Министерство образования и науки Российской Федерации

ФГАОУ ВО «Уральский Федеральный Университет имени первого Президента России Б. Н. Ельцина»

Институт радиоэлектроники и информационных технологий – РТФ

Оценка работы

\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Члены комиссии

\_\_\_\_\_\_\_\_\_\_\_\_\_\_

БАЗА ДАННЫХ С ИСПОЛЬЗОВАНИЕМ ACCESS, SQL И SQL SERVER  
НА ТЕМУ «КИБЕРСПОРТ»

Пояснительная записка к проекту  
по модулю «Базы данных и интеллектуальные системы»

09.03.02 0000 002 ПЗ

РуководительП. В. Добряк

Студент гр. РИ-360005М. А. Куприянов

# РЕФЕРАТ

Куприянов М. А. база данных с использованием Access, SQL и SQL Server на тему «киберспорт».

Пояснительная записка, 22 стр., 28 рис., 5 источников, 1 прил.

БАЗЫ ДАННЫХ, SQL, SQL SERVER, ACCESS, КИБЕРСПОРТ

Аннотация: в данной работе была создана база данных на тему киберспорт, написаны запросы к базе данных, триггеры, созданы формы презентации базы данных в Access, написаны хранимые процедуры, реализована клиентская программа на языке программирования высокого уровня C#.

# СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 4](#_Toc11269004)

[1. ПОСТАНОВКА ЗАДАЧИ 5](#_Toc11269005)

[2. ДИАГРАММА СУЩНОСТЬ-СВЯЗЬ ЧЕНА 6](#_Toc11269006)

[3. СХЕМА БД ACCESS 7](#_Toc11269007)

[4. СОДЕРЖИМОЕ БАЗЫ ДАННЫХ В ACCESS 8](#_Toc11269008)

[4.1. Запросы 8](#_Toc11269009)

[4.2. Формы 14](#_Toc11269010)

[5. СОДЕРЖИМОЕ БАЗЫ ДАННЫХ В SQL SERVER 17](#_Toc11269011)

[5.1. Структура базы данных 17](#_Toc11269012)

[5.2. Работа с базой данных 18](#_Toc11269013)

[6. OLAP (СХЕМА, КУБ) 19](#_Toc11269014)

[ЗАКЛЮЧЕНИЕ 20](#_Toc11269015)

[БИБЛИОГРАФИЧЕСКИЙ СПИСОК 21](#_Toc11269016)

[ПРИЛОЖЕНИЕ А 22](#_Toc11269017)

# ВВЕДЕНИЕ

Данная работа – это подведение итогов и создание полного отчета о выполнении работ связанных с базами данных за весь период обучения в течение семестра по предмету “Управление данными”. Создание баз данных предусматривает создание схем структуры баз данных, реализации запросов к базам данных, для эффективного использования содержимого баз, создание форм и клиентских программ.

# ПОСТАНОВКА ЗАДАЧИ

Цель работы: реализовать базу данных, которая будет описывать и содержать данные о киберспортивных командах в определенной дисциплине.

Для достижения поставленной цели потребуется решить следующие задачи:

1. Изучить теорию построения схемы баз данных
2. Изучить непроцедурный язык программирования SQL
3. Изучить правила работы в реляционной системе управления базами данных Access
4. Изучить правила работы в системе управления реляционными базами данных SQL Server
5. Изучить принципы создание клиентской программы на основе баз данных

# ДИАГРАММА СУЩНОСТЬ-СВЯЗЬ ЧЕНА

Данная база данных должна учитывать особенности существующей модели киберспорта. Киберспортивные команды профессионального уровня всегда состоят в организациях, такие организации могут иметь у себя несколько профессиональных команд. Цель таких организаций – это получение прибыли с призовых фондов киберспортивных соревнований и со спонсорских программ. Для достижения данных целей организации обеспечивютт комфортные условия для своих игроков и осуществляют разного вида рекламу для спонсора. Организации привязаны к определенным странам так как являются юридическими лицами, а значит имеют юридический адрес. Игрок как правило может состоят в любой команде и всегда имеет свои особенности, которые должны выделять его среди других профессиональных игроков даже в своей команде. Ниже представлена диаграмма сущность-связь Чена, описывающая данную модель.

