

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Шебзухова Татьяна Александровна
Должность: Директор Пятигорского института (филиал) Северо-Кавказского
федерального университета
Дата подписания: 19.07.2023 11:16:06
Уникальный программный ключ:
d74ce93cd40e39275c3ba2f58486412a1c8ef96f

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Пятигорский институт (филиал) СКФУ

Методические указания
по выполнению лабораторных работ
по дисциплине
«ПРОЕКТНЫЙ ПРАКТИКУМ»
для направления подготовки **09.03.02 Информационные системы и
технологии**
направленность (профиль) **Информационные системы и технологии
обработки цифрового контента**

**Пятигорск
2023**

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

ВВЕДЕНИЕ

ЦЕЛЬ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целью изучения дисциплины является формирование набора общенаучных и профессиональных компетенций будущего бакалавра по направлению 09.03.02 «Информационные системы и технологии».

Задачи освоения дисциплины:

- приобретение практических навыков комплексного использования методов, инструментальных средств проектирования и сопровождения информационных систем; навыков управления ИТ-проектами;
- освоение методик проектирования обеспечивающих подсистем ИС и расчета экономической эффективности ИТ-проекта.

2.Наименование лабораторных работ

№ Темы дисциплины	Наименование тем дисциплины, их краткое содержание	Объем часов	Из них практическая подготовка, часов
8 семестр			
1	Лабораторная работа 1. Проектирование ПО с применением UML.	0,75	
2	Лабораторная работа 2. Построение моделей в BPWin. Построение IDEF0 диаграмм.	0,75	
2	Лабораторная работа 3. Построение моделей в BPWin. Построение IDEF3, DFD диаграмм.	0,75	
2	Лабораторная работа 4. Построение моделей в ERWin	0,75	
3	Лабораторная работа 5. Создание и заполнение таблиц в Microsoft SQL Server 2012.	0,75	
3	Лабораторная работа 6. Создание запросов и фильтров в Microsoft SQL Server 2012.	0,75	
3	Лабораторная работа 7. Создание динамических запросов при помощи хранимых процедур в Microsoft SQL Server 2012.	0,75	
3	Лабораторная работа 8. Создание функций пользователя в Microsoft SQL Server 2012.	0,75	
3	Лабораторная работа 9. Обеспечение целостности данных в Microsoft SQL Server 2012. Создание диаграмм и триггеров.	0,75	
4	Лабораторная работа 10. Создание простых ленточных форм для работы с базами данных в Visual Studio 2012.	0,75	
4	Лабораторная работа 11.	0,75	

Сертификат
Владелец:

Документ подписан
Электронно-цифровой
2C0000043E9AB820572305E7BA4500060000043E
Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

	Создание сложных ленточных форм для работы с базами данных в Visual Studio 2012		
4	Лабораторная работа 12. Создание табличных форм для работы с базами данных в Visual Studio 2012.	0,75	
4	Лабораторная работа 13. Создание отчетов в Visual Studio 2012.	0,75	
3	Лабораторная работа 14. Выполнение индивидуальных заданий по проектированию информационных систем в Microsoft SQL Server 2012.	0,75	
4	Лабораторная работа 15. Выполнение индивидуальных заданий по проектированию информационных систем в Visual Studio 2012. Создание ленточных и табличных форм для работы с базами данных.	0,75	
4	Лабораторная работа 16. Выполнение индивидуальных заданий по проектированию информационных систем в Visual Studio 2012. Создание отчетов и диаграмм.	0,75	
5	Лабораторная работа 17. Подготовка документации IT проекта.	1,5	
6	Лабораторная работа 18. Расчет экономической эффективности проекта.	1,5	
	Итого за 8 семестр	15	
	Итого	15	

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E

Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

СТРУКТУРА И СОДЕРЖАНИЕ ЛАБОРАТОРНЫХ ЗАНЯТИЙ

ЛАБОРАТОРНЫЕ РАБОТЫ

Лабораторная работа № 1

«Проектирование ПО с применением UML.»

Цель работы:

- Ознакомиться с целями и задачами проектирования ПО
- Понять и усвоить методологию моделирования с применением UML
- Научиться создавать диаграммы Use Case
- Научиться создавать диаграмму вариантов использования.
- Научиться создавать диаграммы действий.

Порядок выполнения лабораторной работы.

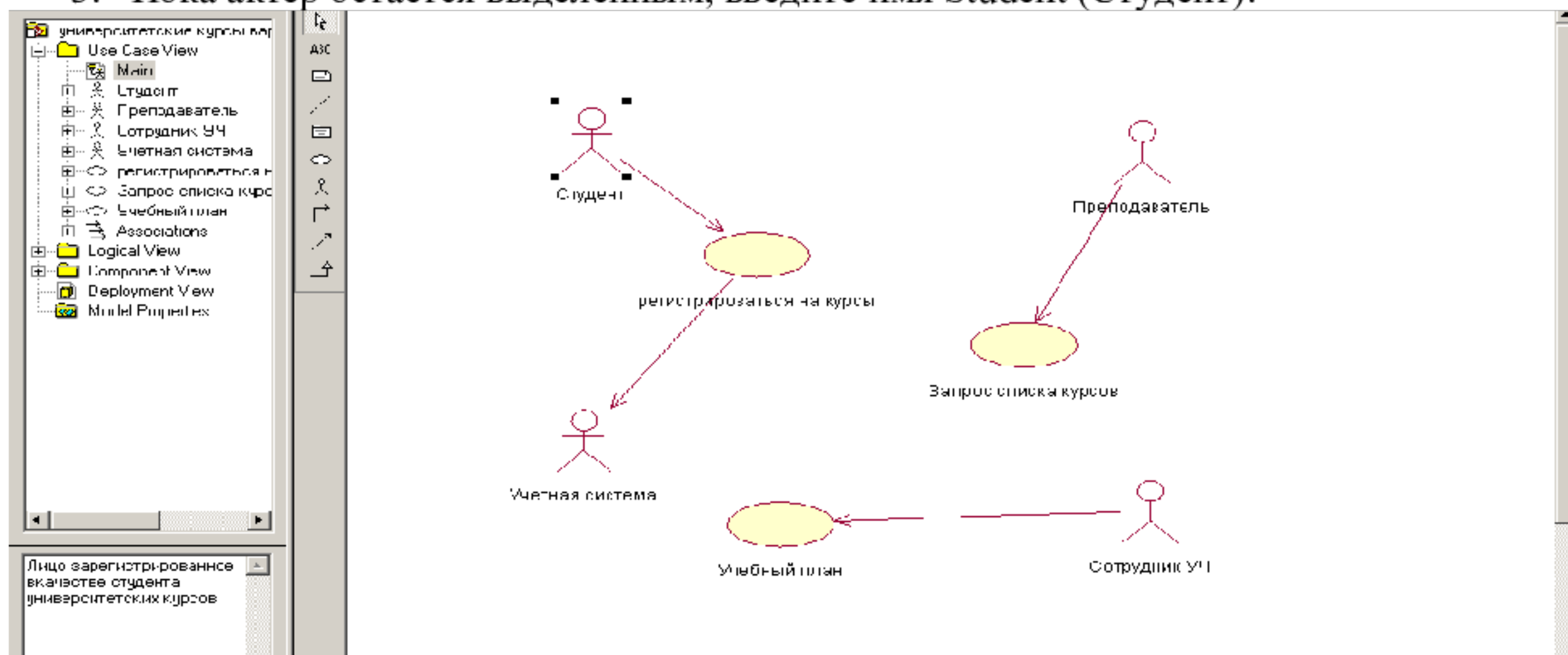
Создание диаграммы Use Case

Моделирование проблемы регистрации курсов начнем с создания диаграммы Use Case. Этот тип диаграммы представляется актерами, элементами Use Case и отношениями между ними. Откроем главную диаграмму Use Case.

1. В окне браузера щелкнем по значку + слева от пакета Use Case View.
2. Для открытия диаграммы выполним двойной щелчок по значку Main.

Первый шаг построения этой диаграммы состоит в определении актеров, фиксирующих роли внешних объектов, взаимодействующих с системой. В нашей проблемной области можно выделить 4 актера - Student (Студент), Professor (Преподаватель), Registrar (Сотрудник УЧ) и Billing System (Учетная система).

1. На панели инструментов щелкните по значку актера.
2. Для добавления актера в диаграмму щелкните в нужном месте диаграммы.
3. Пока актер остается выделенным, введите имя Student (Студент).



4. Повторите предыдущие шаги для ввода трех других актеров (Professor, Registrar и Billing System - Преподаватель, Сотрудник УЧ, Учетная система).

Далее для каждого актера нужно определить соответствующие элементы Use Case. Элемент Use Case представляет определенную часть функциональности, обеспечиваемой системой. Вы можете идентифицировать элементы Use Case путем рассмотрения каждого актера и его взаимодействия с системой. В нашей модели актер Student хочет регистрироваться на курсы (Register for Courses). Актер Billing System получает

ДОКУМЕНТ ПОДПИСАН
Электронная подпись
Сертификат: 2C0009043E59AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

информацию о регистрации. Актер Professor хочет запросить список курса (Request a Course Roster). Наконец, актер Registrar должен управлять учебным планом (Manage Curriculum).

1. На панели инструментов щелкните по значку элемента Use Case.
2. Для добавления элемента Use Case в диаграмму щелкните в нужном месте диаграммы.
3. Пока элемент Use Case остается выделенным, введите имя Register for Courses.
4. Повторите предыдущие шаги для ввода других элементов Use Case (Request Course Roster, Manage Curriculum).

Далее между актерами и элементами Use Case рисуются отношения. Чтобы показать направление взаимодействия (кто инициирует взаимодействие), используются однонаправленные стрелки (uni-directional arrows). В системе регистрации курсов актер Student инициирует элемент Use Case Register for Courses, который, в свою очередь, взаимодействует с актером Billing System. Актер Professor инициирует элемент Use Case Request Course Roster. Актер Registrar инициирует элемент Use Case Manage Curriculum.

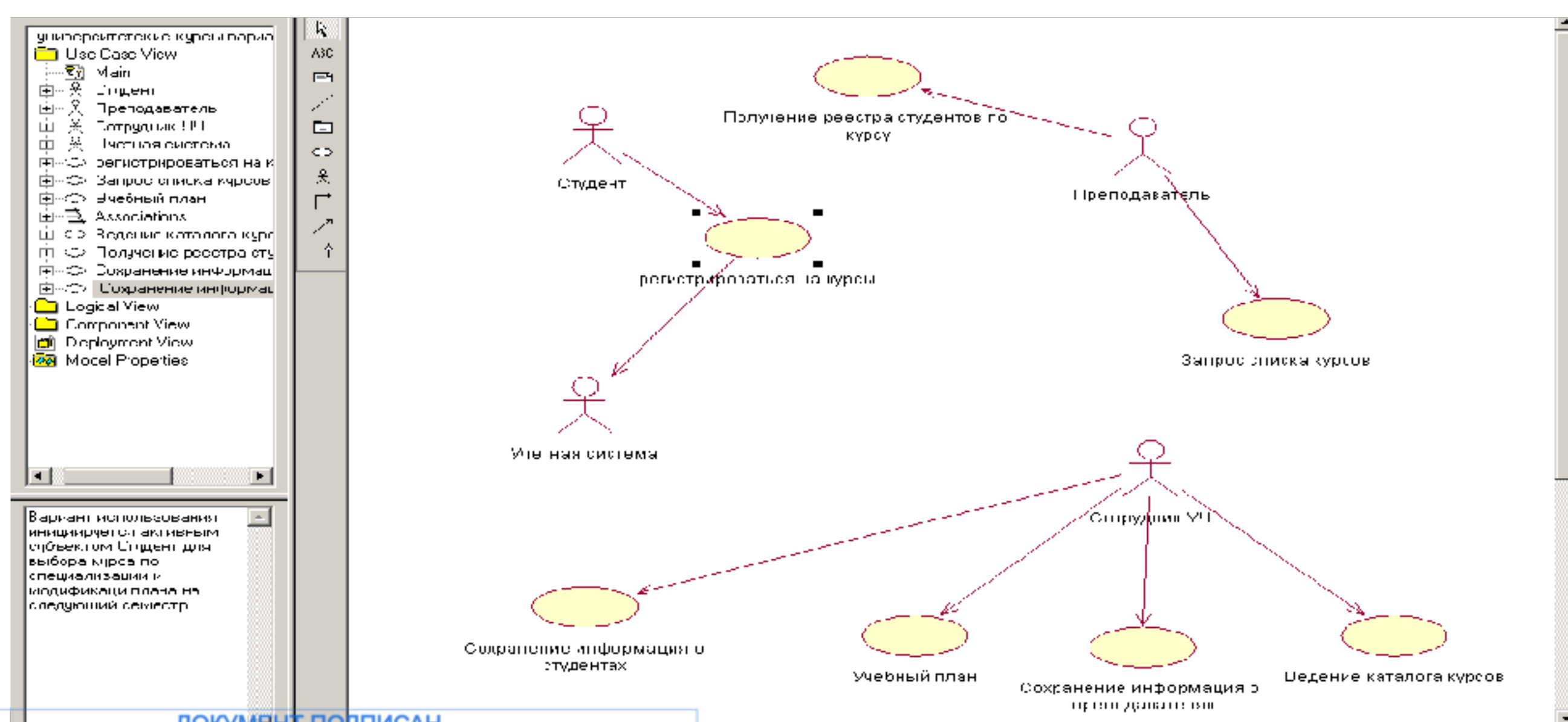
1. На панели инструментов щелкните по значку однонаправленной ассоциации (стрелке).
2. Щелкните по актеру Student и перетащите линию на элемент Use Case Register for Courses.
3. На панели инструментов щелкните по значку однонаправленной ассоциации (стрелке).
4. Щелкните по элементу Use Case Register for Courses и перетащите линию на актера Billing System.
5. Повторите предыдущие шаги для ввода других отношений (от актера Professor к элементу Use Case Request Course Roster и от актера Registrar к элементу Use Case Manage Curriculum).

В модель следует включить краткое описание каждого активного субъекта, которое должно определять роль в процессе использования системы.

Окно документирования можно открыть в меню View – Documentation.

Заполните документирование для наших актеров.

Варианты использования позволяют моделировать диалог между активным субъектом и системой и отображают функции последней, предоставляемые в распоряжение субъекта.



ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебажова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

Создадим потоки событий для варианта использования, которые документируются при помощи следующего шаблона, который создается в виде текстового файла:

- 1.0 Краткое описание
- 1.1 Краткое описание.
- 2.0 Потоки событий
- 2.1 Основной поток
- 2.2 Альтернативные потоки
- 2.2.1 Альтернативные поток
- 3.0 Специальные требования
- 3.1 Специальные требования
- 4.0 Предусловия
- 4.1 Предусловие
- 5.0 Постусловие
- 5.1 Постусловие
- 6.0 Дополнительные замечания
- 6.1 Дополнительные замечания

Теперь свяжем файл спецификации потока событий с вариантом использования, для этого расположим курсор мыши в окне Browser вызовем контекстное меню и выбрать элемент меню Open specification и перейти на вкладку Files диалогового окна Use Case specification, расположите курсор в пределах области окна и вновь активизируйте контекстное меню и выбрав элемент меню Insert file выберите необходимый файл после чего выберите Открыть и нажмите ОК.

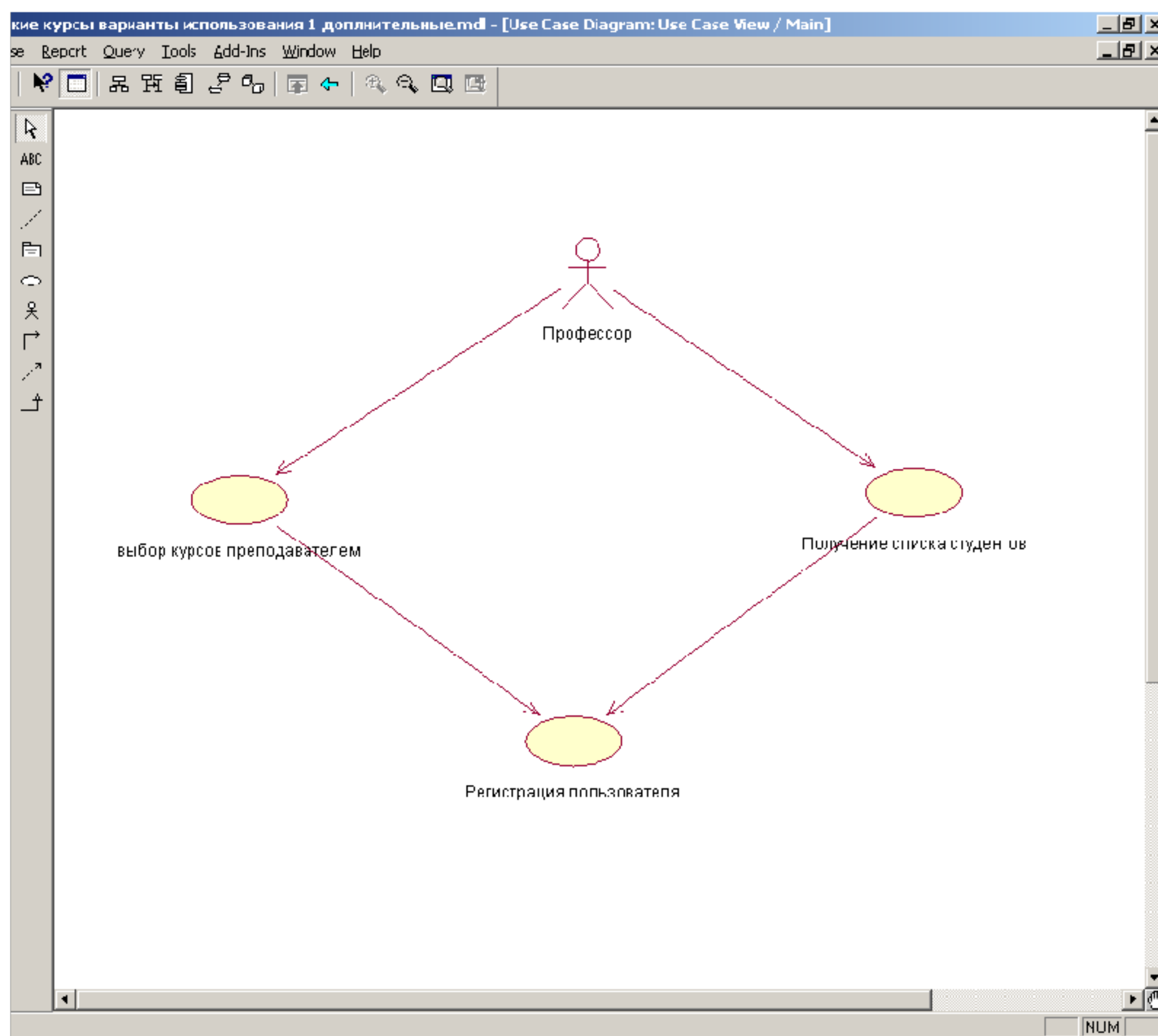
Создадим дополнительную диаграмму вариантов использования.

Расположим курсор мыши над элементом Use Case View окна Browser и щелкнуть правой кнопкой чтобы активировать контекстное меню.

Выбрать элемент New - Use Case Diagram и дерево пополнится элементом NewDiagram соответствующим новой диаграмме вариантов использования.

Изменить элемент NewDiagram

Двойным щелчком на элементе открыть окно дополнительной диаграммы вариантов использования и включить в нее требуемые активные субъекты, варианты использования и связи.



Самостоятельно.

Создать две дополнительные диаграммы для студента и Сотрудника УЧ.

Диаграммы действий воспроизводят поток функций управления, показывают, какие ветви процесса могут выполняться параллельно и определять альтернативные пути достижения целей. Диаграммы действий конструируются на начальных фазах жизненного цикла системы, представляют потоки которые охватывают несколько вариантов использования или протекают на уровне определенного варианта.

Расположить курсор мыши над элементом Use Case View и щелкнуть правой кнопкой чтобы активизировать контекстное меню.

Выбрать элемент меню New – Activity Diagram, выбрать новый элемент и изменить его название и двойным щелчком открыть окно диаграммы действий.

Действия описывают некоторый фрагмент поведения системы в контексте потока функций управления.

Элемент перехода применяются в целях обозначения направления передачи управления от одного действия к другому.

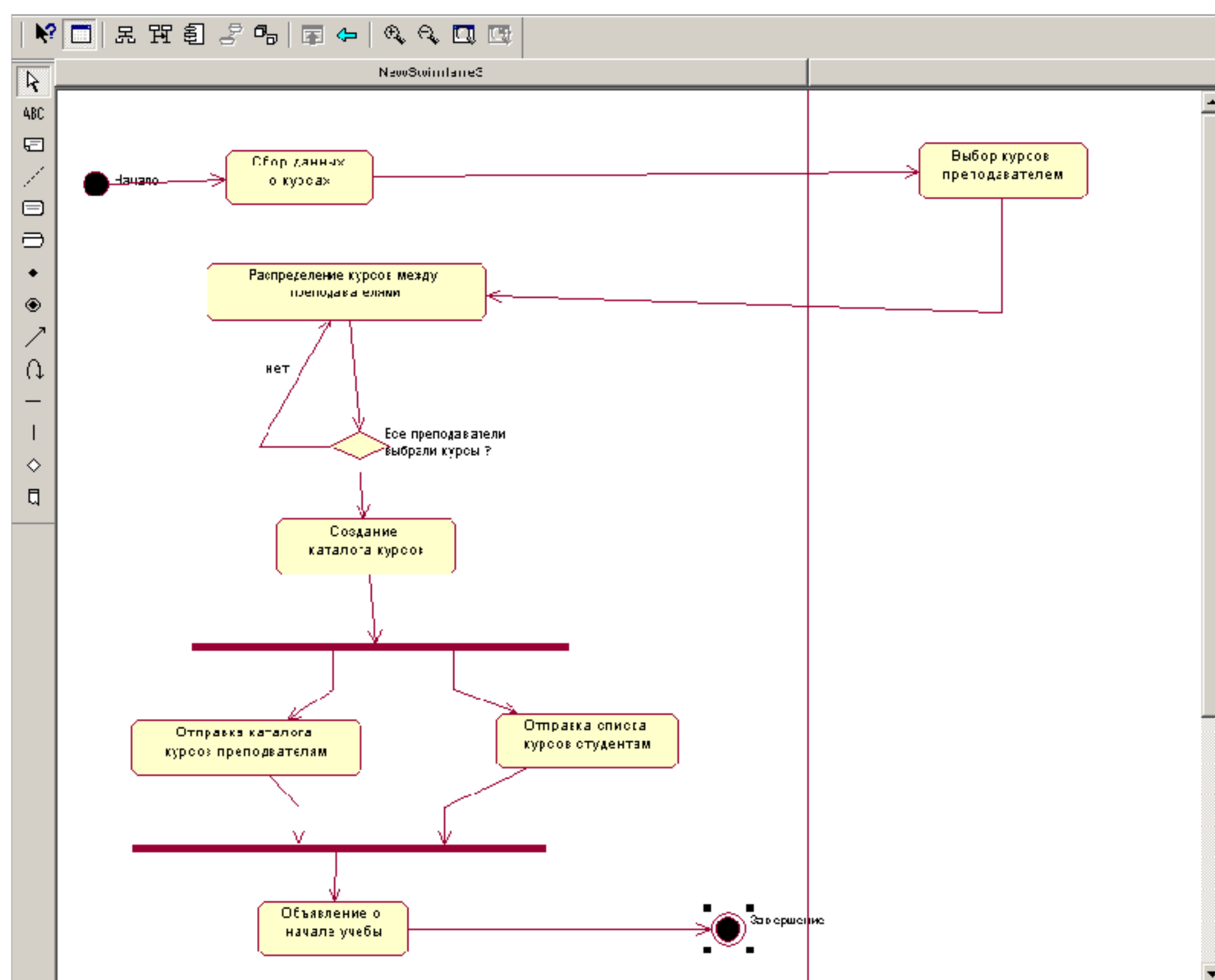
Точки принятия решения - в таких точках поток обычно претерпевает ветвление в зависимости от принимаемых системой или пользователем решений.

Полосы синхронизации позволяют указать допускают одновременное выполнение или подлежат логическому выполнению.

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023



Зоны – диаграмма может быть разделена на зоны каждая из которых связана с определенным активным субъектом.

Исходное и завершающее действия – обычно поток управления содержит одно исходное и несколько завершающих действий.

Самостоятельно написать диаграмму действий для учебной части.

Контрольные вопросы

При защите лабораторной работы студент должен ориентироваться в проделанной работе, знать:

- Цели и задачи проектирования ПО.
- Методологию функционального моделирования с применением UML.
- Как создавать диаграммы Use Case.
- Как создавать диаграмму вариантов использования.
- Как создавать диаграммы действий.

Студент должен уметь изменить модель (добавить новые или удалить элементы) по требованию преподавателя.

Лабораторная работа № 2

«Построение моделей в BPWin. Построение IDEF0 диаграмм.»

Цель работы:

- Ознакомиться с целями и задачами предпроектного обследования
- Понять и усвоить методологию функционального моделирования
- Научиться определять границы моделирования
- Научиться определять точку зрения при построении модели.
- Изучить принципы композиции.

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Порядок выполнения лабораторной работы.

Действителен: с 19.08.2022 по 19.08.2023

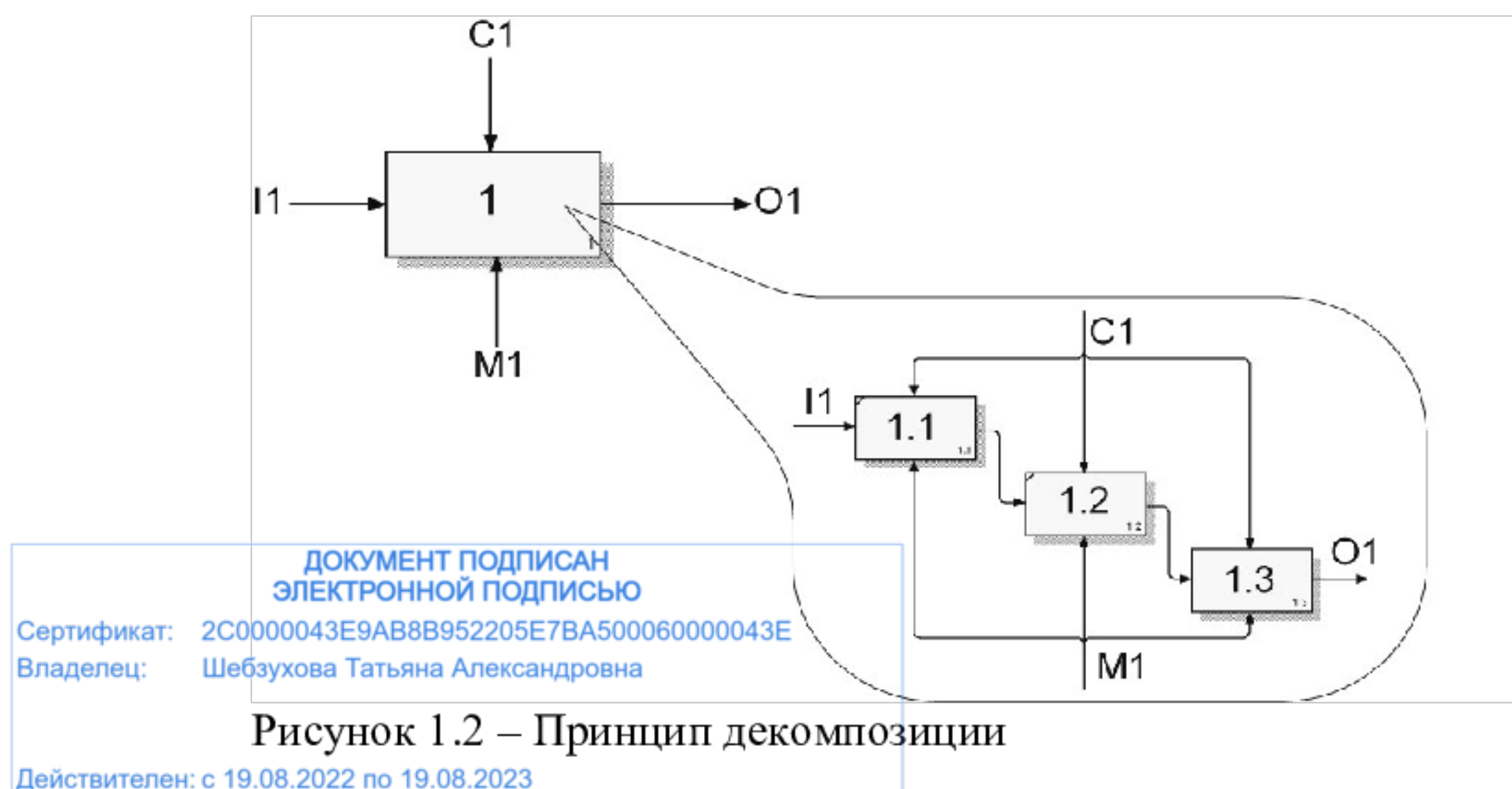
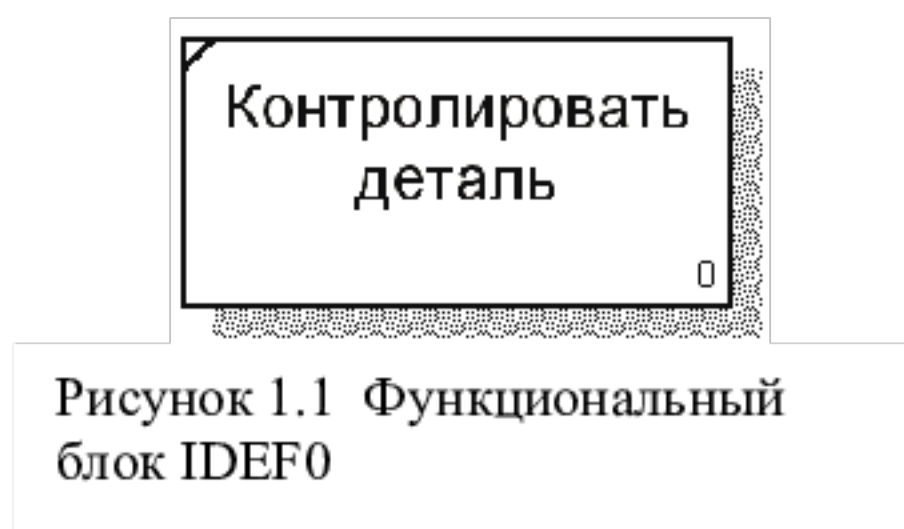
Первый шаг при построении модели IDEF0 заключается в определении назначения модели – набора вопросов, на которые должна отвечать модель. **Границы моделирования** предназначены для обозначения ширины охвата предметной области и глубины детализации и являются логическим продолжением уже определенного назначения модели. Следующим шагом указывается предполагаемая целевая аудитория, для нужд которой создается модель. Зачастую от выбора целевой аудитории зависит уровень детализации, с которым должна создаваться модель. Перед построением модели необходимо иметь представление о том, какие сведения о предмете моделирования уже известны, какие дополнительные материалы и (или) техническая документация для понимания модели могут быть необходимы целевой аудитории, какие язык и стиль изложения являются наиболее подходящими.

Под **точкой зрения** понимается перспектива, с которой наблюдалась система при построении модели. Точка зрения выбирается таким образом, чтобы учесть уже обозначенные границы моделирования и назначение модели. Однажды выбранная точка зрения остается неизменной для всех элементов модели. При необходимости могут быть созданы другие модели, отображающие систему с других точек зрения. Вот несколько примеров точек зрения при построении моделей: клиент, поставщик, владелец, редактор.

Действие, обычно в IDEF0 называемое функцией, обрабатывает или переводит входные параметры (сырье, информацию и т.п.) в выходные. Поскольку модели IDEF0 представляют систему иерархических функций, быть определена система в целом – Функции изображаются поименованные функциональные блоки. подбираются по именами действий – с как множество в первую очередь должна функция, описывающая **контекстная функция**. на диаграммах как прямоугольники, или Имена функций в IDEF0 сходным правилам с использованием глаголов или отглагольных существительных. Важно подбирать имена таким образом, чтобы они отражали систему так, как если бы она обзревалась с точки зрения, выбранной для моделирования. Пример функционального блока приведен на рисунке 1.

Каждый блок, не имеющей декомпозиции, помечается небольшой диагональной чертой, расположенной в левом верхнем углу блока.

Любой блок может быть декомпозирован на составляющие его блоки. Функция декомпозиции позволяет разбить сложные процессы на составляющие его операции. При этом уровень детализации процесса определяется непосредственно разработчиком модели (рисунок 2).



Декомпозиция позволяет постепенно и структурировано представлять модель системы в виде иерархической структуры отдельных диаграмм, что делает ее менее перегруженной и легко читаемой.

1.2.3 Границы и связи

Чтобы быть полезным, описание любого блока должно, как минимум, включать в себя описание объектов, которые блок создает в результате своей работы («выхода»), и объектов, которые блок потребляет или преобразует («вход»).

В IDEF0 также моделируются управление и механизмы исполнения. Под управлением понимаются объекты, воздействующие на способ, которым блок преобразует вход в выход. Механизм исполнения – объекты, которые непосредственно выполняют преобразование входа в выход, но не потребляются при этом сами по себе.

Для отображения категорий информации, присутствующих на диаграммах IDEF0, существует аббревиатура ICOM, отображающая четыре возможных типа стрелок:

I (Input) – вход – нечто, что потребляется в ходе выполнения процесса;

C (Control) – управление – ограничения и инструкции, влияющие на ход выполнения процесса;

O (Output) – выход – нечто, являющееся результатом выполнения процесса;

M (Mechanism) – исполняющий механизм – нечто, что используется для выполнения процесса, но не потребляется само по себе (рисунок 3).

Рисунок 3 показывает четыре возможных типа стрелок в IDEF0, каждый из типов соединяется со своей стороной функционального блока.

