**Часть 1 - Необходимые настройки до начала проектирования.**

ПС: На компьютере установлена виртуальная машина WMware 7 с ОС WINDOWS Server 2008 Datacenter R2 и запущенным сервером БД MS SQL Server 2008 R2.

**1. Настройка сети в WMWARE 7.**

**1.1. Настройки Virtual Network Editor VMnet8 в программе WMWARE.**

Главное меню: EDIT ==> Virtual Network Editor ==> WMnet8;

Почему WMnet8? Потому что виртуальная сеть WMnet8 настроена в режиме NAT(преобразование сетевых адресов).

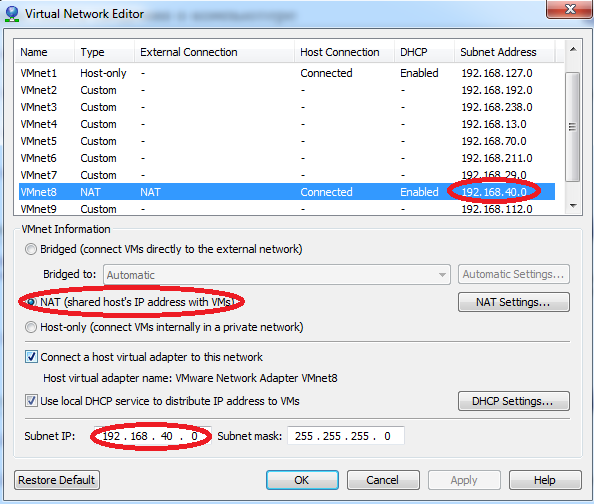


Рисунок 6 - Настройки WMnet8.

Выбираем режим «преобразование сетевых адресов» сети NAT:

* IP-подсети: 192.168.40.0;
* макска подсети: 255.255.255.0;

**1.2. Настройки NAT для виртуальной сети VMnet8.**

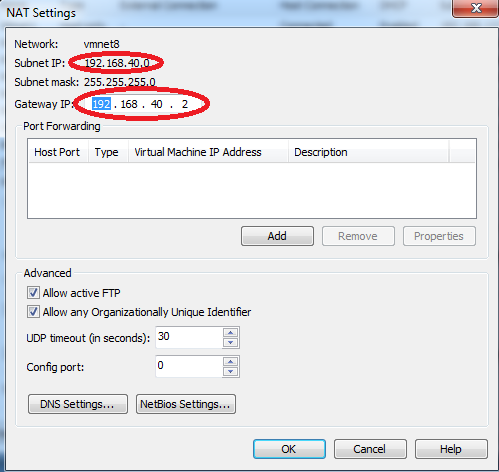


Рисунок 7 - Настройки NAT для виртуальной сети WMnet8.

Нажимаем кнопку NAT Settings... и при необходимости можно отредактировать значение Getway IP.

**1.3. Настройка системы автоматического распространения IP-адресов для VMnet8.**

Нажимаем кнопку DHCP Settings... и при необходимости можно отредактировать значения:

"Start IP Address" и "End IP Address"

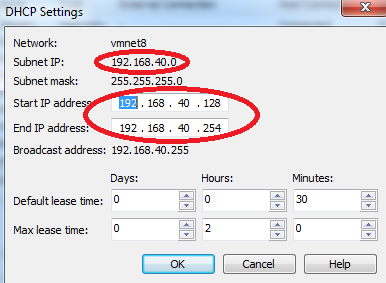


Рисунок 8 - Редактирование значений Start IP address и End IP address.

**2. Настройка сетевых адаптеров на физической машине.**

После установки WMWare во вкладке Сетевые подключения, наблюдаем появления 2 сетевых виртуальных адаптера: VWware Network Adapter VMnet1 и VWware Network Adapter VMnet8. Нас интересует сеть VMnet8 т.к. она настроена в режиме NAT (преобразование сетевых адресов).

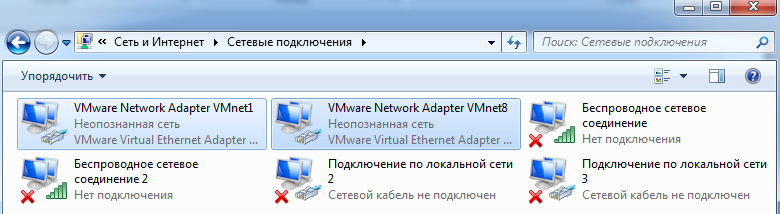


Рисунок 1 - Появление 2 новых виртуальных сетевых адаптера.

**2.1. Настройка виртуального сетевого адаптера VMnet8.**

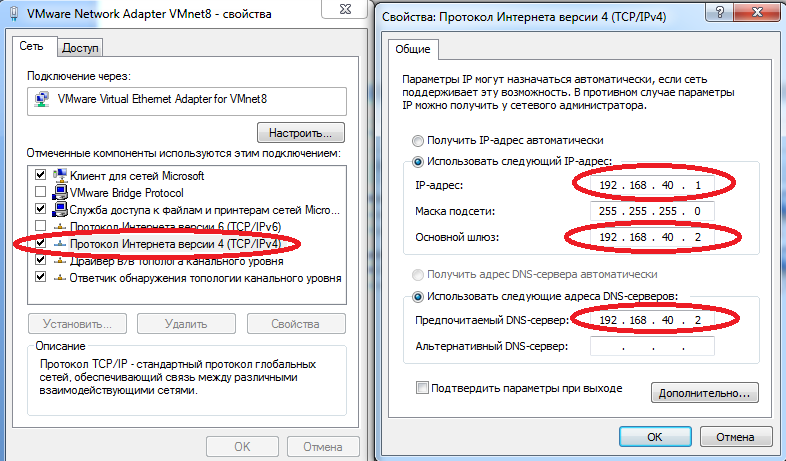


Рисунок 3 - Настройка протокола интернета TCP/IPv4 для виртуального сетевого адаптера VMnet8.

