

Семантическое моделирование данных

Реляционная модель данных достаточна для моделирования предметных областей. Однако проявляется ограниченность реляционной модели данных в следующих аспектах:

- Модель не предоставляет достаточных средств для представления смысла данных.
- Для многих приложений трудно моделировать предметную область на основе плоских таблиц.
- Хотя весь процесс проектирования происходит на основе учета зависимостей, реляционная модель не предоставляет каких-либо средств для представления этих зависимостей.
- Несмотря на то, что процесс проектирования начинается с выделения некоторых существенных для приложения объектов предметной области («сущностей») и выявления связей между этими сущностями, реляционная модель данных не предлагает какого-либо аппарата для разделения сущностей и связей.

Указанные ограничения вызвали к жизни направление семантических (концептуальных, инфологических) моделей данных. Любая развитая семантическая модель данных, как и реляционная модель, включает структурную, манипуляционную и целостную части. Главным назначением семантических моделей является обеспечение возможности выражения семантики данных. На практике семантическое моделирование используется на первой стадии проектирования базы данных. При этом в терминах семантической модели производится концептуальная схема базы данных, которая затем

- а) Либо вручную преобразуется к реляционной (или какой-либо другой) схеме.
- б) Либо реализуется автоматизированная компиляция концептуальной схемы в реляционную.
- в) Либо происходит работа с базой данных в семантической модели, т.е. под управлением СУБД, основанных на семантических моделях данных. (Третья возможность еще не вышла за пределы исследовательских и экспериментальных проектов.)

Наиболее известным представителем класса семантических моделей предметной области является модель «сущность-связь» или *ER-модель*, предложенная Питером Ченом в 1976 году. Модель сущность-связь — модель данных, позволяющая описывать концептуальные схемы предметной области. Предметная область — часть реального мира, рассматриваемая в пределах данного контекста. Под контекстом здесь может пониматься, например, область исследования или область, которая является объектом некоторой деятельности. ER-модель используется при высокоуровневом (концептуальном) проектировании баз данных. С её помощью можно выделить ключевые сущности и обозначить связи, которые могут устанавливаться между этими сущностями. Во время проектирования баз данных происходит преобразование ER-модели в конкретную схему базы данных на основе выбранной модели данных (реляционной, объектной, сетевой или др.). ER-модель представляет собой формальную конструкцию, которая сама по себе не предписывает никаких графических средств её визуализации. В качестве стандартной графической нотации, с помощью которой можно визуализировать ERM, была предложена диаграмма сущность-связь (entity-relationship diagram, ERD). На практике понятия *ER-модель* и *ER-диаграмма* часто не различают, хотя для визуализации ER-моделей предложены и другие графические нотации. Основными понятиями ER-модели являются *сущность*, *связь* и *атрибут* (свойство).

