

2. СРЕДА БАЗЫ ДАННЫХ

2.1. Трехуровневая архитектура ANSI-SPARC

2

- Модель ANSI/SPARC представляет собой основу для понимания некоторых функциональных особенностей СУБД.
- Цель трехуровневой архитектуры заключается в отделении пользовательского представления базы данных от ее физического представления.



2.1.1. Внешний уровень

3

Внешний уровень. *Представление базы данных с точки зрения пользователей. Этот уровень описывает ту часть базы данных, которая относится к каждому пользователю.*

- Внешний уровень состоит из нескольких различных внешних представлений базы данных.
- Каждый пользователь имеет дело с представлением "реального мира", выраженным в наиболее удобной для него форме.
- Внешнее представление содержит только те сущности, атрибуты и связи "реального мира", которые интересны пользователю.

2.1.2. Концептуальный уровень

4

***Концептуальный уровень.** Обобщающее представление базы данных. Этот уровень описывает то, какие данные хранятся в базе данных, а также связи, существующие между ними.*

- Концептуальный уровень.- промежуточный уровень в трехуровневой архитектуре.
- На концептуальном уровне представлены следующие компоненты:
 - все сущности, их атрибуты и связи;
 - накладываемые на данные ограничения;
 - семантическая информация о данных;
 - информация о мерах обеспечения безопасности и поддержки целостности данных.
- Этот уровень не содержит никаких сведений о методах хранения данных.

2.1.3. Внутренний уровень

5

***Внутренний уровень.** Физическое представление базы данных в компьютере. Этот уровень описывает, как информация хранится в базе данных.*

- **Предназначен** для достижения оптимальной производительности и обеспечения экономного использования дискового пространства.
- **Содержит** описание структур данных и организации отдельных файлов, используемых для хранения данных на запоминающих устройствах.
- На внутреннем уровне **хранится** следующая информация:
 - распределение дискового пространства для хранения данных и индексов;
 - описание подробностей сохранения записей (с указанием реальных размеров сохраняемых элементов данных);
 - сведения о размещении записей;
 - сведения о сжатии данных и выбранных методах их шифрования.
- Ниже внутреннего уровня находится **физический уровень** (physical level), который контролируется операционной системой, но под управлением СУБД.

2.1.4. Схемы, отображения и экземпляры

6

- Общее описание базы данных называется **схемой базы данных**.
- Существуют три различных типа схем базы данных, которые определяются в соответствии с уровнями абстракции трехуровневой архитектуры:
 - несколько внешних схем или подсхем, которые соответствуют разным представлениям данных;
 - концептуальная схема;
 - внутренняя схема.
- Для каждой базы данных существует только одна концептуальная и одна внутренняя схема.
- СУБД отвечает за установление соответствия между этими тремя типами схем, а также за проверку их непротиворечивости.
- СУБД должна обеспечивать, чтобы каждую внешнюю схему можно было вывести на основе концептуальной схемы.
- Для установления соответствия между любыми внешней и внутренней схемами СУБД должна использовать информацию из концептуальной схемы.

Отображения

7

- Концептуальная схема связана с внутренней схемой посредством **концептуально внутреннего отображения** оно позволяет СУБД:
 - **найти** фактическую запись или набор записей на физическом устройстве хранения, которые образуют логическую запись в концептуальной схеме, с учетом любых ограничений, установленных для операций, выполняемых над данной логической записью.
 - **обнаружить** любые различия в именах объектов, именах атрибутов, порядке следования атрибутов, их типах данных и т.д.
- Каждая внешняя схема связана с концептуальной схемой с помощью **концептуально внешнего отображения**. С его помощью СУБД может отображать имена поль - зовательского представления на соответствующую часть концептуальной схемы.

Отличие БД от её описания

8

Описанием базы данных является схема базы данных.

- Схема создается в процессе проектирования базы данных, изменяется достаточно редко.
- Содержащаяся в базе данных информация может меняться часто (например, всякий раз при вставке новых сведений)
- Совокупность информации, хранящейся в базе данных в любой определенный момент времени, называется **состоянием базы данных**.
- Одной и той же схеме базы данных может соответствовать множество ее различных состояний.
- Схема базы данных иногда именуется **содержанием базы данных**, а ее состояние — **детализацией**.

