

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ
МОСКОВСКИЙ ЭНЕРГЕТИЧЕСКИЙ ИНСТИТУТ
(ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ)

Сидорова Н.П.

**Базы данных. Методические указания к курсовому
проектированию**

Для студентов, обучающихся по направлениям «Информатика и
вычислительная техника», «Прикладная математика и информатика»

Москва Издательский дом МЭИ 2010

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

МОСКОВСКИЙ ЭНЕРГЕТИЧЕСКИЙ ИНСТИТУТ
(ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ)

Сидорова Н.П.

**Базы данных. Методические указания к курсовому
проектированию**

Для студентов, обучающихся по направлениям «Информатика и
вычислительная техника», «Прикладная математика и информатика»

Москва Издательский дом МЭИ 2010

УДК

621.398

С 374

Рецензент доц., к.т.н. Крюков А.А.

Сидорова Н.П. Базы данных. Методические указания к курсовому проектированию. М.: Издательский дом МЭИ, 2010. – 40 с.

Рассматриваются вопросы применения методологии структурного моделирования для проектирования реляционных баз данных. В пособии описывается CASE-средство AllFusion Modeller, используемое для анализа предметной области и проектирования баз данных.

Для студентов, обучающихся по направлениям «Информатика и вычислительная техника», «Прикладная математика и информатика», а также будет полезно студентам других специальностей при изучении дисциплин, связанных с проектированием реляционных баз данных.

СОДЕРЖАНИЕ

Введение	5
1. ОБЩИЕ ПОЛОЖЕНИЯ	5
2. ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ ОТЧЕТА	6
3. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	7
3.1. Методология анализа предметной области	7
3.2. Разработка IDEF0 модели в AllFusion Process Modeler	11
3.3.Создание отчетов в AllFusion Process Modeler	19
4. Проектирование БД	21
4.1. Инфологическое проектирование	22
4.2. Логическое проектирование БД	24
4.3. Физическое проектирование БД	27
4.4. Проектирование БД средствами AllFusion ERwin Data Modeler	27
4.4.1. Общие сведения о AllFusion ERwin Data Modeler	27
4.4.2. Создание логической модели данных в AllFusion ERwin Data Modeler.....	29
4.4.3. Создание физической модели в AllFusion ERwin Data Modeler	34
4.4.4. Создание отчета в AllFusion ERwin Data Modeler	37
5. РАЗРАБОТКА ИНТЕРФЕЙСА С БД.....	38
6. ПРИМЕР ПРОЕКТИРОВАНИЯ БД.....	38
Библиографический список	46

Введение

Использование информационных систем (ИС) в настоящее время носит повсеместный характер. Трудно найти сферу деятельности, в которой бы не применялись ИС: транспорт, медицина, управление финансами, банковские операции и многое другое. Результаты использования информационных систем зависят от того, насколько точно определены цели их внедрения и задачи, которые ставятся перед ними. Поэтому особенно важно принять правильные и обоснованные решения на начальных этапах разработки информационной системы. Эти решения принимаются на основе анализа предметной области, в которой предполагается внедрить ИС. Основой построения ИС являются базы данных (БД). Для проведения анализа предметной области, сведения о которой будут содержаться в БД, в настоящее время используются различные методологии. В рамках данного пособия рассматривается методология анализа предметной области на основе структурного анализа бизнес-процессов предметной области – IDEF-технологии. Реализация этой технологии в пособии описывается с использованием CASE-средства AllFusion Modeling Suite.

Пособие предназначено как для студентов, обучающихся по направлениям «Информатика и вычислительная техника», «Прикладная математика и информатика».

Данное методическое пособие рекомендуется для выполнения курсового проекта по дисциплине Базы данных, Базы данных и экспертные системы.

1. ОБЩИЕ ПОЛОЖЕНИЯ

Курсовой проект является самостоятельной работой студентов, в котором закрепляются теоретические знания и практические навыки, изучаемые в рамках дисциплин «Базы данных» и «Базы данных и экспертные системы». Целью курсового проекта является разработка проекта информационной системы в заданной предметной области.

В ходе курсового проектирования студенты должны решить следующие задачи:

- провести анализ предметной области и выделить объекты предметной области и связи между ними, данные о которых должны содержаться в БД;
- выполнить проектирование БД, включая этапы концептуального и логического проектирования;
- реализовать БД средствами выбранной СУБД;
- определить структуру интерфейса с БД и реализовать его средствами языка программирования высокого уровня;
- подготовить отчет по курсовому проекту.

2. ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ ОТЧЕТА

Отчет по курсовому проекту должен содержать следующие разделы:

- Титульный лист
- Оглавление.
- Введение.
- Пояснительная записка.
- Заключение.
- Список используемых источников.
- Приложения, содержащие графические модели БД и тексты программ, реализующих интерфейс с БД.

Отчет по курсовому проекту оформляется на компьютере в текстовом редакторе Microsoft Word и может содержать таблицы, рисунки и иллюстрации и формулы. Для набора сложных математических формул используются редактор Microsoft Equation. Отчет должен быть распечатан шрифтом Times New Roman 14 через 1 интервал между строками на одной стороне белой бумаги формата А4 (210×297 мм) с полями: верхнее – 20 мм, правое – 10 мм, нижнее и левое – не менее 20 мм. Отчет должен быть пронумерован, начиная с Введения. В тексте отчета, таблицах и на

иллюстрациях не допускается сокращений слов, кроме общепринятых. Заголовки текста нумеруют арабскими цифрами (например, Часть 1, подразд. 1.1) . Рекомендуемый объем курсового проекта – 25-30 страниц.

3. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

3.1. Методология анализа предметной области

Одним из основных понятий при проектировании БД является предметная область. Предметная область представляется множеством объектов, например, предприятие - цехами, дирекцией, бухгалтерией и т.д. Каждый объект предметной области характеризуется множеством объектов и процессов, использующих объекты, а также множеством пользователей, характеризующихся различными взглядами на предметную область.

Анализ предметной области, в которой будет функционировать ИС, проводится на основе выделенных в предметной области бизнес-процессов. Бизнес-процесс можно определить как множество взаимосвязанных видов деятельности, приводящих к важному для компании результату. Этот результат не обязательно является конечным видом деятельности компании: продуктом или услугой. Он может иметь ценность для какого-либо его подразделения компании. Например, процесс закупки комплектующих для промышленного предприятия, или процесс подбора кадров для кадрового агентства. Бизнес-процессы, происходящие на предприятии, можно разделить на производственные (основные и вспомогательные) и управленческие. Каждый из них характеризуется технологией реализации, которая зависит от структуры предприятия. Для их реализации используется различное оборудование, средства автоматизации, персонал. Как правило, не существует стандартного списка бизнес-процессов, каждое предприятие должно разрабатывать свой состав основных бизнес-процессов. Выделение бизнес-процессов предприятия требует проведения специального обследования, которое осуществляется либо специальными подразделениями на предприятии, либо с помощью специалистов консалтинговых служб. В

рамках выполнения курсового проекта состав бизнес-процессов является исходными данными.

В большинстве случаев основные бизнес-процессы для предприятий, которые занимаются одинаковым видом деятельности, похожи, например, организация процессов розничной торговли, процессов предоставления банковских услуг и др. Хотя существуют сферы деятельности, где такие процессы выстраиваются предприятием по уникальной схеме, например, транспортировка нефти. Для автоматизации таких процессов используют специализированные отраслевые решения.

Вспомогательные бизнес-процессы связаны с организацией производственного процесса и управлением им (например, бухгалтерия, склад, кадры и т.п.). В большинстве организаций эти бизнес-процессы подчиняются общим правилам. Они достаточно хорошо изучены, описаны и формализованы и унифицированы, что является основой их успешной автоматизации.

Для понимания функционирования основных процессов в предметной области необходимо построить модель, в которой будут отражены все бизнес-процессы, происходящие в ней. Для этого используются формализованные методы, которые включают четкие правила описания таких моделей. Процессная модель предметной области, в которой работает компания, должна строиться с учетом следующих положений:

- Верхний уровень модели должен отражать только контекст диаграммы, в которой отражается взаимодействие компании с внешним миром. Она отражает, что является окружающей средой компании, какова роль компании, какую функцию она выполняет.
- На втором уровне должны быть отражены тематически сгруппированные бизнес-процессы и их взаимосвязи. Они отражают основные направления деятельности компании.
- Детализация бизнес-процессов связана с декомпозицией (разбиением) бизнес-процессов на бизнес – функции.

В настоящее время одним из наиболее эффективных методов построения и анализа бизнес-процессов в предметной области является технология структурного анализа и проектирования систем. В её основе лежит использование IDEF-моделей: IDEF0, DFD, IDEF3.

Основу модели IDEF0 составляет функциональное описание процесса в виде совокупности взаимосвязанных действий, для которых определены правила их взаимодействия и необходимые для них ресурсы. Действие определяет работу (функцию или задачу), которая выполняется при реализации бизнес-процесса. Входные воздействия отражают материалы или информацию, которые изменяются при выполнении работы и преобразуются в результаты работы (выходные воздействия). Управленческие воздействия отражают правила, нормативы, стандарты, которыми руководствуются при выполнении работы. Они не изменяются в процессе выполнения работы. Ресурсы, необходимые для выполнения работы, позволяют учитывать персонал, оборудование и другие элементы, которые используются для выполнения работы. Таким образом, модель IDEF0 представляет собой формализованное описание системы в виде совокупности бизнес-процессов. При выполнении курсового проекта модель IDEF0 позволяет определить информационные ресурсы, необходимые для выполнения выделенных бизнес-процессов. Поэтому при её построении необходимо особое внимание уделить описанию информации (данных), используемых в бизнес-процессах.

Рассматриваемая предметная область, в которой анализируются бизнес-процессы, имеет границу, отделяющую её от других объектов внешнего мира. Граница определяется исходя из целей разработки системы. Определение границ важно для того, чтобы определить взаимодействие разрабатываемой системы с другими элементами предметной области и определить место системы в ней. Взаимодействие системы с окружающей средой описывается как вход (то, что перерабатывается системой) и выход (то, что является результатом действия системы). Такой подход позволяет четко отследить назначение отдельных компонентов системы и отделить

функциональные аспекты отдельных подсистем от их физической реализации. При выполнении курсового проекта для исследования и анализа предметной области и проектирования систем на логическом уровне используется технология декомпозиции моделей бизнес-процессов. Целью анализа модели IDEF0 в курсовом проекте является выделение объектов предметной области и правил работы с ними. Решение этой задачи особенно важно на ранних этапах разработки проекта информационной системы.

Процесс моделирования предметной области в технологии IDEF0 начинается с определения наиболее общего уровня описания системы, в которой определяется краткая характеристика моделируемого объекта, цель моделирования, точка зрения и т.д. При этом необходимо четко определить, что входит в систему, а что находится за ее границами.

Формулируя цель, необходимо ответить на вопросы: что должна показывать модель и для чего она будет использована. Таким образом, цель моделирования определяет вопросы, на которые построенная модель должна дать ответ. Например, «Описать функционирование предприятия для определения требований к ИС».

Точка зрения отражает взгляд на систему человека, который отвечает за работу системы в целом, и соответствует цели моделирования. Очевидно, что процесс функционирования системы будет существенно различаться с точки зрения руководителя и бухгалтера. Поэтому важно определить единую точку зрения. Модель IDEF0 предполагает наличие единого объекта моделирования, четко сформулированной цели и единой точки зрения.

Разработанная модель IDEF0 может быть использована для различных видов анализа: выявления наиболее слабых мест в функционировании системы, возможность модернизации системы, анализ преимуществ новых бизнес-процессов и т.д. В рамках курсового проектирования модель используется для анализа предметной области и выделения основных информационных объектов, которые нужны для выполнения бизнес-

процессов. Поэтому для решения поставленных задач модель IDEF0 следует дополнить моделью DFD.

