# Министерство образования и науки Украины

## Харьковский национальный университет радиоэлектроники

Кафедра информатики

# **Пояснительная записка**

# **к курсовой работа**

# **по дисциплине**

# **“Организация баз данных и информационных систем ”**

# **Тема:“Информационная система компьютерного клуба** ”

Выполнил: ст. гр. xxxxxx

xxxxxxx

Проверила:

Работа защищена с оценкой:

Харьков 2008

# **РЕФЕРАТ**

# Целью данного курсового проекта является разработка актуальной, автоматизированной информационно-справочной системы хранения и обработки информации справочной системы компьютерного клуба, которая способствовала бы упрощению администрирования компьютера в клубе, могла вести наблюдение за каждым компьютером, игроком, администратором, а так же находить необходимые данные при минимальных затратах времени. Курсовая работа состоит из введения, четырех частей, заключения, списка литературы, содержащего \_ источников. Содержит \_ страницы,\_ рисунков, \_ таблиц, \_ SQL запросов, \_ приложения .

# Во введении описывается область применения задачи и перечислены существующие методы ее решения в рамках единого комплекса, где информационной основой служит использование концепции баз данных, реализация которой в среде реляционных моделей считается перспективной. В курсовой работе принят метод решения задачи от построения инфологической схемы предметной области, которая затем отображается в логическую схему базы данных с последующей ее оптимизацией (нормализацией).

Практическая реализация выполнена с использованием СУБД ACCESS и Delphi. Приведены программы построения отношений и запросов в среде реляционных моделей с использованием языка SQL.

# В заключении подводится итог проделанной работы и дается критический анализ полученных результатов. Приложение содержит распечатки общего вида форм .

Ключевые слова и выражения:

БАЗАДА ННЫХ, ДИАГРАММА СУЩНОСТЬ-СВЯЗЬ, ОТНОШЕНИЕ, ДАТАЛОГИЧЕСКАЯ МОДЕЛЬ, ИНФОЛОГИЧЕСКАЯ МОДЕЛЬ, ПОТЕНЦИАЛЬНЫЕ КЛЮЧИ, ССЫЛОЧНАЯ ЦЕЛОСТНОСТЬ, АТРИБУТ, КОРТЕЖ, ФУНКЦИОНАЛЬНАЯ ЗАВИСИМОСТЬ, ПЕРВИЧНЫЙ КЛЮЧ, ВНЕШНИЙ КЛЮЧ, НОРМАЛИЗАЦИЯ, НОМАЛЬНЫЕ ФОРМЫ, ЯЗЫК ЗАПРОСОВ, СУБД, COM-технологии, DELPHI.

# **СОДЕРЖАНИЕ**

Задание

Календарный план

РЕФЕРАТ

ВВЕДЕНИЕ

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1Построение инфологической модели предметной области методом ER- диаграммы

1.2Описание диаграммы «сущность-связь»

2 ПОСТАНОВКА ЗАДАЧИ

3 РАЗРАБОТКА СХЕМЫ БАЗЫ ДАННЫХ

3.1Описание логической схемы базы данных

4 РАЗРАБОТКА ПРОЕКТА ПРИЛОЖЕНИЯ

4.1Описание данных

4.2 Создание отношений БД с помощью языка SQL

4.3 Заполнение базы данных

4.4 Создание запросов к базе данных компьютерного клуба

4.5 Создание отчетов по базам данных с помощью приложений Office

4.5.1 Создание генерации отчетов с помощью Microsoft Excel

4.5.1 Создание отчета с помощью Microsoft Word

ЗАКЛЮЧЕНИЕ

СПИСОК ЛИТЕРАТУРЫ

ПРИЛОЖЕНИЕ А

ПРИЛОЖЕНИЕ Б

ПРИЛОЖЕНИЕ Б

**Введение**

В современном мире информационных технологий информационно-справочные системы с использованием концепций баз данных нашли широкое применение. Любая отрасль не может развиваться без сотрудничества с информатикой, так как предприятиям необходимо обрабатывать огромное количество разнообразной информации, и без применения и информационно-справочных систем это было бы невозможным. Отсутствие хранилища данных и возможности организации быстрого к ним доступа, поиска и их обработки значительно затруднило бы работу и развитие крупных предприятий. Еще недавно бумажной базой данных можно было обойтись. Но сегодня без электронных баз данных, не обойтись, потому что процесс улучшения жизни общества требует увеличения объема необходимой информации ежегодно на 5-7%. А уже через десять объем необходимой информации удвоится.

Занесение базы данных на ПК дает большое преимущество уже для однопользовательской системы – это компактность, скорость, низкие трудозатраты и актуальность.

Базы данных – это компьютеризированная система хранения записей , то есть компьютеризированная система, назначение которой хранить информацию, предоставляя пользователям средства ее извлечения идентификации.

Поскольку БД (база данных) – хранилище определенно города информации, то для управления или анализа их содержимого применяется программное обеспечение, позволяющее установить связь между пользователями системы и физическими записями данных. Таким программным обеспечением является СУБД – системы управления базами данных. Основная задача СУБД – предоставление возможности пользователю базы данных работать с ней, не вникая с детали ее архитектуры.

Термин СУБД так же часто используются в отношении конкретных программных продуктов, одним из которых является Microsoft Access, который и был использован при выполнении данной курсовой работы.

Этот программный продукт предлагает механизм формирования таблиц и достаточно прост в применении, обеспечивая этим пользователю – непрофессионалу комфорт при работе с базой данных.

База данных обеспечивает хранение информации, а также удобный и быстрый доступ к данным. Она представляет собой совокупность данных различного характера, организованных по определенным правилам.

Используется так же современные технологии программирования, базирующиеся на технологии COM и позволяющие использовать приложения Microsoft Office в качестве серверов автоматизации.

COM является платформно-независимой, объектно-ориентированной технологией, позволяющей создавать бинарные компоненты. Эти компоненты можно использовать как локально, так и в распределенном сетевом окружении. COM служит основой для: OLE (технология составных документов), ActiveX-объектов и элементов управления ActiveX, DCOM, COM+.

На базе COM создано большинство новейших продуктов (MS Office, MTS, …) и технологий Windows (Automation, Drag & Drop, ...).

COM – это технология, позволяющая объектам взаимодействовать, несмотря на границы процесса или машины, так же легко, как и объектам внутри одного процесса. COM обеспечивает такое взаимодействие, определяя, что единственный путь управления данными, ассоциированными с объектом, лежит через интерфейс объекта. Термин «интерфейс», о котором речь пойдет чуть ниже, означает реализацию в коде COM-совместимого двоичного интерфейса, ассоциированного с объектом.

Генерация отчетов по базам данных с помощью приложений Microsoft Office применяется на практике довольно часто. Причиной этого является желание пользователей получать отчеты в виде файла одного из стандартных форматов, иметь возможность редактировать их и обмениваться ими с клиентами и партнерами. Дело в том, что имеющиеся в комплекте поставки Delphi компоненты QuickReport обладают весьма ограниченными возможностями, связанными с созданием отчетов в одном из стандартных форматов, и применение вместо них приложений Microsoft Office, как правило, уже имеющихся в компании, нередко предпочтительнее приобретения профессиональных генераторов отчетов типа Crystal Reports (Crystal Decisions).

