

Лекция № 16.

Семантические модели данных

Семантические модели данных

1. Характеристика подхода

Широкое распространение реляционных СУБД и их использование в самых разнообразных приложениях показывает, что реляционная модель данных (далее – РМД) достаточна для моделирования многих предметных областей (ПО).

Однако РМД имеет и свои недостатки, что вызвало появление новых моделей данных.

Семантические модели данных

Главными недостатками РМД являются:

- 1) в РМД нет достаточных средств для представления смысла (семантики) данных. Семантика реальной ПО независимым от РМД способом представляется в голове (или рабочих материалах) проектировщика;
- 2) в РМД нет средств для представления зависимостей между данными (отношениями);
- 3) для многих приложений недостаточно представления данных в виде плоских таблиц;
- 4) несмотря на то что процесс проектирования РБД начинается с выделения некоторых существенных для приложения объектов ПО (сущностей) и выявления связей между этими сущностями, РМД не предлагает никакого аппарата для разделения сущностей и связей;

Семантические модели данных

Указанные недостатки РМД и привели к созданию семантических моделей данных (далее – СМД).

СМД, как и РМД, имеют три части – структурную, манипуляционную и целостную.

На практике СМД используются на первой стадии проектирования БД. При этом в терминах СМД описывается концептуальная (понятийная) схема БД, которая затем вручную преобразуется к реляционной или другой схеме. Этот процесс описан соответствующими методиками.

Иногда выполняется автоматическая компиляция СМД в реляционную схему. При этом производится реляционная схема БД в третьей нормальной форме.

Семантические модели данных

В последнее время ведутся исследования и проводятся эксперименты по созданию СУБД, основанных на семантической модели данных. При этом рассматриваются два варианта:

- 1) обеспечение пользовательского интерфейса на основе СМД с автоматическим отображением конструкций в РМД;
- 2) прямая реализация СУБД, основанной на какой-либо семантической модели данных.

В популярной СУБД Microsoft Access используются первый из этих вариантов. Наиболее близки ко 2-му варианту современные объектно-ориентированные СУБД, модели данных которых по многим параметрам подобны СМД.

Семантические модели данных

2. Модель "сущность – связи"

Одной из наиболее популярных СМД является модель "сущность – связи" (entity – relationship) или ER-модель, предложенная Ченом (Chen) в 1976 г. Она основана на графических диаграммах, включающих небольшое число разнородных компонентов.

ER-модели получили распространение в CASE-системах, поддерживающих проектирование РБД.

Мы будем рассматривать ER-модель из CASE-системы фирмы ORACLE.

Модель "сущность – связи"

2.1. Основные понятия ER-модели

Основными понятиями ER-модели являются *сущность*, *связь* и *атрибут*.

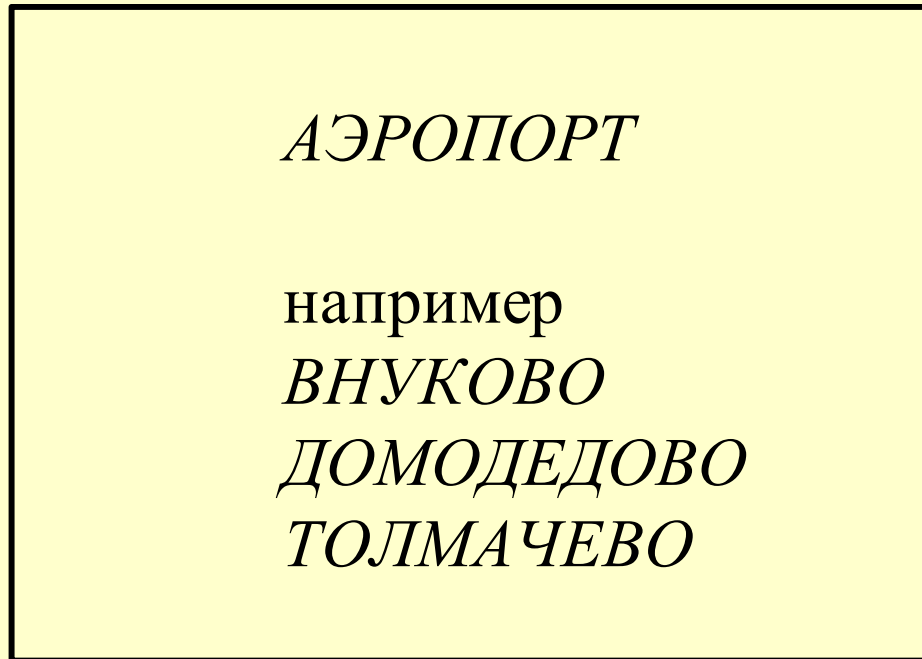
Сущность – это реальный или абстрактный объект, информация о котором хранится в базе данных.

