**МИНИСТЕРСТВО ВЫСШЕГО И СРЕДНЕГО СПЕЦИАЛЬНОГО ОБРАЗОВАНИЯ РЕСПУБЛИКИ УЗБЕКИСТАН**

**САМАРКАНДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ФАКУЛЬТЕТ ЦИФРОВЫХ ТЕХНОЛОГИЙ**

Курсовая работа

по дисциплине

“Технология баз данных”

Тема : “Проектирование БД для Книжного магазина”

**Студентка группы 304: Мирзоева А.**

**Преподаватель: Туракулов И.**

**Самарканд 2022**

**План:**1. Термины и определения используемые в данной курсовой

2.Анализ предметной области

3. Код программы

4. Работа программы

5. Заключение

6. Использованная литература

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

В данной курсовой работе применяются следующие термины и определения:

База данных - это поименованная совокупность данных, отражающая состояние объектов и их отношения в рассматриваемой предметной области.

Система управления базой данных (СУБД) - это совокупность языковых и программных средств, предназначенных для создания, ведения и совместного применения

Таблица - некоторая регулярная структура, состоящая из конечного набора однотипных записей.

База данных - это организованная структура, предназначенная для хранения информации. В современных базах данных хранятся не только данные, но и информация.

Это утверждение легко пояснить, если, например, рассмотреть базу данных крупного банка. В ней есть все необходимые сведения о клиентах, об их адресах, кредитной истории, состояние расчетных счетов, финансовых операциях и т.д. Доступ к этой базе данных имеется у достаточно большого количества сотрудников банка, но среди них вряд ли найдется такое лицо, которое имеет доступ ко всей базе полностью и при этом способно единолично вносить в нее произвольные изменения. Кроме данных, база содержит методы и средства, позволяющие каждому из сотрудников оперировать только с теми данными, которые входят в его компетенцию. В результате взаимодействия данных, содержащихся в базе, с методами, доступными конкретным сотрудникам, образуется информация, которую они потребляют и на основании которой в пределах собственной компетенции производят ввод и редактирование данных.

С понятием базы данных тесно связано понятие системы управления базой данных. Это комплекс программных средств, предназначенных для создания структуры новой базы, наполнение ее содержимым, редактирование содержимого и визуализации информации. Под визуализацией информации базы понимается отбор отображаемых данных в соответствии с заданным критерием, их упорядочение, оформление и последующая выдача на устройства вывода или передачи по каналам связи.

В мире существует множество систем управления базами данных. Несмотря на то что они могут по-разному работать с разными объектами и предоставляют пользователю различные функции и средства, большинство СУБД опираются на единый устоявшийся комплекс основных понятий. Это дает нам возможность рассмотреть одну систему и обобщить ее понятия, приемы и методы на весь класс СУБД. В качестве такого учебного объекта мы выберем SQLite.

**Анализ предметной области**

Основным назначением информационных систем является оперативное обеспечение пользователя информацией о внешнем мире путем реализации вопросно-ответного отношения. Вопросно-ответные отношения, получая интерпретацию во внешнем мире (мире вне информационной системы), позволяют выделить для информационной системы определенный его фрагмент - предметную область, - который будет воплощен в автоматизированной информационной системе. Информация о внешнем мире представляется в информационной системе в форме данных. Это ограничивает возможности смысловой интерпретации информации и конкретизирует семантику ее представления в информационной системе. Совокупность этих выделенных для информационных систем данных, связей между ними и операций над ними образует информационную и функциональную модели предметной области, описывающие ее состояние с определенной точностью.

Важно понимать, что информационная и функциональная модели предметной области создаются на этапе анализа требований к базе данных и не содержат предположений о технологии реализации базы данных. Они строятся независимо от выбираемой модели данных (сетевой, иерархической, реляционной, объектно-ориентированной, многомерной и т.д.), поддерживаемой СУБД, модели вычислений, программно-аппаратной платформы для базы данных. Информационная и функциональная модели предметной области являются входными данными для процесса проектирования базы данных. Поэтому проектировщик должен уметь правильно интерпретировать их в ходе решения своих проектных задач.

Понятие предметной области базы данных является одним из базовых понятий информатики и не имеет точного определения. Его использование в контексте ИС предполагает существование устойчивой во времени соотнесенности между именами, понятиями и определенными реалиями внешнего мира, не зависящей от самой ИС и ее круга пользователей. Таким образом, введение в рассмотрение понятия предметной области базы данных ограничивает и делает обозримым пространство информационного поиска в ИС и позволяет выполнять запросы за конечное время.

Совокупность реалий (объектов) внешнего мира - объектов, о которых можно задавать вопросы, - образует объектное ядро предметной области, которое имеет онтологический статус. Нельзя получить в ИС ответ на вопрос о том, что ей неизвестно. Термин объект является первичным, неопределяемым понятием. Синонимами термина "объект" являются "реалия, сущность, вещь". Однако термин сущность понимается нами несколько уже, как компонент модели предметной области, т.е. как уже выделенный на концептуальном уровне объект для базы данных. Таким образом, выделяемые в предметной области объекты превращаются аналитиками (а не проектировщиками базы данных) в сущности. Сущность предметной области является результатом абстрагирования реального объекта путем выделения и фиксации набора его свойств. Сущность является результатом абстрагирования реального объекта, т.е. в нашем контексте имеет гносеологический статус.