Таким образом, путем автоматизации приложений Microsoft Office мы мо­жем создавать сложные отчеты, содержащие таблицы, диаграммы, вычисляемые значения, равно как и использовать для этой цели другие возможности Microsoft Office (например, сервисы построения сводных таблиц).

**1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ**

* 1. **Построение инфологической модели предметной области методом ER- диаграммы**

Начальным шагом в проектировании базы данных является постановка задачи, описание предметной области и построении инфологической модели данных, то есть модели «сущность - связь», известной еще как диаграмма Чена.

Она помогает логическому программированию РБД. В логической схеме РБД семантика данных передается посредством первичных и внешних ключей и функциональных зависимостей. Для построения диаграммы сущность/связь прежде всего необходимо:

* определить данные, которые будут использоватся РБД;
* определить множество отношений в которых будут связаны эти данные;
* вывести правила целостности и определить связи для работы с отношениями;

ER-диаграмма строится с помощью таких основных правил:

* cвойство всегда связано с одним определенным отношением, (если только отношения не являются подчиненными общему типу). В диаграмме свойства обозначаются овалами;
* если отношение имеет связи, то между их связью должна быть описана логическая привязка обозначенная ромбом. Отношения обозначаются прямоугольниками.

Все связи должны быть обязательно вида «один ко многим» или «многие к одному».

# Отношения в модели сущность-связь могут иметь тип один-к-одному, один-ко-многим, многие-к-одному, многие-ко-многим (для упрощения изложения далее предполагается, что все отношения бинарные).

**1.2 Описание диаграммы «сущность-связь» для информационной системы компьютерного клуба**

Сильными объектами являются Центр Данных, Компьютер, Услуги, Инфо игрока, Администратор, Модернизация, Устройства.

# Свойствами объекта Данные являются Номер компьютера, Дата аренды, Время аренды, Продолжительность, Код услуги, Игрок, Цена, Администратор.

# Свойствами объекта Компьютер являются Номер компьютера, Платформа, Частота, Монитор, Видеокарта, Жесткий диск, Мышь, Клавиатура, Наушники.

# Свойствами объекта Услуги являются - Номер Услуги, Название, Тариф.

# Свойствами объекта Инфо Игрока являются – Игрок, ФИО, Адрес, Команда.

# Свойствами объекта Администратор являются – Администратор, ФИО, Адрес, Доход.

# Инфологическая (ER-диаграмма) предметной области показана на рис. 1.1

**2 ПОСТАНОВКА ЗАДАЧИ**

Разрабатываемая информационная система компьютерного игрового клуба должна быть удобной в использовании, как для Администратора, обслуживающего компьютер, так и бухгалтера, директора клуба, налогового инспектора, а так же любого желающего игрока. Она дает возможность получить полную информацию о точной дате, времени, оказываемой услуге для определённого компьютера, посетителя или обслуживающего администратора.

Система должна помочь работникам компьютерного клуба в ведении отчетов об аренде компьютеров посетителями, в поиске нужной информации об игроках, ведении статистики посещения определённого игрока. Также, помогает администратору без потери времени определить дату дежурства, дневную и общую прибыль.

Данная информационная система должна решать задачу о нахождении подходящего компьютера соответствующего уровня в зависимости от услуги, которую выбирает посетитель.

Система дает возможность администратору вести учет о составляющих устройствах компьютера, что позволяет проводить своевременную модернизацию, а так же подсчет расхода на неё.

При построении информационной системы известно следующее: компьютер, его платформа, частота, монитор, установленные на него жесткий диск, видеокарта, манипуляторы, наушники , администратор и информация о нём, услуги, их описание и тариф. Данные, занесенные в таблицу, со временем могут обновляться.

**3 РАЗРАБОТКА СХЕМЫ БАЗЫ ДАННЫХ**

**3.1 Описание логической схемы базы данных**

В представляемой базе данных семь отношений. Все отношения находятся в НФБК, так как каждая доминанта (атрибут функционально определяющий другой атрибут) является потенциальным ключом.

Описание логической схемы базы данных:

Отношение «Данные» имеет атрибуты: Номер\_компа, Дата\_аренды, Время\_аренды, Продолжительность, Код\_услуги, Игрок, Цена, Админ.

Отношение «Услуги» имеет атрибуты: Номер\_услуги, Название, Тариф.

Отношение «Инфо игрока» имеет атрибуты: Игрок, ФИО, Адрес, Команда.

Отношение «Админ» имеет атрибуты: Админ, ФИО, Адрес, Ставка.

Отношение «Девайс» имеет атрибуты: Устройство, Магазин, Фирма, Цена.

Отношение «Комп» имеет атрибуты: Номер\_компа, Платформа, MHz, Monitor, Video, Hdd, Mouse, Keyboard, Headphones .

Схема данных в Microsoft Access изображена на Рис. 3.1

**4 РАЗРАБОТКА ПРОЕКТА ПРИЛОЖЕНИЯ**

**4.1 Описание данных БД**

В таблице 4.1 приведено описание данных БД «Работа клуба». Таблица 4.1 - Описание данных БД «Работа клуба»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Элемент данных | Описание | Объект | Тип данных | Условие на значеие |
| Номер\_компа | Номер компьютера | Центр данных | Текстовый | Обязательное (ключ) |
| Дата\_Аренды | Дата аренды компьютера | Центр данных | Дата/Время | Обязательное  (ключ) |
| Время\_Аренды | Время аренды компьютера | Центр данных | Дата/Время | Обязательное  (ключ) |
| Продолжительность | Часы аренды компьютера | Центр данных | Числовой | Обязательное |
| Код\_Услуги | Код услуги | Центр данных | Текстовый | Обязательное |
| Игрок | Посетитель клуба | Центр данных | Текстовый | Обязательное |
| Цена | Стоимость | Центр данных | Денежный | Обязательное |
| Админ | Администратор | Центр данных | Текстовый | Обязательное (ключ) |
| Номер\_услуги | Кодовый номер услуги | Услуги | Текстовый | Обязательное (ключ) |
| Название | Описание услуги | Услуги | Текстовый | Обязательное |
| Тариф | Стоимость услуги | Услуги | Денежный | Обязательное  (ключ) |
| Админ | «Никнейм» администратора | Админ | Текстовый | Обязательное  (ключ) |
| ФИО | ФИО | Админ | Текстовый | Обязательное |
| Адрес | Адрес | Админ | Текстовый | Обязательное |
| Ставка | Зарплата | Админ | Текстовый | Обязательное |
| Игрок | «Никнейм» игрока | Инфо игрока | Текстовый | Обязательное(ключ) |
| ФИО | ФИО игрока | Инфо игрока | Текстовый | Обязательное |
| Адрес | Адрес игрока | Инфо игрока | Текстовый | Обязательное |
| Команда | Команда игрока | Инфо игрока | Текстовый | Необязательное |

**4.2 Создание отношений с помощью языка SQL**

В языке SQL оператором создания таблиц является оператор CREATE TABLE.

Ниже приводится построение таблиц БД «Компьютерный клуб» с помощью оператора CREATE TABLE.

