

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	5
Практическая работа 1. ПРОЕКТИРОВАНИЕ ER-МОДЕЛИ БД.....	8
Практическая работа 2. РАЗРАБОТКА РЕЛЯЦИОННОЙ МОДЕЛИ БД.....	26
Практическая работа 3. ПРОЕКТИРОВАНИЕ ПРАВИЛ ЦЕЛОСТНОСТИ БД И ФИЗИЧЕСКОЙ МОДЕЛИ БД.....	33
Практическая работа 4. СОЗДАНИЕ ТАБЛИЦ БД В СУБД ACCESS.....	42
Практическая работа 5. СОЗДАНИЕ ЗАПРОСОВ В СУБД ACCESS.....	51
Практическая работа 6. СОЗДАНИЕ ОТЧЕТОВ В СУБД ACCESS.....	63
Практическая работа 7. СОЗДАНИЕ ИНТЕРФЕЙСА К БД ACCESS	69
Практическая работа 8. ЦЕЛОСТНОСТЬ ДАННЫХ В БАЗЕ ДАННЫХ	79
ЗАКЛЮЧЕНИЕ	84
ЛИТЕРАТУРА	85
Приложение. ВАРИАНТЫ ПРЕДМЕТНОЙ ОБЛАСТИ ДЛЯ ПРАКТИЧЕСКИХ ЗАДАНИЙ	86

ВВЕДЕНИЕ

Широкое применение информационных систем вызывает большой интерес к вопросам проектирования баз данных (БД), которые являются их основой [4, 5, 11]. Качество полученной БД во многом определяется результатами её проектирования. В настоящее время широко используются БД, основанные на реляционной модели данных – реляционные базы данных (РБД). Проектирование модели РБД предполагает определение набора таблиц БД и формирование структуры каждой таблицы. Модели БД должны удовлетворять следующим свойствам [5, 6, 10]:

- представление семантики предметной области, которое предполагает отражение в модели реально существующих в предметной области объектов;
- поддержка эффективного управления данными, связанного с их хранением во внешней памяти и обработкой;
- устранение избыточности данных, которая обусловлена дублированием данных в БД;
- поддержка целостности данных.

Модель БД играет особую роль в процессе проектирования, т.к. представляет информационную модель предметной области [4, 7]. Понятие предметной области, как и многие понятия, используемые в информатике, не имеет точного формального определения. Однако оно широко применяется для того, чтобы ограничить область использования данных из БД. Выделение предметной области БД позволяет сформировать перечень тех сведений об объектах реального мира (сущностей), данные о которых будет содержать БД. Анализ предметной области БД позволяет выделить в ней информационные объекты [7], сведения о которых будут храниться и обрабатываться в БД.

Построение модели БД является непростой задачей, поэтому её решение разбивается на несколько этапов. На каждом этапе разрабатываются модели различного уровня:

- 1) **семантическая модель** предметной области формируется на основе основных понятий, используемых в ней, определяет уровень инфологического моделирования;
- 2) **логическая модель** строится в терминах абстрактных моделей данных и формирует концептуальное представление БД;

3) **физическая модель** определяет, как будут храниться данные во внешней памяти, какие управляющие элементы необходимо предусмотреть для обеспечения эффективной работы с ней.

Такой подход определяет следующие этапы проектирования БД.

1. **Проектирование инфологической модели.** Оно выполняется на основе анализа предметной области. Анализ предметной области позволяет выделить основные информационные объекты, правила их взаимодействия, информационные процессы. Как правило, результатом анализа является формализованное описание процессов предметной области, которое служит основой для построения инфологической модели.

2. **Разработка логической модели** связана, в первую очередь, с выбором типа логической модели.

3. **Проектирование физической модели** позволяет определить способы размещения элементов логической модели во внешней памяти таким образом, чтобы их хранение и управление ими было оптимально.

Для проектирования БД в настоящее время широко применяются специальные программные средства, автоматизирующие разработку моделей различного уровня – CASE-средства (Computer Aided Software Engineering – компьютерные средства разработки программного обеспечения). CASE-средства позволяют применять современные подходы к процессу проектирования программного обеспечения, применять методологию нисходящего проектирования:

- широко использовать автоматизацию для разработки, визуализации и документирования моделей различного уровня;

- создавать библиотеку (репозиторий) моделей, в которой хранятся их описания, на основе репозитория можно создавать новые модели посредством изменения или доработки уже созданных;

- организовывать одновременную работу с моделью БД нескольких разработчиков;

- автоматизировать стандартные процедуры, связанные с документированием, проверкой, интеграцией модели БД.

CASE-средства позволяют решать следующие задачи в процессе проектирования БД.

1. **Разработка моделей процессов обработки информации** в предметной области. Такие модели, как правило, представляются в виде структурных моделей. Они позволяют представить процессы в

виде иерархических диаграмм, задающих функции процесса. Анализ полученных моделей позволяет сформировать состав информационных объектов и определить функции обработки данных. Эта информация является основой для разработки инфологической модели БД.

2. Разработка инфологической модели базируется на использовании модели «сущность-связь». Она позволяет конкретизировать набор характеристик каждого объекта и определить связи между ними.

3. Преобразование модели «сущность-связь» в реляционную модель БД.

4. Автоматическое создание описания БД на языке SQL выбранной СУБД.

5. Автоматическое создание описаний на языке программирования программных компонентов (описание структуры таблиц БД, создание индексов и триггеров БД, экранные формы).

Для анализа информационных процессов могут быть использованы такие программные продукты, как AllFusion Process Modeler, Microsoft Visio, Modelio Open Source.

Для проектирования моделей БД при выполнении практических работ используется CASE-средств AllFusion ERwin Data Modeler или Microsoft Visio. Физическая модель при выполнении практических заданий основывается на модели выбранной СУБД. Для создания БД на основе разработанных моделей при выполнении практических заданий используется СУБД Access.

Практические работы 1–3 позволяют приобрести навыки использования CASE-средств построения моделей БД. В практической работе 1 студенты получают навыки проектирования ER-модели заданной предметной области. В практической работе 2 проводится проектирование реляционной модели БД. В практической работе 3 приобретаются навыки проектирования статических и ссылочных правил целостности для реляционной модели.

Практические работы 4–8 позволяют приобрести навыки использования СУБД Access для реализации БД. В практической работе 4 изучается технология переноса разработанной модели БД в СУБД Access. Практическая работа 5 посвящена приобретению навыков разработки QBE-запросов в СУБД Access. Практическое занятие 6 позволяет приобрести навыки использования инструментальных средств СУБД Access для разработки интерфейса с объектами БД.

Практическая работа 1

ПРОЕКТИРОВАНИЕ ER-МОДЕЛИ БД

Цель работы – приобретение навыков разработки ER-модели заданной предметной области с помощью CASE-средств.

