

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: Соловьев И.?.

Шифр учебной группы: БСБО-??-18

Руководитель: Иванова И.А.

Москва 2021 г.

**Оглавление**

# Katana - 0

Переделать надо короче, автоматическое оглавление накати

# Задание на курсовую работу - 3

# Введение – 4

Анализ предметной области - 5

Структура БД - 5

- ER-Модель - 5

- Содержание БД и таблиц - 6

- Нормализация - 8

Программный продукт - 11

- Восстановление базы данных - 11

- Клиентское приложение - 12

Результат работы - 16

Заключение - 25

# **Задание на курсовую работу**

Тема: «Разработка клиент-серверного приложения»

Предметная область (ПрО): выкуп автомобилей с пробегом.

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.

**Введение**

База данных — это организованная структура, предназначенная

для хранения информации. Для управления базами данных необходимы СУБД - системы управления базами данных. Такие системы имеют следующие функции:

- 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 используется СУБД PostgreSQL, клиент написан на C#, интерфейс создан с помощью WPF, подключение к СУБД осуществляется через фреймворк Npgsql.

На выполнение курсовой работы были поставлены следующие задачи:

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.

**Анализ предметной области**

Автоцентр «Автоград», который было взято в качестве «живого» примера организации, осуществляющей продажу автомобилей с пробегом, и от структуры и функций которого я отталкивался при построении даталогической модели, имеет следующую структуру подразделений (Рисунок 1).

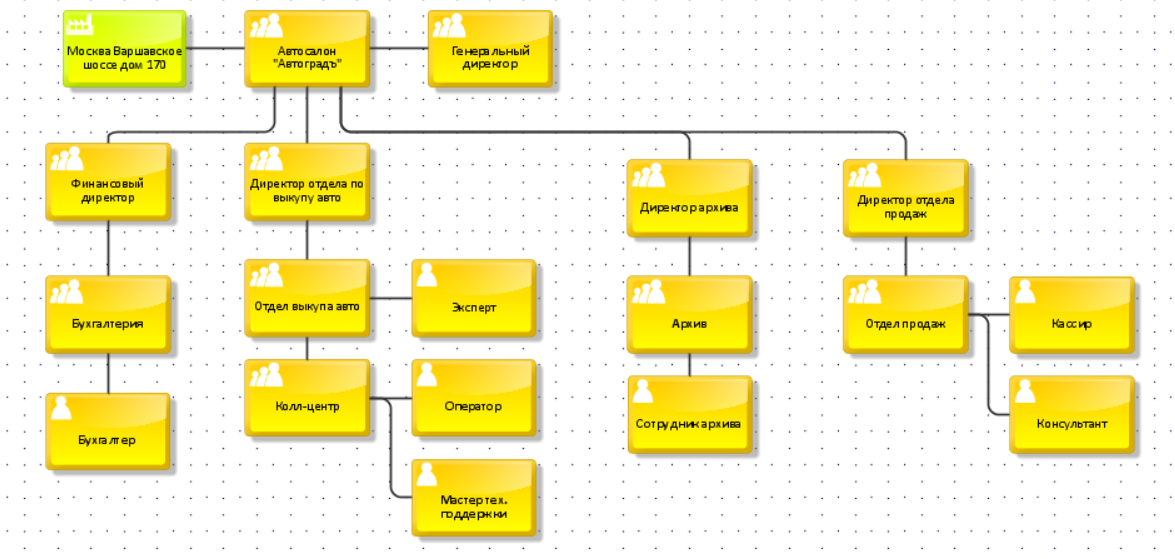


Рисунок 1 – Структура подразделений

В выкупе автомобиля с пробегом ту или иную роль играют:

1. Бухгалтерия.
2. Отдел выкупа авто.

При анализе должностной и функциональной структуры выше перечисленных подразделений (разумеется, с учётом специфики предметной области) я опирался на состав сотрудников автосалона и соответствующие должностные инструкции. Информация о составе рабочего коллектива взята из открытых источников (официального сайта).

Ниже представлена краткая концентрированная информационная «выжимка».

Отдел выкупа авто: Эксперт, колл-Центр. Колл-центр принимает заявки на выкуп авто от продавцов. С точки зрения ПрО, важны дата заявки, ФИО продавца, а также табельный номер оператора, принявшего заявку и название выкупаемой машины. Эксперт из отдела выкупа авто оценивает автомобиль, который хочет выкупить компания, устанавливает стоимость выкупа, восстановления и ремонта, а так же стоимость продажи. С точки зрения ПрО, важны лишь ФИО, а также табельный номер работника.

Бухгалтерия: Финансовый директор, Бухгалтер. Бухгалтер одобряет или отклоняет заявки Эксперта по выкупу автомобилей.

Полное и исчерпывающее понимание рассматриваемого бизнес-процесса возможно только на основании тщательного анализа потока нормативных документов, «циркулирующих» в организации.

Значимыми и основными такими документами являются:

1. Договор о покупке (Договор) – основополагающий документ покупки машины у Продавца.

Содержит информацию о сторонах договора: Эксперте и Продавце, а также о выкупаемом автомобиле.

Подписывается сторонами договора.

1. Паспорт транспортного средства (ПТС).

Документ, в котором содежатся сведения о автомобиле, его технической части и прежних владельцах.

1. Свидетельство о регистрации транспортного средства (СТС)

Документ, в котором так же содержатся сведения о автомобиле, необходим для постановки его в будущем на учет.

1. Расписка о получении денежных средств (РПДС).

Расписка, которую дает Продавец, необходима для отсутсвия конфликтов в будущем.

1. Доверенности.

Данный документ необходим для того, чтобы Эксперт мог заключать договора от имени компании.

1. Налоговая декларация

Документ, составляемый Бухгалтером, для налоговой инспекции

Исходя из вышесказанного, приблизительная обобщённая схема и хронология рассматриваемого бизнес-процесса (а также прямо или косвенно, но близко с ним связанных процессов) выглядят следующим образом.

Этап 1. Создание заявки на продажу.

На данном этапе происходит запись Оператором информации от Продавца о машине: марка, модель, ее состояние и тд. А так же информацию о самом Продавце. После заявка передается Эксперту.

