

Основные понятия и определения

Термин «база данных» (database) был введен в обиход в области вычислительной техники примерно в 1962 году. Этот термин страдает от обилия различных интерпретаций. Примеры определения термина «баз данных» из авторитетных источников:

База данных (БД) - совокупность данных, организованных по определенным правилам, предусматривающая общие принципы описания, хранения и манипулирования данными, независимая от прикладных программ. Является информационной моделью предметной области.

[К. Дейт, Введение в системы баз данных, 1980]

База данных – совокупность хранимых **операционных** данных, используемых прикладными системами некоторого предприятия.

[К. Дейт, Введение в системы баз данных, 6-е издание, 2000]

База данных состоит из некоторого набора **постоянных** данных, которые используются прикладными системами для какого-либо предприятия.

[К. Дейт, Введение в системы баз данных, 7-е издание, 2001]

База данных – это некоторый набор **перманентных** (постоянных) данных, используемых прикладными системами какого-либо предприятия.

1. Операционные данные – мир **OnLine Transaction Processing** (Оперативная обработка транзакций) или **OLTP**.
2. Перманентные данные – мир **OnLine Analytical Processing** (Аналитическая обработка в реальном времени) или **OLAP**. OLAP используется в **системах поддержки принятия решений** (Decision Support System – DSS) и **управленческих информационных системах** (Executive Information System – EIS). Базу данных для поддержки принятия решения обычно называют **хранилищем данных** (data warehouse).

[Джеффри Д. Ульман, Дженнифер Уидом, Введение в системы баз данных, 2000]

По сути дела БД - это просто множество информации, существующее долгое время, часто в течении многих лет. Обычно термином "база данных" обозначается множество данных, управляемое СУБД, или просто СБД.

[Д. Кренке, Теория и практика построения баз данных, 2003]

База данных – это **самодокументированное** собрание **интегрированных** записей. Важно понять обе части этого определения.

1. БД является самодокументированной (self describing): она содержит описание собственной структуры. Это описание называется **словарем данных** (data dictionary), **каталогом данных** (data directory) или **метаданными** (metadata).
2. БД – это собрание интегрированных записей: она содержит:
 - a. **Файлы данных**,
 - b. **Метаданные**,
 - c. **Индексы** (indexes), которые представляют связи между данными, а также служат для повышения производительности приложений базы данных.
 - d. Может содержать **метаданные приложений** (application metadata).
3. БД является **информационной моделью** пользовательской модели (user model) предметной области.

База данных - совокупность взаимосвязанных данных некоторой предметной области, хранимых в памяти ЭВМ и организованных таким образом, что эти данные могут быть использованы для решения многих задач многими пользователями. Основные требования к организации данных:

1. **Неизбыточность данных** - каждое данное присутствует в БД в единственном экземпляре.
2. **Совместное использование данных** многими пользователями.
3. **Эффективность доступа к БД** - высокое быстродействие, т. е. малое время отклика на запрос.
4. **Целостность данных** - соответствие имеющейся в БД информации её внутренней логике, структуре и всем явно заданным правилам.
5. **Безопасность данных** – защита данных от преднамеренного или непреднамеренного искажения или разрушения данных.
6. **Восстановление данных** после программных и аппаратных сбоев.
7. **Независимость данных** от прикладных программ.

Существует множество других определений, отражающих скорее субъективное мнение тех или иных авторов о том, что означает база данных в их понимании, однако общепризнанная единая формулировка отсутствует.

Система управления базами данных (СУБД) - приложение, обеспечивающее создание, хранение, обновление и поиск информации в базах данных.

Система баз данных (или банк данных в отечественной терминологии) - совокупность одной или нескольких баз данных и комплекса информационных, программных и технических средств, обеспечивающих накопление, обновление, корректировку и многоаспектное использование данных в интересах пользователей. (См. Григорьев Ю.А., Ревунков Г.И. Банки данных. – М.: МГТУ, 2002.)

Состав системы баз данных

- Одна или несколько баз данных.
- Аппаратное обеспечение.
- Программное обеспечение.
 - СУБД.
 - Прочие компоненты: утилиты, средства разработки приложений, средства проектирования, генераторы отчетов, диспетчер транзакций и т. д.
- Пользователи.
 - Прикладные программисты, которые отвечают за написание прикладных пакетных и интерактивных программ.
 - Конечные пользователи, которые получают доступ к базе данных, применяя одно из интерактивных приложений, интерфейс, интегрированный в программное обеспечение самой СУБД, командный интерфейс, реализуемый процессором языка запросов или некомандный интерфейс, основанный на меню и формах.
 - Администраторы базы данных.

