

Системы управления базами данных (СУБД)

Олег Анатольевич Тарлапан
Сергей Вячеславович Морозов
Институт системного программирования РАН
снс, кфмн

Основной учебник

учебное пособие «Базы данных» автор С.Д.Кузнецов (изд.центр Академия 2012)
учебное пособие «Основы баз данных» автор С.Д.Кузнецов (изд. Бином 2007)
учебное пособие «Базы данных. Модели и языки» автор С.Д.Кузнецов (изд. 2008)

Интернет

Интерактивный курс С.Д. Кузнецова «Введение в реляционные базы данных»
Университет Информационных технологий www.intuit.ru (текстовый www.citforum.ru)

Дополнительная литература:

К. Дж. Дейт. Введение в системы баз данных, 8-е издание. М., Вильямс, 2005
К. Дж. Дейт, Хью Дарвен. Основы будущих баз данных. Третий манифест. М., Янус-К, 2004
Марков А.С., Лисовский К.Ю. Базы данных. Введение в теорию и методологию. Учебник.
(Исправл. изд.). - М., "Финансы и статистика", 2006.

Просьба старостам групп оставить контактные телефоны. Фамилия, имя, отчество, группа и телефон (для форс-мажорных случаев)

Информационная система
Файловая система
Система Управления Базами Данных (СУБД)

Информационная система - программно-аппаратный комплекс, выполняющий три основные функции:

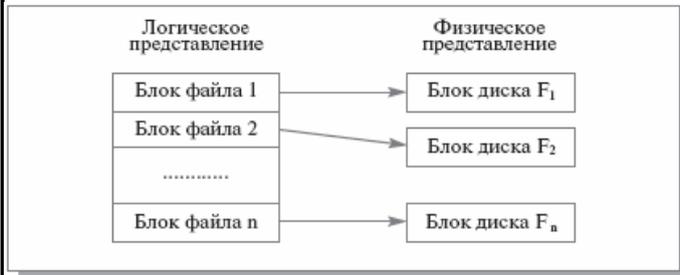
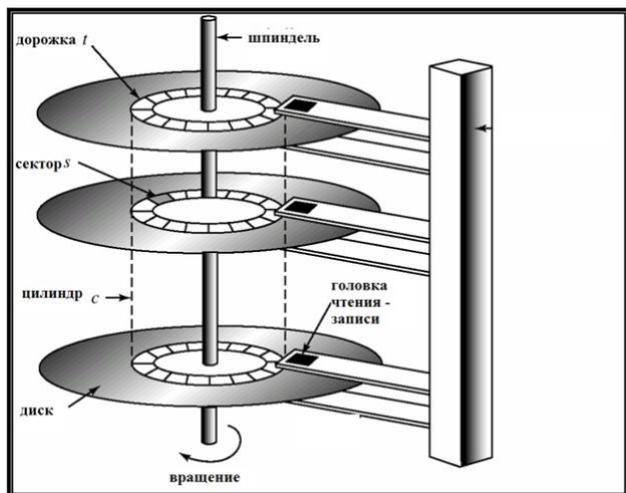
- поддержка надежного хранения информации,
- выполнение специфических для данной системы преобразований и вычислений,
- предоставление пользователю удобного интуитивного интерфейса.

Файл - это именованная область внешней памяти, в которую можно записывать и из которой можно считывать данные.

Функции системы управления файлами:

- распределение внешней памяти,
- отображение имен файлов в соответствующие адреса во внешней памяти,
- обеспечение доступа к данным.

Структура файлов

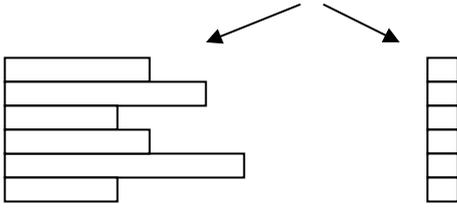


В настоящее время возможность обмениваться с магнитными дисками порциями меньше объема блока не используется в файловых системах.

Во всех файловых системах явно или неявно выделяется **базовый уровень**, обеспечивающий работу с файлами, представляющими набор прямо адресуемых в адресном пространстве файла блоков. Размер логических блоков равен или кратен размеру физического блока диска и размеру страницы виртуальной памяти.