1. Теоретические основы

Для концептуального описания предметной области при проектировании БД используют семантические модели. Они позволяют, с одной стороны, сохранить смысловое описание предметной области, а с другой – представить предметную область в виде формализованной модели. Такие модели используются на этапе инфологического моделирования. В них представляются те смысловые понятия, которые существуют в предметной области и будут отражаться в БД.

Для решения проблемы представления семантики предметной области было создано несколько моделей. Наиболее популярной является модель, предложенная П. Ченом [2, 3], которая известна как модель «сущность-связь» (ER-модель). Её преимуществом является относительная простота, которая, тем не менее, позволяет разрабатывать большие и сложные семантические модели БД. Следует отметить, что существует большое количество CASE-средств, которые включают широкий набор инструментов для работы с такой моделью. При выполнении практической работы используется ER-модель.

В основе построения ER-модели лежит представление предметной области в виде набора взаимосвязанных сущностей. Поэтому основными элементами модели являются сущность (Entity) и связь (Relationship). Каждый реально существующий объект предметной области обладает набором характеристик (свойств), которые необходимо представить в информационной модели. Для представления таких объектов в ER-модели используется сущность. Свойства сущности описывают характеристики реально существующего объекта, а связи определяют смысловые ассоциации, которые можно выделить между сущностями в предметной области.

Сущность может определяться как для реально существующего объекта, так и для некоторого абстрактного понятия предметной области [5, 6, 7, 10, 11]. Конкретные значения характеристик каждого объекта будут храниться в БД. Реальный объект (например,

Преподаватель) задаёт физический объект, в то время как абстрактный объект (например **Лекция**) определяет некоторое абстрактное понятие, связанное с реальным объектом в предметной области. Все объекты предметной области, для которых можно определить общий набор характеристик, необходимо объединить в одну сущность-понятие (далее сущность). Характеристики сущности-понятия называются атрибутами. Они должны определять существенные для заданной предметной области свойства сущности и иметь уникальные (неповторяющиеся) имена. Задавая имена сущностей и атрибутов, следует придерживаться терминологии предметной области. При этом надо учитывать, что наименование характеристик разных сущностей в предметной области могут совпадать (например, свойство **Фамилия** можно определить для различных сущностей: преподаватель, студент, автор учебника, клиент и т.п.). Желательно, однако, определять имена атрибутов разных сущностей уникальными, это позволит легче понимать модель БД.

Экземпляр сущности определяет отдельный объект из этого класса и задаётся конкретными значениями его атрибутов.

При определении набора атрибутов сущности необходимо выделить идентифицирующий атрибут, значение которого позволяет определить конкретную сущность-экземпляр.

В ER-модели можно использовать сущности различных типов:

- сильные и слабые сущности,
- простые и сложные сущности.

Существование сильной сущности не зависит от наличия каких-либо других сущностей. Слабые сущности могут существовать только при наличии связи с некоторой сильной сущностью. Например, слабая сущность **Кредит** зависит от существования сильной сущности **Клиент**.

Простая сущность содержит все свои характеристики. Для сложного объекта предметной области такие характеристики могут быть определены в нескольких сущностях. В этом случае для представления сложного объекта в ER-модели используется сложная сущность.

Вторым элементом в ER-модели является связь. **Связь** в самом общем смысле определяет ассоциацию между сущностями. Связь задаёт смысловое взаимодействие сущностей. В ER-модели следует использовать только бинарные связи, т.е. связи только между двумя

сущностями, которые можно отнести к одному из допустимых типов связей [4, 5, 6].

Связь **ОДИН-К-ОДНОМУ** (1:1) можно задать между сущностями тогда, когда в каждый момент времени один экземпляр родительской сущности (например **Преподаватель**) ассоциируется в предметной области с 1 (или 0) экземпляром дочерней сущности (например **Кафедра**).

Связь **ОДИН-КО-МНОГИМ** (1:M) определяется между сущностями в том случае, когда один экземпляр родительской сущности связан с несколькими экземплярами дочерней сущности, например, между сущностями **Преподаватель** и **Дисциплина**.

Связь **МНОГИЕ-КО-МНОГИМ** (M:N) определяет такое взаимодействие сущностей, при котором некоторое множество экземпляров родительской сущности связано с несколькими экземплярами дочерней сущности. Например, такую связь можно определить между сущностями **Товар** и **Заказ** в том случае, если в один заказ можно включить несколько товаров.

Для однозначной интерпретации ER-модели все её элементы должны иметь описание:

- сущность описывается именем, желательно также определить смысловую интерпретацию сущности, которая определяет её назначение в модели;

- атрибут описывается своим именем, типом допустимых значений, описанием особенностей использования атрибута, также желательно привести краткую смысловую интерпретацию атрибута, которая позволит пояснить его появление в структуре сущности;

- связь описывается именем, которое определяет его смысловую нагрузку, и типом.

Одним из обязательных свойств БД является **целостность** данных. Целостность данных описывается набором правил, которое позволяет учесть специфику работы с данными в предметной области. Это позволяет защитить данные от потери или неправильных действий. В общем случае выделение смысловых правил целостности не является обязательным элементом проектирования ER-модели, но их выделение позволит в дальнейшем построить более качественную БД.

Рассмотрим процесс разработки ER-модели для конкретного примера предметной области: работу курсов дополнительного обучения, в которой организуется работа по проведению групповых

занятий по различным дисциплинам. В этой предметной области можно выделить следующие сущности: **Студент**, **Группа**, **Дисциплина**, **Преподаватель**.

Сущность **Студент** может быть задана следующим набором атрибутов: фамилия студента, номер зачётной книжки, дата рождения. Для конкретного студента эти атрибуты задаются конкретными значениями, например, Фамилия: Смирнов; номер зачётной книжки: 12345; дата рождения 12.10.2000.

Для сущности **Дисциплина** определим следующие атрибуты: название, уровень подготовки, вид итогового документа.

Для сущности **Группа** выделим следующие атрибуты: номер группы, дата формирования.

Задавая связи между сущностями, необходимо основываться на тех правилах, которые существуют в предметной области. Например, если один студент может учиться только в одной группе, то между сущностью **Студент** (родительская сущность) и сущностью **Группа** (дочерняя сущность) определяется связь типа 1:1 (рис. 1.1).

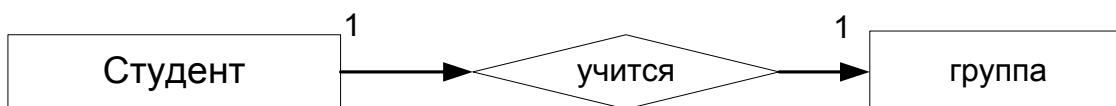


Рисунок 1.1 – Пример связи 1:1

С другой стороны, т.к. одна группа состоит из нескольких студентов, то между родительской сущностью **Группа** и дочерней сущностью **Студент** определена связь 1:M (рис. 1.2).

