

МИНОБРНАУКИ РОССИИ

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«МИРЭА - Российский технологический университет»**

**РТУ МИРЭА**

Институт комплексной безопасности и специального приборостроения

Кафедра КБ-4 «Интеллектуальные системы информационной безопасности»

**ОТЧЕТ по дисциплине**

**«Клиент-серверные СУБД»**

По курсовой работе

Студент: Челышев Степан

Шифр учебной группы: БСБО-??-18

Руководитель: Иванова И.А.

Москва 2021 г.

**Оглавление**

# Здесь будет оглавление

# **Задание на курсовую работу**

Тема: «Разработка клиент-серверного приложения»

Предметная область (ПрО): овощебаза.

Цель работы: Закрепление теоретического материала по предмету «Клиент-серверные СУБД», а также по другим ранее изученным предметам.

Задание: Разработать клиент-серверное приложение, серверная часть которого реализована на PostgreSQL, а клиентская часть - на любом языке программирования. Приложение должно содержать модель предметной области в соответствии с вариантом.

Также курсовая работа должна удовлетворять следующим требованиям:

1. Содержит в себе не менее заполненных 10 таблиц, в каждой таблице не менее 3-8 кортежей (записей). Все записи должны содержать корректные данные, соответствующие предметной области курсовой работы.

2. В БД должно реализовано многопользовательский доступ и разграничение прав пользователя (минимально - пользователь и администратор). У пользователя должен быть ограничен функционал работы с базой данных;

3. Пароли всех пользователей должны хранится в зашифрованном виде.

4. В базе данных должна присутствовать хотя б одна хранимая процедура (или триггер), которая(-ый) должна(-ен) выполнять автоматизацию функционала конкретной БД.

5. В базе данных должно присутствовать хотя б одно представление, которое коррелирует с темой курсовой

6. В базе данных должна присутствовать хотя б одна функция, которая коррелирует с темой курсовой

7. В базе данных должна присутствовать хотя б одна транзакция, которая коррелирует с темой курсовой

8. В базе данных должна быть реализована защита от SQL-инъекций. Метод защиты может быть любой, но его необходимо обосновать.

9. Для работы с базой данных должен быть реализован графический пользовательский интерфейс (GUI), выполняющий упрощённую работу с бд (автоматизация запросов, поиск, вывод таблиц БД и так далее в зависимости от темы курсовой работы). Интерфейс может быть любой - веб-интерфейс/ интерфейс на любом языке программирования/мобильный интерфейс и т.д.

**Введение**

База данных — это организованная структура, предназначенная для хранения, изменения и обработки взаимосвязанной информации, преимущественно больших объемов. Для управления базами данных необходимы СУБД - системы управления базами данных. СУБД связаны со следующими возможностями:

- Разграничение доступа к данным для разных пользователей

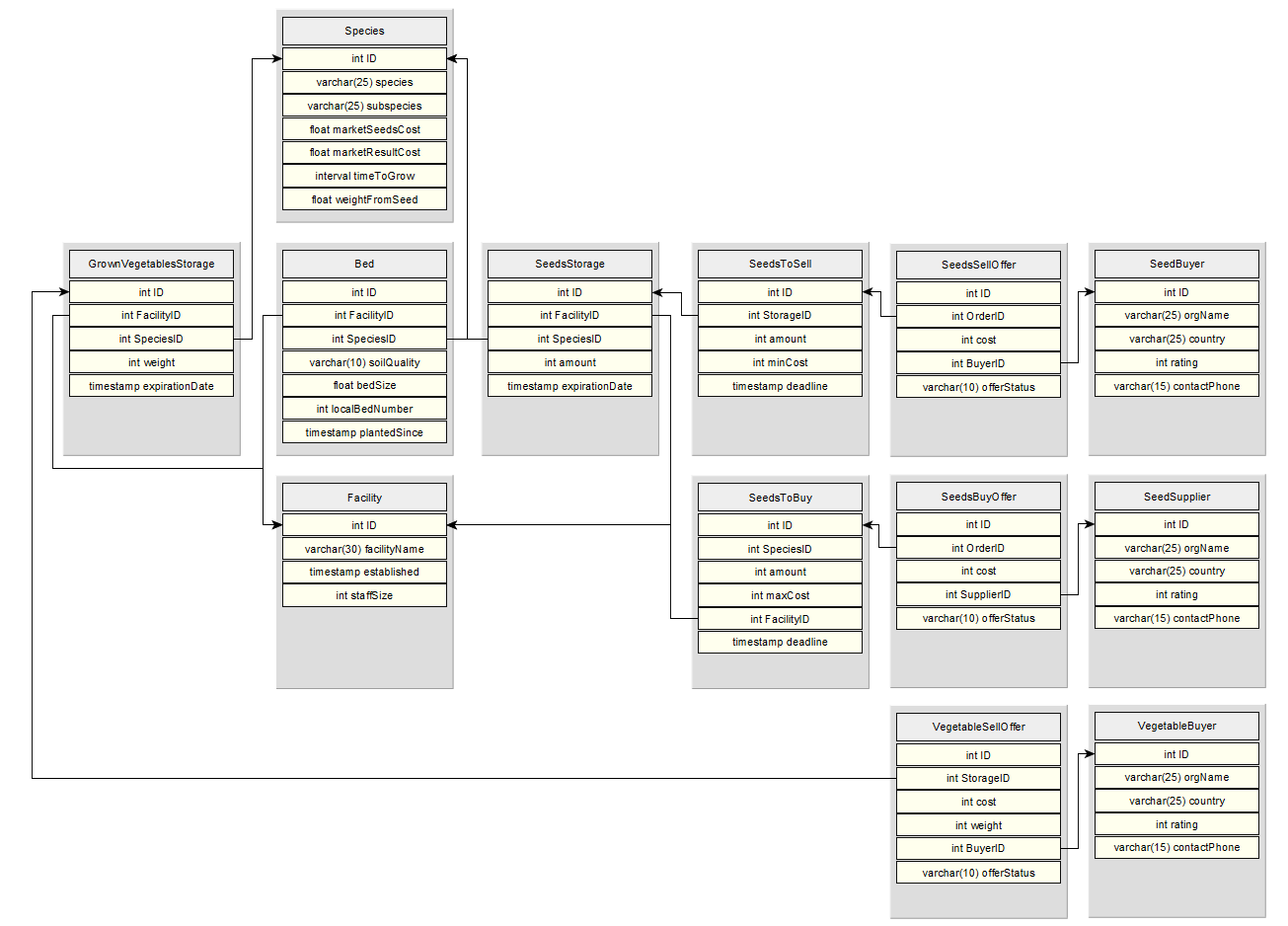
- Управление данными во внешней и оперативной памяти

