*Федеральное государственное бюджетное образовательное учреждение* *высшего образования*

|  |  |
| --- | --- |
|  | ***«Московский государственный технический университет  имени Н.Э. Баумана»***  ***(национальный исследовательский университет)***  ***(МГТУ им. Н.Э. Баумана)*** |

ФАКУЛЬТЕТ \_\_\_\_ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_\_\_Компьютерные Системы и сети (ИУ6)\_\_\_\_\_\_\_\_\_\_\_

**Отчет**

**по практической работе**

Студент гр. ИУ6-22Б  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. П. Плютто**

(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ О. А. Веселовская**

(Подпись, дата) (И.О. Фамилия)

Москва, 2023

Выполнить объектную декомпозицию, разработать формы интерфейса, диаграмму состояний интерфейса, диаграммы классов интерфейсной и предметной областей, диаграмму последовательности действий одной из реализуемых операций. Разработать, протестировать и отладить программу.

База данных футбольного чемпионата содержит сведения о матчах: стадион проведения, команды-соперники (две строки), счет (два числа). Программа должна в интерактивном режиме позволять добавлять, просматривать и удалять данные, а также давать ответ на перечисленные запросы.

1. Показать все матчи заданной команды (стадион, соперник).

2. Определить команды, выигравшие указанное число матчей или более.

3. Показать список матчей, сыгранных вничью (стадион, команды, счет).

4. Построить гистограмму, показывающую число матчей, проведенных на каждом стадионе.

Организовать ввод и вывод данных через табличное представление. Не позволять вводить записи-дубликаты. Для выбора параметра поиска (футбольная команда) использовать выпадающий список.

**Решение**

Так как в задаче написано, что можно использовать базу данных, я выбрал для реализации решения одну из наиболее популярных баз данных для обучения MySQL. Используя встроенные средства Qt и 5-6 запросов MySQL мне удалось создать требуемое приложение.

**Table.h**

#ifndef TABLE\_H

#define TABLE\_H

#include <QMainWindow>

#include <QtSQL>

#include <QtDebug>

#include <QFileInfo>

#include <QSqlDatabase>

#include <QMessageBox>

QT\_BEGIN\_NAMESPACE

namespace Ui { class table; }

QT\_END\_NAMESPACE

class table : public QMainWindow

{

Q\_OBJECT

public:

table(QWidget \*parent = nullptr);

~table();

QSqlDatabase db;

private slots:

void on\_pushButton\_6\_clicked();

void on\_pushButton\_5\_clicked();

void on\_pushButton\_clicked();

void on\_pushButton\_3\_clicked();

void on\_pushButton\_7\_clicked();

void on\_pushButton\_4\_clicked();

void on\_pushButton\_2\_clicked();

void on\_pushButton\_8\_clicked();

private:

Ui::table \*ui;

};

#endif // TABLE\_H

**Table.cpp**

#include "table.h"

#include "ui\_table.h"

#include "string"

#include "unordered\_map"

#include <QtCharts/QBarSeries>

#include <QtCharts/QBarSet>

#include <QtCharts/QChart>

#include <QtCharts/QChartView>

#include <QtCharts/QValueAxis>

#include <QtCharts/QCategoryAxis>

#include <QtCharts>

#include <QVBoxLayout>

#include <QVector>

QChartView\* createHistogram(std::unordered\_map<std::string, int> data)

{

// Создаем объекты для данных гистограммы

// Создаем объект гистограммы и добавляем данные

QBarSeries \*series = new QBarSeries();

for(auto i:data){

QBarSet \*set = new QBarSet(QString::fromUtf8(i.first));

\*set << i.second;

series->append(set);

}

// Создаем объект диаграммы и добавляем на него гистограмму

QChart \*chart = new QChart();

chart->addSeries(series);

chart->setTitle("Гистограмма стадионов");

// Устанавливаем оси координат

// QCategoryAxis \*axisX = new QCategoryAxis();

int count{}, mx{};

for(auto i:data){

count++;

// axisX->append(QString::fromUtf8(i.first),count);

if (i.second > mx) mx = i.second;

qDebug()<<QString::fromUtf8(i.first)<< QString::number(i.second);

}

// axisX->setLabelsAngle(30);

// chart->addAxis(axisX, Qt::AlignBottom);

// series->attachAxis(axisX);

QValueAxis \*axisY = new QValueAxis();

axisY->setTitleText("Количество игр");

axisY->setLabelFormat("%d");

axisY->setTickCount(mx+1);

chart->addAxis(axisY, Qt::AlignLeft);

series->attachAxis(axisY);

// Создаем объект отображения гистограммы и устанавливаем его в макет

QChartView \*chartView = new QChartView(chart);

QVBoxLayout \*layout = new QVBoxLayout;

layout->addWidget(chartView);

// Устанавливаем макет для главного окна и возвращаем объект отображения гистограммы

QWidget \*window = new QWidget;

window->setLayout(layout);

window->resize(1200, 600);

window->show();

return chartView;

}

table::table(QWidget \*parent)

: QMainWindow(parent)

, ui(new Ui::table)

{

ui->setupUi(this);

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

db.setDatabaseName("/Users/pluttan/Desktop/\_Сем2/BMSTU-2/OOP/Мое\_прак/pr3/pr3.db");

if (db.open()){

QMessageBox::information(this,"connection","connect");

}

on\_pushButton\_7\_clicked();

}

table::~table()

{

delete ui;

}

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

{

QSqlQueryModel \*modal = new QSqlQueryModel();

QSqlQuery\* q = new QSqlQuery(db);

q->prepare(ui->lineEdit\_7->text());

if(q->exec()){

modal->setQuery(\*q);

ui -> tableView -> setModel(modal);

}else QMessageBox::information(this,"error","Something went wrong :(");

}

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

{

QSqlQuery que;

if (que.exec("INSERT INTO games (stadium, team1, team2, score1, score2) VALUES ('"

+ ui->lineEdit\_2->text() + "','" + ui->lineEdit\_3->text() + "','"

+ ui->lineEdit\_4->text() + "'," + ui->lineEdit\_5->text() + ","

+ ui->lineEdit\_6->text() + ')')){

} else QMessageBox::information(this,"error","Something went wrong :(");

on\_pushButton\_7\_clicked();

}

void table::on\_pushButton\_clicked()

{

QModelIndexList selected\_indexes = ui->tableView->selectionModel()->selectedIndexes();

if (!selected\_indexes.isEmpty()) {

QModelIndex model\_index = selected\_indexes.first();

QSqlQuery que;

if (que.exec("DELETE FROM games WHERE id = " + QString::number(model\_index.row()+1))){

} else QMessageBox::information(this,"error","Something went wrong :(");

on\_pushButton\_7\_clicked();

}

}

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

{

QSqlQueryModel \*modal = new QSqlQueryModel();

QSqlQuery\* q = new QSqlQuery(db);

q->prepare(QString::fromUtf8(

(std::string)"SELECT team, COUNT(\*) as wins FROM " +

(std::string)"(SELECT team1 as team, score1, score2 "+

(std::string)"FROM games UNION ALL SELECT team2 as "+

(std::string)"team, score2, score1 FROM games) as games "+

(std::string)"WHERE score1 > score2 GROUP BY team HAVING wins >= ") +

ui->lineEdit->text());

if(q->exec()){

modal->setQuery(\*q);

ui -> tableView -> setModel(modal);

}else QMessageBox::information(this,"error","Something went wrong :(");

}

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

{

QSqlQueryModel \*modal = new QSqlQueryModel();

QSqlQuery\* q = new QSqlQuery(db);

q->prepare("select stadium, team1, team2, score1, score2 from games");

if(q->exec()){

modal->setQuery(\*q);

ui -> tableView -> setModel(modal);

}else QMessageBox::information(this,"error","Something went wrong :(");

modal = new QSqlQueryModel();

q = new QSqlQuery(db);

q->prepare("SELECT DISTINCT team1 as team FROM games UNION SELECT DISTINCT team2 as team FROM games");

if(q->exec()){

modal->setQuery(\*q);

ui -> listView -> setModel(modal);

}else QMessageBox::information(this,"error","Something went wrong :(");

}

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

{

QSqlQueryModel \*modal = new QSqlQueryModel();

QSqlQuery\* q = new QSqlQuery(db);

q->prepare("SELECT stadium, team1, team2, score1, score2 FROM games where score1 = score2");

if(q->exec()){

modal->setQuery(\*q);

ui -> tableView -> setModel(modal);

