

## **Лабораторная работа №2**

### **Работа с SQLAlchemy и Alembic**

Студент(ка): Зырянова Татьяна Евгеньевна

Группа: РИМ – 150950

Предмет: Разработка приложений

## Цель работы

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

## Ход работы

### 1) Создание ОРМ для пользователя

```
from datetime import datetime
from uuid import uuid4
from sqlalchemy.orm import declarative_base, relationship, Mapped, mapped_column
from sqlalchemy import String, DateTime, Boolean, ForeignKey
from sqlalchemy.dialects.postgresql import UUID

Base = declarative_base()

class User(Base): 2 usages
    __tablename__ = "users"

    id: Mapped[UUID] = mapped_column(UUID(as_uuid=True), primary_key=True, default=uuid4)
    username: Mapped[str] = mapped_column(String, nullable=False, unique=True)
    email: Mapped[str] = mapped_column(String, nullable=False, unique=True)
    created_at: Mapped[datetime] = mapped_column(DateTime, default=datetime.utcnow)
    updated_at: Mapped[datetime] = mapped_column(DateTime, default=datetime.utcnow, onupdate=datetime.utcnow)

    addresses: Mapped[list["Address"]] = relationship("Address", back_populates="user", cascade="all, delete-orphan")
```

### 2) Инициализация миграций

```
(.venv) PS C:\Users\dom\PycharmProjects\PythonProject5> alembic init migrations
Creating directory 'C:\\Users\\dom\\PycharmProjects\\PythonProject5\\migrations' ... done
Creating directory 'C:\\Users\\dom\\PycharmProjects\\PythonProject5\\migrations\\versions' ... done
Generating C:\\Users\\dom\\PycharmProjects\\PythonProject5\\alembic.ini ... done
Generating C:\\Users\\dom\\PycharmProjects\\PythonProject5\\migrations\\env.py ... done
Generating C:\\Users\\dom\\PycharmProjects\\PythonProject5\\migrations\\README ... done
Generating C:\\Users\\dom\\PycharmProjects\\PythonProject5\\migrations\\script.py.mako ... done
Please edit configuration/connection/logging settings in 'C:\\Users\\dom\\PycharmProjects\\PythonProject5\\alembic.ini' before proceeding.
(.venv) PS C:\\Users\\dom\\PycharmProjects\\PythonProject5> █
```

3) В env.py подставляем метаданные для миграции

```
target_metadata = Base.metadata
```

4) Создание и применение к БД последней миграции

```
(.venv) PS C:\Users\дом\PycharmProjects\PythonProject5> alembic revision --autogenerate
INFO [alembic.runtime.migration] Context impl PostgresqlImpl.
INFO [alembic.runtime.migration] Will assume transactional DDL.
INFO [alembic.autogenerate.compare] Detected added table 'users'
INFO [alembic.autogenerate.compare] Detected added table 'addresses'
Generating C:\Users\дом\PycharmProjects\PythonProject5\migrations\versions\748c2474a6f1_.py ... done
(.venv) PS C:\Users\дом\PycharmProjects\PythonProject5> alembic upgrade head
INFO [alembic.runtime.migration] Context impl PostgresqlImpl.
INFO [alembic.runtime.migration] Will assume transactional DDL.
INFO [alembic.runtime.migration] Running upgrade -> 748c2474a6f1, empty message
(.venv) PS C:\Users\дом\PycharmProjects\PythonProject5>
```

5) Создание фабрики подключений, наполнение БД данными 5 пользователями и их адресами

```
from sqlalchemy import create_engine
from sqlalchemy.orm import sessionmaker

engine = create_engine(
    "postgresql+psycopg2://app:app@localhost:5432/lab2",
    echo=True,
    future=True,
)

SessionLocal = sessionmaker(bind=engine, autoflush=False, autocommit=False)
```

```

from app.db import session_factory
from app.models import User, Address

def run():  # usage
    rows = [
        ("john_doe", "john@example.com", "Baker St 221B", "London", "LDN", "NW1", "UK"),
        ("jane_smith", "jane@example.com", "5th Ave 1", "New York", "NY", "10001", "USA"),
        ("mike_miller", "mike@example.com", "Hauptstr 7", "Berlin", "BE", "10115", "DE"),
        ("olga_ivanova", "olga@example.com", "Nevsky 1", "Saint-Petersburg", "SPB", "190000", "RU"),
        ("li_wei", "li@example.com", "Nanjing Rd 10", "Shanghai", "SH", "200000", "CN"),
    ]
    with session_factory() as session:
        for u, e, st, ct, zipc, c in rows:
            user = User(username=u, email=e)
            user.addresses.append(Address(
                street=st, city=ct, state=stt, zip_code=zipc, country=c, is_primary=True
            ))