Модель DFD (Data Flow Diagrams) используется для структурной разработки и структурного анализа информационных систем. В рамках этой модели система представляется как последовательность действий, связанных между собой процессами передачи информации. Диаграммы позволяют описать процесс обмена информацией между элементами системы. В этой модели отображаются источники данных, получатели данных, идентифицируются процессы и группы данных, связывающие в потоки одну функцию с другой. В модели DFD определяются хранилища данных, которые используются в исследуемом процессе. Эта модель наиболее успешно применяется при проектировании программного обеспечения. Диаграммы потоков данных являются основным средством моделирования функциональных требований к проектируемой системе.

К преимуществам методики DFD относятся:

- возможность однозначно определить внешние сущности, анализируя потоки информации внутри и вне системы;
- возможность проектирования сверху вниз, что облегчает построение модели "как должно быть";
- наличие спецификаций процессов нижнего уровня, что позволяет преодолеть логическую незавершенность функциональной модели и построить полную функциональную спецификацию разрабатываемой системы.

Для построения указанных моделей в ходе курсового проектирования используется CASE-средство AllFusion Modeling Suite.

3.2. Разработка IDEF0 модели в AllFusion Process Modeler

Основу разработки модели IDEF0 составляет графический язык описания бизнес-процессов. Модель представляет собой совокупность иерархически упорядоченных диаграмм, каждая из которых располагается на

отдельном листе. В рамках модели IDEF0 могут быть разработаны следующие виды диаграмм:

- контекстная – одна для каждой модели, она представляет самое общее описание системы и ее взаимодействие с окружающей средой;
- декомпозиции – отражает более детальное описание отдельной части бизнес-процесса;
- дерева узлов – описывает иерархическую зависимость работ;
- иллюстративная – для иллюстрации отдельных видов моделей.

Основными элементами модели IDEF0 являются: функциональный блок и стрелка.

Функциональный блок графически изображается в виде прямоугольника и задает собой конкретный процесс (функцию) в рамках рассматриваемой системы. Блок отражает перевод входной информации в выходной продукт, с помощью чего или кого реализуется (отражает используемые механизмы) и что регламентирует выполнение функции (отражает ограничения).

Для каждого функционального блока в рамках модели определяется:

- уникальный номер, который присваивается автоматически
- название, которое должно отражать действие, оно задается только в виде глагола или отглагольного существительного, например, ИЗГОТОВИТЬ ДЕТАЛЬ, СОБРАТЬ ДАННЫЕ, ИЗГОТОВЛЕНИЕ ДЕТЕЛИ, СБОР ДАННЫХ;
- описание, которое дает развернутую смысловую характеристику действия. Например, сбор данных по телефону, сбор данных по электронной почте, проведение опроса и др.

При создании новой модели автоматически создается контекстная диаграмма с единственным функциональным блоком, отображающим систему в целом (рис. 1).

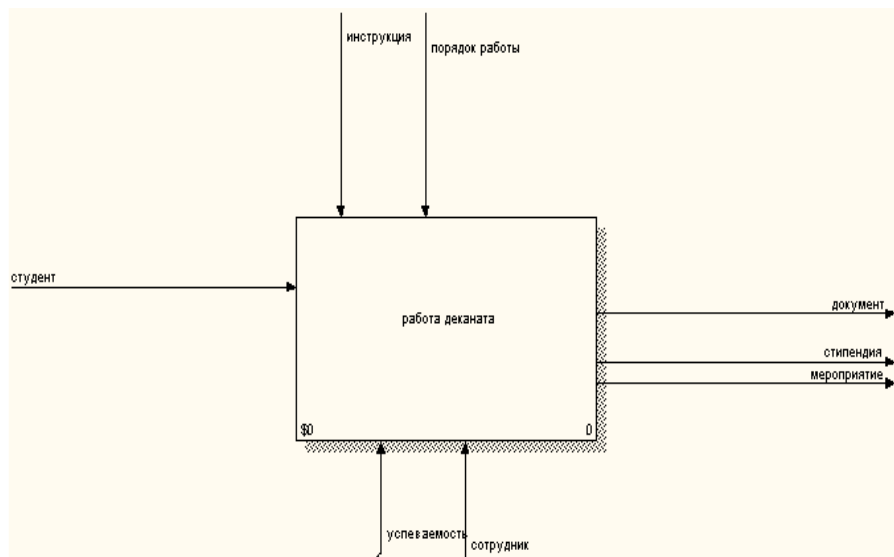


Рис.1. Пример контекстной диаграммы

Для того чтобы задать другие свойства блока необходимо нажать правой клавишей мыши на изображении блока и выбрать нужное свойство *Activity properties* (рис. 2). Для каждого действия необходимо задать его имя (*Name*) и описание (*Definition*). Разработка модели начинается с определения контекстной диаграммы, в которой определяется взаимодействие модели с внешними компонентами (рис. 1).

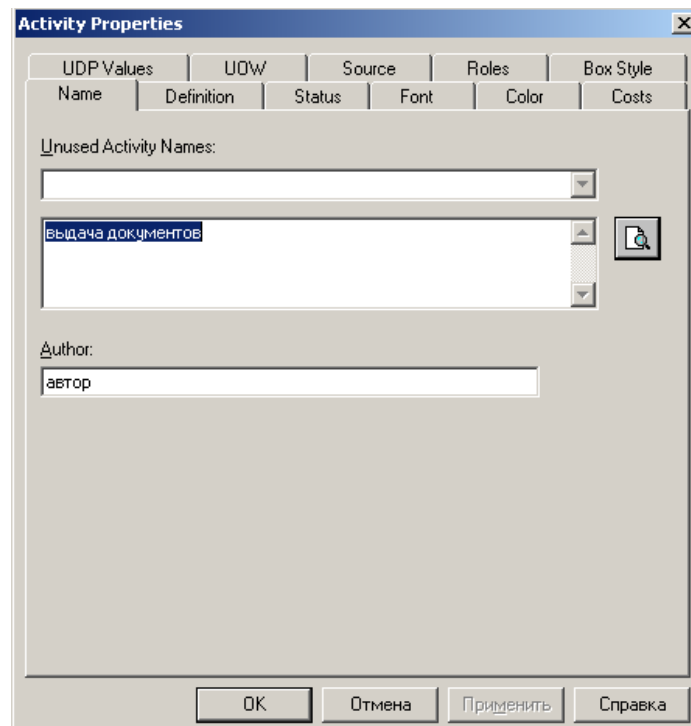


Рис. 2. Редактор задания свойств действия.

Второй основной элемент IDEF0-методологии — это стрелка. *Стрелка* бывает четырех типов: стрелка-вход, выход, механизм и управление.

1. Вход (***Input***) рисуется, как входящая в левую грань функционального блока. Вход показывает, что требуется для выполнения функции, например: КЛИЕНТ.

2. Выход (***Output***) – исходящая из правой грани блока. Выход — результат функции, например: РЕАЛИЗОВАННАЯ УСЛУГА

3. Механизм (***Mechanism***) входящая в нижнюю грань стрелка. Механизм с помощью чего или кого выполняется функция, например: ПЕРСОНАЛ

4. Управление (***Control***) рисуется входящей в верхнюю грань блока. Управление ограничивает (регламентирует) выполнение функции, например, ПРАВИЛА, ЛИЦЕНЗИЯ.

Каждая стрелка (рис.3) определяется уникальным именем (***Name***), которое задается существительным в единственном числе им. падеже, и описанием (***Definition, Note***) (рис. 4). Имена вновь внесенных стрелок автоматически заносятся в словарь (***Arrow Dictionary***).

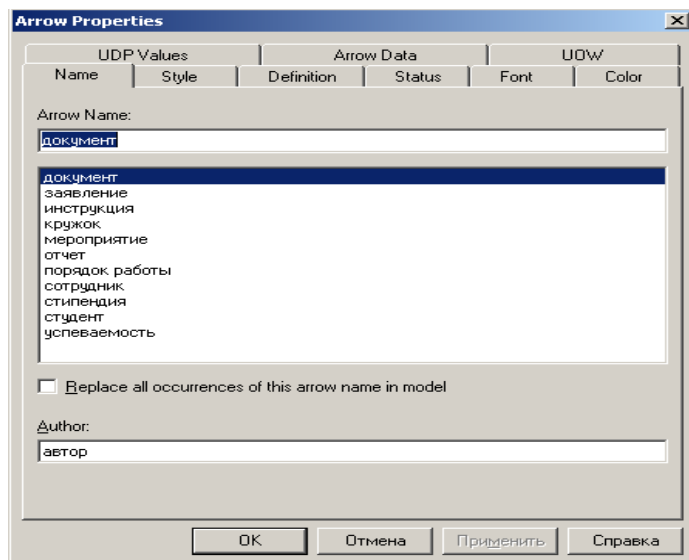



Рис. 3. Задание имени стрелки.

Стрелки могут быть **внутренними и граничными**. Внутренние стрелки соединяют блоки между собой. Граничные стрелки служат для описания взаимодействия с внешней средой. Они могут начинаться у блока, а

заканчиваться у границы диаграммы (на контекстной диаграмме используются только граничные стрелки). Для задания и изменения свойств стрелки необходимо выбрать на панели инструментов элемент  и дважды щелкнуть левой кнопкой мыши на стрелке.

Одни и те же данные или объекты, порожденные одним процессом, могут использоваться сразу в нескольких других процессах. С другой стороны, стрелки, порожденные в разных работах, могут представлять собой одинаковые или однородные данные или объекты, которые в дальнейшем используются или перерабатываются в одном месте.

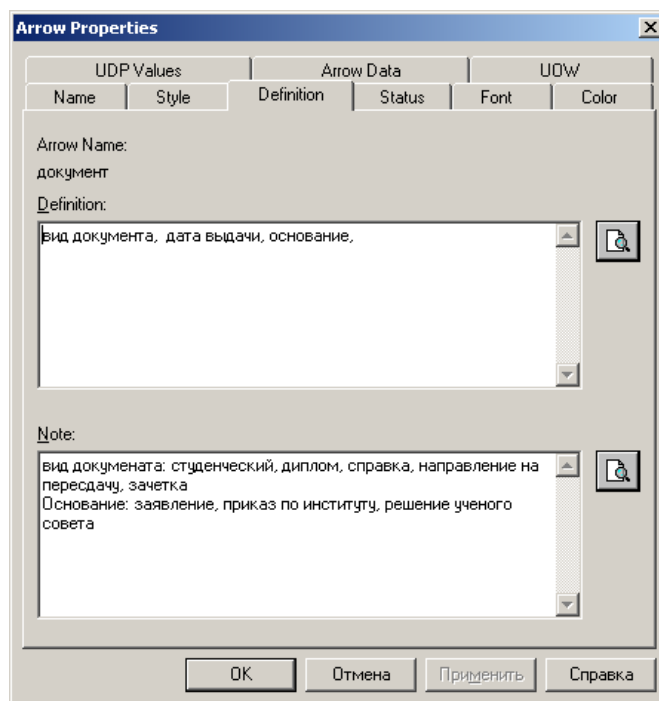


Рис. 4. Задание свойств Definition и Note

Для моделирования таких ситуаций в IDEF0 используются разветвляющиеся и сливающиеся стрелки. Для разветвления стрелки нужно в режиме рисования стрелки щелкнуть по фрагменту стрелки и по соответствующему сегменту блока. Для слияния двух стрелок выхода нужно в режиме рисования стрелки сначала щелкнуть по сегменту выхода блока, а затем по соответствующему сегменту стрелки.

Существуют определенные правила именования таких стрелок. Если стрелка именована до разветвления, а после разветвления ни одна из ветвей не именована, то подразумевается, что каждая ветвь моделирует те же

данные или объекты, что и ветвь до разветвления. Если стрелка именована до разветвления и одна из ветвей после разветвления, то подразумевается, что именованная ветвь моделирует данные соответствующие ее имени, а не именованные те же данные, что и ветвь до разветвления.

