

Министерство образования и науки Украины  
Национальный технический университет  
“ХПИ”

## Проектирование информационных систем

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**  
к лабораторным работам и самостоятельной работе  
по курсам “Управление информационными проектами” и  
“Управление проектами разработки информационных компьютерных  
технологий”  
для студентов специальности 8.000003 “Управление проектами” и  
направления 0804 “Компьютерные науки”

Утверждено  
редакционно-издательским  
советом университета,  
протокол № 1 от 24.06.2010.

Харьков  
НТУ «ХПИ»  
2010

Проектирование информационных систем. Методические указания к лабораторным работам по курсу “Управление информационными проектами” для студентов специальности 8.000003 “Управление проектами” и направления 0804 “Компьютерные науки” / Сост. Н.В. Шатохина. – Х.: НТУ «ХПИ», 2010, – 40 с.

Составитель Н.В. Шатохина

Рецензент А.И. Роговой

Кафедра стратегического управления

## ВСТУПЛЕНИЕ

Методические указания “Проектирование информационных систем” предназначены для студентов специальностей “Компьютерные науки” и “Управление проектами”, изучающих дисциплины “Управление информационными проектами” и “Управление проектами разработки информационных компьютерных технологий”, которые владеют хотя бы одним языком программирования.

Основой данных методических материалов являются принципы моделирования информационных систем (ИС), позволяющие обеспечивать необходимую функциональность системы и учитывать конкретную предметную область, для которой реализуется проект.

Цель данных методических указаний – дать студентам ориентиры в области моделирования и инженерии программного обеспечения; ознакомить студентов с применением инструментальных средств поддержки проектирования информационных систем.

Структурно методические указания состоят из трёх разделов, изложение которых методически подчинено таким аспектам, как основы менеджмента программных проектов, использование BPwin и применение ERwin для построения структурных моделей информационной системы.

Лабораторные работы 1–4 посвящены изложению основных принципов построения диаграмм при помощи CASE средства BPwin 7.3. На примере предметной области «Трудоустройство выпускников кафедры» построены IDEF0, DFD и IDEF 3 диаграммы. Вопросы построения IDEF 1X моделей при помощи CASE средства ERwin 7.3 рассматриваются в лабораторной работе №5.

Рассмотренные модели в совокупности дают достаточно полное описание системы независимо от того, является ли она существующей или вновь разрабатываемой. Состав диаграмм в каждом конкретном случае зависит от необходимой полноты описания системы.

Необходимо отметить, что проведение структурного и объектно-ориентированного анализа существенно снижает риски при создании сложной программной системы.

## Лабораторная работа № 1

### ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

*Цель лабораторной работы:* разработать описание системы “Трудоустройство студентов кафедры”.

*Ход работы:*

1. Описание назначения и задач, выполняемых системой.
2. Декомпозиция системы на подсистемы.
3. Характеристика функций каждой подсистемы.
4. Описание возможностей, предоставляемых различным категориям пользователей.

Процесс создания диаграмм начинается с этапа изучения предметной области, которая описывается в данной лабораторной работе. Система предназначена для того, чтобы помочь студенту найти работу еще в процессе его обучения в университете. Подав заявку в информационную систему, студент становится ее клиентом и начинает обслуживаться на протяжении своего обучения. Заявка представляет собой анкету.

Особое внимание уделяется обучению студента, по итогам успеваемости преподавателями составляются экспертные оценки. На основе собранной информации составляется резюме, представляющее собой полную характеристику человека. Это резюме отсылается всем организациям, имеющим необходимые вакансии.

Основным назначением системы является автоматизация ввода и хранения отчетных данных по студентам, составление характеристик и резюме, поиска вакансий в фирмах. Система позволяет изменять, дополнять, вести поиск и просмотр информации о студентах, накладывать ограничения доступа к системе, хранить списки студентов, закончивших обучение, в виде архива, контролировать выдачу студенту заданий на курсовые работы и проекты, связывать институт с фирмами, заинтересованными в поиске сотрудников. Также данная система может быть использована для составления отдельных списков групп, для печати зачетных ведомостей, для печати полной базы данных и для статистики.

Система состоит из ряда подсистем:

1-я: учет успеваемости;

2-я: экспертные оценки преподавателей кафедры;

3-я: определение категорий полномочий пользователей.

Рассмотрим детальнее подсистему учета успеваемости студентов.

Эта подсистема отвечает за статистическую отчетность по успеваемости отдельного студента, группы, а также за хранение и правильность ее ввода.

Исходными данными подсистемы являются: оценки, даты сдачи экзаменов, имена студентов, номера групп, факультет. На выходе подсистема выдает обработанные данные: средний балл по студенту, группе или факультету, процентное соотношение оценок у студента в группе или на факультете, имена и количество стипендиатов в группе или на факультете. Подсистема "Учет успеваемости студентов" может функционировать отдельно от всей системы, что дает возможность установить и использовать ее независимо, если это необходимо. Подсистема "Учет успеваемости студентов" включает следующие функции:

- ввод, вывод и редактирование информации по информационным объектам подсистемы;
- сохранение информации, поступившей от подсистемы контроля успеваемости студентов;
- расчет процентного соотношения оценок у студента в группе или на факультете и вывод его в виде таблиц, графиков и диаграмм;
- расчет среднего балла по студенту, группе или факультету;
- формирование данных по студенту, группе или факультету;
- выявление сильнейших и слабых студентов в группе или на факультете;
- расчет количества факультетских стипендиатов в группе;
- проверка правильности ввода данных [8].

Рассмотрим подсистему обработки запросов и определения категорий пользователей.

Данная подсистема предназначена для определения категории, полномочий и обработки запросов пользователей службы занятости. В частности, она выполняет следующие функции:

- регистрацию новых фирм;
- регистрацию новых студентов;
- определение прав доступа зарегистрированного пользователя;
- обработку запросов;
- прием регистрационных данных от фирм;
- прием регистрационных данных от студентов;
- составление резюме;
- запись данных в БД студентов;

- запись данных в БД фирм.

В соответствии с выполняемыми функциями система работает со следующими данными: регистрационными данными студентов; регистрационными данными фирм; личными данными студентов; информацией о студентах (получаемой фирмами); информацией о фирмах (получаемой студентами); идентификационными данными пользователей; информацией о системе; запросом; служебной информацией (для обслуживающего персонала); результатами тестирования; экспертными оценками преподавателей.

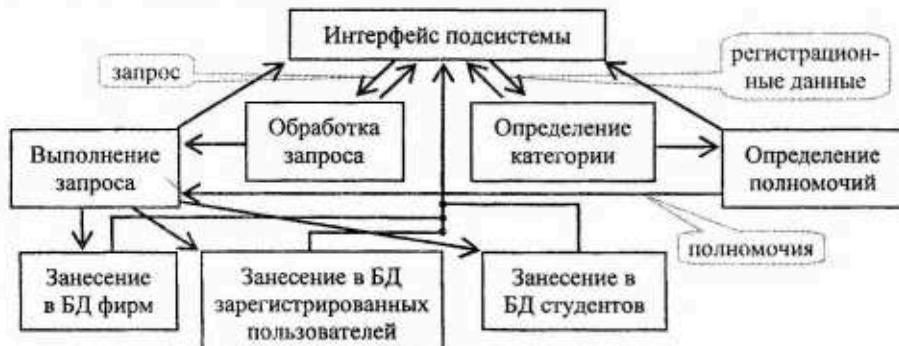


Рисунок 1.1 – Схематическое представление модульной структуры подсистемы обработки запросов и определения категорий пользователей

#### Рассмотрим подсистему оценок преподавателей кафедры.

Эта подсистема предназначена для установки и просмотра экспертной оценки. Она дает краткую информацию преподавателю о студенте или группе (см. рис. 1.2). Студент может с помощью ее ориентироваться в учебе.

Кроме того, подсистема может дать представителю фирмы информацию о студенте при рассмотрении его кандидатуры в качестве нового сотрудника.

Клиент входит в подсистему с идентификационным номером. По номеру присваивается уровень доступа. Подсистема выдает меню работы, соответствующее уровню доступа.

Модуль, получающий из БД экспертные оценки, проводит их обработку и передает обработанные данные в общую БД.