В данном курсовом проекте в качестве предметной области рассматривается книжный магазин. Наша база данных решает следующие задачи: учёт товара, выдача данных о поставщиках и поставляемых ими товарах, наличие определенного вида книг.

Реализует запросы упорядочения по полям: автор, название книги, поиск админа. Осуществляет поиск сведений какого-либо товара. Производит подсчет стоимости и количества оставшегося в магазине товара, а также выдает отчет об отсутствующих товарах.

Предметной областью называется фрагмент реальности, который описывается или моделируется с помощью БД и ее приложений. В предметной области выделяются информационные объекты - идентифицируемые объекты реального мира, процессы, системы, понятия и т.д., сведения о которых хранятся в БД.

В данной курсовой работе разработана база данных Книжного магазина для автоматизации и уменьшения времени при работе в книжном магазине.

Таблица «Books» состоит из из следующих полей: - автор(Author); - название книги(Book name); -год издания(Year); -стоимость(Cost); количество книг(Amount).

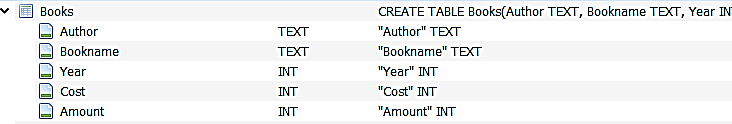
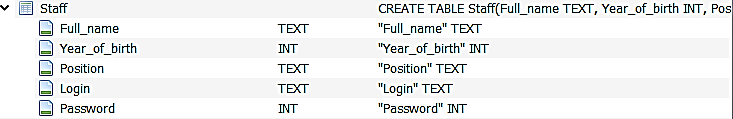


Таблица «Staff» состоит из из следующих полей: -полное имя(Full name), -дата рождения(Year of birth), -должность(Position), -логин(Login), -пароль(Password).



3. Код программы

Рассмотрим содержание файла mainwindow.cpp

#include "mainwindow.h"

#include "ui\_mainwindow.h"

#include <QDebug>

#include <QPixmap>

MainWindow::MainWindow(QWidget \*parent) :

QMainWindow(parent),

ui(new Ui::MainWindow)

{

ui->setupUi(this);

knigibase=QSqlDatabase::addDatabase("QSQLITE");

knigibase.setDatabaseName("./knigibase.db");

if(knigibase.open())

{ qDebug("open");}

else

{qDebug("no open");}

query = new QSqlQuery(knigibase);

query->exec("CREATE TABLE Books(Author TEXT, Bookname TEXT,Genre TEXT, Year INT, Cost INT, PageCount INT, Amount INT );");

model= new QSqlTableModel(this,knigibase);

model->*setTable*("Books");

model->*select*();

ui->tableView->*setModel*(model);

ui->tableView->horizontalHeader()->setSectionResizeMode(0, QHeaderView::Stretch);

ui->tableView->horizontalHeader()->setSectionResizeMode(1, QHeaderView::Stretch);

ui->tableView->horizontalHeader()->setSectionResizeMode(2, QHeaderView::Stretch);

ui->tableView->horizontalHeader()->setSectionResizeMode(3, QHeaderView::Stretch);

ui->tableView->horizontalHeader()->setSectionResizeMode(4, QHeaderView::Stretch);

query\_2 = new QSqlQuery(knigibase);

query\_2->exec("CREATE TABLE Staff(Full\_name TEXT, Year\_of\_birth INT, Position TEXT, Login TEXT, Password INT );");

model\_2= new QSqlTableModel(this,knigibase);

model\_2->*setTable*("Staff");

model\_2->*select*();

ui->tableView\_2->*setModel*(model\_2);

ui->tableView\_2->horizontalHeader()->setSectionResizeMode(0, QHeaderView::Stretch);

ui->tableView\_2->horizontalHeader()->setSectionResizeMode(1, QHeaderView::Stretch);

ui->tableView\_2->horizontalHeader()->setSectionResizeMode(2, QHeaderView::Stretch);

ui->tableView\_2->horizontalHeader()->setSectionResizeMode(3, QHeaderView::Stretch);

ui->tableView\_2->horizontalHeader()->setSectionResizeMode(4, QHeaderView::Stretch);

ui->stackedWidget->setCurrentIndex(0);

QPixmap myPixmap(":/resour/1635141742\_1.jpg");

ui->label\_3->setPixmap(myPixmap);

QPixmap myPixmap\_2(":/resour/410-640x400.jpg");

ui->label\_7->setPixmap(myPixmap\_2);

ui->lineEdit\_7->hide();

ui->pushButton\_20->hide();

ui->listWidget->clear();

ui->listWidget\_2->clear();

}

MainWindow::~*MainWindow*()

{

delete ui;

}

void MainWindow::on\_pushButton\_clicked()

{

ui->stackedWidget->setCurrentIndex(2);