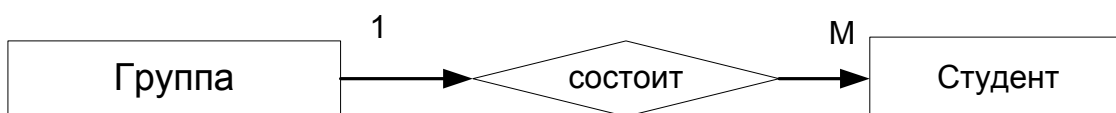


Рисунок 1.2 – Пример связи 1:M

Каждый студент может изучать произвольное количество дисциплин, поэтому между сущностью **Студент** и **Дисциплина** следует определить связь типа M:N (рис. 1.3).

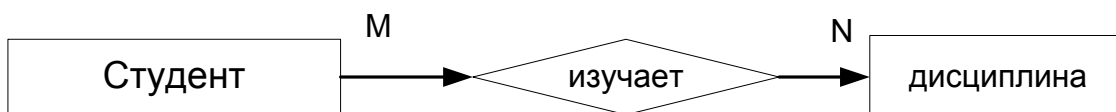


Рисунок 1.3 – Пример связи N:M

2. CASE-средства проектирования инфологической модели

2.1. Проектирование ER-модели в среде AllFusion ERwin Data Modeler

CASE-средство AllFusion ERwin Data Modeler (ERwin DM) широко применяется для проектирования БД [8, 9]. Оно позволяет создать модели различного типа (рис. 1.4): **Logical**, **Physical**, **Logical/Physical**, **Match temptale**.

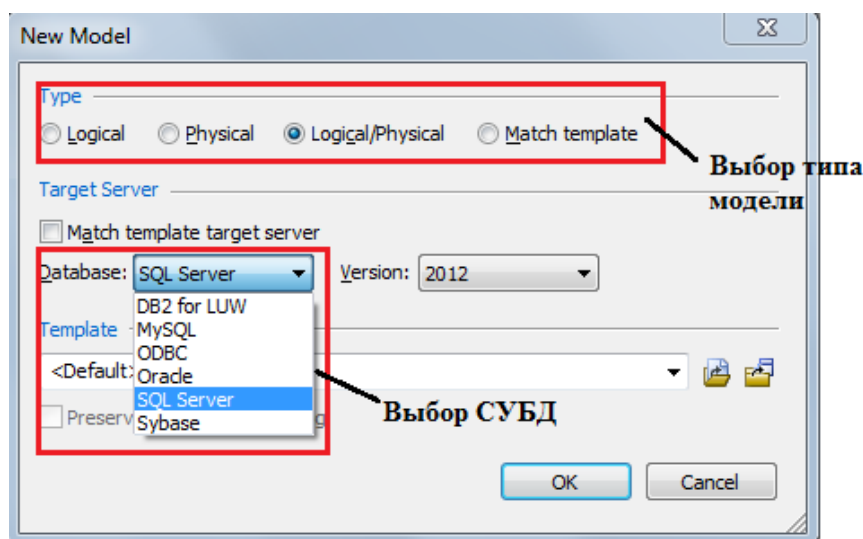


Рисунок 1.4 – Окно выбора типа модели и СУБД

Модель типа **Logical** используется для работы только с ER-моделью предметной области. Модель типа **Physical** используется для работы только с реляционной моделью БД. Модель типа **Logical/Physical** позволяет моделировать БД на инфологическом и логическом уровнях её проектирования и строить ER-модель и реляционную модель БД. Выбор типа модели осуществляется в начале создания модели. В рамках выполнения практических работ разрабатываются модели БД, имеющие тип **Logical/Physical**.

Если выбрана модель типа **Logical/Physical** или **Physical**, то можно выбрать тип целевой СУБД (рис.1.4).

Основное меню среды включает (рис. 1.5) возможности разработки логической и физической модели, документирование модели, выполнение процессов прямого и обратного проектирования. Это CASE-средство позволяет создавать графические модели БД и документировать их. **ERwin DM** использует удобный графический интерфейс, который позволяет разрабатывать ER-модели любой сложности. В среде **ERwin DM** ER-модель строится в терминах

логической модели, элементами которой являются сущности, атрибуты и связи.

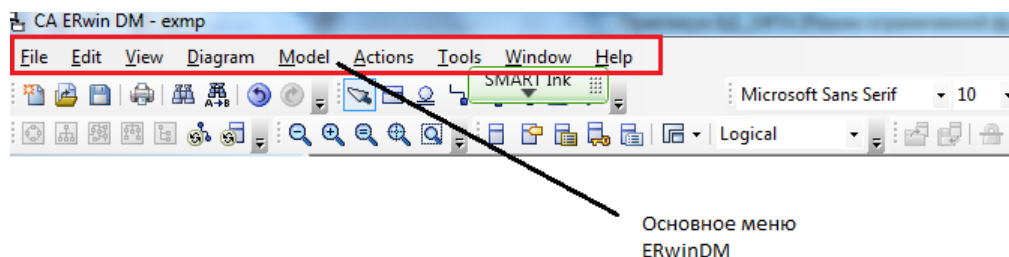
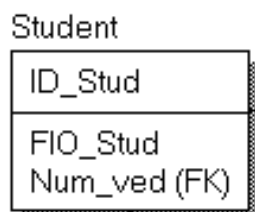


Рисунок 1.5 – Главное меню CASE-средства AllFusion ERwin Data Modeler

В **ERwin DM** прямоугольник (рис. 1.6.) обозначает сущность. Имя сущности располагается над прямоугольником. Для описания сущности используется контекстное меню, которое вызывается правой кнопкой мыши. Для того чтобы задать описание сущности, необходимо выбрать пункт **Properties** и задать уникальное имя, её смысловое описание и особенности взаимодействия с другими сущностями. От того, насколько полно будет представлена эта информация, в дальнейшем зависит качество модели БД.

Логическая модель в **Erwin** может быть представлена различными способами: на уровне сущностей, на уровне атрибутов, на уровне ключей. Наиболее полное представление модели задаётся уровнем атрибутов.

Рисунок 1.6 – Пример сущности в Erwin DM







Для задания описателей атрибутов сущности в контекстном меню выбирается пункт **Attributes**, меню которого позволяет вызвать **Редактор свойств**. В окне **Редактора свойства** задаются основные описатели атрибута: имя атрибута, тип принимаемого значения. Полезно также задать краткое описание атрибута, смысловые правила использования значений атрибута. Это позволит, во-первых, правильно интерпретировать назначение атрибута в модели, а во-вторых, позволит создать описание модели средствами **Erwin DM**. Имена атрибутов отображаются внутри прямоугольника. В верхней части прямоугольника, задающего сущность, отражаются идентифицирующие атрибуты, в нижней – остальные атрибуты.

