

МИНОБРНАУКИ РОССИИ

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«МИРЭА ­- Российский технологический университет»**

**РТУ МИРЭА**

Институт комплексной безопасности и специального приборостроения

Кафедра КБ-2 «Прикладные информационные технологии»

**ОТЧЕТ по дисциплине**

**«Клиент-серверные базы данных»**

**По курсовой работе**

Вариант 67

Студент: Старков А. С.

Шифр учебной группы: БСБО-02-18

Руководитель: Кульков Д.А.

Москва 2021г.

# **Задание на курсовую работу**

Тема: «Разработка клиент-серверного приложения»

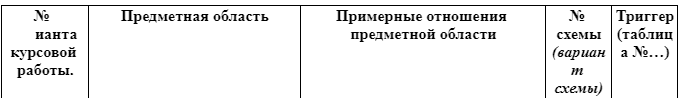
Цель работы: Закрепление теоретического материала по предмету «клиент-серверные базы данных», а также по ранее изученным предметам.

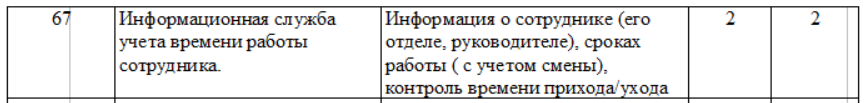
Задание:  Разработать клиент-серверное приложение, серверная часть которой реализована на любом SQL сервере, представляющая собой модель предметной области в соответствии с вариантом задания. В рамках заданной предметной области реализовать заданную (по варианту) схему отношений, т.е. выделить сущности и их атрибуты, так чтобы связи между сущностями соответствовали представленной схеме. В рамках курсовой работы необходимо на стороне сервера реализовать и использовать при демонстрации приложения следующие компоненты:

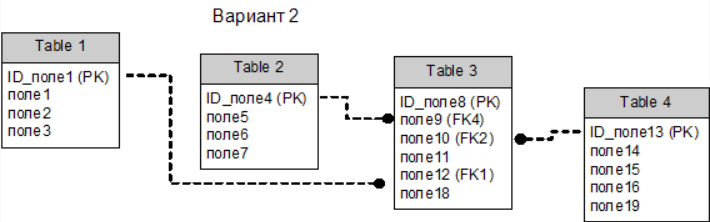
1. Постоянные таблицы и связи между ними, количество таблиц и наличие связей должно соответствовать заданию, допускается увеличение числа таблиц и их полей для более адекватного представления предметной области;
2. В приложении реализовать не менее пяти запросов, включая (для демонстрации навыков работы)
   1. Составной многотабличный запрос с параметром, включающий соединение таблиц и CASE-выражение;
   2. На основе обновляющего представления (многотабличного VIEW), в котором критерий упорядоченности задает пользователь при выполнении;
   3. Запрос, содержащий коррелированные и некоррелированные подзапросы в разделах SELECT, FROM и WHERE (в каждом хотя бы по одному);
   4. Многотабличный запрос, содержащий группировку записей, агрегативные функции и параметр, используемый в разделе HAVING;
   5. Запрос, содержащий предикат ANY(SOME) или ALL;
3. Создать индексы для увеличения скорости выполнения запросов;
4. В таблице (в соответствии с вариантом) предусмотреть поле, которое заполняется автоматически по срабатыванию триггера при добавлении, обновлении и удалении данных, иметь возможность продемонстрировать работу триггера при работе приложения. Триггеры должны обрабатывать только те записи, которые были добавлены, изменены или удалены в ходе текущей операции (транзакции).
5. Операции добавления, удаления и обновления реализовать в виде хранимых процедур (функций) с параметрами для всех таблиц;
6. Реализовать отдельную хранимую процедуру(функцию), состоящую из нескольких отдельных операций в виде единой транзакции, которая при определенных условиях может быть зафиксирована или откатана;
7. В триггере или хранимой процедуре реализовать курсор на обновления отдельных данных;
8. В запросе (из пункта 2 или в дополнительном к тому перечню) использовать собственную скалярную функцию, а в хранимой процедуре – векторную (или табличную) функцию. Функции сохранить в базе данных
9. Распределение прав пользователей: предусмотреть не менее двух пользователей с разным набором привилегий. Каждый набор привилегий оформить в виде роли.

Клиент должен обеспечивать добавление, модификацию и удаление данных по всей предметной области. Добавление, редактирование данных в таблице производить в отдельном окне.

Запрещено в качестве вводимых данных, в том числе для связи таблиц, указывать значения первичных и внешних ключей – для обеспечения ссылочной целостности пользователь должен выбирать значения из справочника, а соответствующие значения должны подставляться программно (тем или иным способом – автоматически).





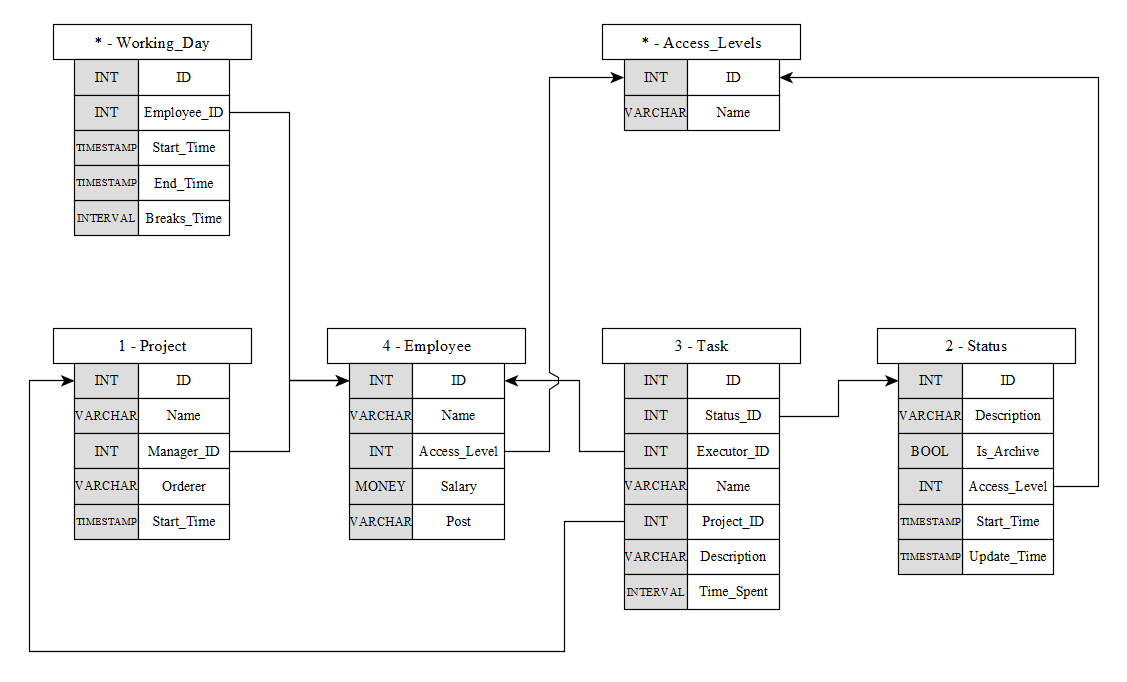


# **Ход работы**

# Введение

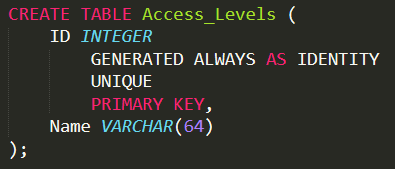
Для выполнения данной курсовой работы мной была выбрана СУБД PostgreSQL и язык C# с его пакетом WPF для удобной работы с данной БД. Подключение от клиента к БД осуществлялось посредством Npgsql. Первоначальная загрузка БД осуществлялась посредством pgAdmin.

