ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

АНГАРСКАЯ ГОСУДАРСТВЕННАЯ ТЕХНИЧЕСКАЯ АКАДАМИЯ

ФАКУЛЬТЕТ ТЕХНИЧЕСКОЙ КИБЕРНЕТИКИ

КАФЕДРА “ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ И КОМПЛЕКСЫ ”

**КУРСОВОЙ ПРОЕКТ**

по дисциплине «Проектирование клиент/серверных систем»

Тема:

**Информационная система «Data-центр»**

Выполнил: Зиннер Е.С.

Проверила: Засухина О. А.

Ангарск 2009 г.

**Содержание**

Введение

1. Техническое задание

1.1 Обоснование необходимости разработки приложения

1.1.1 Постановка задачи

1.1.1.1 Входные данные

1.1.1.2 Выходные данные

1.2 Научно исследовательские работы

1.2.1 Выбор СУБД

1.2.2 Выбор визуальной среды программирования

1.2.3 Механизм доступа данных к ADO

1.2.4 Выбор технических средств

2. Технический проект

2.1 Разработка структурированного приложения

2.2 Определение структуры входных и выходных данных

2.2.1 Структура таблиц

2.2.2 Схема данных

2.3 Определение формы представления входных и выходных данных

2.3.1 Формы

3. Рабочий проект

3.1 Текст программы

3.1.1 Сценарий в Microsoft SQL Server

3.1.2 Клиентское приложение

3.2 Руководство пользователя

Заключение

Литература

**Введение**

Целью курсовой работы является разработка информационной системы «Библиотека Дорам» в архитектуре клиент/сервер. БД «Библиотека Дорам» разрабатывается в серверной системе управления БД Microsoft SQL Server. Клиентское приложение разрабатывается в Code Gear RAD Studio механизм доступа ADO.

**Microsoft SQL Server** — система управления реляционными базами данных (СУБД), разработанная корпорацией Microsoft. Основной используемый язык запросов — Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями. Используется для от небольших и средних по размеру баз данных до крупных баз данных масштаба предприятия, конкурирует с другими СУБД в этом сегменте рынка.

Последняя версия SQL Server — SQL Server 2008 (кодовое название «Katmai»). Была выпущена 6 августа 2008 года.

SQL Server 2008 направлен на то, чтобы сделать управление данными самонастраивающимся, самоорганизующимся и самообслуживающимся механизмом — для реализации этих возможностей были созданы технологии SQL Server Always On. Это позволит уменьшить до нуля время нахождения сервера в нерабочем состоянии.

В SQL Server 2008 была добавлена поддержка структурированных и частичноструктурированных данных, включая цифровые форматы для изображений, звуков, видео и других типов мультимедиа. Поддержка мультимедиа форматов внутри СУБД позволила специализированным функциям взаимодействовать с этими типами данных.

Кроме этого, были включены специализированные форматы даты и времени и пространственный (англ. Spatial) тип для пространственно зависимых данных. Для неструктурированных данных были добавлены специализированные типы, например, тип File.

Для повышения эффективности администрирования в SQL Server были включены библиотеки Declarative Management Framework, позволяющие распределять полномочия для баз данных или отдельных таблиц. Были улучшены методы компрессии данных. SQL Server Katmai поддерживает набор библиотек ADO.NET Entity Framework и средства оповещения, репликации и определения данных.

**CodeGear RAD Studio 2007** - среда быстрой разработки приложений (RAD) для Microsoft Windows. CodeGear RAD Studio, ранее известный как проект Highlander, объединяет Delphi для Win32, C++Builder и новый Delphi.NET 2.0 в единую интегрированную среду. Это единственная интегрированная среда разработки (IDE), которая поддерживает быструю разработку как Windows, так и.NET-приложений для Microsoft Windows 2000, XP, и Vista. Подобная универсальность позволяет разработчикам строить Web, клиент/серверные и десктоп Windows-приложения для всех трех ОС и использовать такие приложения на любой из этих платформ.

CodeGear RAD Studio дает разработчикам гибкость в выборе операционной системы, которая наиболее полно удовлетворяет их требованиям при создании приложения для нескольких ОС Windows, включая программы для Windows Vista и Web-приложения.

Новые функции и возможности CodeGear RAD Studio 2007:

Расширенная поддержка Delphi для пользовательского интерфейса Microsoft Vista Aero, включая прикладные приложения, библиотеки визуальных компонентов (VCL) и поддержку VCL.NET для 3D эффектов и анимации пользовательского интерфейса, эффекта прозрачности, компонентов диалоговых окон для работы с файлами и задачами.

Поддержка языка Delphi для разработки в среде Microsoft.NET 2.0 (совместимо с.NET 3.0) и ASP.NET 2.0. ASP.NET представляет собой набор технологий в рамках.NET framework для создания Web-приложений и XML Web-сервисов.

Delphi для.NET имеет поддержку параметризованных типов, позволяющую разработчикам.NET применять Delphi для создания и использования классов, используя любой тип структуры данных в качестве параметров.

Blackfish SQL - сервер баз данных, полностью написанный на платформе.NET, который поддерживает SQL и обеспечивает легкое развертывание XCopy, поддержку транзакций, а также создание триггеров и хранимых процедур на языке.NET (Delphi.NET, C#, и Visual Basic.NET).

Enterprise Core Objects (ECO) IV - мощная среда для разработки на основе моделей (Model-Driven Development) для.NET 2.0. ECO позволяет разработчикам использовать визуальные диаграммы для построения объектов, а также связей и логики взаимодействия между ними, а затем автоматически создает и поддерживает инфраструктуру базы данных (с использованием Microsoft ADO.NET) и интерфейс взаимодействия данных, оставляя разработчику только заботу о бизнес-логике и внешнем виде приложения.

Обновленная архитектура доступа к базам данных dbExpress 4 с поддержкой ADO.NET 2.0. dbExpress 4 - это единое решение для доступа к базам данных для.NET и Windows с поддержкой ADO.NET, которое упрощает разработку и сопряжение десктоп и клиент/серверных приложений баз данных.

Как и платформа Windows, аппаратное обеспечение компьютеров и сеть постоянно эволюционируют, так и продукты CodeGear для Delphi и C++ также совершенствуются, чтобы дать разработчикам возможность извлечь из перемен пользу, создавая новые возможности для независимых поставщиков программного обеспечения (ISV) и создателей клиент/сервер приложений на основе баз данных.

# 1. Техническое задание

## 

## 1.1 Обоснование необходимости разработки приложения

СУБД используются практически во всех сферах человеческой деятельности;

Подготовка и издание данного приложения позволит отслеживать новинки, а так же узнавать ход перевода.

# 

# 1.1.1 Постановка задачи

Разработать БД «Библиотека Дорам» в серверной системе управления БД Microsoft SQL Server, клиентское приложение разрабатывается в Code Gear RAD Studio. БД должна хранить и предоставлять пользователю информацию о сериалах и переводчиках, подсчитывает количество записей в базе, возможность поиска определенной записи по режиссеру через фильтрацию, сортировку по выбранным параметрам (по стране или по году), вывод отчетов для последующей печати.

### 1.1.1.1 Входные данные

Технические характеристики оборудования

Цена оборудования

Клиенты

Время аренды

### 1.1.1.2 Выходные данные

Технические характеристики оборудования

Цена оборудования

Клиенты

Время аренды

## 1.2 Научно исследовательские работы

### 

### 1.2.1 Выбор СУБД

**Microsoft SQL Server** — система управления реляционными базами данных (СУБД), разработанная корпорацией Microsoft. Основной используемый язык запросов — Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями. Используется для от небольших и средних по размеру баз данных до крупных баз данных масштаба предприятия, конкурирует с другими СУБД в этом сегменте рынка.

Последняя версия SQL Server — SQL Server 2008 (кодовое название «Katmai»). Была выпущена 6 августа 2008 года.

