

**СОЗДАНИЕ БАЗ ДАННЫХ
С ИСПОЛЬЗОВАНИЕМ ПАКЕТА ERWIN**

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

Содержание

Введение.....	4
Цель работы	5
Назначение пакета ERWin.....	5
Описание интерфейса программы ERWin	5
Основные приемы работы с пакетом ERWin	8
Работа с сущностями.....	8
Работа с атрибутами.....	9
Работа со связями	10
Пример выполнения работы.....	11
Формулировка задачи	11
Порядок выполнения работы	11
Задание	19
Варианты заданий	19

Введение

Создание современных информационных систем представляет собой сложнейшую задачу, решение которой требует применения специальных методик и инструментов. Неудивительно, что в последнее время среди системных аналитиков и разработчиков значительно вырос интерес к CASE-технологиям и инструментальным CASE-средствам, позволяющим максимально систематизировать и автоматизировать все этапы разработки программного обеспечения.

Технология создания информационных систем (ИС) предъявляет особые требования к методикам реализации и программным инструментальным средствам, а именно:

- Реализацию проектов по созданию ИС принято разбивать на стадии анализа (прежде чем создавать ИС, необходимо понять и описать бизнес-логику предметной области), проектирования (необходимо определить модули и архитектуру будущей системы), непосредственного кодирования, тестирования и сопровождения. Известно, что исправление ошибок, допущенных на предыдущей стадии, обходится примерно в 10 раз дороже, чем на текущей, откуда следует, что наиболее критическими являются первые стадии проекта. Поэтому крайне важно иметь эффективные средства автоматизации ранних этапов реализации проекта.
- Проект по созданию сложной ИС невозможно организовать в одиночку. Коллективная работа существенно отличается от индивидуальной, поэтому при реализации крупных проектов необходимо иметь средства координации и управления коллективом разработчиков.
- Жизненный цикл создания сложной ИС сопоставим с ожидаемым временем ее эксплуатации. Другими словами, в современных условиях компании перестраивают свои бизнес-процессы примерно раз в два года, столько же требуется (если работать в традиционной технологии) для создания ИС. Может оказаться, что к моменту сдачи ИС она уже никому не нужна, поскольку компания, ее заказавшая, вынуждена перейти на новую технологию работы. Следовательно, для создания ИС жизненно необходим инструмент, значительно (в несколько раз) уменьшающий время разработки ИС.

Вследствие значительного жизненного цикла может оказаться, что в процессе создания системы внешние условия изменились. Обычно внесение изменений в проект на поздних этапах создания ИС - весьма трудоемкий и дорогостоящий процесс. Поэтому для успешной реализации крупного проекта необходимо, чтобы инструментальные средства, на которых он реализуется, были достаточно гибкими к изменяющимся требованиям.

На современном рынке средств разработки ИС достаточно много систем, в той или иной степени удовлетворяющих перечисленным требованиям. CASE-средства ERwin и PRwin, разработанные фирмой PLATINUM, входят в число лучших на сегодняшний день.

ERwin - средство разработки структуры базы данных (БД). ERwin сочетает графический интерфейс Windows, инструменты для построения ER-диаграмм, редакторы для создания логического и физического описания модели данных и прозрачную поддержку ведущих реляционных СУБД и настольных баз данных. С помощью ERwin можно создавать или проводить обратное проектирование (реинжиниринг) баз данных.

Возможны две точки зрения на информационную модель и, соответственно, два уровня модели. Первый - логический (точка зрения пользователя) - описывает данные, задействованные в бизнесе предприятия. Второй - физический - определяет представление информации в БД. ERwin объединяет их в единую диаграмму, имеющую несколько уровней представления.

Цель работы

Целью работы является проведение процесса информационного моделирования для заданной предметной области с помощью инструментальной среды ERWin.

Назначение пакета ERWin

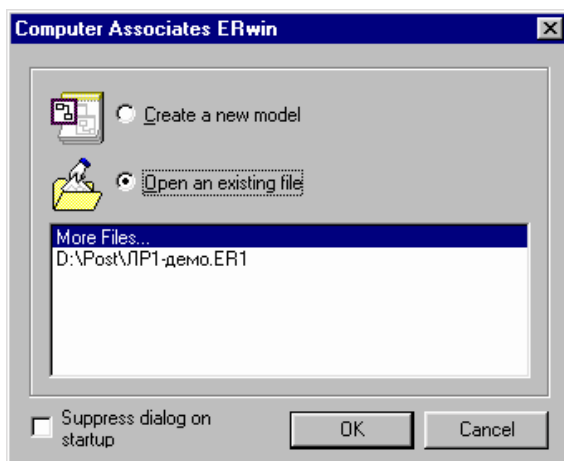
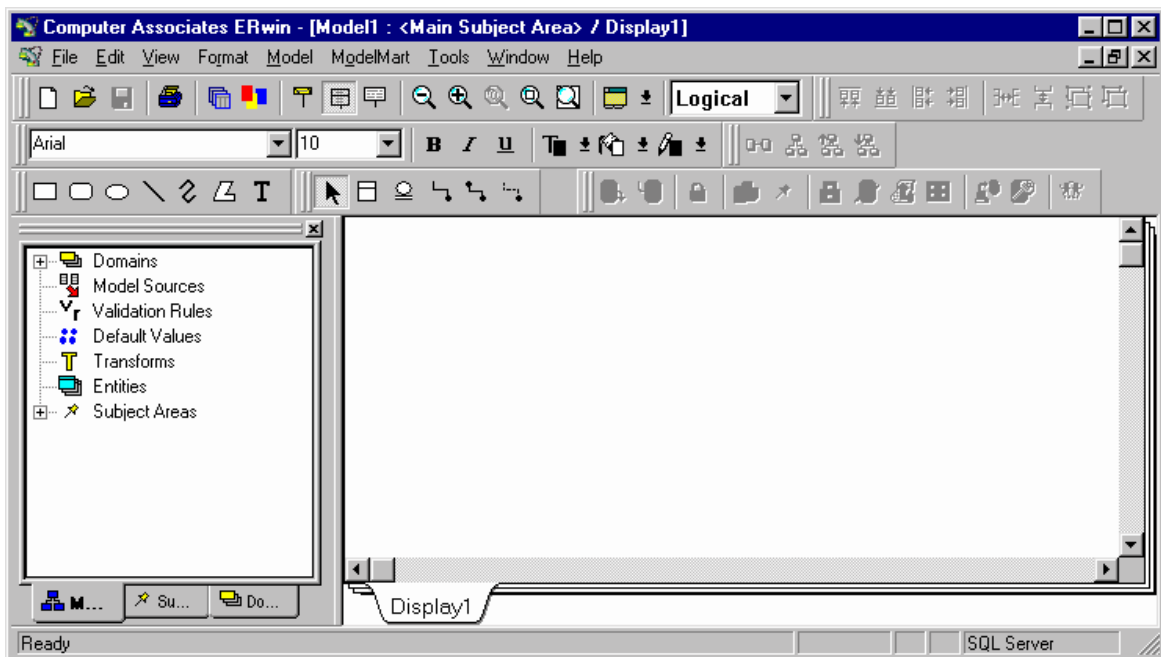
Пакет ERWin предназначен для создания информационной модели базы данных. Он позволяет формировать модель данных с учетом всех связей, а также ограничений на исходные данные.