Основные функции СУБД

- **Непосредственное управление данными во внешней памяти.** В развитых СУБД пользователи не обязаны знать, как организованы файлы данных.
- **Управление буферами оперативной памяти.** Для увеличения скорости обмена данными с внешней памятью используется буферизация данных в оперативной памяти. В развитых СУБД поддерживается собственный набор буферов оперативной памяти с собственной дисциплиной замены буферов, даже если операционная система производит общесистемную буферизацию. Существует даже отдельное направление СУБД, которое ориентировано на постоянное присутствие в оперативной памяти всей БД. Это направление основывается на предположении, что объем оперативной памяти компьютеров настолько велик, что можно не беспокоиться о буферизации.
- **Управление транзакциями.** Транзакция – это последовательность операций над данными, рассматриваемая СУБД как единое целое. Реализуется принцип «либо все, либо ничего». Поддержание механизма транзакций является обязательным условием даже однопользовательских СУБД. Но понятие транзакции гораздо более важно в многопользовательских СУБД. То свойство, что каждая транзакция начинается при целостном состоянии БД и оставляет это состояние целостным после своего завершения, делает удобным использование понятия транзакции как единицы активности пользователя по отношению к БД.
- **Журнализация.** Одним из основных требований к СУБД является требование надежного 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. В любом случае для восстановления БД нужно располагать некоторой дополнительной информацией. Наиболее распространенным методом поддержания такой избыточной информации является ведение журнала изменений БД. Журнал - это особая часть БД, недоступная пользователям СУБД (иногда поддерживаются две копии журнала, располагаемые на разных физических носителях), в которую поступают записи обо всех изменениях основной части БД.
- **Поддержка языков БД.** Для работы с БД используются специальные языки, называемые языками баз данных. В ранних СУБД (иерархических и сетевых) поддерживалось несколько специализированных по своим функциям языков. В современных СУБД (реляционных) поддерживается язык SQL (Structured Query Language), содержащий все необходимые средства для работы с БД, начиная от ее создания, и обеспечивающий базовый пользовательский интерфейс.

Основные компонентами СУБД

- **Ядро,** которое отвечает за управление данными во внешней и оперативной памяти и журнализацию.
- **Процессор языка базы данных,** обеспечивающий оптимизацию запросов на извлечение и изменение данных, и создание, как правило, машинно-независимого исполняемого внутреннего кода,
- **Подсистема поддержки времени исполнения,** которая интерпретирует программы манипуляции данными, создающие пользовательский интерфейс с СУБД.
- **Сервисные программы** (внешние утилиты), обеспечивающие ряд дополнительных возможностей по обслуживанию информационной системы.

Классификация СУБД

Классификация СУБД может выполняться по нескольким признакам:

1. По модели данных:

- a. Дореляционные**
 - i. Инвертированные списки (файлы)**
 - ii. Иерархические**
 - iii. Сетевые**
- b. Реляционные**
- c. Постреляционные**

БД на основе инвертированных списков представляет собой совокупность файлов, содержащих записи (таблиц). Для записей в файле определен некоторый порядок, диктуемый физической организацией данных. Для каждого файла может быть определено произвольное число других упорядочений на основании значений некоторых полей записей (инвертированных списков). Обычно для этого используются индексы. В такой модели данных отсутствуют ограничения целостности как таковые. Все ограничения на возможные экземпляры БД задаются теми программами, которые работают с БД. Одно из немногих ограничений, которое все-таки может присутствовать - это ограничение, задаваемое уникальным индексом.

Иерархическая модель БД состоит из объектов с указателями от родительских объектов к потомкам, соединяя вместе связанную информацию. Иерархические БД могут быть представлены как дерево.

К основным понятиям сетевой модели БД относятся: элемент (узел), связь. Узел — это совокупность атрибутов данных, описывающих некоторый объект. Сетевые БД могут быть представлены в виде графа. В сетевой БД логика процедуры выборки данных зависит от физической организации этих данных. Поэтому эта модель не является полностью независимой от приложения. Другими словами, если необходимо изменить структуру данных, то нужно изменить и приложение.

Реляционная модель данных включает следующие компоненты:

- **Структурный** (данные в базе данных представляют собой набор отношений),
- **Целостностный** (отношения (таблицы) отвечают определенным условиям целостности),
- **Манипуляционный** (манипулирование отношениями осуществляется средствами реляционной алгебры и/или реляционного исчисления). Кроме того, в состав реляционной модели данных включают теорию нормализации.

В 1985 году доктор Кодд сформулировал двенадцать правил, которым должна соответствовать настоящая реляционная база данных. Они являются полуофициальным определением понятия реляционная база данных. Строгое изложение теории реляционных баз данных (реляционной модели данных) в современном понимании можно найти в книге К. Дж. Дейта.