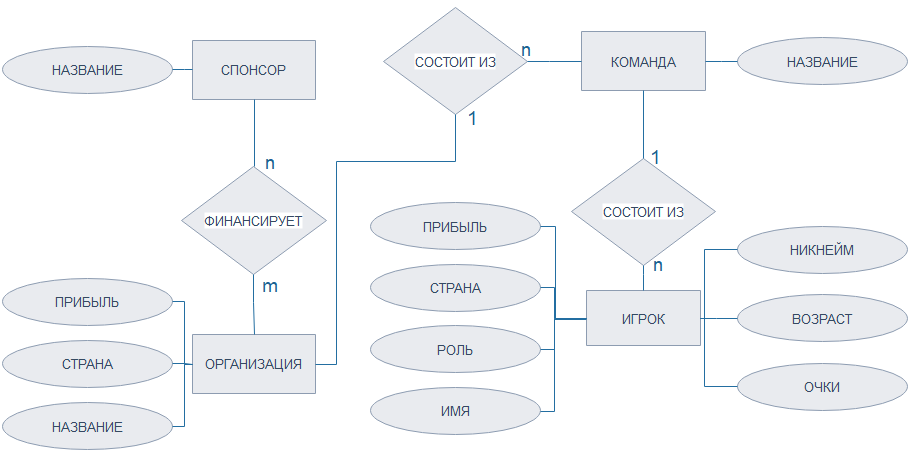


Диаграмма 1 – Сущность-свзять Чена

# СХЕМА БД ACCESS

По представленной выше диаграмме была составлена схема реляционной базы данных в системе управления базами данных Access.

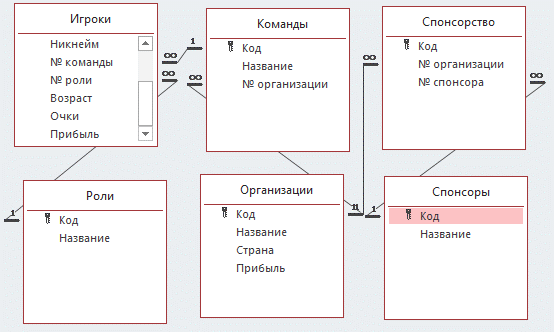


Схема 1 – Схема базы данных Access

# СОДЕРЖИМОЕ БАЗЫ ДАННЫХ В ACCESS

## Запросы

Ниже буду приведены запросы с кодом SQL и результатом работы этого кода (рисунки 1 – 16) на разработанной базе данных.

Запрос с Any

SELECT DISTINCT Спонсоры.Название

FROM Спонсоры, Игроки

WHERE (((Игроки.Прибыль)>1000000) AND ((Игроки.[№ команды])=Any (SELECT Команды.Код

FROM Команды, Организации, Спонсорство

WHERE Организации.Код = Команды.[№ организации]

AND Организации.Код = Спонсорство.[№ организации]

AND Спонсоры.Код = Спонсорство.[№ спонсора])));

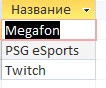


Рисунок 1 - Результат выполнения запроса с Any

Запрос с Exists

SELECT Название

FROM Спонсоры

WHERE not exists (SELECT \* FROM Спонсорство WHERE Спонсоры.Код = Спонсорство.[№ спонсора]);



Рисунок 2 - Результат выполнения запроса с Exists

Запрос с Left join Игроки и команды

SELECT Игроки.Имя, Игроки.Никнейм, Игроки.Фамилия, Команды.Название

FROM Команды RIGHT JOIN Игроки ON Команды.Код = Игроки.[№ команды];