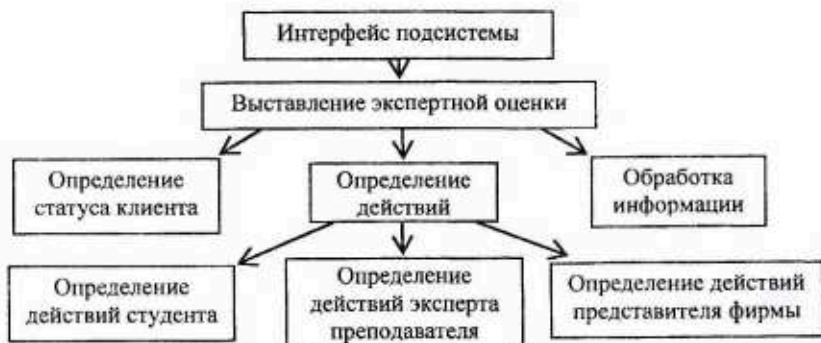


Рисунок 1.2 – Схематическое представление модульной структуры подсистемы экспертных оценок

**База данных содержит в себе:**

- экспертные оценки (предоставляются подсистемой экспертных оценок);
- успеваемость студента (предоставляется подсистемой контроля студенческой успеваемости);
- успеваемость группы (предоставляется подсистемой контроля успеваемости).

**Критерии экспертных оценок:**

- 1) по 12-балльной системе оценить студента в начале, середине, конце семестра;
- 2) по 12-балльной системе оценить степень запоминания курса студентом в начале, середине, конце семестра;
- 3) по 12-балльной системе оценить степень применения знаний студентом в начале, середине, конце семестра.

**Предоставляемые возможности пользователям системы:**

#### 1. Студенту:

- ввод личных анкетных данных;
- просмотр экспертных оценок по отчетным заданиям и результатам тестов;
- просмотр сводных таблиц и графиков;
- получение и сдача контрольных заданий;
- доступ к справочным материалам (данные из службы удаленного обучения, а именно, методическое обеспечение);

- просмотр сообщений и внесение изменений в сообщения доски объявлений;
- поиск вакансий в БД по запросу.

#### 2. Преподавателю (эксперту):

- предоставление экспертной оценки, а также изменение ее;
- просмотр других оценок;
- просмотр программы курса и внесение изменений в нее;
- ввод контрольных заданий и назначение их студенту;
- контроль ответов на задания;
- доступ к интеллектуальным ресурсам;
- составление резюме (характеристик).

#### 3. Администрации университета (заведующему кафедрой, декану и т.д.):

- просмотр программы курса;
- просмотр динамики успеваемости курса, группы, отдельного студента;
- просмотр сводных таблиц и графиков;
- просмотр экспертных оценок и характеристик преподавателей.

#### 4. Администратору:

- определение прав доступа;
- ввод и корректировка системных данных;
- контроль работы системы;
- осуществление контроля защиты системы от несанкционированного доступа;
- изменение физической модели данных системы;
- составление сводных таблиц и графиков;
- заполнение полей БД системы (ввод информации).

#### 5. Фирме:

- поиск в БД данных о студенте по запросу;
- просмотр резюме студентов.

### **Контрольные вопросы**

1. Из каких подсистем состоит «Трудоустройство студентов кафедры»?
2. Укажите цели системы.
3. Изобразите модульную структуру всей системы.
4. Как формируется БД и из чего она состоит?
5. Опишите порядок занесения студента в БД.
6. Перечислите функциональные характеристики системы.
7. Какие данные поступают на вход подсистем?
8. Назовите категории пользователей системы и предоставляемые им возможности.
9. Каково назначение подсистем?
10. Опишите критерии экспертных оценок.

## Лабораторная работа № 2

### МЕТОДОЛОГИЯ IDEF0

*Цель работы:* напоминание основных принципов методологии IDEF0 и создание нового проекта в BPWin для системы трудоустройства студентов кафедры.

*Ход работы:*

1. Создать новую модель.
2. Описать цели моделирования, предметную область и границы, свойства модели.
3. Сформировать контекстную диаграмму по системе согласно методологии IDEF0.
4. Описать главный процесс, нанести входы, выходы, механизмы и управления.
5. Заполнить необходимые поля бланка диаграммы.
6. Декомпозиционировать контекстную диаграмму.
7. Объединить работы, нанести и описать связи.
8. Сохранить проект в отдельный файл.

На начальных этапах создания информационной системы необходимо понять, как работает организация, которую собираются автоматизировать. Поэтому для описания работы организации необходимо построить модель, которая будет адекватна предметной области и будет содержать в себе знания всех участников рассматриваемых бизнес-процессов. Наиболее удобным языком моделирования бизнес-процессов является IDEF0, где система представляется как совокупность взаимодействующих работ или функций. Такая чисто функциональная ориентация является принципиальной – функции системы анализируются независимо от объектов, которыми они оперируют. Это позволяет более четко смоделировать логику и взаимодействие процессов организации.

IDEF0-модель может содержать четыре типа диаграмм [2].

Процесс моделирования системы в IDEF0 начинается с создания контекстной диаграммы – диаграммы наиболее абстрактного уровня описания системы в целом, содержащей определение субъекта моделирования, цели и точки зрения на модель.

Под субъектом понимается сама система, при этом необходимо точно установить, что входит в систему, а что лежит за ее пределами, т.е. определить, что будет в дальнейшем рассматриваться как компоненты

системы, а что как внешнее воздействие. На определение субъекта системы будут существенно влиять позиция, с которой рассматривается система, и цель моделирования. Так будет определена область моделирования.

После построения контекстной диаграммы, проводится функциональная декомпозиция – система разбивается на подсистемы и каждая подсистема описывается отдельно (диаграммы декомпозиции). Затем каждая подсистема разбивается на более мелкие и так далее до достижения нужной степени подробности.

Каждая IDEF0-диаграмма содержит работы и стрелки, которые связывают блоки (работы) вместе и отображают взаимодействия и взаимосвязи между ними.

Работы (Activity) обозначают поименованные процессы, функции или задачи, которые происходят в течение определенного времени и имеют распознаваемые результаты. Работы изображаются в виде прямоугольников. Имя работы должно быть выражено отлагательным существительным, обозначающим действие.

Каждая сторона блока имеет особое, вполне определенное назначение. Левая сторона блока предназначена для входов, верхняя – для управления, правая – для выходов, нижняя – для механизмов. Такое обозначение отражает определенные системные принципы: входы преобразуются в выходы, управление ограничивает или предписывает условия выполнения преобразований, механизмы показывают, что и как выполняет функция.

Блоки в IDEF0 размещаются по степени важности, как ее понимает автор диаграммы. Этот относительный порядок называется доминированием. Доминирование понимается как влияние, которое один блок оказывает на другие блоки диаграммы. Например, самым доминирующим блоком диаграммы может быть либо первый из требуемой последовательности функций, либо планирующая или контролирующая функция, влияющая на все другие. Наиболее доминирующий блок обычно размещается в верхнем левом углу диаграммы, а наименее доминирующий – в правом углу. Таким образом, топология диаграммы показывает, какие функции оказывают большее влияние на остальные.

Стрелки (Arrow) описывают взаимодействие работ и представляют собой некоторую информацию, выраженную существительными. Стрелки представляют собой некую информацию, именуются существительными и

изображаются одинарными линиями со стрелками на концах.

В IDEF0 различают пять типов стрелок.

1) Вход (Input) – объекты, используемые и преобразуемые работой для получения результата (выхода). Допускается, что работа может не иметь ни одной стрелки входа. Стрелка входа рисуется как входящая в левую грань работы.

2) Управление (Control) – информация, управляющая действиями работы. Обычно управляющие стрелки несут информацию, которая указывает, что должна выполнять работа. Каждая работа должна иметь хотя бы одну стрелку управления, которая изображается как входящая в верхнюю грань работы.

3) Выход (Output) – объекты, в которые преобразуются входы. Каждая работа должна иметь хотя бы одну стрелку выхода, которая рисуется как исходящая из правой грани работы.

4) Механизм (Mechanism) – ресурсы, выполняющие работу. Стрелка механизма рисуется как входящая в нижнюю грань работы. По усмотрению аналитика стрелки механизма могут не изображаться на модели.

5) Вызов (Call) – специальная стрелка, указывающая на другую модель работы. Стрелка вызова рисуется как исходящая из нижней части работы и используется для указания того, что некоторая работа выполняется за пределами моделируемой системы.