ER-диаграмма:

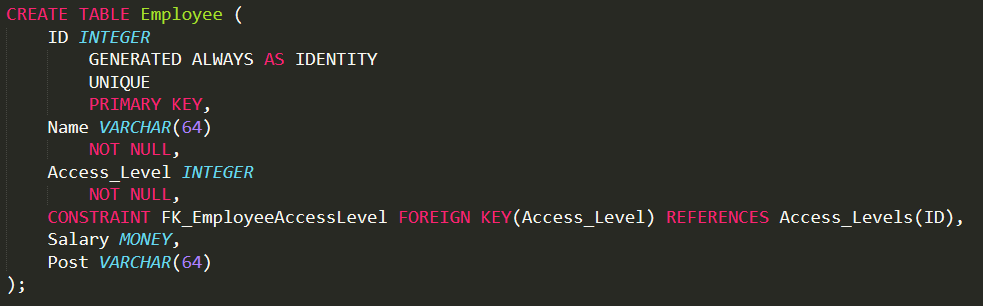


# Создание таблиц

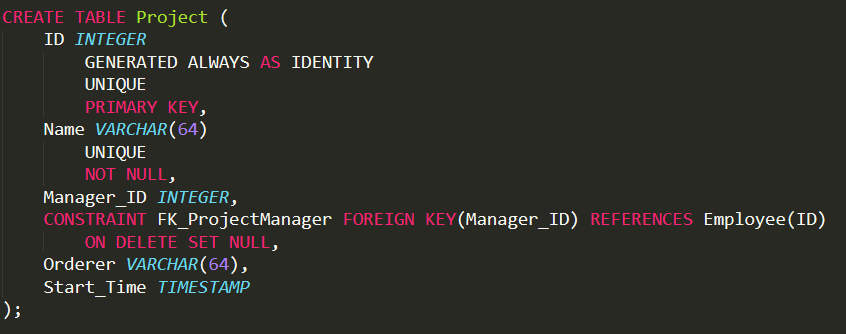
В таблице Access\_Levels хранится кодификатор уровней доступа с ID от 1 до 3 (Рабочий, менеджер и директор)



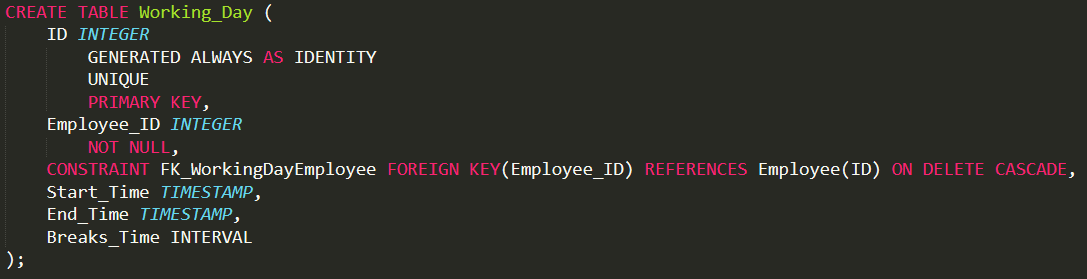
Employee хранит информацию о работниках, их имя, уровень доступа, зарплату и должность



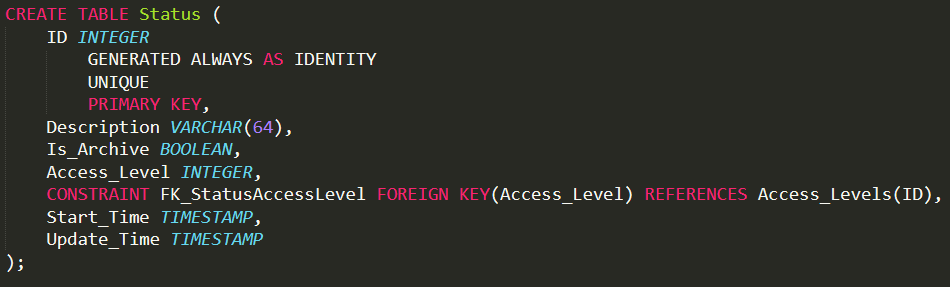
Project хранит информацию о проектах, включая заказчика, время начала, название и ID менеджера



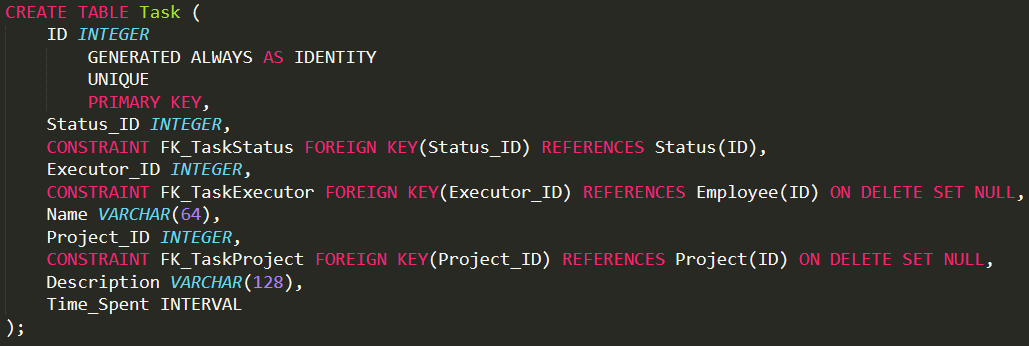
Working\_Day содержит начало и конец конкретного рабочего дня конкретного работника, а так же время, затраченное работником на перерывы в течение рабочего дня



Status - статусы рабочих заданий, объявленных далее. Здесь хранятся описания статуса, архивировано ли задание, уровень доступа необходимый для дальнейших действий, время начала и последнего действия

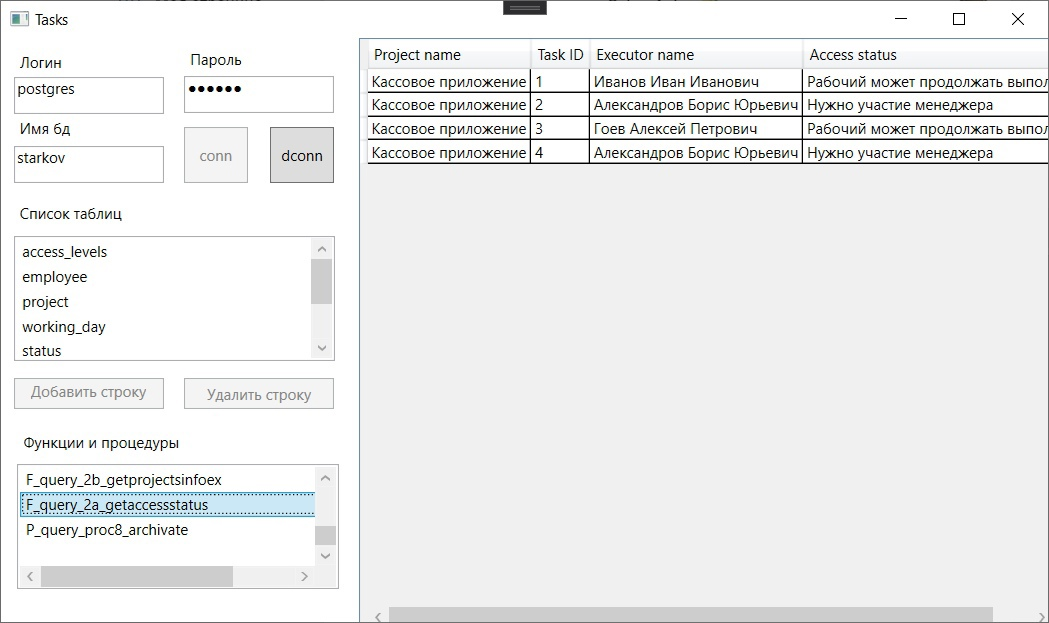
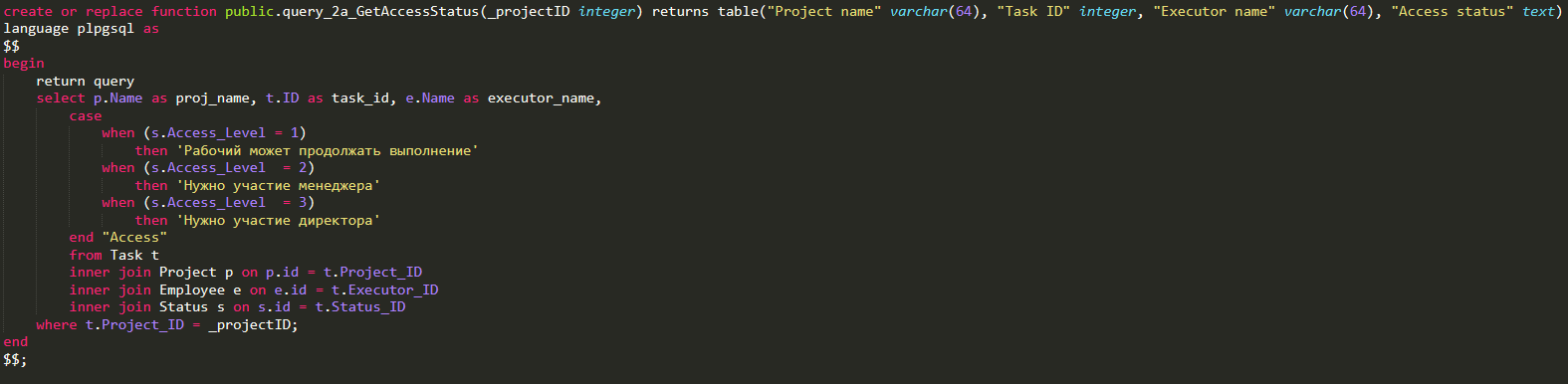