Этап 2. Оценка машины.

Эксперт связывается с Продавцом, узнает все необходимые данные о машине, после чего Продавец обязан привезти машину в автосалон, где будет произведен ее осмотр Экспертом.

Эксперт проводит осмотр машины и оценивает ее. Называя стомость покупки Продавцу, а так же передает в Бухгалтерию информации о стоимости восстановления машины и предполагаемой стоимости продажи в автосалоне.

Также Эксперт может отказать Продавцу в покупке в случае, если машина окажется неликвидным товаром.

Этап 3. Покупка и оформление.

После осмотра и оценки Эксперт заключает Договор о покупке с Продавцом. После чего передает деньги Продавцу и составляется расписка о получении денег. Затем Договор о покупке подписывается обеими сторонами.

Продавец передает Эксперту все документы на машину (СТС и ПТС).

Таким образом, получается внятная целостная картина исследуемого бизнес-процесса, в котором чётко обозначены участники, их функции, а также последовательность «субакций», на которые данный процесс разбивается.

# **Структура БД**

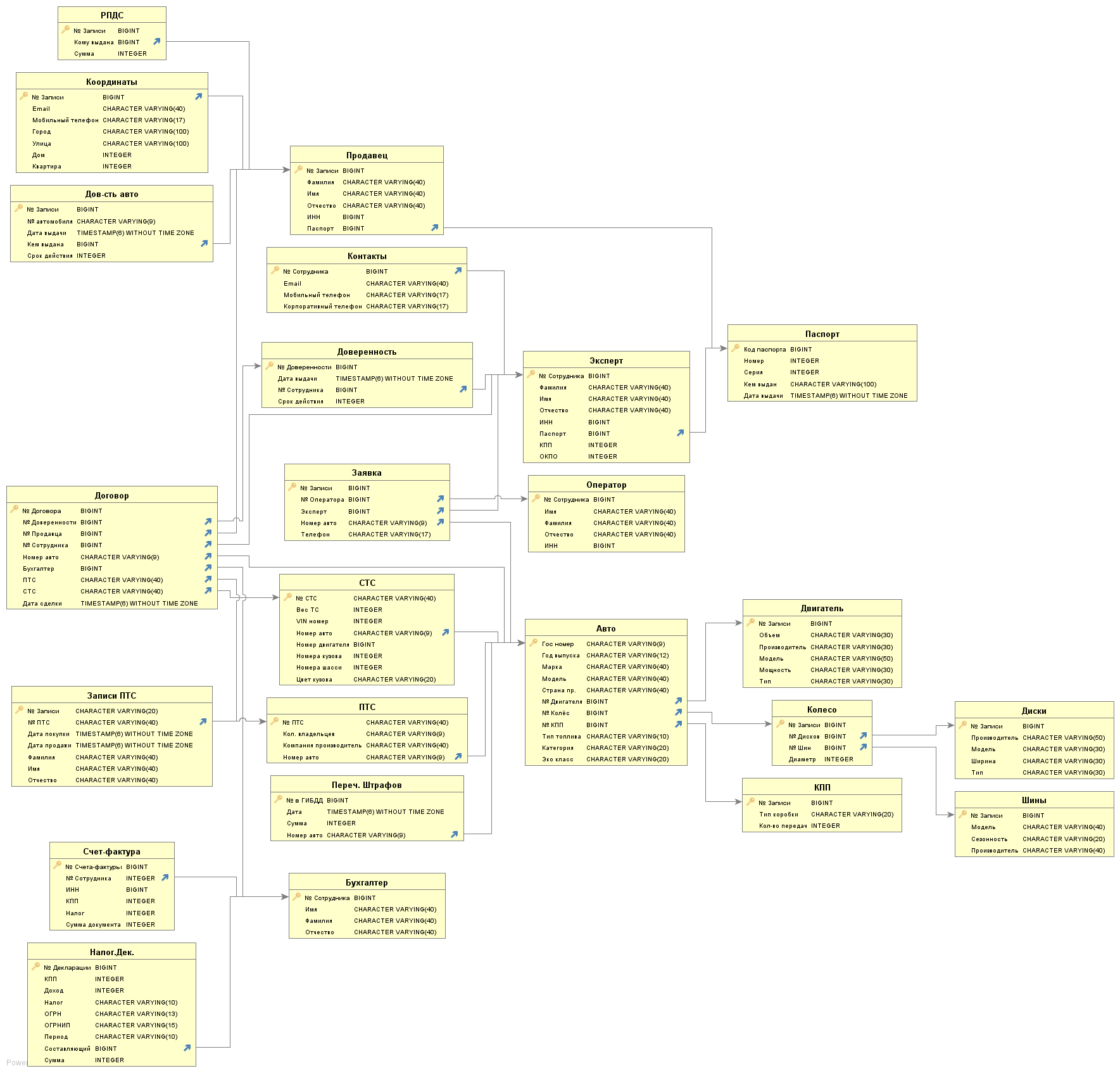


Рисунок 2 – модель «Сущность – Связь» иследуемой области ПрО.

На рисунке 2 изображена модель «Сущность-Связь» исследуемой ПрО.

Полагаю, рисунок нуждается в комментарии. Спецификация сущностей и их атрибутов представлена в формате «Сущность: первичный ключ, *альтернативный ключ*, атрибут».

Записи ПТС: № Записи [varchar 20], № ПТС [varchar 40], Дата покупки [timestamp], Дата продажи [timestamp], Фамилия [varchar 40] Отчество [varchar 40], Имя [varchar 40].

ПТС: № ПТС[integer], Кол. владельцев [varchar 9], Компания производитель [varchar 40], Номер авто [varchar 9].

Авто: Гос номер[varchar 9], Год выпуска [varchar 12], Марка [varchar 40], Модель[varchar 40], Страна пр.[varchar 40], № Двигателя [bigint], № Колёс [bigint], № КПП [bigint], Тип топлива [varchar 10], Категория [varchar 20], Эко класс [varchar 20].

Бухгалтер: № сотрудника [bigint], Имя [varchar 40], Фамилия[varchar 40], Отчество[varchar 40].