SQL Server 2008 направлен на то, чтобы сделать управление данными самонастраивающимся, самоорганизующимся и самообслуживающимся механизмом — для реализации этих возможностей были созданы технологии SQL Server Always On. Это позволит уменьшить до нуля время нахождения сервера в нерабочем состоянии.

### 

### 1.2.2 Выбор визуальной среды программирования

**CodeGear RAD Studio 2007** - среда быстрой разработки приложений (RAD) для Microsoft Windows. CodeGear RAD Studio, ранее известный как проект Highlander, объединяет Delphi для Win32, C++Builder и новый Delphi.NET 2.0 в единую интегрированную среду. Это единственная интегрированная среда разработки (IDE), которая поддерживает быструю разработку как Windows, так и.NET-приложений для Microsoft Windows 2000, XP, и Vista. Подобная универсальность позволяет разработчикам строить Web, клиент/серверные и десктоп Windows-приложения для всех трех ОС и использовать такие приложения на любой из этих платформ. CodeGear RAD Studio 2007 открывает новые возможности для разработчиков, которые интересуются разработкой высокопроизводительных Windows-приложений с высокими требованиями к GUI, прикладных программ, работающих с базами данных, Web-приложений для бизнеса, а также стремятся извлечь максимальную пользу из усовершенствований аппаратного обеспечения и операционной системы Windows.

### 

### 1.2.3 Механизм доступа данных к ADO

ADO (от англ. ActiveX Data Objects — «объекты данных ActiveX») — интерфейс программирования приложений для доступа к данным, разработанный компанией Microsoft (MS Access, MS SQL Server) и основанный на технологии компонентов ActiveX. ADO позволяет представлять данные из разнообразных источников (реляционных баз данных, текстовых файлов и т. д.) в объектно-ориентированном виде. Компонент ADOQuery обеспечивает выполнение SQL запросов и получение требуемого набора данных. ADOConnection предназначен для хранения данных ADOTable, ADOQuery, ADOCommand. Основное свойство Connection string позволяет определить параметры Настроить вручную или при помощи редактора. ADOTable используется для доступа к хранилищам данных ADO и придоставляет информацию из них в табличном виде. Компонент связывается с данными при помощи Connection->ADOConnection, TableName-> Имя Таблицы, Active->True.

Объектная модель ADO состоит из следующих объектов высокого уровня и семейств объектов:

Connection (представляет подключение к удалённому источнику данных)

Recordset (представляет набор строк, полученный от источника данных)

Command (используется для выполнения команд и SQL-запросов с параметрами)

Record (может представлять одну запись объекта Recordset или же иерархическую структуру, состоящую из текстовых данных)

Stream (используется для чтения и записи потоковых данных, например, документов XML или двоичных объектов)

Errors (представляет ошибки)

Fields (представляет столбцы таблицы базы данных)

Parameters (представляет набор параметров SQL-инструкции)

Properties (представляет набор свойств объекта)

1.2.4. Выбор технических средств

Основанная база хранится на сервере, клиентские приложения хранятся на ПК в сети, доступ к базе осуществляется посредством локальной сети:

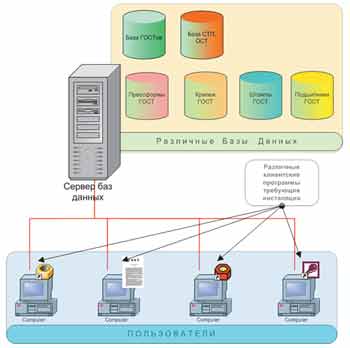


Рисунок 1.1 Архитектура локальной сети для клиент-серверных приложений

# 2. ТЕХНИЧЕСКИЙ ПРОЕКТ

## 

## 2.1 Разработка структурированного приложения

FRMMain

UnitMain

DM

UnitDM

FRMAdd

UnitAdd

FRMTable

UnitTable

FRMSrav

UnitSprav

Схема 2.1 Дерево модулей и форм.

## 

## 2.2 Определение структуры входных и выходных данных

### 

### 2.2.1 Структура таблиц

Таблица 2.1 TB\_DOGOVOR

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Пояснение | Тип данных | Ключ поля |
| ID\_DOGOVOR | Код договора | Int | Первичный |
| ID\_KLIENT | Код клиента | Int | Внешний |
| ID\_MASHIN | Код конфигурации | Int | Внешний |
| Date\_zakl | Дата заключения | DateTime |  |
| Date\_okonch | Дата окончания | DateTime |  |
| Stoimost | Общая стоимостьтоимость | DateTime |  |

Таблица 2.2 TB\_MASHIN

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Пояснение | Тип данных | Ключ поля |
| ID\_MASHIN | Код конфигурации | Int | Первичный |
| ID\_PROC | Код процессора | Int | Внешний |
| ID\_HDD | Код конфигурации | Int | Внешний |
| ID\_MEM | Код оперативной памяти | Int | Внешний |
| ID\_SPEED | Код скорости подключения | Int | Внешний |
| NAIM | Название конфигурации | nchar |  |
| CENA | Цена аренды в месяц | Real |  |

Таблица 2.3 TB\_HDD

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Пояснение | Тип данных | Ключ поля |
| ID\_HDD | Код жесткого диска | Int | Первичный |
| NAIM | Наименование | nchar |  |
| HARACT | Характеристики | nchar |  |

Таблица 2.4 TB\_KLIENTS

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Пояснение | Тип данных | Ключ поля |
| ID\_KLIENT | Код клиента | Int | Первичный |
| NAIM | ФИО | nchar |  |
| HARACT | Контактные данные | nchar |  |

Таблица 2.5 TB\_MEM

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Пояснение | Тип данных | Ключ поля |
| ID\_MEM | Код памяти | Int | Первичный |
| NAIM | Наименование | nchar |  |
| HARACT | Характеристики | nchar |  |

Таблица 2.6 TB\_PROC

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Пояснение | Тип данных | Ключ поля |
| ID\_PROC | Код процессора | Int | Первичный |
| NAIM | Наименование | nchar |  |
| HARACT | Характеристики | nchar |  |

Таблица 2.4 TB\_SPEED

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Пояснение | Тип данных | Ключ поля |
| ID\_SPEED | Код скорости | Int | Первичный |
| NAIM | Наименование | nchar |  |
| HARACT | Характеристики | nchar |  |