- Журнализация вносимых изменений

- Резервное копирование и восстановление базы данных после сбоев

В рамках курсовой работы используются такие технологии как PostgreSQL (субд), C# (ядро клиента), WPF (интерфейс), Npgsql (подключение к бд).

# **Структура БД**



На данном изображении видны все поля в БД и их типы. Ключевым полем каждой таблицы является поле ID. Поля, от которых ведут стрелки, содержат в себе внешний ключ, который указывает туда, куда ведёт стрелка.

Facility представляет собой ядро структуры бд и означает подразделение овощебазы, содержит в себе GrownVegetablesStorage - хранилище созревших овощей, Bed - грядки и SeedsStorage - хранилище семян. Эти три элемента в свою очередь связаны с классификатором сортов овощей Species.

Все созревшие овощи по умолчанию находятся в состоянии продажи, для покупки-продажи семян же

**Содержание БД и таблиц**

Вся БД представляет собой информацию об автоцентре и его клиентах.

Названия таблиц и столбцов соответствуют их содержимому, далее идёт краткое их описание и пояснение уникальности некоторых записей.

**"Двигатель"** - таблица для хранения двигателей и их характеристик (мощность, производитель, тип и т.д.).

**"КПП"** – таблица, содержащая характеристики различных коробок передач (вид и кол-во передач).

**"Шины"** – таблица, содержащая различные наименования шин (отличаются сезонностью, производителем и моделью).

**"Диски"** - таблица, содержащая различные наименования дисков.

**"Колесо"** - таблица, объединяющая в себе **"Шины"** и **"Диски"**

**"Авто"** - таблица для всех автомобилей, представленных к продаже в автосалоне. Включает себя данные из всех вышеперечисленных таблиц, дополняя их госномером и различной справочной информацией

**"ПТС"** -

**"Записи ПТС"** -

**"Переч. Штрафов"** -

**"СТС"** -

**"Бухгалтер"** -

**"Паспорт"** -

**"Продавец"** -

**"Эксперт"** -

**"Оператор"** -

**"Доверенность"** -

**"Дов-сть авто"** –

**"Заявка"** -

**"Контакты"** -

**"Координаты"** -

**"Налог.Дек."** –

**"РПДС"** -

**"Счет-фактура"** -

**"Договор"** -

**Программный продукт**

Для дальнейшей работы с программным продуктом и базой данных нужно совершить некоторые подготовительные действия — поднятие базы данных PostgreSQL и компиляция приложения в Visual Studio.

Для работы с PostgreSQL необходимо его скачать: <https://www.postgresql.org/download/>

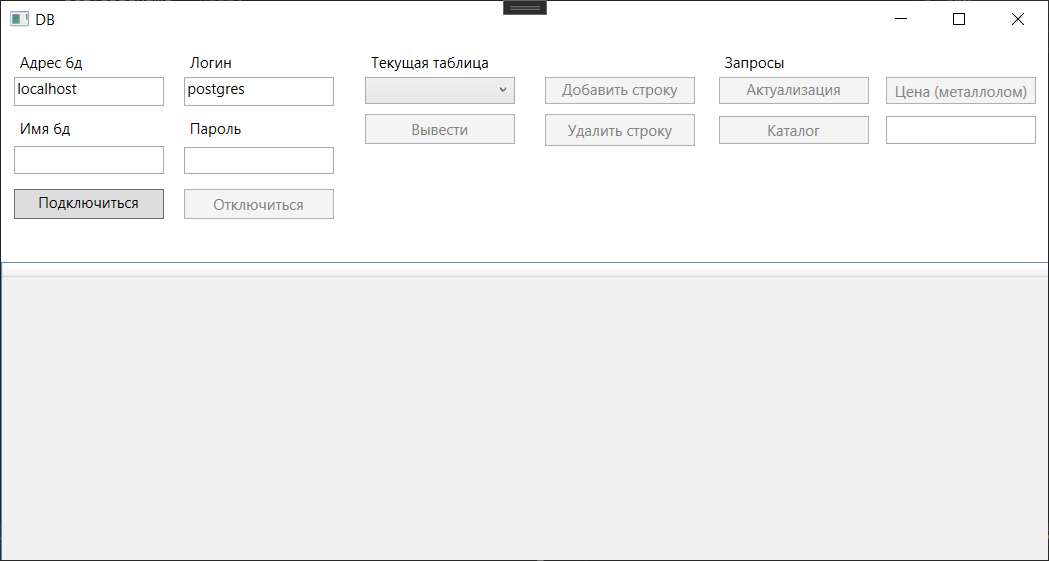
Закончив процесс установки и настройки PostgreSQL, нужно запустить pgAdmin, идущий с ним в комплекте, зайти под созданным ранее пользователем и воссоздать БД из файлов .sql, идущих с курсовой работой. В pgAdmin необходим вызвать контекстное меню для одной из баз данных (первоначальной или новой) и перейти к инструменту "Query Tool". Туда нужно скопировать текст из .sql файлов и выполнить.

