



Лекция 1

Введение в основы базы данных



Введение в базы данных

- Введение в системы баз данных
- Системы управления базами данных (СУБД)
- Тип базы данных
- Дизайн базы данных
- Рекомендации по проектированию базы данных
- Компонент систем баз данных

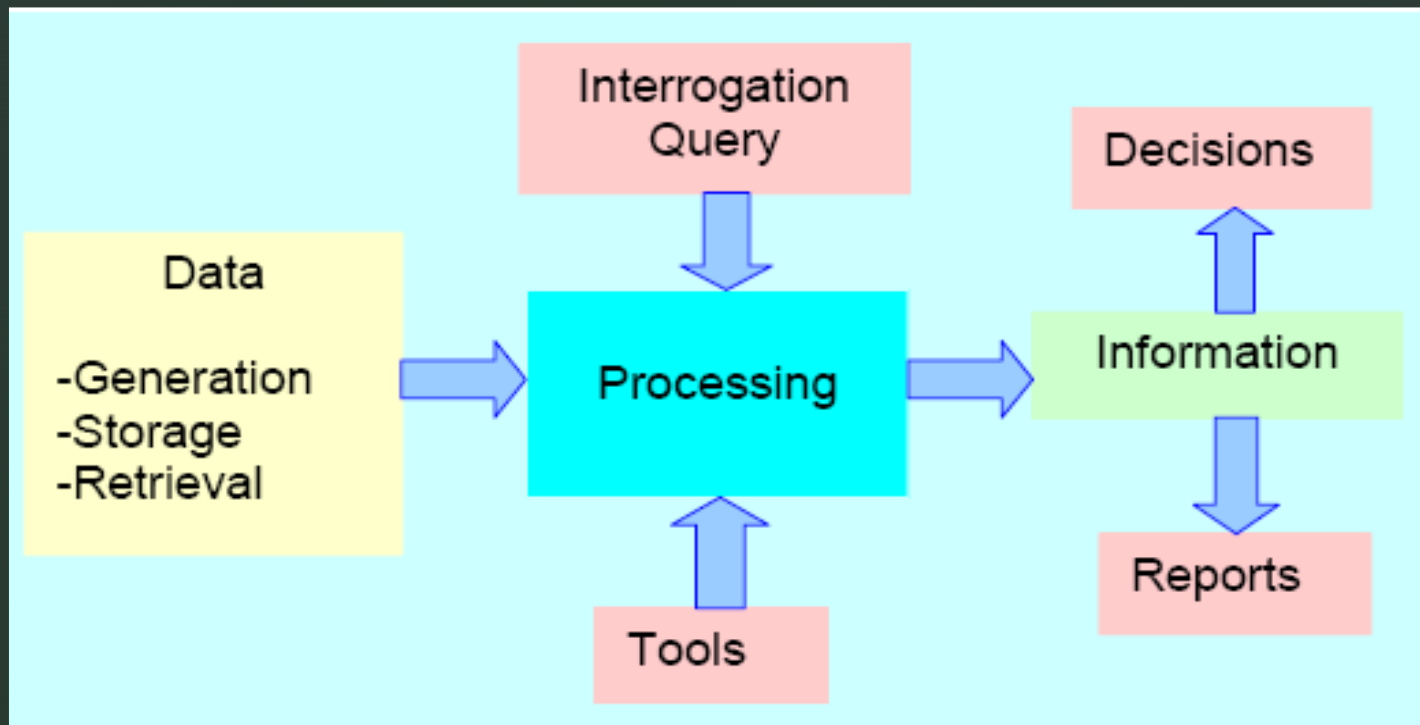
Введение в системы баз данных

- База данных - это программа, в которой хранится информация, относящаяся к определенной деятельности или цели.
- Примеры:
 - Банку необходимо хранить информацию, относящуюся к счетам клиентов.
 - Больнице необходимо хранить данные о пациентах и выданных лекарствах,
 - Университет должен вести учет своих студентов,
 - Бизнесу интернет-продаж необходимо ..

Введение в базы данных

- База данных включает инструменты, помогающие организовать и поддерживать данные.
- Инженер-проектировщик базы данных должен знать и уметь использовать эти инструменты, чтобы сделать базу данных эффективной и точной.
- В правильно спроектированной базе данных информация обновляется один раз, а все остальные ее появления в базе данных будут обновляться автоматически.
- Основным требованием к любой системе баз данных является способность эффективно генерировать, хранить и извлекать данные.