```

## 6) Создание запроса связанных данных

```

(.venv) PS C:\Users\дом\PycharmProjects\PythonProject5> python -m app.seed
2025-10-22 23:26:23,267 INFO sqlalchemy.engine.Engine select pg_catalog.version()
2025-10-22 23:26:23,267 INFO sqlalchemy.engine.Engine [raw sql] {}
2025-10-22 23:26:23,273 INFO sqlalchemy.engine.Engine select current_schema()
2025-10-22 23:26:23,273 INFO sqlalchemy.engine.Engine [raw sql] {}
2025-10-22 23:26:23,279 INFO sqlalchemy.engine.Engine show standard_conforming_strings
2025-10-22 23:26:23,279 INFO sqlalchemy.engine.Engine [raw sql] {}
2025-10-22 23:26:23,284 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2025-10-22 23:26:23,284 INFO sqlalchemy.engine.Engine INSERT INTO users (id, username, email, created_at, updated_at) VALUES (%(id_0)s, %(username_0)s, %(email_0)s, %(created_at_0)s, %(updated_at_0)s), %(id_1)s, %(username_1)s, %(email_1)s, %(created_at_1)s, %(updated_at_1)s), %(id_2)s, %(username_2)s, %(email_2)s, %(created_at_2)s, %(updated_at_2)s), %(id_3)s, %(username_3)s, %(email_3)s, %(created_at_3)s, %(updated_at_3)s), %(id_4)s, %(username_4)s, %(email_4)s, %(created_at_4)s, %(updated_at_4)s)
2025-10-22 23:26:23,289 INFO sqlalchemy.engine.Engine [generated by: insertmanyvalues] { 'created_at_0': datetime.datetime(2025, 10, 22, 18, 26, 23, 284388), 'username_0': 'john@example.com', 'id_0': '78cc55a0-4d5c-476f-aff2-79cbcd9d828', 'updated_at_0': datetime.datetime(2025, 10, 22, 18, 26, 23, 284388), 'email_0': 'john_doe', 'created_at_1': datetime.datetime(2025, 10, 22, 18, 26, 23, 284388), 'email_1': 'jane@example.com', 'id_1': '40e4fdf8-bd34-448b-aec0-88ba0bada3', 'updated_at_1': datetime.datetime(2025, 10, 22, 18, 26, 23, 284388), 'username_1': 'jane_smith', 'created_at_2': datetime.datetime(2025, 10, 22, 18, 26, 23, 284388), 'email_2': 'mike@example.com', 'id_2': 'daccbb1b-555b-47bc-9432-0013b2c2ceef7', 'updated_at_2': datetime.datetime(2025, 10, 22, 18, 26, 23, 284388), 'username_2': 'mike_miller', 'created_at_3': datetime.datetime(2025, 10, 22, 18, 26, 23, 284388), 'email_3': 'olga@example.com', 'id_3': 'b7287f2d-fbc5-41f8-bc4b-f5f8200e4bc2', 'updated_at_3': datetime.datetime(2025, 10, 22, 18, 26, 23, 284388), 'username_3': 'olga_ivanova', 'created_at_4': datetime.datetime(2025, 10, 22, 18, 26, 23, 284388), 'email_4': 'li@example.com', 'id_4': '21445591-5a8c-4934-81f9-b69e7fb8785f', 'updated_at_4': datetime.datetime(2025, 10, 22, 18, 26, 23, 284388), 'username_4': 'li_wei' }
2025-10-22 23:26:23,308 INFO sqlalchemy.engine.Engine INSERT INTO addresses (id, user_id, street, city, state, zip_code, country, is_primary, created_at, updated_at) VALUES (%(id_0)s, %(user_id_0)s, %(street_0)s, %(city_0)s, %(state_0)s, %(zip_code_0)s, %(country_0)s, %(is_primary_0)s, %(created_at_0)s, %(updated_at_0)s), %(id_1)s, %(user_id_1)s, %(street_1)s, %(city_1)s, %(state_1)s, %(zip_code_1)s, %(country_1)s, %(is_primary_1)s, %(created_at_1)s, %(updated_at_1)s)
2025-10-22 23:26:23,308 INFO sqlalchemy.engine.Engine [generated by: insertmanyvalues] { 'created_at_0': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'country_0': 'UK', 'id_0': 'bf04918-bd35-49d5-8021-eac9pe8af0fe', 'zip_code_0': 'NW1', 'is_primary_0': True, 'updated_at_0': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_0': 'Baker St 221B', 'state_0': 'LDN', 'city_0': 'London', 'created_at_1': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'user_id_1': '40e4fdf8-bd34-448b-aec0-88ba0bada3', 'country_1': 'USA', 'id_1': '1c50559d8-afb2-4cd5-b31d-9783shf662h7', 'zip_code_1': '10001', 'is_primary_1': True, 'updated_at_1': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_1': '5th Ave 1', 'city_1': 'New York', 'state_1': 'NY', 'country_2': 'RU', 'id_2': 'b7287f2d-fbc5-41f8-bc4b-f5f8200e4bc2', 'zip_code_2': '190000', 'is_primary_2': True, 'updated_at_2': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_2': 'Hauptstr 7', 'city_2': 'Berlin', 'state_2': 'BE', 'country_3': 'CN', 'id_3': '21445591-5a8c-4934-81f9-b69e7fb8785f', 'zip_code_3': '200000', 'is_primary_3': True, 'updated_at_3': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_3': 'Nanjing Rd 10', 'city_3': 'Shanghai', 'state_3': 'SH', 'country_4': 'DE', 'id_4': 'daccbb1b-555b-47bc-9432-0013b2c2ceef7', 'zip_code_4': '10115', 'is_primary_4': True, 'updated_at_4': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_4': 'Neckarstr 11', 'city_4': 'Heidelberg', 'state_4': 'HE', 'country_5': 'FR', 'id_5': '40e4fdf8-bd34-448b-aec0-88ba0bada3', 'zip_code_5': '69115', 'is_primary_5': True, 'updated_at_5': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_5': 'Avenue de la Paix', 'city_5': 'Paris', 'state_5': 'FR', 'country_6': 'ES', 'id_6': 'b7287f2d-fbc5-41f8-bc4b-f5f8200e4bc2', 'zip_code_6': '28000', 'is_primary_6': True, 'updated_at_6': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_6': 'Calle de Alcalá 31', 'city_6': 'Madrid', 'state_6': 'ES', 'country_7': 'IT', 'id_7': '1c50559d8-afb2-4cd5-b31d-9783shf662h7', 'zip_code_7': '08008', 'is_primary_7': True, 'updated_at_7': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_7': 'Via XX Settembre 20', 'city_7': 'Milano', 'state_7': 'IT', 'country_8': 'PT', 'id_8': '21445591-5a8c-4934-81f9-b69e7fb8785f', 'zip_code_8': '10000', 'is_primary_8': True, 'updated_at_8': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_8': 'Avenida da Liberdade 120', 'city_8': 'Lisboa', 'state_8': 'PT', 'country_9': 'NL', 'id_9': 'daccbb1b-555b-47bc-9432-0013b2c2ceef7', 'zip_code_9': '1000', 'is_primary_9': True, 'updated_at_9': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_9': 'Prinsengracht 287', 'city_9': 'Amsterdam', 'state_9': 'NL', 'country_10': 'NO', 'id_10': '40e4fdf8-bd34-448b-aec0-88ba0bada3', 'zip_code_10': '1018', 'is_primary_10': True, 'updated_at_10': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_10': 'Oslo gate 21', 'city_10': 'Oslo', 'state_10': 'NO', 'country_11': 'SE', 'id_11': 'b7287f2d-fbc5-41f8-bc4b-f5f8200e4bc2', 'zip_code_11': '1024', 'is_primary_11': True, 'updated_at_11': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_11': 'Kungsgatan 32', 'city_11': 'Stockholm', 'state_11': 'SE', 'country_12': 'DK', 'id_12': '1c50559d8-afb2-4cd5-b31d-9783shf662h7', 'zip_code_12': '1000', 'is_primary_12': True, 'updated_at_12': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_12': 'Frederiksberg Allé 30', 'city_12': 'Copenhagen', 'state_12': 'DK', 'country_13': 'NO', 'id_13': '21445591-5a8c-4934-81f9-b69e7fb8785f', 