

ГЛАВА 3. МОДЕЛИ ДАННЫХ СУБД КАК ИНСТРУМЕНТ ПРЕДСТАВЛЕНИЯ КОНЦЕПТУАЛЬНОЙ МОДЕЛИ

В соответствии с основными этапами проектирования базы данных после построения концептуальной модели выбирается система управления базой данных, с помощью которой будет организована база данных и работа с ней. Каждая СУБД поддерживает определенные виды и типы данных, а также средства представления связей между данными, составляющими модель данных СУБД. Следующий этап проектирования базы данных состоит в представлении концептуальной модели средствами модели СУБД или в отображении концептуальной модели в модель данных СУБД. Этот этап часто называют логическим проектированием базы данных. Полученная при этом модель часто также называется концептуальной моделью или схемой (но специфицированной к понятиям модели данных СУБД). В некоторых источниках полученную модель называют логической структурой данных.

3.1. Общие представления о модели данных

Можно по-разному характеризовать понятие модели данных. С одной стороны, модель данных – это способ структурирования данных, которые рассматриваются как некоторая абстракция в отрыве от предметной области. С другой стороны, модель данных – это инструмент представления концептуальной модели предметной области и динамики ее изменения в виде базы данных.

Учитывая обе вышеуказанные стороны, определим основные структуры моделей данных, используемые для представления концептуальной модели предметной области (сущностей, атрибутов, связей).

Элемент данных (поле) – наименьшая поименованная единица данных. Используется для представления значения атрибута.

Запись – поименованная совокупность полей. Используется для представления совокупности атрибутов сущности (записи о сущности).

Экземпляр записи – запись с конкретными значениями полей.

Агрегат данных – поименованная совокупность элементов данных внутри записи, которую можно рассматривать как единое целое.

Место рождения		
Область	Район	Населенный пункт

Файл – поименованная совокупность экземпляров записей одного типа. Используется для представления однородного набора сущностей.

Набор файлов – поименованная совокупность файлов, обрабатываемых в системе. Используется для представления нескольких наборов сущностей.

Введем понятие «группа», обобщающее понятия «агрегат» и «запись».

Группа – это поименованная совокупность элементов данных или элементов данных и других групп.

Важнейшим понятием концептуальной модели является понятие связи между сущностями (наборами сущностей). В моделях данных соответствующее понятие отражается понятием «групповое отношение».

Групповое отношение – поименованное бинарное отношение, заданное на двух множествах экземпляров рассматриваемых групп. По характеру бинарных связей различают групповые отношения вида 1:1, 1:М, М:1, М:М. Пары чисел называют коэффициентами группового отношения. В групповом отношении один член группы назначается владельцем отношения, другой – членом.

База данных – поименованная совокупность экземпляров групп и групповых отношений.

Для представления группового отношения используется две формы:

а) *Графовая*. Группы изображаются вершинами графа, связи между группами – дугами, направленными от группы-владельца к групп-члену с указанием имени отношения и коэффициента.

По типу графов различают:

- иерархическую модель (граф без циклов – дерево);
- сетевую модель (ориентированный граф общего вида).

б) *Табличная*. Связь между группами изображается таблицей, столбцы которой представляют ключи соответствующих групп. Для формального описания таблицы используется математическое (теоретико-множественное) понятие отношения. Соответствующая модель данных называется реляционной моделью.

Модель данных описывается следующим образом:

- определяются типы и характеристики логических структур данных (полей, записей, файлов);
- описываются правила составления структур более общего типа из структур более простых типов;
- описываются возможные действия над структурами и правила их выполнения, включающие:
 - основные элементарные операции над данными;
 - обобщенные операции (процедуры);
 - средства контроля относительно простых условий корректности ввода данных (ограничения);
 - средства контроля сколь угодно сложных условий корректности выполнения определенных действий (правила).

В качестве основных элементарных операций обычно рассматриваются следующие: поиск записи с заданным значением ключа, чтение нужной записи, добавление записи, корректировка, удаление.

В моделях данных также предусматриваются специальные операции для установления групповых отношений.

Обобщенные операции или процедуры – последовательность операций, реализующая определенный алгоритм обработки данных. Процедуры могут инициироваться СУБД автоматически, а также могут запускаться пользователем. Примерами процедур являются процедуры копирования БД, восстановления БД, процедуры, вычисляющие значения определенных атрибутов в БД по значениям других атрибутов, и т.п.