}else QMessageBox::information(this,"error","Something went wrong :(");

}

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

{

QModelIndexList selected\_indexes = ui->listView->selectionModel()->selectedIndexes();

if (!selected\_indexes.isEmpty()) {

QModelIndex model\_index = selected\_indexes.first();

int row = model\_index.row();

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

QVariant data = model->data(model->index(row, 0));

QSqlQueryModel \*modal = new QSqlQueryModel();

QSqlQuery\* q = new QSqlQuery(db);

q->prepare("SELECT stadium, team1, team2, score1, score2 FROM games where team1 = '" + data.toString() +

"' or team2 = '" + data.toString() + "'");

if(q->exec()){

modal->setQuery(\*q);

ui -> tableView -> setModel(modal);

}else QMessageBox::information(this,"error","Something went wrong :(");

}

}

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

{

std::unordered\_map<std::string, int> sel;

QSqlQuery query("SELECT stadium FROM games");

while (query.next()) {

std::string name = query.value(0).toString().toStdString();

sel[name]++;

};

QChartView\* chartView = createHistogram(sel);

chartView->resize(800, 600);

chartView->show();

}

**Main.cpp**

#include "table.h"

#include <QApplication>

int main(int argc, char \*argv[])

{

QApplication a(argc, argv);

table w;

w.show();

return a.exec();