Туннелирование стрелок. Вновь внесенные граничные стрелки на диаграмме декомпозиции нижнего уровня изображаются в квадратных скобках (рис. 5) и автоматически не появляются на диаграмме верхнего уровня. Квадратный туннель является предупреждением для разработчика о возможной ошибке (случайное удаление или добавление стрелки). Квадратный туннель должен быть заменен на круглый туннель либо стрелка добавлена на родительской диаграмме.

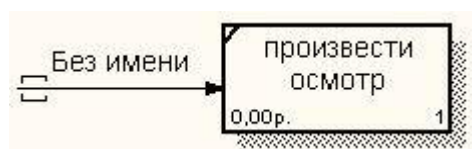


Рис. 5. Туннельная стрелка

Для их «перетаскивания» вверх нужно щелкнуть правой кнопкой мыши по квадратным скобкам граничной стрелки и выбрать из выпадающего меню Arrow Tunnel. Появится диалоговое окно Border Arrow Editor (рис. 6).



Рис. 6. Диалог Border Arrow Editor

Дальнейшая работа по созданию моделей IDEF0 связана с декомпозицией контекстной диаграммы. Декомпозиция – это разбиение действия бизнес-процесса на его составные части. Для каждого выделенного в рамках бизнес-процесса действия строится своя IDEF0-модель. Для

перехода к следующему уровню декомпозиции необходимо выбрать элемент панели инструментов ▼, который автоматически откроет новую страницу диаграммы.

Таким образом, строится иерархическая система диаграмм, на каждом уровне которой бизнес-процесс описывается подробнее. Пример диаграммы приведен на рис. 7.

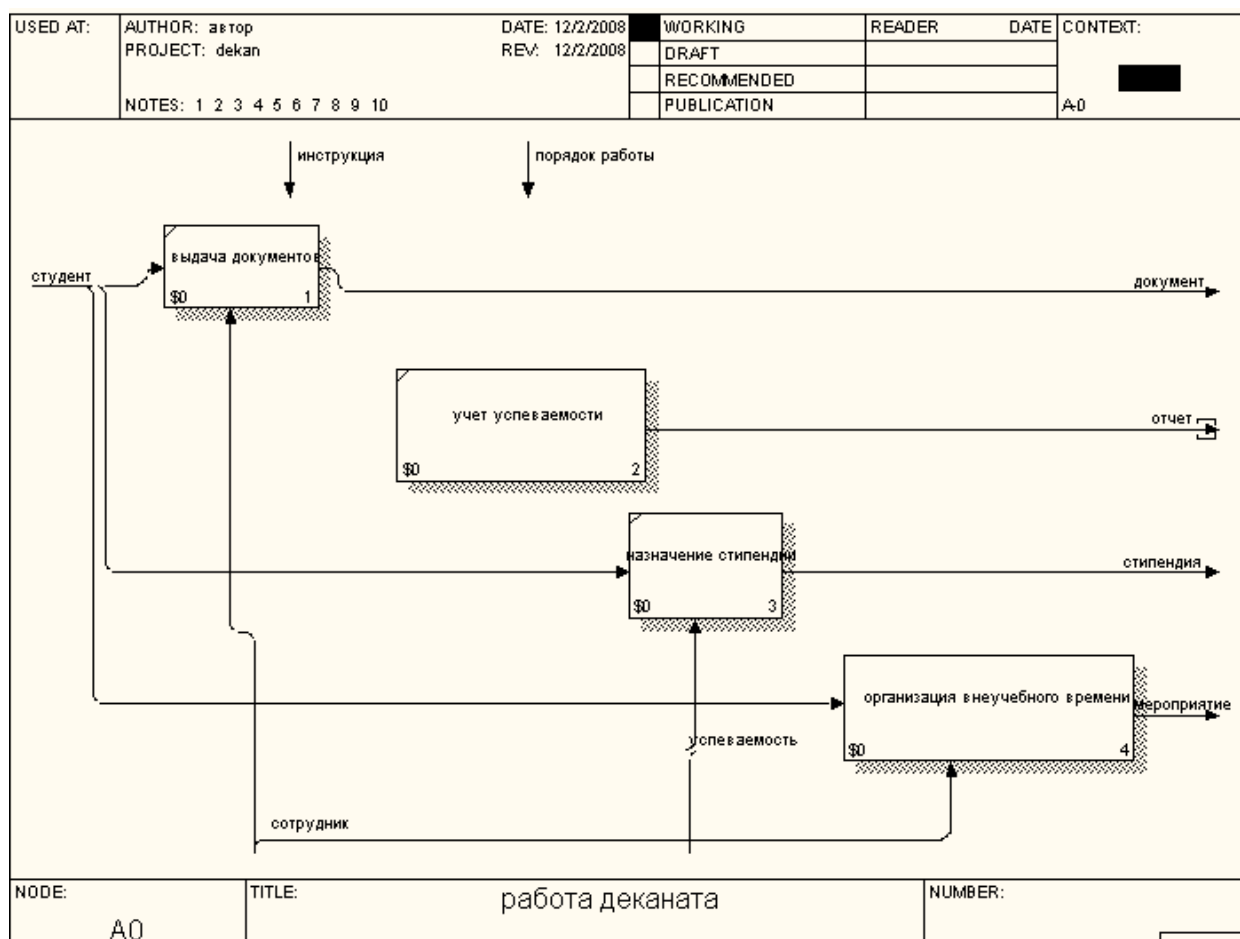



Рис. 7. Декомпозиция диаграммы процесса


Разработанные модели типа IDEF0 дополняются моделями типа DFD, которые создаются на нижних уровнях декомпозиции. При создании диаграммы потоков данных используются четыре основных понятия: потоки данных, процессы (работы) преобразования входных потоков данных в выходные, внешние сущности, накопители данных (хранилища).

Потоки данных используются для моделирования передачи информации от одного действия к другому. Потоки на диаграммах

изображаются именованными стрелками, ориентация которых указывает направление движения информации.

Назначение процесса аналогично процессу в модели IDEF0. В этой модели применяются специальные элементы: хранилище данных и внешняя сущность.

Хранилище данных (*Data Store*) позволяет определять данные, которые будут сохраняться в памяти между процессами. Информация, которую оно содержит, можно использовать в любое время после ее получения, при этом данные могут выбираться в любом порядке. Имя хранилища должно определять его содержимое и быть существительным. Для того чтобы добавить хранилище данных в модель необходимо выделить элемент  панели инструментов и поместить его в поле модели.

Внешняя сущность представляет собой внешний источник или получатель данных. Ее имя должно содержать существительное, например, "клиент". Предполагается, что объекты, представленные как внешние сущности, не должны участвовать ни в какой обработке. На панели инструментов внешняя сущность задана пиктограммой . Пример DFD диаграммы приведен на рис.8.

Для всех внешних сущностей строится таблица событий, описывающая их взаимодействие с основным потоком. Таблица событий включает в себя наименование внешней сущности, событие, его тип (типичный для системы или исключительный, реализующийся при определенных условиях) и реакцию системы. После построения потоков данных диаграмма должна быть проверена на полноту и непротиворечивость. Полнота диаграммы обеспечивается, если в системе нет "повисших" процессов, не используемых в процессе преобразования входных потоков в выходные. Непротиворечивость системы обеспечивается выполнением наборов формальных правил о возможных типах процессов: на диаграмме не может быть потока, связывающего две внешние сущности – это взаимодействие удаляется из рассмотрения; ни одна сущность не может непосредственно

получать или отдавать информацию в хранилище данных – хранилище данных является пассивным элементом, управляемым с помощью интерфейсного процесса; два хранилища данных не могут непосредственно обмениваться информацией – эти хранилища должны быть объединены.

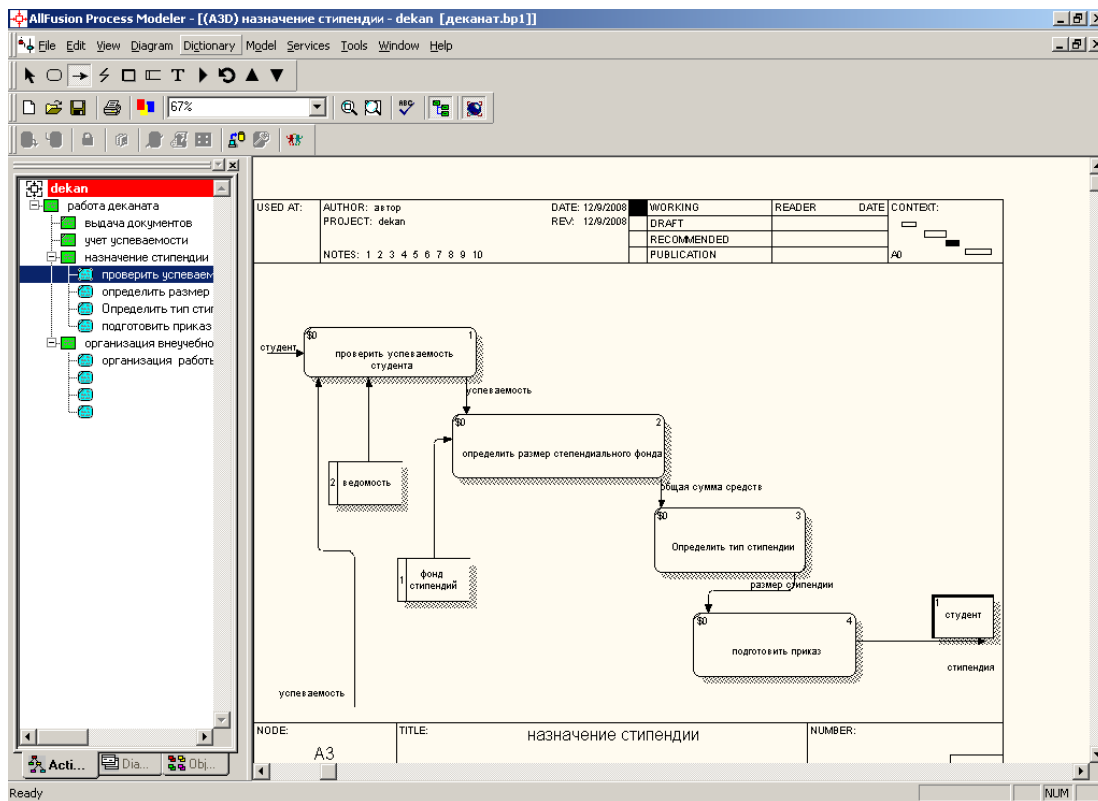


Рис. 8. Диаграмма потока данных

3.3. Создание отчетов в AllFusion Process Modeler

AllFusion Process Modeler имеет мощный инструмент генерации отчетов – Builder Report. Всего имеется семь типов шаблонов отчетов:

1. Model Report. Включает информацию о контексте модели — имя модели, точку зрения, область, цель, имя автора, дату создания и др.
2. Diagram Report. Отчет по конкретной диаграмме. Включает список объектов (работ, стрелок, хранилищ данных, внешних ссылок и т. д.).
3. Diagram Object Report. Наиболее полный отчет по модели. Может включать полный список объектов модели (работ, стрелок с указанием их типа и др.) и свойства, определяемые пользователем.
4. Activity Cost Report. Отчет о результатах стоимостного анализа. Будет рассмотрен ниже.

5. Arrow Report. Отчет по стрелкам. Может содержать информацию из словаря стрелок, информацию о работе-источнике, работе-назначении стрелки и информацию о разветвлении и слиянии стрелок.

