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

- Либо вручную преобразуется к реляционной (или какой-либо другой) схеме.
- Либо реализуется автоматизированная компиляция концептуальной схемы в реляционную.
- 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 Ченомв 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).

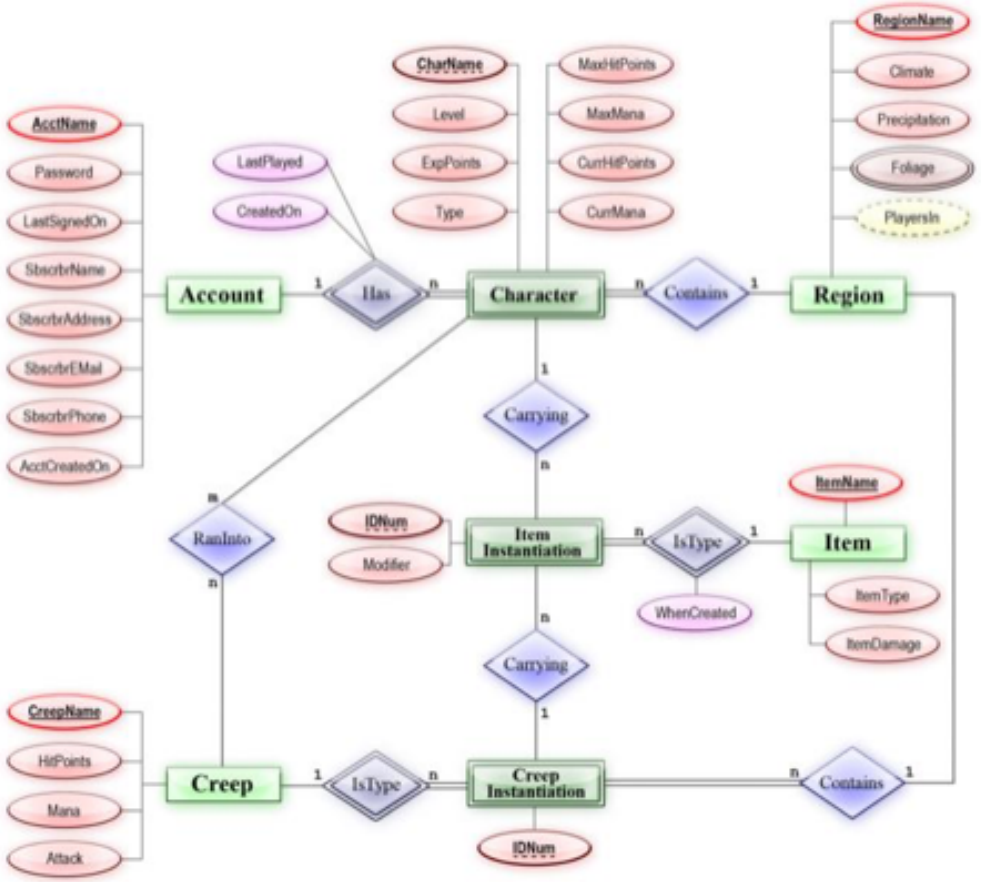
Сущность - это реальный или представляемый объект, информация о котором должна сохраняться и быть доступна. В диаграммах ER-модели сущность представляется в виде прямоугольника, содержащего имя сущности. При этом имя сущности - это имя типа, а не некоторого конкретного экземпляра этого типа. Для большей 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 единственным 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. Каждый 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:

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

Реляционная модель данных

Основоположником теории реляционных баз данных является британский учёный Эдгар Кодд, который в 1970 году опубликовал первую работу по реляционной модели данных. Наиболее распространенная трактовка реляционной модели данных принадлежит Кристоферу Дейту. Согласно Дейту, реляционная модель состоит из трех частей: структурной, целостностной и манипуляционной.

Структурная часть реляционной модели

Структурная часть реляционной модели описывает, из каких объектов состоит реляционная модель. Постулируется, что основной структурой данных, используемой в реляционной модели, являются нормализованные «n-арные» отношения. Основными понятиями структурной части реляционной модели являются тип данных, домен, атрибут, схема отношения, схема базы данных, кортеж, отношение, потенциальный, первичный и альтернативные ключи, реляционная база данных.

Понятие типа данных в реляционной модели полностью адекватно понятию типа данных в языках программирования.

Понятие домена можно считать уточнением типа данных. Домен можно рассматривать как подмножество значений некоторого типа данных, имеющих определенный смысл. Домен характеризуется следующими свойствами:

- домен имеет уникальное имя (в пределах базы данных),
- домен определен на некотором типе данных или на другом домене,
- домен может иметь некоторое логическое условие, позволяющее описать подмножество данных, допустимых для данного домена,
- домен несет определенную смысловую нагрузку.

Говорят, что домен отражает семантику, определенную предметной областью. Может быть несколько доменов, совпадающих как подмножества, но несущие различный смысл. Основное значение доменов состоит в том, что домены ограничивают сравнения. Некорректно, с логической точки зрения, сравнивать значения из различных доменов, даже если они имеют одинаковый тип.
Понятие домена помогает правильно моделировать предметную область. Не все домены обладают логическим условием, ограничивающим возможные значения домена. В таком случае множество возможных значений домена совпадает с множеством возможных значений типа данных.
Атрибут отношения – это пара вида <имя_атрибута, имя_домена >. Имена атрибутов должны быть уникальны в пределах отношения. Часто имена атрибутов отношения совпадают с именами соответствующих доменов.
Схема отношения – это именованное множество упорядоченных пар <имя_атрибута, имя_домена>. Степенью или «арностью» схемы отношения является мощность этого множества. Схема базы данных в реляционной модели – это множество именованных схем отношений. Понятие схемы отношения близко к понятию структурного типа в языках программирования (например, record в языке Pascal или struct в языке C).
Кортеж, соответствующий данной схеме отношения – это множество упорядоченных пар <имя_атрибута, значение_атрибута>, которое содержит одно вхождение каждого имени атрибута, принадлежащего схеме отношения. Значение атрибута должно быть допустимым значением домена, на котором определен данный атрибут. Степень или «арность» кортежа совпадает с «арностью» соответствующей схемы отношения.
Отношение, определенное на множестве из n доменов (не обязательно различных), содержит

две части: заголовок (схему отношения) и тело (множество из m кортежей). Значения n и m называются соответственно степенью и кардинальностью отношения. Отношения обладают следующими свойствами.

