

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Пермская государственная сельскохозяйственная академия
имени академика Д.Н. Прянишникова»

ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ
направление 230700 «Прикладная информатика»

ЛАБОРАТОРНОЕ ЗАНЯТИЕ № 5

Тема: ЛОГИЧЕСКАЯ МОДЕЛЬ ДАННЫХ В МЕТОДОЛОГИИ IDEF1X

Учебные вопросы:

1. Уровни логической модели данных.
2. Состав модели данных.
3. Нормализация отношений.
4. Построение логической модели данных.

Литература, техническое и программное обеспечение:

1. Методическая разработка по теме занятия.
2. Класс ПЭВМ.
3. AllFusion Erwin Data Modeler 7.

Вопрос 1. Уровни логической модели данных

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

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

Диаграмма сущность-связь – это модель данных верхнего уровня. Она включает сущности и взаимосвязи, отражающие основные бизнес-правила предметной области. Такая диаграмма не слишком детализирована, в нее включаются основные сущности и связи между ними, которые удовлетворяют основным требованиям, предъявляемым к ИС. Диаграмма сущность-связь может включать связи «многие-ко-многим» и не включать описание ключей. Как правило, ERD используется для презентаций и обсуждения структуры данных с экспертами предметной области. Целью этой диаграммы является формирование общего взгляда на систему для ее дальнейшей детализаций.

Модель данных, основанная на ключах, – более подробное представление данных. Она включает описание всех сущностей и первичных ключей и предназначена для представления структуры данных и ключей, которые соответствуют предметной области. Целью этой модели является детализация модели сущность-связь, после чего модель данных может начать реализовываться.

Полная атрибутивная модель – наиболее детальное представление структуры данных: представляет данные в третьей нормальной форме и включает все сущности, атрибуты и связи.

Поскольку основными компонентами диаграммы в ERwin являются сущности, атрибуты и связи, то построение модели данных предполагает определение сущностей и атрибутов, т. е. необходимо определить, какая информация будет храниться в конкретной сущности или атрибуте, а также связей между сущностями.

Вопрос 2. Состав модели данных

Сущности

Сущность – это множество реальных или абстрактных предметов (людей, объектов, мест, событий, состояний и т.д.), обладающих общими атрибутами и характеристиками.

Отдельный элемент этого множества называется **экземпляром сущности**. Каждый экземпляр индивидуален и должен отличаться от всех остальных экземпляров.

Правила сущностей

1. Каждая сущность должна иметь уникальное имя в единственном числе; к одному и тому же имени должна применяться одна и та же интерпретация.
2. Сущность обладает одним или несколькими атрибутами, которые либо принадлежат сущности, либо наследуются через отношение.
3. Сущность обладает одним или несколькими атрибутами, которые однозначно идентифицируют каждый экземпляр сущности.
4. Каждая сущность может обладать любым количеством отношений с другими сущностями модели.

5. Отношение связи – это связь между сущностями, при которой каждый экземпляр одной сущности, называемой родительской сущностью, ассоциирован с произвольным количеством экземпляром второй сущности, называемой сущностью-потомком, а каждый экземпляр сущности-потомка ассоциирован в точности с одним экземпляром сущности-родителя.

В IDEF1X различают **зависимые** и **независимые сущности**. Тип сущности определяется ее связью с другими сущностями.

Независимая от идентификаторов или просто независимая сущность – это сущность, у которой каждый экземпляр сущности может быть однозначно идентифицирован без определения его отношений с другими сущностями.

Зависимая от идентификаторов или просто зависимая сущность – это сущность, у которой однозначная идентификация экземпляра сущности зависит от его отношения к другой сущности.

Если сущности были описаны при моделировании в VPwin, то их можно просто импортировать в ERwin.

Атрибуты

Атрибут – это описательное свойство или характеристика сущности.

С точки зрения БД (физическая модель) сущности соответствует таблица, экземпляру сущности – строка в таблице, а атрибуту – колонка таблицы.

Правила атрибутов

- Каждый атрибут должен иметь уникальное имя.
- Сущность может обладать любым количеством атрибутов.
- Сущность может обладать любым количеством наследуемых атрибутов.
- Каждый экземпляр сущности должен иметь значение для каждого атрибута (правило необращения в ноль).
- Ни один из экземпляров сущности не может обладать более чем одним значением для связанного с сущностью атрибута (правило неповторяемости).