6. Data Usage Report. Отчет о результатах связывания модели процессов и модели данных.

Отчеты используются как средство документирования модели и как основа для проведения дальнейшего анализа. Вызвать генератор отчетов можно из меню Tools/Report Builder, который открывает диалог Report Templates (рис. 9). Этот диалог позволяет сформировать шаблон отчета, определив в нем те разделы, описание которых будет помещено в автоматически создаваемый отчет. Кнопка New открывает диалог создания нового отчета. Окно *Output Type* позволяет задать формат отчета, который выбирается из списка: текстовый (*TXT*), *RTF*, в виде Web-страницы (*HTML*). Для того чтобы сформировать шаблон отчета, необходимо определить разделы, из которых он будет состоять. Экран диалога Report Templates состоит из двух окон и панели инструментов.

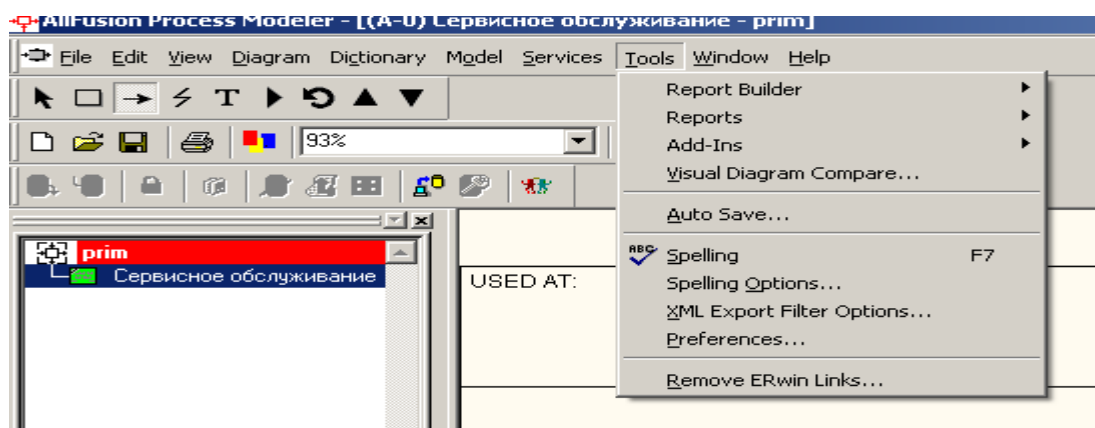



Рис. 9. Вызов генератора отчетов

В левом окне расположен список всех возможных разделов отчета, в правой – те разделы, которые вы включаете в отчет. Для переноса нужного раздела из левой части в правую, его необходимо выделить и нажать кнопку ►. Для определения того, какая информация будет отражаться в заданный

раздел, необходимо выделить его и щелкнуть правой кнопкой мыши. В открывшейся вкладке свойств выделенного раздела отмечаются те свойства, которые вы хотите включить в отчет (рис. 10). Для генерации отчета следует нажать кнопку .

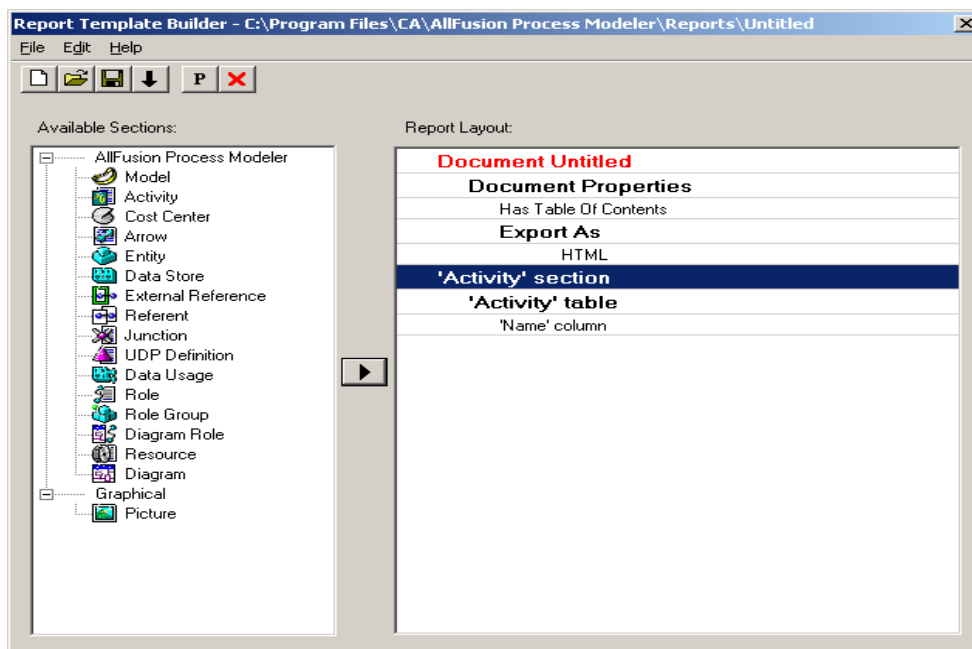


Рис. 10. Диалог создания структуры отчета

Данные, полученные в отчете, позволяют провести анализ выбранной предметной области и выделить в ней объекты, их свойства и связи между ними.

4. Проектирование БД

Проектирование БД – один из наиболее сложных и ответственных этапов. В результате проектирования необходимо определить структуру БД, эффективный для всех пользователей способ организации данных и интерфейс с БД. Процесс проектирования включает в себя следующие этапы:

- Инфологическое проектирование.
- Определение требований к операционной среде, в которой будет функционировать информационная система, и выбор СУБД и инструментальных программных средств.
- Логическое проектирование БД.
- Физическое проектирование БД.

Основная цель процесса проектирования БД состоит в получении моделей БД, которые удовлетворяют следующим требованиям:

- корректность схемы БД, т.е. база должна адекватно отображать моделируемую предметную область, в которой где каждому объекту предметной области соответствуют данные в структуре БД, а каждому процессу – процедуры обработки данных;
- обеспечение ограничений на целостность данных;
- эффективность функционирования (соблюдение ограничений на время реакции системы на запрос и обновление данных).
- простота и удобство эксплуатации
- гибкость, т.е. возможность развития БД и адаптации к изменениям предметной области и/или требований пользователей.

4.1. Инфологическое проектирование

Инфологическая модель предметной области представляет собой описание её структуры и динамики, характера информационных потребностей пользователей в терминах, понятных пользователю и не зависящих от реализации БД. Это описание выражается в терминах не конкретных объектов предметной области, а их типов, связанных с ними ограничений целостности и тех процессов, которые приводят к переходу предметной области из одного состояния в другое. Одним из наиболее часто используемых моделей является модель «сущность-связь» (ER-модель). Основными элементами этой модели являются сущность, атрибут и связь.

Сущность – это объект, о котором в системе будет накапливаться информация. Сущности бывают как физически существующие (например, СОТРУДНИК или АВТОМОБИЛЬ), так и абстрактные (например, ЭКЗАМЕН или ДИАГНОЗ).

Для сущностей различают тип сущности и экземпляр сущности. Тип характеризуется именем и списком свойств, а экземпляр – конкретными значениями свойств.

Выделяют сильные и слабые типы сущностей. Сильные сущности существуют сами по себе, а существование слабых сущностей зависит от существования сильных. Например, читатель библиотеки – сильная сущность, а заказ литературы от этого читателя – слабая, которая зависит от наличия соответствующего читателя. Для каждой сущности выбираются свойства (атрибуты). Различают:

1. Идентифицирующие и описательные атрибуты. Идентифицирующие атрибуты имеют уникальное значение для сущностей данного типа и являются потенциальными ключами. Они позволяют однозначно распознавать экземпляры сущности. Из потенциальных ключей выбирается один первичный ключ. Первичный ключ должен включать в свой состав минимально необходимое для идентификации количество атрибутов. Остальные атрибуты называются описательными и включают в себе интересующие свойства сущности.

2. Составные и простые атрибуты. Простой атрибут состоит из одного компонента, его значение неделимо. Составной атрибут является комбинацией нескольких компонентов, возможно, принадлежащих разным типам данных (например, ФИО или адрес). Решение о том, использовать составной атрибут или разбивать его на компоненты, зависит от характера его обработки и формата пользовательского представления этого атрибута.

3. Однозначные и многозначные атрибуты (могут иметь соответственно одно или много значений для каждого экземпляра сущности).

4. Основные и производные атрибуты. Значение основного атрибута не зависит от других атрибутов. Значение производного атрибута вычисляется на основе значений других атрибутов (например, возраст студента вычисляется на основе даты его рождения и текущей даты).

Связи могут иметь различный содержательный смысл (семантику). Различают связи типа "сущность-сущность", "сущность-атрибут" и "атрибут-атрибут" для отношений между атрибутами, которые характеризуют одну и ту же сущность или одну и ту же связь типа "сущность-сущность".

По типу различают следующие виды связей:

- связи один к одному – (1:1),
- один ко многим – (1:n)
- многие ко многим – (m:n).

ER-модель, содержащая различные типы связей, приведена на рис. 11.

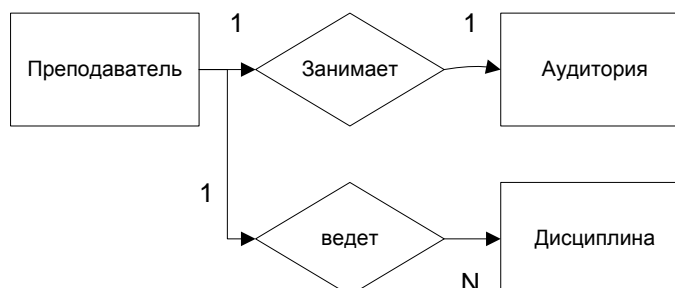


Рис.11. Пример ER-модели.

4.2. Логическое проектирование БД

На этапе логического проектирования разрабатывается логическая структура БД. Решение этой задачи существенно зависит от модели данных, поддерживаемой выбранной СУБД. В рамках выполнения курсового проекта в качестве логической модели используется реляционная модель БД. Проектирование реляционной БД (РБД) имеет свои особенности. Оно должно отвечать дополнительным требованиям, связанным с минимизацией дублирования данных и упрощения процедур их обработки и обновления. В случае неправильно спроектированной РБД могут возникнуть аномалии модификации данных, которые обусловлены недостаточно полно проведенным анализом типов связей между объектами предметной области и неполнотой анализа правил целостности данных. Для устранения таких недостатков применяется процедура нормализации отношений, образующих схему РБД и определяются правила целостности БД.

Автор реляционной модели данных Э.Ф. Кодд (E.F. Codd) предложил процедуру нормализации отношений и разработал механизм, позволяющий любое отношение преобразовать к виду, который удовлетворяет приведенным выше требованиям.

Процесс нормализации схемы отношения, входящего в состав РБД, выполняется путём её декомпозиции. **Декомпозицией** схемы отношения R называется замена её совокупностью схем отношений A_i таких, что

$$R = \bigcup_i A_i,$$

При этом не требуется, чтобы отношения A_i были непересекающимися.

Для определения понятия нормальной формы было введено понятие простого и сложного атрибута. Простой атрибут – это атрибут, значения которого атомарны (т.е. неделимы с точки зрения их обработки). Сложный атрибут может иметь значение, представляющее собой объединение нескольких значений, каждое из которых может обрабатываться самостоятельно. Примером сложного атрибута может быть агрегат данных или повторяющийся агрегат данных.

Ниже приведены определения нормальных форм (НФ), используемых при проектировании РБД.

Первая нормальная форма (1НФ).

Отношение находится в 1НФ, если все его атрибуты простые.

Для того чтобы привести к 1НФ отношение, содержащее повторяющиеся атрибуты (агрегаты), нужно построить декартово произведение всех повторяющихся агрегатов с кортежами, к которым они относятся. Для идентификации кортежа отношения в этом случае понадобится составной ключ, включающий первичный ключ исходного отношения и идентифицирующие атрибуты агрегатов.