2.1.5. Независимость от данных

- Основным назначением трехуровневой архитектуры является обеспечение не зависимости от данных, которая означает, что изменения на нижних уровнях не влияют на верхние уровни.
- Различают два типа независимости от данных:
 - логическую;
 - физическую.
- **Логическая независимость от данных.** Логическая независимость от данных означает полную защищенность внешних схем от изменений, вносимых в концептуальную схему.
- **Физическая независимость от данных.** Физическая независимость от данных означает защищенность концептуальной схемы от изменений, вносимых во внутреннюю схему.

Реализация независимости от данных в трехуровневой архитектуре ANSI/SPARC

10



2.2. Языки баз данных

- Внутренний язык СУБД для работы с данными состоит из двух частей:
 - языка определения данных (Data Definition Language — DDL) используется для определения схемы базы данных;
 - языка манипулирования данными (Data Manipulation Language — DML) используется для чтения и обновления данных, хранимых в базе..
- Эти языки называются подъязыками данных- в них отсутствуют конструкции для выполнения всех вычислительных операций, обычно используемых в языках программирования высокого уровня, таких как условные операторы или операторы цикла.
- Во многих СУБД предусмотрена возможность внедрения операторов подъязыка данных в программы, написанные на таких языках программирования высокого уровня, как COBOL, Fortran, Pascal, Ada, C, C++, Java или Visual Basic.
- В этом случае язык высокого уровня принято называть базовым языком (host language).

2.2.1. Язык определения данных — DDL

- **Язык DDL.** Описательный язык, который позволяет АБД или пользователю описать и именовать сущности и атрибуты, необходимые для работы некоторого приложения, а также связи, имеющиеся между различными сущностями, кроме того, указать ограничения целостности и защиты.
- **Схема базы данных** состоит из набора определений, выраженных на специальном языке определения данных — DDL.
- Язык DDL используется как для определения новой схемы, так и для модификации уже существующей.
- Результатом компиляции DDL-операторов является набор таблиц, хранимый в особых файлах, называемых системным каталогом.
- В **системном каталоге** интегрированы метаданные — т.е. данные, которые описывают объекты базы данных, а также позволяют упростить способ доступа к ним и управления ими.
- **Метаданные** включают определения записей, элементов данных, а также другие объекты, представляющие интерес для пользователей или необходимые для работы СУБД.

2.2.2. Язык управления данными —DML

13

- Язык DML. Язык, содержащий набор операторов для поддержки основных операций манипулирования содержащимися в базе данными.
- К операциям управления данными относятся:
 - вставка в базу данных новых сведений;
 - модификация сведений, хранимых в базе данных;
 - извлечение сведений, содержащихся в базе данных;
 - удаление сведений из базы данных.
- Часть непроцедурного языка DML, которая отвечает за извлечение данных, называется **языком запросов**.
- Существуют два типа языков DML:
 - процедурный ;
 - непроцедурный..
- Основное различие между ними заключается в том, что процедурные языки указывают то, как можно получить результат оператора языка DML, тогда как непроцедурные языки описывают то, какой результат будет получен.

Процедурные языки DML

14

- **Процедурный язык DML.** Язык, который позволяет сообщить системе о том, какие данные необходимы, и точно указать, как их можно извлечь.
- Языки DML сетевых и иерархических СУБД обычно являются процедурными

Непроцедурные языки DML

15

- **Непроцедурный язык DML.** Язык, который позволяет указать лишь то, какие данные требуются, но не то, как те следует извлекать.
- Реляционные СУБД в той или иной форме обычно включают поддержку непроцедурных языков манипулирования данными — чаще всего это язык структурированных запросов SQL (Structured Query Language) или язык запросов по образцу QBE (Query-by-Example).

2.2.3. Языки 4GL

16

- Аббревиатура 4GL представляет собой сокращенный английский вариант написания термина язык четвертого поколения (Fourth-Generation Language).
- Языки третьего поколения - процедурные;
- Языки 4GL - непроцедурные (поскольку пользователь определяет, что должно быть сделано, но не сообщает, как именно должен быть достигнут желаемый результат.)
- Выделяют следующие типы языков четвертого поколения:
 - **языки представления информации**, например языки запросов или генераторы отчетов;
 - **специализированные языки**, например языки электронных таблиц и баз данных;
 - **генераторы приложений**, которые при создании приложений обеспечивают определение, вставку, обновление или извлечение сведений из базы данных;
 - **языки генерации кода приложений**.