При задании типа атрибута имеется возможность использовать домены.

Домен – это абстрактный пользовательский тип, который присваивается любому физическому типу данных. При этом каждый домен может иметь свои значения по умолчанию и правила проверки вводимых данных.

ERwin предоставляет возможность документировать все действия по созданию собственных типов данных. С использованием концепции домена обеспечивается переносимость БД на различные аппаратные платформы.

Связи

Связь – это логическое соотношение между сущностями. Каждая связь должна именоваться глаголом или глагольной фразой. Имя связи выражает некоторое ограничение или бизнес-правило и облегчает чтение диаграммы. По умолчанию имя связи на диаграмме не показывается.

Связь может дополнительно определяться с помощью указания степени или мощности.

Мощность – это количества экземпляров сущности-потомка, которое может существовать для каждого экземпляра сущности-родителя.

В IDEF1X могут быть выражены следующие **мощности связей**:

- каждый экземпляр сущности-родителя может иметь ноль, один или более связанных с ним экземпляров сущности-потомка;
- каждый экземпляр сущности-родителя должен иметь не менее одного связанного с ним экземпляра сущности-потомка;
- каждый экземпляр сущности-родителя должен иметь не более одного связанного с ним экземпляра сущности-потомка;
- каждый экземпляр сущности-родителя связан с некоторым фиксированным числом экземпляров сущности-потомка.

На логическом уровне можно установить идентифицирующую связь «один-ко-многим», связь «многие-ко-многим» и неидентифицирующую связь «один-ко-многим».

Идентифицирующая связь устанавливается между **независимой** (родительский конец связи) и **зависимой** (дочерний конец связи) **сущностями**. Когда рисуется идентифицирующая связь, ERwin автоматически преобразует дочернюю сущность 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 внешний ключ.

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

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

Некоторые реально существующие сущности являются **категориями** других сущностей. В IDEF1X-модели они связаны друг с другом через категориальное отношение. Сущности-категории являются взаимоисключающими. Существует также отношение неполной категоризации, когда имеется экземпляр общей сущности, не связанный ни с каким экземпляром из сущностей-категорий.

Правила отношений категоризации

- Сущность-категория может иметь только одну общую сущность.
- Сущность-категория не может быть сущностью-потомком в идентифицирующем отношении.
- Атрибуты первичного ключа сущности-категории должны совпадать с атрибутами первичного ключа общей сущности.
- Атрибуты представляют тип характеристик или свойств, ассоциированных с множеством реальных или абстрактных объектов. Экземпляр атрибута определяется типом характеристики и ее значением. Атрибуты изображаются в виде списка имен внутри блока сущности.

Ключи

Как было сказано выше, каждый экземпляр сущности должен быть уникален и должен отличаться от других атрибутов.

Первичный ключ (Primary Key) – это потенциальный ключ, который будет в большинстве случаев использован для однозначной идентификации единственного экземпляра

сущности. Атрибуты первичного ключа находятся в списке атрибутов выше горизонтальной линии.

В одной сущности могут оказаться несколько атрибутов или наборов атрибутов, претендующих на роль первичного ключа. Такие претенденты называются **потенциальными ключами (Candidate Key)**.

Альтернативный ключ (Alternate Key) – это любой непотенциальный ключ, который не установлен в качестве первичного.

Правила первичных и альтернативных ключей

- Каждая сущность должна обладать первичным ключом.
- Атрибуты первичного ключа не должны содержать нулевых значений.
- Значение атрибутов первичного ключа не должно меняться в течение всего времени существования экземпляра сущности.
- Каждая сущность может обладать любым числом альтернативных ключей.
- Первичный или альтернативный ключ могут состоять из одного атрибута или комбинации атрибутов.
- Один атрибут может быть частью более чем одного первичного или альтернативного ключа.
- Атрибуты, входящие в первичные ключи сущности, могут быть собственными для сущности или наследоваться через отношения.
- Первичные и альтернативные ключи должны содержать только необходимые для однозначной идентификации атрибуты, т.е. при исключении из ключа любого атрибута не все экземпляры сущности могут быть однозначно определены.
- Если первичный ключ состоит более чем из одного атрибута, то значение любого неключевого атрибута должно функционально зависеть от всего первичного ключа (правило полной функциональной зависимости).
- Каждый неключевой атрибут должен функционально зависеть только от первичного и альтернативных ключей, т.е. значение неключевого атрибута не может определяться значением другого неключевого атрибута (правило отсутствия транзитивной зависимости).