}

**Table.ui**

<?xml version="1.0" encoding="UTF-8"?>

<ui version="4.0">

<class>table</class>

<widget class="QMainWindow" name="table">

<property name="geometry">

<rect>

<x>0</x>

<y>0</y>

<width>776</width>

<height>715</height>

</rect>

</property>

<property name="windowTitle">

<string>table</string>

</property>

<widget class="QWidget" name="centralwidget">

<layout class="QHBoxLayout" name="horizontalLayout\_5">

<item>

<layout class="QVBoxLayout" name="verticalLayout\_2">

<item>

<widget class="QTableView" name="tableView"/>

</item>

<item>

<spacer name="horizontalSpacer\_5">

<property name="orientation">

<enum>Qt::Horizontal</enum>

</property>

<property name="sizeHint" stdset="0">

<size>

<width>40</width>

<height>20</height>

</size>

</property>

</spacer>

</item>

<item>

<layout class="QVBoxLayout" name="verticalLayout">

<item>

<layout class="QHBoxLayout" name="horizontalLayout\_4">

<item>

<widget class="QLabel" name="label\_2">

<property name="text">

<string>Стадион</string>

</property>

</widget>

</item>

<item>

<spacer name="horizontalSpacer\_2">

<property name="orientation">

<enum>Qt::Horizontal</enum>

</property>

<property name="sizeHint" stdset="0">

<size>

<width>40</width>

<height>20</height>

</size>

</property>

</spacer>

</item>

<item>

<widget class="QLabel" name="label\_3">

<property name="text">

<string> Команда 1</string>

</property>

</widget>

</item>

<item>

<spacer name="horizontalSpacer\_3">

<property name="orientation">

<enum>Qt::Horizontal</enum>

</property>

<property name="sizeHint" stdset="0">

<size>

<width>40</width>

<height>20</height>

</size>

</property>

</spacer>

</item>