Обязательным описателем каждого атрибута является тип значений, который должен быть выбран в среде **ERwin DM** из набора допустимых типов:

- данные целого типа – Integer;
- вещественные данные – Real;
- данные, задающие дату и время, – Date;
- символьные данные – Varchar;
- числовые данные в денежном формате – Money.

Между сущностями в **ERwin DM** допускаются следующие типы связей [8]:

-  – задаёт связь «многие-ко-многим»;
-  – задаёт идентифицирующую связь «один-ко-многим»;
-  – задает неидентифицирующую связь «один-ко-многим»;
-  – определяет категориальную связь, которая позволяет определять в модели сложные сущности, реализуя отношения типа тип-подтип.

Для выбора типа связи между сущностями необходимо выделить соответствующую связь (рис. 1.7) и задать её для двух выделенных сущностей.

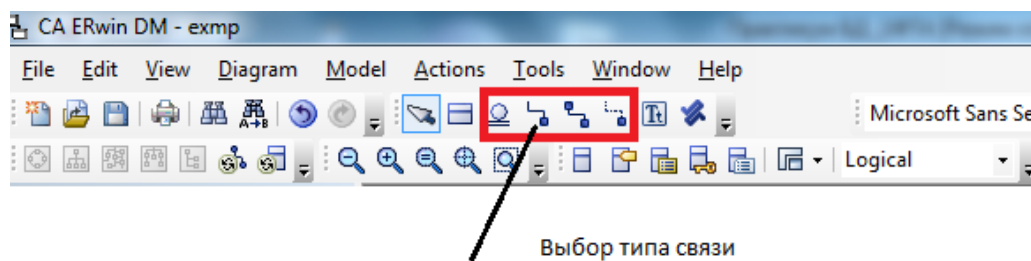


Рисунок 1.7 – Типы связи между сущностями

Для определения связи 1:N в **ERwin DM** используется два типа связей: идентифицирующая связь «один-ко-многим» и неидентифицирующая связь «один-ко-многим». Идентифицирующая связь связывает сильную (родительскую) сущность и слабую сущность (дочернюю). При использовании такой связи дочерняя сущность автоматически преобразуется в зависимую сущность. На графической модели она представляется прямоугольником с закругленными углами. Поскольку экземпляр зависимой сущности не может существовать без существования связанного с ним экземпляра родительской сущности, то в этом случае в среде **ERwin DM** [8, 9] идентифицирующие атрибуты родительской сущности автоматически

копируются в идентифицирующие атрибуты дочерней сущности (миграция атрибутов). В дочерней сущности скопированные атрибуты помечаются символами FK (внешний ключ). На рисунке 1.8. приведён пример миграции ключей при определении неидентифицирующей связи между сущностями **Преподаватель** и **Курс**.

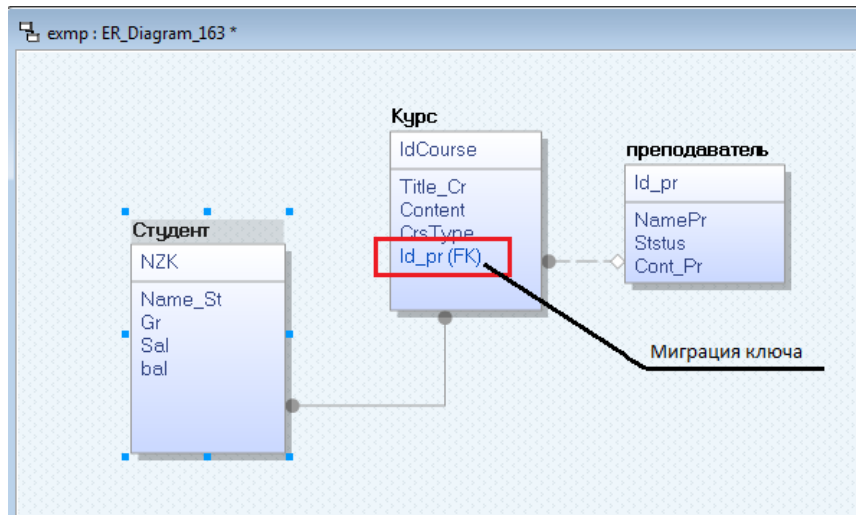


Рисунок 1.8– Задание связи между сущностями

Чтобы описать связь, необходимо в контекстном меню, вызываемом правой кнопкой мыши, выбрать пункт **Properties** (свойства), который вызывает редактор связей, и определить имя связи, её тип и мощность (рис. 1.9).

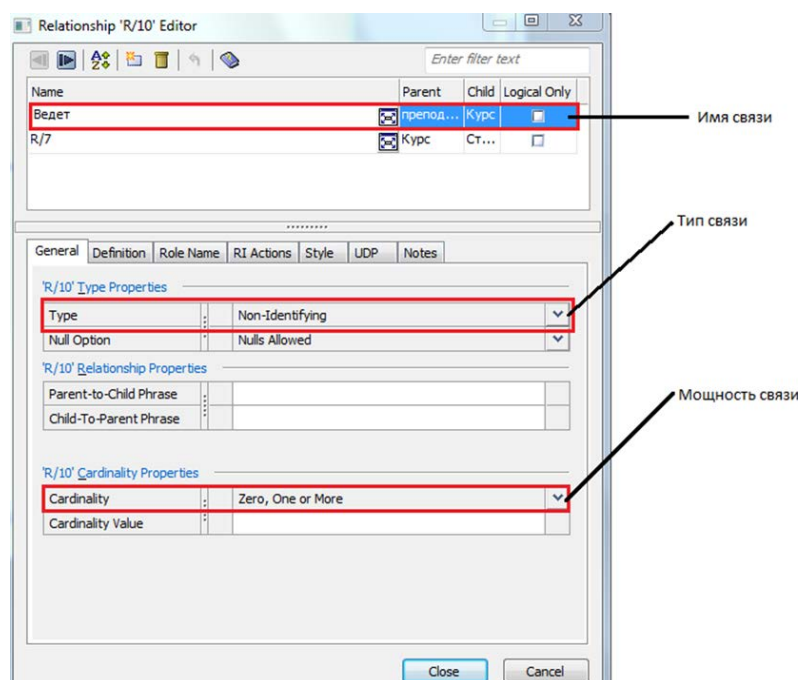


Рисунок 1.9 – Задание свойств связи

Мощность выбирается из определённых в среде **ERwin DM** типов (рис. 1.10):

– связи типа **Zero, One or More** и **One or More (P)** используются для обозначения связи типа 1:M (отличие между этими двумя типами в **ERwin DM** заключается в том, что в случае использования типа **Zero, One or More** в дочерней сущности могут отсутствовать связанные с родительской сущностью экземпляры, а при использовании связи типа **One or More (P)** в дочерней сущности обязательно должны быть экземпляры, связанные с родительской сущностью, такая связь на графической модели помечается символом **P**);

– **Zero or One (Z)** определяет связь 1:1;

– связь, помечаемая на модели цифрой, позволяет определить точное количество экземпляров дочерней сущности, которое связано с одним экземпляром родительской сущности.