Скомпилировать проект можно установив Visual Studio <https://visualstudio.microsoft.com/ru/downloads/> . В ходе установки необходимо в числе прочих выбрать пакет разработки под платформу .NET.

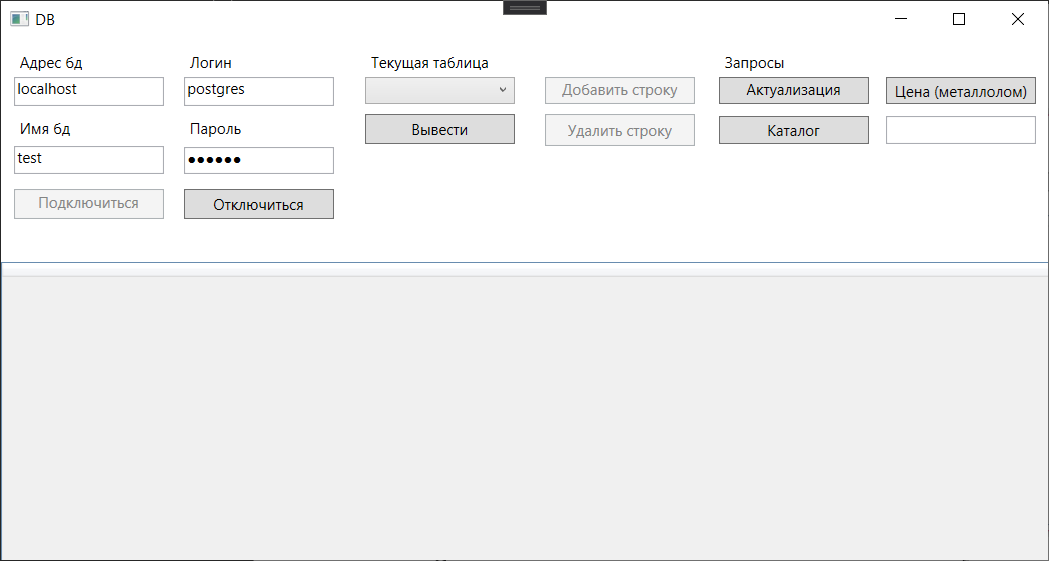
Открыв проект в Visual Studio, нажимаем на «Пуск» в верхней части экрана, приложение компилируется и запускается.

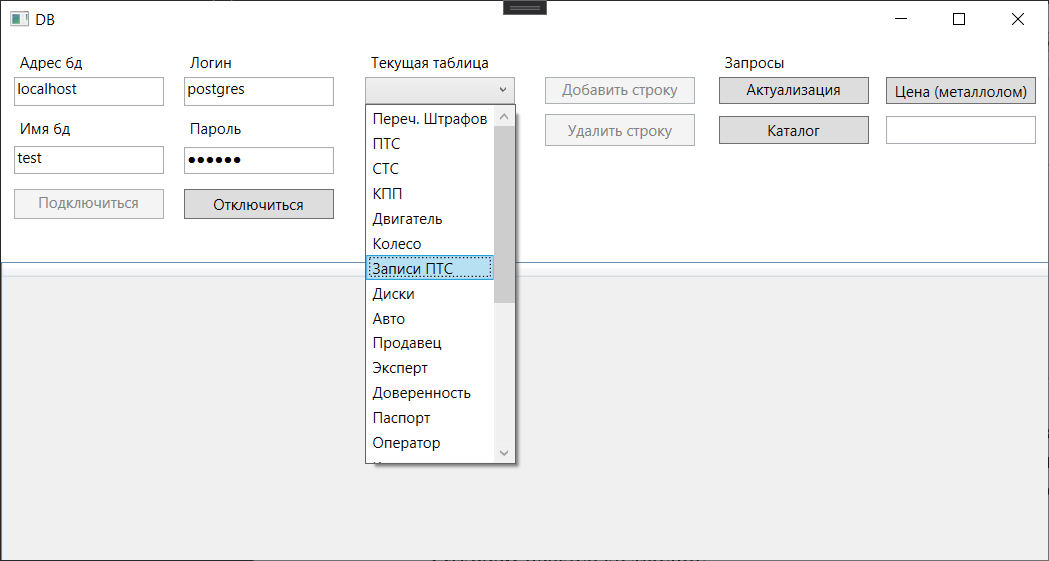
Среда для работы с БД готова.

Открыв приложение, можно увидеть следующий экран:

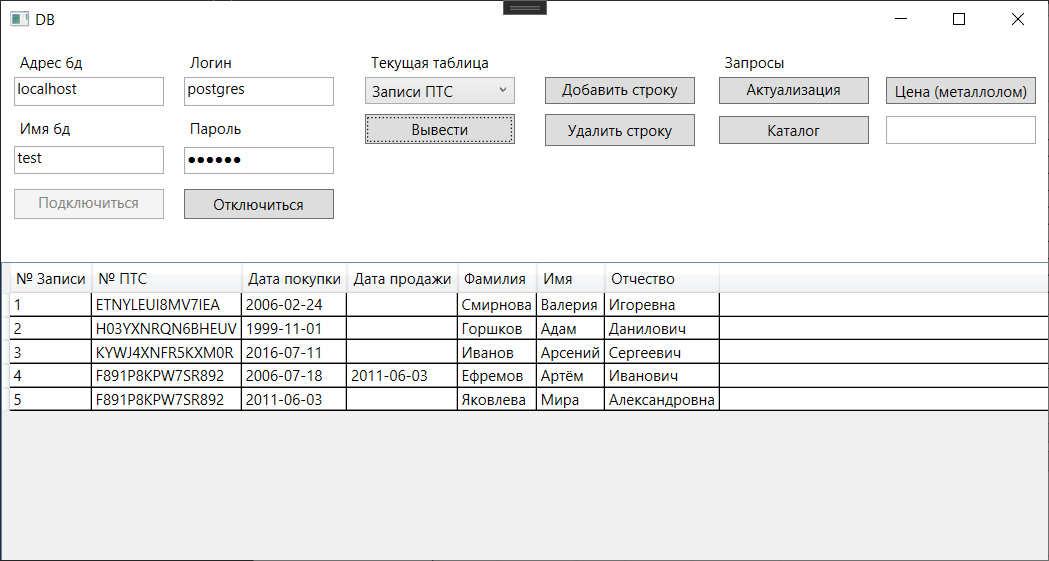
Это единственный и основной экран приложения. Для подключения к БД необходимо ввести данные для входа в соответствующие поля. Поле "Адрес бд" заполняется в соответствии с тем, где находится БД - localhost либо IP-адрес хоста. Поле "Имя" соответствует имени БД. Поля "Логин" и "Пароль" заполняются данными созданной при первоначальной настройке PostgreSQL учётной записи либо других тестовых учетных записей с разными привилегиями, список которых можно найти в "03 - roles.sql". Стандартный логин - postgres.