В диаграммах ER-модели сущность задается в виде прямоугольника с именем сущности. Имя сущности – это тип объекта, а не конкретный его экземпляр.

Для наглядности в диаграммах имя сущности часто сопровождается примерами. Например, диаграмма сущности АЭРОПОРТ выглядит следующим образом:

Модель "сущность – связи"

Пример диаграммы



Каждый экземпляр сущности должен быть отличим от другого экземпляра этой же сущности. (Заметим, что это требование аналогично требованию отсутствия кортежей-дубликатов в отношении РБД.)

Модель "сущность – связи"

Связь – это графически изображаемая ассоциация в виде линии между двумя сущностями. Связь всегда бинарная, т.е. связывает две разных сущности или сущность саму с собой. В последнем случае связь называется *рекурсивной*.

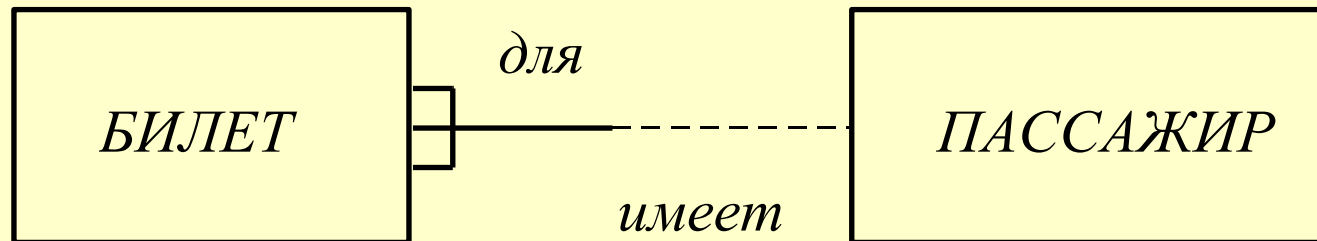
Связь имеет два конца. На каждом конце связи указывается его *имя*, *степень* (сколько экземпляров данной сущности связывается) и *обязательность* связи.

В месте стыковки связи с сущностью используется *трехточечный вход* в прямоугольник сущности, если в связи могут участвовать много (несколько) экземпляров сущности, или *одноточечный*, если один экземпляр.

Обязательный конец связи изображается сплошной линией, а *необязательный* – прерывистой.

Модель "сущность – связи"

Пример ER-схемы



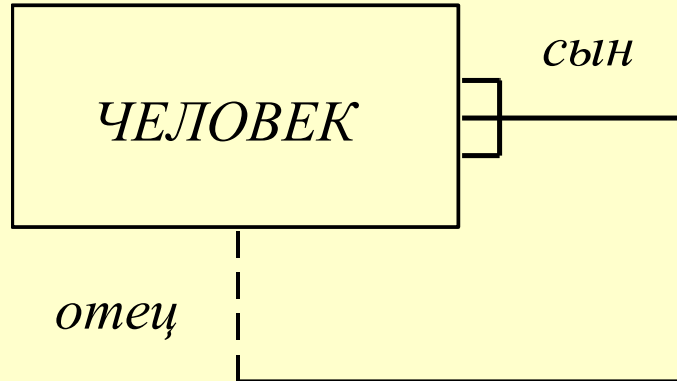
На приведенной схеме конец связи с именем *для* позволяет связывать с одним пассажиром несколько билетов, причем каждый билет должен быть обязательно связан с каким-либо пассажиром. Конец связи с именем *имеет* означает, что каждый билет принадлежит только одному пассажиру, причем пассажир может не иметь билета.