### CREATE TABLE Админ

( Админ VARCHAR (50) NOT NULL,

ФИО VARCHAR (50) NOT NULL,

Адрес VARCHAR (50) NOT NULL,

Ставка NUMBER NOT NULL,

PRIMARY KEY (Админ),

FOREIGN KEY (Админ),

REFERENCES Центр данных,

ON DELETE CASCADE,

ON UPDATE CASCADE)

### CREATE TABLE Инфо игрока

( Игрок VARCHAR (50) NOT NULL,

ФИО VARCHAR (50) NOT NULL,

Адрес VARCHAR (50) NOT NULL,

Комманда VARCHAR (50) NOT NULL,

PRIMARY KEY (Игрок),

ON DELETE CASCADE,

ON UPDATE CASCADE)

### CREATE TABLE Девайс

( Устройство VARCHAR (50) NOT NULL,

Магазин VARCHAR (50) NOT NULL,

Фирма VARCHAR (50) NOT NULL,

Цена VARCHAR (50) NOT NULL,

PRIMARY KEY (Устройство),

FOREIGN KEY (Устройство),

REFERENCES Модернизация,

ON DELETE CASCADE,

ON UPDATE CASCADE)

### CREATE TABLE Услуги

(Номер\_услуги VARCHAR (10) NOT NULL,

Название VARCHAR (50) NOT NULL,

Тариф MONEY NOT NULL,

PRIMARY KEY (Номер\_услуги),

ON DELETE CASCADE,

ON UPDATE CASCADE)

### CREATE TABLE Комп

(Номер\_компа VARCHAR (50) NOT NULL,

Платформа VARCHAR (50) NOT NULL,

MHz VARCHAR (50) NOT NULL,Monitor

Video VARCHAR (50) NOT NULL,

Hdd VARCHAR (50) NOT NULL,

Mouse VARCHAR (50) NOT NULL,

Keyboard VARCHAR (50) NOT NULL,

Headphones VARCHAR (50) NOT NULL,

PRIMARY KEY (Номер\_компа),

ON DELETE CASCADE,

ON UPDATE CASCADE)

### CREATE TABLE Центр данных

(Номер\_компа VARCHAR (50) NOT NULL,

Дата\_аренды DATE NOT NULL,

Время\_аренды DATE NOT NULL,

Продолжительность NUMBER NOT NULL,

Код\_услуги VARCHAR (50) NOT NULL,

Игрок VARCHAR (50) NOT NULL,

Цена MONEY NOT NULL,

Админ VARCHAR (50) NOT NULL,

PRIMARY KEY (Номер\_компа, Дата\_аренды, Время\_аренды, Продолжительность, Админ),

FOREIGN KEY (Код\_услуги),

REFERENCES Услуги,

ON DELETE CASCADE,

ON UPDATE CASCADE,

FOREIGN KEY ( Игрок)

REFERENCES Инфо игрока,

ON DELETE CASCADE,

ON UPDATE CASCADE,

FOREIGN KEY (Номер\_компа),

REFERENCES Комп,

ON DELETE CASCADE,

ON UPDATE CASCADE,

)

**4.3 Заполнение базы данных**

Данные заносятся с использованием форм) приведенных в приложении. При занесении данных следует обратить внимание на поддержание ограничений целостности по первичным и внешним ключам отношений согласно схеме базы данных,приведенной на рис.4.2.

Рекомендуется вначале установить связи в схеме базы данных, а затем заносить данные, что предотвратит ошибки, возникающие при попытке внести данные, нарушающие ограничения целостности в базу данных и связанные с их идентификацией.

Особое внимание следует уделить занесению NULL-значений. В теоретическом плане NULL-значения запрещены (значения атрибута атомарны), однако требования практики обусловили их использование в коммерческих пакетах. Следует учитывать разницу в описании NULL-значений в каждом из них. В разрабатываемом проекте NULL-значения задавались средствами пакета ACCESS.

**4.4 Создание запросов к базе данных компьютерный клуб**

В основном запросы используются для анализа данных. В данной курсовой работе были созданы такие запросы:

1. Данные о дежурстве админа:

SELECT Данные.Дата\_аренды, Данные.админ

FROM Комп INNER JOIN (Услуги INNER JOIN ([Инфо игрока] INNER JOIN Данные ON [Инфо игрока].Игрок=Данные.Игрок) ON Услуги.Номер\_услуги=Данные.Код\_услуги) ON Комп.Номер\_компа=Данные.Номер\_компа

WHERE (((Данные.Дата\_аренды)=[ Введите дату аренды ]));

2.Доход в заданную дату.

SELECT Sum([Центр данных].Цена) AS Сумма

FROM [Центр данных]

WHERE ((([Центр данных].Дата\_аренды)=[ Введите дату аренды ]));

3.Вывести игроков, которые не состоят в командах

SELECT [Инфо игрока].игрок, [Инфо игрока].фио, [Инфо игрока].адрес

FROM [инфо игрока]

WHERE [Инфо игрока].комманда Is Null;

4.Вывести информацию, если заработано более 5 грн.

SELECT [Центр данных].игрок, [Центр данных].Цена, [Центр данных].Номер\_компа, [Центр данных].Дата\_аренды, [Центр данных].Код\_услуги, [Центр данных].Админ

FROM [Центр данных]

GROUP BY [Центр данных].игрок, [Центр данных].Цена, [Центр данных].Номер\_компа, [Центр данных].Дата\_аренды, [Центр данных].Код\_услуги, [Центр данных].Админ

HAVING ((([Центр данных].Цена)>5));

5.Вывести данные о том, когда и на каком компьютере пользовались определённой услугй.

SELECT Услуги.Название, [Центр данных].Номер\_компа, [Центр данных].Дата\_аренды, [Центр данных].Время\_аренды, [Центр данных].Продолжительность

FROM Комп INNER JOIN (Услуги INNER JOIN ([Инфо игрока] INNER JOIN [Центр данных] ON [Инфо игрока].Игрок=[Центр данных].Игрок) ON Услуги.Номер\_услуги=[Центр данных].Код\_услуги) ON Комп.Номер\_компа=[Центр данных].Номер\_компа

WHERE (((Услуги.Номер\_услуги)=[ Код услуги ]))

ORDER BY [Центр данных].Дата\_аренды;

6.Вывести сумму, заработанную определённым администратором

SELECT Sum([Центр данных].Цена) AS Сумма

FROM [Центр данных]

WHERE ((([Центр данных].админ)=[ Администратор ]));

7.Вывести данные о работе клуба, отсортировать по дате и игроку

SELECT [Центр данных].Номер\_компа, [Центр данных].Дата\_аренды, [Центр данных].Время\_аренды, [Центр данных].Продолжительность, [Центр данных].Код\_услуги, [Центр данных].Игрок, [Центр данных].Цена, [Центр данных].Админ

FROM [Центр данных]

ORDER BY [Центр данных].Дата\_аренды, [Центр данных].Игрок;

8.Вывести информацию о самой поздней аренде компьютера

SELECT MAX([центр данных].время\_аренды) AS Время

FROM [центр данных];

9.Вывести информацию о самой ранней аренде компьютера

SELECT MIN([центр данных].время\_аренды) AS Время

FROM [центр данных];

10.Вывести данные об аренде компьютера в заданное время

SELECT [Центр данных].Время\_аренды, [Центр данных].Код\_услуги, [Центр данных].Игрок, [Центр данных].Номер\_компа

FROM Комп INNER JOIN (Услуги INNER JOIN ([Инфо игрока] INNER JOIN [Центр данных] ON [Инфо игрока].Игрок=[Центр данных].Игрок) ON Услуги.Номер\_услуги=[Центр данных].Код\_услуги) ON Комп.Номер\_компа=[Центр данных].Номер\_компа

WHERE ((([Центр данных].Время\_аренды)=[ Введите время аренды ]));

11.Вывести данные об аренде компьютера в заданную дату.