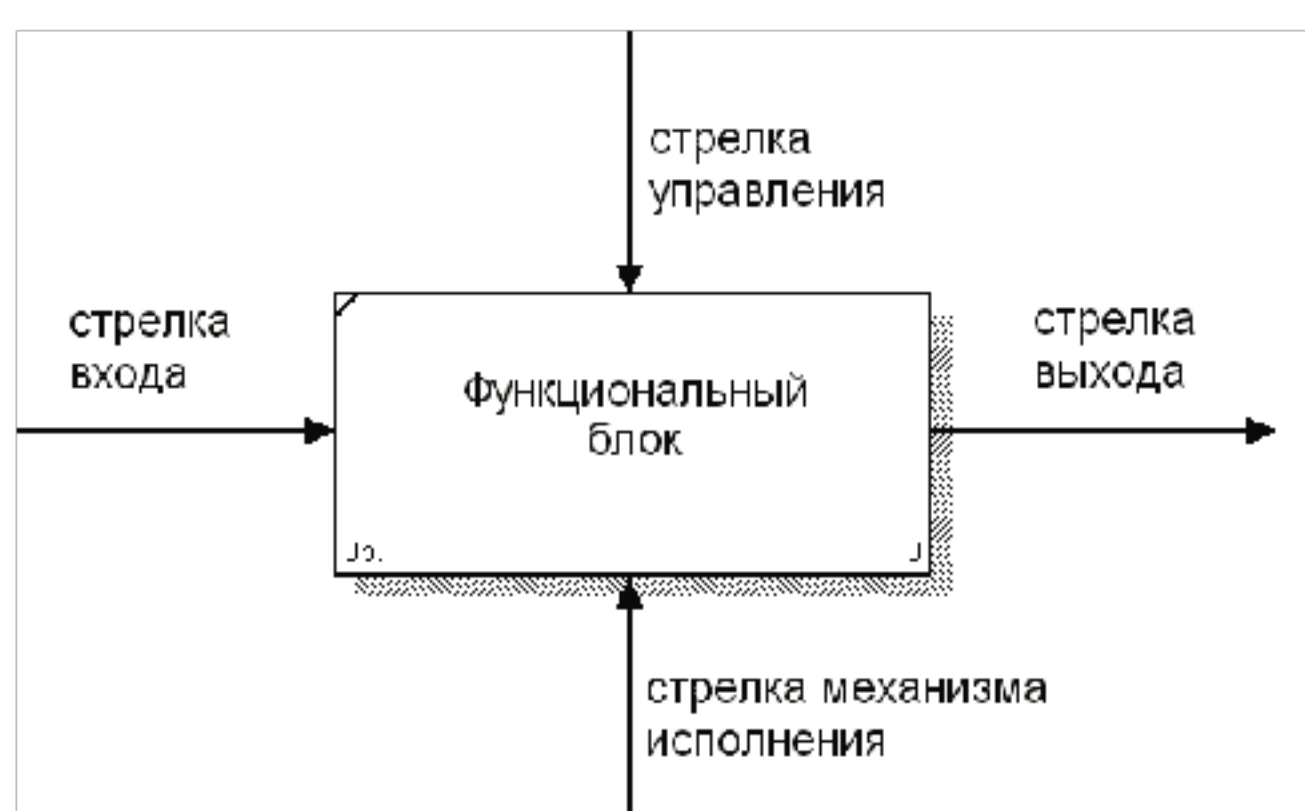


Рисунок 1.3 – Типы стрелок функционального блока

Для названия стрелок, как правило, употребляются имена существительные. Стрелки могут представлять собой людей, места, вещи, идеи или события. Как и в случае с функциональными блоками, присвоение имен всем стрелкам на диаграмме является только необходимым условием для понимания читателем сути изображенного. Отдельное описание каждой стрелки в текстовом виде может оказаться критическим фактором для построения точной и полезной модели.

Стрелки входа. Вход представляет собой сырье, или информацию, потребляемую или преобразуемую функциональным блоком для производства выхода. Стрелки входа всегда направлены в левую сторону прямоугольника, обозначающего в IDEF0 функциональный блок. Наличие входных стрелок на диаграмме не является обязательным, так как возможно, что некоторые блоки ничего не преобразуют и не изменяют. Примером блока, не имеющего входа, может служить «принятие решения руководством», где для принятия решения анализируется несколько факторов, но ни один из них непосредственно не преобразуется и не потребляется в результате принятия какого-либо решения.

Стрелки управления. Стрелки управления отвечают за регулирование того, как и когда выполняется функциональный блок, и, если он выполняется, какой выход получается в результате его выполнения. Так как управление контролирует поведение функционального блока для обеспечения создания желаемого выхода, каждый функциональный блок должен иметь, как минимум, одну стрелку управления. Стрелки управления всегда входят в функциональный блок сверху.

Управление часто существует в виде правил, инструкций, законов, политики, набора необходимых процедур или стандартов. Влияя на работу блока, оно непосредственно не потребляется и не трансформируется в результате. Может оказаться, что целью функционального блока является как раз изменение того или иного правила, инструкции, стандарта и т.п. В этом случае стрелка, содержащая соответствующую информацию, должна рассматриваться не как управление, а как вход функционального блока.

Управление можно рассматривать как специфический вид входа. В случаях, когда неясно, относить ли стрелку к входу или к управлению, предпочтительно относить ее к управлению до момента, пока неясность не будет разрешена.

Стрелки выхода. Выход – это продукция или информация, получаемая в результате работы функционального блока. Каждый блок должен иметь, как минимум, один выход. Действие, которое не производит никакого четко определяемого выхода, не должно моделироваться вообще (по меньшей мере, должно рассматриваться в качестве одного из первых кандидатов на исключение из модели).

При моделировании непроеизводственных предметных областей выходами, как правило, являются данные, в каком-либо виде обрабатываемые функциональным блоком. В этом случае важно, чтобы названия стрелок входа и выхода были достаточно различимы по своему смыслу. Например, блок «Прием пациентов» может иметь стрелку «Данные о пациенте» как на входе, так и на выходе. В такой ситуации входящую стрелку можно назвать «Предварительные данные о пациенте», а исходящую – «Подтвержденные данные о пациенте».

Стрелки механизма исполнения. Механизмы являются ресурсом, который непосредственно исполняет моделируемое действие. С помощью механизмов исполнения могут моделироваться: ключевой персонал, техника и (или) оборудование. Стрелки механизма исполнения могут отсутствовать в случае, если оказывается, что они не являются необходимыми для достижения поставленной цели моделирования.

Комбинированные стрелки. В IDEF0 существует пять основных видов комбинированных стрелок: выход – вход, выход – управление, выход – механизм исполнения, выход – обратная связь на управление и выход – обратная связь на вход.

Стрелка *выход – вход* применяется, когда один из блоков должен полностью завершить работу перед началом работы другого блока (рисунок 4).

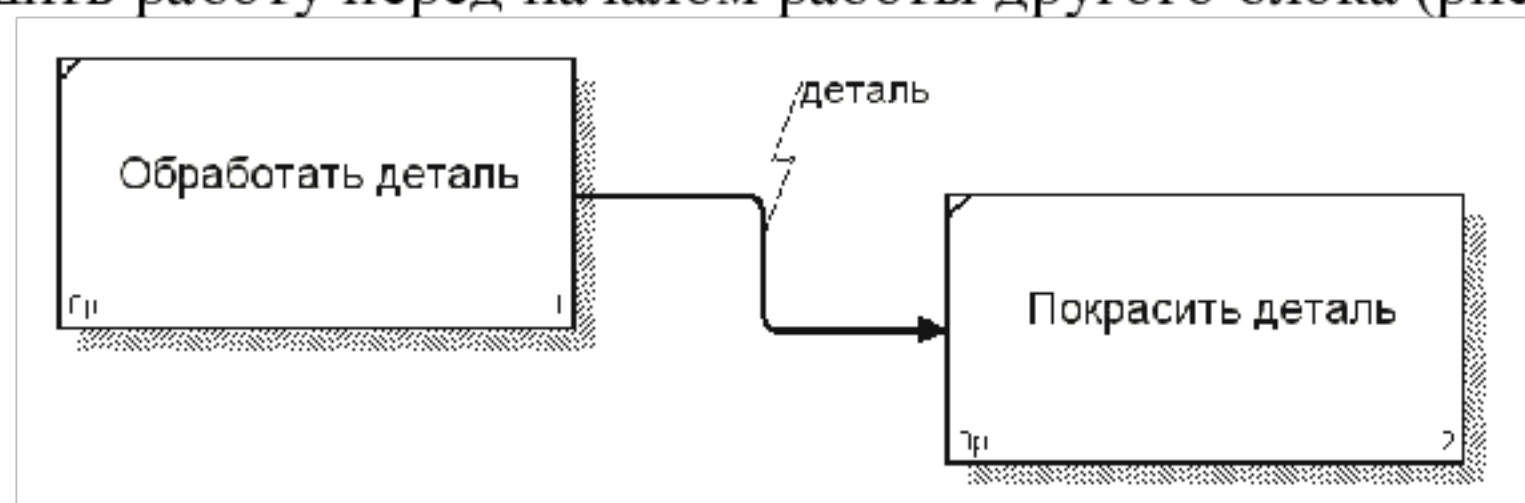


Рисунок 1.4 – Комбинированная стрелка *выход – вход*

Стрелка *выход – управление* отражает ситуацию преобладания одного блока над другим, когда один блок управляет работой другого. На рисунке 5 принципы формирования инвестиционного портфеля управляют поведением брокеров на бирже.

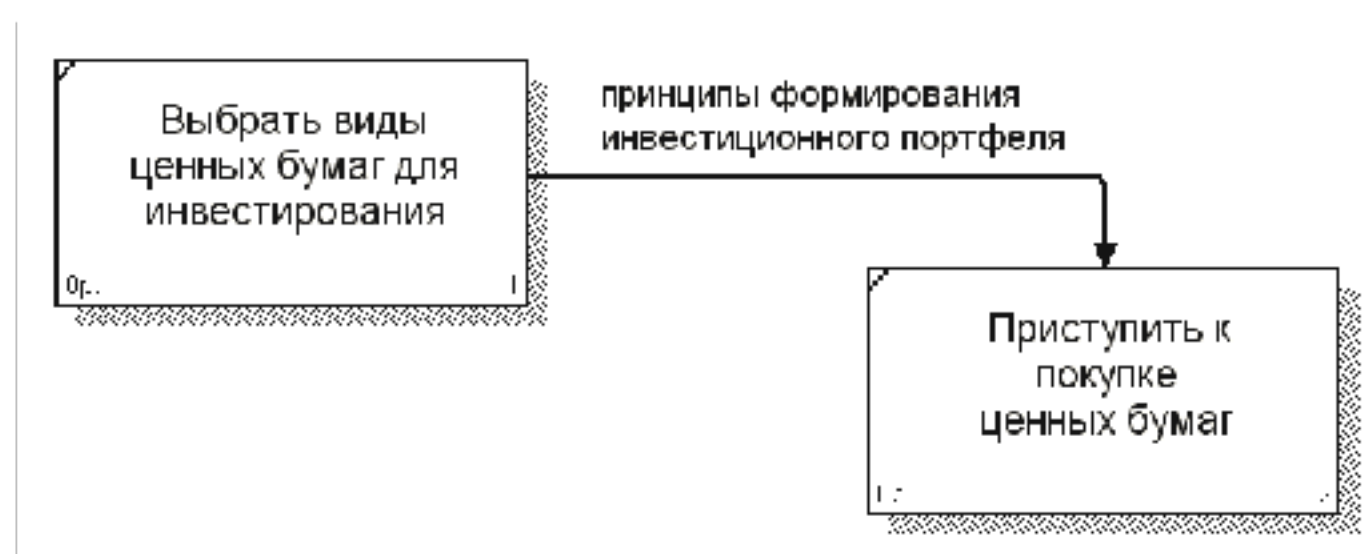


Рисунок 1.5 – Комбинированная стрелка *выход – управление*

Стрелки *выход - механизм исполнения* встречаются реже и отражают ситуацию, когда выход одного функционального блока применяется в качестве оборудования для работы другого блока. На рисунке 6 зажим - устройство, используемое для закрепления детали во время ее сборки, должен быть собран для того, чтобы выполнить сборку детали.

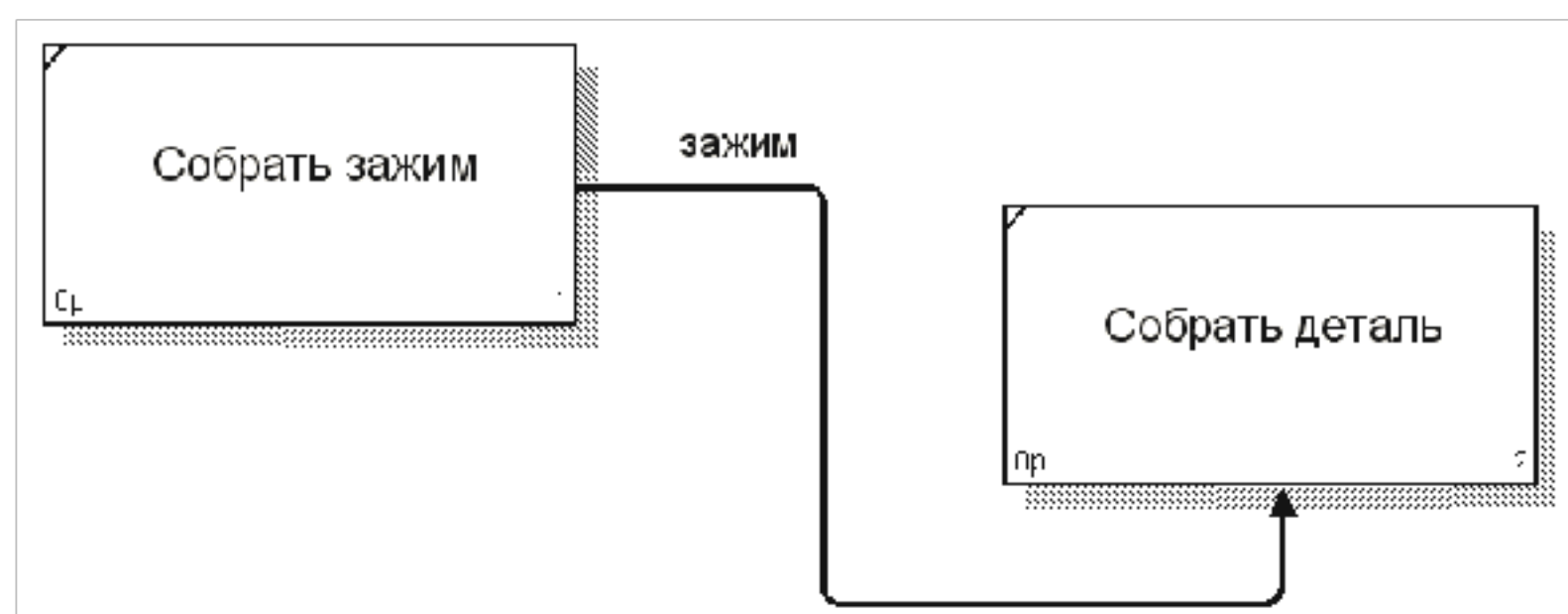


Рисунок 1.6 – Комбинированная стрелка *выход - механизм исполнения*

Обратные связи на вход и на управление применяются в случаях, когда зависимые блоки формируют обратные связи для управляющих ими блоков. На рисунке 7 получаемая от брокеров информация о текущих биржевых курсах применяется для корректировки стратегии игры на бирже.

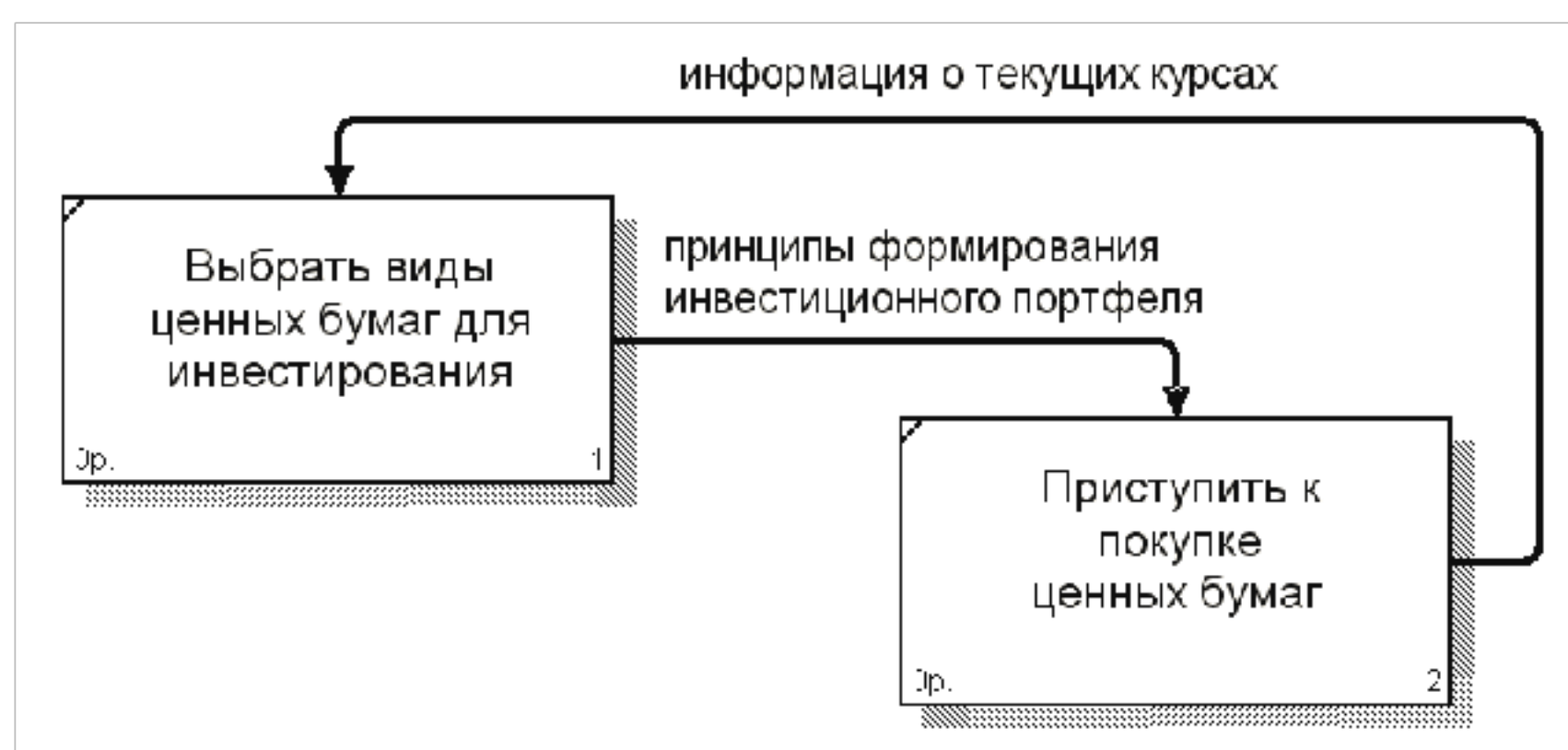


Рисунок 1.7 – Комбинированная стрелка *выход – обратная связь на управление*

Стрелка *выход - обратная связь на вход* обычно применяется для описания циклов повторной обработки чего-либо. Рисунок 8 может служить примером применения стрелки такого типа. Кроме того, связи *выход – обратная связь на вход* могут применяться в случае, если бракованная продукция может заново использоваться в качестве сырья, как это происходит, например, при производстве оконного стекла, когда разбитое в процессе производства стекло перемалывается и переплавляется заново вместе с обыкновенным сырьем.

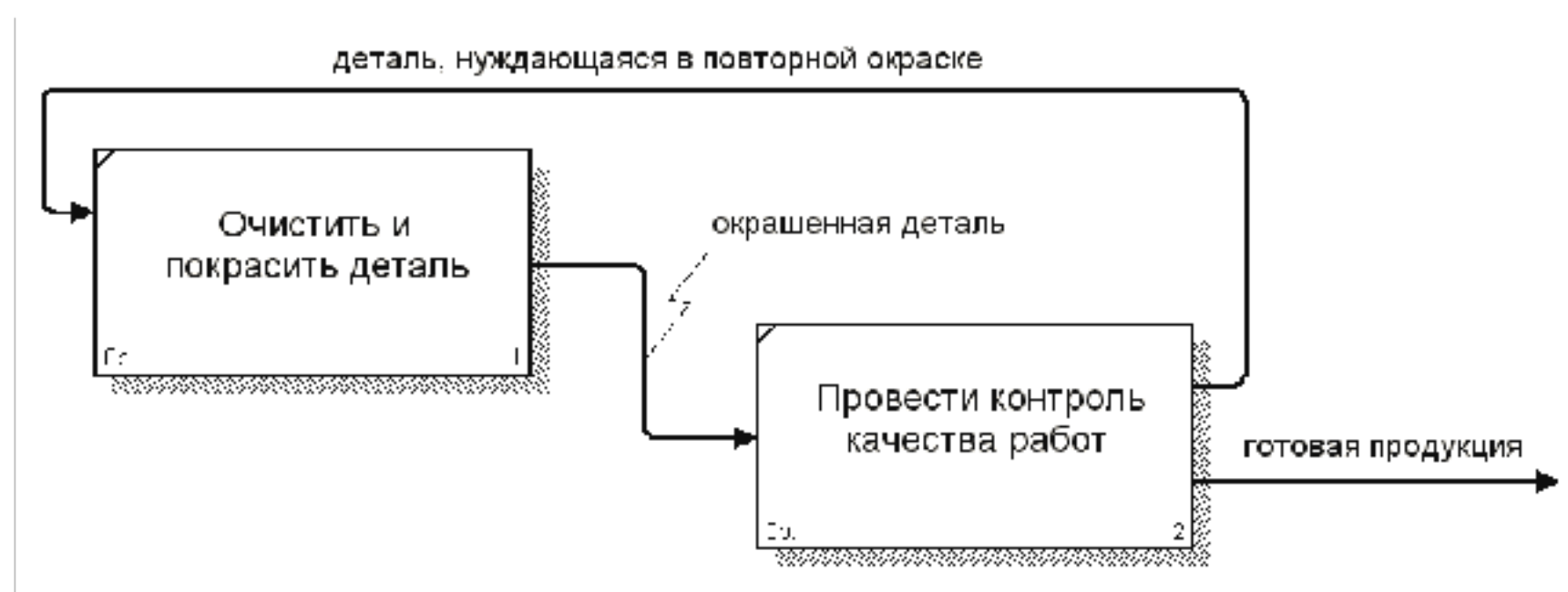
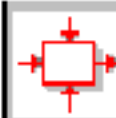


Рисунок 1.8 – Комбинированная стрелка выход – обратная связь на вход

Ветвление и слияние стрелок. Выход функционального блока может использоваться в нескольких других блоках. Фактически чуть ли не главная ценность IDEF0 заключается в том, что эта методология помогает выявить взаимозависимости между блоками системы. Соответственно IDEF0 предусматривает как ветвление, так и слияние стрелок на диаграмме. Разбитые на несколько частей стрелки могут иметь наименования, отличающиеся от наименования исходной стрелки. Исходная и разбитые (или объединенные) стрелки в совокупности называются связанными. Такая техника обычно применяется для того, чтобы отразить использование в процессе только части сырья или информации, обозначаемых исходной стрелкой.

Задание на лабораторную работу

- 1 Запустите **BPwin**. (Кнопка Start  /BPwin ).
- 2 Если появляется диалог **ModelMart Connection Manager**, нажмите на кнопку **Cancel** (Отмена).
- 3 Щелкните по кнопке  . Появляется диалоговое окно **I would like to** (рисунок 1.9). Внесите в текстовое поле **Name** имя модели "Деятельность компании" и выберите **Type** – **Business Process (IDEF0)**. Нажмите кнопку **OK**.

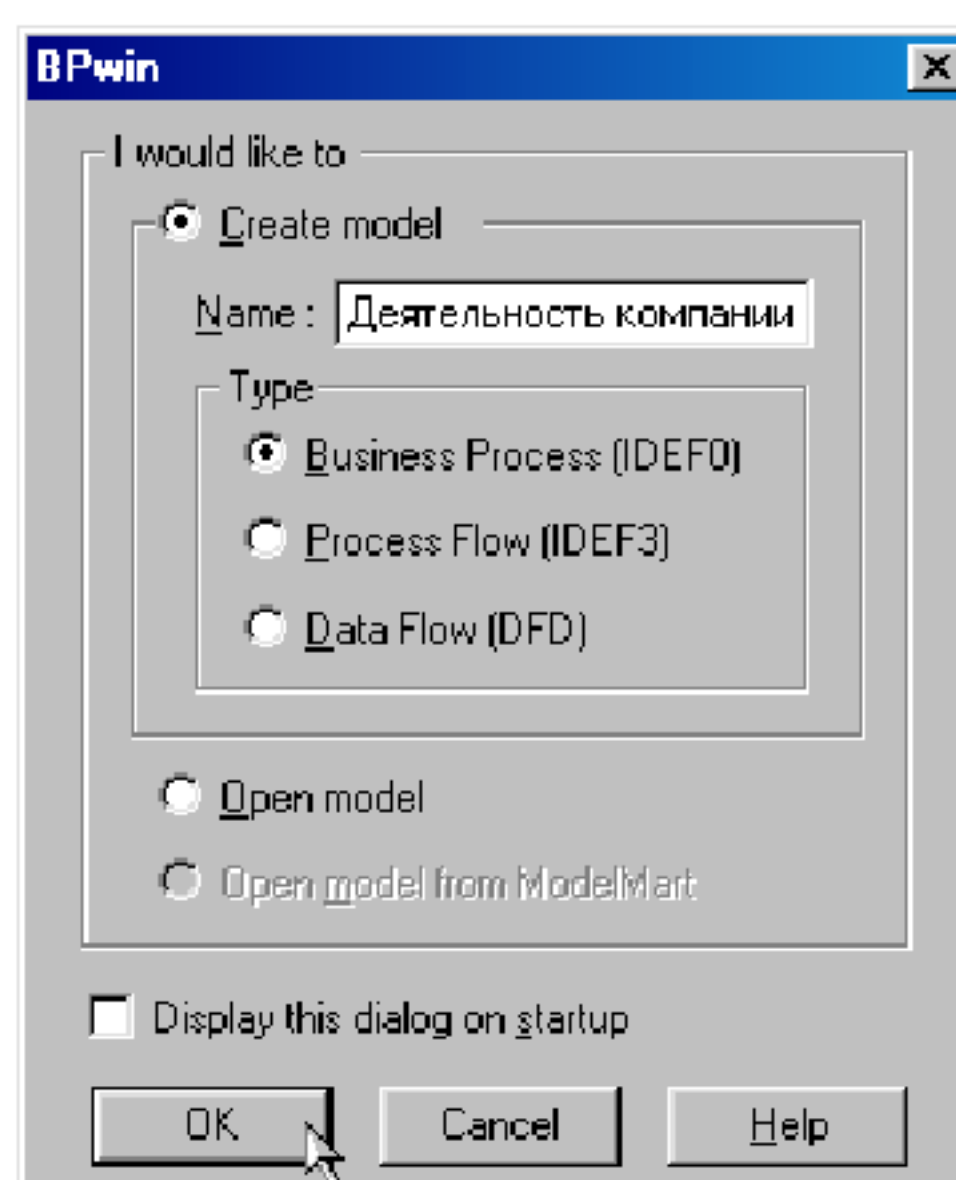
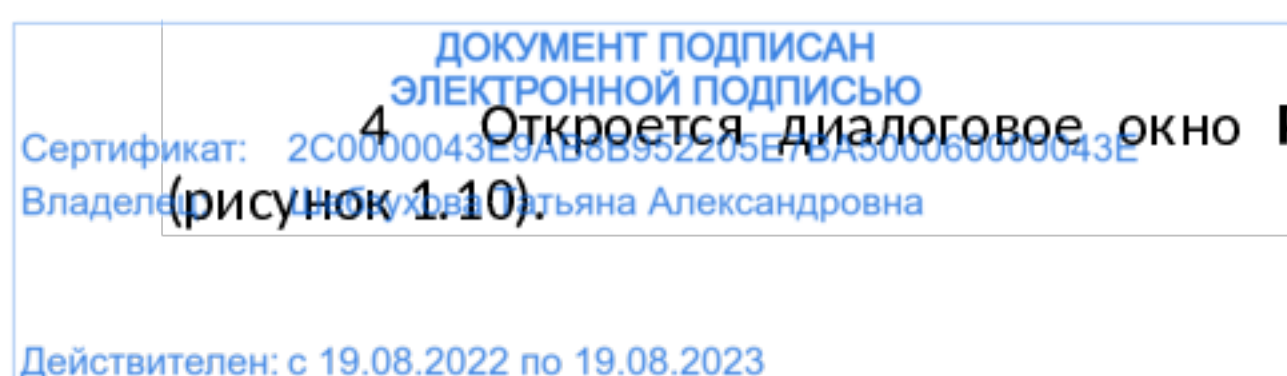


Рисунок 1.9



- 4 Откроется диалоговое окно **Properties for New Models** (Свойства новой модели) (рисунок 1.10).

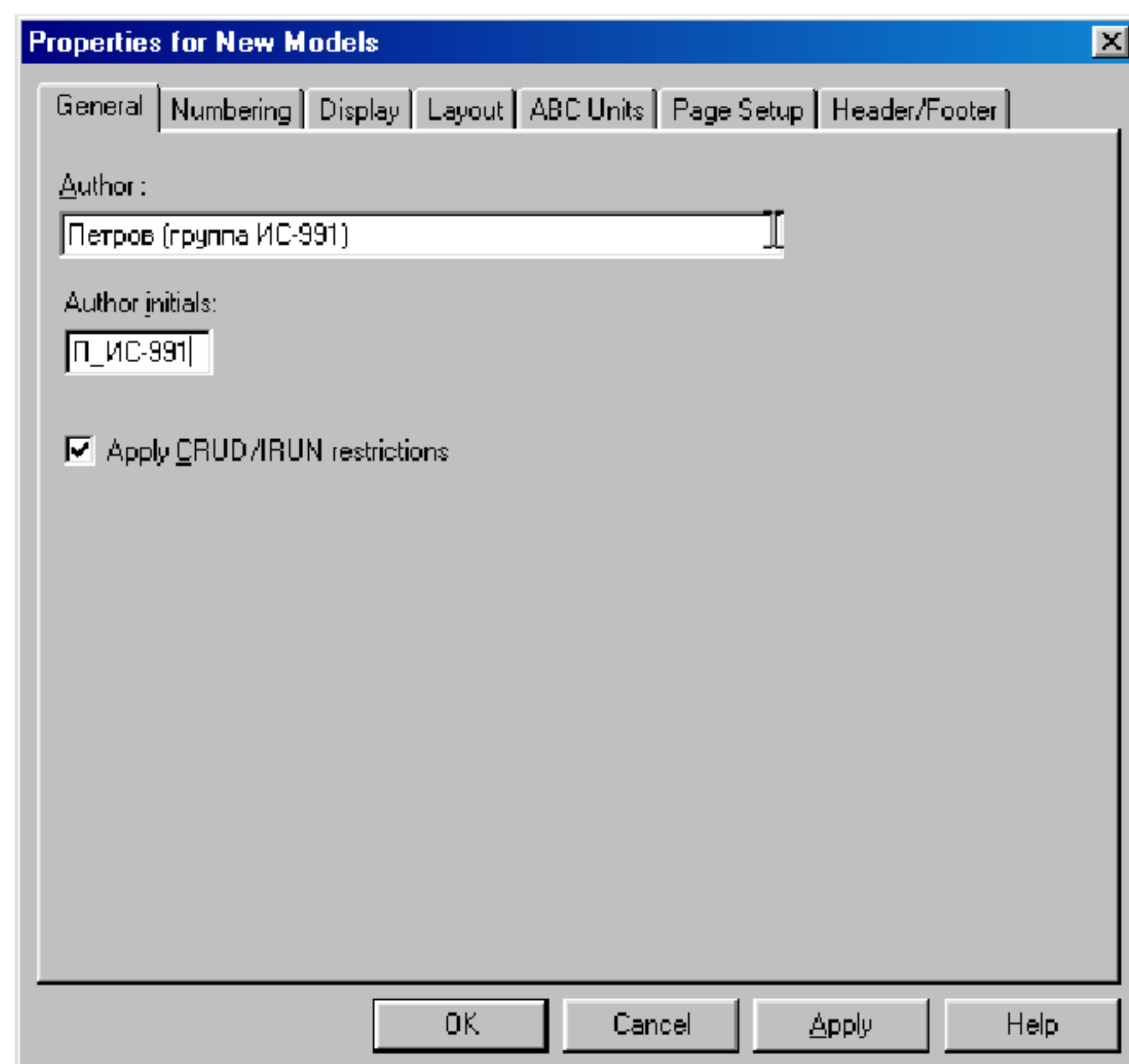


Рисунок 1.10

Введите в текстовое поле **Author** (Автор) имя автора модели и в текстовое поле **Author initials** его инициалы. Нажмите последовательно кнопки **Apply** и **OK**.

- 5 Автоматически создается незаполненная контекстная диаграмма (рисунок 1.11).