При работе с пакетом ERWin пользователю вначале предлагается выбрать тип создаваемой модели: только логическая / только физическая / включающая логическую и физическую. После этого пользователь получает возможность формировать структуру будущей базы данных с использованием средств пакета.

Кроме того, пакет ERWin позволяет созданную модель данных перенести в выбранную базу данных в автоматическом режиме. При этом отпадает необходимость вручную создавать структуру таблиц и связей между ними. Возможна и обратная операция – из существующей базы данных сформировать модель данных в программе.

Описание интерфейса программы ERWin

Основное окно программы имеет следующий вид:



После запуска программы ERWin на экране появится окно программы с диалогом, предлагающий открыть существующую модель или создать новую.

Если пользователь выбрал открытие существующей модели, ему предложат стандартный диалог Windows для выбора файла модели. Либо пользователь может сразу выбрать имя файла из списка недавно использованных.

При выборе пункта «*Create a new model*» пользователю предлагается выбрать тип

создаваемой модели: логическая или физическая, а также воспользоваться одним из ранее созданных шаблонов. Если среди опций была выбрана модель типа «*Physical*» или «*Logical/Physical*», будет предложено выбрать также тип целевой базы данных, куда будет транслироваться готовая модель.

Далее на экране появляется основное окно программы (Рис. 1).

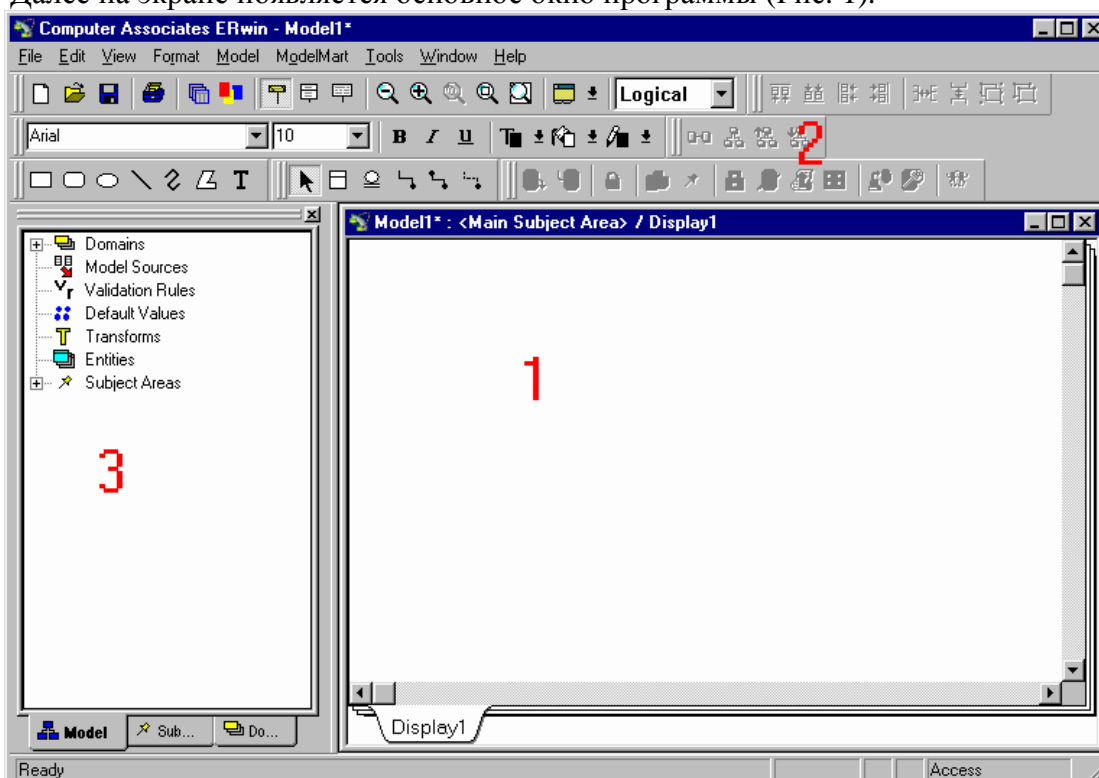


Рис. 1. Основное окно ERWin

Основное окно программы содержит следующие части:

- 1) Область модели данных или рабочая область
- 2) Панели инструментов
- 3) Браузер элементов модели

Рассмотрим подробнее содержимое каждой из частей программы:

Область модели данных – содержит собственно разрабатываемую модель. Если создана новая модель – это окно пустое, если открыта существующая – содержит сущности и связи между ними.

Панели инструментов: эти панели содержат практически все используемые при работе элементы. По умолчанию все панели отображаются на экране. При необходимости пользователь может отключить или, наоборот, включить требуемые модели, используя меню «*View – Toolbars*». Имеются следующие панели инструментов:

- **Standard** – содержит кнопки для управления файлами (новый, открыть, сохранить), кнопки отображения сущностей (только сущности, сущности и атрибуты, сущности и их определение), кнопки управления масштабом изображения, переключатель уровней модели (логический/физический).
- **Toolbox** – инструментальные кнопки создания элементов модели, сущностей и связей. Набор кнопок незначительно различается для логической и физической модели. Содержит кнопки: стрелка – выбор объекта, создание независимой сущности, создание связи-подкатегории, создание сущности-представления, создание идентифицирующей и неидентифицирующей связи «один-к-одному», создание связи «многие-ко-многим», создание связи для сущности-представления.

- **Font & Color** – управление шрифтом, начертанием и цветом различных элементов модели. Позволяет выделить цветом и шрифтом какие-либо смысловые части модели для облегчения восприятия.
- **ModelMart** – панель кнопок специального инструментального средства, предназначенного для связывания пакета BPWin и пакета ERWin.
- **Transforms** – преобразования сущностей (объединение, разделение и т.д.)
- **Drawing** – дополнительные рисуемые элементы (линии, прямоугольники, круги и т.д.)
- **Alignment** – выравнивание элементов модели

Браузер элементов модели помогает ориентироваться среди множества элементов модели, выделяя и группируя их. Имеет три вкладки: «**Model**», «**Subject Areas**» и «**Domains**».

Вкладка «**Model**» объединяет все элементы, в том числе элементы вкладок «**Subject Areas**» и «**Domains**». Здесь можно просмотреть список всех сущностей, связей, узнать, какая сущность с какой связана, и каким отношением (родительская/дочерняя), просмотреть созданные правила проверки значений, типы данных и т.д.

Вкладка «**Subject Areas**» содержит список предметных областей. Предметная область – это выборка из полной модели отдельных сущностей и их областей только тех, которые относятся к заданной области. Например, можно из общей информационной модели вуза выбрать только те сущности и связи, которые относятся, например, к деятельности деканата.

Вкладка «**Domains**» позволяет просматривать используемые в модели типы данных для атрибутов.

При создании новой модели пользуются панелью инструментов «**Toolbox**» (Рис. 2).



Рис. 2. Панели **Toolbox** для логического (а) и физического (б) уровней модели

Рассмотрим элементы этой панели (слева направо по порядку).