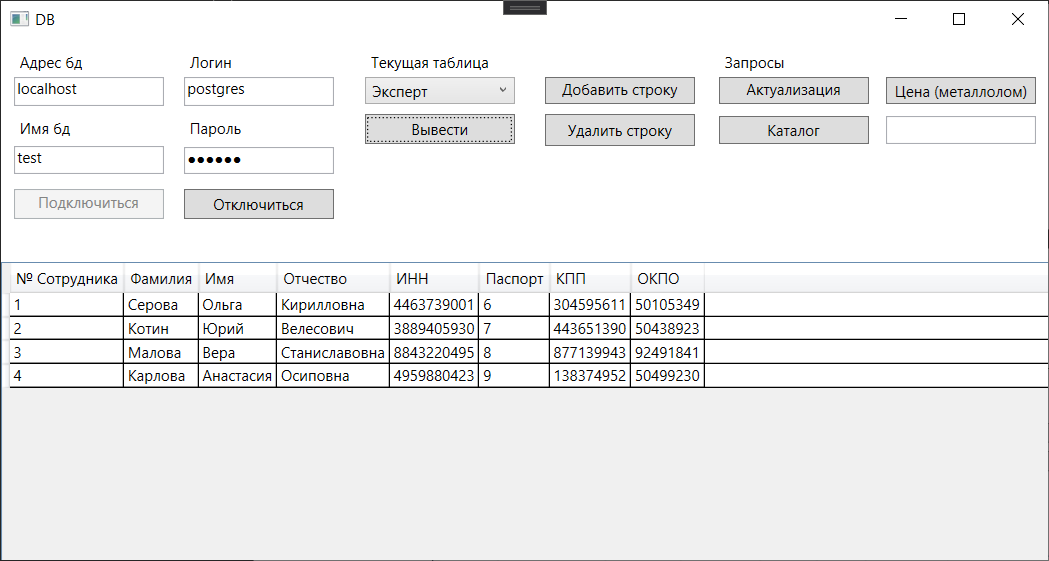
После заполнения этих полей, нужно нажать на кнопку подключения.

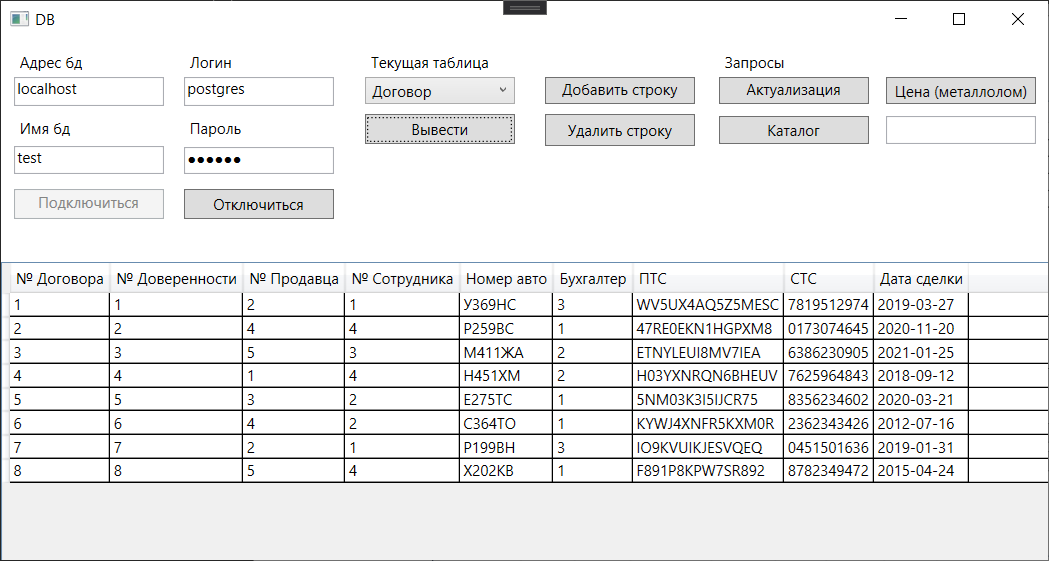


Активировались кнопки, в ComboBox загрузились все доступные для пользователя таблицы.

Проверим содержимое этих таблиц:







Чтобы изменить какое-то значение в таблице, достаточно ввести новое значение на его место в отображаемой сетке. Завершение редактирования поля отправит UPDATE запрос с этим значением на сервер. При недостаточных для этого правах пользователь не сможет изменить значение и получит сообщение об ошибке.

