

КОНЦЕПЦИЯ БАЗ ДАННЫХ

Лекция 1

Базы данных: основные понятия и определения

Система баз данных – это компьютерная система хранения записей, т.е. компьютеризированная система, основное назначение которой – хранить информацию, предоставляя пользователем средства ее извлечения и модификации.

К информации можно отнести все, что заслуживает внимания отдельного пользователя или организации, использующей систему, иначе говоря, все необходимое для текущей работы данного пользователя или предприятия.

Упрощенная схема систем баз данных состоит из 4-х главных компонентов: данные, аппаратное обеспечение, программное обеспечение и пользователи.

База данных - поименованная совокупность структурированных данных, относящихся к определенной предметной области, организованных по определенным правилам, предусматривающим общие принципы описания, хранения и манипулирования данными, а также их оптимальное использование для одного или нескольких приложений. При этом данные должны быть непротиворечивы, минимально избыточны и целостны.

Обычно БД создается для хранения и доступа к данным, содержащим сведения о некоторой предметной области, то есть некоторой области человеческой деятельности или области реального мира. Всякая БД должна представлять собой **систему** данных о предметной области. БД, относящиеся к одной и той же предметной области, в различных случаях содержат более или менее детализированную информацию о ней. Степень детализации определяется рядом факторов, прежде всего целью использования информации из базы данных и сложностью производственных (деловых) процессов, существующих в пределах предметной области в конкретных условиях.

В современной технологии баз данных предполагается, что создание базы данных, ее поддержка и обеспечение доступа пользователей к ней осуществляются централизованно с помощью специального программного инструментария - *системы управления базами данных*.

Система управления базами данных (СУБД) - это комплекс программных и языковых средств, необходимых для создания баз данных,

поддержания их в актуальном состоянии и организации поиска в них необходимой информации.

Использование СУБД при построении информационных систем призвано реализовать физическую и логическую независимость прикладного программирования от данных. Физическая независимость от данных заключается в том, что работа программного обеспечения ИС не будет зависеть от изменений, которые могут происходить на внутреннем, физическом уровне. Эти изменения могут заключаться, например, в том, что будет изменена файловая система или же в том, что изменится структура тех файлов, которые составляют базу данных.

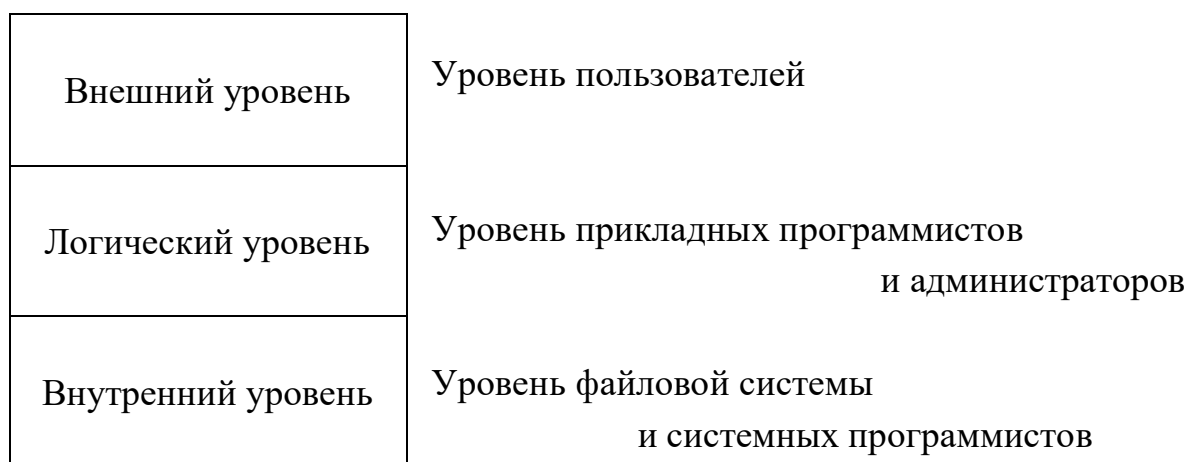


Рисунок 1.1. Три уровня восприятия данных

Логическая независимость прикладного программирования от данных при использовании СУБД в трехуровневой структуре доступа к данным (рис. 1.1) заключается, прежде всего, в том, что добавление новых элементов (например, добавление нового столбца в таблицу) в структуру данных никак не влияет на функционирование программного обеспечения. Добиться логической независимости от операций удаления элементов из базы данных в общем случае невозможно, поскольку на удаляемые элементы, как правило, имеются явные ссылки в программном обеспечении.

