

image not found or type unknown



Ядром любой базы данных является модель данных. Модель данных – это совокупность структур данных и операций их обработки. С помощью модели данных могут быть представлены информационные объекты и взаимосвязи между ними. Рассмотрим три основных типа моделей данных: иерархическую, сетевую и реляционную.

Иерархическая модель данных представляет собой совокупность элементов данных, расположенных в порядке их подчинения и образующих по структуре перевернутое дерево. К основным понятиям иерархической модели данных относятся: уровень, узел и связь.

Сетевая модель данных основана на тех же основных понятиях (уровень, узел, связь), что и иерархическая модель, но в сетевой модели каждый узел может быть связан с любым другим узлом.

Реляционная модель данных использует организацию данных в виде двумерных таблиц. Каждая такая таблица, называемая реляционной таблицей или отношением, представляет собой двумерный массив.

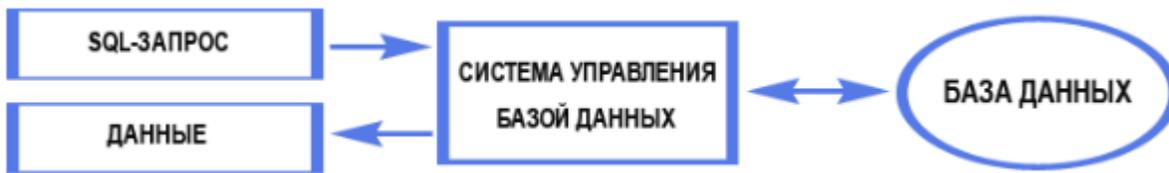
Основными структурными элементами реляционной таблицы являются поле и запись. Поле (столбец реляционной таблицы) – элементарная единица логической организации данных, которая соответствует конкретному атрибуту информационного объекта. Запись (строка реляционной таблицы) – совокупность логически связанных полей, соответствующая конкретному экземпляру информационного объекта.

Основная часть

Базы данных представляют собой синтез структур данных и файловых структур.

Термин база данных относится к совместно используемому набору логически связанных данных (и описанию этих данных), предназначенному для удовлетворения информационных потребностей организации. База данных - набор сведений, хранящихся некоторым упорядоченным способом. Можно сравнить базу данных со шкафом, в котором хранятся документы. Иными словами, база данных - это хранилище данных.

В случае баз данных на внешнем носителе хранятся данные совместно с их описанием (метаданные), в то время как в файлах хранятся просто двоичные данные, логическое описание которых содержится в соответствующих программах.



Файловая система используется для обозначения программной системы, управляющей файлами, хранящимися во внешней памяти.

- Файловая система – это дополнительный программный слой, обеспечивающий возможность работы прикладного программиста на уровне логической организации файлов.

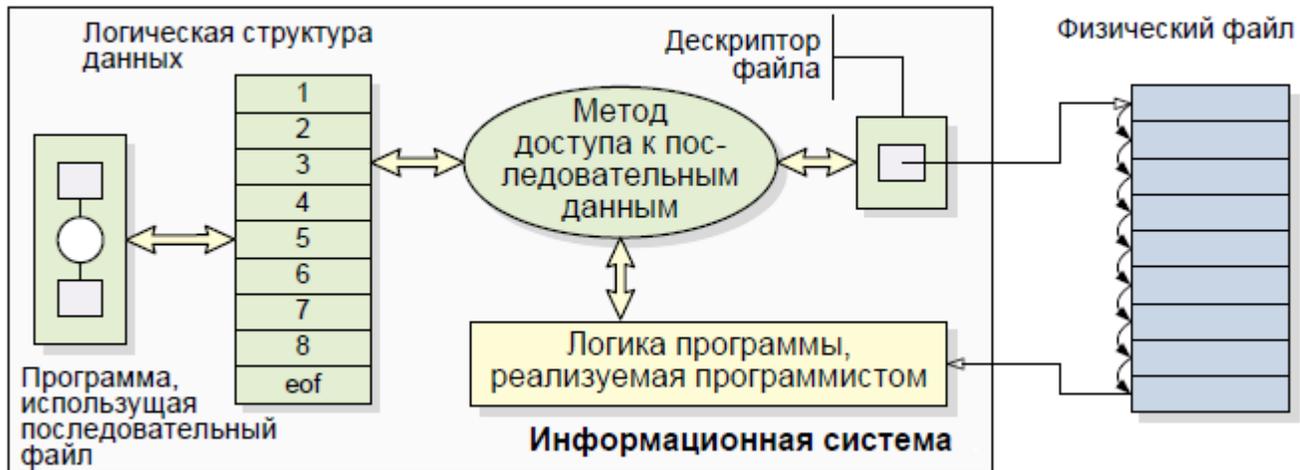
Включает в себя следующие компоненты:

- – совокупность всех файлов на диске с их физической организацией;
- – наборы структур данных, используемых для управления файлами (каталоги файлов, дескрипторы файлов, таблицы распределения свободного и занятого пространства на диске, т. е. логическая организация файловых структур);
- – комплекс системных программных средств, реализующих управление файлами (создание, уничтожение, чтение, запись, поиск и другие операции над файлами)

Файловая информационная система Информационная система с базой данных



Последовательный и ассоциативный доступ в файловых системах.



Последовательный доступ

Использовался самый простой способ локализации записи – сканирование файла до выявления требуемой записи. Для реализации данного способа доступа достаточно было знать начальный адрес файла. Есть ключевое поле, составной ключ и первичный ключ.

Неотъемлемой частью процесса обработки последовательного файла является определение конца файла, а для этого мы должны иметь возможность распознавать записи по какому-нибудь признаку. Логические записи распознаются, как правило, по одному полю в записи. В файле сотрудников это может быть поле, содержащее идентификационный номер сотрудника. Такое поле называется ключевым. Хотя во многих приложениях требуется идентифицировать записи по ключам, которые не являются уникальными (например, Ф, И, О), но при этом все равно должен существовать один уникальный ключ, используемый для идентификации записи в файле. Такой ключ называется первичным или идентификатором. Иногда бывает необходимо объединить несколько полей, чтобы обеспечить уникальность ключа, который в этом случае называется составным ключом.

Ассоциативный доступ

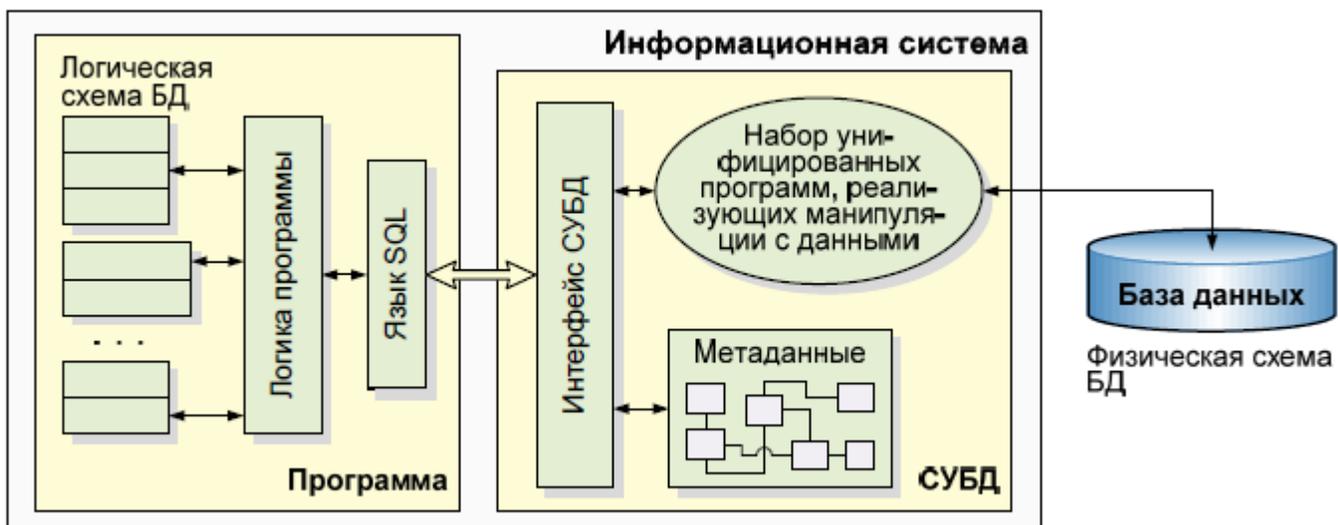
Для создания индексированных файлов на основе ключей были реализованы специальные таблицы, переводящие ассоциативный запрос в соответствующий адрес. Эти таблицы были названы списками ссылок или индексами.

