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

**ЛАБОРАТОРНАЯ РАБОТА № 3  
   
«Ограничения целостности. Триггеры, представления**»

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 обработки событий и гранулирования доступа при работе с СУБД.

Задание

Варианты выполнения лабораторной работы, представлены ниже. Для выполнения работы используется схема данных из лабораторной работы 1.

1. Реализовать заданное ограничение целостности.

2. Реализовать представления в соответствии с требованиями по разграничению прав из работы 1. Выполнить все условия разграничения.

3. Дать соответствующим пользователям права только в отношении созданных представлений.

4. Для прав на модификацию данных обеспечить обновляемость представлений.

5. Сделать выводы в отношении применения стандартных способов разграничения прав доступа и построения разграничения на основе представлений.

6. Обеспечить аудит действий пользователя по манипулированию данными с использованием вспомогательной таблицы для логгирования в отношении таблицы по выбору студента.

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.

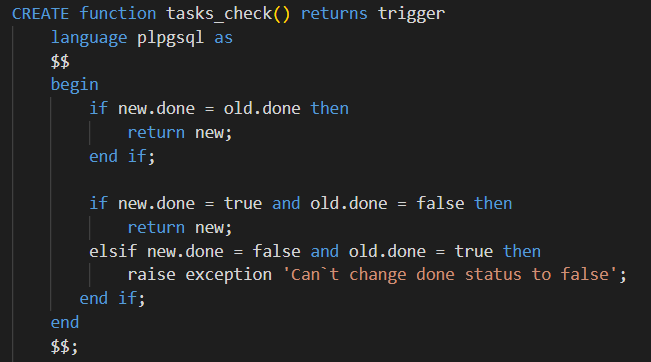
**Ограничение целостности**:

Отметка о принятии задачи руководителем может быть проставлена «в одном направлении» (0 на 1 и т.д.) и не может быть изменена обратно.

Часть 1. Ограничение целостности

Согласно варианту задания, необходимо сделать так, чтобы отметку о принятии решения можно было менять только в одну сторону (только с false на true). Для этого был создан триггер.

В PL/pgSQL в качестве триггера создается триггерная процедура с помощью команды CREATE FUNCTION, при этом у функции не должно быть аргументов, а в качестве возвращаемого значения должен быть trigger (рисунок 1.1).

  
Рисунок 1.1 – Триггерная функция

Далее с помощью команды CREATE TRIGGER необходимо указать, в каком случае будет срабатывать триггер (рисунок 1.2).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 1.2 – Условия активации триггера

Проверим, как работает триггер. На рисунке 1.3 видно отметки о принятии.

Изображение выглядит как стол

Автоматически созданное описание  
Рисунок 1.3 – Значения таблицы Projects

Попробуем изменить значения, представленные в строке 13 (риc. 1.4).

Изображение выглядит как текст

Автоматически созданное описание  
Рисунок 1.4 – Демонстрация работы триггера

Часть 2. Разграничение прав доступа посредством представлений

2.1. Матрица доступа

Разработанные матрицы доступа представлены в таблицах 2.1, 2.2. К таблице Employees и сотрудники-исполнители, и сотрудники-руководители имеют одинаковы доступ.

Таблица 2.1. Матрица доступа к таблице Projects

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Код проекта | Название проекта | Наименование задачи | ФИО исполнителя | Трудоемкость в часах | Плановая дата выполнения | Реальная дата выполнения | Описание задачи | Отметка о принятии задачи руководителем |
| Исполнитель | CREATE | - | - | - | - | - | - | - | - | - |
| READ | - | + | + | + | + | + | + | + | - |
| UPDATE | - | - | - | - | - | - | + | - | - |
| DELETE | - | - | - | - | - | - | - | - | - |
| Руководитель | CREATE | - | - | - | - | - | - | - | - | - |
| READ | + | + | + | + | + | + | + | + | + |
| UPDATE | - | - | - | - | - | - | - | - | + |
| DELETE | - | - | - | - | - | - | - | - | - |

Таблица 2.2. Матрица доступа к таблице Employees

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ФИО сотрудника | Должность | Подразделение | Код проекта |
| CREATE | - | - | - | - |
| READ | + | + | + | + |
| UPDATE | - | - | - | - |
| DELETE | - | - | - | - |

2.2. Представление отношения Employees

Так как данные из отношения Employees могут видеть и сотрудники-исполнители, и сотрудники-руководители, представление не накладывает никаких ограничений на выбор данных (рисунок 2.2.1).

Изображение выглядит как текст, подключен, снимок экрана

Автоматически созданное описание  
Рисунок 2.2.1 – Создание представления

2.3. Представление отношения Project

Реализация представления представлена на рисунке 2.3.1.

Поля project\_code и done выводятся пользователю только если у работника есть отметка с кодом проекта, которым он руководит.

Изображение выглядит как текст

Автоматически созданное описание  
Рисунок 2.3.1 – Создание представления

Для созданного представления определим права доступа (рисунок 2.3.2).

Изображение выглядит как текст

Автоматически созданное описание  
Рисунок 2.3.2 – Права доступа

Результат запроса к представлению представлен на рисунке 2.3.3. Несмотря на то, что пользователь запросил информацию о значении поля done, значение поля не вывелось.

