**Лабораторная работа № 6**

РАЗРАБОТКА ПРИЛОЖЕНИЯ РАБОТЫ С БАЗОЙ ДАННЫХ

**Вариант 10.**

**Цель работы:** изучить возможности взаимодействия Python с реляционными базами данных с помощью DB-API 2.0.

**Краткая теория**

Почти каждая программа, которая работает с данными, требует места для их хранения. В качестве такого хранилища могут выступать обычные текстовые файлы, файлы JSON или XML. Однако для хранения большого количества данных и удобной организации работы с ними существуют СУБД. В данной лабораторной работе не рассматриваются понятия и термины баз данных, синтаксис SQL. Здесь будут рассмотрены механизмы для работы с базами данных с помощью Python.

DB-API 2.0

Несмотря на существующий стандарт SQL (ISO/IEC 9075) каждая СУБД имеет ряд отличий от других. Для того, чтобы программисту не вникать в их реализации в Python существует общее API, описанное стандартом PEP 249. Стоит учесть, что PEP 249 это всего лишь спецификация, реализация которой выполняется самостоятельно. Однако существуют уже готовые реализации данного API для ряда СУБД. Стоит отметить, что Python имеет в качестве встроенного модуля одну из реализаций данного API для работы с SQLite. Ниже приведены некоторые реализации для других СУБД:

− Oracle: cx\_Oracle, mxODBC, pyodbc;

− MySQL: mysql-python, PyMySQL;

− MS SQL Server: adodbapi, pymssql, mxODBC, pyodbc;

− PostgreSQL: psycopg2, txpostgres.

Для подключения реализации рассматриваемого API при работе с Sqlite необходимо импортировать пакет sqlite3 из стандартной библиотеки Python. Чтобы открыть соединение с базой необходимо использовать функцию connect(db\_file\_path), которая в качестве значения аргумента принимает путь к файлу SQLite. При работе с другими СУБД потребуется указать дополнительные параметры функции, такие как хост, имя пользователя, пароль и наименование базы.

Для выполнения запросов и получения результатов требуется использовать специальный объект Cursor, который можно получить, вызвав метод cursor() из объекта Connection, который возвращается из функции connect().

Вызов метода execute(query) для объекта Cursor выполняет запрос и в случае наличия результата возвращает ответ, который можно получить методом fetchall(). Стоит отметить, что данный метод возвращает значение только один раз, повторное обращение к нему без предварительного запроса вернет пустой список.

Для закрытия соединения с базой данных используется метод close() объекта Connection.

**Задание.** **Вариант 10.**

Написать функцию, которая строит ER модель базы данных. Можно сгенерировать схему в формате HTML или XML, либо воспользоваться сторонними приложениями и библиотеками.

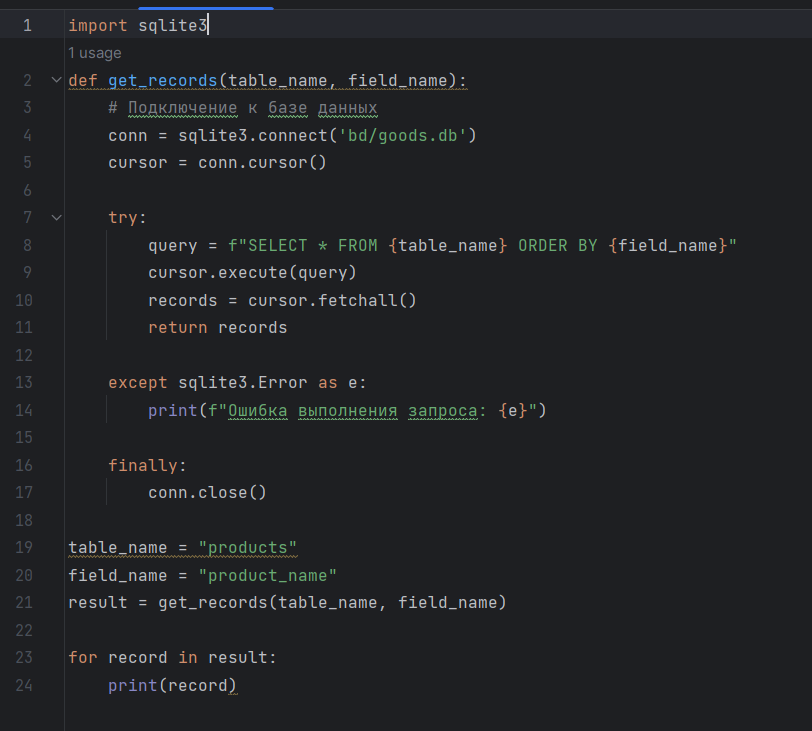


Рисунок 6.1 – код программы.