Task содержит непосредственно сами задания со ссылкой на их статус, исполнителя, проект, их название, описание и потраченное время

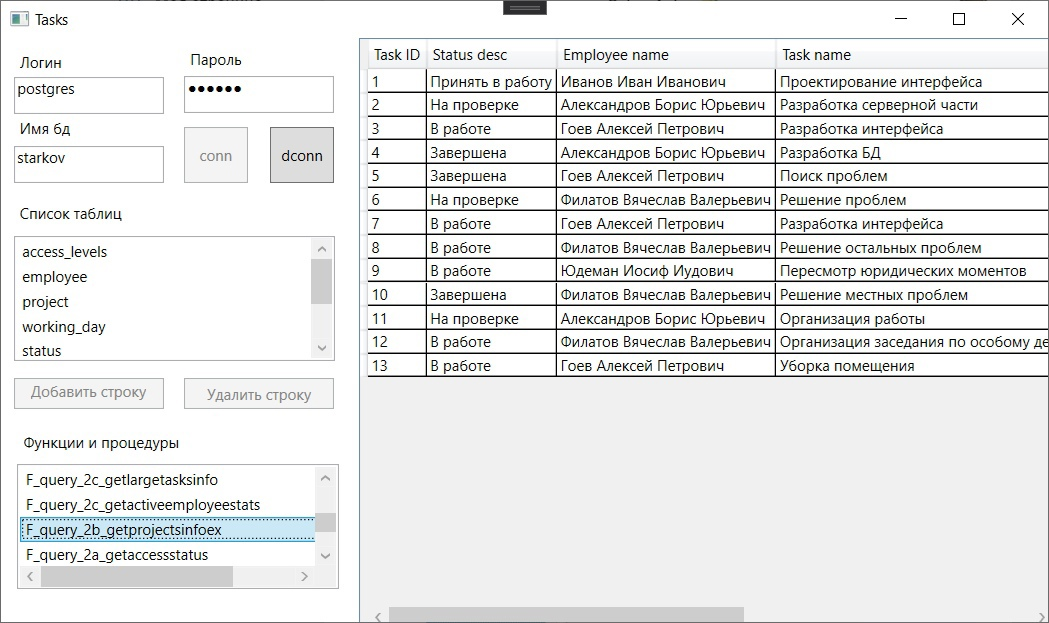
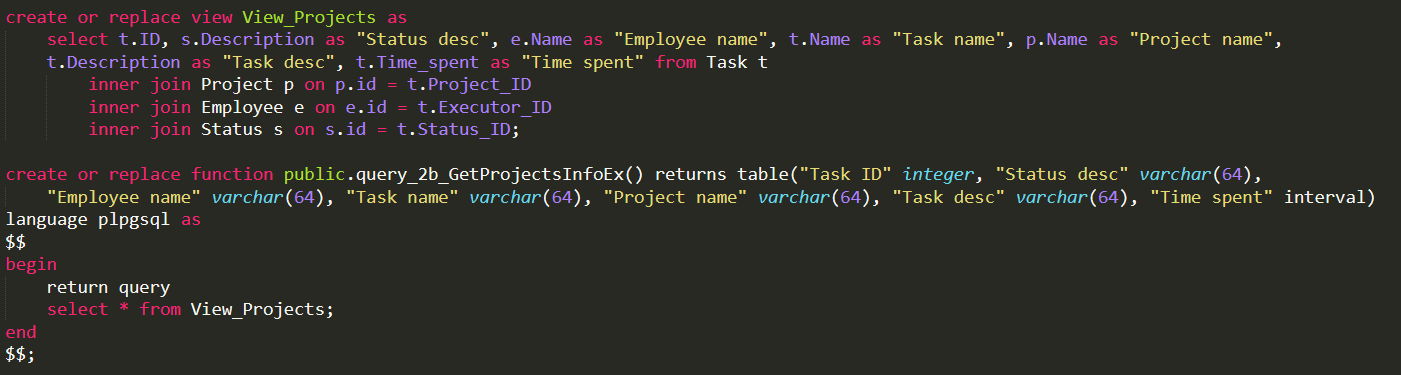