### 

### 2.2.2 Схема данных

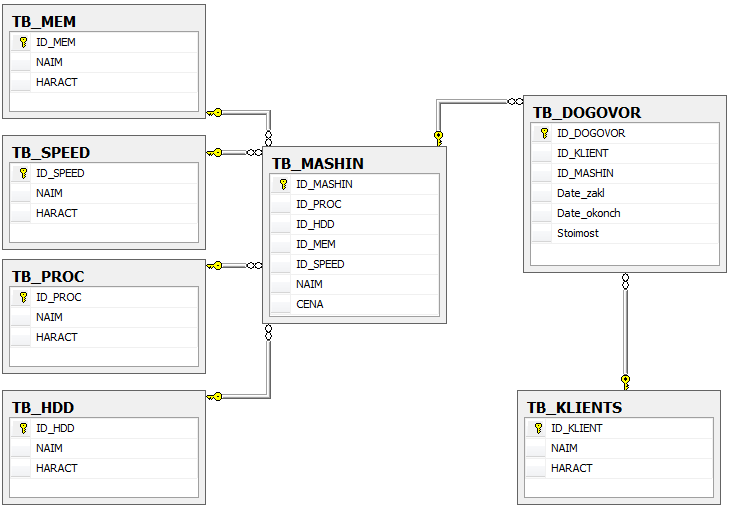


Схема 2.2 Структура таблиц

## 

## 2.3 Определение формы представления входных и выходных данных

### 

### 2.3.1 Формы

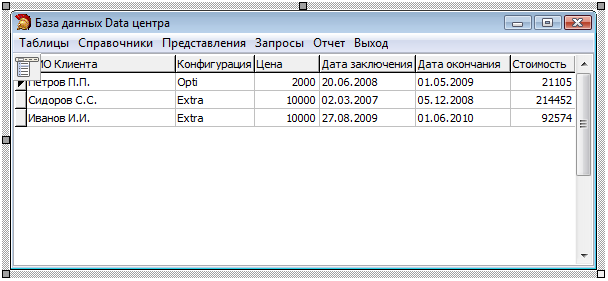


Рисунок 2.2 Главная форма

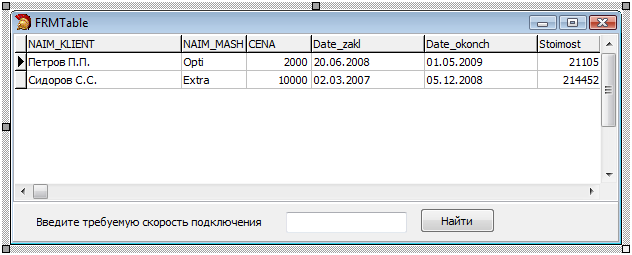


Рисунок 2.3 Форма запросов, таблиц и представлений.

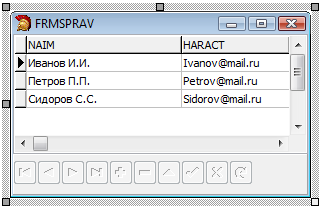


Рисунок 2.4 Форма справочников

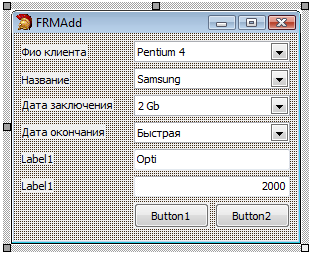


Рисунок 2.5 Форма добавления элементов в таблицы.

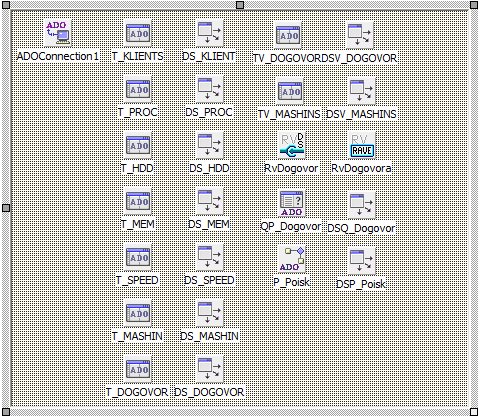


Рисунок 2.5 Data Module

# 3. РАБОЧИЙ ПРОЕКТ

## 

## 3.1 Текст программы

### 

### 3.1.1 Сценарий в Microsoft SQL Server

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

IF NOT EXISTS (SELECT \* FROM sys.objects WHERE object\_id = OBJECT\_ID(N'[dbo].[TB\_SPEED]') AND type in (N'U'))

BEGIN

CREATE TABLE [dbo].[TB\_SPEED](

[ID\_SPEED] [int] IDENTITY(1,1) NOT NULL,

[NAIM] [nchar](10) NULL,

[HARACT] [nchar](10) NULL,

CONSTRAINT [PK\_TB\_SPEED] PRIMARY KEY CLUSTERED

(

[ID\_SPEED] ASC

)WITH (IGNORE\_DUP\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

END

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

IF NOT EXISTS (SELECT \* FROM sys.objects WHERE object\_id = OBJECT\_ID(N'[dbo].[TB\_KLIENTS]') AND type in (N'U'))

BEGIN

CREATE TABLE [dbo].[TB\_KLIENTS](

[ID\_KLIENT] [int] IDENTITY(1,1) NOT NULL,

[NAIM] [nchar](25) NULL,

[HARACT] [nchar](50) NULL,

CONSTRAINT [PK\_TB\_KLIENTS] PRIMARY KEY CLUSTERED

(

[ID\_KLIENT] ASC

)WITH (IGNORE\_DUP\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

END

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

IF NOT EXISTS (SELECT \* FROM sys.objects WHERE object\_id = OBJECT\_ID(N'[dbo].[TB\_PROC]') AND type in (N'U'))

BEGIN

CREATE TABLE [dbo].[TB\_PROC](

[ID\_PROC] [int] IDENTITY(1,1) NOT NULL,

[NAIM] [nchar](10) NULL,

[HARACT] [nchar](10) NULL,

CONSTRAINT [PK\_TB\_PROC] PRIMARY KEY CLUSTERED

(

[ID\_PROC] ASC

)WITH (IGNORE\_DUP\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

END

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

IF NOT EXISTS (SELECT \* FROM sys.objects WHERE object\_id = OBJECT\_ID(N'[dbo].[TB\_MEM]') AND type in (N'U'))

BEGIN

CREATE TABLE [dbo].[TB\_MEM](

[ID\_MEM] [int] IDENTITY(1,1) NOT NULL,

[NAIM] [nchar](10) NULL,

[HARACT] [nchar](10) NULL,

CONSTRAINT [PK\_TB\_MEM] PRIMARY KEY CLUSTERED

(

[ID\_MEM] ASC

)WITH (IGNORE\_DUP\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

END

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

IF NOT EXISTS (SELECT \* FROM sys.objects WHERE object\_id = OBJECT\_ID(N'[dbo].[TB\_HDD]') AND type in (N'U'))

BEGIN

CREATE TABLE [dbo].[TB\_HDD](

[ID\_HDD] [int] IDENTITY(1,1) NOT NULL,

[NAIM] [nchar](10) NULL,

[HARACT] [nchar](10) NULL,

CONSTRAINT [PK\_TB\_HDD] PRIMARY KEY CLUSTERED

(

[ID\_HDD] ASC

)WITH (IGNORE\_DUP\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

END

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

IF NOT EXISTS (SELECT \* FROM sys.objects WHERE object\_id = OBJECT\_ID(N'[dbo].[TB\_MASHIN]') AND type in (N'U'))

BEGIN

CREATE TABLE [dbo].[TB\_MASHIN](

[ID\_MASHIN] [int] IDENTITY(1,1) NOT NULL,

[ID\_PROC] [int] NULL,

[ID\_HDD] [int] NULL,

[ID\_MEM] [int] NULL,

[ID\_SPEED] [int] NULL,

[NAIM] [nchar](10) NULL,

[CENA] [real] NULL,

CONSTRAINT [PK\_TB\_MASHIN] PRIMARY KEY CLUSTERED

(

[ID\_MASHIN] ASC

)WITH (IGNORE\_DUP\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

END

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

IF NOT EXISTS (SELECT \* FROM sys.objects WHERE object\_id = OBJECT\_ID(N'[dbo].[TB\_DOGOVOR]') AND type in (N'U'))

BEGIN

CREATE TABLE [dbo].[TB\_DOGOVOR](

[ID\_DOGOVOR] [int] IDENTITY(1,1) NOT NULL,

[ID\_KLIENT] [int] NULL,

[ID\_MASHIN] [int] NULL,

[Date\_zakl] [datetime] NULL,

[Date\_okonch] [datetime] NULL,

[Stoimost] [real] NULL,

CONSTRAINT [PK\_TB\_DOGOVOR] PRIMARY KEY CLUSTERED

(

[ID\_DOGOVOR] ASC

)WITH (IGNORE\_DUP\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

END

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

IF NOT EXISTS (SELECT \* FROM sys.views WHERE object\_id = OBJECT\_ID(N'[dbo].[PR\_MASHIN]'))

EXEC dbo.sp\_executesql @statement = N'CREATE VIEW [dbo].[PR\_MASHIN]