Централизованный характер управления данными в базе данных предполагает необходимость существования некоторого лица (группы лиц), на которое возлагаются функции администрирования данных, хранимых в базе (администратор БД).

База данных предполагает централизованное управление данными, что обеспечивает ряд преимуществ:

- сокращение избыточности хранимых данных благодаря однократному хранению каждого сообщения в базе данных,
- совместное использование хранимых данных всеми пользователями ИС,
- стандартизацию представления данных, упрощающую проблемы эксплуатации БД и обмена данными между ИС,
- обеспечение процедур проверки достоверности информации и процедур ограничения доступа к данным,
- совмещение требований к использованию БД со стороны различных пользователей ИС.

База знаний - организованная по особым принципам совокупность знаний, относящихся к какой-либо предметной области.

База знаний отличается от *базы данных* тем, что если единицы информации в базе данных представляют собой не связанные друг с другом сведения, формулы, теоремы, аксиомы, то в базе знаний те же элементы уже связаны как между собой, так и с понятиями внешнего мира определенными отношениями и сами содержат в себе эти отношения.

Виды баз данных

1. По *технологии обработки* данных базы данных подразделяются на *централизованные* и *распределенные*.

Централизованная база данных хранится в памяти одной вычислительной системы. Если эта вычислительная система является компонентом сети ЭВМ, возможен распределенный доступ к такой базе. Такой способ использования баз данных часто применяют в локальных сетях.

Распределенная база данных состоит из нескольких, возможно пересекающихся или даже дублирующих друг друга частей, хранимых в различных ЭВМ вычислительной сети. Работа с такой базой осуществляется с помощью системы управления распределенной базой данных (СУРБД).

2. По *способу доступа к данным* базы данных разделяются на *базы данных с локальным доступом* и *базы данных с удаленным (сетевым доступом)*.

3. Системы централизованных баз данных с сетевым доступом предполагают различные *архитектуры* подобных систем:

- файл-сервер;
- клиент-сервер.

Файл-сервер. Архитектура систем БД с сетевым доступом предполагает выделение одной из машин сети в качестве центральной (сервер файлов). На такой машине хранится совместно используемая централизованная БД. Все другие машины сети выполняют функции рабочих станций, с помощью которых поддерживается доступ пользовательской системы к централизованной базе данных. Файлы базы данных в соответствии с пользовательскими запросами передаются на рабочие станции, где в основном и производится обработка. При большой интенсивности доступа к одним и тем же данным производительность информационной системы падает. Пользователи могут создавать также на рабочих станциях локальные БД, которые используются ими монопольно. Концепция файл-сервер условно отображена на рисунке 1.2.

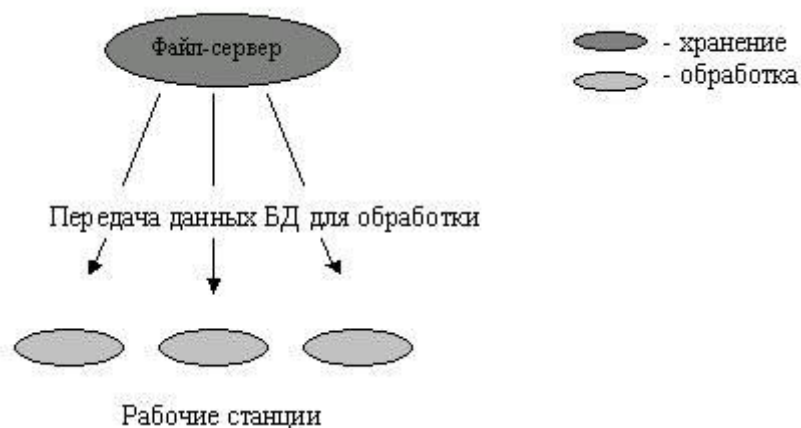


Рисунок 1.2. Схема обработки информации в БД по принципу файл-сервер

Клиент-сервер. В этой концепции подразумевается, что помимо хранения централизованной базы данных центральная машина (сервер базы данных) должна обеспечивать выполнение основного объема обработки данных. Запрос на данные, выдаваемый клиентом (рабочей станцией), порождает поиск и извлечение данных на сервере. Извлеченные данные (но не файлы) транспортируются по сети от сервера к клиенту. Спецификой архитектуры клиент-сервер является использование языка запросов SQL. Концепция клиент-сервер условно изображена на рисунке 1.3.