Обычно в файловых системах базовый уровень не доступен пользователю, и закрывается интерфейсом более высокого уровня.

Пользовательские интерфейсы файловых систем



1. представление **файла как последовательность записей**.

Каждая запись - это последовательность байтов постоянного или переменного размера. Записи можно читать или записывать последовательно или позиционировать файл на запись с указанным номером.

2. представление **файла как последовательность байтов**.

Из файла можно прочитать указанное число байтов либо начиная с его начала, либо предварительно произведя его позиционирование на байт с указанным номером. Аналогично, можно записать указанное число байтов в конец файла, либо предварительно произведя позиционирование файла.

Именованние файлов. Каталог.

Классификация файловых систем по способу именования файлов

Современные файловые системы поддерживают многоуровневое именованние файлов за счет поддержания во внешней памяти дополнительных файлов со специальной структурой - **каталОгов**.

Полное (квалифицированное) имя файла состоит из цепочки имен всех родительских каталогов начиная с корневого уровня и заканчивая собственным именем файла.

Изолированные файловые системы

Каждый архив файлов (полное дерево каталогов) целиком располагался на одном логическом диске, т.е. устройстве представляемом с помощью средств операционной системы как отдельный диск. Полное имя файла начинается с имени дискового устройства.

Полностью централизованные файловые системы

Вся совокупность каталогов и файлов представляется как единое дерево, физически распределенное по всем внешним запоминающим устройствам (операционная система Multics).

Компромиссное решение применено в файловых системах ОС UNIX.

На базовом уровне в этих файловых системах поддерживаются изолированные архивы файлов. Один из этих архивов объявляется корневой файловой системой. После запуска системы можно "смонтировать" корневую файловую систему и остальные изолированные файловые архивы в одну общую файловую систему.

Защита файлов от несанкционированного доступа (авторизация доступа)

Мандатный способ защиты

По отношению к каждому зарегистрированному пользователю данной вычислительной системы для каждого существующего файла указываются действия, которые разрешены или запрещены данному пользователю.

Каждый пользователь имеет отдельный **мандат** для работы с каждым файлом.

Дискреционный способ защиты

Каждому зарегистрированному пользователю соответствует пара целочисленных идентификаторов: собственный идентификатор и идентификатор группы, к которой относится пользователь.

Этими же идентификаторами снабжается каждый процесс, запущенный от имени данного пользователя и имеющий возможность обращаться к файловой системе.

Каждый файл системы хранит полный идентификатор пользователя, который создал этот файл, и информацию о том, какие действия с файлом может производить он сам, какие действия с файлом доступны для остальных пользователей той же группы и что могут делать с файлом пользователи других групп.

Для каждого файла контролируется возможность выполнения трех действий: чтение, запись и выполнение.

Достоинства

- компактность (два целых числа для представления идентификаторов и шкала из 9 бит для характеристики возможных действий),
- скорость проверки (требуется небольшое количество действий)

Недостатки

Предопределенный ограниченный набор проверяемых действий

Обеспечение многопользовательского доступа

Контроль совместимости режима при открытии файла

Области разумного применения файлов

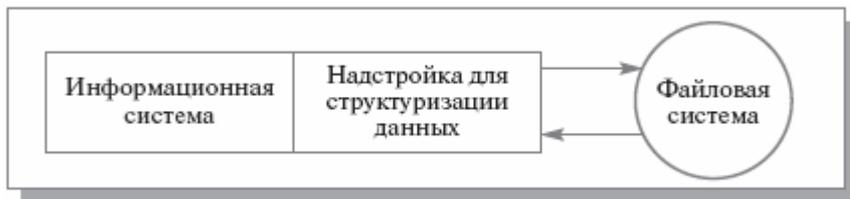


Цепочка связей между программными компонентами при написании программы

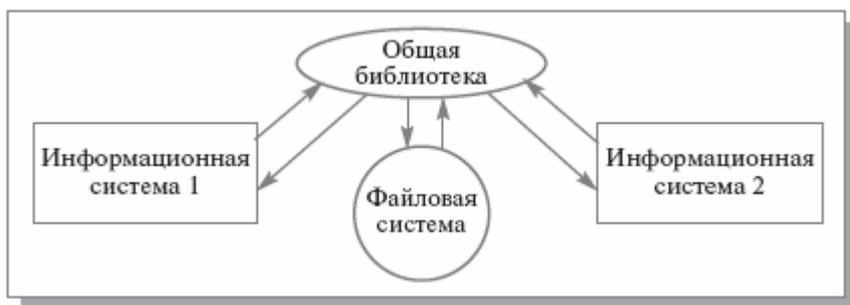
Файловые системы обеспечивают хранение слабо структурированной информации, оставляя дальнейшую структуризацию прикладным программам.