Дальнейший процесс нормализации отношений связан с анализом функциональных зависимостей, существующих в отношении. Пусть X и Y – атрибуты некоторого отношения. Если в любой момент времени каждому значению X соответствует единственное значение Y, то Y **функционально зависит** от X ($X \rightarrow Y$). В нормализованном отношении все неключевые атрибуты функционально зависят от ключа отношения. Говорят, что

неключевой атрибут функционально полно зависит от составного ключа, если он функционально зависит от ключа, но не находится в функциональной зависимости ни от какой части составного ключа.

Вторая нормальная форма (2НФ).

Отношение находится во 2НФ, если оно находится в 1НФ и каждый неключевой атрибут функционально полно зависит от ключа.

Для того чтобы привести отношение ко 2НФ, нужно:

- построить его проекцию, исключив атрибуты, которые не находятся в функционально полной зависимости от составного ключа;
- построить дополнительно одну или несколько проекций на часть составного ключа и атрибуты, функционально зависящие от этой части ключа.

Особое значение при анализе функциональных зависимостей имеют транзитивные зависимости. Пусть X, Y, Z – атрибуты некоторого отношения R . При этом $X \rightarrow Y$ и $Y \rightarrow Z$. Тогда говорят, что Z транзитивно зависит от X .

Третья нормальная форма (3НФ).

Отношение находится в 3НФ, если оно находится во 2НФ и каждый неключевой атрибут нетранзитивно зависит от первичного ключа.

Для того чтобы привести отношение к 3НФ, нужно:

- построить его проекцию, исключив транзитивно зависящие от ключа атрибуты;
- построить дополнительно одну или несколько проекций на детерминанты исходного отношения и атрибуты, функционально зависящие от них.

Для получения качественной РБД в большинстве случаев достаточно приведение её схемы в 3НФ, при которой все её отношения также находятся в 3 НФ.

Нормализация отношений позволяет сократить дублирование данных, но появление новых отношений порождает проблему поддержки семантической целостности данных. Это требует разработки

дополнительных процедур поддержки целостности данных средствами СУБД. При выполнении курсового проекта необходимо провести анализ функциональных зависимостей отношений модели БД и привести их в 3НФ. Для обеспечения целостности БД требуется разработать триггеры, обеспечивающие согласованное изменение данных, находящихся в разных отношениях.

4.3. Физическое проектирование БД

Этап физического проектирования предполагает реализацию логической структуры БД с помощью физической среды хранения. В рамках курсового проекта не осуществляется физическое проектирование. Эта задача решается выбором СУБД, в которой реализуется логическая модель БД.

4.4. Проектирование БД средствами AllFusion ERwin Data Modeler

4.4.1. Общие сведения о AllFusion ERwin Data Modeler

Для построения ER-модели предметной области используется CASE-средство AllFusion ERwin Data Modeler (ранее ERwin). Это CASE-средство используется для проектирования и документирования БД. AllFusion ERwin Data Modeler позволяет наглядно отображать сложные структуры данных. Это средство имеет удобный в использовании графический интерфейс, который упрощает разработку базы данных и автоматизирует множество трудоемких задач. Использование AllFusion ERwin Data Modeler уменьшает сроки создания качественных и производительных реляционных баз данных. Применение средств системы существенно повышает эффективность деятельности разработчиков информационных систем и позволяет получить следующие преимущества:

- существенное повышение скорости разработки за счет мощного редактора диаграмм, автоматической генерации базы данных, автоматической подготовки документации;

- автоматическое создание SQL-предложений для описания схемы БД;
- возможность легко вносить изменения в модель при разработке и расширении системы;
- возможность автоматической подготовки отчетов по базе данных;
- предоставление разработчикам прикладного программного обеспечения наглядной и полной документацией БД;
- тесная интеграция со средствами 4GL позволяет уже на стадии информационного моделирования задавать отображение данных в приложениях;
- поддержка обратного проектирования позволяет документировать и вносить изменения в существующие БД и осуществлять перенос модели БД из одной СУБД в другую;
- поддержка однопользовательских СУБД позволяет использовать для персональных систем современные технологии, что значительно упрощает переход от настольных систем к системам в технологии клиент-сервер (upsizing).

AllFusion ERwin Data Modeler позволяет проектировать модели двух типов: логическую и физическую. Для переключения между логической и физической моделью данных служит список выбора в центральной части панели инструментов (рис. 12). Логическая модель отражает абстрактный взгляд на данные в предметной области и является по существу инфологической моделью предметной области. В ней данные представляются так, как выглядят в реальном мире, и могут называться так, как они называются в реальном мире, например "Постоянный клиент", "Отдел" или "Фамилия сотрудника". Логическая модель AllFusion ERwin Data Modeler данных является универсальной и не связана с возможностью реализации в конкретной СУБД. Элементами логической модели являются сущности и отношения между ними.

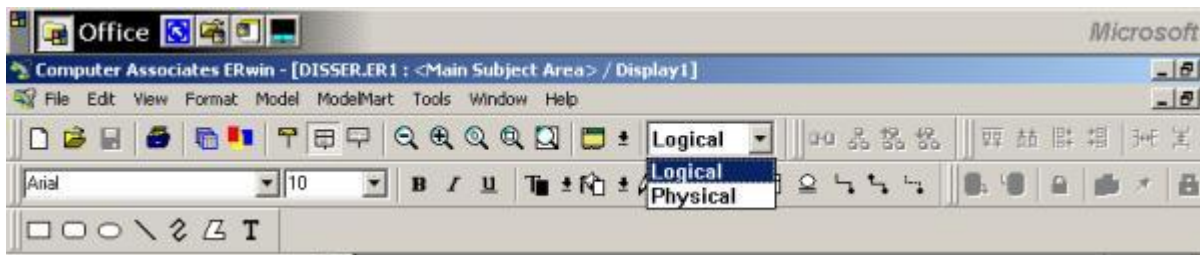


Рис. 12. Выбор типа модели в AllFusion ERwin Data Modeler.

Каждому уровню отображения модели соответствует своя палитра инструментов. На **логическом уровне** палитра инструментов имеет следующие кнопки (рис. 13):





- кнопку указателя (режим мыши) — в этом режиме можно установить фокус на каком-либо объекте модели ;
- кнопку внесения *сущности* ;
- кнопку категории (категория, или категориальная *связь*) — специальный тип *связи* между *сущностями* ;
- кнопки создания *связей*: идентифицирующую, "многие-ко-многим" и неидентифицирующую .



Рис. 13. Панель инструментов логической модели AllFusion ERwin Data Modeler.

4.4.2. Создание логической модели данных в AllFusion ERwin Data Modeler

Различают три уровня отображения логической модели, отличающихся по глубине представления информации о данных:

- *диаграмма сущность-связь (Entity Relationship Diagram, ERD)*;
- *модель данных, основанная на ключах (Key Based model, KB)*;
- *полная атрибутивная модель (Fully Attributed model, FA)*.

Диаграмма сущность-связь представляет собой модель данных верхнего уровня. Она включает только название сущностей и взаимосвязи, отражающие основные объекты предметной области. Её полезно использовать для презентаций и обсуждения структуры данных.

Модель данных, основанная на ключах, является более подробным представлением данных, которое включает описание всех сущностей и первичных ключей.

Полная атрибутивная модель — наиболее детальное представление структуры данных и включает все сущности, атрибуты, ключи и связи. Переключиться между первыми тремя уровнями можно с использованием кнопок панели инструментов. Переключиться на другие уровни отображения можно при помощи контекстного меню, которое появляется, если щелкнуть по любому свободному месту диаграммы. В контекстном меню следует выбрать пункт *Display Level* (Рис. 14.) и затем — необходимый уровень отображения.

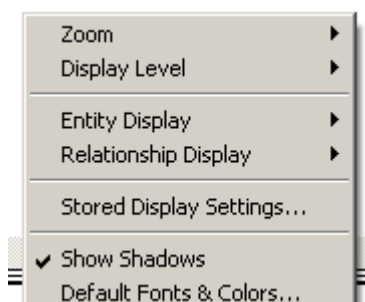


Рис. 14. Выбор уровней отображения диаграммы

Основными элементами диаграммы AllFusion ERwin Data Modeler являются сущности, атрибуты и связи. Каждая сущность является множеством подобных объектов, называемых экземплярами. Каждый экземпляр индивидуален и должен отличаться от всех остальных экземпляров. Для каждой сущности необходимо задать имя (*Name*) и текстовое описание в закладке *Definition*. Закладки *Note*, *Note2*, *Note3*, *UDP* (User Defined Properties - свойства, определенные пользователем) служат для внесения дополнительных комментариев и определений сущности. В закладке *Icon* каждой сущности можно поставить в соответствие

изображение (файл bmp), которое будет отображаться в режиме просмотра модели на уровне иконок. Для того, чтобы создать новую сущность, достаточно нажать соответствующую кнопку панели инструментов.

Атрибут выражает определенное свойство объекта. С точки зрения БД (физическая модель) сущности соответствует таблица, экземпляру сущности — строка в таблице, а атрибуту — колонка таблицы. Каждый атрибут хранит информацию об определенном свойстве сущности. Каждый экземпляр сущности должен быть уникальным. Атрибут или группа атрибутов, которые идентифицируют сущность, называется первичным ключом.

Построение модели данных предполагает определение сущностей и атрибутов, т. е. необходимо определить, какая информация будет храниться в конкретной сущности или атрибуте. Сущность можно определить как объект, событие или концепцию, информация о которых должна сохраняться. Сущности должны иметь имя с четким смысловым значением. Имя задается существительным в единственном числе, например, сущность Студент (но не Студенты!) с атрибутами Номер студента, Фамилия студента и Адрес студента. Каждая сущность должна быть полностью определена с помощью текстового описания, которое определяет имя сущности и набор атрибутов, задающих свойства сущности.

Очень важно дать атрибуту правильное имя. Атрибуты должны именоваться в единственном числе и иметь четкое смысловое значение. Соблюдение этого правила позволяет частично решить проблему нормализации данных уже на этапе определения атрибутов. Для определения атрибута необходимо задать его имя, тип принимаемого значения, дополнительные свойства (рис. 15). Если атрибут является первичным ключом сущности, необходимо отметить в этом окне свойство **Primary Key**.

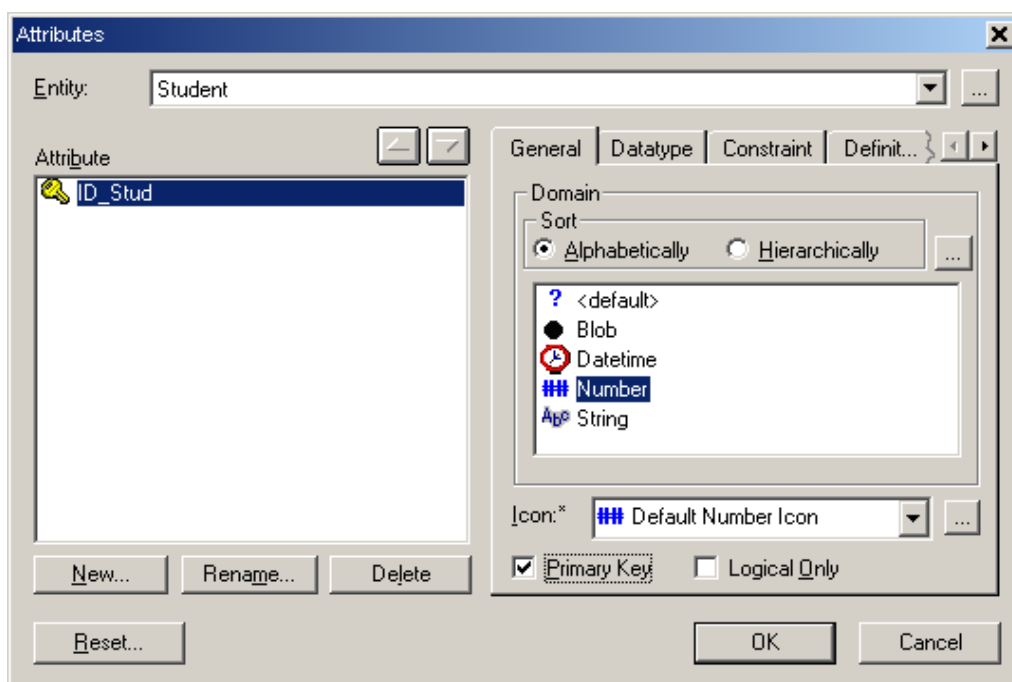


Рис. 15. Задание свойств для атрибутов сущности.