*Границные стрелки* на контекстной диаграмме служат для описания взаимодействия системы с окружающим миром. Они могут начинаться у границы диаграммы и заканчиваться у работы, или наоборот.

В некоторых случаях удобно использовать механизм туннелирования стрелок. Существует два вида туннелирования связей – со скрытым приемником и со скрытым источником [4].

После изучения исходных документов и опроса заказчиков и пользователей системы необходимо сформулировать цель моделирования и определить точку зрения на модель. Рассмотрим технологию ее построения на примере системы “Трудоустройство выпускников кафедры”, основные возможности которой были описаны в лабораторной работе № 1.

Сформулируем цель моделирования: описать функционирование системы, которое было бы понятно ее пользователю, не вдаваясь в подробности, связанные с реализацией. Модель будем строить с точки зрения пользователей (студент, преподаватель, системный администратор,

деканат, фирма).

Разрабатываемая модель должна содержать контекстную диаграмму и два уровня декомпозиции. Необходимо предусмотреть наличие связей со скрытым источником и скрытым приемником (туннелирование связей) [9].

Начнем с построения контекстной IDEF0-диаграммы. Согласно описанию системы основной функцией является обслуживание ее клиентов посредством обработки запросов, от них поступающих. Таким образом, определим единственную работу контекстной диаграммы: "Обслужить клиента системы". Далее определим входные и выходные данные, а также механизмы и управление.

Для того чтобы обслужить клиента, необходимо зарегистрировать его в системе, открыть доступ к БД и обработать его запрос. В качестве входных данных будут использоваться: "имя клиента", "пароль клиента", "исходная БД", "запрос клиента". Выполнение запроса ведет либо к получению информации от системы, либо к изменению содержащегося БД (например, при составлении экспертных оценок), поэтому выходными данными будут являться "отчеты" и "измененная БД". Процесс обработки запросов будет выполняться монитором системы под контролем администратора.

Определим контекстную диаграмму системы (рис. 2.1).



Рисунок 2.1 – Контекстная диаграмма подсистемы

Проведем декомпозицию контекстной диаграммы, описав последовательность обслуживания клиента:

1. Определение уровня доступа в систему.
2. Выбор подсистемы.
3. Обращение к подсистеме.
4. Изменение БД (при необходимости).

Получим диаграмму, изображенную на рис. 2.2.

Закончив декомпозицию контекстной диаграммы, переходим к декомпозиции диаграммы следующего уровня. Обычно при рассмотрении третьего и более низких уровней модели возвращаются к "родительским" диаграммам и корректируют их.

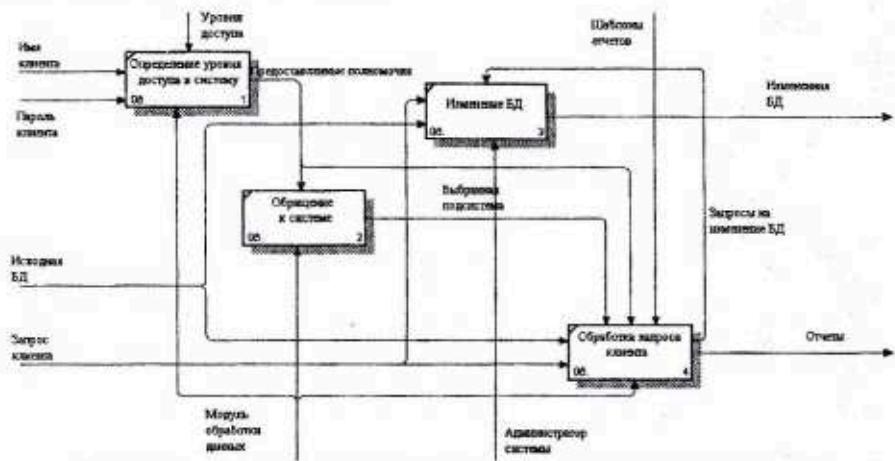


Рисунок 2.2 – Декомпозиция работы “Обслуживание клиента системы”

Декомпозириуем последовательно все блоки полученной диаграммы. Первым этапом при определении уровня доступа в систему является определение категории пользователя. По имени клиента осуществляется поиск в базе пользователей, возможно определить его категорию. Согласно определенной категории выясняются полномочия, предоставляемые пользователю системы. Проводим процедуру доступа в систему, проверяя имя и пароль доступа. Объединяя информацию о полномочиях и уровне доступа в систему, для пользователя формируем набор разрешенных действий. Таким образом, определение уровня доступа в систему будет выглядеть так, как показано на рис. 2.3.

После прохождения процедуры доступа в систему, монитор анализирует запрос клиента, выбирая подсистему, которая будет обрабатывать запрос.

Декомпозиция работы “Обращение к подсистеме” не отвечает цели и точке зрения модели. Пользователя системы не интересуют внутренние алгоритмы ее работы. В данном случае ему важно, что выбор подсистемы будет произведен автоматически, без его вмешательства, поэтому декомпозиция обращения к подсистеме только усложнит модель.

Декомпозириуем работу “Обработка запроса клиента”, выполняемую подсистемой обработки запросов, определения категорий и полномочий пользователей.

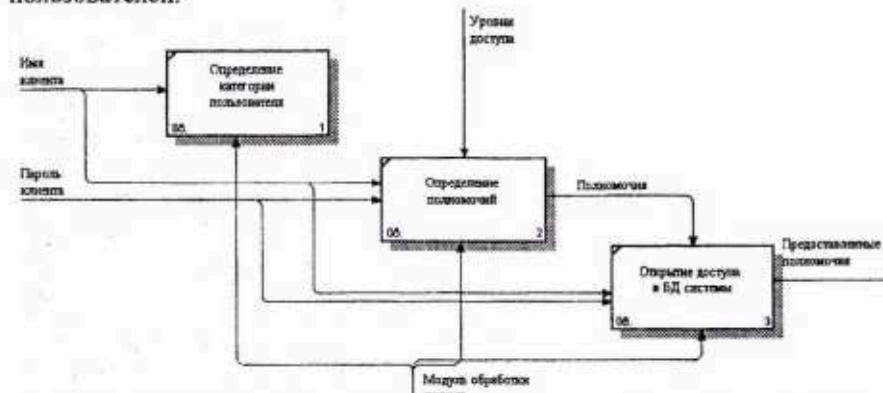


Рисунок 2.3 – Декомпозиция работы “Определение уровня доступа в систему”

Перед осуществлением поиска ответа на запрос необходимо открыть БД (подключиться к ней). В общем случае БД может находиться на удаленном сервере, поэтому может потребоваться установление соединения с ней. Определим последовательность работ: 1) открытие БД; 2) выполнение запроса; 3) генерация отчетов.

После открытия БД необходимо сообщить системе об установлении соединения с БД, после чего выполнить запрос и сгенерировать отчеты для пользователя (рис. 2.4).

Необходимо отметить, что в “Выполнение запроса” включается работа различных подсистем. Например, если запрос включает в себя тестирование, то его будет выполнять подсистема профессиональных и психологических тестов. На этапе выполнения запроса может потребоваться

изменение содержимого БД, например при составлении экспертных оценок. Поэтому на диаграмме необходимо предусмотреть такую возможность.



Рисунок 2.4 – Декомпозиция работы «Обработка запроса клиента»

При анализе полученной диаграммы возникает вопрос, по каким правилам происходит генерация отчетов? Необходимо наличие заранее сформированных шаблонов, по которым будет производиться выборка из БД, причем эти шаблоны должны соответствовать запросам и должны быть заранее определены. Кроме того, клиенту должна быть предоставлена возможность выбора формы отчета.

Туннельная стрелка “Клиент системы” применена для того, чтобы не выносить стрелку на диаграмму верхнего, так как функция выбора формы отчета не является достаточно важной для отображения ее на “родительской” диаграмме.

Декомпозицию работы “Выполнение запроса” целесообразно провести при помощи диаграммы DFD (лабораторная работа № 3), так как методология IDEF0 рассматривает систему как совокупность взаимосвязанных работ, что плохо отражает процессы обработки информации.

Перейдем к декомпозиции последнего блока “Изменение БД”.

Согласно цели моделирования, клиенту важно понимать, что поступившие данные не сразу обновляются в системе, а проходят дополнительный этап обработки и контроля. Алгоритм изменения можно сформулировать следующим образом:

1. Определяется БД, в которой будет изменяться информация.
2. Оператором формируется временный набор данных и предоставляется

администратору.

