ПРАКТИЧЕСКИЕ ЗАДАНИЯ ПО ДИСЦИПЛИНЕ «CASE- СРЕДСТВА ПРОЕКТИРОВАНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ»

Для направления подготовки

44.03.01 Педагогическое образование, Направленность (профиль): Информатика

Практическое занятие № 1.

Тема: Изучение основных функций пакета BPwin

BPwin позволяет аналитику создавать сложные модели бизнес-процессов при минимальных усилиях. BPwin поддерживает три методологии - IDEF0, IDEF3 и DFD. Каждая из них призвана решать свои специфические задачи. Также можно строить смешанные модели. Модель в BPwin рассматривается как совокупность работ, каждая из которых оперирует с некоторым набором данных. Работы изображаются в виде прямоугольников (блоков), данные - в виде стрелок (дуг).

Основу методологии IDEF0 составляет графический язык описания бизнес-процессов. Модель в IDEF0 представлена совокупностью иерархически упорядоченных и логически связанных диаграмм. Каждая диаграмма располагается на отдельном листе. Можно выделить четыре типа диаграмм:

- контекстную диаграмму A-0 (в каждой модели может быть только одна контекстная диаграмма);

- диаграммы декомпозиции (в том числе диаграмма первого уровня декомпозиции A0, раскрывающая контекстную);

- диаграммы дерева узлов;

- диаграммы только для экспозиции (FEO).

Контекстная диаграмма является вершиной древовидной структуры диаграмм и представляет собой самое общее описание системы и ее взаимодействия с внешней средой (как правило, здесь описывается основное назначение моделируемого объекта). После описания системы в целом проводится разбиение ее на крупные фрагменты. Этот процесс называется функциональной декомпозицией, а диаграммы, которые описывают каждый фрагмент и взаимодействие фрагментов, называются диаграммами декомпозиции. После декомпозиции контекстной диаграммы (т.е., получения диаграммы A0) проводится декомпозиция каждого блока диаграммы A0 на более мелкие фрагменты и так далее, до достижения нужного уровня

подробности описания. После каждого сеанса декомпозиции проводятся сеансы экспертизы - эксперты предметной области (обычно это интервьюируемые аналитиками сотрудники предприятий) указывают на соответствие реальных бизнес-процессов созданным диаграммам. Найденные несоответствия исправляются, и только после прохождения экспертизы без замечаний можно приступать к следующему сеансу декомпозиции. Так достигается соответствие модели реальным бизнес-процессам на любом и каждом уровне модели. Синтаксис описания системы в целом и каждого ее фрагмента одинаков во всей модели. Диаграмма дерева узлов показывает иерархическую зависимость работ, но не взаимосвязи между работами. Диаграмм деревьев узлов может быть в модели сколько угодно, поскольку дерево может быть построено на произвольную глубину и не обязательно с корня.

Диаграммы для экспозиции (FEO) строятся для иллюстрации отдельных фрагментов модели, для иллюстрации альтернативной точки зрения, либо для специальных целей.

Каркас диаграммы.

На рис.1 показан типичный пример контекстной диаграммы с граничными рамками, которые называются каркасом диаграммы. Каркас содержит заголовок (верхняя часть рамки, табл.3) и подвал (нижняя часть, табл.4). Заголовок каркаса используется для отслеживания диаграммы в процессе моделирования. Нижняя часть используется для идентификации и позиционирования в иерархии диаграмм. Значения полей каркаса задаются в диалоге Diagram Properties (в меню Edit/Diagram Properties).

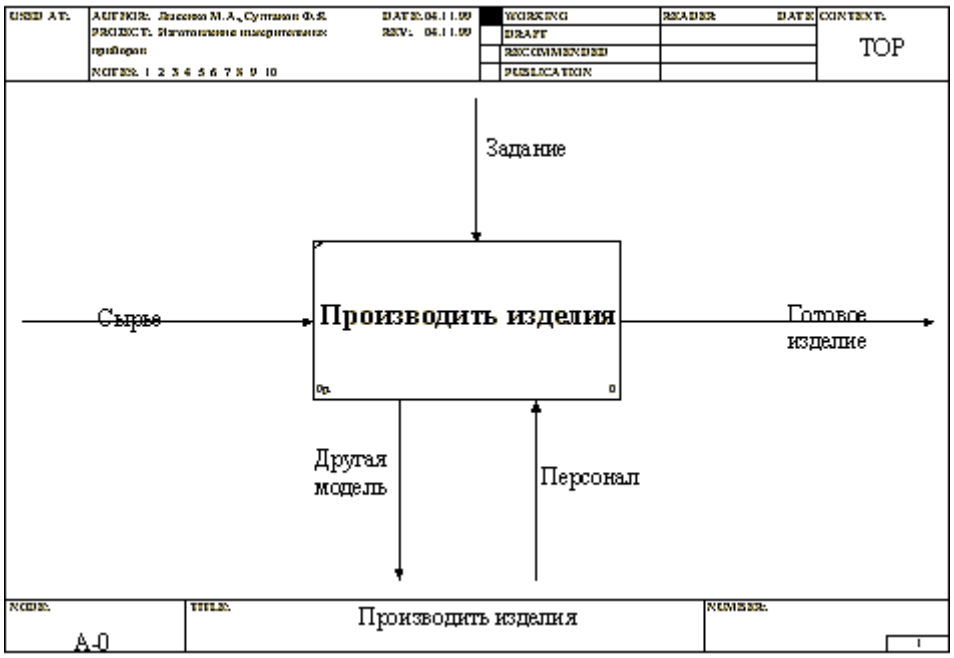

Рис.1.Контекстная диаграмма

Поля заголовка каркаса (слева направо)

Табл. 1

Поле	**Смысл**
Used At	Используется для указания на родительскую работу в случае, если на текущую диаграмму ссылались посредством стрелки вызова.
Author, Date, Rev, Project	Имя создателя диаграммы, дата создания и имя проекта, в рамках которого была создана диаграмма. REV - дата последнего редактирования диаграммы.
Notes 1 2 3 4 5 6 7 8 9 10	Используется при проведении сеанса экспертизы. Эксперт должен (на бумажной копии диаграммы) указать число замечаний, вычеркивая цифру из списка каждый раз при внесении нового замечания.
Status	Статус отображает стадию создания диаграммы, отображая все этапы публикации.
Working	Новая диаграмма, кардинально обновленная диаграмма или новый автор диаграммы.
Draft	Диаграмма прошла первичную экспертизу и готова к дальнейшему обсуждению.
Recommended	Диаграмма и все ее сопровождающие документы прошли экспертизу. Новых изменений не ожидается.
Publication	Диаграмма готова к окончательной печати и публикации.
Reader	Имя читателя (эксперта).
Date	Дата прочтения (экспертизы).
Context	Схема расположения работ в диаграмме верхнего уровня. Работа, являющаяся родительской, показана темным прямоугольником, остальные - светлым. На контекстной диаграмме (A-0) показывается надпись TOP. В левом нижнем углу показывается номер по узлу родительской диаграммы.

Поля подвала каркаса (слева направо)

Табл. 2

Поле	**Смысл**
Node	Номер узла диаграммы (номер родительской работы)

Title	Имя диаграммы. По умолчанию - имя родительской работы
Number	C-Number, уникальный номер версии диаграммы
Page	Номер страницы, может использоваться как номер страницы при формировании папки

Задание.

На основе резюме, описывающих функционирование конкретного отдела РГУ нефти и газа им. И.М. Губкина, создать контекстную диаграмму А-0. Выделить основные его функции и создать диаграмму А0. Разбить каждую функцию на подфункции и диаграммы третьего уровня. Предоставить иерархию диаграмм.

Контрольные вопросы.

1. Каковы стадии жизненного цикла информационных систем, их основное содержание?
2. Что такое реинжиниринг бизнес-процессов?
3. Какие виды работ рекомендуется выполнить при построении моделей деятельности, какие средства и методологии при этом используются?
4. Каковы основные функции CASE-средства BPwin?
5. Как представляется функциональная модель деятельности в методологии IDEF0?

Практическое занятие № 2.

Тема: Изучение объектов диаграмм функциональной модели

Работы (Activity).

Работы обозначают поименованные процессы, функции или задачи, которые происходят в течение определенного времени и имеют распознаваемые результаты. Работы изображаются в виде прямоугольников (блоков). Все работы должны быть названы и определены. Имя работы должно быть глаголом (например, "Изготовить деталь", "Принять заказ" и т.д.). Работу можно добавить в диаграмму, щелкнув по кнопке  на палитре инструментов, а затем по свободному месту на диаграмме. Работы на диаграммах декомпозиции располагаются по диагонали от левого верхнего угла к правому нижнему (рис.2). Такой порядок называется порядком доминирования. Согласно этому принципу расположения в левом верхнем углу располагается самая важная работа или работа, выполняемая по времени первой. Далее вправо вниз располагаются менее важные или выполняемые позже работы.