AS

SELECT dbo.TB\_MASHIN.NAIM AS NAIM\_MASH, dbo.TB\_MEM.NAIM AS NAIM\_MEM, dbo.TB\_SPEED.NAIM AS NAIM\_SPEED, dbo.TB\_PROC.NAIM AS NAIM\_PROC,

dbo.TB\_HDD.NAIM AS NAIM\_HDD, dbo.TB\_MASHIN.CENA

FROM dbo.TB\_MASHIN INNER JOIN

dbo.TB\_MEM ON dbo.TB\_MASHIN.ID\_MEM = dbo.TB\_MEM.ID\_MEM INNER JOIN

dbo.TB\_PROC ON dbo.TB\_MASHIN.ID\_PROC = dbo.TB\_PROC.ID\_PROC INNER JOIN

dbo.TB\_SPEED ON dbo.TB\_MASHIN.ID\_SPEED = dbo.TB\_SPEED.ID\_SPEED INNER JOIN

dbo.TB\_HDD ON dbo.TB\_MASHIN.ID\_HDD = dbo.TB\_HDD.ID\_HDD

'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_DiagramPane1', @value=N'[0E232FF0-B466-11cf-A24F-00AA00A3EFFF, 1.00]

Begin DesignProperties =

Begin PaneConfigurations =

Begin PaneConfiguration = 0

NumPanes = 4

Configuration = "(H (1[40] 4[20] 2[20] 3))"

End

Begin PaneConfiguration = 1

NumPanes = 3

Configuration = "(H (1 [50] 4 [25] 3))"

End

Begin PaneConfiguration = 2

NumPanes = 3

Configuration = "(H (1 [50] 2 [25] 3))"

End

Begin PaneConfiguration = 3

NumPanes = 3

Configuration = "(H (4 [30] 2 [40] 3))"

End

Begin PaneConfiguration = 4

NumPanes = 2

Configuration = "(H (1 [56] 3))"

End

Begin PaneConfiguration = 5

NumPanes = 2

Configuration = "(H (2 [66] 3))"

End

Begin PaneConfiguration = 6

NumPanes = 2

Configuration = "(H (4 [50] 3))"

End

Begin PaneConfiguration = 7

NumPanes = 1

Configuration = "(V (3))"

End

Begin PaneConfiguration = 8

NumPanes = 3

Configuration = "(H (1[56] 4[18] 2))"

End

Begin PaneConfiguration = 9

NumPanes = 2

Configuration = "(H (1 [75] 4))"

End

Begin PaneConfiguration = 10

NumPanes = 2

Configuration = "(H (1[66] 2))"

End

Begin PaneConfiguration = 11

NumPanes = 2

Configuration = "(H (4 [60] 2))"

End

Begin PaneConfiguration = 12

NumPanes = 1

Configuration = "(H (1))"

End

Begin PaneConfiguration = 13

NumPanes = 1

Configuration = "(V (4))"

End

Begin PaneConfiguration = 14

NumPanes = 1

Configuration = "(V (2))"

End

ActivePaneConfig = 0

End

Begin DiagramPane =

Begin Origin =

Top = 0

Left = 0

End

Begin Tables =

Begin Table = "TB\_MEM"

Begin Extent =

Top = 6

Left = 245

Bottom = 108

Right = 414

End

DisplayFlags = 280

TopColumn = 0

End

Begin Table = "TB\_PROC"

Begin Extent =

Top = 6

Left = 452

Bottom = 108

Right = 621

End

DisplayFlags = 280

TopColumn = 0

End

Begin Table = "TB\_SPEED"

Begin Extent =

Top = 108

Left = 245

Bottom = 210

Right = 414

End

DisplayFlags = 280

TopColumn = 0

End

Begin Table = "TB\_HDD"

Begin Extent =

Top = 108

Left = 452

Bottom = 210

Right = 621

End

DisplayFlags = 280

TopColumn = 0

End

Begin Table = "TB\_MASHIN"

Begin Extent =

Top = 6

Left = 38

Bottom = 123

Right = 207

End

DisplayFlags = 280

TopColumn = 3

End

End

End

Begin SQLPane =

End

Begin DataPane =

Begin ParameterDefaults = ""

End

End

Begin CriteriaPane =

Begin ColumnWidths = 11

Column = 1440

Alias = 900

Table = 1170

Output = 720

Append = 1400

NewValue = 1170

SortType = 1350

SortOrder = 1410

GroupBy = 1350

Filter = 1350

Or = 1350

Or = 1350

Or = 135',@level0type=N'SCHEMA', @level0name=N'dbo', @level1type=N'VIEW', @level1name=N'PR\_MASHIN'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_DiagramPane2', @value=N'0

End

End

End

',@level0type=N'SCHEMA', @level0name=N'dbo', @level1type=N'VIEW', @level1name=N'PR\_MASHIN'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_DiagramPaneCount', @value=2,@level0type=N'SCHEMA', @level0name=N'dbo', @level1type=N'VIEW', @level1name=N'PR\_MASHIN'

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

IF NOT EXISTS (SELECT \* FROM sys.views WHERE object\_id = OBJECT\_ID(N'[dbo].[PR\_DOGOVOR]'))

EXEC dbo.sp\_executesql @statement = N'CREATE VIEW [dbo].[PR\_DOGOVOR]

AS

SELECT dbo.TB\_KLIENTS.NAIM AS NAIM\_KLIENT, dbo.TB\_MASHIN.NAIM AS NAIM\_MASH, dbo.TB\_MASHIN.CENA, dbo.TB\_DOGOVOR.Date\_zakl,

dbo.TB\_DOGOVOR.Date\_okonch, dbo.TB\_DOGOVOR.Stoimost

FROM dbo.TB\_DOGOVOR INNER JOIN

dbo.TB\_KLIENTS ON dbo.TB\_DOGOVOR.ID\_KLIENT = dbo.TB\_KLIENTS.ID\_KLIENT INNER JOIN

dbo.TB\_MASHIN ON dbo.TB\_DOGOVOR.ID\_MASHIN = dbo.TB\_MASHIN.ID\_MASHIN

'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_DiagramPane1', @value=N'[0E232FF0-B466-11cf-A24F-00AA00A3EFFF, 1.00]

Begin DesignProperties =

Begin PaneConfigurations =

Begin PaneConfiguration = 0

NumPanes = 4

Configuration = "(H (1[40] 4[20] 2[20] 3))"

End

Begin PaneConfiguration = 1

NumPanes = 3

Configuration = "(H (1 [50] 4 [25] 3))"

End

Begin PaneConfiguration = 2

NumPanes = 3

Configuration = "(H (1 [50] 2 [25] 3))"

End

Begin PaneConfiguration = 3

NumPanes = 3

Configuration = "(H (4 [30] 2 [40] 3))"

End

Begin PaneConfiguration = 4

NumPanes = 2

Configuration = "(H (1 [56] 3))"

End

Begin PaneConfiguration = 5

NumPanes = 2

Configuration = "(H (2 [66] 3))"

End

Begin PaneConfiguration = 6

NumPanes = 2

Configuration = "(H (4 [50] 3))"

End

Begin PaneConfiguration = 7

NumPanes = 1

Configuration = "(V (3))"

End

Begin PaneConfiguration = 8

NumPanes = 3

Configuration = "(H (1[56] 4[18] 2))"

End

Begin PaneConfiguration = 9

NumPanes = 2

Configuration = "(H (1 [75] 4))"

End

Begin PaneConfiguration = 10

NumPanes = 2

Configuration = "(H (1[66] 2))"

End

Begin PaneConfiguration = 11

NumPanes = 2

Configuration = "(H (4 [60] 2))"

End

Begin PaneConfiguration = 12

NumPanes = 1

Configuration = "(H (1))"

End

Begin PaneConfiguration = 13

NumPanes = 1

Configuration = "(V (4))"

End

Begin PaneConfiguration = 14

NumPanes = 1

Configuration = "(V (2))"