Основныe компоненты системы баз данных



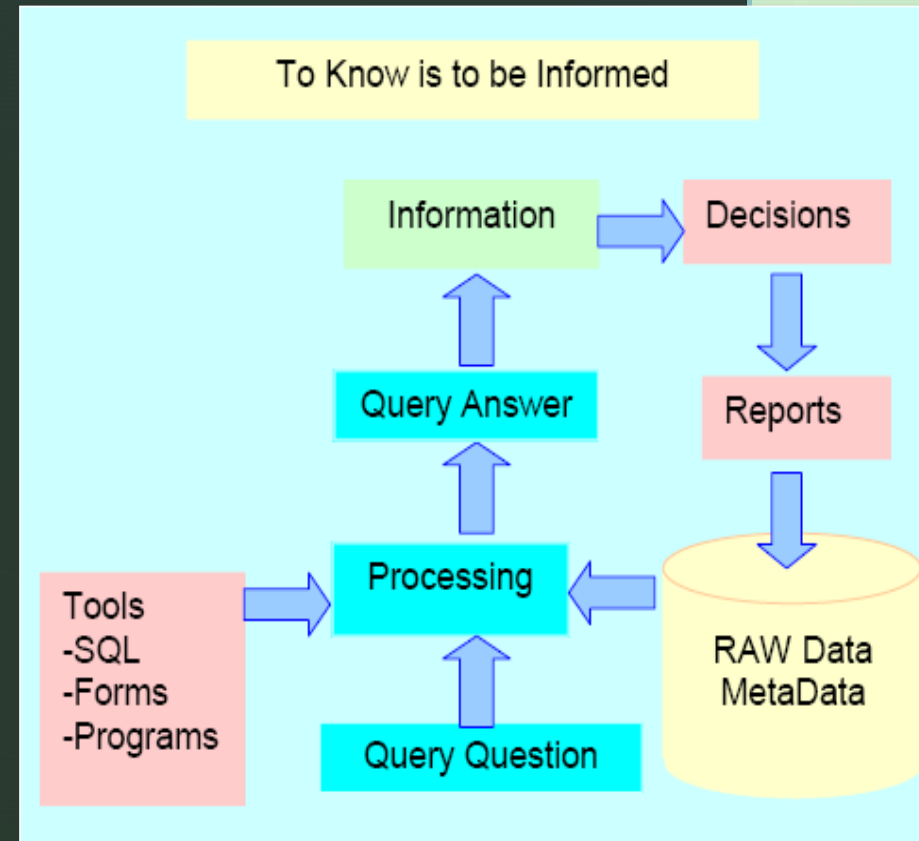
Для преобразования данных в полезную информацию необходим набор программных инструментов, SQL, Form и т. д.

Введение в базы данных

- Введение в системы баз данных
- Системы управления базами данных (СУБД)
- Тип базы данных
- Дизайн базы данных
- Рекомендации по проектированию базы данных
- Компонент систем баз данных

Системы управления базами данных (СУБД)

- СУБД - это набор программ, предоставляемых поставщиком, которые позволяют получать доступ к данным, фильтровать и в целом эффективно обрабатывать их, чтобы предоставить полезную информацию для пользователя.



▀ Системы управления базами данных (СУБД)

- В принципе, пользователь сгенерирует вопрос (запрос), а СУБД будет искать ответ, который будет возвращен пользователю.
- СУБД будет включать дополнительные инструменты для предоставления графических пользовательских интерфейсов и программ генерации отчетов для облегчения доставки информации пользователю.
- Он также предоставит инструменты для создания, редактирования, обработки и изменения данных, а также для программирования базы данных. SQL - это стандартный язык для доступа к реляционным базам данных.

Что такое система управления базами данных?

- Система управления базами данных (СУБД) - это программа, которая управляет базой данных:
 - Поддерживает язык доступа высокого уровня (например, SQL).
 - Приложение описывает доступ к базе данных на этом языке.
 - СУБД интерпретирует операторы языка для выполнения запрошенного доступа к базе данных.

Примеры SQL

- Example 1:

```
SELECT Course_Number  
FROM StudentRecords  
WHERE student_name = ' ';
```
- Example 2:

```
SELECT student_number  
FROM StudentRecords  
WHERE BEngCourse = '2388' AND  
average_mark >70;
```
- These examples show that SQL is a very useful tool for interrogating a database