SELECT [Центр данных].Дата\_аренды, [Центр данных].Код\_услуги, [Центр данных].Игрок, [Центр данных].Номер\_компа

FROM Комп INNER JOIN (Услуги INNER JOIN ([Инфо игрока] INNER JOIN [Центр данных] ON [Инфо игрока].Игрок=[Центр данных].Игрок) ON Услуги.Номер\_услуги=[Центр данных].Код\_услуги) ON Комп.Номер\_компа=[Центр данных].Номер\_компа

WHERE ((([Центр данных].дата\_аренды)=[ Введите дату аренды ]));

12.Вывести данные об услугах

SELECT Услуги.Номер\_услуги, Услуги.Название, Услуги.Тариф

FROM Комп INNER JOIN (Услуги INNER JOIN ([Инфо игрока] INNER JOIN [Центр данных] ON [Инфо игрока].Игрок=[Центр данных].Игрок) ON Услуги.Номер\_услуги=[Центр данных].Код\_услуги) ON Комп.Номер\_компа=[Центр данных].Номер\_компа

GROUP BY Услуги.Номер\_услуги, [Центр данных].Код\_услуги, Услуги.Название, Услуги.Тариф

HAVING (("номер\_компа"="номер\_компа"));

**4.5 Создание отчетов по базам данных с помощью приложений Office**

Созданное приложение должно позволять создать отчеты по базе данных club.mdb, входящую в комплект стандартной поставки пакета Microsoft Office.

**4.5.1 Создание отчета с помощью Microsoft Word. Откройте новое приложение в Delphi**

Поместить на форму компонент TADOConection, два компонента TADODataSet и три компонента TButton ( рис.4.3). Свойствам Caption и Name присвоить значения из таблицы 1.

Таблица 4.1

|  |  |  |
| --- | --- | --- |
| Компонент | Свойство Caption | Свойство Name |
| Tbutton1 | Отчет с использованием MS Word | Report\_Word |
| Tbutton2 | Отчет с использованием MS Excel | Report\_Excel |

Свойству ConnectionString объекта TADODataSet установить значение, которое связывало бы его с базой данных club.mdb .

Объявить глобальные переменные для объектов Application, Document, Range и Table:

Var Forml: TForm1;

// Range и Table

Wd, Doc, Rng, Tbl: Variant;

Свойствам CommandType и CommandText компонентов присвоить значения из таблицы 4.2.

Таблица 4.2

|  |  |  |
| --- | --- | --- |
| Компонент | Свойство CommandType | Свойство CommandText |
| TADODataSet1 | cmdText | Select КодКлиента, Название, Город From Страны |

Реализовываем генерацию табличного отчета в обработчике события OnClick компонента Buttonl:

procedure TForm1.Report\_WordClick(Sender: TObject);

var I, Rcnt:integer;

begin

// Открываем базу данных и подсчитываем количество записей

ADODataSet1.Open;

Rcnt:=ADODataSet1.RecordCount;

// Запускаем MS Word

wd:= CreateOleObject('Word.Application');

// Отображаем на экране окно MS Word

wd.Visible :=True;

// Создаем новый документ

wd.Documents.Add;

Doc:= wd.Documents.Item(1);

// Добавляем новый абзац

Doc.Paragraphs.Add;

// Меняем его стиль

Doc.Paragraphs.Item(1).Style:='Заголовок 1';

// Создаем заголовок отчета

Rng := Doc.Range(0);

Rng.InsertBefore('Центр данных');

// Создаем заголовки колонок

Doc.Paragraphs.Add;

Rng.InsertAfter('Комп: Игрок: Дата: Продолжительность: Код услуги: Админ');

// Перемещаемся на первую запись набора данных

ADODataSet1.First;

for I := 1 to Rcnt do begin

// Добавляем новый абзац

Doc.Paragraphs.Add;

// Добавляем поля из текущей записи в новый абзац

Rng.InsertAfter(ADODataSet1.Fields[0].AsString+':'+

ADODataSet1.Fields[1].ASString+':'+

ADODataSet1.Fields[2].AsString+':'+

ADODataSet1.Fields[3].ASString+':'+

ADODataSet1.Fields[4].ASString+':'+

ADODataSet1.Fields[5].ASString);

ADODataSet1.Next;

end;

// Превращаем текст в таблицу

Rng:=Doc.Range(Doc.Paragraphs.Item(3).Range.Start, Doc.Paragraphs.Item(Rcnt+3).Range.End);

Tbl:=Rng.ConvertToTable(':',Rcnt,6);

// Изменяем размеры колонок таблицы

Tbl.Columns.Item(1).Width:=Tbl.Columns.Item(1).Width-30;

Tbl.Columns.Item(2).Width:=Tbl.Columns.Item(2).Width+20;

Tbl.Columns.Item(3).Width:=Tbl.Columns.Item(3).Width-3;

Tbl.Columns.Item(4).Width:=Tbl.Columns.Item(4).Width+20;

Tbl.Columns.Item(5).Width:=Tbl.Columns.Item(5).Width-5;

Tbl.Columns.Item(6).Width:=Tbl.Columns.Item(6).Width+5;

// Подавляем вывод диагностических сообщений

Wd.DisplayAlerts:=False;

// Сохраняем документ

Doc.SaveAs(D:\SOT\Custrep.doc');

// Закрываем Word и освобождаем ресурсы

//Wd.Quit;

Wd:=Unassigned;

end;

Откомпилируйте приложение и проверьте работу кнопки «Отчет с использованием MS Word». При выполнении этого приложения создается документ с отчетом по базе данный club.mdb (рис.4.4).

Пояснения к приведенному выше фрагменту кода.

Во-первых, мы должны создать копию Microsoft Word, сделать ее видимой и создать новый документ.

wd:= CreateOleObject('Word.Application');

// Отображаем на экране окно MS Word

wd.Visible :=True;

// Создаем новый документ

wd.Documents.Add;

Затем нужно создать заголовок отчета и заголовки колонок будущей таблицы, добавляя соответствующие абзацы и меняя их стили:

// Добавляем новый абзац

Doc.Paragraphs.Add;

// Меняем его стиль

Doc.Paragraphs.Item(1).Style:='Заголовок 1';

// Создаем заголовок отчета

Rng := Doc.Range(0);

Rng.lnsertBefore(Центр данных');

// Создаем заголовки колонок

Doc.Paragraphs.Add;

Rng.InsertAfter('Комп: Игрок: Дата: Продолжительность: Код услуги: Админ');

Затем следует, перемещаясь по записям набора данных, добавить в документ строки, соответствующие этим записям:

// Перемещаемся на первую запись набора данных

ADODataSet1.First;

for I := 1 to Rcnt do begin

// Добавляем новый абзац

Doc.Paragraphs.Add;

// Добавляем поля из текущей записи в новый абзац

Rng.InsertAter(ADODataSet1.Fields[0].AsString+':'+

ADODataSet1.Fields[1].ASString+':'+

ADODataSet1.Fields[2].AsString+':'+

ADODataSet1.Fields[3].ASString+':'+

ADODataSet1.Fields[4].ASString+':'+

ADODataSet1.Fields[5].ASString);