При проведении связи между двумя сущностями в дочерней сущности автоматически образуются **внешние ключи (Foreign Key)**. Они могут использоваться в качестве части или целого первичного ключа или неключевого атрибута. Связь образует ссылку на атрибуты первичного ключа в дочерней сущности, и эти атрибуты образуют внешний ключ в дочерней сущности. Внешний ключ изображается с помощью помещения внутрь блока сущности имен атрибутов, после которых следуют буквы FK в скобках.

Правила внешних ключей

- Каждая сущность должна содержать отдельный внешний ключ для каждого отношения связи или категоризации, в котором эта сущность является сущностью-потомком и сущностью-категорией.
- Первичный ключ общей сущности должен наследоваться в качестве первичного ключа для каждой сущности-категории.
- Каждый наследуемый атрибут сущности-потомка или сущности-категории должен представлять атрибут из первичного ключа связанной родительской или общей сущности.
- Каждый атрибут первичного ключа родительской или общей сущности должен быть наследуемым атрибутом в связанной сущности-потомке или сущности-категории.

Вопрос 3. Нормализация отношений

Нормализация отношений – это процесс построения оптимальной структуры таблиц и связей в реляционной БД (процесс уменьшения избыточности информации).

В процессе нормализации данные группируются в таблицы, представляющие классы объектов и их взаимодействие.

Цели, которые преследуются при построении наиболее эффективной структуры данных:

1. обеспечить быстрый доступ к данным;
2. исключить ненужное повторение данных, которое может являться причиной ошибок при вводе, а также привести к нерациональному использованию дискового пространства;
3. обеспечить целостность данных, т.о. чтобы при изменении одних объектов автоматически происходило соответствующее изменение связанных с ними объектов.

Теория нормализации отношений работает с 6 нормальными формами таблиц. Каждая последующая форма должна отвечать требованиям предыдущих форм плюс некоторым дополнительным требованиям. На практике обычно ограничиваются приведением данных к третьей нормальной форме.

Первая нормальная форма (1НФ)

Таблица, находящаяся в первой нормальной форме должна отвечать следующим требованиям:

- таблица не должна иметь повторяющихся записей,
- в таблице должны отсутствовать повторяющиеся группы полей.

Пример 1 НФ представлен на рис. 3.1.

Проект
Название проекта
Табельный номер руководителя
Дата начала
Дата завершения
Фамилия
Имя
Отчество
Должность
Оклад

Рисунок 3.1 – Сущность «Проект» в 1НФ

Для приведения к 1НФ можно использовать следующий алгоритм:

1. Определить поле, которое можно назначить первичным ключом. Если такого поля нет, то добавить новое уникальное ключевое поле.
2. Определить группы повторяющихся полей.
3. Вынести группы повторяющихся полей в отдельные таблицы, в основной таблице остается одно поле для организации связи между таблицами.
4. Назначить первичные ключи в новых таблицах. В качестве ключевых полей можно использовать поля таблицы или добавить новое поле. Если ключевое поле имеет большой размер, предпочтительней добавлять новое поле.
5. Определить тип отношения между таблицами.

Вторая нормальная форма (2НФ)

Таблица, находящаяся во второй нормальной форме должна отвечать всем требованиям 1НФ, а также любое неключевое поле однозначно идентифицируется полным набором

ключевых полей. 2НФ применяется к таблицам, которые имеют составной ключ. Частично зависимое поле – это поле, зависящее только от части ключа (рис. 3.2).

