

## **Лабораторная работа №2. Разработка логической модели данных и формулирование отношений между сущностями.**

**ЦЕЛЬ РАБОТЫ:** Ознакомиться более подробно с особенностями моделирования данных, основными правилами выделения сущностей, атрибутов, отношений, научиться строить произвольные модели с использованием специализированных редакторов ERWIN и применить полученные знания для разработки модели базы данных.

### **Создание логической модели**

Первым шагом при создании логической модели является создание диаграммы зависимостей сущностей (Entity Relationship Diagram, ERD) – модели данных высокого уровня, описывающей широкие области бизнеса.

ERD состоит из трех основных блоков: сущностей, атрибутов и связей. Если рассматривать диаграмму как некий графический язык, то можно заметить, что объекты являются существительными, атрибуты – прилагательными, а зависимости – глаголами этого языка. Построение модели данных в ERwin заключается в, как бы, нахождении правильного набора существительных, глаголов и прилагательных и в правильном их сочетании.

Основная задача ERD – оценить, какие требования к бизнес - информации будут достаточными для обеспечения нужд планирования разработки информационной системы. Эти модели не очень подробны (включены только основные сущности), атрибуты тоже слабо детализируются. Разрешены отношения многие ко многим, ключи, как правило, не включаются. Эта модель, в основном, предназначена для презентации и обсуждения.

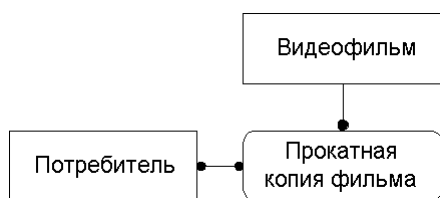


Рис. 1. Пример диаграммы зависимостей сущностей

Так же, как таблицы и колонки образуют физическую модель реляционной базы данных, Диаграмма зависимостей сущностей (и другие логические модели) включают в себя компоненты, которые позволяют смоделировать структуры данных для вашего бизнеса.

Логическим эквивалентом таблицы является сущность, а логическим эквивалентом колонки является атрибут.

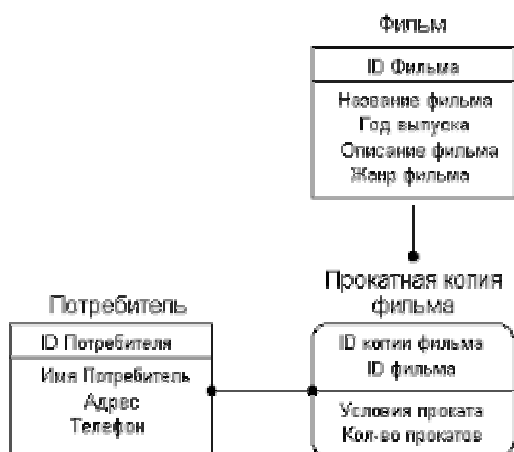


Рис. 2. ERD с атрибутами

В ERD названия сущностей могут быть только в единственном числе – потребитель (а не потребители), фильм (а не фильмы), страна (а не страны).

Взаимосвязи между таблицами являются жизненно важными компонентами реляционных баз данных. Эти связи создаются за счет использования общих ключей, т. е. записи в одной таблице ссылаются на записи в другой таблице. Связь между двумя сущностями также включает то, что факты одной сущности ссылаются или ассоциированы с фактами другой сущности. Так, на рис.2 информация о потребителе и копии фильма взаимосвязана, и эта связь может быть выражена так: Потребитель берет напрокат одну или более Прокатных копий фильма.

### Определение сущностей и атрибутов

*Сущность* служит для представления набора реальных или абстрактных предметов (людей, мест, событий и т.п.), которые обладают общими атрибутами или характеристиками. Сущность в ERwin обычно описывается тремя компонентами: атрибуты, являющиеся первичными ключами, неключевые атрибуты и тип сущности.

ERwin имеет два типа сущностей: независимые и зависимые. *Независимая сущность* - это сущность, экземпляры которой могут быть уникальным образом идентифицированы без определения ее связи с другой сущностью. *Зависимая сущность* - это сущность, экземпляры которой не могут быть уникальным образом идентифицированы без определения ее связи с другой сущностью или сущностями.