ADODataSet1.Next;

end;

Далее мы превращаем набор строк в таблицу Word и изменяем ширину ее колонок так, чтобы корректно отобразить содержащиеся в ней данные:

// Превращаем текс в таблицу

Rng:=Doc.Range(Doc.Paragraphs.Item(3).Range.Start, Doc.Paragraphs.Item(rcnt+3).Range.End); Tbl:=Rng.ConvertToTable(':',rcnt.3);

// Изменяем размеры колонок таблицы

Tbl.Columns.Item(1).Width:=Tbl.Columns.Item(1).Width-30;

Tbl.Columns.Item(2).Width:=Tbl.Columns.Item(2).Width+20;

Tbl.Columns.Item(3).Width:=Tbl.Columns.Item(3).Width-3;

Tbl.Columns.Item(4).Width:=Tbl.Columns.Item(4).Width+20;

Tbl.Columns.Item(5).Width:=Tbl.Columns.Item(5).Width-5;

Tbl.Columns.Item(6).Width:=Tbl.Columns.Item(6).Width+5;Теперь нам нужно сохранить документ, подавив при этом вывод диагностических сообщений Word:

// Подавляем вывод диагностических сообщений

Wd.DisplayAlerts:=False;

Почему нужно избавиться от вывода диагностических сообщений? В общем случае приложения, подобные Word, можно запускать удаленно, например, с помощью средств DCOM или универсальных СОМ-клиентов, доступ к которым осуществляется по протоколам TCP/IP или HTTP/HTTPS. В этом случае у пользователя не будет возможности отвечать на вопросы диалоговых окон Word — ведь физически он находитсяна другом компьютере; кроме того, некоторые режимы применения DCOM таковы, что пользовательский интерфейс DCOM-сервера (включая обработку им событий мыши и клавиатуры) может быть просто недоступен никому из пользователей. В этом случае диалоговое окно, созданное в оперативной памяти, получит сообщения о событии, связанном со щелчком на одной из кнопок этого окна, и не будет закрыто, и у пользователя создастся впечатление, что приложение «зависло».

И, наконец, нам следует сохранить документ и освободить ресурсы:

// Сохраняем документ

Doc.SaveAs('D:\SOT\Custrep.doc');

// Закрываем Word и освобождаем ресурсы

Wd.Quit;

Wd:=Unassigned;

**4.5.2 Создание генерации отчетов с помощью Microsoft Excel**

Как и в предыдущем случае, нам следует также объявить глобальные переменные для объектов Application, WorkBook и WorkSheet:

var

Forml: TForm1;

. . .

// Переменные для объектов Excel Application, WorkBook и WorkSheet

Xl, Wb, Ws: Variant;

Реализуем генерацию табличного отчета в обработчике события OnClick компонента Button2:

procedure TForm1.Report\_ExcelClick(Sender: TObject);

var I, Rcnt:integer;

begin

// Открываем базу данных и подсчитываем количество записей

ADODataSet1.Open;

Rcnt := ADODataSet1.RecordCount;

// Запускаем Microsoft Excel

Xl := CreateOleObject('Excel.Application');

// Отображаем окно Microsoft Excel

Xl.Visible := True;

// Создаем рабочую книгу

Xl.WorkBooks.Add;

Wb := XL.WorkBooks[1];

Ws := Wb.WorkSheets[1];

Ws.Name := 'Центр данных';

// Создаем заголовок отчета

Ws.Cells[1,1] := 'Центр данных';

Ws.Cells[1,1].Font.Bold := True;

Ws.Cells[1,1].Font.Size := 16;

Ws.Cells[2,1] := 'Комп';

Ws.Cells[2,2] := 'Игрок';

Ws.Cells[2,3] := 'Дата';

Ws.Cells[2,4] := 'Продолжительность';

Ws.Cells[2,5] := 'Код услуги';

Ws.Cells[2,6] := 'Админ';

for I:=1 to 6 do

Ws.Cells[2,i].Font.Bold := True;

// Перемещаемся на первую запись набора данных

ADODataSet1.First;

for I:=1 to Rcnt do

begin

// Добавляем значения полей текущей записи в новую строку

Ws.Cells[i+2,1] := ADODataSet1.Fields[0].AsString;

Ws.Cells[i+2,2] := ADODataSet1.Fields[1].AsString;

Ws.Cells[i+2,3] := ADODataSet1.Fields[2].AsString;

Ws.Cells[i+2,4] := ADODataSet1.Fields[3].AsString;

Ws.Cells[i+2,5] := ADODataSet1.Fields[4].AsString;

Ws.Cells[i+2,6] := ADODataSet1.Fields[5].AsString;

ADODataSet1.Next;

end;

// Изменяем ширину колонок

Xl.Columns[1].ColumnWidth:=Xl.Columns[1].ColumnWidth+5;

Xl.Columns[2].ColumnWidth:=Xl.Columns[2].ColumnWidth+5;

Xl.Columns[3].ColumnWidth:=Xl.Columns[3].ColumnWidth+5;

Xl.Columns[4].ColumnWidth:=Xl.Columns[4].ColumnWidth+5;

Xl.Columns[5].ColumnWidth:=Xl.Columns[5].ColumnWidth+5;

Xl.Columns[6].ColumnWidth:=Xl.Columns[6].ColumnWidth+5;

// Подавляем вывод диагностических сообщений

Xl.DisplayAlerts:=false;

// Сохраняем документ

Wb.SaveAs('D:\SOT\Custrer.xls');

// Закрываем Excel и освобождаем ресурсы

//Xl.Quit;

Xl:=Unassigned;

end;

Откомпилируем приложение и проверьте работу кнопки «Отчет с использованием MS Excel». При выполнении этого приложения создается документ с отчетом по базе данный club.mdb (рис.4.5).

Пояснения к приведенному выше фрагменту кода

Во-первых, мы должны создать копию Microsoft Excel, сделав ее видимой и создать новую рабочую книгу:

// Запускаем Microsoft Excel

Xl := CreateOleObject('Excel.Application');

// Отображаем окно Microsoft Excel

Xl.Visible := True;

// Создаем рабочую книгу

Xl.WorkBooks.Add;

Wb := XL.WorkBooks[1];

Ws := Wb.WorkSheets[1];

Ws.Name := ' Список клиентов ';

Затем нужно создать заголовок отчета и заголовки колонок будущей таблицы, добавляя текст в соответствующие ячейки т меняя характеристики шрифта ячеек:

// Создаем заголовок отчета

Ws.Cells[1,1] := 'Центр данных';

Ws.Cells[1,1].Font.Bold := True;

Ws.Cells[1,1].Font.Size := 16;

Ws.Cells[2,1] := 'Комп';

Ws.Cells[2,2] := 'Игрок';

Ws.Cells[2,3] := 'Дата';

Ws.Cells[2,4] := 'Продолжительность';

Ws.Cells[2,5] := 'Код услуги';

Ws.Cells[2,6] := 'Админ';

for I:=1 to 6 do

Ws.Cells[2,i].Font.Bold := True;Затем следует, перемещаясь по записям набора данных, добавит в документ строки, соответствующие этим записям:

// Перемещаемся на первую запись набора данных

ADODataSet1.First;

for I:=1 to Rcnt do

begin

// Добавляем значения полей текущей записи в новую строку

Ws.Cells[i+2,1] := ADODataSet1.Fields[0].AsString;