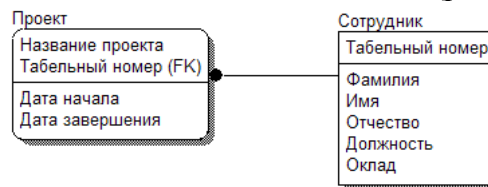


Рисунок 3.2 – Сущность «Проект» во 2НФ

Для приведения к 2НФ необходимо:

1. Вынести все частично зависимые поля в отдельную таблицу.
2. Определить ключевые поля.
3. Установить отношения между таблицами.

Третья нормальная форма (3НФ)

Таблица, находящаяся в третьей нормальной форме должна отвечать всем требованиям 2НФ, а также ни одно из неключевых полей не идентифицируется при помощи другого неключевого поля. Другими словами в таблице нет полей, которые не зависят от ключа (рис. 3.3).

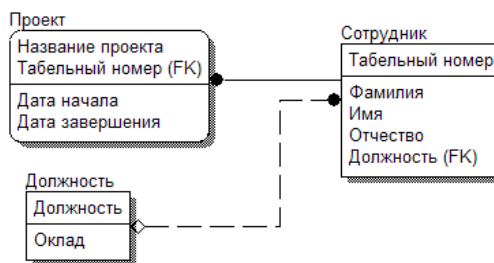


Рисунок 3.3 – Сущность «Проект» в 3НФ

Вопрос 4. Построение логической модели данных

1. Запустите ERwin. В появившемся диалоговом окне ModelMart Connection Manager нажмите Cancel.
2. В диалоговом окне Computer Associates ERwin выберите позицию Create a new model и нажмите ОК. Необходимо отметить, что модель данных будет строиться сразу в третьей нормальной форме.
3. Появится диалоговое окно Create Model – Select Template (рис. 4.1). В нем выберите тип модели – Logical / Physical, выберите в нижней части окна из списка СУБД из списка и нажмите ОК. В результате откроется окно диаграммы логического уровня представления данных (рис. 4.2).

