

УО «Полоцкий государственный университет»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ №8

**к выполнению лабораторной работы
по курсу «Базы данных»
для специальности Программное обеспечение информационных
технологий 1-40 01 01**

ТЕМА: Доступ к данным SQL Server при помощи Server Enterprise Manager

Новополоцк 2008

Методические указания разработали:

Старший преподаватель кафедры технологий программирования Бураченко Ирина Брониславовна

Начальник Центра информационных технологий, старший преподаватель кафедры технологий программирования Бураченко Алексей Леонидович

ТЕМА: Доступ к данным SQL Server при помощи Server Enterprise Manager.

ЦЕЛЬ: Научиться способам доступа к информации, хранимой в базах данных SQL Server, при помощи Server Enterprise Manager, используя средство Query Designer. Научиться импортировать внешние данные в Excel из базы данных SQL Server.

Результат обучения:

После успешного завершения занятия пользователь должен:

- Уметь осуществлять доступ к данным в SQL Server при помощи Server Enterprise Manager.
- Уметь обрабатывать данные в SQL Server при помощи языка Transact-SQL
- Уметь создавать представления (View) средствами SQL Server.
- Уметь разрабатывать хранимые процедуры в SQL Server.
- Уметь осуществлять импорт внешних данных в MS Excel из базы данных SQL Server.

Используемая программа: Microsoft SQL Server 2000, Microsoft Excel.

План занятия:

1. Доступ к данным SQL Server.
2. Доступ к данным при помощи Server Enterprise Manager
3. Представления.
4. Хранимые процедуры.
5. Импорт данных в Excel.

1. ДОСТУП К ДАННЫМ SQL SERVER

Большинство приложений, работающих с базами данных, используют в качестве интерактивного интерфейса программы, созданные в среде Microsoft Visual Basic или Microsoft Access. Эти приложения представляют данные, находящиеся в базах SQL Server и предназначенные для передачи в них, в удобной для пользователя форме. Реальная же обработка данных происходит при помощи языка Transact-SQL, который используют сами интерфейсные приложения для «общения» с SQL Server. Поэтому начинающему разработчику баз данных следует обязательно ознакомиться с возможностями использования языка Transact-SQL для обработки баз данных. Причем нужно сразу понять, что основная работа с Transact-SQL будет происходить не с использованием SQL Server Enterprise Manager, а именно при помощи интерфейсных программ, хотя очень удобно отлаживать, например, представления и хранимые процедуры непосредственно на SQL Server (т.е. с помощью SQL Server Enterprise Manager).

В данной лабораторной работе рассматриваются способы доступа к информации, хранимой в базах данных SQL Server, при помощи утилиты SQL Server Enterprise Manager.

2. ДОСТУП К ДАННЫМ ПРИ ПОМОЩИ SERVER ENTERPRISE MANAGER

SQL Server Enterprise Manager имеет средство Query Designer, очень похожее на конструктор запросов в Microsoft Access. Query Designer позволяет создавать запросы к базе данных посредством различных инструментов. Вам остается только выбирать наиболее подходящий из них. Для работы с Query Designer найдите, например, таблицу **Запасы** базы данных **Фирма** (рис. 1).

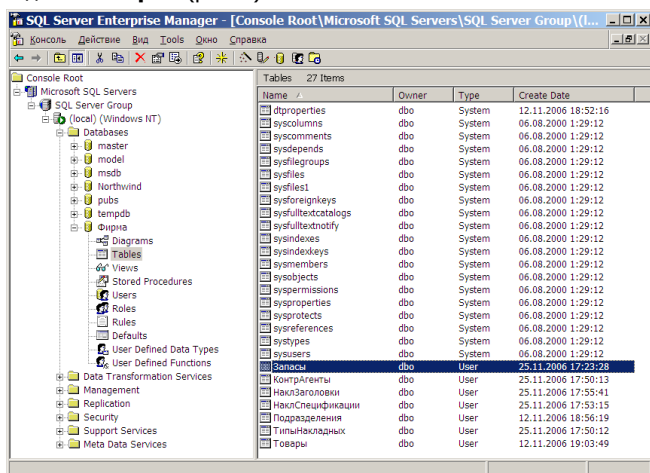


Рис. 1. Для работы с Query Designer найдите таблицу **Запасы** базы данных **Фирма**

Щелкните на значке таблицы **Запасы** правой кнопкой мыши и в контекстном меню выберите **Open Table | Return all rows**. Если вы проделали лабораторные работы и корректно создали базу данных **Фирма**, то неудивительно будет, если вы увидите окно, похожее на представленное на рис. 2.

The screenshot shows the Query Designer window with the 'Запасы' table selected. The table data is displayed in a grid with three columns: 'КодПодразделения', 'КодТовара', and 'Количество'.

КодПодразделения	КодТовара	Количество
1506	006019165803	3000
1506	006019165803	200
1518	006019165803	1200
0429	006019170049	1000
1506	006019170049	500
1518	006019170049	1000
1506	0104414114438	400
1958	0104414114438	3200
1506	010428144348	800
1958	010428144348	4200
0429	036019164741	2000
1506	036019164741	600
1518	036019164741	1400
1506	174500175359	600
1958	174500175359	2200
0429	174500175359	1500
1958	174500175359	2300
0429	186100175232	2000
1958	186100175232	1300

Рис. 2. Таблица **Запасы** базы данных **Фирма** в окне Query Designer

2.1. Панели окна Query Designer

Щелкните на кнопке **Show/Hide SQL Panel** (с текстом «SQL»), и окно **SQL Server Enterprise Manager** будет разделено на две части: в верхней (**SQL Pane**) — SQL-инструкция, в нижней (**Results Pane**) — ее результат (рис. 3). В данном случае инструкция соответствует выбору всех полей и всех строк таблицы **Запасы**.

Чтобы убрать окно с SQL-инструкцией, достаточно щелкнуть на кнопке **Show/Hide SQL Panel**.

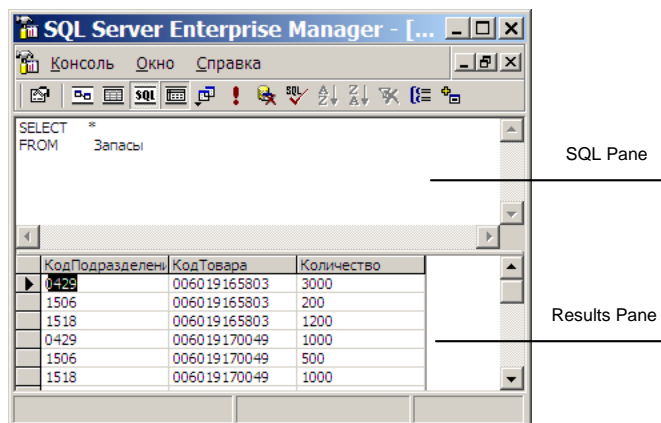


Рис. 3. Две части окна **Query Designer**: в верхней — SQL-инструкция, в нижней — ее результат

Как вы уже могли догадаться, инструкция (можно говорить «команда») **SELECT** является одной из инструкций языка Transact-SQL (или T-SQL) и ее синтаксис не намного отличается от синтаксиса подобной инструкции SQL Microsoft Jet. Для несложных SQL-инструкций вам вполне может хватить той информации по языку SQL, которую вы получили на лекциях, но для серьезной работы, конечно, следует приобрести литературу по SQL Server с описанием Transact-SQL.

Кроме кнопки **Show/Hide SQL Panel**, на панели инструментов окна **SQL Server Enterprise Manager** находятся кнопки **Show/Hide Diagram Pane**, **Show/Hide Grid Pane**, **Show/Hide Results Pane** (рис. 4), первые две из которых предназначены для отображения панелей формирования таблиц с информацией из базы данных.

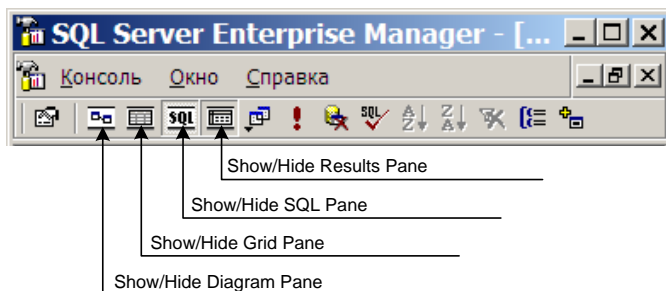


Рис. 4. Кнопки панели инструментов окна **SQL Server Enterprise Manager**

2.2. Панель диаграмм (Diagram Pane)

На этой панели отображается запрос к базе данных в виде, аналогичном диаграмме базы данных. Щелкните, например, на кнопке **Show/Hide Diagram Pane**. В результате появляется панель **Diagram Pane**, на которой схематично отображается текущий запрос к базе данных. Из рис. 5 видно, что текущий запрос должен вернуть содержимое всех полей таблицы **Запасы**.

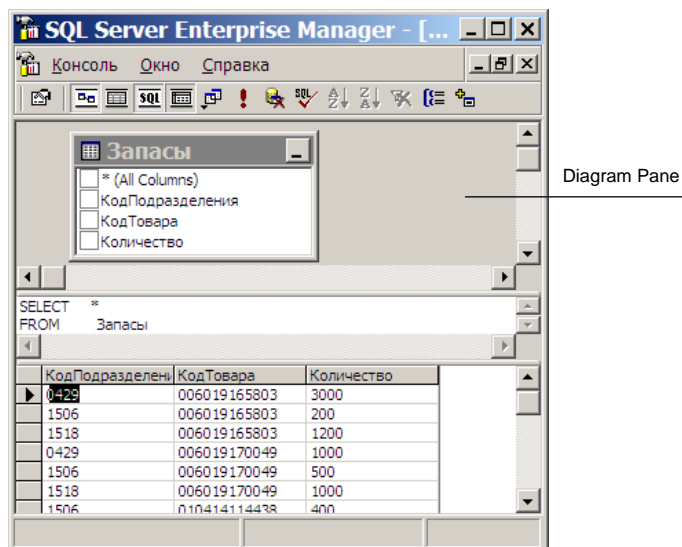


Рис. 5. Текущий запрос должен вернуть содержимое всех полей таблицы **Запасы**

Чтобы получить более содержательную диаграмму текущего запроса, введите в **SQL Pane** следующую инструкцию:

```
SELECT а.НаимТовара, б.Количество
FROM Товары а, Запасы б
WHERE а.КодТовара=б.КодТовара AND б.КодПодразделения='0429'
```

и щелкните кнопку **Run**. Query Designer изменит содержимое и **SQL Pane**, и **Diagram Pane** (рис. 6). Обратите внимание на то, как преобразована SQL-инструкция. Query Designer «умело» использовал предложение **INNER JOIN** для объединения двух таблиц по ключевому полю **КодТовара**.