Средства контроля используются для реализации ограничений целостности концептуальной модели.

Простейшие средства контроля – ограничения – используются для реализации как внешних ограничений концептуальной модели, так и внутренних ограничений модели данных. В качестве последних ограничений, в частности, реализованы ограничения на ввод данных несоответствующего типа, несоответствующей характеристики (по числу битов, по числу полей, по количеству записей и т.п.). Более сложные средства контроля (правила) позволяют вызывать выполнение определенной последовательности операций (сколь угодно сложной) при изменении или добавлении данных в БД и тем самым реализовывать ограничения целостности, описанные с помощью специальных конструкций.

3.2. Сетевая модель данных

Это одна из наиболее ранних моделей данных. Типовая сетевая модель данных была предложена рабочей группой по базам данных (Data Base Task Group – DBTG) системного комитета CODASYL (Conference of Data System Languages), основными функциями которого были анализ известных фирменных систем обработки управленческих данных с единых позиций и в единой терминологии, обобщение опыта организации таких систем и разработка рекомендаций по созданию соответствующих систем. Результаты работы комитета CODASYL изложены в [12]. Структура данных сетевой модели определяется в терминах раздела 3.1 (элемент, агрегат, запись, группа, групповое отношение, файл, база данных).

Реализация групповых отношений в сетевой модели осуществляется с использованием указателей (адресов связи или ссылок), которые устанавливают связь между владельцем и членом группового отношения. Запись может состоять в отношениях разных типов (1:1, 1: N , M : N). Заметим, что если один из вариантов установления связи 1:1 очевиден (в запись – владелец отношения, поля которой соответствуют атрибутам сущности, включается дополнительное поле – указатель на запись – член отношения), то возможность представления связей 1: N и M : N таким же образом весьма проблематична. Поэтому наиболее распространенным способом организации связей в сетевых СУБД является введение дополнительного типа записей, полями которых являются указатели.

Рассмотрим для примера представление группового отношения M : N . В модель вводится дополнительная группа (дополнительный вид записей). Элементы этой записи представляют собой указатели на две исходные группы и указатели на экземпляры рассматриваемой дополнительной записи, связывающие их в список (цепь), соответствующий M и (или) N членам группового отношения (рис. 30).

Представление связей 1:1, 1: M , N :1 является частным случаем связи типа M : N и осуществляется аналогично рассмотренному выше.

Заметим, что группа может быть членом более чем одного группового отношения. В этом случае вводится несколько дополнительных групп-указателей, а в группе – владельце отношений вводится несколько полей – указателей на дополнительные группы.

Тогда множество записей (групп) и связей между ними образует некую сетевую структуру (ориентированный граф общего вида). Вер-

шинами графа являются группы; дугами графа, направленными от владельца к члену группового отношения, – связи между группами.

Сетевая модель данных поддерживает все необходимые операции над данными, реализованные как действия со списками структурами.

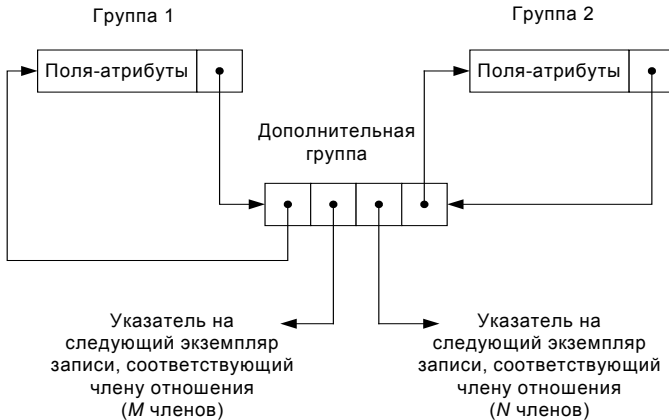


Рис. 30. Представление связей типа $M:N$

Сетевая модель данных является, вероятно, наиболее общей по возможностям представления концептуальной модели. По сути, любая ER-диаграмма без каких-либо изменений представляется средствами сетевой модели. К недостаткам сетевой модели обычно относят сложность получаемой на её основе концептуальной схемы и большую трудоемкость понимания соответствующей схемы внешним пользователем.