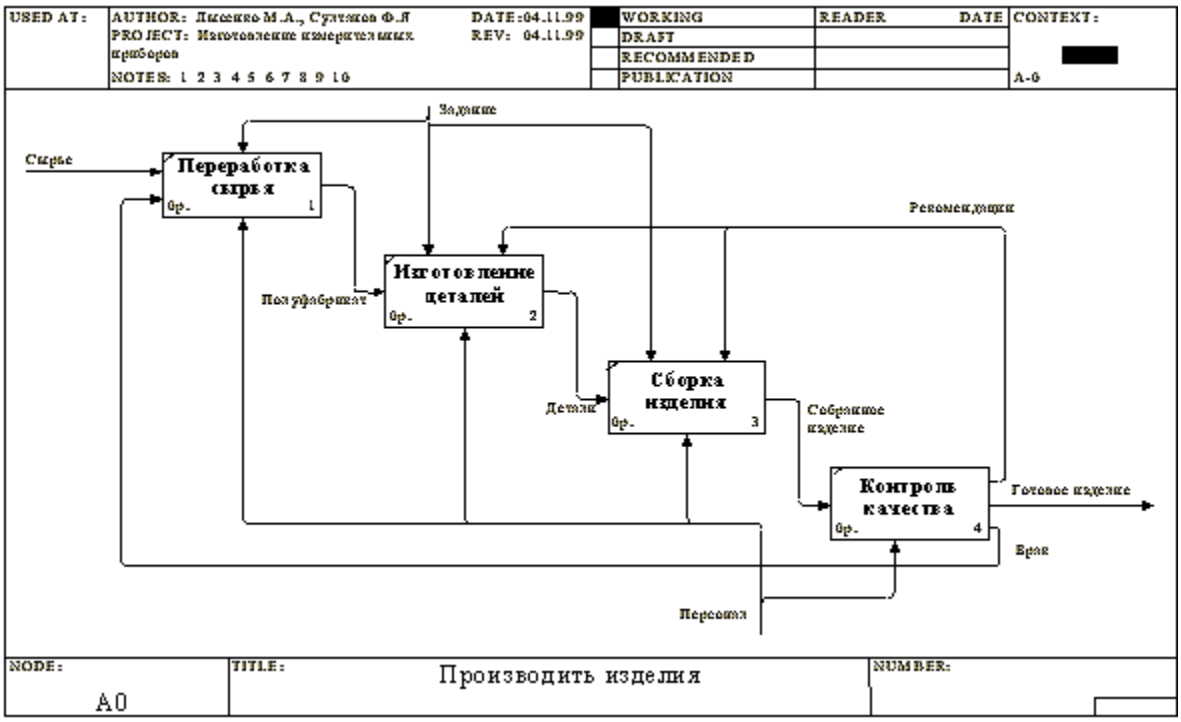

Рис.2.Диаграмма декомпозиции

Для внесения имени работы следует щелкнуть по работе правой кнопкой мыши, выбрать в меню пункт Name Editor и в появившемся диалоге внести имя работы (рис.3). Диаграммы декомпозиции содержат родственные работы, т.е. дочерние работы, имеющие общую родительскую работу.

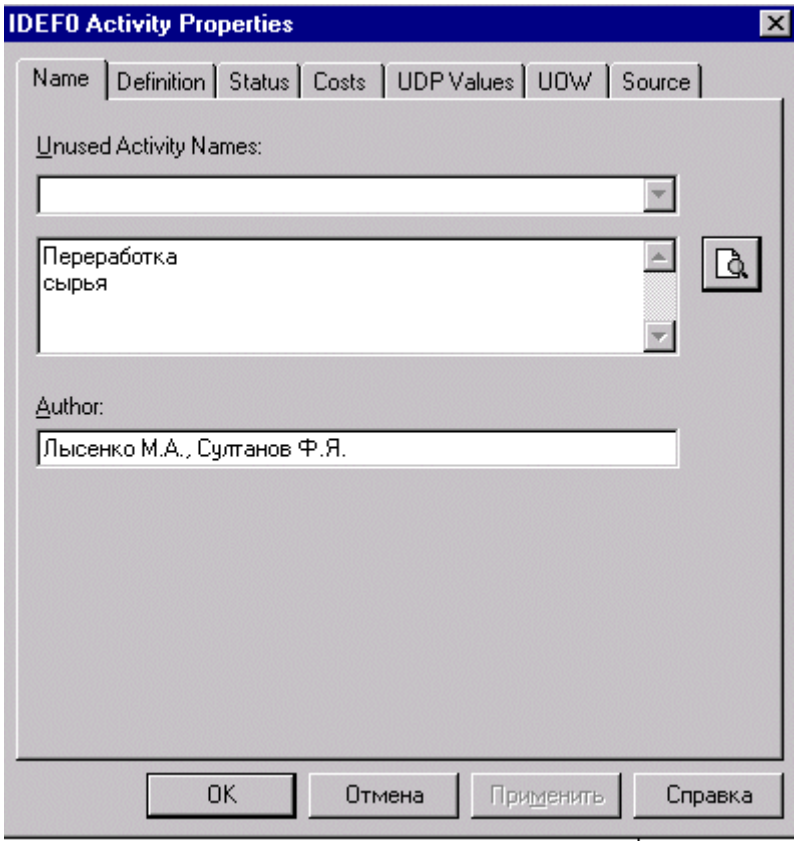

Рис.3.Внесение имени работы

Для создания диаграммы декомпозиции следует щелкнуть по кнопке  и выбрать на диаграмме работу, которую необходимо декомпозировать.
Возникает диалог Activity Box Count (рис.4), в котором следует указать нотацию новой диаграммы. Надо выбрать IDEF0 и надавить ОК.

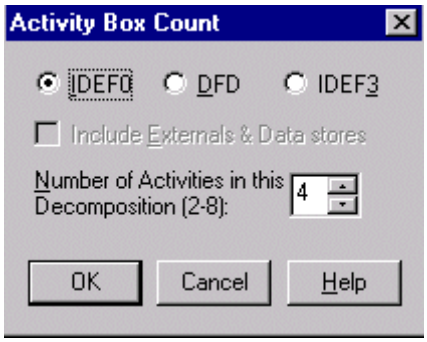

Рис.4.Выбор нотации диаграммы

На диаграмме декомпозиции работы нумеруются автоматически слева направо. Номер работы показывается в правом нижнем углу. В левом верхнем углу изображается небольшая диагональная черта, которая показывает, что данная работа не была декомпозирована.

Стрелки (Arrows).

Взаимодействие работ с внешним миром описывается в виде стрелок. Стрелки представляют собой некую информацию и именуются существительными (например, "Заготовка", "Изделие", "Заказ"). В IDEF0 различают пять типов стрелок.

- Вход (Input) - материал или информация, которая используется или преобразуется работой для получения результата (выхода). Допускается, что работа может не иметь ни одной стрелки входа. Каждый тип стрелок подходит к определенной стороне блока, или выходит из нее. Очень часто сложно определить, являются ли данные входом или управлением. В этом случае подсказкой может служить то, перерабатываются/изменяются ли данные в работе или нет. Если изменяются, то скорее всего это вход, если нет - управление.

- Управление (Control) - правила, стратегии, процедуры или стандарты, которыми руководствуется работа. Каждая работа должна иметь хотя бы одну стрелку управления. Управление влияет на работу, но не преобразуется ей. Если цель работы - изменить процедуру или стратегию, то такая процедура или стратегия будет для работы входом.

- Выход (Output) - материал или информация, которые производятся работой. Каждая работа должна иметь хотя бы одну стрелку выхода. Работа без результата не имеет смысла.

- Механизм (Mechanism) - ресурсы, которые выполняют работу, например персонал предприятия, станки, устройства и т.д.

- Вызов (Call) - специальная стрелка, указывающая на другую модель работы. Рисуется как исходящая из нижней грани работы. Стрелка вызова используется для указания того, что некоторая работа выполняется за пределами моделируемой системы. Используются в механизме слияния и разделения моделей.

Каждый тип стрелок подходит к определенной стороне блока, или выходит из нее. Стрелка входа рисуется как входящая в левую грань работы. Стрелка управления рисуется как входящая в верхнюю грань. Выход рисуется как исходящая стрелка из правой грани. Механизм - входит в нижнюю.

Граничные стрелки.

Стрелки на контекстной диаграмме служат для описания взаимодействия системы с окружающим миром. Они могут начинаться у границы диаграммы и заканчиваться у работы, или наоборот. Такие стрелки называются граничными. Для внесения граничной стрелки надо:

- щелкнуть по кнопке с символом стрелки  в палитре инструментов. Дальше перенести курсор к левой стороне экрана, пока не появится начальная штриховая полоска;

- щелкнуть один раз по полоске (откуда выходит стрелка) и еще раз в левой части работы со стороны входа (где заканчивается стрелка);

- вернуться в палитру инструментов и выбрать опцию редактирования стрелки 

- щелкнуть правой кнопкой мыши на линии стрелки, во всплывающем меню выбрать пункт Name Editor и добавить имя стрелки в закладке Name диалога IDEF0 Arrow Properties.

Стрелки управления, входа, механизма и выхода изображаются аналогично. Для рисования стрелки выхода, например, следует щелкнуть по кнопке с символом стрелки в палитре инструментов, щелкнуть в правой части работы со стороны выхода (где начинается стрелка), перенести курсор к правой стороне экрана, пока не появится штриховая полоска, и щелкнуть один раз по ней. Имена вновь внесенных стрелок автоматически заносятся в словарь (Arrow Dictionary).

Словарь стрелок (Arrow Dictionary) редактируется при помощи специального редактора Arrow Dictionary Editor (рис.5), в котором определяется стрелка и вносится относящийся к ней комментарий. Словарь стрелок решает очень важную задачу. Диаграммы создаются аналитиком для того, чтобы провести сеанс экспертизы, т.е. обсудить диаграмму со специалистом предметной области. В любой предметной области формируется профессиональный жаргон, причем очень часто жаргонные выражения имеют нечеткий смысл и воспринимаются разными специалистами по-разному. В то же время аналитик - автор диаграмм должен употреблять те выражения, которые наиболее понятны экспертам.

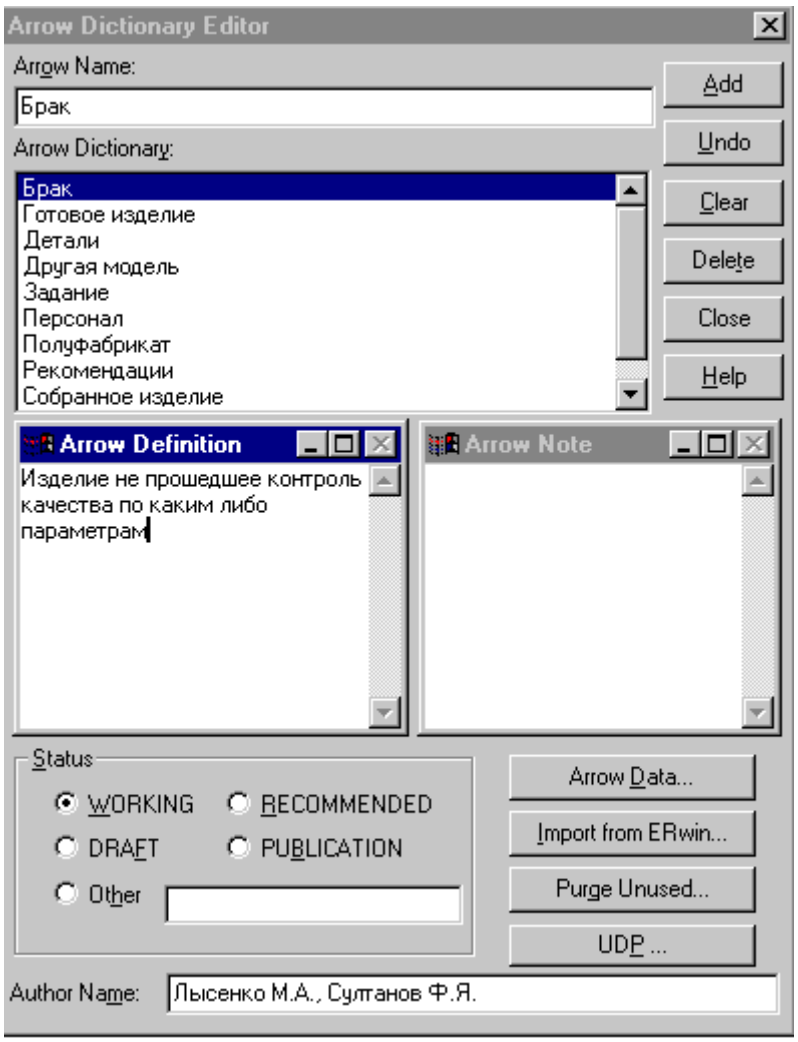

Рис.5.Редактор словаря стрелок

Поскольку формальные определения часто сложны для восприятия, аналитик вынужден употреблять профессиональный жаргон, а чтобы не возникало неоднозначных трактовок, в словаре стрелок каждому понятию можно дать расширенное и, если это необходимо, формальное определение.