2.3. Модели данных и концептуальное моделирование

17

Модель данных. *Интегрированный набор понятий для описания и обработки данных, связей между ними и ограничений, накладываемых на данные в некоторой организации.*

- **Модель** является представлением "реального мира" объектов и событий, а также существующих между ними связей.
- **Модель данных** можно рассматривать как сочетание трех компонентов:
 - *Структурная часть*, т.е. набор правил, по которым может быть построена база данных.
 - *Управляющая часть*, определяющая типы допустимых операций с данными (сюда относятся операции обновления и извлечения данных, а также операции изменения структуры базы данных).
 - *Набор (необязательный) ограничений* поддержки целостности данных, гарантирующих корректность используемых данных.

2.3.1. Объектные модели данных

18

- При создании объектных моделей данных используются понятия:
 - сущность;
 - атрибут;
 - связь.
- **Сущность** — это отдельный элемент деятельности организации (сотрудник или клиент, место или вещь, понятие или событие), который должен быть представлен в базе данных.
- **Атрибут** — это свойство, которое описывает некоторый аспект объекта и значение которого следует зафиксировать.
- **Связь** — является ассоциативным отношением между сущностями.
- Некоторые наиболее общие типы объектных моделей данных:
 - Модель типа "сущность-связь", или ER-модель (Entity-Relationship model).
 - Семантическая модель.
 - Функциональная модель.
 - Объектно-ориентированная модель.

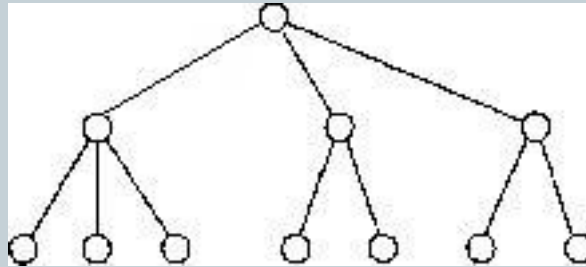
2.3.2. Модели данных на основе записей

- В модели на основе записей база данных состоит из нескольких записей фиксированного формата, которые могут иметь разные типы.
- Каждый тип записи определяет фиксированное количество полей, каждое из которых имеет фиксированную длину.
- Существуют три основных типа логических моделей данных на основе записей:
 - реляционная модель данных (relational data model);
 - сетевая модель данных (network data model) ;
 - иерархическая модель данных (hierarchical data model).

Иерархическая модель данных

20

- В иерархической модели узел может иметь только одного родителя.
- Иерархическая модель может быть представлена как древовидный граф с записями в виде узлов (которые также называются сегментами) и множествами в виде ребер. его родителя.



- В иерархической модели связи между данными можно описать с помощью упорядоченного графа (или дерева).
- Для описания структуры (схемы) иерархической БД на некотором языке программирования используется тип данных "дерево".
- Тип "дерево" является составным. Он включает в себя подтипы ("поддережья"), каждый из которых, в свою очередь, является типом "дерево". Каждый из типов "дерево" состоит из одного "корневого" типа и упорядоченного набора (возможно пустого)
- Самой распространенной иерархической СУБД является система IMS корпорации IBM

Иерархическая модель данных

21

● Пример иерархической БД

Список составных частей изделия по своей природе является иерархической структурой



Иерархическая база данных, содержащая информацию о составных частях машины.

Иерархическая модель данных

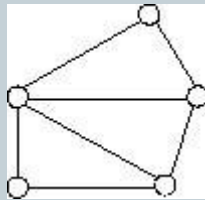
22

- Получая доступ к информации, содержащейся в базе данных, программа могла:
 - найти конкретную деталь (левую дверь) по ее номеру;
 - перейти «вниз» к первому потомку (ручка двери);
 - перейти «вверх» к предку (кузов);
 - перейти «в сторону» к другому потомку (правая дверь)

Сетевая модель

23

- Сетевая модель данных позволяет отображать разнообразные взаимосвязи элементов данных в виде произвольного графа, обобщая тем самым иерархическую модель данных



Представление связей в сетевой модели

- Для описания схемы сетевой БД используется две группы типов:
 - "запись"
 - "связь".
- Тип "связь" определяется для двух типов "запись": предка и потомка.
- Переменные типа "связь" являются экземплярами связей.
- Сетевая БД состоит из набора записей и набора соответствующих связей.
- На формирование связи особых ограничений не накладывается. Если в иерархических структурах запись-потомок могла иметь только одну запись-предка, то в сетевой модели данных запись-потомок может иметь произвольное число записей-предков (сводных родителей).