}

void MainWindow::on\_pushButton\_2\_clicked()

{

ui->stackedWidget->setCurrentIndex(1);

}

void MainWindow::on\_pushButton\_4\_clicked()

{

ui->stackedWidget->setCurrentIndex(0);

}

void MainWindow::on\_pushButton\_3\_clicked()

{

QString login=ui->lineEdit->text(),password=ui->lineEdit\_2->text();

if(!knigibase.isOpen()){

QMessageBox::about(this,"inf","inco");

return;}

QSqlQuery sq;

if(sq.exec("select \* from Staff where Login='"+login +"'and Password='"+ password+"'"))

{

int c=0;

if(sq.next())

c++;

if(c==1){

ui->stackedWidget->setCurrentIndex(3);

}

else

QMessageBox::warning(this,"Ошибка","Неверный логин или пароль");}

}

void MainWindow::on\_pushButton\_6\_clicked()

{

ui->stackedWidget->setCurrentIndex(0);

}

void MainWindow::on\_pushButton\_8\_clicked()

{

model->insertRow(model->*rowCount*());

}

void MainWindow::on\_pushButton\_9\_clicked()

{

model->removeRow(ui->tableView->currentIndex().row());

}

void MainWindow::on\_pushButton\_10\_clicked()

{

ui->stackedWidget->setCurrentIndex(4);

}

void MainWindow::on\_pushButton\_15\_clicked()

{

model\_2->insertRow(model\_2->*rowCount*());

}

void MainWindow::on\_pushButton\_16\_clicked()

{

model\_2->removeRow(ui->tableView\_2->currentIndex().row());

}

void MainWindow::on\_pushButton\_18\_clicked()

{

ui->stackedWidget->setCurrentIndex(3);

}

void MainWindow::on\_pushButton\_12\_clicked()

{

ui->stackedWidget->setCurrentIndex(0);

}

void MainWindow::on\_pushButton\_5\_clicked()

{

QString author=ui->lineEdit\_3->text();

QString bookn=ui->lineEdit\_4->text();

QSqlQuery quer;

QSqlQuery sq;

if(sq.exec("select \* from Books where Author='"+author +"'and Bookname='"+ bookn+"'"));

{

int c=0;

if(sq.next())

c++;

if(c==1){

QMessageBox::about(this,"Поиск","Данная книга есть в наличии. Цена книги- "+ sq.value(3).toString()+" сумов");

// ui->pushButton\_10->show();

ui->listWidget->addItem(sq.value(0).toString()+" - "+sq.value(1).toString()+" - "+sq.value(3).toString()+" сумов");

}

else

{

QMessageBox::warning(this,"Ошибка","Данной книги нет в наличии");

ui->listWidget->addItem(author+" - "+bookn+" нет в наличии");}

}

}

void MainWindow::on\_pushButton\_7\_clicked()

{

QString author=ui->lineEdit\_3->text();

QString bookn=ui->lineEdit\_4->text();

ui->listWidget\_2->addItem(author+" - "+bookn);

}

void MainWindow::on\_pushButton\_11\_clicked()

{

QSqlQuery sq;

sq.exec("select \* from Books");

while(sq.next()){

ui->plainTextEdit->appendPlainText(sq.value(0).toString()+" - "+sq.value(1).toString()+" "+sq.value(2).toString()+" год "+sq.value(3).toString()+" сумов"+sq.value(4).toString()+" штук");}

QPrinter printer(QPrinter::HighResolution);

printer.setOutputFormat(QPrinter::PdfFormat);

printer.setOutputFileName("output.pdf");

ui->plainTextEdit->document()->print(&printer);

}

void MainWindow::on\_pushButton\_13\_clicked()

{

QString author=ui->lineEdit\_5->text();

QString bookn=ui->lineEdit\_6->text();

QSqlQuery quer;

QSqlQuery sq;

if(sq.exec("select \* from Books where Author='"+author +"'and Bookname='"+ bookn+"'"));

{

int c=0;

if(sq.next())

c++;

if(c==1)

QMessageBox::about(this,"Поиск","Данная книга есть в наличии. Цена книги- "+ sq.value(3).toString()+" сумов");

else

QMessageBox::warning(this,"Ошибка","Данной книги нет в наличии");

}

}

void MainWindow::on\_pushButton\_20\_clicked()

{

QString log=ui->lineEdit\_7->text();

QSqlQuery quer;

QSqlQuery sq;

if(sq.exec("select \* from Staff where Login='"+log+"'"));

{

int c=0;

if(sq.next())

c++;

if(c==1)

QMessageBox::about(this,"Поиск","Владелец логина найден. Ф\_И\_О владельца - "+ sq.value(0).toString());

else

{

QMessageBox::warning(this,"Ошибка","Данного логина не существует");

}

}

}

void MainWindow::on\_pushButton\_17\_clicked()

{

ui->lineEdit\_7->show();