Внутренние стрелки.

Для связи работ между собой используются внутренние стрелки, т.е. стрелки, которые не касаются границы диаграммы, начинаются у одной и кончаются у другой работы. Для рисования внутренней стрелки необходимо в режиме рисования стрелок щелкнуть по сегменту (например, выхода) одной работы и затем по сегменту (например, входа) другой. В IDEF0 различают пять типов связей работ:

- связь по входу (output-input), когда стрелка выхода вышестоящей работы (далее - просто выход) направляется на вход нижестоящей;

- связь по управлению (output-control), когда выход вышестоящей работы направляется на управление нижестоящей. Связь по входу показывает доминирование вышестоящей работы. Данные или объекты выхода вышестоящей работы не меняются в вышестоящей;

- обратная связь по входу (output-input feedback), когда выход нижестоящей работы направляется на вход вышестоящей. Такая связь, как правило, используется для описания циклов;

- обратная связь по управлению (output-control feedback), когда выход нижестоящей работы направляется на управление вышестоящей. Обратная связь по управлению часто свидетельствует об эффективности бизнес-процесса;

- связь выход-механизм (output-mechanism), когда выход одной работы направляется на механизм другой. Эта взаимосвязь используется реже остальных и показывает, что одна работа подготавливает ресурсы, необходимые для проведения другой работы.

Явные стрелки.

Явная стрелка имеет источником одну-единственную работу и назначением тоже одну-единственную работу.

Разветвляющиеся и сливающиеся стрелки.

Одни и те же данные или объекты, порожденные одной работой, могут использоваться сразу в нескольких других работах. 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. Для моделирования таких ситуаций IDEF0 используются разветвляющиеся и сливающиеся стрелки. Для разветвления стрелки нужно в режиме редактирования стрелки щелкнуть по фрагменту стрелки и по соответствующему сегменту работы. Для слияния двух стрелок выхода нужно в режиме редактирования стрелки сначала щелкнуть по сегменту выхода работы, а затем по соответствующему фрагменту стрелки.

Тоннелирование стрелок.

Вновь внесенные граничные стрелки на диаграмме декомпозиции нижнего уровня изображаются в квадратных скобках и автоматически не появляются на диаграмме верхнего уровня. Для их "перетаскивания" наверх нужно сначала выбрать кнопку  на палитре инструментов и щелкнуть по квадратным скобкам граничной стрелки. Появится диалог Border Arrow Editor (рис.6).

Рис.6.Диалог для тоннелирования стрелок

Если щелкнуть по кнопке Resolve Border Arrow, стрелка мигрирует на диаграмму верхнего уровня, если по кнопке Change To Tunnel - стрелка будет затоннелирована и не попадет на другую диаграмму. Тоннельная стрелка изображается с круглыми скобками на конце. Тоннелирование может быть применено для изображения малозначимых стрелок. Если на какой-либо диаграмме нижнего уровня необходимо изобразить малозначимые данные или объекты, которые не обрабатываются или не используются работами на текущем уровне, то их необходимо направить на вышестоящий уровень. Если эти данные не используются на родительской диаграмме, их нужно направить еще выше и т.д. В результате малозначимая стрелка будет изображена на всех уровнях и затруднит чтение всех диаграмм, на которых она присутствует. Выходом является тоннелирование стрелки на самом нижнем уровне. Такое тоннелирование называется "Не-в-родительской-диаграмме". Другим примером тоннелирования может быть ситуация, когда стрелка механизма мигрирует с верхнего уровня на нижний, причем на нижнем уровне этот механизм используется одинаково во всех работах без исключения. В этом случае стрелка механизма на нижнем уровне может быть удалена, после чего на родительской диаграмме она может быть затоннелирована ("Не-в-дочерней-работе").

Задание.

Исходя из результатов предыдущей лабораторной работы, создать все диаграммы в программе, расположить на них все блоки и дуги, описывающие заданный отдел. Получить законченную модель функционирования отдела.

Контрольные вопросы.

1. Что такое CASE-технологии, их достоинства и преимущества?
2. Проведите сравнительный анализ традиционной технологии разработки и разработки с помощью CASE-технологии.
3. Каковы основные объекты диаграмм функциональной модели по методологии IDEF0?

4. Что обозначают работы в диаграммах функциональной модели, как они отображаются по методологии IDEF0?
5. Для чего предназначены стрелки в диаграммах функциональной модели, каковы их типы и виды?
6. Для чего предназначен словарь стрелок?
7. Каковы типы связей работ по методологии IDEF0?
8. Что такое тоннелирование стрелок, для чего оно нужно, каковы виды тоннелирования?

Практическое занятие № 3.

Тема: Составление отчетов в пакете BPwin

BPwin имеет мощный инструмент генерации отчетов. Отчеты по модели вызываются из пункта меню Report. Всего имеется семь типов отчетов:

1. Model Report. Этот отчет 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.
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.
7. Model Consistency Report. Отчет, содержащий список синтаксических ошибок модели.

Синтаксические ошибки IDEF0 с точки зрения BPwin разделяются на три типа:

- во-первых, это ошибки, которые BPwin выявить не в состоянии. BPwin не позволяет анализировать синтаксис естественного языка (английского и русского) и смысл имен объектов и поэтому игнорирует ошибки этого типа. Выявление таких ошибок - ручная работа.

- ошибки второго типа BPwin просто не допускает. Например, каждая грань работы предназначена для определенного типа стрелок. BPwin просто не

позволит создать на диаграмме IDEF0 внутреннюю стрелку, выходящую из левой грани работы и входящую в правую грань.

- третий тип ошибок BPwin позволяет допустить, но отмечает их. Полный их список можно получить в отчете Model Consistency Report. Это единственный неопциональный отчет в BPwin. Список ошибок может содержать, например, неименованные работы и стрелки (unnamed arrow, unnamed activity), несвязанные стрелки (unconnected border arrow), неразрешенные стрелки (unresolved (square tunneled) arrow connections), работы, не имеющие по крайней мере одной стрелки выхода и одной стрелки управления, и т.д.

При выборе пункта меню, который соответствует какому-либо отчету, появляется диалог настройки отчета. Для каждого из семи типов отчетов он выглядит по-своему. Рассмотрим типичный диалог Arrow Report (рис.7). Раскрывающийся список Standard Reports позволяет выбрать один из стандартных отчетов. Стандартный отчет - это запоминаемая комбинация переключателей, флажков и других элементов управления диалога. Для создания собственного стандартного отчета необходимо задать опции отчета, ввести имя отчета в поле списка выбора и щелкнуть по кнопке New. BPwin сохраняет информацию о стандартном отчете в файле BPWINRPT.INI. Все определения этого файла доступны из любой модели. Единственное ограничение - свойства, определяемые пользователем (User Defined Properties). Они сохраняются в виде указателя и поэтому доступны только из родной модели. Стандартный отчет можно изменить или удалить.

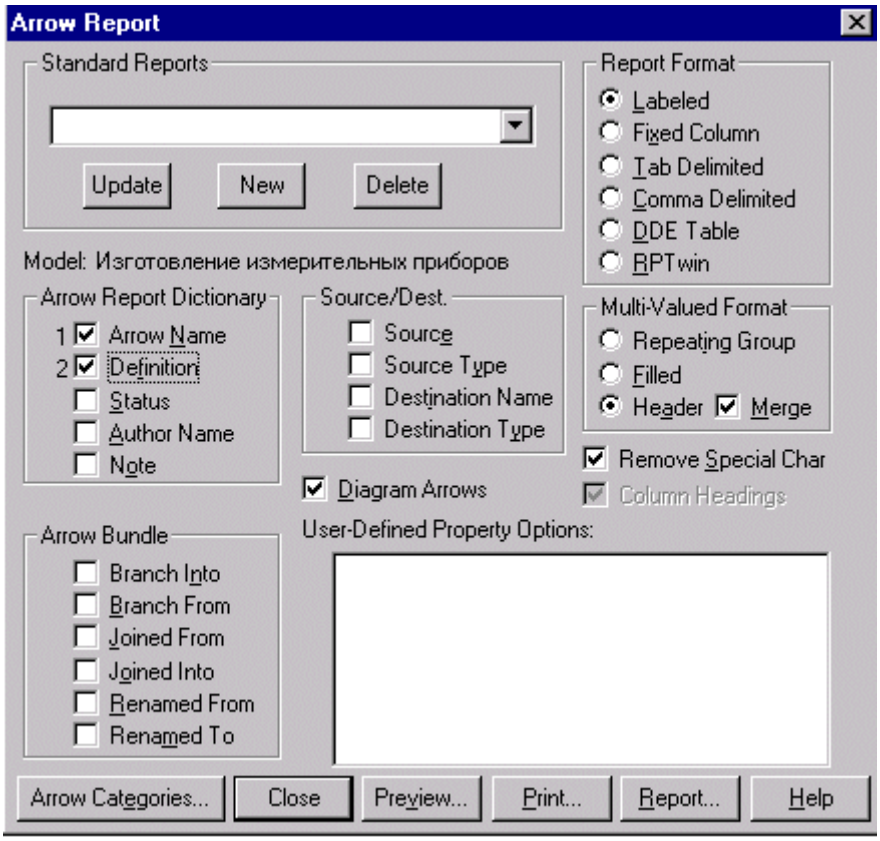

Рис.7.Диалог настройки отчета

В правом верхнем углу диалога находится группа управляющих элементов для выбора формата отчета. Доступны следующие форматы:

- Labeled - отчеты включают метку поля, затем, в следующей строке, печатается содержимое поля;
- Fixed Column - каждое поле печатается в собственной колонке;
- Tab-Comma Delimited - каждое поле печатается в собственной колонке. Колонки разделяются знаком табуляции или запятыми;
- DDE Table - данные передаются по DDE приложению, например MS Word или Excel;
- RPTwin - отчет создается в формате Platinum RPTwin - специализированного генератора отчетов, который входит в поставку BPwin.