**Результат работы**

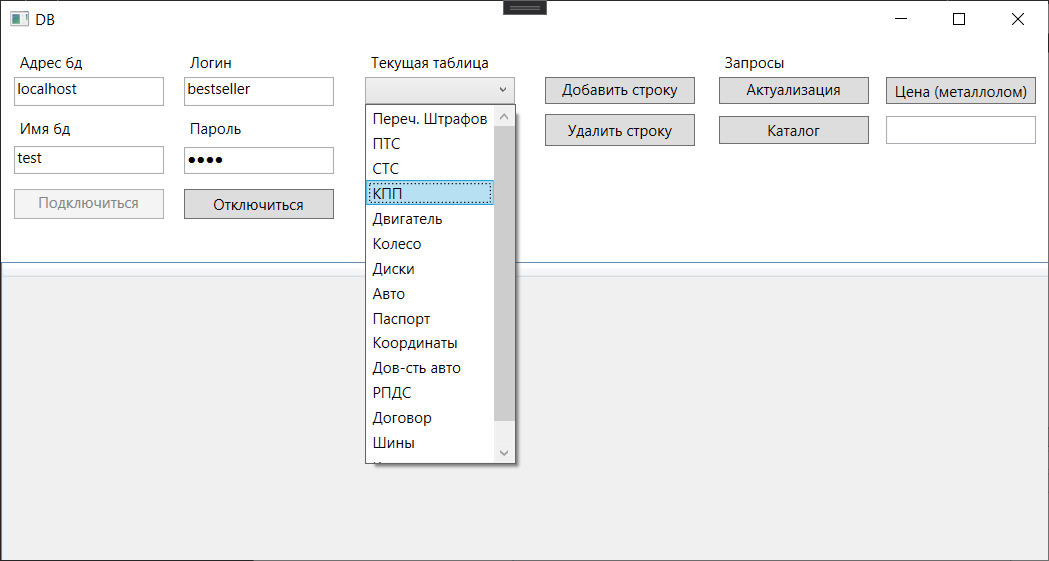
Для демонстрации результата вернёмся к задачам, поставленным перед выполнением курсовой работы и перечисленным в пункте "Введение".

1. Содержит в себе не менее заполненных 10 таблиц, в каждой таблице не менее 3-8 кортежей (записей). Все записи должны содержать корректные данные, соответствующие предметной области курсовой работы.

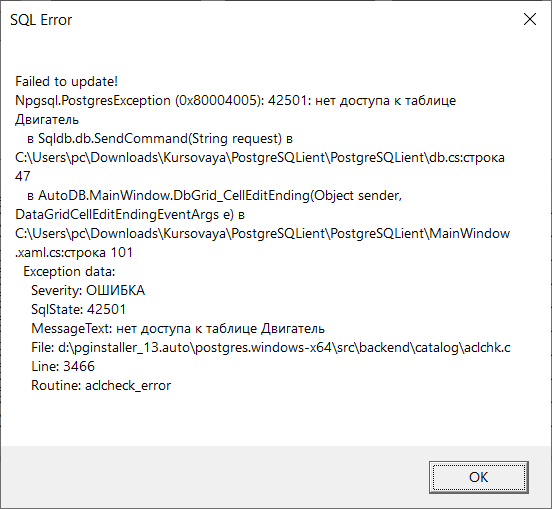
- Уже было продемонстрировано в разделе "Структура БД"

2. В БД должно реализовано многопользовательский доступ и разграничение прав пользователя (минимально - пользователь и администратор). У пользователя должен быть ограничен функционал работы с базой данных;

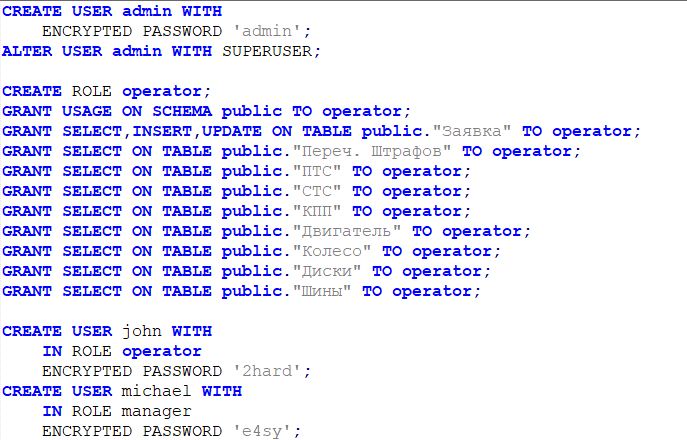
- Наличие многопользовательского доступа было продемонстрировано в пункте "Программный продукт", но не были напрямую продемонстрированы ограничения функционала для разных пользователей. Чтобы показать это разграничение, воспользуемся одним из дополнительных пользователей для входа - bestseller.



Уже видно, что не все таблицы доступны. Попробуем изменить данные в одной из таблиц, к которой у данного пользователя неполный доступ:



Всех доступных пользователей можно увидеть в файле "03 - roles.sql" вместе с их разрешениями. Здесь видны роли operator, seller и accountant с конкретными разрешениями и пользователи, привязанные к данным ролям, а также пользователь admin-admin.