<item>

<widget class="QLabel" name="label\_4">

<property name="text">

<string> Команда 2</string>

</property>

</widget>

</item>

<item>

<spacer name="horizontalSpacer\_4">

<property name="orientation">

<enum>Qt::Horizontal</enum>

</property>

<property name="sizeHint" stdset="0">

<size>

<width>40</width>

<height>20</height>

</size>

</property>

</spacer>

</item>

<item>

<widget class="QLabel" name="label\_5">

<property name="text">

<string>Счет </string>

</property>

</widget>

</item>

</layout>

</item>

<item>

<layout class="QHBoxLayout" name="horizontalLayout">

<item>

<widget class="QLineEdit" name="lineEdit\_2">

<property name="placeholderText">

<string>Стадион</string>

</property>

</widget>

</item>

<item>

<widget class="QLineEdit" name="lineEdit\_3">

<property name="placeholderText">

<string>Команда 1</string>

</property>

</widget>

</item>

<item>

<widget class="QLineEdit" name="lineEdit\_4">

<property name="placeholderText">

<string>Команда 2</string>

</property>

</widget>

</item>

<item>

<widget class="QLineEdit" name="lineEdit\_5">

<property name="minimumSize">

<size>

<width>20</width>

<height>0</height>

</size>

</property>

<property name="maximumSize">

<size>

<width>20</width>

<height>16777215</height>

</size>

</property>

</widget>

</item>

<item>

<widget class="QLabel" name="label\_6">

<property name="text">

<string>:</string>

</property>

</widget>

</item>

<item>

<widget class="QLineEdit" name="lineEdit\_6">

<property name="maximumSize">

<size>

<width>20</width>

<height>16777215</height>

</size>

</property>

</widget>

</item>

</layout>

</item>

<item>

<layout class="QHBoxLayout" name="horizontalLayout\_2">