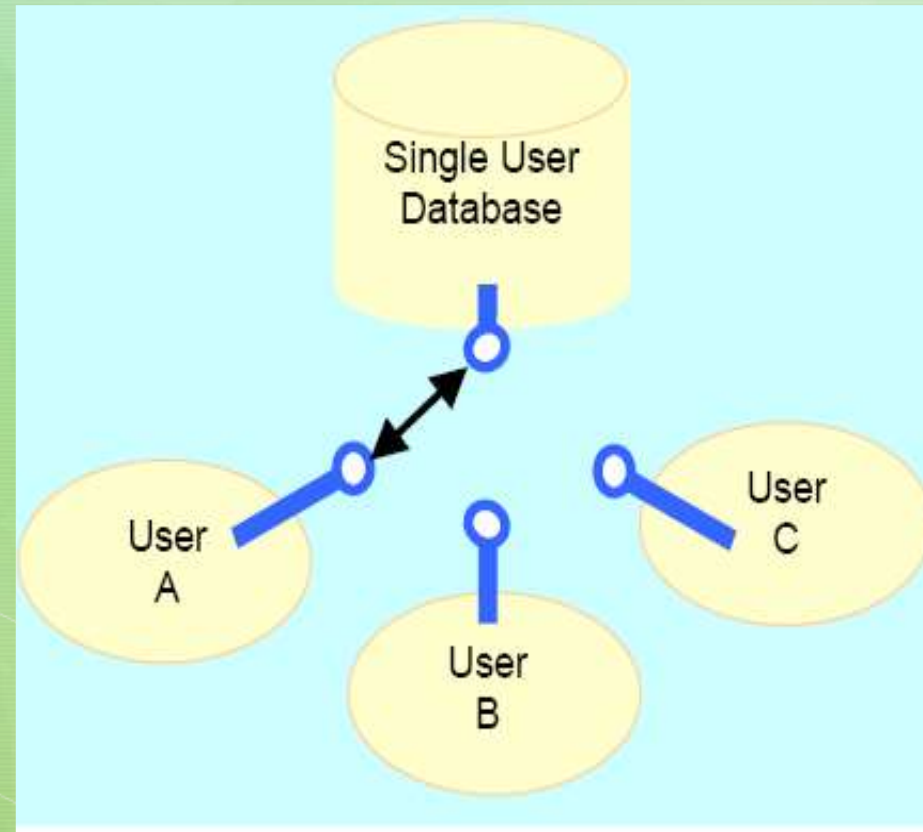
Введение в основы базы данных

- Введение в системы баз данных
- Системы управления базами данных (СУБД)
- Тип баз данных
- Дизайн базы данных
- Рекомендации по проектированию базы данных
- Компонент систем баз данных

Types of Database

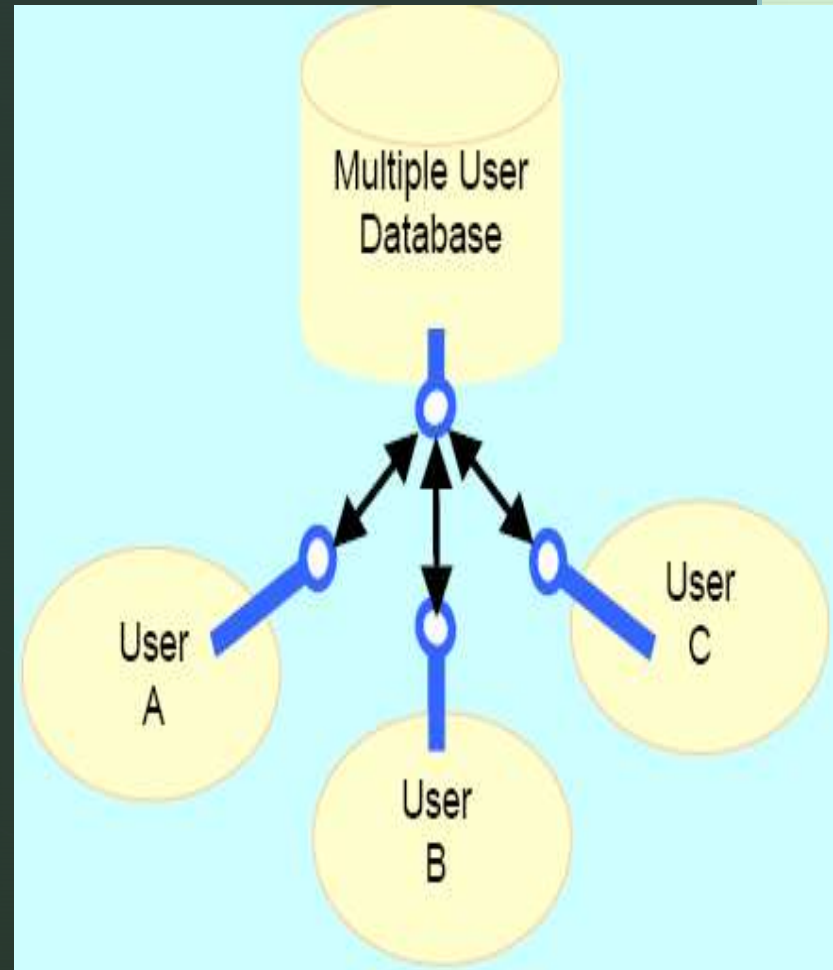
Classification by user

- **Single user databases:** Access was initially designed as a single user database
- If Single user database system used for multiple user, there are two problems need to be concern: Data corruption and Performance & Speed.



Типы баз данных

- **Классификация по пользователю**
 - Однопользовательские базы данных: Access изначально проектировался как однопользовательская база данных.
 - Если однопользовательская система базы данных используется для нескольких пользователей, необходимо учитывать две проблемы: повреждение данных и производительность и скорость.