# Создание запросов

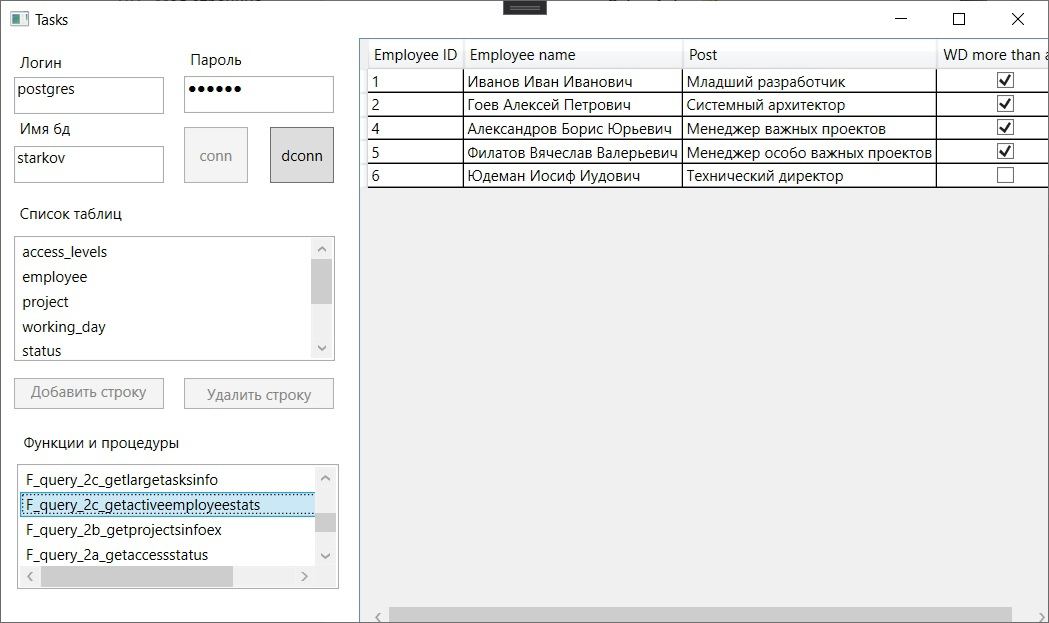
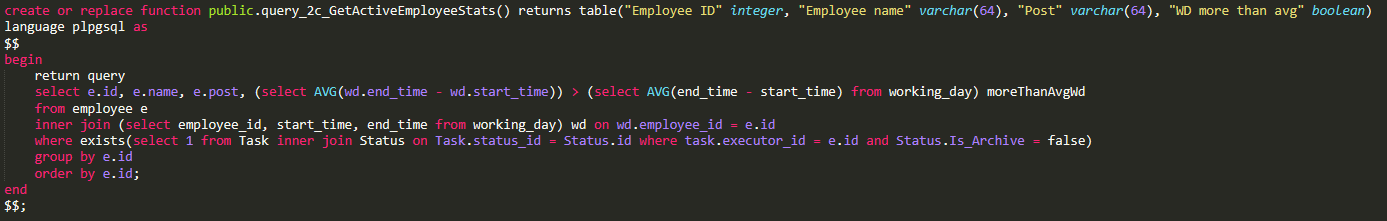
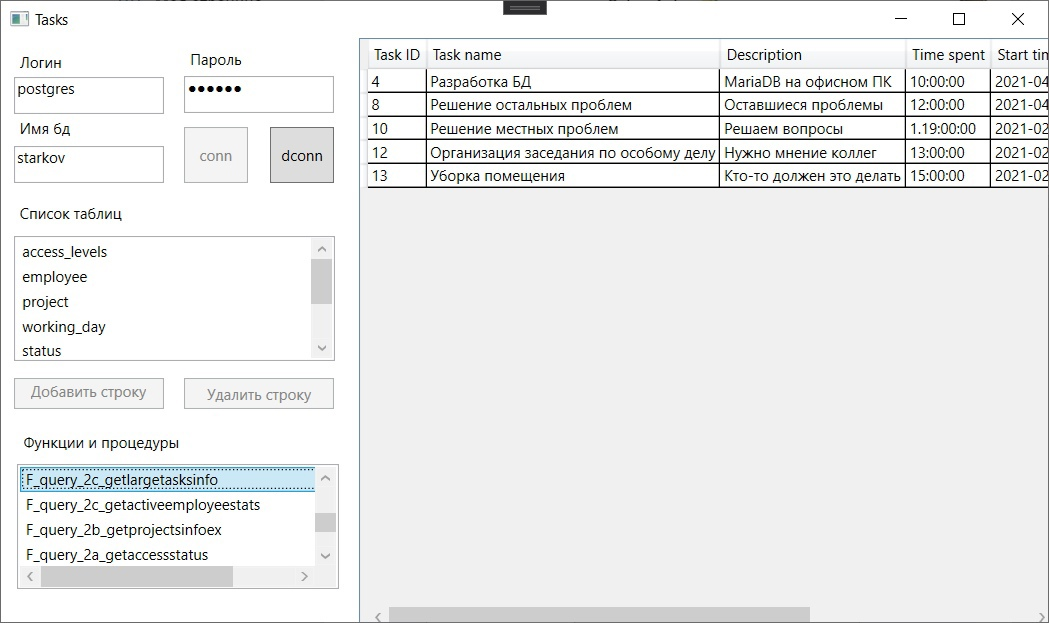
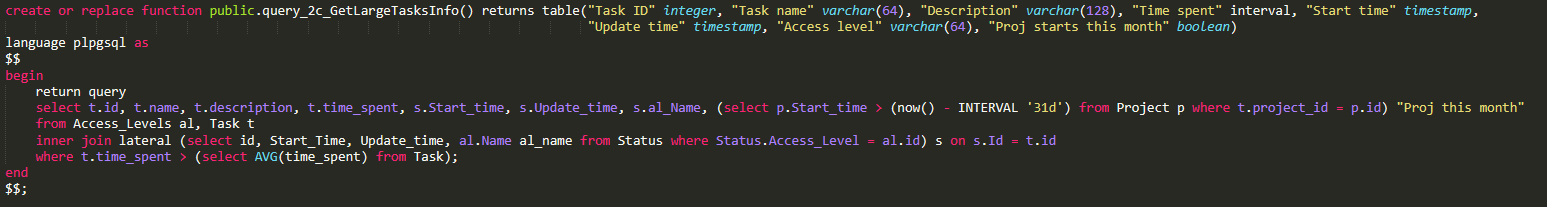
1. Составной многотабличный запрос с параметром, включающий соединение таблиц и CASE-выражение - запрос отображает для определённого проекта список заданий, исполнителя и может ли рабочий продолжать работу над заданием без вмешательства менеджера:



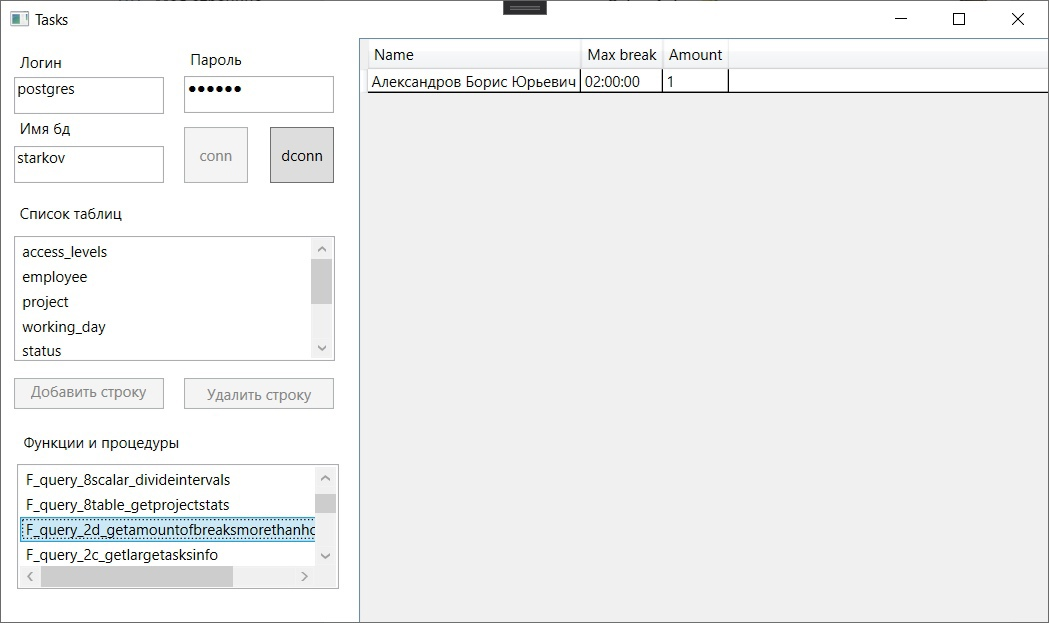
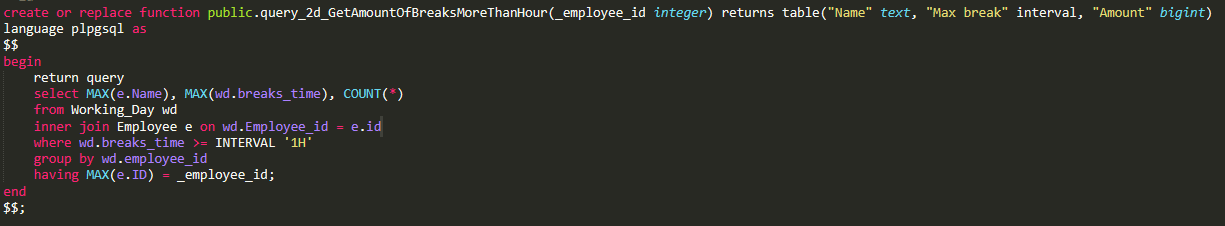
1. На основе обновляющего представления (многотабличного VIEW), в котором критерий упорядоченности задает пользователь при выполнении - представление отображает все задания, их проекты, затраченное время, ответственных и статус заданий:



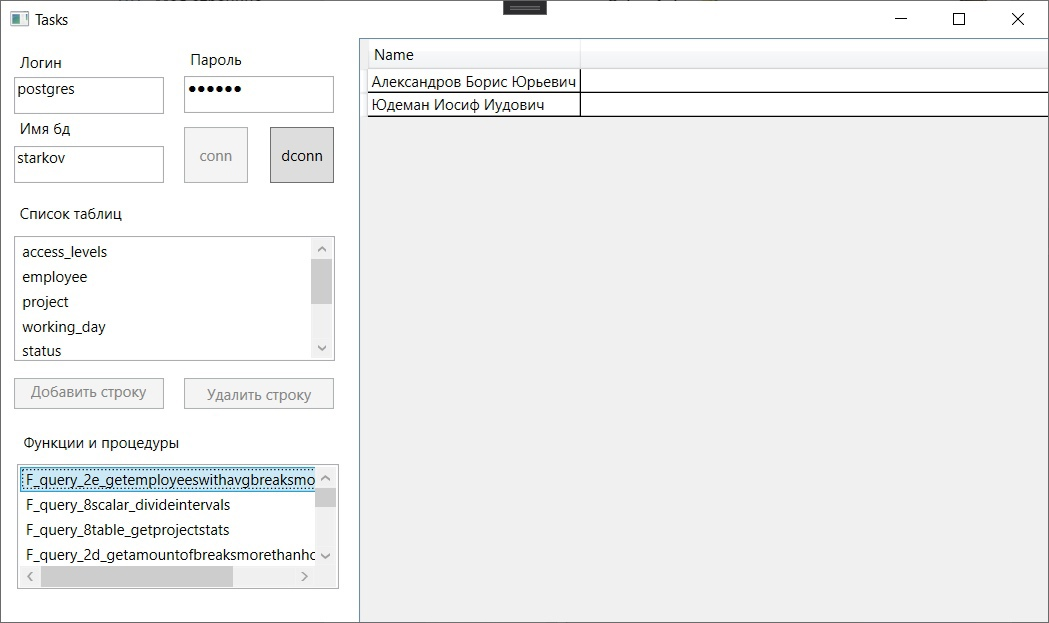
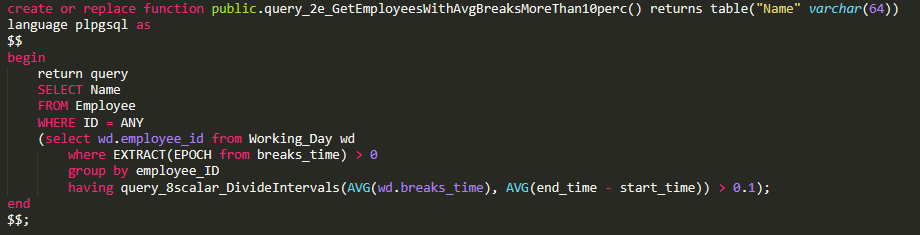
1. Запрос, содержащий коррелированные и некоррелированные подзапросы в разделах SELECT, FROM и WHERE (в каждом хотя бы по одному) - первый возвращает задания с длиной выше среднего и некоторую дополнительную информацию, второй считает больше ли чем в среднем провёл каждый сотрудник на рабочем месте за время своих рабочих дней при условии, что он вообще имеет какие-то активные задания.



1. Многотабличный запрос, содержащий группировку записей, агрегативные функции и параметр, используемый в разделе HAVING - отображает количество и максимальный перерыв сотрудника за рабочие дни (в примере это сотрудник с ID 4):



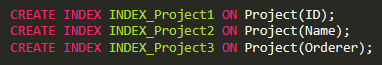
1. Запрос, содержащий предикат ANY(SOME) или ALL - возвращает список людей с интервалами, занимающими большую часть рабочего дня чем положено:



Создание индексов

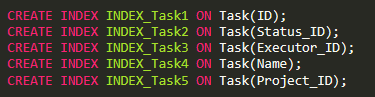
Каждой таблице кроме Access\_Levels были присвоены индексы