Ws.Cells[i+2,2] := ADODataSet1.Fields[1].AsString;

Ws.Cells[i+2,3] := ADODataSet1.Fields[2].AsString;

Ws.Cells[i+2,4] := ADODataSet1.Fields[3].AsString;

Ws.Cells[i+2,5] := ADODataSet1.Fields[4].AsString;

Ws.Cells[i+2,6] := ADODataSet1.Fields[5].AsString;

ADODataSet1.Next;

end;

Как и в предыдущем случае, нам следует изменить размер колонок на листе рабочей книги, чтобы корректно отобразить содержащиеся в них данные:

// Изменяем ширину колонок

Xl.Columns[1].ColumnWidth:=Xl.Columns[1].ColumnWidth+5;

Xl.Columns[2].ColumnWidth:=Xl.Columns[2].ColumnWidth+5;

Xl.Columns[3].ColumnWidth:=Xl.Columns[3].ColumnWidth+5;

Xl.Columns[4].ColumnWidth:=Xl.Columns[4].ColumnWidth+5;

Xl.Columns[5].ColumnWidth:=Xl.Columns[5].ColumnWidth+5;

Xl.Columns[6].ColumnWidth:=Xl.Columns[6].ColumnWidth+5;

Теперь нам нужно сохранить документ, подавив при этом вывод диагностических сообщений Excel. Как и в Word, приложение Excel может быть запущено удаленно, и в этом случае пользователь также может не иметь возможности взаимодействовать с диалоговыми окнами Excel:

// Подавляем вывод диагностических сообщений

Xl.DisplayAlerts:=false;

// Сохраняем документ

Wb.SaveAs('D:\SOT\Custrep.xls');

Наконец, нам следует закрыть Excel и освободить ресурсы:

// Закрываем Excel и освобождаем ресурсы

Xl.Quit;

Xl:=Unassigned;

Добавление строки с новыми записями в базу данных.

procedure TForm1.Button6Click(Sender: TObject);

begin

with form1.ADOQuery2 do begin

//Close; // закрыть файл-результат выполнения предыдущего запроса

SQL.Clear; // удалить текст предыдущего запроса

// записываем новый запрос в свойство SQL

SQL.Add('INSERT INTO Админ (Админ, ФИО, Адрес, Ставка) VALUES ("Annie", "Бойко А.К.", "Засекречено:)", 1000);');

ExecSQL;

end;

Так же программа имеет возможность выполнять набранные запросы (Рис

Код программы приведен в Додатке В.

**ЗАКЛЮЧЕНИЕ**

В ходе данной курсовой работы была разработана автоматизированная информационно-справочная система хранения и обработки информации компьютерного клуба, которая способствует быстрому поиску необходимых данных при минимальных затратах времени, а также программная реализация системы, в основе которой лежит проектирование предметной области и логической схемы БД, являющейся информационным ресурсом разрабатываемой системы.

Для решения современных проблем применяются современные методы их решения. Проблема методов управления в организациях любых форм собственности предполагает использование системы управления базами данных для принятия решений на основе эффективного использования информационного ресурса, что позволяет получить большую экономическую отдачу и решать задачи планирования и управления с использованием современных информационных технологий.

Хорошо спроектированная система может расширяться с ростом информационных потребностей пользователей и при этом сохраняет уже существующую информацию. Современные методы проектирования предполагают в качестве первого этапа проектирования разработку инфологической модели предметной области. Разработанная в курсовом проекте инфологическая схема позволяет уяснить структуру предметной области с целью отображения ее логической схемы в базе данных.

При проектировании информационных систем использование реляционной модели базы данных является самым подходящим методом. Нормализация отношений разработанной базы данных позволила устранить ошибки внесения, удаления, обновления, дублирования данных, что особенно важно при работе с базой данных пользователей непрофессионалов. Практическая реализация информационной системы выполнена с использованием СУБД Access , однако приведенные в работе программы создания таблиц средствами SQL позволяют реализовать разработку в ином коммерческом пакете реляционного типа, где кроме языка запросов используется язык определения данных для создания базовых таблиц.

Таким образом, путем автоматизации приложений Microsoft Office мы можем создавать сложные отчеты, содержащие таблицы, диаграммы, вычисляемые значения, равно как и использовать для этой цели другие возможности Microsoft Office (например, сервисы построения сводных таблиц).

Разработанная база данных является законченным программным продуктом для поддержания информационных потребностей и может быть легко расширена при изменении информационных потребностей пользователя.

СПИСОК ЛИТЕРАТУРЫ

1. Буслик М.М. Модели и структуры данных.
2. Мартин Дж. Организация баз данных в вычислительных системах.-М.: Мир, 1980.-608с.
3. Глушаков С.В., Мельников И.В. Базы данных,включая SQL2 и SQL SERVER:Учебник для вузов.-Харьков.: Фолио,2000-250 с.
4. Вейскас Дж. Эффективная работа с Access.-СПб:Питер.1997-848с.
5. Боуман Дж.,Эмерсон С. Дарновски М. Практичесское руководство по SQL.-К:Диалектика,1998-565с.
6. Дейт К. Введение в системы баз данных.- 6-е изд.-К:Диалектика, 1998.-784с.
7. Холигвэрт, Джарод, Баттерфилд, Дэн, Сворт, Боб, и др. С++Builder 5. Руководство разработчика, том 2. Сложные вопросы программирования : Пер. с англ. – М.: Издательский дом “Вильямс”, 2001. - 832 с.: ил. – Парал. тит. англ.
8. Архангельский А.Я. Приемы программирования в Delphi. Изд. 2-е, пере раб. и доп. – М.: ООО «Бином-Пресс», 2004 г. – 848 с.: ил.

**Приложение В**

**Код программы**

unit Unit1;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, ComObj, ActiveX, DB, ADODB, StdCtrls, Grids, DBGrids, ComCtrls,

OleServer, ExcelXP;

type

TForm1 = class(TForm)

Report\_Word: TButton;

Report\_Excel: TButton;

CreatePivotTable: TButton;

ADOConnection1: TADOConnection;

ADODataSet1: TADODataSet;

PageControl1: TPageControl;

TabSheet1: TTabSheet;

TabSheet2: TTabSheet;

TabSheet3: TTabSheet;

ADOTable1: TADOTable;

DataSource1: TDataSource;

DBGrid1: TDBGrid;

RadioButton1: TRadioButton;

ComboBox1: TComboBox;

RadioButton2: TRadioButton;

ComboBox2: TComboBox;

DBGrid2: TDBGrid;

ADOQuery1: TADOQuery;

DataSource2: TDataSource;

Select: TMemo;

Button1: TButton;

DBGrid3: TDBGrid;

DataSource3: TDataSource;

TabSheet4: TTabSheet;

ADOQuery2: TADOQuery;

DataSource4: TDataSource;

DBGrid4: TDBGrid;

Button4: TButton;

ADODataSet3: TADODataSet;

Button5: TButton;

TabSheet5: TTabSheet;

DBGrid5: TDBGrid;

DataSource5: TDataSource;

Button6: TButton;

procedure Report\_WordClick(Sender: TObject);

procedure Report\_ExcelClick(Sender: TObject);

procedure CreatePivotTableClick(Sender: TObject);

procedure RadioButton1Click(Sender: TObject);

procedure ComboBox1Click(Sender: TObject);

procedure RadioButton2Click(Sender: TObject);

procedure ComboBox2Click(Sender: TObject);