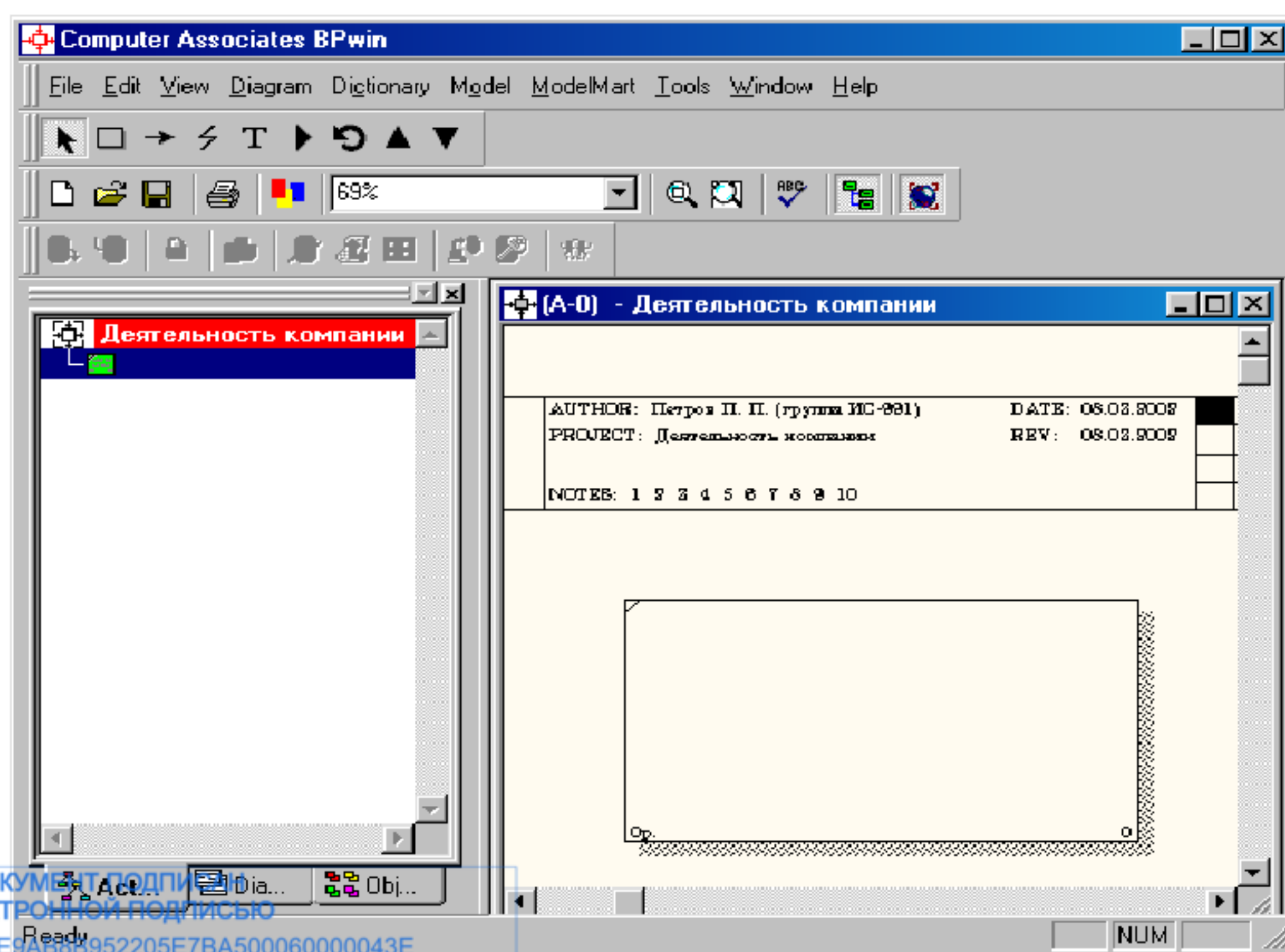


Рисунок 1.11.

6 Обратите внимание на кнопку  на панели инструментов. Эта кнопка включает и выключает инструмент просмотра и навигации – **Model Explorer** (Браузер модели). **Model Explorer** имеет три вкладки – **Activities** (, **Diagrams** () и **Objects** (). Во вкладке **Activities** щелчок правой кнопкой по объекту в браузере модели позволяет выбрать опции редактирования его свойств (рисунок 1.12).

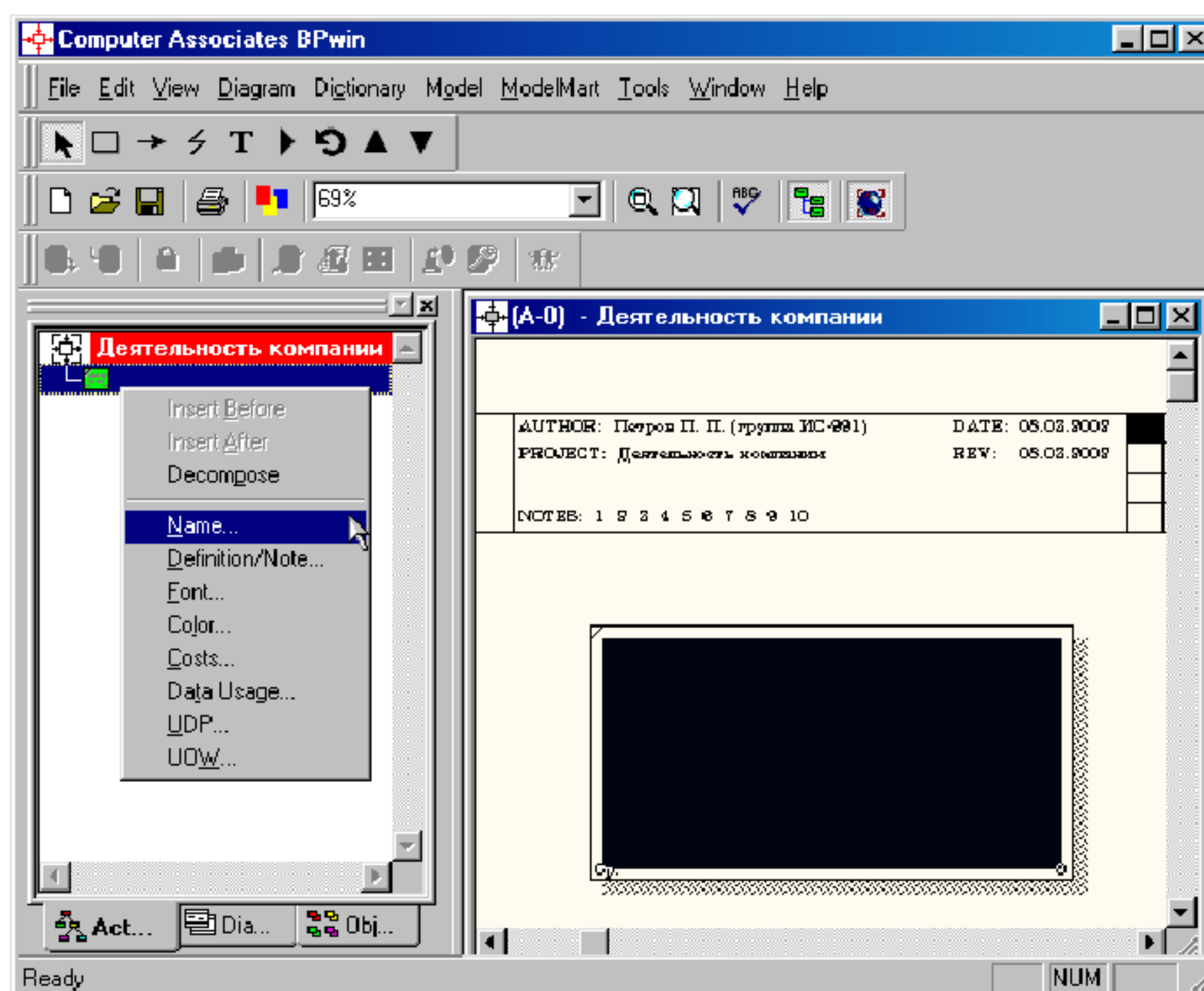


Рисунок 1.12.

7 Если вам непонятно, как выполнить то или иное действие, вы можете вызвать контекстную помощь – клавиша **F1** или воспользоваться меню **Help**.

8 Перейдите в меню **Model/Model Properties**. Во вкладке **General** диалогового окна **Model Properties** в текстовое поле **Model name** следует внести имя модели "Деятельность компании", а в текстовое поле **Project** имя проекта "Модель деятельности компании", и, наконец, в текстовое **Time Frame** (Временной охват) – **AS-IS** (Как есть) (рисунок 1.13).

9 Во вкладке **Purpose** диалогового окна **Model Properties** в текстовое поле **Purpose** (цель) внесите данные о цели разработки модели – " Моделировать текущие (AS-IS) бизнес-процессы компании", а в текстовое поле **Viewpoint** (точка зрения) – "Директор".

Model Properties

Layout | ABC Units | Page Setup | Header/Footer | Shapes | Draw Style

General | Purpose | Definition | Source | Status | Numbering | Display

Model name:
Деятельность компании

Project:
Модель деятельности компании

Author :
Петров П. П. (группа ИС-991)

Author initials:
П_ИС-991 ☒ Apply CRUD/IRUN restrictions

Time Frame:
☒ AS-IS
☐ TO-BE

OK Cancel Apply Help

Рисунок 1.13.

Model Properties

Layout | ABC Units | Page Setup | Header/Footer | Shapes | Draw Style

General | Purpose | Definition | Source | Status | Numbering | Display

Model Name:
Деятельность компании

Purpose:
Моделировать текущие (AS-IS) бизнес процессы компании

Viewpoint:
Директор

OK Cancel Apply Help

Рисунок 1.14.

10 Во вкладке **Definition** диалогового окна **Model Properties** в текстовое поле **Definition** (Определение) внесите "Это учебная модель, описывающая деятельность компании" и в текстовое поле **Scope** (охват) – "Общее управление бизнесом компании: исследование рынка, закупка компонентов, сборка, тестирование и продажа продуктов".

ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
 Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

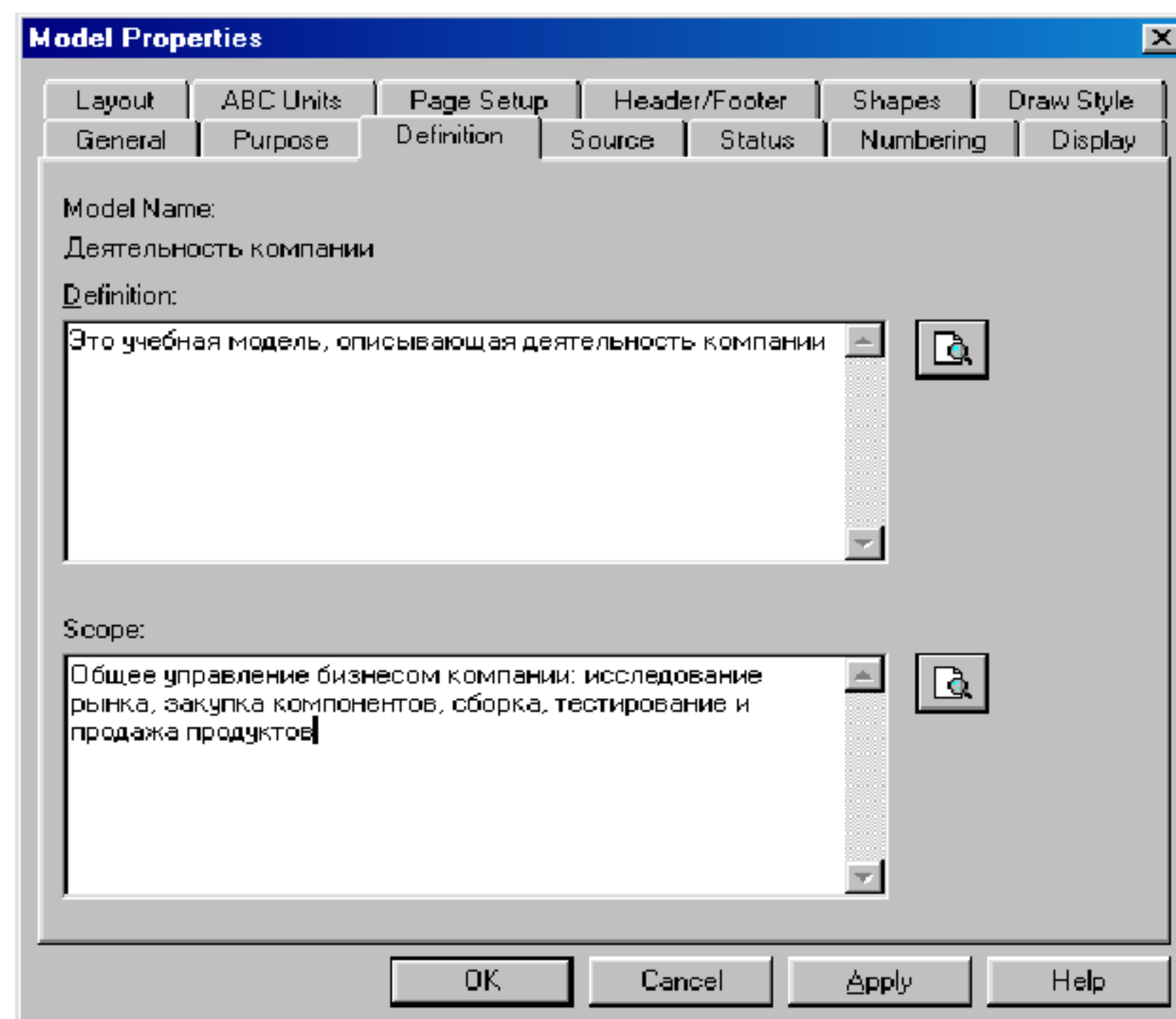


Рисунок 1.15

11 Перейдите на контекстную диаграмму и правой кнопкой мыши щелкните по прямоугольнику представляющему, в нотации **IDEF0**, условное графическое обозначение работы. В контекстном меню выберите опцию **Name** (рисунок 1.16). Во вкладке **Name** внесите имя "Деятельность компании" (рисунок 1.17).

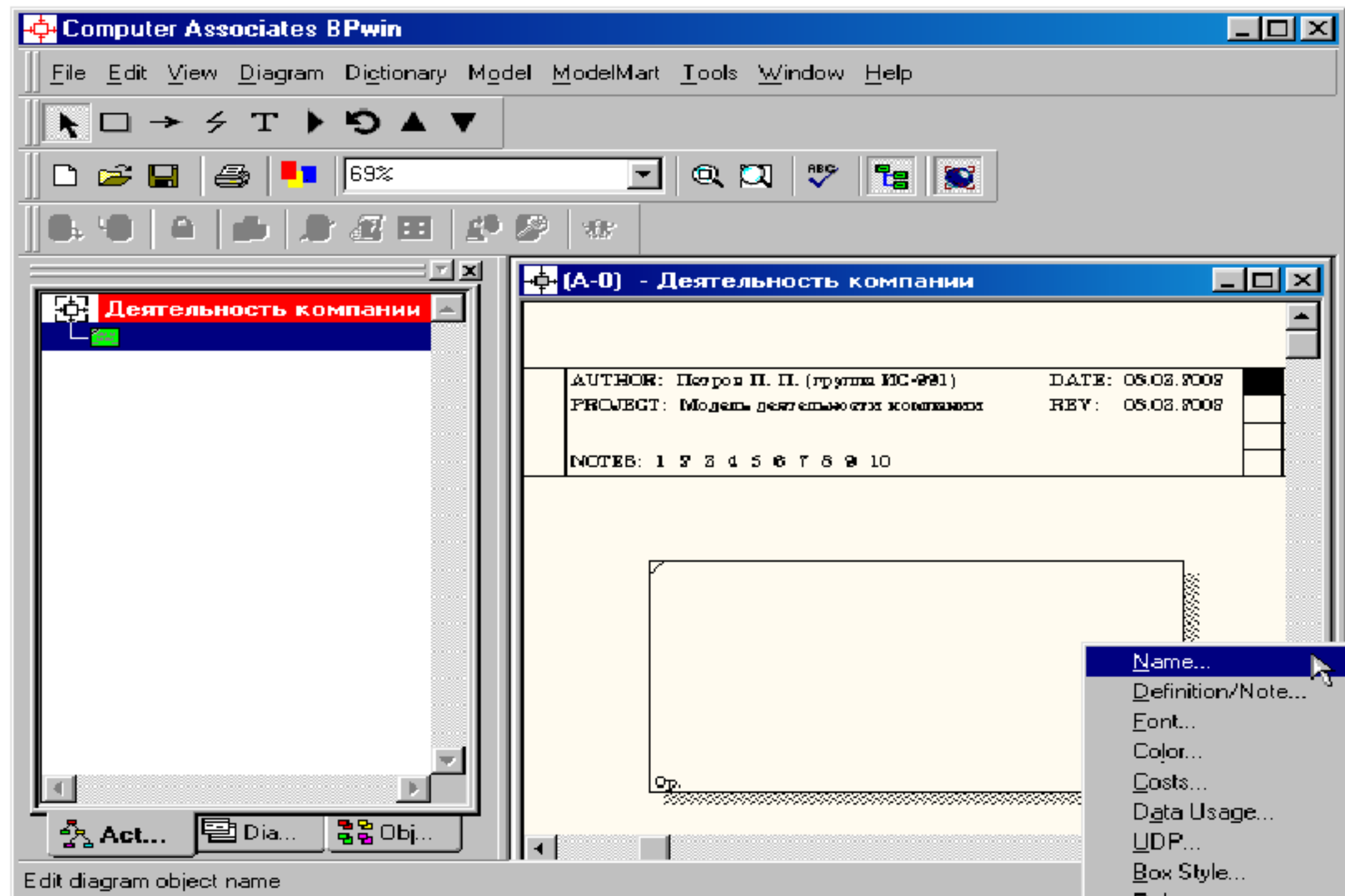


Рисунок 1.16.

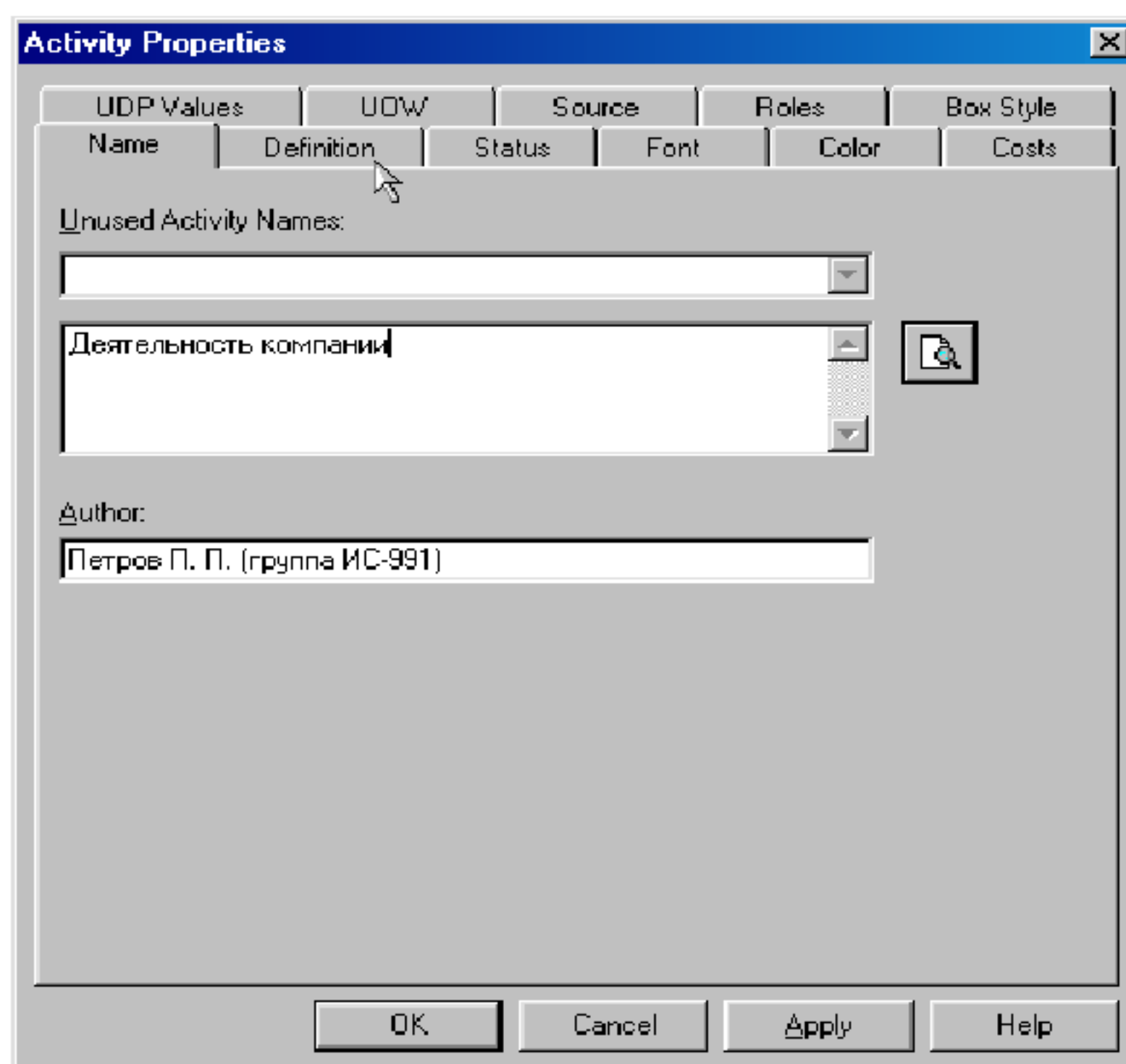


Рисунок 1.17.

Во вкладке **Definition** диалогового окна **Activity Properties** в текстовое поле **Definition** (Определение) внесите "Текущие бизнес-процессы компании" (рисунок 1.18). Текстовое поле **Note** (Примечания) оставьте незаполненным.

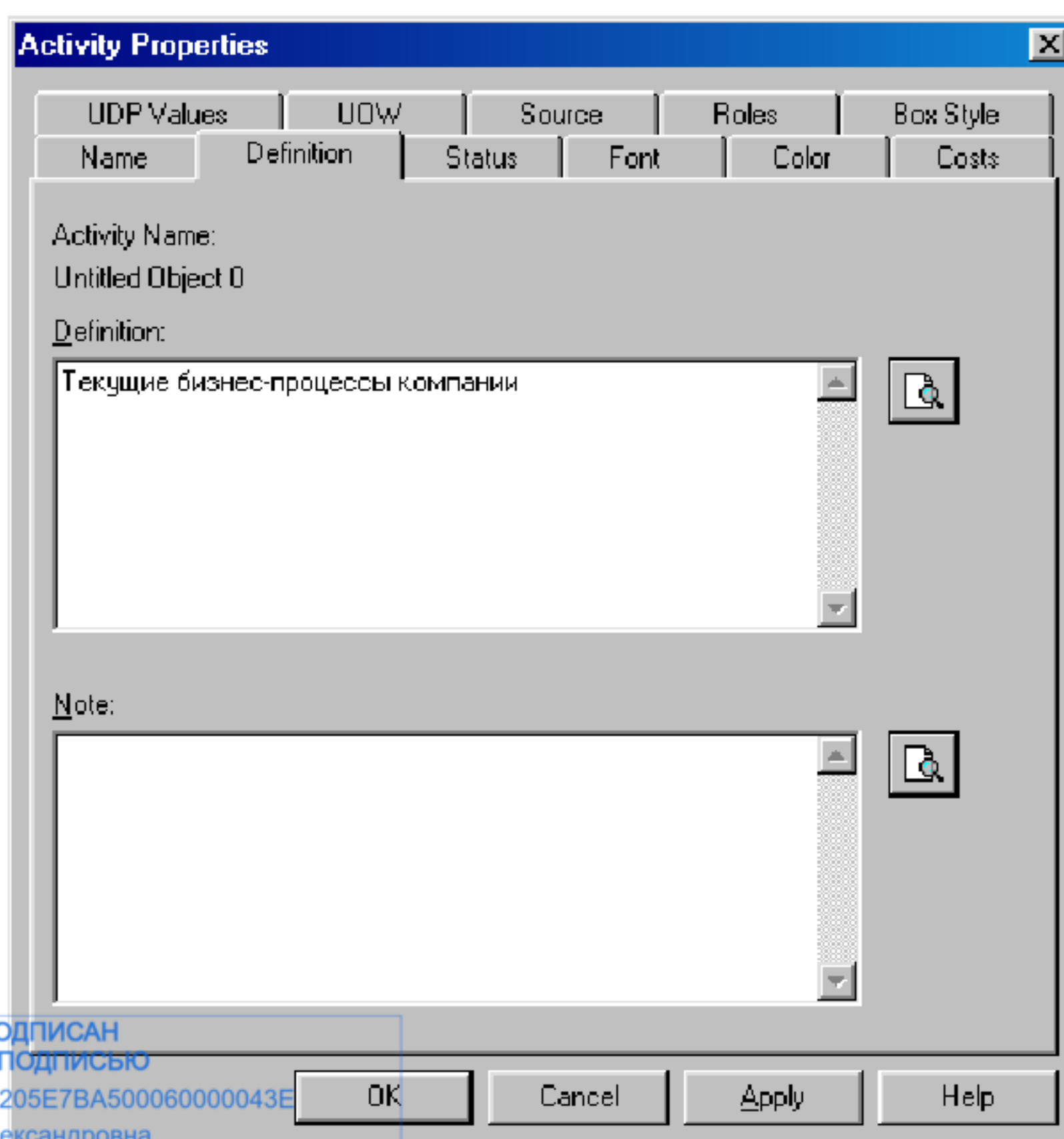


Рисунок 1.18

12 Создайте **ICOM**-стрелки на контекстной диаграмме (таблица 1).
Таблица 1. Стрелки контекстной диаграммы

Таблица 1. Стрелки контекстной диаграммы

Название стрелки (Arrow Name)	Определение стрелки (Arrow Definition)	Тип стрелки (Arrow Type)
Звонки клиентов	Запросы информации, заказы, техподдержка и т. д.	Input
Правила и процедуры	Правила продаж, инструкции по сборке, процедуры тестирования, критерии производительности и т. д.	Control
Проданные продукты	Настольные и портативные компьютеры	Output
Бухгалтерская система	Оформление счетов, оплата счетов, работа с заказами	Mechanism

13. С помощью кнопки **Т** внесите текст в поле диаграммы - точку зрения и цель (рисунок 1.19). Результат показан на рисунке 1.20.

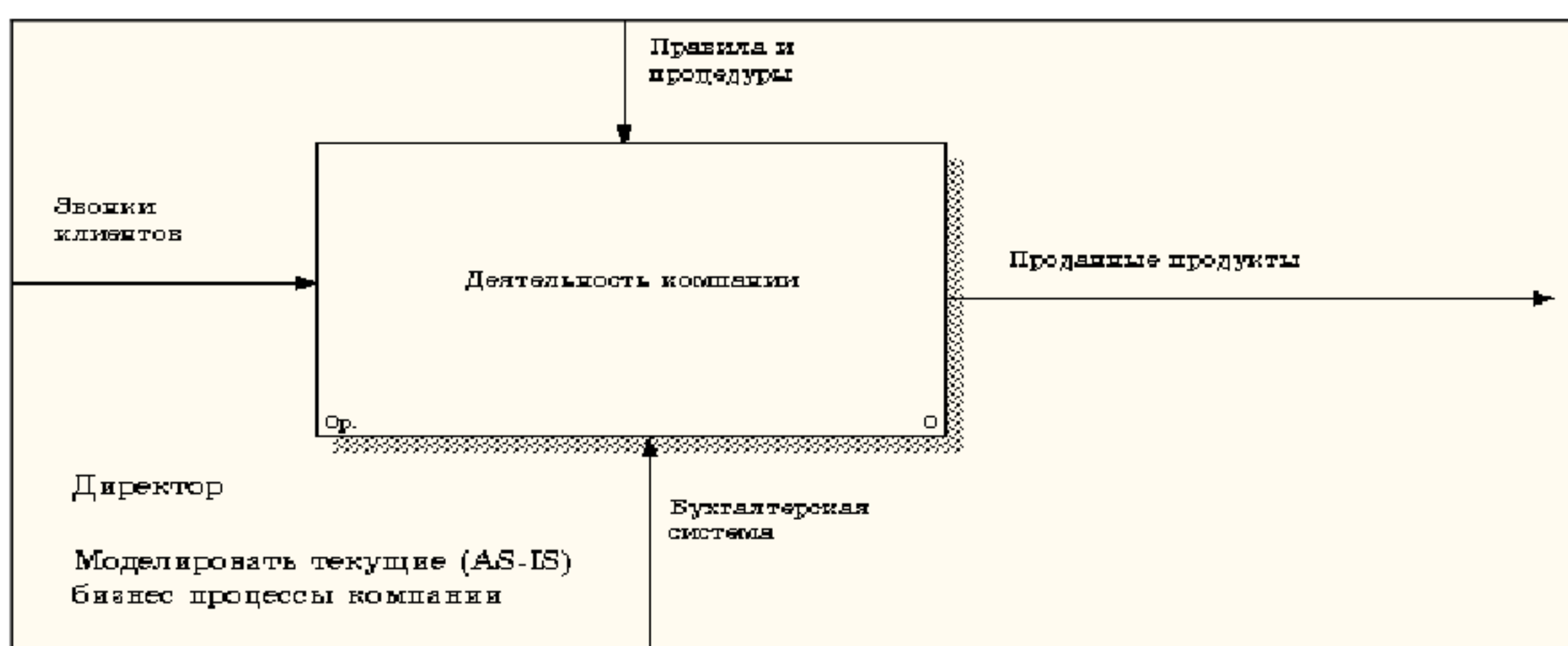
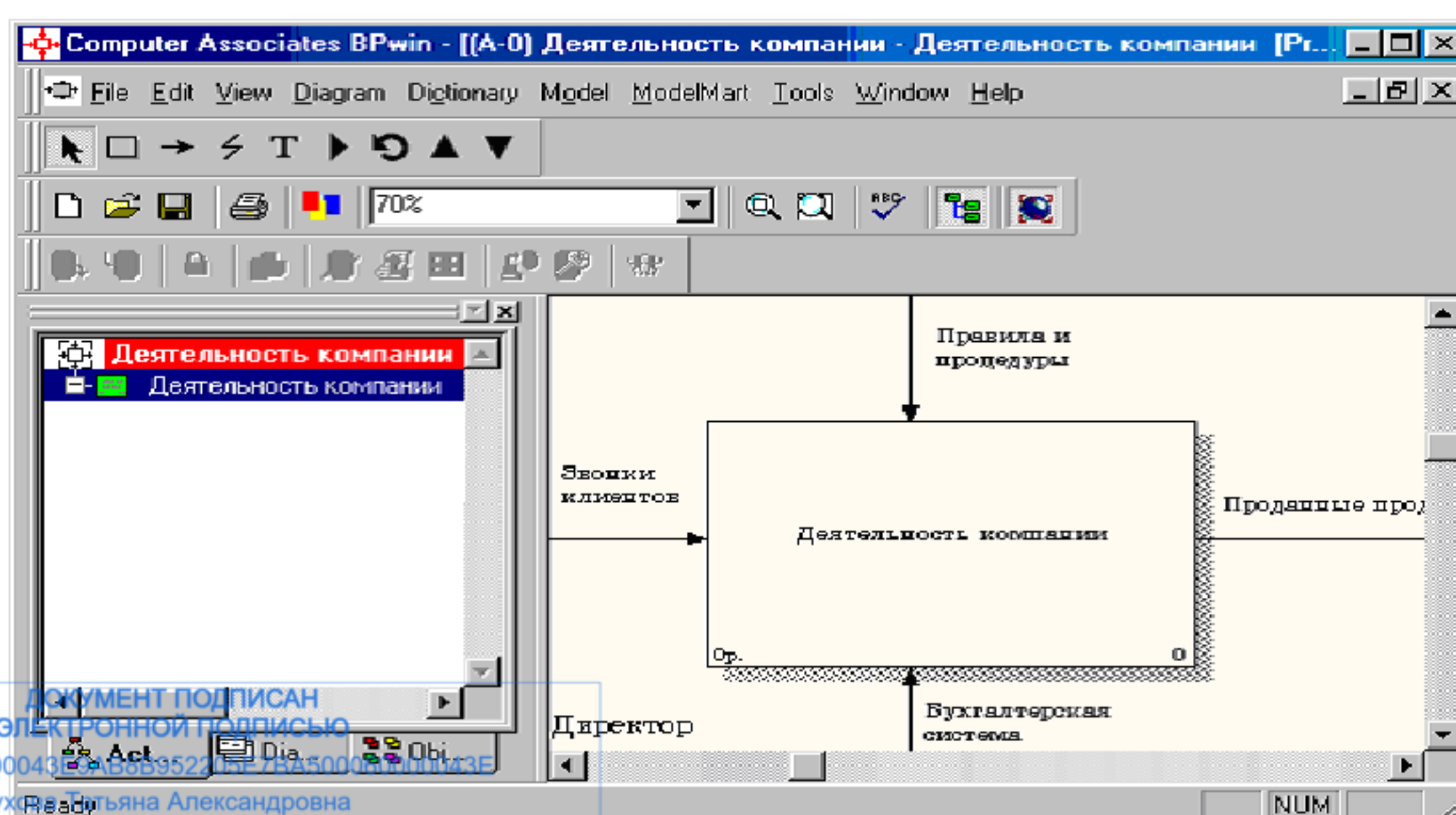


Рисунок 1.19.



Сертификат: 2C0000043E1A8B55234E78A500000000000043E
Владелец: Шебзухов Владислав Евгеньевич

Действителен: с 19.08.2022 по 19.08.2023

Рисунок 1.20.

Создайте отчет по модели. В меню **Tools/Reports/Model Report** (рисунок 1.21) задайте опции генерирования отчета (установите галочки) и нажмите кнопку **Preview** (Предварительный просмотр) (рисунок 1.22).