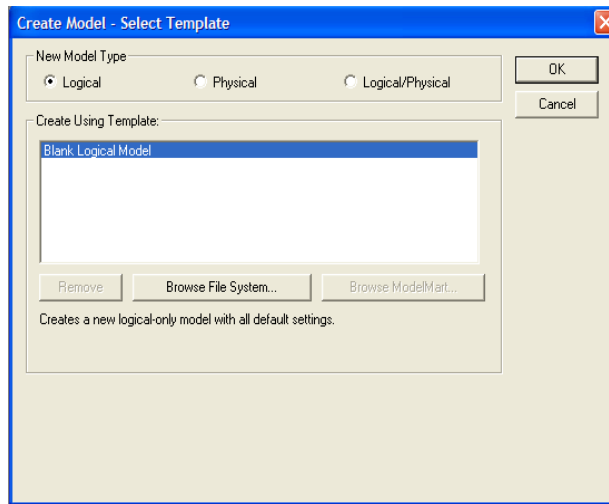


Рисунок 4.1 – Диалоговое окно Create Model – Select Template

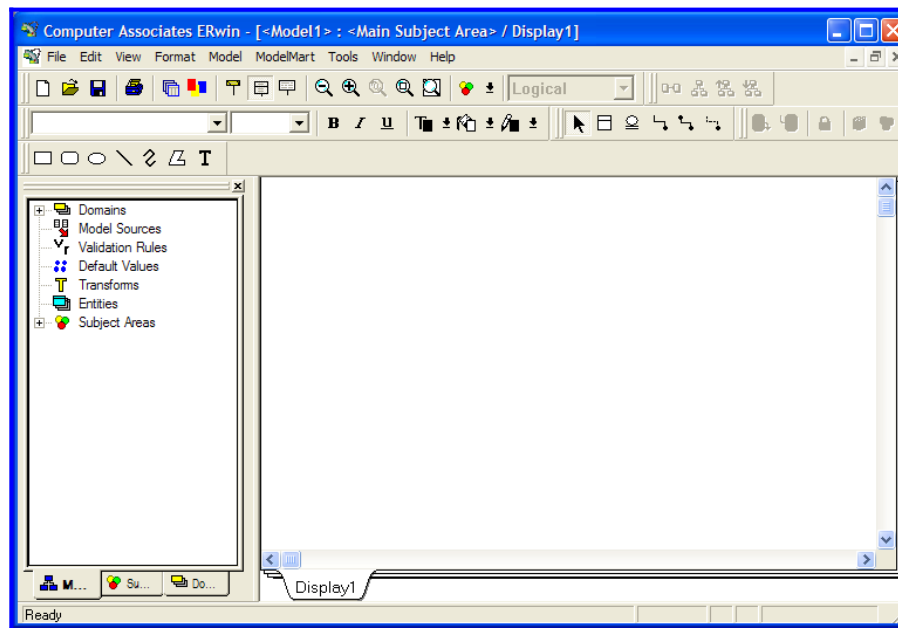








Рисунок 4.2 – Окно диаграммы логического уровня представления данных

На логическом уровне палитра инструментов имеет следующие кнопки:

-  – указатель (режим мыши) – в этом режиме можно установить фокус на каком-либо объекте модели;
-  – сущность;
-  – категория (или категориальная связь – специальный тип связи между сущностями). Для установления категориальной связи нужно щелкнуть левой кнопкой мыши по кнопке категории, затем один раз щелкнуть по сущности – родовому предку, затем – по сущности-потомку;
-  – идентифицирующая связь;
-  – связь «многие-ко-многим»;
-  – неидентифицирующая связь.

4. Установите шрифт и цвет объектов модели данных. Для этого можно воспользоваться одним из трех описанных ранее способов. Например, измените шрифт и цвет для всех объектов модели через пункт меню Format/ Default Font&Color. При этом во всех вкладках установите значение Font – Arial CYR (рис. 4.3).

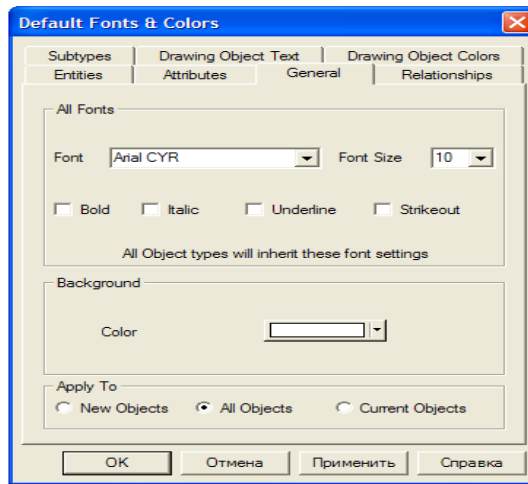


Рисунок 4.3 – Диалоговое окно Default Font & Color

5. Перед тем как приступить к построению сохраните модель данных под именем «Обучение студентов». Обратите внимание на переименовавшийся заголовок модели в верхней части окна диаграммы.

6. Первым этапом построения модели является определение состава сущностей. Для нашего примера необходимо в модель включить следующие сущности: Студент, Факультет, Специальность, Курс, Группа, Подгруппа, Список группы, Дисциплина, Преподаватель, Должность, Приказ об отчислении, Приказ о переводе, Зачетно-экзаменационная ведомость. Для этого необходимо воспользоваться кнопкой «Сущность» (Entity) на палитре инструментов.

7. По умолчанию ERwin автоматически всем сущностям присваивает имя E и их порядковый номер в зависимости от очередности создания (например, E/1). Для переименования сущности есть несколько способов.

Во-первых, ее переименовать можно в окне диаграммы. Для этого сделайте двойной щелчок по имени сущности. Появится блок редактирования вокруг имени. Наберите имя сущности вместо имени по умолчанию и кликните мышью вне блока редактирования (рис. 4.4).

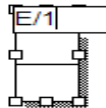


Рисунок 4.4 – Сущность

Во-вторых, в Model Explorer щелкните правой кнопкой мыши по сущности и из контекстного меню выберите «Rename». Наберите имя и щелкните мышкой вне блока редактирования.

В-третьих, если щелкнуть два раза левой клавишей мыши по сущности, появится окно Attributes (рис. 4.5).