Сетевая модель

24

Пример схемы сетевой БД



- **Физическое** размещение данных в базах сетевого типа может быть организовано практически теми же методами, что и в иерархических базах данных.
- К числу важнейших **операций манипулирования данными** баз сетевого типа можно отнести следующие: 1. поиск записи в БД; 2. переход от предка к первому потомку; 3. переход от потомка к предку; 4. создание новой записи; 4. удаление текущей записи; 5. обновление текущей записи; 6. включение записи в связь; 7. исключение записи из связи; 8. изменение связей и т. д.
- **Достоинством** сетевой модели данных является возможность эффективной реализации по показателям затрат памяти и оперативности. В сравнении с иерархической моделью сетевая модель предоставляет большие возможности в смысле допустимости образования произвольных связей.

Сетевая модель

25

- **Недостатком** сетевой модели данных является высокая сложность и жесткость схемы БД, построенной на ее основе, а также сложность для понимания и выполнения обработки информации в БД обычным пользователем. Кроме того, в сетевой модели данных ослаблен контроль целостности связей вследствие допустимости установления произвольных связей между записями.
- Системы на основе сетевой модели не получили широкого распространения на практике.
- Наиболее известными сетевыми СУБД являются следующие: IDMS, db VistaIII, СЕТЬ, СЕТОР и КОМПАС

Реляционная модель

26

- **Реляционная модель данных** предложена сотрудником фирмы IBM Эдгаром Коддом и основывается на понятии отношение (relation)
- **Отношение** представляет собой множество элементов, называемых кортежами.. Наглядной формой представления отношения является привычная для человеческого восприятия двумерная таблица.
- Таблица имеет:
 - строки (записи)
 - столбцы (колонки).
- Каждая строка таблицы имеет одинаковую структуру и состоит из полей. Строкам таблицы соответствуют кортежи, а столбцам - атрибуты отношения.
- С помощью одной таблицы удобно описывать простейший вид связей между данными, а именно: деление одного объекта (явления, сущности, системы и проч.), информация о котором хранится в таблице, на множество подобъектов, каждому из которых соответствует строка или запись таблицы.
- Каждый из подобъектов имеет одинаковую структуру или свойства, описываемые соответствующими значениями полей записей.

- Поскольку в рамках одной таблицы не удастся описать более сложные логические структуры данных из предметной области, применяют **связывание** таблиц
- **Достоинство** реляционной модели данных:
 - простота;
 - понятность;
 - удобство физической реализации на ЭВМ.

Именно простота и понятность для пользователя явились основной причиной их широкого использования. Проблемы же эффективности обработки данных этого типа оказались технически вполне разрешимыми.

Недостатки реляционной модели :

отсутствие стандартных средств идентификации отдельных записей;
сложность описания иерархических и сетевых связей.

2.3.3. Физические модели данных

28

- Физические модели данных описывают то, как данные хранятся в компьютере, представляя информацию о структуре записей, их упорядоченности и существующих путях доступа.
- Физических моделей данных не так много, как логических, а самыми популярными среди них являются :
 - обобщающая модель (unifying model) ;
 - модель памяти кадров (frame memory).

2.3.4. Концептуальное моделирование

- Концептуальная схема является "сердцем" базы данных.
- Она поддерживает все внешние представления, а сама поддерживается средствами внутренней схемы.
- Именно концептуальная схема призвана быть полным и точным представлением требований к данным некоторого предприятия. В противном случае определенная часть информации о предприятии будет упущена или искажена.
- Концептуальное моделирование (или концептуальное проектирование) базы данных — это процесс конструирования модели использования информации на некотором предприятии.
- Этот процесс не зависит от таких подробностей реализации, как используемая СУБД, прикладные программы, языки программирования или любые другие вопросы физической организации информации.
- Концептуальные модели в литературе иногда также называют **логическими моделями**.

