

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 (да?)

# **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

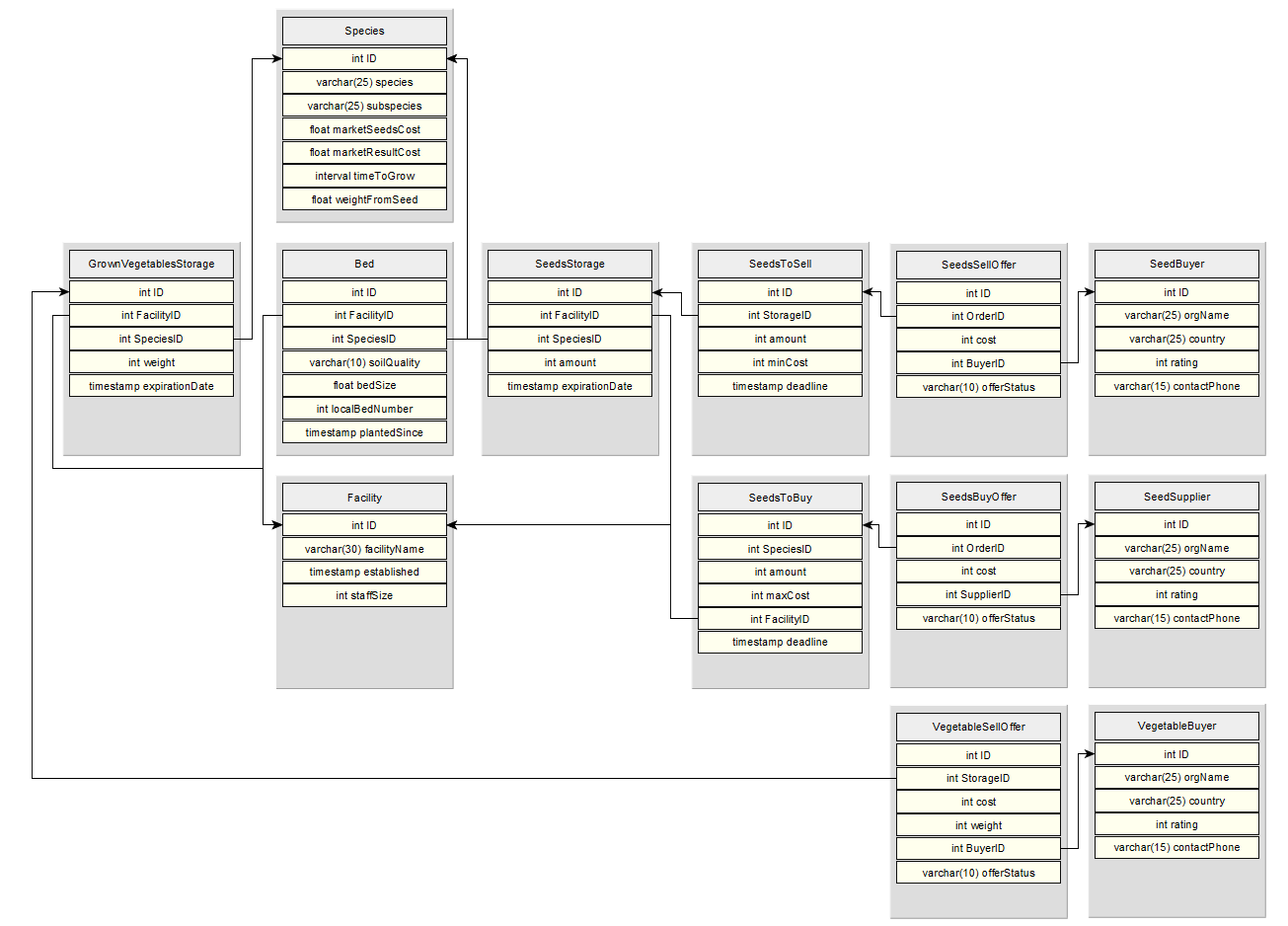
- Управление данными во внешней и оперативной памяти

- 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 позиций покупки-продажи семян же существуют отдельные таблицы SeedsToBuy и SeedsToSell с информацией о заказах на покупку и продажу соответственно.