Настроим IP адресов для виртульного сетевого адаптера VMnet8: VWware Network Adapter VMnet1 ==> свойства ==> протокол интернета версии 4 (TCP/IPv4):

IP: 192.168.40.1

Маска подсети: 255.255.255.0

Основной шлюз: 192.168.40.2

Предпочитаемый DNS-сервер: 192.168.40.2

**3. Настройка сетевых адаптеров на виртуальной машине.**

Во вкладке Сетевые подключения настроим сетевые адреса сетевого адаптера. Настроим IP адресов для виртуальной машине Сетевые подключения ==> свойства ==> протокол интернета версии 4 (TCP/IPv4):

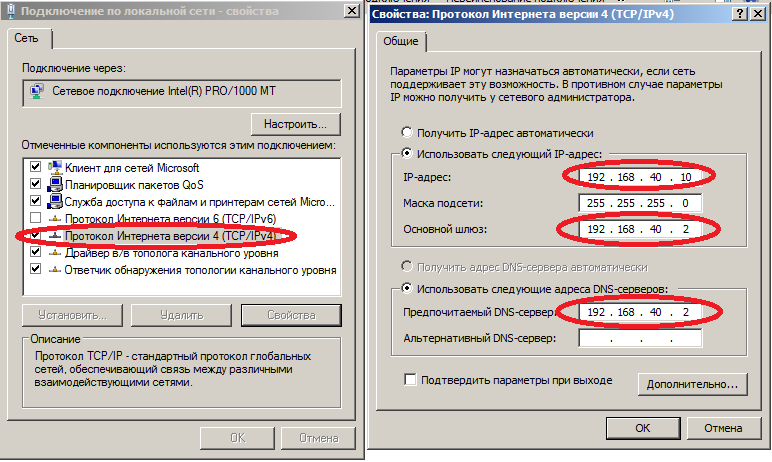


Рисунок 9 - Настройка протокола интернета TCP/IPv4 для виртуальной машине.

IP: 192.168.40.10

Маска подсети: 255.255.255.0

Основной шлюз: 192.168.40.2

Предпочитаемый DNS-сервер: 192.168.40.2

Настройка сети выполнена. Таким образом привязали виртуальную машину в локальную сеть.

IP основной машины: 192.168.40.1

IP виртуальной машины: 192.168.40.19

Getway IP основной шлюз: 192.168.40.2

**4. Настройки брандмауэра и разрешения доступа.**

**4.1 Настройки разрешения доступа**

Настройки защиты могут серьёзно ограничить доступ в локальной сети. Для их отключения заходим: Панель управления ==> Сеть и Интернет ==> Центр управления сетями и общим доступом ==> Дополнительные параметры общего доступа

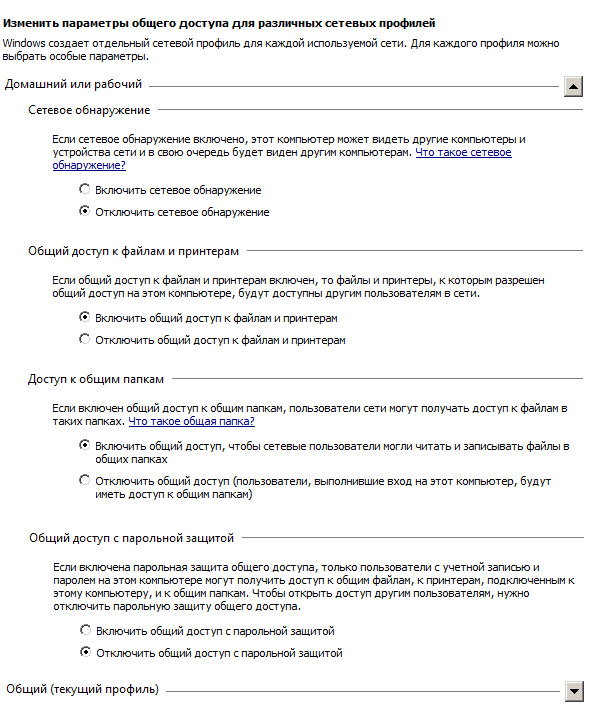


Рисунок - Дополнительные настройки общего доступа для домашней или рабочей группы на физической и виртуальной машинах.

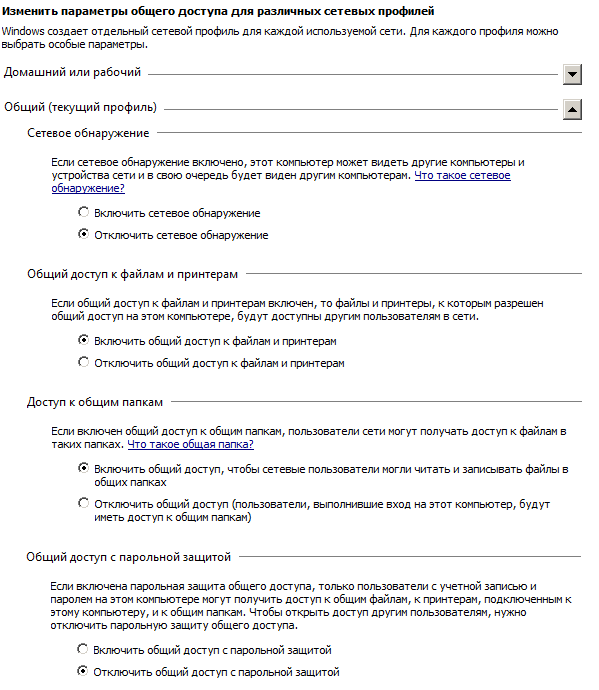


Рисунок - Дополнительные настройки общего доступа для домашней или общей группы на физической и виртуальной машинах.