Лаконичная трактовка диаграммы:

- каждый билет предназначен для одного и только одного пассажира,
- каждый пассажир может иметь ноль или более билетов.

Модель "сущность – связи"

Рекурсивная связь



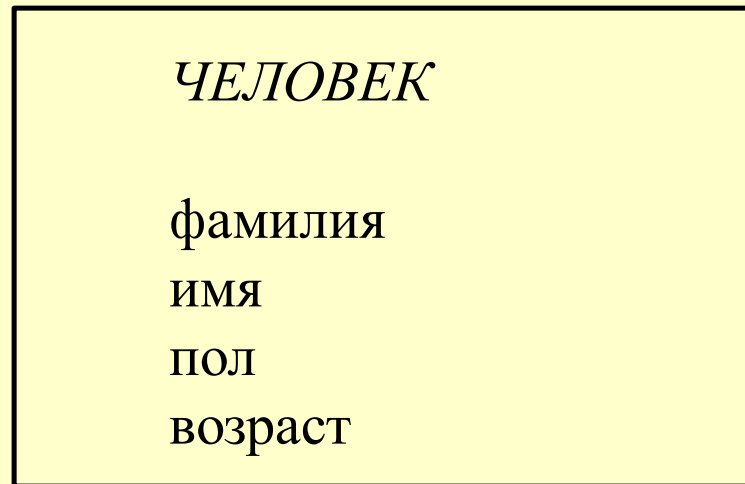
В ER-схеме, включающей рекурсивную связь, конец связи с именем *сын* означает: у одного отца может быть более чем один сын. Смысл конца связи *отец*: не у каждого человека могут быть сыновья.

Краткая трактовка этой схемы:

- каждый человек является сыном одного и только одного человека,
- каждый человек может являться отцом для нуля и более людей.

Модель "сущность – связи"

Атрибутом сущности является любая деталь, которая служит для уточнения, классификации, идентификации, числовой характеристики или выражения состояния сущности. Имена атрибутов пишутся под именем сущности строчными буквами:



Уникальным идентификатором сущности является атрибут, комбинация атрибутов, комбинация связей или комбинация атрибутов и связей, однозначно отличающая любой экземпляр сущности от других экземпляров сущности того же типа.

Модель "сущность – связи"

2.2. Расширение ER-модели

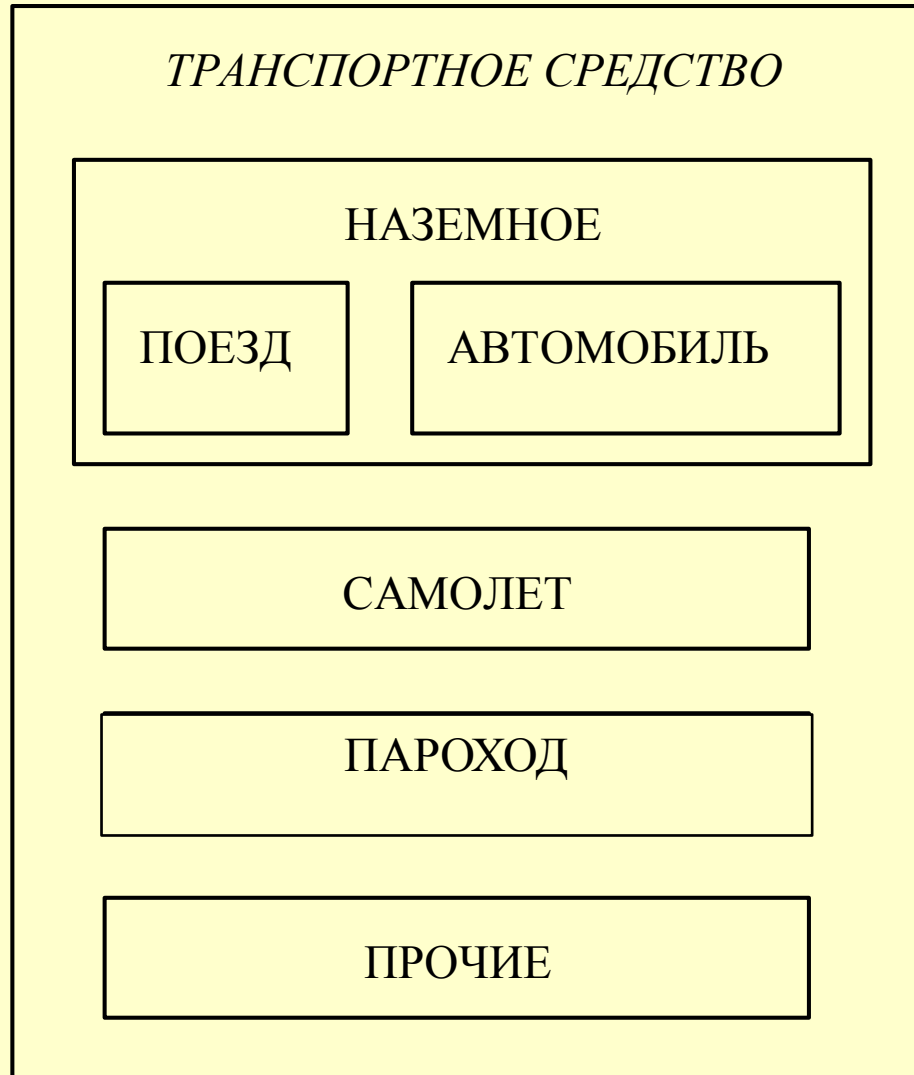
В ER-модель могут входить следующие более сложные элементы и средства:

- 1) подтипы и супертипы сущностей;
- 2) связи типа "многие-ко-многим" ("many-to-many");
- 3) уточняемые степени связи;
- 4) каскадное удаление экземпляров сущности;
- 5) возможность определения доменов.

В расширенной ER-модели сущности могут строиться с использованием иерархии. Сущность, на основе которой определяются подтипы, называется *супертипом*. *Подтипы* наследуют свойства (атрибуты) супертипа. Подтипы должны образовывать полное множество, т.е. любой экземпляр супертипа должен относиться к некоторому подтипу. Иногда определяют дополнительный подтип ПРОЧИЕ.

Модель "сущность – связи"

Супертип ТРАНСПОРТНОЕ СРЕДСТВО



Модель "сущность – связи"

Часто удобно иметь два и более разбиений сущности на подтипы, например, сущность ЧЕЛОВЕК может быть разбита на подтипы по профессиональному признаку (ПРОГРАММИСТ, ШАХТЕР) или по половому или возрастному признаку.

В расширенной ER-модели вводится также возможность уточнения *количества экземпляров сущности*, участвующих в данной связи, например, программисту разрешается участвовать не более чем в трех проектах. Это семантическое ограничение задается указанием на конце связи ее *максимальной* или *обязательной степени* (в виде целого числа).

При определении сущности может формулироваться каскадное удаление экземпляров сущности. Некоторые связи типа "*один-ко-многим*" могут быть настолько сильными, что при удалении опорного элемента, соответствующего концу связи "*один*", необходимо удалить и все сущности на конце связи "*многие*".

Модель "сущность – связи"

2.3. Нормальные формы ER-схем

В ER-схемах, как и в реляционных схемах, вводится понятие нормальных форм, близкое по смыслу к тому же понятию в РМД.

Приведем неформальные определения трех нормальных форм в ER-схемах.

В НФ-1 устраняются повторяющиеся атрибуты или группы атрибутов, т.е. выявляются неявные сущности, замаскированные под атрибуты.

В НФ-2 устраняются атрибуты, зависящие от части уникального идентификатора. Эта часть уникального идентификатора определяет отдельную сущность.

В НФ-3 устраняются атрибуты, зависящие от атрибутов, не входящих в уникальный идентификатор. Эти атрибуты являются основой отдельной сущности.

Модель "сущность – связи"

2.4. Получение реляционной схемы из ER-схемы

Приведем методику получения из ER-схемы реляционной схемы.