<item>

<widget class="QPushButton" name="pushButton\_7">

<property name="text">

<string>Сбросить</string>

</property>

</widget>

</item>

<item>

<spacer name="horizontalSpacer\_6">

<property name="orientation">

<enum>Qt::Horizontal</enum>

</property>

<property name="sizeHint" stdset="0">

<size>

<width>40</width>

<height>20</height>

</size>

</property>

</spacer>

</item>

<item>

<widget class="QPushButton" name="pushButton\_8">

<property name="text">

<string>Показать гистограмму</string>

</property>

</widget>

</item>

<item>

<spacer name="horizontalSpacer">

<property name="orientation">

<enum>Qt::Horizontal</enum>

</property>

<property name="sizeHint" stdset="0">

<size>

<width>40</width>

<height>20</height>

</size>

</property>

</spacer>

</item>

<item>

<widget class="QPushButton" name="pushButton\_5">

<property name="text">

<string>Добавить</string>

</property>

</widget>

</item>

</layout>

</item>

<item>

<layout class="QHBoxLayout" name="horizontalLayout\_3">

<item>

<widget class="QLineEdit" name="lineEdit\_7">

<property name="placeholderText">

<string>Создать запрос в базу данных (SQL syntax)</string>

</property>

</widget>

</item>

<item>

<widget class="QPushButton" name="pushButton\_6">

<property name="text">

<string>Выполнить</string>

</property>

</widget>

</item>

</layout>

</item>

</layout>

</item>

</layout>

</item>

<item>

<layout class="QVBoxLayout" name="verticalLayout\_3">

<item>

<widget class="QPushButton" name="pushButton">

<property name="maximumSize">

<size>

<width>200</width>

<height>16777215</height>

</size>

</property>

<property name="text">