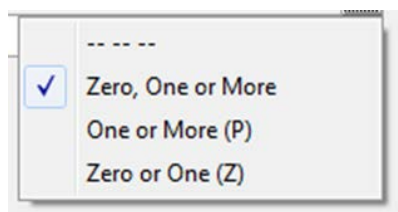


Рисунок 1.10 – Типы мощности связи

В общем случае выделяют следующие виды слабых сущностей:

– **характеристическая** – дополняет характеристики родительской сущности и связана только с одной сущностью;

– **ассоциативная** – используется для задания взаимодействия двух сущностей и связана с каждой из них;

– **категориальная** – используется для определения сложной сущности на основе нескольких слабых сущностей.

Пример слабой сущности **Контракт** представлен на рисунке 1.11.

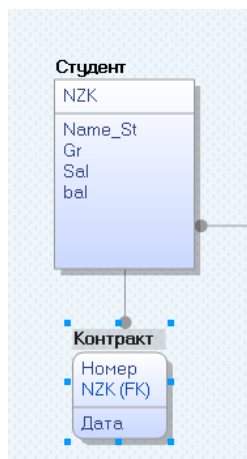


Рисунок 1.11 – Зависимая сущность

Задание идентифицирующей связи в дочерней сущности **Контракт** автоматически создаётся копия идентифицирующего атрибута родительской сущности **Студент – NZK (FK)**, которые помечаются символами (FK).

2.2. Проектирование ER-модели в среде Microsoft Visio

Microsoft Visio нельзя в полной мере отнести к CASE-средствам проектирования БД. Это среда разработки графических объектов. Однако благодаря большому количеству шаблонов, определённых в ней, её можно использовать для разработки модели БД.

Для разработки ER-модели при выполнении практического занятия рекомендуется использовать шаблон для создания БД в нотации **IDEF1X** (рис. 1.12), который находится в категории «Программное обеспечение».

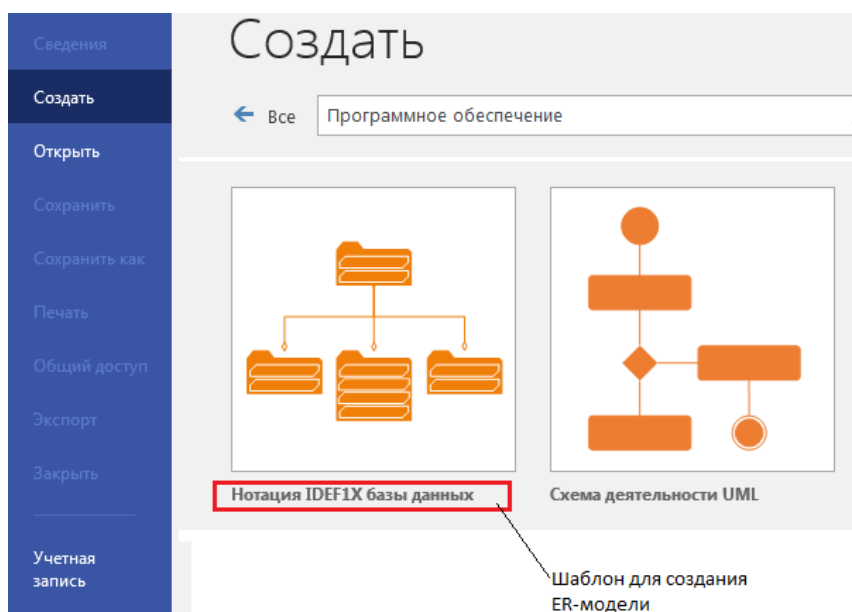


Рисунок 1.12 – Выбор шаблона для разработки ER-модели в среде Microsoft Visio

Основное меню **Microsoft Visio** (рис. 1.13) позволяет выполнять следующие действия по работе с графическим объектом:

- выполнять стандартные действия с файлом – открытие файла, создание нового файла, сохранение – вкладка **Файл**;
- создавать и редактировать графический объект – вкладка **Главная**;
- вставлять различные элементы графического объекта: иллюстрации, части схемы, ссылки, текст – вкладка **Вставка**;
- изменять параметры страницы – вкладка **Конструктор**;

- определять связь с внешними данными – вкладка **Данные**;
- создавать новый процесс – вкладка **Процесс**;
- проверять правописание, добавлять примечания, отмечать исправления и др. – вкладка **Рецензирование**;
- определять вид отображения графического объекта на экране дисплея – вкладка **Вид**.

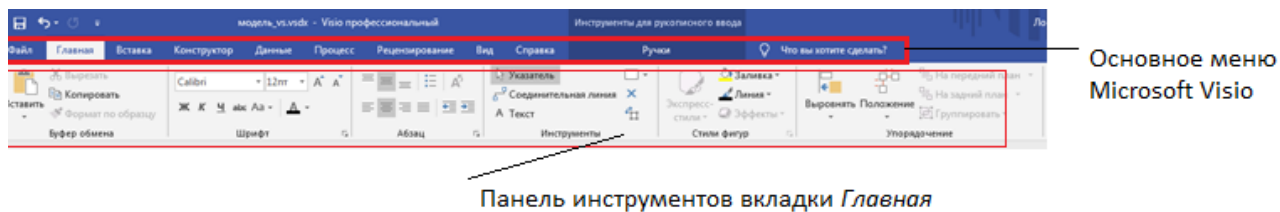
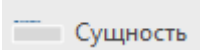
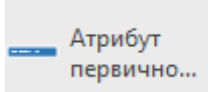
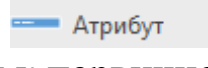
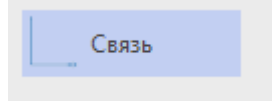



Рисунок 1.13 – Основное меню Microsoft Visio

При использовании шаблона **Нотация IDEF1X** для базы данных **Microsoft Visio** предоставляет следующий набор элементов (рис. 1.14):

- сущность  **Сущность** используется для определения имени сущности и включает описание атрибутов;
- атрибут первичного ключа  позволяет определить атрибут, идентифицирующий экземпляр сущности (атрибутов первичного ключа может быть несколько, в этом случае он является составным);
- атрибут  позволяет определить атрибут сущности, не относящейся к первичному ключу;
- связь  определяет все виды связей, предусмотренные в ER-модели (рекомендуется обозначать тип связи, используя инструмент добавления текста – инструмент  панели инструментов (рис. 1.13)).

