**Лабораторная работа №6**

**«ИССЛЕДОВАНИЕ СПОСОБОВ РАБОТЫ С БАЗАМИ ДАННЫХ В QT-ПРИЛОЖЕНИЯХ»**

**Цель работы**

Исследование способов взаимодействия с базами данных в Qt-приложениях. Приобретение навыков разработки приложений на основе баз данных на примере SQLite.

**Постановка задачи**

1. Изучить принципы работы с базами данных в Qt, способы соединения с БД, способы выполнения SQL-запросов (выполняется в ходе самостоятельной подготовки к лабораторной работе).

2. Установить SQLite. Выполнить скрипт из приложения Б для создания таблицы и добавления тестовых записей.

3. Создать Qt GUI приложение.

4. В дизайнере добавить на форму QTableView и необходимые элементы управления.

5. Реализовать логику приложения по варианту задания (Приложение В).

6. Исследовать работу созданного приложения, проанализировать работоспособность программы при вводе ошибочных данных.

**Ход работы**

1. Было создано Qt GUI приложение и добавлены на форму QTableView и две PushButton

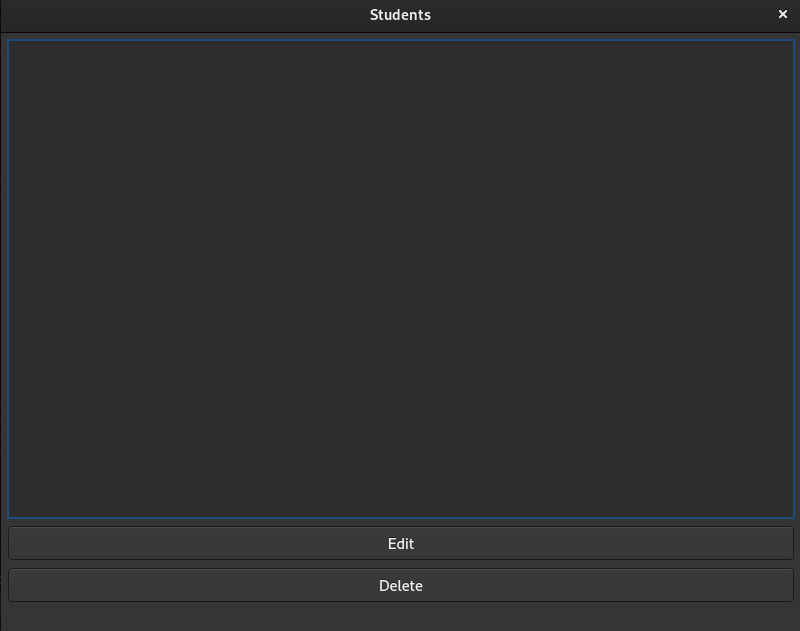


Рисунок 1 – Графический интерфейс приложения

2. Была реализована логика приложения, при нажатии на кнопку Edit появляется возможность редактирования выбранной ячейки таблицы, а при нажатии на кнопку Delete запись удаляется из таблицы.



Рисунок 2 – Изменение записи



Рисунок 3 – Удаление записи

**Выводы**

В ходе выполнения лабораторной работы были исследованы способы взаимодействия с базами данных в Qt-приложениях. Приобретены навыки разработки приложений на основе баз данных на примере SQLite.

В QT-приложениях для работы с данными из базы данных используются модели данных, которые позволяют удобно отображать данные в виде таблиц и выполнять операции с ними. Также QT имеет удобный механизм подключения к базам данных, при использовании SQLite необходимо было указать драйвер, файл и выполнить команду для подключения.

**ПРИЛОЖЕНИЕ А**

**«ИСХОДНЫЙ КОД ПРОГРАММЫ»**

Листинг А1 – Файл exec.sql

create table Student

(

Id int,

Name varchar(255),

GPA float,

EducForm varchar(255),

GroupId int

);

INSERT INTO Student VALUES (1, 'Ivanov Ivan', 4.1, 'Zaochnaja', 'I33-z');

INSERT INTO Student VALUES (2, 'Petr Pervyj', 4.3, 'Zaochnaja', 'M24-z');

INSERT INTO Student VALUES (3, 'Ovchinnikova Svetlana', 4.8,'Ochnaja', 'I52-d');

INSERT INTO Student VALUES (4, 'Zhelenkov Oleg', 4.4, 'Ochnaja', 'I51-d');

INSERT INTO Student VALUES (5, 'Zhilin Andrej', 4.8, 'Ochnaja', 'I51-d');