Для логического уровня:

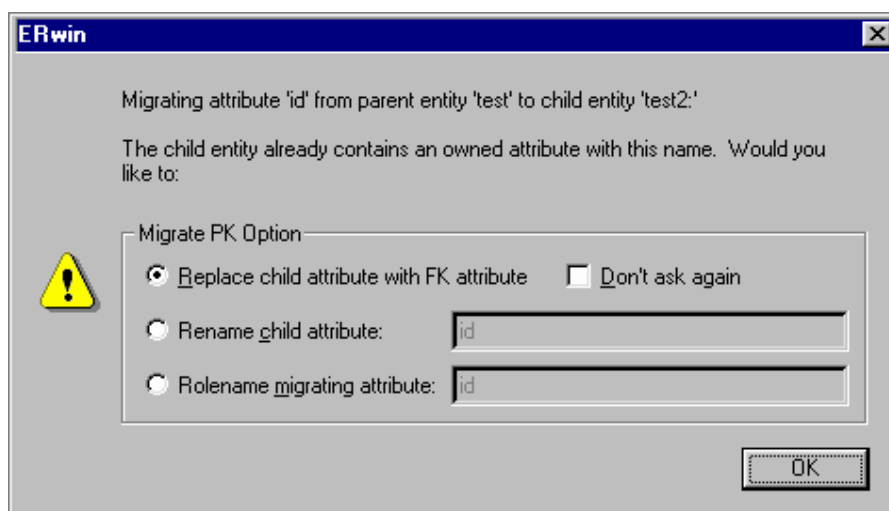
- **Select** – выбор объектов в модели
- **Create entity** – создание новой сущности
- **Create complete sub-category** – создание связки-разветвления
- **Identifying relationship** – создание идентифицирующей связи «один-ко-многим».
- **Many-to-many relationship** – создание связи «многие-ко-многим»
- **Non-identifying relationship** – создание неидентифицирующей связи «один-ко-многим».

Для создания сущности надо выбрать соответствующую кнопку, после чего щелкнуть мышью на свободном месте. Создастся новая сущность, при этом активизируется ввод ее имени. После ввода имени и нажатия Enter выделение переходит в часть ввода ключевых атрибутов. Имена атрибутов также набираются текстом. После ввода имени ключевого атрибута можно либо клавишей Enter добавить еще один ключевой атрибут, либо клавишей Tab перейти к вводу неключевых атрибутов.

Для создания связи между сущностями надо выбрать требуемый тип связи и щелкнуть последовательно на родительской и дочерней сущности. Для связей «многие-ко-многим» порядок щелканья безразличен. При создании связи «один-ко-многим» ключевые атрибуты родительской таблицы переходят в качестве внешних ключей в дочернюю таблицу. Если в дочерней таблице уже есть атрибуты с таким именем, пользователю предложат сделать выбор из следующих вариантов:

- Заменить существующий атрибут. В дочерней таблице существовавший атрибут становится внешним ключом для первичного ключа родительской таблице.
- Переименовать атрибут в дочерней таблице. При этом внешний ключ, приходящий из дочерней таблице, остается неизменным, меняется имя дочернего атрибута, совпадающее внешним ключом.
- Переименовать атрибут, переходящий из родительской таблице. При этом в родительской таблице атрибут свое имя не меняет, в дочерней таблице появляется новый атрибут с указанным именем.

При разработке БД чаще всего применяется именно третий вариант – переименование мигрирующего атрибута.



Для физического уровня панель инструментов содержит следующие кнопки:

- **Select** – выбор объектов в модели
- **Create entity** – создание новой сущности
- **View table** – создание сущности-представления
- **Identifying relationship** – создание идентифицирующей связи «один-ко-многим».
- **View relationship** – создание связи для сущности-представления.
- **Non-identifying relationship** – создание неидентифицирующей связи «один-ко-многим».

Сущность-представление – это виртуальная сущность, получаемая выборкой атрибутов из существующих сущностей. В терминах баз данных соответствует запросу.

Использование этих инструментов производится также, как в логической модели.

Основные приемы работы с пакетом ERWin

Работа с сущностями

Чтобы добавить в модель новую сущность, необходимо на панели «*Toolbox*» выбрать инструмент «*Create entity*», после чего щелкнуть мышью на свободном месте рабочей области. Появится рамка новой сущности, и программа перейдет в режим ввода имени сущности. Завершается ввод имени нажатием на клавишу Enter.

После ввода имени сущности в зависимости от режима отображения возможны три варианта:

- 1) в режиме отображения имен сущностей программа просто отобразит вновь добавленную сущность;

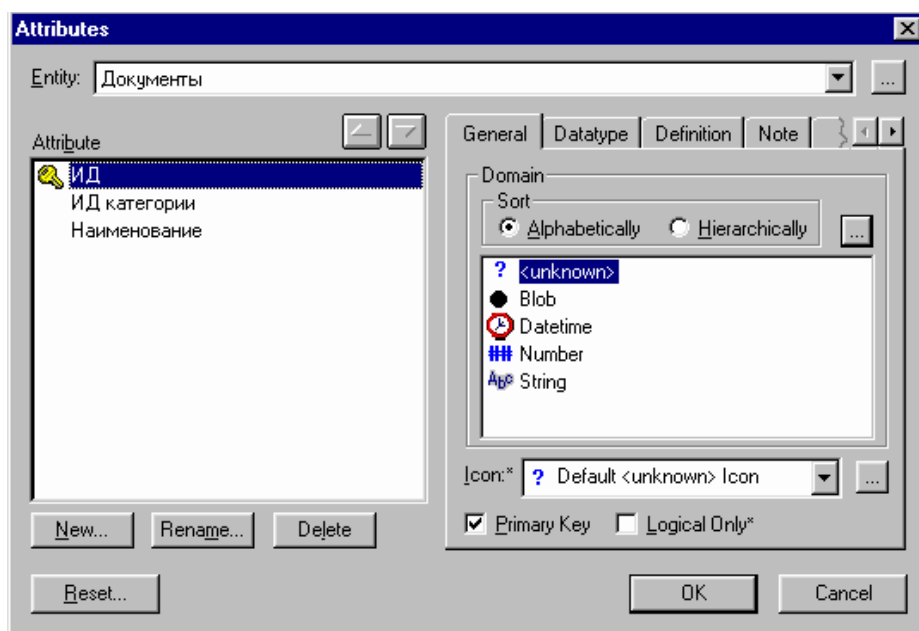
- 2) в режиме отображения атрибутов предложит вводить имена для ключевых атрибутов (этот режим обычно включен по умолчанию);
- 3) в режиме ввода определений перейдет к вводу определения для вновь добавленной сущности.

В режиме ввода атрибутов после ввода каждого имени атрибута нажатие клавиши Enter приводит к добавлению нового атрибута. Сначала вводятся имена ключевых атрибутов. После ввода всех ключевых атрибутов можно нажать клавишу Tab и перейти к вводу неключевых атрибутов.