End

ActivePaneConfig = 0

End

Begin DiagramPane =

Begin Origin =

Top = 0

Left = 0

End

Begin Tables =

Begin Table = "TB\_MASHIN"

Begin Extent =

Top = 6

Left = 38

Bottom = 123

Right = 207

End

DisplayFlags = 280

TopColumn = 3

End

Begin Table = "TB\_KLIENTS"

Begin Extent =

Top = 6

Left = 245

Bottom = 108

Right = 414

End

DisplayFlags = 280

TopColumn = 0

End

Begin Table = "TB\_DOGOVOR"

Begin Extent =

Top = 6

Left = 452

Bottom = 123

Right = 621

End

DisplayFlags = 280

TopColumn = 0

End

End

End

Begin SQLPane =

End

Begin DataPane =

Begin ParameterDefaults = ""

End

End

Begin CriteriaPane =

Begin ColumnWidths = 11

Column = 1440

Alias = 900

Table = 1170

Output = 720

Append = 1400

NewValue = 1170

SortType = 1350

SortOrder = 1410

GroupBy = 1350

Filter = 1350

Or = 1350

Or = 1350

Or = 1350

End

End

End

',@level0type=N'SCHEMA', @level0name=N'dbo', @level1type=N'VIEW', @level1name=N'PR\_DOGOVOR'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_DiagramPaneCount', @value=1,@level0type=N'SCHEMA', @level0name=N'dbo', @level1type=N'VIEW', @level1name=N'PR\_DOGOVOR'

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

IF NOT EXISTS (SELECT \* FROM sys.objects WHERE object\_id = OBJECT\_ID(N'[dbo].[PPoisk]') AND type in (N'P', N'PC'))

BEGIN

EXEC dbo.sp\_executesql @statement = N'-- -- Author:<Author,,Name>

-- Create date: <Create Date,,>

-- Description:<Description,,>

CREATE PROCEDURE [dbo].[PPoisk]

@p CHAR(10)

AS

BEGIN

SELECT NAIM\_MASH, NAIM\_MEM, NAIM\_SPEED, NAIM\_PROC, NAIM\_HDD, CENA

FROM PR\_MASHIN

WHERE (NAIM\_SPEED = @p)

END

'

END

GO

IF NOT EXISTS (SELECT \* FROM sys.foreign\_keys WHERE object\_id = OBJECT\_ID(N'[dbo].[FK\_TB\_MASHIN\_TB\_HDD]') AND parent\_object\_id = OBJECT\_ID(N'[dbo].[TB\_MASHIN]'))

ALTER TABLE [dbo].[TB\_MASHIN] WITH CHECK ADD CONSTRAINT [FK\_TB\_MASHIN\_TB\_HDD] FOREIGN KEY([ID\_HDD])

REFERENCES [dbo].[TB\_HDD] ([ID\_HDD])

GO

IF NOT EXISTS (SELECT \* FROM sys.foreign\_keys WHERE object\_id = OBJECT\_ID(N'[dbo].[FK\_TB\_MASHIN\_TB\_MEM]') AND parent\_object\_id = OBJECT\_ID(N'[dbo].[TB\_MASHIN]'))

ALTER TABLE [dbo].[TB\_MASHIN] WITH CHECK ADD CONSTRAINT [FK\_TB\_MASHIN\_TB\_MEM] FOREIGN KEY([ID\_MEM])

REFERENCES [dbo].[TB\_MEM] ([ID\_MEM])

GO

IF NOT EXISTS (SELECT \* FROM sys.foreign\_keys WHERE object\_id = OBJECT\_ID(N'[dbo].[FK\_TB\_MASHIN\_TB\_PROC]') AND parent\_object\_id = OBJECT\_ID(N'[dbo].[TB\_MASHIN]'))

ALTER TABLE [dbo].[TB\_MASHIN] WITH CHECK ADD CONSTRAINT [FK\_TB\_MASHIN\_TB\_PROC] FOREIGN KEY([ID\_PROC])

REFERENCES [dbo].[TB\_PROC] ([ID\_PROC])

GO

IF NOT EXISTS (SELECT \* FROM sys.foreign\_keys WHERE object\_id = OBJECT\_ID(N'[dbo].[FK\_TB\_MASHIN\_TB\_SPEED]') AND parent\_object\_id = OBJECT\_ID(N'[dbo].[TB\_MASHIN]'))

ALTER TABLE [dbo].[TB\_MASHIN] WITH CHECK ADD CONSTRAINT [FK\_TB\_MASHIN\_TB\_SPEED] FOREIGN KEY([ID\_SPEED])

REFERENCES [dbo].[TB\_SPEED] ([ID\_SPEED])

GO

IF NOT EXISTS (SELECT \* FROM sys.foreign\_keys WHERE object\_id = OBJECT\_ID(N'[dbo].[FK\_TB\_DOGOVOR\_TB\_KLIENTS]') AND parent\_object\_id = OBJECT\_ID(N'[dbo].[TB\_DOGOVOR]'))

ALTER TABLE [dbo].[TB\_DOGOVOR] WITH CHECK ADD CONSTRAINT [FK\_TB\_DOGOVOR\_TB\_KLIENTS] FOREIGN KEY([ID\_KLIENT])

REFERENCES [dbo].[TB\_KLIENTS] ([ID\_KLIENT])

GO

IF NOT EXISTS (SELECT \* FROM sys.foreign\_keys WHERE object\_id = OBJECT\_ID(N'[dbo].[FK\_TB\_DOGOVOR\_TB\_MASHIN]') AND parent\_object\_id = OBJECT\_ID(N'[dbo].[TB\_DOGOVOR]'))

ALTER TABLE [dbo].[TB\_DOGOVOR] WITH CHECK ADD CONSTRAINT [FK\_TB\_DOGOVOR\_TB\_MASHIN] FOREIGN KEY([ID\_MASHIN])

REFERENCES [dbo].[TB\_MASHIN] ([ID\_MASHIN])

### 

### 3.1.2 Клиентское приложение

unit UnitMain;

interface

uses

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

Dialogs, Grids, DBGrids, Menus, ExtCtrls, DBCtrls, StdCtrls;

type

TFRMMain = class(TForm)

MainMenu1: TMainMenu;

N1: TMenuItem;

N2: TMenuItem;

N3: TMenuItem;

N4: TMenuItem;

N11: TMenuItem;

N21: TMenuItem;

N5: TMenuItem;

N6: TMenuItem;

N7: TMenuItem;

N8: TMenuItem;

N9: TMenuItem;

N10: TMenuItem;

N12: TMenuItem;

N13: TMenuItem;

N14: TMenuItem;

N15: TMenuItem;

N16: TMenuItem;

DBGrid1: TDBGrid;

procedure N10Click(Sender: TObject);

procedure N9Click(Sender: TObject);

procedure N7Click(Sender: TObject);

procedure N8Click(Sender: TObject);

procedure N12Click(Sender: TObject);

procedure N6Click(Sender: TObject);

procedure N15Click(Sender: TObject);

procedure N16Click(Sender: TObject);

procedure N5Click(Sender: TObject);

procedure N13Click(Sender: TObject);

procedure N14Click(Sender: TObject);

procedure N11Click(Sender: TObject);

procedure N21Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

FRMMain: TFRMMain;

implementation

uses UnitDM, UnitSprav, UnitTable;

{$R \*.dfm}

procedure TFRMMain.N7Click(Sender: TObject);

begin

FRMSPRAV.DBGrid1.DataSource:=DM.DS\_KLIENT;

FRMSPRAV.DBNavigator1.DataSource:=DM.DS\_KLIENT;

FRMSPRAV.Caption:='Клиенты';

FRMSPRAV.Show;

end;

procedure TFRMMain.N8Click(Sender: TObject);

begin

FRMSPRAV.DBGrid1.DataSource:=DM.DS\_PROC;