Потребности информационных систем

Структура типовой информационной системы на начальном этапе использования вычислительной техники



Возможная схема интеграции сложных информационных систем.
Попытка создания первых прототипов СУБД
(две информационные системы с общей библиотекой работы с файловой системой)



Система учета служащих некоторой организации

Система должна поддерживать возможность получения следующих данных:

Для «служащего»:

- номера удостоверения по полному имени служащего (для простоты мы полагаем, что имена всех служащих различны);
- полного имени по номеру удостоверения;
- информации о соответствии занимаемой должности, и размере зарплаты.

Для «отдела»:

- имени руководителя отдела;
- общей численности отдела;
- общей суммы зарплаты служащих отдела, среднего размера зарплаты и т. д.

Для «организации»:

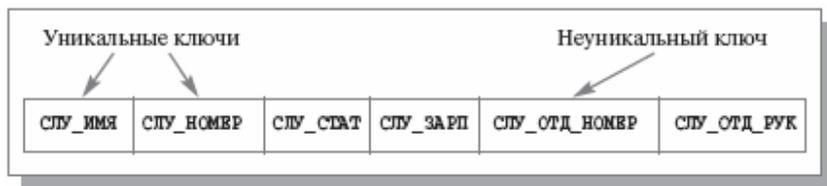
- списка служащих по отделам;

Кроме того, система обеспечивать средства поддержки:

- перевода служащего из одного отдела в другой;
- приема на работу новых и увольнения работающих служащих.

Необходимые структуры данных

Структура файла СЛУЖАЩИЕ (случай одного файла)



Пример файла СЛУЖАЩИЕ

| | | | | | |
|---------|----|-----|------------|----|---------|
| Иванов | 33 | Да | 100000 | 10 | Астахов |
| Астахов | 22 | Да | 200000 | 10 | Астахов |
| Яковлев | 11 | Нет | 30 00 0 | 20 | Яковлев |

Вспомогательные **индексы** ускоряющие доступ

| СЛУ_ИМЯ | СЛУ_НОМЕР | СЛУ_ОТД_НОМЕР |
|---------|-----------|---------------|
| 1 | 2 | 0 |
| 0 | 1 | 1 |
| 2 | 0 | 2 |

Сложности:

- требуется создание достаточно сложной надстройки для многоключевого доступа к файлам;
- возникает существенная избыточность данных (для каждого служащего повторяется имя руководителя его отдела);
- требуется выполнение массовой выборки и вычислений для получения суммарной информации об отделах.

Структура файла СЛУЖАЩИЕ и ОТДЕЛЫ (случай двух файлов)
Каждый из файлов содержит только не дублируемую информацию.



Минимизированы динамические вычисления суммарной информации по отделам.

Сложности:

- система управления данными должна знать об их структуре,
- система управления данными должна поддерживать согласованность данных.

Обеспечение Согласованности(Целостности) данных

1. Если в файле СЛУЖАЩИЕ содержится запись со значением поля СЛУ_ОТД_НОМЕР, равным N, то и в файле ОТДЕЛЫ должна содержаться запись со значением поля ОТД_НОМЕР, также равным N

$\forall tr \in Bt(\text{СЛУЖАЩИЕ}) \Rightarrow \exists ts \in Bs(\text{ОТДЕЛЫ}) : tr.\text{СЛУ_ОТД_НОМЕР} = ts.\text{ОТД_НОМЕР}$

2. Если в файле ОТДЕЛЫ содержится запись со значением поля ОТД_РУК, равным M, то и в файле СЛУЖАЩИЕ должна содержаться запись со значением поля СЛУ_НОМЕР, также равным M