Связь указывает, что между сущностями существует смысловая зависимость. Каждая связь должна именоваться глаголом или глагольной фразой. Имя связи выражает некоторое ограничение или бизнес-правило предметной области. На логическом уровне можно установить следующие типы связи:

- идентифицирующую связь "один-ко-многим",
- связь "многие-ко-многим"
- неидентифицирующую связь "один-ко-многим".

AllFusion ERwin Data Modeler различают зависимые и независимые сущности. Тип сущности определяется ее связью с другими сущностями. Идентифицирующая связь устанавливается между независимой (родительский конец связи) и зависимой (дочерний конец связи) сущностями. При использовании идентифицирующей связи дочерняя сущность автоматически преобразуется в зависимую, которая изображается прямоугольником со скругленными углами. Экземпляр зависимой сущности определяется только через отношение к родительской сущности, т.е. не он может существовать, если для него нет экземпляра родительской сущности. При установлении идентифицирующей связи атрибуты первичного ключа

родительской сущности автоматически переносятся в состав первичного ключа дочерней сущности. Эта операция дополнения атрибутов дочерней сущности при создании связи называется миграцией атрибутов. В дочерней сущности новые атрибуты помечаются как внешний ключ — FK (рис. 16).

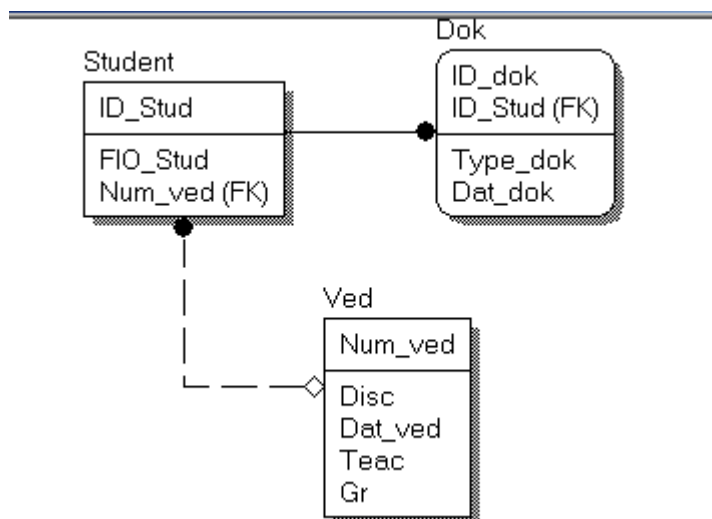


Рис. 16. Фрагмент логической модели в AllFusion ERwin Data Modeler

Неидентифицирующая связь служит для связывания независимых сущностей. Если задается неидентифицирующая связь, дочерняя сущность остается независимой, а атрибуты первичного ключа родительской сущности мигрируют в состав неключевых компонентов родительской сущности.

Идентифицирующая связь показывается на диаграмме сплошной линией с жирной точкой на дочернем конце связи, неидентифицирующая – пунктирной (рис. 13).

Для каждой идентифицирующей и неидентифицирующей связи задается мощность и имя связи. Имя связи (**Verb Phrase**) — фраза, характеризующая отношение между родительской и дочерней сущностями. Мощность связей (**Cardinality**) — служит для обозначения отношения числа экземпляров родительской сущности к числу экземпляров дочерней. Мощность задается одним из четырех типов:

- общий случай, когда одному экземпляру родительской сущности соответствуют 0, 1 или много экземпляров дочерней сущности; такая связь не помечается каким-либо символом;

- символом **P** помечается случай, когда одному экземпляру родительской сущности соответствуют 1 или много экземпляров дочерней сущности (исключено нулевое значение);
- символом **Z** помечается случай, когда одному экземпляру родительской сущности соответствуют 0 или 1 экземпляр дочерней сущности (исключены множественные значения);
- цифрой помечается случай точного соответствия, когда одному экземпляру родительской сущности соответствует заранее заданное число экземпляров дочерней сущности.

4.4.3. Создание физической модели в AllFusion ERwin Data Modeler

Физическая модель данных AllFusion ERwin Data Modeler зависит от конкретной СУБД, фактически являясь отображением системного каталога. Физическая модель является реляционной моделью БД. В физической модели содержится информация обо всех объектах реляционной БД: таблицы, связи, триггеры. Так как не существует стандартов на объекты реляционной БД (например, нет стандарта на типы данных), физическая модель зависит от конкретной реализации СУБД.

Физическая модель данных может быть создана средством AllFusion ERwin Data Modeler автоматически, если в окне типа модели выбрать значение **Physical**. При этом генерируются имена таблиц и индексов физической модели по шаблону на основе имен соответствующих сущностей и ключей логической модели. Эти имена можно изменить вручную. Сформированная модель является описанием реляционной модели БД. При генерации схемы физической БД в СУБД AllFusion ERwin Data Modeler автоматически создает индекс на основе первичного ключа каждой таблицы.

Однако средствами AllFusion ERwin Data Modeler можно построить реляционную модель, которая находится во второй нормальной форме. Этого недостаточно для получения качественной реляционной модели. Поэтому необходимо проверить, соответствует ли полученная модель требованиям

третьей нормальной формы (3НФ), и при необходимости внести в неё необходимые изменения. Для этого необходимо выделить и проанализировать функциональные зависимости для каждой таблицы БД.

Модель реляционной базы данных содержит не только структурную, так и семантическую информацию. Структура базы данных определяется числом и видом включенных в нее отношений. Семантика (смысл) отношений определяется множеством функциональных зависимостей, существующих между атрибутами этих отношений. Дадим определение функциональной зависимости: если даны два атрибута X и Y некоторого отношения, то говорят, что Y функционально зависит от X, если в любой момент времени каждому значению X соответствует ровно одно значение Y. Корректной считается такая схема базы данных, в которой отсутствуют нежелательные функциональные зависимости. Для приведения модели БД к требуемому виду при выполнении курсового проекта необходимо повести декомпозицию тех отношений (таблиц), которые не отвечают требованиям 3НФ.

На этапе проектирования реляционной модели БД необходимо определить правила целостности, которые должны быть отражены в этой модели. Для этого в среде AllFusion ERwin Data Modeler при определении физической модели можно задать правила валидации для атрибутов (столбцов), а также значение, присваиваемое им по умолчанию. Правило валидации задает список допустимых значений для конкретной колонки и/или правила проверки допустимых значений.

AllFusion ERwin Data Modeler существуют средства поддержки правил целостности данных на уровне таблицы: проверка допустимости значений и начальных значений (по умолчанию). Для этого используется специальный диалог (рис. Хх). При задании правила или умолчания для клиентской части эти атрибуты переносятся в репозиторий средства 4GL.

AllFusion ERwin Data Modeler имеет также собственный макроязык для создания прототипов триггеров и процедур. Он позволяет подготовить

шаблоны для различных типов триггеров (например, триггер, реализующий логику каскадного удаления - ON DELETE CASCADE). В AllFusion ERwin Data Modeler определены базовые шаблоны, но пользователь может определить свои собственные шаблоны и использовать их вместо стандартных.

На основе физической модели ERwin может сгенерировать системный каталог БД в заданной СУБД (или соответствующий SQL-скрипт рис. 18). Этот процесс называется прямым проектированием (*Forward Engineering*). Для реализации этого процесса необходимо выполнить команду *Tools/Forward Engineer* (рис. 17).

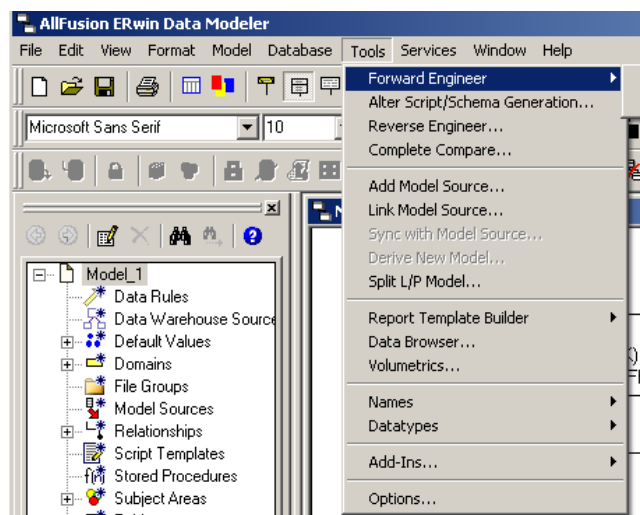


Рис. 17. Прямое проектирование.

Физический уровень представления модели зависит от выбранного сервера. AllFusion ERwin Data Modeler поддерживает более 20 СУБД, среди них:

- Oracle,
- Access,
- DB2,
- SQL Server,
- Teradata,
- Sybase,
- Informix.

Результат создания описания схемы БД приведен на рис. 18. Для создания системного каталога необходимо выбрать команду *Tools/ Forward Engineer/ Schema Generation* или нажать кнопку *Generate* в открывшемся диалоговом окне (рис. 18). Системный каталог в выбранной СУБД определяет состав и структуру таблиц, созданных на основе физической модели, а также связи между ними.

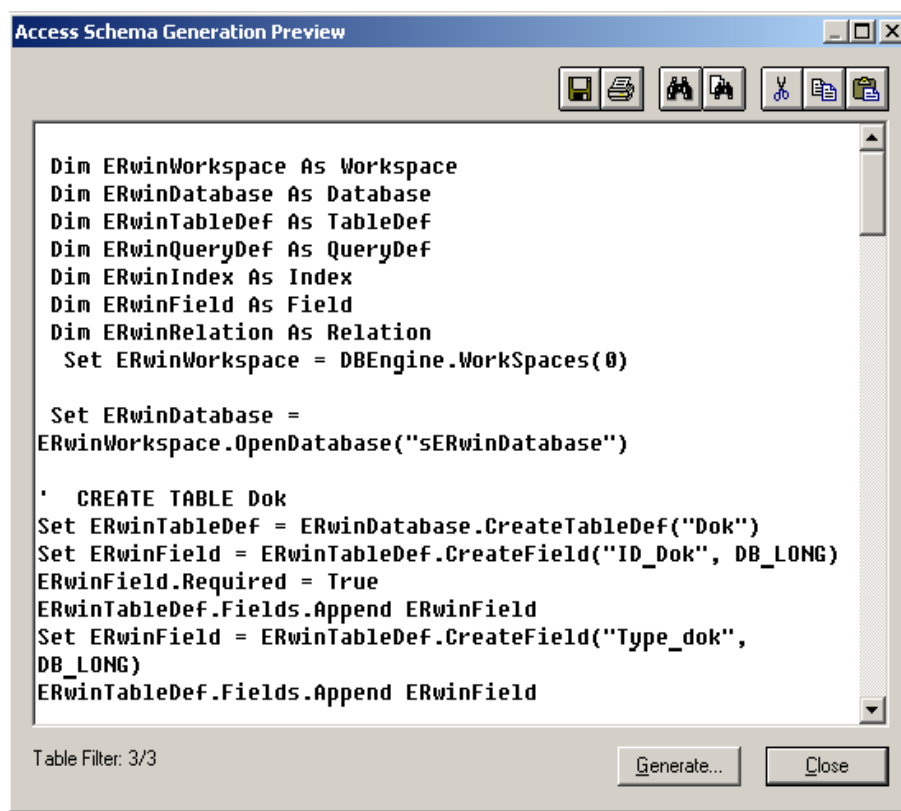


Рис. 18. Результат прямого проектирования для СУБД SQL Server

4.4.4. Создание отчета в AllFusion ERwin Data Modeler

По завершении работы над информационной моделью БД, необходимо подготовить её документацию. Для этого в AllFusion ERwin Data Modeler используют средства автоматически создаваемого отчета, который позволяет распечатывать логический и физический уровни диаграммы, а также отчет по соответствиям сущность-таблица, атрибут-имя колонки, сущность-атрибуты. Диаграмма физической модели является необходимым, почти достаточным и очень удобным материалом для разработчиков программ. Дополнительная

информация для группы разработчиков прикладных программ содержится в отчете "имена таблиц и колонок", который может быть легко построен с помощью генератора отчетов. Создание отчетов в AllFusion ERwin Data Modeler аналогично созданию отчета в AllFusion Process Modeler, которое было описано в разделе 3.3.