<string>Удалить</string>

</property>

</widget>

</item>

<item>

<widget class="QListView" name="listView">

<property name="maximumSize">

<size>

<width>200</width>

<height>16777215</height>

</size>

</property>

</widget>

</item>

<item>

<widget class="QPushButton" name="pushButton\_2">

<property name="minimumSize">

<size>

<width>200</width>

<height>0</height>

</size>

</property>

<property name="maximumSize">

<size>

<width>200</width>

<height>16777215</height>

</size>

</property>

<property name="text">

<string>Найти игры</string>

</property>

</widget>

</item>

<item>

<widget class="QLineEdit" name="lineEdit">

<property name="maximumSize">

<size>

<width>200</width>

<height>16777215</height>

</size>

</property>

</widget>

</item>

<item>

<widget class="QPushButton" name="pushButton\_3">

<property name="maximumSize">

<size>

<width>200</width>

<height>16777215</height>

</size>

</property>

<property name="text">

<string>Найти команды</string>

</property>

</widget>

</item>

<item>

<widget class="QPushButton" name="pushButton\_4">

<property name="maximumSize">

<size>

<width>200</width>

<height>16777215</height>

</size>

</property>

<property name="text">

<string>Матчи, сыгранные в ничью</string>

</property>

</widget>

</item>

</layout>

</item>

</layout>

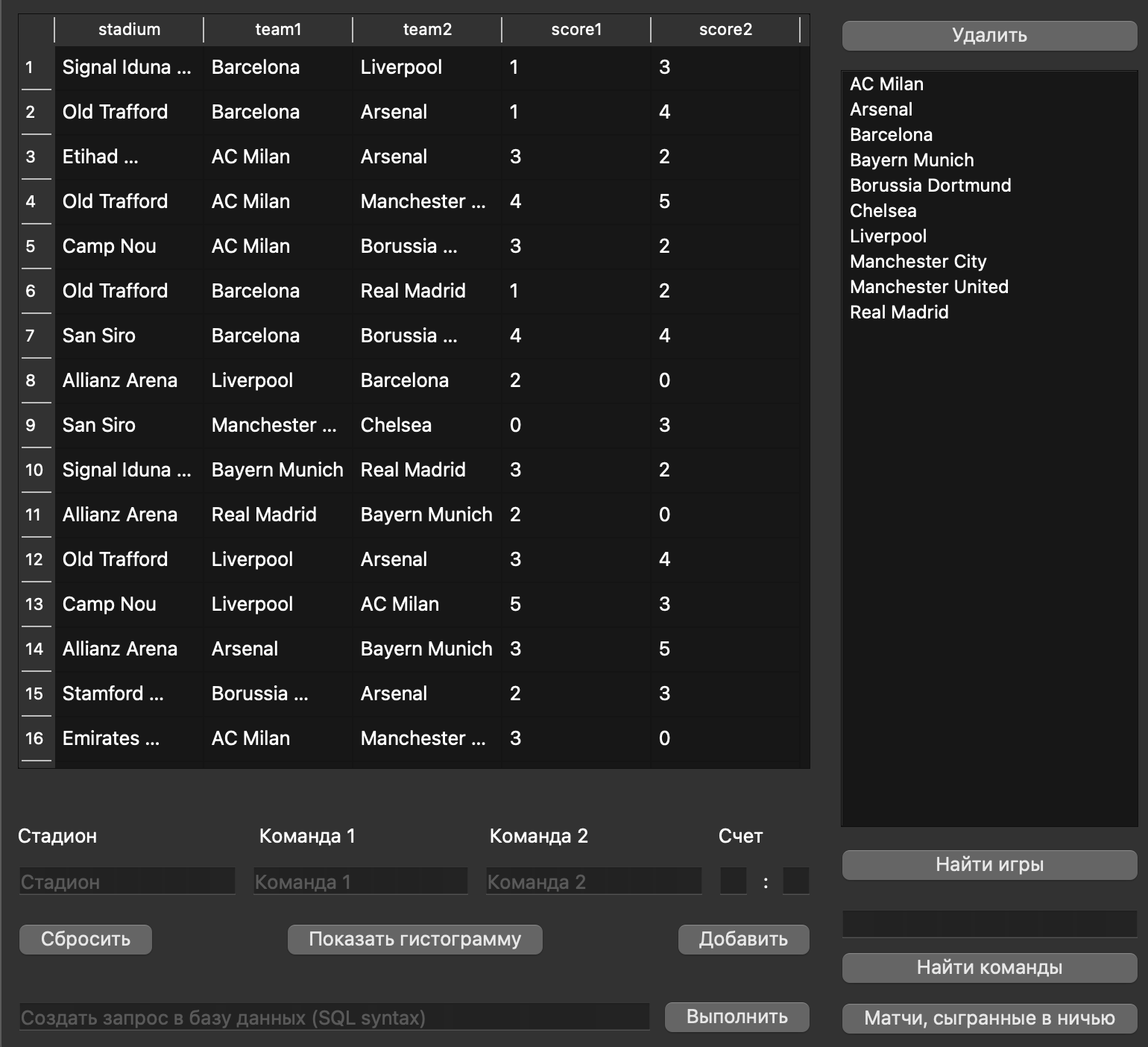
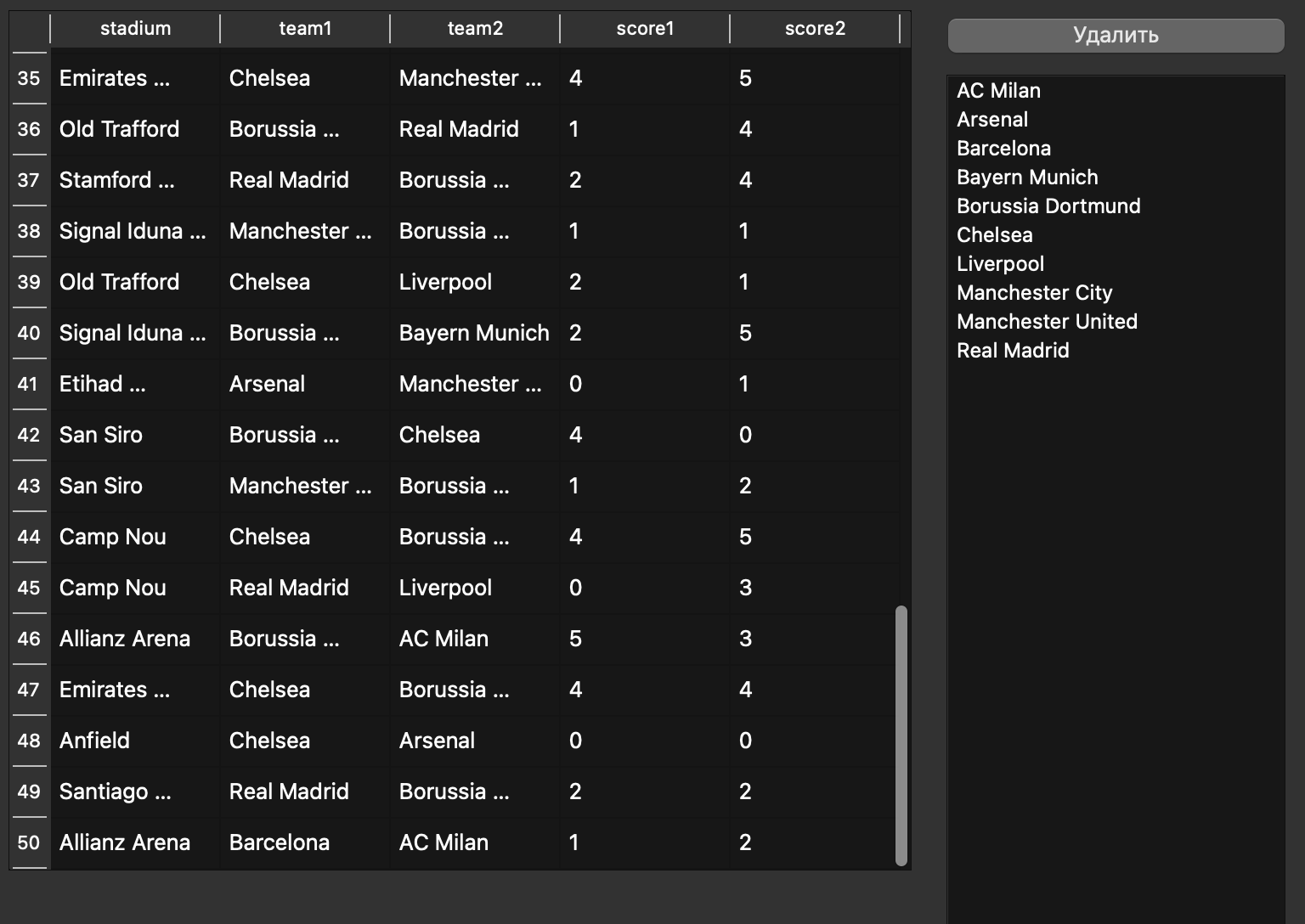
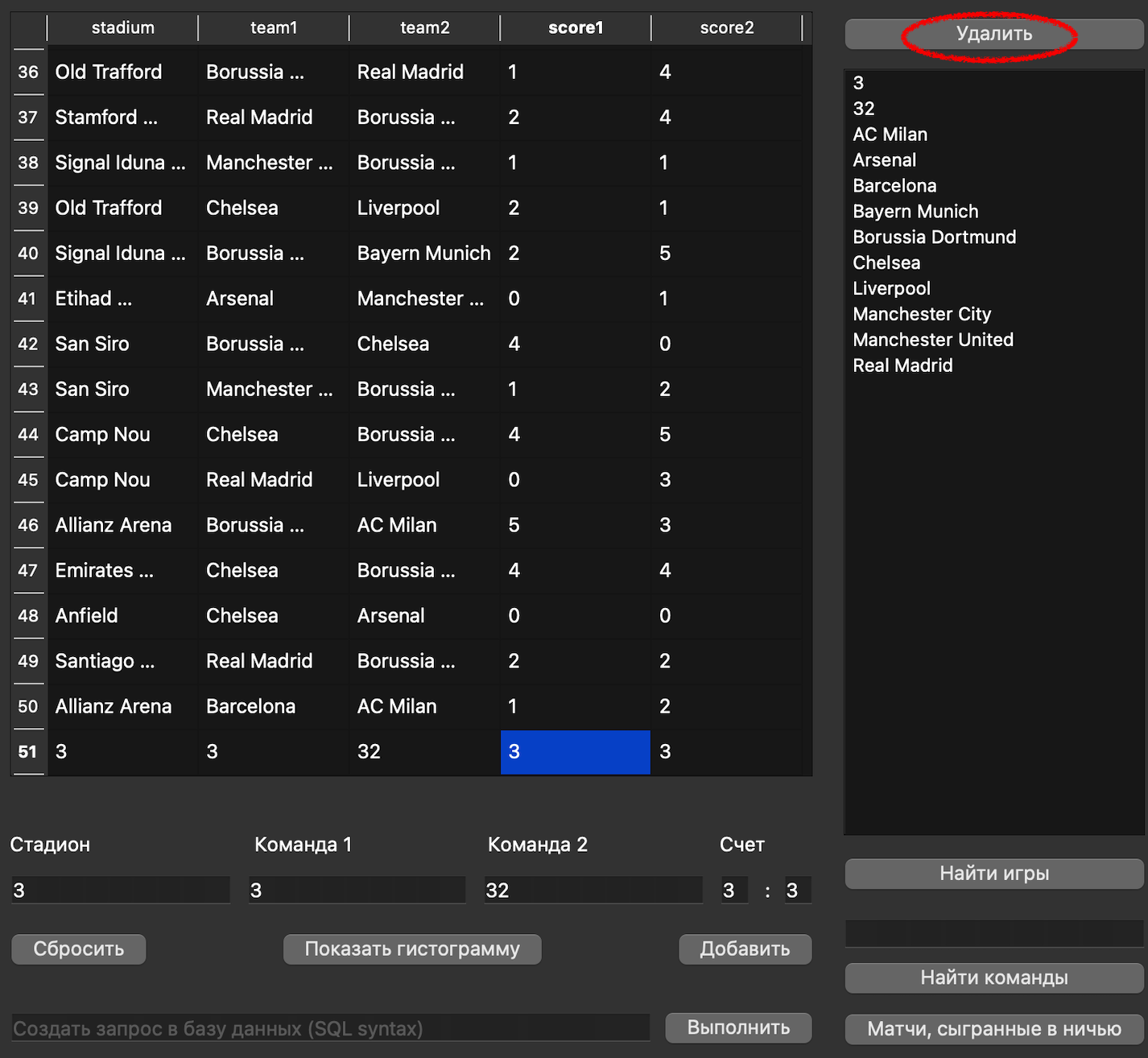
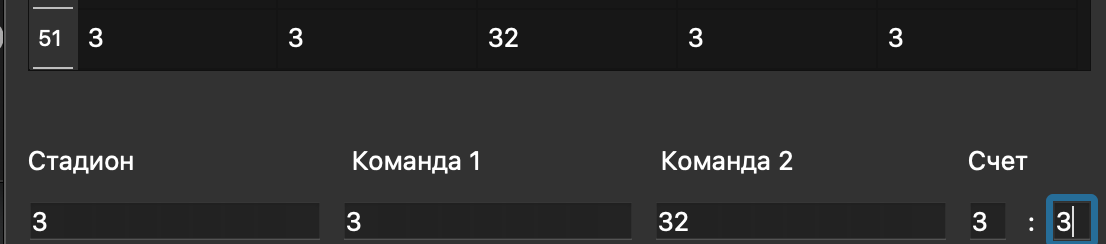
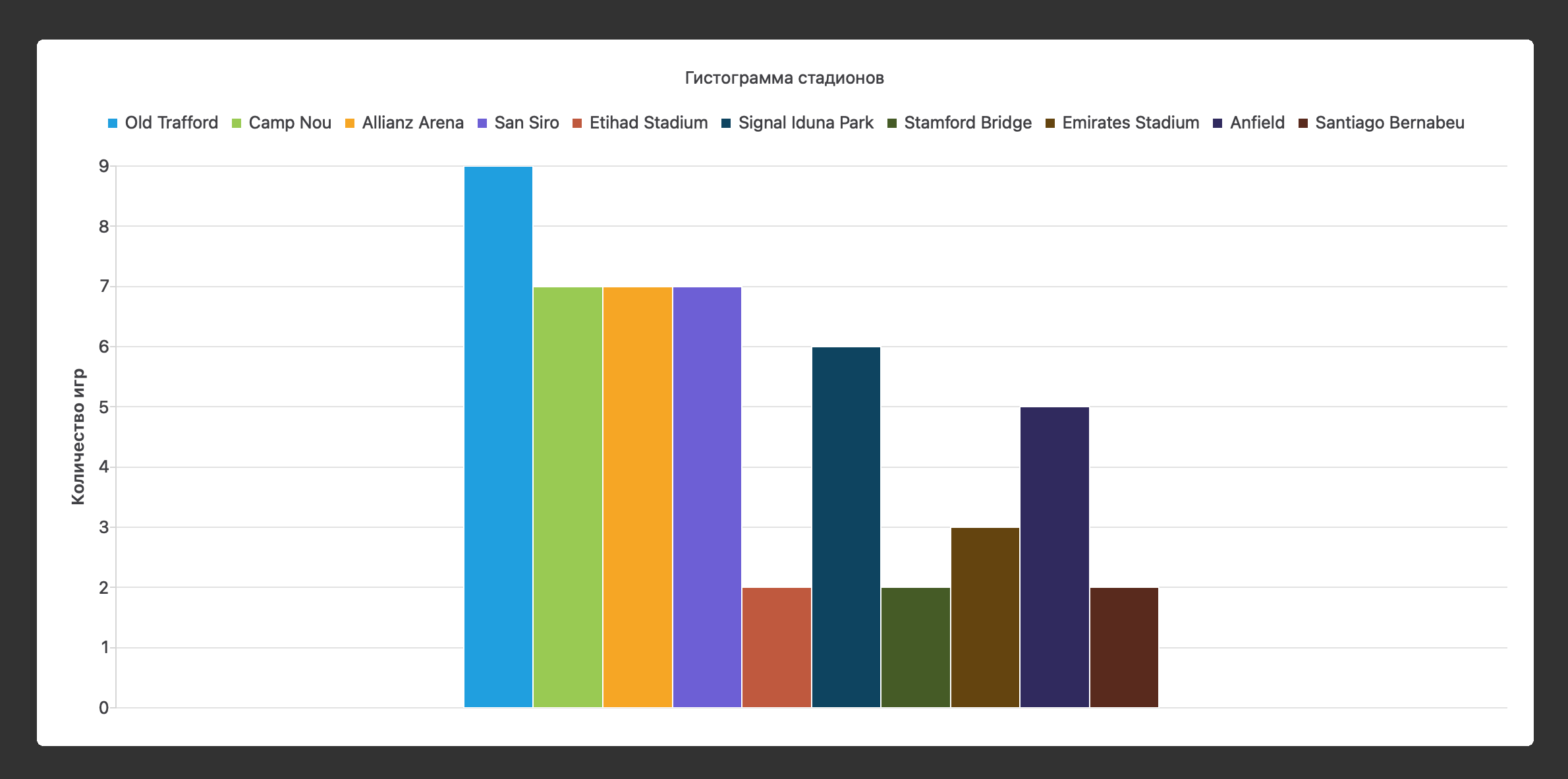
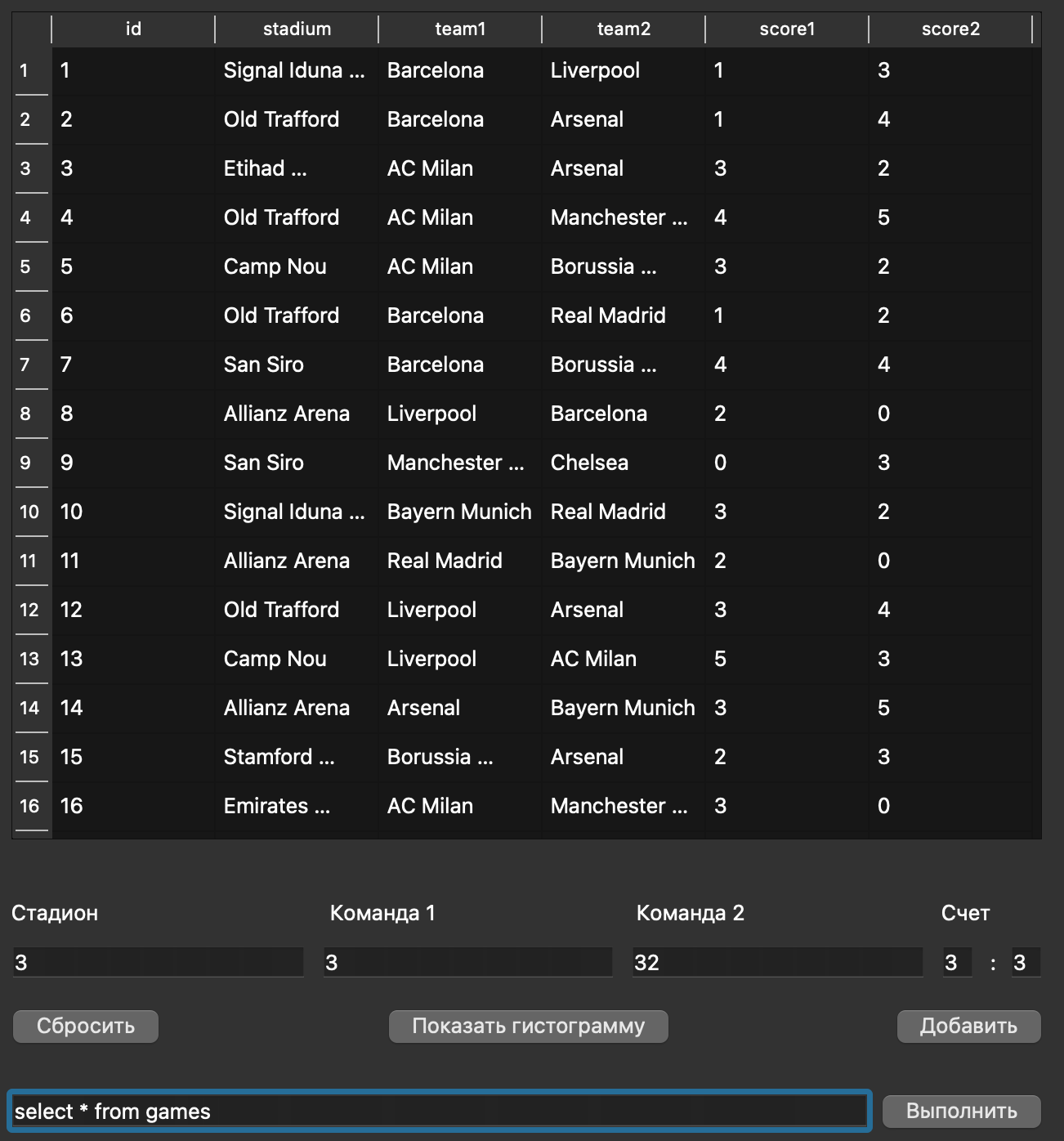
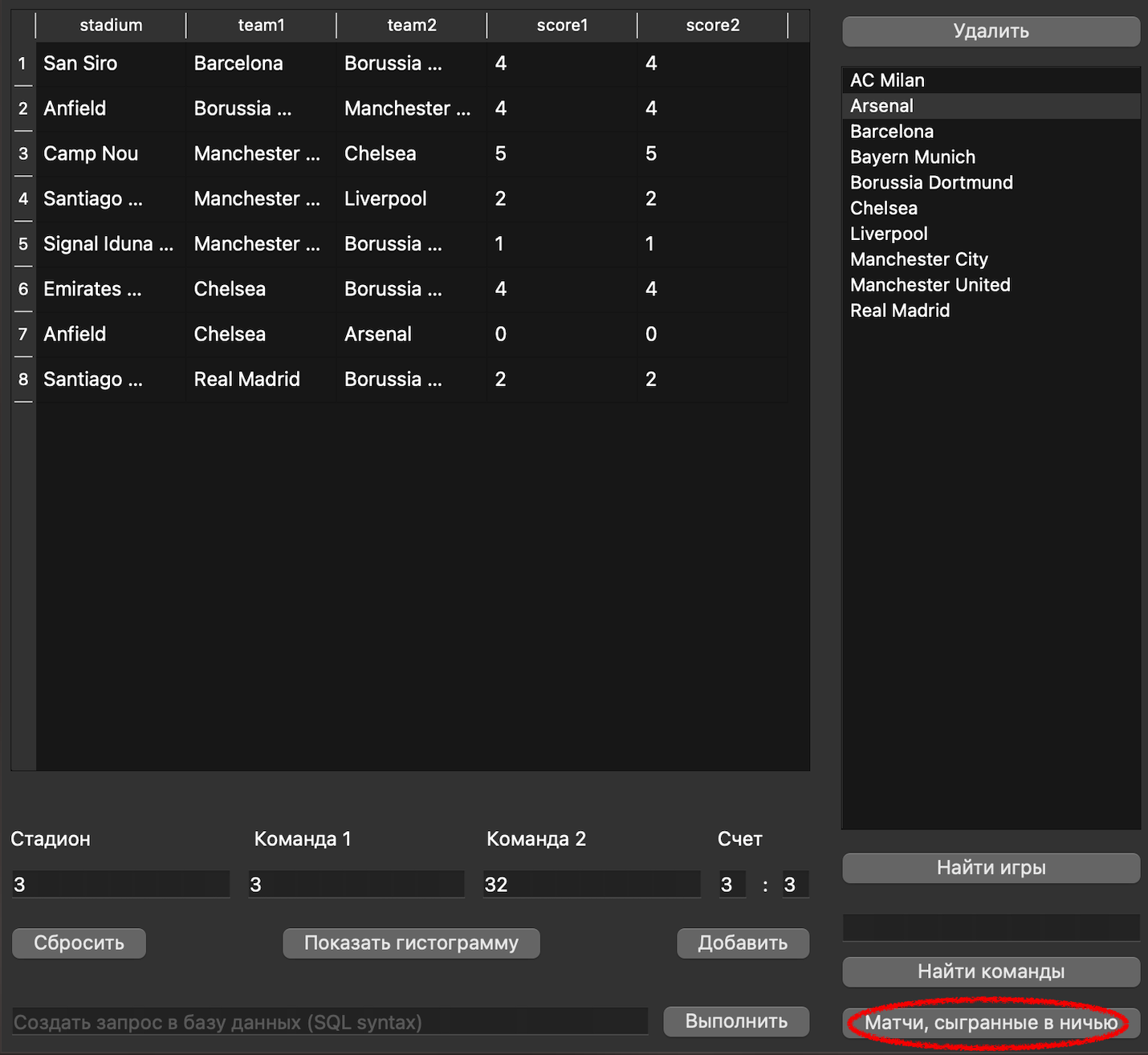
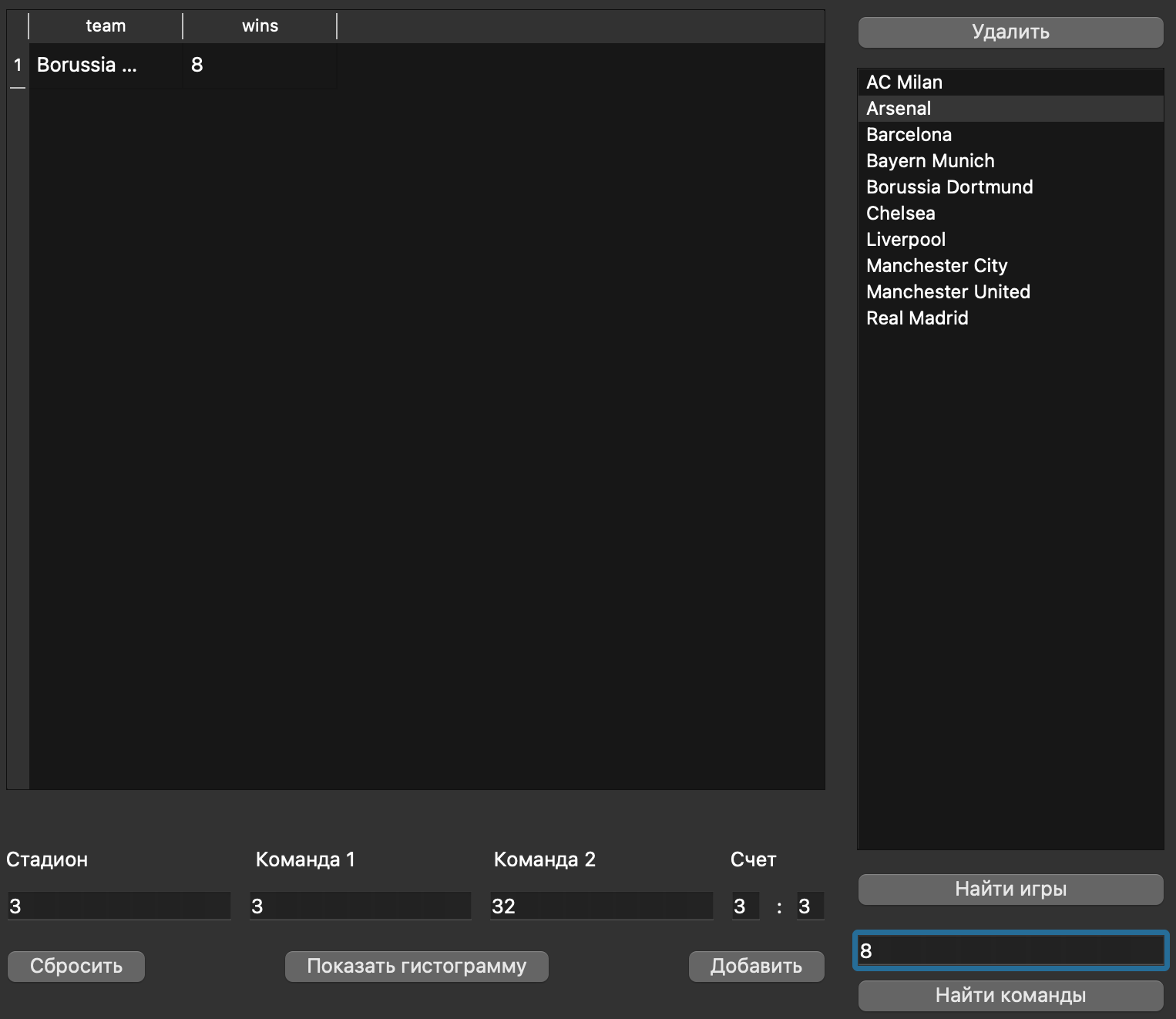
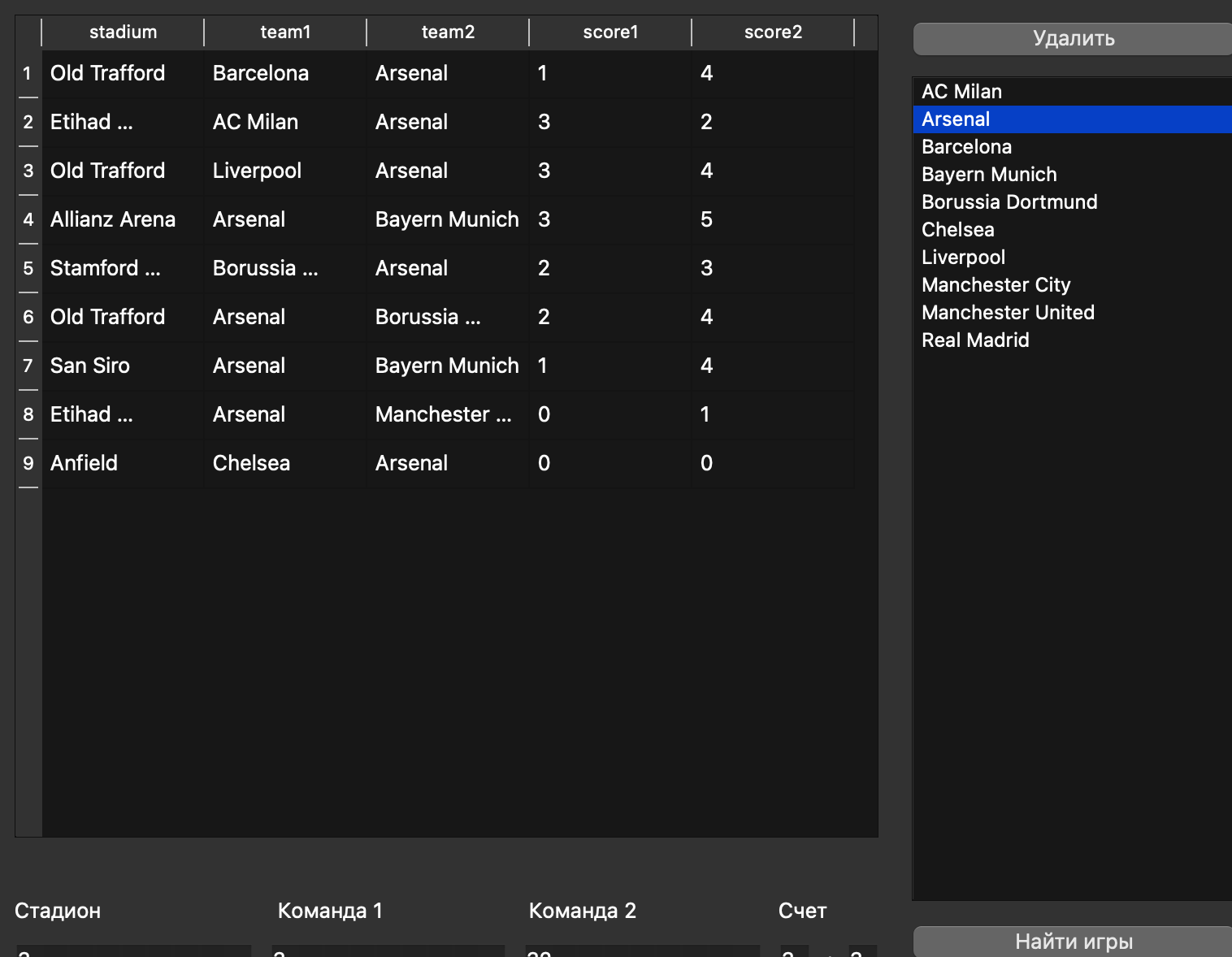
</widget>

