



Введение

Развитие средств вычислительной техники обеспечило для сотворения и широкого использования систем обработки данных разнообразного назначения.

Разрабатываются информационные системы для обслуживания разных систем деятельности, всевозможные тренажеры и обучающие системы. Одной из принципиальных предпосылок сотворения таких систем стала возможность оснащения их "памятью" для скопления, хранения и систематизация огромных размеров данных. Другой значимой предпосылкой необходимо признать разработку подходов, а также создание программных и технических средств конструирования систем, предназначенных для коллективного использования. В данной связи потребовалось создать особые способы и механизмы управления такого рода вместе используемыми ресурсами данных, которые стали называться базами данных. Исследования и разработки, связанные с проектированием, созданием и эксплуатации баз данных, а также нужных для этих целей языковых и программных инструментальных средств, привели к появлению самостоятельной ветки информатики, получившей заглавие системы управления данными.

В настоящее время разработаны и употребляются на персональных компьютерах около двадцати систем управления базами данных.

Начиная с 60-х годов для работы с данными, стали употреблять особенные программные комплексы, называемые системами управления базами данных (СУБД). Системы управления базами данных отвечают за:

физическое размещение данных и их описаний;

поиск данных;

поддержание баз данных в актуальном состоянии;

защиту данных от некорректных обновлений и несанкционированного доступа;

Модели данных

В зависимости от способа организации (модели) данных в базах данных (БД) их разделяют на а) иерархические, б) сетевые модели, в) реляционные модели и г) объектно-ориентированные модели СУБД. Однако в большинстве учебных материалов обычно различают три класса СУБД, обеспечивающих работу:

- 1) иерархических
- 2) сетевых
- 3) реляционных моделей.

"Иерархические структуры" наиболее подробно описывают положительные и отрицательные черты иерархической модели. Воспринимая окружающий мир, наше сознание в процессе мышления манипулирует иерархическими понятиями. При работе с иерархиями используется "семейная" терминология (родители, внуки, предки, потомки), поскольку семья является самым распространённым примером объектов (в данном случае - людей), объединённых иерархическими отношениями. ("родитель" может иметь множество "детей", но не наоборот), организована в виде иерархического дерева.

В то же время, место объекта в иерархическом дереве - не более чем условное обозначение связи с другими объектами. Иерархическая структура всего лишь помогает сохранить, упорядочить и найти объект.

Структура данных

Организация данных в СУБД иерархического типа определяется в терминах: элемент, агрегат, запись (группа), групповое отношение, база данных.

Атрибут (элемент данных) - наименьшая единица структуры данных. Обычно каждому элементу при описании базы данных присваивается уникальное имя. По этому имени к нему обращаются при обработке. Элемент данных также часто называют полем.

Запись - именованная совокупность атрибутов. Использование записей позволяет за одно обращение к базе получить некоторую логически связанную совокупность данных. Именно записи изменяются, добавляются и удаляются. Тип записи определяется составом ее атрибутов. Экземпляр записи - конкретная запись с конкретным значением элементов

Групповое отношение - иерархическое отношение между записями двух типов. Родительская запись (владелец группового отношения) называется исходной записью, а дочерние записи (члены группового отношения) - подчиненными. Иерархическая база данных может хранить только такие древовидные структуры.

Корневая запись каждого дерева обязательно должна содержать ключ с уникальным значением. Ключи некорневых записей должны иметь уникальное значение только в рамках группового отношения. Каждая запись идентифицируется полным сцепленным ключом, под которым понимается совокупность ключей всех записей от корневой по иерархическому пути.

При графическом изображении групповые отношения изображают дугами ориентированного графа, а типы записей - вершинами (диаграмма Бахмана).

Для групповых отношений в иерархической модели обеспечивается автоматический режим включения и фиксированное членство. Это означает, что для запоминания любой некорневой записи в БД должна существовать ее родительская запись (подробнее о режимах включения и исключения записей сказано в параграфе о сетевой модели). При удалении родительской записи автоматически удаляются все подчиненные.

Операции над данными

ДОБАВИТЬ в базу данных новую запись. Для корневой записи обязательно формирование значения ключа.

ИЗМЕНИТЬ значение данных предварительно извлеченной записи. Ключевые данные не должны подвергаться изменениям.

УДАЛИТЬ некоторую запись и все подчиненные ей записи.

ИЗВЛЕЧЬ корневую запись по ключевому значению, допускается также последовательный просмотр корневых записей

извлечь следующую запись (следующая запись извлекается в порядке левостороннего обхода дерева)

В операции **ИЗВЛЕЧЬ** допускается задание условий выборки (например, извлечь сотрудников с окладом более 1 тысячи руб.)

Все операции изменения применяются только к одной "текущей" записи (которая предварительно извлечена из базы данных). Такой подход к манипулированию данными получил название "навигационного".

Заключение

С ростом популярности СУБД в 70-80-х годах появилось множество различных моделей данных. У каждой из них имелись свои достоинства и недостатки, которые сыграли ключевую роль в развитии реляционной модели данных, появившейся во многом благодаря стремлению упростить и упорядочить первые модели данных.

Современные БД основываются на использовании моделей данных, позволяющих описывать объекты предметных областей и взаимосвязи между ними. Модели данных используются, как для концептуального, так и для логического и физического представления данных.

Основное различие между этими моделями данных состоит в способах описания взаимодействий между объектами и атрибутами.

Библиографический список

1. Зеленков Ю.А. "Введение в базы данных". Учебный курс.

<http://www.mstu.edu.ru/education/materials/zelenkov/toc.html>

2. Bachman C. W. The Programmer as Navigator, CACM 16.11, Nov. 1973.

3. Журнал "СУБД" № 1, 1995. Реляционная модель данных для больших совместно используемых банков данных

<http://www.osp.ru/dbms/1995/01/01.htm>

4. New Data Management Markets. Gartner Group, August 1999