**Министерство науки и высшего образования Российской Федерации**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Факультет безопасности информационных технологий**

**Дисциплина:**

«Информационная безопасность баз данных»

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №6**

«Доступ к БД с уровня приложений. SQL-инъекции. Защита и фильтрация данных, получаемых от пользователя»

Содержание

[Введение 3](#_Toc195789950)

[1 Доступ к БД с уровня приложений. SQL-инъекции. Защита и фильтрация данных, получаемых от пользователя. 4](#_Toc195789951)

[1.1 Ход работы 4](#_Toc195789952)

[Заключение 12](#_Toc195789953)

[Список использованных источников 13](#_Toc195789954)

Введение

Цель работы – познакомиться с доступом к БД с уровня приложений, SQL-инъекциями, защитой и фильтрацией данных, получаемых от пользователя.

Для достижения поставленной цели необходимо решить следующие задачи:

* создать таблицы;
* написать и протестировать запросы на языке python для взаимодействия с бд;
* написать параметризованный запрос, протестировать его;
* сделать выводы.

# Доступ к БД с уровня приложений. SQL-инъекции. Защита и фильтрация данных, получаемых от пользователя.

## Ход работы

1. Для выполнения дальнейших заданий создадим 3 таблицы и заполним их данными.

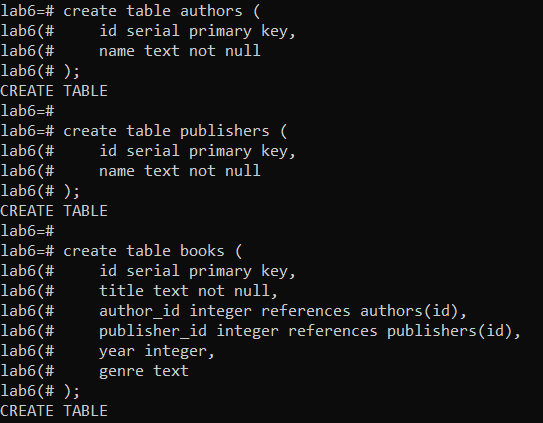


Рисунок 1 – создание таблиц

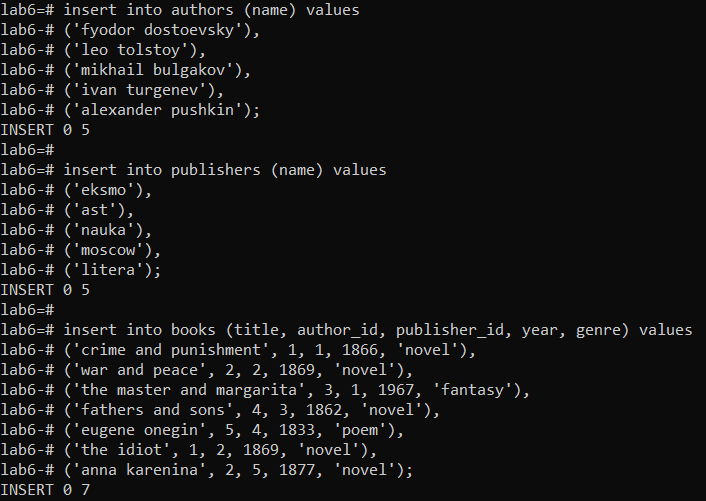


Рисунок 2 – заполнение таблиц

1. Необходимо продемонстрировать один из способов взаимодействия с БД с уровня приложения. Для этого воспользуемся языком программирования python и это адаптер PostgreSQL для python psycopg2.

Продемонстрируем работу с БД. Сначала реализуем просто выборку всех книг с жанром роман.

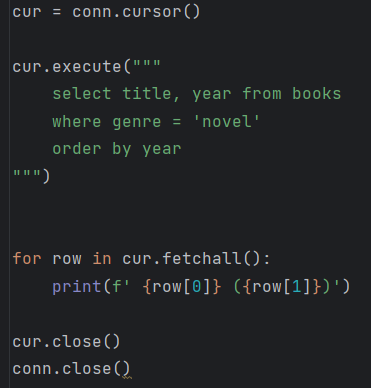


Рисунок 3 – код для выборки

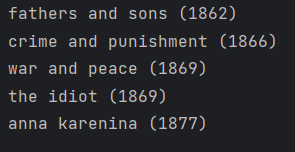


Рисунок 4 – выборка

Затем добавим новые элементы в таблицу. Выведем таблицу из нашей БД для проверки.