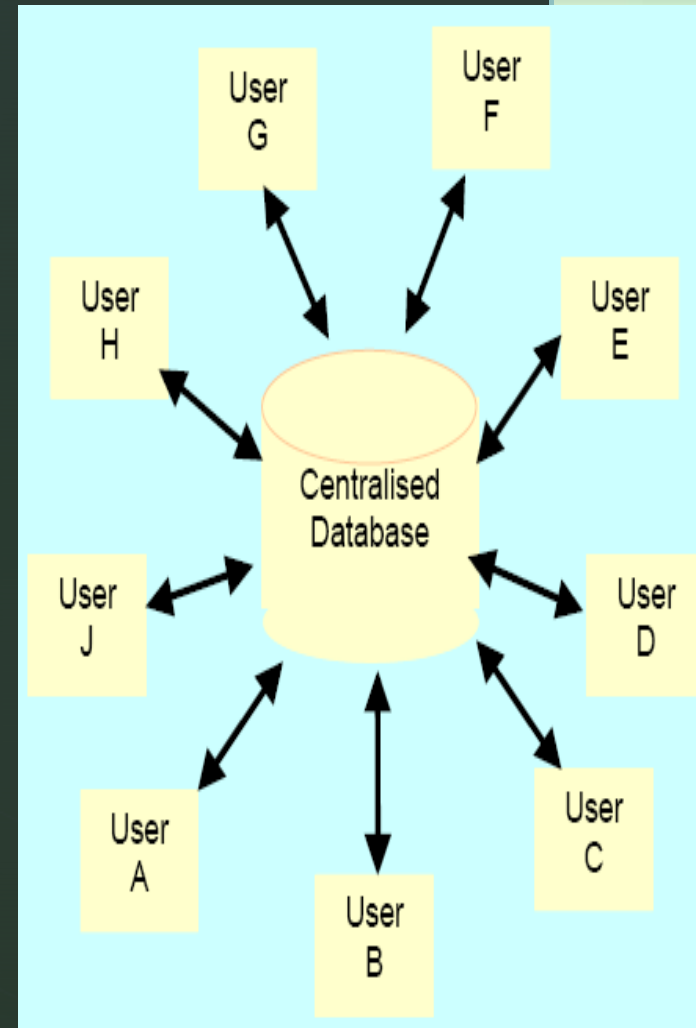
Решения Microsoft для многопользовательского рынка

- Файловый сервер: база данных Access настроена как общий сетевой ресурс, поэтому несколько пользователей могут использовать ее одновременно.
- Клиент / Сервер: Доступ работает как клиент, на стороне сервера есть MS SQL сервер
- Репликация базы данных: сеть MS поддерживает репликацию через свои службы каталогов.
- Интернет-решения для баз данных. Сервер ASP / Access или ASP / SQL

