей БД



В env.py подставил метаданные для миграции  
Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Создал миграцию

Изображение выглядит как текст, снимок экрана, Шрифт, черный

Контент, сгенерированный ИИ, может содержать ошибки.  
Применил к БД последнюю миграцию

Изображение выглядит как текст, Шрифт, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

После этого создал фабрику подключений и наполнил БД данными 5 пользователей и их адресами

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, Шрифт, черный

Контент, сгенерированный ИИ, может содержать ошибки.

## **Запрос связанных данных**

В классе у нас создана связь между таблицами с помощью relationship.

Получил и вывел данные из двух таблиц



Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

## **Последующие работы с БД и миграции**

В ОРМ добавил пользователю дополнительное строковое поле description. Добавил дополнительную таблицу для продукции и заказов. Произвёл миграцию данных и добавил 5 продукций и 5 заказов в БД.

Изображение выглядит как текст, снимок экрана, Шрифт, черный

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, черный, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки. Изображение выглядит как текст, снимок экрана, Шрифт, черно-белый

Контент, сгенерированный ИИ, может содержать ошибки.

## **Вопросы**

1. В SQLAlchemy есть разные подходы маппинга, в основном используют декларативный, потому что он проще и нагляднее.
2. Alembic отслеживает текущую версию базы через таблицу alembic\_version, где хранится идентификатор последней применённой миграции.
3. В работе я использовал связи один-ко-многим и многие-к-одному между пользователями, адресами, заказами и продуктами.
4. Миграция базы — это способ безопасно менять структуру таблиц и колонок без потери данных, и она важна для поддержки проекта.
5. Отношения многие-ко-многим в SQLAlchemy делают через отдельную таблицу-связку, которая хранит пары ключей двух таблиц.
6. Если возникает конфликт версий в Alembic, нужно либо восстановить отсутствующую миграцию, либо заштамповать базу текущей версией (stamp head) и затем создавать новые ревизии.

## **Вывод**

В работе я освоил создание и изменение структуры базы данных через ORM и миграции, научился работать с зависимостями между таблицами, получать связанные данные и наполнять базу реальными объектами. Полученная база готова для дальнейшего использования в проекте, и теперь понимаю, как правильно поддерживать структуру и историю изменений данных.