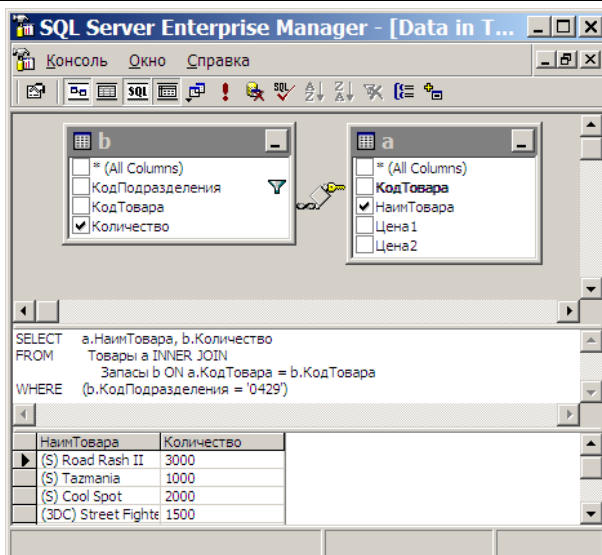


Рис. 6. Использование предложения **INNER JOIN** для объединения двух таблиц по ключевому полю **КодТовара**

Самое удивительное — в том, что **Diagram Pane** предназначена не только для графического отображения текущего запроса, но и для графического построения запроса. Щелкните, например, в таблице **a (Товары)** флажок **Цена1**. В **SQL Pane** к SQL-инструкции к предложению **SELECT** будет добавлено поле **Цена1** (рис. 7).

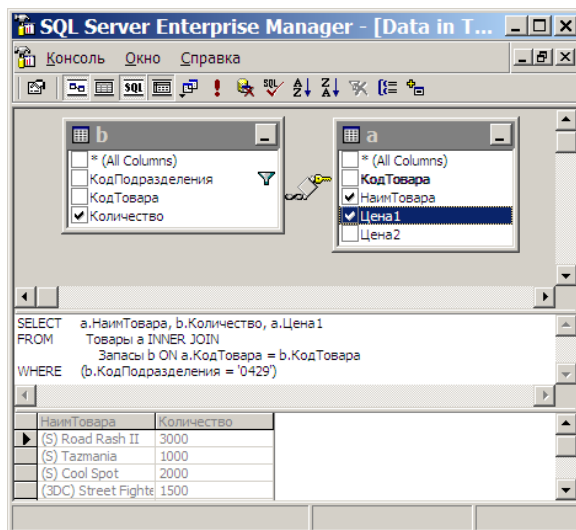


Рис. 7. Изменения в **Diagram Pane** сразу же отображаются в **SQL Panel**

Для выполнения запроса остается щелкнуть кнопку **Run** (результат выполнения запроса — на рис. 8).

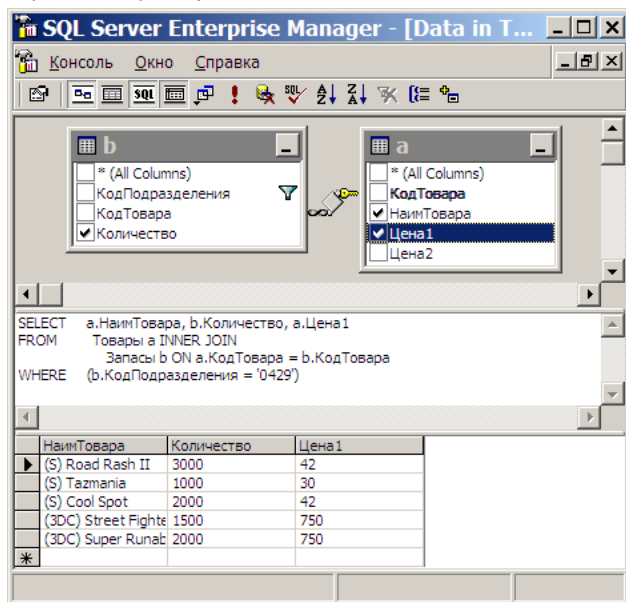


Рис. 8. Результат выполнения запроса, сформированного с использованием **Diagram Pane**

Построим при помощи **Diagram Pane** простой запрос. Удалите с **Diagram Pane** все таблицы. Это можно сделать, щелкнув на заголовке таблицы правой кнопкой мыши и выбрав из контекстного меню опцию **Remove**. Поместите на **Diagram Pane** таблицы **Товары**, **Запасы** и **Подразделения**. Это можно сделать, щелкнув на правой кнопкой мыши и выбрав в контекстном меню опцию **Add Table** для вызова одноименного диалогового окна (рис. 9).

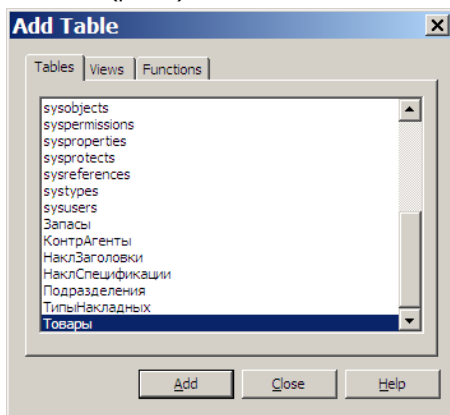


Рис. 9. Окно **Add Table** позволяет добавлять объекты на **Diagram Pane**

Быстро удалить объекты с панели диаграмм можно следующим образом. Очистить панель с SQL-инструкцией, а затем просто щелкнуть на панели диаграмм, которая будет приведена в «соответствие» с SQL-инструкцией.

После добавления указанных таблиц окно **SQL Enterprise Manager** будет похожим на отображенное на рис. 10. Связи между таблицами были добавлены автоматически — **Enterprise Manager** «помнит» недавнюю нашу работу по созданию объекта **DIAGRAM1**.

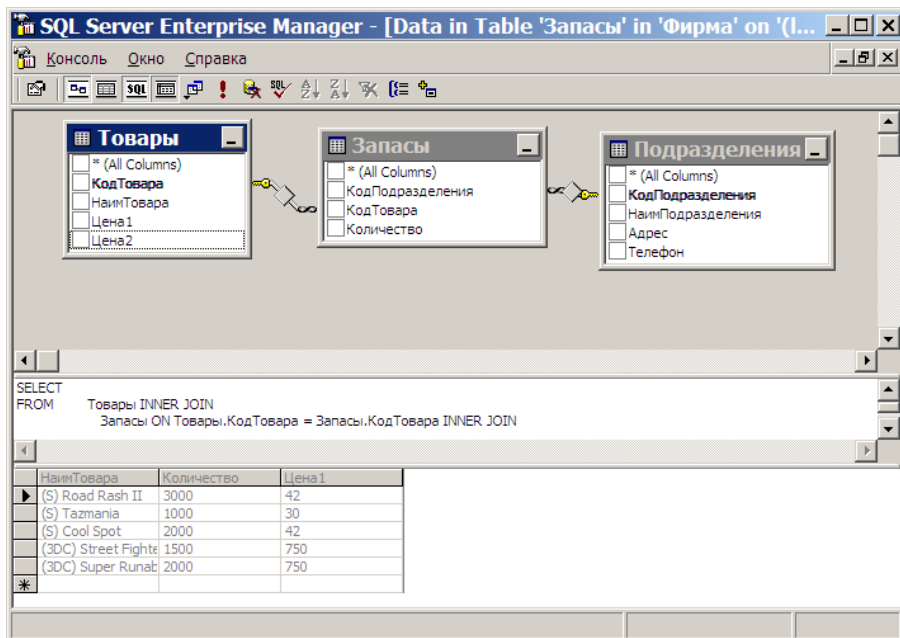


Рис. 10. Окно **SQL Enterprise Manager** после добавления таблиц на **Diagram Pane**

Информация на **Results Pane** в данный момент не соответствует запросу (кнопка **Run** еще не использовалась), поэтому эта панель затенена. Панель для работы с SQL-инструкциями — **SQL Pane** — соответствует графической информации на **Diagram Pane**.

Построим следующий запрос: наименование товара, количество, цена, его местонахождение. Для этого последовательно устанавливаем флажки на **Diagram Pane** (наблюдайте за изменениями на **SQL Pane**):

- **НаимТовара** в таблице **Товары**;
- **Количество** в таблице **Запасы**;
- **Цена1** в таблице **Товары**;
- **НаимПодразделения** в таблице **Подразделения**.

В результате работы с **Diagram Pane** можно увидеть сформированную SQL-инструкцию на **SQL Pane** (рис. 11). Между прочим, инструкцию можно подправить прямо на **SQL Pane**.

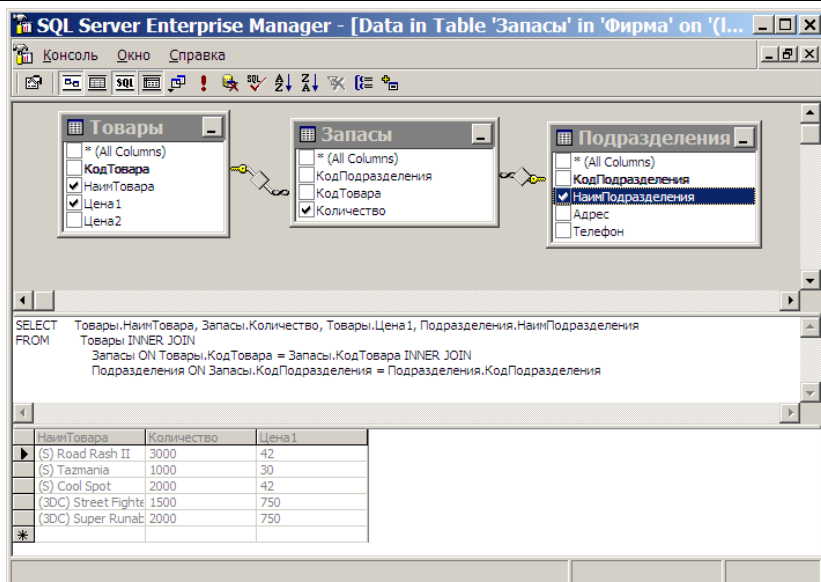


Рис. 11. В результате работы с **Diagram Pane** можно увидеть сформированную SQL-инструкцию на **SQL Pane**

Не долго думая, можно щелкнуть на кнопке **Run** для приведения данных на **Results Pane** в соответствии с запросом на **SQL Pane** (рис. 12).

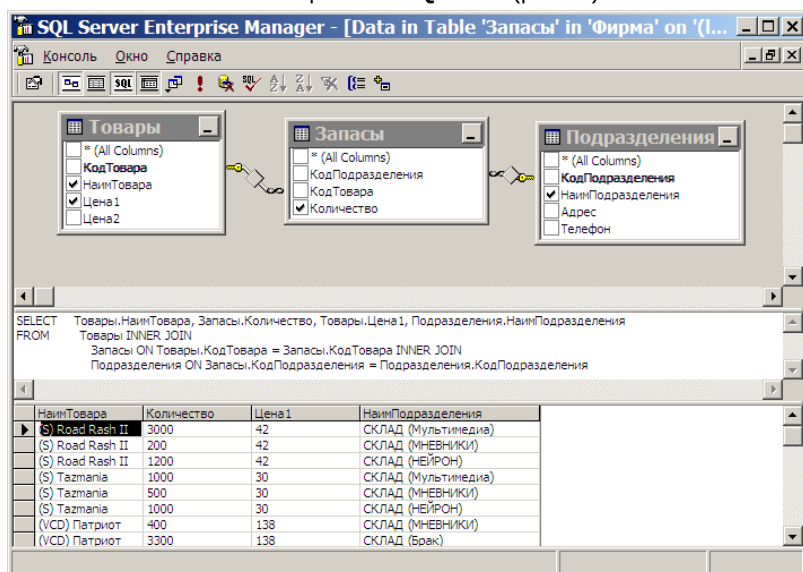


Рис. 12. Данные на **Results Pane** соответствуют запросу на **Diagram Pane** и **SQL Pane**

2.3. Панель сетки (Grid Pane)

Панель сетки аналогична (по виду) Query Design Grid в Microsoft Access и позволяет изменять заголовки столбцов в запросе, задавать порядок отображения строк, условия отбора строк. Для отображения **Grid Pane** следует щелкнуть на кнопке **Show/Hide Grid Pane**. Содержимое этой панели зависит от текущего запроса (рис. 13).