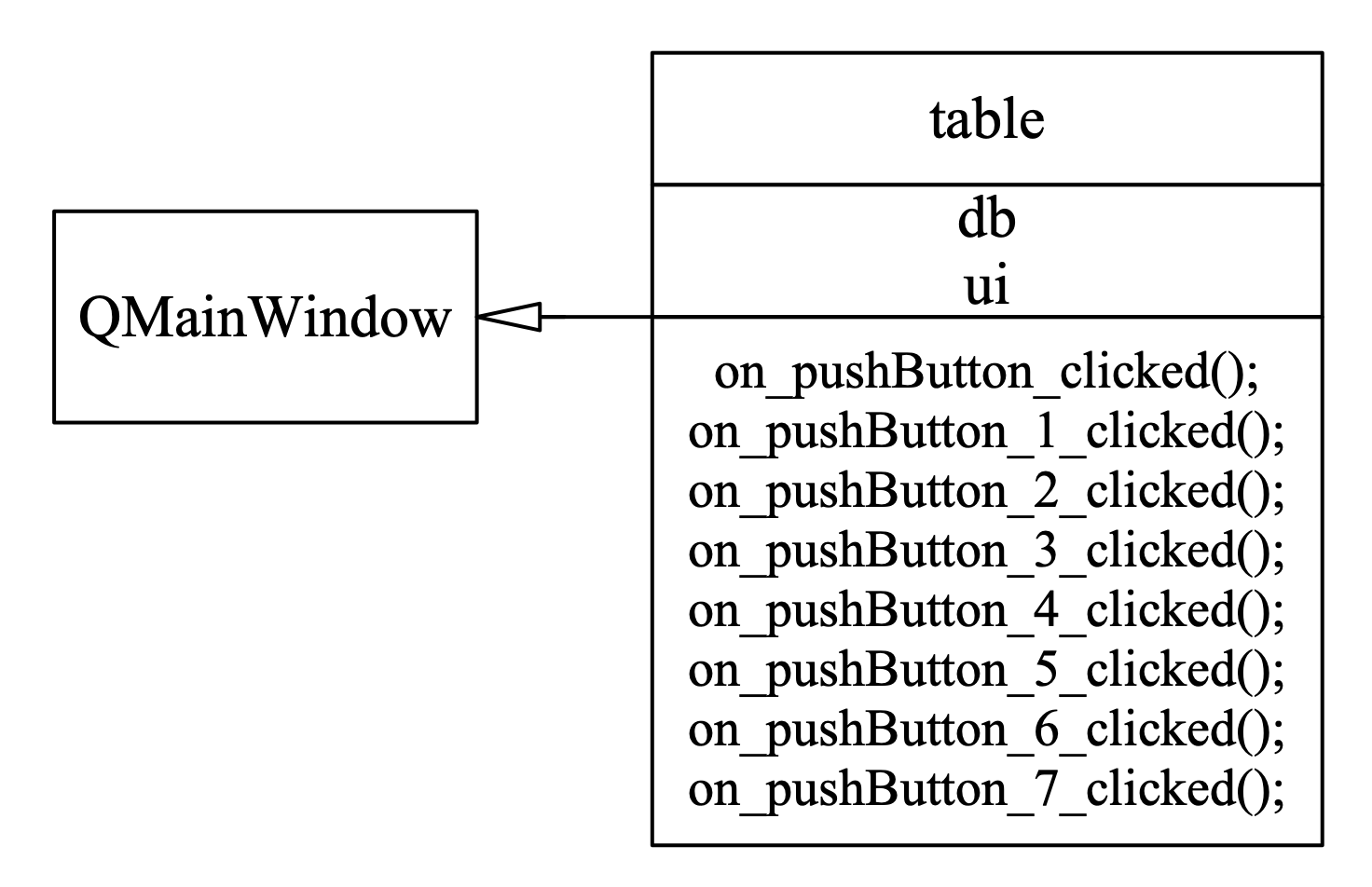
</widget>

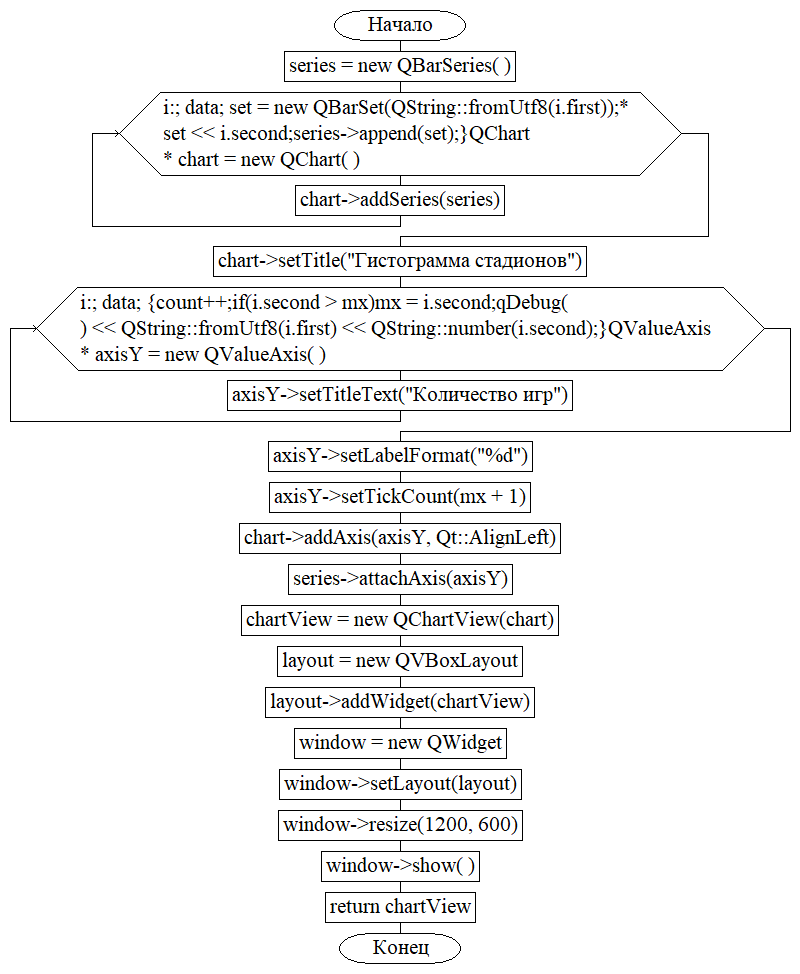
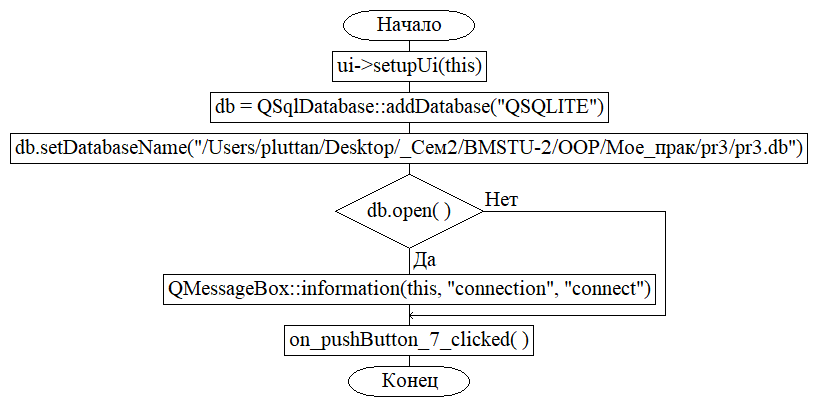
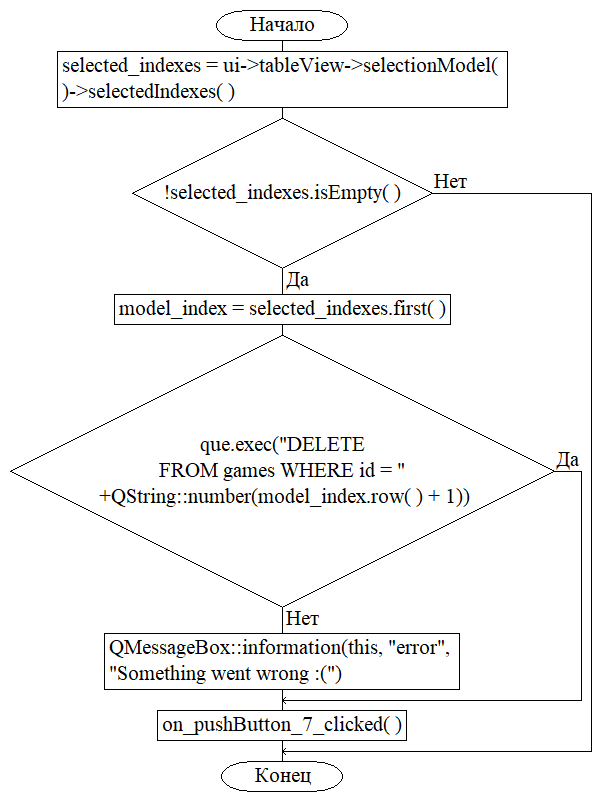
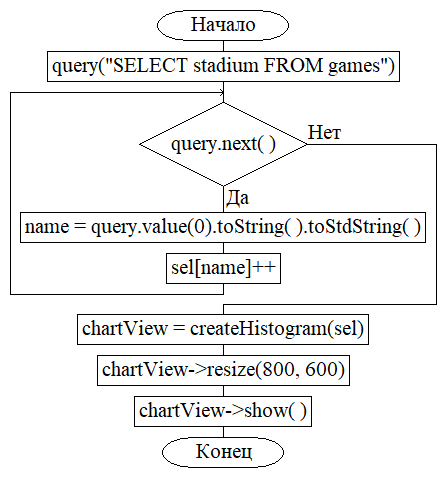
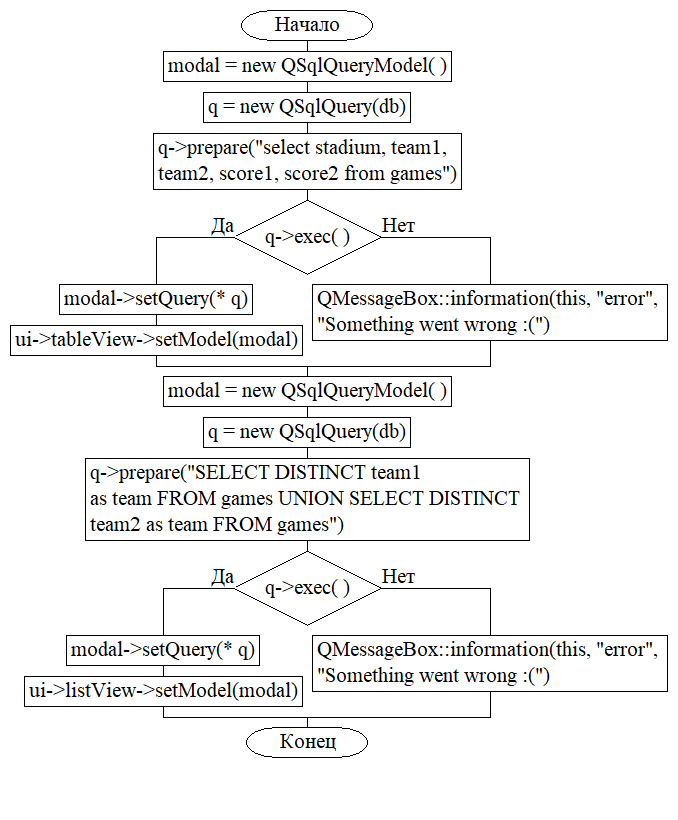