$\forall ts \in Bs(\text{ОТДЕЛЫ}) \Rightarrow \exists tr \in Br(\text{СЛУЖАЩИЕ}) : ts.\text{ОТД_РУК} = tr.\text{СЛУ_НОМЕР}$

(1) и (2) **ограничения ссылочной целостности.**

3. При любом корректном состоянии информационной системы значение поля ОТД_РАЗМЕР любой записи отд_k файла ОТДЕЛЫ должно быть равно числу всех тех записей файла СЛУЖАЩИЕ, в которых значение поля СЛУ_ОТД_НОМЕР совпадает со значением поля ОТД_НОМЕР записи отд_k

4. При любом корректном состоянии информационной системы значение поля ОТД_СЛУ_ЗАРП любой записи отд_k файла ОТДЕЛЫ должно быть равно сумме значений поля СЛУ_ЗАРП всех тех записей файла СЛУЖАЩИЕ, в которых значение поля СЛУ_ОТД_НОМЕР совпадает со значением поля ОТД_НОМЕР записи отд_k.

(3) и (4) **ограничения целостности общего вида**

Поддержка согласованности данных – **основной признак СУБД.**

Информацию, описывающую структуру хранимых данных и накладываемые на них ограничения целостности принято называть **метаданными.**

Формулировка запросов к системе

Наиболее естественный способ получения информации – формулировка запроса на некотором языке максимально близком пользователям.

Такие языки называются **языками запросов к базам данных**.

Получения общей численности отдела, в котором работает Иванов Иван Иванович

```
SELECT ОТД_РАЗМЕР  
FROM СЛУЖАЩИЕ, ОТДЕЛЫ  
WHERE СЛУ_ИМЯ = 'ИВАНОВ ИВАН ИВАНОВИЧ' AND  
СЛУ_ОТД_НОМЕР = ОТД_НОМЕР;
```

Получение списка служащих, не соответствующих занимаемой должности

```
SELECT СЛУ_ИМЯ, СЛУ_НОМЕР  
FROM СЛУЖАЩИЕ  
WHERE СЛУ_СТАТ = "НЕТ";
```

Запросы носят **декларативный** характер

Поддержка надежности хранения данных (восстановление после сбоев)

В **файловой системе** сбой при добавлении служащего может вызвать рассогласованное состояние файлов СЛУЖАЩИЕ и ОТДЕЛЫ. Будет нарушено правило (3) (численность), вероятно правило (4) (зарплата), и возможно правило (1) (номер отдела).

В процессе восстановления система должна это обнаружить и исправить. Системы управления файлами для этого мало пригодны.

СУБД берут заботу о поддержании целостности данных при различных сбоях системы на себя, поддерживая **транзакционное управление и журнализацию изменений базы данных**.

Поддержка параллельной многопользовательской работы

В **файловых системах** любая операция одного из пользователей по изменению данных будет приводить к полной блокировке чтения информации из базы данных всеми пользователями.

В **СУБД** реализуются механизмы, обеспечивающие гораздо более тонкую синхронизацию параллельного доступа к данным, на уровне отдельных записей.

СУБД как независимый системный компонент

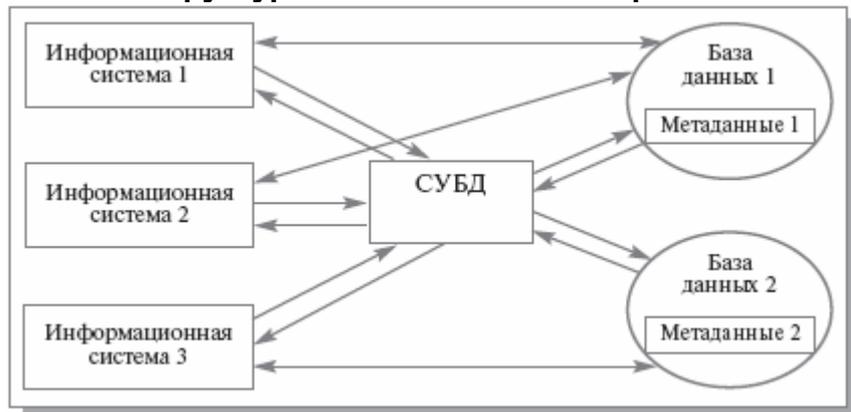
СУБД в составе информационной системы



Дефекты:

1. СУБД должны располагаться вне информационной системы.
2. Метаданные должны располагаться вместе с данными вне информационной системы.