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Button4Click(Sender: TObject);

procedure Button5Click(Sender: TObject);

procedure Button6Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

Wd, Doc, Rng, Tbl, Xl, Wb, Ws, Ch, Z : Variant;

implementation

{$R \*.dfm}

procedure TForm1.Report\_WordClick(Sender: TObject);

var I, Rcnt:integer;

begin

// Открываем базу данных и подсчитываем количество записей

ADODataSet1.Open;

Rcnt:=ADODataSet1.RecordCount;

// Запускаем MS Word

wd:= CreateOleObject('Word.Application');

// Отображаем на экране окно MS Word

wd.Visible :=True;

// Создаем новый документ

wd.Documents.Add;

Doc:= wd.Documents.Item(1);

// Добавляем новый абзац

Doc.Paragraphs.Add;

// Меняем его стиль

Doc.Paragraphs.Item(1).Style:='Заголовок 1';

// Создаем заголовок отчета

Rng := Doc.Range(0);

Rng.InsertBefore('Центр данных');

// Создаем заголовки колонок

Doc.Paragraphs.Add;

Rng.InsertAfter('Комп: Игрок: Дата: Продолжительность: Код услуги: Админ');

// Перемещаемся на первую запись набора данных

ADODataSet1.First;

for I := 1 to Rcnt do begin

// Добавляем новый абзац

Doc.Paragraphs.Add;

// Добавляем поля из текущей записи в новый абзац

Rng.InsertAfter(ADODataSet1.Fields[0].AsString+':'+

ADODataSet1.Fields[1].ASString+':'+

ADODataSet1.Fields[2].AsString+':'+

ADODataSet1.Fields[3].ASString+':'+

ADODataSet1.Fields[4].ASString+':'+

ADODataSet1.Fields[5].ASString);

ADODataSet1.Next;

end;

// Превращаем текст в таблицу

Rng:=Doc.Range(Doc.Paragraphs.Item(3).Range.Start, Doc.Paragraphs.Item(Rcnt+3).Range.End);

Tbl:=Rng.ConvertToTable(':',Rcnt,6);

// Изменяем размеры колонок таблицы

Tbl.Columns.Item(1).Width:=Tbl.Columns.Item(1).Width-30;

Tbl.Columns.Item(2).Width:=Tbl.Columns.Item(2).Width+20;

Tbl.Columns.Item(3).Width:=Tbl.Columns.Item(3).Width-3;

Tbl.Columns.Item(4).Width:=Tbl.Columns.Item(4).Width+20;

Tbl.Columns.Item(5).Width:=Tbl.Columns.Item(5).Width-5;

Tbl.Columns.Item(6).Width:=Tbl.Columns.Item(6).Width+5;

// Подавляем вывод диагностических сообщений

Wd.DisplayAlerts:=False;

// Сохраняем документ

Doc.SaveAs('D:\SOT\Custrep.doc');

// Закрываем Word и освобождаем ресурсы

//Wd.Quit;

Wd:=Unassigned;

end;

procedure TForm1.Report\_ExcelClick(Sender: TObject);

var I, Rcnt:integer;

begin

// Открываем базу данных и подсчитываем количество записей

ADODataSet1.Open;

Rcnt := ADODataSet1.RecordCount;

// Запускаем Microsoft Excel

Xl := CreateOleObject('Excel.Application');

// Отображаем окно Microsoft Excel

Xl.Visible := True;

// Создаем рабочую книгу

Xl.WorkBooks.Add;

Wb := XL.WorkBooks[1];

Ws := Wb.WorkSheets[1];

Ws.Name := 'Центр данных';

// Создаем заголовок отчета

Ws.Cells[1,1] := 'Центр данных';

Ws.Cells[1,1].Font.Bold := True;

Ws.Cells[1,1].Font.Size := 16;

Ws.Cells[2,1] := 'Комп';

Ws.Cells[2,2] := 'Игрок';

Ws.Cells[2,3] := 'Дата';

Ws.Cells[2,4] := 'Продолжительность';

Ws.Cells[2,5] := 'Код услуги';

Ws.Cells[2,6] := 'Админ';

for I:=1 to 6 do

Ws.Cells[2,i].Font.Bold := True;

// Перемещаемся на первую запись набора данных

ADODataSet1.First;

for I:=1 to Rcnt do

begin

// Добавляем значения полей текущей записи в новую строку

Ws.Cells[i+2,1] := ADODataSet1.Fields[0].AsString;

Ws.Cells[i+2,2] := ADODataSet1.Fields[1].AsString;

Ws.Cells[i+2,3] := ADODataSet1.Fields[2].AsString;

Ws.Cells[i+2,4] := ADODataSet1.Fields[3].AsString;

Ws.Cells[i+2,5] := ADODataSet1.Fields[4].AsString;

Ws.Cells[i+2,6] := ADODataSet1.Fields[5].AsString;

ADODataSet1.Next;

end;

// Изменяем ширину колонок

Xl.Columns[1].ColumnWidth:=Xl.Columns[1].ColumnWidth+5;

Xl.Columns[2].ColumnWidth:=Xl.Columns[2].ColumnWidth+5;

Xl.Columns[3].ColumnWidth:=Xl.Columns[3].ColumnWidth+5;

Xl.Columns[4].ColumnWidth:=Xl.Columns[4].ColumnWidth+5;

Xl.Columns[5].ColumnWidth:=Xl.Columns[5].ColumnWidth+5;

Xl.Columns[6].ColumnWidth:=Xl.Columns[6].ColumnWidth+5;

// Подавляем вывод диагностических сообщений

Xl.DisplayAlerts:=false;

// Сохраняем документ

Wb.SaveAs('D:\SOT\Custrer.xls');

// Закрываем Excel и освобождаем ресурсы

//Xl.Quit;

Xl:=Unassigned;

end;

procedure TForm1.CreatePivotTableClick(Sender: TObject);

var

WB,PC,PT:Variant;

const

// Константы Excel

xlExternal = $00000002;

xlCmdSql = $00000002;

xlColumnField = $00000002;

xlDataField = $00000004;

xlPageField = $00000003;

xlRowField = $00000001;

begin

// ADODataSet2.Open;

// Запускаем Microsoft Excel

Xl := CreateOleObject('Excel.Application');

// Делаем окно Excel видимым

Xl.Visible:=true;

// Создаем одну рабочую книгу с одним листом

Xl.WorkBooks.Add(1);

Wb := Xl.WorkBooks[1];

Ws := Wb.WorkSheets[1];

// Создаем кэш для хранения данных

PC := WB.PivotCaches.Add(xlExternal);

// Выбираем источник данных и текст запроса

PC.Connection := 'OLEDB;Provider=Microsoft.Jet.OLEDB.4.0;'+

'Data Source=D:\SOT\club.mdb';

PC.CommandType := xlCmdSql;

PC.CommandText := 'SELECT Игрок, Номер\_компа,'+' Админ, Код\_услуги, Цена FROM 01Cost\_more\_5';

// Создаем сводную таблицу

PC.CreatePivotTable(WB.Worksheets[1].Cells[1,1],'PivotTable1');

PT := WB.Worksheets[1].PivotTables('PivotTable1');