Сущности могут быть определены в виде какого-либо лица, места, предмета, события, о которых содержится соответствующая информация. Если говорить точнее, то сущность

можно представлять как набор отдельных экземпляров (записей). Экземпляр – это конкретная реализация сущности. Каждый экземпляр должен быть отличен от остальных.

В примере (рис. 2), каждый экземпляр сущности Потребитель содержит следующую информацию: Id потребителя, имя потребителя, адрес потребителя. В логической модели эти свойства называются атрибутами сущности. Каждый атрибут содержит только часть информации о сущности.

### Логические связи

Данные, относящиеся к связям, очень важны и часто являются критическими данными, которые мы используем в повседневном бизнесе. Например, важно знать о каком-то типе инструмента, но знание того, к кому относится конкретный инструмент (связь между человеком и инструментом) может иметь критическую важность. Связь - это соотношение либо между двумя сущностями, либо между сущностью и этой же сущностью. Связь - “логический” объект, представленный одним или несколькими атрибутами - внешними ключами. Связи играют роль ссылок, соединений и ассоциаций между сущностями. Связи – это, собственно, глаголы, которые показывают, как соотносятся сущности между собой.

Простые для понимания правила помогают определять ограничения, накладываемые на данные, а также точно определять атрибуты связей. Примером связи может быть:

Автобус <перевозит> нескольких Пассажира.

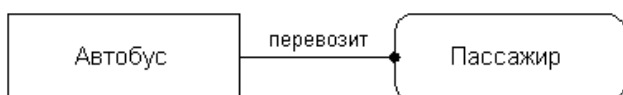


Рис. 3. Пример связи

В этом примере взаимосвязи между сущностями соответствуют схеме один ко многим. Это означает, что один экземпляр первой сущности связан с несколькими экземплярами второй сущности. Причем первая сущность называется родительской, а вторая – дочерней. Взаимосвязи отображаются в виде линии между двумя сущностями с точкой на одном конце и глагольной фразой, отображаемой над линией.

### Отношение многие - ко - многим

Отношение многие ко многим, также называемое неопределенным отношением, отображает ситуацию, когда экземпляр в одной сущности относится к одному или нескольким экземплярам второй сущности, а экземпляр во второй сущности относится к одному или нескольким экземплярам первой сущности. Отношения многие ко многим обычно

используются на начальной стадии разработки диаграммы, например, в диаграмме зависимости сущностей (ERD) и отображаются в IDEF1X в виде сплошной линии с точками на обоих концах.

Так как отношения многие ко многим могут скрыть другие бизнес правила или ограничения, они должны быть полностью исследованы на одном из этапов моделирования. Например, иногда отношение многие ко многим на ранних стадиях моделирования идентифицируется неправильно, на самом деле представляя два случая отношений один-ко-многим между связанными сущностями.

## Разработка сущностей и атрибутов

ERwin предлагает набор специальных окон-диалогов (редакторов), которые используются для определения характеристик сущности, атрибута, связи или другого объекта на диаграмме. Если Вы щелкнете по объекту в окне диаграммы, а затем откроете контекстное меню, то в его начале будут перечислены одни или несколько редакторов, доступных для этого объекта. Каждый редактор ERwin содержит специальный набор управляющих функций, позволяющих Вам задавать характеристики, применимые к данному типу объекта.

Для сущности или ее атрибута доступны три редактора: редактор сущности (*Entity Editor*), редактор атрибутов (*Attribute Editor*) и редактор ключей (*Key Group Editor*).