Двигатель: № Записи[bigint], Модель [varchar 50], Мощность [varchar 30], Объем [varchar 30], Производитель[varchar 30], Тип[varchar 30].

Диски: № Записи [bigint], Производитель [varchar 50], Модель [varchar 30], Ширина [varchar 30], Тип [varchar 30].

Доверенность:№ доверенности[bigint], Дата выдачи [timestamp], № сотрудника[bigint], Срок действия [integer].

Дов-сть авто (Доверенность на продажу автомобиля): :№ записи [bigint], № автомобиля[varchar 9], Дата выдачи [timestamp], Кем выдана[bigint], Срок действия [integer].

Договор: № Договора [bigint], № доверенности[bigint], № продавца [bigint], № сотрудника. [bigint], Бухгалтер [bigint], № автомобиля [varchar 9], Дата сделки[timestamp], ПТС. [varchar 40], СТС [varchar 40].

Заявка: № Записи [bigint], № оператора [bigint], Эксперт [bigint], Номер Авто[varchar 9], Телефон [varchar 17].

Колесо: № Записи [bigint], № дисков [bigint], № шин [bigint], Диаметр[integer].

Контакты: № сотрудника [bigint], Email [varchar 40], Мобильный телефон [varchar 17], Корпоративный телефон [varchar 17].

Коорд. (Координаты продавца): № Записи [bigint], Email [varchar 40], Мобильный телефон [varchar 17], Город [varchar 100], Улица [varchar 100], Дом [integer], Квартира [integer].

КПП: № Записи [bigint], Тип коробки[varchar 40], Кол-во передач [integer].

Налог.Дек.(Налоговая декларация): № декларации [bigint], КПП [integer], Доход [integer], Налог [varchar 10], ОГРН [varchar 13], ОГРНИП [varchar 15], Период[varchar 10], Составляющий[bigint], Сумма[integer].

Оператор: № сотрудника [bigint], Имя [varchar 40], Фамилия [varchar 40], Отчество [varchar 40], ИНН [bigint].

Паспорт: Код паспорта [bigint], Номер [integer], Серия [integer], Кем выдан [varchar 100], Дата выдачи [timestamp].

Переч. Штрафов (Перечень штрафов): № в ГИБДД[bigint], Дата[timestamp], Сумма[integer], Номер авто [varcahr 9].

Продавец: № Записи[bigint], Имя [varchar 40], Фамилия [varchar 40], Отчество [varchar 40], ИНН [bigint], Паспорт [bigint].

РПДС(Расписка о получении денежных средств): № Записи [bigint], Кому выдана [bigint], Сумма [integer],

СТС: № СТС [varchar 40], Вес ТС[integer], VIN номер[integer], Номер авто[varchar 9], Номер двигателя[bigint], Номера кузова[integer], Номера шасси[integer], Цвет кузова[varchar 20].

Счет-фактура: № счета-фактуры[bigint], № сотрудника[integer], ИНН[bigint], КПП[integer], Налог[integer], Сумма документа[integer].

Шины: № Записи [bigint] , Модель[varchar 40], Сезонность[varchar 20], Производитель [varchar 40].

Эксперт: № сотрудника[bigint], Имя[varchar 40], Фамилия[varchar 40], Отчество[varchar 40], ИНН[bigint], КПП[integer], ОКПО[integer], Паспорт [bigint].

**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/>

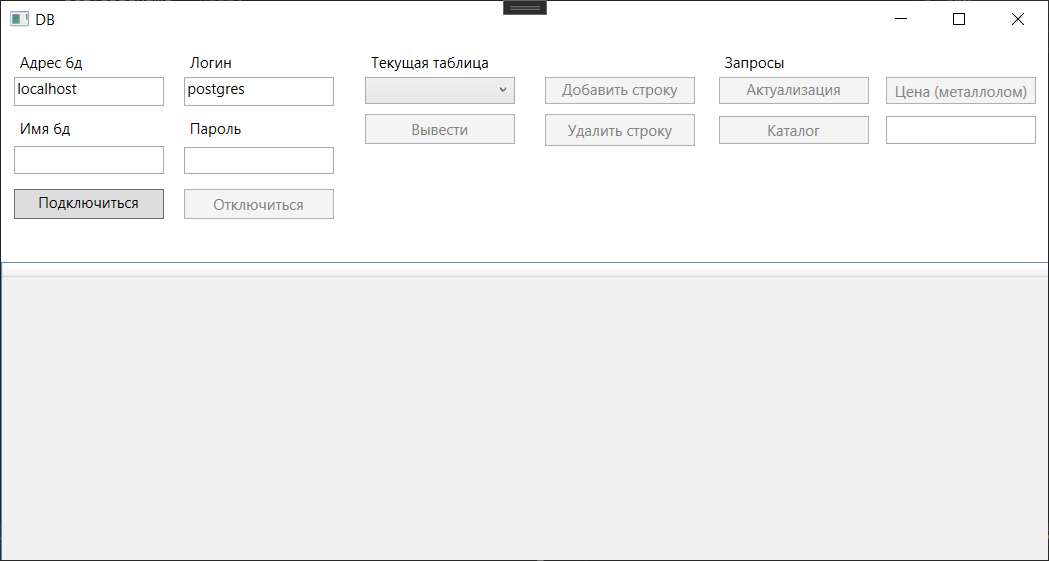
Закончив процесс установки и настройки PostgreSQL, нужно запустить pgAdmin, идущий с ним в комплекте, зайти под созданным ранее пользователем и воссоздать БД из файлов .sql, идущих с курсовой работой. В pgAdmin необходим вызвать контекстное меню для одной из баз данных (первоначальной или новой) и перейти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