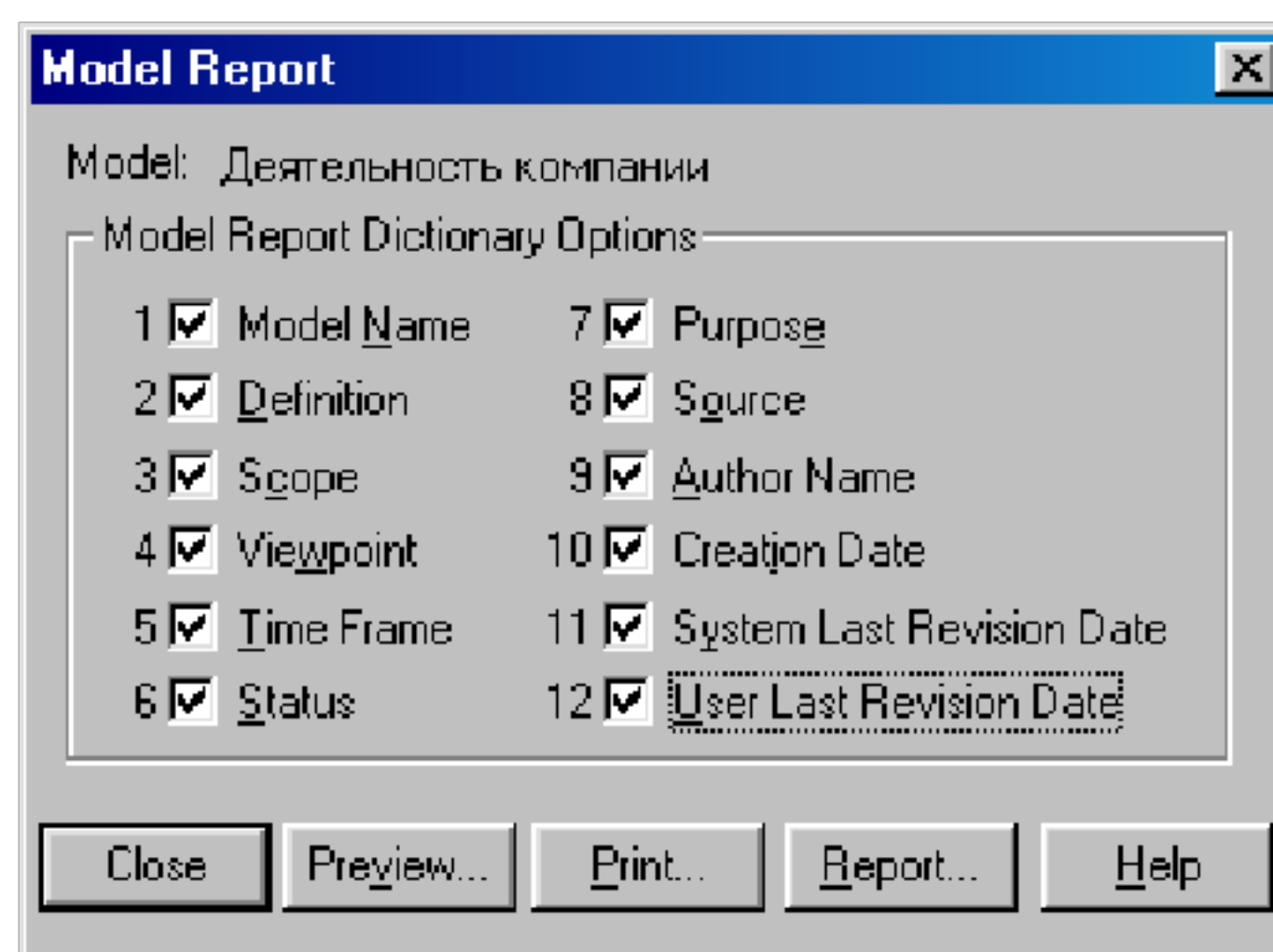


Рисунок 1.21.

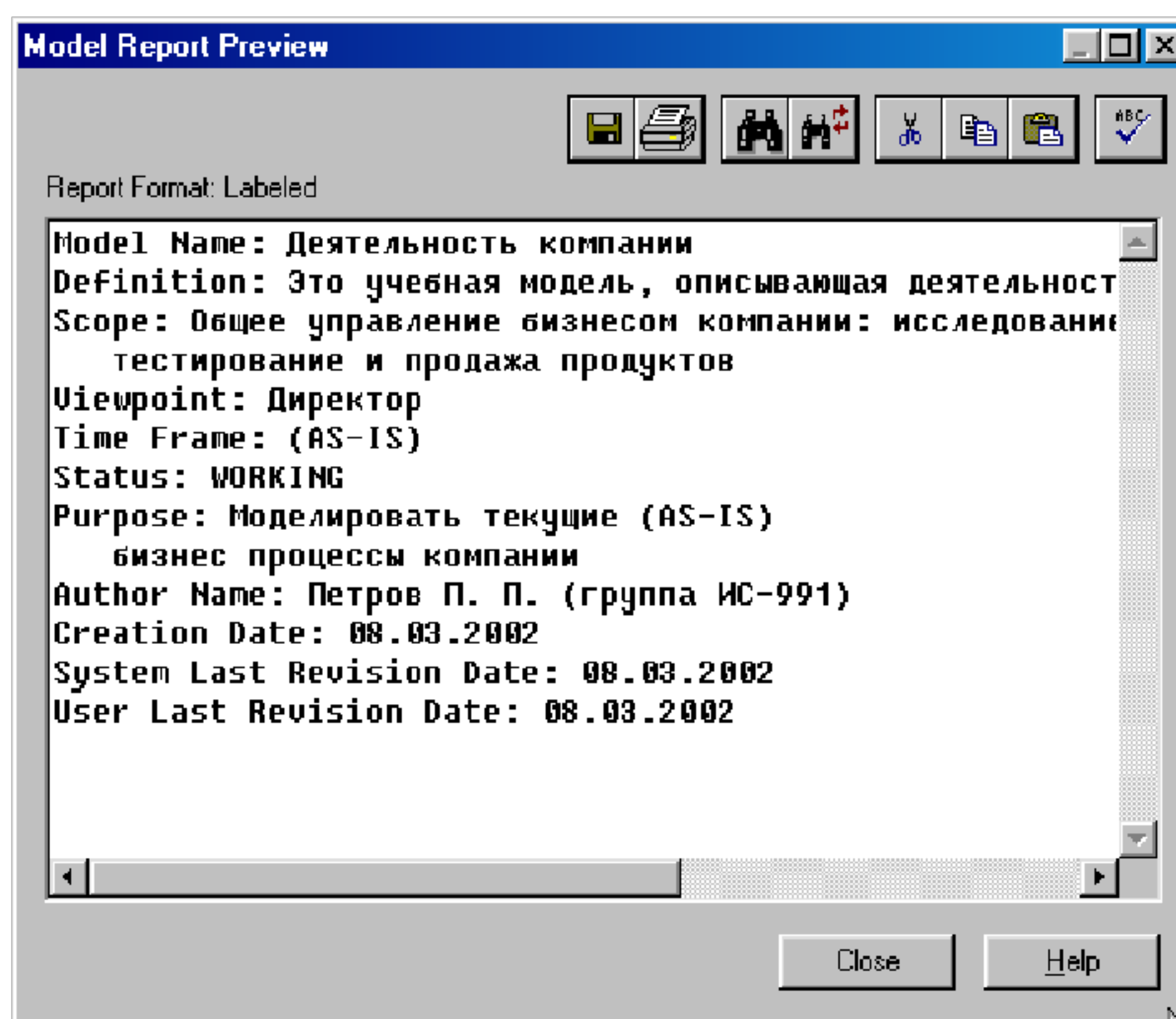


Рисунок 1.22.

Создание диаграммы декомпозиции

1 Выберите кнопку  перехода на нижний уровень в палитре инструментов и в диалоговом окне **Activity Box Count** (рисунок 1.23) установите число работ на диаграмме нижнего уровня – 3 и нажмите кнопку **OK**.

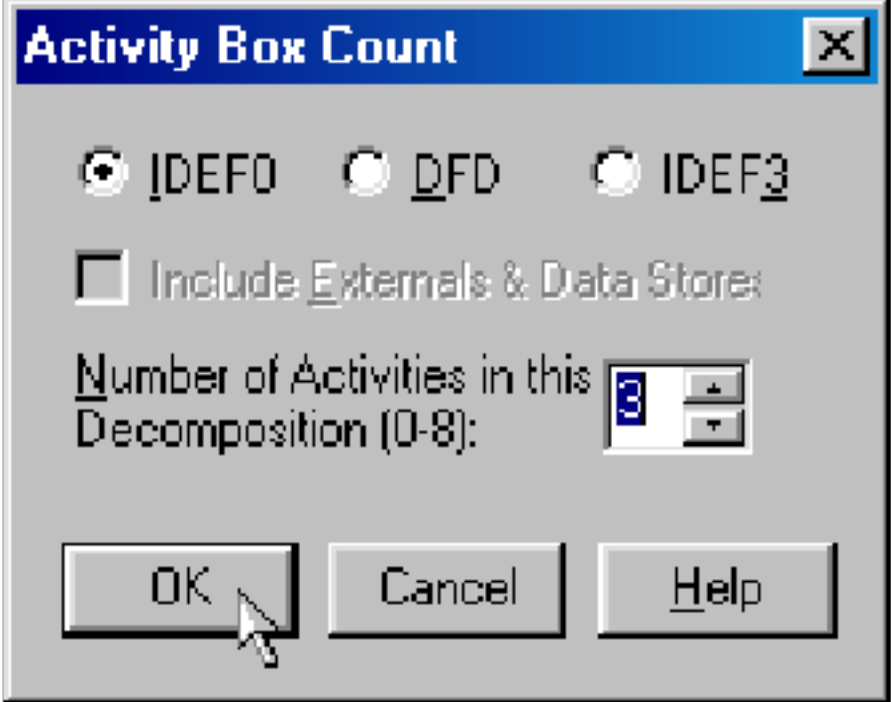


Рисунок 1.23.

2 Автоматически будет создана диаграмма декомпозиции (рисунок 1.24).

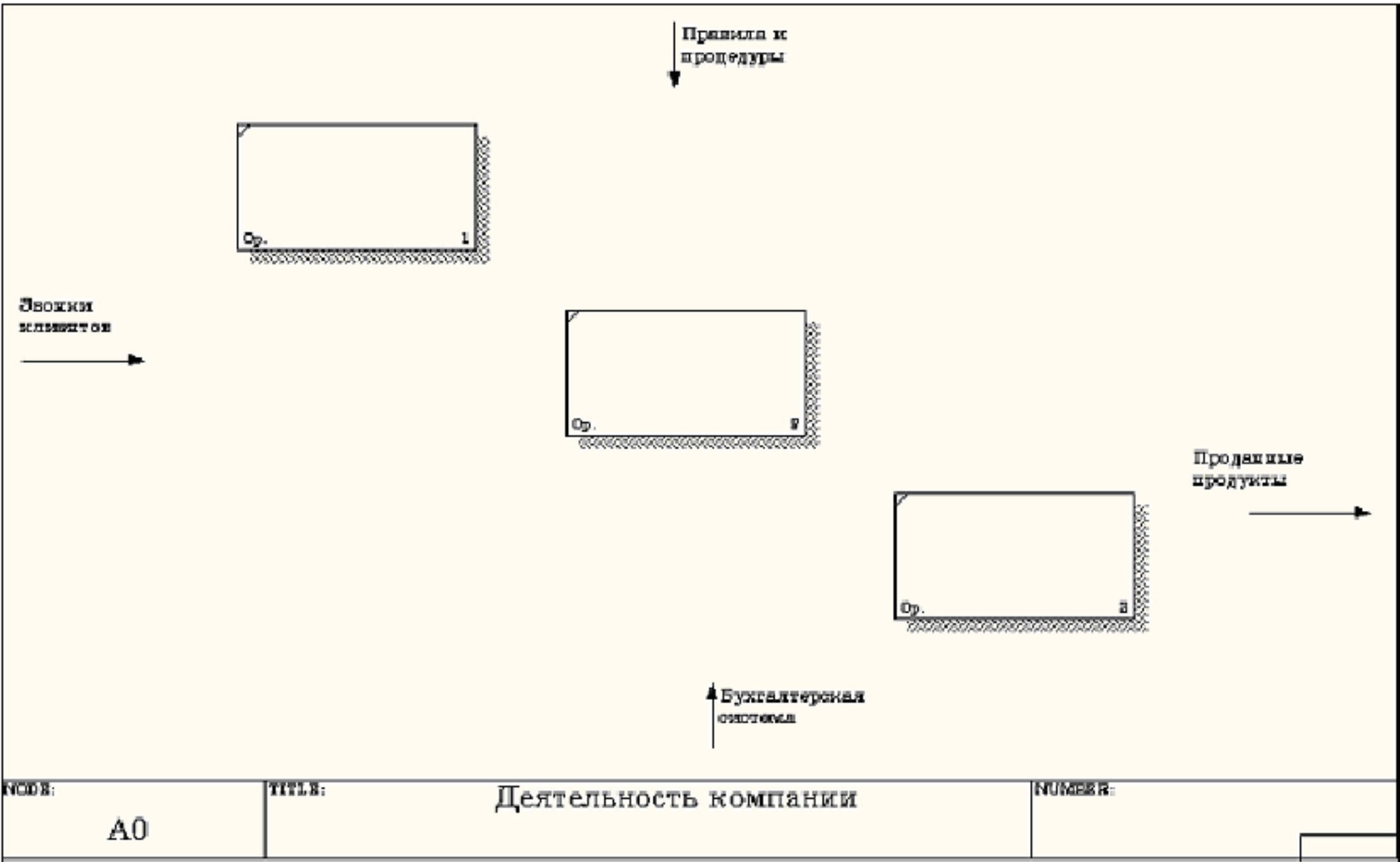


Рисунок 1.24.

Правой кнопкой мыши щелкните по работе расположенной в левом верхнем углу области редактирования модели, выберите в контекстном меню опцию **Name** и внесите имя работы. Повторите операцию для оставшихся двух работ. Затем внесите определение для каждой работы согласно данным таблицы 2.

Таблица 2

Название работы (Activity Name)	Определение работы (Activity Definition)
Продажи и маркетинг	Телемаркетинг и презентации, выставки

Сборка и тестирование компьютеров	Сборка и тестирование настольных и портативных компьютеров
Отгрузка и получение	Отгрузка заказов клиентам и получение компонентов от поставщиков

Диаграмма декомпозиции примет вид представленный на рисунке 1.25.

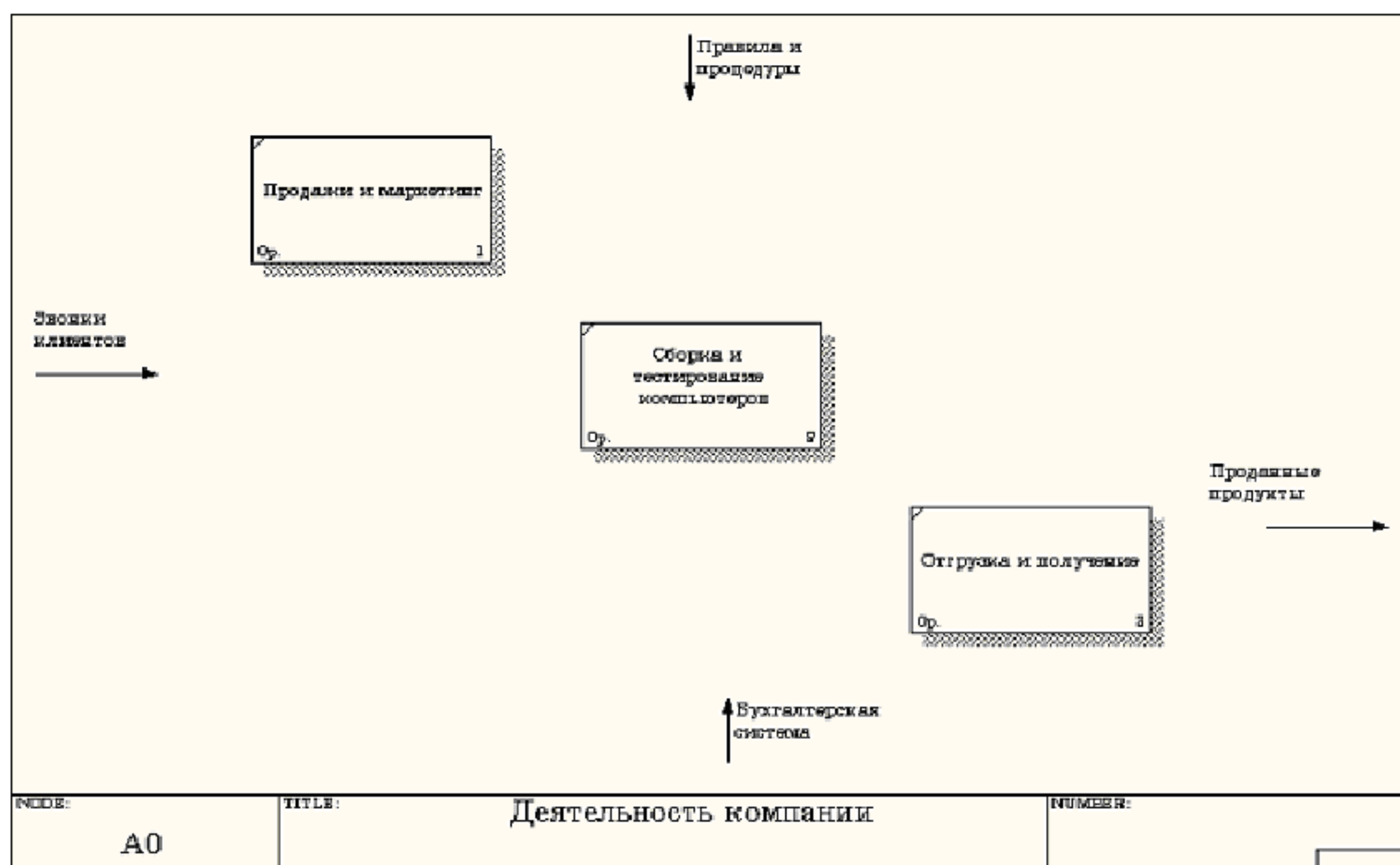


Рисунок 1.25.

3 Для изменения свойств работ после их внесения в диаграмму можно воспользоваться словарем работ (рисунок 1.26). Вызов словаря производится при помощи пункта главного меню **Dictionary /Activity**.

Activity Dictionary		
Name	Definition	Author
Деятельность	Текущие бизнес-процессы компании	Петров П. П. (груп
Отгрузка и пол	Отгрузка заказов клиентам и получение компонентов от поставщиков	Петров П. П. (груп
Продажи и мар	Телемаркетинг и презентации, выставки	Петров П. П. (груп
Сборка и тестирование компьютеров	Сборка и тестирование настольных и портативных компьютеров	Петров П. П. (группа ИС-991)

Рисунок 1.26.

Если описать имя и свойства работы в словаре, ее можно будет внести в диаграмму позже с помощью кнопки  в палитре инструментов. Невозможно удалить работу из словаря, если она используется на какой-либо диаграмме. Если работа удаляется из диаграммы, из словаря она не удаляется. Имя и описание такой работы может быть использовано в дальнейшем. Для добавления работы в словарь необходимо перейти в конец списка и щелкнуть правой кнопкой по последней строке. Возникает новая строка, в которой нужно внести имя и свойства работы. Для



Рисунок 1.27.

тестирование компьютеров" и переименуйте ее в "Правила сборки и тестирования" (рисунок 1.28).

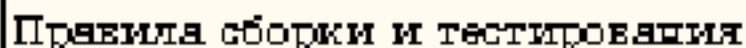


Рисунок 1.28.

критерии производительности и т. д." Правой кнопкой мыши щелкните по ветви стрелки механизма работы "Продажи и маркетинг" и переименуйте ее как "Система оформления заказов" (рисунок 1.29).

Владелец: Шебазова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

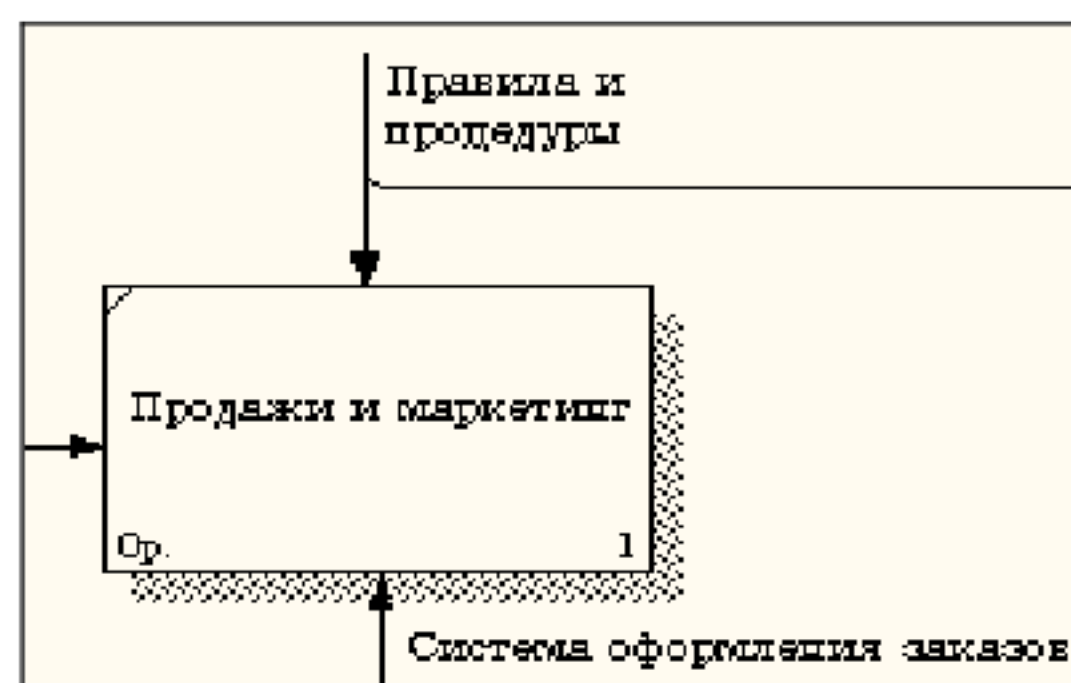


Рисунок 1.29

Альтернативный метод внесения имен и свойств стрелок – использование словаря стрелок (вызов словаря - меню **Dictionary/ Arrow**). Если внести имя и свойства стрелки в словарь (рисунок 1.30), ее можно будет внести в диаграмму позже.

Arrow Dictionary			
Name	Definition	Author	Status
Бухгалтерская с		Петров П. П. (группа)	WORKING
Звонки клиентов		Петров П. П. (группа)	WORKING
Маркетинговые		Петров П. П. (группа)	WORKING
Правила и проце		Петров П. П. (группа)	WORKING
Правила сборки	Инструкции по сборке, процедуры тестирования, критерии	Петров П. П. (группа)	WORKING
Прданные продк	Настольные и портативные компьютеры	Петров П. П. (группа)	WORKING
Проданные продч		Петров П. П. (группа)	WORKING
Система оформл		Петров П. П. (группа)	WORKING
			WORKING

Рисунок 1.30

Стрелку нельзя удалить из словаря, если она используется на какой-либо диаграмме. Если удалить стрелку из диаграммы, из словаря она не удаляется. Имя и описание такой стрелки может быть использовано в дальнейшем. Для добавления стрелки необходимо перейти в конец списка и щелкнуть правой кнопкой по последней строке. Возникает новая строка, в которой нужно внести имя и свойства стрелки.

6 Создайте новые внутренние стрелки так, как показано на рисунке 1.31.

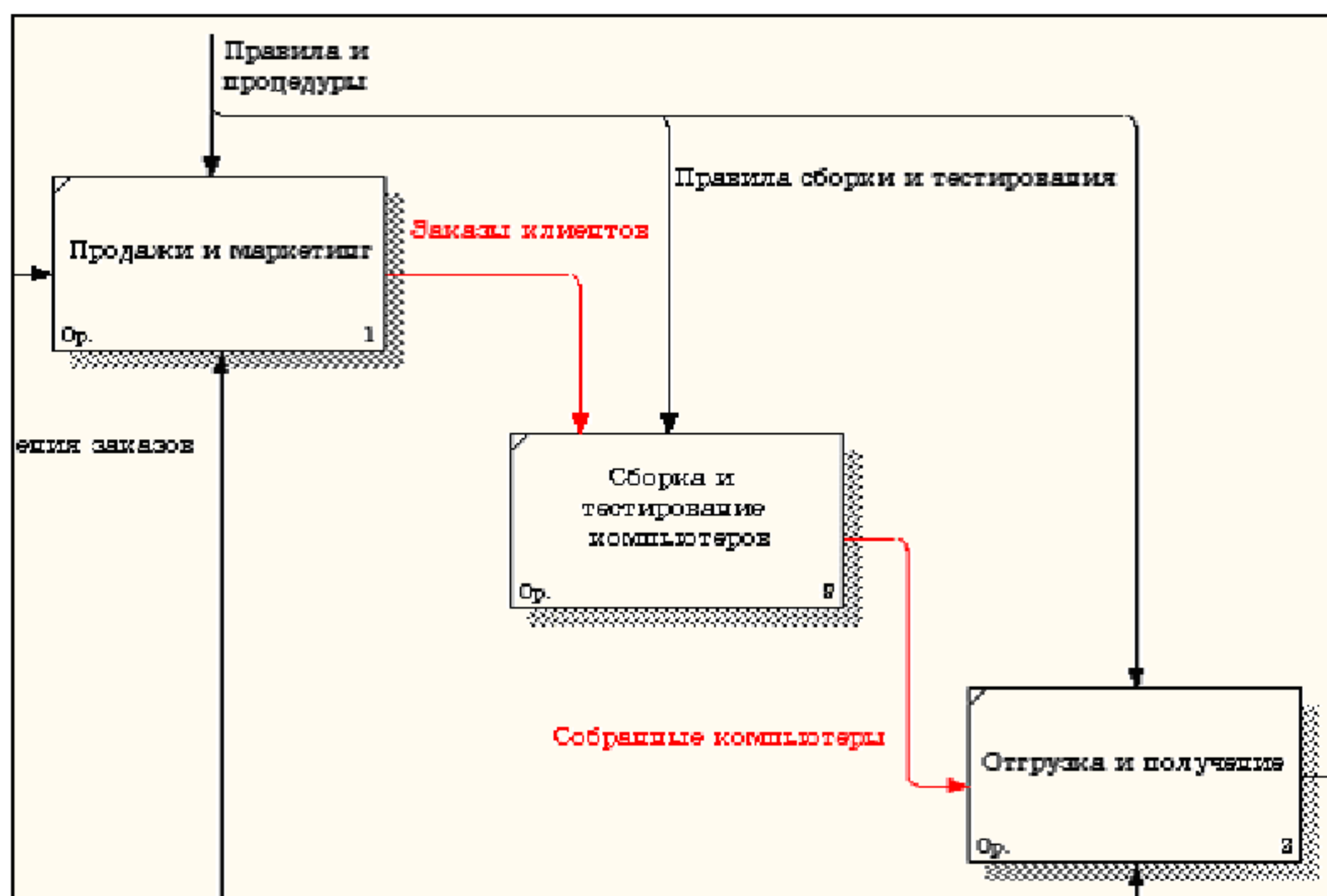


Рисунок 1.31.

Создайте стрелку обратной связи (по управлению) "Результаты сборки и тестирования", идущую от работы "Сборка и тестирование компьютеров" к работе "Продажи и маркетинг". Измените, при необходимости, стиль стрелки (толщина линий) и установите опцию **Extra Arrowhead** (Дополнительный Наконечник стрелы) (из контекстного меню). Методом **drag&drop** перенесите имена стрелок так, чтобы их было удобнее читать. Если необходимо, установите из контекстного меню **Squiggle** (Загогулину). Результат возможных изменений показан на рисунке 1.32.

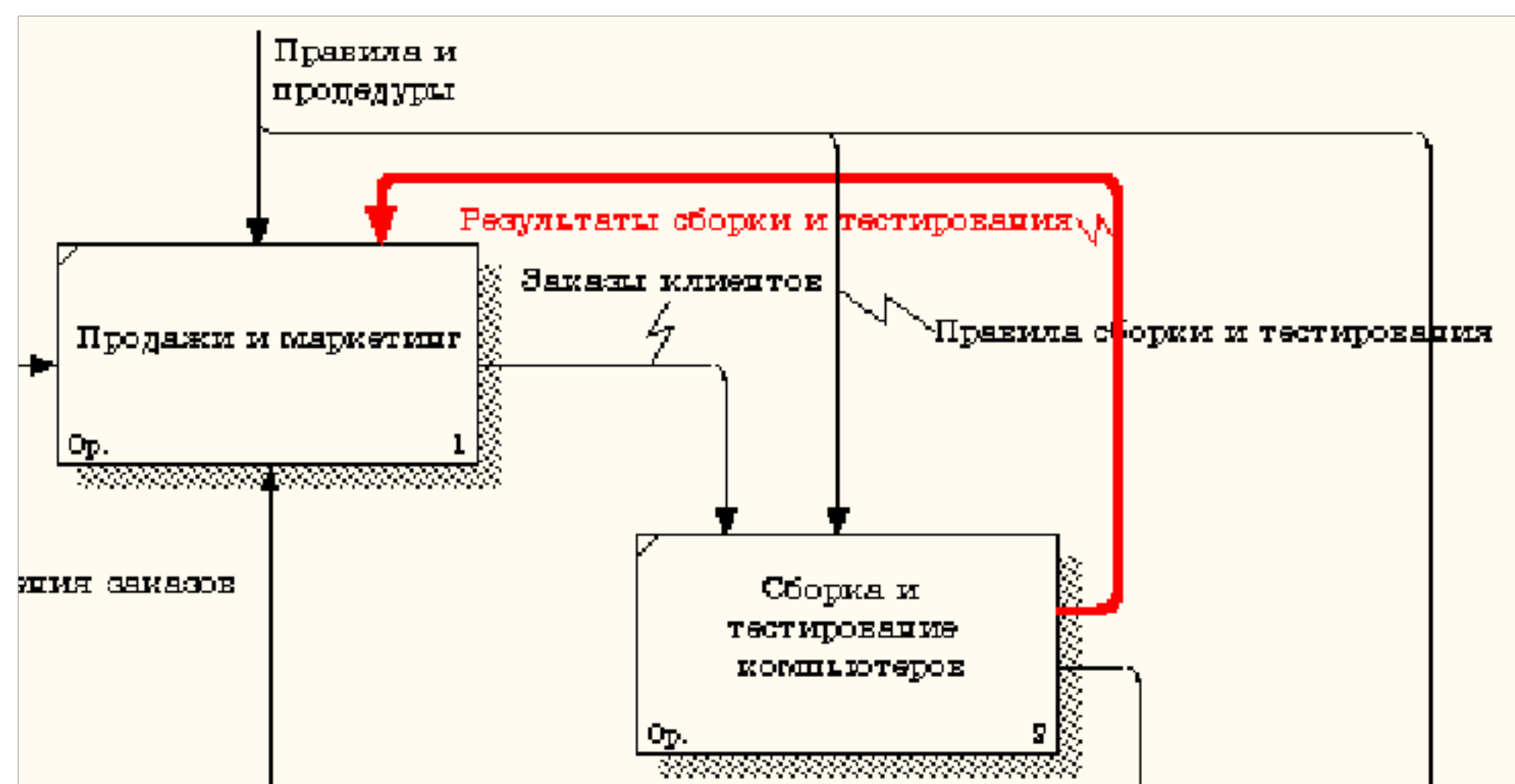


Рисунок 1.32.

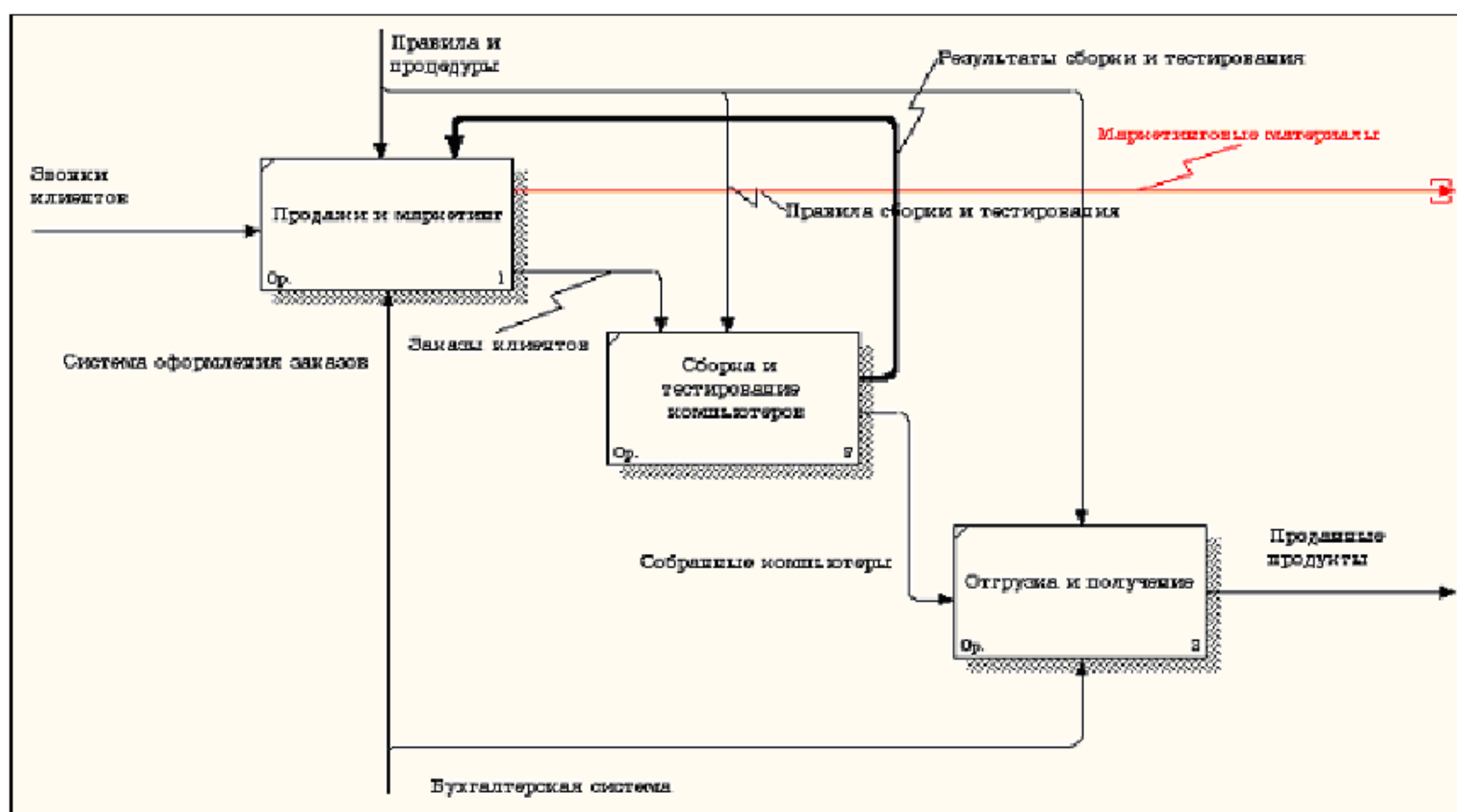


Рисунок 1.33.

Щелкните правой кнопкой мыши по квадратным скобкам и выберите пункт меню **Arrow Tunnel** (рисунок 1.34).