Опция Ordering (на отчете по стрелкам отсутствует) сортирует данные по какому-либо значению. Опция Multi-Valued Format регулирует вывод полей в отчете при группировке данных:

- Repeating Group - детальные данные объединяются в одно поле, между значениями вставляется +.
- Filled - дублирование данных для каждого заголовка группы;
- Header (опция по умолчанию) - печатается заголовок группы, затем - детальная информация.

Задание.

По полученной модели получить основные отчеты: по дугам и блокам модели. Проанализировать полученные отчеты.

Контрольные вопросы.

1. Какие компоненты должны входить в полный комплекс CASE-средств, обеспечивающий поддержку жизненного цикла ПО?
2. По каким признакам можно классифицировать CASE-средства?
3. По каким основные типам классифицируются CASE-средства, какие конкретные системы им соответствуют?
4. Какие существуют типы отчетов в пакете BPwin, для чего каждый из них предназначен?
5. Какого рода синтаксические ошибки выявляет пакет BPwin?

Практическая работа № 4.

Тема: Изучение объектов DFD-диаграмм

Диаграммы потоков данных (DFD, Data Flow Diagramming) используются для описания документооборота и обработки информации. Подобно IDEF0, DFD представляет модельную систему как сеть связанных между собой работ. Их можно использовать как дополнение к модели IDEF0 для более наглядного отображения текущих операций документооборота в корпоративных системах обработки информации.
DFD описывает:

- функции обработки информации (работы, activities);
- документы (стрелки, arrows), объекты, сотрудников или отделы, которые участвуют в обработке информации;
- внешние ссылки (external references), которые обеспечивают интерфейс с внешними объектами, находящимися за границами моделируемой системы;
- таблицы для хранения документов (хранилище данных, data store).

В BPwin для построения диаграмм потоков данных используется нотация Гейна-Сарсона. Для того чтобы дополнить модель IDEF0 диаграммой DFD, нужно в процессе декомпозиции в диалоге Activity Box Count надавить на радио-кнопку DFD. В палитре инструментов на новой диаграмме появляются кнопки:

 - добавить в диаграмму внешнюю ссылку. Внешняя ссылка является источником или приемником данных извне модели;
 - добавить в диаграмму хранилище данных. Хранилище данных позволяет описать данные, которые необходимо сохранить в памяти прежде, чем использовать в работах;
 - ссылка на другую страницу. В отличие от IDEF0 инструмент off-page reference 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 - движение объектов (data flow), хранение объектов (data stores), поставка и распространение объектов. В отличие от IDEF0, где система рассматривается как взаимосвязанные работы, DFD рассматривает систему как совокупность предметов. Контекстная диаграмма часто включает работы и внешние ссылки. Работы обычно именуются по названию системы, например "Система обработки

информации". Включение внешних ссылок в контекстную диаграмму не отменяет требования методологии четко определить цель, область и единую точку зрения на моделируемую систему.

Работы.

В DFD работы представляют собой функции системы, преобразующие входы в выходы. Хотя работы изображаются прямоугольниками со скругленными углами, смысл их совпадает со смыслом работ IDEF0. Внешние сущности. Изображают входы в систему и/или выходы из нее. 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. Обычно такой прием используют, чтобы не рисовать слишком длинных и запутанных стрелок.

Стрелки (Потоки данных).

Стрелки описывают движение объектов из одной части системы в другую. Поскольку в DFD каждая сторона работы не имеет четкого назначения, как в IDEF0, стрелки могут подходить и выходить из любой грани прямоугольника работы. В DFD также применяются двунаправленные стрелки для описания диалогов типа "команда-ответ" между работами, между работой и внешней сущностью и между внешними сущностями.

Хранилище данных.

В отличие от стрелок, описывающих объекты в движении, хранилища данных изображают объекты в покое. В материальных системах хранилища данных изображаются там, где объекты ожидают обработки, например в очереди. В системах обработки информации хранилища данных являются механизмом, который позволяет сохранить данные для последующих процессов. Слияние и разветвление стрелок. В DFD стрелки могут сливаться и разветвляться, что позволяет описать декомпозицию стрелок. Каждый новый сегмент сливающейся или разветвляющейся стрелки может иметь собственное имя.

Построение диаграмм DFD.

Диаграммы DFD могут быть построены с использованием традиционного структурного анализа, подобно тому, как строятся диаграммы IDEF0. Сначала строится физическая модель, отображающая текущее состояние дел. Затем эта модель преобразуется в логическую модель, которая отображает требования к существующей системе. После этого строится модель, отображающая требования к будущей системе. И наконец, строится физическая модель, на основе которой должна быть построена новая система.

Альтернативным является подход, популярный при создании программного обеспечения, называемый событийным разделением, в котором различные диаграммы DFD выстраивают модель системы.
Логическая модель строится как совокупность работ и документирования того, что эти работы должны делать.
Модель окружения описывает систему как объект, взаимодействующий с событиями из внешних сущностей. Модель окружения обычно содержит описание цели системы, одну контекстную диаграмму и список событий. Контекстная диаграмма содержит один прямоугольник работы, изображающий систему в целом, и внешние сущности, с которыми система взаимодействует.

Наконец, модель поведения показывает, как система обрабатывает события. Эта модель состоит из одной диаграммы, в которой каждый прямоугольник изображает каждое событие из модели окружения. Хранилища могут быть добавлены для моделирования данных, которые необходимо запоминать между событиями. Потоки добавляются для связи с другими элементами, и диаграмма проверяется с точки зрения соответствия модели окружения. Полученные диаграммы могут быть преобразованы с целью более наглядного представления системы, в частности, работы на диаграммах могут быть декомпозированы.

Нумерация объектов.

В DFD номер каждой работы может включать префикс, номер родительской работы (A) и номер объекта. Номер объекта - это уникальный номер работы на диаграмме. Например, работа может иметь номер A.12.4. Уникальный номер имеют хранилища данных и внешние сущности независимо от их расположения на диаграмме. Каждое хранилище данных имеет префикс D и уникальный номер, например D5. Каждая внешняя сущность имеет префикс E и уникальный номер, например E4.

Задание.

По заданному отделу построить диаграмму верхнего уровня взаимодействия отдела с внешними данными.

Контрольные вопросы.

1. Какие CASE-средства наиболее известны на российском рынке программного обеспечения?
2. Каковы основные функции наиболее известного российского CASE-средства функционального моделирования?
3. В чем особенности CASE-средства Rational Rose?
4. В чем особенности DFD-диаграмм, что в них описывается?
5. В чем особенности объектов DFD-диаграмм?

6. В чем различия функциональной, логической, физической моделей, а также моделей окружения и поведения?

Практическое занятие № 5.

Тема: Изучение основных функций пакета ERwin. Создание логической модели

ERwin - средство концептуального моделирования БД, использующее методологию IDEF1X. ERwin реализует проектирование схемы БД, генерацию ее описания на языке целевой СУБД (ORACLE, Informix, Ingres, Sybase, DB/2, Microsoft SQL Server, Progress и др.) и реинжиниринг существующей БД. ERwin выпускается в нескольких различных конфигурациях, ориентированных на наиболее распространенные средства разработки приложений 4GL. Версия ERwin/OPEN полностью совместима со средствами разработки приложений PowerBuilder и SQLWindows и позволяет экспортировать описание спроектированной БД непосредственно в репозитории данных средств. Для ряда средств разработки приложений (PowerBuilder, SQLWindows, Delphi, Visual Basic) выполняется генерация форм и прототипов приложений. Сетевая версия ERwin ModelMart обеспечивает согласованное проектирование БД и приложений в рабочей группе.

Основные получаемые преимущества:

- существенное повышение скорости разработки за счет мощного редактора диаграмм, автоматической генерации базы данных, автоматической подготовки документации;
- нет необходимости ручной подготовки SQL-предложений для создания базы данных;
- возможность легко вносить изменения в модель при разработке и расширении системы;
- возможность автоматической подготовки отчетов по базе данных; важно, что эти отчеты всегда в точности соответствуют реальной структуре БД;
- разработчики прикладного программного обеспечения снабжены удобными в работе диаграммами;
- тесная интеграция со средствами 4GL позволяет уже на стадии информационного моделирования задавать отображение данных в приложениях;
- обратное проектирование позволяет документировать и вносить изменения в существующие информационные системы;
- поддержка однопользовательских СУБД позволяет использовать для персональных систем современные технологии, что значительно упрощает переход от настольных систем к системам в технологии клиент-сервер (upsizing).

Построение моделей в ERwin

Возможны две точки зрения на информационную модель и, соответственно, два уровня модели. Первый - логический уровень (точка зрения пользователя) означает прямое отображение фактов из реальной жизни. Например, люди, столы, отделы, собаки и компьютеры являются реальными объектами. Они именуются на естественном языке, с любыми разделителями слов (пробелы, запятые и т.д.). На физическом уровне модели рассматривается использование конкретной СУБД, определяются типы данных (например, целое или вещественное число), индексы для таблиц. ERwin предоставляет возможности создавать и управлять этими двумя различными уровнями представления одной диаграммы (модели), равно как и иметь много вариантов отображения на каждом уровне. Термин "логический уровень" в ERwin соответствует концептуальной модели.