FRMSPRAV.DBNavigator1.DataSource:=DM.DS\_PROC;

FRMSPRAV.Caption:='Процессоры';

FRMSPRAV.Show;

end;

procedure TFRMMain.N9Click(Sender: TObject);

begin

FRMSPRAV.DBGrid1.DataSource:=DM.DS\_HDD;

FRMSPRAV.DBNavigator1.DataSource:=DM.DS\_HDD;

FRMSPRAV.Caption:='Жесткие диски';

FRMSPRAV.Show;

end;

procedure TFRMMain.N10Click(Sender: TObject);

begin

FRMSPRAV.DBGrid1.DataSource:=DM.DS\_MEM;

FRMSPRAV.DBNavigator1.DataSource:=DM.DS\_MEM;

FRMSPRAV.Show;

FRMSPRAV.Caption:='Память';

end;

procedure TFRMMain.N11Click(Sender: TObject);

begin

FRMTable.Label1.Caption:=('Введите название машины');

FRMTable.DBGrid1.DataSource:=DM.DSQ\_Dogovor;

FRMTable.ClientHeight:=211;

FRMTable.Show;

FRMTable.Panel1.Visible:=false;

FRMTable.Panel2.Visible:=true;

FRMTable.Caption:='Поиск арендаторов';

end;

procedure TFRMMain.N12Click(Sender: TObject);

begin

FRMSPRAV.DBGrid1.DataSource:=DM.DS\_SPEED;

FRMSPRAV.DBNavigator1.DataSource:=DM.DS\_SPEED;

FRMSPRAV.Show;

FRMSPRAV.Caption:='Скорость';

end;

procedure TFRMMain.N13Click(Sender: TObject);

begin

FRMTable.DBGrid1.DataSource:=DM.DSv\_DOGOVOR;

FRMTable.Show;

FRMTable.ClientHeight:=169;

FRMTable.Caption:='Договора';

end;

procedure TFRMMain.N14Click(Sender: TObject);

begin

FRMTable.DBGrid1.DataSource:=DM.DSV\_MASHINS;

FRMTable.Show;

FRMTable.ClientHeight:=169;

FRMTable.Caption:='Прайс-лист';

end;

procedure TFRMMain.N15Click(Sender: TObject);

begin

FRMTable.DBGrid1.DataSource:=DM.DS\_DOGOVOR;

FRMTable.ClientHeight:=211;

FRMTable.Show;

FRMTable.Panel1.Visible:=true;

FRMTable.Panel2.Visible:=false;

FRMTable.Caption:='Договора';

end;

procedure TFRMMain.N16Click(Sender: TObject);

begin

FRMTable.DBGrid1.DataSource:=DM.DS\_MASHIN;

FRMTable.ClientHeight:=211;

FRMTable.Show;

FRMTable.Panel1.Visible:=true;

FRMTable.Panel2.Visible:=false;

FRMTable.Caption:='Прайс-лист';

end;

procedure TFRMMain.N21Click(Sender: TObject);

begin

FRMTable.Label1.Caption:=('Введите требуемый объем памяти');

FRMTable.DBGrid1.DataSource:=DM.DSP\_Poisk;

FRMTable.ClientHeight:=211;

FRMTable.Show;

FRMTable.Panel1.Visible:=false;

FRMTable.Panel2.Visible:=true;

FRMTable.Caption:='Поиск машины';

end;

procedure TFRMMain.N5Click(Sender: TObject);

begin

DM.RvDogovora.Execute;

end;

procedure TFRMMain.N6Click(Sender: TObject);

begin

close;

end;

end.

unit UnitDM;

interface

uses

SysUtils, Classes, DB, ADODB, RpCon, RpConDS, RpDefine, RpRave, RpRender,

RpRenderCanvas, RpRenderPreview, DBClient;

type

TDM = class(TDataModule)

T\_KLIENTS: TADOTable;

T\_DOGOVOR: TADOTable;

T\_MASHIN: TADOTable;

T\_SPEED: TADOTable;

T\_MEM: TADOTable;

T\_HDD: TADOTable;

T\_PROC: TADOTable;

DS\_DOGOVOR: TDataSource;

DS\_MASHIN: TDataSource;

DS\_SPEED: TDataSource;

DS\_MEM: TDataSource;

DS\_HDD: TDataSource;

DS\_PROC: TDataSource;

DS\_KLIENT: TDataSource;

ADOConnection1: TADOConnection;

T\_HDDID\_HDD: TAutoIncField;

T\_HDDNAIM: TWideStringField;

T\_HDDHARACT: TWideStringField;

T\_PROCID\_PROC: TAutoIncField;

T\_PROCNAIM: TWideStringField;

T\_PROCHARACT: TWideStringField;

T\_MEMID\_MEM: TAutoIncField;

T\_MEMNAIM: TWideStringField;

T\_MEMHARACT: TWideStringField;

T\_KLIENTSID\_KLIENT: TAutoIncField;

T\_KLIENTSNAIM: TWideStringField;

T\_SPEEDID\_SPEED: TAutoIncField;

T\_SPEEDNAIM: TWideStringField;

T\_SPEEDHARACT: TWideStringField;

RvDogovora: TRvProject;

RvDogovor: TRvDataSetConnection;

TV\_DOGOVOR: TADOTable;

DSV\_DOGOVOR: TDataSource;

TV\_MASHINS: TADOTable;

DSV\_MASHINS: TDataSource;

TV\_MASHINSNAIM\_MASH: TWideStringField;

TV\_MASHINSNAIM\_MEM: TWideStringField;

TV\_MASHINSNAIM\_SPEED: TWideStringField;

TV\_MASHINSNAIM\_PROC: TWideStringField;

TV\_MASHINSNAIM\_HDD: TWideStringField;

TV\_MASHINSCENA: TFloatField;

TV\_DOGOVORNAIM\_MASH: TWideStringField;

TV\_DOGOVORCENA: TFloatField;

TV\_DOGOVORDate\_zakl: TDateTimeField;

TV\_DOGOVORDate\_okonch: TDateTimeField;

TV\_DOGOVORStoimost: TFloatField;

TV\_DOGOVORNAIM\_KLIENT: TWideStringField;

QP\_Dogovor: TADOQuery;

DSQ\_Dogovor: TDataSource;

P\_Poisk: TADOStoredProc;

DSP\_Poisk: TDataSource;

T\_KLIENTSHARACT: TWideStringField;

T\_DOGOVORID\_DOGOVOR: TAutoIncField;

T\_DOGOVORID\_KLIENT: TIntegerField;

T\_DOGOVORID\_MASHIN: TIntegerField;

T\_DOGOVORDate\_zakl: TDateTimeField;

T\_DOGOVORDate\_okonch: TDateTimeField;

T\_DOGOVORStoimost: TFloatField;

T\_MASHINID\_MASHIN: TAutoIncField;

T\_MASHINID\_PROC: TIntegerField;

T\_MASHINID\_HDD: TIntegerField;

T\_MASHINID\_MEM: TIntegerField;

T\_MASHINID\_SPEED: TIntegerField;

T\_MASHINNAIM: TWideStringField;

T\_MASHINCENA: TFloatField;

QP\_DogovorNAIM\_KLIENT: TWideStringField;

QP\_DogovorNAIM\_MASH: TWideStringField;

QP\_DogovorCENA: TFloatField;

QP\_DogovorDate\_zakl: TDateTimeField;

QP\_DogovorDate\_okonch: TDateTimeField;

QP\_DogovorStoimost: TFloatField;

P\_PoiskNAIM\_MASH: TWideStringField;

P\_PoiskNAIM\_MEM: TWideStringField;

P\_PoiskNAIM\_SPEED: TWideStringField;

P\_PoiskNAIM\_PROC: TWideStringField;

P\_PoiskNAIM\_HDD: TWideStringField;