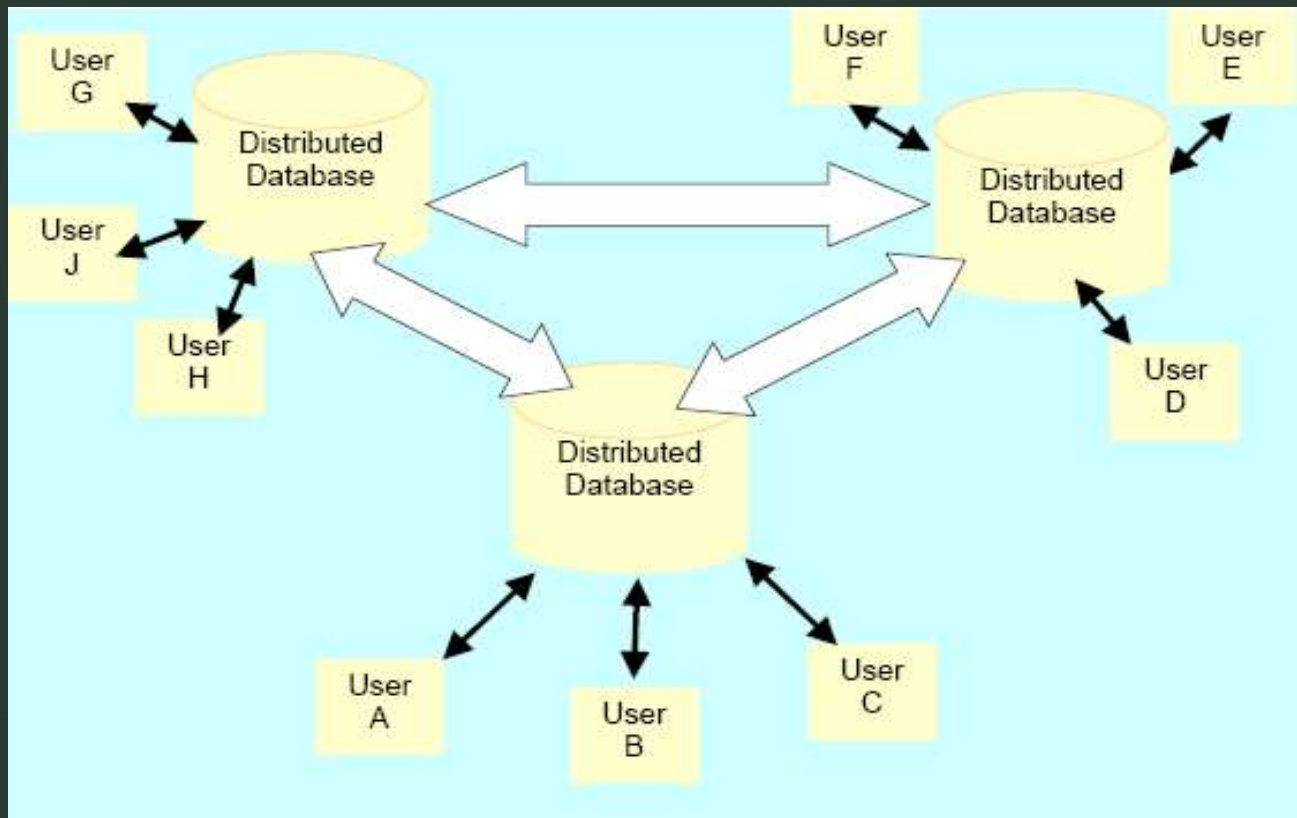
Классификация по местонахождению

Централизованная база данных: в централизованной системе баз данных есть только одна база данных, которая используется совместно пользователями.

- В:** В чем главный недостаток централизованных баз данных? Как решить эту проблему?
- О:** Возможно, основным недостатком этого типа базы данных является то, что отказ сервера, на котором работает база данных, не позволит всем пользователям использовать базу данных до тех пор, пока сервер не вернется в рабочее состояние. Чтобы решить эту проблему, более крупные приложения баз данных, как правило, относятся к распределенному типу.
- В:** В чем главный недостаток централизованных баз данных? Как решить эту проблему?
- О:** Возможно, основным недостатком этого типа базы данных является то, что отказ сервера, на котором работает база данных, не позволит всем пользователям использовать базу данных до тех пор, пока сервер не вернется в рабочее состояние. Чтобы решить эту проблему, более крупные приложения баз данных, как правило, относятся к распределенному типу.



Классификация по местонахождению - Распределенная база данных



Классификация по местонахождению Распределенная база данных

- Распределенная база данных обычно встречается в сетях. Здесь текущая версия базы данных хранится на разных серверах. Любые изменения реплицируются по сети, поэтому все копии базы данных актуальны.

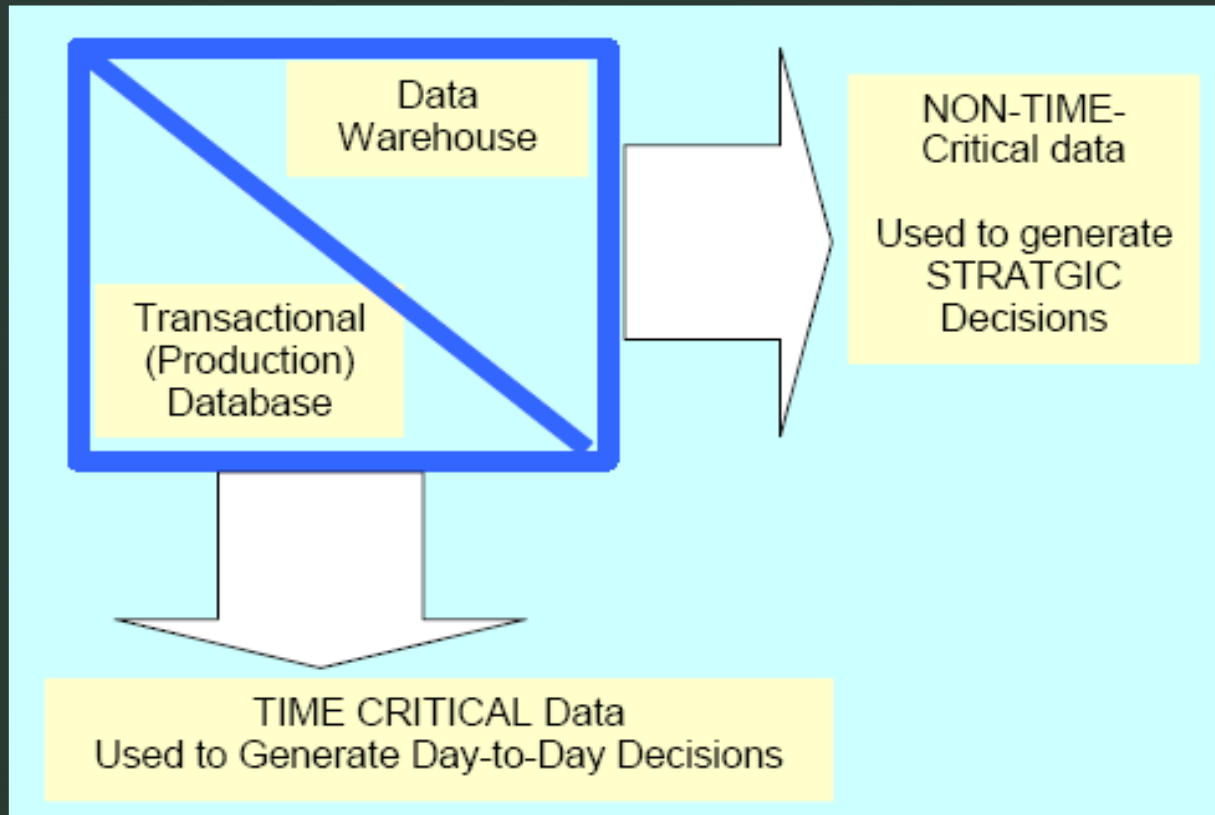
- Каковы основные недостатки и недостатки распределенных баз данных?