Этапы построения информационной модели:

- определение сущностей;
- определение зависимостей между сущностями;
- задание первичных и альтернативных ключей;
- определение атрибутов сущностей;
- приведение модели к требуемому уровню нормальной формы;
- переход к физическому описанию модели: назначение соответствий имя сущности - имя таблицы, атрибут сущности - атрибут таблицы;
- задание триггеров, процедур и ограничений;
- генерация базы данных.

Erwin создает визуальное представление (модель данных) для решаемой задачи. Это представление может использоваться для детального анализа, уточнения и распространения документации, необходимой в цикле разработки. Однако ERwin далеко не только инструмент для рисования. ERwin автоматически создает базу данных (таблицы, индексы, хранимые процедуры, триггеры для обеспечения ссылочной целостности и другие объекты, необходимые для управления данными).

Создание сущности.

Для внесения сущности в модель необходимо щелкнуть по кнопке сущности на панели инструментов (Erwin Toolbox)  , затем - по тому месту на диаграмме, где необходимо расположить новую сущность. Щелкнув правой кнопкой мыши по сущности и выбрав из всплывающего меню пункт Entity Editor, можно вызвать диалог Entity Editor, в котором определяются имя, описание и комментарии сущности. Каждая сущность должна быть полностью определена с помощью текстового описания в закладке Definition. Эти определения полезны как на логическом уровне, поскольку позволяют понять, что это за объект, так и на физическом уровне, поскольку их можно экспортировать как часть схемы и использовать в реальной БД (CREATE

COMMENT on entity_name). Закладки Note, Note2, Note3, UDP (User Defined Properties - Свойства, определенные пользователем) служат для внесения дополнительных комментариев и определений к сущности.
В закладке Icon каждой сущности можно поставить в соответствие изображение, которое будет отображаться в режиме просмотра модели на уровне иконок и изображение, которое будет отображаться на всех других уровнях.

Закладка UDP диалога Entity Editor служит для определения свойств, определяемых пользователем (User - Defined Properties). При нажатии на кнопку  этой закладки вызывается диалог User - Defined Property Editor (также вызывается из меню Edit/UDPs). В нем необходимо указать вид объекта, для которого заводится UDP (диаграмма в целом, сущность, атрибут и т.д.) и тип данных. Для внесения нового свойства следует щелкнуть в таблице по кнопке  и внести имя, тип данных, значение по умолчанию и определение.

Создание атрибутов.

Для описания атрибутов следует, щелкнув правой кнопкой по сущности, выбрать в появившемся меню пункт Attribute Editor. Появится диалог Attribute Editor. Если щелкнуть по кнопке New, то в появившемся диалоге New Attribute можно указать имя атрибута, имя соответствующей ему в физической модели колонки и домен. Домен атрибута будет использоваться при определении типа колонки на уровне физической модели.

Для атрибутов первичного ключа в закладке General диалога Attribute Editor необходимо сделать пометку в окне выбора Primary Key.
Закладки Definition, Note и UDP несут те же функции, что и при определении сущности, но на уровне атрибутов.
Для большей наглядности диаграммы каждый атрибут можно связать с иконкой. Это можно сделать при помощи списка выбора Icon в закладке General. Очень важно дать атрибуту правильное имя. Атрибуты должны именоваться в единственном числе и иметь четкое смысловое значение.
Согласно синтаксису IDEF1X, имя атрибута должно быть уникальным в рамках модели (а не только в рамках сущности!). По умолчанию при попытке внесения уже существующего имени атрибута ERwin переименовывает его. Например, если атрибут Комментарий уже существует в модели, другой атрибут (в другой сущности) будет назван Комментарий/2, затем Комментарий/3 и т.д.
При переносе атрибутов внутри и между сущностями можно воспользоваться техникой drag&drop, выбрав кнопку  в палитре инструментов.

Создание связи.

Для создания новой связи следует выбрать идентифицирующую или неидентифицирующую связь в палитре инструментов (ERwin Toolbox), щелкнуть сначала по родительской, а затем по дочерней сущности. В палитре инструментов кнопка  соответствует идентифицирующей связи, кнопка  связи многие-ко-многим и кнопка  соответствует неидентифицирующей связи. Для редактирования свойств связи следует щелкнуть правой кнопкой мыши по связи и выбрать на контекстном меню пункт Relationship Editor. В закладке General появившегося диалога можно задать мощность, имя и тип связи.

Мощность связи (Cardinality) - служит для обозначения отношения числа экземпляров родительской сущности к числу экземпляров дочерней. Различают четыре типа мощности: общий случай, когда одному экземпляру родительской сущности соответствуют 0, 1 или много экземпляров дочерней сущности, 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, когда одному экземпляру родительской сущности соответствует заранее заданное число экземпляров дочерней сущности. По умолчанию символ, обозначающий мощность связи, не показывается на диаграмме. Для отображения имени следует в контекстном меню, которое появляется, если щелкнуть правой кнопкой мыши по любому месту диаграммы, не занятому объектами модели, выбрать пункт Display Options/Relationship и затем включить опцию Cardinality.

Тип связи (идентифицирующая/неидентифицирующая).

В IDEF1X 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. Когда рисуется идентифицирующая связь, ERwin автоматически преобразует дочернюю связь в зависимую. Зависимая сущность изображается прямоугольником со скругленными углами. Экземпляр зависимой сущности определяется только через отношение к 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 внешние ключи - (FK). При установлении неидентифицирующей связи

дочерняя сущность остается независимой, а атрибуты первичного ключа родительской сущности мигрируют в состав неключевых компонентов дочерней. Неидентифицирующая связь служит для связи независимых сущностей.

Идентифицирующая связь показывается на диаграмме сплошной линией с жирной точкой на дочернем конце связи, неидентифицирующая - пунктирной.

Для неидентифицирующей связи можно указать обязательность (Nulls в закладке General диалога Relationship Editor). В случае обязательной связи (No Nulls) при генерации схемы БД атрибут внешнего ключа получит признак NOT NULL, несмотря на то, что внешний ключ не войдет в состав первичного ключа дочерней сущности. В случае необязательной связи (Nulls Allowed) внешний ключ может принимать значение NULL. Необязательная неидентифицирующая связь помечается прозрачным ромбом со стороны родительской сущности

Имя связи (Verb Phrase) - фраза, характеризующая отношение между родительской и дочерней сущностями. Для связи один-ко-многим идентифицирующей или неидентифицирующей достаточно указать имя, характеризующей отношение от родительской к дочерней сущности (Parent-to-Child). Для связи многие-ко-многим следует указывать имена как Parent-to-Child, так и Child-to-Parent. Для отображения имени следует в контекстном меню, которое появляется, если щелкнуть правой кнопкой мыши по любому месту диаграммы, не занятому объектами модели, выбрать пункт Display Options/Relationship и затем включить опцию Verb Phrase.

Имя роли или функциональное имя (Rolename) - это синоним атрибута внешнего ключа, который показывает, какую роль играет атрибут в дочерней сущности. Задать имя роли можно в закладке Rolename/RI Actions диалога Relationship Editor.

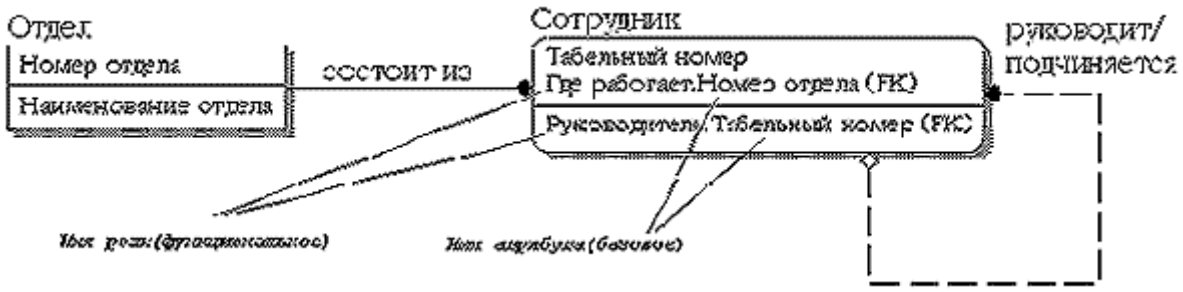

Рис.8. Имена ролей внешних ключей

В примере, приведенном на рис.8, в сущности Сотрудник внешний ключ Номер отдела имеет имя роли "Где работает", которое показывает,

какую роль играет этот атрибут в сущности. По умолчанию в списке атрибутов показывается только имя роли. Для отображения полного имени атрибута (как функционального имени, так и имени роли) следует в контекстном меню, которое появляется, если щелкнуть правой кнопкой мыши по любому месту диаграммы, не занятому объектами модели, выбрать пункт Display Options/Entities и затем включить опцию Rolename/Attribute. Полное имя показывается как функциональное имя и базовое имя, разделенные точкой (рис.8). Обязательным является применение имен ролей в том случае, когда два или более атрибутов одной сущности определены по одной и той же области, т.е. они имеют одну и ту же область значений, но разный смысл.

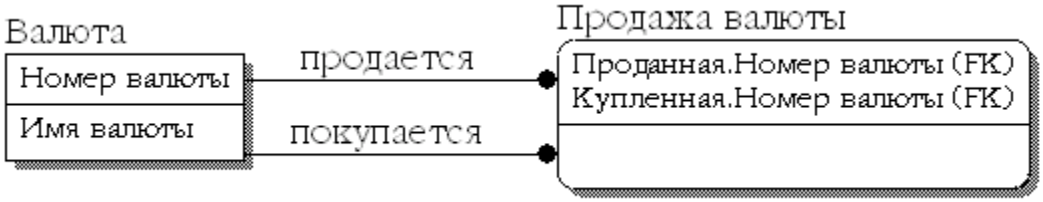

Рис.9. Случай обязательности имен ролей

На рис.9 сущность Продажа валюты содержит информацию об акте обмена валюты, в котором участвуют две валюты - проданная и купленная. Информация о валютах содержится в сущности Валюта. Следовательно, сущности Продажа валюты и Валюта должны быть связаны дважды, и первичный ключ - Номер валюты должен дважды мигрировать в сущность Валюта в качестве внешнего ключа. Необходимо различать эти атрибуты, которые содержат информацию о номере проданной и купленной валюты (имеют разный смысл), но ссылаются на одну и ту же сущность Валюта (имеют общую область значений). В примере на рис.9 атрибуты получили имена ролей Проданная и Купленная. Другим примером обязательного применения имен ролей являются рекурсивные связи, когда одна и та же сущность является и родительской и дочерней одновременно.