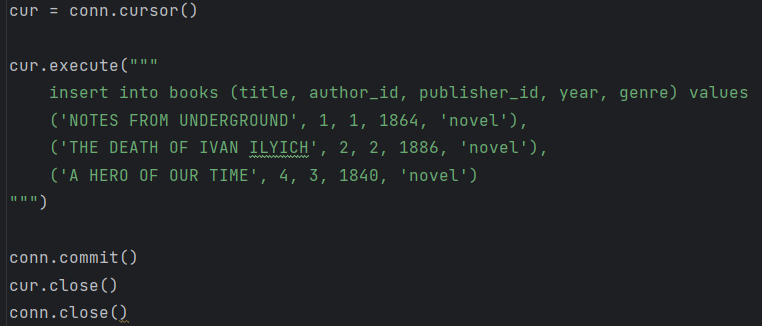


Рисунок 5 – код для вставки

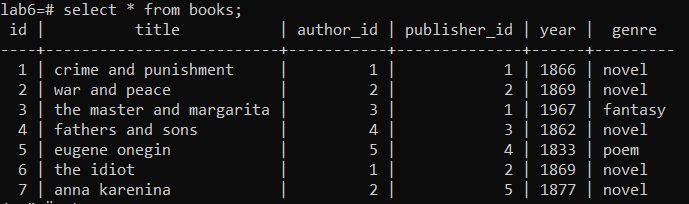


Рисунок 6 – таблица до вставки

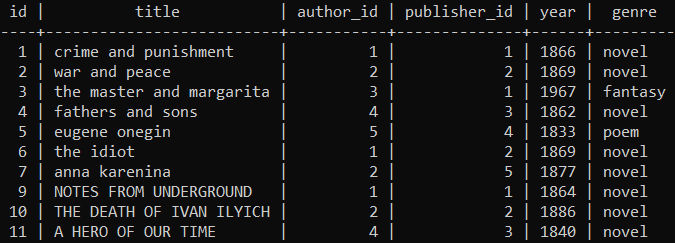


Рисунок 7 – таблица после вставки

Затем удалим данные из таблицы. Пусть будут удалены все книги, написанные Толстым. Выведем таблицу для проверки.



Рисунок 8 – код для удаления

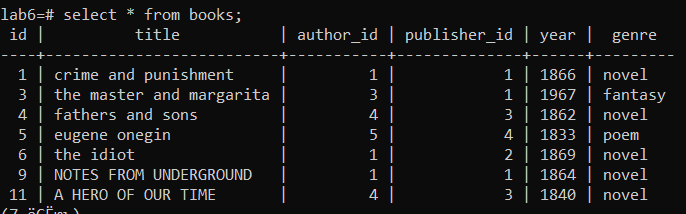


Рисунок 9 – таблица после удаления

Теперь реализуем два сложных запроса. Пусть первый изменит жанр с романа на классику у всех книг издательства АСТ. А второй выведет все книги Пушкина.