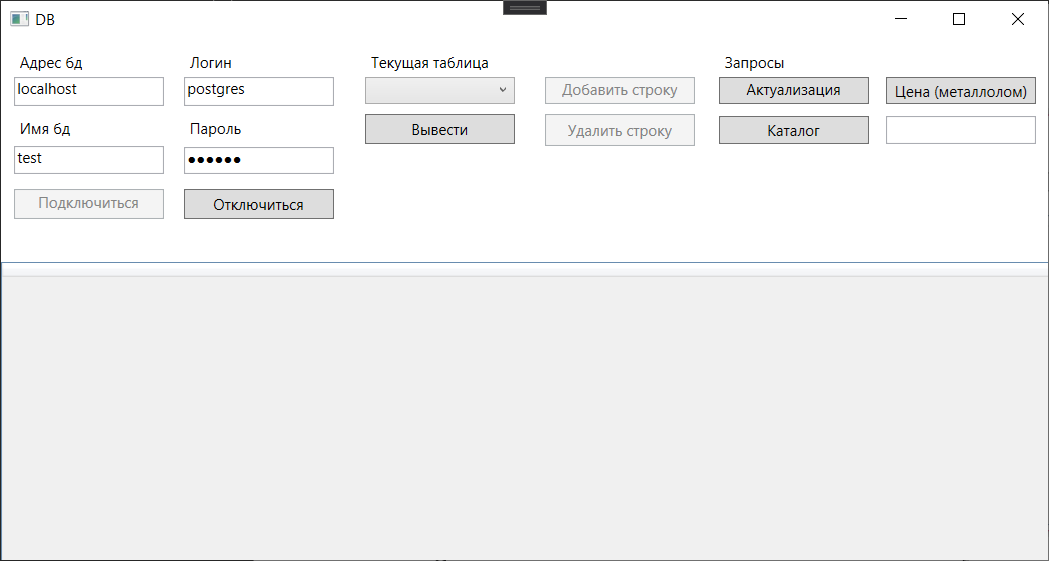
Открыв проект в Visual Studio, нажимаем на «Пуск» в верхней части экрана, приложение компилируется и запускается.

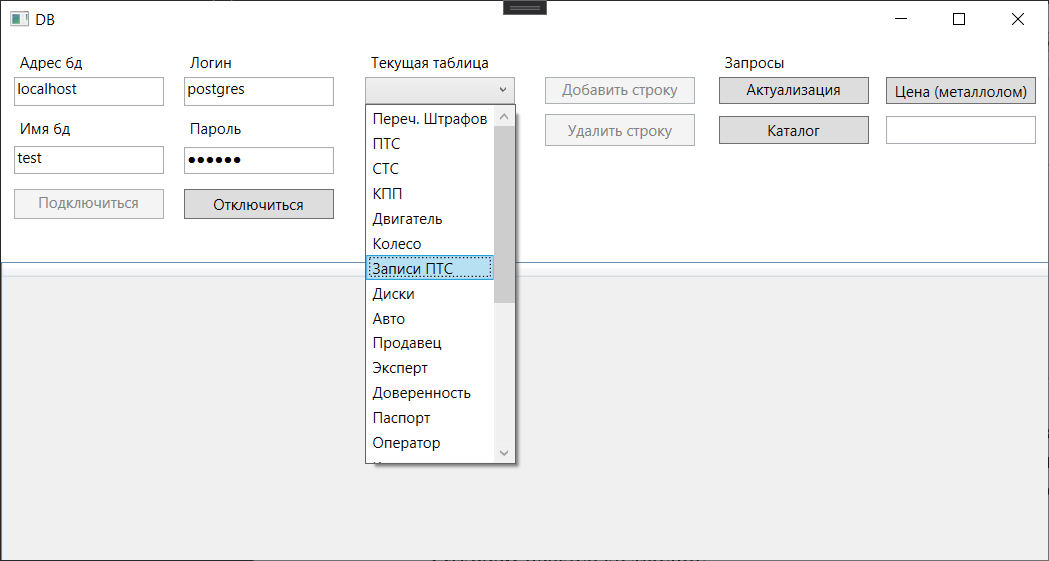
Среда для работы с БД готова.

Открыв приложение, можно увидеть следующий экран:

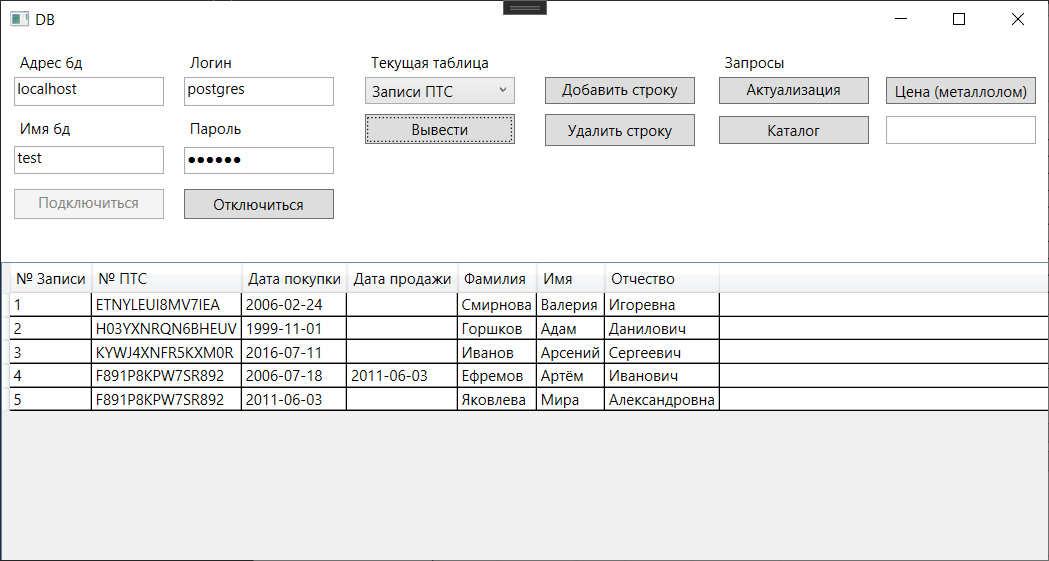
Это единственный и основной экран приложения. Для подключения к БД необходимо ввести данные для входа в соответствующие поля. Поле "Адрес бд" заполняется в соответствии с тем, где находится БД - localhost либо IP-адрес хоста. Поле "Имя" соответствует имени БД. Поля "Логин" и "Пароль" заполняются данными 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.