'zip_code_13': '1300', 'is_primary_13': True, 'updated_at_13': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_13': 'Oslo gate 21', 'city_13': 'Oslo', 'state_13': 'NO', 'country_14': 'NO', 'id_14': 'daccbb1b-555b-47bc-9432-0013b2c2ceef7', 'zip_code_14': '0100', 'is_primary_14': True, 'updated_at_14': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_14': 'Frederiksberg Allé 30', 'city_14': 'Copenhagen', 'state_14': 'DK', 'country_15': 'DK', 'id_15': '40e4fdf8-bd34-448b-aec0-88ba0bada3', 'zip_code_15': '1300', 'is_primary_15': True, 'updated_at_15': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_15': 'Oslo gate 21', 'city_15': 'Oslo', 'state_15': 'NO', 'country_16': 'DK', 'id_16': 'b7287f2d-fbc5-41f8-bc4b-f5f8200e4bc2', 'zip_code_16': '0100', 'is_primary_16': True, 'updated_at_16': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_16': 'Frederiksberg Allé 30', 'city_16': 'Copenhagen', 'state_16': 'DK', 'country_17': 'DK', 'id_17': '1c50559d8-afb2-4cd5-b31d-9783shf662h7', 'zip_code_17': '1300', 'is_primary_17': True, 'updated_at_17': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_17': 'Oslo gate 21', 'city_17': 'Oslo', 'state_17': 'NO', 'country_18': 'DK', 'id_18': '21445591-5a8c-4934-81f9-b69e7fb8785f', 'zip_code_18': '0100', 'is_primary_18': True, 'updated_at_18': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_18': 'Frederiksberg Allé 30', 'city_18': 'Copenhagen', 'state_18': 'DK', 'country_19': 'DK', 'id_19': 'daccbb1b-555b-47bc-9432-0013b2c2ceef7', 'zip_code_19': '1300', 'is_primary_19': True, 'updated_at_19': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_19': 'Oslo gate 21', 'city_19': 'Oslo', 'state_19': 'NO', 'country_20': 'DK', 'id_20': '40e4fdf8-bd34-448b-aec0-88ba0bada3', 'zip_code_20': '0100', 'is_primary_20': True, 'updated_at_20': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_20': 'Frederiksberg Allé 30', 'city_20': 'Copenhagen', 'state_20': 'DK', 'country_21': 'DK', 'id_21': 'b7287f2d-fbc5-41f8-bc4b-f5f8200e4bc2', 'zip_code_21': '1300', 'is_primary_21': True, 'updated_at_21': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_21': 'Oslo gate 21', 'city_21': 'Oslo', 'state_21': 'NO', 'country_22': 'DK', 'id_22': '1c50559d8-afb2-4cd5-b31d-9783shf662h7', 'zip_code_22': '0100', 'is_primary_22': True, 'updated_at_22': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_22': 'Frederiksberg Allé 30', 'city_22': 'Copenhagen', 'state_22': 'DK', 'country_23': 'DK', 'id_23': '21445591-5a8c-4934-81f9-b69e7fb8785f', 'zip_code_23': '1300', 'is_primary_23': True, 'updated_at_23': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_23': 'Oslo gate 21', 'city_23': 'Oslo', 'state_23': 'NO', 'country_24': 'DK', 'id_24': 'daccbb1b-555b-47bc-9432-0013b2c2ceef7', 'zip_code_24': '0100', 'is_primary_24': True, 'updated_at_24': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_24': 'Frederiksberg Allé 30', 'city_24': 'Copenhagen', 'state_24': 'DK', 'country_25': 'DK', 'id_25': '40e4fdf8-bd34-448b-aec0-88ba0bada3', 'zip_code_25': '0100', 'is_primary_25': True, 'updated_at_25': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_25': 'Oslo gate 21', 'city_25': 'Oslo', 'state_25': 'NO', 'country_26': 'DK', 'id_26': 'b7287f2d-fbc5-41f8-bc4b-f5f8200e4bc2', 'zip_code_26': '0100', 'is_primary_26': True, 'updated_at_26': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_26': 'Frederiksberg Allé 30', 'city_26': 'Copenhagen', 'state_26': 'DK', 'country_27': 'DK', 'id_27': '1c50559d8-afb2-4cd5-b31d-9783shf662h7', 'zip_code_27': '0100', 'is_primary_27': True, 'updated_at_27': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_27': 'Oslo gate 21', 'city_27': 'Oslo', 'state_27': 'NO', 'country_28': 'DK', 'id_28': '21445591-5a8c-4934-81f9-b69e7fb8785f', 'zip_code_28': '0100', 'is_primary_28': True, 'updated_at_28': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_28': 'Frederiksberg Allé 30', 'city_28': 'Copenhagen', 'state_28': 'DK', 'country_29': 'DK', 'id_29': 'daccbb1b-555b-47bc-9432-0013b2c2ceef7', 'zip_code_29': '0100', 'is_primary_29': True, 'updated_at_29': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_29': 'Oslo gate 21', 'city_29': 'Oslo', 'state_29': 'NO', 'country_30': 'DK', 'id_30': '40e4fdf8-bd34-448b-aec0-88ba0bada3', 'zip_code_30': '0100', 'is_primary_30': True, 'updated_at_30': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_30': 'Frederiksberg Allé 30', 'city_30': 'Copenhagen', 'state_30': 'DK', 'country_31': 'DK', 'id_31': 'b7287f2d-fbc5-41f8-bc4b-f5f8200e4bc2', 'zip_code_31': '0100', 'is_primary_31': True, 'updated_at_31': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_31': 'Oslo gate 21', 'city_31': 'Oslo', 'state_31': 'NO', 'country_32': 'DK', 'id_32': '1c50559d8-afb2-4cd5-b31d-9783shf662h7', 'zip_code_32': '0100', 'is_primary_32': True, 'updated_at_32': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_32': 'Frederiksberg Allé 30', 'city_32': 'Copenhagen', 'state_32': 'DK', 'country_33': 'DK', 'id_33': '21445591-5a8c-4934-81f9-b69e7fb8785f', 'zip_code_33': '0100', 'is_primary_33': True, 'updated_at_33': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_33': 'Oslo gate 21', 'city_33': 'Oslo', 'state_33': 'NO', 'country_34': 'DK', 'id_34': 'daccbb1b-555b-47bc-9432-0013b2c2ceef7', 'zip_code_34': '0100', 'is_primary_34': True, 'updated_at_34': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_34': 'Frederiksberg Allé 30', 'city_34': 'Copenhagen', 'state_34': 'DK', 'country_35': 'DK', 'id_35': '40e4fdf8-bd34-448b-aec0-88ba0bada3', 'zip_code_35': '0100', 'is_primary_35': True, 'updated_at_35': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_35': 'Oslo gate 21', 'city_35': 'Oslo', 'state_35': 'NO', 'country_36': 'DK', 'id_36': 'b7287f2d-fbc5-41f8-bc4b-f5f8200e4bc2', 'zip_code_36': '0100', 'is_primary_36': True, 'updated_at_36': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_36': 'Frederiksberg Allé 30', 'city_36': 'Copenhagen', 'state_36': 'DK', 'country_37': 'DK', 'id_37': '1c50559d8-afb2-4cd5-b31d-9783shf662h7', 'zip_code_37': '0100', 'is_primary_37': True, 'updated_at_37': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_37': 'Oslo gate 21', 'city_37': 'Oslo', 'state_37': 'NO', 'country_38': 'DK', 'id_38': '21445591-5a8c-4934-81f9-b69e7fb8785f', 'zip_code_38': '0100', 'is_primary_38': True, 'updated_at_38': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_38': 'Frederiksberg Allé 30', 'city_38': 'Copenhagen', 