- В отношении нет одинаковых кортежей. Действительно, тело отношения есть множество кортежей и, как всякое множество, не может содержать неразличимые элементы.
- Кортежи не упорядочены (сверху вниз). Причина следующая – тело отношения есть множество, а множество не упорядочено.
- Атрибуты не упорядочены (слева направо). Т.к. каждый атрибут имеет уникальное имя в пределах отношения, то порядок атрибутов не имеет значения. В таблицах в отличие от отношений столбцы упорядочены.
- Каждый кортеж содержит ровно одно значение для каждого атрибута. Отношение, удовлетворяющее этому свойству, называется нормализованным или представленным в первой нормальной форме (1NF).
- Все значения атрибутов атомарны, т. е. не обладают структурой. Трактовка этого свойства в последнее время претерпела существенные изменения. Исторически в большинстве публикаций по базам данных считалось недопустимым использовать атрибуты со структурированными значениями. (А в большинстве изданий это считается таковым и поныне.) В настоящее время принимается следующая точка зрения: тип данных атрибута может быть произвольным, а, следовательно, возможно существование отношения с атрибутами, значениями которых также являются отношения. Именно так в некоторых постреляционных базах данных реализована работа со сколь угодно сложными типами данных, создаваемых пользователями.

В реляционной модели каждый кортеж любого отношения должен отличатся от любого другого кортежа этого отношения (т.е. любое отношение должно обладать уникальным ключом).

Непустое подмножество множества атрибутов схемы отношения будет потенциальным ключом тогда и только тогда, когда оно будет обладать свойствами уникальности (в отношении нет двух различных кортежей с одинаковыми значениями потенциального ключа) и неизбыточности (никакое из собственных подмножеств множества потенциального ключа не обладает свойством уникальности).

В реляционной модели по традиции один из потенциальных ключей должен быть выбран в качестве первичного ключа, а все остальные потенциальные ключи будут называться альтернативными.

Реляционная база данных – это набор отношений, имена которых совпадают с именами схем отношений в схеме базы данных.

Целостностная часть реляционной модели

В целостностной части реляционной модели фиксируются два базовых требования целостности, которые должны выполняться для любых отношений в любых реляционных базах данных. Это целостность сущностей и ссылочная целостность (или целостность внешних ключей).

Простой объект реального мира представляется в реляционной модели как кортеж некоторого отношения. Требование целостности сущностей заключается в следующем: каждый кортеж любого отношения должен отличатся от любого другого кортежа этого отношения (т.е. любое отношение должно обладать потенциальным ключом). Вполне очевидно, что если данное требование не соблюдается (т.е. кортежи в рамках одного отношения не уникальны), то в базе данных может храниться противоречивая информация об одном и том же объекте.

Поддержание целостности сущностей обеспечивается средствами СУБД. Это осуществляется с помощью двух ограничений:

- при добавлении записей в таблицу проверяется уникальность их первичных ключей,
- не позволяется изменение значений атрибутов, входящих в первичный ключ.

Сложные объекты реального мира представляются в реляционной модели данных в виде кортежей нескольких нормализованных отношений, связанных между собой. При этом

- связи между данными отношениями описываются в терминах функциональных зависимостей,
- для отражения функциональных зависимостей между кортежами разных отношений используется дублирование первичного ключа одного отношения (родительского) в другое (дочернее). Атрибуты, представляющие собой копии ключей родительских отношений, называются внешними ключами.

Внешний ключ в отношении R2 – это непустое подмножество множества атрибутов FK этого отношения, такое, что: a) существует отношение R1 (причем отношения R1 и R2 необязательно различны) с потенциальным ключом CK; b) каждое значение внешнего ключа FK в текущем значении отношения R2 обязательно совпадает со значением ключа CK некоторого кортежа в текущем значении отношения R1.

Требование ссылочной целостности состоит в следующем: Для каждого значения внешнего ключа, появляющегося в дочернем отношении, в родительском отношении должен найтись кортеж с таким же значением первичного ключа.

Как правило, поддержание ссылочной целостности также возлагается на СУБД. Например, она может не позволить пользователю добавить запись, содержащую внешний ключ с несуществующим (неопределенным) значением.

Манипуляционная часть реляционной модели

Манипуляционная часть реляционной модели описывает два эквивалентных способа манипулирования реляционными данными – реляционную алгебру и реляционное исчисление. Принципиальное различие между реляционной алгеброй и реляционным исчислением заключается в следующем: реляционная алгебра в явном виде предоставляет набор операций, а реляционное исчисление представляет систему обозначений для определения требуемого отношения в терминах данных отношений. Формулировка запроса в терминах реляционной алгебры носит предписывающий характер, а в терминах реляционного исчисления – описательный характер. Говоря неформально, реляционная алгебра носит процедурный характер (пусть на очень высоком уровне), а реляционное исчисление – непроцедурный характер.