1. Министерство образования и науки Российской Федерации
2. Санкт-Петербургский Политехнический Университет Петра Великого
3. —
4. Институт кибербезопасности и защиты информации

**ЛАБОРАТОРНАЯ РАБОТА № 2  
   
«SQL-запросы и SQL-инъекции**»

1. по дисциплине «Системы управления базами данных»
2. Выполнил
3. студент гр. 4851004/80101 Терещенко Е. А.

<*подпись*>

1. Проверил
2. Ассистент преподавателя Полтавцева М. А.

<*подпись*>

1. Санкт-Петербург
2. 2021

Цель работы

Получение навыков составления запросов и защиты от SQL – инъекций в СУБД..

Задание

Варианты выполнения лабораторной работы представлены ниже.

1. Реализовать представленные запросы. При этом:

1. Заданные данные оформить в виде входной переменной любого типа
2. Каждый пункт «выбрать» необходимо выполнить в виде одного SQL – запроса без промежуточных таблиц (структура with).
3. Обратите внимание, что формулировки и уточнения к запросам направлены на то, чтобы вы правильно поняли, что от вас требуется, но НЕ подсказывают способ решения.

2. Изучить представленный код (SQL - файл из папки «ЛР2 варианты»). Обратите внимание, что вариантов этого задания БОЛЬШЕ, чем схем в таблице 1. Можно модифицировать под свою СУБД.

3. Составить и реализовать цепочку SQL – инъекций к доступным процедурам (процедуре) для получения пароля администратора, заданного в коде и хранящегося в СУБД.

4. Составить набор методов защиты от инъекций для данного примера. Продемонстрировать его эффективность.

5. Предложите по крайней мере один пример инъекции, которая осталась возможна.

Вариант задания

В рамках работы был выполнен 7 вариант задания.

Реализованы отношения:

* **Отношение 1**

Код проекта (PK), Название проекта (AK1), Наименование задачи (PK, AK1), ФИО исполнителя (FK), Трудоемкость в часах, Плановая дата выполнения, Реальная дата выполнения если есть, Описание задачи, Отметка о принятии задачи руководителем.

* **Отношение 2**

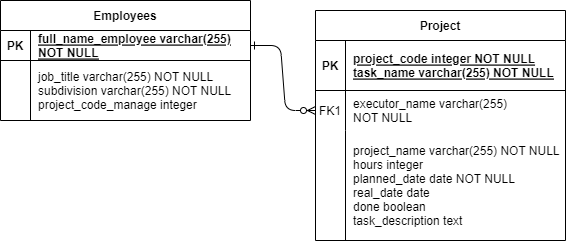
ФИО сотрудника (PK), Должность, Подразделение, Код проекта которым руководит сотрудник, если он есть.

**Описание запросов**:

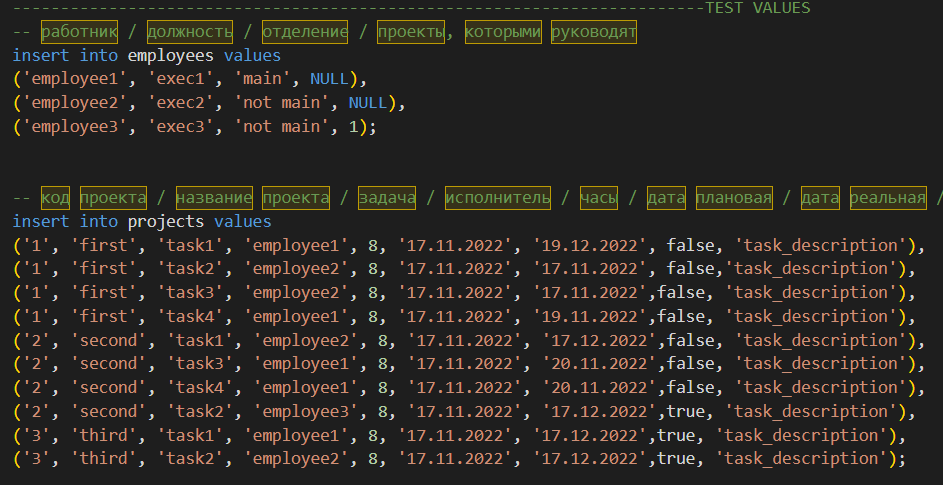
* Выберите все коды и названия проектов, в которых больше трех задач, еще не принятых руководителем.
* Выберите все наименования задач и ФИО их исполнителей, у которых реальная дата выполнения не лежит в заданном интервале.
* Выберите ФИО сотрудников, поработавших на всех проектах и ФИО сотрудников, выполнивших все задачи хотя бы одного своего проекта.

Часть 1. Реализованные запросы

Схема данных совпадает со схемой из первой лабораторной работы (рисунок 1.1).

  
Рисунок 1.1. – Логическая схема

Таблицы были заполнены тестовыми данными (рисунок 1.2).

  
Рисунок 1.2 – Тестовые данные

* 1. Запрос 1

Формулировка запроса: выберите все коды и названия проектов, в которых больше трех задач, еще не принятых руководителем.

Текст запроса и результат выполнения представлены на рисунке 1.1.1. Согласно тестовым данным (рисунок 1.2), под данный запрос должен попадать только один проект.