2.4. Функции СУБД

30

- Кодд предложил перечень из восьми служб, которые должны быть реализованы в любой полномасштабной СУБД.
- 1. Хранение, извлечение и обновление данных**
СУБД должна предоставлять пользователям возможность сохранять, извлекать и обновлять данные в базе данных.
- 2. Каталог, доступный конечным пользователям**
СУБД должна иметь доступный конечным пользователям каталог, в котором хранится описание элементов данных.
- Ключевой особенностью архитектуры ANSI-SPARC является наличие интегрированного системного каталога с данными о схемах, пользователях, приложениях и т.д.
- Предполагается, что каталог доступен как пользователям, так и функциям СУБД
- Обычно в системном каталоге хранятся следующие сведения:
 - Имена;
 - типы и размеры элементов данных;
 - накладываемые на данные ограничения поддержки целостности;

- имена пользователей, которым предоставлено право доступа к данным;
- внешняя, концептуальная и внутренняя схемы и отображения между ними
- статистические данные, например частота транзакций и счетчики обращений к объектам базы данных.

3. Поддержка транзакций

СУБД должна иметь механизм, который гарантирует выполнение либо всех операций обновления данной транзакции, либо ни одной из них.

- Транзакция представляет собой набор действий, выполняемых отдельным пользователем или прикладной программой с целью доступа или изменения содержимого базы данных.
- *Пример транзакции* - удаление сведений о сотруднике из базы данных и передача ответственности за все курируемые им объекты недвижимости другому сотруднику.

В этом случае в базу данных потребуется внести сразу несколько изменений. Если во время выполнения транзакции произойдет сбой, например из-за выхода из строя компьютера, база данных попадает в противоречивое состояние, поскольку некоторые изменения уже будут внесены, а остальные — еще нет. Поэтому все частичные изменения должны быть отменены для возвращения базы данных в прежнее, непротиворечивое состояние.

4. Службы управления параллельной работой

СУБД должна иметь механизм, который гарантирует корректное обновление базы данных при параллельном выполнении операций обновления многими пользователями.

- Одна из основных целей создания и использования СУБД заключается в том, чтобы множество пользователей могло осуществлять параллельный доступ к совместно обрабатываемым данным.
- Параллельный доступ сравнительно просто организовать, если все пользователи выполняют только чтение данных, поскольку в этом случае они не могут помешать друг другу.
- Когда два или больше пользователей одновременно получают доступ к базе данных, конфликт с нежелательными последствиями легко может возникнуть, например, если хотя бы один из них попытается обновить данные.
- СУБД должна гарантировать, что при одновременном доступе к базе данных многих пользователей подобных конфликтов не произойдет.

Службы СУБД

33

5. Службы восстановления

СУБД должна предоставлять средства восстановления базы данных на случай какого-либо ее повреждения или разрушения.

- Сбой может произойти в результате выхода из строя системы или запоминающего устройства, ошибки аппаратного или программного обеспечения, которые могут привести к останову СУБД.
- Во всех этих случаях СУБД должна предоставить механизм восстановления базы данных и возврата ее к непротиворечивому состоянию.

6. Службы контроля доступа к данным

СУБД должна иметь механизм, гарантирующий возможность доступа к базе данных только санкционированных пользователей.

- Термин безопасность относится к защите базы данных от преднамеренного или случайного несанкционированного доступа. Предполагается, что СУБД обеспечивает механизмы подобной защиты данных.

7. Поддержка обмена данными

СУБД должна обладать способностью к интеграции с коммуникационным программным обеспечением.

- СУБД должны поддерживать работу в локальной сети, чтобы вместо нескольких разрозненных баз данных для каждого отдельного пользователя можно было бы установить одну централизованную базу данных и использовать ее как общий ресурс для всех существующих пользователей.
- При этом предполагается, что не база данных должна быть распределена в сети, а удаленные пользователи должны иметь возможность доступа к централизованной базе данных. Такая организация работы называется **распределенной обработкой**.

Службы СУБД

35

8. Службы поддержки целостности данных

СУБД должна обладать инструментами контроля за тем, чтобы данные и их изменения соответствовали заданным правилам.

- Целостность базы данных означает корректность и непротиворечивость хранимых данных.
- Целостность может рассматриваться как еще один тип защиты базы данных.
- Целостность связана с качеством самих данных.
- Целостность обычно выражается в виде ограничений или правил сохранения непротиворечивости данных, которые не должны нарушаться в базе.

9. Вспомогательные службы

СУБД должна предоставлять некоторый набор различных вспомогательных служб.

- Вспомогательные утилиты обычно предназначены для оказания помощи АБД в эффективном администрировании базы данных.