**4.2. Настройки брандмауэра**

Настройки защиты брандмауэра могут ограничить доступ в локальной сети. Для их отключения заходим: Панель управления ==> Система и безопасность ==> Брандмауэр Windows ==> Настроить параметры. Если у антивируса своя личная система брандмауэр (например norton internet security), её нужно будет тоже отключить.

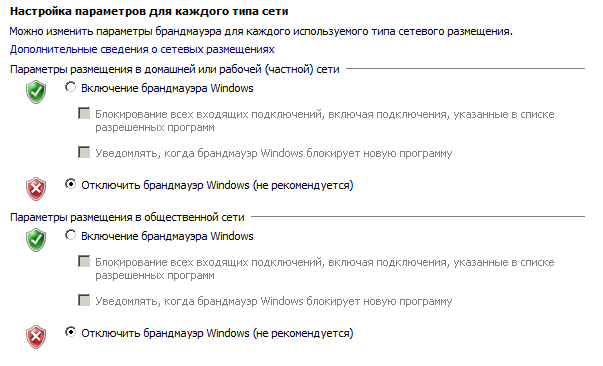


Рисунок - Отключение брандмауэра на физической и виртуальной машинах.

В пунктах 1 по 4 указаны все проблемы доступа, которые могут возникать в локальной сети (это может быть реальная сеть или виртуальная). Если сервер БД MS SQL SERVER установлен на той же машине, где установлен VISUAL STUDIO 2010, тогда переходим к пункту 5.

**5. Настройка сервера БД MS SQL Server 2008 R2.**

**5.1. Проверка сервисов сервера БД.**

Пуск ==> Все программы ==> Microsoft SQL Sever 2008 R2 ==> Configuration tools ==> SQL Server configuration manager ==> SQL server services.

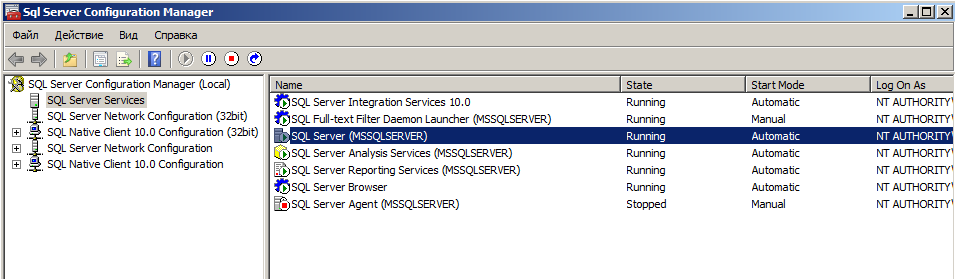


Рисунок - Службы MS SQL Server 2008 R2.

Как видно служба SQL Server должна быть запущена (state = Running) если служба остановлена, её необходимо запустить (правая кнопка мыши ==> start).

**5.2. Проверка протоколов сети сервера БД.**

Пуск ==> Все программы ==> Microsoft SQL Sever 2008 R2 ==> Configuration tools ==> SQL Server configuration manager ==> SQL Server Network Configuration ==> Protocols for MSSQLSERVER.

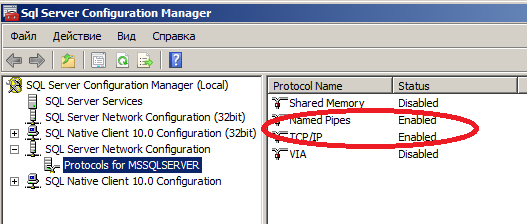


Рисунок - Протоколы сети: Protocols for MSSQLSERVER

Активными протоколами являются: Named Pipes и TCP/IP.

**Часть 2 - Пример выполнения ТР по дисциплине БД.**

Задание: Структуризация текстового содержания HTML страниц.

Не секрет, что в WWW огромное количество информационных сайтов (например: news.yandex.ru; auto.mail.ru; babyblog.ru и другие). У каждого сайта огромное количество страниц (URL внутри сайта), каждая страница содержит специальные элементы (тэги), которые могут содержать текстовую информацию. Вся текстовая информация записывается в БД.

**1. Нормализация БД.**

**1-ая нормальная форма:**

Из определения 1-ой нормальной формы:

Переменная отношения находится в первой нормальной форме (1НФ) [тогда и только тогда](http://ru.wikipedia.org/wiki/%D0%A2%D0%BE%D0%B3%D0%B4%D0%B0_%D0%B8_%D1%82%D0%BE%D0%BB%D1%8C%D0%BA%D0%BE_%D1%82%D0%BE%D0%B3%D0%B4%D0%B0), когда в любом допустимом значении отношения каждый его [кортеж](http://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D1%80%D1%82%D0%B5%D0%B6) содержит только одно значение для каждого из атрибутов (атомарность значений).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Сотрудник** | **отдел** | **должность** | **tel/fax** | **наличие PC** |
| Антон | Программирования | Программист | 345-5487/345-5488 | Да |
| Иван | Секретариат | Бухгалтер | 345-5488/345-5489 | Да |
| Григорий | Охрана | Охранник | 345-6767 | Нет |
| Елена | Секретариат | Секретарь | 345-5488/345-5489 | Да |
| Анна | Снабжения | Менеджер | 345-5490 | Нет |
| Александр | Выдачи | Менеджер | 345-0000/345-0001 | Нет |
| Василий | Выдачи | Менеджер | 345-0000/345-0001 | Нет |
| Сергей | Программирования | Админ БД | 345-5487/345-5488 | Да |

Для получения 1-ой нормальной формы необходимо разбить столбец " **tel/fax** " на 2 отдельных столбца:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Сотрудник** | **отдел** | **должность** | **tel** | **fax** | **наличие PC** |
| Антон | Программирования | Программист | 345-5487 | 345-5488 | Да |
| Иван | Секретариат | Бухгалтер | 345-5488 | 345-5489 | Да |
| Григорий | Охрана | Охранник | 345-6767 |  | Нет |
| Елена | Секретариат | Секретарь | 345-5488 | 345-5489 | Да |
| Анна | Снабжения | Менеджер | 345-5490 |  | Нет |
| Александр | Выдачи | Менеджер | 345-0000 | 345-0001 | Нет |
| Василий | Выдачи | Менеджер | 345-0000 | 345-0001 | Нет |
| Сергей | Программирования | Админ БД | 345-5487 | 345-5488 | Да |

Все столбцы содержат сейчас только атомарные значения. 1НФ достигнута.