P\_PoiskCENA: TFloatField;

procedure T\_DOGOVORCalcFields(DataSet: TDataSet);

private

{ Private declarations }

public

{ Public declarations }

end;

var

DM: TDM;

implementation

uses UnitMain;

{$R \*.dfm}

procedure TDM.T\_DOGOVORCalcFields(DataSet: TDataSet);

begin

T\_Dogovor.Fields[5].AsFloat:=round(T\_Dogovor.Fields[4].AsDateTime-T\_Dogovor.Fields[3].AsDateTime)\*(T\_Mashin.Fields[6].AsInteger/30);

end;

end.

unit UnitTable;

interface

uses

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

Dialogs, Grids, DBGrids, StdCtrls, ExtCtrls;

type

TFRMTable = class(TForm)

DBGrid1: TDBGrid;

Panel1: TPanel;

Button3: TButton;

Button2: TButton;

Button1: TButton;

Panel2: TPanel;

Edit1: TEdit;

Button4: TButton;

Label1: TLabel;

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Button3Click(Sender: TObject);

procedure Button4Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

FRMTable: TFRMTable;

implementation

uses UnitDM, UnitAdd;

{$R \*.dfm}

procedure set\_add\_form\_dogovor;

begin

FRMAdd.DBLookupComboBoxD1.Visible:=true;

FRMAdd.DBLookupComboBoxD2.Visible:=true;

FRMAdd.DBEditD1.Visible:=true;

FRMAdd.DBEditD2.Visible:=true;

FRMAdd.DBLookupComboBoxM1.Visible:=false;

FRMAdd.DBLookupComboBoxM2.Visible:=false;

FRMAdd.DBLookupComboBoxM3.Visible:=false;

FRMAdd.DBLookupComboBoxM4.Visible:=false;

FRMAdd.DBEditm1.Visible:=false;

FRMAdd.DBEditm2.Visible:=false;

FRMAdd.Label1.Caption:=('ФИО Клиента');

FRMAdd.Label2.Caption:=('Название машины');

FRMAdd.Label3.Caption:=('Дата заключения');

FRMAdd.Label4.Caption:=('Дата окончания');

FRMAdd.Label5.Caption:=('');

FRMAdd.Label6.Caption:=('');

end;

procedure set\_add\_form\_mashin;

begin

FRMAdd.DBLookupComboBoxD1.Visible:=false;

FRMAdd.DBLookupComboBoxD2.Visible:=false;

FRMAdd.DBEditD1.Visible:=false;

FRMAdd.DBEditD2.Visible:=false;

FRMAdd.DBLookupComboBoxM1.Visible:=true;

FRMAdd.DBLookupComboBoxM2.Visible:=true;

FRMAdd.DBLookupComboBoxM3.Visible:=true;

FRMAdd.DBLookupComboBoxM4.Visible:=true;

FRMAdd.DBEditm1.Visible:=true;

FRMAdd.DBEditm2.Visible:=true;

FRMAdd.Label1.Caption:=('Процессор');

FRMAdd.Label2.Caption:=('Жесткий диск');

FRMAdd.Label3.Caption:=('Память');

FRMAdd.Label4.Caption:=('Скорость подключения');

FRMAdd.Label5.Caption:=('Название');

FRMAdd.Label6.Caption:=('Цена');

end;

procedure TFRMTable.Button1Click(Sender: TObject);

begin

if DBGrid1.DataSource=DM.DS\_DOGOVOR then begin

DM.T\_DOGOVOR.Append;

set\_add\_form\_dogovor;

FRMAdd.Caption:='Добавить договор';

end else

if DBGrid1.DataSource=DM.DS\_MASHIN then begin

DM.T\_MASHIN.Append;

set\_add\_form\_mashin;

FRMAdd.Caption:='Добавить конфигурацию';

end;

FRMAdd.show;

end;

procedure TFRMTable.Button2Click(Sender: TObject);

begin

if DBGrid1.DataSource=DM.DS\_DOGOVOR then begin

DM.T\_DOGOVOR.Edit;

set\_add\_form\_dogovor;

FRMAdd.Caption:='Редактировать договор';

end else

if DBGrid1.DataSource=DM.DS\_MASHIN then begin

DM.T\_MASHIN.Edit;

set\_add\_form\_mashin;

FRMAdd.Caption:='Редактировать конфигурацию';

end;

FRMAdd.show;

end;

procedure TFRMTable.Button3Click(Sender: TObject);

begin

if DBGrid1.DataSource=DM.DS\_DOGOVOR then begin

DM.T\_DOGOVOR.Delete;

end else

if DBGrid1.DataSource=DM.DS\_MASHIN then begin

DM.T\_MASHIN.Delete;

end;

end;

procedure TFRMTable.Button4Click(Sender: TObject);

var s:string;

begin

if DBGrid1.DataSource=DM.DSQ\_DOGOVOR then begin

DM.QP\_Dogovor.Close;

s:='SELECT NAIM\_KLIENT, NAIM\_MASH, CENA, Date\_zakl, Date\_okonch, ';

s:=s+' Stoimost FROM PR\_DOGOVOR ';

s:=s+'WHERE NAIM\_MASH =('''+edit1.Text+''')';

DM.QP\_Dogovor.SQL.Clear;

DM.QP\_Dogovor.SQL.Add(s);

try

dm.QP\_Dogovor.Open;

except

showmessage('ошибка');

end;

end else

if DBGrid1.DataSource=DM.DSP\_Poisk then begin

dm.P\_Poisk.close;

dm.P\_Poisk.Parameters.ParamByName('@p').Value:=Edit1.Text;

dm.P\_Poisk.Open;

end;

end;

end.

unit UnitAdd;

interface

uses

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

Dialogs, StdCtrls, DBCtrls, Mask;

type

TFRMAdd = class(TForm)

Button1: TButton;

DBEditD1: TDBEdit;

DBEditD2: TDBEdit;

DBLookupComboBoxD1: TDBLookupComboBox;

DBLookupComboBoxD2: TDBLookupComboBox;

Button2: TButton;

DBLookupComboBoxM1: TDBLookupComboBox;

DBLookupComboBoxM2: TDBLookupComboBox;

DBLookupComboBoxM3: TDBLookupComboBox;

DBLookupComboBoxM4: TDBLookupComboBox;

DBEditM1: TDBEdit;

DBEditM2: TDBEdit;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Label5: TLabel;

Label6: TLabel;

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

FRMAdd: TFRMAdd;

implementation

uses UnitDM, UnitTable;

{$R \*.dfm}

procedure TFRMAdd.Button1Click(Sender: TObject);

begin

if FRMTable.DBGrid1.DataSource=DM.DS\_DOGOVOR then begin

DM.T\_Dogovor.Fields[5].AsFloat:=round(DM.T\_Dogovor.Fields[4].AsDateTime-DM.T\_Dogovor.Fields[3].AsDateTime)\*(round(DM.T\_Mashin.Fields[6].AsInteger/30));

DM.T\_DOGOVOR.Post;

DM.T\_DOGOVOR.Active:=false;

DM.T\_DOGOVOR.Active:=true;

DM.T\_DOGOVOR.Append;

end

else if FRMTable.DBGrid1.DataSource=DM.DS\_MASHIN then begin

DM.T\_MASHIN.Post;

DM.T\_MASHIN.Active:=false;

DM.T\_MASHIN.Active:=true;

DM.T\_MASHIN.Append;

end;

end;

procedure TFRMAdd.Button2Click(Sender: TObject);

begin

close;

end;

end.

## 3.2 Руководство пользователя

Главная форма (рис. 3.1) позволяет получить доступ ко всем функциям программы, путем использования главного меню. Так же на ней отображен весь список договоров.