Правила ссылочной целостности (Referential Integrity (RI)) - логические конструкции, которые выражают бизнес-правила использования данных и представляют собой правила вставки, замены и удаления. Задать правила ссылочной целостности можно в закладке Rolename/RI Actions диалога Relationship Editor.
При генерации схемы БД на основе опций логической модели будут сгенерированы правила декларативной ссылочной целостности, которые должны быть предписаны для каждой связи, и триггеры, обеспечивающие ссылочную целостность.

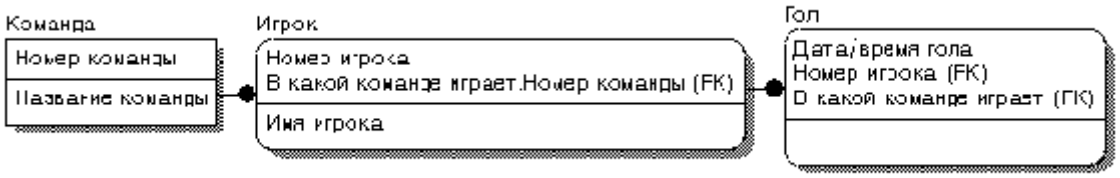

Рис.10. Миграция имен ролей

На рис.10 существует идентифицирующая связь между сущностями Команда и Игрок. Что будет, если удалить команду? Экземпляр сущности Игрок не может существовать без команды (атрибут первичного ключа *В какой команде играет. Номер команды* не может принимать значение NULL), следовательно нужно либо запретить удаление команды, пока в ней числится хотя бы один игрок, либо удалять вместе с командой и всех ее игроков. Такие правила удаления (Parent Delete) называются Parent Restrict (ограничение) и Parent Cascade (каскад). Сущности Игрок и Гол, в свою очередь, тоже связаны идентифицирующей связью и, если на удаление игрока наложено правило каскадного удаления всех записей о его голах, то при удалении команды будут удалены все игроки команды и все голы, забитые этими игроками. Связь многие-ко-многим возможна только на уровне логической модели данных. Такая связь обозначается сплошной линией с двумя точками на концах. Для внесения связи следует сначала нажать на кнопку  в палитре инструментов (ERwin Toolbox), а затем по очереди щелкнуть по обеим связанным сущностям. Связь многие-ко-многим должна именоваться (Verb Phrase) двумя фразами - в обе стороны. Это облегчает чтение диаграммы.

Создание ключей.

Каждый экземпляр сущности должен быть уникален и отличаться от других атрибутов.

Первичный ключ (primary key) - это атрибут или группа атрибутов, однозначно идентифицирующие экземпляр сущности. Атрибуты первичного ключа на диаграмме не требуют специального обозначения - это те атрибуты, которые находятся в списке атрибутов выше горизонтальной линии. При внесении нового атрибута в диалоге Attribute Editor для того, чтобы сделать его атрибутом первичного ключа, нужно включить флажок Primary Key в нижней части закладки General. На диаграмме ключевой атрибут можно внести в состав первичного ключа, воспользовавшись режимом переноса атрибутов (кнопка  в палитре инструментов). В одной сущности может оказаться несколько атрибутов или наборов атрибутов, претендующих на роль первичного ключа. Такие претенденты называются **потенциальными ключами (candidate key).** Ключи могут быть сложными, т.е. содержащими

несколько атрибутов. Сложные первичные ключи не требуют специального обозначения - это список атрибутов выше горизонтальной линии. При выборе первичного ключа предпочтение должно отдаваться более простым ключам, т.е. ключам, содержащим меньшее количество атрибутов. Многие сущности имеют только один потенциальный ключ. Такой ключ становится первичным. Некоторые сущности могут иметь более одного возможного ключа. Тогда один из них становится первичным, а остальные - альтернативными ключами.

Альтернативный ключ (Alternative Key) - это потенциальный ключ, не ставший первичным. Каждому ключу соответствует индекс, имя которого также присваивается автоматически. Имена ключа и индекса при желании можно изменить вручную.

На диаграмме атрибуты альтернативных ключей обозначаются как (Akn.m.), где n - порядковый номер ключа, m - порядковый номер атрибута в ключе. Когда альтернативный ключ содержит несколько атрибутов, (Akn.m.) ставится после каждого.

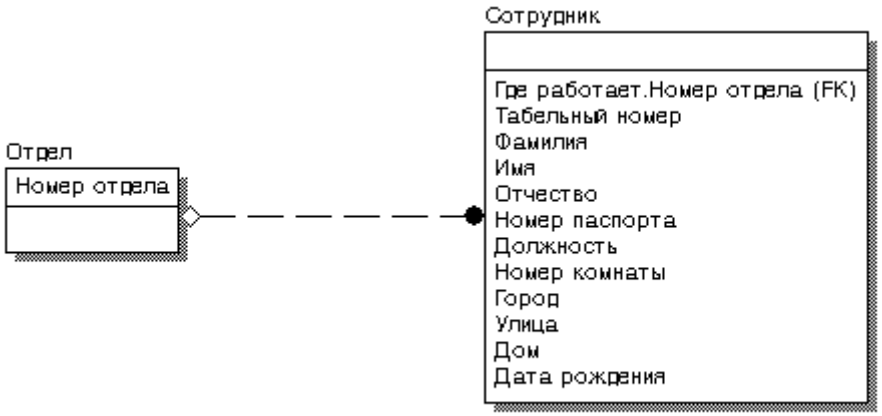

Рис.11. Сущность "Сотрудник" с отображением ключей

Внешние ключи (Foreign Key) создаются автоматически, когда связь соединяет сущности: связи образуют ссылку на атрибуты первичного ключа в дочерней сущности и эти атрибуты образуют внешний ключ в дочерней сущности (миграция ключа). Атрибуты внешнего ключа обозначаются символом (FK) после своего имени (рис.11). Атрибуты внешнего ключа Где работает.Номер отдела ("Где работает" - имя роли) сущности Сотрудник является атрибутом первичного ключа (PK) в сущности Отдел. Зависимая сущность может иметь один и тот же ключ из нескольких родительских сущностей. Сущность может также получить один и тот же внешний ключ несколько раз от одного и того же родителя через несколько разных связей. Когда ERwin обнаруживает одно из этих событий, он распознает, что два атрибута одинаковы, и помещает атрибуты внешнего

ключа в зависимой сущности только один раз. Это комбинирование или объединение идентичных атрибутов называется унификацией. Есть случаи, когда унификация нежелательна. Например, когда два атрибута имеют одинаковые имена, но на самом деле они отличаются по смыслу, и необходимо, что бы это отличие отражалось в диаграмме. В этом случае необходимо использовать имена ролей внешнего ключа (рис.9).

Домены.

Домен можно определить как совокупность значений, из которых берутся значения атрибутов. Каждый атрибут может быть определен только на одном домене, но на каждом домене может быть определено множество атрибутов. В понятие домена входит не только тип данных, но и область значений данных. Например, домен "Возраст" можно определить как положительное целое число и определить атрибут Возраст сотрудника как принадлежащий этому домену. В ERwin домен может быть определен только один раз и использоваться как в логической, так и в физической модели . На логическом уровне домены можно описать без конкретных физических свойств. На физическом уровне они получают специфические свойства, которые можно изменить вручную. Так, домен "Возраст" может иметь на логическом уровне тип Number, на физическом уровне домену будет присвоен тип INTEGER. Для создания домена в логической модели служит диалог Domain Dictionary Editor. Его можно вызвать из меню Edit/Domain Dictionary по кнопке, расположенной в верхней левой части закладки General диалога Attribute Editor. Для создания нового домена в диалоге Domain Dictionary Editor следует:

1. щелкнуть по кнопке New. Появляется диалог New Domain;
2. выбрать родительский домен из списка Domain Parent. Новый домен можно создать на основе уже созданного пользователем домена, либо на основе изначально существующего. По умолчанию Erwin имеет четыре предопределенных доменов (String, Number, Blob, Datetime). Новый домен наследует все свойства родительского домена. Эти свойства в дальнейшем можно переопределить;
3. набрать имя домена в поле Logical Name. Можно также указать имя домена на физическом уровне в поле Physical Name. Если физическое имя не указано, по умолчанию оно принимает значение логического имени;
4. щелкнуть по кнопке OK;

В диалоге Domain Dictionary Editor можно связать домен с иконкой, с которой он будет отображаться в списке доменов (Domain Icon), иконкой, с которой атрибут, определенный на домене будет отображаться в модели (Icon Inherited by Attribute). Каждый домен может быть описан в закладке Definition, снабжен комментарием в закладке Note или свойством

определенным пользователем в закладке UDP. ERwin имеет специальный инструмент, который значительно облегчает создание новых атрибутов в модели, используя описание доменов, - Independent Attribute Browser. Этот диалог вызывается (и скрывается) по горячему ключу CTRL+B. С его помощью можно выбрать в списке домен и по методу drag&drop перенести его в какую-либо сущность. В ней будет создан новый атрибут с именем, которое следует задать в окне Name Inherited by Attribute диалога Domain Dictionary Editor. Если значение поля не задано, по умолчанию принимается имя домена. На физическом уровне диалог Domain Dictionary Editor позволяет редактировать физические свойства домена. Имя этой закладки зависит от выбранного сервера БД. На ней можно задать конкретный тип данных, соответствующих домену, правила присвоения NULL - значений, правила валидации (правила проверки допустимых значений) и задания значения по умолчанию. Правила валидации и значения по умолчанию должны быть предварительно описаны и именованы. Для вызова диалогов редактирования правил валидации и значений по умолчанию служат кнопки  справа от соответствующего списка выбора (Valid и Default). Функции других закладок диалога Domain Dictionary Editor: **General.** Задание родительского домена (Domain Parent) и имени, присваиваемого колонке при ее создании с помощью Independent Column Browser. С помощью опции Phisical Only домен можно определить только на уровне физической модели.
Comment. Внесение комментария к атрибуту.
UDP. Свойства, определяемые пользователем.
Visual Basic - PowerBuilder. Задание специальных свойств домена для кодогенерации клиентского приложения.

Задание.