**2-ая нормальная форма:**

Из определения 2-ой нормальной формы:

Переменная отношения находится во второй нормальной форме тогда и только тогда, когда она находится в [первой нормальной форме](http://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%B2%D0%B0%D1%8F_%D0%BD%D0%BE%D1%80%D0%BC%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%84%D0%BE%D1%80%D0%BC%D0%B0) и каждый не ключевой атрибут неприводимо зависит от ее потенциального ключа.

Как видно наличие компьютера зависить только от должности (охраннику PC не нужен; программисту просто необходим). Для получения 2-ой нормальной формы необходимо разбить таблицу (отношение) на 3 отдельных отношений.

|  |  |  |
| --- | --- | --- |
| **Сотрудник** | **отдел** | **должность** |
| Антон | Программирования | Программист |
| Иван | Секретариат | Бухгалтер |
| Григорий | Охрана | Охранник |
| Елена | Секретариат | Секретарь |
| Анна | Снабжения | Менеджер |
| Александр | Выдачи | Менеджер |
| Василий | Выдачи | Менеджер |
| Сергей | Программирования | Админ БД |

и

|  |  |
| --- | --- |
| **должность** | **наличие PC** |
| Программист | Да |
| Бухгалтер | Да |
| Охранник | Нет |
| Секретарь | Да |
| Менеджер | Нет |
| Админ БД | Да |

и

|  |  |  |
| --- | --- | --- |
| **отдел** | **tel** | **fax** |
| Программирования | 345-5487 | 345-5488 |
| Секретариат | 345-5488 | 345-5489 |
| Охрана | 345-6767 |  |
| Снабжения | 345-5490 |  |
| Выдачи | 345-0000 |  |

Отношение находится во 2-ой НФ.

**3-ая нормальная форма:**

Из определения 3-ой нормальной формы: отношение находится в третьей нормальной форме если оно уже 2НФ и любой её не ключевой атрибут функционально зависит только от первичного ключа.

отношение оказалось в 3НФ. Нормализация выполнена.

**2. Наименование таблиц, типы и размерность данных.**

**Таблица 1 на псевдокоде:**

Create table worker

(

worker\_name varchar(100) not null PRIMARY KEY

,section varchar(100) not null FOREIGN KEY REFERENCES section\_table

,duty varchar(100) not null FOREIGN KEY REFERENCES duty\_table

)

**Описание столбцов:**

worker\_name - полное имя сотрудника. тип строка (до 100 символов), первичный ключ.

section - отдел. тип строка (до 100 символов), внешний ключ .

duty - должность. тип строка (до 100 символов), внешний ключ.

**Таблица 2 на псевдокоде:**

Create table duty\_table

(

duty varchar(100) not null PRIMARY KEY

,PC tinyint not null DEFAULT 0

)

**Описание столбцов:**

duty - Должность. тип строка (до 100 символов), первичный ключ.

PC - Признак наличия PC, тип число (tinyint), по умалчанию 0;

**Таблица 3 на псевдокоде:**

Create table section\_table

(

section varchar(100) not null PRIMARY KEY

,tel varchar(15) NOT NULL

,fax varchar(15) NULL

)

**Описание столбцов:**

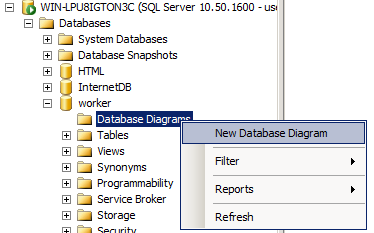
section - отдел. тип строка (до 100 символов), первичный ключ.

tel - Номер телефона в формате (495)-123-45-67.

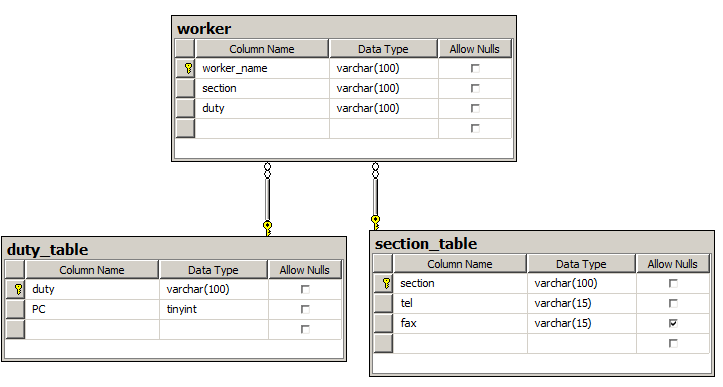
fax - Номер факса в формате (495)-123-45-67.

**3. Физическая Модель: Первичные ключи, родительские и дочерние таблицы.**

Создана БД "worker" и нужные таблицы. Переходим к диаграммы базы данных "worker":



Добавляем (или создаём) нужные нам таблицы. Получаем схему (диаграмму) БД "worker".



В физической модели БД можно разглядеть все ключи и отношения между таблицами.