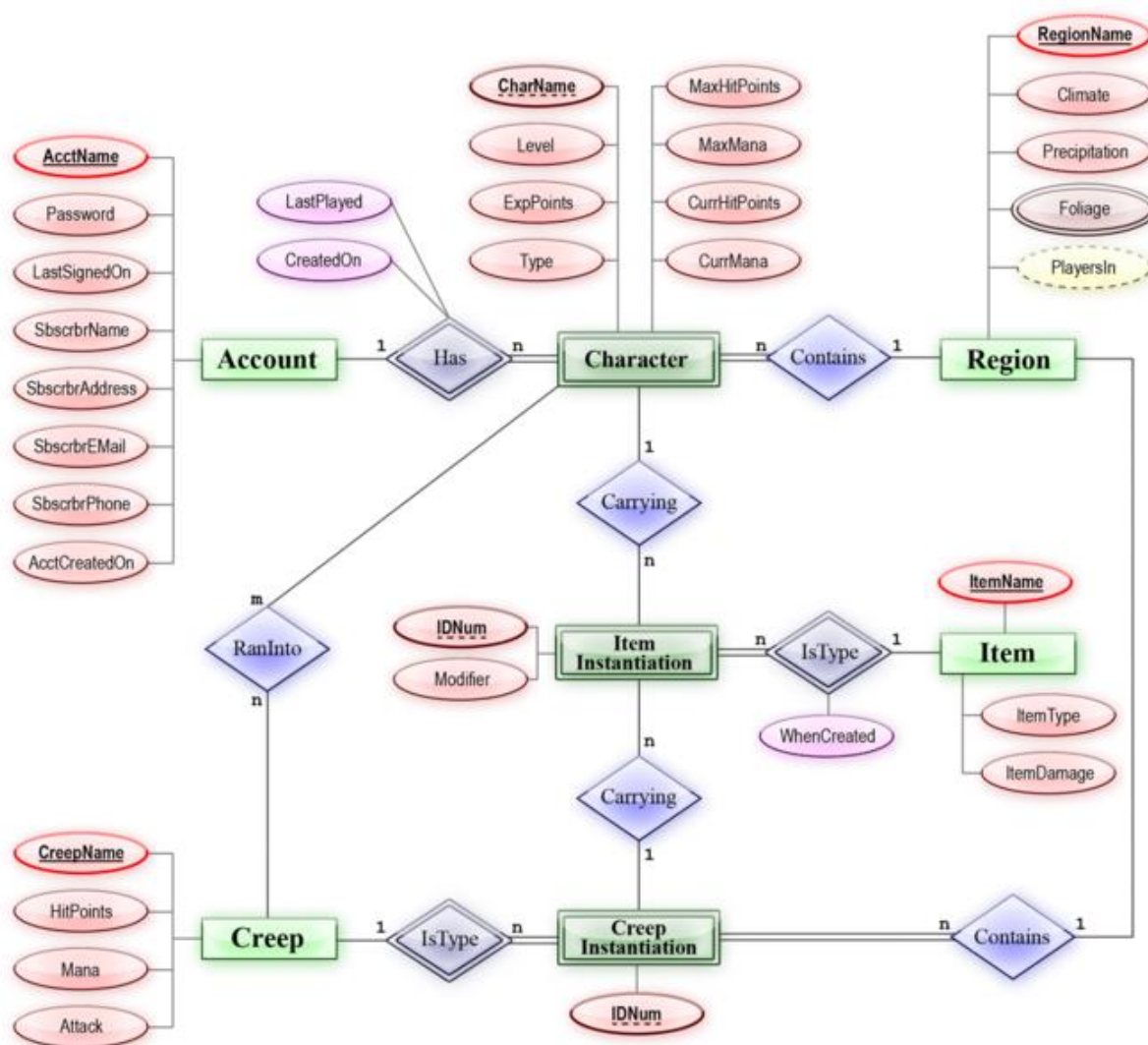
Сущность - это реальный или представляемый объект, информация о котором должна сохраняться и быть доступна. В диаграммах ER-модели сущность представляется в виде прямоугольника, содержащего имя сущности. При этом имя сущности - это имя типа, а не некоторого конкретного экземпляра этого типа. Для большей выразительности и лучшего понимания имя сущности может сопровождаться примерами конкретных объектов этого типа. Каждый экземпляр сущности должен быть отличим от любого другого экземпляра той же сущности (это требование в некотором роде аналогично требованию отсутствия кортежей-дубликатов в реляционных таблицах). Сущности подразделяются на *сильные* и *слабые*. Сильные сущности существуют сами по себе, а существование слабых сущностей зависит от существования сильных.

Связь - это ассоциация, устанавливаемая между сущностями. Эта ассоциация может существовать между разными сущностями или между сущностью и ей же самой (рекурсивная связь). Сущности, включенные в связь, называются ее *участниками*, а количество участников связи называется ее *степенью*. Участие сущности в связи может быть как *полным*, так и *частичным*. Связи в ER-модели могут иметь тип «один к одному», «один ко многим», «многие ко многим». Именно тип связи «многие ко многим» является единственным типом, представляющим истинную связь, поскольку это единственный тип связи, который требует для своего представления отдельного отношения. Связи типа «один к одному» и «один ко многим» всегда могут быть представлены с помощью механизма внешнего ключа, помещаемого в одно из отношений.