- Основное преимущество такой схемы, пожалуй, в отказоустойчивости. Если один из серверов не работает, другие могут взять на себя рабочую нагрузку. Эти типы баз данных распространены в корпоративных базах данных.
- Главный недостаток - стоимость, как с точки зрения лицензирования программного обеспечения, так и с точки зрения стоимости оборудования в сети. В условиях интенсивного использования сетевой трафик также может быть проблемой.

▸ Классификация по типу

- Классификация по типу различает **транзакционную базу данных и хранилище данных** .
- Это различие довольно интуитивно понятно, поскольку транзакционная база данных используется для обработки повседневной информации о транзакциях и ведения записей о решениях, принятых в отношении этих данных. В связи с тем, что в этой системе баз данных может происходить множество транзакций и что эти транзакции будут влиять на решения, принятые в этот день, эти типы баз данных считаются **критическими по времени** .
- С другой стороны, **хранилище данных** используется для выработки стратегических решений, а хранимые данные имеют тенденцию быть более долгосрочными. Следовательно, эти данные считаются **НЕ критичными по времени** .
- С точки зрения количества приложений транзакционная база данных (также называемая производственной базой данных) является более распространенной из двух типов баз данных.

Классификация по типу



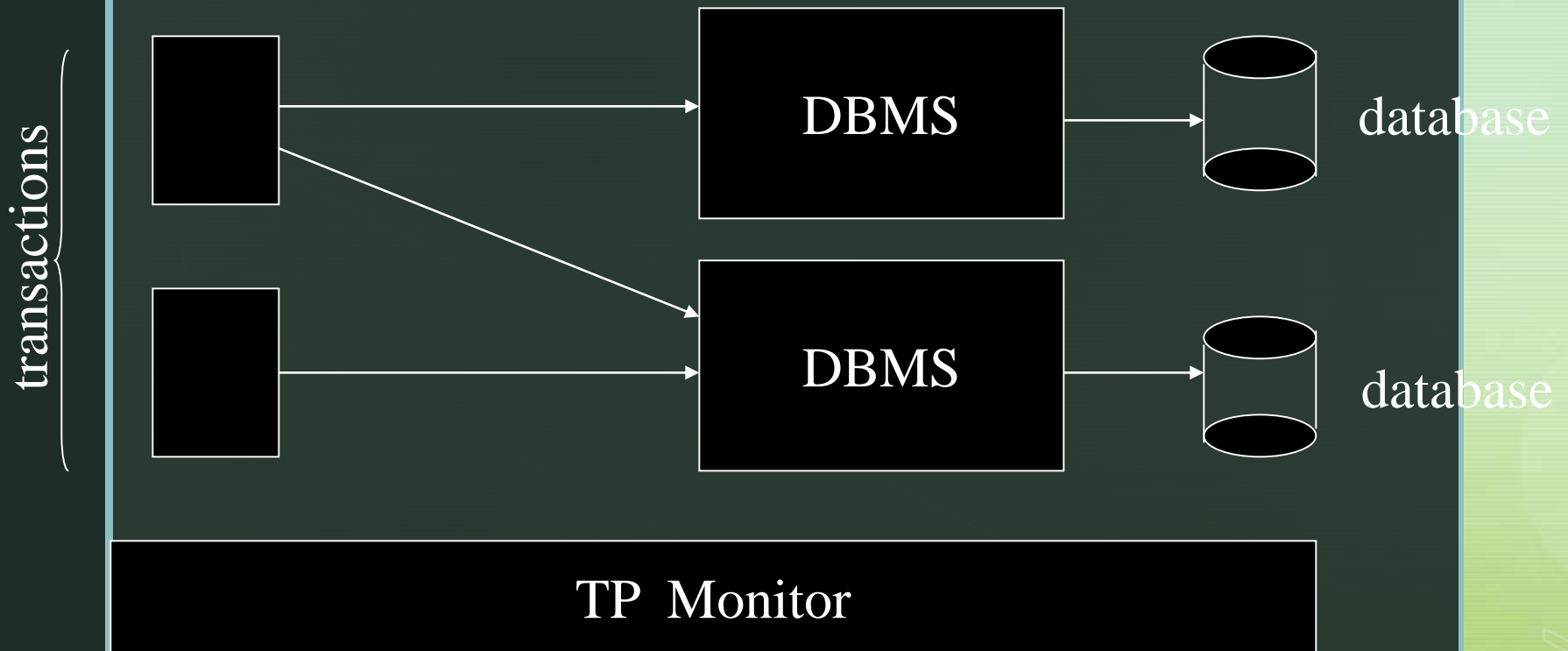
Что такое транзакция?