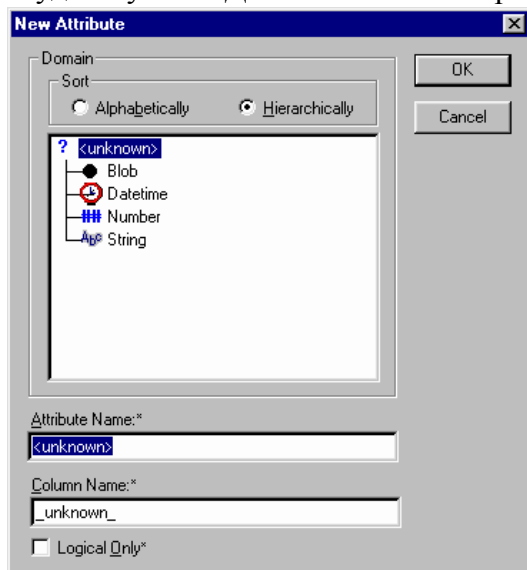
Работа с атрибутами

После создания сущности и задания ее имени необходимо указать атрибуты, относящиеся к данной сущности. Это можно сделать двумя способами: после создания сущности в режиме отображения атрибутов ввести перечень ключевых и неключевых атрибутов, или после добавления сущности в режиме редактора атрибутов.

Редактор атрибутов вызывается двойным щелчком мышью на сущности.



Если модель была создана без атрибутов, список атрибутов в левой части диалога будет пустой. Добавить новый атрибут можно кнопкой «New...» в нижней левой части диалога. Если атрибуты уже есть, их можно переименовать кнопкой «Rename...» или удалить кнопкой «Delete».



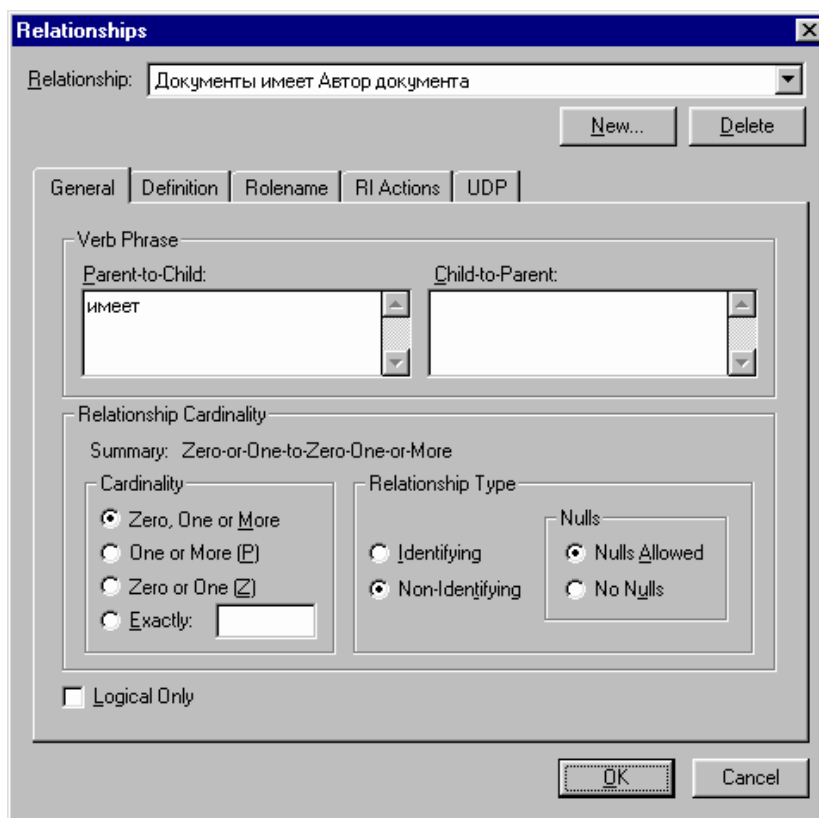
В правой части перечислены элементы управления для настройки свойств одного атрибута, выделенного в левой части. Здесь можно указать тип атрибута, отметить, что атрибут является или не является первичным ключом (галочка «Primary Key»). По умолчанию атрибутам присваивается тип «unknown» - неопределенный.

При добавлении нового атрибута на экране появляется диалог добавления нового атрибута, где предлагается указать его тип данных, логическое и физическое имя.

Работа со связями

Для создания связи необходимо выбрать тип требуемой связи и последовательно щелкнуть на двух связываемых сущностях. При этом для связей типа «один-ко-многим» первая сущность будет родительской, а вторая – дочерней. Для связей типа «многие-ко-многим» порядок сущностей безразличен.

Далее можно задать свойства связи, открыв окно редактора связей. Для этого надо сделать двойной щелчок на линии связи. Появится окно редактора связей.



В этом окне на вкладке «**General**» можно задать общие характеристики связи, изменить ее тип и характеристики. В разделе «**Relationship Cardinality**» можно указать конкретный тип связи, который обеспечивает ERWin:

- ноль, один или много;
- один или много;
- ноль или один;
- точное количество.

Кроме того, можно указать, является связь идентифицирующей или нет, а также допустимость пустых значений.

На вкладке «**Rolename**» можно указать, будет ли переименовываться атрибут, отвечающий за организацию связи, и если будет, то под каким именем он будет присутствовать в дочерней таблице (поле ввода «**Rolename**»).

На вкладке «**RI Actions**» можно указать, какие действия следует выполнить с одной сущностью при различных операциях с другой сущностью. Эти настройки делаются для обеспечения целостности данных при изменении информации в базе данных.

Пример выполнения работы

В данном примере рассматривается задача разработки информационной модели для реализации каталога документов.

Формулировка задачи

Необходимо создать информационную модель библиотечного каталога. Обеспечить хранение сведений об авторах, книгах и издательствах. Обеспечить получение информации из базы данных в следующих видах:

- 1) перечень книг, изданных в указанном издательстве;
- 2) список книг указанного автора;
- 3) перечень издательств с указанием количества книг данного издательства, имеющихся в библиотеке.

Примеры представления информации приведены ниже:

Порядок выполнения работы

Общий порядок разработки информационной модели можно представить следующим образом:

- 1) Выделение сущностей (таблиц);
- 2) Выделение связей между таблицами;
- 3) Формализация модели и устранение избыточности.

Обратите внимание на последний этап. Этот этап во многом повторяет первый и второй этапы и может показаться лишним. На данном этапе в модели выделяются избыточные элементы, и модель модифицируется таким образом, чтобы минимизировать избыточность.

Первичный анализ постановки задачи позволяет выделить следующие сущности: книга, издательство, автор. Эти сущности предварительно имеют следующие атрибуты:

Сущность «Книга»:

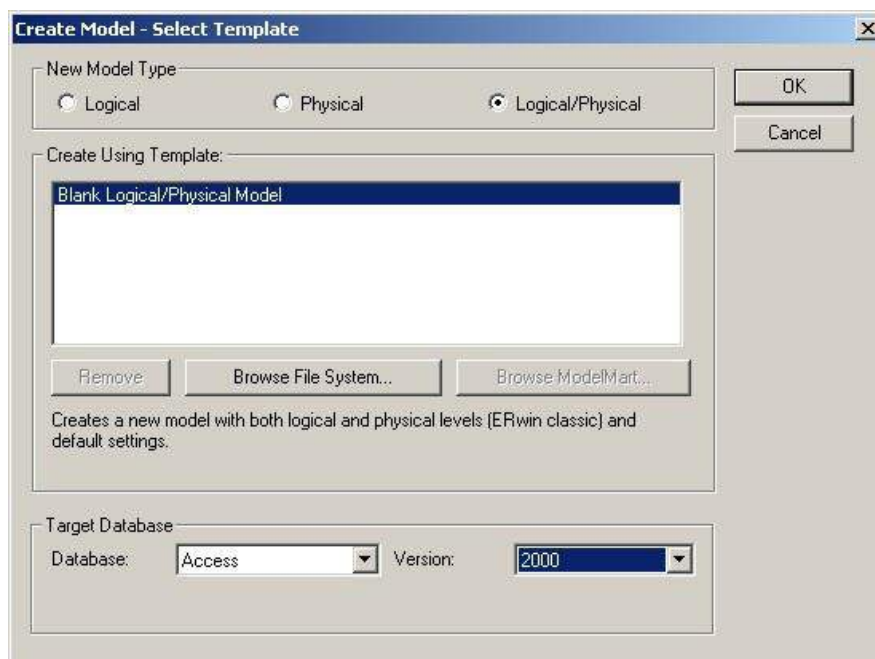
- авторы;
- название;
- издательство;
- год издания;
- жанр книги;
- прочие атрибуты.