Есть три отдельных категории компаний, с которыми ведётся торговля - поставщики семян SeedSupplier, покупатели семян SeedBuyer и покупатели овощей VegetableBuyer. Торговля специализированная, поэтому такое разделение и произошло - так, среди покупателей овощей в основном всегда будут продуктовые магазины, а среди покупателей семян такие же овощебазы и хозяйства.

Соответственно, между этими компаниями и позициями овощебазы на продажу/покупку есть промежуточные таблицы для хранения предложений о сделке от других фирм. Это SeedsSellOffer - предложение продать семена, SeedsBuyOffer - предложение купить семена и VegetableSellOffer - предложение продать овощи.

**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/>

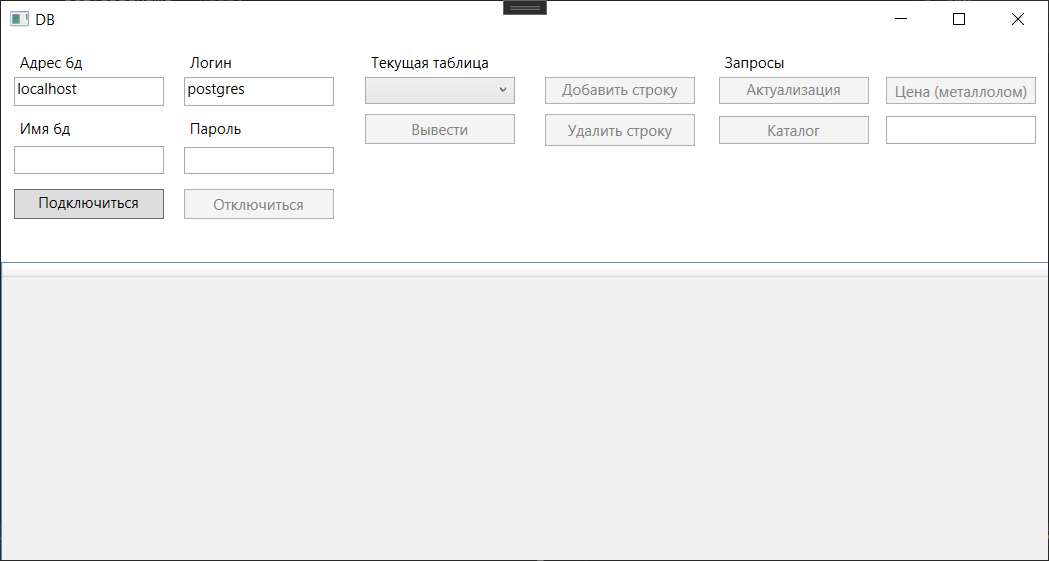
Закончив процесс установки и настройки PostgreSQL, нужно запустить pgAdmin, идущий с ним в комплекте, зайти под созданным ранее пользователем и воссоздать БД из файлов .sql, идущих с курсовой работой. В pgAdmin необходим вызвать контекстное меню для одной из баз данных (первоначальной или новой) и перейти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

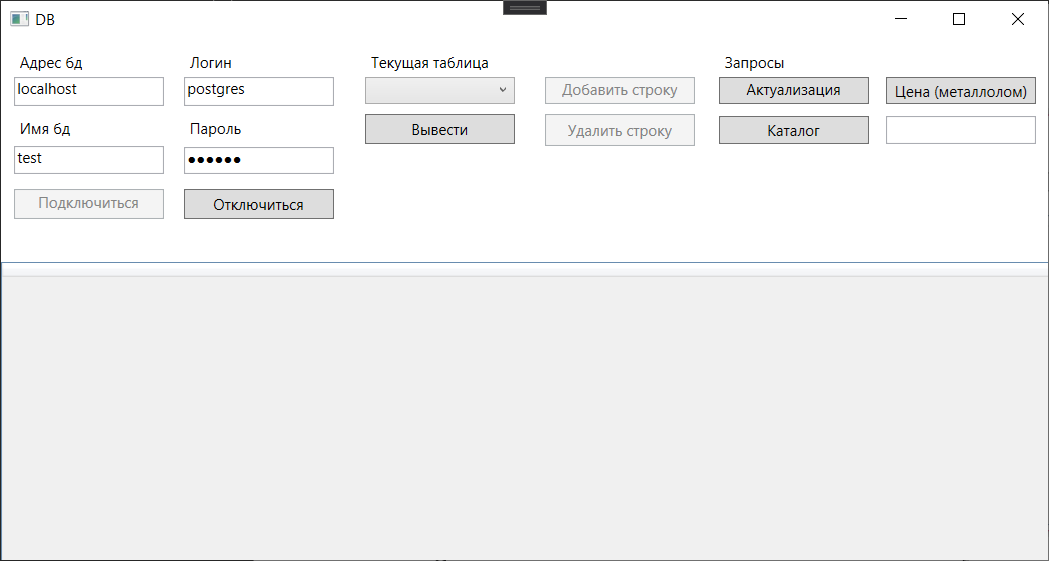
Открыв проект в Visual Studio, нажимаем на «Пуск» в верхней части экрана, приложение компилируется и запускается.