INSERT INTO Student VALUES (6, 'Lunjov Dmitrij', 4.2, 'Ochnaja', 'I51-d');

INSERT INTO Student VALUES (7, 'Belous Tatjana', 4.5, 'Ochnaja', 'I52-d');

INSERT INTO Student VALUES (8, 'Opanashhenko Irina', 4.8, 'Ochnaja','I52-d');

INSERT INTO Student VALUES (9, 'Neponjatnyj Tip', 3.1, 'Zaochnaja','I33-z');

INSERT INTO Student VALUES (10, 'Sjusjukajlo Dmitrij', 3.8, 'Ochnaja','I51-d');

INSERT INTO Student VALUES (11, 'Margaza Artjom', 4.0, 'Ochnaja','I52-d');

INSERT INTO Student VALUES (12, 'Melnikov Oleg', 4.3, 'Ochnaja', 'I52-d');

INSERT INTO Student VALUES (13, 'Krikunenko Dmitrij', 3.9, 'Ochnaja','I51-d');

INSERT INTO Student VALUES (14, 'Jakunina Anastasija', 5.0, 'Ochnaja','I51-d');

INSERT INTO Student VALUES (15, 'Trishina Elena', 4.4, 'Ochnaja', 'I51-d');

INSERT INTO Student VALUES (16, 'Mazurenko Elena', 3.7, 'Ochnaja','I52-d');

INSERT INTO Student VALUES (17, 'Adzhigeldieva Nurie', 4.1, 'Ochnaja','I52-d');

INSERT INTO Student VALUES (18, 'Sidorov Denis', 3.0, 'Zaochnaja','I33-z');

INSERT INTO Student VALUES (19, 'Golovach Elena', 4.1, 'Zaochnaja','M24-z');

INSERT INTO Student VALUES (20, 'Kuzmenko Dmitrij', 4.7, 'Zaochnaja','I33-z');

INSERT INTO Student VALUES (21, 'Gricenko Ivan', 4.7, 'Zaochnaja','I33-z');

INSERT INTO Student VALUES (22, 'Horolich Vladimir', 4.1, 'Zaochnaja','M24-z');

INSERT INTO Student VALUES (23, 'Azarov Nikolaj', 4.1, 'Zaochnaja','M24-z');

INSERT INTO Student VALUES (24, 'Napoleon Bonapart', 4.1,'Zaochnaja', 'M24-z');

INSERT INTO Student VALUES (25, 'Petrov Petr', 4.1, 'Zaochnaja', 'I33-z');

INSERT INTO Student VALUES (26, 'Koshkina Ljubov', 4.1, 'Zaochnaja','I33-z');

Листинг А2 – Файл data.sql

INSERT INTO Student VALUES (1, 'Ivanov Ivan', 4.1, 'Zaochnaja', 'I33-z');

INSERT INTO Student VALUES (2, 'Petr Pervyj', 4.3, 'Zaochnaja', 'M24-z');

INSERT INTO Student VALUES (3, 'Ovchinnikova Svetlana', 4.8,'Ochnaja', 'I52-d');

INSERT INTO Student VALUES (4, 'Zhelenkov Oleg', 4.4, 'Ochnaja', 'I51-d');

INSERT INTO Student VALUES (5, 'Zhilin Andrej', 4.8, 'Ochnaja', 'I51-d');

INSERT INTO Student VALUES (6, 'Lunjov Dmitrij', 4.2, 'Ochnaja', 'I51-d');

INSERT INTO Student VALUES (7, 'Belous Tatjana', 4.5, 'Ochnaja', 'I52-d');

INSERT INTO Student VALUES (8, 'Opanashhenko Irina', 4.8, 'Ochnaja','I52-d');

INSERT INTO Student VALUES (9, 'Neponjatnyj Tip', 3.1, 'Zaochnaja','I33-z');

INSERT INTO Student VALUES (10, 'Sjusjukajlo Dmitrij', 3.8, 'Ochnaja','I51-d');

INSERT INTO Student VALUES (11, 'Margaza Artjom', 4.0, 'Ochnaja','I52-d');

INSERT INTO Student VALUES (12, 'Melnikov Oleg', 4.3, 'Ochnaja', 'I52-d');

INSERT INTO Student VALUES (13, 'Krikunenko Dmitrij', 3.9, 'Ochnaja','I51-d');

INSERT INTO Student VALUES (14, 'Jakunina Anastasija', 5.0, 'Ochnaja','I51-d');

INSERT INTO Student VALUES (15, 'Trishina Elena', 4.4, 'Ochnaja', 'I51-d');

INSERT INTO Student VALUES (16, 'Mazurenko Elena', 3.7, 'Ochnaja','I52-d');