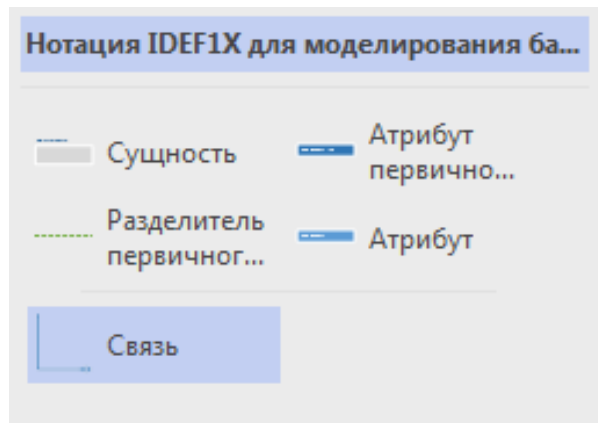


Рисунок 1.14 – Элементы для создания ER-модели в Microsoft Visio

Пример построения модели БД в среде Microsoft Visio представлен на рисунке 1.15.

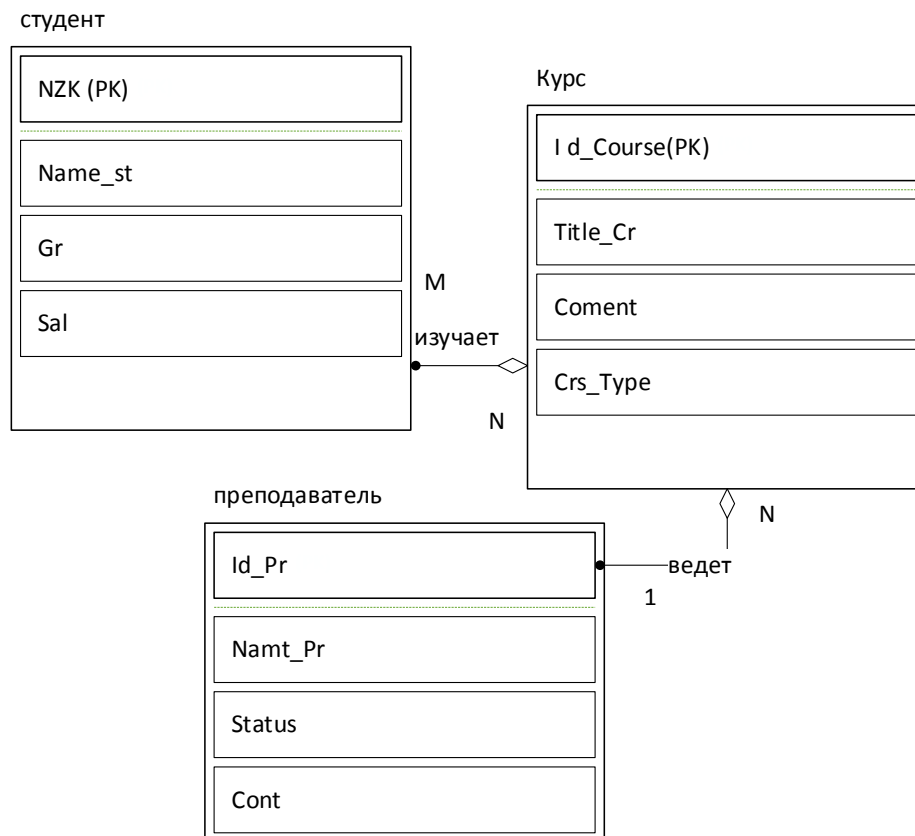


Рисунок 1.15 – ER-модель в среде Microsoft Visio

3. Средства документирования ER-модели

Документирование модели является важным и обязательным элементом разработки модели. Различные CASE-средства имеют разные инструменты для документирования.

3.1. Документирование модели в ERwin Data Modeler

CASE-средство **ERwin DM** обладает специальными возможностями для автоматического создания описания разработанной модели. В его состав включены возможности использования стандартных видов отчёта по модели и средства разработки отчёта на основе созданного шаблона. Эти возможности определены во вкладке **Tools** основного меню (рис. 1.5.). Набор стандартных отчётов определяется имеющимися шаблонами. Для создания собственного шаблона отчёта по модели используется редактор отчётов – **Report Designer** (рис. 1.16).