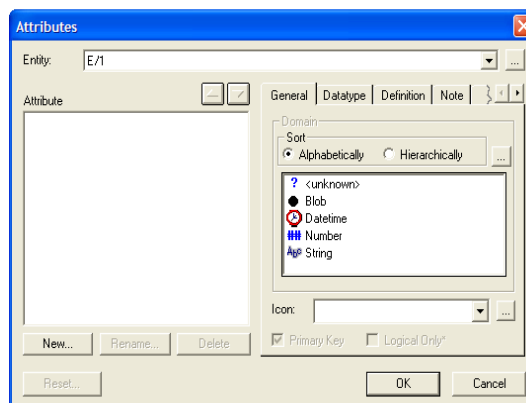



Рисунок 4.5 – Окно Attributes

После нажатия кнопки  в окне Entities в текстовом поле Name можно написать имя сущности (рис. 4.6).

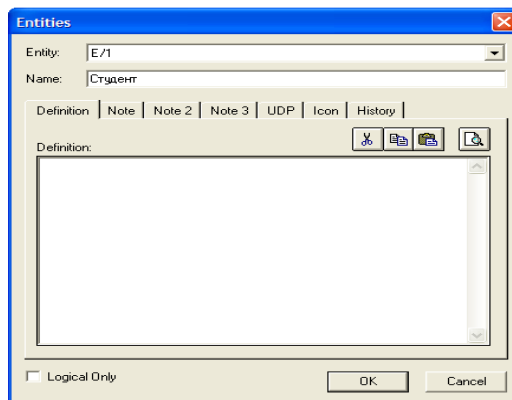


Рисунок 4.6 – Окно Entities

Таким образом, внесите в модель данных вышеперечисленные сущности.

8. Следующий этап создания модели состоит в задании атрибутов для каждой сущности. Для этого в окне Attributes в поле Attribute с помощью кнопки New заносятся все необходимые атрибуты сущностей.

В нем для работы с атрибутами используются следующие вкладки:

General – позволяет указать домен атрибута, сделать отметку в окне выбора Primary Key для атрибутов первичного ключа и связать иконку с атрибутом.

Datatype – позволяет задать тип данных, правила валидации и значение по умолчанию. *Valid:** – ограничение целостности; *Default:** – значение по умолчанию.

Definition – позволяет записывать определения отдельных атрибутов.

Note – позволяет добавлять замечания об одном или нескольких атрибутах сущности, которые не вошли в определения.

UDP – задает значение свойств, определяемых пользователем. Предварительно эти свойства должны быть внесены в диалог User-Defined Property Editor как свойства атрибутов.

Key Group – позволяет включить атрибут в состав первичного, альтернативного или инвертированного ключа.

History – отображает историю создания и изменения свойств атрибутов.

Отметьте необходимые атрибуты в качестве первичных ключей в окне выбора Primary Key. Задайте атрибуты сущностям согласно рис. 4.7.