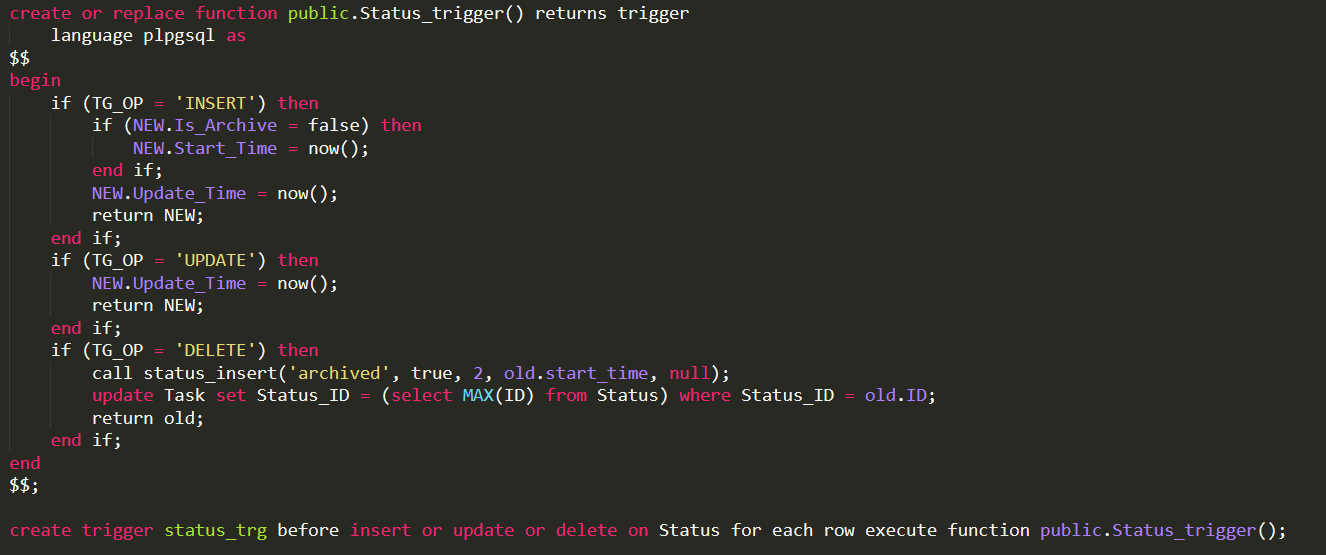






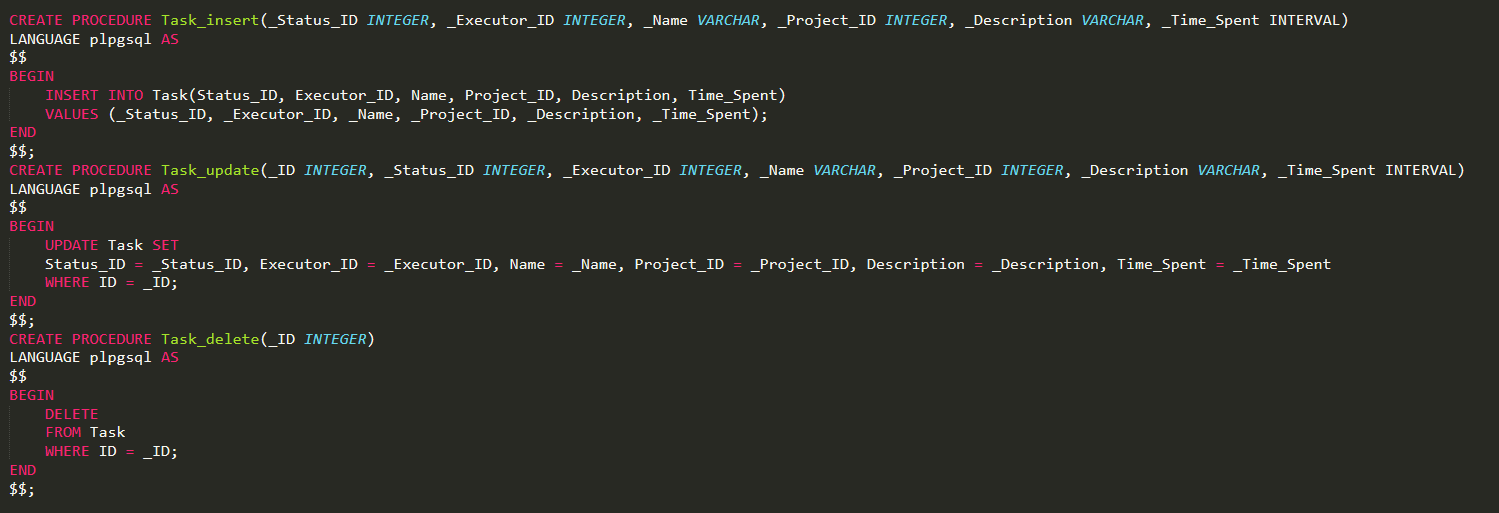
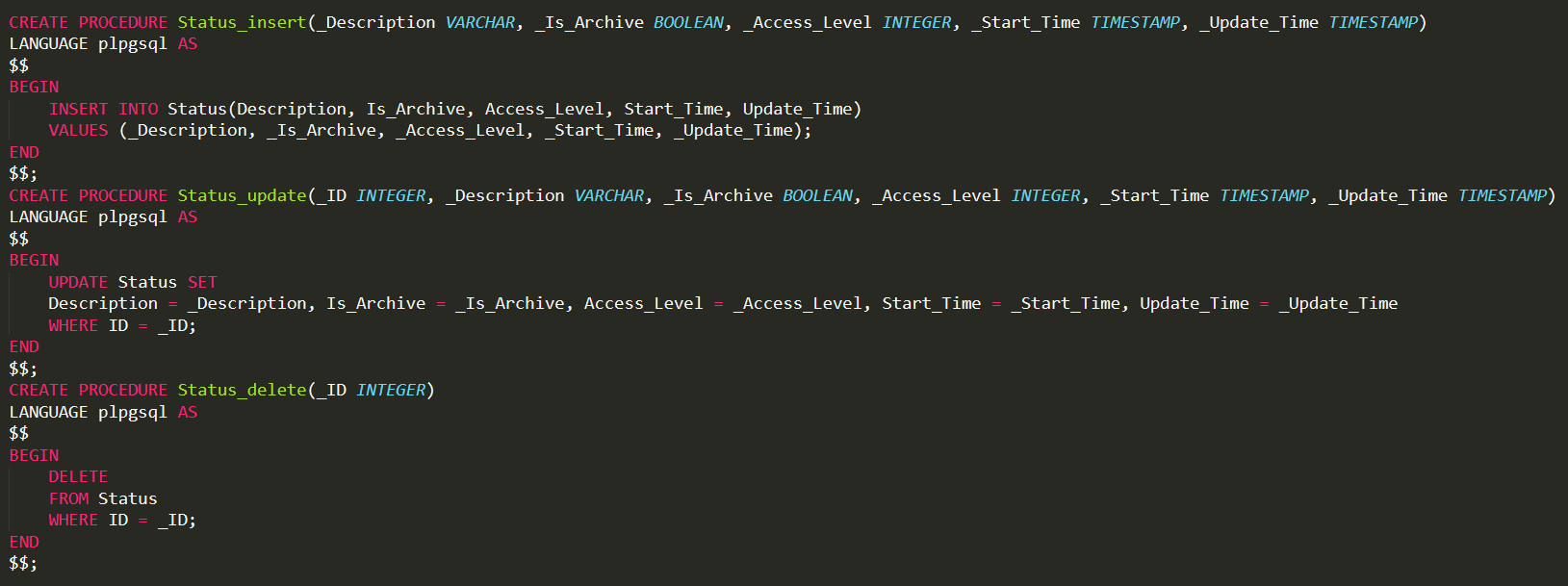
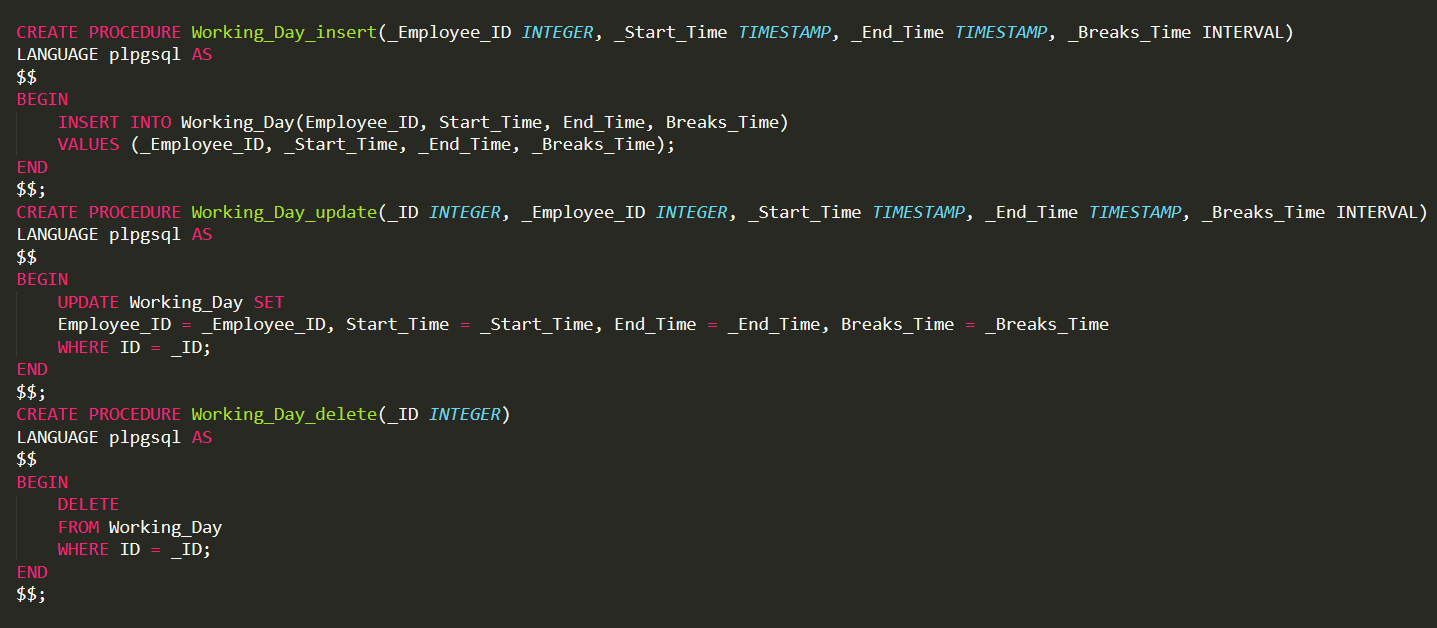
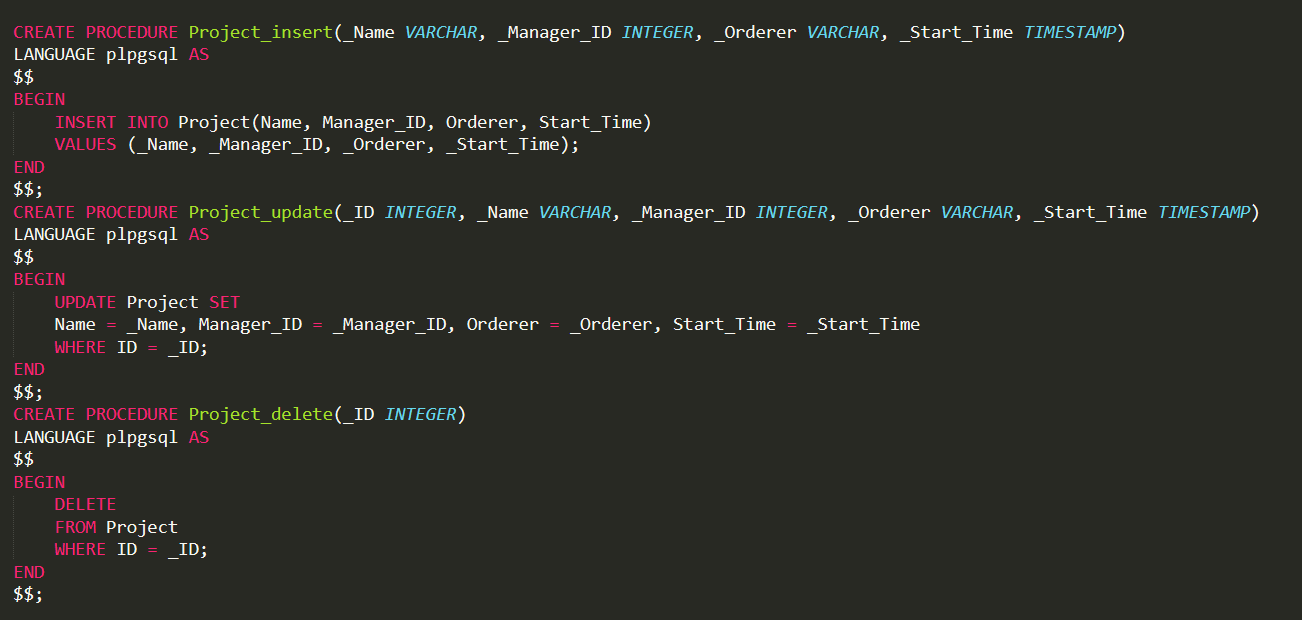
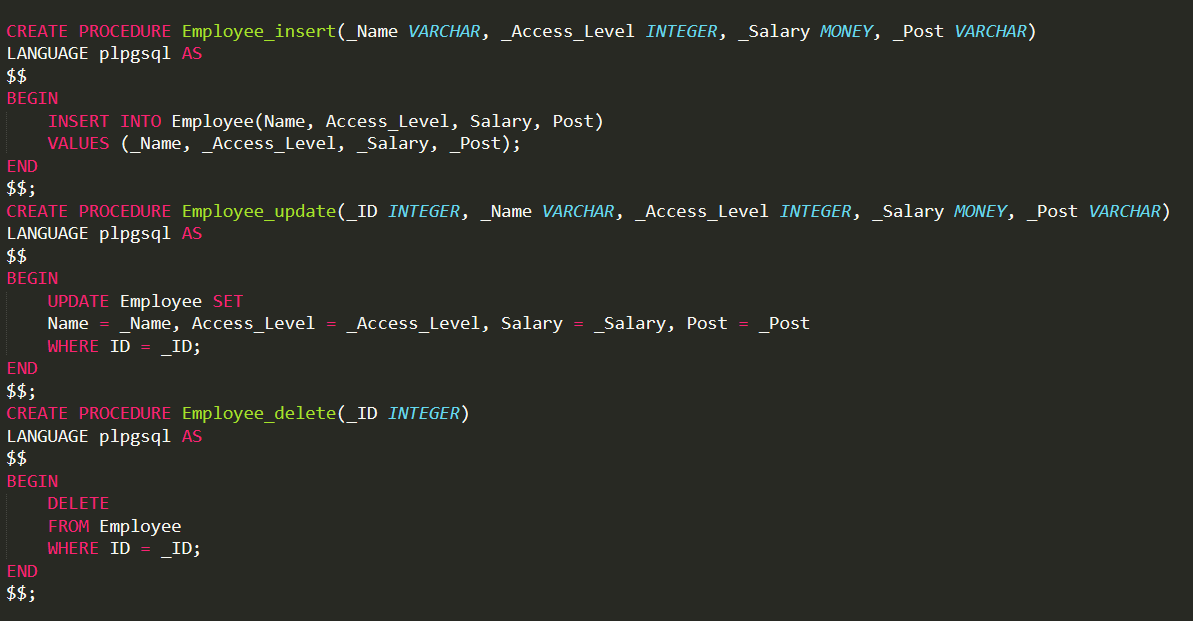
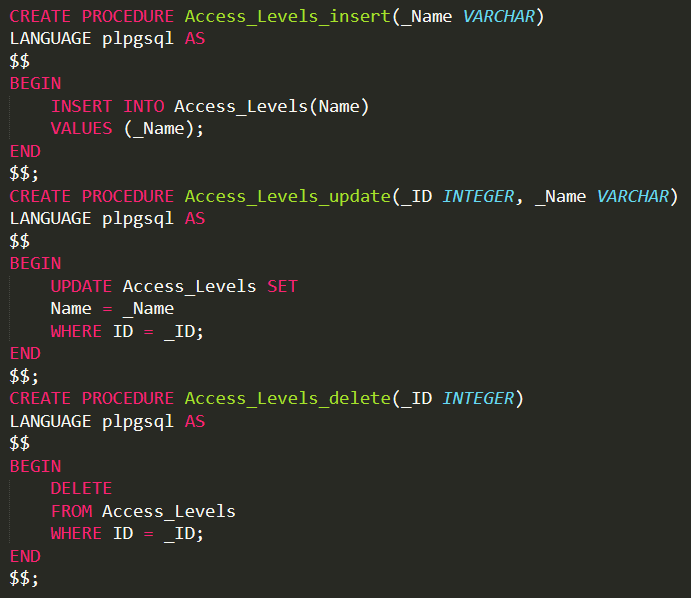
# Создание триггеров