3. Администратор осуществляет контроль данных и вносит их в БД.

Проведение дальнейшей декомпозиции “Изменения БД” будет усложнять модель, объясняя, как осуществляется физическое изменение БД в системе. При этом пользователь не получит никакой дополнительной информации о работе системы службы занятости. Декомпозицию этой работы целесообразно проводить в процессе проектирования БД системы на этапе создания логической модели БД (см. рис. 2.7).

Декомпозиция работы “Выполнение запроса” будет проведена в следующей лабораторной работе с иллюстрацией применения диаграмм DFD для описания процессов обработки информации.

### Контрольные вопросы

1. Назначение и сущность IDEF0 диаграмм.
2. Направления IDEF0 моделирования.
3. Что обозначают работы в IDEF0? Назовите порядок их наименования.
4. Какое количество работ должно присутствовать на одной диаграмме?
5. Что называется порядком доминирования? Как располагаются работы по принципу доминирования?
6. Назначение сторон функциональных блоков в IDEF0 диаграмме.
7. Каково назначение сторон прямоугольников работ на диаграммах?
8. Перечислите типы стрелок. Назовите виды взаимосвязей.
9. Укажите этапы жизненного цикла ИС, для которых наиболее эффективно использование методологии IDEF0.
10. Что называется граничными стрелками?
11. Объясните принцип именования разветвляющихся и сливающихся стрелок.
12. Назначение туннелирования стрелок.
13. Назначение кнопок палитры инструментов для построения IDEF0 диаграмм.
14. Как провести связь между работами? Как задать имя работы?
15. Опишите процесс декомпозиции работы.
16. Как добавить работу на диаграмму?
17. Как разрешить туннелированные стрелки?
18. Может ли модель BPWin содержать диаграммы нескольких методологий?

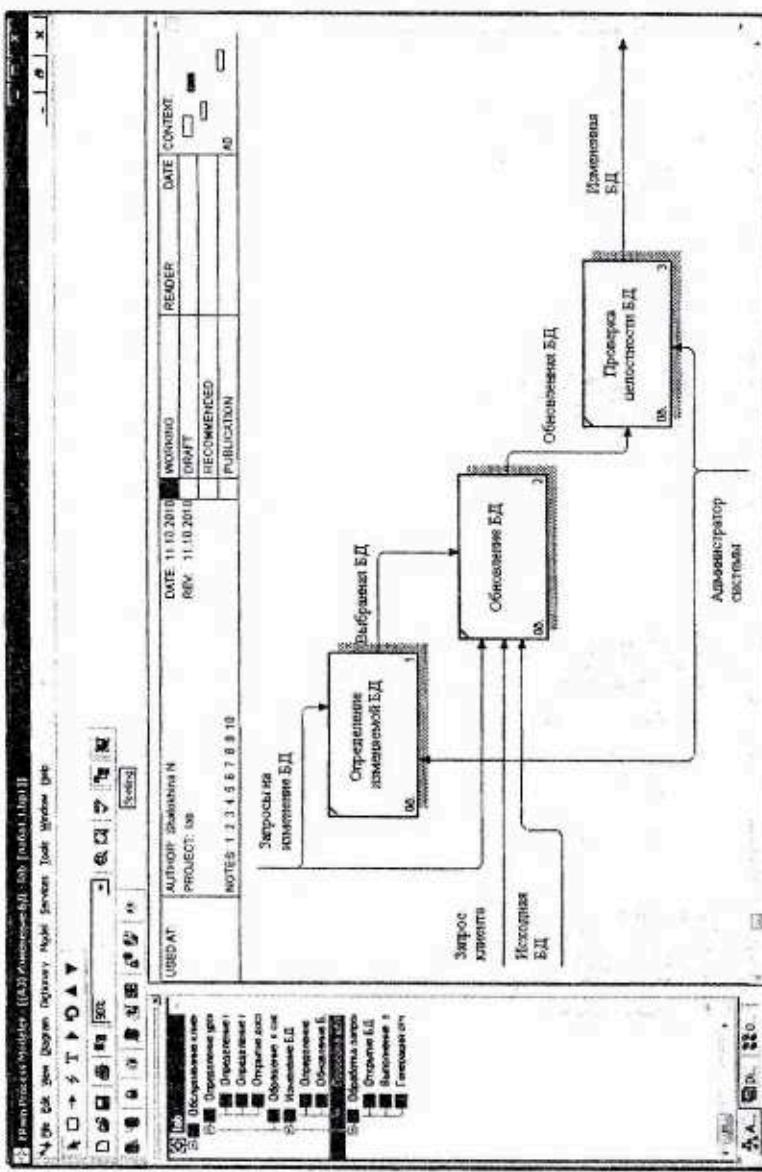


Рисунок 2.7 – Декомпозиция работы «Изменения БД»

## Лабораторная работа № 3

### ПОСТРОЕНИЕ DFD ДИАГРАММ (Data Flow Diagrams)

*Цель работы – построение диаграмм потоков данных (DFD).*

*Ход работы:*

1. Дополнить созданную диаграмму IDEF0 диаграммой DFD.
2. Добавить на диаграмму DFD внешнюю сущность и хранилище данных.
3. Связать диаграмму и внешнюю сущность, связать диаграмму и хранилище.
4. Определить имя связи с внешней сущностью.

Диаграммы DFD используются для представления механизмов передачи и обработки информации в моделируемой системе. Их применение целесообразно на этапе проведения системного анализа при проектировании информационных систем разного уровня сложности.

Часто диаграммы DFD используются в качестве дополнения к функциональной модели, выполненной при помощи диаграмм IDEF0, для отражения отдельных аспектов предметной области [1]. Их удобно использовать для представления сети связанных между собой работ, например, описания документооборота и обработки информации.

Для построения этих диаграмм используется нотация Гейна-Сарсона [3].

Диаграмма 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).

В BPWin нотация Гейна-Сарсона предполагает наличие таких компонентов как “поток данных”, “процесс”, “хранилище данных”, “внешняя сущность” [7].

В палитре инструментов на новой диаграмме DFD появляются новые кнопки:

- External Reference – добавить в диаграмму внешнюю ссылку;
- Data store – добавить в диаграмму хранилище данных;

- Diagram Dictionary Editor – ссылка на другую страницу. В отличие от IDEF0 этот инструмент позволяет направить стрелку на любую диаграмму (а не только на верхний уровень).

В отличие от стрелок IDEF0, которые представляют собой жесткие взаимосвязи, стрелки DFD показывают, как объекты (включая данные) двигаются от одной работы к другой. Это представление потоков совместно с хранилищами данных и внешними сущностями делает модели DFD более похожими на физические характеристики системы

Потоки данных являются механизмами, использующимися 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.

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 собой "резы" 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 нет 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.

В DFD стрелки могут сливаться и разветвляться, что позволяет описать декомпозицию стрелок. Каждый новый сегмент сливающейся или разветвляющейся стрелки может иметь собственное имя.

Рассмотрим работу “Обработка запросов клиента” из лабораторной работы №1. Диаграмма может быть построена с использованием традиционного структурного анализа.

Запросы в систему поступают от пользователей, поэтому запросы от каждой категории будут обрабатываться отдельно. Выделим внешние сущности диаграммы согласно каждой категории пользователей, определяя потоки данных, которыми они обмениваются с системой. Получим диаграмму, изображенную на рис. 3.1.

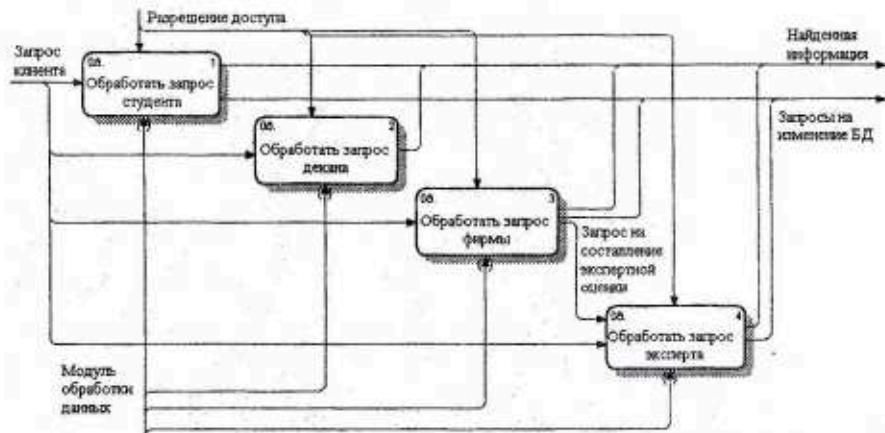


Рисунок 3.1 – DFD-декомпозиция работы “Выполнение запроса”

Согласно описанию системы проведем декомпозицию полученных блоков (рис. 3.2–3.5).