Процесс получения требуемой схемы состоит из следующих шагов:

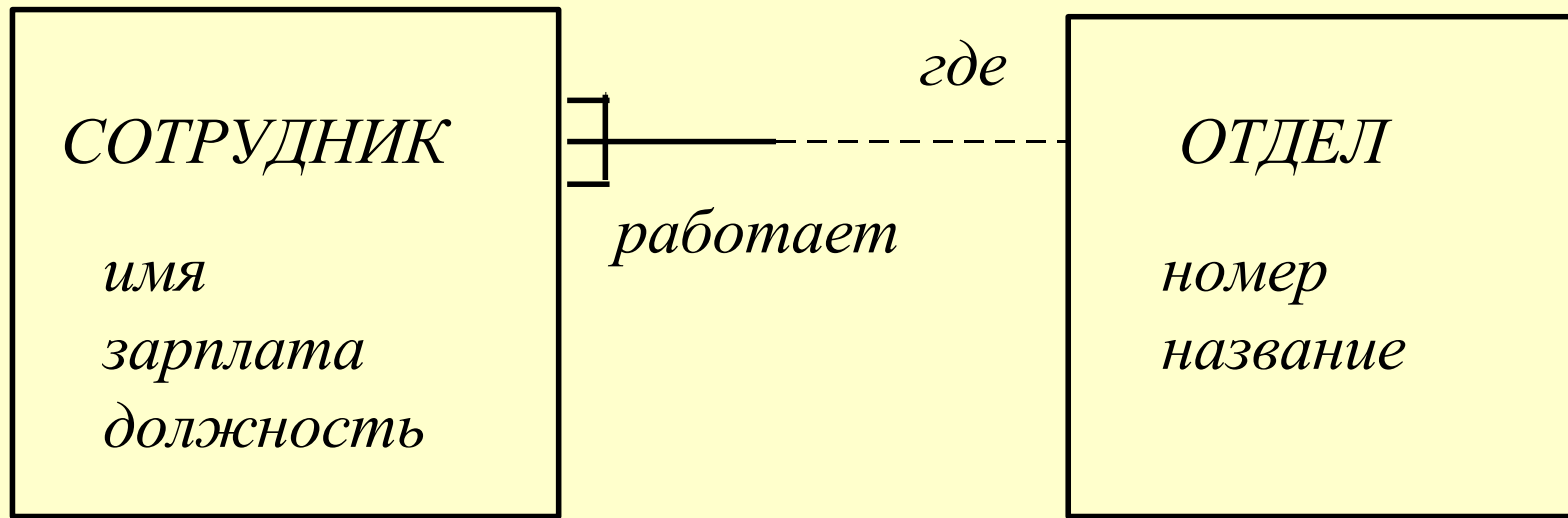
- 1) Каждая простая сущность превращается в таблицу.
(Простой называется сущность, не являющаяся подтипом и не имеющая подтипов.) При этом имя сущности становится именем таблицы.
- 2) Каждый атрибут сущности становится возможным столбцом таблицы с тем же именем; при этом может выбираться более точный формат значений атрибута. Столбцы, соответствующие необязательным атрибутам, могут содержать неопределенные значения, обязательным – не могут.

Модель "сущность – связи"

- 3) Компоненты уникального идентификатора сущности превращаются в первичный ключ таблицы. 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.
- 4) Связи "многие-к-одному" и "один-к-одному" становятся внешними ключами, т.е. делается копия уникального идентификатора с конца связи "один" и соответствующие ему столбцы составляют внешний ключ. При этом необязательные связи соответствуют столбцам, допускающим неопределенные значения, обязательные - столбцам, не допускающим неопределенные значения.

Модель "сущность – связи"

Пример связи "многие-к-одному"



Модель "сущность – связи"

5) Если в ER-схеме присутствовали подтипы, то имеется два способа их перевода:

- а) все подтипы помещаются в одной таблице,
- б) для каждого подтипа создается своя таблица.

При применении способа (а) таблица создается для наиболее внешнего супертипа. В таблицу добавляется столбец, содержащий код ТИПА; он становится частью первичного ключа.

При применении способа (б) для каждого подтипа первого уровня создается своя таблица; супертип воссоздается путем выбора из всех таблиц подтипов общих столбцов.

Оба способа имеют свои достоинства и недостатки (они различаются числом таблиц, обязательностью столбцов, логикой работы).

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