Рисунок 1.3. Схема обработки информации в БД по принципу клиент-сервер

Выбор хранимых данных

Для анализа информационных потоков в управляемой системе исходными являются данные о парных взаимосвязях, или отношениях (т.е. есть отношение или нет отношения), между наборами информационных элементов. Под информационными элементами понимают различные типы входных, промежуточных и выходных данных, которые составляют наборы входных N_1 , промежуточных N_2 и выходных N_3 элементов данных.

Формализованно связи (парные отношения) между наборами информационный элементов отображаются в виде матрицы смежности B , под которой понимают квадратную бинарную матрицу, проиндексированную по обеим осям множеством информационных элементов $D = \{c\}$ где s - число этих элементов:

$$B = \begin{matrix} & \begin{matrix} d_1 & d_2 & \dots & d_j & \dots & d_s \end{matrix} \\ \begin{matrix} d_1 \\ d_2 \\ \dots \\ d_i \\ \dots \\ d_s \end{matrix} & \left| \begin{array}{cccccc} q_{11} & q_{12} & \dots & q_{1j} & \dots & q_{1s} \\ q_{21} & q_{22} & \dots & q_{2j} & \dots & q_{2s} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ q_{i1} & q_{i2} & \dots & q_{ij} & \dots & q_{is} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ q_{s1} & q_{s2} & \dots & q_{sj} & \dots & q_{ss} \end{array} \right. \end{matrix} \quad (1.1)$$

где $q_{ij} = \begin{cases} 1, & \text{если между } d_i \text{ и } d_j \text{ отношение существует;} \\ 0 & \text{в противном случае.} \end{cases}$

В позиции (i, j) матрицы смежности записывают 1 (т.е. $q_{ij}=1$), если между информационными элементами d_i и d_j существует отношение R_0 , такое, что для получения значения информационного элемента d_j необходимо непосредственное обращение к элементу d_i . Наличие такого отношения между

d_i и d_j обозначают в виде $d_i \text{ Ro } d_j$, чему соответствует $q_{ij} = 1$, а отсутствие - в виде $d_i \text{ Ro } d_j$, т.е. $q_{ij} = 0$. Для простоты принимают, что каждый информационный элемент недостижим из самого себя:

$$d_i \text{ Ro } d_j; i=1, \overline{S} \quad (1.2)$$

Матрице B ставится в соответствие информационный граф $G = (D, \text{Ro})$. Множеством вершин графа $G = (D, \text{Ro})$ является множество D информационных элементов, а каждая дуга (d_i, d_j) соответствует условию $d_i \text{ Ro } d_j$, т.е. записи 1 в позиции (ij) матрицы B .

Например, задано множество D из четырех наборов информационных элементов, т.е. $D = \{d_1, d_2, d_3, d_4\}$. Пусть матрица смежности B этих элементов имеет вид:

$$B = \begin{matrix} & \begin{matrix} d_1 & d_2 & d_3 & d_4 \end{matrix} \\ \begin{matrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{matrix} & \begin{vmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{vmatrix} \end{matrix} \quad (1.3)$$

Из этой матрицы видно, что для вычисления элемента d_3 необходимо обращение к элементам d_1 и d_2 , а для получения элемента d_4 — к элементу d_3 . Чтобы получить элемент d_1 , надо обратиться к d_3 . Элемент d_2 не зависит от других элементов матрицы. Информационный граф в этом простейшем случае будет соответствовать рисунку 1.4.

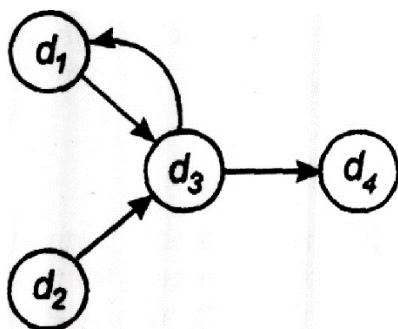


Рисунок 1.4. Информационный граф $G = (D, \text{Ro})$

В общем случае структура графа $G = (D, \text{Rq})$ вследствие неупорядоченности сложна для восприятия и анализа. Составленная на основе инфологической модели, она не гарантирована от неточностей, ошибок, избыточности и транзитивности. Для формального выделения входных,

промежуточных и выходных наборов информационных элементов, определения последовательности операций их обработки, анализа и уточнения взаимосвязей на основе графа $G=(D,R_0)$ строят матрицу достижимости.