'state_38': 'DK', 'country_39': 'DK', 'id_39': 'daccbb1b-555b-47bc-9432-0013b2c2ceef7', 'zip_code_39': '0100', 'is_primary_39': True, 'updated_at_39': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_39': 'Oslo gate 21', 'city_39': 'Oslo', 'state_39': 'NO', 'country_40': 'DK', 'id_40': '40e4fdf8-bd34-448b-aec0-88ba0bada3', 'zip_code_40': '0100', 'is_primary_40': True, 'updated_at_40': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_40': 'Frederiksberg Allé 30', 'city_40': 'Copenhagen', 'state_40': 'DK', 'country_41': 'DK', 'id_41': 'b7287f2d-fbc5-41f8-bc4b-f5f8200e4bc2', 'zip_code_41': '0100', 'is_primary_41': True, 'updated_at_41': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_41': 'Oslo gate 21', 'city_41': 'Oslo', 'state_41': 'NO', 'country_42': 'DK', 'id_42': '1c50559d8-afb2-4cd5-b31d-9783shf662h7', 'zip_code_42': '0100', 'is_primary_42': True, 'updated_at_42': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_42': 'Frederiksberg Allé 30', 'city_42': 'Copenhagen', 'state_42': 'DK', 'country_43': 'DK', 'id_43': '21445591-5a8c-4934-81f9-b69e7fb8785f', 'zip_code_43': '0100', 'is_primary_43': True, 'updated_at_43': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_43': 'Oslo gate 21', 'city_43': 'Oslo', 'state_43': 'NO', 'country_44': 'DK', 'id_44': 'daccbb1b-555b-47bc-9432-0013b2c2ceef7', 'zip_code_44': '0100', 'is_primary_44': True, 'updated_at_44': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_44': 'Frederiksberg Allé 30', 'city_44': 'Copenhagen', 'state_44': 'DK', 'country_45': 'DK', 'id_45': '40e4fdf8-bd34-448b-aec0-88ba0bada3', 'zip_code_45': '0100', 'is_primary_45': True, 'updated_at_45': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_45': 'Oslo gate 21', 'city_45': 'Oslo', 'state_45': 'NO', 'country_46': 'DK', 'id_46': 'b7287f2d-fbc5-41f8-bc4b-f5f8200e4bc2', 'zip_code_46': '0100', 'is_primary_46': True, 'updated_at_46': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_46': 'Frederiksberg Allé 30', 'city_46': 'Copenhagen', 'state_46': 'DK', 'country_47': 'DK', 'id_47': '1c50559d8-afb2-4cd5-b31d-9783shf662h7', 'zip_code_47': '0100', 'is_primary_47': True, 'updated_at_47': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_47': 'Oslo gate 21', 'city_47': 'Oslo', 'state_47': 'NO', 'country_48': 'DK', 'id_48': '21445591-5a8c-4934-81f9-b69e7fb8785f', 'zip_code_48': '0100', 'is_primary_48': True, 'updated_at_48': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_48': 'Frederiksberg Allé 30', 'city_48': 'Copenhagen', 'state_48': 'DK', 'country_49': 'DK', 'id_49': 'daccbb1b-555b-47bc-9432-0013b2c2ceef7', 'zip_code_49': '0100', 'is_primary_49': True, 'updated_at_49': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_49': 'Oslo gate 21', 'city_49': 'Oslo', 'state_49': 'NO', 'country_50': 'DK', 'id_50': '40e4fdf8-bd34-448b-aec0-88ba0bada3', 'zip_code_50': '0100', 'is_primary_50': True, 'updated_at_50': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_50': 'Frederiksberg Allé 30', 'city_50': 'Copenhagen', 'state_50': 'DK', 'country_51': 'DK', 'id_51': 'b7287f2d-fbc5-41f8-bc4b-f5f8200e4bc2', 'zip_code_51': '0100', 'is_primary_51': True, 'updated_at_51': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_51': 'Oslo gate 21', 'city_51': 'Oslo', 'state_51': 'NO', 'country_52': 'DK', 'id_52': '1c50559d8-afb2-4cd5-b31d-9783shf662h7', 'zip_code_52': '0100', 'is_primary_52': True, 'updated_at_52': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_52': 'Frederiksberg Allé 30', 'city_52': 'Copenhagen', 'state_52': 'DK', 'country_53': 'DK', 'id_53': '21445591-5a8c-4934-81f9-b69e7fb8785f', 'zip_code_53': '0100', 'is_primary_53': True, 'updated_at_53': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_53': 'Oslo gate 21', 'city_53': 'Oslo', 'state_53': 'NO', 'country_54': 'DK', 'id_54': 'daccbb1b-555b-47bc-9432-0013b2c2ceef7', 'zip_code_54': '0100', 'is_primary_54': True, 'updated_at_54': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_54': 'Frederiksberg Allé 30', 'city_54': 'Copenhagen', 'state_54': 'DK', 'country_55': 'DK', 'id_55': '40e4fdf8-bd34-448b-aec0-88ba0bada3', 'zip_code_55': '0100', 'is_primary_55': True, 'updated_at_55': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_55': 'Oslo gate 21', 'city_55': 'Oslo', 'state_55': 'NO', 'country_56': 'DK', 'id_56': 'b7287f2d-fbc5-41f8-bc4b-f5f8200e4bc2', 'zip_code_56': '0100', 'is_primary_56': True, 'updated_at_56': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_56': 'Frederiksberg Allé 30', 'city_56': 'Copenhagen', 'state_56': 'DK', 'country_57': 'DK', 'id_57': '1c50559d8-afb2-4cd5-b31d-9783shf662h7', 'zip_code_57': '0100', 'is_primary_57': True, 'updated_at_57': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_57': 'Oslo gate 21', 'city_57': 'Oslo', 'state_57': 'NO', 'country_58': 'DK', 'id_58': '21445591-5a8c-4934-81f9-b69e7fb8785f', 'zip_code_58': '0100', 'is_primary_58': True, 'updated_at_58': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_58': 'Frederiksberg Allé 30', 'city_58': 'Copenhagen', 'state_58': 'DK', 'country_59': 'DK', 'id_59': 'daccbb1b-555b-47bc-9432-0013b2c2ceef7', 'zip_code_59': '0100', 'is_primary_59': True, 'updated_at_59': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_59': 'Oslo gate 21', 'city_59': 'Oslo', 'state_59': 'NO', 'country_60': 'DK', 'id_60': '40e4fdf8-bd34-448b-aec0-88ba0bada3', 'zip_code_60': '0100', 'is_primary_60': True, 'updated_at_60': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_60': 'Frederiksberg Allé 30', 'city_60': 'Copenhagen', 'state_60': 'DK', 'country_61': 'DK', 'id_61': 'b7287f2d-fbc5-41f8-bc4b-f5f8200e4bc2', 'zip_code_61': '0100', 'is_primary_61': True, 'updated_at_61': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_61': 'Oslo gate 21', 'city_61': 'Oslo', 'state_61': 'NO', 'country_62': 'DK', 'id_62': '1c50559d8-afb2-4cd5-b31d-9783shf662h7', 'zip_code_62': '0100', 'is_primary_62': True, 'updated_at_62': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_62': 'Frederiksberg Allé 30', 'city_62': 'Copenhagen', 'state_62': 'DK', 'country_63': 'DK', 'id_63': '21445591-5a8c-4934-81f9-b69e7fb8785f', 'zip_code_63': '0100', 'is_primary_63': True, 'updated_at_63': datetime.datetime(2025, 10, 22, 18, 26, 23, 307810), 'street_63': 'Oslo gate 21', 'city_63': 'Oslo', 'state_63': 'NO', 'country_64': 'DK', 'id_64': 'daccbb1b-555b-47bc-9432-0013b2c2ceef7', 'zip_code_64': '0100', 'is_primary_64': True, 'updated_at_64': datetime.datetime(202
```