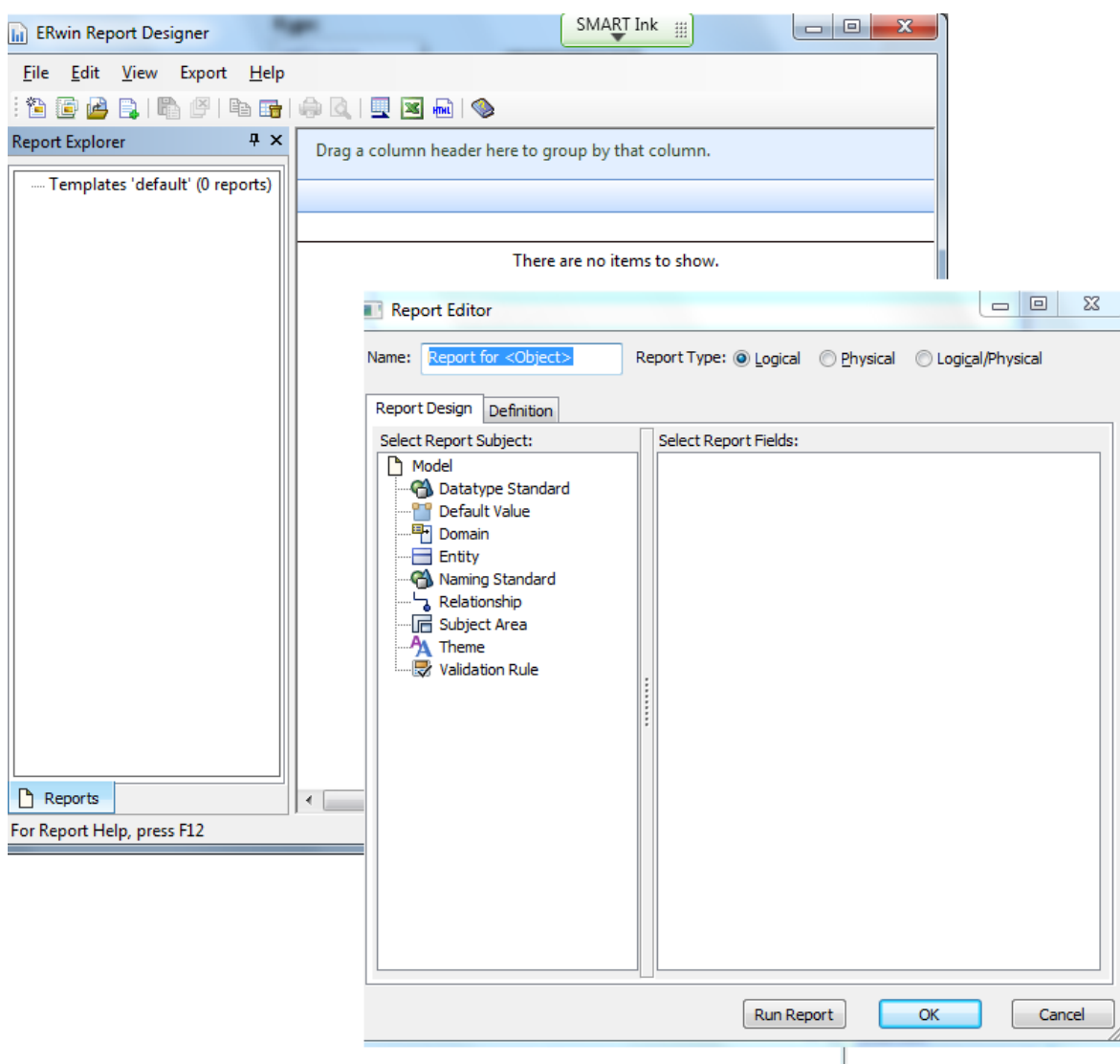


Рисунок 1.16 – Создание шаблона отчета в редакторе отчетов

Для формирования своего шаблона отчёта необходимо выделить в правой части **Редактора отчётов** элемент модели. В левой части **Редактора отчётов** будут отображаться свойства выбранного

элемента. Для включения выбранного свойства в шаблон отчёта достаточно поставить знак у этого элемента (рис. 1.17).

Для формирования отчёта достаточно нажать кнопку **Run Report**. Созданный отчёт (рис. 1.18) можно сохранить в различных форматах. Примеры сформированных отчётов по модели представлены на рисунках 1.18 и 1.19.

При использовании среды **Microsoft Visio** отчёт создаётся отдельно средствами **Microsoft Word**. Для этого можно использовать табличную форму описания модели, аналогичную представленной на рисунке 1.18.