2.5. Компоненты СУБД

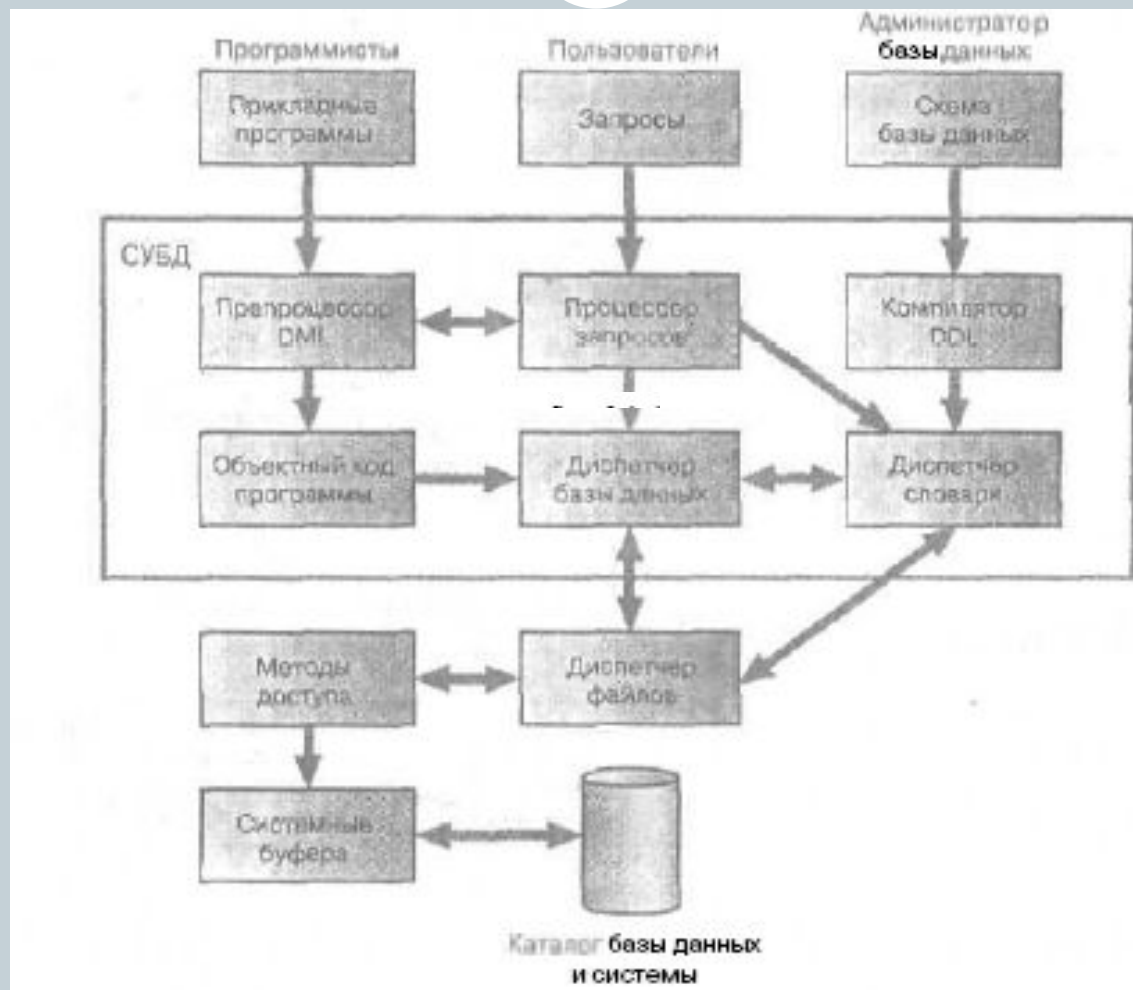
36

СУБД является сложным видом программного обеспечения, предназначенным для предоставления ресурсов.

- СУБД состоит из нескольких программных компонентов (модулей), каждый из которых предназначен для выполнения специфической операции.
- Основные программные компоненты среды СУБД:
 - **Процессор запросов** - основной компонент СУБД, который преобразует запросы в последовательность низкоуровневых команд для диспетчера базы данных.
 - **Диспетчер базы данных** - компонент взаимодействует с запущенными пользователями, прикладными программами и запросами.
 - **Диспетчер файлов**. Манипулирует предназначенными для хранения данных файлами и отвечает за распределение доступного дискового пространства.
 - **Препроцессор языка DML**. Этот модуль преобразует внедренные в прикладные программы DML-операторы в вызовы стандартных функций базового языка.
 - **Компилятор языка DDL**. Компилятор языка DDL преобразует DDL-команды в набор таблиц, содержащих метаданные.
 - **Диспетчер словаря**. Диспетчер словаря управляет доступом к системному каталогу и обеспечивает работу с ним.