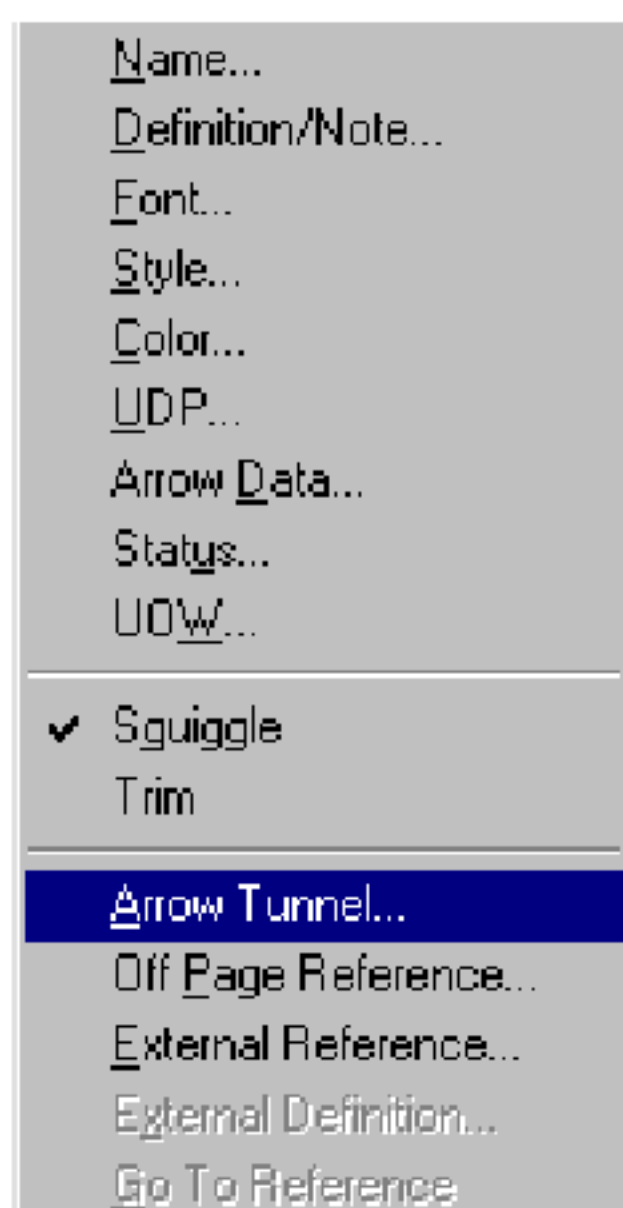


Рисунок 1.34.

В диалоговом окне **Border Arrow Editor** (Редактор Граничных Стрелок) выберите опцию **Resolve it to Border Arrow** (Разрешить как Граничную Стрелку) (рисунок 1.35).

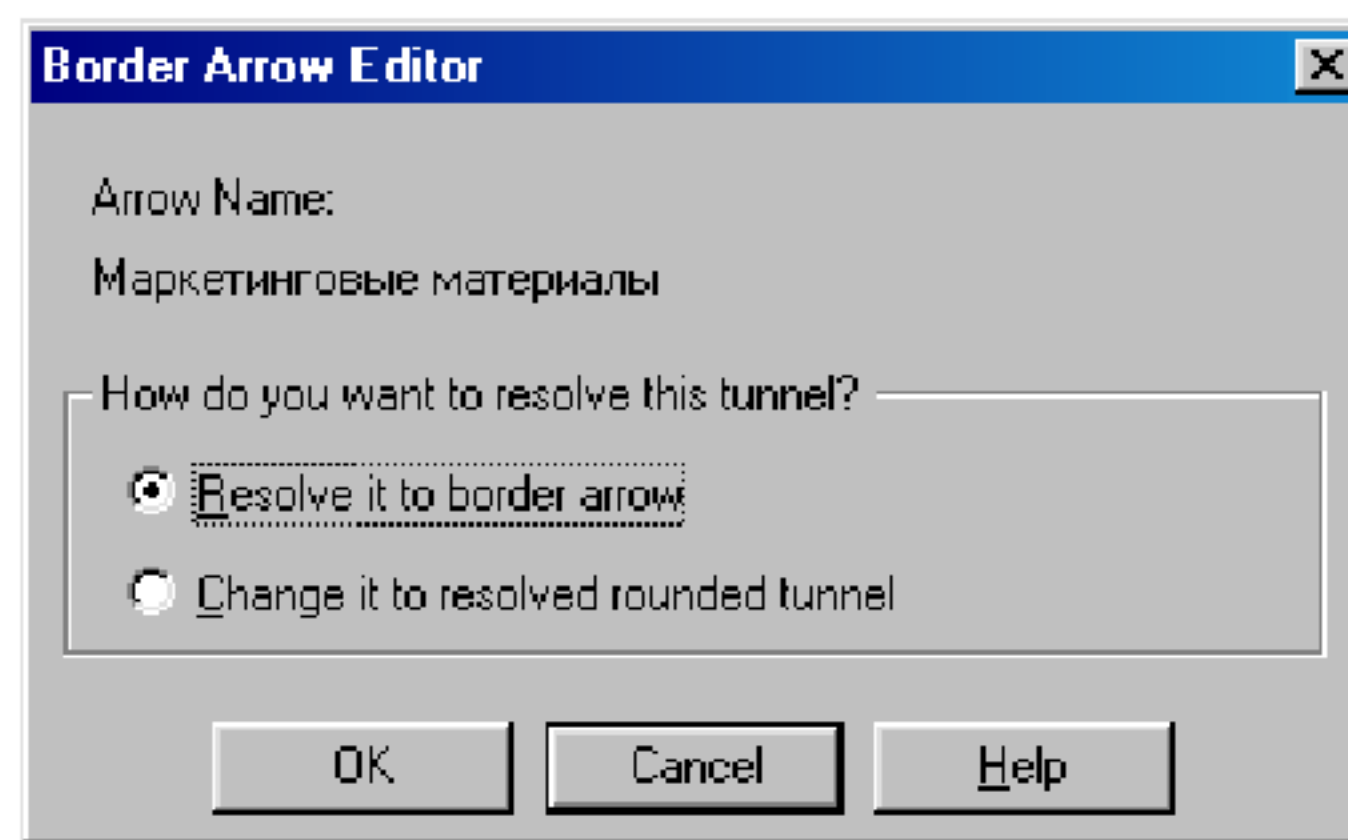


Рисунок 1.35.

Для стрелки "Маркетинговые материалы" выберите опцию **Trim** (Упорядочить) из контекстного меню. Результат показан на рис. 1.36.

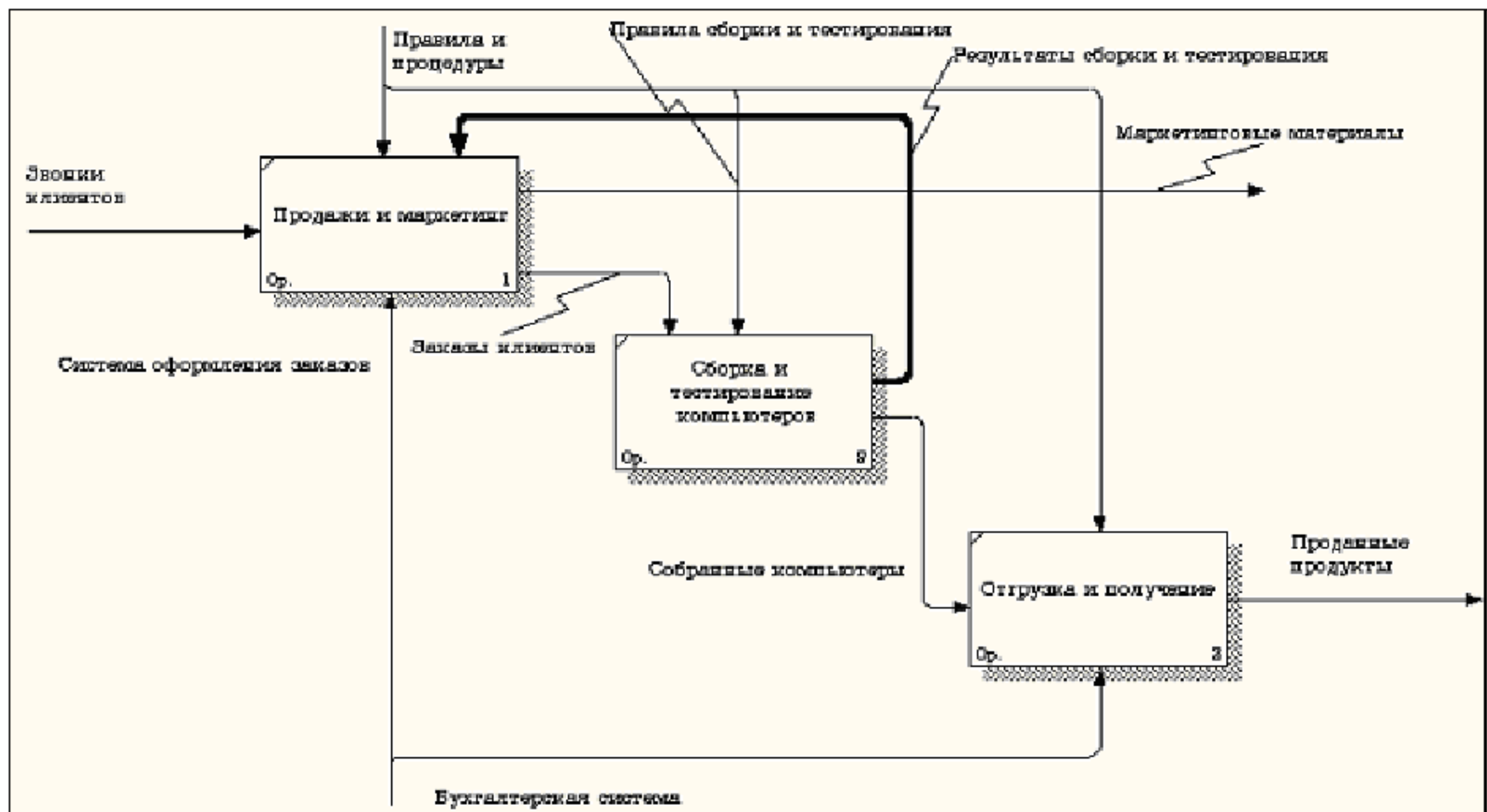


Рисунок 1.36.

Контрольные вопросы

При защите лабораторной работы студент должен ориентироваться в проделанной работе, знать:

- Цели и задачи предпроектного обследования.
- Методологию функционального моделирования.
- Как определять границы моделирования
- Как определять точку зрения при построении модели.
- Принцип декомпозиции.

Студент должен уметь изменить модель (добавить новые или удалить элементы) по требованию преподавателя.

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Илья Владимирович
Цель работы:
Действителен: с 19.08.2022 по 19.08.2023

Лабораторная работа №3

«Построение моделей в BPWin. Построение IDEF3, DFD диаграмм.»

Освоение принципов построения IDEF3, DFD диаграмм. Приобретение практических навыков в создании IDEF3, DFD диаграмм.

Создание диаграммы IDEF3

1) Перейдите на диаграмму A2 и декомпозируйте работу «Сборка настольных компьютеров». В диалоге Activity Box Count (рис. 1) установите число работ 4 и нотацию IDEF3.

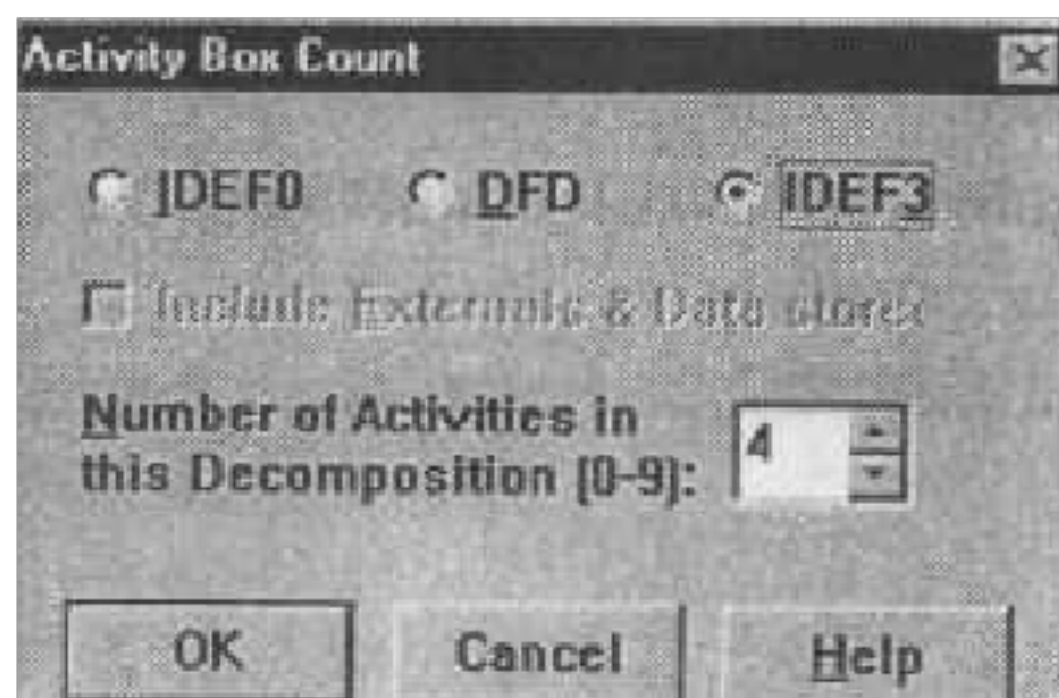


Рис. 1. Выбор нотации IDEF3 в диалоге Activity Box Count

Возникает диаграмма IDEF3, содержащая работы (UOW). Правой кнопкой мыши щелкните по работе, выберите в контекстном меню Name и внесите имя работы «Подготовка компонентов». Затем во вкладке Definition внесите определение "Подготавливаются все компоненты компьютера согласно спецификации заказа".

2) Во вкладке UOW внесите свойства работы (табл. 1).

Таблица 1. Свойства UOW

<i>Objects</i>	Компоненты: винчестеры, корпуса, материнские платы, видеокарты, звуковые карты, дисководы CD-ROM и флоппи, модемы, программное обеспечение
<i>Facts</i>	Доступные операционные системы: Windows 98, Windows NT, Windows 2000
<i>Constrains</i>	Установка модема требует установки дополнительного программного обеспечения

3) Внесите в диаграмму еще 3 работы (кнопка). Внесите имена работ:

- Установка материнской платы и винчестера;
- Установка модема;
- Установка дисководы CD-ROM;
- Установка флоппи- дисководы;
- Установка операционной системы;
- Установка дополнительного программного обеспечения.

4) С помощью кнопки – палитры инструментов создайте объект ссылки. Внесите имя объекта внешней ссылки "Компоненты". Свяжите стрелкой объект ссылки и работу "Подготовка компонентов".

5) Свяжите стрелкой работы "Подготовка компонентов" (выход) и "Установка материнской платы и винчестера". Измените стиль стрелки на Object Flow.

В IDEF3 имя стрелки может отсутствовать, хотя VPwin показывает отсутствие имени как ошибку. Результат показан на рис. 2.



Рис. 2. Результат создания UOW и объекта ссылки

6) С помощью кнопки  на палитре инструментов внесите два перекрестка типа "асинхронное или" и свяжите работы с перекрестками, как показано на рис. 3.

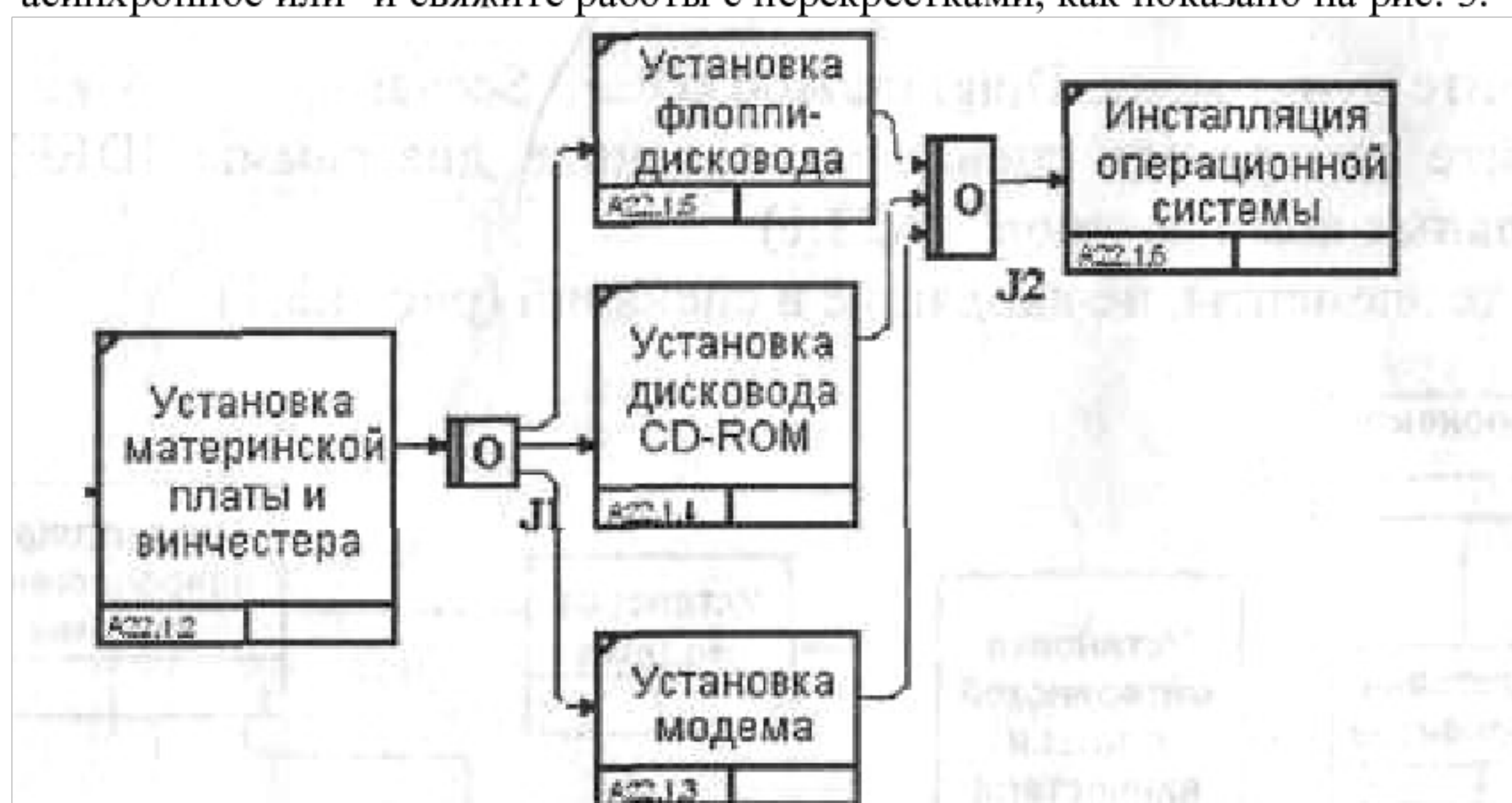


Рис. 3. Диаграмма IDEF3 после создания перекрестков

7) Правой кнопкой щелкните по перекрестку для разветвления (fan-out), выберите Name и внесите имя "Компоненты, требуемые в спецификации заказа".

Создайте два перекрестка типа исключающего "ИЛИ" и свяжите работы, как показано на рис. 4.



Рис. 4. Результат создания диаграммы IDEF3

Создание сценария

- 1) Выберите пункт меню Diagram/Add IDEF3 Scenario. Создайте диаграмму сценария на основе диаграммы IDEF3 "Сборка настольных компьютеров" (A22.1).
- 2) Удалите элементы, не входящие в сценарий (рис. 5).



Рис. 5. Результат создания сценария

Дополнение моделей процессов диаграммой DFD

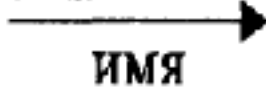
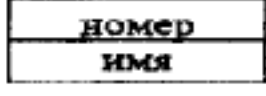
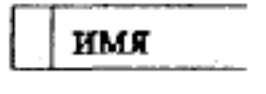
Эти диаграммы представляют сеть связанных между собой работ. Их удобно использовать для описания документооборота и обработки информации.

DFD описывает:

1. функции обработки информации (работы);

2. документы (стрелки, arrow), объекты, сотрудников или отделы, которые участвуют в обработке информации;
3. внешние ссылки (external reference), которые обеспечивают интерфейс с внешними объектами, находящимися за границами моделируемой системы;
4. таблицы для хранения документов (хранилища данных, data store).

Для построения диаграмм DFD в BPWin используется нотация Гейна – Сарсона:

Компонент	Обозначение
Поток данных	 ИМЯ
Процесс	 ИМЯ
Хранилище	 ИМЯ
Внешняя сущность	 ИМЯ

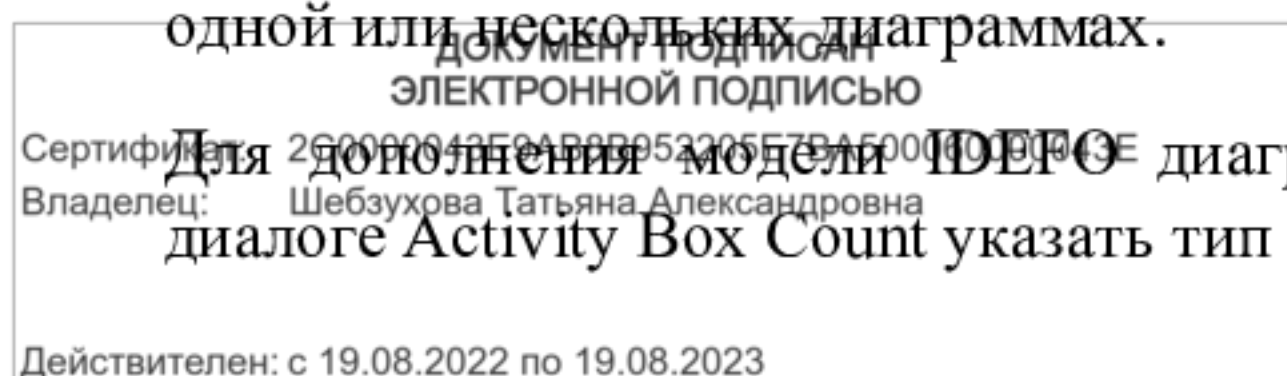
Потоки данных являются механизмами, используемыми для моделирования передачи информации (или физических компонентов) из одной части системы в другую. Потоки изображаются на диаграмме именованными стрелками, ориентация которых указывает направление движения информации. Стрелки могут подходить к любой грани прямоугольника работы и могут быть двунаправленными для описания взаимодействия типа «команда-ответ».

Назначение процесса состоит в продуцировании выходных потоков из входных в соответствии с действием, задаваемым именем процесса. Каждый процесс должен иметь уникальный номер для ссылок на него внутри диаграммы. Этот номер может использоваться совместно с номером диаграммы для получения уникального индекса процесса во всей модели.

Хранилище данных позволяет на определенных участках определять данные, которые будут сохраняться в памяти между процессами. Фактически хранилище представляет «срезы» потоков данных во времени. Информация, которую оно содержит, может использоваться в любое время после ее определения, при этом данные могут выбираться в любом порядке. Имя хранилища должно идентифицировать его содержимое. В случае, когда поток данных входит в хранилище или выходит из него и его структура соответствует структуре хранилища, он должен иметь то же самое имя, которое не имеет необходимости отражать на диаграмме.

Внешняя сущность представляет сущность вне контекста системы, являющуюся источником или приемником данных системы. Предполагается, что объекты, представленные такими узлами, не должны участвовать ни в какой обработке. Внешние сущности изображаются в виде прямоугольника с тенью и обычно располагаются по краям диаграммы. Одна внешняя сущность может быть использована многократно на одной или нескольких диаграммах.

Для дополнения модели IDEFO диаграммой DFD нужно в процессе декомпозиции в диалоге Activity Box Count указать тип диаграммы DFD.



Контрольные вопросы

При защите лабораторной работы студент должен ориентироваться в проделанной работе, знать:

- Методологию функционального моделирования с использованием IDEF3 и DFD.
- Как создать диаграммы IDEF3
- Как создать сценарии
- Как дополнить модели процессов диаграммой DFD.
- Принцип декомпозиции.

Студент должен уметь изменить модель (добавить новые или удалить элементы) по требованию преподавателя.

Лабораторная работа № 4

«Построение моделей в ERWin.»

Цель работы:

Приобретение практических навыков построения логической модели данных выбранной предметной области в нотации IDEF1X в ERWin.

IDEF1X основан на подходе Чена и позволяет построить модель данных, эквивалентную реляционной модели в третьей нормальной форме. Нотация Чена и сам процесс построения диаграмм сущность-связь изучалась в курсе "Организация баз данных и знаний", поэтому здесь мы рассмотрим только отличия IDEF1X от нотации Чена.

Сущность (Entity) - реальный либо воображаемый объект, имеющий существенное значение для рассматриваемой предметной области. Каждая сущность должна иметь наименование, выраженное существительным в единственном числе. Каждая сущность должна обладать уникальным идентификатором. Каждый экземпляр сущности должен однозначно идентифицироваться и отличаться от всех других экземпляров данного типа сущности.

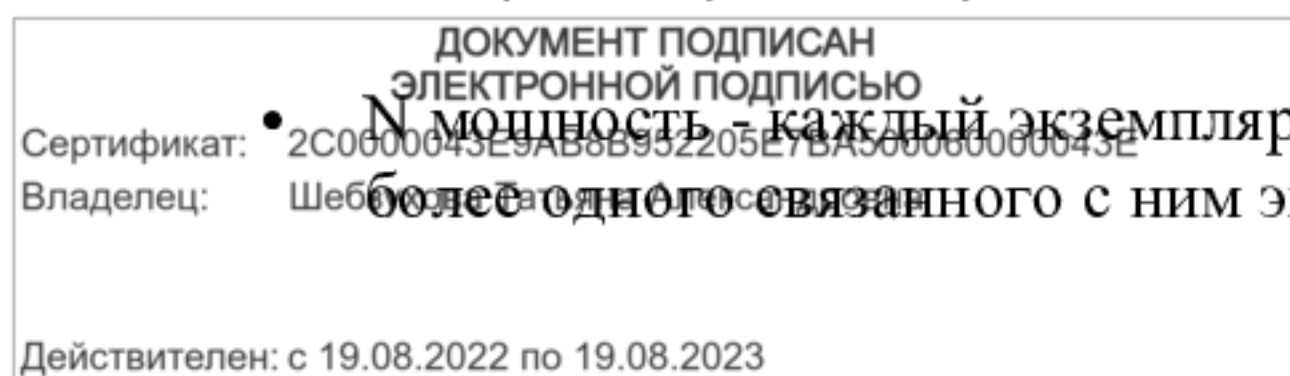
Атрибут (Attribute) - любая характеристика сущности, значимая для рассматриваемой предметной области и предназначенная для квалификации, идентификации, классификации, количественной характеристики или выражения состояния сущности. Наименование атрибута должно быть выражено существительным в единственном числе.

Связь (Relationship) - поименованная ассоциация между двумя сущностями, значимая для рассматриваемой предметной области.

В методе IDEF1X все сущности делятся на зависимые и независимые от идентификаторов. Сущность является независимой от идентификаторов или просто независимой, если каждый экземпляр сущности может быть однозначно идентифицирован без определения его отношений с другими сущностями. Сущность называется зависимой от идентификаторов или просто зависимой, если однозначная идентификация экземпляра сущности зависит от его отношения к другой сущности. Независимая сущность изображается в виде обычного прямоугольника, зависимая - в виде прямоугольника с закругленными углами.

В IDEF1X существуют следующие виды мощностей связей:

- **0-мощность** - каждый экземпляр сущности-родителя может иметь ноль, один или более экземпляров сущности-потомка (по умолчанию);



- Р мощность - каждый экземпляр сущности-родителя должен иметь не менее одного связанного с ним экземпляра сущности-потомка;
- Z мощность - каждый экземпляр сущности-родителя должен иметь не более одного связанного с ним экземпляра сущности-потомка;
- конкретное число - каждый экземпляр сущности-родителя связан с некоторым фиксированным числом экземпляров сущности-потомка.

Связь изображается линией, проводимой между сущностью-родителем и сущностью-потомком, с точкой на конце линии у сущности-потомка. По умолчанию мощность связи принимается равной N. Если экземпляр сущности-потомка однозначно определяется своей связью с сущностью-родителем, то связь называется идентифицирующей, в противном случае — неидентифицирующей. Идентифицирующая связь изображается сплошной линией, неидентифицирующая - пунктирной линией.

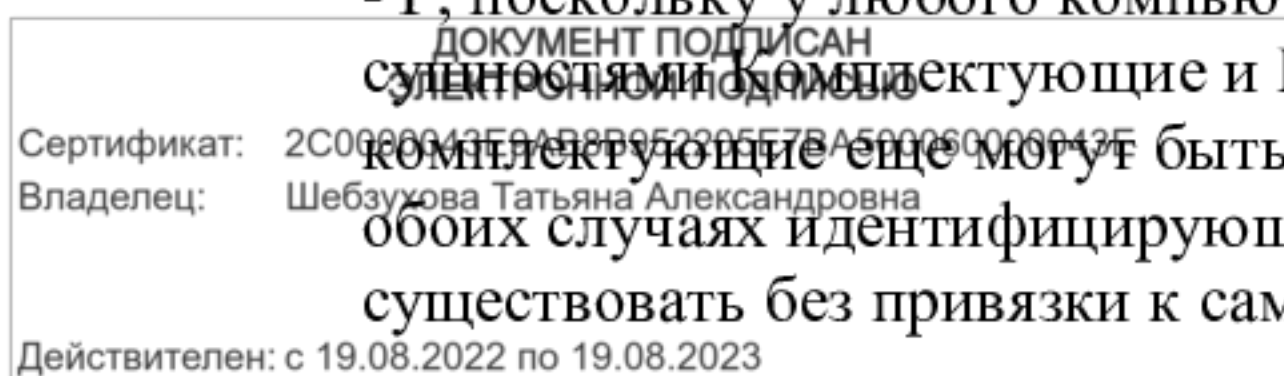
В ERwin'e при установлении идентифицирующей связи атрибуты первичного ключа родительской сущности автоматически переносятся в состав первичного ключа дочерней сущности. Эта операция называется миграцией атрибутов. В дочерней сущности новые атрибуты помечаются как внешний ключ (FK). При установке неидентифицирующей связи атрибуты первичного ключа родительской сущности мигрируют в состав неключевых полей дочерней сущности.

Построение логической модели данных предприятия по сборке и продаже компьютеров и ноутбуков. Построение модели данных начинается с выделения сущностей данной предметной области. В нашем случае были выделены следующие сущности:

- клиент - человек, который покупает компьютеры
- заказ - список компьютеров, которые покупает клиент
- компьютер
- комплектующие - то, из чего собирают компьютеры
- сотрудник - сотрудник предприятия, собирающий конкретный компьютер

Далее рассмотрим связи между сущностями:

- Клиент - Заказ. Один клиент может делать несколько заказов. При этом если данные о клиенте имеются в базе данных, то он сделал минимум один заказ. Поэтому мощность связи - Р. Связь идентифицирующая, т.к. заказ без клиента существовать не может;
- Заказ - Компьютер. В рамках одного заказа клиент может заказать несколько компьютеров, но как минимум заказ должен состоять из одного компьютера. Поэтому мощность связи - Р. Связь идентифицирующая, т.к. компьютер без заказа существовать не может;
- Компьютер - Комплектующие. В состав одного компьютера входит много различных комплектующих; один и тот же тип комплектующего может входить в состав разных компьютеров. Мощность связи - много ко многим. В IDEF1X такой тип связи отсутствует, поэтому вводим промежуточную (ассоциативную) сущность - Конфигурация. Мощность связи между сущностями Компьютер и Конфигурация - Р, поскольку у любого компьютера должна быть конфигурация, мощность между сущностями Комплектующие и Конфигурация - N, поскольку какие-то комплектующие еще могут быть не установлены ни в один компьютер. Связь в обоих случаях идентифицирующая, т.к. конфигурация компьютера не может существовать без привязки к самому компьютеру и к комплектующим;



- Комплектующие - Тип комплектующих. Поскольку перечень типов комплектующих, которые могут быть установлены в компьютер, ограничен, но используется очень часто, то мы приняли решение создать еще одну сущность - Тип комплектующих. Мощность связи - Р. Связь идентифицирующая;
- Компьютер - Сотрудник. Каждый компьютер собирается каким-то одним сотрудником. Какие-то сотрудники могут собирать множество компьютеров. Мощность связи - N. Тип связи - неидентифицирующая, поскольку экземпляр сущности Компьютер уже может существовать, но за ним еще может быть не закреплен ни один сотрудник. Именно из этих же соображений в свойствах этой связи мы выбрали переключатель "Nulls Allowed" (на диаграмме это отображается в виде незакрашенного ромбика со стороны сущности-родителя.