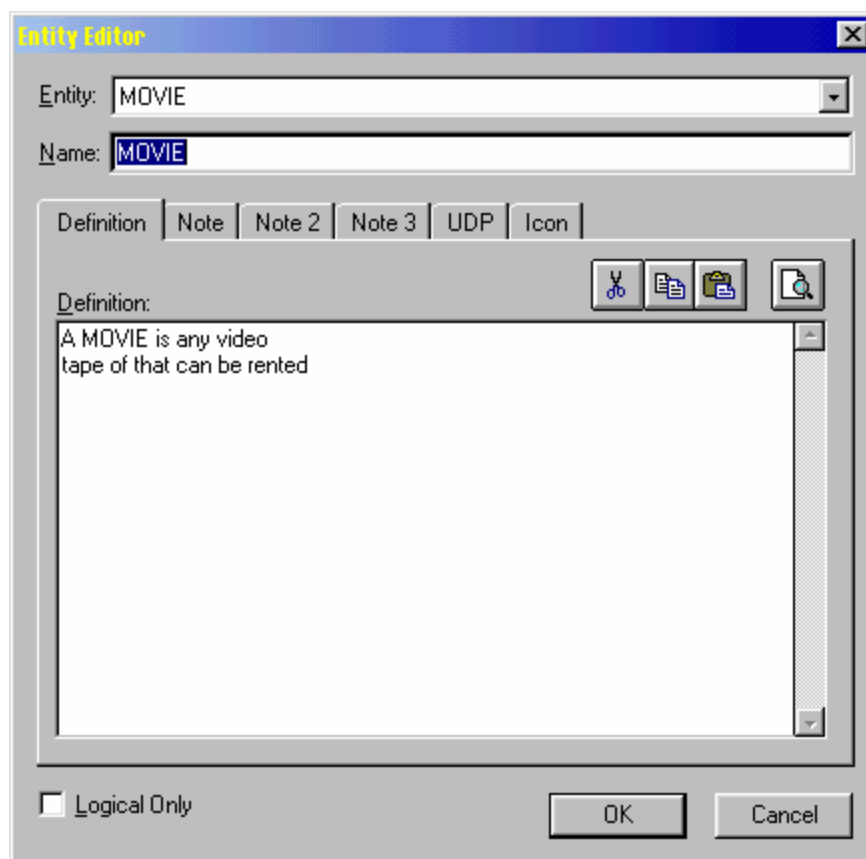


Рис.4. Редактор сущностей

Редактор сущностей (см. рис. 4) наиболее прост. Параметры сущности в большинстве своем носят описательный характер (Definition, Note, Icon). Флажок Logical Only позволяет создать чисто логическую сущность (не отражается в физическую модель). С помощью списка Entity можно переключаться между различными сущностями модели. Редактор Entity Definition используется для ввода определения сущности. Эти определения полезны на логическом уровне, поскольку они помогают людям, читающим модель, понять, что это за объект. Они полезны и на физическом уровне, поскольку их можно экспортировать как часть Вашей схемы и использовать в реальной базе данных (CREATE COMMENT on entity\_name). Редактор Entity Note позволяет Вам добавлять дополнительные замечания (notes) о сущности, которые не были отражены в определении, введенном в редакторе Entity Definition. Здесь можно ввести полезное замечание, описывающее правило (Business Rule) или соглашение по организации диаграммы, которое Вы приняли.

Редактор атрибутов (см. рис. 5) позволяет настраивать все аспекты атрибутов сущности. Кнопки New, Rename, Delete управляют списком атрибутов, а блокнот справа – параметрами отдельного атрибута. Группа Domain позволяет менять домен атрибута, флажок Primary Key определяет атрибут как атрибут первичного ключа, а Required – обязательный атрибут (см. рис 5). Редактор Attribute Definition позволяет Вам записывать определения отдельных атрибутов. Определения используются для обращения к каждому объекту модели. Определения атрибутов можно также сгенерировать как часть схемы (CREATE COMMENT on entity\_name.attribute\_name).