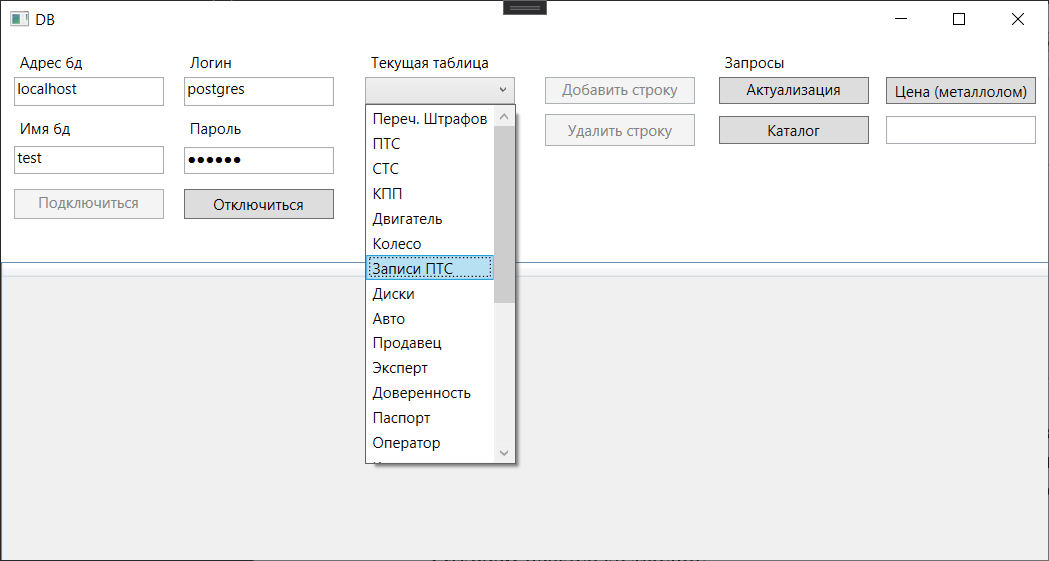
Среда для работы с БД готова.

Открыв приложение, можно увидеть следующий экран:

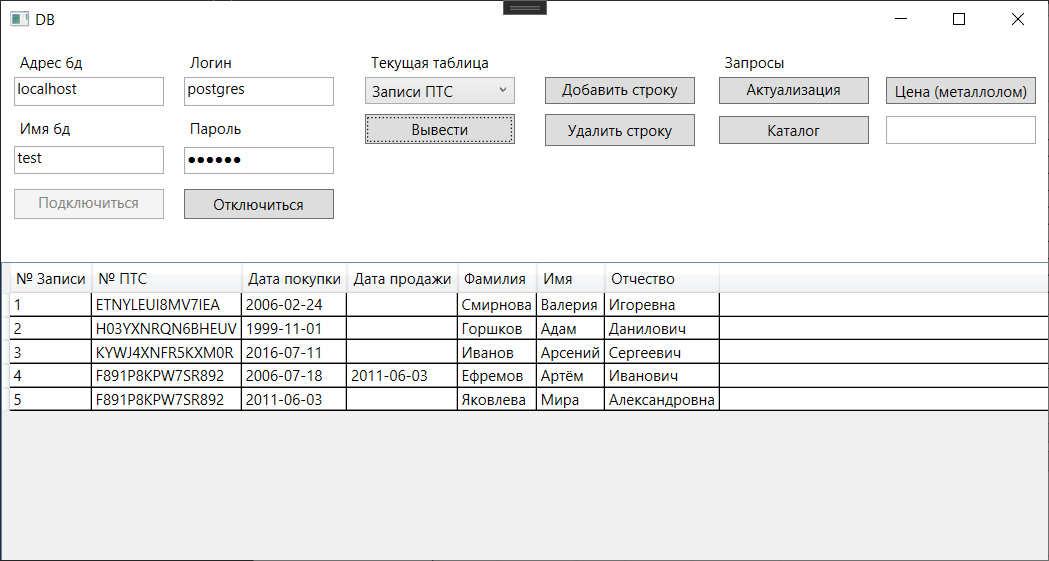
Это единственный и основной экран приложения. Для подключения к БД необходимо ввести данные для входа в соответствующие поля. Поле "Адрес бд" заполняется в соответствии с тем, где находится БД - localhost либо IP-адрес хоста. Поле "Имя" соответствует имени БД. Поля "Логин" и "Пароль" заполняются данными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

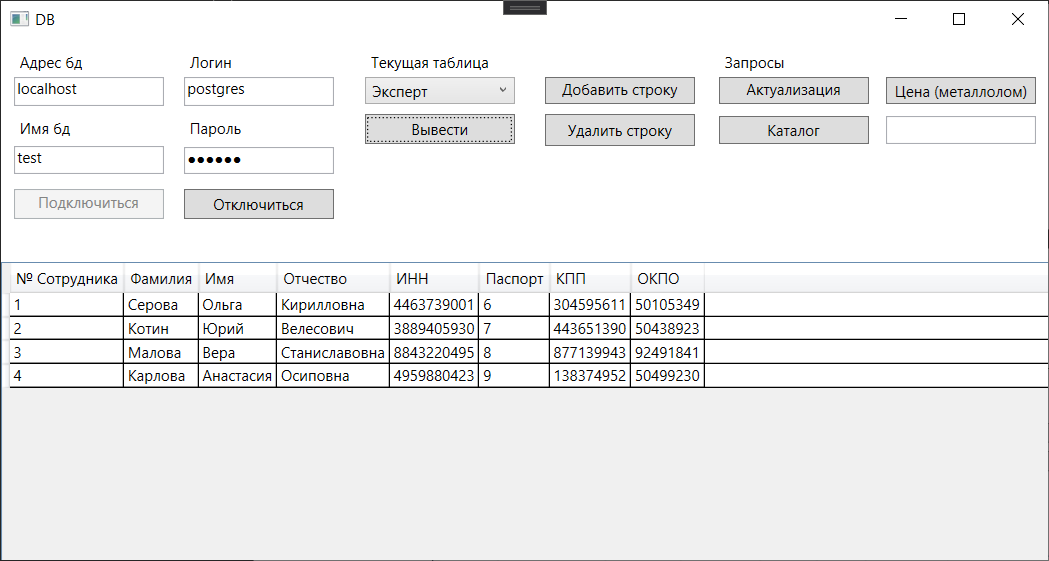
После заполнения этих полей, нужно нажать на кнопку подключения.