## 7) Запрос связанных данных

```
from sqlalchemy import select
from sqlalchemy.orm import selectinload

from app.db import session_factory
from app.models import User

def main(): 1 usage
    with session_factory() as session:
        stmt = select(User).options(selectinload(User.addresses))
        for u in session.scalars(stmt):
            print(u.username, "->", [a.street for a in u.addresses])

> if __name__ == "__main__":
    main()
```

```
(.venv) PS C:\Users\dom\PycharmProjects\PythonProject5> python -m app.query_examples
2025-10-22 23:30:36,593 INFO sqlalchemy.engine.Engine select pg_catalog.version()
2025-10-22 23:30:36,593 INFO sqlalchemy.engine.Engine [raw sql] {}
2025-10-22 23:30:36,596 INFO sqlalchemy.engine.Engine select current_schema()
2025-10-22 23:30:36,597 INFO sqlalchemy.engine.Engine [raw sql] {}
2025-10-22 23:30:36,599 INFO sqlalchemy.engine.Engine show standard_conforming_strings
2025-10-22 23:30:36,599 INFO sqlalchemy.engine.Engine [raw sql] {}
2025-10-22 23:30:36,603 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2025-10-22 23:30:36,606 INFO sqlalchemy.engine.Engine SELECT users.id, users.username, users.email, users.created_at, users.updated_at
FROM users
2025-10-22 23:30:36,606 INFO sqlalchemy.engine.Engine [generated in 0.00043s] {}
2025-10-22 23:30:36,611 INFO sqlalchemy.engine.Engine SELECT addresses.user_id AS addresses_user_id, addresses.id AS addresses_id, addresses.street
AS addresses_street, addresses.city AS addresses_city, addresses.state AS addresses_state, addresses.zip_code AS addresses_zip_code, addresses.county
AS addresses_country, addresses.is_primary AS addresses_is_primary, addresses.created_at AS addresses_created_at, addresses.updated_at AS address
es_updated_at
FROM addresses
WHERE addresses.user_id IN (%(primary_keys_1)s, %(primary_keys_2)s, %(primary_keys_3)s, %(primary_keys_4)s, %(primary_keys_5)s)
2025-10-22 23:30:36,611 INFO sqlalchemy.engine.Engine [generated in 0.00056s] {'primary_keys_1': '78cc55a0-4d5c-476f-aff2-79cbcd9d828', 'primary_ke
ys_2': '40e4fdf8-bd34-4488-aec0-88ba0badba3', 'primary_keys_3': 'daccb8b1-555b-476c-9432-0013b2c2eeff', 'primary_keys_4': 'b7287f2d-fbc5-41f8-bc4b-
f5fb200e4bc2', 'primary_keys_5': '21445591-5a8c-4934-81f9-b69e7fb8785f'}
john_doe -> ['Baker St 221B']
jane_smith -> ['5th Ave 1']
mike_miller -> ['Hauptstr 7']
olga_ivanova -> ['Nevsky 1']
li_wei -> ['Nanjing Rd 10']
2025-10-22 23:30:36,626 INFO sqlalchemy.engine.Engine ROLLBACK
(.venv) PS C:\Users\dom\PycharmProjects\PythonProject5>
```