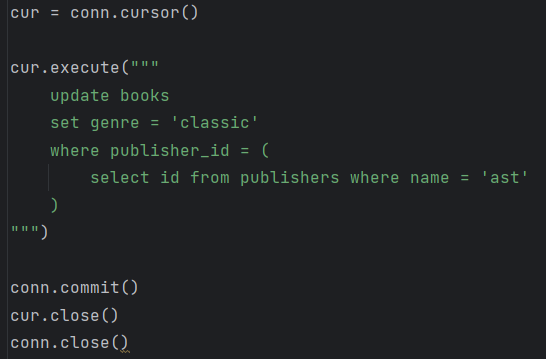


Рисунок 10 – первый запрос

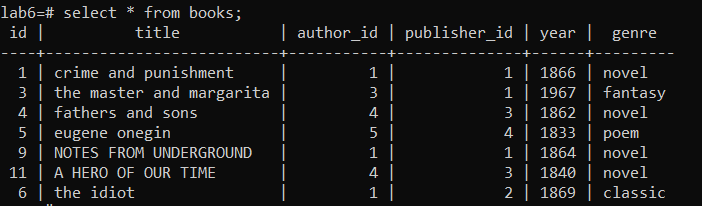


Рисунок 11 – таблица после первого запроса

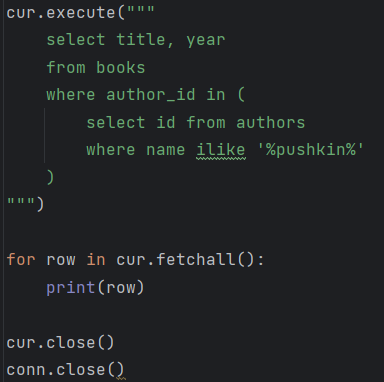


Рисунок 12 – второй запрос



Рисунок 13 – результат второго запроса

1. Для изучения проблемы фильтрации данных подготовим пример аналогичный, заданному в указаниях к данной лабораторной работе (Пример 1.). Один из популярных методов предотвращения SQL-инъекций – использование параметризованных запросов. Такой подход позволяет вместо конкатенации строк использовать специальную структуру с входными данными в качестве параметров. Все известные языки программирования могут использовать подготовленные параметры.

Пример будет подготовлен на python. Предусмотрим в примере два случая подготовки SQL запросов (подготовленные запросы, конкатенация параметров со строкой запроса). Таким образом, мы покажем разницу между безопасным (подготовленные запросы) и небезопасным способом (конкатенация) работы с SQL-запросами. В первом случае параметры (логин и пароль) вставляются напрямую в строку запроса, что делает код уязвимым к SQL-инъекциям. Во втором случае используется подготовленный запрос, где значения передаются отдельно, поэтому попытки инъекции не срабатывают.

Создадим также новую таблицу с пользователями и их паролями.