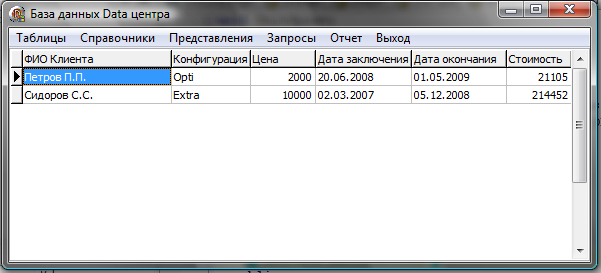


Рисунок 3.1 Главная форма

С помощью главного меню открывается редактор таблиц, путем использования кнопок «Добавить», «Редактировать», «Удалить» можно внести изменения в состав таблиц (рис. 3.2, 3.3). При нажатии кнопки «Ок» на форме добавления/редактирования записи автоматически посчитается сумма договора на основании длительности договора и стоимости аренды данной конфигурации.

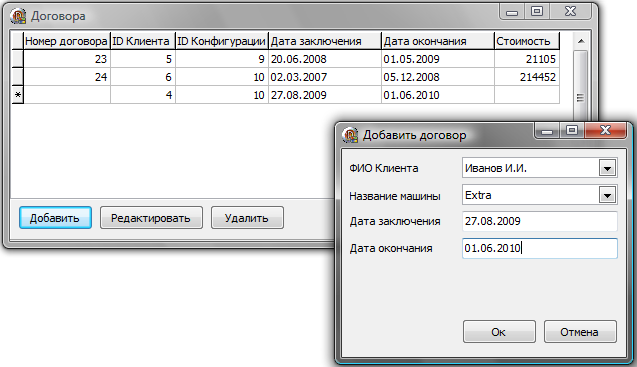


Рисунок 3.2 Таблица «Договора» и форма для внесения изменений.

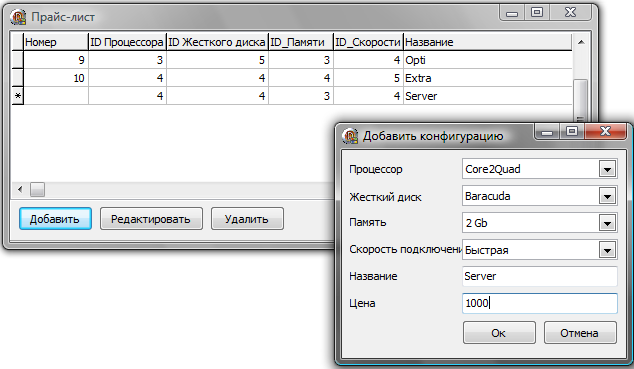


Рисунок 3.3 Таблица «Прайс-лист» и форма для внесения изменений.

При использовании второго пункта меню «Справочники» Можно внести изменения в имеющиеся в базе данных справочники (рис. 3.4 – 3.8)

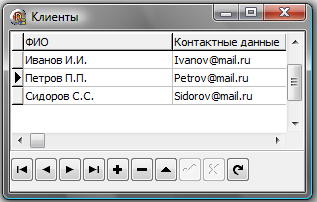


Рисунок 3.4 Справочник «Клиенты».

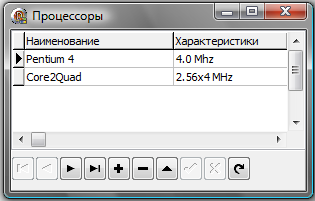


Рисунок 3.5 Справочник «Процессоры».

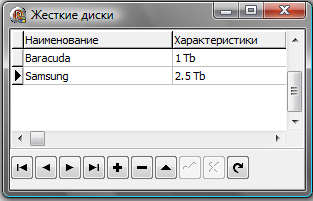


Рисунок 3.6 Справочник «Жесткие диски».

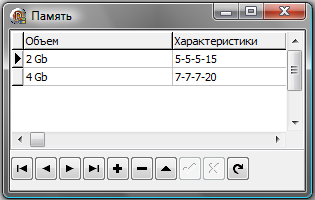


Рисунок 3.7 Справочник «Оперативная память».

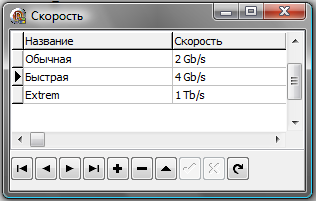


Рисунок 3.8 Справочник «Скорость».

Для удобства просмотра хранящейся в базе данных информации сделаны представления, которые можно увидеть с помощью пункта меню «Представления» (рис. 3.9, 3.10)

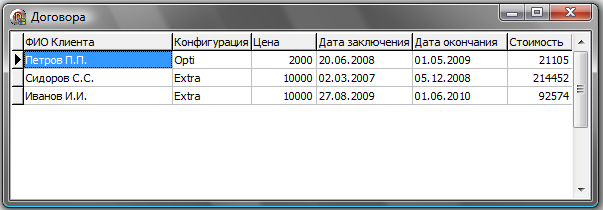


Рисунок 3.9 Представление Договоров.

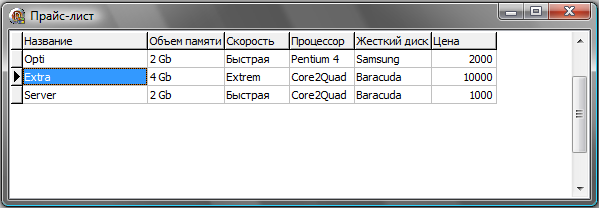


Рисунок 3.10 Представление Прайс-листа.

Для получения каких-либо сведений из базы данных используются запросы. На рис 3.11 показан результат запроса на выборку всех арендаторов заданной конфигурации.

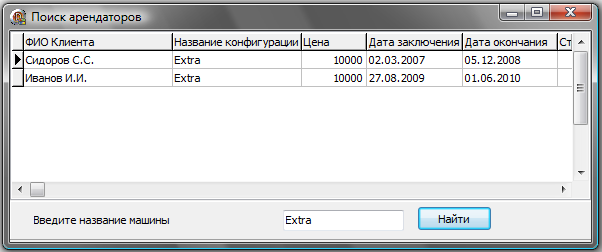


Рисунок 3.11 Запрос.

Для удобства просмотра и возможности вывода информации на печать была использована программа RaveReports с помощью которой можно создавать отчеты. На рис. 3.12 показан внешний вид получаемого отчета.

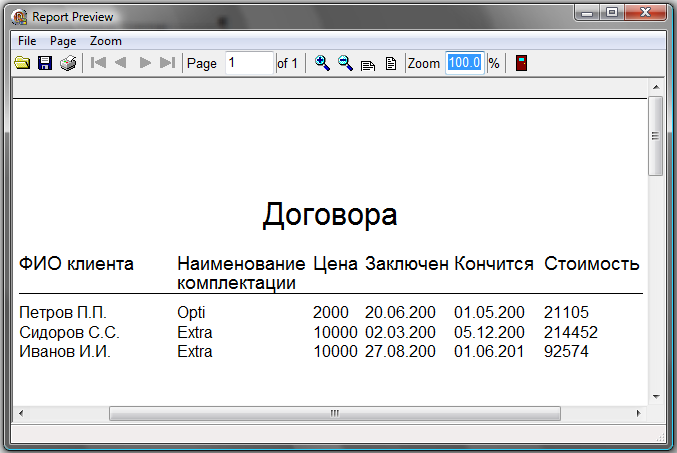


Рисунок 3.12 Вид отчета.

# Заключение

В результате работы была создана реляционная база данных в Microsoft SQL Server, содержащая 7 таблиц, 5 из которых справочники. Клиентское приложение написанное с помощью CodeGear Delphi позволяющее просматривать, изменять и искать нужную информацию. Были рассмотрены различные способы внесения и получения данных из базы данных. Построен отчет для вывода на печать с помощью программы RaveReports. Изучены свойства и методы компонентов CodeGear Delphi для работы с базами данных.