Итоговая диаграмма показана на рис. 1:

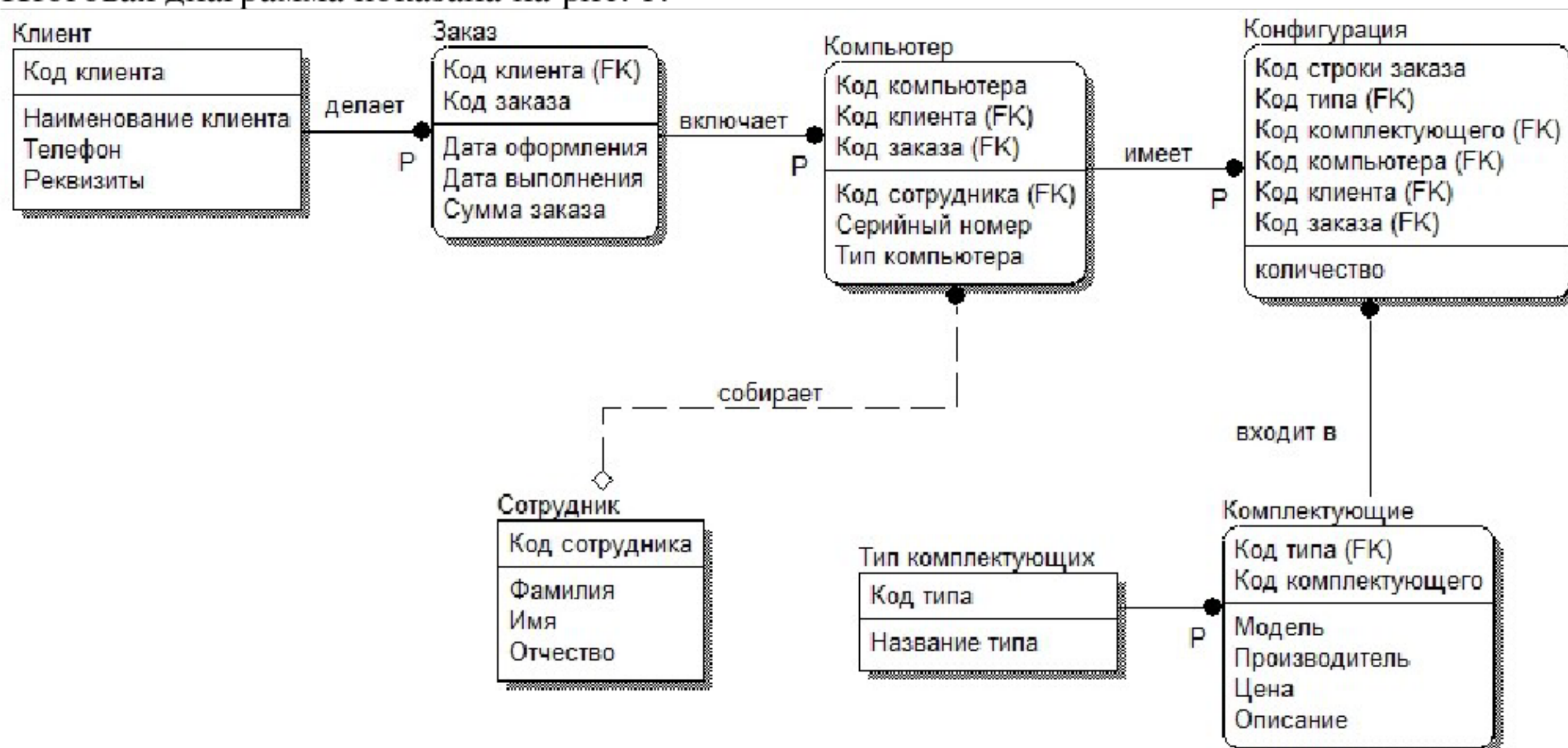


Рисунок 1. Логическая модель данных предприятия по сборке компьютеров и ноутбуков

Контрольные вопросы

1. Какие типы сущностей вы знаете?
2. Какие типы связей вы знаете?
3. Какие виды атрибутов вы знаете?
3. Какие виды мощностей связей вы знаете?

Лабораторная работа №5

«Создание и заполнение таблиц в Microsoft SQL Server 2012.»

Цель работы:

- Изучение основных конструкций структурированного языка запросов SQL.
- Изучения среды MS SQL Server Management Studio.
- Приобретение навыков проектирования структур данных.

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Базы данных составляют основу для построения информационных систем любого масштаба и предназначения. В теории баз данных одними из основных являются вопросы,

Действителен: с 19.08.2022 по 19.08.2023

связанные с анализом предметной области и моделированием структуры данных, управлением данными и их анализом.

Основой любой базы данных является реализованная в ней модель данных, представляющая собой множество структур данных, ограничений целостности и операций манипулирования данными. С помощью модели данных могут быть представлены объекты предметной области и существующие между ними связи.

Результатом лабораторной работы будет создание реляционной базы данных на основе MS SQL Server 2005.

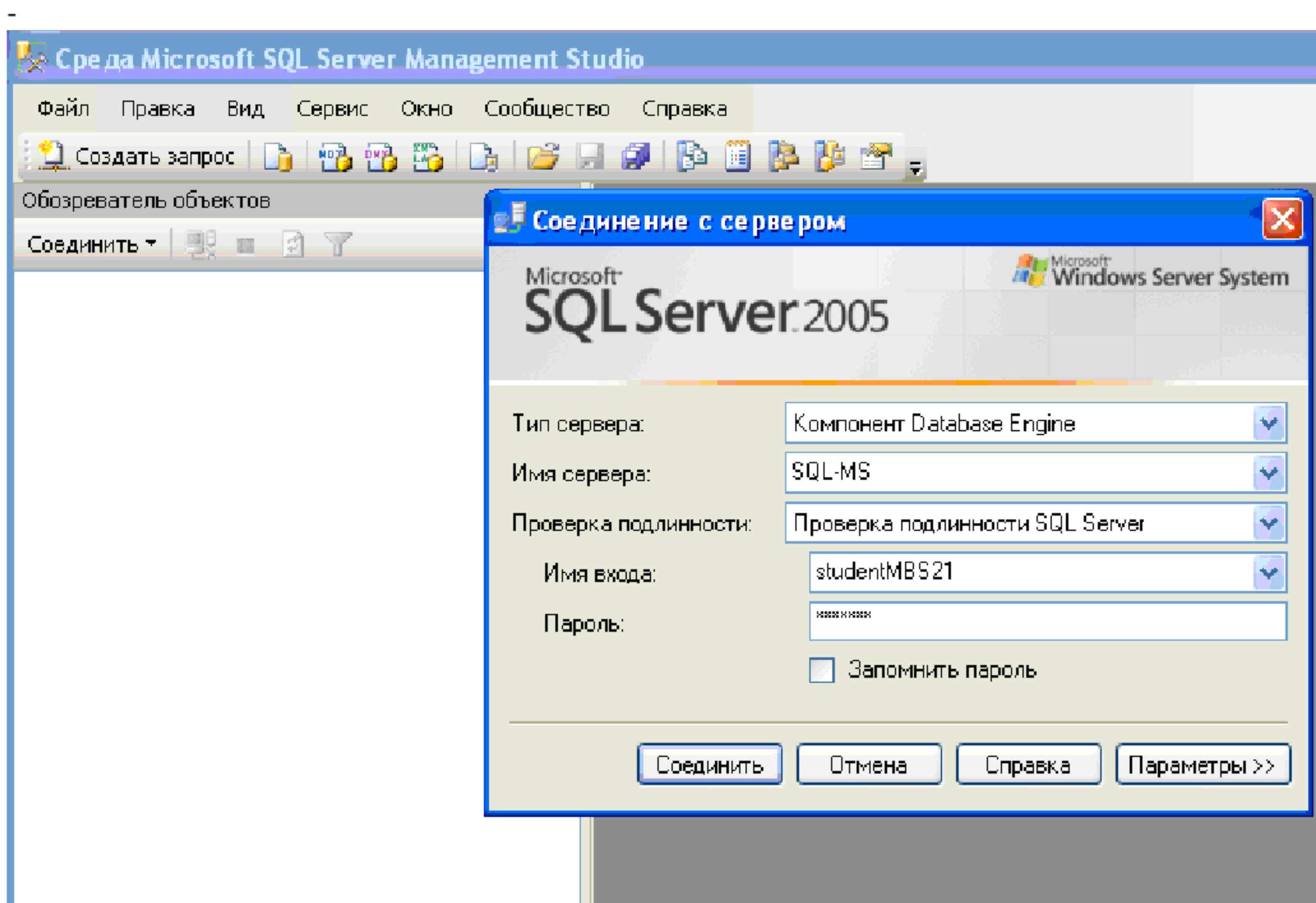
В реляционной базе данных данные представлены в виде собрания таблиц. Таблица состоит из определенного числа столбцов (полей) и произвольного числа строк (записей). Планируемая база данных будет представлять собой информационное хранилище данных об успеваемости студентов и состоять из следующих таблиц:

- **Speciality** (специальность)
- **Course** (курс)
- **Group** (группа)
- **Discipline** (дисциплина)
- **Account** (тип отчетности)
- **Mark** (отметка)
- **Status** (академический статус студента)
- **Position** (должность)
- **People** (люди)
- **Student** (студент)
- **Teacher** (преподаватель)
- **SemesterResults** (результаты сессии, семестра)
- Структура данных таблиц приведена в Приложении.

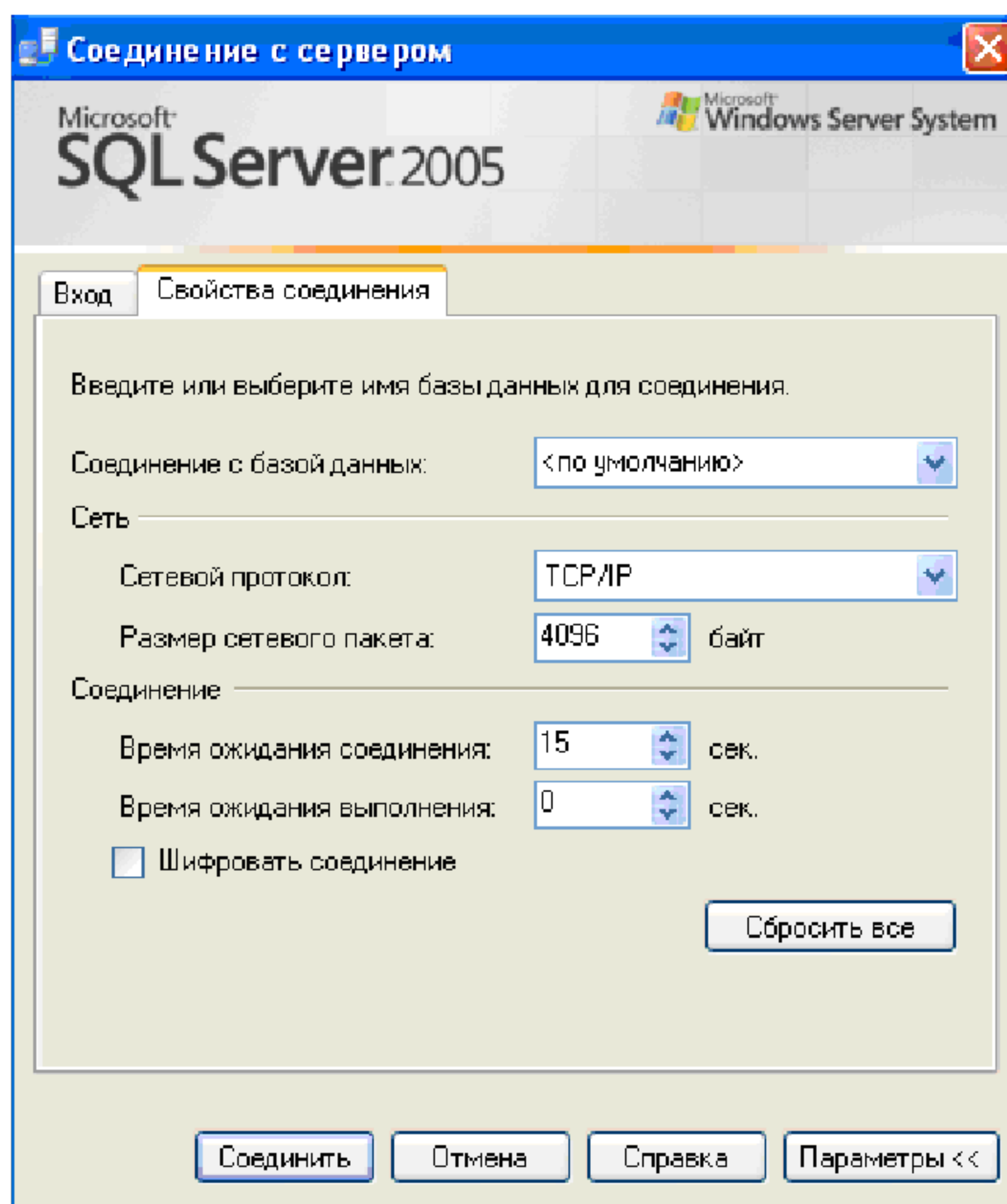
Начало работы в Microsoft SQL Server Management Studio

Для создания баз данных используем среду Microsoft SQL Server Management Studio. На запрос соединения с сервером выбираем (рис. 1):

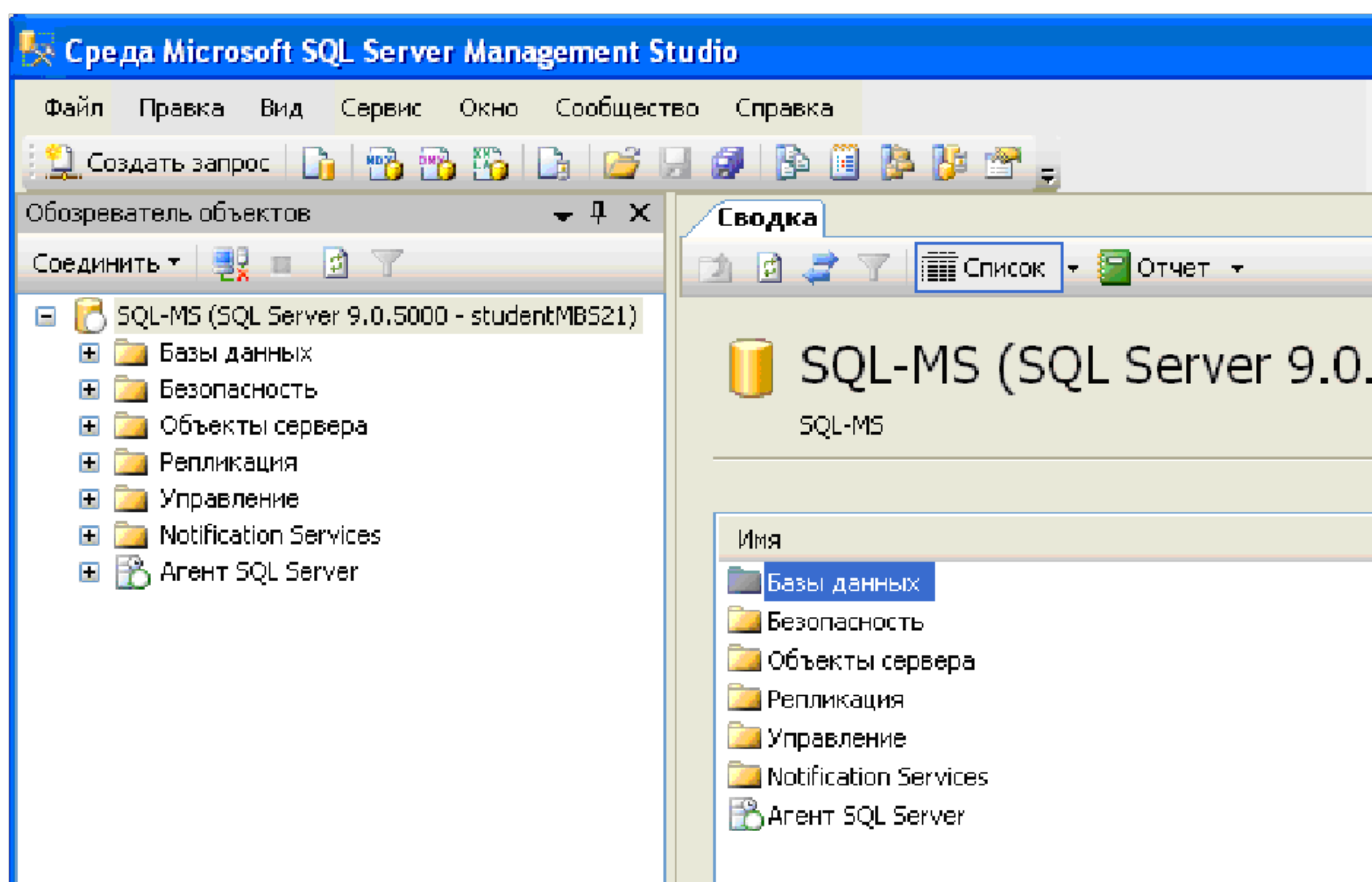
- *Тип сервера:* **Компонент Database Engine**
- *Имя сервера:* **SQL-MS.**
- Под таким именем в домене fizmat.vspu.ru. доступна машина, на которой установлены серверные компоненты MS SQL Server 2005. Можно попробовать выбрать сервер из выпадающего списка серверов. Можно также обратиться к этой машине по IP-адресу 192.168.10.152 из локальной сети.
- *Проверка подлинности:* **Проверка подлинности SQL Server.**
- Такая настройка позволяет создавать пользователей данного экземпляра SQL Server независимо от компьютера, с которого производится вход.
- *Имя входа:* **studentMBS21.**
- *Пароль:* **student.**



- Рисунок 1. Окно входа в Microsoft SQL Server Management Studio 2005
- *Примечание.* Пользователь studentMBS21 обладает большими полномочиями на этом сервере, поэтому пользоваться им надо очень аккуратно. Под этим пользователем мы создадим базу данных, а заполнять её и производить поиск по ней мы будем под другими пользователями. Предпочтительнее всего использовать свою учетную запись в домене fizmat.vspu.ru. В этом случае надо выбирать проверку подлинности Windows.
- Теперь нажимаем кнопку «Параметры» и выбираем (рис. 2):
- Соединение с базой данных → Обзор сервера... → Пользовательские базы данных → trial_base.
- Сетевой протокол → TCP/IP
- Нажимаем кнопку «Соединить».



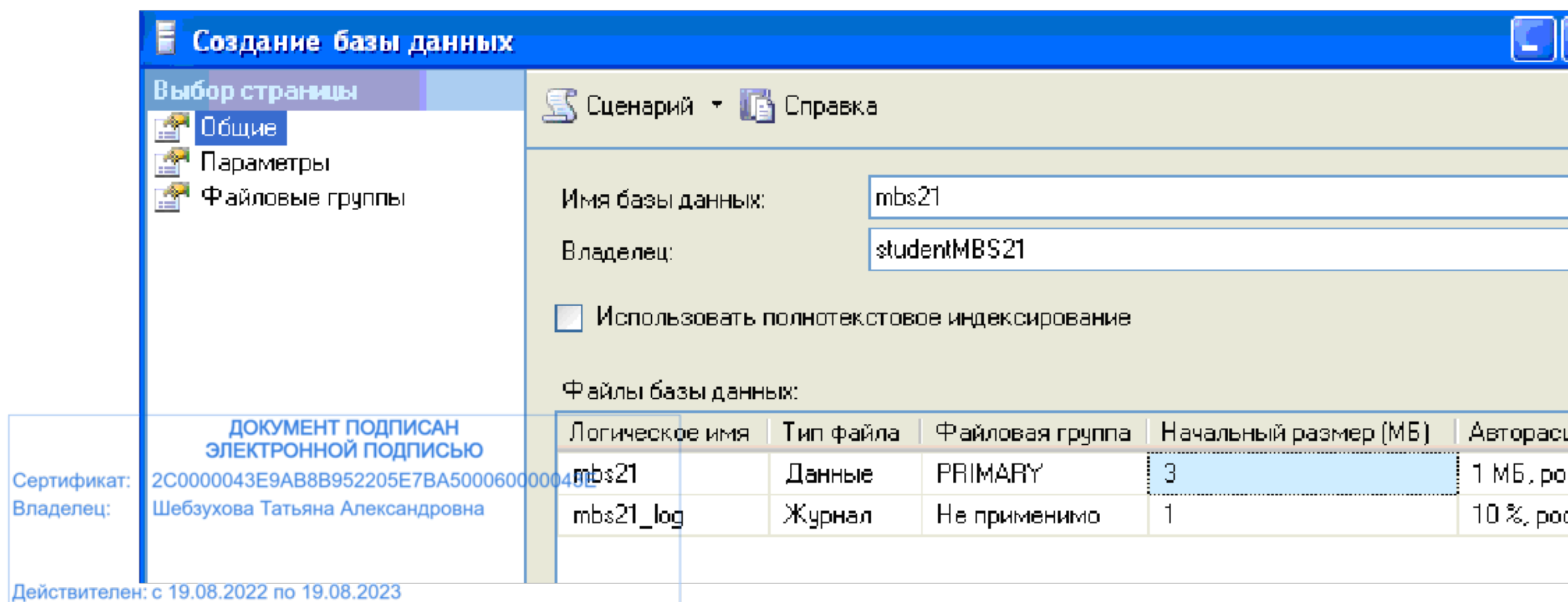
-
- Рисунок 2. Окно входа в Microsoft SQL Server Management Studio 2005 (вкладка Параметры)
- *Примечание.* База данных trial_base является базой данной по умолчанию для пользователя studentMBS21, она была создана при регистрации этого пользователя. В случае, когда права доступа пользователя не ограничены (как в рассматриваемом случае), вкладку Параметры можно не открывать. Если же пользователь имеет доступ только к определенным базам данных, при подключении к серверу нужно одну из этих баз указывать.
- После успешного соединения с базой данных на экране видим следующую картинку (рис. 3):



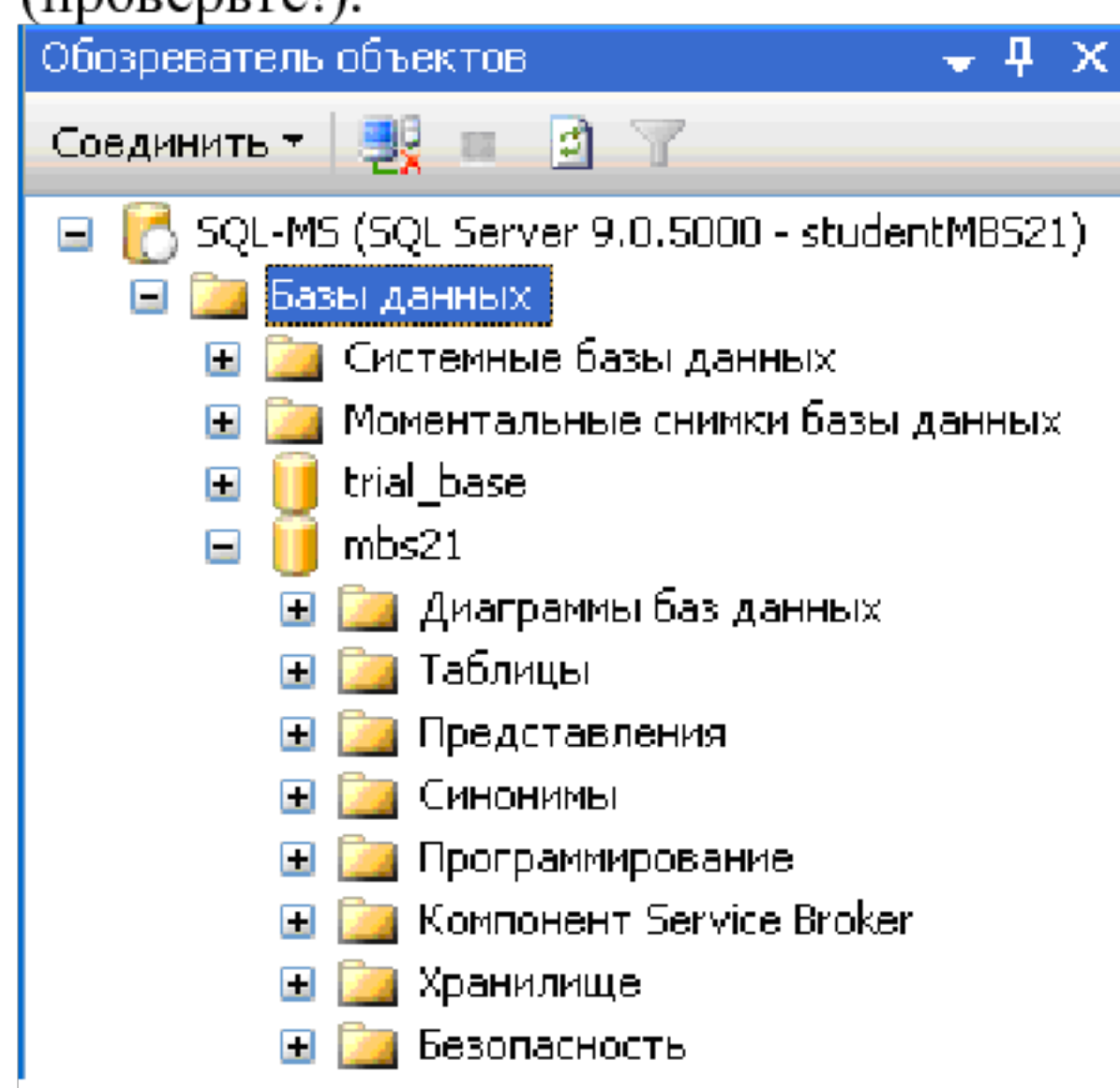
- Рисунок 3. Подключение к SQL - серверу установлено
- Среда MS SQL Management Studio предоставляет удобный инструмент для создания, редактирования, заполнения баз данных. Но настоящие профессионалы в своей работе редко пользуются этой средой, а для выполнения своих задач используют SQL-запросы. Мы будем пользоваться, когда это удобно и наглядно, графическим режимом, но основной упор будем делать на освоении базы языка SQL.

Создание базы данных в среде Microsoft SQL Server Management Studio

- В разделе «Базы данных» правой кнопкой выбираем «Создать базу данных...» (рис. 4). Назовем базу данных по индексу группы – mbs21. Владелец базы данных назначим пользователя, под именем которого был произведен вход – studentMBS21. В разделе «Параметры» выбираем тип сортировки Cyrillic_General_BIN (для примера), нажимаем OK.



- Рисунок 4. Создание базы данных
- В разделе «Базы данных» Обозревателя объектов появилась вновь созданная mbs21 (проверьте!):



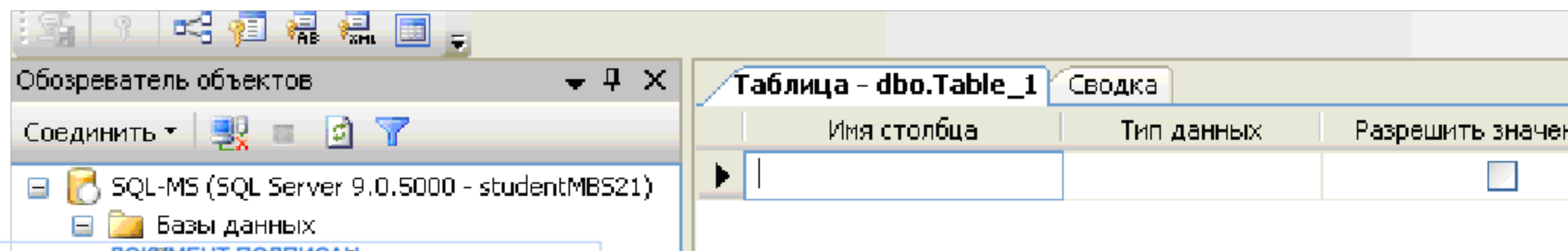
- Рисунок 5. Обозреватель объектов

Создание таблиц базы данных в среде Microsoft SQL Server Management Studio

- Начнем с создания таблицы Speciality. Структура таблицы приведена ниже:

Имя поля (столбца)	Содержание	Тип данных	Возможность содержать NULL
Num	Первичный ключ	int	нет
Name	Название специальности	varchar(60)	нет

- В реляционных базах данных первичный ключ используется как уникальный идентификатор записи. Это поле является обязательным, оно используется для связи таблиц по внешним ключам (примеры такого связывания будут рассмотрены далее). Первичный ключ должен иметь целочисленный тип (в данном случае - *int*). Во втором поле будет храниться название специальности - некоторая строка, поэтому мы выбираем для этого поля тип *varchar(60)*. Число в скобках означает максимальное число символов в строке. Детальную информацию об этих типах можно посмотреть в справке.
- Простейшим образом можно создавать таблицы средствами MS SQL Server Management Studio (правая кнопка мыши на заголовке «Таблицы» > Создать таблицу.). Получаем следующее:



- Рисунок 6. Создание таблицы

Сертификат: 2C0690043E9AB8B952205E7BA500060000042E
Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

- Вводим имя первого столбца Num (первичный ключ – в том столбце хранится номер записи), выбираем из выпадающего списка тип данных int. Первичный ключ не может быть пустым, поэтому и оставляем неотмеченным поле «Разрешить

значения null». Затем аналогичным образом вводим имя второго столбца, задаем тип, запрещаем полю иметь значение null. Таблица принимает следующий вид:

Таблица - dbo.Table_1*			
Сводка			
	Имя столбца	Тип данных	Разрешить значения null
	Num	int	☐
▶	Name	varchar(60)	☐
			☐

- Рисунок 7.
- Теперь необходимо указать, что поле *Num* будет являться первичным ключом. Правой кнопкой мыши щелкаем по этому полю и выбираем «Задать первичный ключ»:

Таблица - dbo.Table_1*			
Сводка			
	Имя столбца	Тип данных	Разрешить значения null
▶	Num	int	☐
	Name	varchar(60)	☐
			☐

- Задать первичный ключ
- Вставить столбец
- Удалить столбец
- Отношения

- Рисунок 8.
- Сохраняем таблицу под именем *Speciality* (после этого таблица должна появиться в обозревателе объектов). Теперь можно перейти к заполнению этой таблицы (для этого нужно в обозревателе объектов выбрать эту таблицу и в контекстном меню нажать «Открыть таблицу»):

Таблица - dbo.Speciality		
Таблица - dbo.Speciality		
Сводка		
	Num	Name
	1	Математика
	2	Информатика
	3	Физика
▶*	Null	Null

- Рисунок 9.
- При заполнении вы обнаружите, что каждый раз приходится вводить не только полезную информацию (название специальности), но и номер записи. Чтобы вводить номер записи автоматически, нужно задать спецификацию идентифицирующего столбца. Для этого необходимо в свойствах столбца указать, что данный столбец является идентифицирующим (рис. 10):

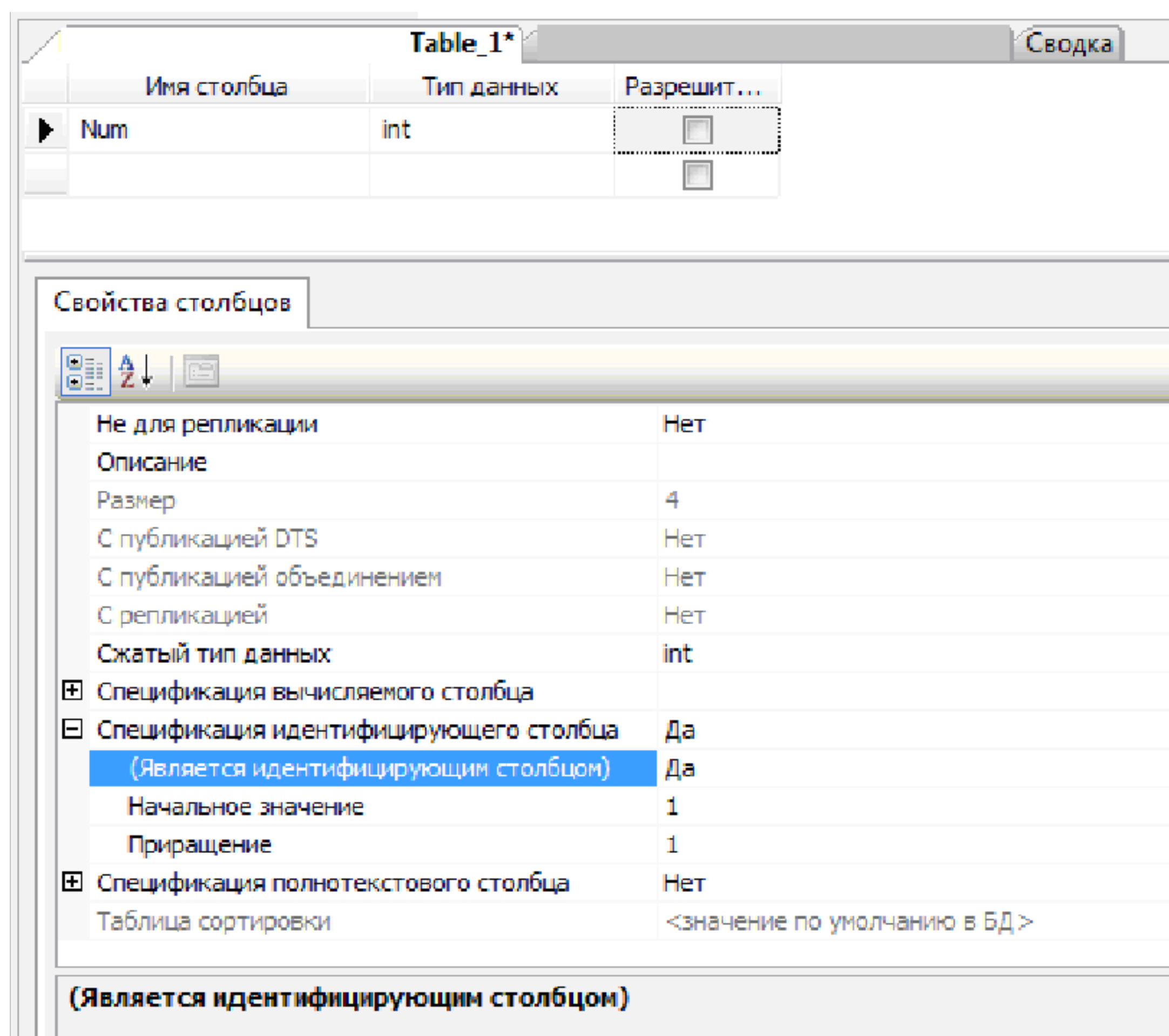


Рисунок 10. Определение свойств идентифицирующего столбца

Создание таблиц базы данных с помощью SQL-запроса

- Создание таблиц в графическом режиме, безусловно, удобно, однако не универсально. При использовании других средств разработки баз данных (например, IBM DB2) придется привыкать к новым приемам работы. Использование конструкций языка SQL позволяет работать с базами данных, исходя из единого подхода, в любой среде управления базами данных.
- Выберите на панели инструментов «Создать запрос»:

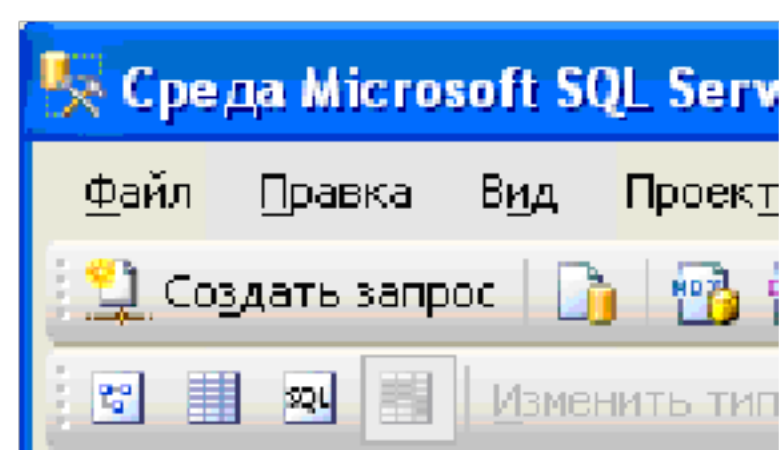


Рисунок 11.