Триггеры, реализованные в данной БД, автоматизируют работу со статусами заданий. По добавлению записи у неё ставятся даты на текущий момент, по обновлению меняется дата обновления, а вместо удаления создаётся новая запись статуса с архивным статусом, ссылка на которую передаётся в запись задания.



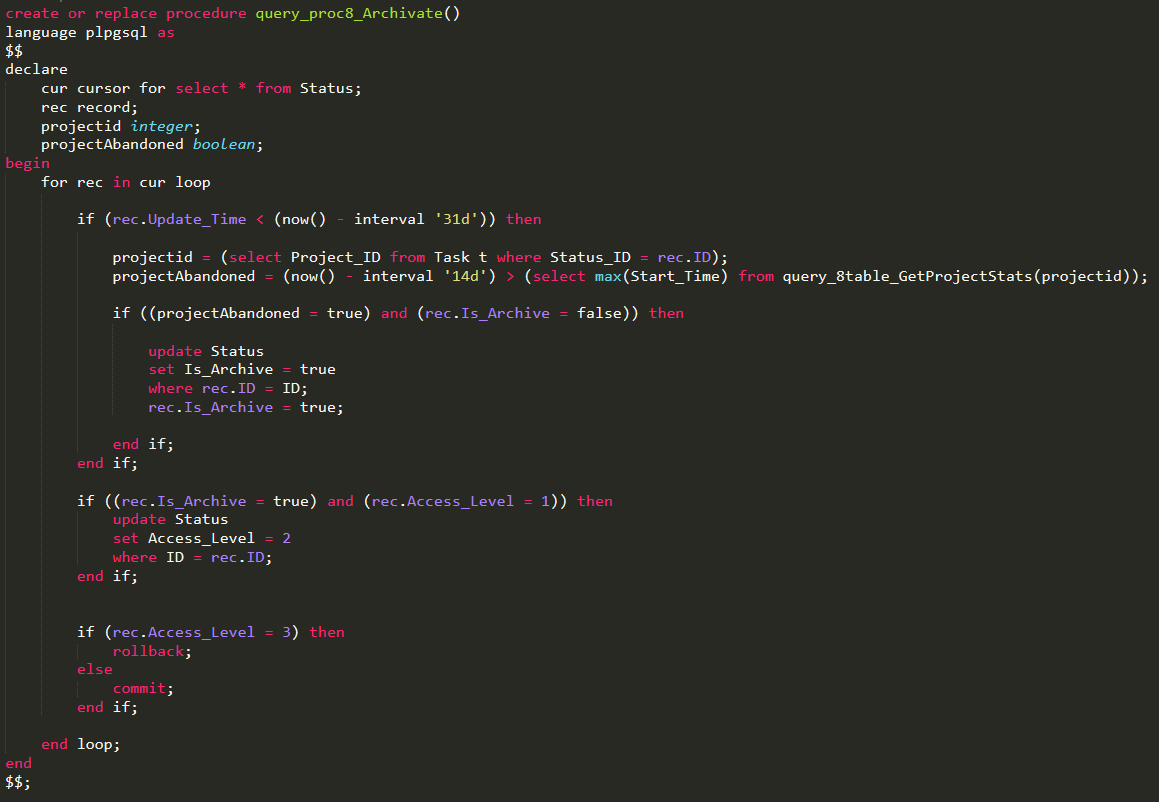
Хранимые процедуры

Для всех таблиц было создано по 3 процедуры для создания, изменения и удаления записей:

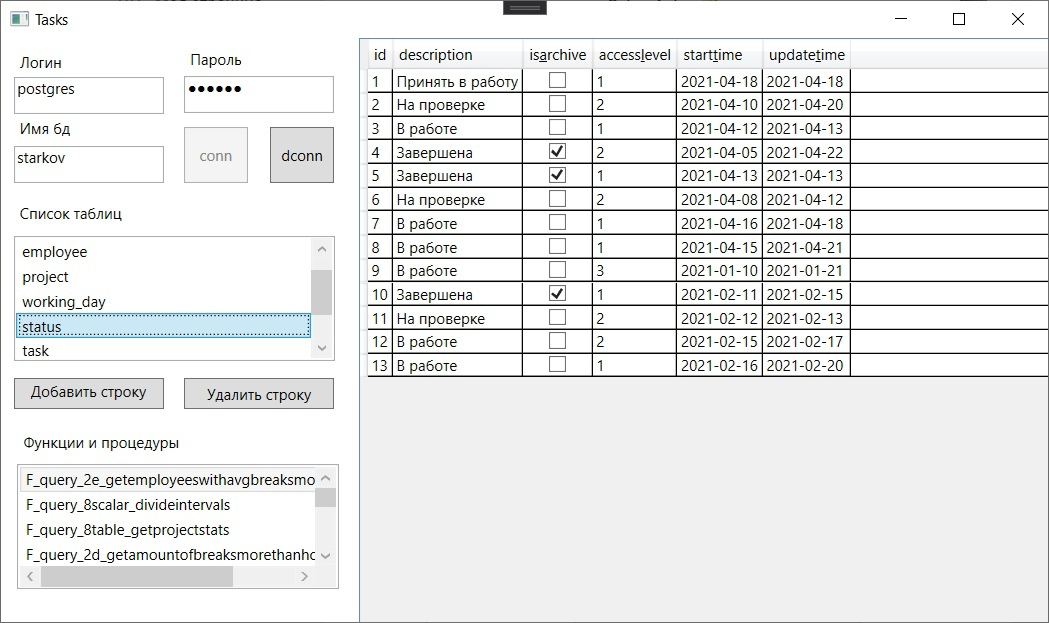


Транзакция

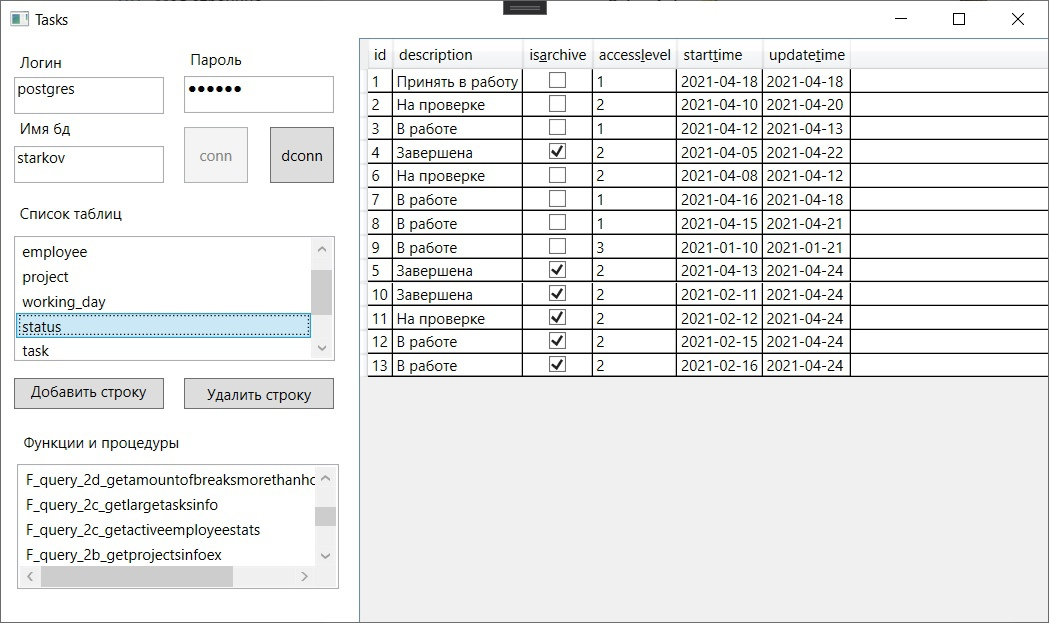
Транзакция была реализована в идущей далее процедуре, меняющей статус заданий в заброшенных проектах на архивный и уровень прав, необходимый для продолжения этих заданий, на как минимум второй. Откат производится при попытке архивировать задание, в котором ожидается действие лица с третьим уровнем доступа - директором предприятия, в других ситуациях транзакция фиксируется.



До:



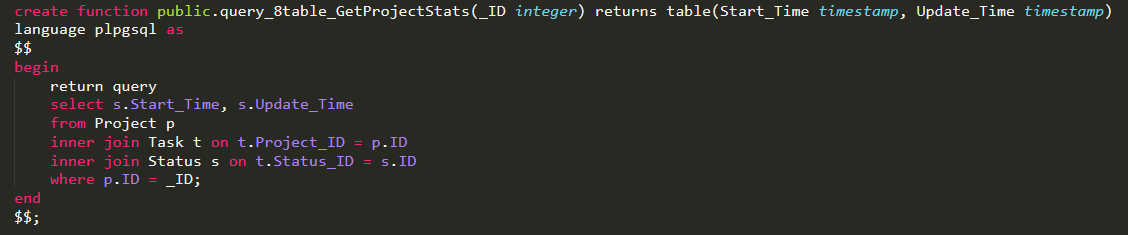
После:



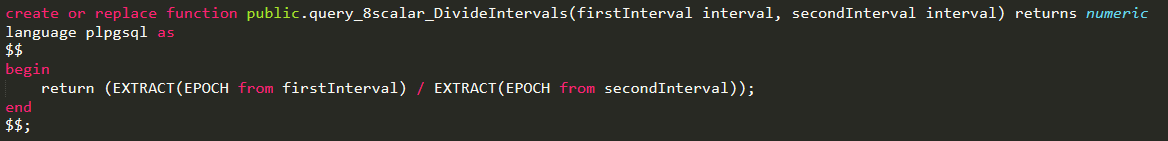
Курсор

Курсор был реализован в той же самой процедуре (см. предыдущий пункт) и используется для обхода всей таблицы Status и обновления данных.

Процедура и использование функций

Процедура была описана выше. Из неё вызывается табличная функция, возвращающая время начала и обновления всех заданий связанных с конкретным проектом:

В запросе 2е используется скалярная функция для деления интервалов друг на друга:



# Распределение прав

В данной бд есть три роли с разным набором привилегий и по пользователю на каждую



Заключение

В рамках данной курсовой работы мной была спроектирована и создана база данных, создан клиент для работы с ней. Я получил большое количество практического опыта, подкрепившего теоретические знания, полученные при изучении предмета.

Список литературы

1. Concurrent Programming on Windows, Joe Duffy (2008)

2. C# 7.0 in a Nutshell: The Definitive Reference, Joseph Albahari (2017)

3. Beginning Databases with MySQL, Neil Matthew (2002)