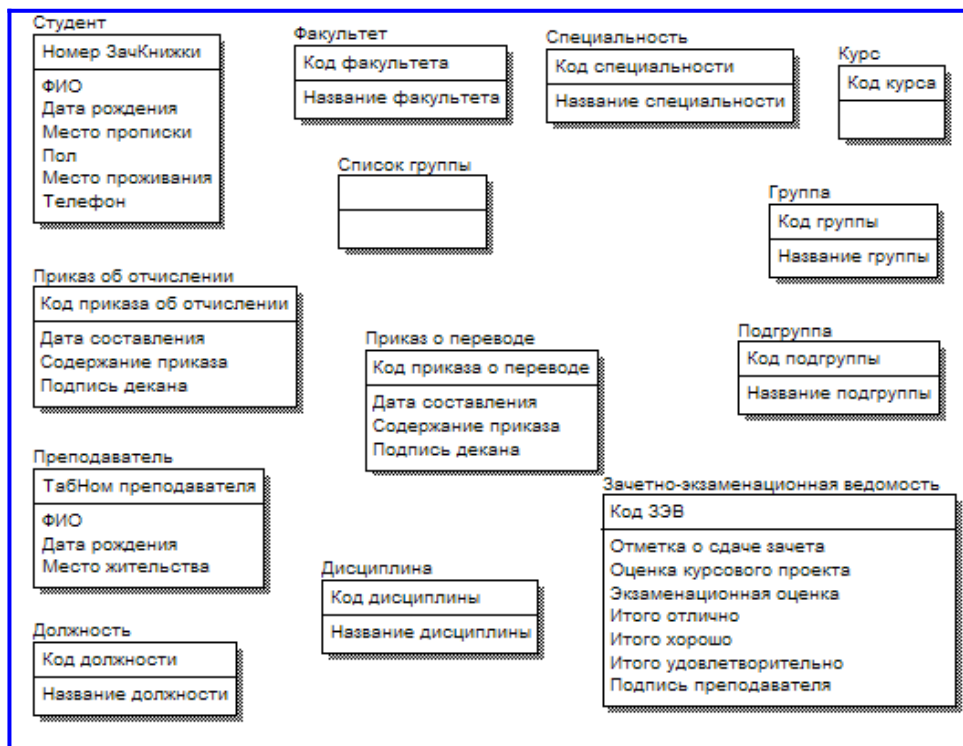


Рисунок 4.7 – Модель данных с сущностями и атрибутами

Также добавить атрибуты можно через Model Explorer. Для этого щелкните в плюсе около рассматриваемой сущности. Это расширит объектный список. Щелкните правой кнопкой мыши по «Attributes» и выберите «New». Появится новый атрибут, который надо будет переименовать.

9. После определения состава сущностей и их атрибутов необходимо расставить связи между ними, воспользовавшись кнопками идентифицирующей связи «один-ко-многим» и неидентифицирующей связи «один-ко-многим» на палитре инструментов. Свяжите сущности согласно рис. 4.8.