Изображение выглядит как текст

Автоматически созданное описание  
Рисунок 2.3.3 – Результат запроса сотрудника-исполнителя

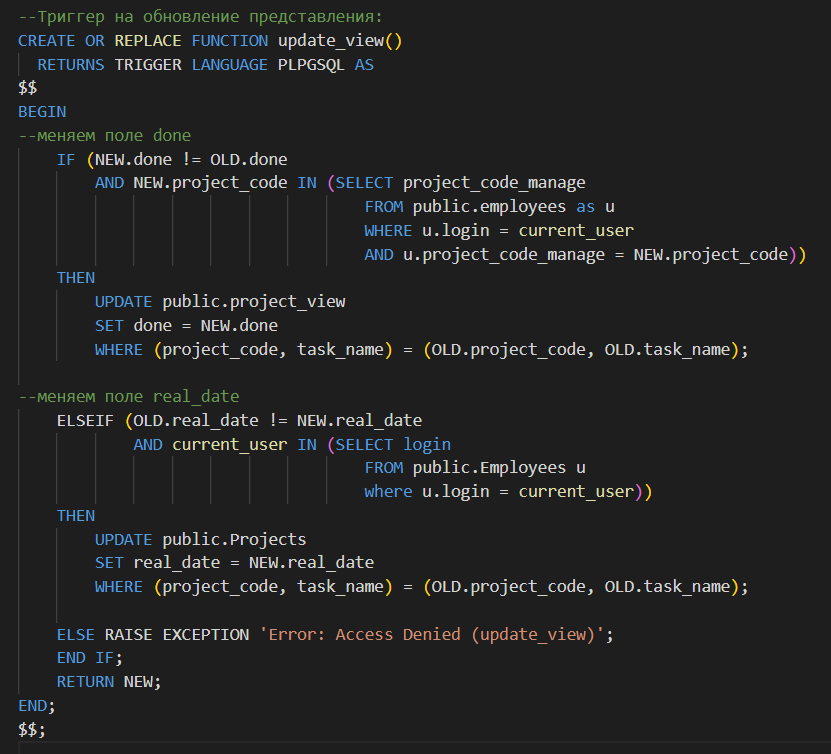
2.4. Обновление отношений

Чтобы информацию в отношениях можно было обновлять, необходимо создать триггеры для обновления отношений, одной команды GRANT UPDATE в данном случае недостаточно.

Необходимо определить триггер на обновление исходного отношения (рисунок 2.4.1) и триггер на обновление представления (рисунок 2.4.2), а также зарегистрировать их (рисунок 2.4.3).

Изображение выглядит как текст

Автоматически созданное описание  
Рисунок 2.4.1 – Триггер на обновление исходной таблицы

  
Рисунок 2.4.2 – Триггер на обновление представления

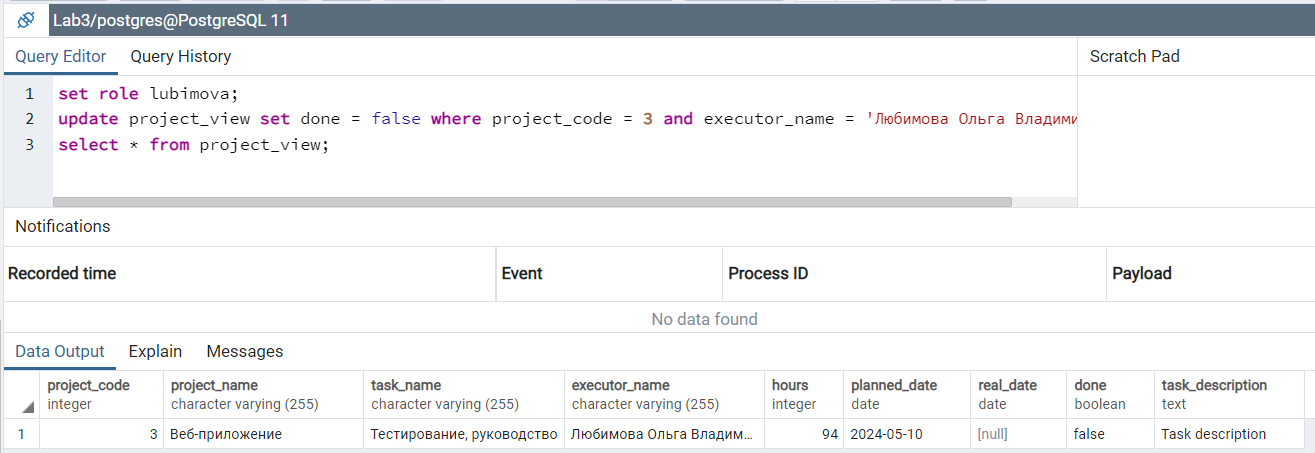
Изображение выглядит как текст

Автоматически созданное описание  
Рисунок 2.4.3 – Регистрация триггеров

Теперь, если запись в отношении попробует изменить не авторизированный на то пользователь, то появится сообщение ошибки (рис. 2.4.4). Однако, если пользователь имеет право выполнить данную операцию, то она выполнится (рис. 2.4.5).

Изображение выглядит как текст

Автоматически созданное описание  
Рисунок 2.4.4 – Изменение значения поля пользователем без прав

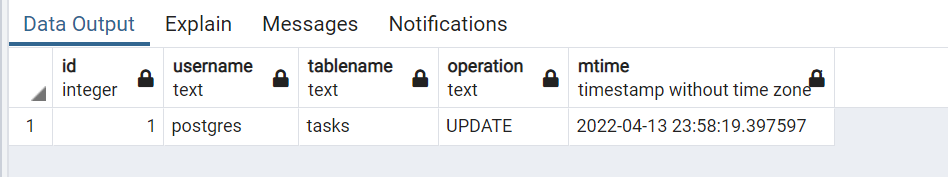
  
Рисунок 2.4.5 – Изменение значения авторизованным пользователем

Часть 3. Логгирование

Для логгирования было выбрано отношение 1, создана отдельная таблица и написана следующая триггерная функция (рис. 3.1).

Изображение выглядит как текст

Автоматически созданное описание  
Рисунок 3.1 – Триггер для логгирования

****  
Рисунок 3.2 – Формат логгируемых данных

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.