

Лабораторная работа № 10

Тема: Python и БД. ORM

Цель: освоение концепции ORM.

Задача. Для своей схемы базы данных:

- инициализировать проект с использованием SQLAlchemy.
- определить модели для представления различных сущностей.
- произвести добавление, редактирование и удаление данных.
- выполнить запросы на выборку данных из базы данных.

Отчет должен содержать:

1. Текст задания.
2. Схема базы данных.
3. Листинг файла *.py.
4. Скриншоты с результатами выполнения запросов.


```

1  from sqlalchemy import create_engine, Column, Integer, String, ForeignKey, Text
2  from sqlalchemy.orm import relationship, declarative_base, sessionmaker
3
4  Base = declarative_base()
5
6  class Author(Base):
7      __tablename__ = 'authors'
8      id = Column(Integer, primary_key=True)
9      name = Column(String, nullable=False)
10     books = relationship('Book', back_populates='author')
11
12     class Book(Base):
13         __tablename__ = 'books'
14         id = Column(Integer, primary_key=True)
15         title = Column(String, nullable=False)
16         author_id = Column(Integer, ForeignKey('authors.id'))
17         author = relationship('Author', back_populates='books')
18         publisher_id = Column(Integer, ForeignKey('publishers.id'))
19         publisher = relationship('Publisher', back_populates='books')
20         reviews = relationship('Review', back_populates='book')
21
22     class Publisher(Base):
23         __tablename__ = 'publishers'
24         id = Column(Integer, primary_key=True)
25         name = Column(String, nullable=False)
26         books = relationship('Book', back_populates='publisher')
27
28     class Review(Base):
29         __tablename__ = 'reviews'
30         id = Column(Integer, primary_key=True)
31         content = Column(Text)
32         book_id = Column(Integer, ForeignKey('books.id'))
33         book = relationship('Book', back_populates='reviews')
34         user_id = Column(Integer, ForeignKey('users.id'))
35         user = relationship('User', back_populates='reviews')
36

```

```

36
37 class User(Base):
38     __tablename__ = 'users'
39     id = Column(Integer, primary_key=True)
40     name = Column(String, nullable=False)
41     reviews = relationship('Review', back_populates='user')
42
43     # Создаем engine
44     engine = create_engine('postgresql://user20:1@localhost:5432/db20')
45     Base.metadata.create_all(engine)
46
47     def initialize_data(session):
48         author1 = Author(name='Имя Автора 1')
49         author2 = Author(name='Имя Автора 2')
50
51         publisher1 = Publisher(name='Издательство 1')
52         publisher2 = Publisher(name='Издательство 2')
53
54         book1 = Book(title='Книга 1', author=author1, publisher=publisher1)
55         book2 = Book(title='Книга 2', author=author2, publisher=publisher2)
56
57         user1 = User(name='Пользователь 1')
58         review1 = Review(content='Отзыв на Книгу 1', book=book1, user=user1)
59
60         session.add_all([author1, author2, publisher1, publisher2, book1, book2, user1, review1])
61         session.commit()
62
63         # Добавим еще несколько записей для тестирования
64         user2 = User(name='Пользователь 2')
65         review2 = Review(content='Отзыв на Книгу 2', book=book2, user=user2)
66
67         session.add_all([user2, review2])
68         session.commit()
69
70     def retrieve_and_print_books(session):
71         selected_books = session.query(Book).all()
72         for book in selected_books:
73             print(f'Книга: {book.title}, Автор: {book.author.name}, Издательство: {book.publisher.name}')
74

```

```

74
75 ~ def delete_review_and_print_remaining_reviews(session):
76     selected_reviews = session.query(Review).all()
77     print(f'Количество отзывов до удаления: {len(selected_reviews)}')
78
79     if selected_reviews:
80         session.delete(selected_reviews[0])
81         session.commit()
82
83     selected_reviews = session.query(Review).all()
84     print(f'Количество отзывов после удаления: {len(selected_reviews)}')
85
86 ~ def select_and_update_data(session):
87     # Пример запроса на выборку: Получение всех авторов и их книг
88     authors_and_books = session.query(Author, Book).join(Book).all()
89     for author, book in authors_and_books:
90         print(f'Автор: {author.name}, Книга: {book.title}')
91
92     # Пример запроса на выборку с фильтром: Получение книг определенного издательства
93     selected_publisher_name = 'Издательство 1'
94     books_from_selected_publisher = session.query(Book).join(Publisher).filter(Publisher.name == selected_publisher_name).all()
95     for book in books_from_selected_publisher:
96         print(f'Книга от издательства {selected_publisher_name}: {book.title}')
97
98     # Пример запроса на редактирование данных: Изменение имени пользователя
99     user_to_update = session.query(User).filter(User.name == 'Пользователь 1').first()
100 ~ if user_to_update:
101     new_user_name = 'Обновленный Пользователь 1'
102     user_to_update.name = new_user_name
103     session.commit()
104     print(f'Имя пользователя обновлено: {new_user_name}')
105 else:
106     print('Пользователь не найден')
107
108 > if __name__ == "__main__":
109     # Создаем сессию
110     Session = sessionmaker(bind=engine)
111     session = Session()
112

```

```

112
113     initialize_data(session)
114     retrieve_and_print_books(session)
115     delete_review_and_print_remaining_reviews(session)
116
117     # Выполняем запросы на выборку и редактирование данных
118     select_and_update_data(session)
119

```

```
db20=> SELECT * FROM authors;
```

id	name
1	Имя Автора 1
2	Имя Автора 2
3	Имя Автора 1
4	Имя Автора 2
5	Имя Автора 1
6	Имя Автора 2

(6 строк)

```
db20=> SELECT * FROM books;
```

id	title	author_id	publisher_id
1	Книга 1	1	1
2	Книга 2	2	2
3	Книга 1	3	3
4	Книга 2	4	4
5	Книга 1	5	5
6	Книга 2	6	6

(6 строк)

```
db20=> SELECT * FROM publishers;
```

id	name
1	Издательство 1
2	Издательство 2
3	Издательство 1
4	Издательство 2
5	Издательство 1
6	Издательство 2

(6 строк)

```
db20=> SELECT * FROM reviews;
```

id	content	book_id	user_id
4	Отзыв на Книгу 1	5	4
5	Отзыв на Книгу 2	6	5

(2 строки)

```
db20=> SELECT * FROM users;
```

id	name
2	Пользователь 1
3	Пользователь 2
4	Пользователь 1
5	Пользователь 2
1	Обновленный Пользователь 1

(5 строк)

```
db20=>
```

Итог лабораторной работы:

Мы научились работать с БД и Python, совмещая эти две программы