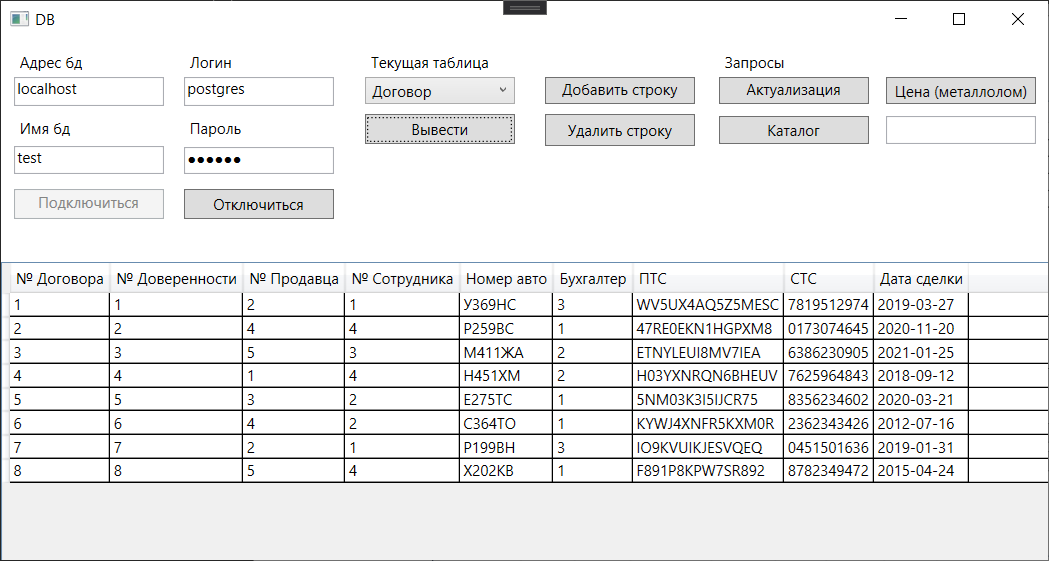


Активировались кнопки, в ComboBox загрузились все доступные для пользователя таблицы.

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.

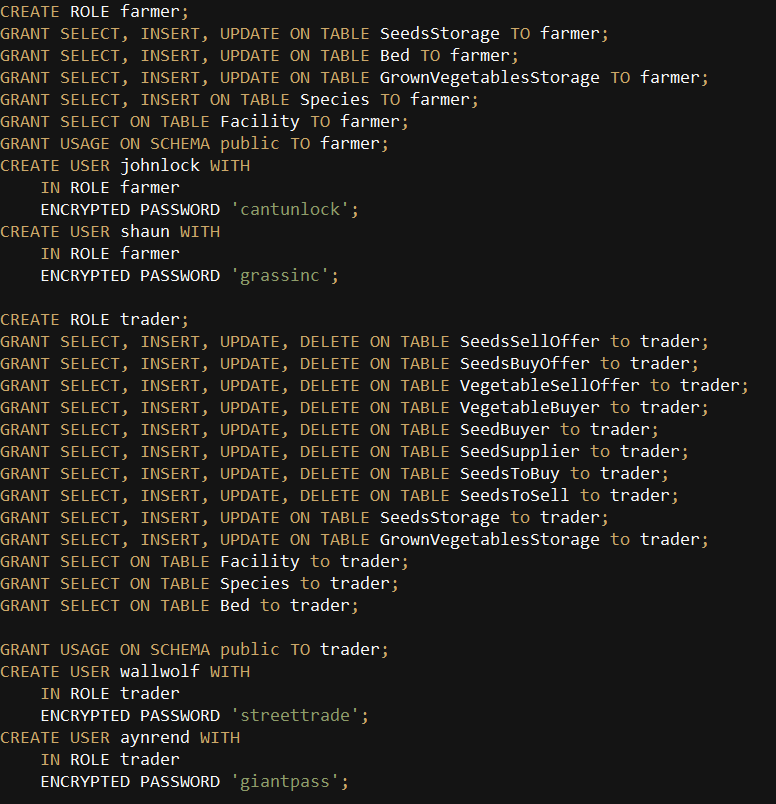
1. Содержит в себе не менее заполненных 10 таблиц, в каждой таблице не менее 3-8 кортежей (записей). Все записи должны содержать корректные данные, соответствующие предметной области курсовой работы.

База данных содержит 13 таблиц - Facility, SeedsStorage, Bed, GrownVegetablesStorage, Species, SeedsToSell, SeedsToBuy, SeedsSellOffer, SeedsBuyOffer, VegetableSellOffer, SeedBuyer, SeedSupplier, VegetableBuyer. Все они содержат как минимум 3 кортежа (в основном больше).

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.

- 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.



На данном скриншоте видно, что в базе данных присутствуют две роли с разным уровнем доступа и по два пользователя на каждую роль. Также, у каждого пользователя перед паролем есть ключевое слово 'ENCRYPTED', означающее, что пароль должен быть сохранён 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.

а

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

а

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

а

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

а

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

а

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.

а

**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.