Типовая структура использования современной СУБД



СУБД

Потребности информационных систем, которые не покрываются возможностями систем управления файлами:

- поддержание логически согласованного набора файлов;
- обеспечение языка описания и манипулирования данными;
- восстановление информации после разного рода сбоев;
- эффективная параллельная работа нескольких пользователей.

СУБД – система управления данными выполняющая следующий набор функций:

- управление **логически согласованными данными** в долговременной памяти;
- управление буферами оперативной памяти;
- управление транзакциями;
- журнализация и восстановление БД после сбоев;
- поддержание языков БД.

Управление транзакциями

Транзакция - это последовательность операций над БД, рассматриваемых СУБД как единое целое. Либо транзакция успешно выполняется, и СУБД фиксирует (COMMIT) изменения БД, произведенные этой транзакцией, во внешней памяти, либо ни одно из этих изменений никак не отражается на состоянии БД.

Основные свойства транзакций **АСИД(ACID)**:

- **Атомарность/atomicity**. Транзакция выполняется как атомарная операция - либо выполняется вся транзакция целиком, либо она целиком не выполняется.
- **Согласованность/consistency**. Транзакция переводит базу данных из одного согласованного (целостного) состояния в другое согласованное (целостное) состояние. Внутри транзакции согласованность базы данных может нарушаться.
- **Изоляция/isolation**. Транзакции разных пользователей не должны мешать друг другу (например, как если бы они выполнялись строго по очереди).
- **Долговечность/durability**. Если транзакция выполнена, то результаты ее работы должны сохраниться в базе данных, даже если в следующий момент произойдет сбой системы.

Сериализация транзакций (сериальный план выполнения смеси транзакций) - такой порядок планирования их работы, при котором суммарный эффект смеси транзакций эквивалентен эффекту их некоторого последовательного выполнения.

Надежность СУБД

Под **надежностью** СУБД понимают ее способность восстановить последнее согласованное состояние БД после любого сбоя.

Классификация сбоев по типу происхождения:

- **аппаратные,**
- **программные.**

Классификация сбоев по степени результирующего воздействия на БД:

- **мягкие,**
- **жесткие.**

Поддержание надежности хранения данных в БД достигается **избыточностью** хранимых данных за счет **ведения журнала изменений БД.**

Журнализация

Журнал - особая часть БД, недоступная пользователям СУБД и поддерживаемая с особой тщательностью, в который поступают записи обо всех изменениях основной части БД.

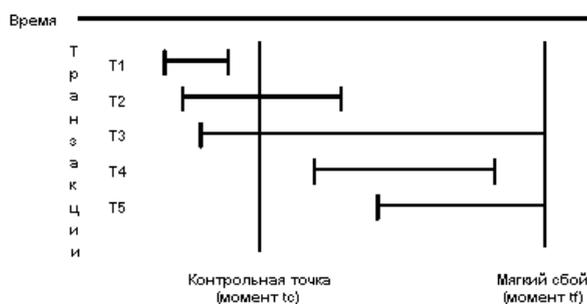
Уровни журнализация БД:

- **логический**, когда запись в журнале соответствует некоторой логической операции изменения БД (например, операции удаления строки из таблицы реляционной БД),
- **физический**, когда запись в журнале соответствует минимальной внутренней операции модификации страницы внешней памяти. (например, изменение сегмента внешней памяти в таком то интервале адресов)

Основная идея журнализации это использование стратегии "**упреждающей**" записи в журнал (протокола **WAL (Write Ahead Log)** – первичная запись лога).

Способы использования журнала в СУБД:

- индивидуальный откат транзакции (ROLLBACK),
- восстановление БД после мягкого сбоя,
- восстановление БД после жесткого сбоя.



Поддержка языков БД

SDL - Schema Definition Language

DML - Data Manipulation Language

SQL - Structured Query Language

Единый интегрированный язык, содержащий все необходимые средства для работы с БД, начиная от ее создания, и обеспечивающий базовый пользовательский интерфейс с базами данных.