Индекс определяется как таблица, содержащая список ключевых значений, каждому из которых соответствует указатель, локализующий блок записей на

носителе данных. Чтобы найти определенный блок информации, сначала необходимо отыскать в индексе его ключ, а потом получить сам блок, который хранится по адресу, связанным с этим ключом.

Классическим примером использования индексированного файла является обслуживание записей сотрудников. За счет создания индекса можно избежать длительных операций поиска для получения отдельной записи. В частности, если файл записей сотрудников индексирован по идентификационным номерам сотрудников, то определенную запись можно быстро получить, если этот номер известен.

Системы управления базами данных (СУБД).



В ходе эволюции файловой системы, а затем и концепции баз данных, фрагмент логики программы, отвечающий за **управление данными**, был выделен в отдельный компонент, названный **СУБД**.

Основные функции СУБД и их реализация.

- Непосредственное управление данными во внешней памяти:

Эта функция включает обеспечение необходимых структур внешней памяти как для хранения данных, непосредственно входящих в БД, так и для служебных целей, например, для ускорения доступа к данным в некоторых случаях (обычно для этого используются индексы). В некоторых реализациях СУБД активно используются возможности существующих файловых систем, в других работа производится вплоть до уровня устройств внешней памяти.

- Управление буферами оперативной памяти:

СУБД обычно работают с БД значительного размера; по крайней мере, этот размер обычно существенно больше доступного объема оперативной памяти. Понятно, что если при обращении к любому элементу данных будет производиться обмен с внешней памятью, то вся система будет работать со скоростью устройства внешней памяти. Практически единственным способом реального увеличения этой скорости является буферизация данных в оперативной памяти. Поэтому в развитых СУБД поддерживается собственный набор буферов оперативной памяти с собственной дисциплиной замены буферов.

- Управление транзакциями

Транзакция - это последовательность операций над БД, рассматриваемых СУБД как единое целое.

- - Либо транзакция успешно выполняется, и СУБД фиксирует изменения БД, произведенные этой транзакцией, во внешней памяти, либо ни одно из этих изменений никак не отражается на состоянии БД.
 - Понятие транзакции необходимо для поддержания логической целостности БД. Приведем пример информационной системы с файлами СОТРУДНИКИ и ОТДЕЛЫ, единственным способом не нарушить целостность БД при выполнении операции приема на работу нового сотрудника является объединение элементарных операций над файлами СОТРУДНИКИ и ОТДЕЛЫ в одну транзакцию.
- Журнализация

Одним из основных требований к СУБД является надежность хранения данных во внешней памяти.

Под надежностью хранения понимается то, что СУБД должна быть в состоянии восстановить последнее согласованное состояние БД после любого аппаратного или программного сбоя.

Обычно рассматриваются два возможных вида аппаратных сбоев:

- - так называемые мягкие сбои, которые можно трактовать как внезапную остановку работы компьютера (например, аварийное выключение питания),

- и жесткие сбои, характеризующиеся потерей информации на носителях внешней памяти.

Журнал - это особая часть БД, недоступная пользователям СУБД и поддерживаемая с особой тщательностью (иногда поддерживаются две копии журнала, располагаемые на разных физических дисках), в которую поступают записи обо всех изменениях основной части БД.

В разных СУБД изменения БД журналируются на разных уровнях: иногда запись в журнале соответствует некоторой логической операции изменения БД (например, операции удаления строки из таблицы реляционной БД), иногда - минимальной внутренней операции модификации страницы внешней памяти; в некоторых системах одновременно используются оба подхода.

- Поддержка языков БД

Для работы с базами данных используются специальные языки, в целом называемые языками баз данных. В ранних СУБД поддерживалось несколько специализированных по своим функциям языков. Чаще всего выделялись два языка

- ○ язык определения схемы БД (SDL - Schema Definition Language) и
- язык манипулирования данными (DML - Data Manipulation Language).

В современных СУБД обычно поддерживается единый интегрированный язык, содержащий все необходимые средства для работы с БД, начиная от ее создания, и обеспечивающий базовый пользовательский интерфейс с базами данных. Стандартным языком наиболее распространенных в настоящее время реляционных СУБД является язык запросов SQL (Structured Query Language).

Список литературы

- Проектирование баз данных. СУБД Microsoft Access. Учебное пособие — Н. Н. Гринченко, Е. В. Гусев, Н. П. Макаров.
- Самоучитель Microsoft Access 2013 — Бекаревич Ю.Б., Пушкина Н. В.
- Microsoft Access 2007. Разработка приложений на реальном примере — Геннадий Гурвиц.