Результат работы программы представлен на рисунке 6.2

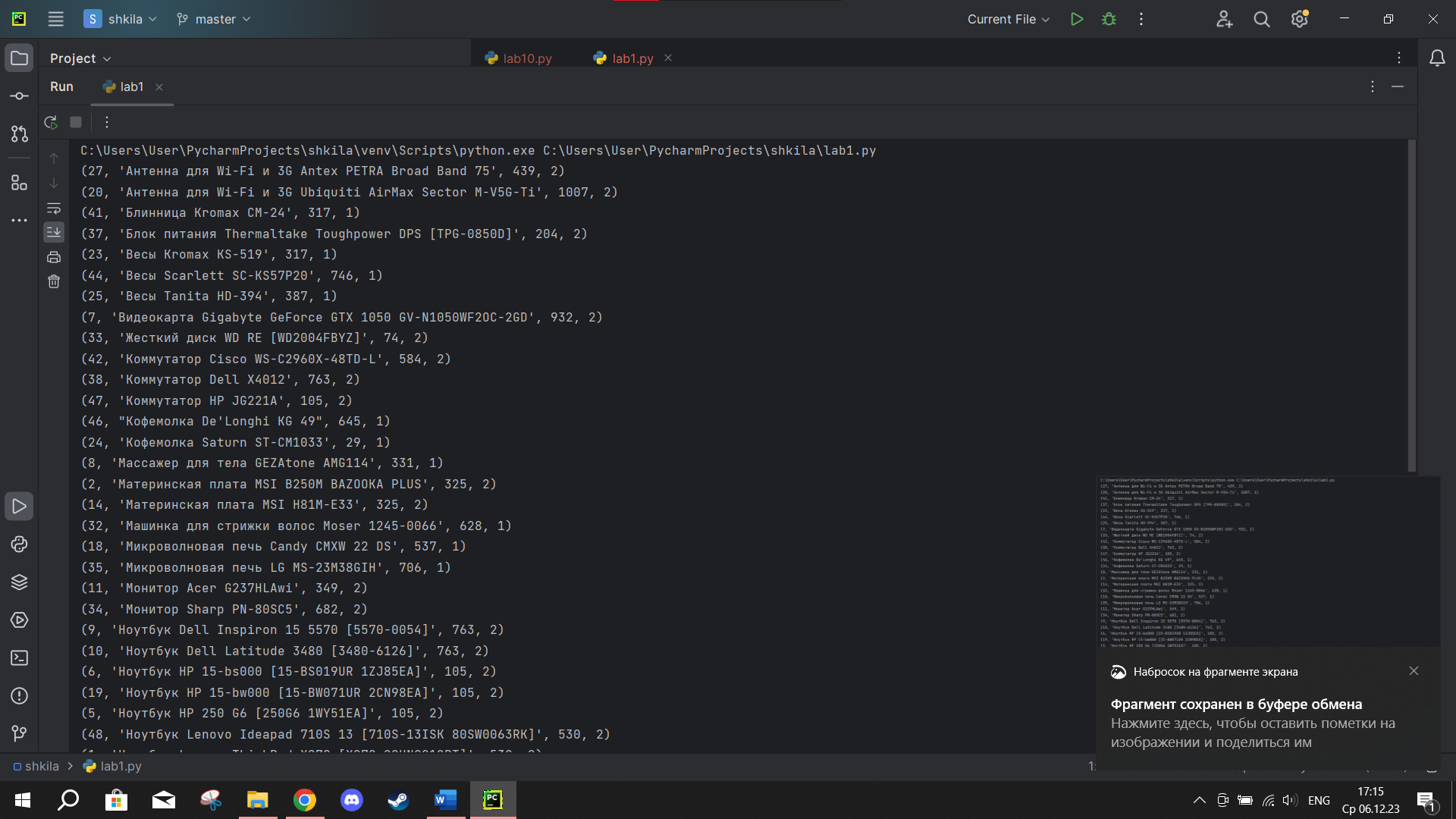


Рисунок 6.2 – результат работы программы

**Вывод**: в результате проделанной работы была изучена возможность взаимодействия Python с реляционными базами данных с помощью DB-API 2.0.

|  |  |
| --- | --- |
| Выполнил | Рогов М.А. |
| Проверил | Елкин Н.С. |