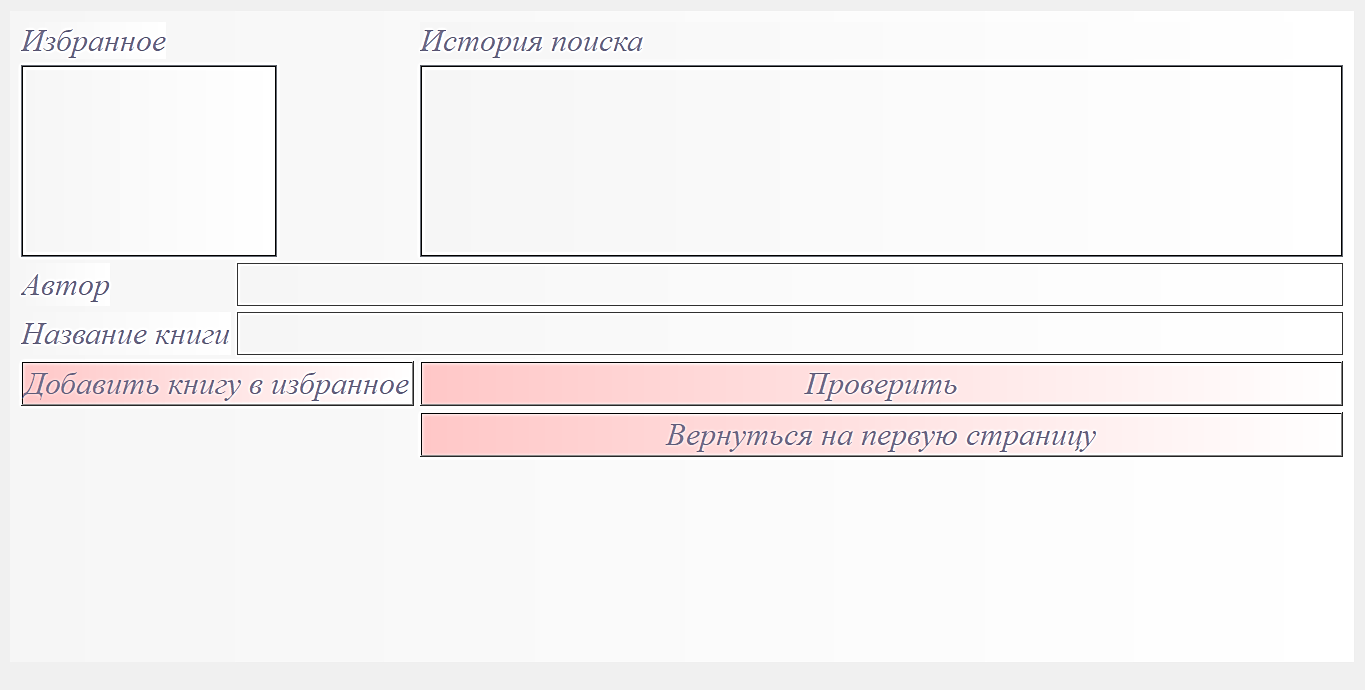
ui->pushButton\_20->show();