// Указываем расположение осей и суммируем данные

PT.PivotFields('Игрок').Orientation := xlRowField;

PT.PivotFields('Игрок').Position := 1;

PT.PivotFields('Номер\_компа').Orientation := xlRowField;

PT.PivotFields('Номер\_компа').Position := 2;

PT.PivotFields('Админ').Orientation := xlPageField;

PT.PivotFields('Код\_услуги').Orientation := xlColumnField;

PT.PivotFields('Цена').Orientation := xlDataField;

WB.Worksheets[1].Columns[2].ColumnWidth := 15;

WB.Worksheets[1].Columns[1].ColumnWidth := 20;

end;

procedure TForm1.RadioButton1Click(Sender: TObject);

var i:integer;

begin

ADOConnection1.GetTableNames(ComboBox1.Items);

for i:=Combobox1.Items.Count-1 downto 0 do

if Pos ('0', Combobox1.Items[i])>0 then Combobox1.Items.Delete(i) ;

for i:=Combobox1.Items.Count-1 downto 0 do

if Pos ('1', Combobox1.Items[i])>0 then Combobox1.Items.Delete(i) ;

for i:=Combobox1.Items.Count-1 downto 0 do

if Pos ('2', Combobox1.Items[i])>0 then Combobox1.Items.Delete(i) ;

end;

procedure TForm1.ComboBox1Click(Sender: TObject);

begin

if ADOTable1.Active then ADOTable1.Active:=False;

ADOTable1.TableName:=ComboBox1.Text;ADOTable1.Active:=True;

TabSheet1.Caption:= ComboBox1.Text;

end;

procedure TForm1.RadioButton2Click(Sender: TObject);

var i:integer;

begin

ADOConnection1.GetTableNames(ComboBox2.Items);

for i:=Combobox2.Items.Count-1 downto 0 do

if Pos ('Админ', Combobox2.Items[i])>0 then Combobox2.Items.Delete(i) ;

for i:=Combobox2.Items.Count-1 downto 0 do

if Pos ('Данные', Combobox2.Items[i])>0 then Combobox2.Items.Delete(i) ;

for i:=Combobox2.Items.Count-1 downto 0 do

if Pos ('Инфо игрока', Combobox2.Items[i])>0 then Combobox2.Items.Delete(i) ;

for i:=Combobox2.Items.Count-1 downto 0 do

if Pos ('Комп', Combobox2.Items[i])>0 then Combobox2.Items.Delete(i) ;

for i:=Combobox2.Items.Count-1 downto 0 do

if Pos ('Услуги', Combobox2.Items[i])>0 then Combobox2.Items.Delete(i) ;

for i:=Combobox2.Items.Count-1 downto 0 do

if Pos ('~', Combobox2.Items[i])>0 then Combobox2.Items.Delete(i) ;

end;

procedure TForm1.ComboBox2Click(Sender: TObject);

begin

if ADOTable1.Active then ADOTable1.Active:=False;

ADOTable1.TableName:=ComboBox2.Text;ADOTable1.Active:=True;

TabSheet1.Caption:= ComboBox2.Text;

end;

procedure TForm1.Button1Click(Sender: TObject);

begin

if ADOQuery2.Active then ADOQuery2.Active:=False;

ADOQuery2.SQL:=Select.Lines;ADOQuery2.Active:=True;

end;

procedure TForm1.Button2Click(Sender: TObject);

begin

if ADOTable1.Active then ADOTable1.Active:=False;

ADOTable1.TableName:=ComboBox1.Text;ADOTable1.Active:=True;

TabSheet1.Caption:= ComboBox1.Text;

end;

procedure TForm1.Button4Click(Sender: TObject);

{var

fam: string[30];

begin

fam:=InputBox('Выборка информации из БД',

'Укажите фамилию и щелкните на OK.', '');

if fam <> '' // пользователь ввел фамилию

then }

begin

with form1.ADOQuery2 do begin

Close; // закрыть файл-результат выполнения предыдущего запроса

SQL.Clear; // удалить текст предыдущего запроса

// записываем новый запрос в свойство SQL

SQL.Add('SELECT Админ, ФИО, Ставка');

SQL.Add('FROM Админ');

Open; // активизируем выполнение запроса

end;

{if ADOQuery2.RecordCount <> 0 then

DataSource1.DataSet:=ADOQuery2 // отобразить рез-т выполнения запроса

else begin

ShowMessage('В БД нет записей, удовлетворяющих критерию запроса.');

DataSource1.DataSet:=ADOTable1;

end; }

end;

procedure TForm1.Button5Click(Sender: TObject);

var I, Rcnt:integer;

begin

// Открываем базу данных и подсчитываем количество записей

ADODataSet3.Open;

Rcnt := ADODataSet3.RecordCount;

// Запускаем Microsoft Excel

Xl := CreateOleObject('Excel.Application');

// Отображаем окно Microsoft Excel

Xl.Visible := True;

// Создаем рабочую книгу

Xl.WorkBooks.Add;

Wb := XL.WorkBooks[1];

Ws := Wb.WorkSheets[1];

Ws.Name := 'Админ';

// Создаем заголовок отчета

Ws.Cells[1,1] := 'Админ';

Ws.Cells[1,1].Font.Bold := True;

Ws.Cells[1,1].Font.Size := 16;

Ws.Cells[2,1] := 'Админ';

Ws.Cells[2,2] := 'ФИО';

Ws.Cells[2,3] := 'Адрес';

Ws.Cells[2,4] := 'Ставка';

for I:=1 to 4 do

Ws.Cells[2,i].Font.Bold := True;

// Перемещаемся на первую запись набора данных

ADODataSet3.First;

for I:=1 to Rcnt do

begin

// Добавляем значения полей текущей записи в новую строку

Ws.Cells[i+2,1] := ADODataSet3.Fields[0].AsString;

Ws.Cells[i+2,2] := ADODataSet3.Fields[1].AsString;

Ws.Cells[i+2,3] := ADODataSet3.Fields[2].AsString;

Ws.Cells[i+2,4] := ADODataSet3.Fields[3].AsString;

ADODataSet3.Next;

end;

// Изменяем ширину колонок

Xl.Columns[1].ColumnWidth:=Xl.Columns[1].ColumnWidth+5;

Xl.Columns[2].ColumnWidth:=Xl.Columns[2].ColumnWidth+5;

Xl.Columns[3].ColumnWidth:=Xl.Columns[3].ColumnWidth+5;

Xl.Columns[4].ColumnWidth:=Xl.Columns[4].ColumnWidth+5;

// Подавляем вывод диагностических сообщений

Xl.DisplayAlerts:=false;

// Сохраняем документ

Wb.SaveAs('D:\SOT\Custrer2.xls');

// Закрываем Excel и освобождаем ресурсы

//Xl.Quit;

Xl:=Unassigned;

end;

procedure TForm1.Button6Click(Sender: TObject);

begin

with form1.ADOQuery2 do begin

//Close; // закрыть файл-результат выполнения предыдущего запроса

SQL.Clear; // удалить текст предыдущего запроса

// записываем новый запрос в свойство SQL

SQL.Add('INSERT INTO Админ (Админ, ФИО, Адрес, Ставка) VALUES ("Annie", "Бойко А.К.", "Засекречено:)", 1000);');

ExecSQL;

// активизируем выполнение запроса

end;

end;

end.