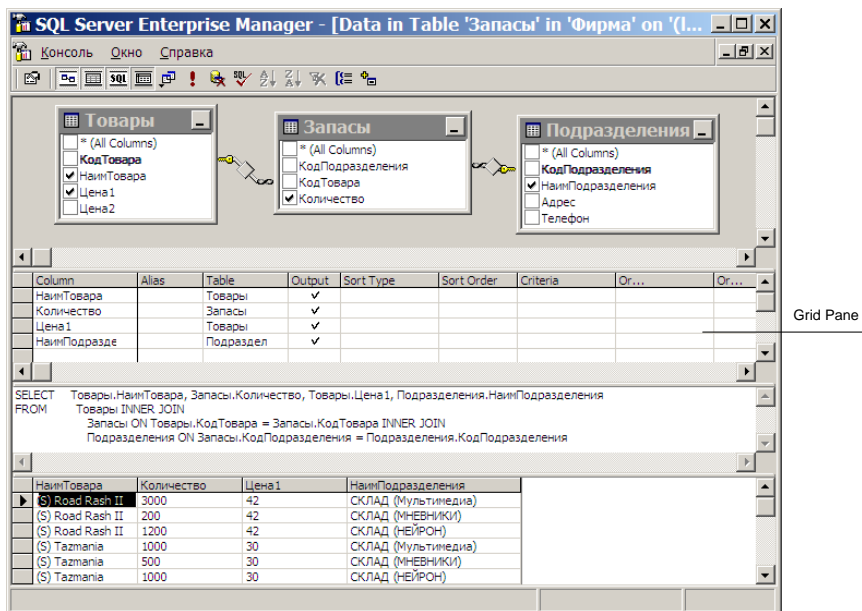


Рис. 13. Содержимое этой панели зависит от текущего запроса

Для рассмотрения возможностей **Grid Pane** скройте **Diagram Pane**. Установите, например, режим сортировки для наименований товаров: щелкните в поле **Sort Type** и в появившемся меню выберите **Ascending**. При этом к SQL-инструкции будет добавлено предложение

ORDER BY Товары.НаимТовара

Для выполнения нового запроса щелкните кнопку **Run**. Результат выполнения запроса представлен на рис. 14. Нетрудно понять, что при помощи этой панели можно задать сортировку по нескольким полям.

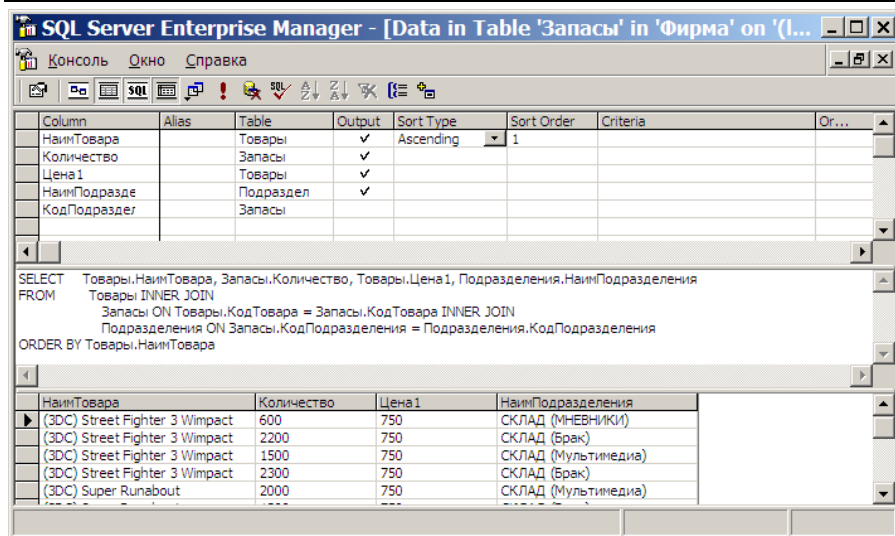


Рис. 14. Использование **Grid Panel** для сортировки записей запроса

Grid Panel позволяет задать критерии для выводимых записей. Например, запросите вывод записей только для подразделения, код которого совпадает со строкой "0429". Во второй строке (напротив таблицы **Запасы**) столбца **Criteria** введите **КодПодразделения='0429'**, как показано на рис. 15.

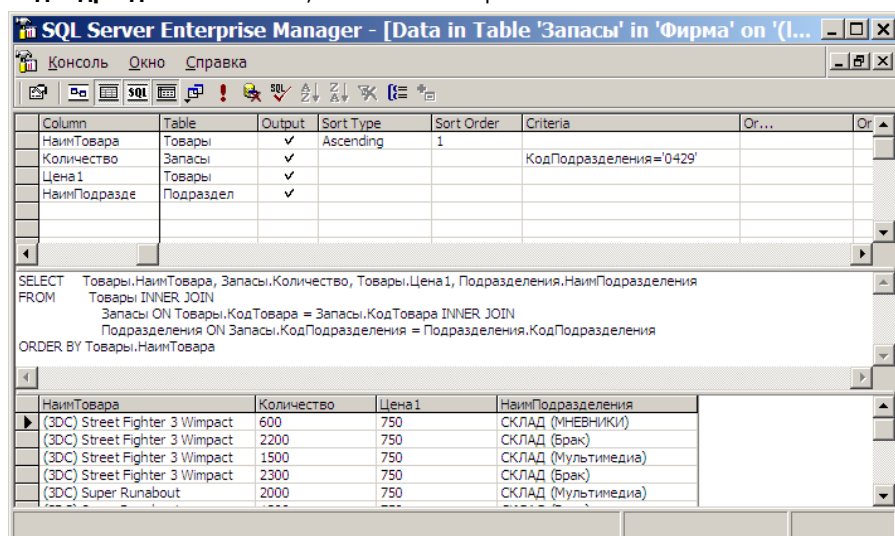


Рис. 15. Использование **Grid Panel** для получения записей, удовлетворяющих некоторому критерию

Как всегда, Query Designer изменит содержимое и **SQL Pane**, и **Grid Pane** (рис. 16).

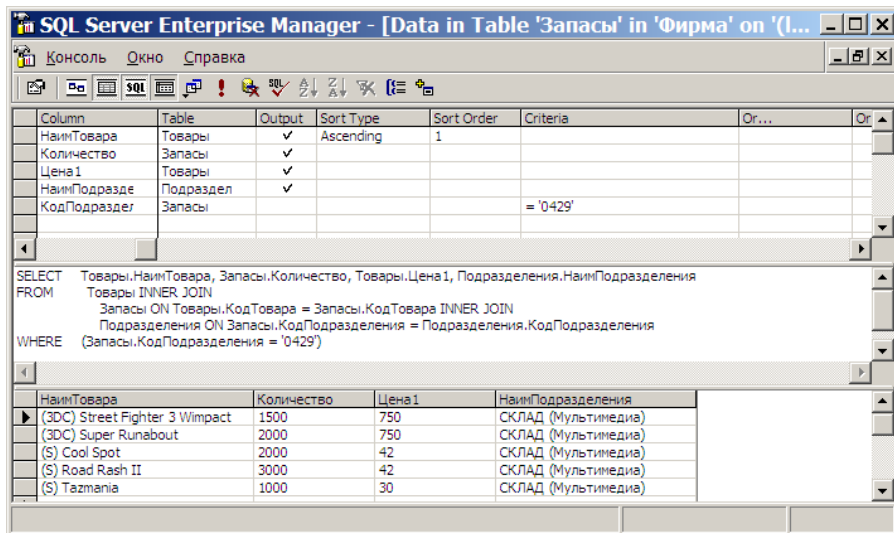


Рис. 16. Изменение содержимого **SQL Pane** и **Grid Pane** при выполнении запроса

В заключение следует отметить, что наиболее гибким и мощным способом создания запросов является использование панели с SQL-инструкциями. Всегда для любой другой панели можно придумать невыполнимую или почти невыполнимую задачу. Для **SQL Pane** такую задачу придумать сложно.

3. ПРЕДСТАВЛЕНИЯ

Представления (*Views*) — это способ извлечения данных из базы, результатом которого является (виртуальная) таблица, полученная из (возможно) многих таблиц базы данных. Свои первые представления вы можете создавать в SQL Enterprise Manager. Это удобно и потому, что можно легко и быстро получить первые результаты от применения этой технологии, и потому, что синтаксис написанных здесь представлений сразу же проверяется.

Утилита SQL Enterprise Manager позволяет создать новое представление двумя способами: при помощи **Create View Wizard** (мастер создания представления) и командой **New View** (конструктор представлений — **View Designer**).

3.1. Создание представления при помощи View Designer

Для создания представления при помощи **View Designer** выделите на «дереве» базы данных **Фирма** значок **Views** и при помощи правой кнопки мыши вызовите контекстное меню. Выберите из него опцию **New View**. Окно, которое вы увидите на экране, практически, ни чем не отличается от того что рассмотренного (рис. 17) — здесь те же панели (**SQL Pane**, **Diagram Pane**, **Grid Pane** и **Results Pane**), отображение и скрытие которых управляется теми же кнопками (**Show/Hide SQL Pane**, **Show/Hide Diagram Pane**, **Show/Hide Grid Pane** и **Show/Hide Results Pane**). Это не очень удивляет, если понимать, что представление — не более чем именованный

запрос. Сначала мы создаем запрос при помощи всех известных способов, а потом сохраняем его (под некоторым именем) для дальнейшего использования.

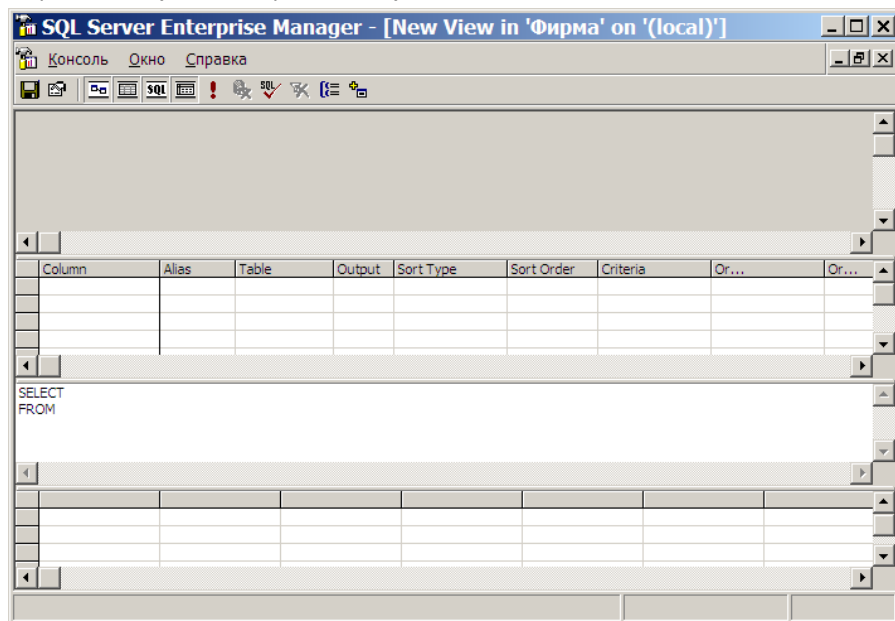


Рис. 17. Окно ***View Designer***, практически, ни чем не отличается от окна ***Query Designer***

Поместите на **Diagram Pane**, например, таблицу **Товары**. Установите флажки для полей **КодТовара** и **НаимТовара** (такое представление может пригодиться приложениям, которым для работы необходимо в диалоговом окне представлять наименование товара, а в программе — оперировать кодами товаров). На **Diagram Pane** тут же появится SQL-инструкция, соответствующая изменениям на **Diagram Pane** (рис. 18). Это представление можно уже сохранить.