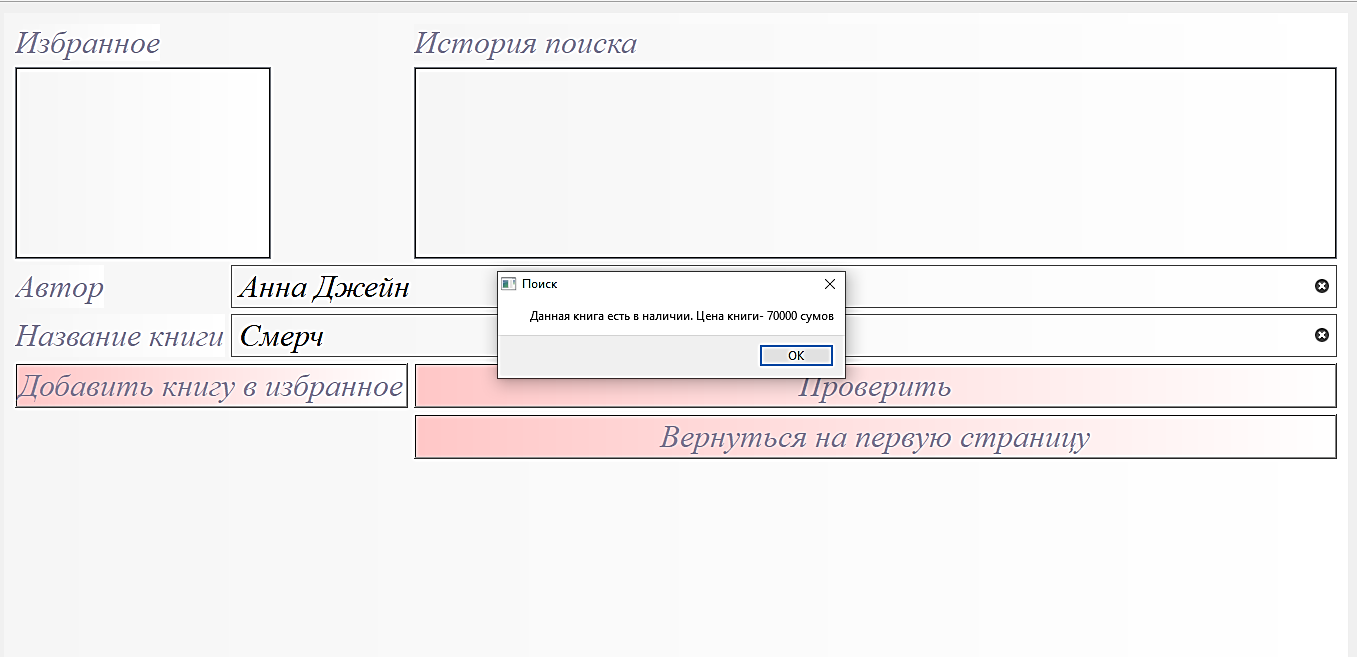
}

4. Работа программы

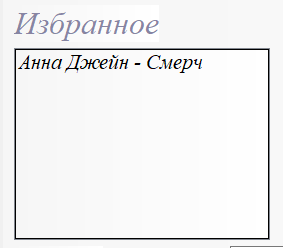


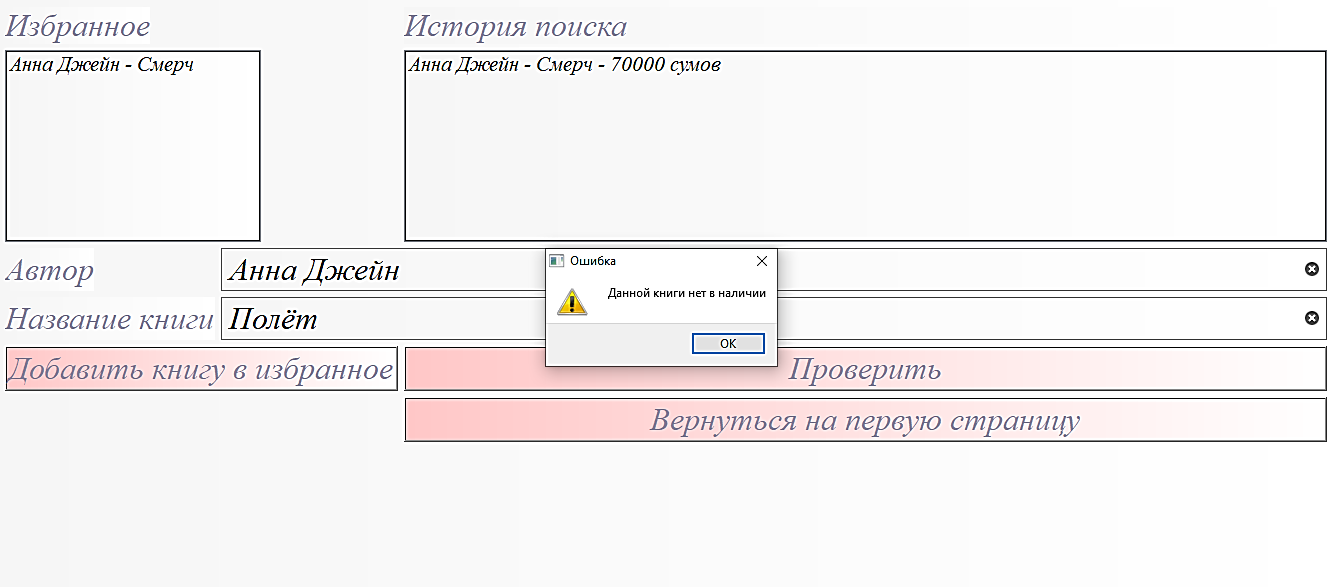
Вошли как пользователь:

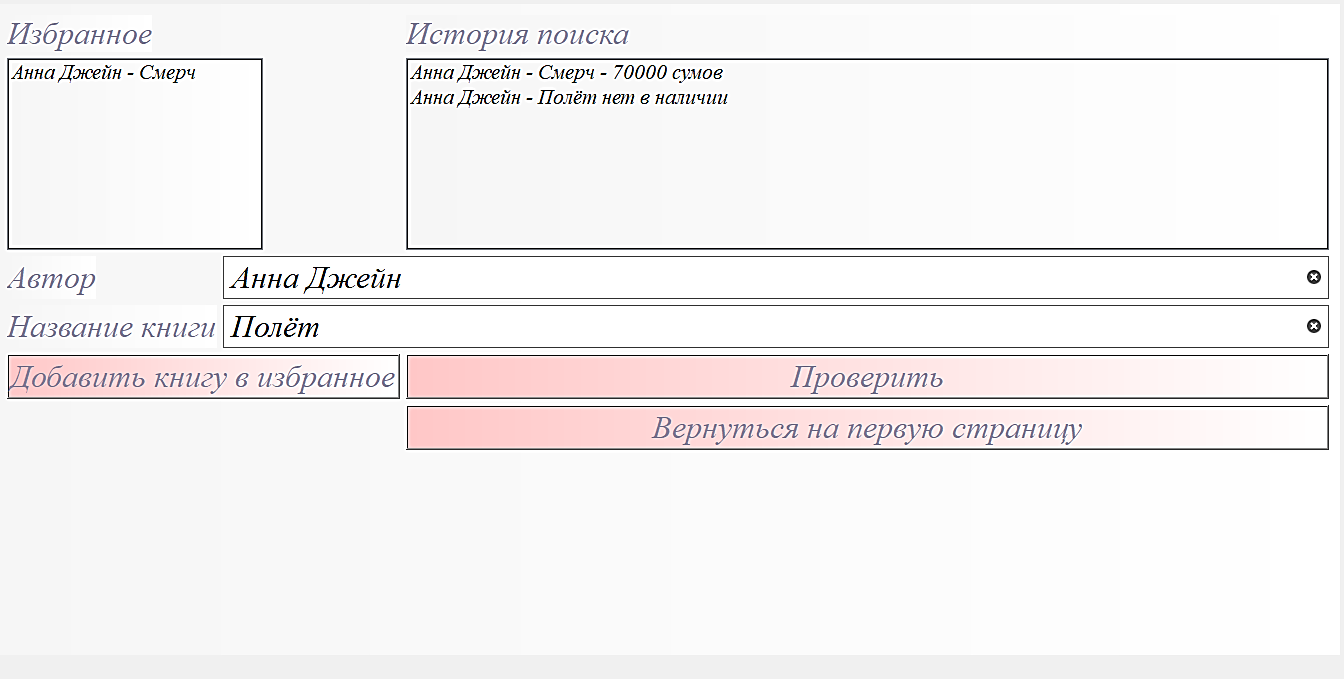




Добавили книгу в избранное:



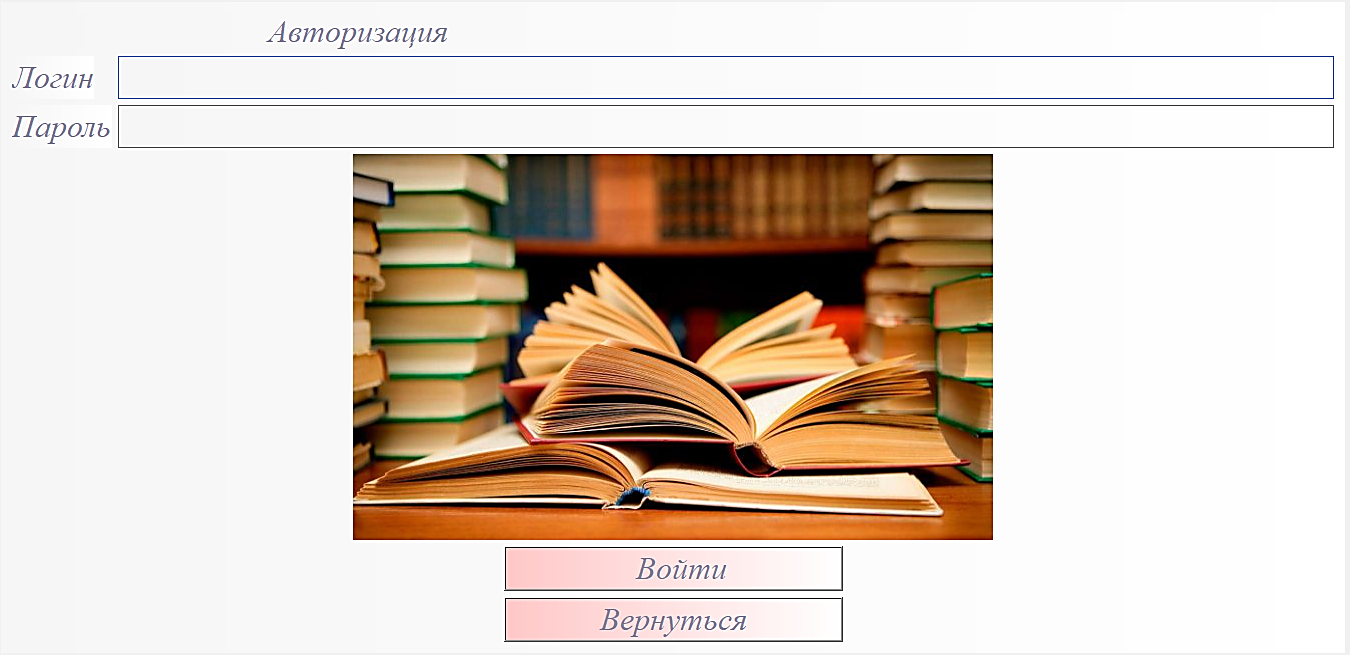




С помощью кнопки «Вернуться на первую страницу» вернёмся назад



Теперь можем войти как администратор:





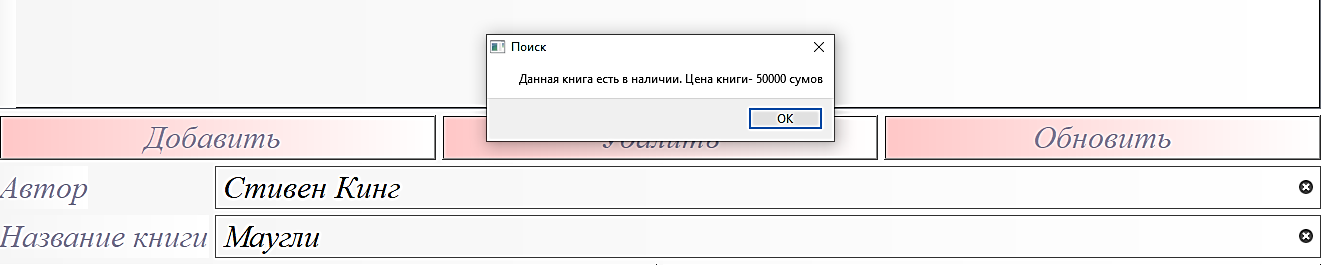
Видим нашу базу данных.