2. По архитектуре организации хранения данных

- a. локальные СУБД** (все части локальной СУБД размещаются на одном компьютере)
- b. распределенные СУБД** (части СУБД могут размещаться на двух и более компьютерах)

3. По способу доступа к БД (а может быть по местоположению БД):

- a. Файл-серверные.** При работе в архитектуре "файл-сервер" БД и приложение расположены на файловом сервере сети. Возможна многопользовательская работа с одной и той же БД, когда каждый пользователь со своего компьютера запускает приложение, расположенное на сетевом сервере. Тогда на компьютере пользователя запускается копия приложения. По каждому запросу к БД из приложения данные из таблиц БД перегоняются на компьютер пользователя, независимо от того, сколько реально нужно данных для выполнения запроса. После этого выполняется запрос. Каждый пользователь имеет на своем компьютере локальную копию данных, время от времени обновляемых из реальной БД, расположенной на сетевом сервере. При этом изменения, которые каждый пользователь вносит в БД, могут быть до определенного момента неизвестны другим пользователям, что делает актуальной задачу систематического обновления данных на компьютере пользователя из реальной базы данных. Другой актуальной задачей является блокирование записей, которые изменяются одним из пользователей; это необходимо для того, чтобы в это время другой пользователь не внес изменений в те же данные. В архитектуре "файл-сервер" вся тяжесть выполнения запросов к базе данных и управления целостностью базы данных ложится на приложение пользователя. База данных на сервере является пассивным источником данных.
- b. Клиент-серверные.** Клиент-сервер - сетевая архитектура, в которой устройства являются либо *клиентами*, либо *серверами*. Клиентом (front end) является запрашивающая машина, сервером (back end) - машина, которая отвечает на запрос. Оба термина (клиент и сервер) могут применяться как к физическим устройствам, так и к программному обеспечению. Характерной особенностью архитектуры "клиент-сервер" является перенос вычислительной нагрузки на сервер базы данных (sql-сервер) и максимальная разгрузка приложения клиента от вычислительной работы, а также существенное укрепление безопасности данных — как от злонамеренных, так и просто ошибочных изменений. БД в этом случае помещается на сетевом

сервере, как и в архитектуре "файл-сервер", однако прямого доступа к базе данных из приложений не происходит. Функцию прямого обращения к базе данных осуществляет СУБД. При этом ресурсы клиентского компьютера не участвуют в физическом выполнении запроса; клиентский компьютер лишь отправляет запрос к СУБД и получает результат, после чего интерпретирует его необходимым образом и представляет пользователю. Так как клиентскому приложению посылается результат выполнения запроса, по сети "путешествуют" только те данные, которые необходимы клиенту. В итоге снижается нагрузка на сеть. Кроме того, СУБД, если это возможно, оптимизирует полученный запрос таким образом, чтобы он был выполнен в минимальное время с наименьшими накладными расходами. При выполнении запросов сервером существенно повышается степень безопасности данных, поскольку правила целостности данных определяются в базе данных на сервере и являются едиными для всех приложений, использующих эту БД. Таким образом, исключается возможность определения противоречивых правил поддержания целостности. Аппарат транзакций, поддерживаемый СУБД, позволяет исключить одновременное изменение одних и тех же данных различными пользователями и предоставляет возможность откатов к первоначальным значениям при внесении в БД изменений, закончившихся аварийно.

- c. **Встраиваемые.** Встраиваемая СУБД — библиотека, которая позволяет унифицированным образом хранить большие объёмы данных на локальной машине. Доступ к данным может происходить через SQL либо через особые функции СУБД. Встраиваемые СУБД быстрее обычных клиент-серверных и не требуют установки сервера, поэтому востребованы в локальном ПО, которое имеет дело с большими объёмами данных.
- d. **Сервисно-ориентированные.** БД является хранилищем сообщений, промежуточных состояний, метаинформации об очередях сообщений и сервисах. Отправка сообщений в очередь и прием сообщений из очереди производится в одной транзакции с изменением данных, что обеспечивает транзакционную целостность системы. Так как очереди сообщений и данные хранятся и обрабатываются в базе единообразно, это обеспечивает гарантированную доставку и обработку сообщений в случае сбоев оборудования или питания с таким же успехом, как и прочих данных, хранящихся в той же базе данных. Кроме этого, в базе данных хранится информация о самих сервисах и обрабатываемых ими очередях сообщений, что обеспечивает восстановление после сбоя состояний не только данных и сообщений, но и настроек сервисов и очередей сообщений.
- e. **Прочие.** Пространственная (spatial database), Временная, или темпоральная (temporal database), Пространственно-временная (spatial-temporal database).