## 8) Последующие работы с БД и миграции

В ОРМ добавляем пользователю дополнительное строковое поле `description`. Добавляем дополнительную таблицу для продукции и заказов. Заказ должен в себе содержать информацию о пользователе, адресе доставки и продукции. Производим миграцию данных и добавляем 5 продукции и 5 заказов в БД.

```
from datetime import datetime
from uuid import uuid4, UUID as UUID_TYPE
from typing import List, Optional
from sqlalchemy.orm import declarative_base, relationship, Mapped, mapped_column
from sqlalchemy import String, DateTime, Boolean, ForeignKey, Integer
from sqlalchemy.dialects.postgresql import UUID as PG_UUID

Base = declarative_base()

class User(Base): 7 usages
    __tablename__ = "users"
    id: Mapped[UUID_TYPE] = mapped_column(PG_UUID(as_uuid=True), primary_key=True, default=uuid4)
    username: Mapped[str] = mapped_column(String, nullable=False, unique=True)
    email: Mapped[str] = mapped_column(String, nullable=False, unique=True)
    description: Mapped[Optional[str]] = mapped_column(String, nullable=True)
    created_at: Mapped[datetime] = mapped_column(DateTime, default=datetime.utcnow)
    updated_at: Mapped[datetime] = mapped_column(DateTime, default=datetime.utcnow, onupdate=datetime.utcnow)
    addresses: Mapped[List["Address"]] = relationship("Address", back_populates="user", cascade="all, delete-orphan")

class Address(Base): 4 usages
    __tablename__ = "addresses"
    id: Mapped[UUID_TYPE] = mapped_column(PG_UUID(as_uuid=True), primary_key=True, default=uuid4)
    user_id: Mapped[UUID_TYPE] = mapped_column(PG_UUID(as_uuid=True), ForeignKey("users.id"), nullable=False)
    street: Mapped[str] = mapped_column(String, nullable=False)
    city: Mapped[str] = mapped_column(String, nullable=False)
    state: Mapped[str] = mapped_column(String, nullable=False)
    zip_code: Mapped[int] = mapped_column(Integer, nullable=False)
```