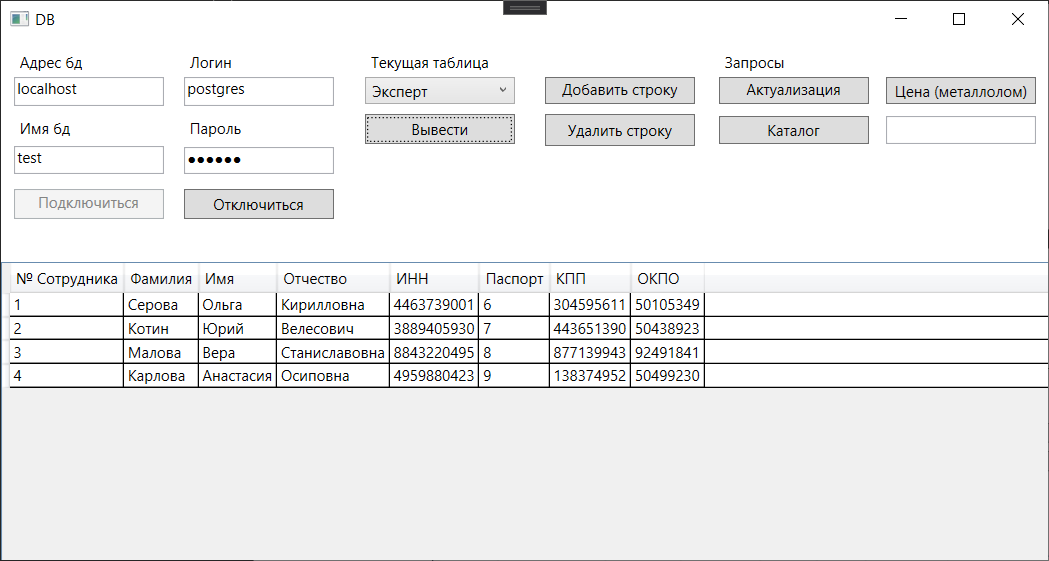
После заполнения этих полей, нужно нажать на кнопку подключения.

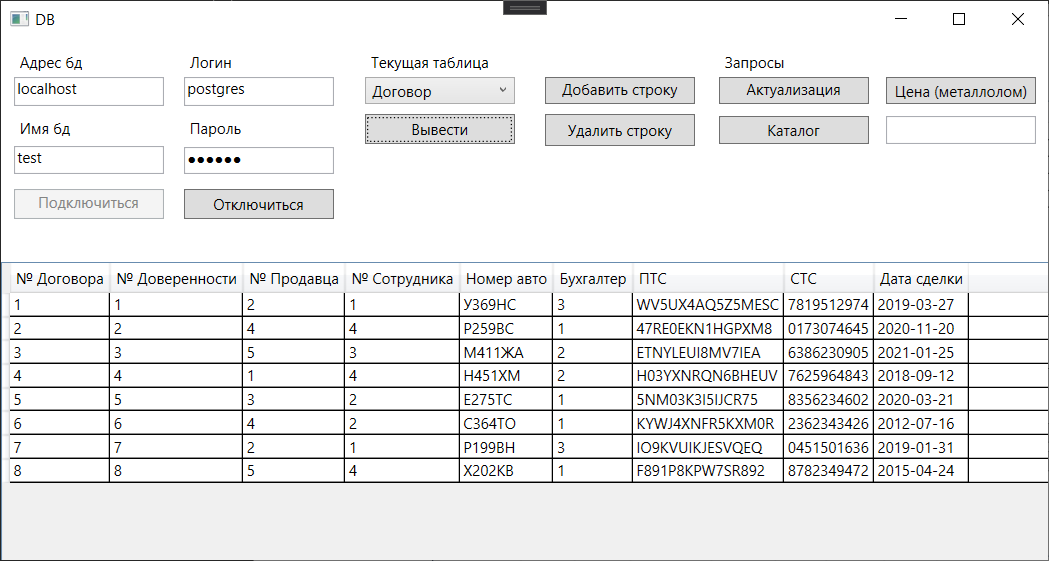


Активировались кнопки, в ComboBox загрузились все доступные для пользователя таблицы.

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**

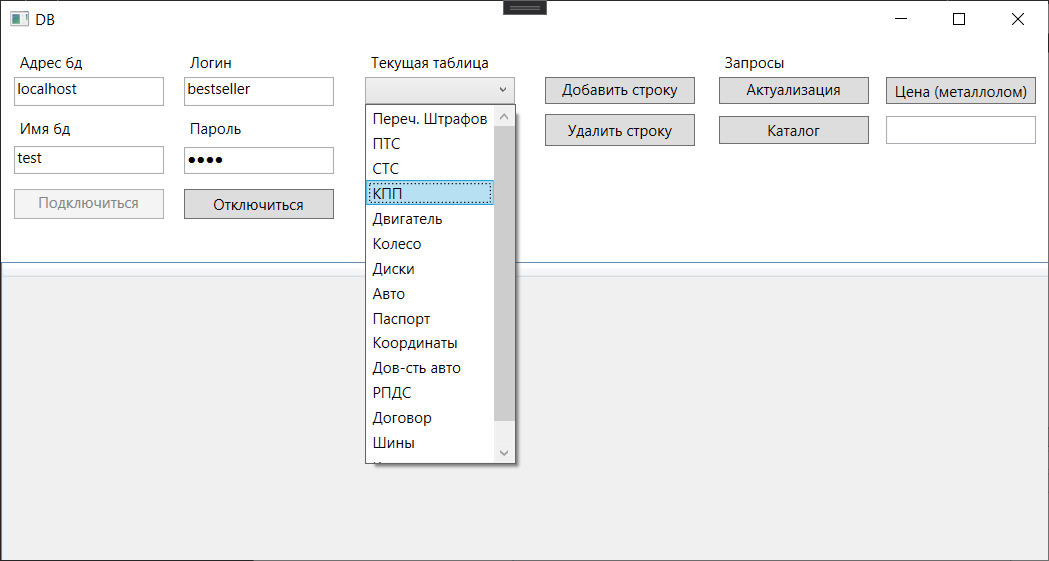
Для демонстрации результата вернёмся к задачам, поставленным перед выполнением курсовой работы и перечисленным в пункте "Введение".

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.