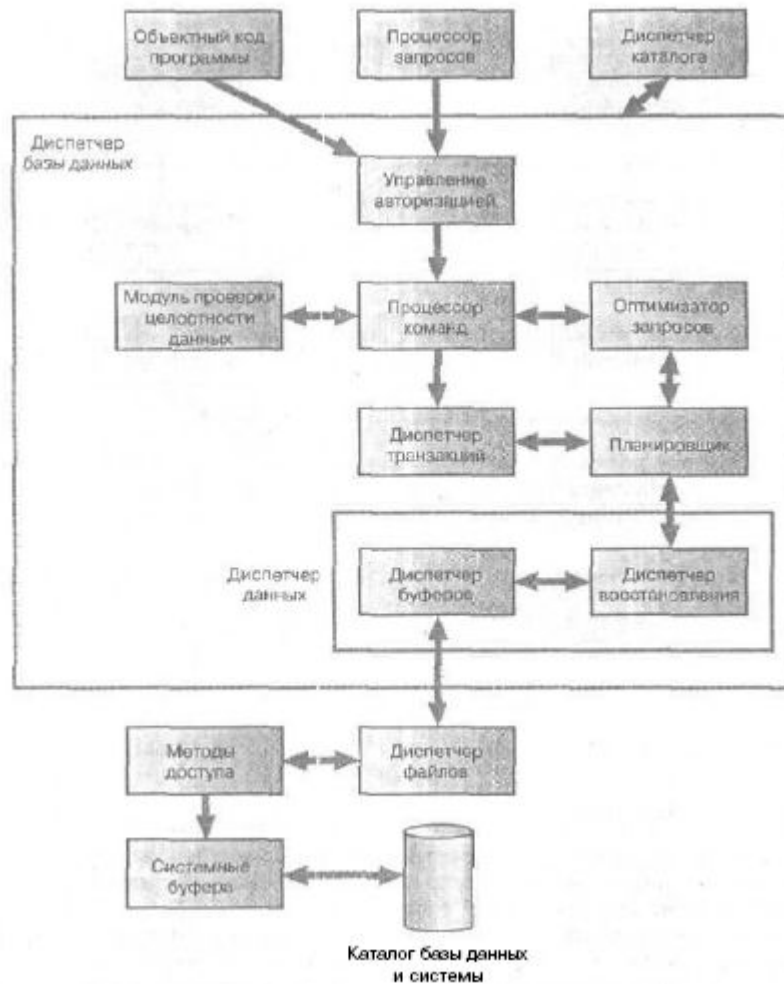
2.6 Основные компоненты типичной системы управления базами данных

37



Компоненты диспетчера базы данных

38



Компоненты диспетчера базы данных

39

- **Модуль контроля прав доступа.** Модуль проверяет наличие у данного пользователя полномочий для выполнения затребованной операций.
- **Процессор команд.** После проверки полномочий пользователя для выполнения затребованной операции управление передается процессору команд.
- **Средства контроля целостности.** В случае операций, которые изменяют содержимое базы данных, средства контроля целостности выполняют проверку того, удовлетворяет ли затребованная операция всем установленным ограничениям поддержки целостности данных (например, требованиям, установленным для ключей).
- **Оптимизатор запросов.** Этот модуль определяет оптимальную стратегию выполнения запроса.
- **Диспетчер транзакций.** Этот модуль осуществляет требуемую обработку операций, поступающих в процессе выполнения транзакций.
- **Планировщик.** Этот модуль отвечает за бесконфликтное выполнение параллельных операций с базой данных. Он управляет относительным порядком выполнения операций, затребованных в отдельных транзакциях.

Компоненты диспетчера базы данных

40

- **Диспетчер восстановления.** Этот модуль гарантирует восстановление базы данных до непротиворечивого состояния при возникновении сбоев, отвечает за фиксацию и отмену результатов выполнения транзакций.
- **Диспетчер буферов.** Этот модуль отвечает за перенос данных между оперативной памятью и вторичным запоминающим устройством.
Например, жестким диском или магнитной лентой.
- **Диспетчер восстановления.**