Сущность «Автор»:

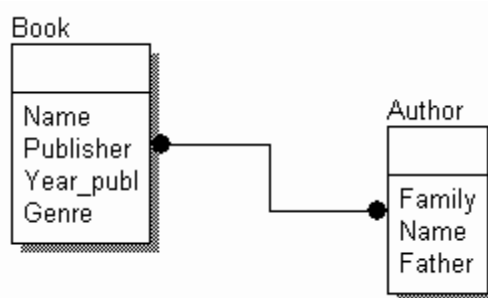
- ФИО автора;

Так как должна быть взаимосвязь между книгами и авторами, то между этими сущностями появляется связь. Она имеет следующие особенности: книга обязательно имеет одного или нескольких авторов, один и тот же автор может написать несколько книг. То есть напрашивается связь типа «многие-ко-многим».

Сформируем в среде пакета ERWin модель в соответствии с полученными результатами. Для этого запускаем ERWin, и создаем новую модель. В качестве типа модели указываем «*Logical/Physical*», сервер – Access 2000.



Далее на логическом уровне создаем нашу модель. Для создания модели воспользуемся панелью инструментов «*Toolbox*». В готовом виде должна получиться примерно такая картинка (используем названия на английском языке):



Проанализируем полученный результат. Можно выявить следующие недостатки:

- 1) Если атрибут «жанр» рассматривать как текст, то для многих книг это поле будет иметь одно и то же значение. Например, все учебники или детективные романы. В то же время самих жанров книг относительно немного – их количество измеряется десятками, в то время как количество самих книг может исчисляться десятками и сотнями тысяч;
- 2) То же самое можно сказать про атрибут «Издательство» – самих издательств гораздо меньше, чем публикуемых ими книг;
- 3) Присутствует связь «многие-ко-многим». Большинство современных СУБД не поддерживают в чистом виде реализацию связи данного типа.

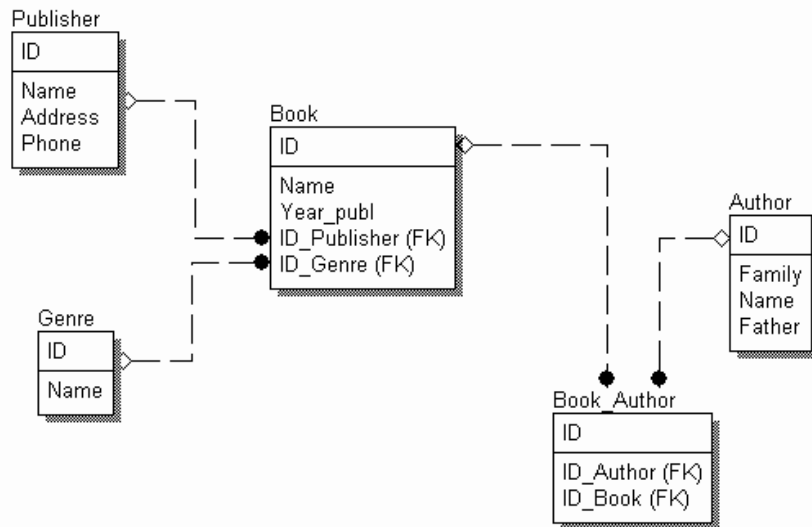
Следовательно, полученная модель требует оптимизации.

Оптимизация осуществляется выделением атрибута «Жанр» в отдельную сущность. При этом возникает связь со следующими требованиями: одна книга может относиться только к одному жанру, в одном жанре могут быть написаны множество книг. Этим требованиям отвечает связь типа «один-ко-многим». Аналогично выполняется выделение сущности «Издательство».

Далее, связь типа «многие-ко-многим» формализуется введением дополнительной сущности «автор книги» и заменой связи «многие-ко-многим» двумя связями «один-ко-многим».

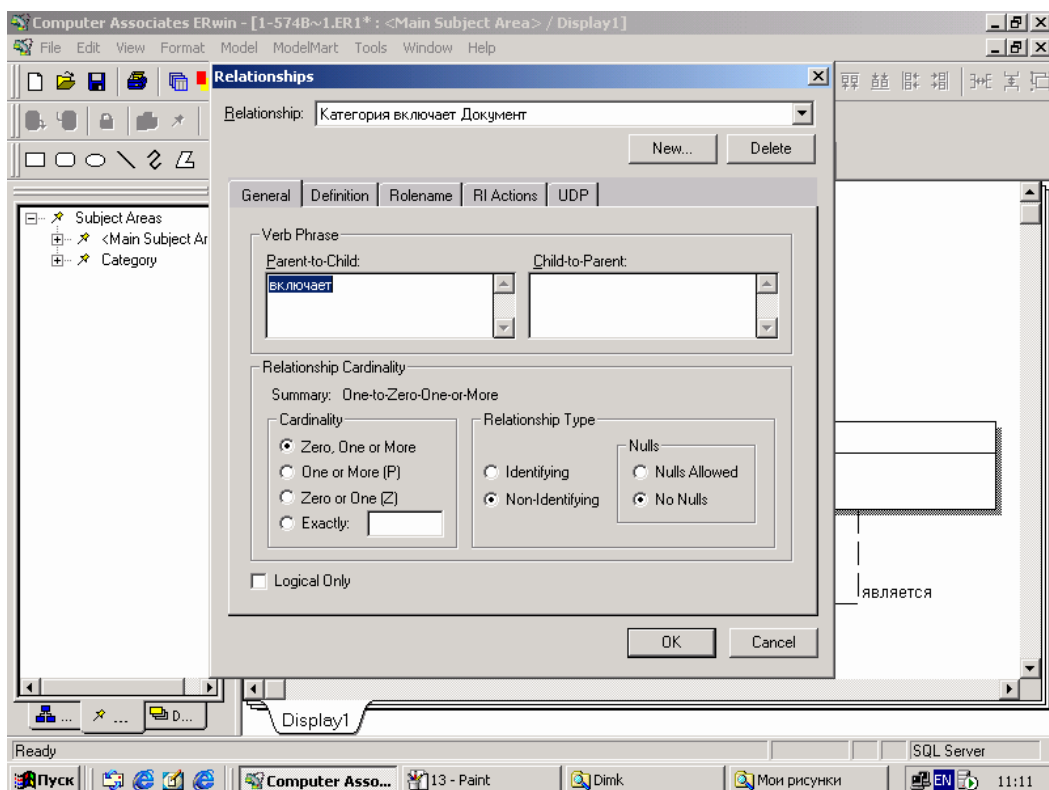
При связывании таблиц удобно задавать специальные поля-идентификаторы (первичные ключи), которые будут использоваться для перекрестных ссылок между таблицами. Такие связи рекомендуется делать неидентифицирующими.