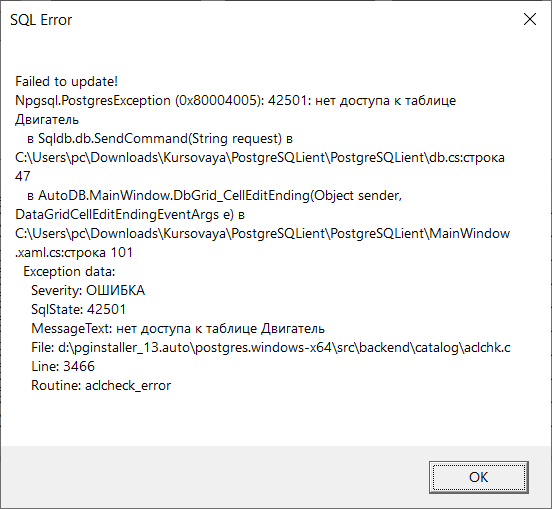
- Уже было продемонстрировано в разделе "Структура БД"

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;

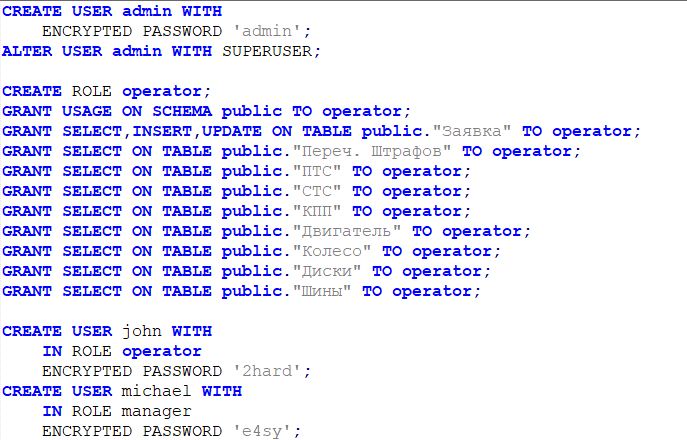
- Наличие многопользовательского доступа было продемонстрировано в пункте "Программный продукт", но не были напрямую продемонстрированы ограничения функционала для разных пользователей. Чтобы показать это разграничение, воспользуемся одним из дополнительных пользователей для входа - bestseller.



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.



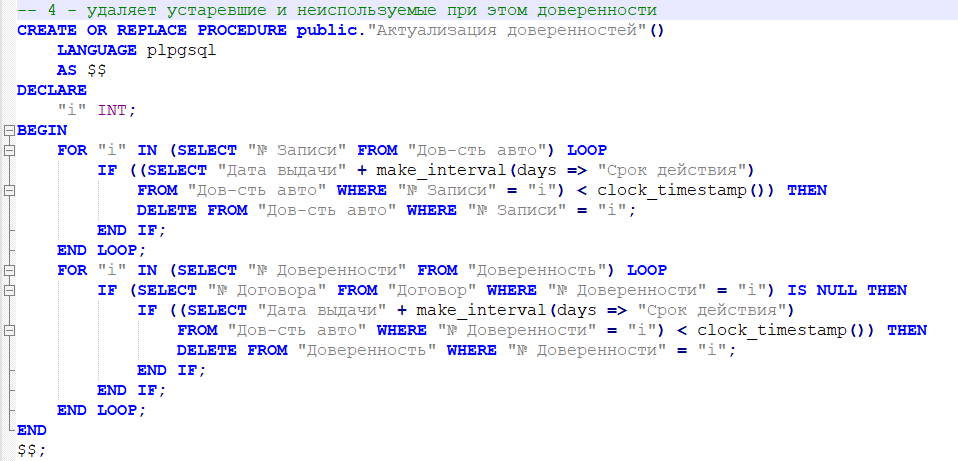


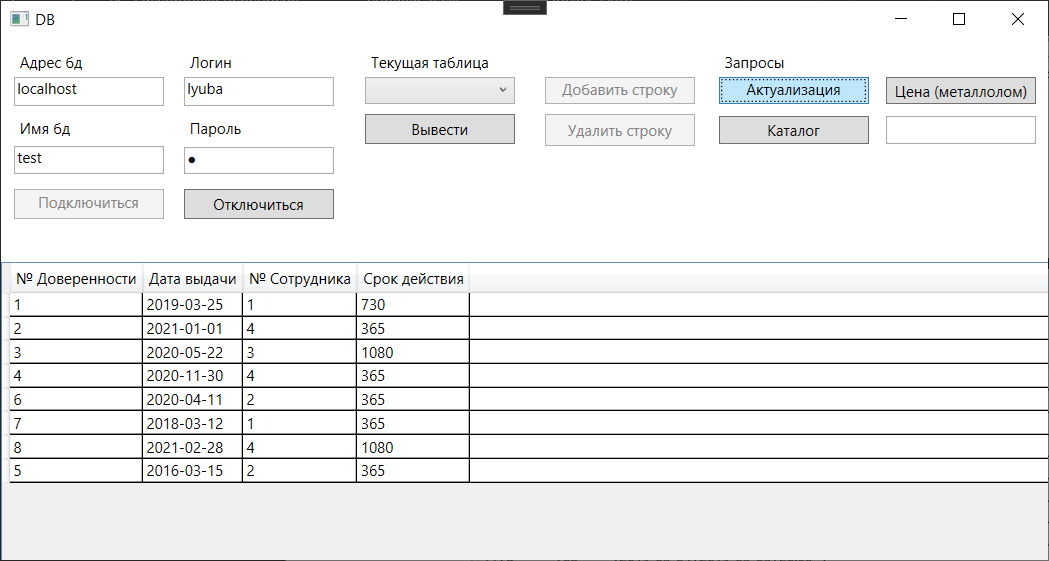
3. Пароли всех пользователей должны хранится в зашифрованном виде.

- 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.