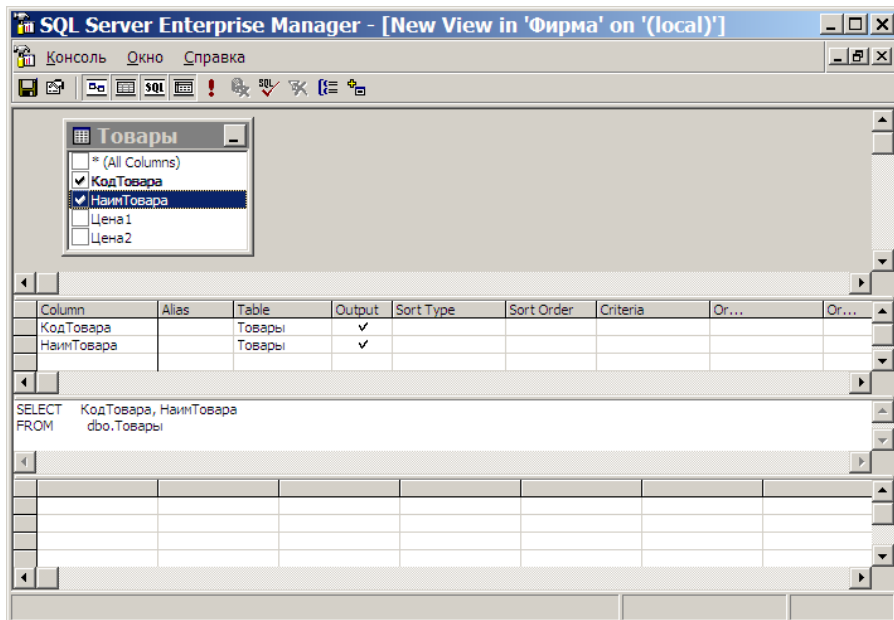


Рис. 18. Это представление уже можно сохранить

Щелкните кнопку **Save** на панели инструментов и в окне **Save As** введите наименование представления (рис. 19)

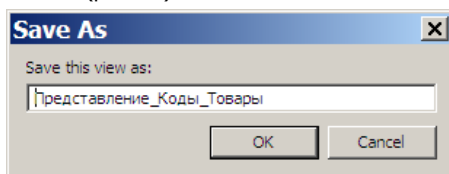


Рис. 19. Окно **Save As**, предназначенное для ввода наименования представления

Если вы не собираетесь создавать следующее представление после только что созданного, то необязательно для сохранения представления щелкать кнопку **Save**. Можно просто закрыть **View Designer**. При наличии несохраненного представления вам обязательно будет предложено сохранить его (задать для него наименование).

3.2. Использование представления

Созданное однажды представление можно в дальнейшем использовать точно так же, как таблицу. Представление можно открыть в Enterprise Manager с использованием View Designer и даже включить в другие запросы.

Щелкните дважды на значке **Views** на «дереве» базы данных **Фирма** для получения списка всех представлений (рис. 20).

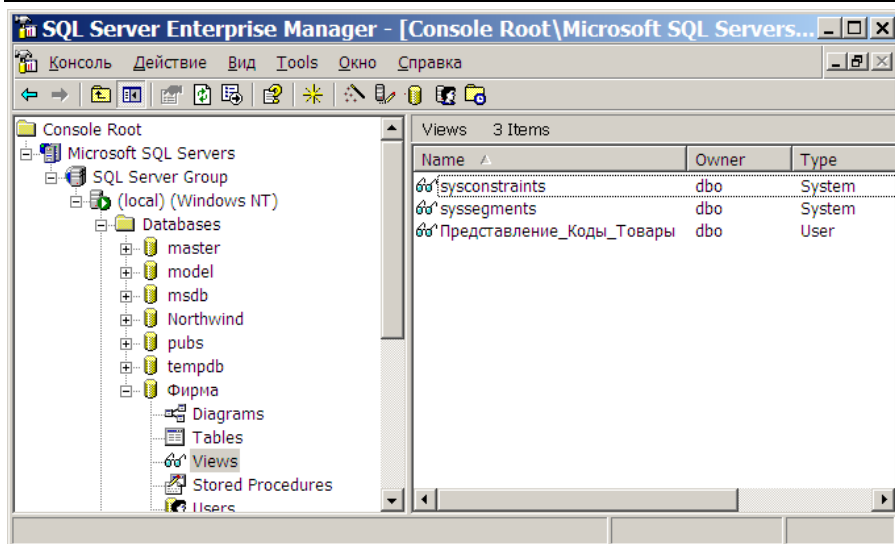


Рис. 20. Список всех представлений базы данных **Фирма**

Для получения данных, соответствующих представлению **Представление_Коды_Товары**, вызовите контекстное меню для этого представления и выберите **Open View | Return all rows** (рис. 21).

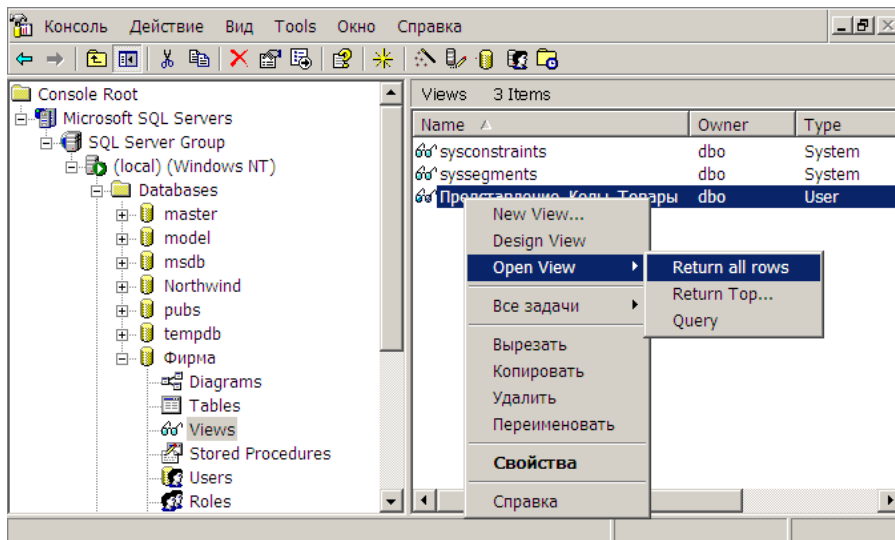


Рис. 21. Получение данных, соответствующих представлению **Представление_Коды_Товары**, из контекстного меню **Open View | Return all rows**

В результате будет выдана таблица с полями **КодТовара** и **НаимТовара** из таблицы **Товары** (рис. 22). Таким образом, использование представления мало чем отличается от использования таблицы. Более того, если посмотреть на SQL-инструкцию и диаграмму (используя кнопки **Show/Hide SQL Pane** и **Show/Hide Diagram Pane**), которые создаются при этом (рис. 23), то можно сказать, что отличий вообще нет.

The screenshot shows the 'SQL Server Enterprise Manager - [Data...]' window. The 'Консоль' (Console) pane displays a table with two columns: 'КодТовара' and 'НаимТовара'. The table contains 13 rows of data.

КодТовара	НаимТовара
006019165803	(S) Road Rash II
006019170049	(S) Tasmania
010414114438	(VCD) Патриот
010415090732	(VCD) Пастырь
010415090937	(VCD) Шоссе в ник
010428144348	(VCD) Доберман
028019095823	(VCD) Diber Marouani
028019100018	(VCD) Wham! The b
028019100222	(VCD) Голая мише
036019164741	(S) Cool Spot
174500175359	(3DC) Street Fighte
174500175521	(3DC) Star Gladiato
186100175232	(3DC) Super Runab

Рис. 22. *Результат выполнения представления
Представление_Коды_Товары*

The screenshot shows the 'SQL Server Enterprise Manager - [Data in ...]' window. The 'Представ...' (View) pane shows a diagram with a table named 'Представление_Коды_Товары'. The 'SQL' pane shows the following query:

```
SELECT *
FROM Представление_Коды_Товары
```

The 'Table' pane shows a table with two columns: 'КодТовара' and 'НаимТовара'. The table contains 13 rows of data.

КодТовара	НаимТовара
006019165803	(S) Road Rash II
006019170049	(S) Tasmania
010414114438	(VCD) Патриот
010415090732	(VCD) Пастырь
010415090937	(VCD) Шоссе в никуда
010428144348	(VCD) Доберман
028019095823	(VCD) Diber Marouani
028019100018	(VCD) Wham! The best of

Рис. 23. *SQL-инструкция и диаграмма для получения результата представления ничем не отличается от SQL-инструкции и диаграммы для таблицы*

Чтобы окончательно убедиться в том, работа с представлением очень похожа на работу с таблицей, построим запрос, включающий представление и таблицы. Добавим на **Diagram Pane** таблицу **Запасы** и **Подразделения** (рис. 24).

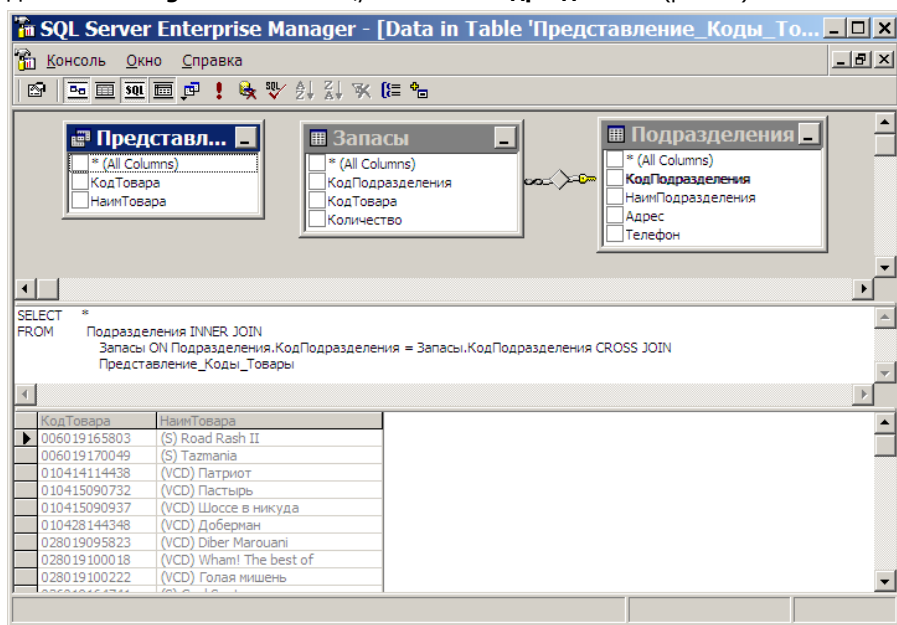


Рис. 24. Добавим на **Diagram Pane** таблицу **Запасы** и **Подразделения**

Установим связь между представлением **Представление_Коды_Товары** и таблицей **Запасы**. Эта операция выполняется так же, как и при построении диаграммы: нужно щелкнуть на поле **КодТовара** на представлении и, не отпуская кнопку мыши, перетащить курсор на одноименное поле таблицы **Запасы** (рис. 25).