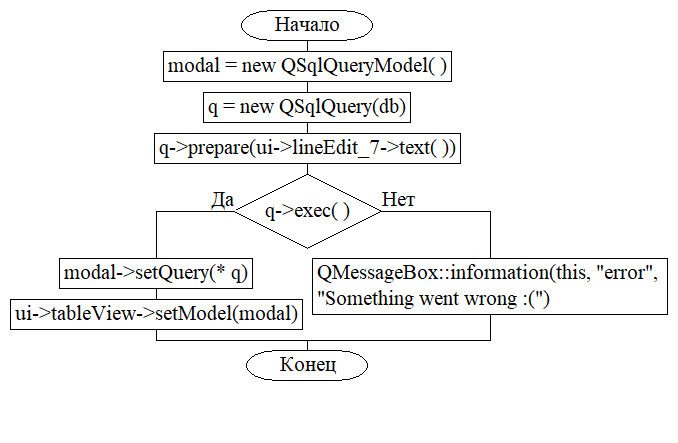
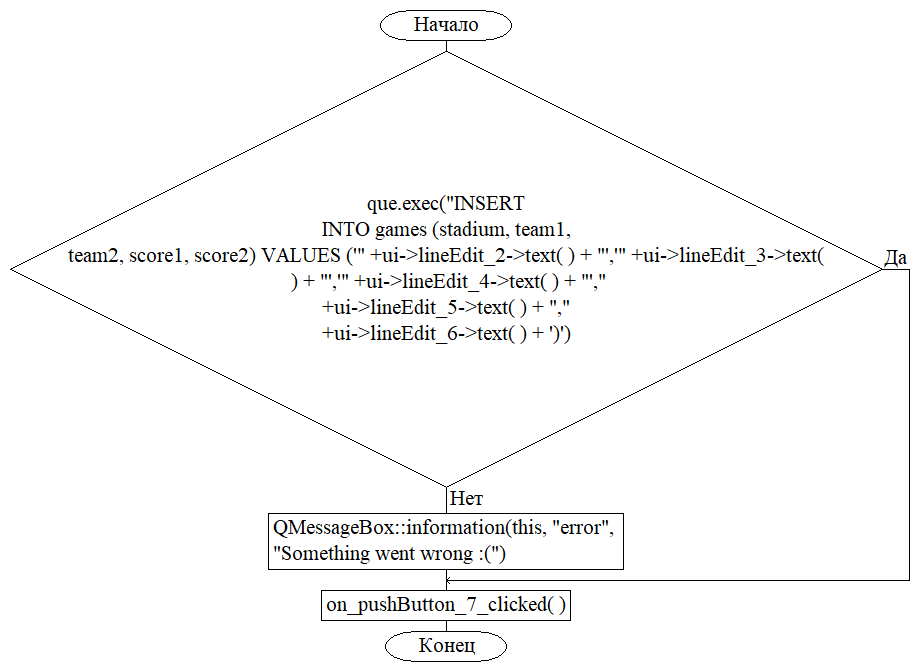
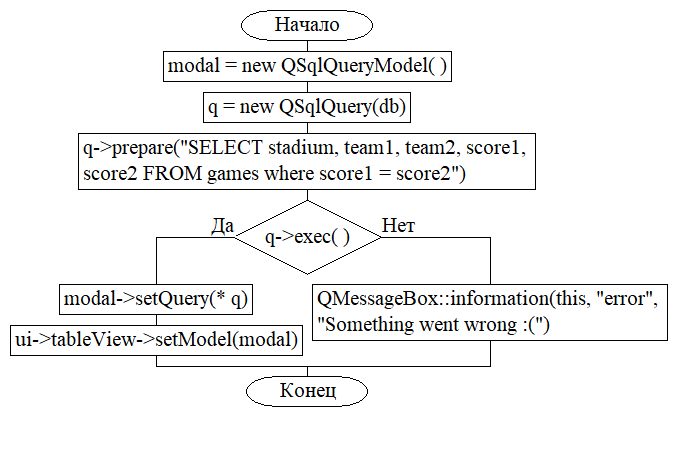
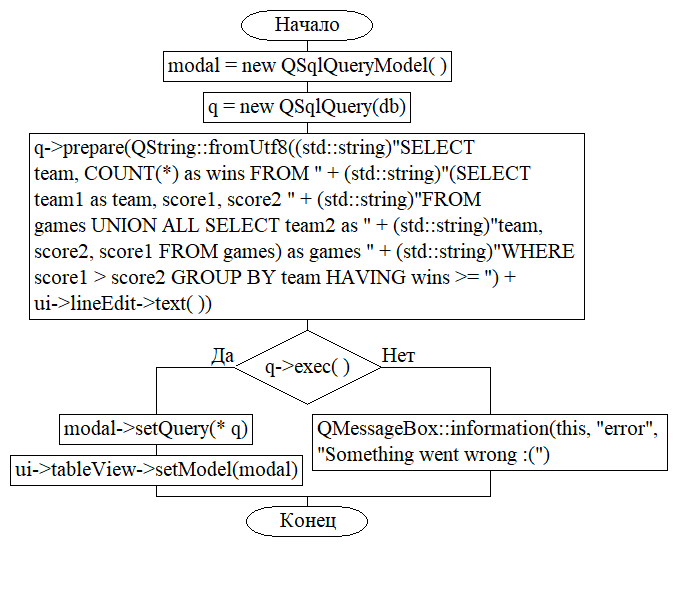
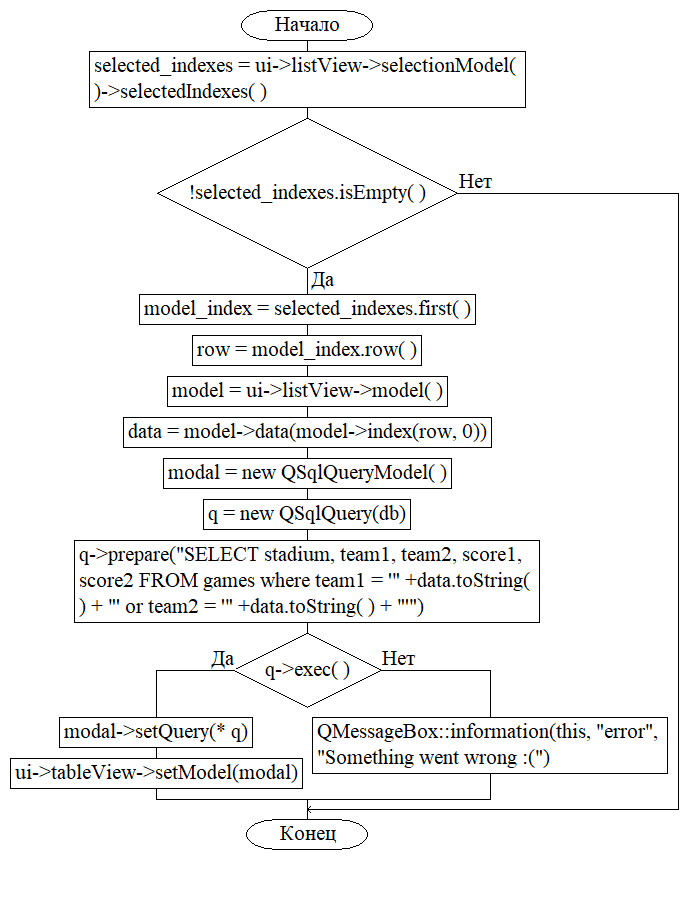
<resources/>

<connections/>

</ui>







Вывод: я научился работать с базами данных в Qt.2