## Наполнение БД: добавлены 5 пользователей и их адреса

```
(.venv) PS C:\Users\дом\PycharmProjects\PythonProject5> python -m app.seed
2025-10-22 23:36:37,088 INFO sqlalchemy.engine.Engine select pg_catalog.version()
2025-10-22 23:36:37,088 INFO sqlalchemy.engine.Engine [raw sql] {}
2025-10-22 23:36:37,092 INFO sqlalchemy.engine.Engine select current_schema()
2025-10-22 23:36:37,093 INFO sqlalchemy.engine.Engine [raw sql] {}
2025-10-22 23:36:37,096 INFO sqlalchemy.engine.Engine show standard_conforming_strings
2025-10-22 23:36:37,096 INFO sqlalchemy.engine.Engine [raw sql] {}
2025-10-22 23:36:37,100 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2025-10-22 23:36:37,104 INFO sqlalchemy.engine.Engine INSERT INTO users (id, username, email, description, created_at, updated_at) VALUES (%(id_0::UUID, %(username_0)s, %(email_0)s, %(description_0)s, %(created_at_0)s, %(updated_at_0)s), %(id_1)s::UUID, %(username_1)s, %(email_1)s, %(description_1)s, %(created_at_1)s, %(updated_at_1)s)
2025-10-22 23:36:37,104 INFO sqlalchemy.engine.Engine [generated in 0.00027s (insertmanyvalues) 1/1 (unordered)] {'id_0': UUID('21120b54-7974-4bf4-9e3c-26a481afab'), 'id_1': UUID('545616f2-7422-47b3-919b-28181ebefbe'), 'description_0': None, 'updated_at_0': datetime.datetime(2025, 10, 22, 18, 36, 37, 104436), 'created_at_0': datetime.datetime(2025, 10, 22, 18, 36, 37, 104436), 'email_0': 'john@example.com', 'username_0': 'john_doe', 'id_1': UUID('455616f2-7422-47b3-919b-28181ebefbe'), 'description_1': None, 'updated_at_1': datetime.datetime(2025, 10, 22, 18, 36, 37, 104436), 'created_at_1': datetime.datetime(2025, 10, 22, 18, 36, 37, 104436), 'email_1': 'jane@example.com', 'username_1': 'jane_smith', 'id_2': UUID('564cd332-b4a8-45c4-922c-a63659a0972'), 'description_2': None, 'updated_at_2': datetime.datetime(2025, 10, 22, 18, 36, 37, 104436), 'created_at_2': datetime.datetime(2025, 10, 22, 18, 36, 37, 104436), 'email_2': 'mike@example.com', 'username_2': 'mike_miller', 'id_3': UUID('c3f7400f-2ee2-4cc4-afff-c45f40e40742'), 'description_3': None, 'updated_at_3': datetime.datetime(2025, 10, 22, 18, 36, 37, 104436), 'created_at_3': datetime.datetime(2025, 10, 22, 18, 36, 37, 104436), 'email_3': 'olga@example.com', 'username_3': 'olga_ivanova', 'id_4': UUID('318e3942-8b17-454c-9900-0792e3db3d79'), 'description_4': None, 'updated_at_4': datetime.datetime(2025, 10, 22, 18, 36, 37, 104436), 'created_at_4': datetime.datetime(2025, 10, 22, 18, 36, 37, 104436), 'email_4': 'li@example.com', 'username_4': 'li_wei'}
2025-10-22 23:36:37,114 INFO sqlalchemy.engine.Engine INSERT INTO addresses (id, user_id, street, city, state, zip_code, country, is_primary, created_at, updated_at) VALUES (%(id_0)s::UUID, %(user_id_0)s::UUID, %(street_0)s, %(city_0)s, %(state_0)s, %(zip_code_0)s, %(country_0)s, %(is_primary_0)s)
2025-10-22 23:36:37,114 INFO sqlalchemy.engine.Engine [generated in 0.00034s (insertmanyvalues) 1/1 (unordered)] {'state_0': 'LDN', 'id_0': UUID('0924151a-26e4-43c8-85d3-ab79629cd3c9'), 'street_0': 'Baker St 221B', 'zip_code_0': 'NW1', 'is_primary_0': True, 'country_0': 'UK', 'updated_at_0': datetime.datetime(2025, 10, 22, 18, 36, 37, 113870), 'user_id_0': UUID('21120b54-7974-4bf4-9e3c-26a481afab'), 'created_at_0': datetime.datetime(2025, 10, 22, 18, 36, 37, 113870), 'id_1': UUID('1276b2c6-761c-4d70-a67d-78f951493a64'), 'street_1': 'Ave 1', 'zip_code_1': '1100011', 'is_primary_1': True, 'country_1': 'USA', 'updated_at_1': datetime.datetime(2025, 10, 22, 18, 36, 37, 113870)}
2025-10-22 23:44:34 migrations > versions > 5e5a20b7e28a.py
```

## Наполнение БД товарами

```
(.venv) PS C:\Users\дом\PycharmProjects\PythonProject5> python -m app.seed_products
2025-10-22 23:44:34,359 INFO sqlalchemy.engine.Engine select pg_catalog.version()
2025-10-22 23:44:34,359 INFO sqlalchemy.engine.Engine [raw sql] {}
2025-10-22 23:44:34,365 INFO sqlalchemy.engine.Engine select current_schema()
2025-10-22 23:44:34,365 INFO sqlalchemy.engine.Engine [raw sql] {}
2025-10-22 23:44:34,369 INFO sqlalchemy.engine.Engine show standard_conforming_strings
2025-10-22 23:44:34,369 INFO sqlalchemy.engine.Engine [raw sql] {}
2025-10-22 23:44:34,373 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2025-10-22 23:44:34,379 INFO sqlalchemy.engine.Engine SELECT products.title AS products_title
FROM products
2025-10-22 23:44:34,379 INFO sqlalchemy.engine.Engine [generated in 0.00053s] {}
2025-10-22 23:44:34,394 INFO sqlalchemy.engine.Engine INSERT INTO products (id, title, price_cents) VALUES (%(id_0)s::UUID, %(title_0)s, %(price_cents_0)s), %(id_1)s::UUID, %(title_1)s, %(price_cents_1)s), %(id_2)s::UUID, %(title_2)s, %(price_cents_2)s), %(id_3)s::UUID, %(title_3)s, %(price_cents_3)s), %(id_4)s::UUID, %(title_4)s, %(price_cents_4)s)
2025-10-22 23:44:34,395 INFO sqlalchemy.engine.Engine [generated in 0.00029s (insertmanyvalues) 1/1 (unordered)] {'price_cents_0': 1500, 'id_0': UUID('a391fde4-7d4a-4003-81a2-40ee6e6ab20b'), 'title_0': 'Book', 'price_cents_1': 200, 'id_1': UUID('80b8c7fb-7330-4c3c-b7f9-4b16b2d2f7e3'), 'title_1': 'Pen', 'price_cents_2': 150000, 'id_2': UUID('756c96e0-6df6-426a-a239-fc1f172a4e30'), 'title_2': 'Laptop', 'price_cents_3': 4500, 'id_3': UUID('0a428af9-6daf-47a3-9c78-fd7361a59c1'), 'title_3': 'Backpack', 'price_cents_4': 2500, 'id_4': UUID('3fa0414b-c9e7-4e19-a9c3-5df59f7d6432')}
```