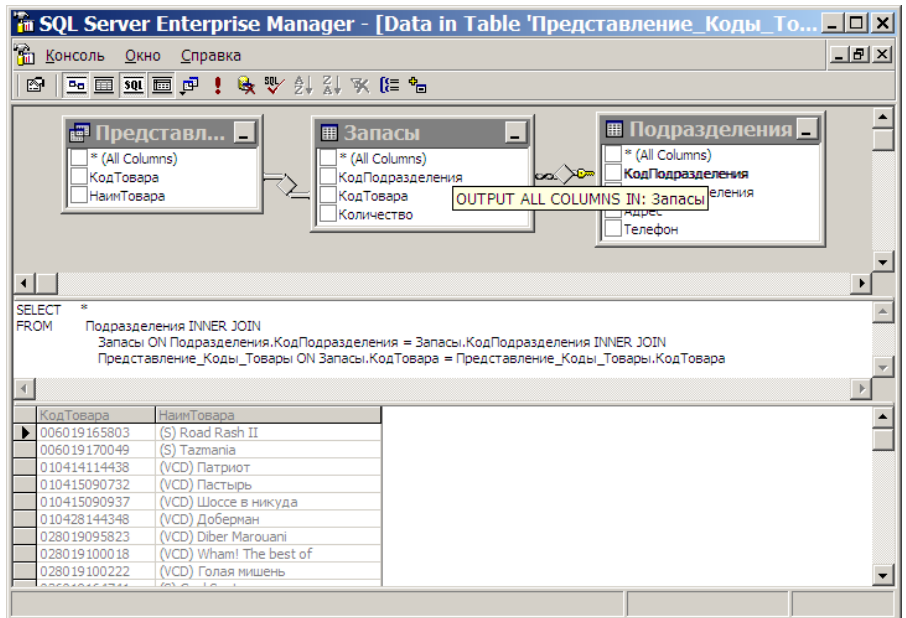


Рис. 25. Установим связь между представлением **Представление_Коды_Товары** и таблицей **Запасы**

Остается указать поля для вывода в запросе — щелкните последовательно:

- поле **НаимТовара** в представлении **Представление_Коды_Товары**;
- поле **Количество** в таблице **Запасы**;
- поле **НаимПодразделения** в таблице **Подразделения**.

Уберите из SQL-инструкции символ звездочки (*). Теперь запрос с использованием представления готов (рис. 26). Остается щелкнуть кнопку **Run** для получения результата (рис. 27).

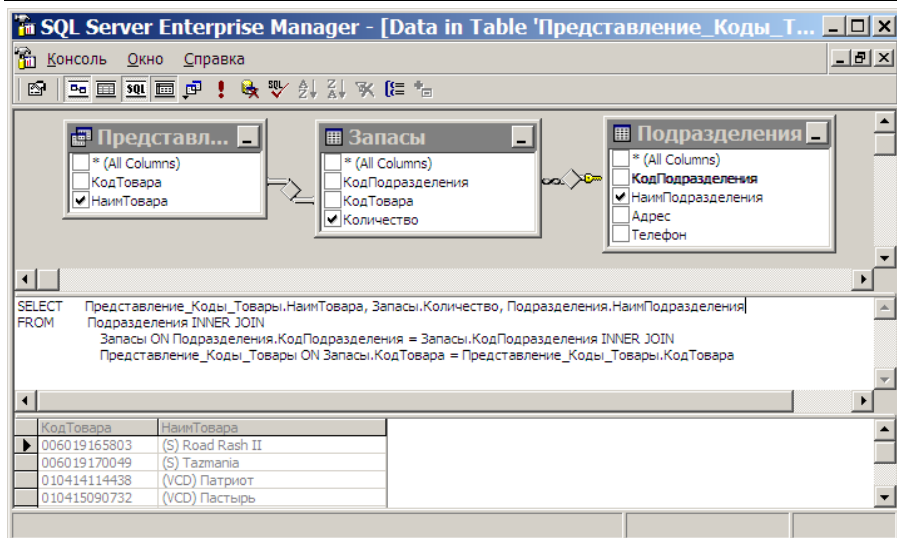


Рис. 26. Пример запроса с использованием представления

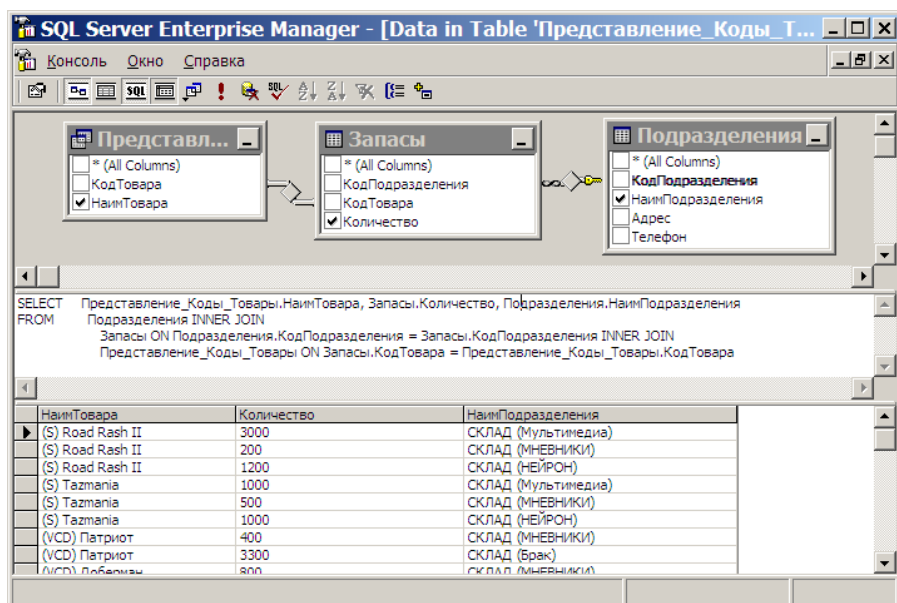


Рис. 27. Результат выполнения запроса с использованием представления

3.3. Проверка синтаксиса представления

Если после получения списка представлений щелкнуть (дважды) на некотором представлении (или в контекстном меню выбрать **Properties**), то на экране появится

диалоговое окно **View Properties - <наименование представления>**, как, например, на рис. 28.

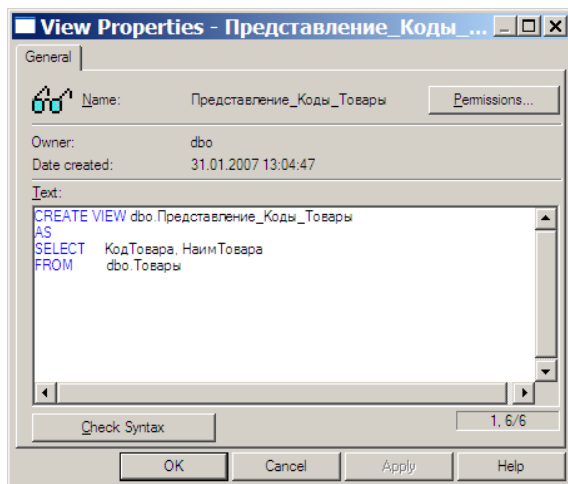


Рис. 28. Окно свойств представления

В этом окне можно изменить SQL-инструкцию представления и проверить синтаксис новой инструкции. Если новая инструкция не содержит ошибок, то после щелчка на кнопке **Check Syntax** Enterprise Manager выдаст положительный результат (рис. 29), иначе — отрицательный (рис. 30).

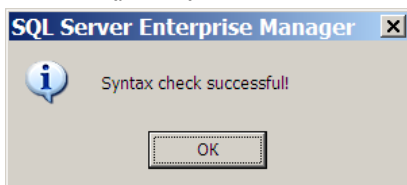


Рис. 29. Если новая инструкция не содержит ошибок, то после щелчка на кнопке **Check Syntax** Enterprise Manager выдаст положительный результат

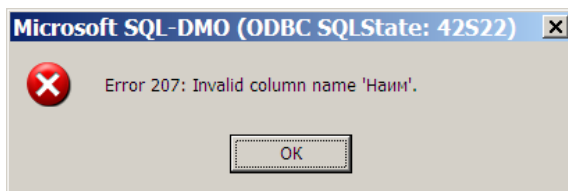


Рис. 30. Если новая инструкция содержит ошибки, то после щелчка на кнопке **Check Syntax** Enterprise Manager сообщает об ошибке

Если новая SQL-инструкция для представления не содержит ошибок, ее можно сохранить, используя кнопку **Apply**. Инструкцию с ошибками сохранить нельзя.

3.4. Создание представления при помощи Create View Wizard

Для создания представления при помощи команды **Create View Wizard** щелкните кнопку **Run a Wizard** на панели инструментов Enterprise Manager (рис. 31). Enterprise Manager отображает окно **Select Wizard** (рис. 32).



Рис. 31. Кнопка **Run a Wizard** на панели инструментов Enterprise Manager

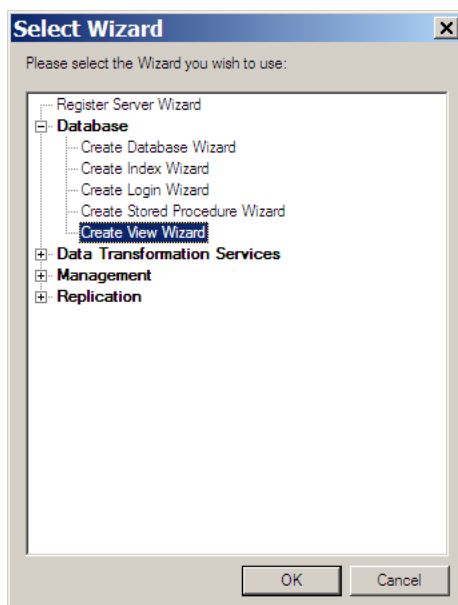


Рис. 32. Enterprise Manager отображает окно **Select Wizard**

Если необходимо, раскройте раздел **Database**. Выберите **Create View Wizard** и щелкните кнопку **OK**. Enterprise Manager, как обычно, «поприветствует» вас (рис. 33). Щелкните кнопку **Next** в окне **Create View Wizard**, если не передумали создавать представление.

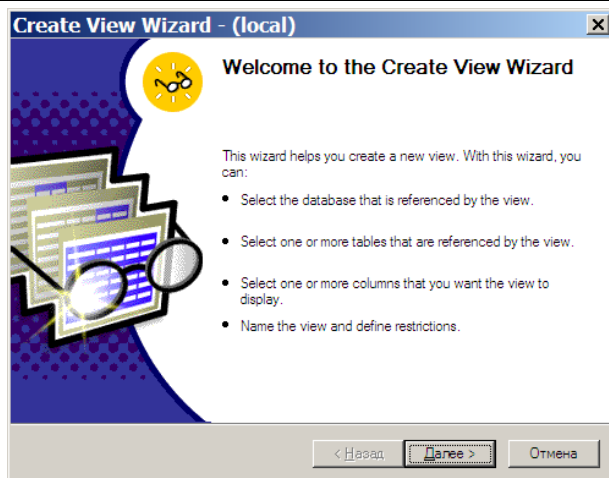


Рис. 33. Щелкните кнопку **Next** в этом окне, если не передумали создавать представление

В следующем окне (рис. 34) можно изменить базу данных, для которой создается представление. Здесь также следует щелкнуть кнопку **Next**.

