# Лабараторная работа №2

**Цель работы**: создать приложение в Qt для работы с базой данных. Реализовать валидатор и написать к нему Unitest.

**Задача:** Цветочная оранжерея выращивает различные виды цветов и продает на заказ составленные из них композиции. Каждая композиция имеет свое название и может состоять как из цветов одного вида, так и из цветов разного вида**.**

Дирекция оранжереи владеет информацией:  
о цветах: название цветка, сорт, стоимость, одного цветка;  
о композициях: название композиции, название входящего в композицию цветка, количество единиц;  
о выполнении заказов: дата принятия заказа, название композиции, количество единиц, покупатель;  
о покупателе: логин, пароль.

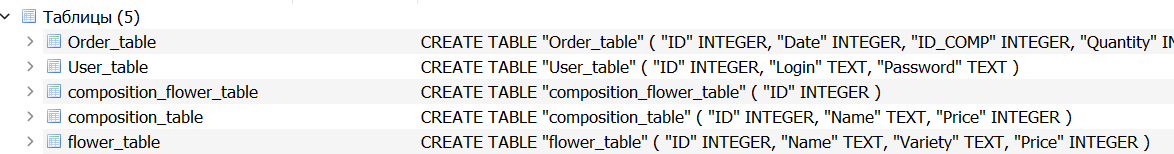
Необходимо выполнить:

1. Создать таблицы ДБ  
 2. Обеспечить с помощью операторов Select, Insert, Update, Delete обновление и выборку информации в указанных таблицах  
 3. Создать оконное приложение, которое позволяет манипулировать данными с использованием вышеупомянутых операторов.  
 4. Написать тесты для валидатора.

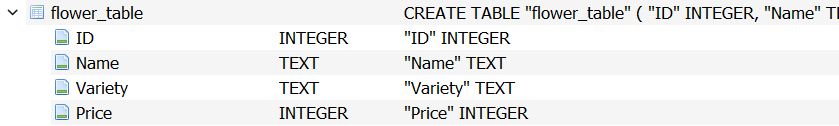
Часть 1. Разработка необходимых таблиц, связи между собой и наполнения каждой таблицы.  
flower\_table (название text, сорт text, cost integer, id PK integer )  
composition\_table (id integer, name text, cost integer)  
composition\_flower\_table (id integer, id\_flower integer, id\_comp integer, count integer)  
Order\_table (id integer, date\_v text, id\_comp integer, count integer, user\_id integer, summa integer)  
User\_table (id integer, login text, password text)

Часть 2. Создание базы данных с помощью DB Browser

В DB Browser создал 5 таблиц:



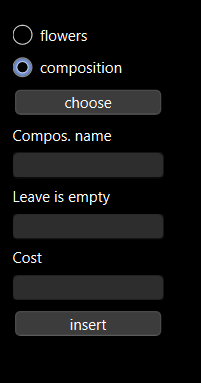
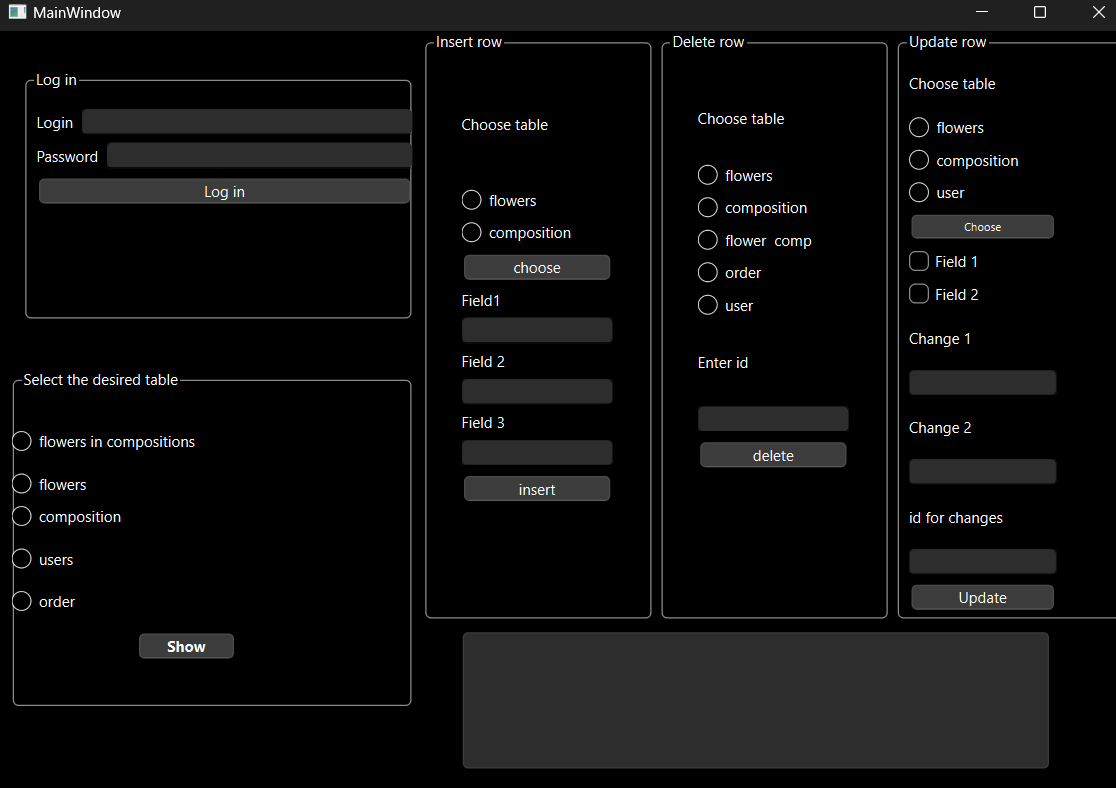
Далее таблицы были заполнены вручную. Пример:  
  
После чего, сохранил файл с расширением db.



Часть 3. Реализация работы с базой данных через QT. Создание интерфейса для простейших операций над базами данных.

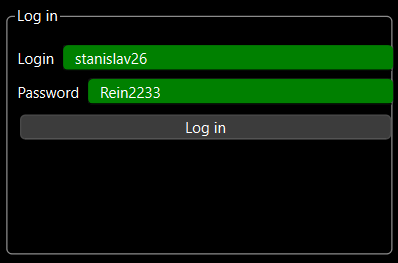
Основной целью было создание удобного пользовательского интерфейса, позволяющего выполнять операции, аналогичные тем, которые доступны в DB Browser. Для обеспечения лучшей структурированности информации, мы использовали такие элементы интерфейса QT, как QSqlTableModel для отображения таблиц в актуальном состоянии для выбора, GroupBox для группировки элементов, решающих общую задачу, а также TexyBrowser и Label для вывода инструкций пользователю.

Вот финальный вид приложения:  
  
  
Пользователь сам может выбирать с какой таблицей ему работать:



Часть 4. Написание валидатора

Для того чтобы создать нового пользователь, нужно было придумать логин и пароль, которые соответствуют стандартам(они прописываются при указании неверного логина или пароля).



Часть 5. Написание теста к валидатору

Тестирование проводилось с помощью Visual Studio, где был выбран шаблон google test.   
Я подключал файл .h для дальнейшего подключения к тестам. Т.к. класс писался в Qt, он содержал библиотеки Qt.