В результате получаем такую модель:



Полученная модель уже близка к корректной модели, которую можно после доработки реализовывать с использованием какой-либо СУБД.

Далее, после формирования сущностей и связей между ними следует настроить все свойства связей. Именно свойствами связей, как правило, задаются ограничения целостности данных для готовой БД. Пользователь в любой момент может изменить тип связи, используемой в модели. Это достигается двойным щелчком на связи, и в появившемся диалоговом окне на вкладке «*General*» выбирается тип связи (область «*Relationship Cardinality*»).



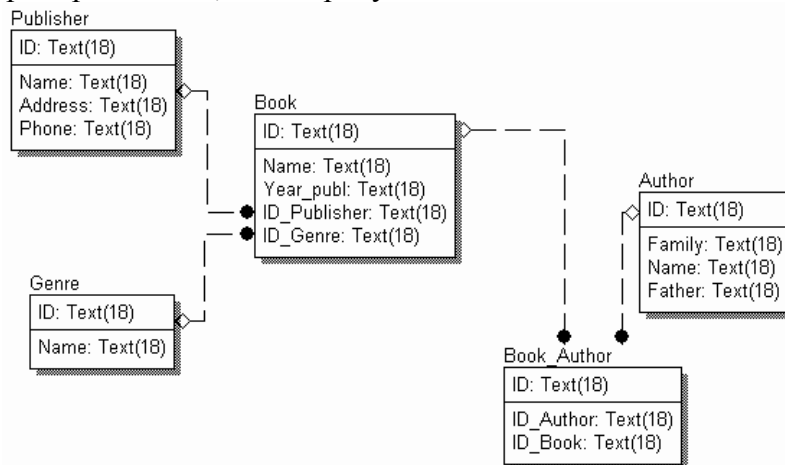
Здесь есть следующие варианты:

- **Zero, One or More** – связь «Ноль, один или много»
- **One or More** – связь «Один или много»
- **Zero or One** – связь «Ноль или один»
- **Exactly** – связь «Точно указанное количество»

Далее в секции «**Relationship Type**» задается характер связи – идентифицирующая или неидентифицирующая. В первом случае первичный ключ родительской таблицы входит составной частью в первичный ключ дочерней таблицы, во втором случае – не входит. Если связь задана неидентифицирующая, можно дополнительно указать допустимость пустых значений (NULL values). Если пустые значения не допускаются, при всех операциях с базой данных в этом поле обязательно должно быть какое-то значение. Значение «NULL» не является нулем, пустой строкой и т.д. Оно говорит о том, что в данном поле вообще отсутствует какое-либо значение.

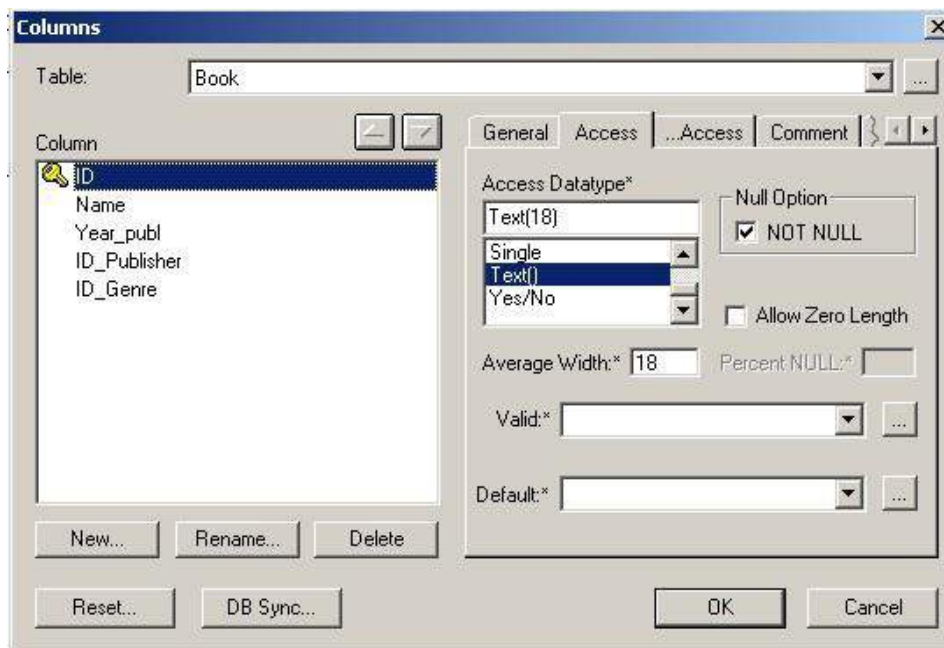
После того, как общий вид связей определен, следует переходить к редактированию физической модели. Для этого в панели инструментов выбирается из выпадающего списка вид модели «**Physical**».

Модель приобретает вид, как на рисунке:

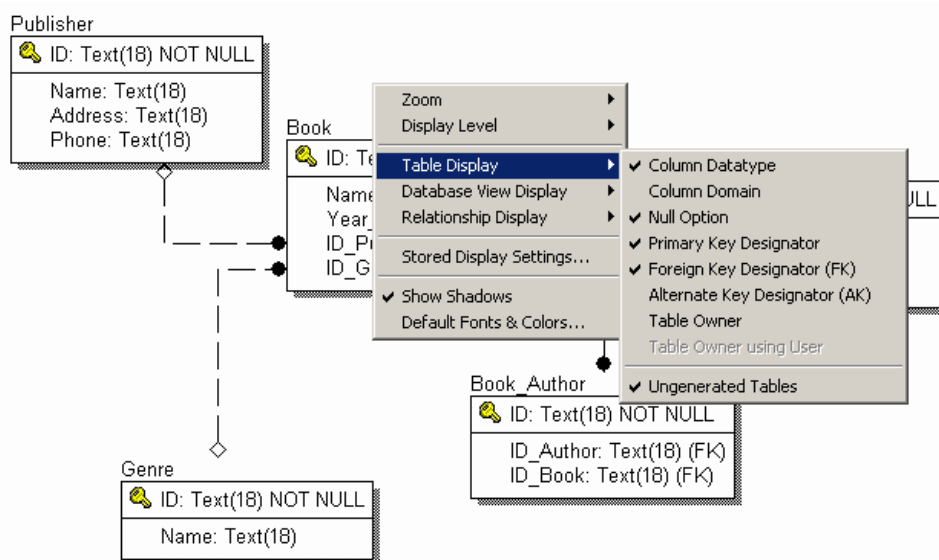


На физическом уровне модель выглядит так, как она должна быть реализована средствами СУБД. При модификации физической модели рекомендуется придерживаться английских наименований атрибутов и сущностей. Хотя СУБД MS Access поддерживает русские наименования атрибутов и сущностей, желательно использовать английские. Это связано с тем, что далеко не все приложения, работающие с разработанной БД, смогут корректно обработать русские наименования полей и таблиц.