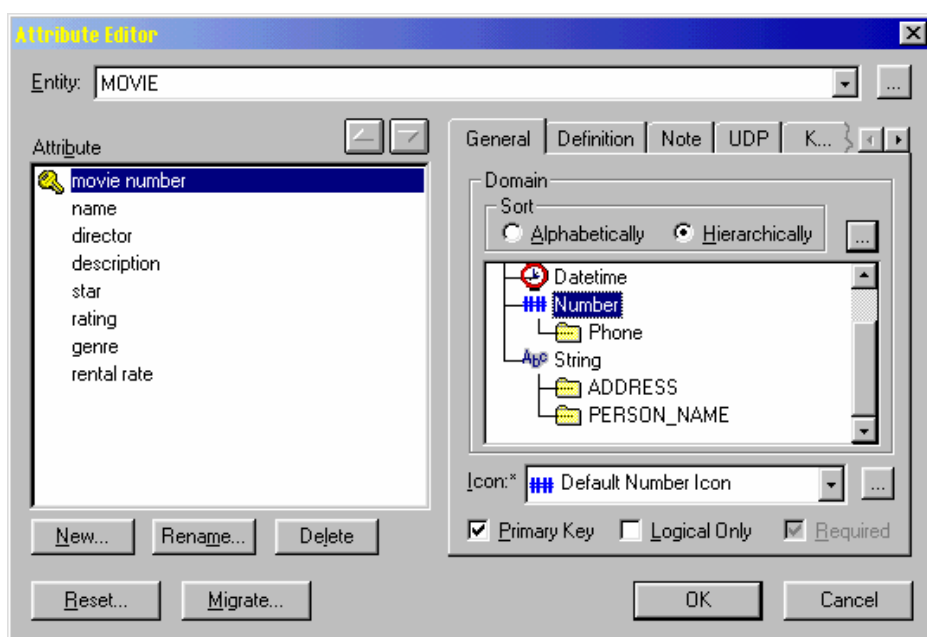


Рис.5. Интерфейс редактора атрибутов.

Редактор ключей используется в тех случаях, когда требуется указать дополнительные ключи (помимо первичного) сущности или изменить их параметры. Интерфейс этого редактора (см. рис. 6) стандартен – вверху находится перечень сущностей, под ним – список ключей текущей сущности, а внизу – параметры отдельного ключа. В ERwin ключи бывают 4 типов: первичный (primary key – PK), внешний (foreign key – FK), альтернативный (alternative key – AK) и инверсный вход (inversion entry – IE). Альтернативным ключом называется атрибут или группа атрибутов, уникальным образом определяющие экземпляр сущности. Inversion Entry - это атрибут, который не определяет экземпляр сущности уникальным образом, но часто используется для обращения к экземплярам сущности.

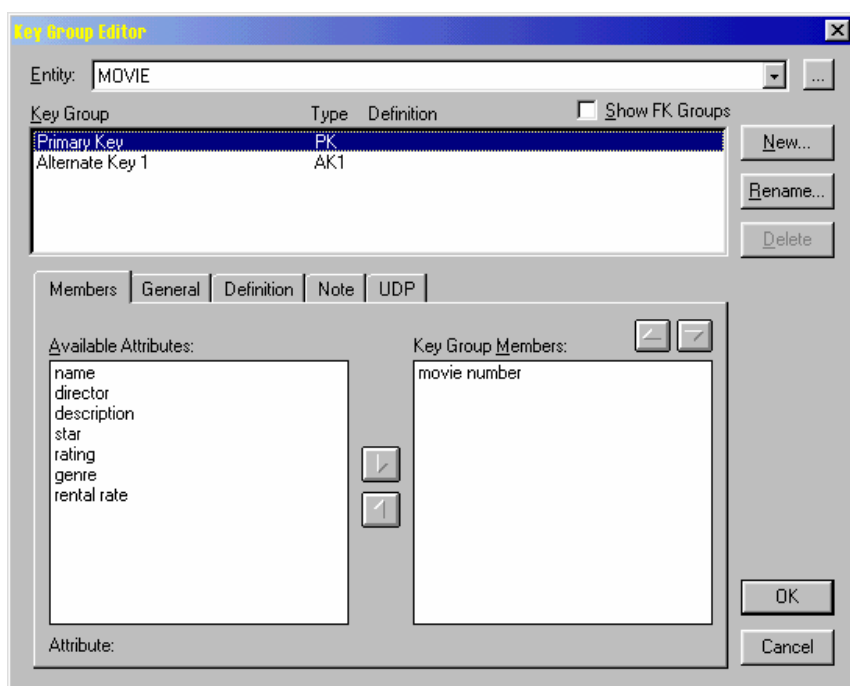


Рис.6. Редактор ключей сущности

### Выбор первичного ключа

Выбор первичного ключа для сущности является очень важным шагом. В качестве первичных ключей могут быть использованы несколько атрибутов или наборов атрибутов. Атрибуты или группы атрибутов, которые могут быть выбраны первичными ключами, называются кандидатами в ключевые атрибуты. Кандидаты в ключи должны уникально идентифицировать каждую запись сущности. В соответствии с этим, ни одна из частей ключа не может быть NULL, не заполненной или отсутствующей. Правила устанавливают, что атрибуты и группы атрибутов должны:

- уникальным образом идентифицировать экземпляр.
- не использовать NULL значений.

- не изменяться со временем. Экземпляр идентифицируется при помощи ключа. При изменении ключа, меняется экземпляр.