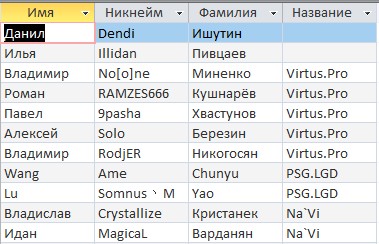


Рисунок 3 - Результат выполнения запроса с Left join Игроки и команды

Вложенный запрос с From

SELECT [%$##@\_Alias].Название

FROM (SELECT [№ организации], Название, COUNT(Спонсорство.Код) AS Количество\_спонсорств

FROM Спонсорство, Организации

WHERE Организации.Код = Спонсорство.[№ организации]

GROUP BY [№ организации], Название) AS [%$##@\_Alias]

WHERE ((([%$##@\_Alias].[Количество\_спонсорств])>1));



Рисунок 4 - Результат выполнения вложенного запроса с From

Запрос с группировкой

SELECT Игроки.Страна, Sum(Игроки.Прибыль) AS [Sum-Прибыль]

FROM Игроки

GROUP BY Игроки.Страна;

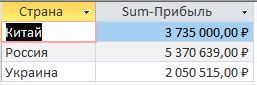


Рисунок 5 - Результат выполнения запроса с группировкой

Запрос с Select

SELECT Фамилия, Никнейм

FROM Игроки;



Рисунок 6 - Результат выполнения запроа с Select

Запрос с WHERE AND

SELECT Никнейм

FROM Игроки

WHERE Прибыль > 1000000

AND Очки > 2000;

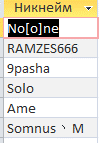


Рисунок 7 - Результат выполнения запроса с WHERE AND

Запрос с Having

SELECT A.Никнейм, A.Прибыль

FROM Игроки AS A

GROUP BY A.Страна, A.Никнейм, A.Прибыль

HAVING A.Прибыль > (SELECT avg(B.Прибыль)

FROM Игроки B WHERE A.Страна = B.Страна);

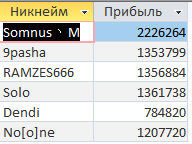


Рисунок 8 - Результат выполнения запроса с Having

Запрос с вложенным Select

SELECT Игроки.Никнейм, Avg(Очки)-(SELECT avg(Очки) FROM Игроки) AS Карьера

FROM Игроки

GROUP BY Игроки.Никнейм;



Рисунок 9 - Результат выполнения запроса с вложенным Select

Запрос с псевдонимом

SELECT DISTINCT B.Название

FROM Организации AS A, Организации AS B, Спонсорство AS C, Спонсорство AS D

WHERE A.Название = 'Virtus.pro' And B.Название<>'Virtus.pro' And A.Код = C.[№ организации] And B.Код = D.[№ организации] And C.[№ спонсора] = D.[№ спонсора];



Рисунок 10 - Результат выполнения запроса с псевдонимом

Запрос – ‘Игроки’ без подчиненных в ‘Команды’

SELECT Игроки.Имя, Игроки.Никнейм, Игроки.Фамилия

FROM Игроки LEFT JOIN Команды ON Игроки.[№ команды] = Команды.[Код]

WHERE (((Команды.Код) Is Null));

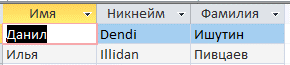


Рисунок 11 - Результат выполнения запроса – ‘Игроки’ без подчиненных в ‘Команды’

Запрос с поиском повторений возрастов игроков

SELECT Игроки.Возраст, Игроки.Никнейм

FROM Игроки

WHERE (((Игроки.Возраст) In (SELECT [Возраст] FROM [Игроки] As Tmp GROUP BY [Возраст] HAVING Count(\*)>1 )))

ORDER BY Игроки.Возраст;

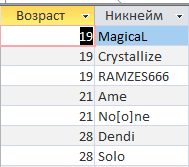


Рисунок 12 - Результат выполнения запроса с поиском повторений возрастов игроков

Запрос на селекцию по роли ‘мидер’