Наиболее существенным недостатком, на наш взгляд, является «жесткость» получаемой концептуальной схемы. Связи закреплены в записях в виде указателей. При появлении новых аспектов использования этих же данных может возникнуть необходимость установления новых связей между ними. Это требует введения в записи новых указателей, т.е. изменения структуры БД, и, соответственно, реформирования всей базы данных.

СУБД, поддерживающие сетевую модель, широко использовались на вычислительных системах серии IBM 360/370 (ЕС ЭВМ). В качестве примеров таких систем можно указать IDMS, UNIBAD (БАНК), аналоги СЕДАН, СЕТОР. На персональных компьютерах сетевые СУБД не получили широкого распространения. Примером сетевой

СУБД для персонального компьютера является db_VISTA III. Отметим, что система db_VISTA реализована на языке С и поэтому является переносимой. Система может эксплуатироваться на ПЭВМ типа IBM PC, SUN, Macintosh.

3.3. Иерархическая модель данных

Это также одна из наиболее ранних моделей данных. Реализация групповых отношений в иерархической модели, как и в сетевой, может осуществляться с помощью указателей и представляется в виде графа. Однако, в отличие от сетевой модели, здесь существует ряд принципиальных особенностей.

1. Групповые отношения являются отношениями соподчиненности. Группа (запись) – владелец отношения имеет подчиненные группы – члены отношений. Исходная группа называется предком, подчиненная – потомком.
2. Групповые отношения образуют иерархическую структуру, которую можно описать как ориентированный граф следующего вида:
 - имеется единственная особая вершина (соответствующая группе), называемая корнем, в которую не заходит ни одно ребро (группа не имеет предков);
 - во все остальные вершины входит только одно ребро (все остальные группы имеют одного предка), а исходит произвольное количество ребер (группы имеют произвольное количество потомков);
 - отсутствуют циклы.
3. Иерархическая модель данных может представлять совокупность нескольких деревьев. В терминологии иерархической модели дерева, описывающие структуру данных, называются деревьями описания данных, а сами структурированные данные (база данных) – деревьями данных.

Особенностью реализации операций поиска в иерархической модели является то, что операция всегда начинает поиск с корневой вершины и специфицирует иерархический путь (последовательность связанных вершин) от корня до вершины, экземпляры которой удовлетворяют условиям поиска.

Необходимо отметить, что программы, реализующие операции иерархической модели, существенно проще, чем аналогичные программы для сетевой модели, т.к. здесь много легче осуществлять навигацию по структуре.

Целесообразность появления иерархической модели обусловлена, конечно, тем, что большинство организационных систем реального мира имеют иерархическую структуру (административное деление страны, организационная структура предприятия и т.п.). Соответствующее концептуальное представление также будет иметь иерархическую структуру и естественным образом может быть описано в терминах иерархической модели.

В качестве недостатков иерархической модели можно назвать вышеуказанные недостатки сетевой.

СУБД, поддерживающие иерархическую модель, достаточно широко использовались на вычислительных системах IBM 360/370 (ЕС ЭВМ). В качестве примеров таких систем можно указать IMS, ОКА и широко тиражируемую в СССР отечественную разработку ИНЕС. Примером иерархической СУБД для персональных ЭВМ является отечественная система НИКА (адаптация системы ИНЕС к IBM PC).

3.4. Реляционная модель данных

Учитывая отмеченные в предыдущих разделах недостатки сетевых и иерархических моделей, можно сформулировать желательные требования к модели данных:

- модель должна быть понятна пользователю, не имеющему особых навыков в программировании;
- появление новых аспектов использования данных и необходимость введения новых связей не должны приводить к реструктуризации всей модели данных и базы данных в целом.

Моделью данных, удовлетворяющей вышеуказанным требованиям, является реляционная модель, часто называемая также табличной.

Основными используемыми понятиями здесь также являются поле, запись и файл. Структура записи определяет структуру таблицы, содержащей экземпляры соответствующей записи. Столбцы таблицы представляют собой имена полей записи, строки таблицы – экземпляры записи. Таким образом, понятие «таблица» здесь соответствует понятию «файл» модели данных.