INSERT INTO Student VALUES (17, 'Adzhigeldieva Nurie', 4.1, 'Ochnaja','I52-d');

INSERT INTO Student VALUES (18, 'Sidorov Denis', 3.0, 'Zaochnaja','I33-z');

INSERT INTO Student VALUES (19, 'Golovach Elena', 4.1, 'Zaochnaja','M24-z');

INSERT INTO Student VALUES (20, 'Kuzmenko Dmitrij', 4.7, 'Zaochnaja','I33-z');

INSERT INTO Student VALUES (21, 'Gricenko Ivan', 4.7, 'Zaochnaja','I33-z');

INSERT INTO Student VALUES (22, 'Horolich Vladimir', 4.1, 'Zaochnaja','M24-z');

INSERT INTO Student VALUES (23, 'Azarov Nikolaj', 4.1, 'Zaochnaja','M24-z');

INSERT INTO Student VALUES (24, 'Napoleon Bonapart', 4.1,'Zaochnaja', 'M24-z');

INSERT INTO Student VALUES (25, 'Petrov Petr', 4.1, 'Zaochnaja', 'I33-z');

INSERT INTO Student VALUES (26, 'Koshkina Ljubov', 4.1, 'Zaochnaja','I33-z');

Листинг А3 – Файл 6th.pro

QT += core gui

QT += sql

greaterThan(QT\_MAJOR\_VERSION, 4): QT += widgets

CONFIG += c++17

# You can make your code fail to compile if it uses deprecated APIs.

# In order to do so, uncomment the following line.

#DEFINES += QT\_DISABLE\_DEPRECATED\_BEFORE=0x060000 # disables all the APIs deprecated before Qt 6.0.0

SOURCES += \

main.cpp \

mainwindow.cpp

HEADERS += \

mainwindow.h

FORMS += \

mainwindow.ui

# Default rules for deployment.

qnx: target.path = /tmp/$${TARGET}/bin

else: unix:!android: target.path = /opt/$${TARGET}/bin

!isEmpty(target.path): INSTALLS += target

Листинг А4 – Файл mainwindow.h

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QMainWindow>

#include <QtSql>

#include <QMessageBox>

QT\_BEGIN\_NAMESPACE

namespace Ui { class MainWindow; }

QT\_END\_NAMESPACE

class MainWindow : public QMainWindow

{

Q\_OBJECT

public:

MainWindow(QWidget \*parent = nullptr);

~MainWindow();

public slots:

void delete\_student();

void edit\_student();

private:

Ui::MainWindow \*ui;

QSqlDatabase db;

QSqlTableModel \*db\_model;

};

#endif // MAINWINDOW\_H

Листинг А5 – Файл mainwindow.cpp

#include "mainwindow.h"

#include "ui\_mainwindow.h"

#include <QSqlDatabase>

#include <QSqlError>

#include <QSqlQuery>

MainWindow::MainWindow(QWidget \*parent)

: QMainWindow(parent)

, ui(new Ui::MainWindow)

{

ui->setupUi(this);

QString DBpath = "student.db";

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

db.setDatabaseName(DBpath);

if(db.open()){

QMessageBox msgBox;

msgBox.setText("Connected");

msgBox.exec();

}

else

{

QMessageBox::critical(this,tr("SQLite connection"), tr("Unable connect to DB."));

exit(1);

}

db\_model = new QSqlTableModel(this, db);

db\_model->setTable("Student");

db\_model->setEditStrategy(QSqlTableModel::OnFieldChange);

db\_model->select();

ui->tableView->setModel(db\_model);

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

connect(ui->pushButton, SIGNAL(clicked()), SLOT(edit\_student()));

connect(ui->pushButton\_2, SIGNAL(clicked()), SLOT(delete\_student()));

}

MainWindow::~MainWindow()

{

delete ui;

}

void MainWindow::delete\_student()

{

QModelIndexList ids = ui->tableView->selectionModel()->selection().indexes();

QSet<int> \*rowsToDelete = new QSet<int>();

for (int i = 0; i < ids.count(); i++)

{

QModelIndex id = ids.at(i);

rowsToDelete->insert(id.row());

}

QAbstractItemModel \*model = ui->tableView->model();

QSet<int>::iterator i;

for (i = rowsToDelete->begin(); i != rowsToDelete->end(); ++i)

{

model->removeRow(\*i);

}

}

void MainWindow::edit\_student()

{

QModelIndex currentIndex = ui->tableView->currentIndex();

if (currentIndex.isValid())

{

ui->tableView->edit(currentIndex);

}

}