## Создание заказов

```
(.venv) PS C:\Users\дом\PycharmProjects\PythonProject5> python -m app.seed_orders
2025-10-22 23:44:48,030 INFO sqlalchemy.engine.Engine select pg_catalog.version()
2025-10-22 23:44:48,031 INFO sqlalchemy.engine.Engine [raw sql] {}
2025-10-22 23:44:48,037 INFO sqlalchemy.engine.Engine select current_schema()
2025-10-22 23:44:48,040 INFO sqlalchemy.engine.Engine [raw sql] {}
2025-10-22 23:44:48,046 INFO sqlalchemy.engine.Engine show standard_conforming_strings
2025-10-22 23:44:48,046 INFO sqlalchemy.engine.Engine [raw sql] {}
2025-10-22 23:44:48,053 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2025-10-22 23:44:48,066 INFO sqlalchemy.engine.Engine SELECT users.id AS users_id, users.username AS users_username, users.email AS users_email, users.description AS users_description, users.created_at AS users_created_at, users.updated_at AS users_updated_at
FROM users ORDER BY users.username
LIMIT %(param_1)s
2025-10-22 23:44:48,066 INFO sqlalchemy.engine.Engine [generated in 0.00115s] {'param_1': 5}
2025-10-22 23:44:48,087 INFO sqlalchemy.engine.Engine SELECT products.id AS products_id, products.title AS products_title, products.price_cents AS products_price_cents
FROM products ORDER BY products.title
LIMIT %(param_1)s
2025-10-22 23:44:48,087 INFO sqlalchemy.engine.Engine [generated in 0.00102s] {'param_1': 5}
2025-10-22 23:44:48,108 INFO sqlalchemy.engine.Engine SELECT addresses.id AS addresses_id, addresses.user_id AS addresses_user_id, addresses.street AS addresses_street, addresses.city AS addresses_city, addresses.state AS addresses_state, addresses.zip_code AS addresses_zip_code, addresses.country AS addresses_country, addresses.is_primary AS addresses_is_primary, addresses.created_at AS addresses_created_at, addresses.updated_at AS addresses_updated_at
FROM addresses
```

## Обновление описаний пользователей

```
AttributeError: type object 'Order' has no attribute 'user'
(.venv) PS C:\Users\dom\PycharmProjects\PythonProject5> python -m app.update_descriptions
2025-10-22 23:45:23,996 INFO sqlalchemy.engine.Engine select pg_catalog.version()
2025-10-22 23:45:23,996 INFO sqlalchemy.engine.Engine [raw sql] {}
2025-10-22 23:45:24,002 INFO sqlalchemy.engine.Engine select current_schema()
2025-10-22 23:45:24,003 INFO sqlalchemy.engine.Engine [raw sql] {}
2025-10-22 23:45:24,007 INFO sqlalchemy.engine.Engine show standard_conforming_strings
2025-10-22 23:45:24,007 INFO sqlalchemy.engine.Engine [raw sql] {}
2025-10-22 23:45:24,012 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2025-10-22 23:45:24,017 INFO sqlalchemy.engine.Engine SELECT users.id AS users_id, users.username AS users_username, users.email AS users_email,
rs.description AS users_description, users.created_at AS users_created_at, users.updated_at AS users_updated_at
FROM users
WHERE users.username IN (%(username_1_1)s, %(username_1_2)s)
2025-10-22 23:45:24,017 INFO sqlalchemy.engine.Engine [generated in 0.00070s] {'username_1_1': 'john_doe', 'username_1_2': 'jane_smith'}
2025-10-22 23:45:24,038 INFO sqlalchemy.engine.Engine UPDATE users SET description=%(description)s, updated_at=%(updated_at)s WHERE users.id = %(rs_id)s:UUID
2025-10-22 23:45:24,039 INFO sqlalchemy.engine.Engine [generated in 0.00069s] [{"description": "VIP customer from NY", "updated_at": datetime.datetime(2025, 10, 22, 18, 45, 24, 38583), "users_id": UUID('21120b54-7974-4bf3-b49f-e5c26a481afb')}, {"description": "Frequent buyer", "updated_at": datetime.datetime(2025, 10, 22, 18, 45, 24, 38583), "users_id": UUID('455616f2-7422-47b3-919b-28181ebefbe2')}]
2025-10-22 23:45:24,045 INFO sqlalchemy.engine.Engine COMMIT
Обновлено пользователей (description): 2
(.venv) PS C:\Users\dom\PycharmProjects\PythonProject5>
```

## **Ответы на вопросы**

1. Подходы маппинга в SQLAlchemy: Declarative (классы), Classical (ручное сопоставление таблиц и классов) и Automap (автогенерация классов по существующей схеме). Declarative — основной и наиболее удобный в новых проектах.
2. Alembic хранит текущую версию БД в служебной таблице alembic\_version, а история описывается ревизиями в каталоге versions.
3. В работе реализованы связи: один-ко-многим (User → Address) и многие-ко-многим (Order ↔ Product через таблицу order\_products).
4. Миграция БД — контролируемое изменение схемы (создание/изменение таблиц, полей, индексов) без потери данных; позволяет последовательно развивать структуру.
5. Отношения многие-ко-многим в SQLAlchemy реализуются через вспомогательную таблицу (association table), указанную в параметре secondary.
6. При конфликте версий Alembic либо выполняют merge-ревизию (объединение веток), либо вручную переносят изменения и создают новую ревизию; затем приводят БД к единому head.

## **Вывод**

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