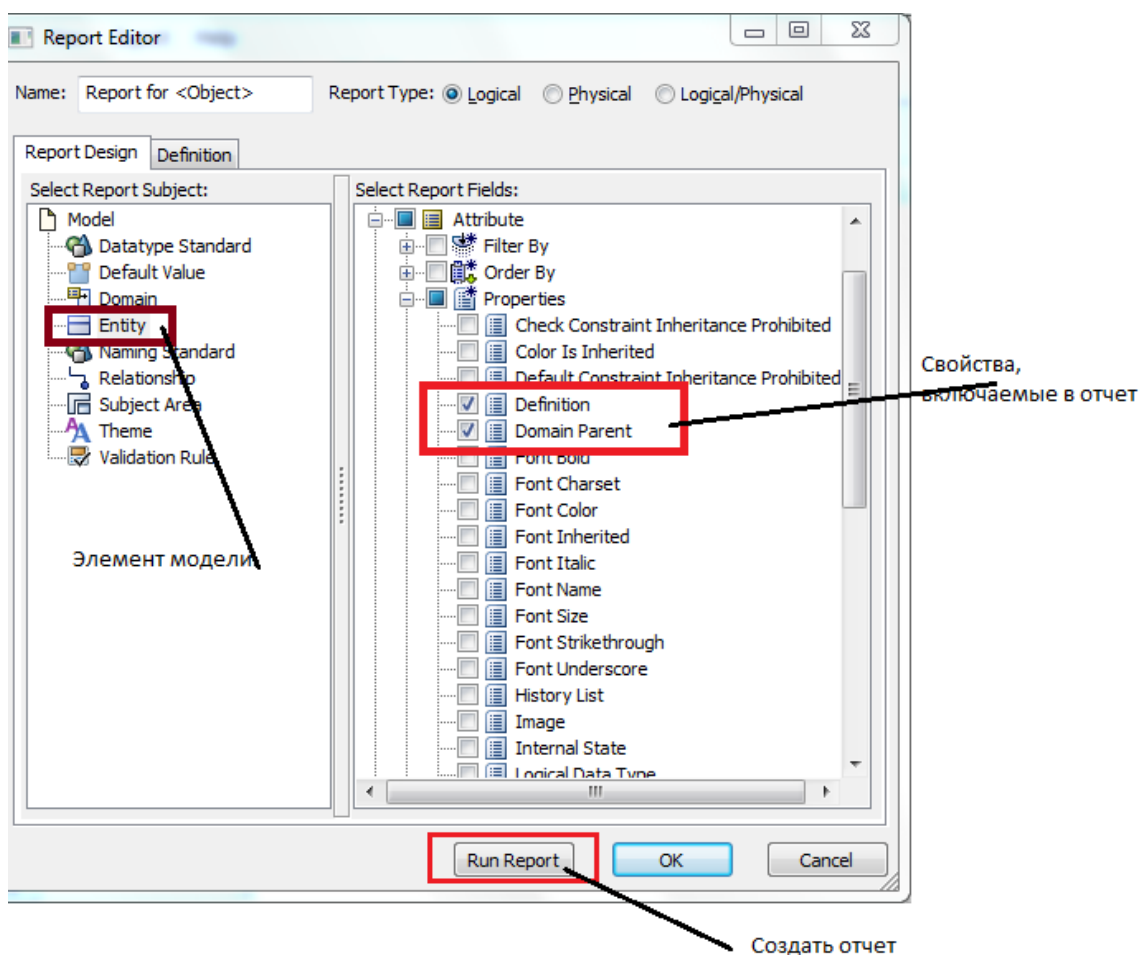


Рисунок 1.17 – Формирование шаблона отчета

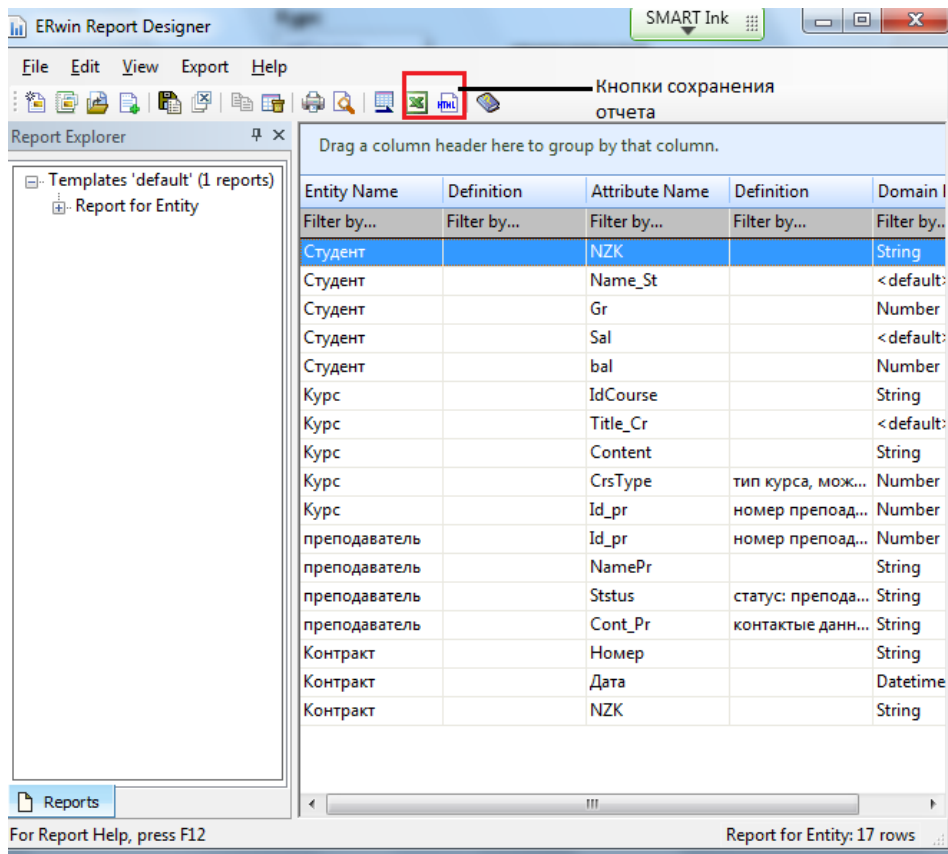


Рисунок 1.18 – Отчёт по модели, созданный в среде Erwin DM

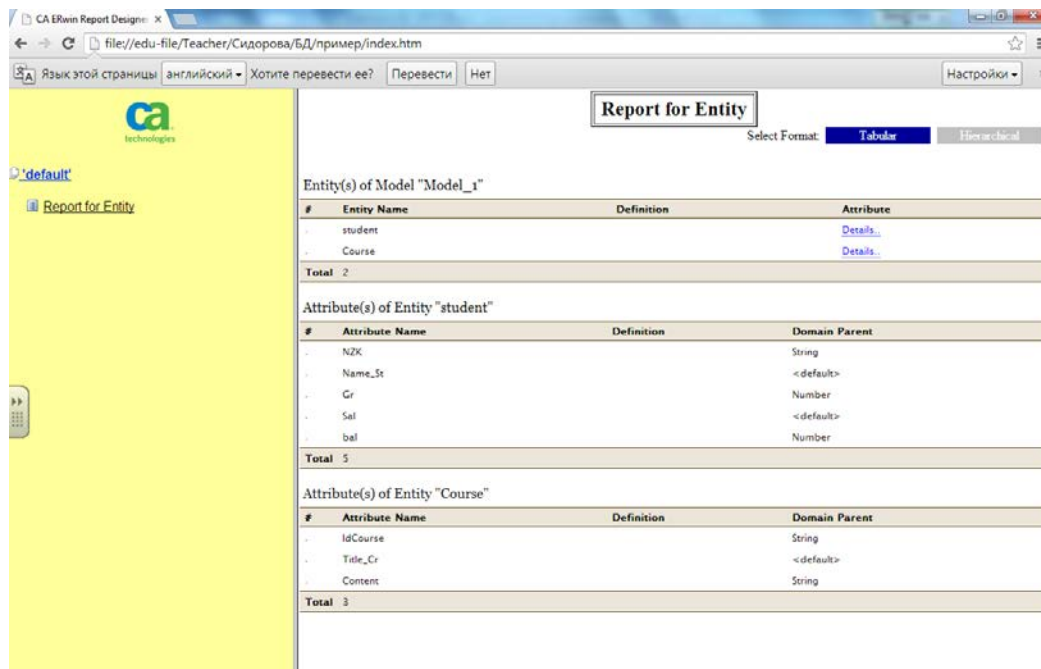


Рисунок 1.19 – Отчёт по модели в формате html

3.2. Документирование модели в среде Microsoft Visio

Среда Microsoft Visio не располагает средствами документирования модели. Поэтому для описания созданной модели

рекомендуется использовать средства разработки текстовых документов, например, **Word**. Описания ER-модели должно включать следующие разделы:

- состав сущностей и их краткая характеристика;
- состав атрибутов каждой сущности и их краткое описание;
- описание связей между сущностями.

Такое описание может быть выполнено в форме таблиц. Фрагмент описания ER-модели, представленной на рисунке 1.15, приведён в таблицах 1.1 – 1.3.

Таблица 1.1 – Описание сущностей ER-модели

№	Имя сущности	Описание
1.	студент	Учащийся вуза, который может обучаться по одному направлению подготовки
2.	дисциплина	Дисциплина учебного плана направления подготовки
3.	преподаватель	Сотрудник вуза, который ведёт занятия по дисциплинам учебного плана

Таблица 1.2 – Описание свойств сущностей

№	Имя свойства	Описание	Ограничения
Сущность студент			
1.	NZK	Номер зачётной книжки	Уникальное значение для каждого студента
2.	Name_st	Полное имя студента, содержит фамилию имя и отчество	
3.	Gr	Номер группы, в которой обучается студент	
4.	Sal	Стипендия	Размер стипендии ≥ 0
Сущность дисциплина			
1.	Id_Course	Код дисциплины	Задаётся номером дисциплины по учебному плану, уникальное значение для каждой дисциплины
2.	Titile_Cr	Название дисциплины по учебному плану	
3.	Coment	Краткое описание дисциплины	
4.	Crs_Type	Тип дисциплины	Может принимать значения: 1 – 4

Таблица 1.3 – Описание связей ER-модели

№	Имя связи	Тип связи	Описание
1.	изучает	многие-ко-многим	Много студентов изучает несколько дисциплин
2.	ведёт	один-ко-многим	Один преподаватель ведёт несколько дисциплин

4. Последовательность выполнения практической работы

1. Запустить CASE-средство разработки модели БД.

2. Ознакомиться с её интерфейсом среды моделирования, её элементами для построения ER-модели.

3. Провести анализ описания предметной области и выделить набор сущностей.

Указание. *Варианты заданий для выбора предметной области приведены в Приложении I. Студент может конкретизировать описание предметной области, приведённое в варианте задания.*

Набор сущностей должен обязательно включать указанные в варианте задания сущности, но также может включать дополнительные сущности на основе уточнённого описания предметной области.

При определении состава свойств (атрибутов) сущности следует учитывать указанный в задании минимальный набор атрибутов и запросы на обработку.

4. Используя CASE-средство, построить и описать ER-модель для заданной предметной области. Описание каждой сущности должно содержать её имя, назначение в модели БД.

Выделить атрибуты каждой сущности.

Указание. *При определении состава свойств (атрибутов) сущности следует учитывать указанный в задании минимальный набор атрибутов и запросы на обработку.*

5. Определить для каждой сущности набор атрибутов и задать их описание. Описание каждого атрибута должно содержать его имя, тип, назначение атрибута в модели, смысловые правила, определяющие специфику использования атрибута.

6. Задать идентифицирующий атрибут для каждой сущности.

7. При необходимости выделить сложные сущности.

8. Провести анализа бизнес-правил, приведённых в варианте задания, и определить связи между сущностями. Описать каждую

Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

e-Univers.ru