- Когда событие в реальном мире меняет состояние предприятия, выполняется транзакция, вызывающая соответствующее изменение состояния базы данных.
 - В интерактивной базе данных событие вызывает выполнение транзакции в реальном времени.
- **Транзакция** - это прикладная программа со специальными свойствами, которые будут обсуждаться позже, для обеспечения корректности базы данных.

▀ Что такое система обработки транзакций?

- Выполнение транзакции контролируется монитором TP
 - Создает абстракцию транзакции, аналогично тому, как операционная система создает абстракцию процесса.
 - TP Monitor и СУБД вместе гарантируют особые свойства транзакций
- Система обработки транзакций состоит из монитора транзакций, баз данных и транзакций.

Система обработки транзакций



OLTP vs. OLAP

- **On-line Transaction Processing (OLTP)**
 - Повседневная обработка транзакций, возникающих в результате работы предприятия
 - Поддерживает соответствие между состоянием базы данных и состоянием предприятия
- **On-line Analytic Processing (OLAP)**
 - Анализ информации в базе данных для принятия управленческих решений

OLAP

- Анализирует исторические данные (терабайты) с помощью сложных запросов
- Из-за объема данных и сложности запросов OLAP часто использует хранилище данных.
- **Хранилище данных** - (автономное) хранилище исторических данных, созданных из OLTP или других источников.
- **Data Mining** - использование данных хранилища для *обнаружения* взаимосвязей, которые могут повлиять на стратегию предприятия.