Все процессы обработки запросов контролируются и выполняются модулем обработки информации системы, поэтому стрелка-механизм “Модуль обработки информации” будет повторяться на декомпозированных диаграммах. Точка зрения модели, определенная в лабораторной работе № 1, не требует рассмотрения внутренних особенностей функционирования системы, поэтому затуннелируем данную стрелку с тем, чтобы не переносить ее на диаграммы нижнего уровня.



Рисунок 3.2 – DFD-декомпозиция работы “Обработать запрос студента”

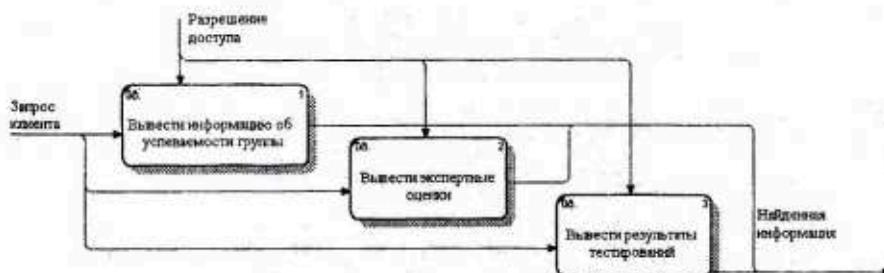


Рисунок 3.3 – DFD-декомпозиция работы “Обработать запрос декана”

Перед проведением экспертной оценки эксперт должен выбрать студента (см. рис. 3.5). Термин “определить” может подразумевать вмешательство эксперта, при котором он на основании каких-либо предпочтений выбирает студента для проведения оценки. На самом деле студент просто выбирается из общего списка. Хотя здесь могут существовать внешние факторы, например требование фирмы на составление экспертной оценки для студента.

Эта ситуация должна отражаться на диаграмме “Обработать запрос фирмы”, т.е. необходимо внести дополнительную работу “Найти экспертные оценки студента”. Причем если такие оценки не найдены, то необходимо генерировать запрос для эксперта на их составление.

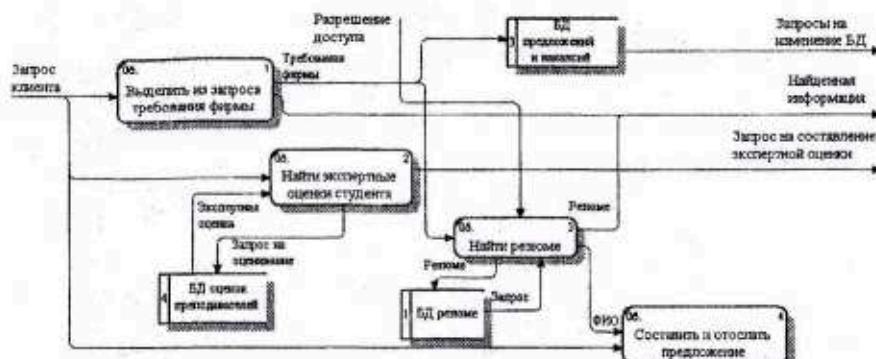


Рисунок 3.4 – DFD-декомпозиция работы “Обработать запрос фирмы”

### Контрольные вопросы

- Понятие CASE средств и их назначение.
- Назначение и сущность DFD диаграмм.
- Компоненты диаграммы DFD.
- В чем состоит назначение процесса?
- Что называется внешней сущностью?
- Что описывают хранилища?
- Объясните механизм дополнения диаграммы IDEF0 диаграммой DFD.
- Какая нотация используется в BPWin для построения диаграмм DFD?

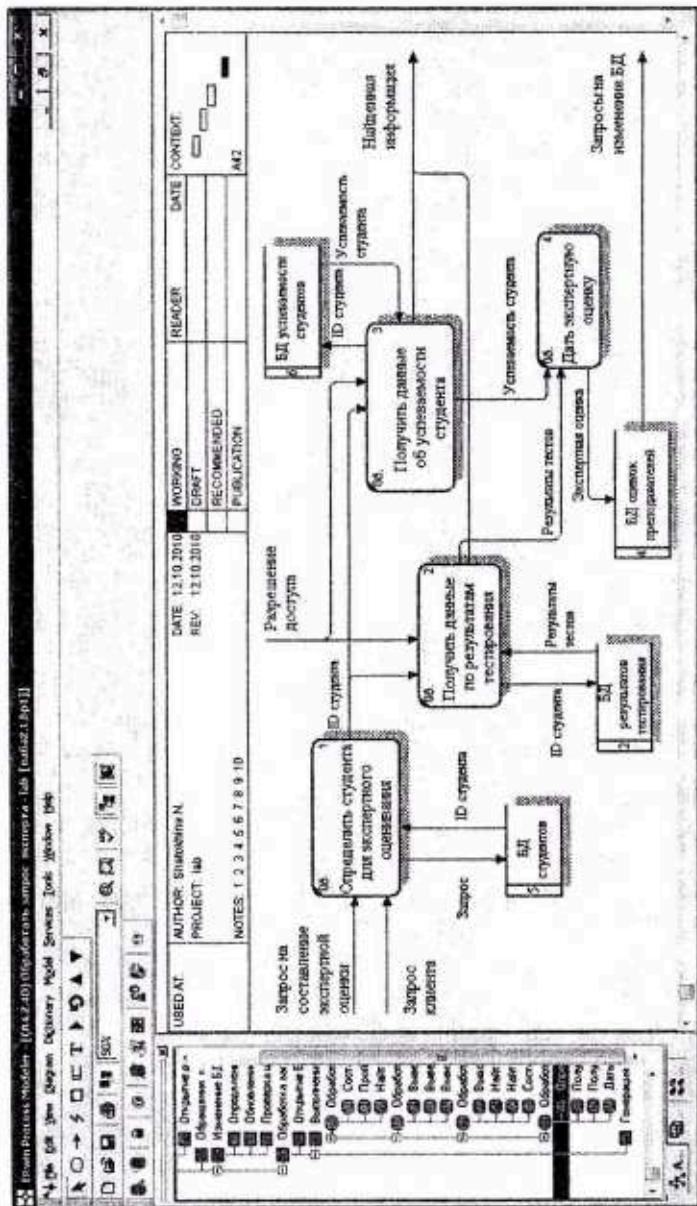


Рисунок 3.5 – DFD-декомпозиция работы “Обработка запроса” эксперта

## Лабораторная работа № 4

### ПОСТРОЕНИЕ WORKFLOW ДИАГРАММ (IDEF3)

*Цель работы – построение диаграмм IDEF3 для получения возможности передать аналитикам описание последовательности выполнения процессов, а также объектов, участвующих совместно в одном процессе.*

*Ход работы:*

1. Создать диаграмму IDEF3, определяющую последовательность заполнения БД.
2. Связать работы на диаграмме.
3. Добавить на диаграмму перекрестки, моделирующие параллельные события при заполнении БД.
4. Добавить объект-ссылку и связать его с диаграммой.

Наличие в диаграммах DFD описания источников, приемников и хранилищ данных позволяет описать процесс движения информации в системе. Для описания логики взаимодействия информационных потоков модель системы дополняют диаграммами IDEF3. Их ещё называют диаграммами потоков работ.

Методология моделирования IDEF3 (WorkFlow diagramming) позволяет описывать графически течение процессов во времени а также отношения процессов и объектов, являющихся частями этих процессов [2].

В методологии существует два типа моделей [3]:

- 1) модель отражающая процессы в их логической последовательности и позволяющая увидеть функционирование системы;
- 2) модель, отражающая “сеть переходных состояний” объекта и позволяющая увидеть последовательность состояний, в которых может оказаться объект при прохождении через определенный процесс.

Диаграмма WorkFlow может использоваться для:

- анализа динамики событий;
- анализа процедур обработки информации;
- как метод создания процессов;
- с их помощью можно описывать сценарии действий сотрудников.