На основе ранее созданной функциональной модели и описания заданного отдела создать логическую модель с использованием пакета ERwin.

Контрольные вопросы.

1. Каково назначение пакета ERwin и его основные функции?
2. В чем состоят главные преимущества пакета ERwin?
3. Опишите этапы построения информационной модели.
4. Из каких элементов состоит диаграмма "сущность-связь"?
5. Опишите характеристики связей в методологии IDEF1X.
6. Какие типы ключей используются в пакете ERwin, каково их назначение?
7. Каково предназначение доменов, приведите примеры доменов различного вида.

Практическое занятие № 6.

Тема: Создание физической модели в ERwin.

Сервер

Физический уровень представления модели зависит от выбранного сервера. Для выбора СУБД служит редактор **Target Server** (меню Server/Target Server... доступно только на физическом уровне). ERwin поддерживает практически все распространенные СУБД, всего более 20 реляционных и нереляционных БД. Диалог **Target Server** позволяет задать тип данных и опцию NULL для новых колонок, а также правила ссылочной целостности, принимаемые по умолчанию. Группа кнопок Default Non-Key Null Option позволяет разрешить или запретить значения NULL для неключевых колонок. По умолчанию ERwin генерирует имена таблиц и индексов по шаблону на основе имен соответствующих сущностей и ключей логической модели. Окна Table Name Macro и Index Name Macro позволяют изменить шаблон генерации имен. Кнопка Reset Names вызывает диалог **Globally Reset DBMS Property,** который позволяет заменить все имена таблиц, связей, индек-сов, колонок и соответствующих свойств, заданных вручную, на значения по умолчанию. Имена таблиц и колонок по умолчанию будут сгенерированы на основе имен сущностей и атрибутов логической модели. Если в имени сущности или атрибута встречается пробел, он будет заменен на символ "_". При смене СУБД ERwin предлагает автоматически преобразовать тип данных, связанный с каждым атрибутом, на ближайший доступный для новой СУБД.

Таблицы

Для внесения новой таблицы в модель на физическом уровне служит кнопка на палитре инструментов. Связи между таблицами создаются так же, как на логическом уровне. Щелкнув правой клавишей мыши по таблице и выбрав во всплывающем меню пункты *Table Editor* или*Column Editor*, можно вызвать редакторы для задания свойств таблиц и колонок. ERwin автоматически создает имена таблиц и колонок на основе имен соответствующих сущностей и атрибутов, учитывая максимальную длину имени и другие синтаксические ограничения, накладываемые СУБД. При генерации имени таблицы или колонки по умолчанию все пробелы автоматически преобразуются в символы подчеркивания, а длина имени обрезается до максимальной длины, допустимой для выбранной СУБД. Все изменения, сделанные в *Table Editor* или *Column Editor*, не отражаются на именах сущностей и атрибутов, поскольку информация на логическом и физическом уровнях в ERwin хранится отдельно. Редактор Table Editor позволяет задать свойства любой таблицы модели, отличные от значения по

умолчанию, в том числе имя таблицы, синонимы, правила валидации, процедуры и т. д. Переключиться на другую таблицу можно при помощи раскрывающегося списка выбора в верхней части диалога. Окно Name служит для задания имени текущей таблицы. Окно Owner позволяет внести имя владельца таблицы, отличное от имени пользователя, производящего генерацию схемы БД. Окно выбора Physical Only служит для создания объектов только на физическом уровне. Если выбрана опция Generate, при генерации схемы БД будет выполняться команда CREATE TABLE. Кнопка DB Sync служит для немедленной синхронизации модели с системным каталогом БД.

Колонки

Для задания свойств колонок, отличных от значения по умолчанию, служит редактор *Column Editor*. Чтобы вызвать его, нужно щелкнуть правой клавишей мыши по таблице и выбрать во всплывающем меню пункт Column Editor. По умолчанию ERwin присваивает режимы нулевых значений всем не-ключевым колонкам, исходя из значений по умолчанию, устанавливаемых в редакторе **Target Server**. Для колонок первичного ключа и альтернативных ключей устанавливается режим NOT NULL. При создании связи колонки первичного ключа родительской таблицы мигрируют в состав колонок дочерней таблицы в качестве внешнего ключа. Кнопка Migrate вызывает диалог**Migrate Column Property** , который позволяет определять, какие характеристики мигрировавшей колонки будут сохранены в дочерней таблице.

Для переноса каких-либо характеристик колонки необходимо включить соответствующую опцию в диалоге Migrate Column Property, для отказа от переноса - выключить. Опциями диалога следует пользоваться осторожно, во-первых, потому, что новые свойства колонки перезаписывают старые, а во-вторых, поскольку установленные опции действуют в рамках всей диаграммы, а не только текущей таблицы.

Представления

Представления (view), или, как их иногда называют, временные или производные таблицы, представляют собой объекты БД, данные в которых не хранятся постоянно, как в таблице, а формируются динамически при обращении к представлению. Представление не может существовать само по себе, а определяется только в терминах одной или нескольких таблиц. Применение представлений позволяет разработчику БД обеспечить каждому пользователю или группе пользователей свой взгляд на данные, что решает проблемы простоты использования и безопасности данных. ERwin имеет специальные инструменты для создания и редактирования представлений. Палитра инструментов на физическом уровне содержит

кнопки внесения представлений и установления связей между таблицами и представлениями. Для внесения представления нужно щелкнуть по кнопке  в палитре инструментов, затем по свободному месту диаграммы. По умолчанию представление получает номер V_n, где n - уникальный порядковый номер представления. Для установления связи нужно щелкнуть по кнопке  , затем по родительской таблице и, наконец, по представлению. Связи с представлениями и прямоугольники представлений показываются на диаграмме пунктирными линиями. Для редактирования представления служит диалог **View Editor**. Для его вызова следует щелкнуть правой кнопкой мыши по представ-лению и выбрать в меню пункт *View Editor*.

Правила валидации и значения по умолчанию

ERwin поддерживает правила валидации для колонок, а также значение, присваеваемое колонкам по умолчанию. Правило валидации задает список допустимых значений для конкретной колонки и/или правила проверки допустимых значений. Значение по умолчанию - значение, которое нужно ввести в колонку, если никакое другое значение не задано явным образом во время ввода данных. С каждой колонкой или доменом можно связать значение по умолчанию (если выбранная СУБД поддерживает домены).

Если щелкнуть по кнопке  , расположенной справа от раскрывающегося списка Valid , появляется диалог **Validation Rule Editor**, который служит для задания правил валидации. В нем можно задать максимальное и минимальное значение и тип валидации (где проверять - на сервере или в клиентском приложении). Например, значение, вводимое в колонку *Age*, должно быть больше 18, но меньше 180. Для описания этого правила можно создать правило валидации с именем "Проверка_возраста", которое содержит выражение: *Age* BETWEEN 18 AND 180. Использование этого правила валидации гарантирует, что диапазон вводимых значений будет от 18 до 180. СУБД выдаст сообщение об ошибке, если вводимый возраст находится вне границ заданного диапазона. После создания правила валидации и значения по умолчанию можно присвоить одной или нескольким колонкам или доменам.

Индексы

Чтобы решить проблему поиска данных, СУБД использует особый объект, называемый индексом. Он подобен содержанию книги, которое указывает на все номера страниц, посвященных конкретной теме. Индекс содержит отсортированную по колонке или нескольким колонкам информацию и указывает на строки, в которых хранится конкретное значение колонки.

Например, если необходимо найти клиента по имени (рис.12), можно создать индекс по колонке

CustomerName таблицы **CUSTOMER.** В индексе имена клиентов будут отсортированы в алфавитном порядке. Для имени индекс будет содержать ссылку, указывающую, в каком месте таблицы хранится эта строка. Для поиска клиента серверу направляется запрос с критерием поиска (***CustomerName***="Иванов"). При выполнении запроса СУБД просматривает индекс, вместо того чтобы просматривать по порядку все строки таблицы **CUSTOMER.** Поскольку значения в индексе хранятся в определенном порядке, просматривать нужно гораздо меньший объем данных, что значительно уменьшает время выполнения запроса. Индекс можно создать для всех колонок таблицы, по которым часто производится поиск.

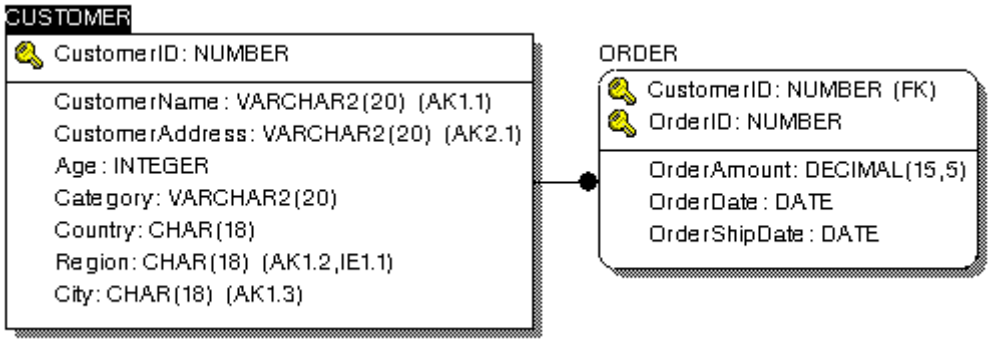

Рис.12. Поиск с помощью индекса

При генерации схемы физической БД ERwin автоматически создает отдельный индекс на основе первичного ключа каждой таблицы, а также на основе всех альтернативных ключей, внешних ключей и инверсионных входов, поскольку эти колонки наиболее часто используются для поиска данных. Можно отказаться от генерации индексов по умолчанию и для повышения производительности создать собственные индексы. Администратор СУБД должен анализировать наиболее часто выполняемые запросы и создавать индексы с различными колонками и порядком сортировки для увеличения эффективности поиска при работе конкретных приложений.

Изменить характеристики существующего индекса или создать новый можно в редакторе **Index Editor**. Для его вызова следует щелкнуть правой кнопкой мыши по таблице и выбрать во всплывающем меню пункт Index.