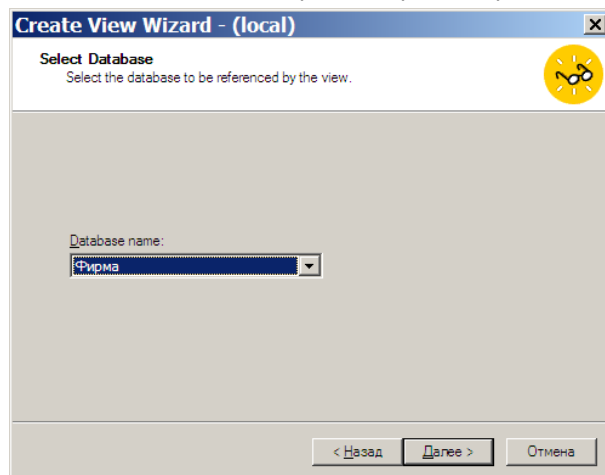


Рис. 34. В этом окне можно изменить базу данных, для которой создается представление

Теперь необходимо указать таблицы, которые будут использоваться в SQL-инструкции создаваемого представления. Укажите таблицы **Запасы**, **Подразделения** и **Товары** для получения SQL-инструкции, которая ранее использовалась при работе с **Diagram Pane** (рис. 35) и щелкните кнопку **Next**.

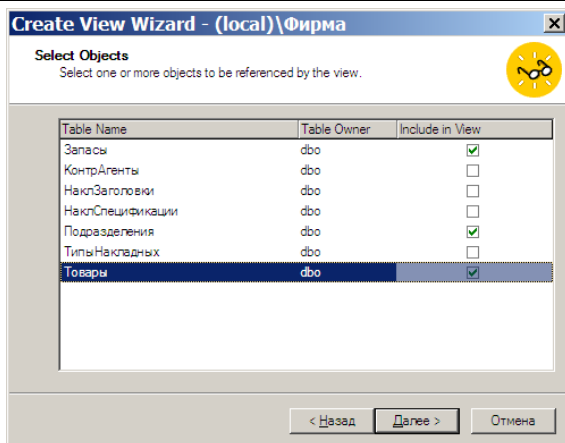


Рис. 35. Укажите таблицы **Запасы**, **Подразделения** и **Товары**

Далее нужно указать поля, данные из которых попадут в запрос (рис. 36). К сожалению, здесь нельзя определить порядок появления полей в предложении **SELECT** SQL-инструкции.

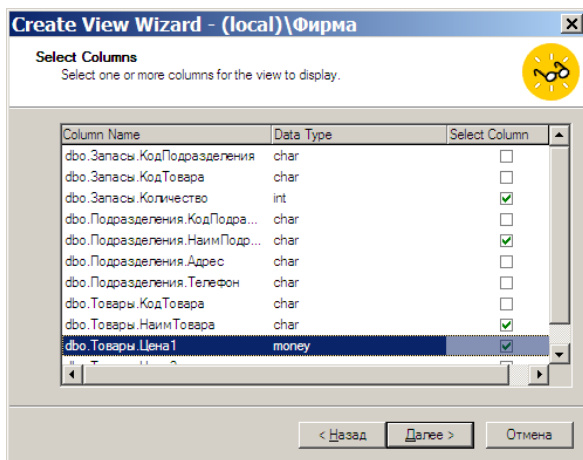


Рис. 36. Здесь нужно указать поля, данные из которых попадут в запрос

Create View Wizard позволяет вставить в SQL-инструкцию предложение **WHERE** (рис. 37). Мы не будем использовать эту возможность, поэтому нужно просто щелкнуть кнопку **Next**.

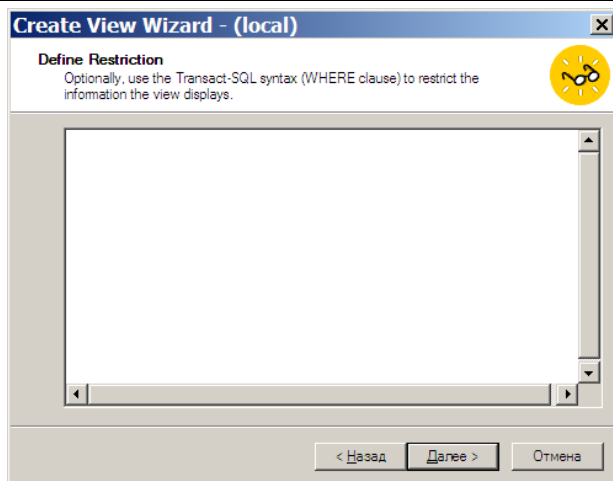


Рис. 37. *Create View Wizard* позволяет вставить предложение **WHERE**

В следующем окне необходимо указать имя нового представления или оставить то, которое предлагается по умолчанию (рис. 38). Поскольку в данном случае у нас нет никакой цели, кроме тестирования Create View Wizard, оставляем имя по умолчанию. При разработке же представлений для «настоящего» приложения следует давать такие наименования представлениям, чтобы позже можно было бы легко вспомнить их назначение.

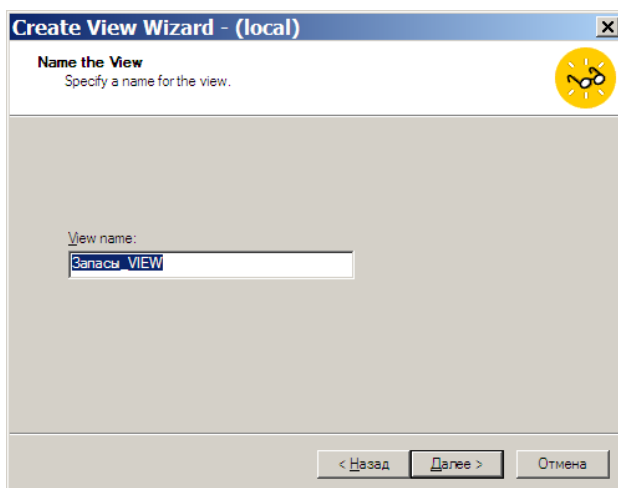


Рис. 38. *В этом окне необходимо указать имя нового представления*

В следующем окне (рис. 39) отображается текстовое значение представления — SQL-инструкция, которую можно отредактировать, прежде чем нажать на кнопку **Finish**, завершающую построение представления. Поскольку нас, действительно, не

устроить такой порядок выводимых полей, следует поменять их местами. Это нетрудно сделать используя приемы копирования и вставки.

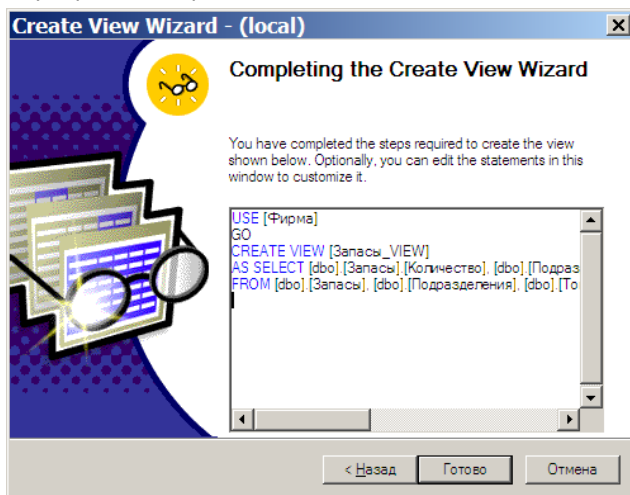


Рис. 39. Текстовое значение представления

После того как представление отредактировано (рис. 40), можно щелкнуть на кнопке **Finish**. Если никаких проблем с синтаксисом представления не возникает, Create View Wizard выдаст сообщение об окончании создания представления (рис. 41).

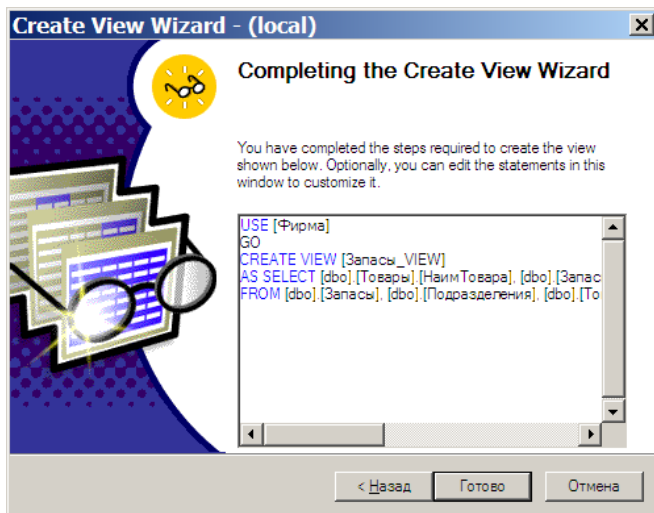


Рис. 40. После того как представление отредактировано, можно щелкнуть на кнопке **Finish**

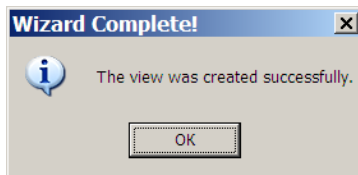


Рис. 41. Если никаких проблем с синтаксисом представления не возникает, **Create View Wizard** выдаст сообщение об окончании создания представления

Теперь новое представление можно увидеть в списке представлений базы данных Фирма (рис. 42) и использовать в дальнейшем.

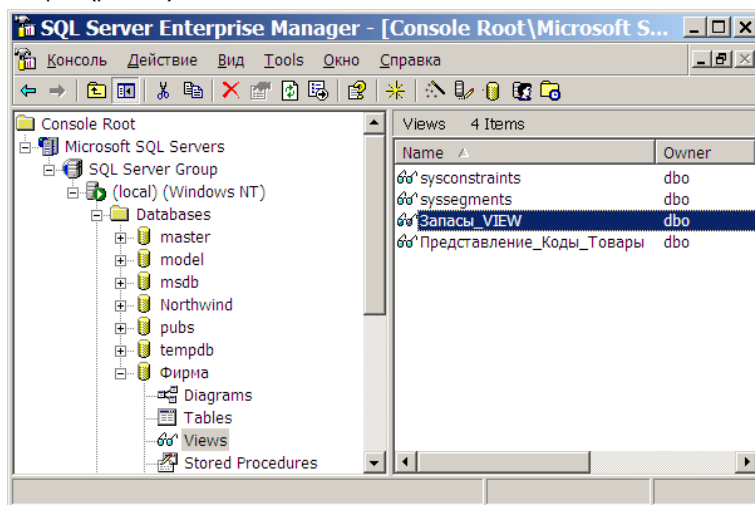


Рис. 42. Новое представление можно увидеть в списке представлений базы данных **Фирма**

4. ХРАНИМЫЕ ПРОЦЕДУРЫ

Хранимые процедуры (*Stored Procedures*) — это подпрограммы, работающие на сервере. Одним из результатов работы хранимой процедуры может быть возвращение таблицы подобно тому, как это делает представление. Как и обычная подпрограмма, хранимая процедура может принимать аргументы и возвращать значение.

Хранимые процедуры являются наиболее важной и широко рекламируемой особенностью SQL Server и представляют собой скомпилированные и сохраненные в базе именованные последовательности операторов Transact-SQL. Причем к операторам языка, используемым в процедурах, относятся не только инструкции типа **SELECT**, **INSERT**, **UPDATE** и т.д., но и такие операторы, как **IF ELSE**, **WHILE** и др. Кроме того, в Transact-SQL, как в обычном языке используются переменные, операции и функции. Описание языка Transact-SQL — это тема отдельной книги.