**4. Шаблоны процедур, функций или триггеров.**

**Процедура:**

CREATE procedure proc\_workers\_in\_section @section varchar(100) as

SELECT S.section, count(\*) as cnt

into #t -- записали результат во временную таблицу

FROM worker.dbo.section\_table S with (nolock) -- Грязное чтение

inner join

worker.dbo.worker W with (nolock) -- Грязное чтение

ON S.section = W.section

WHERE S.section = @section

GROUP BY S.section

-- Выводим результат из временной таблицы.

SELECT \* from #t;

drop table #t;

**Функция типа "table-valued":**

CREATE FUNCTION tfunc\_workers\_in\_section (@section varchar(100))

RETURNS @t1 table

(

section varchar(100)

,cnt int

)

as

BEGIN

INSERT @t1 (section, cnt)

SELECT S.section, count(\*) as cnt

FROM worker.dbo.section\_table S with (nolock) -- Грязное чтение

inner join

worker.dbo.worker W with (nolock) -- Грязное чтение

ON S.section = W.section

WHERE S.section = @section

GROUP BY S.section

RETURN

END

Функция типа "scalar-valued":

CREATE FUNCTION sfunc\_workers\_in\_section (@section varchar(100))

RETURNS int

AS

BEGIN

declare @cnt int

SELECT @cnt = count(\*)

FROM worker.dbo.section\_table S with (nolock) -- Грязное чтение

inner join

worker.dbo.worker W with (nolock) -- Грязное чтение

ON S.section = W.section

WHERE S.section = @section

GROUP BY S.section

RETURN @cnt

END

триггер:

CREATE trigger t\_worker on worker.dbo.worker

after /\*BEFORE\*/ update

as

declare @old\_value varchar(50), @host\_name varchar(50);

declare @new\_value varchar(50);

select @old\_value=deleted.worker\_name, @host\_name = HOST\_NAME() from deleted;

select @new\_value=inserted.worker\_name from inserted;

-- Заполняем уже созданную таблицу

INSERT WORKER\_history (worker\_name, HOST\_NAME)

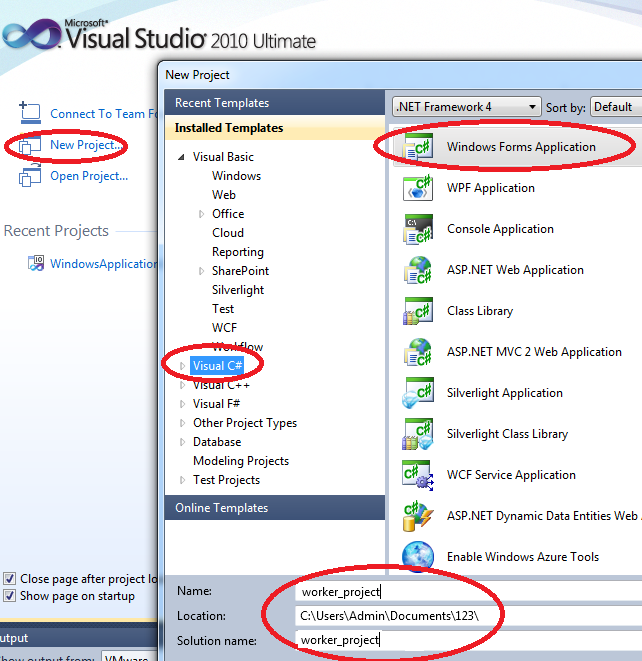
values (@old\_value, @host\_name), (@new\_value, @host\_name)

Проектирование БД завершено. БД создана, таблицы созданы, отношение между таблицами установлено.

**5. Разработка клиентского модуля с помощью VISUAL STUDIO 2010 для работы с БД "worker".**

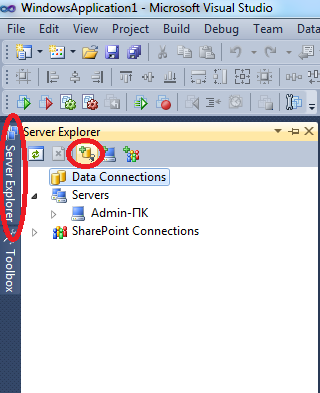
5.1. Новый проект в среде VISUAL STUDIO 2010.

В среде Visual Studio 2010 создаём новый проект "Windows Forms Application", указав при этом папку и название проекта.

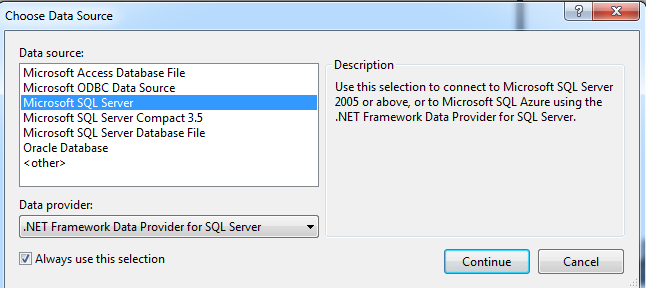


5.2. Подключение сервера БД.

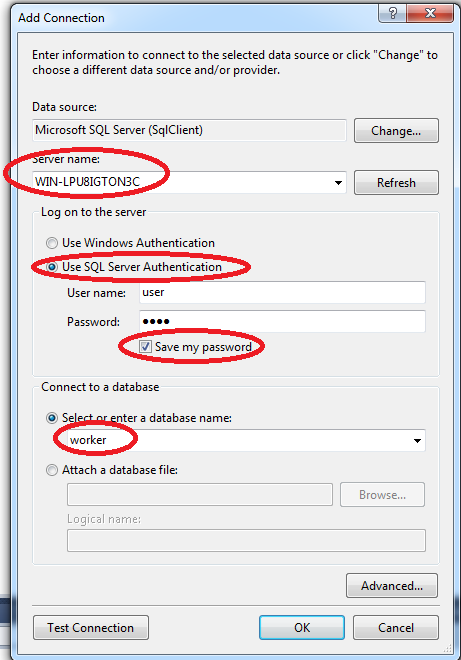
Во вкладке Server Explorer выбираем "Connect to Database"