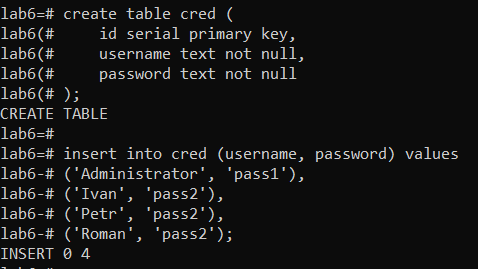


Рисунок 14 – создание таблицы с пользователями

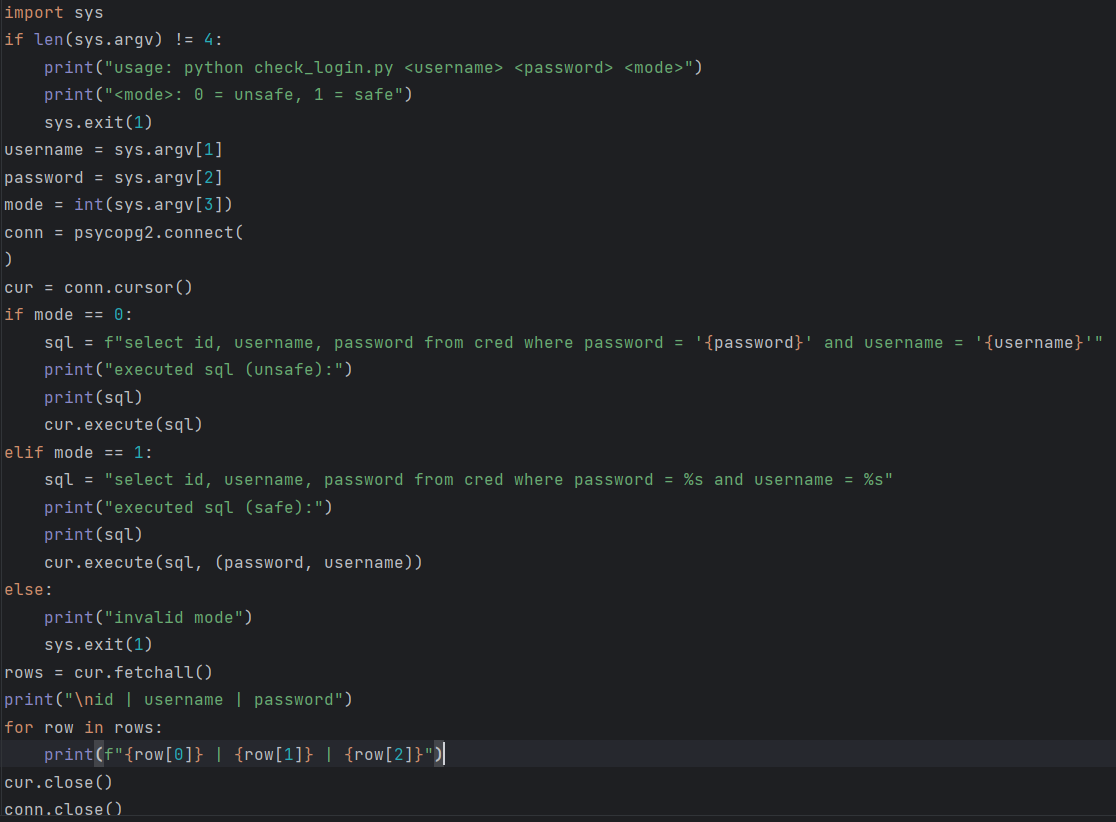


Рисунок 15 – код параметризованного запроса

1. Продемонстрируем возможные варианты проведения SQL-инъекций.

Запустим код сначала используя конкатенация (параметры вставляются в строку запроса напрямую) (Administrator "' or '1'='1 --" 1). То есть происходи поиск пользователя по логину и паролю, и в этом случае инъекция будет выполнена, пользователь будет найден.

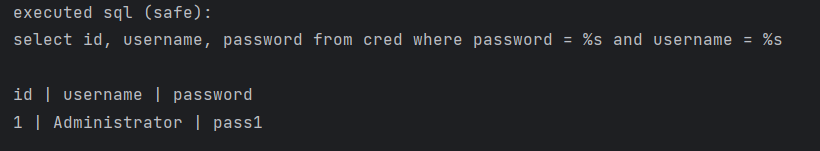


Рисунок 16 – использование конкатенацию строк

Затем используем подготовленный оператор с обработкой символов-пропусков. В этом случае пользователь уже не будет найден.

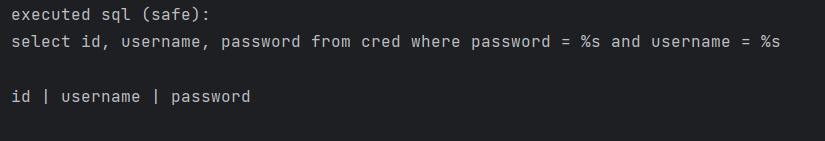


Рисунок 17 – использование подготовительного оператора

Можно также продемонстрировать и другой вариант SQL инъекции с использованием UNION. Для этого создадим таблицу с пользователями и купленными ими книгами.

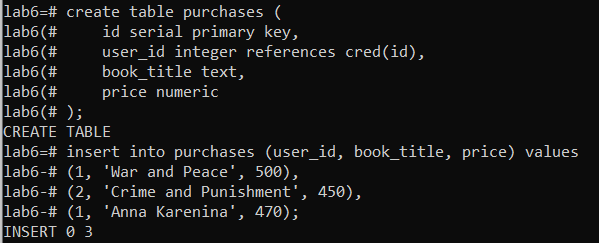


Рисунок 18 – создание новой таблицы

Заключение

В ходе данной лабораторной работы были изучены доступ к БД с уровня приложений, SQL-инъекции, защита и фильтрация данных, получаемых от пользователя. Были написаны на языке python запросы для взаимодействия с БД на уровне приложения, написан параметризованный запрос. Были проведены тесты. Это позволило закрепить полученные знания и навыки. В ходе выполнения лабораторной работы были выполнены все задачи и достигнуты поставленные цели.

Список использованных источников

1. Основы технологий баз данных: учебное пособие Б. А. Новиков, Е. А. Горшкова, Н. Г. Графеева; под ред. Е. В. Рогова. — 2-е изд. — М.: ДМК Пресс, 2020. — 582 с.