## Разработка отношений в модели

Связь в ERwin обычно содержит пять типов информации: тип связи, родительский конец связи, дочерний конец связи, знак “обязательности” связи и кардинальность связи.

На ERwin Toolbox есть пять типов связей: идентифицирующая, неидентифицирующая, полный кластер подтипов, неполный кластер подтипов, неопределенная (связь ”многие-ко-многим”). *Идентифицирующая связь* - такая связь, при которой экземпляр дочерней сущности идентифицируется через свою ассоциацию с родительской сущностью. Атрибуты первичного ключа родительской сущности становятся атрибутами первичного ключа дочерней. *Неидентифицирующая связь* - это такая связь, при которой экземпляр дочерней сущности не идентифицируется через свою ассоциацию с родительской сущностью. Атрибуты первичного ключа родительской сущности становятся неключевыми атрибутами дочерней.

### Создание связи

Создать связь в ERwin легко. Для этого необходимо:

- в Toolbox выбрать инструмент, соответствующий нужному типу связи
- щелкнуть сначала по родительской, а затем по дочерней сущности.

Если Вам не нравится форма линии связи, ее можно изменить. Для этого Вам нужно захватывать мышью углы линии связи и переносить их с места на место, пока линия не начнет выглядеть лучше.

Когда связь сформирована, то ключ родительской сущности автоматически появится в качестве части ключа дочерней (для идентифицирующей связи) или в качестве неключевого атрибута дочерней сущности (для неидентифицирующей связи), причем рядом с ним будет стоять (FK), то есть внешний ключ, Foreign Key. Эта операция изменения дочерней сущности вследствие создания связи называется *миграцией ключа*.

Для связей, также как и для сущностей и атрибутов, имеются собственные редакторы – редактор связи (Relationship Editor) и редактор шаблонов связей (Relationship Template Editor). Редактор связей (см. рис. 7) позволяет определить мощность (Cardinality), тип (Type), обязательность (Nulls), глагольные фразы (Verb Phrase) связи, роли мигрирующих атрибутов и правила ссылочной целостности. Для отношений «многие-ко-многим» возможно их разрешение (с выделением промежуточной сущности). Для этого применяется команда Resolve many-to-many контекстного меню.

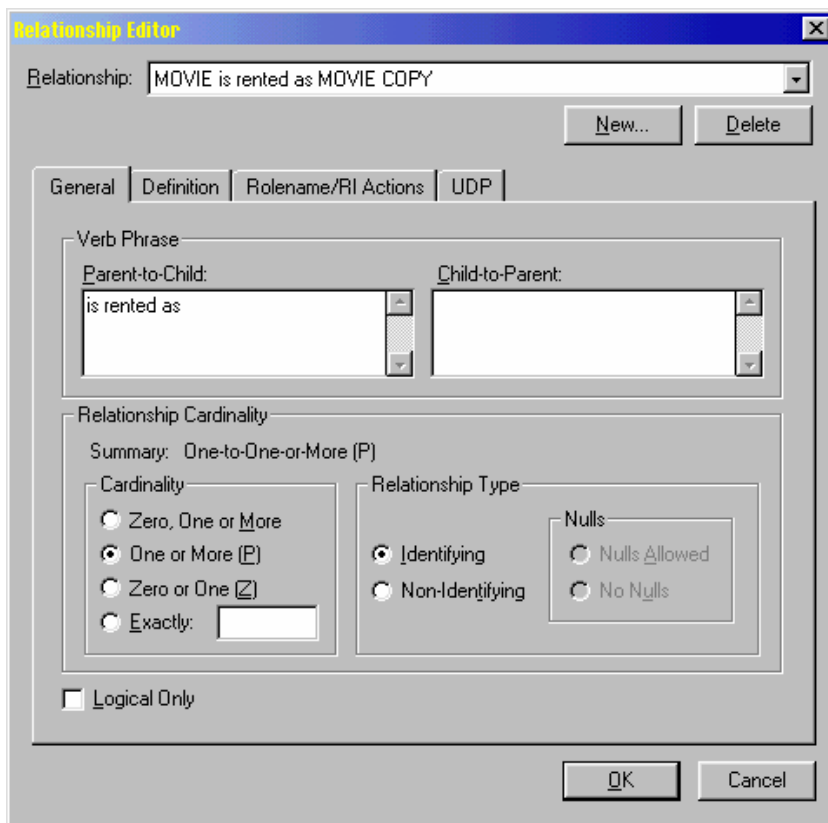


Рис. 7. Редактор связей модели

### Задание кардинальности связи

*Кардинальность связи (мощность)* служит для обозначения отношения числа экземпляров родительской сущности к числу экземпляров дочерней. Родительская сущность может связываться с дочерней одним из четырех способов. В IDEF1X кардинальность бинарных отношений равна 1:n, где n может принимать значения:

- 0, 1 или более - обозначается “пустым местом”. Каждая родительская сущность связана с 0, 1 или более экземпляров дочерней.
- 1 или более - обозначается буквой “P”. Каждая родительская сущность связана с 1 или более экземпляров дочерней.
- 0 или 1 - обозначается буквой “Z”. Каждая родительская сущность связана с 0 или 1 экземпляром дочерней.
- ровно n, где “n” - некоторое число. Каждая родительская сущность связана с ровно n экземплярами дочерней.

### Задание режима Null (нулевых значений)

В редакторе Relationship Вы можете задать также режим “нулевых значений”. Он относится к случаю, когда значения нет: это пустое множество или “нет колонки”. Атрибут, который может принимать нулевое значение, считается “необязательным” и на диаграмме изображается как ромбик на родительском конце связи. Вы можете задать этот режим и в



других местах, но вообще говоря, первая такая возможность предоставляется именно в этом редакторе.

### Имена ролей

Когда атрибут первичного ключа мигрирует из родительской сущности в дочернюю, то там он становится атрибутом внешнего ключа. Поскольку у внешнего ключа может быть иная роль, отличающаяся от роли связанного с ним первичного ключа, то ERwin разрешает Вам связать с внешним ключом *имя роли*. Имя роли - это псевдоним, присваиваемый атрибуту внешнего ключа. Имя роли состоит из трех частей: *присваиваемое имя роли*, *точка-разграничитель* и *базовое имя*. Базовое имя - это имя атрибута, которое мигрировало из родительской сущности в дочернюю.

При рекурсивной связи, когда сущность связана сама с собой, один и тот же атрибут может быть и первичным, и внешним ключом, и имя роли помогает прояснить вторую роль атрибута.

Зависимая сущность может унаследовать один и тот же внешний ключ из нескольких родительских сущностей. В этом случае ERwin распознает, что два атрибута одинаковы, и предполагает, что Вы хотите видеть атрибут в зависимой сущности только один раз. Это комбинирование, или объединение идентичных атрибутов называется *унификацией*.

## Задание на лабораторную работу

1. Внимательно изучить материалы введения и разобраться со структурой интерфейса упомянутых редакторов.
2. Разработать модель базы данных для решения следующей задачи. Пусть имеется библиотека, в которой находится определенное количество книг. Для каждой книги в библиотеке имеется учетная карточка, в которой указаны: полное название, дата и место издания, объем, автор или авторы, код по библиотечному каталогу, аннотация, прочие параметры. Требуется построить такую модель базы данных, которая обеспечит непротиворечивое хранение таких сведений и возможность автоматизированного поиска литературы по названию, году издания, издательству, каждому автору и т.п.
  - 2.1. Разработать полную атрибутивную модель данной предметной области. Сущность, описывающая отдельную книгу должна называться «книга».
  - 2.2. Связать сущности модели между собой в случае необходимости. Конкретный вид связей выбрать исходя из жизненного опыта. С помощью поля Notes редактора связей сформулировать принципы выделения найденных связей.
  - 2.3. Указать для каждой связи глагольные фразы.

- 2.4. Выделить для сущности книга альтернативный ключ и оформить его в редакторе ключей.
- 2.5. Для одной из сущностей модели выделить инверсный вход и зафиксировать его.
3. Сохранить полученную модель в своей папке под именем task2-1.er1

### **Контрольные вопросы**

1. Что такое ERD и какие элементы используются для построения ERD? Каковы правила выделения и именования элементов ERD?
2. Перечислите инструменты, используемые для создания и настройки сущностей и атрибутов, поясните устройство из интерфейса.
3. Что такое роль атрибута и в каких случаях необходимо ее использовать?
4. Какие типы ключей имеются в ERwin и каковы их особенности?
5. Какие правила существуют для выделения первичного ключа и какую роль для сущности играет первичный ключ ?