Прямое и обратное проектирование

Процесс генерации физической схемы БД из логической модели данных называется прямым проектированием (Forward Engineering). При генерации физической схемы ERwin включает триггеры ссылочной целостности, хранимые процедуры, индексы, ограничения и другие возможности, доступные при определении таблиц в выбранной СУБД. Процесс генерации логической модели из физической БД называется обратным проектированием (Reverse Engineering). ERwin позволяет создать модель данных путем обратного проектирования имеющейся БД. После того как модель создана, можно переключиться на другой сервер (модель будет конвертирована) и про-извести прямое проектирование структуры БД для другой СУБД. Кроме режима прямого и обратного проектирования ERwin поддерживает синхронизацию между логической моделью и системным каталогом СУБД на протяжении всего жизненного цикла создания ИС. Для генерации системного каталога БД следует выбрать пункт меню *Tasks/Forward Engineer/Schema Generation* или нажать кнопку  на панели инструментов. Появляется диалог **Schema Generation** . Диалог Schema Generation имеет три закладки:
Options. Служит для задания опций генерации объектов БД - триггеров, таблиц, представлений, колонок, индексов и т. д. Для задания опций генерации какого-либо объекта следует выбрать объект в левом списке закладки. после чего включить соответствующую опцию в правом списке .
В закладке **Summary** отображаются все опции, заданные в закладке Options. Список опций в Summary можно редактировать так же, как и в Options.
Comment. Позволяет внести комментарий для каждого набора опций. Каждый набор опций может быть именован (окно Option Set, кнопки New, Rename и Delete) и использован многократно.
Кнопка Preview вызывает диалог **Schema Generation Preview**, в котором отображается SQL-скрипт, создаваемый ERwin для генерации системного каталога СУБД. Нажатие на кнопку Generate приведет к запуску процесса генерации схемы. Кнопка Print предназначена для вывода на печать создаваемого ERwin SQL-скрипта. нопка Report сохраняет тот же скрипт в ERS или SQL текстовом файле. Эти команды можно в дальнейшем редактировать любым текстовым редактором и выполнять при помощи соответствующей утилиты сервера.
Кнопка Generate запускает процесс генерации схемы. Возникает диалог связи с БД, устанавливается сеанс связи с сервером и начинает выполняться SQL-скрипт. При этом возникает диалог **Generate Database Schema.** Для выполнения обратного проектирования следует выбрать пункт меню T*asks/Reverse Engineer*. При этом возникает диалог **ERwin Template Selection**, в ко-тором нужно выбрать шаблон диаграммы, затем диалог выбора СУБД и, наконец, диалог задания опций обратного проектирования Reverse Engineer - Set Options . В диалоге *Reverse Engineer - Set Options* можно задать следующие опции:

Группа **Reverse Engineer From** позволяет задать источник обратного проектирования - БД или SQL(DDL)-скрипт. При помощи кнопки Browse можно выбрать текстовый файл, содержащий SQL-скрипт.

Группа **Items to Reverse Engineer** позволяет задать объекты БД, на основе которых будет создана модель. При помощи списка выбора Option Set, а также кнопок New, Update и Delete можно создавать и редактировать име-нованные конфигурации объектов БД, которые могут быть использованы многократно при других сеансах обратного проектирования.

Группа **Reverse Engineer** (доступна только при обратном проектировании из БД) позволяет включить в модель системные объекты (окно выбора System Objects) и установить фильтр на извлекаемые таблицы по их владельцу. В процессе работы модель может изменяться и дополняться. С другой стороны, системный каталог БД может редактироваться другими проектировщиками. В результате спустя некоторое время после последнего сеанса обратного проектирования могут возникнуть расхождения между реальным состоянием системного каталога и моделью данных.

Для **синхронизации системного каталога** БД и текущей модели следует выбрать пункт меню *Tasks/Complete* Compare или нажать кнопку  на панели инструментов. Возникает диалог **Complete Compare - Set Options**, который во многом похож на описанный выше диалог **Reverse Engineer-Set Options**. Разница заключается в том, что в отличие от обратного проек-тирования сравнивать текущую модель можно не только с БД или SQL-скриптом, но и с другой моделью ERwin, хранящейся в файле или репозитории ModelMart.

Задание.

На основе логической модели автоматически получить в пакете ERwin физическую модель. Модифицировать модель по различным параметрам - серверам, таблицам, представлениям и т.п. Представить отчет в виде исходной и модифицированной моделей.

Контрольные вопросы.

1. Когда возникает необходимость в редактировании физической модели?
2. Для чего предназначены представления, как их можно создать?
3. Что такое правила валидации, каким образом они задаются?
4. Каким образом в СУБД предусмотрено ускорение поиска информации?
5. Какой смысл в обратном проектировании базы данных? Что создается в результате этого процесса?

Практическое занятие № 7.

Тема: Создание отчетов в пакете ERwin

Для генерации отчетов в ERwin имеется эффективный и простой в использовании инструмент - Report Browser. Он позволяет выполнять предопределенные отчеты (объединенные по типам), сохранять результаты их выполнения, создавать собственные отчеты, печатать и экспортировать их в распространенные форматы. Каждый отчет может быть настроен индивидуально, данные в нем могут быть отсортированы и отфильтрованы. Диалог Report Browser вызывается кнопкой  в панели инструментов ERwin. Диалог Report Browser имеет собственное меню и панель инструментов. Назначение кнопок панели инструментов показано в табл.5.

Кнопки панели инструментов Report Browser

Табл.3

кнопки	**назначение кнопок**
	Создание нового отчета или папки
	Печать отчета
	Просмотр результата выполнения отчета
	Выполнение отчета
	Фиксация изменений (для редактируемого отчета)
	Поиск элементов отчета: задание условий поиска, поиск следующей строки и поиск другого отчета, соответствующего строке
	Включение и выключение дерева отчетов
	Показать список выполненных отчетов в хронологическом порядке
	Перейти к предыдущему отчету (при создании нового отчета на основе строки существующего)
	Выбор колонок и сортировка выполненного отчета
	Ассоциирование строки отчета с иконкой
	Сохранение выполненного отчета в виде представления

В верхней левой части диалога расположено окно, отображающее дерево отчетов. Отчеты могут быть сгруппированы в папки. Каждый отчет может включать несколько результирующих наборов данных, каждое из которых генерируется при очередном выполнении отчета. Каждый элемент дерева помечен иконкой:

 - папка;

 - отчет;

- редактируемый отчет;
- результирующий набор данных;
- представление.

По умолчанию Report Browser содержит предварительно определенные отчеты, позволяющие наглядно представить информацию об основных объектах модели данных - как логической, так и физической. Для выполнения отчета достаточно дважды щелкнуть по нему в дереве отчетов или щелкнуть по соответствующей кнопке на панели инструментов. Результат выполнения отчета будет отображен в правом окне диалога Report Browser. Иконка результирующего набора будет также добавлена в дерево отчетов.

В левом нижнем окне Report Browser отображается комментарий к отчету (вносится в диалоге ERwin Report Editor). В нижней части диалога содержится дополнительная панель инструментов для управления деревом отчетов (табл.6)

Кнопки нижней панели инструментов Report Browser

Табл.4

кнопки	**назначение кнопок**
	Редактировать выделенный отчет
	Удалить отчет
	Показать только верхний уровень дерева
	Сделать выбранную папку корнем дерева (показать только выбранную ветку дерева)
	Сделать корнем дерева родительскую папку (по отношению к выбранной)

Создание нового отчета

Для создания нового отчета следует выбрать пункт меню File/New ERwin Report или щелкнуть по кнопке  на панели инструментов. Появляется диалог ERwin Report Editor.
В поле Name следует внести имя отчета. Категория отчета (Category) указывает на тип объектов модели, по которым будет создаваться отчет (атрибуты, сущности, домены, связи и т. д.). Закладки Definition и Note служат соответственно для внесения определения и комментария к отчету. Закладка Options отображает информацию, которая будет включена в отчет. В левой части закладки находится иерархический список категорий (Category). Папки в этом списке могут раскрываться и сворачиваться. Окно выбора ☑ позволяет включить соответствующий пункт

списка в отчет. Иконка показывает, что соответствующую колонку в полученном отчете можно будет редактировать. Папка с символом позволяет выбрать условия фильтрации данных отчета, а с символом - условия сортировки.
Кроме списка, закладка содержит следующие элементы управления:

- группу **Options** - позволяет выбрать режим отображения элементов в списке - показывать все возможные или только выбранные;
- Collapse All - сворачивает все папки списка;
- Clear All - отменяет все предварительно выбранные опции;
- Show Selected - раскрывает папки с выбранными опциями.

После щелчка по кнопке OK отчет будет добавлен в список отчетов диалога Report Browser. Для выполнения отчета нужно либо дважды щелкнуть по его имени в списке, либо щелкнуть по кнопке в палитре инструментов.

Существующий отчет, в том числе предопределенный, тоже можно изменить с помощью редактора, если в списке щелкнуть правой кнопкой мыши по имени отчета и выбрать во всплывающем меню пункт Edit ERwin Report. Полученный после выполнения отчета результирующий набор данных можно отформатировать, распечатать, экспортировать или сохранить в виде представления. Для форматирования результирующего набора данных следует в списке щелкнуть правой кнопкой мыши по имени набора и выбрать во всплывающем меню пункт Edit report format. В появляющемся диалоге Report Format можно изменить сортировку данных, очередность колонок, сделать колонку невидимой, задать ее стиль. Для редактирования результирующего набора данных следует в списке щелкнуть правой кнопкой мыши по имени набора и выбрать во всплывающем меню пункт Export result set. Допустимые форматы экспорта:

- CSV - текстовый файл;
- HTLM;
- DDE - экспорт в MS Word или MS Excel;
- RPTwin - экспорт в специализированный генератор отчетов.

После форматирования и настройки результирующего набора данных его можно сохранить в качестве именованного представления. Использование представлений облегчает использование отчетов, поскольку все настройки достаточно сделать один раз. Каждый отчет может иметь несколько представлений. Для создания представления следует установить фокус в списке на нужный набор и щелкнуть по кнопке на панели инструментов. В диалоге Save View следует указать имя и определение представления. После щелчка по кнопке OK представление добавится в список отчетов.

Задание.

Создать отчеты по следующим типам: атрибуты, сущности, домены, связи. Внести определения и комментарии к отчетам. Полученный после выполнения отчета результирующий набор данных отформатировать, распечатать, сохранить в виде представления.

Контрольные вопросы.

1. Каковы основные возможности генератора отчетов Report Browser?
2. Какие недостатки в работе данного пакета Вы видите?