- Создадим новую базу данных запросом. Напишем
- CREATE DATABASE mbs21_query
- и нажмем F5. В обозревателе объектов должна появиться новая база (если сразу не появилась, то надо выделить мышью раздел «Базы данных» и в контекстном меню выбрать «Обновить»).

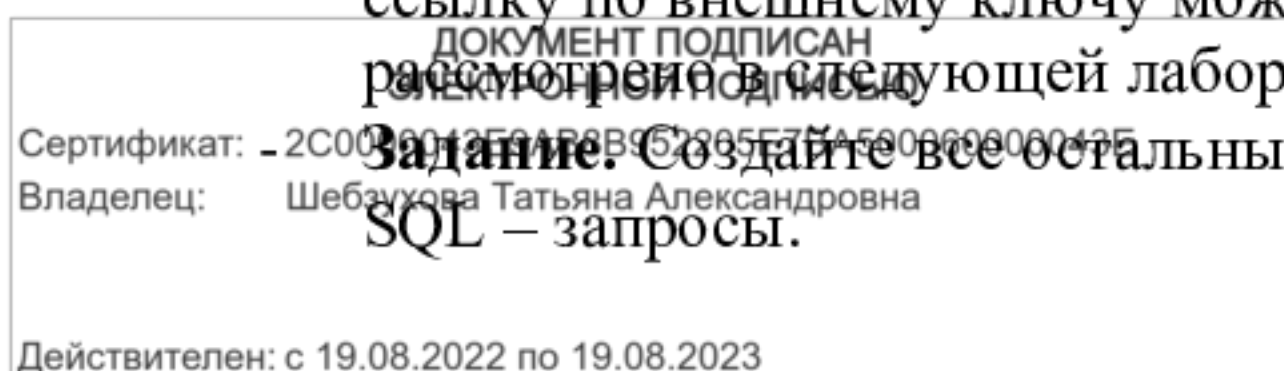
Документ подписан
электронной подписью
Сертификат: 2C00101A52256100
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

- Теперь создадим таблицу **Speciality**. Упрощенный синтаксис создания таблиц следующий:
- `CREATE TABLE <имя таблицы> (`
- `<имя столбца 1> <тип данных> [NOT NULL] [DEFAULT <значение по умолчанию>],`
- `<имя столбца 2> <тип данных> [NOT NULL] [DEFAULT <значение по умолчанию>],`
- `...`
- `)`
- Введем новый запрос:
- `/* создание таблицы Специальность */`
- `USE mbs21_query` -- определяем базу
- `данных, в которую входит таблица`
- `CREATE TABLE Speciality(`
- `Num INT IDENTITY(1,1) PRIMARY KEY NOT NULL, -- первичный ключ`
- `NameSpec VARCHAR(60)` -- название
- `специальности`
- `)`
- В обозревателе объектов видим, что таблица действительно создана. Файл с SQL-запросом сохраняем в своей папке (в конце работы необходимо показать запросы, которые были выполнены, преподавателю). Слово `IDENTITY(1,1)` добавлено, чтобы поле первичного ключа `Num` автоматически нумеровалось начиная с единицы (фактически, эта конструкция определяет спецификацию идентифицирующего столбца).
- Таким же образом необходимо создать остальные таблицы. Рассмотрим таблицу **Course**.
- Таблица **Course** (курс)

Имя поля (столбца)	Содержание	Тип данных	Возможность содержать NULL
Num	Первичный ключ	int	нет
Name	Название специальности	varchar(60)	нет
YearEntry	Год поступления	int	нет
YearFinal	Год выпуска	int	да
Speciality	Специальность (внешний ключ ссылается на первичный ключ таблицы Speciality)	int	нет

- Эта таблица содержит поле *Speciality*, которое ссылается на первичный ключ таблицы **Speciality**. Чтобы создать такую таблицу, необходимо выполнить запрос:
- `/* создание таблицы Курс */`
- `USE mbs21_query` -- определяем базу
- `данных, в которую входит таблица`
- `CREATE TABLE Course(`
- `Num INT IDENTITY(1,1) PRIMARY KEY NOT NULL, -- первичный ключ`
- `YearEntry INT NOT NULL, -- год поступления`
- `YearFinal INT, -- год окончания`
- `Speciality INT FOREIGN KEY REFERENCES Speciality(Num) --`
- `специальность,`
- `-- ссылка по внешнему ключу на поле Num таблицы Speciality`
- `)`
- **Примечание.** Ссылку можно создать только на существующую таблицу. Задать ссылку по внешнему ключу можно и после создания таблицы (подробно будет рассмотрено в следующей лабораторной работе).

Задание. Создайте все остальные таблицы, указанные в Приложении, используя SQL – запросы.



Контрольные вопросы

1. Каким образом можно получить доступ к MS SQL Server?
2. С помощью каких средств можно создать таблицу для MS SQL Server?
3. Что такое первичный ключ?
4. Каким образом можно создать автоматическую нумерацию строк таблицы?
5. Что означают Not Null?

Лабораторная работа № 6

«Создание запросов и фильтров в Microsoft SQL Server 2012.»

Цель работы:

Приобрести практические навыки в создании запросов и фильтров в Microsoft SQL Server 2012.»

Цель работы:

- научиться создавать запросы и фильтры.

Перейдем к созданию статических запросов. В обозревателе объектов "Microsoft SQL Server 2008" все запросы БД находятся в папке **"Views"** ([рис. 8.1](#)).

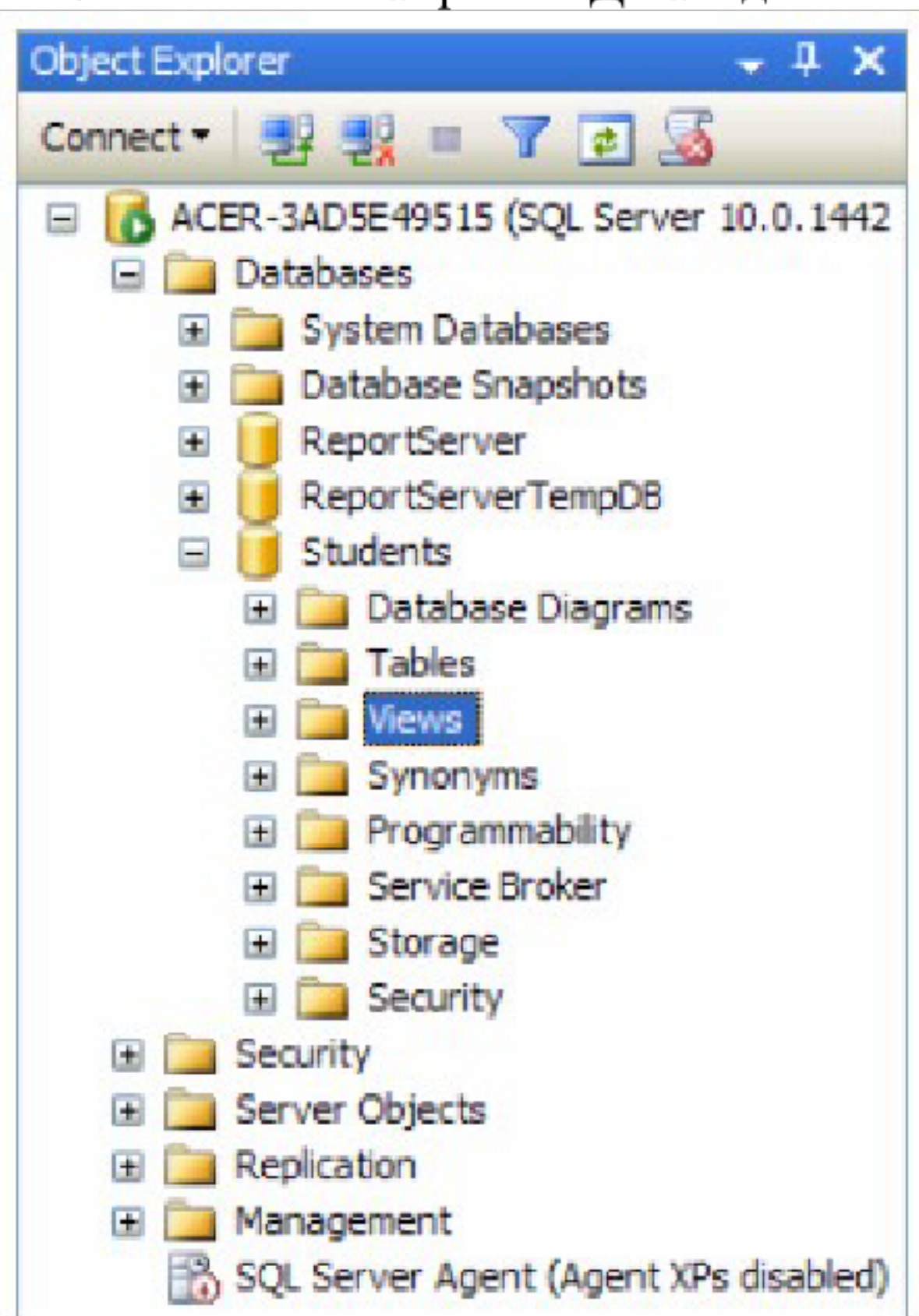


Рис. 8.1.

Создадим запрос **"Запрос Студенты+Специальности"**, связывающий таблицы **"Студенты"** и **"Специальности"** по полю связи **"Код специальности"**. Для создания нового запроса необходимо в обозревателе объектов в БД **"Students"** щелкнуть **ПКМ** по папке **"Views"**, затем в появившемся меню выбрать пункт **"New View"**. Появится окно **"Add Table"** (Добавить таблицу), предназначенное для выбора таблиц и запросов, участвующих в новом запросе ([рис. 8.2](#)).

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

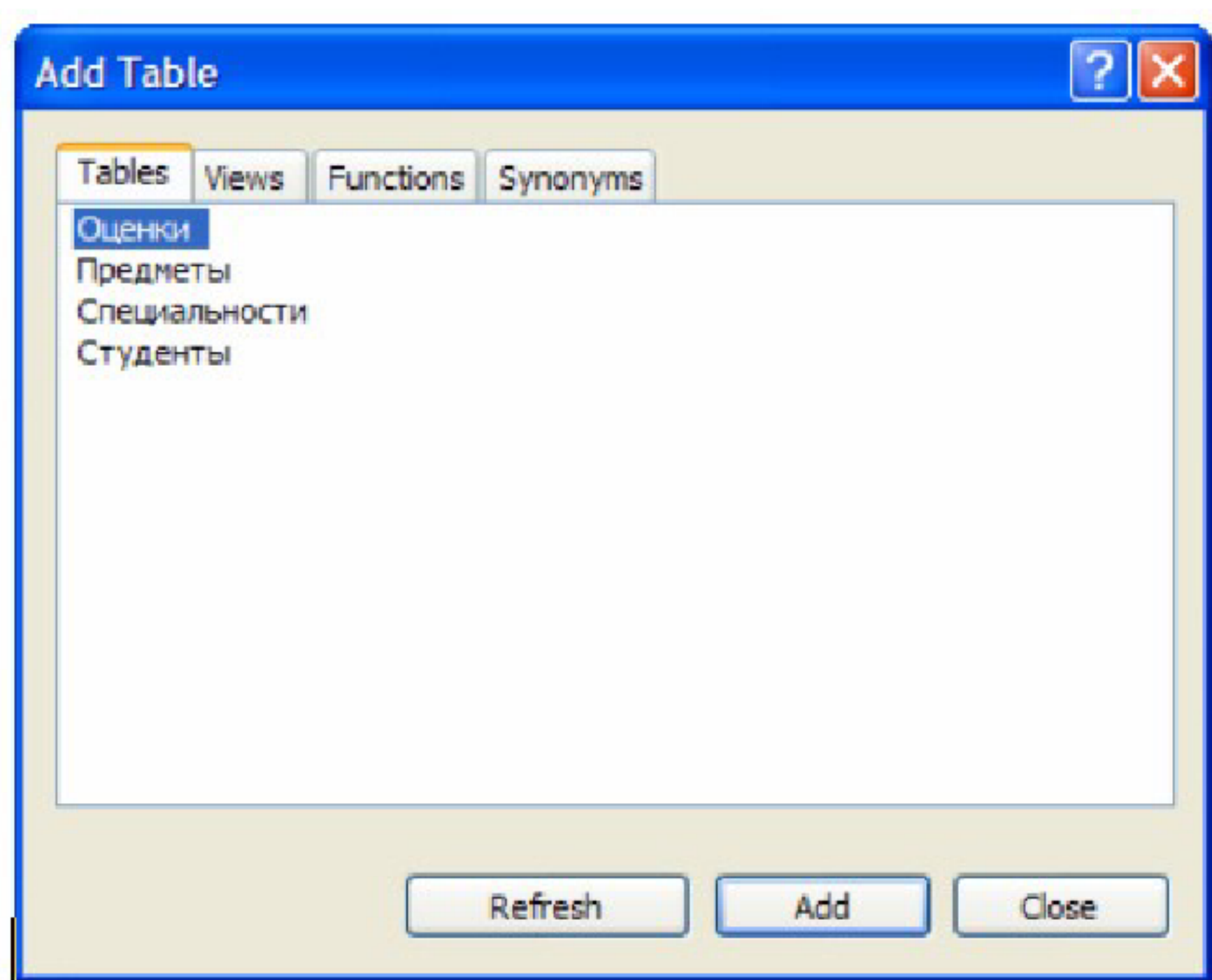


Рис. 8.2.

Добавим в новый запрос таблицы "Студенты" и "Специальности". Для этого в окне "Add Table" выделите таблицу "Студенты" и нажмите кнопку "Add" (Добавить). Аналогично добавьте таблицу "Специальности". После добавления таблиц участвующих в запросе закройте окно "Add Table" нажав кнопку "Close" (Заккрыть). Появится окно конструктора запросов ([рис. 8.3](#)).

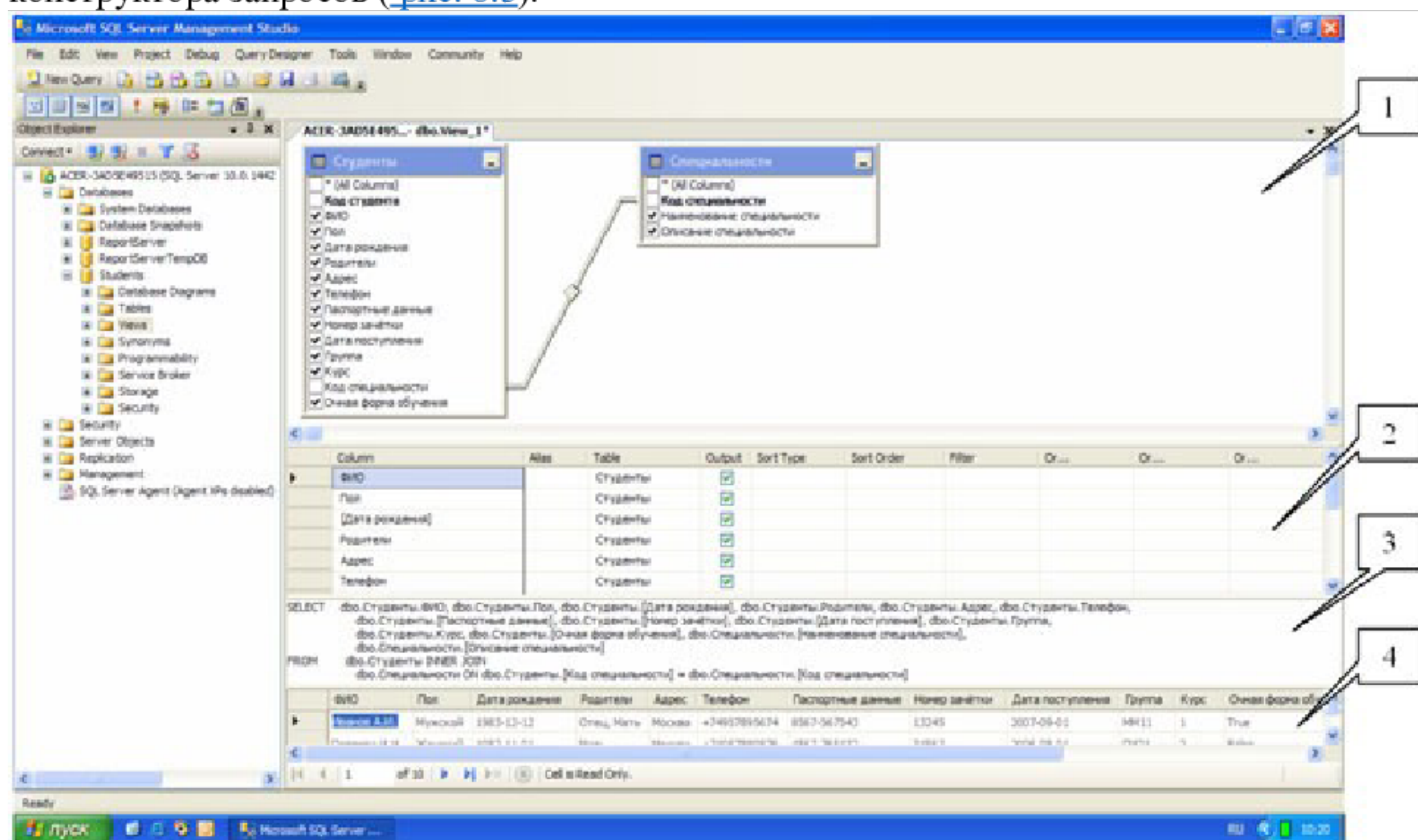
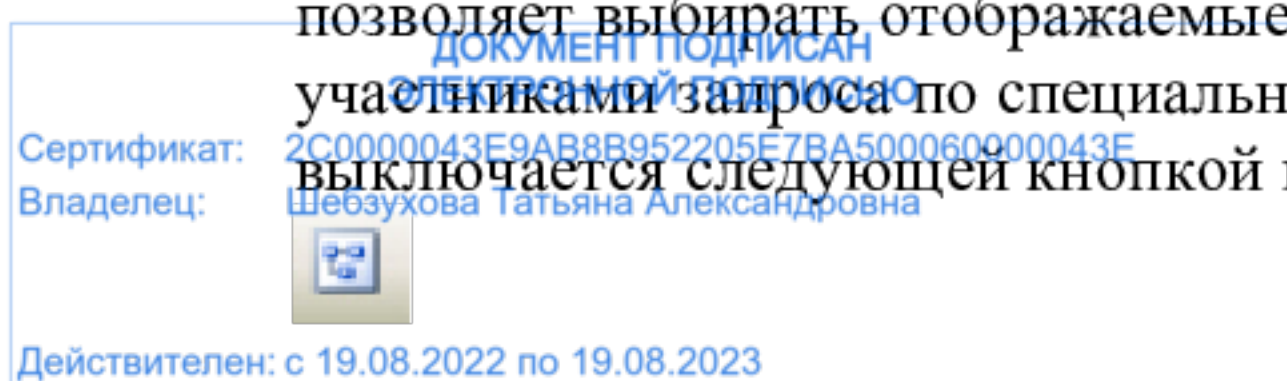


Рис. 8.3.

Замечание: Окно конструктора запросов состоит из следующих панелей:

1. **Схема данных** - отображает поля таблиц и запросов, участвующих в запросе, позволяет выбирать отображаемые поля, позволяет устанавливать связи между участниками запроса по специальным полям связи. Эта панель включается и выключается следующей кнопкой на панели инструментов



2. **Таблица отображаемых полей** - показывает отображаемые поля (столбец **"Column"**), позволяет задавать им псевдонимы (столбец **"Alias"**), позволяет устанавливать тип сортировки записей по одному или нескольким полям (столбец **"Sort Type"**), позволяет задавать порядок сортировки (столбец **"Sort Order"**), позволяет задавать условия отбора записей в фильтрах (столбцы **"Filter"** и **"Or..."**). Также эта таблица позволяет менять порядок отображения полей в запросе. Эта панель включается и выключается следующей кнопкой на панели инструментов



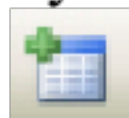
3. **Код SQL** - код создаваемого запроса на языке T-SQL. Эта панель включается и выключается следующей кнопкой на панели инструментов



4. **Результат** - показывает результат запроса после его выполнения. Эта панель включается и выключается следующей кнопкой на панели инструментов



Замечание: Если необходимо снова отобразить окно **"Add Table"** для добавления новых таблиц или запросов, то для этого на панели инструментов **"Microsoft SQL Server 2008"** нужно нажать кнопку



Замечание: Если необходимо удалить таблицу или запрос из схемы данных, то для этого нужно щелкнуть **ПКМ** и в появившемся меню выбрать пункт **"Remove"** (Удалить). Теперь перейдем к связыванию таблиц **"Студенты"** и **"Специальности"** по полям связи **"Код специальности"**. Чтобы создать связь необходимо в схеме данных перетащить мышью поле **"Код специальности"** таблицы **"Специальности"** на такое же поле таблицы **"Студенты"**. Связь отобразится в виде ломаной линии соединяющей эти два поля связи ([рис. 8.3](#)).

Замечание: Если необходимо удалить связь, то для этого необходимо щелкнуть по ней **ПКМ** и в появившемся меню выбрать пункт **"Remove"**.

Замечание: После связывания таблиц (а также при любых изменениях в запросе) в области кода T-SQL будет отображаться T-SQL код редактируемого запроса.

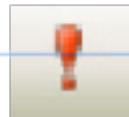
Теперь определим поля, отображаемые при выполнении запроса. Отображаемые поля обозначаются галочкой (слева от имени поля) на схеме данных, а также отображаются в таблице отображаемых полей. Чтобы сделать поле отображаемым при выполнении запроса необходимо щелкнуть мышью по пустому квадрату (слева от имени поля) на схеме данных, в квадрате появится галочка.

Замечание: Если необходимо сделать поле невидимым при выполнении запроса, то нужно убрать галочку, расположенную слева от имени поля на схеме данных. Для этого просто щелкните мышью по галочке.

Замечание: Если необходимо отобразить все поля таблицы, то необходимо установить галочку слева от пункта **"* (All Columns)"** (Все поля), принадлежащего соответствующей таблице на схеме данных.

Определите отображаемые поля нашего запроса, как это показано на [рис. 8.3](#) (Отображаются все поля кроме полей с кодами, то есть полей связи).

На этом настройку нового запроса можно считать законченной. Перед сохранением запроса проверим его работоспособность, выполнив его. Для запуска запроса на панели инструментов нажмите кнопку

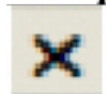


Либо щелкните **ПКМ** в любом месте окна конструктора запросов и в появившемся меню выберите пункт **"Execute SQL"** (Выполнить SQL). Результат выполнения запроса появится в виде таблицы в области результата ([рис. 8.3](#)).

Действителен: с 19.08.2022 по 19.08.2023

Замечание: Если после выполнения запроса результат не появился, а появилось сообщение об ошибке, то в этом случае проверьте, правильно ли создана связь. Ломаная линия связи должна соединять поля **"Код специальности"** в обеих таблицах. Если линия связи соединяет другие поля, то ее необходимо удалить и создать заново, как это описано выше.

Если запрос выполняется правильно, то необходимо сохранить. Для сохранения запроса закройте окно конструктора запросов, щелкнув мышью по кнопке закрытия



расположенной в верхнем правом углу окна конструктора (над схемой данных). Появится окно с вопросом о сохранении запроса ([рис. 8.4](#)).

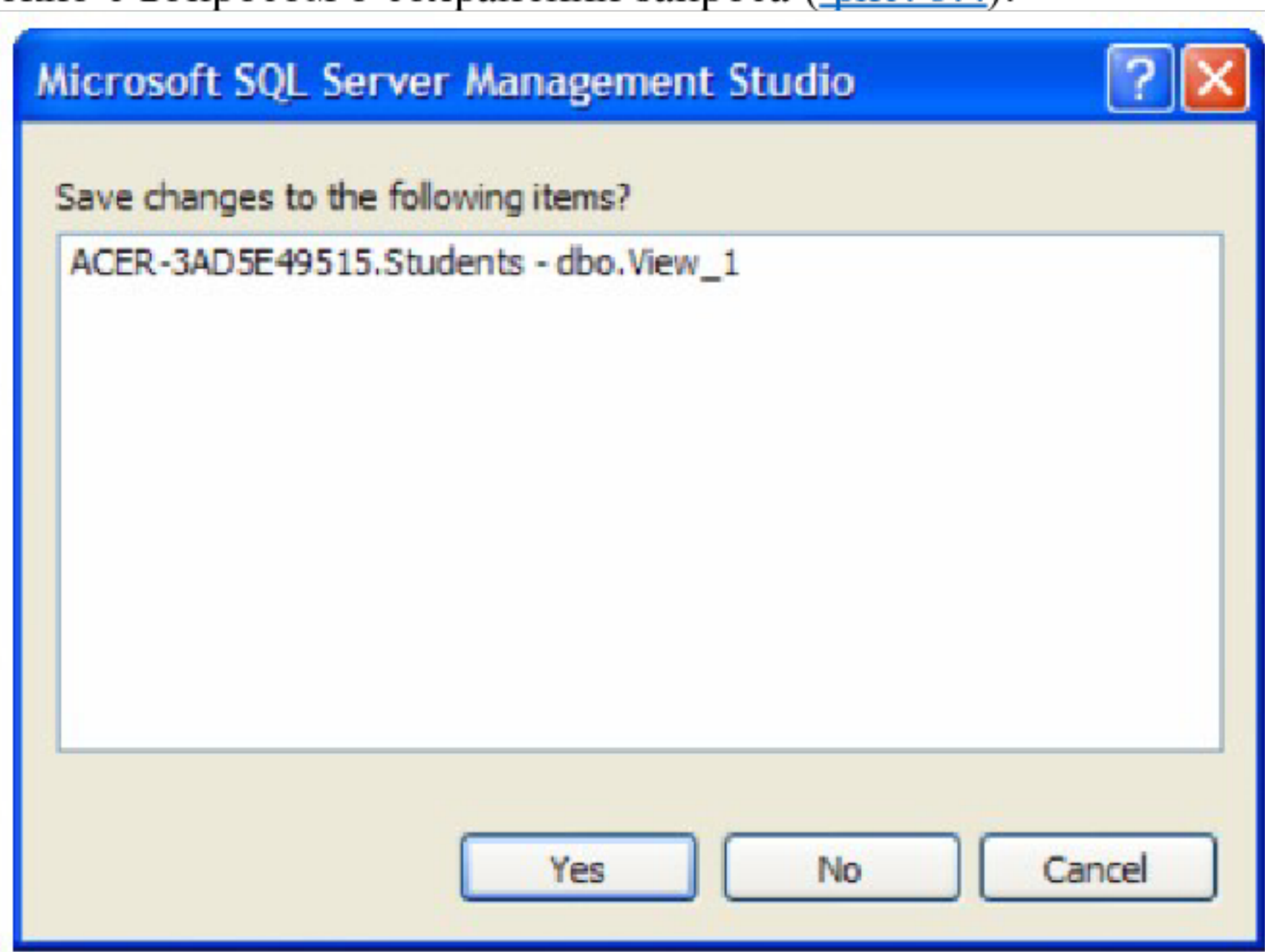


Рис. 8.4.

В данном окне необходимо нажать кнопку **"Yes"** (Да). Появится окно **"Choose Name"** (Выберите имя) ([рис. 8.5](#)).

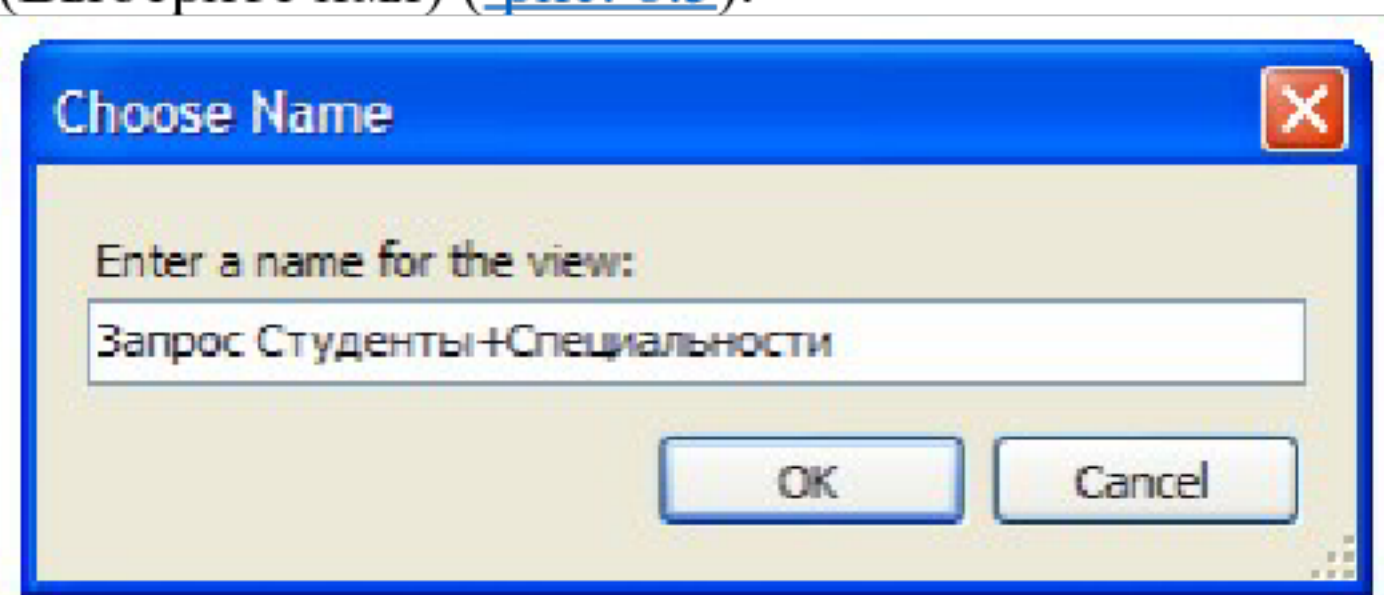


Рис. 8.5.

В данном окне зададим имя нового запроса **"Запрос Студенты+Специальности"** и нажмем кнопку **"Ok"**. Запрос появится в папке **"Views"** БД **"Students"** в обозревателе объектов ([рис. 8.6](#)).

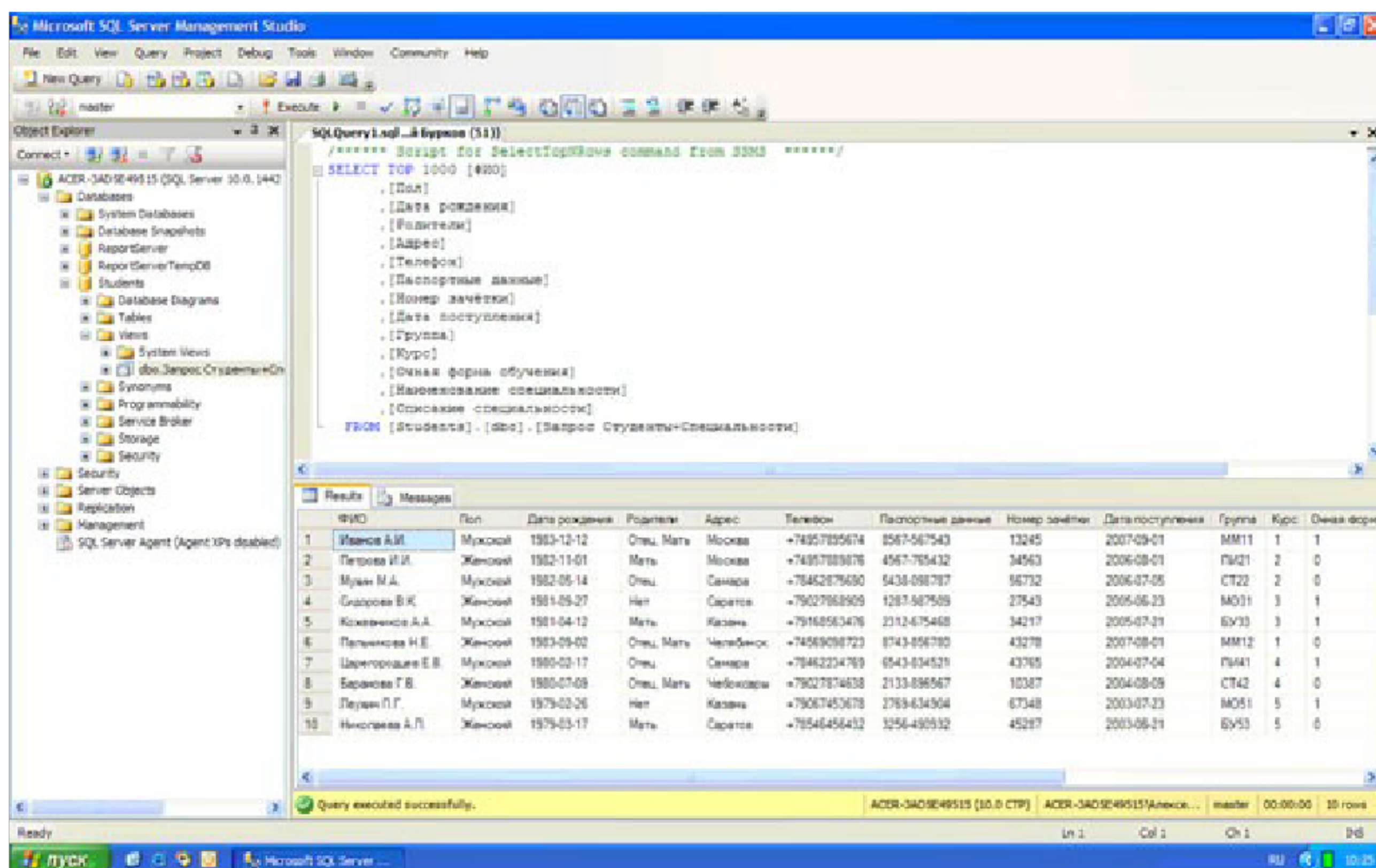


Рис. 8.6.