Мы можем добавить, удалить или изменить наши данные.

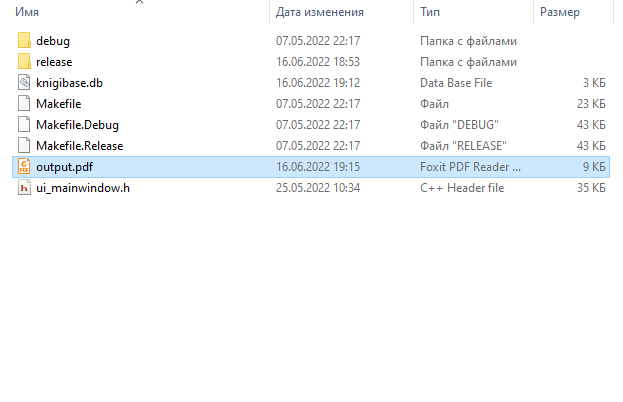




Администратор также может искать книги по имени автора и названию книги.



Также нашу базу данных мы можем записать в формате .pdf.



У нас в папке появился файл с названием output.pdf. Это и есть наша база данных в формате pdf.

С помощью кнопки «Добавить нового админа» мы можем добавить ещё одного админа.



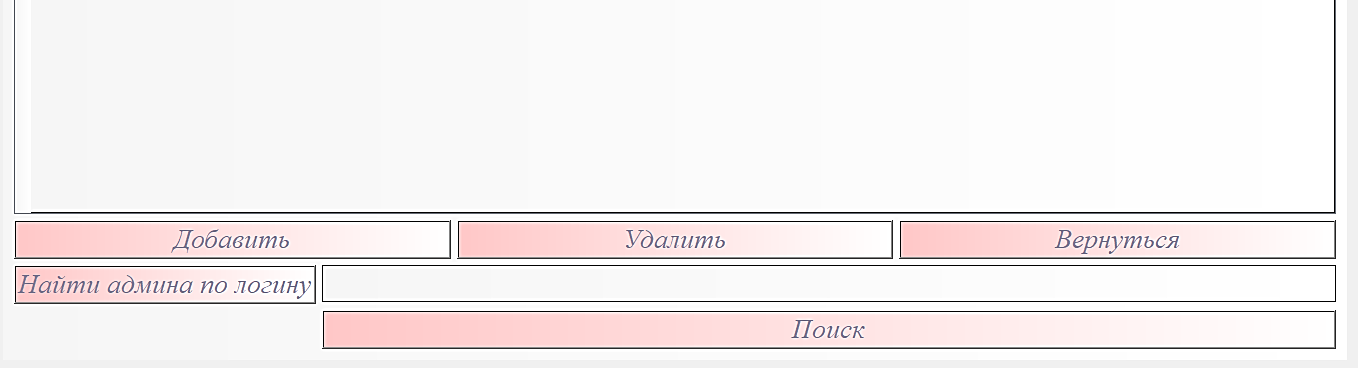
Открылось такое вот окошко.

Как говорилось выше мы можем добавить нового админа, удалить админа, или же изменить информацию об админах.

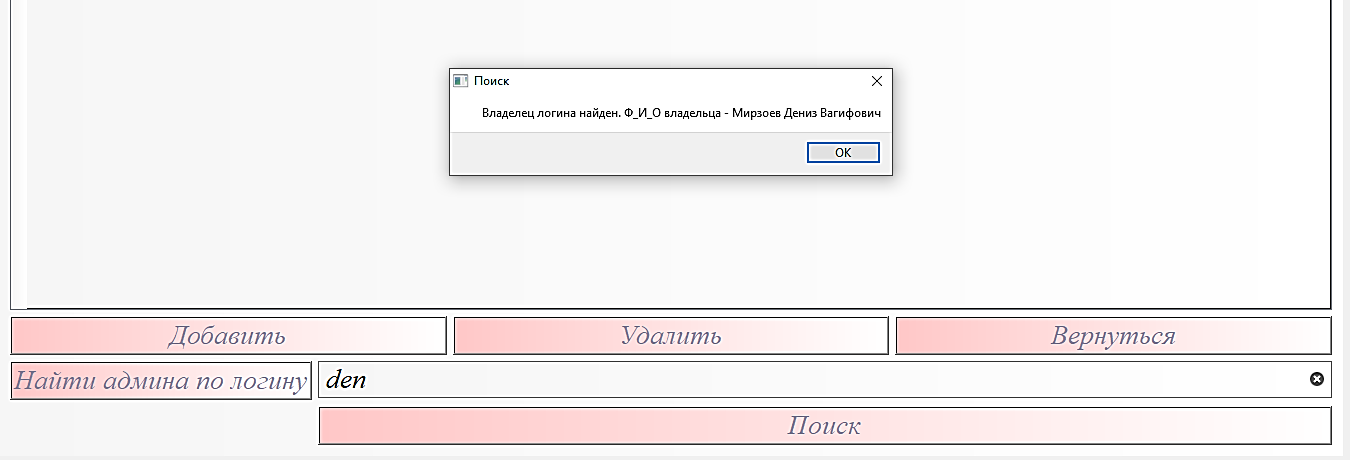
Добавим нового админа:

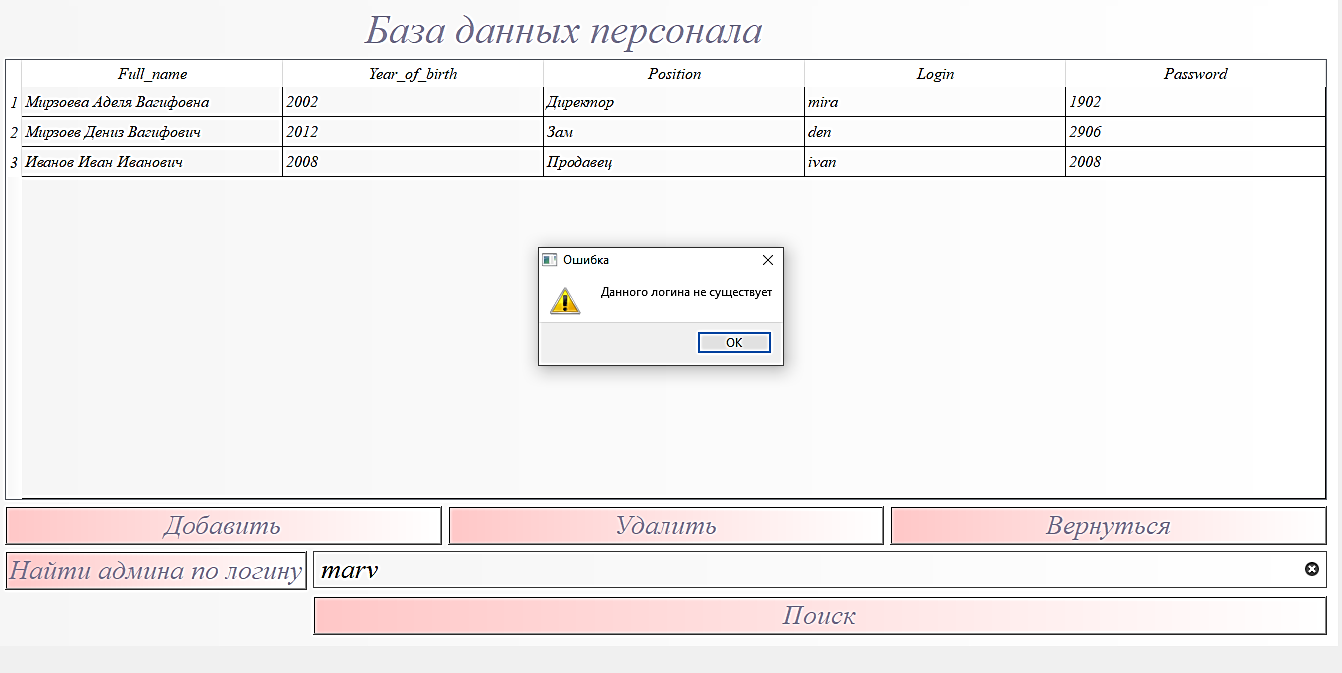


Покажем действие кнопки «Найти админа по логину».



Добавилось окошко для ввода логина.





Заключение В процессе выполнения курсовой работы изучены взаимосвязи в  
информационной структуре книжного магазина, проанализированы различные  
запросы к базе данных определенных таблиц. На основе полученных данных было  
спроектировано и разработано клиентское приложение с использованием кроссплатформенного инструментария Qt.  
 В теоретической части курсовой работы рассмотрены основы разработки  
автоматизированной информационной системы: процесс создания, проектирования  
и использования базы данных SQLite, реализация различных классов при  
разработке приложений с использованием библиотеки Qt.  
 В практической части курсовой работы реализована структура базы данных  
книжного магазина и описан процесс создания клиентского приложения с  
использованием программы Qt.

Cписок используемой литературы

1. Информатика. Базовый курс /Симонович С.В. и др. - СПб: Издательство «Питер», 2000. - 640с.

. Информатика. Учебное пособие /Под ред. В.Г. Кирия. - Иркутск: ИрГТУ ,1998 часть 2. - 382с.

. Джексон Г. Проектирование реляционных баз данных для использования с микроЭВМ. -М.: Мир, 1991.

. Карпова Т.С. Базы данных: модели, разработка, реализация. - СПб.: Питер, 2002