SELECT Игроки.Никнейм

FROM Роли INNER JOIN Игроки ON Роли.Код = Игроки.[№ роли]

WHERE (((Роли.Код)=1));

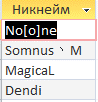


Рисунок 13 - Результат выполнения запроса на селекцию по роли ‘мидер’

Запрос на сортировку игроков по Роли

SELECT Игроки.Никнейм, Роли.Название

FROM Роли INNER JOIN Игроки ON Роли.Код = Игроки.[№ роли]

ORDER BY Роли.Код;



Рисунок 14 - Результат выполнения запроса на сортировку игроков по Роли

Запрос с Union

SELECT Никнейм as Лицо, Прибыль

FROM Игроки

UNION SELECT Название as Лицо, Прибыль

FROM Организации

ORDER BY Прибыль;

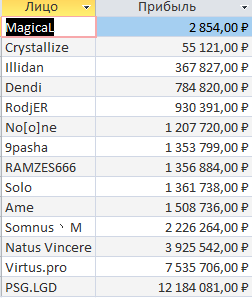


Рисунок 15 - Результат выполнения запроса с Union

Перекрестный запрос

TRANSFORM Sum(Игроки.Прибыль) AS [Sum-Прибыль]

SELECT Игроки.Страна, Sum(Игроки.Прибыль) AS [Итоговое значение Прибыль]

FROM Игроки

GROUP BY Игроки.Страна

PIVOT Игроки.Возраст;

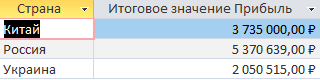


Рисунок 16 - Результат выполнения перекрестного запроса

## Формы

Ниже буду приведены формы для базы данных и запросов к ней (рисунки 17 – 20) на разработанной базе данных.