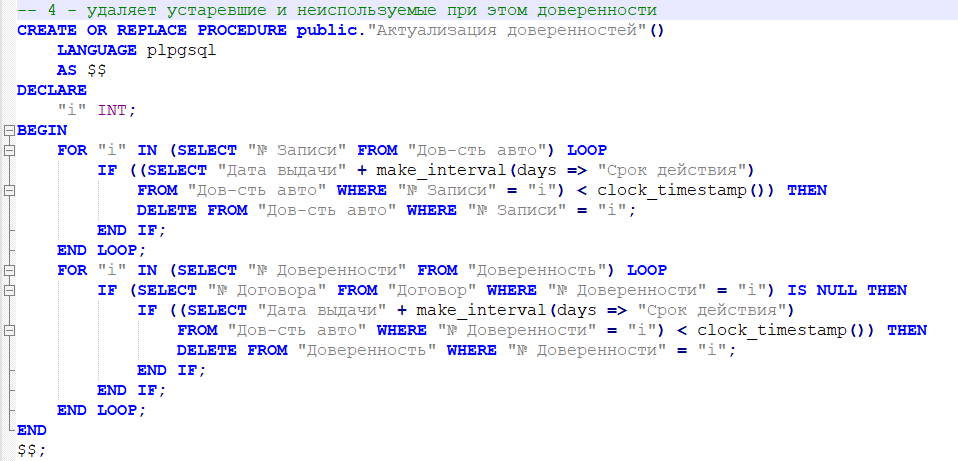


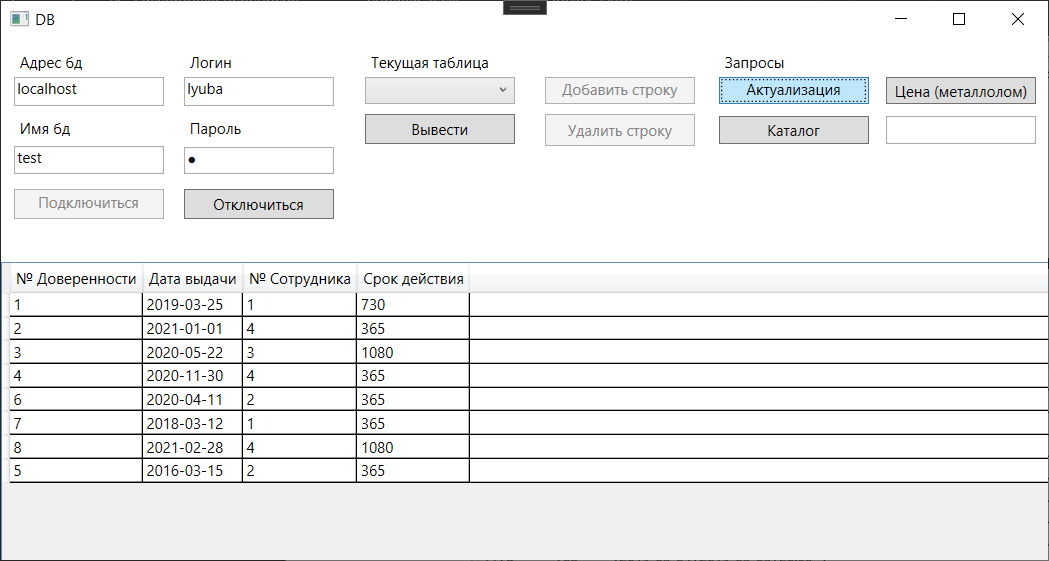
3. Пароли всех пользователей должны хранится в зашифрованном виде.

- Если обратить внимание на код выше, можно заметить, что при создании каждого пользователя пароль задаётся с ключевым словом ENCRYPTED, что сообщает СУБД напрямую, что данный пароль следует хранить зашифрованным.

4. В базе данных должна присутствовать хотя б одна хранимая процедура (или триггер), которая(-ый) должна(-ен) выполнять автоматизацию функционала конкретной БД.

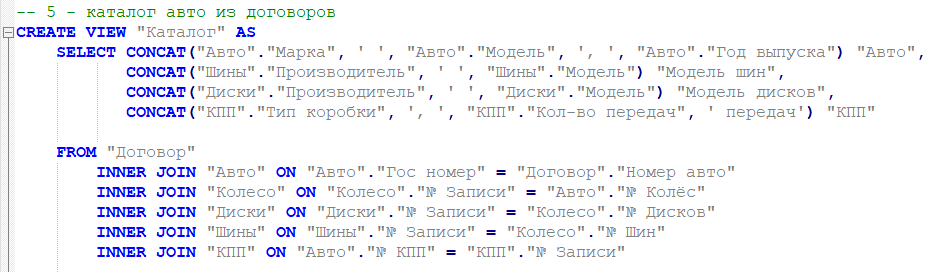
- Была реализована процедура “Актуализация доверенностей”

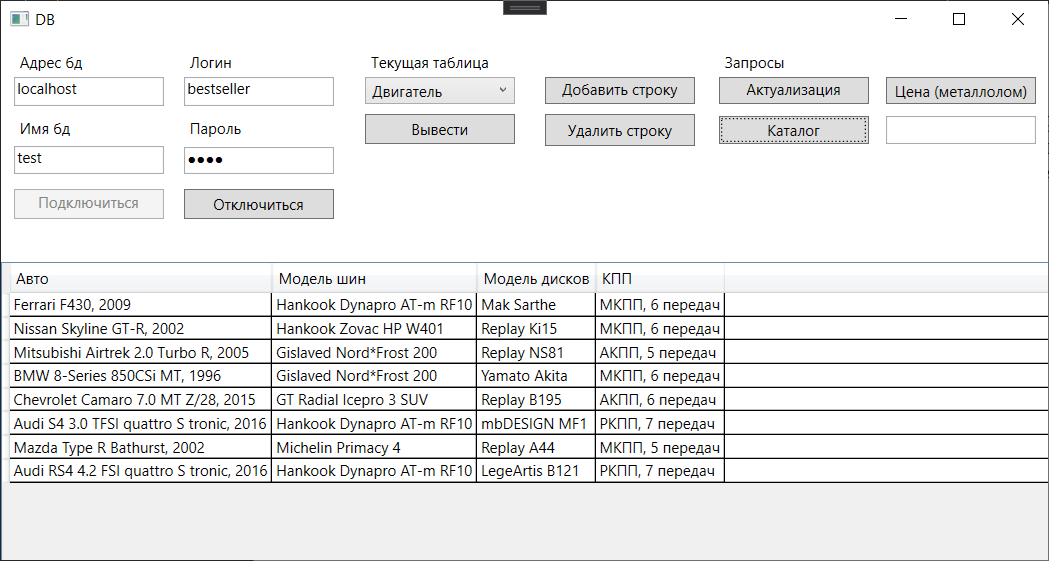


Данная процедура призвана помочь бухгалтерам и убирает доверенности, срок действия которых истёк и которые не фигурируют ни в каких документах.