Для переименования атрибутов необходимо сделать двойной клик на сущности, и выбирая атрибут из списка, кнопкой «Rename» вызывать диалог переименования атрибута. Кроме того, после переименования следует проверить значения типа данных для каждого атрибута. Это делается на вкладке «**Access**» в правой части окна. Следует обратить внимание, что предлагаемые типы данных зависят от типа данных, установленного для атрибута на вкладке «**General**». Если при создании модели или в меню «**Database – Choose Database**» был указан другой тип сервера, то вкладка с типами данных атрибутов (обычно она идет сразу после вкладки **General**) будет иметь имя выбранного типа базы данных. Например, при выборе типа базы SQL Server, эта вкладка будет называться «**SQL Server**».



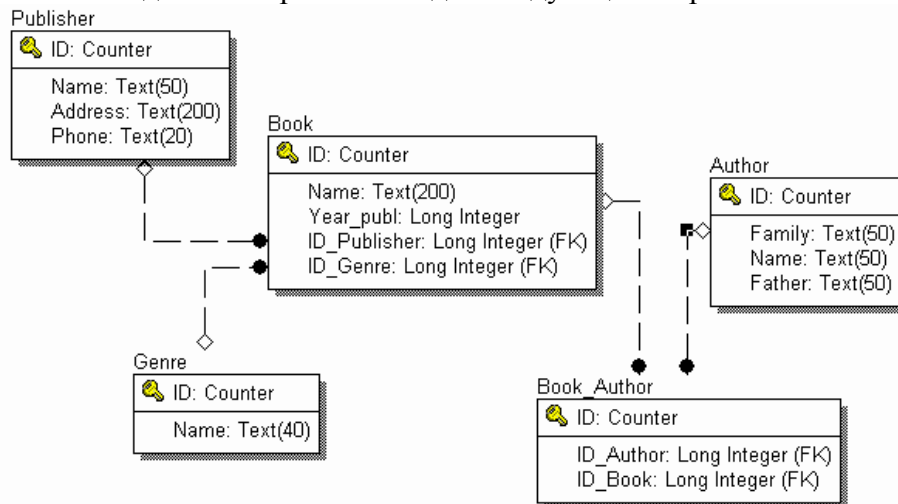
Следующий шаг – проверка типов атрибутов. Следует внимательно проверить все типы данных, используемых для атрибутов, а также допустимость пустых значений. Для облегчения восприятия рекомендуется включить отображение дополнительной информации об атрибутах. Для этого надо щелкнуть правой кнопкой мыши на свободном месте модели (где нет атрибутов и связей), в появившемся меню выбрать пункт «*Table Display*», и далее проставить галочки напротив пунктов «*Column Datatype*», «*Null Option*», «*Primary Key Designator*», «*Foreign Key Designator*».



Для всех первичных ключей следует установить тип данных «*Counter*» (в русской версии Access этот тип называется «Счетчик»). Это означает, что для данного поля будут автоматически генерироваться новые уникальные значения при каждом добавлении новой записи.

Кроме того, следует внимательно проверить допустимость пустых значений для каждого атрибута. Там, где такие значения являются недопустимыми, следует явно указать признак «*NOT NULL*». Пустое значение – это не ноль, не пустая строка или любое другое значение! Значение NULL указывает, что для данного атрибута вообще не задано какое-либо значение.

После этого модель на экране выглядит следующим образом:



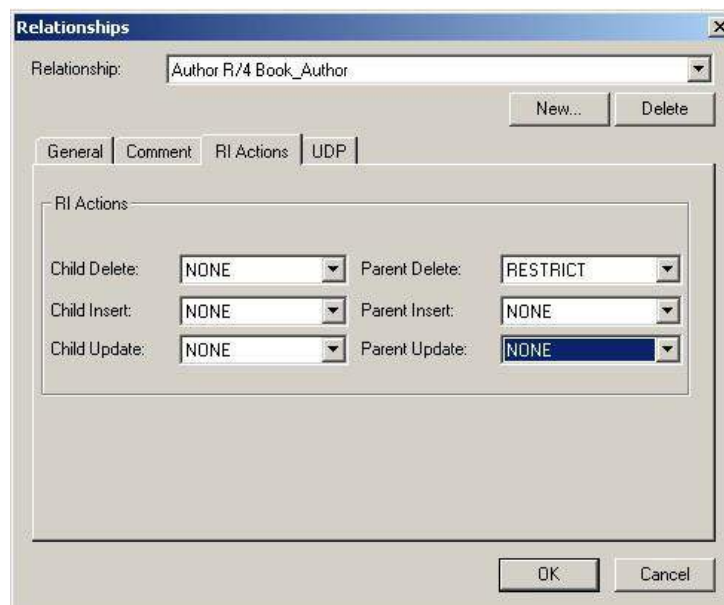
Перед завершением работы над моделью следует проверить действия сервера при различных действиях с БД: добавлении, удалении, изменении записей. Для этого следует открыть окно свойств связи (двойным щелчком на линии связи) и на вкладке «**RI Actions**» установить требуемые ограничения.

Здесь можно установить следующие действия:

- **NONE** – ничего не делать;
- **RESTRICT** – при наличии связанных записей действие запрещено;
- **CASCADE** – действие распространяется на связанные записи;
- **NO ACTION** – аналогично NONE;
- **SET NULL** – установить значение внешнего ключа в NULL.

Например, для связи между таблицами «**Book**» и «**Book_Author**» можно установить в качестве действия на «**Parent Delete**» значение «**CASCADE**». В переводе на человеческий язык это означает, что при удалении книги автоматически будут удалены все связки между этой книгой и ее авторами.

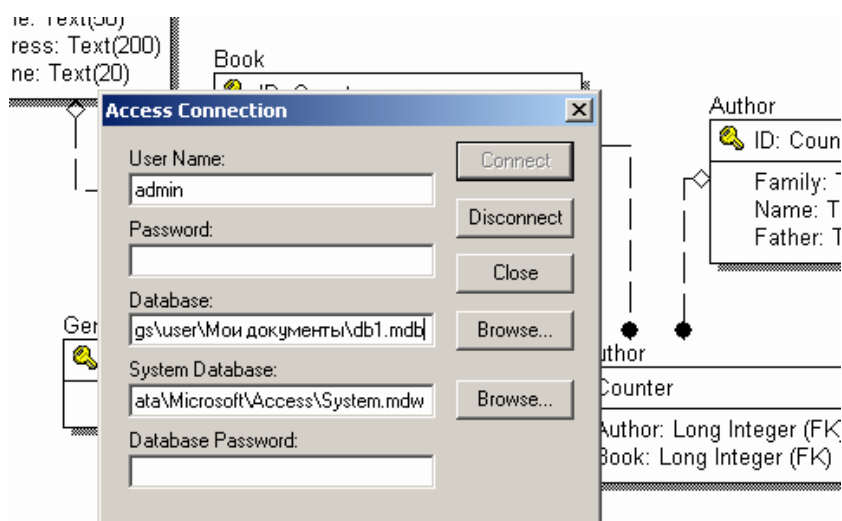
С другой стороны, для аналогичной связки между «**Author**» и «**Book_Author**» для того же действия (**Parent Delete**) следует указать «**RESTRICT**», то есть нельзя удалить автора, пока в библиотеке имеется хотя бы одна книга этого автора.