IDEF3 дополняет IDEF0 и содержит все необходимое для построения моделей, которые могут быть использованы для имитационного моделирования.

Организация диаграмм в IDEF3 является особенно важной, если модель редактируется несколькими людьми. В этом случае разработчик должен определить, какая информация будет входить в ту или иную модель.

В IDEF3 *работы* изображаются прямоугольниками с прямыми углами и имеют имя, выраженное отглагольным существительным, обозначающим процесс действия, одиночным или в составе фразы, и номер (идентификатор). Другое имя существительное в составе той же фразы обычно отображает основной выход (результат) работы. Обычно номер работы состоит из номера родительской работы и порядкового номера на текущей диаграмме.

*Связи* показывают взаимоотношения работ. Все связи в IDEF3 односторонние и могут быть направлены куда угодно, но обычно диаграммы IDEF3 стараются построить так, чтобы связи были направлены слева направо. В IDEF3 различают три типа стрелок:

*Старшая (Precedence) линия* – сплошная линия ( $\rightarrow$ ), связывающая единица работ. Рисуется слева направо или сверху вниз. Показывает, что работа-источник должна закончиться прежде, чем работа-цель начнется.

*Линия отношения (Relation Link)* – пунктирная линия ( $\cdots\rightarrow$ ), использующаяся для изображения связей между единицами работ, а также между единицами работ и объектами ссылок.

*Потоки объектов (Object Flow)* – стрелка с двумя наконечниками ( $\rightarrow\rightarrow$ ), применяется для описания использования объекта в двух или более единицах работы, например, когда объект порождается в одной работе и используется в другой.

*Перекрестки (Junction)* используются для отображения логики взаимодействия стрелок при слиянии и разветвлении или для отображения множества событий, которые могут или должны быть завершены перед началом следующей работы.

Различают перекрестки для слияния (Fan-in Junction) и разветвления (Fan-out Junction) стрелок. Перекресток не может использоваться одновременно для слияния и для разветвления.

Типы перекрестков подробно рассмотрены в [2].

*Объекты-ссылки* являются специальными символами, которые указываются на внешние части описания процесса. Они добавляются на диаграмму для того, чтобы обратить внимание редактора на что-либо важное, что невозможно связать со стрелкой, работой или перекрестком.

Объект-ссылка отображается в виде прямоугольника, который должен быть связан с единицами работ или перекрестками пунктирными линиями. При внесении объектов-ссылок необходимо указать их тип. Типы объектов-ссылок рассмотрены в [2].

С помощью диаграмм IDEF3 обычно моделируют последовательности работ, имеющие технологические и временные связи. К таким моделям можно отнести проект разработки системы службы занятости, который и будет рассмотрен в данном примере.

Перед началом моделирования необходимо создать иерархическую структуру работ, описывающую процесс разработки системы.

1 этап – Разработка технического задания:

- 1.1. Составление технического задания.
- 1.2. Утверждение технического задания.

2 этап – Анализ требований:

- 1.1. Определение объектов системы и их атрибутов.
- 1.2. Определение категорий пользователей.
- 1.3. Создание запросов к системе.

3 этап – Разработка модульной структуры:

- 3.1. Разработка модульной структуры всей системы.
- 3.2. Разработка модульной структуры подсистемы обработки запросов, определения категории пользователей.

3.3. Разработка модульной структуры подсистемы экспертных оценок.

3.4. Разработка модульной структуры контроля успеваемости студентов.

4 этап – Проектирование БД:

- 4.1. Проектирование концептуальной модели БД.
- 4.2. Проектирование логической структуры БД.
- 4.3. Выбор СУБД.

5 этап – Кодирование (программирование модулей системы).

6 этап – Тестирование.

Согласно созданной структуре работ определим диаграммы, добавив на них взаимосвязи между работами.

На стадии разработки технического задания заказчик системы играет важную роль, снабжая разработчиков необходимой информацией для создания системы. Поэтому на диаграмме показан соответствующий объект-ссылка, влияющий на работу «Разработка технического задания».

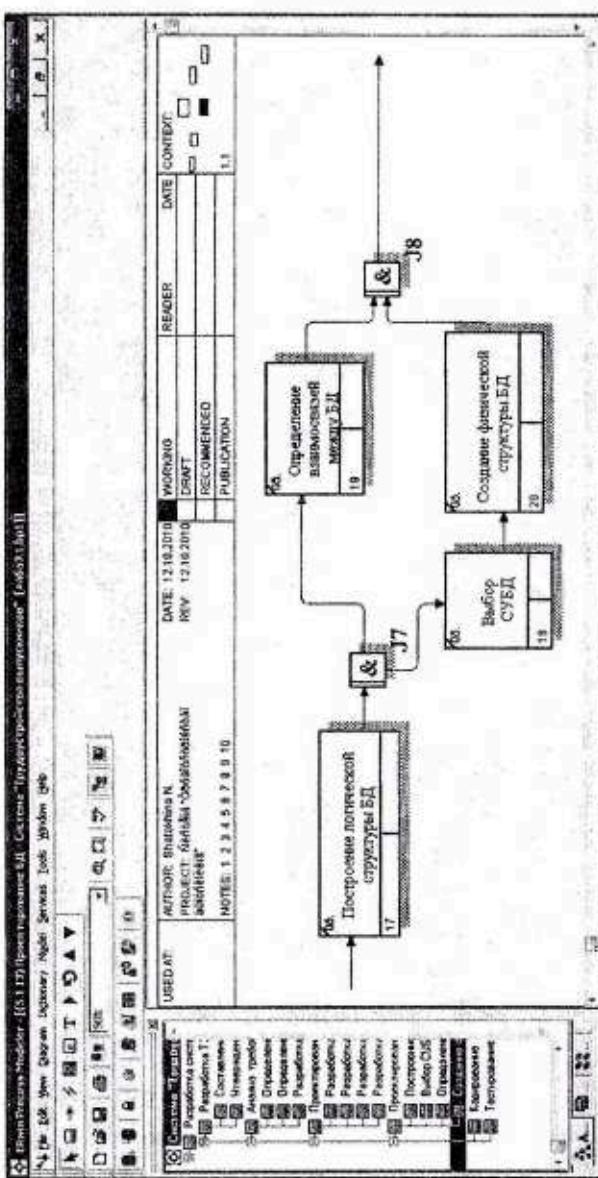


Рисунок 4.5 – Декомпозиция работы “Проксигрирование БД”

## Лабораторная работа № 5

### ПОСТРОЕНИЕ IDEF 1X ДИАГРАММ

*Цель работы – построение диаграмм IDEF 1X при помощи CASE-средства ERwin 7.3.*

*Ход работы:*

1. Изучить методологию функционального проектирования в нотации IDEFIX.

2. Освоить CASE-средство ERwin 7.3 на примере:

- Задать атрибуты сущностей.
- Определить первичные ключи в сущностях.
- Определить состав альтернативных ключей.
- Связать сущности между собой, используя описанные типы связей.
- После проведения связей определить состав внешних ключей.
- Проверить модель на соответствие предметной области. Сохранить полученную диаграмму.
- Нормализовать БД до третьей нормальной формы.
- Создать физическую модель БД системы “Трудоустройство выпускников кафедры”.

ERwin поддерживает функциональное моделирование в нотации IDEF 1X и стандарт IE (Information engineering). Методология IDEFIX подразделяется на уровни, соответствующие проектируемой модели данных системы по принципу “сверху-вниз”.

Диаграмма “сущность-связь” определяет сущности и их отношения. Модель данных, основанная на ключах, дает более подробное представление данных. Она включает описание всех сущностей и первичных ключей, которые соответствуют предметной области.

Нижний уровень состоит из “трансформационной модели” и “полной атрибутивной модели” [6, 7]. Трансформационная модель содержит всю информацию для реализации проекта, который может быть частью общей информационной системы и описывать предметную область. Трансформационная модель позволяет проектировщикам и администраторам БД представлять, какие объекты БД хранятся в словаре данных, и проверить, насколько физическая модель данных удовлетворяет требованиям информационной системы. Фактически из трансформационной модели автоматически можно получить модель СУБД.

Предпосылки использования CASE-средства ERwin:

1. Возможность создания диаграмм структуры БД, позволяющих автоматически решать вопросы, связанные с сохранением ее целостности.
2. Независимость логической модели от используемой СУБД, что позволяет применять универсальные методы для ее экспорта в конкретные СУБД.
3. Возможность формирования большого числа отчетов, отражающих текущее состояние процесса проектирования БД.