5. РАЗРАБОТКА ИНТЕРФЕЙСА С БД.

Разработка интерфейса предполагает реализацию программ на языке высокого уровня, которые обеспечивают выполнение основных функций по обработке данных из БД. Для этого следует выполнить проектирование функциональной структуры интерфейса, которая отражает состав программных компонентов приложения и связь между выделенными компонентами. В ней должны быть отражены компоненты интерфейса с пользователем, компоненты, отражающие выполнение основных процедур обработки данных. Автоматизированные функции управления, реализованные в ИС, могут быть разбиты на функциональные модули. Модульный подход к построению и внедрению ИС обеспечивает простоту и экономичность развития и модернизации ИС.

Разработанный проект интерфейса с БД реализуется на основе программных средств разработки приложений (Delphi, C#, VBA и др.). Для этого рекомендуется использовать технологии ADO или ADO.NET. Рассмотрение применения данных технологий выходит за рамки этого методического пособия. Существует большое количество источников, содержащих исчерпывающую информацию по вопросам разработки приложений с БД в этих технологиях. Ссылки на некоторые из них приведены в списке литературы.

6. ПРИМЕР ПРОЕКТИРОВАНИЯ БД

В качестве примера рассмотрим предметную область, связанную с работой кафедры по распределению студентов на преддипломную практику. Этот процесс включает следующие этапы:

1. Формирование списка баз практики;

2. Назначение руководителя практики от кафедры;
3. Назначение руководителя практики от предприятия;
4. Распределение студентов по базам практики;
5. Формирование документов по прохождению практики;
6. Организация контроля прохождения практики руководителем от кафедры;
7. Защита отчёта по практике на кафедре.

Проведем моделирование процесса средствами AllFusion Process Modeler. Контекстная диаграмма процесса представлена на рис. 19.



Рис. 19. Контекстная диаграмма процесса распределения на преддипломную практику

Декомпозиция контекстной диаграммы предполагает разработку диаграмм этапов, определенных для заданной предметной области. На рис. 20 отражены все основные этапы распределения студентов на преддипломную практику, а на рис. 21 - 22 представлены диаграммы этапов «Формирование списка баз практики» и «Назначение руководителя практики от кафедры». Аналогично строятся диаграммы для всех процессов, отраженных на рис. 20.

Для выделения состава объектов предметной области, информация о которых должна содержаться в БД необходимо провести анализ описанных

процессов. Для проектирования БД особый интерес представляют данные, отражающие информационные взаимосвязи процессов. Для получения этих данных воспользуемся описанием моделей, которое составлено средствами AllFusion Process Modeler. Фрагмент отчета представлен в табл. 1.

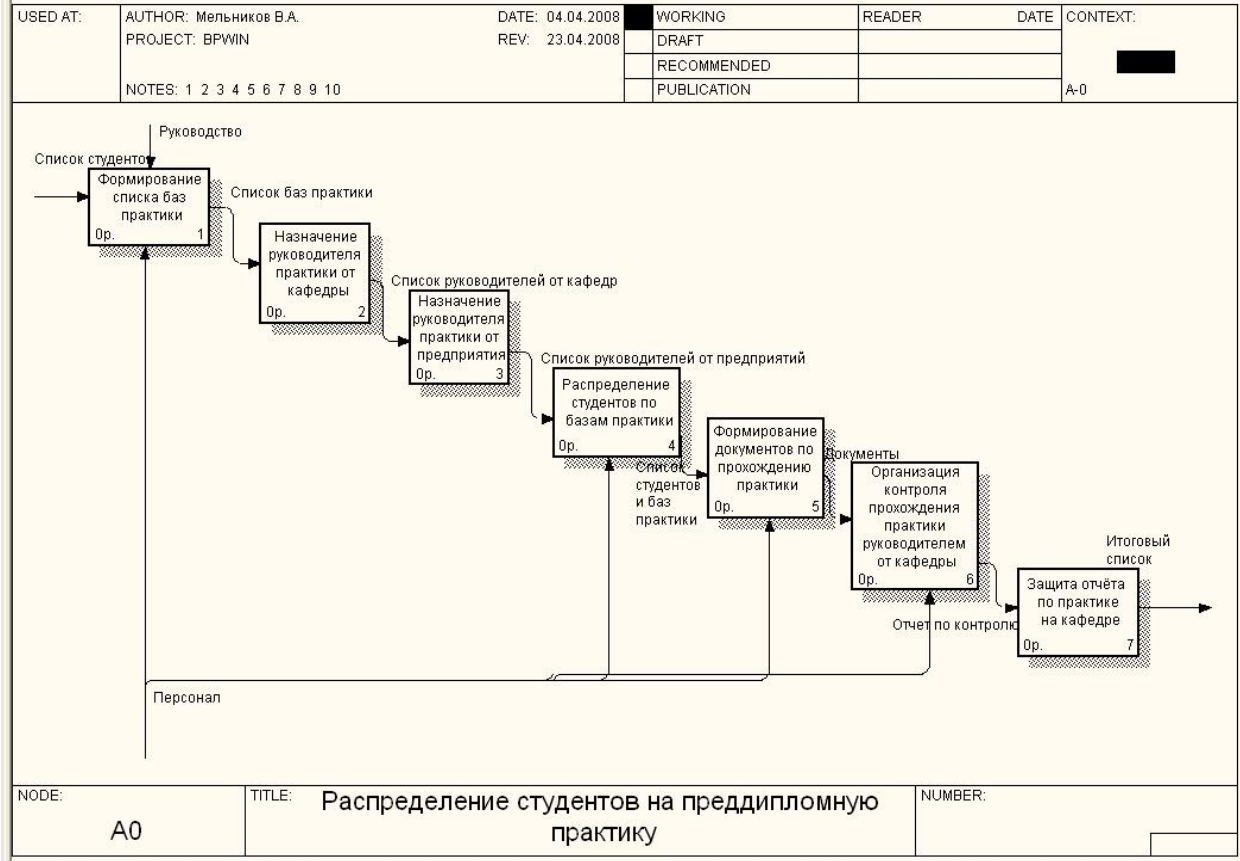


Рис. 20. Декомпозиция контекстной диаграммы процесса

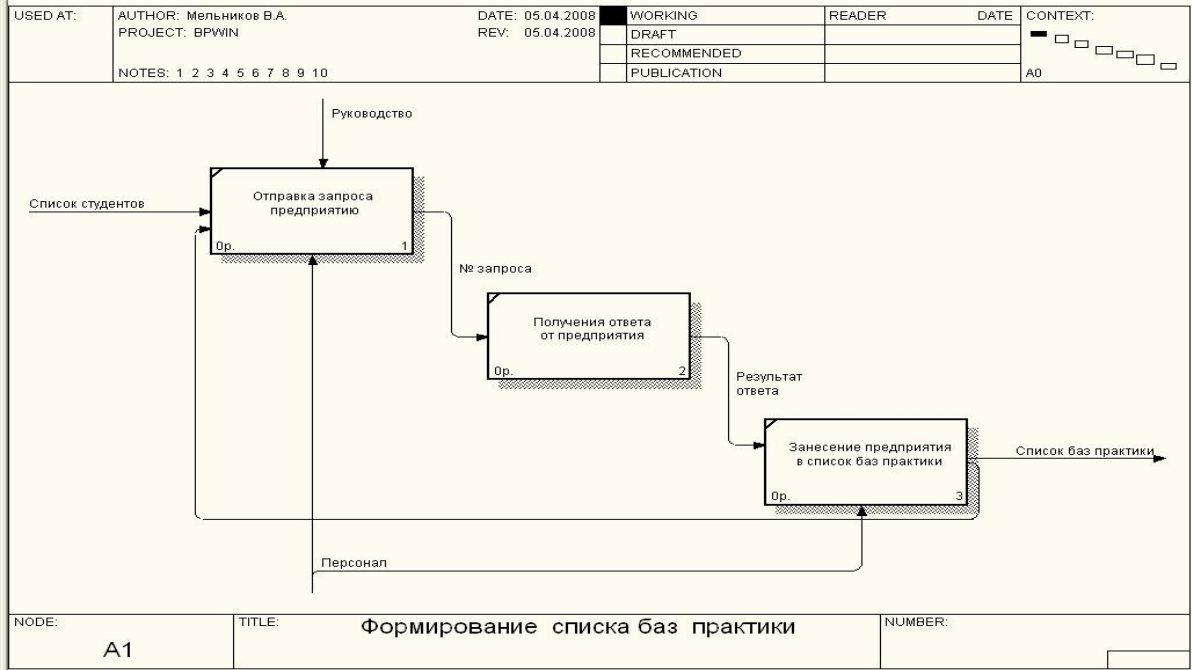


Рис. 21. Декомпозиция процесса «Формирование списка баз практики»

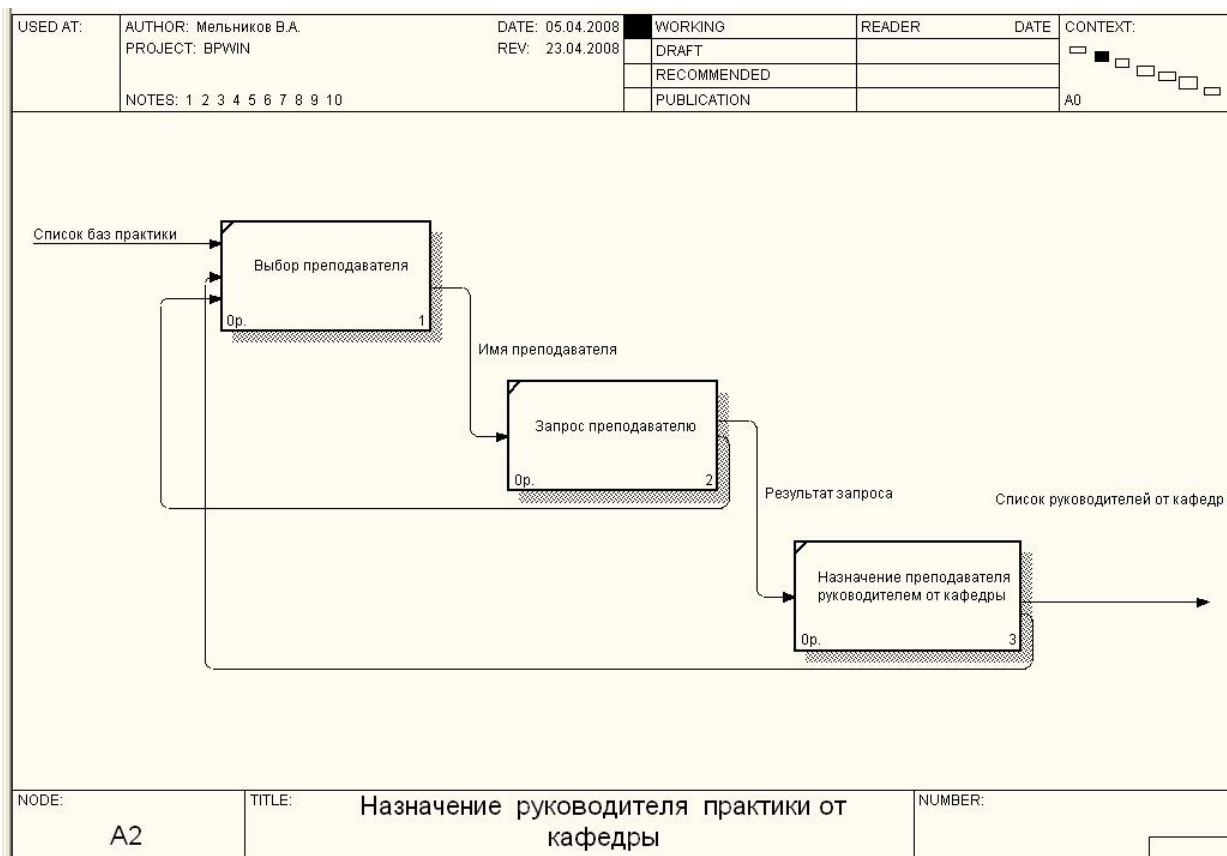


Рис.22. Декомпозиция процесса «Назначение руководителя практики от кафедры»