Разработка комплекса программ (хранимых процедур) — это довольно перспективное направление развития программного обеспечения фирмы, поскольку хранимые процедуры могут содержать операторы, которые взаимодействуют с

программным обеспечением, расположенным на сервере вне самой базы данных. Например, хранимая процедура может послать почтовое сообщение другому пользователю или запустить на сервере программу пользователя. Используя такие возможности SQL Server, можно создать на его основе сервер прикладных задач.

Любая база данных на SQL Server содержит хранимые процедуры, даже если разработчик базы не создал ни одной процедуры сам. Процедуры, создаваемые SQL Server, называются *системными*. Некоторые системные процедуры могут быть полезными и разработчику баз данных, например, процедуры для просмотра списка объектов базы данных, пользователей базы данных и др. На рис. 43 отображен результат выполнения системной хранимой процедуры **sp_helpdb** — таблица со сведениями о базах данных SQL Server Local. На рис. 43 можно видеть способ вызова процедуры.

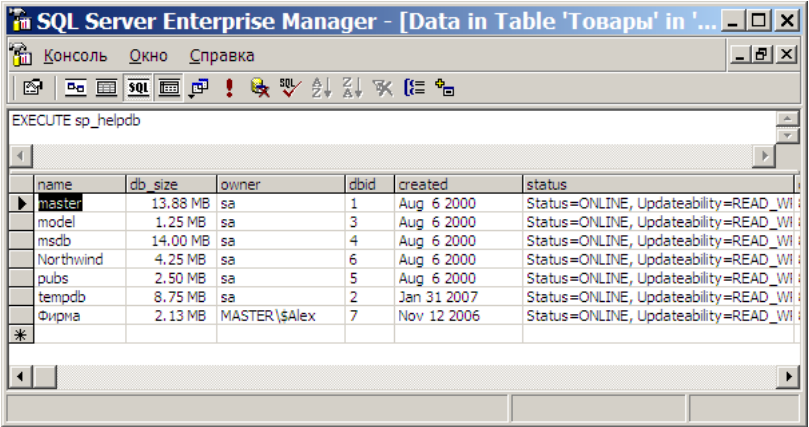


Рис. 43. *Результат выполнения системной хранимой процедуры
sp_helpdb*

Разработчики баз данных также могут создавать хранимые процедуры, которые подобно другим объектам хранятся вместе с базой данных. В каждой базе данных пользователя находятся системные процедуры с именами, начинающимися со строки "dt_".

4.1. Создание хранимой процедуры

Для создания хранимой процедуры можно использовать инструкцию **CREATE PROCEDURE** со следующим синтаксисом:

```
CREATE PROC[EDURE] procedure_name [ ; number ]
[ { @parameter data_type }
  [ VARYING ] [= default ] [ OUTPUT ]
] [ ,...n ]
AS sql_statement [...n ]
```

Параметры процедуры:

- *procedure_name* — имя процедуры, которое должно удовлетворять правилам именования идентификаторов переменных, функций и процедур, в частности, не может превышать 128 символов;
- *number* — необязательное целое число для группировки нескольких процедур под одним именем;

- `{ @parameter data_type }` — список имен параметров процедуры с соответствующим типом данных; хранимая процедура может иметь до 2100 параметров (для SQL Server 2000);
- **VARYING** — ключевое слово, указывающее, что в качестве выходного параметра используется результирующий набор;
- *default* — значение по умолчанию для возможного пропуска параметра при вызове процедуры;
- **OUTPUT** — ключевое слово, определяющее выходной параметр процедуры;
- **AS** — ключевое слово, отмечающее начало текста процедуры;
- *sql_statement [...n]* — набор SQL-инструкций, кроме **CREATE DEFAULT**, **CREATE PROCEDURE**, **CREATE RULE**, **CREATE TRIGGER**, **CREATE VIEW**.

В качестве простого примера создадим процедуру, которая должна выдавать инвентаризационную ведомость склада. Причем код склада задается как входной параметр. Откройте список процедур при помощи «дерева» базы данных **Фирма** и выберите **Action | New Stored Procedure** (рис. 44).

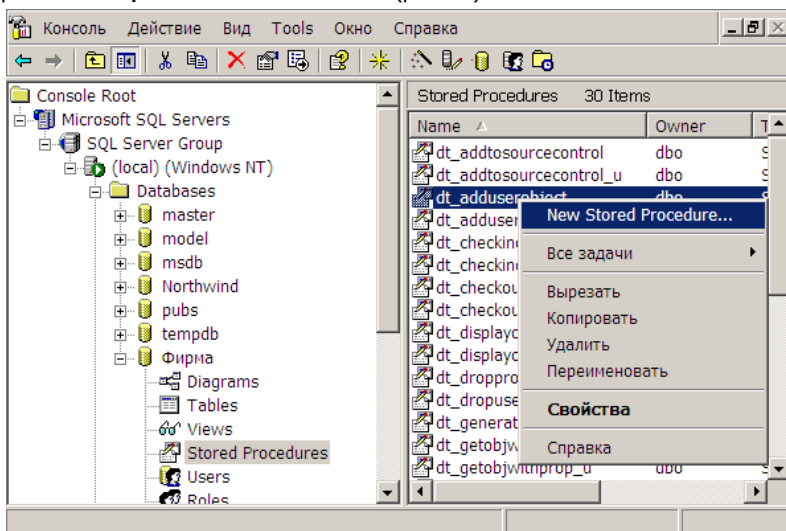


Рис. 44. Откройте список процедур при помощи «дерева» базы данных **Фирма**

Неожиданно для вас Enterprise Manager, не выдавая никаких «поздравлений», тут же предоставит окно для ввода текста хранимой процедуры с шаблоном (рис. 45). Введите в окне **Text** код процедуры, как на рис. 46.

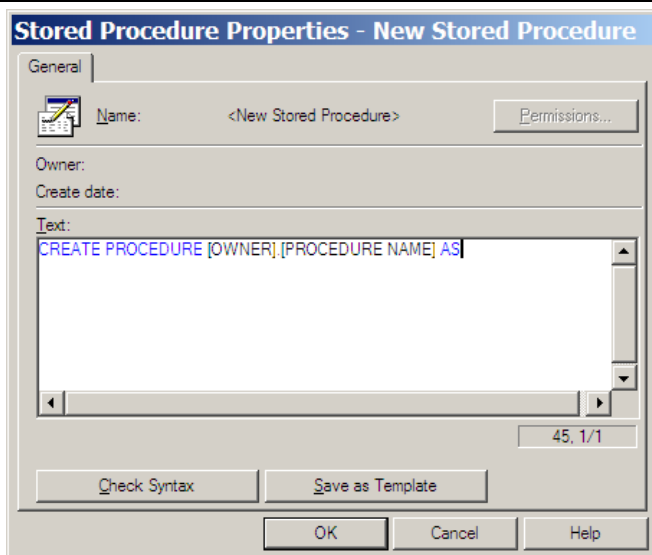


Рис. 45. Окно для ввода текста хранимой процедуры с шаблоном

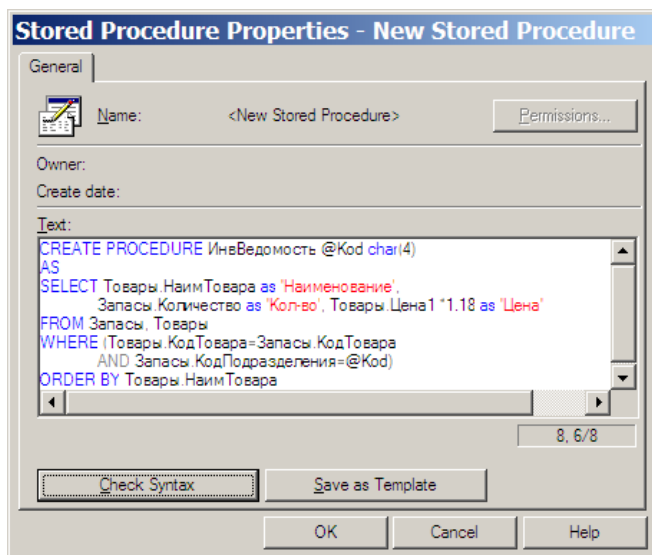


Рис. 46. Окно **Text** для ввода кода процедуры

После проверки синтаксиса (кнопкой **Check Syntax**) щелкните кнопку **OK**. Enterprise Manager поместит только что созданную процедуру в список хранимых процедур базы данных **Фирма** (рис. 47).

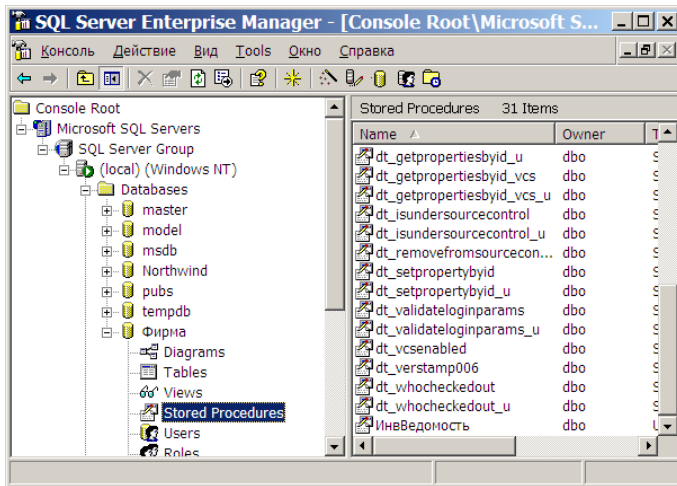


Рис. 47. Enterprise Manager поместит только что созданную процедуру в список хранимых процедур базы данных **Фирма**

На рис. 48 приведен пример использования хранимой процедуры **ИнвВедомость**. Обратите внимание на использование символа @. Он применяется как часть имени параметра, а не только как признак параметра.

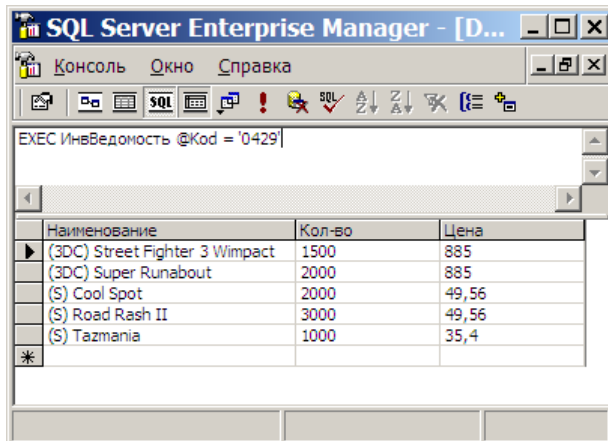


Рис. 48. Пример использования хранимой процедуры **ИнвВедомость**

5. ИМПОРТ ДАННЫХ В EXCEL

Рассмотрим импорт внешних данных в Excel из базы SQL Server для возможной дальнейшей их обработки. Выберите **Импорт внешних данных | Импортировать данные** меню **Данные** и в появившемся диалоговом окне (рис. 49) выделите **+Новое подключение к SQL-серверу.odc** и щелкните кнопку **Создать**.