Указываем тип сервера БД: Microsoft SQL Server и драйвер (Data provider): .NET Framework Data Provider for SQL Server. Нажимаем кнопку "Continue"



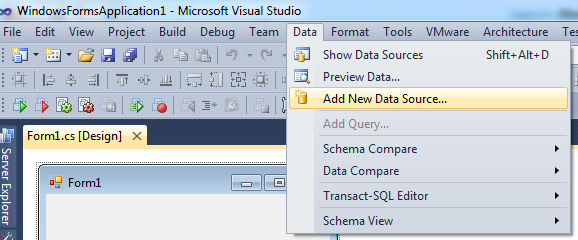
Выбераем имя сервера на котором установлен сервер БД server name : <имя сервера>; выбираем SQL Server Authentication и вводим логин и пароль; Выбираем базу данных <наименование БД>;



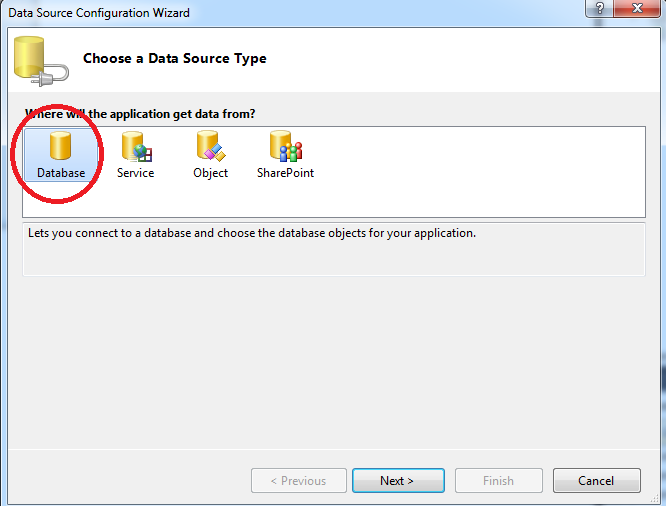
Нажимаем кнопку "ОК" и подключение состоялось.

**5.3. Подключение источника данных.**

Подключаем источник данных: главное меню ==> Data ==> Add New Data Source...

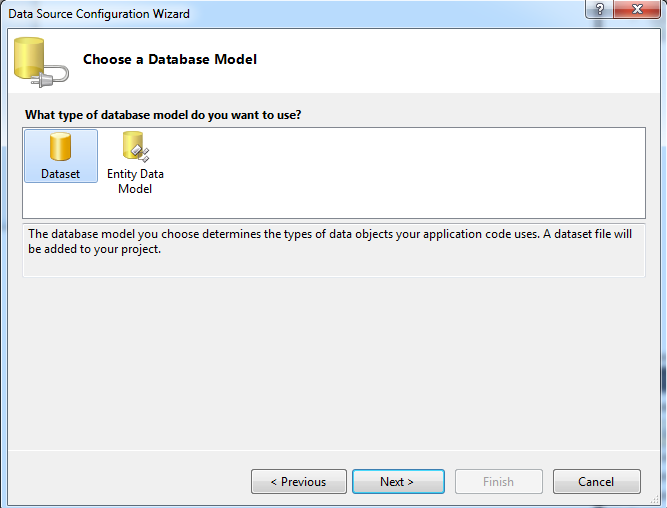


Следующий скриншот: Data Souce Configuration Wizard



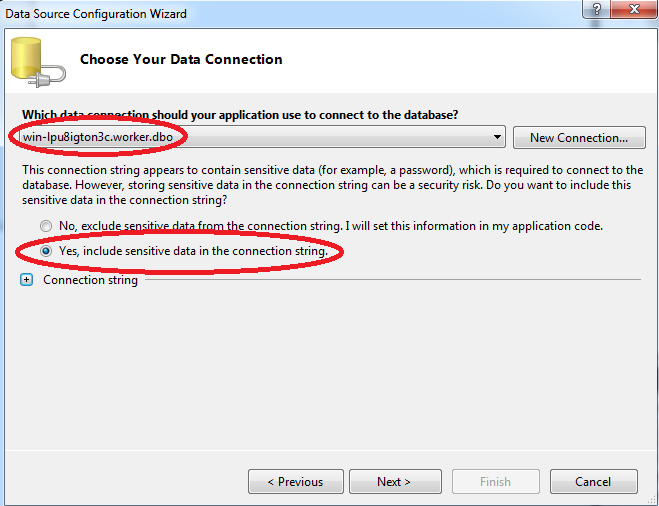
Выбераем Database и нажимаем кнопку "next >"

Следующий скриншот "Shoose a Database Model"



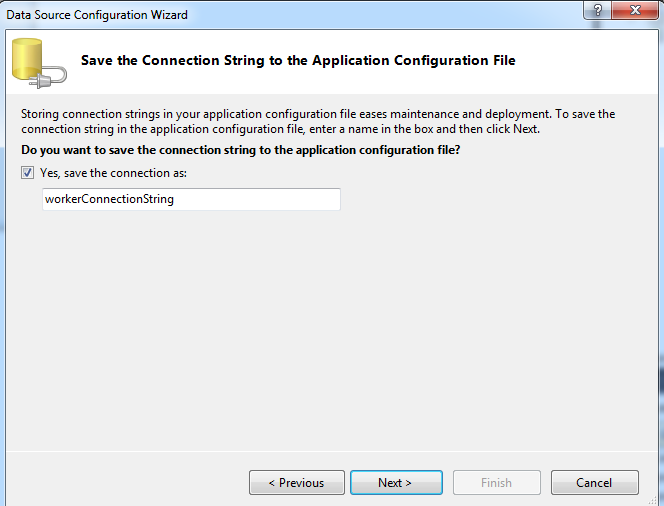
Выбираем "Dataset" и нажимаем кнопку "next >"