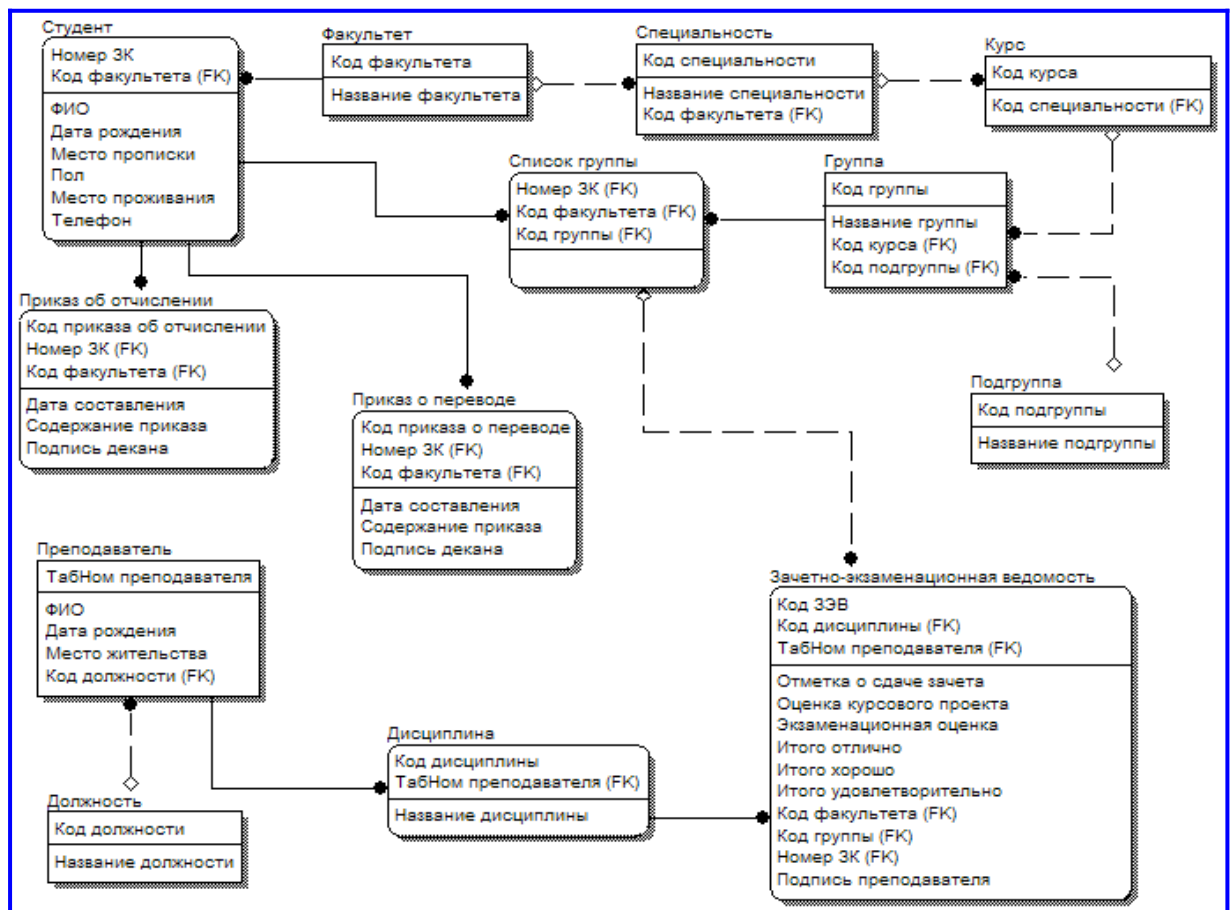


Рисунок 4.8 – Модель данных в третьей нормальной форме

10. ERwin имеет несколько уровней отображения диаграммы: уровень сущностей, уровень атрибутов, уровень определений, уровень первичных ключей и уровень иконок. Переключиться между первыми тремя уровнями можно с использованием кнопок панели инструментов. Переключиться на другие уровни отображения можно при помощи контекстного меню, которое появляется, если «кликнуть» по любому месту рабочего окна, не занятому объектами модели. В контекстном меню следует выбрать пункт Display Level (рис. 4.9) и затем – необходимый уровень отображения.

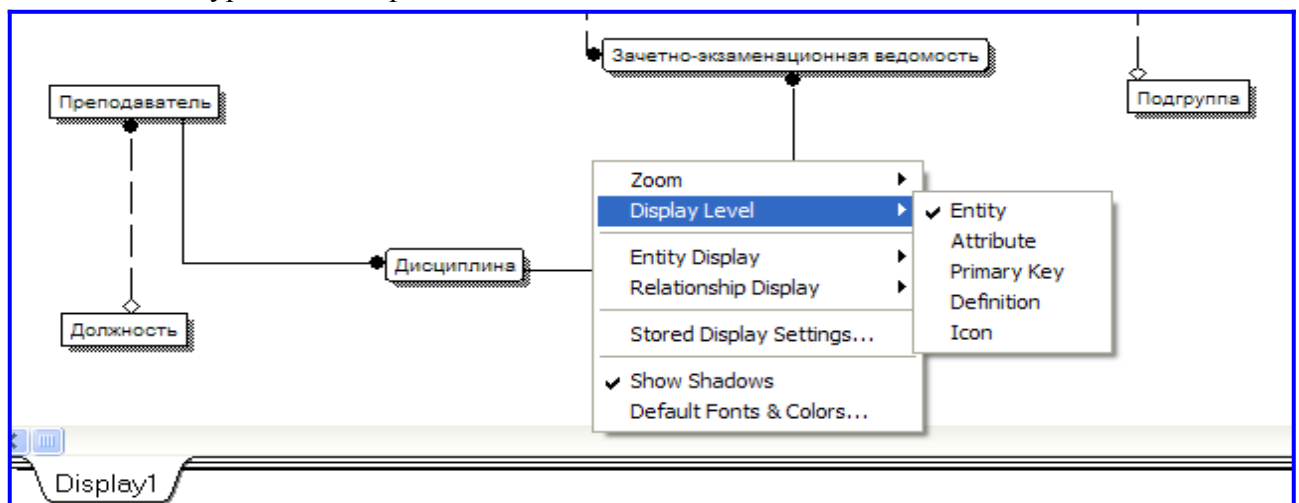


Рисунок 4.9 – Выбор уровней отображения диаграммы

Также уровень отображения можно выбрать через меню Format / Display Level на панели меню.

Контрольные вопросы

1. В чем различие между моделями данных, представленных в форме диаграммы «сущность-связь», на основе ключей и в виде полной атрибутивной модели?
2. Понятие и формы нормализации отношений.
3. Какие основные компоненты содержат модели данных, представленные в методологии IDEF1X?

Задание на СРС

1. Построить логическую модель данных в 3 нормальной форме для рассматриваемого процесса.