5. В базе данных должно присутствовать хотя б одно представление, которое коррелирует с темой курсовой

- Было реализовано представление “Каталог”:

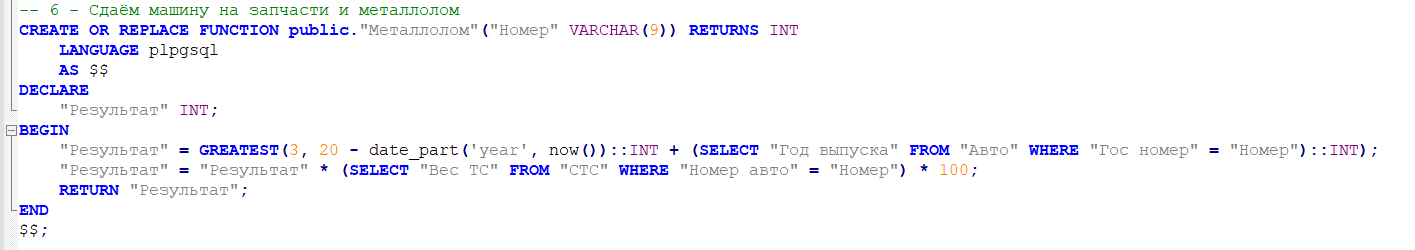




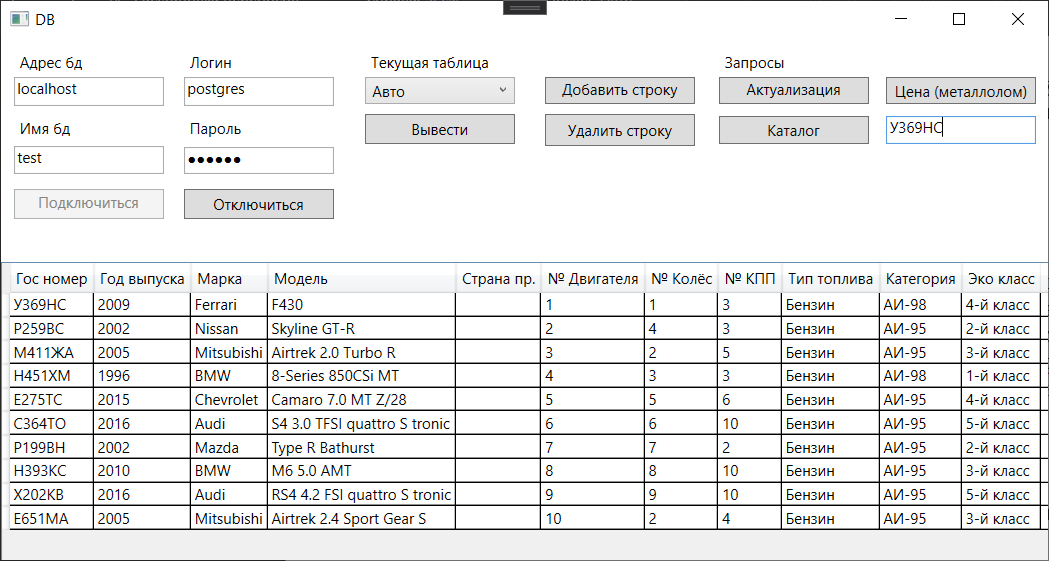
Данное представление отображает содержимое таблиц Авто, Колесо, Диски, Шины и КПП, объединяя их и приводя к удобочитаемый каталог. Доступно всегда (если у пользователя есть права).

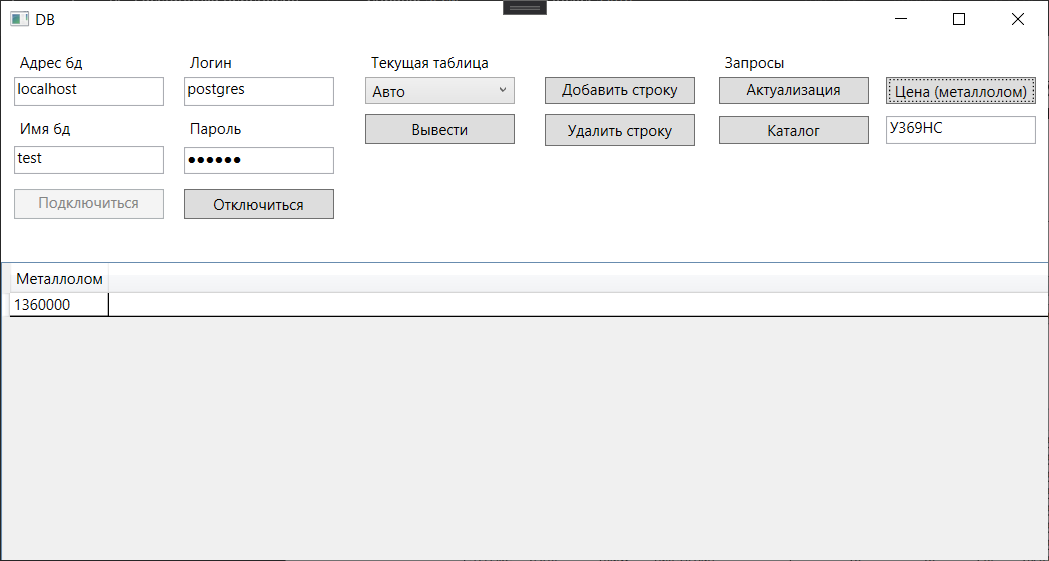
6. В базе данных должна присутствовать хотя б одна функция, которая коррелирует с темой курсовой

- Была реализована скалярная функция “Металлолом”



Данная функция высчитывает сумму, которую можно получить за автомобиль, если продать его на запчасти и металлолом, на вход подается номер автомобиля. Расчёт идёт исходя из года выпуска и веса авто.