Проверим работоспособность созданного запроса вне конструктора запросов. Запустим вновь созданный запрос "Запрос Студенты+Специальности" без использования конструктора запросов. Для выполнения уже сохраненного запроса необходимо щелкнуть ПКМ по запросу и в появившемся меню выбрать пункт "Select top 1000 rows" (Отобразить первые 1000 записей). Выполните эту операцию для запроса "Запрос Студенты+Специальности". Результат представлен на рис. 8.6.

Перейдем к созданию запроса "Запрос Студенты+Оценки". В обозревателе объектов в БД "Students" щелкните ПКМ по папке "Views", затем в появившемся меню выберите пункт "New View". Появится окно "Add Table" (рис. 8.2).

В запросе "Запрос Студенты+Оценки" мы связываем таблицы "Студенты" и "Оценки" по полям связи "Код студента". Следовательно, в окне "Add Table" в новый запрос добавляем таблицы "Студенты" и "Оценки". Более того, в данном запросе таблица "Оценки" связывается с таблицей "Предметы" не по одному полю, а по трем полям. То есть поля "Код предмета 1", "Код предмета 2" и "Код предмета 3" таблицы "Оценки" связаны с полем "Код предмета" таблицы "Предметы". Поэтому добавим в запрос три экземпляра таблицы "Предметы" (по одному экземпляру для каждого поля связи таблицы оценки). В итоге в запросе должны участвовать таблицы "Студенты", "Оценки" и три экземпляра таблицы "Предметы" (в запросе они будут называться "Предметы", "Предметы_1" и "Предметы_2"). После добавления таблиц закройте окно "Add Table", появится окно конструктора запросов.

В окне конструктора запросов установите связи между таблицами и определите отображаемые поля, как показано на рис. 8.7.

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

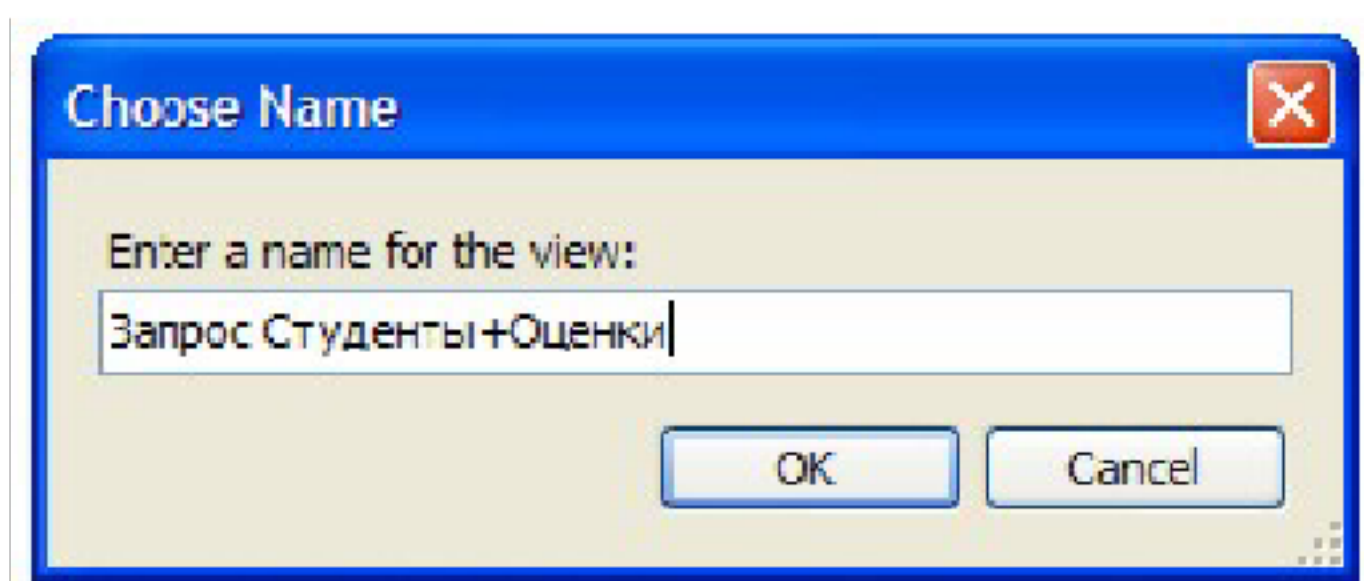
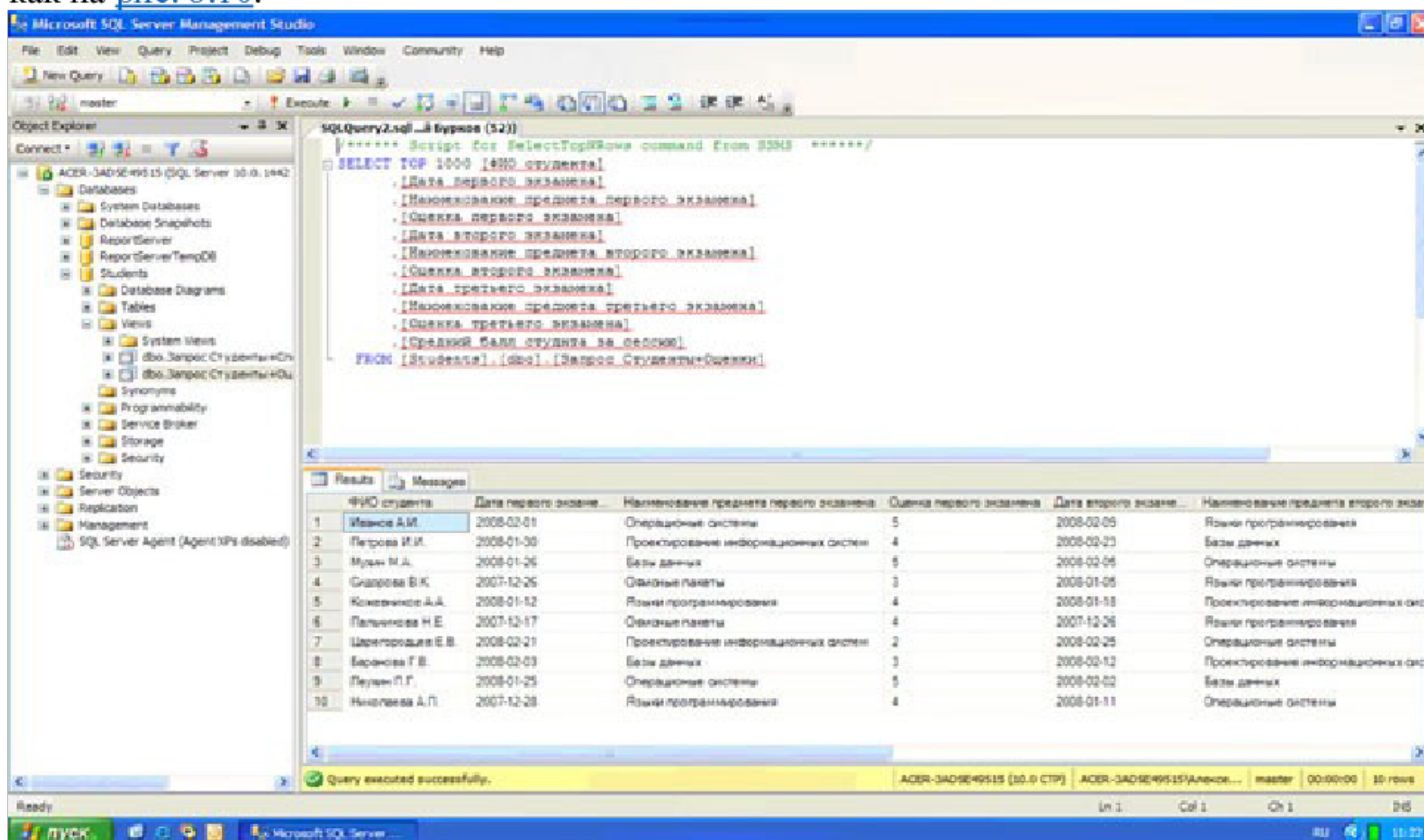


Рис. 8.9.

Проверьте работоспособность нового запроса вне конструктора. Для этого запустите запрос. Результат выполнения запроса "Запрос Студенты+Оценки" должен выглядеть как на [рис. 8.10](#).



[увеличить изображение](#)

Рис. 8.10.

На этом мы заканчиваем рассмотрение обычных запросов и переходим к созданию фильтров.

На основе запроса "Запрос Студенты+Специальности" создадим фильтры, отображающие студентов отдельных специальностей. Создайте новый запрос. Так как он будет основан на запросе "Запрос Студенты+Специальности", то в окне "Add Table" перейдите на вкладку "Views" и добавьте в новый запрос "Запрос Студенты+Специальности" ([рис. 8.11](#)). Затем закройте окно "Add Table".

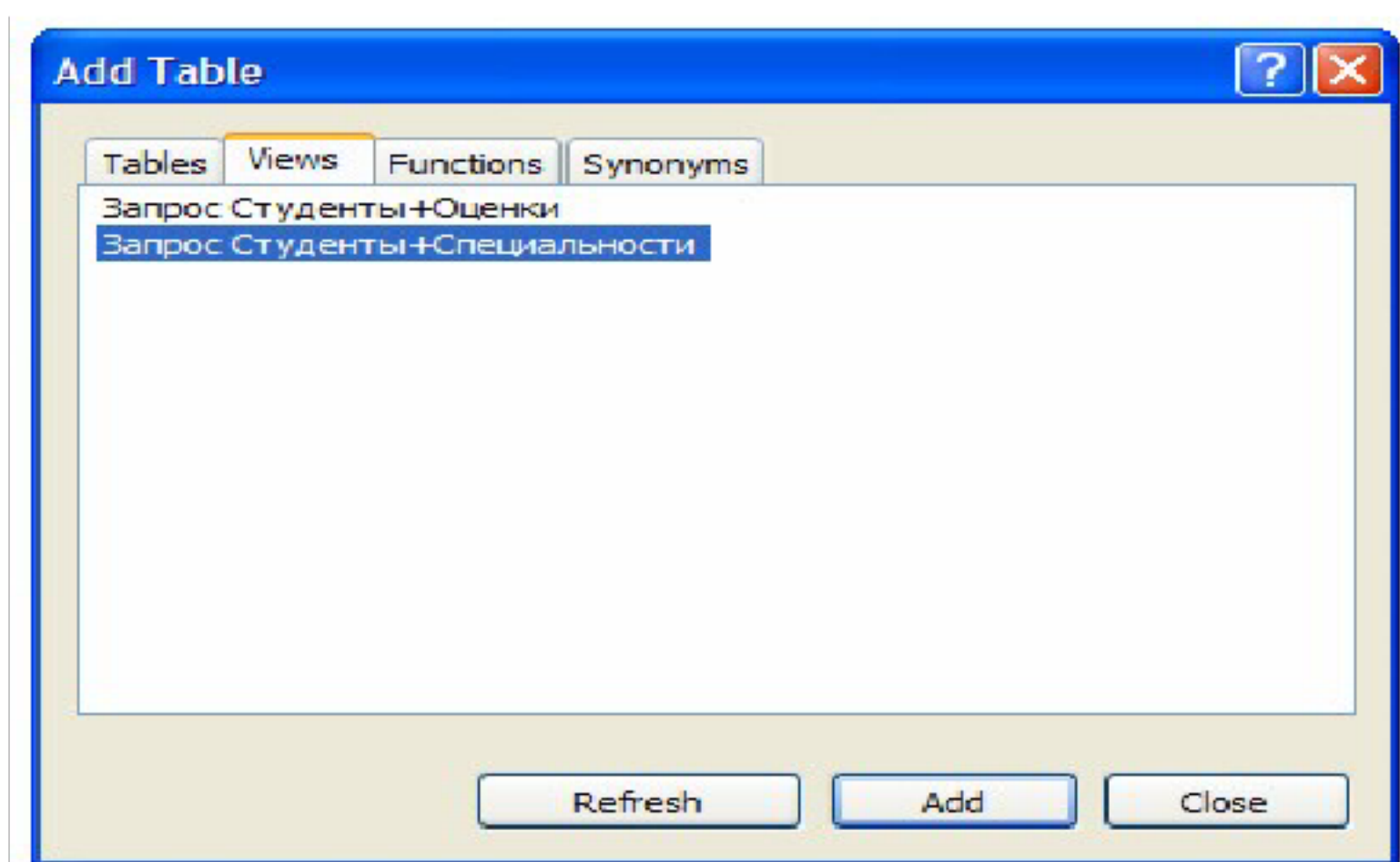
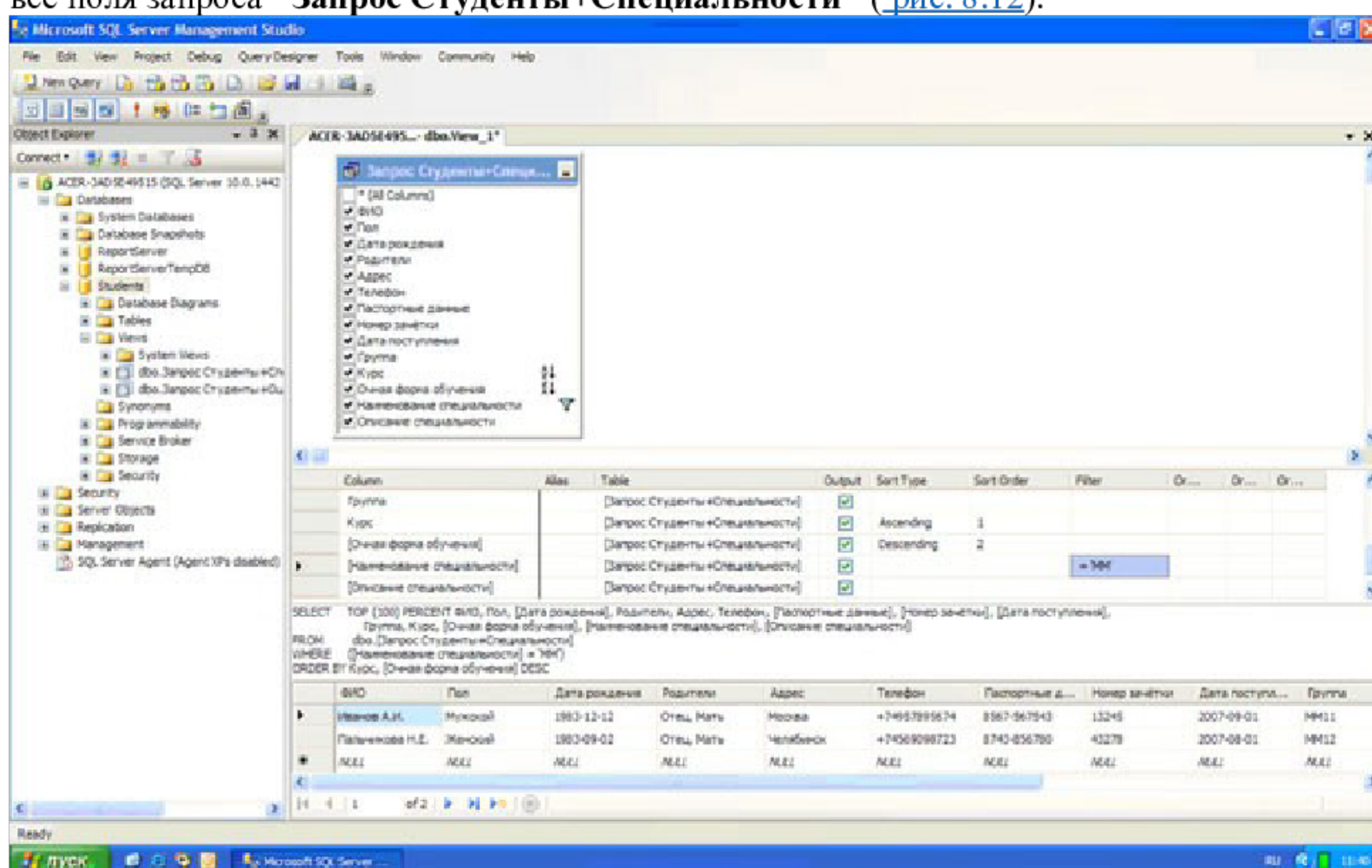


Рис. 8.11.

В появившемся окне конструктора запросов определите в качестве отображаемых полей все поля запроса "Запрос Студенты+Специальности" (рис. 8.12).



[увеличить изображение](#)

Рис. 8.12.

Замечание: Для отображения всех полей запроса, в данном случае, мы не можем использовать пункт **"* (All Columns)"** (Все поля). Так как в этом случае мы не можем устанавливать критерий отбора записей в фильтре, а также невозможно установить сортировку записей.

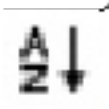
Теперь установим критерий отбора записей в фильтре. Пусть наш фильтр отображает только студентов имеющих специальность "ММ". Для определения условия отбора записей в таблице отображаемых полей в строке, соответствующей полю, на которое накладывается условие, в столбце **"Filter"**, необходимо задать условие. В нашем случае условие накладывается на поле **"Наименование специальности"**. Следовательно, в строке **"Наименование специальности"**, в столбце **"Filter"** нужно задать следующее условие отбора **"= ММ"** (рис. 8.12).

В заключение настроим сортировку записей в фильтре. Пусть при выполнении фильтра сначала происходит сортировка записей по возрастанию по полю **"Очная форма**

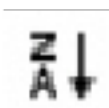
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Чебукова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

обучения", а затем по убыванию по полю **"Курс"**. Для установки сортировки записей по возрастанию, в таблице определяемых полей, в строке для поля **"Очная форма обучения"**, в столбце **"Sort Type"** (Тип сортировки), задайте **"Ascending"** (По возрастанию), а в строке для поля **"Курс"** - задайте **"Descending"** (По убыванию). Для определения порядка сортировки для поля **"Очная форма обучения"** в столбце **"Sort Order"** (Порядок сортировки) поставьте 1, а для поля **"Курс"** поставьте 2 ([рис. 8.12](#)). То есть, при выполнении запроса записи сначала сортируются по полю **"Очная форма обучения"**, а затем по полю **"Курс"**.

Замечание: После установки условий отбора и сортировки записей на схеме данных напротив соответствующих полей появятся специальные значки. Значки



и



обозначают сортировку по возрастанию и убыванию, а значок



показывает наличие условия отбора.

После установки сортировки записей в фильтре проверим его работоспособность, выполнив его. Результат выполнения фильтра должен выглядеть как на [рис. 8.12](#). Закройте окно конструктора запросов. В качестве имени нового фильтра в окне **"Choose Name"** задайте **"Фильтр ММ"** ([рис. 8.13](#)) и нажмите кнопку **"Ok"**.

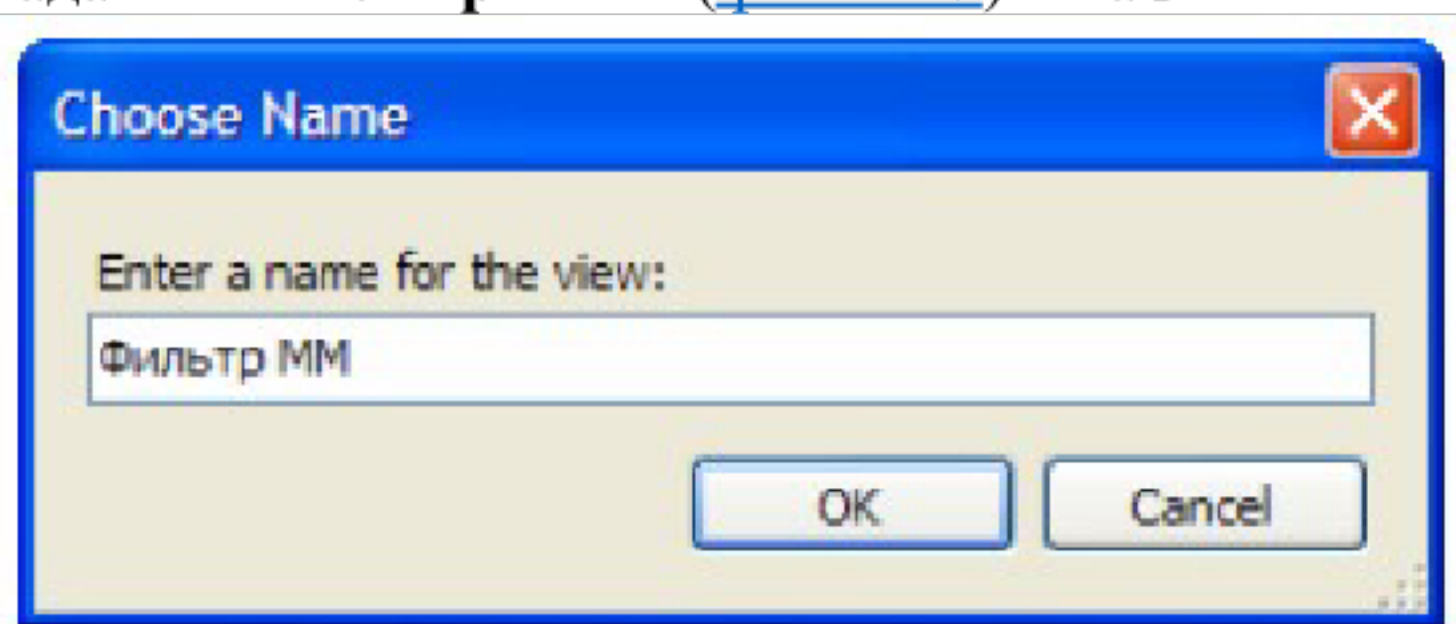
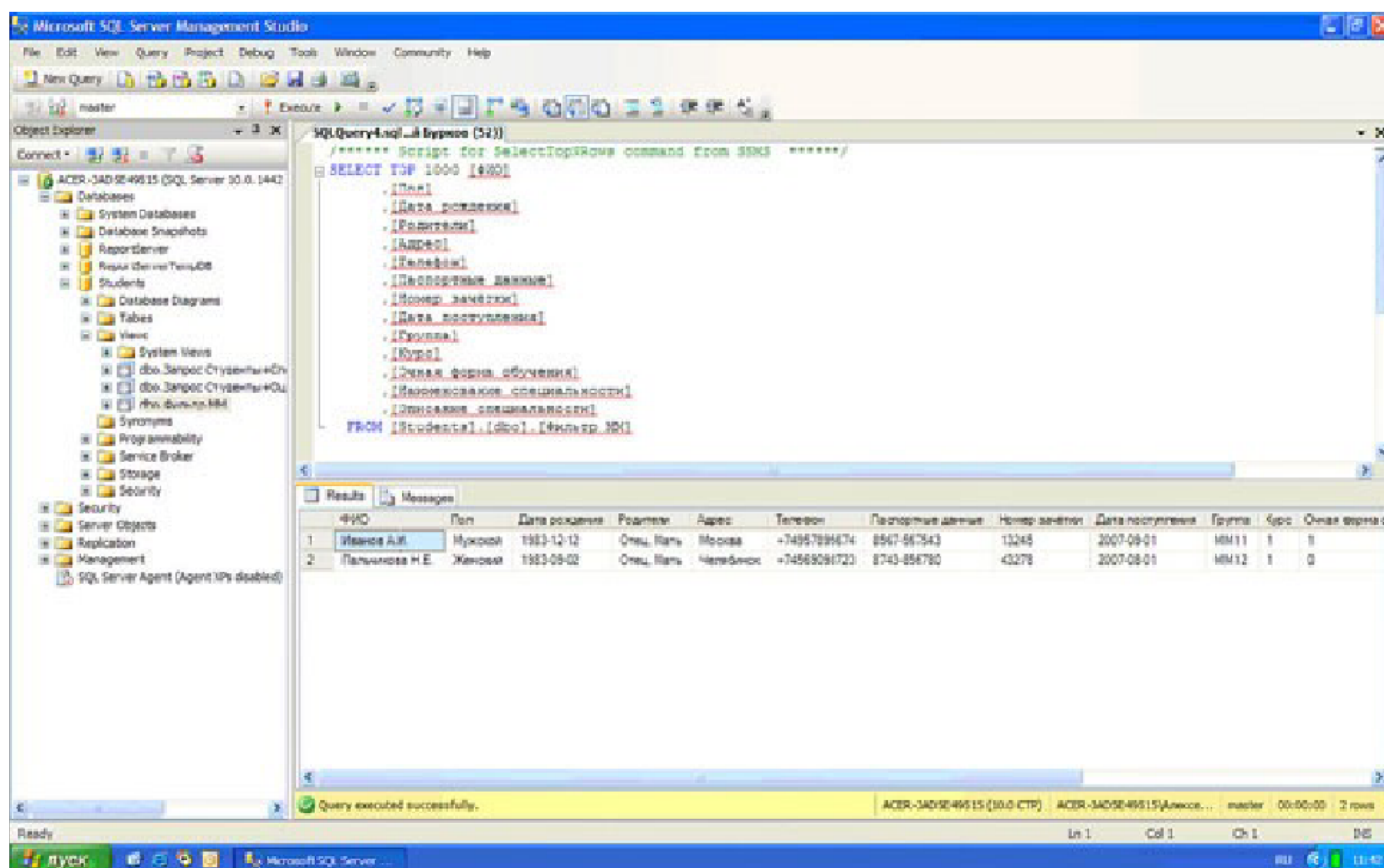


Рис. 8.13.

Фильтр **"Фильтр ММ"** появится в обозревателе объектов. Выполните созданный фильтр вне окна конструктора запросов. Результат должен быть таким же как на [рис. 8.14](#).



[увеличить изображение](#)

Рис. 8.14.

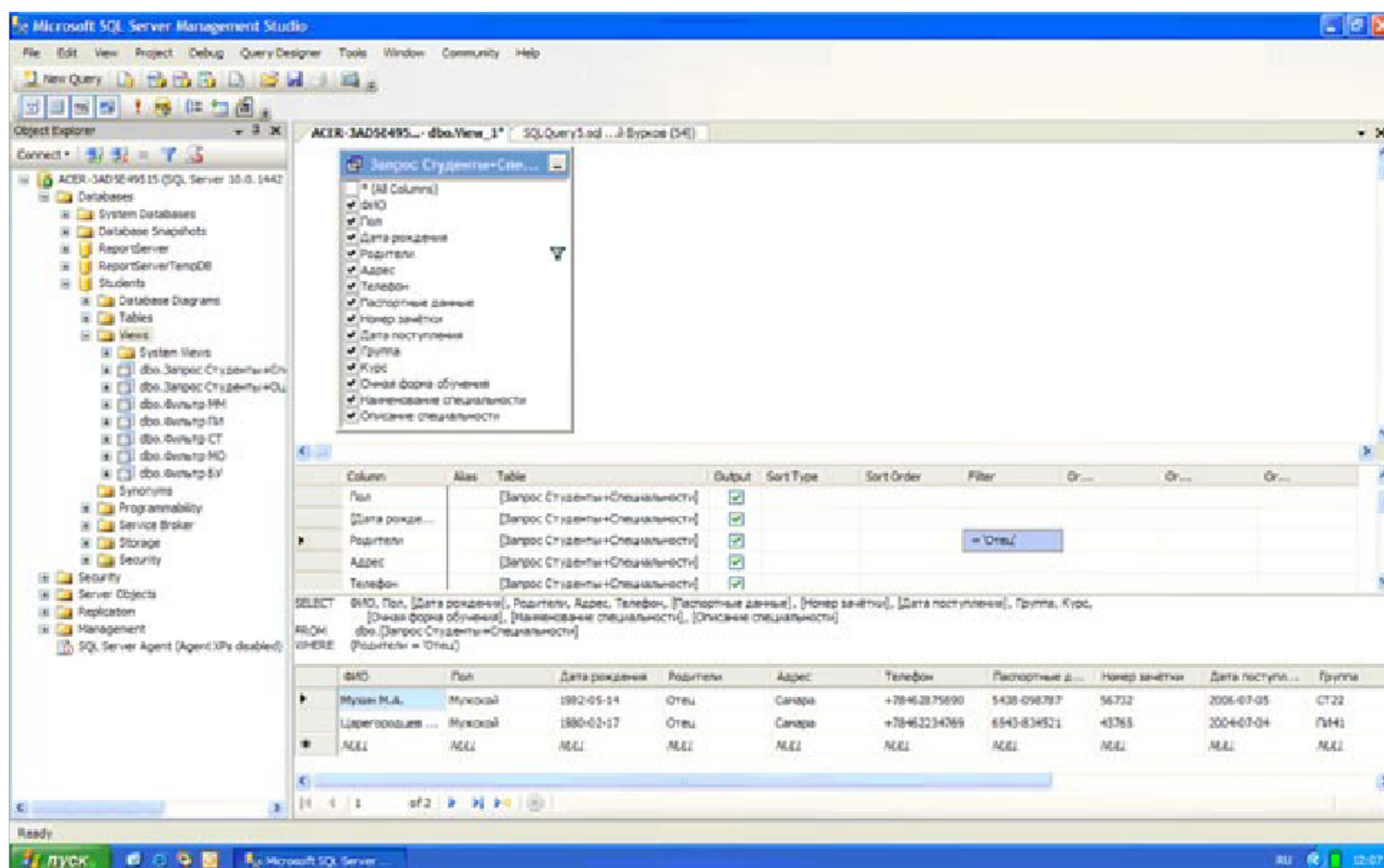
Самостоятельно создайте фильтры для отображения других специальностей. Данные фильтры создаются аналогично фильтру **"Фильтр ММ"** (смотри выше). Единственным отличием является условие отбора, накладываемое на поле **"Наименование специальности"**, оно должно быть не **"='ММ'"**, а **"='ПИ'"**, **"='СТ'"**, **"='МО'"** или **"='БУ'"**. При сохранении фильтров задаем их имена соответственно их условиям отбора, то есть **"Фильтр ПИ"**, **"Фильтр СТ"**, **"Фильтр МО"** или **"Фильтр БУ"**. Проверьте созданные фильтры на работоспособность.

Теперь на основе запроса **"Запрос Студенты+Специальности"** создадим фильтры, отображающие студентов имеющих отдельных родителей. Для начала создадим фильтр для студентов, из родителей только **"Отец"**. Создайте новый запрос и добавьте в него запрос **"Запрос Студенты+Специальности"** ([рис. 8.11](#)). После закрытия окна **"Add Table"** сделайте отображаемыми все поля запроса ([рис. 8.15](#)).

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023



[увеличить изображение](#)

Рис. 8.15.

В таблице отображаемых полей в строке для поля **"Родители"**, в столбце **"Filter"**, задайте условие отбора равное **"=Отец"**. Проверьте работу фильтра, выполнив его. В результате выполнения фильтра окно конструктора запросов должно выглядеть как на [рис. 8.15](#).

Закройте окно конструктора запросов. В окне **"Choose Name"** задайте имя нового фильтра как **"Фильтр Отец"** ([рис. 8.16](#)).

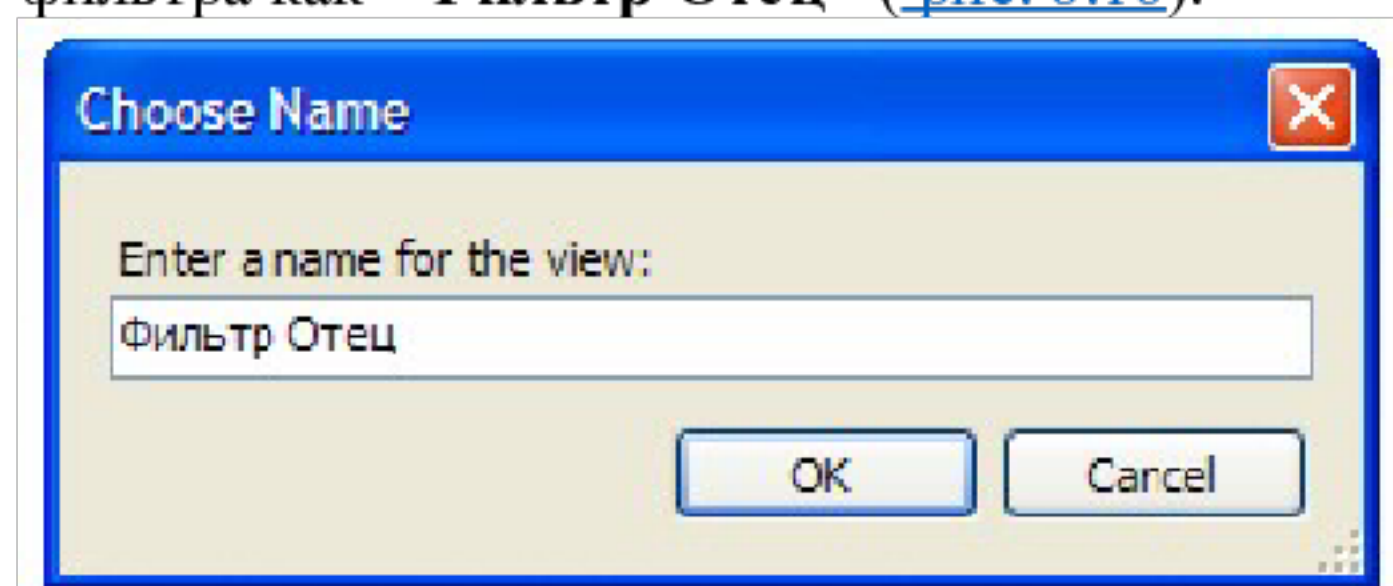


Рис. 8.16.

Выполните фильтр **"Фильтр Отец"** вне конструктора запросов. Результат должен быть аналогичен [рис. 8.17](#).