Групповое отношение может представляться двумя способами. При первом способе в таблицы, соответствующие группам – членам отношения, добавляются столбцы ключевых полей (атрибутов) другого члена отношения (связь описывается через ключевые атрибуты). При втором способе групповое отношение определяется как дополнительная группа (дополнительная таблица). Столбцами этой дополнительной таблицы являются ключи групп – членов отношения. Таким

образом, при любом способе соответствующая модель данных представляет собой совокупность структур таблиц.

Для формального описания таблицы используется теоретико-множественное понятие отношения. Список названий столбцов таблицы (имен полей записи, соответствующих атрибутам) именуют схемой отношения и обозначают $R(A_1, A_2, \dots, A_n)$.

Совокупность схем отношений, используемых для представления концептуальной модели, называется схемой реляционной базы данных, а текущие значения соответствующих отношений – реляционной базой данных.

В качестве основного недостатка реляционной модели можно указать дублирование информации при представлении связей.

Необходимо отметить, что большинство СУБД для персональных ЭВМ поддерживают именно реляционную модель данных. В качестве примеров таких наиболее распространенных СУБД можно указать все dBase-подобные системы, DB2, Paradox, Access, FoxPro, Oracle, MS SQL Server. Более подробно реляционная модель данных будет рассмотрена далее.

3.5. Многомерная модель данных

Вернемся к понятию «сущность» концептуальной модели.

Сущность – это то, о чем накапливается информация в информационной системе. Часто оказывается, что информация об определенной сущности зависит еще от ряда параметров. Рассмотрим, например, сущность ЧИСЛЕННОСТЬ НАСЕЛЕНИЯ.

ЧИСЛЕННОСТЬ НАСЕЛЕНИЯ
Число мужчин
Число женщин
Число лиц обоих полов

Значение атрибутов зависит от параметров «год», «административный район». Если использовать для описания соответствующей концептуальной схемы реляционную модель, то необходимо вводить множество таблиц ЧИСЛЕННОСТЬ НАСЕЛЕНИЯ по каждому году для каждого района. Так, при 60 административных районах и необходимости анализировать данные за 10 лет число таблиц будет равно 600. Дублируются аналогичные структуры всех таблиц, достаточно

сложна обработка данных, связанная с анализом однотипных данных при изменении значения одного из параметров и т.д.

Наиболее подходящей моделью данных для этого случая является так называемая многомерная модель, используемая в технологии OLAP (OnLine Analytical Processing – оперативная аналитическая обработка). Отметим, что многомерность модели данных означает здесь многомерное логическое представление структуры информации и, вообще говоря, не связана с многомерностью визуализации.

Многомерные структуры представляются как гиперкубы данных. Каждая грань куба является размерностью. Основными понятиями, используемыми в многомерных моделях данных, являются «измерение» (dimension) и «ячейка» (cell).

Измерение – упорядоченный набор значений, принимаемых конкретным параметром, соответствующий одной из граней гиперкуба. Для нашего примера можно указать в качестве измерений: год – 1998, 1999, 2000, 2001, 2002, 2003; область – Московская, Ленинградская, Нижегородская и т.д.

Ячейка или показатель – это поле, соответствующее атрибуту сущности, значение которого однозначно определяется фиксированным набором значений параметров (значениями «измерений», например, 2003 г., Нижегородская область).

В многомерной модели данных определяется ряд дополнительных операций, среди которых можно выделить операции «формирование среза» и «агрегация».

При формировании среза пользователю по его запросу предоставляется некоторое подмножество гиперкуба, полученное в результате фиксации пользователем одного или нескольких значений параметров. Операция «агрегация» обеспечивает переход к более общему представлению информации из гиперкуба пользователю, например суммируя значения показателей по всем значениям одного из параметров, допустим, по всем областям.

Такая модель позволяет легко сравнивать данные при разных значениях параметров, строить графики зависимости значений конкретных атрибутов от значений определенных параметров (например, изменение атрибута по годам) и т.п. Поэтому основное назначение технологии OLAP – обработка информации для проведения анализа и принятия решения.

Массовое использование СУБД, поддерживающих многомерную модель данных, только начинается. В качестве наиболее известных СУБД такого типа можно указать Oracle Express Server.