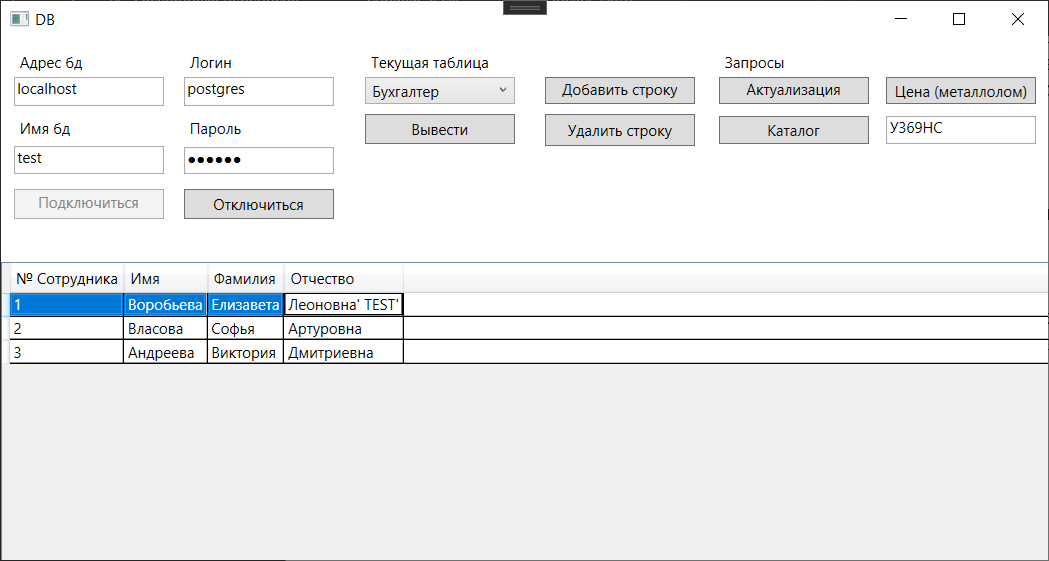


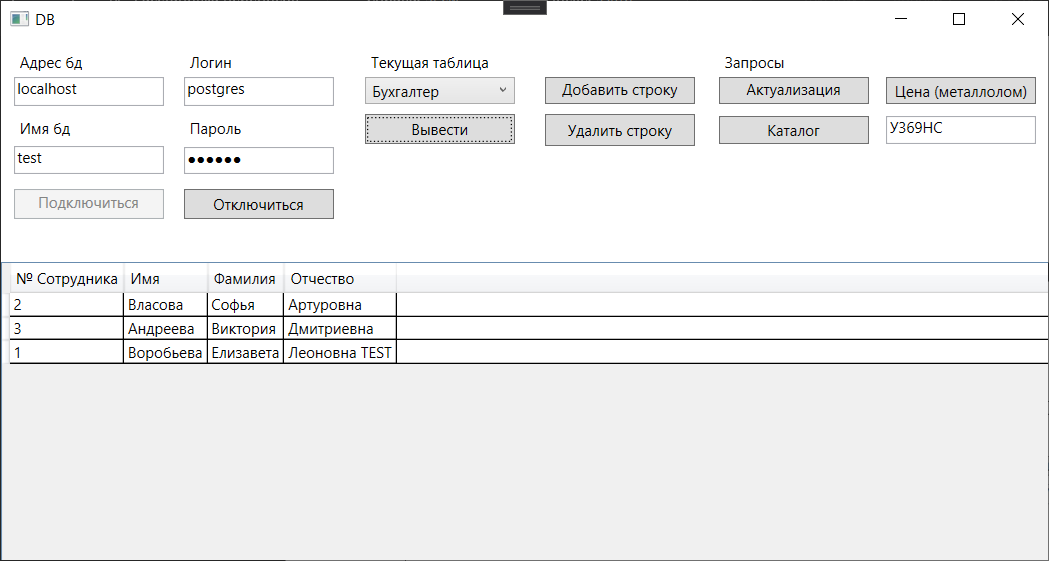
7. В базе данных должна присутствовать хотя б одна транзакция, которая коррелирует с темой курсовой

- Любая процедура/функция в PostgreSQL содержит в себе транзакцию

8. В базе данных должна быть реализована защита от SQL-инъекций. Метод защиты может быть любой, но его необходимо обосновать.

- В любом месте программы, где от пользователя ожидается текстовый ввод, идёт проверка на наличие одиночных кавычек ( ' ), наличие которых может свидетельствовать о попытке осуществления SQL-инъекции. Они автоматически удаляются из ввода, что позволяет предотвращать такие случаи.





9. Для работы с базой данных должен быть реализован графический пользовательский интерфейс (GUI), выполняющий упрощённую работу с бд (автоматизация запросов, поиск, вывод таблиц БД и так далее в зависимости от темы курсовой работы). Интерфейс может быть любой - веб-интерфейс/ интерфейс на любом языке программирования/мобильный интерфейс и т.д.

- Реализованный интерфейс был показан в пункте "Программный продукт" и далее.

**Заключение**

В ходе данной курсовой работы была разработана база данных, пригодная для использования в рамках своей предметной области, а также приложение с пользовательским интерфейсом, упрощающее работу с ней. Также, как показано в пункте "Результат работы", были выполнены все поставленные перед выполнением задачи.