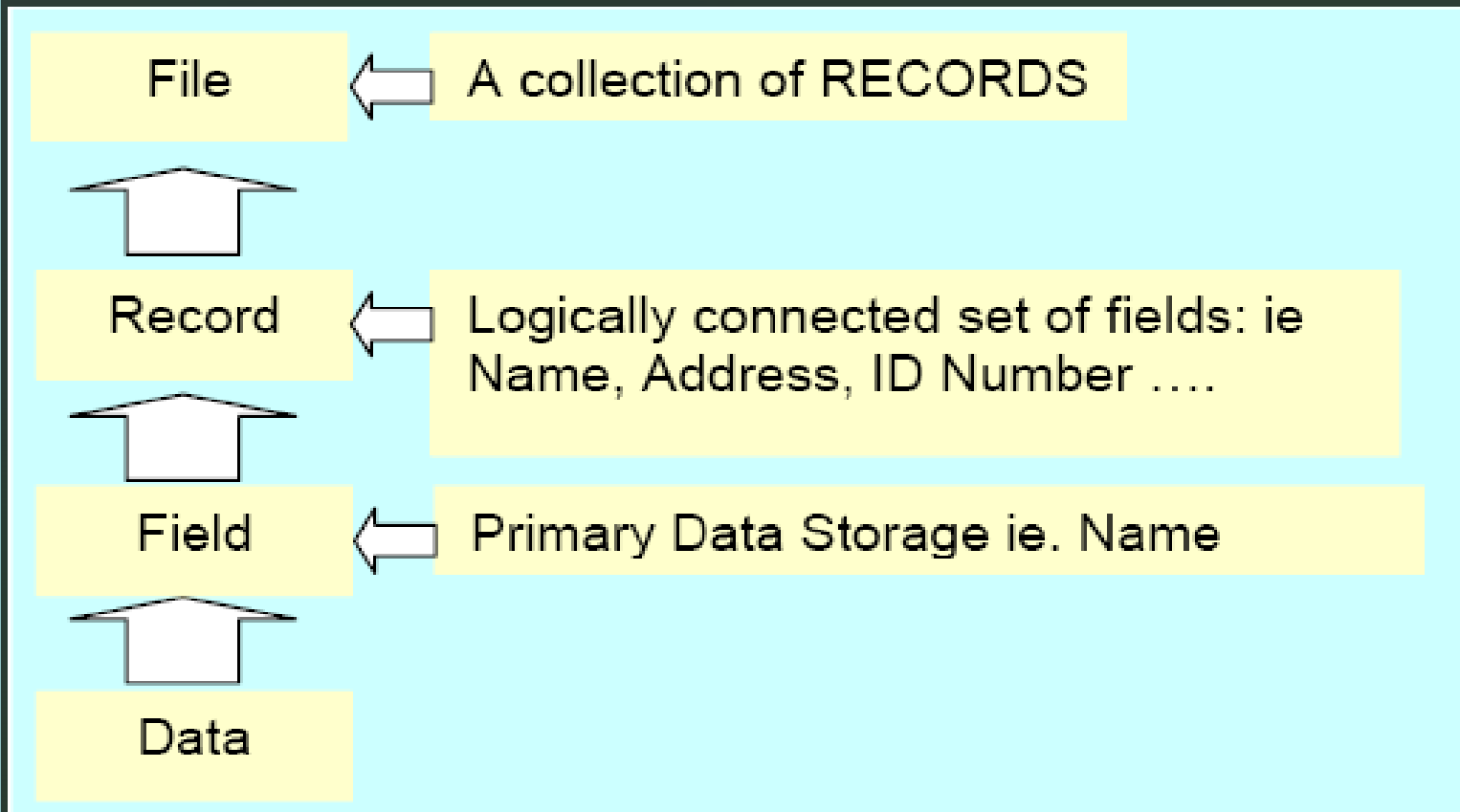
Examples - Supermarket

- **OLTP**
 - Событие: 3 банки супа и 1 коробка сухарей; обновить базу данных, чтобы отразить это событие
- **OLAP**
 - Прошлой зимой во всех магазинах северо-востока, сколько покупателей вместе купили суп и крекеры?
- **Сбор данных**
 - Есть ли какие-нибудь интересные комбинации продуктов, которые клиенты часто покупают вместе?

Введение в основы баз данных

- Введение в системы баз данных
- Системы управления базами данных (СУБД)
- Тип базы данных
- **Дизайн базы данных**
- Рекомендации по проектированию базы данных
- Компонент систем баз данных

Дизайн базы данных



Very basic level database components

Дизайн базы данных

- Дизайн базы данных связан с тем, как эти компоненты организованы, и это определяет, насколько эффективным будет проектное решение.
- Чтобы разработать структуру данных, нам нужно задать себе несколько вопросов, например:
 - 1. Знаем ли мы все поля, которые нам нужно включить, чтобы наши записи и файлы содержали все необходимые данные?
 - 2. Какие поля будет содержать каждая из наших записей.
 - 3. Разрешаем ли мы дублировать поля в записях? Это создаст избыточность данных.
 - 4. Какие записи нужны в файлах?

Введение в базы данных

- Введение в системы баз данных
- Системы управления базами данных (СУБД)
- Тип базы данных
- Дизайн базы данных
- Рекомендации по проектированию базы данных
- Компонент систем баз данных

Рекомендации по проектированию базы данных

- **Зависимость от данных и структурная зависимость**
 - Данные организованы дизайнером в структуру, которая соответствует их применению. Таким образом, существует **структурная зависимость**, связанная с данными в базе данных.
 - Структурная зависимость касается механизма изменения структуры базы данных.
 - Хороший дизайн базы данных должен гарантировать, что внести необходимые изменения в структуру относительно легко.
 - Например, добавление поля к существующей файловой структуре.
 - **зависимость от данных** и занимается вопросами самих данных, а не структуры.
 - Например, зависимость данных будет связана с тем, насколько легко изменить поле, скажем, целое на десятичное.

Database design considerations

- **Избыточность данных**

- Избыточность данных, пожалуй, самая распространенная проблема при проектировании БД. Это означает, что одни и те же данные хранятся в нескольких местах. Например, адрес студента хранится в финансовом файле, а также в файле результатов зачисления и экзаменов.
- Хороший дизайн базы данных гарантирует, что данные не будут дублироваться подобным образом. Это сэкономит место для хранения (памяти), а также повысит скорость доступа к данным и их обработки.
- **целостность данных** относится к согласованности данных, которые дублируются.
- Например, если наш дизайн требует, чтобы мы поддерживали адрес электронной почты студента в нескольких разных местах, то неспособность обновить изменение адреса электронной почты во всех местах приведет к несогласованности данных.

Рекомендации по проектированию базы данных

- **Аномалии данных**

- **Аномалии модификации**, **вставки** и **удаления** относятся к данным во взаимосвязи "один ко многим". Например, у многих учеников есть один личный наставник. Таким образом, это отношение «один ко многим», и в базе данных будет храниться запись с указанием личного репетитора для каждого ученика. Если номер телефона репетитора изменится, поскольку связь является взаимно однозначной, все файлы учеников должны быть обновлены этой новой информацией. Невозможность обновления приведет к аномалии данных, которая в данном случае называется **аномалией модификации**.
- Точно так же мы сталкиваемся с проблемами, например, с добавлением нового репетитора, когда это необходимо, и удалением старого репетитора. Все файлы студентов должны быть обновлены соответствующим образом, и невыполнение этого приведет к **аномалии вставки** или, наоборот, **аномалии удаления**.

Введение в основы баз данных

- Введение в системы баз данных
- Системы управления базами данных (СУБД)
- Тип базы данных
- Дизайн базы данных
- Рекомендации по проектированию базы данных
- **Компоненты систем баз данных**

Компоненты систем баз данных