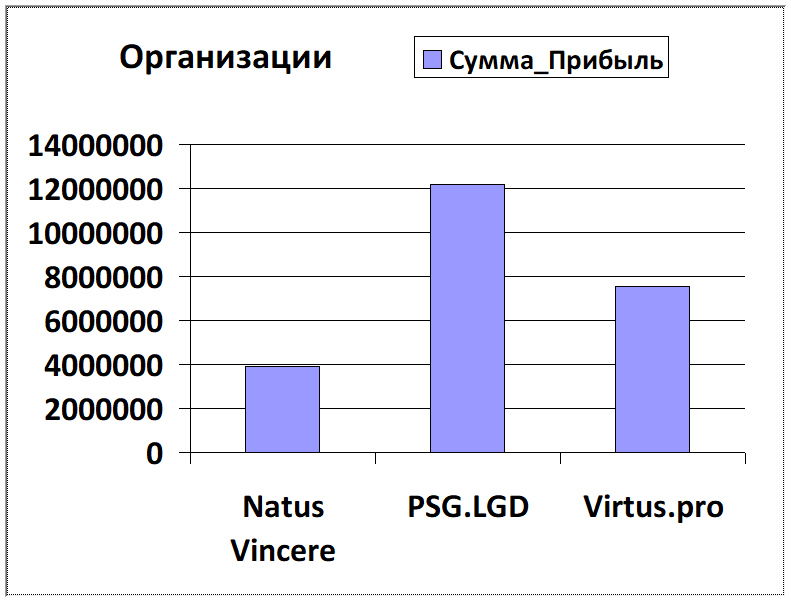


Рисунок 17 – Форма вывода прибыли организаций

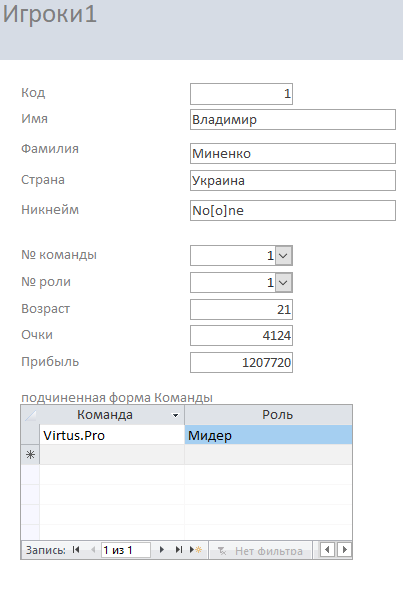


Рисунок 18 – Форма вывода информации по игроку

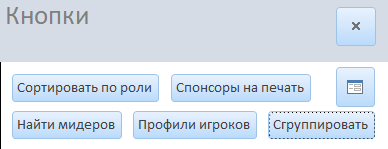


Рисунок 19 – Форма с кнопками для выполнения запросов

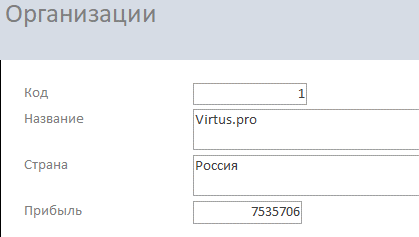


Рисунок 20 – Форма вывод информации об организации

# СОДЕРЖИМОЕ БАЗЫ ДАННЫХ В SQL SERVER

## Структура базы данных

Ниже приведены структуры таблиц (рисунки 21 – 25) в системе управления SQL Server.

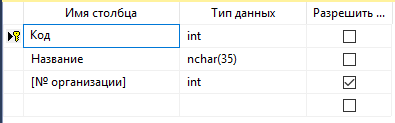


Рисунок 21 – Таблица команд

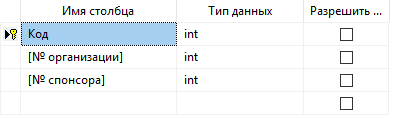


Рисунок 22 – Таблица спонсорств

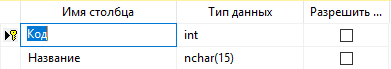


Рисунок 23 – Таблица ролей

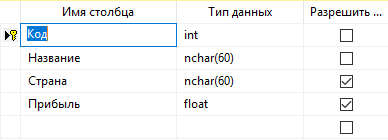


Рисунок 24 – Таблица организаций

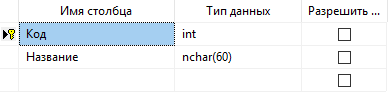


Рисунок 25 – Таблица спонсоров

## Работа с базой данных

К этой базе данных была создана простая пользовательская программа с выводом таблиц (рисунок 26), исходный код будет приведен в приложении А к проекту.

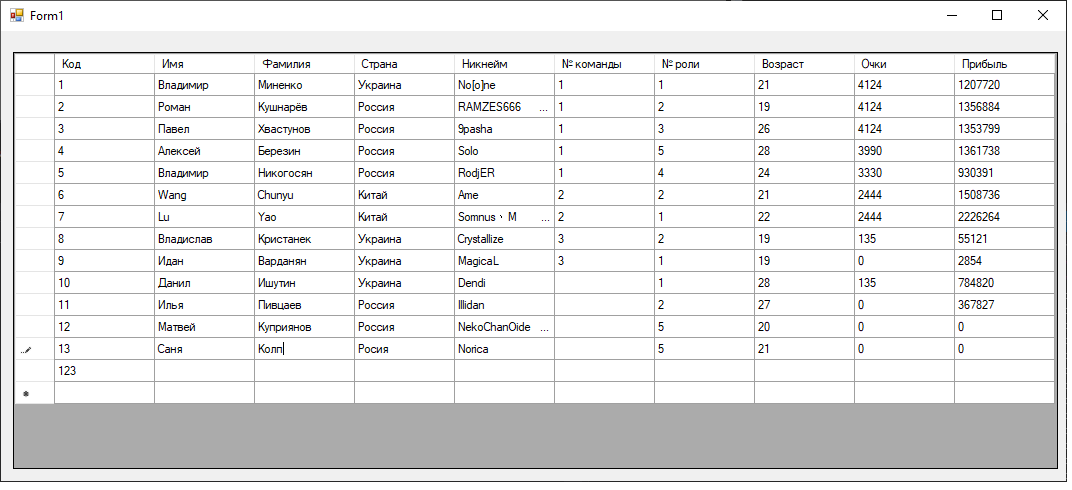


Рисунок 26 – Пользовательская программа

В базе был также реализован триггер (рисунок 27).