Матрицей достижимости M называют квадратную бинарную матрицу, проиндексированную по обеим осям множеством информационных элементов D аналогично матрице смежности B . Запись 1 в каждой позиции (ij) матрицы достижимости соответствует наличию для упорядоченной пары информационных элементов (d_i, d_j) смыслового отношения достижимости R . Элемент d_i достижим из элемента d_j , т.е. выполняется условие $d_j R_0 d_i$, если на графе $G = (D, R_0)$ существует направленный путь от вершины d_i к вершине d_j (в процессе получения значения элемента d_j используется значение элемента d_i). Если $\overline{d_i R_0 d_j}$, то отношение достижимости между элементами d_i и d_j отсутствует и в позиции (ij) матрицы M записывают 0. Отношение достижимости транзитивно, т.е. если $d_i R_0 d_k$ и $d_k R_0 d_j$, то $d_i R_0 d_j$; $i, j, k=1, S$.

Записи 1 j -м столбце матрицы M соответствуют информационным элементам d_i , которые необходимы для получения значений элементов d_j , и образуют множество элементов предшествования для этого элемента. Записи 1 в i -й строке матрицы M соответствуют всем элементам d_j , достижимым из рассматриваемого элемента d_i и образующим множество достижимости $R(d_i)$ этого элемента. Информационные элементы, строки которых в матрице M не содержат единиц (нулевые строки), являются выходными информационными элементами, а информационные элементы, соответствующие нулевым столбцам матрицы M , являются входными. Это условие может служить проверкой правильности заполнения матриц B и M , если наборы входных и выходных информационных элементов известны. Информационные элементы, не имеющие нулевой строки или столбца, являются промежуточными.

Для полученного графа (см. рис. 1.5) матрица M будет выглядеть следующим образом:

$$M = \begin{matrix} & \begin{matrix} d_1 & d_2 & d_3 & d_4 \end{matrix} \\ \begin{matrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{matrix} & \begin{vmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{vmatrix} \end{matrix}. \quad (1.4)$$

Отличие столбцов матриц M и B объясняется тем, что в матрице M учитывается смысловое отношение R между информационными элементами, а в матрице B - только непосредственно R_0 . Например, элемент d_4 в матрице M

достигим из элементов d_1 , d_2 и d_3 т.е. d_1Rd_4 , d_2Rd_4 и d_3Rd_4 , в то время как в матрице B для этих элементов d_4 достигим только из d_3 т.е. только d_3Rd_4 . Из анализа матрицы M следует, что элемент d_3 является входным, d_4 - выходным, остальные - промежуточные. На основе матрицы M строится информационный граф $G_s(D,R)$ системы, структурированный по входным (N_1), промежуточным (N_2) и выходным (N_3) наборам информационных элементов и полученный из анализа множества элементов предшествования $A(d_i)$ и достижимости $R(d_j)$ (рис. 1.5).

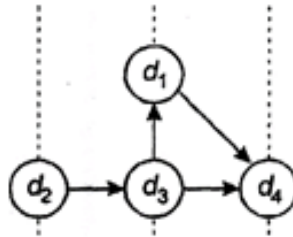


Рисунок 1.5. Информационный граф $G_s(D,R)$

В общем случае информационный граф системы в отличие от вычисленного графа может иметь контуры и петли, что объясняется необходимостью повторного обращения к отдельным элементам данных. Информационный граф системы $G_s(D,R)$ структурируется по уровням (N_1 , N_2 , N_3) с использованием итерационной процедуры, что позволяет определить информационные входы и выходы системы, выделить основные этапы обработки данных, их последовательность и циклы обработки на каждом уровне. Кроме того, удаляются избыточные (лишние) дуги и элементы. Граф, получаемый после структуризации по наборам информационных элементов и удаления избыточных элементов и связей, определяет каноническую структуру информационной базы. Таким образом, каноническая структура задает логически не избыточную информационную базу. Выделение наборов элементов данных по уровням позволяет объединить множество значений конечных элементов в логические записи и тем самым упорядочить их в памяти ЭВМ. От канонической структуры переходят к логической структуре информационной базы, а затем к физической организации информационных массивов. Каноническая структура служит также основой для автоматизации основных процессов предпроектного анализа предметных областей. Процедуры хранения, актуализации и извлечения данных непосредственно связаны с базами данных, поэтому логический уровень этих процедур определяется моделями баз данных.