Рассмотрим процесс построения логической модели на примере БД студентов системы «Трудоустройство выпускников кафедры». Первым этапом является определение сущностей и атрибутов. В БД будут храниться записи о студентах, следовательно, сущностью будет студент.

Таблица 5.1 – Атрибуты сущности “Студент”

Атрибут	Описание
Номер	Уникальный номер для идентификации пользователя
Ф.И.О.	Фамилия, имя и отчество пользователя
Пароль	Пароль для доступа в систему
Возраст	Возраст студента
Пол	Пол студента
Характеристика	Мемо-поле с общей характеристикой пользователя
E-mail	Адреса электронной почты
Телефон	Номера телефонов студента (домашний, рабочий)
Опыт работы	Специальности и опыт работы студента по каждой из них
Специальность	Специальность, получаемая студентом при окончании учебного заведения
Специализация	Направление специальности, по которому обучается студент
Иностранный язык	Список иностранных языков и уровень владения ими
Тестирование	Список тестов и отметки о их прохождении
Экспертная оценка	Список предметов с экспертными оценками по каждому из них
Оценки по экзаменам	Список сданных предметов с оценками

В полученном списке существуют атрибуты, которые нельзя определить в виде одного поля БД. Такие атрибуты требуют дополнительных определений и должны рассматриваться как сущности, состоящие, в свою очередь, из атрибутов. К таковым относятся: опыт работы, иностранный язык, тестирование, экспертная оценка, оценки по экзаменам. Определим их атрибуты.

Таблица 5.2 – Атрибуты сущности “Опыт работы”

Атрибут	Описание
Опыт	Опыт работы по данной специальности в годах
Место	Наименование предприятия, где приобретался опыт

Таблица 5.3 – Атрибуты сущности “Тестирование”

Атрибут	Описание
Название	Название теста, который прошел студент
Описание	Содержит краткое описание теста
Оценка	Оценка, которую получил студент в результате прохождения теста

Таблица 5.4 – Атрибуты сущностей “Иностранный язык”

Атрибут	Описание
Язык	Название иностранного языка, которым владеет студент
Уровень владения	Численная оценка уровня владения иностранным языком

Таблица 5.5 – Атрибуты сущности “Экспертная оценка”

Атрибут	Описание
Дисциплина	Наименование дисциплины, по которой оценивался студент
Ф.И.О. преподавателя	Ф.И.О. преподавателя, который оценивал студента
Оценка	Экспертная оценка преподавателя

Таблица 5.6 – Атрибуты сущности “Оценки по экзаменам”

Атрибут	Описание
Предмет	Название предмета, по которому сдавался экзамен
Оценка	Полученная оценка

Составим ERD-диаграмму, определяя типы атрибутов и проправляя связи между сущностями. Все сущности будут зависимыми от сущности “Студент”. Связи будут типа “один-ко-многим”.

Выберем для каждой сущности ключевые атрибуты, однозначно определяющие сущность. Для сущности "Студент" это будет уникальный номер, для сущности "Опыт работы" все поля являются ключевыми, так как по разным специальностям студент может иметь разный опыт работы в разных фирмах. Сущность "Тест" определяется названием, так как студент по одному тесту может иметь только одну оценку. Оценка по экзамену определяется только названием предмета, экспертная оценка зависит от преподавателя, который ее составил, поэтому в качестве ключевых атрибутов выберем "Дисциплину" и "ФИО преподавателя". У сущности "Иностранный язык" уровень владения зависит только от наименования языка, следовательно, это и будет являться ключевым атрибутом.

Получим новую диаграмму, изображенную на рис. 5.1, где все ключевые атрибуты будут находиться над горизонтальной чертой внутри рамки, изображающей сущность.

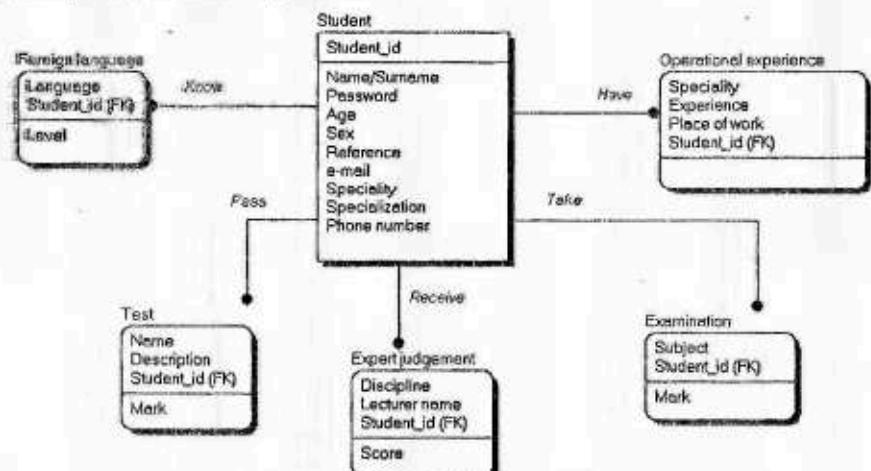


Рисунок 5.1 – ERD-диаграмма с ключевыми атрибутами

Нормализация – это процесс проверки и реорганизации сущностей и атрибутов с целью удовлетворения требований к реляционной модели данных. Нормализация позволяет быть уверенным, что каждый атрибут определен для своей сущности, что позволяет значительно сократить объем памяти для хранения данных.

Существуют следующие виды нормальных форм [5].

*Первая нормальная форма.* Сущность находится в 1-й нормальной форме, только в том случае, если все атрибуты содержат единственно атомарные значения. Среди атрибутов не должно встречаться повторяющихся групп, т. е. нескольких значений для каждого экземпляра.

*Вторая нормальная форма.* Сущность находится во 2-й нормальной форме, если она находится в 1-й нормальной форме и каждый неключевой атрибут полностью зависит от первичного ключа, т.е. не существует зависимостей от части ключа.

*Третья нормальная форма.* Сущность находится в 3-й нормальной форме, если она находится во 2-й нормальной форме и неключевые атрибуты сущности зависят от других атрибутов сущности.

После 3-й нормальной формы существуют нормальная форма Бойсса-Кодда, 4-я и 5-я нормальные формы [5]. Однако на практике, как правило, ограничиваются приведением к третьей нормальной форме.

Нормализуем полученную БД (рис. 5.1) до 3-й нормальной формы. Для приведения БД в 1-ю нормальную форму необходимо выполнить условие, при котором все атрибуты содержат атомарные значения. Рассмотрим атрибуты сущности “Студент”. Студент может иметь несколько адресов электронной почты и несколько телефонных номеров, что является нарушением 1-й нормальной формы. Необходимо создать отдельные сущности “E-mail” и “Телефон” и связать их с сущностью “Студент” (см. рис. 5.2).

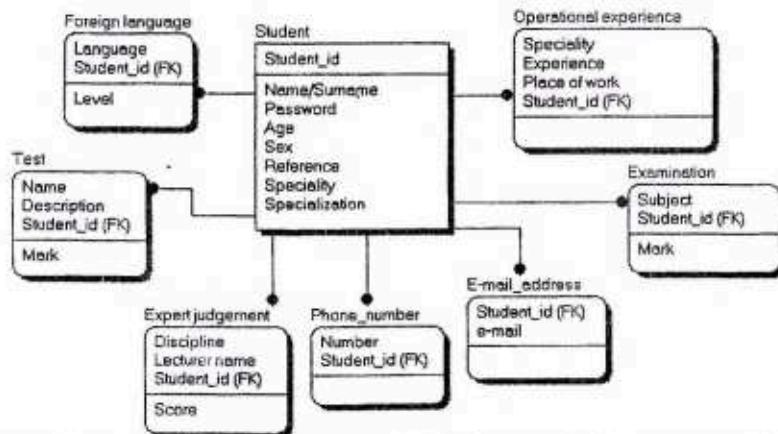


Рисунок 5.2 – Диаграмма “сущность-связь” БД “Студенты” в 1-й нормальной форме

Проверим соответствие БД 2-й нормальной форме. Все неключевые атрибуты полностью должны зависеть от первичного ключа. Нетрудно заметить, что это условие выполняется для всех сущностей БД; следовательно, можно сделать вывод о том, что она находится во 2-й нормальной форме.