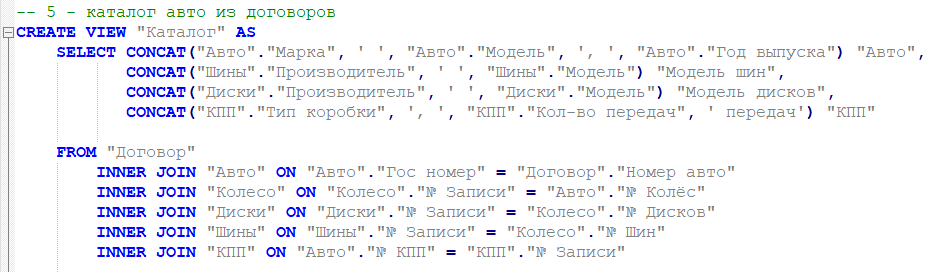
- Была реализована процедура “Актуализация доверенностей”

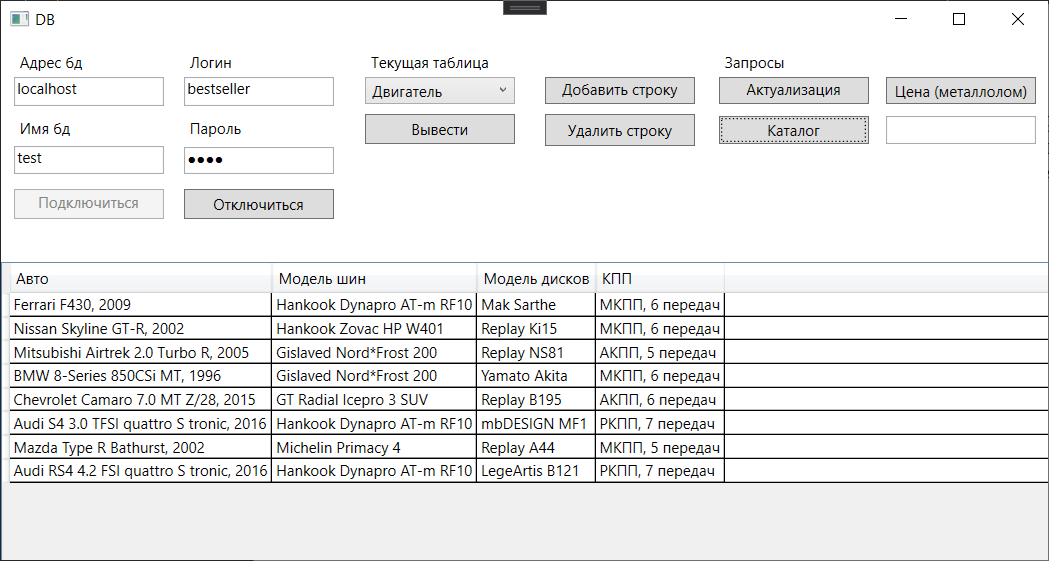


Данная процедура призвана помочь бухгалтерам и убирает доверенности, срок действия которых истёк и которые не фигурируют ни в каких документах.

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

- Было реализовано представление “Каталог”:

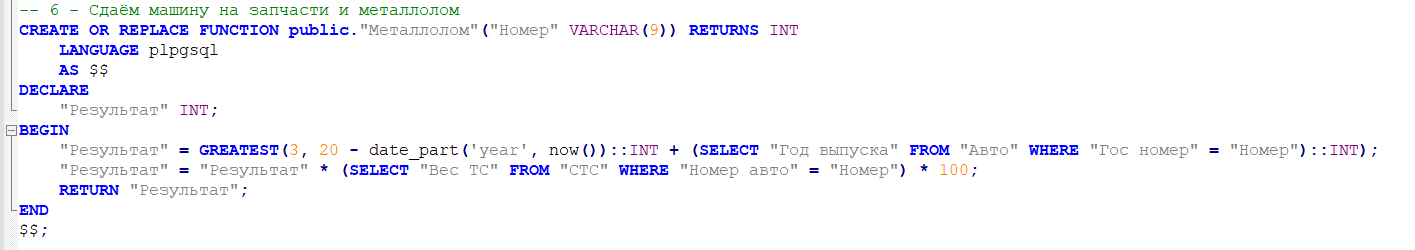




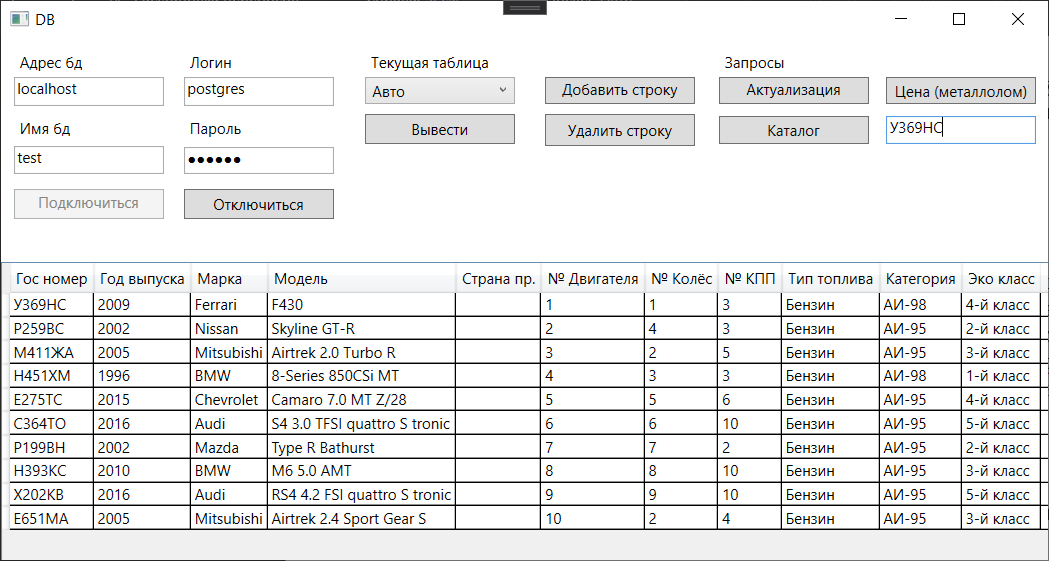
Данное представление отображает содержимое таблиц Авто, Колесо, Диски, Шины и КПП, объединяя их и приводя к удобочитаемый каталог. Доступно всегда (если у пользователя есть права).