Изображение выглядит как текст

Автоматически созданное описание  
Рисунок 1.1.1 – Текст запроса и результат его выполнения

* 1. Запрос 2

Формулировка запроса: выберите все наименования задач и ФИО их исполнителей, у которых реальная дата выполнения не лежит в заданном интервале.

Текст запроса и результат выполнения представлены на рисунках 1.2.1, 1.2.2. Был выбран промежуток дат между “16.11.2022” и “18.11.2022”. Согласно тестовым данным (рисунок 1.2), под данный запрос попадает проекты с реальной датой выполнения “19.11.2022”, “20.11.2022” и “17.12.2022” (всего 8 записей).

Изображение выглядит как текст

Автоматически созданное описание  
Рисунок 1.2.1 –Текст запроса

Изображение выглядит как текст

Автоматически созданное описание  
Рисунок 1.2.2 –Результат выполнения запроса

* 1. Запрос 3

Формулировка запроса: выберите ФИО сотрудников, поработавших на всех проектах и ФИО сотрудников, выполнивших все задачи хотя бы одного своего проекта.

Обе части запроса можно переформулировать через отрицание следующим образом:

1. Выберите ФИО сотрудников, для которых не существует проектов, на которых они бы не проработали;
2. Выберите ФИО сотрудников, для которых в рамках хотя бы одного проекта не существует ни одной задачи, которую они бы не выполнили.

Текст запроса и результат выполнения представлены на рисунке 1.3.1.

Изображение выглядит как текст

Автоматически созданное описание  
Рисунок 1.3.1 – Текст запроса и результат его выполнения

Часть 2. SQL-инъекции

2.1. Получение пароля администратора

Код с вариантом задания был изучен и адаптирован под PostgreSQL.

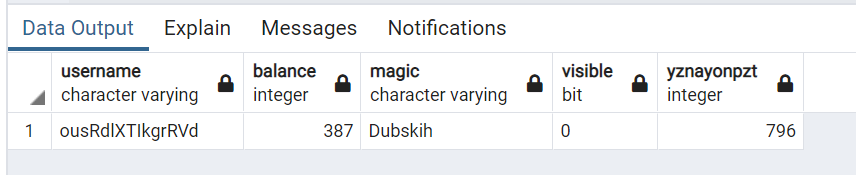
Для получения пароля администратора был составлен запрос с sql-инъекцией, представленной на рисунке 2.1.1.

Изображение выглядит как текст

Автоматически созданное описание  
Рисунок 2.1.1 – Запрос с инъекцией

Результат выполнения запроса представлен на рисунке 2.1.2. В столбце username мы получили имя пользователя, в столбце magic – пароль этого пользователя.

Столбец magic – единственные столбец, через который можно было вывести пароль, т.к. возвращаемые типы столбцов функции должны совпадать, а пароль хранится в varchar.

  
Рисунок 2.1.2 – Результат выполнения запроса с инъекцией

2.2. Методы защиты

В файле с вариантом задания представлены различные «фильтры» для защиты от sql-инъекций, однако проблема в том, что в функции GetUserInfo() нет строчки, вызывающей их. Добавим данную строку (рисунок 2.2.1) и попробуем выполнить инъекцию снова (рисунок 2.2.2).

Изображение выглядит как текст

Автоматически созданное описание  
Рисунок 2.2.1 – Добавление защиты от инъекций

Изображение выглядит как текст

Автоматически созданное описание  
Рисунок 2.2.2 – Попытка выполнения запроса с инъекцией

2.3. Обход реализованных методов защиты

В функции FilterFunc вызывается последовательно несколько фильтров в следующем порядке:

* Замена комментариев '/\*\*/' на '';
* Замена слов 'select' и 'from' на '';
* Удаление лишних пробелов;
* Замена слова admin на пустую строку;
* Замена знака равенства '=' на '';
* Удаление лишних кавычек;
* Замена слова NULL на пустую строку;

Чтобы обойти фильтры, нужно придумать альтернативное написание «запрещенных» слов, то есть слов, которые фильтры отбрасывают. Попробуем следующие варианты:

* Фильтр удаляет только одно «запрещенное» слово, поэтому нужно внутрь слова добавить это же слово (рисунок 2.3.1).

Изображение выглядит как текст

Автоматически созданное описание  
Рисунок 2.3.1 – Попытка 1 обойти фильтр

* Написать вместо символа код символа

Изображение выглядит как текст

Автоматически созданное описание  
Рисунок 2.3.2 – Попытка 2 обойти фильтр

* Заменить слово “admin” байтовой последовательностью

Изображение выглядит как текст

Автоматически созданное описание  
Рисунок 2.3.3 – Попытка 3 обойти фильтр

С учетом всех представленных выше обходов получим следующую инъекцию (рисунок 2.3.4).

Изображение выглядит как текст

Автоматически созданное описание  
Рисунок 2.3.4 – Инъекция

Вывод

В ходе лабораторной работы были изучены принципы применения и защиты от SQL-инъекций. Для тестовой базы данных была SQL инъекция, позволяющая получить имя пользователя и пароль администратора. Была реализована защита от инъекции и попытка обойти эту защиту.