Для приведения БД к 3-й нормальной форме необходимо обеспечить отсутствие транзитивных зависимостей неключевых атрибутов. Такая зависимость наблюдается у атрибутов “Специальность” и “Специализация” у сущности “Студент”: специализация зависит от специальности и от группы, в которой обучается студент.

Создадим новую независимую сущность “Специальность”, перенеся в нее атрибут “Специализация” и создав новый атрибут “Группа”, являющийся ключевым и определяющий атрибуты “Специальность” и “Специализация”. Проведем неидентифицирующую связь от сущности “Специальность” к сущности “Студент”, при этом ключевой атрибут “Группа” мигрирует в сущность “Студент”. Получим БД в 3-й нормальной форме, так как других транзитивных зависимостей неключевых атрибутов не имеется (см. рис. 5.3).

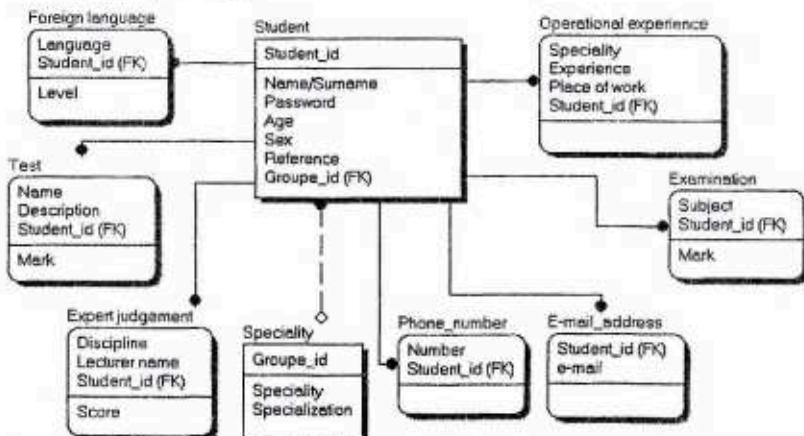


Рисунок 5.3 – Диаграмма “сущность–связь” БД “Студенты” в 3-й нормальной форме

Перед построением физической модели выберем в качестве сервера Microsoft Access 2007, получив физическую модель, сгенерированную ERWin по умолчанию (рис. 5.4).

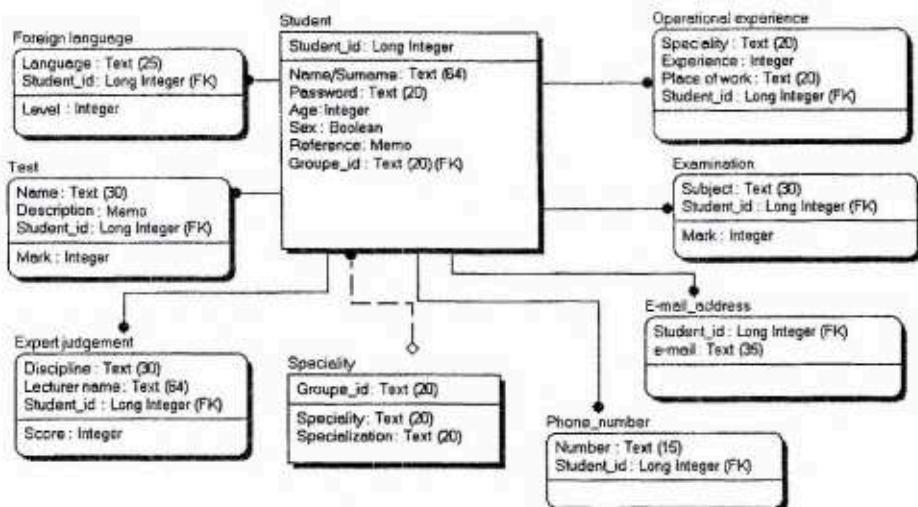


Рисунок 5.4 – Физическая модель БД “Студенты”

В полученной модели необходимо скорректировать типы и размеры полей. Кроме того, на этапе создания физической модели данных вводятся правила валидации колонок, определяющие списки допустимых значений и значения по умолчанию.

### Контрольные вопросы

1. Цель ERD-диаграммы.
2. Что является основным компонентом реляционных БД?
3. Что показывает взаимосвязь между сущностями?
4. Назовите типы логических взаимосвязей.
5. Опишите механизм проверки адекватности логической модели.
6. Что называется первичным ключом? Назовите принципы формирования первичного ключа.
7. Что называется альтернативным ключом?
8. Что называется инверсионным входом?
9. Что называется процессом нормализации?
10. Что называется функциональной зависимостью?
11. Первая нормальная форма. Вторая нормальная форма. Третья нормальная форма.
20. Как осуществляется разрешение связей “многие-ко-многим”?

## СПИСОК ЛИТЕРАТУРЫ

1. Вендрев А.М. Проектирование программного обеспечения экономических информационных систем / А.М. Вендрев. – М. Финансы и статистика, 2000.
2. Калянов Г.Н. CASE-технологии. Консалтинг при автоматизации бизнес-процессов / Г.Н. Калянов. – М. СИНТЕГ, 2000.
3. Маклаков С.В. Создание информационных систем с AllFusion Modeling Suite / С.В. Маклаков. – М. ДИАЛОГ-МИФИ, 2003. – 432 с.
4. Марка Д.А., Мак Гоуен К. Методология структурного анализа и проектирования SADT / Д.А. Марка, К. Мак Гоуен. – М. 1993.
5. Мейер Д. Теория реляционных баз данных / Д. Мейер. – М. Мир, 2003.
6. Смирнова Г.Н. Проектирование экономических информационных систем: учебник / Г.Н. Смирнова, А.А. Сорокин, Ю.Ф. Тельнов; под ред. Ю.Ф. Тельнова. – М. Финансы и статистика, 2001. – 512 с.
7. Соммервилл И. Инженерия программного обеспечения: 6-е изд. / И. Соммервилл. – М. Вильямс, 2002.
8. Федотова Д.Э., Семенов Ю.Д., Чижик К.Н. CASE-технологии: практикум / Д.Э. Федотова, Ю.Д. Семенов, К.Н. Чижик. – М. Горячая линия, 2005. – 160 с.
9. Черемных С.В., Ручкин В.С., Семенов И.О. Структурный анализ систем. IDEF-технологии / С.В. Черемных, В.С. Ручкин, И.О. Семенов. – М. Финансы и статистика, 2001.

## СОДЕРЖАНИЕ:

Вступление .....	3
1. Описание предметной области для информационной системы “Трудоустройство выпускников кафедры”.....	4
Контрольные вопросы.....	9
2. Построение IDEF0 диаграммы при помощи CASE средства BRwin 7.3 ..	10
Контрольные вопросы.....	17
3. Построение DFD диаграммы при помощи CASE средства BRwin 7.3 ...	19
Контрольные вопросы.....	23
4. Построение IDEF3 диаграммы при помощи CASE средства BRwin 7.3 ..	25
Контрольные вопросы.....	29
5. Построение IDEF 1X диаграммы при помощи CASE средства ERwin 7.3 .....	31
Контрольные вопросы.....	37
Список литературы.....	39

## **Навчальне видання**

### **Проектирование информационных систем**

#### **Методичні вказівки**

до лабораторних занять і самостійної роботи з курсів

“Управління інформаційними проектами” і

“Управління проектами розробки інформаційних комп’ютерних технологій”

для студентів спеціальності 8.000003 “Управління проектами” і

напряму 0804 “Комп’ютерні науки”

**Російською мовою**

**Укладач Шатохіна Наталія Володимирівна**

Відповідальний за випуск      I.B. Кононенко

Роботу рекомендував до видання      О.В. Горілій

Редактор      М.П. Єфремова

**План 2010 р., поз. 161**

Підп. до друку 25.10.2010 р. Формат 60 × 84 1/16. Папір офісний.  
Riso-друк. Гарнітура Таймс. Ум. друк. арк. 2,3. Наклад 50 прим.  
Зам. № 323. Ціна договірна.

---

**Видавничий центр НТУ «ХПІ».**

Свідоцтво про державну реєстрацію ДК № 3657 від 24.12.2009 р.  
61002, Харків, вул. Фрунзе, 21

---

Друкарня НТУ «ХПІ», 61002, Харків, вул. Фрунзе, 21