Таблица 1

Имя объекта	Описание объекта
№ запроса	Номер запроса
Документ	Номера документа Дата подписи ФИО ответственного исполнителя
Руководитель1	ФИО, кафедра, факультет
Руководитель2	ФИО, предприятие
Оценка	Оценка за практику
Студент	Информация о студентах: ФИО, номер, пол, дата рождения, факультет, кафедра, семейное положение

На основе анализа отчета по модели можно выделить следующие объекты предметной области, сведения о которых должны содержаться в БД:

- студент

- факультет
- руководитель1
- руководитель2
- отчет.

Следующим этапом проектирования БД является разработка ER-модели средствами AllFusion ERwin Data Modeler. Результат построения ER-модели БД для выбранной предметной области средствами AllFusion ERwin Data Modeler приведен на рис. 23.

Завершающим этапом проектирования БД является построение реляционной модели и приведение её к 3НФ. Для этого проводим анализ функциональных зависимостей каждого отношения в схеме БД. Пример анализа функциональных зависимостей для отношения «Студент» приведен на рис. 24.

Созданная БД используется для поиска, изменения, удаления и добавления данных. Для организации работы с БД разработана программа,

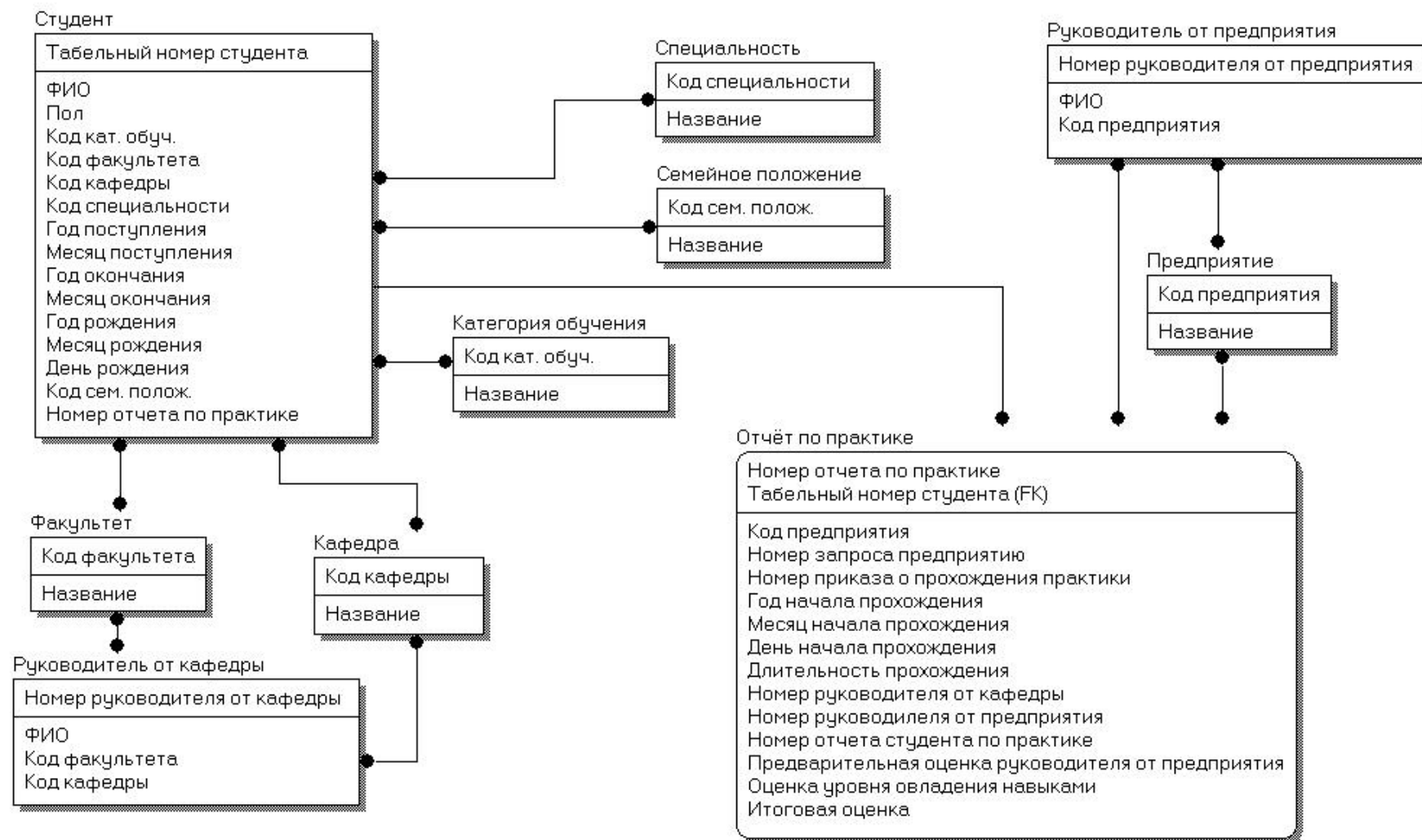


Рис. 23. ER-модель БД

Сущность **Студент**

Ключ: **Номер студента**

Номер студента → ФИО студента

Номер студента → Пол

Номер студента → Код категории обучения

Номер студента → Код факультета

Номер студента → Код кафедры

Номер студента → Код специальности

Номер студента → Год поступления

Номер студента → Месяц поступления

Номер студента → День поступления

Номер студента → Месяц окончания

Номер студента → Месяц окончания

Номер студента → Год рождения

Номер студента → Месяц рождения

Номер студента → День рождения

Номер студента → Код семейного положения

Номер студента → Номер отчёта

Рис. 24. Функциональные зависимости отношения «Студент»

реализующая пользовательский интерфейс. В структуре интерфейса выделены компоненты, реализующие основные операции по работе с таблицами: поиск, удаление, изменение и добавление данных в отдельную таблицу. Пример работы с БД приведен на рис. 25. Для внесения изменений, добавления или удаления записей нужно выбрать соответствующую форму из меню. Интерфейс у всех остальных форм, определяющих работу с другими таблицами БД, поэтому рассмотрим изменение данных на примере одной таблицы - руководитель от кафедры.

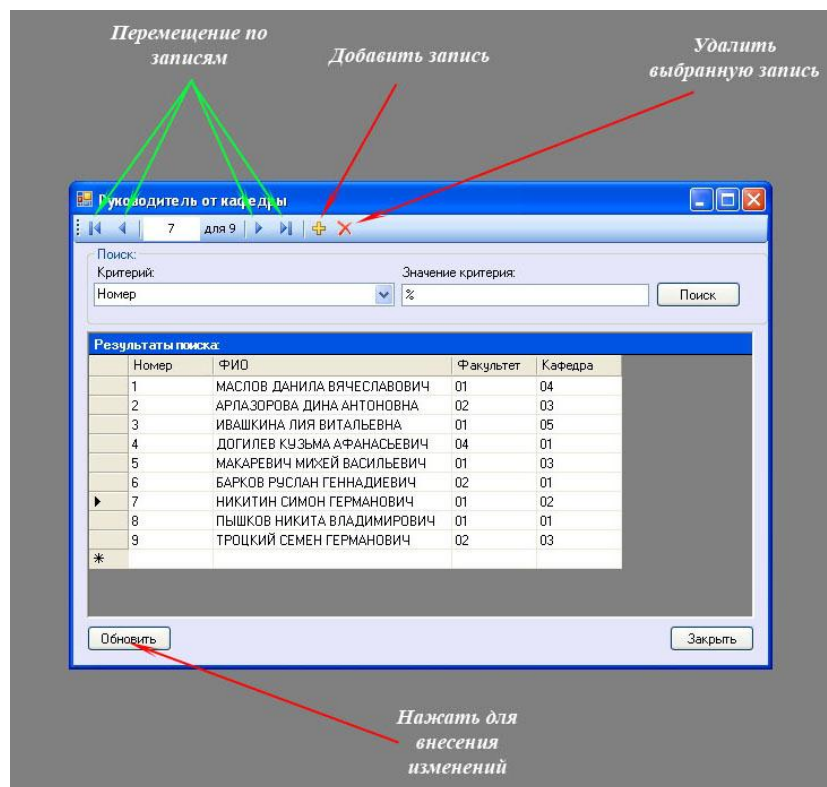


Рис. 25. Пример интерфейса с БД

Библиографический список

1. Маклаков С.В. Создание информационных систем с AllFusion Modeler Suite. – М.: ДИАЛОГ-МИФИ, 2005 – 432 с.
2. Хомоненко А.Д., Гофман В.Э. Работа с базами данных в Delphi. – СПб.: БХВ-Петербург, 2005. – 640 с.
3. Орлов С.А. Технология разработки программного обеспечения. – СПб.: Питер. 2003. – 480 с.
4. Назаров С.В., Мельников П.П. Программирование на MS VISUAL BASIC. – М.: Финансы и статистика, 2002. – 340 с.
5. Пирогов В.Э. Microsoft SQL Server 2005: программирование клиент-серверных приложений. – СПб.: БХВ-Петербург, 2006. – 336 с.
6. Маран, М. М. Программирование на языке C# в среде Microsoft Visual Studio 2005 /. М.: МЭИ, 2007
7. *[www. aspnetmania.com](http://www.aspnetmania.com)*
8. *<http://www.gotdotnet.ru>*
9. Постолинт А. Visual Studio .NET: разработка приложений баз данных
Издательство: БХВ-Петербург, 2003 г. - 544 стр.
10. Баркер Скотт Ф. Создание приложений баз данных в среде Visual Baisic .NET и ADO .NET: Советы, рекомендации, примеры. М.: Вильямс, 2003 г.
- 550 стр.
11. Павловская, Т. А. C/C++. Программирование на языке высокого уровня.
СПб. : Питер, 2007 . - 461 с.

Учебное издание

Сидорова Наталья Петровна

Методические указания к курсовому проекту по дисциплине
«Базы данных».
Методическое пособие.
для студентов, обучающихся по направлениям Информатика и вычислительная техника,
Прикладная математика и информатика

Редактор издательства

Темплан издания МЭИ 2009 (), метод.	Подписано к печати
Формат 60×84/16	Печать офсетная
Физ. печ.л	
Тираж экз.	Изд. №
Заказ №	

ЗАО «издательский дом МЭИ», 111250, Москва, ул. Красноказарменная, д. 14
Отпечатано в типографии ФГУП «НИИ «Геодезия», 141292, Московская область,
Г. Красноармейск, пр-т Испытателей, д.14.