Типовая организация современной СУБД

Современная реляционная СУБД включает следующие компоненты:

- внутреннюю часть - ядро СУБД (часто его называют Data Base Engine),
- компилятор языка БД (обычно SQL),
- подсистему поддержки времени выполнения,
- набор утилит.

Ранние подходы к организации БД

Общие характеристик ранних систем:

- Все ранние системы не основывались на каких-либо абстрактных моделях и не имели какой-либо математической модели.
- В ранних системах доступ к БД производился на уровне записей.
- Вся оптимизация доступа к БД осуществлялась пользователем.
- Большинство ранних систем было впоследствии оснащено "реляционными" интерфейсами.

Иерархические системы

Структура иерархической БД

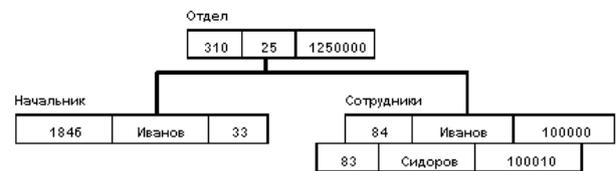
Упорядоченный набор нескольких экземпляров одного типа дерева.

Тип дерева - один "корневой" тип записи и упорядоченный набор из нуля или более типов поддеревьев (каждое из которых является некоторым типом дерева).

Пример схемы БД:



Пример базы данных:



Узел дерева - несколько упорядоченных записей, каждая запись может являться корнем своего собственного поддерева.

Манипулирование данными

- Найти указанное дерево БД (например, отдел 310);
- Перейти от одного дерева к другому;
- Перейти от одной записи к другой внутри дерева (например, от отдела - к первому сотруднику);
- Перейти от одной записи к другой в порядке обхода иерархии;
- Вставить новую запись в указанную позицию;
- Удалить/Изменить текущую запись.

Ограничения целостности

Никакой потомок не может существовать без своего родителя.

Сетевые системы

Структура сетевой БД

Основана на произвольных ориентированных графах и включает набора экземпляров каждого типа записи и набора экземпляров каждого типа связи.



Пример схемы БД:

Манипулирование данными

- Поиск конкретной записи в наборе однотипных записей;
- Перейти от предка к первому потомку по некоторой связи;
- Перейти к следующему потомку в некоторой связи;
- Перейти от потомка к предку по некоторой связи;
- Создать/Уничтожить/Модифицировать запись;
- Включить в связь/Исключить из связи/Переставить в другую связь

Ограничения целостности

в некоторых реализациях поддерживалась целостность по ссылкам.

Системы, основанные на инвертированных списках

Структуры данных

Двумерные таблицы, состоящие из строк – записей и столбцов – атрибутов. Строки упорядочены системой в некоторой физической последовательности.

Пример БД

| № | СЛУ_ИМЯ | СЛУ_НОМЕР | СЛУ_СТАТУС | СЛУ_ЗАРПЛАТА | СЛУ_ОТД_№ | СЛУ_РУК |
|---|---------|-----------|------------|--------------|-----------|---------|
| 0 | Иванов | 33 | Да | 100000 | 10 | Астахов |
| 1 | Астахов | 22 | Да | 200000 | 10 | Астахов |
| 2 | Яковлев | 11 | Нет | 300000 | 20 | Яковлев |
| 3 | Петров | 50 | Да | 200000 | 20 | Яковлев |

Индексы ускоряющие доступ

СЛУ_ИМЯ СЛУ_НОМЕР СЛУ_ОТД_НОМЕР

| | | |
|---|---|---|
| 1 | 2 | 0 |
| 0 | 1 | 1 |
| 3 | 0 | 3 |
| 2 | 3 | 2 |

Манипулирование данными

- прямые поисковые операторы;
- операторы, находящие запись в терминах относительной позиции от предыдущей записи.
- операторы над адресуемыми записями

Ограничения целостности

В некоторых системах поддерживались ограничения уникальности значений некоторых атрибутов.

Достоинства и недостатки ранних дореляционных СУБД

Достоинства

- развитые средства управления данными во внешней памяти на низком уровне, и как следствие возможность разработки высокоэффективных прикладных систем.

Недостатки

- необходимость знания прикладной системой физической организации БД;
- отсутствие или сильная ограниченность правил определения целостности БД.