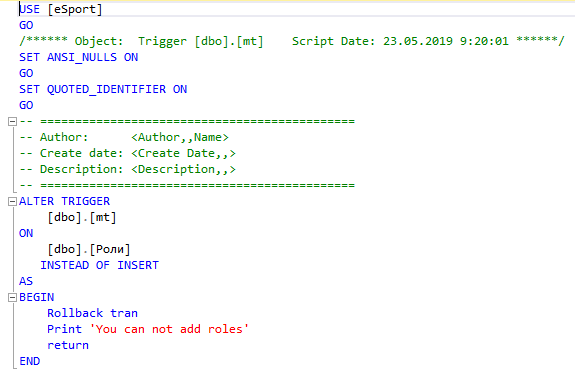


Рисунок 27 - Триггер

# OLAP (СХЕМА, КУБ)

Схема и куб для OLAP представлены на схеме 2 и рисунке 28.

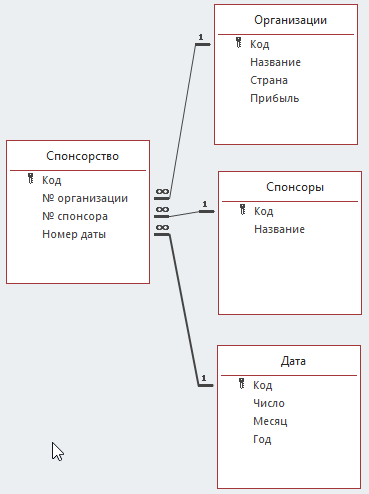


Схема 2 – Схема OLAP

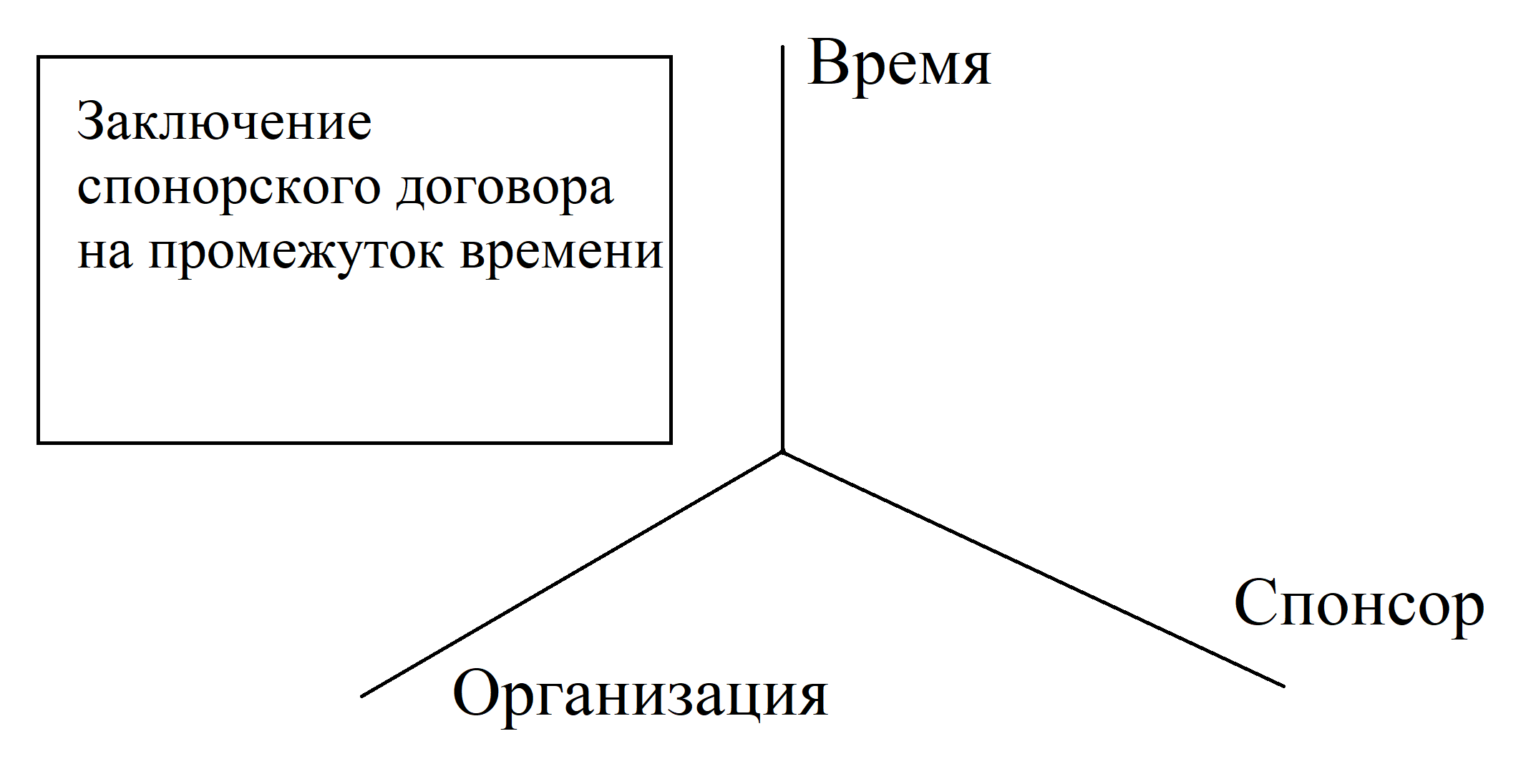


Рисунок 28 – Трехмерный куб OLAP

# ЗАКЛЮЧЕНИЕ

Данная работа представляет собой реализацию полноценной системы базы данных, которая связана с хранением данных о киберспорте, а также содержит примеры запросов к этой базе данных и выполнения анализа по ней.

Была проведена работа в различных средствах работы с базами данных, в том числе для создания локальной базы данных, создания сервера с базой данных и создания клиентского приложения для работы с базой данных.

Рассмотрены и применены на практике теоретические знания в построении и реализации схем релятивистских баз данных, таких как диаграмма Сущность-связь Чена, схема в Access и схема OLAP.

# БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. SQL – язык программирования [Электронный ресурс] – Уровень доступа: https://www.learnsqlonline.org/
2. S# – язык программирования [Электронный ресурс] – Уровень доступа: https://www.learncs.org/
3. Добряк Павел Вадимович. Управление данными./ П. В. Добряк. Екатеринбург : УГТУ-УПИ, 2014.-138 с.
4. Online analytical processing – статья энциклопедии [Электронный ресурс] – Уровень доступа:  
   https://en.wikipedia.org/wiki/Online\_analytical\_processing
5. Ulrich Drepper. What Every Programmer Should Know About Memory./ D. Ulrich. Red Hat, Inc, 2007.-114 с.

# ПРИЛОЖЕНИЕ А

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApp1

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void dataGridView1\_CellContentClick(object sender,

DataGridViewCellEventArgs e)

{

}

private void Form1\_Load(object sender, EventArgs e)

{

/\* TODO: данная строка кода позволяет загрузить данные в таблицу

"dataSet1.Игроки". При необходимости она может быть перемещена или

удалена.\*/

this.игрокиTableAdapter.Fill(this.dataSet1.Игроки);

}

}

static class Program

{

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

}

}