Свойством сущности (и связи) является любая деталь, которая служит для уточнения, идентификации, классификации, числовой характеристики или выражения состояния сущности (или связи). Значения свойств каждого типа извлекаются из соответствующего множества значений, которое в реляционной терминологии называется доменом. Свойства могут быть *простыми* или *составными*, *ключевыми*, *однозначными* или *многозначными*, *опущенными* (т. е. «неизвестными») или «непредставленными», *базовыми* или *производными*.

Более сложными элементами ER-модели являются подтипы и супертипы сущностей. Как в языках программирования с развитыми типовыми системами (например, в языках объектно-ориентированного программирования), вводится возможность наследования типа сущности, исходя из одного или нескольких супертипов.

На ER-диаграммах множества сущностей изображаются в виде прямоугольников, множества отношений изображаются в виде ромбов. Слабый тип сущности изображают в виде прямоугольника с двойным контуром. Слабый тип связи изображают в виде ромба с двойным контуром. Если сущность участвует в отношении, они связаны линией. Тип связи с частичным участием изображают двойной линией. Вид типа связи обозначается над линиями в виде соответствующих надписей возле типов сущностей. Например, если это вид бинарной связи «один ко многим», то делают надписи 1, n (или m), соответственно, возле соответствующих типов сущностей. Атрибуты изображаются в виде овалов и связываются линией с одним отношением или с одной сущностью. Именование сущности обычно выражается уникальным существительным, именование связи обычно выражается глаголом, именование атрибута обычно выражается существительным. Неизбыточный набор атрибутов, значения которых в совокупности являются уникальными для каждого экземпляра сущности, являются ключом сущности.



Существует множество инструментов для работы с ER-моделями, вот некоторые из них: Microsoft Visio, ERwin, Oracle Designer, PowerDesigner, Rational Rose. В справочниках приводятся сведения о 25 таких инструментах.

Получение реляционной схемы из ER-схемы осуществляется с помощью следующей пошаговой процедуры.

Шаг 1. Каждая простая сущность превращается в таблицу. Простая сущность - сущность, не являющаяся подтипом и не имеющая подтипов. Имя сущности становится именем таблицы.

Шаг 2. Каждое свойство (атрибут) становится возможным столбцом с тем же именем; может выбираться более точный формат. Столбцы, соответствующие необязательным атрибутам, могут содержать неопределенные значения; столбцы, соответствующие обязательным атрибутам, - не могут.

Шаг 3. Компоненты уникального идентификатора сущности превращаются в первичный ключ таблицы. Если имеется несколько возможных уникальных идентификаторов, выбирается наиболее используемый. Если в состав уникального идентификатора входят связи, к числу столбцов первичного ключа добавляется копия уникального идентификатора сущности, находящейся на дальнем конце связи (этот процесс может продолжаться рекурсивно). Для именования этих столбцов используются имена концов связей и/или имена сущностей.

Шаг 4. Связи «многие к одному» (и «один к одному») становятся внешними ключами. Т.е. делается копия уникального идентификатора с конца связи «один», и соответствующие столбцы составляют внешний ключ. Необязательные связи соответствуют столбцам, допускающим неопределенные значения; обязательные связи - столбцам, не допускающим неопределенные значения.

Шаг 5. Индексы создаются для первичного ключа (уникальный индекс), внешних ключей и тех атрибутов, на которых предполагается в основном базировать запросы.

Шаг 6. Если в концептуальной схеме присутствовали подтипы, то возможны два способа:

- a) все подтипы в одной таблице,
- b) для каждого подтипа - отдельная таблица.