Следующий скриншот "Shoose Your Data Connection"



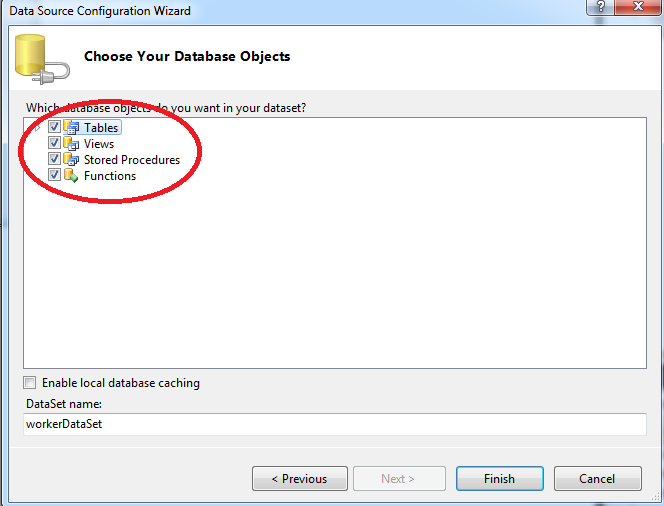
Выбераем недавно созданное подключение и "Yes include sensitive data in the connection string", затем нажимаем "next >";

Следующий скриншот:



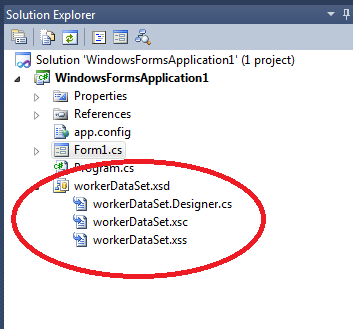
нажимаем "next >";

Следующий скриншот "Choose your Dataset Objects":

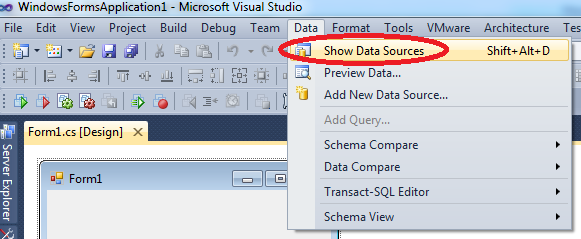


Выбераем нужные нам компоненты: таблицы, представления, хранимые процедуры и функции. И нажимаем кнопку FINISH.

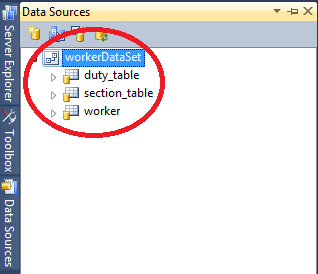
Во вкладке Solution Explorer появляется новый элемент workerDataSet.xsd



Посмотрим источники данных: главное меню ==> Data ==> Show Data Sources



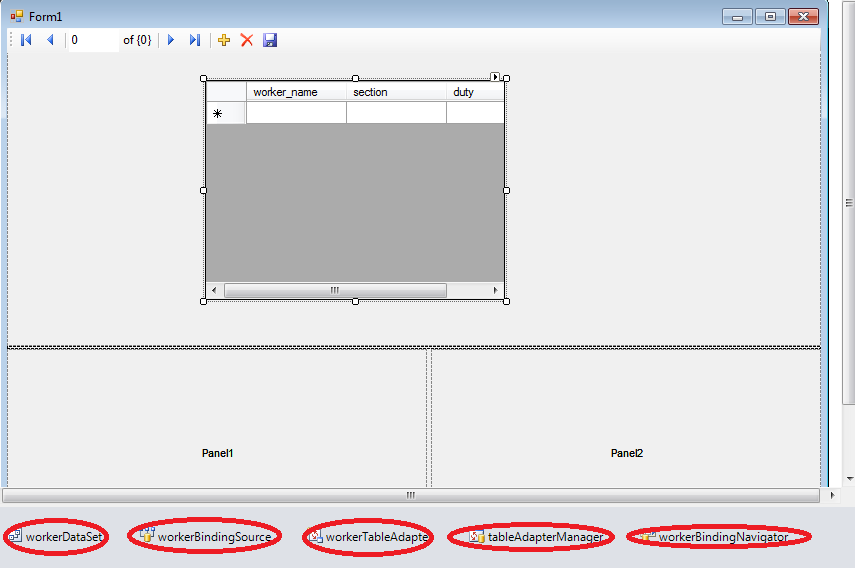
Появляется вкладка Data Source c подключённой нами БД worker:



С помощью возможности Drag&Drop переносим таблицы целиком или конкретные столбцы на рабочую форму.

**5.4. Проектирование клиентского модуля обработки данных.**

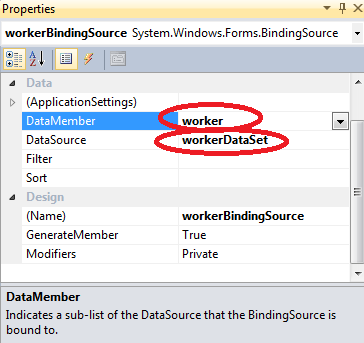
Начинаем с главной информационной таблицей "worker". С помощью возможности DRAG&DROP и переносим её на форму:



Как видно автоматически появляются следующие компоненты:

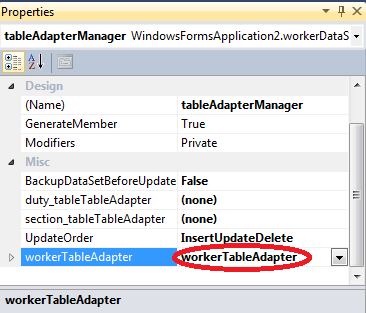
workerDataSet - элемент класса DataSet;

workerBindingSource - элемент класса BindingSource; Основные параметры: DataSource и DataMember:



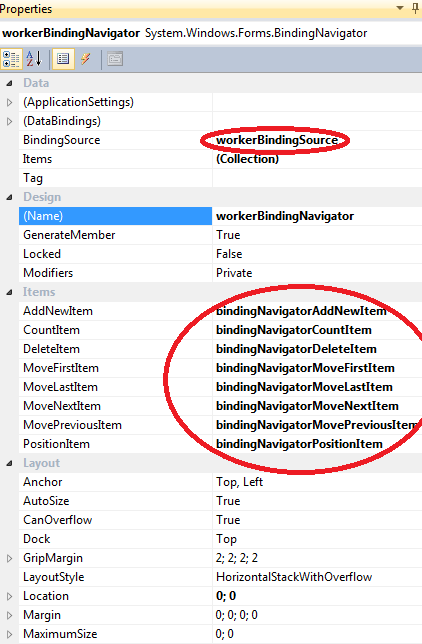
workerTableAdapter - Элемент класса TableAdapter;

tableAdapterManager - Элемент класса TableAdapterManager; Основной параметр: workerTableAdapter;

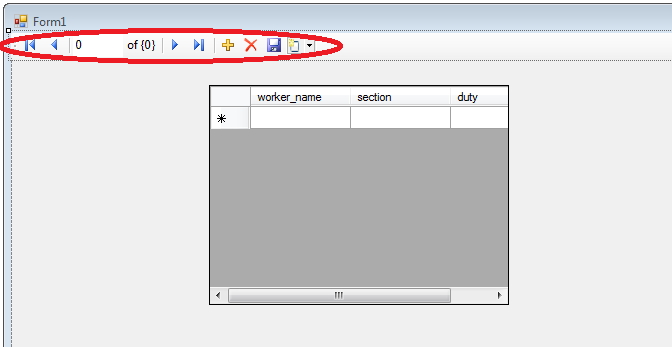


workerBindingNavigator - Элемент класса TableBindingNavigator; Основные параметры:

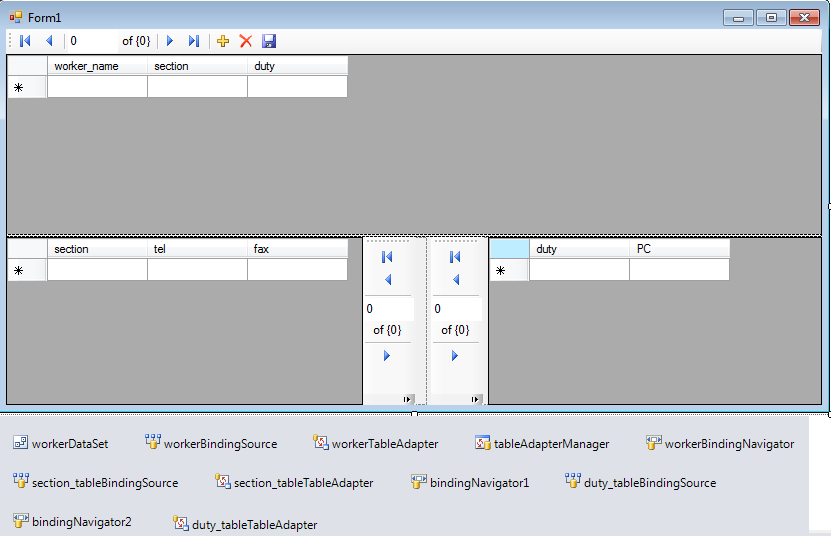
BindingSource, AddNewItem, CountItem, DeleteItem, MoveFirstItem, MoveLastItem, MoveNextItem, MovePreviousItem и PositionItem;



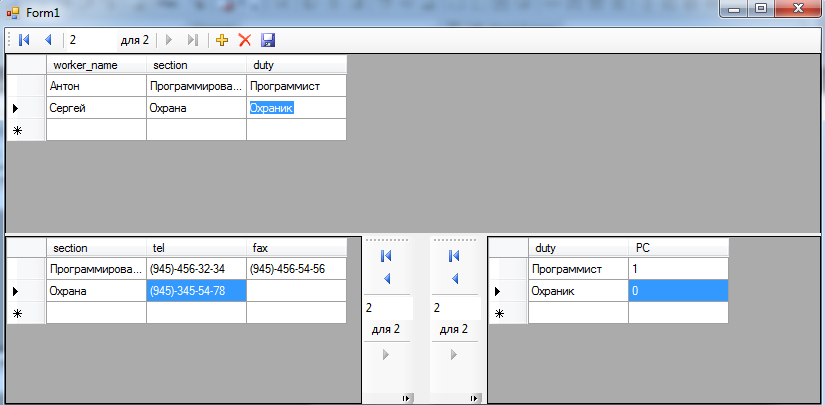
Этот элемент отвечает за редактирование содержания таблиц.



Перетаскиваем (Drag&Drop) остальные таблицы и добавляем 2 элемента TableBindingNavigator и проставляем нужное значение bindingSource: duty\_tableBindingSource и section\_tableBindingSource. Получаем следующую картину:



Всё готово для обработки (редактирование, добавление и удаления) записей, запускаем программу:



добавлены записи во всех таблицах. Работа выполнена.

Провераем число записей в БД:

select count(\*) from worker==> получаем 2;

select count(\*) from section\_table ==> получаем 2;

select count(\*) from duty\_table ==> получаем 2;

Клиентский модуль работает.

Также можно удалять и редактировать все записи.