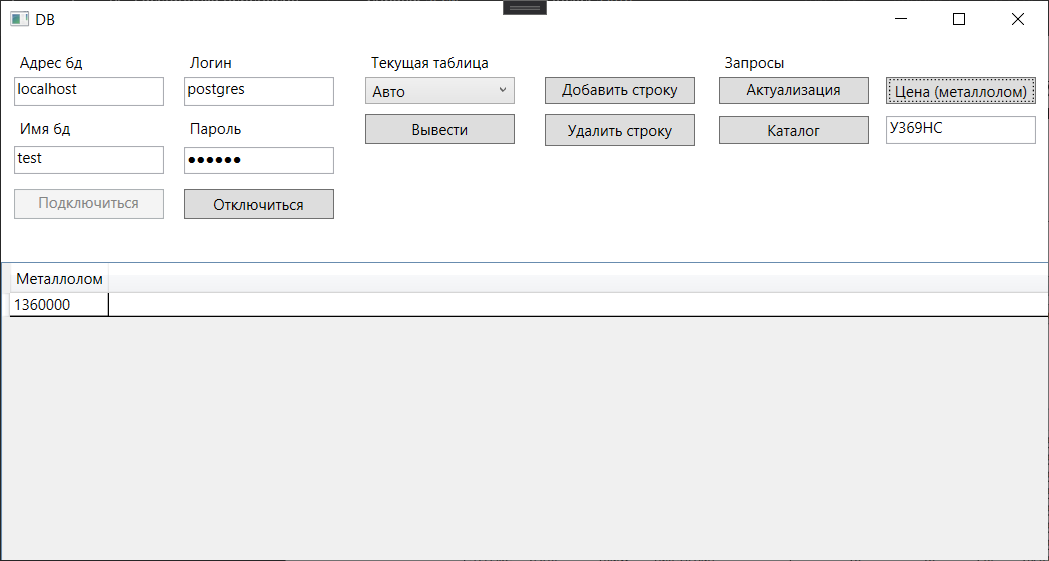
6. В базе данных должна присутствовать хотя б одна функция, которая коррелирует с темой курсовой

- 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.



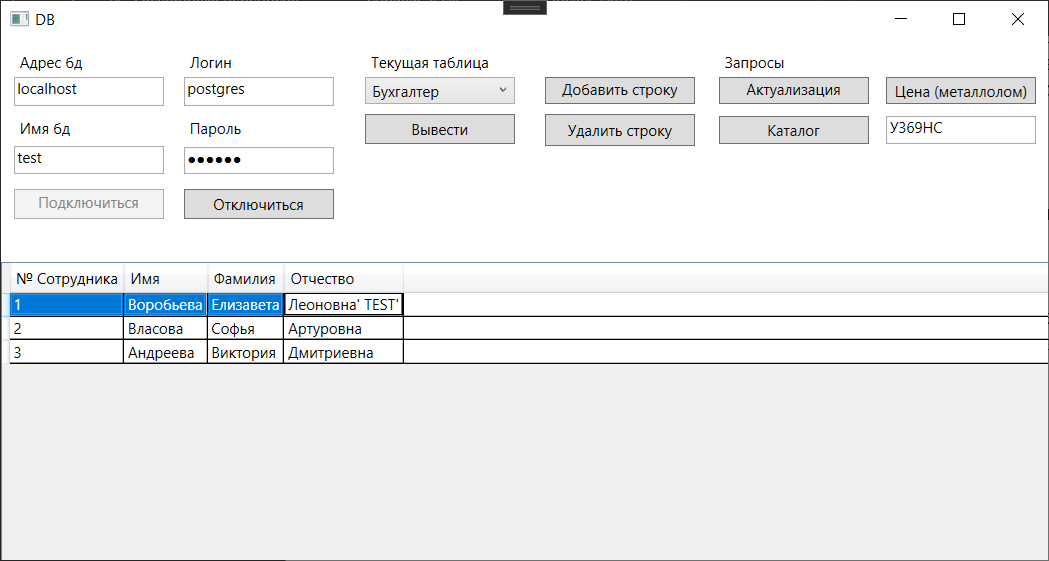


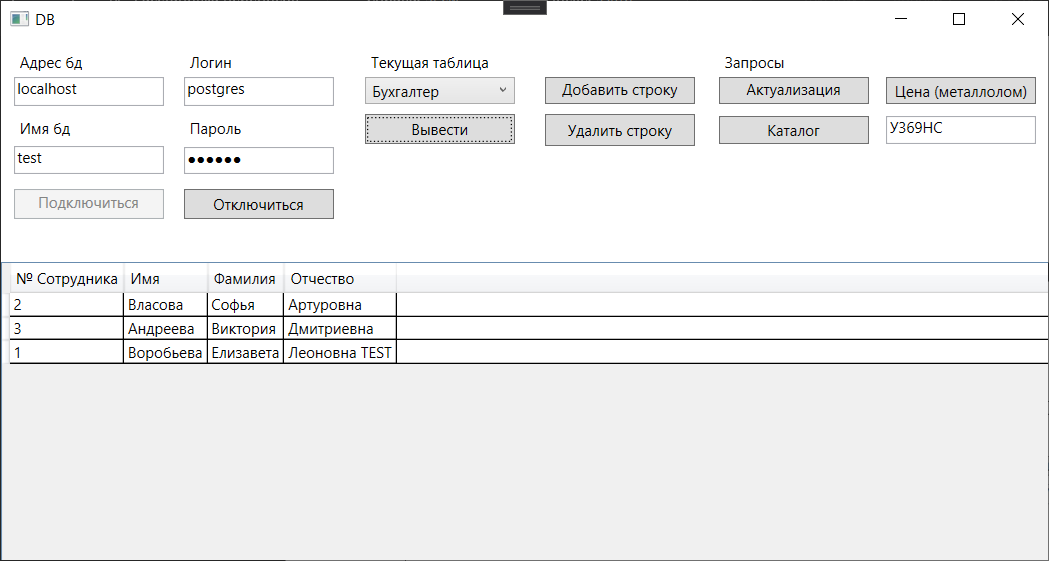
7. В базе данных должна присутствовать хотя б одна транзакция, которая коррелирует с темой курсовой

- 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.