Подготовленную таким образом модель уже можно транслировать в СУБД Access. Для этого необходимо выполнить следующие шаги: выбрать базу данных, в которой будет создаваться структура данных, настроить опции генерации таблиц.

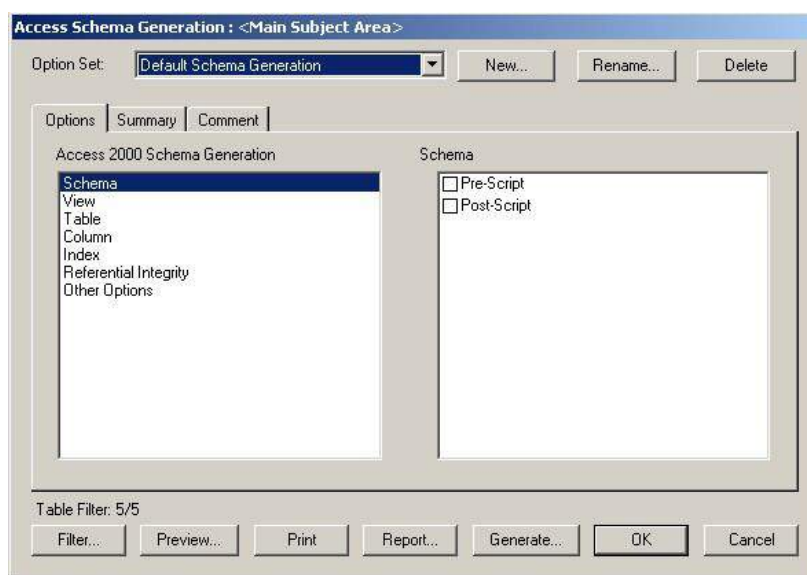
Для выбора базы данных следует в меню «**Database**» выбрать пункт «**Database Connection**». В появившемся диалоге надо указать имя пользователя и пароль для доступа к БД, имя БД пользователя, имя системной БД, и также пароль для доступа к системной БД.

Для работы с Access следует выставить те параметры, которые приведены на рисунке. Расположение системной БД (System Database) следует уточнить у преподавателя.



Предполагается, что тип сервера задан изначально при создании модели. Если это не так, следует выбрать в меню «**Database**» пункт «**Choose Database**». В появившемся диалоге настраивается тип сервера данных.

После настройки соединения с базой данных подготовленную модель данных можно транслировать на сервер БД. Для этого в меню «**Tools**» выбирается пункт «**Forward Engineering/Schema Generation...**».



В этом окне указываются опции генерации структуры данных. Если таблицы уже существуют, в разделе «**Table**» ставится галочка «**Drop Table**». Это означает, что все

существующие таблицы с совпадающими именами будут удалены (естественно, со всеми данными). Поэтому этой опцией следует пользоваться с осторожностью.

После завершения всех настроек после нажатия кнопки «*Generate*» на сервере БД будет создана структура данных по модели. При необходимости предварительного контроля нажатие кнопки «*Preview*» позволяет увидеть SQL-код, который создает все требуемые таблицы.

Задание

Пользуясь пакетом ERWin, по варианту задания сформировать модель данных для реализации БД с указанными требованиями. Структура связей между таблицами определяется правилами: один-ко-многим или многие-ко-многим.

При выполнении работы обеспечить соблюдение следующих требований:

- корректность модели данных;
- полнота информации, представляемой в модели данных. При необходимости студент самостоятельно дополняет модель требуемыми атрибутами и сущностями для адекватного представления рассматриваемой задачи;
- безизбыточность информации, представляемой в модели данных;

После завершения моделирования разработанная структура данных транслируется средствами пакета ERWin в базу данных MS Access.

Варианты заданий

1) Налоговая инспекция

Таблицы:

- налогоплательщик: ФИО, адрес, категория, ...
- налоги: наименование, процентная ставка, дата и размер платежа, ...
- льготы: вид льготы, сумма льготы, процентная ставка, ...

Правила:

- каждый налогоплательщик платит несколько разновидностей налогов
- каждый налогоплательщик может иметь несколько льгот

2) Земельный реестр

Таблицы:

- землевладелец: наименование, адрес, ...
- участок земли: расположение, площадь, категория земельного участка, ...
- строения на участке земли: наименование, площадь, стоимость, ...

Правила:

- один землевладелец может иметь несколько участков, одним участком может распоряжаться несколько землевладельцев;
- на каждом участке может быть несколько строений;

3) Бухгалтерия

Таблицы:

- контрагенты: плательщики и получатели – наименование, адрес, банк, ...
- платежи: дата, сумма, контрагент, ...

Правила:

- каждый контрагент может выполнять несколько платежей
- в одном банке может быть зарегистрировано несколько контрагентов

4) Банк

Таблицы:

- счета: ФИО или наименование клиента, адрес, другие сведения, ...
- операции по счету: зачисление или снятие средств, начисление процентов, ...
- вклады: наименование, срок вклада, порядок начисления процентов, ...

Правила:

- каждый клиент может иметь несколько вкладов

- с каждым счетом выполняется множество операций

5) Инвестиционный фонд

Таблицы:

- объект инвестиций: наименование, срок инвестиций, срок возврата, процент, ...
- фирма-получатель инвестиций: наименование, адрес, ...

Правила:

- фонд одновременно может инвестировать несколько проектов
- фонд получает платежи в несколько приемов
- фонд может осуществлять инвестирование в несколько этапов
- одна фирма может получать инвестиции на несколько различных объектов.

6) Лизинговая компания

Таблицы:

- договор лизинга: дата, номер, контрагент, наименование оборудования, стоимость оборудования, размер выплат, ...
- контрагент, взявший оборудование в лизинг;
- выплата по договору лизинга: дата, сумма, договор, контрагент, ...

Правила:

- один агент может заключить несколько договоров лизинга
- по каждому договору выплаты осуществляются в несколько приемов

7) Отдел кредитования магазина «Электроника»

Таблицы:

- покупатель: ФИО, адрес, место работы, паспортные данные, ...
- кредит: вид кредита, сумма, срок полной выплаты, ...

Правила:

- по кредиту выплаты производятся в несколько приемов
- один покупатель может сделать несколько покупок в кредит

8) Отдел начисления зарплаты

Таблицы:

- сотрудники: ФИО, отдел, должность, размер зарплаты, ...
- выплаты: дата, сумма, за какой месяц, дополнительные выплаты, ...

Правила:

- каждому сотруднику зарплата платится ежемесячно
- каждая зарплата складывается из нескольких выплат

9) Отдел валютных операций банка

Таблицы:

- текущий курс валюты
- кросс-курсы валют
- операции с валютой

Правила:

- каждая валюта имеет свой курс относительно рубля
- пересчет между разными валютами осуществляется с использованием соответствующего кросс-курса

10) Рекламное агентство

Таблицы:

- рекламодатель: наименование, адрес, какие виды рекламы заказывает, ...

- договор рекламы: дата, номер, сумма, правила размещения рекламы, ...
- размещение рекламной информации: вид, дата, параметры, ...

Правила:

- несколько рекламных объявлений по одному договору
- различные виды рекламы (радио, ТВ, баннеры, газетная реклама)