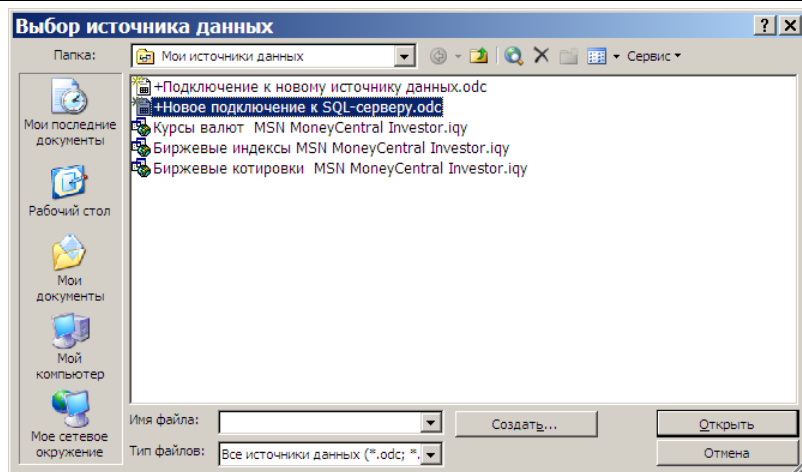


Рис. 49. Выделите **+Новое подключение к SQL-серверу.odc**

В следующем окне (рис. 50) укажите имя SQL-сервера и щелкните кнопку **Далее**.

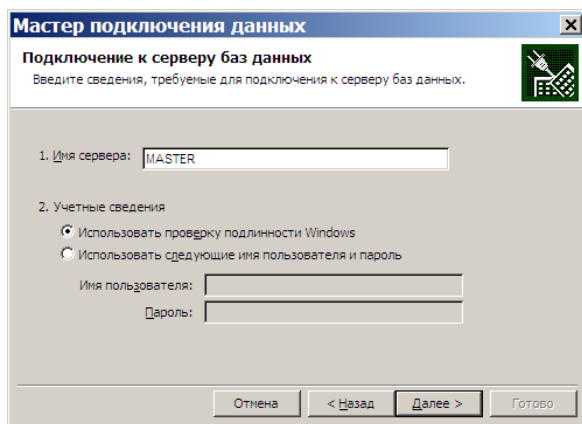


Рис. 50. Укажите имя SQL-сервера и щелкните кнопку **Далее**

Если указанный сервер действительно существует и нет проблем с доступом, то в следующем окне можно уже будет указать имя базы данных (рис. 51).

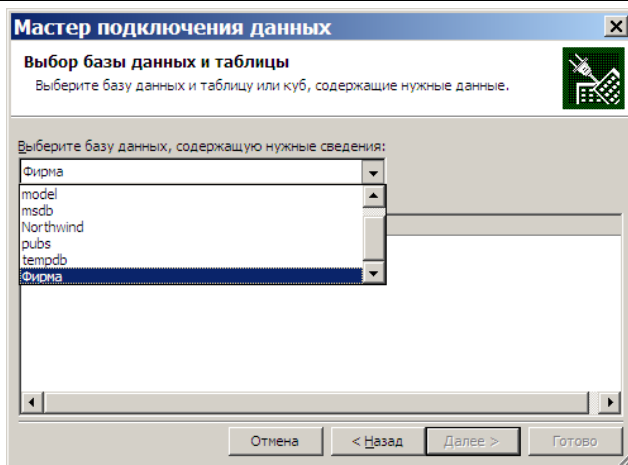


Рис. 51. В этом можно уже указать имя базы данных

Как только имя базы указано, следует выбрать необходимую таблицу (рис. 52).

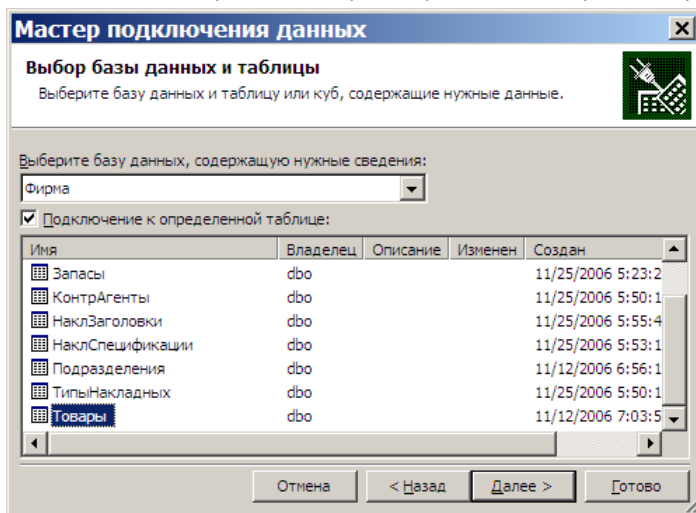


Рис. 52. Как только имя базы указано, следует выбрать необходимую таблицу

В окне, подобном представленному на рис. 53, щелкните кнопку **Готово** или же предварительно введите описание создаваемого запроса.

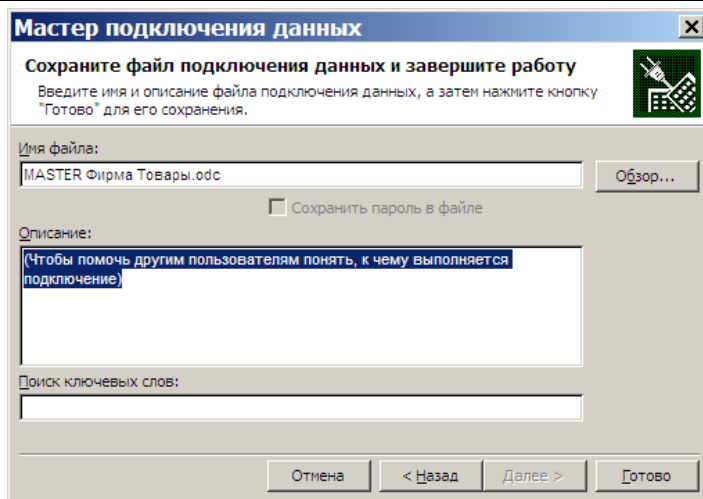


Рис. 53. Щелкните кнопку **Готово**

Остается сделать совсем немного — указать место, куда следует поместить таблицу с данными (рис. 54). Для тестирования созданного запроса можно просто щелкнуть кнопку **ОК**.

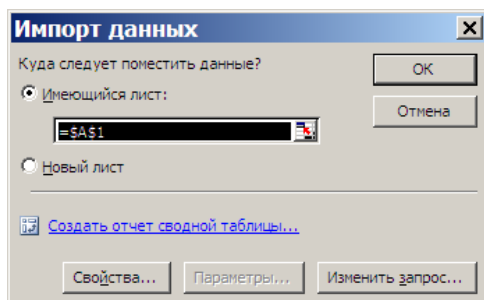


Рис. 54. Здесь следует указать, куда нужно поместить результаты запроса

Цель нашего примера — на рис. 55. На этот раз в запросе была указана таблица, но вы всегда можете изменить запрос, используя диалоговое окно, вызываемое при помощи кнопки **Изменить запрос** в окне **Импорт внешних данных** (рис. 54).

	A	B	C	D	E	F	G
	КодТовара	НаимТовара	Цена1	Цена2			
2	006019165803	(S) Road Rash II	42	49			
3	006019170049	(S) Tazmania	30	35			
4	010414114438	(VCD) Патриот	138	161			
5	010415090732	(VCD) Пастырь	111	130			
6	010415090937	(VCD) Шоссе в никуда	120	140			
7	010428144348	(VCD) Доберман	135	158			
8	028019095823	(VCD) Diber Marouani	90	105			
9	028019100018	(VCD) Wham! The best of	105	123			
10	028019100222	(VCD) Голая мишень	90	105			
11	036019164741	(S) Cool Spot	42	49			
12	174500175359	(3DC) Street Fighter 3 Wimpact	750	875			
13	174500175521	(3DC) Star Gladiator 2	750	875			
14	186100175232	(3DC) Super Runabout	750	875			
15							
16							
17							

Рис. 55. Результат импортирования содержимого таблицы с SQL-сервера

В диалоговом окне **Изменение запроса OLE DB** (рис. 56) в текстовом окне **Текст** команды можно указать любую другую таблицу.

Изменение запроса OLE DB

Подключение:
 Provider=SQLOLEDB.1;Integrated Security=SSPI;Persist Security Info=True;Initial Catalog=Фирма;Data Source=MASTER;Use Procedure for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=MASTER;Use Encryption for Data=False;Tag with column collation when possible=False

Тип команды:
 Таблица

Текст команды:
 "Фирма", "dbo", "Товары"

OK Отмена

Рис. 56. В текстовом окне **Текст команды** можно указать любую другую таблицу

Если необходимо указать не таблицу, а SQL-инструкцию, нужно в списке **Тип команды** выбрать **SQL** (рис. 57) и в текстовом окне **Текст команды** написать эту

инструкцию (рис. 58). В SQL-инструкции можно вместо таблицы использовать имена представлений. Можно также вызывать хранимые процедуры.

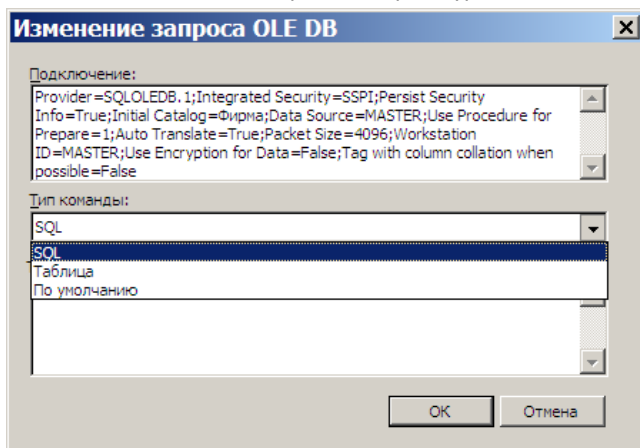


Рис. 57. В списке *Тип команды* выбрать *SQL*

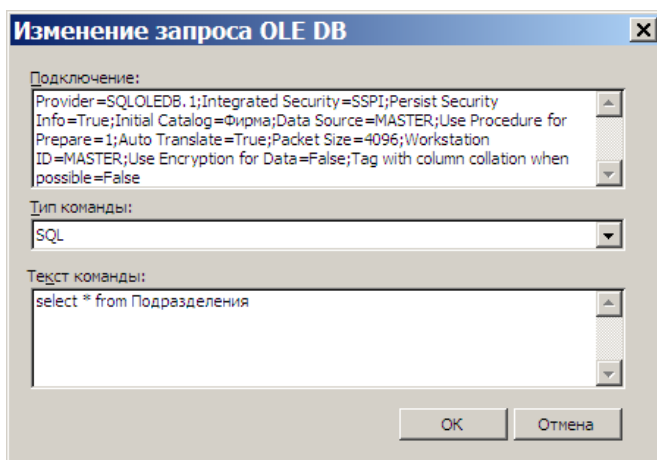


Рис. 58. В текстовом окне *Текст команды* можно написать SQL-инструкцию

Задания:

Продемонстрируйте Вашу работу преподавателю!

КОНТРОЛЬНЫЕ ВОПРОСЫ:

1. Для каких целей в SQL Server Enterprise Manager используют средство Query Designer?
2. Расскажите об основных особенностях языка Transact – SQL.

3. Для каких целей в SQL Server используют SQL Pane, Results Pane, Diagram Pane, Grid Pane?
4. Расскажите о способах создания новых представлений (Views) в SQL Server. Какие существуют особенности работы с созданными представлениями?
5. Как создать хранимую процедуру в SQL Server?
6. Какие из хранимых процедур называются системными?
7. Как осуществить импорт данных в MS Excel из базы данных SQL Server?