第1章　数据库系统引论

2012. 02

目录 Contents

1.1 数据管理

- 一、数据（data）
  - 源于拉丁语，指用于描述自然现象的一组描述符，包括经验、观察、实验的结果，表现为计算机系统中的一组前提或信息。
  - 数据可由数字、文本、图像等所组成，特别地，它通常是一组变量的测量值或观测值。
  - 在计算机科学中，数据通常进一步区分为原始数据（raw data）和处理后的数据（processed data）。
  - 原始数据一般是指直接来自设备的一组输出（如数量、字符或图像），或更广义地说，是由物理量转换而来的符号（symbol）。
  - 原始数据也可以是一个相对术语。数据处理通常是分阶段进行的，因此，来自某个阶段的“处理后的数据”可看作是下一个处理阶段的“原始数据”。

1.1 数据管理

- 一、数据（data）
  - 计算机科学中有一类特殊的数据称元数据（metadata）。
    - 元数据是关于数据的数据，用于描述数据。
    - 例如，数据库模式（database schema）就是一种元数据。
  - 在计算机科学中，通常需区分数据与程序（program）。
    - 程序是规定计算机执行任务的一组指令。
    - 从这种意义上说，数据是指计算机可使用的、除程序代码（code）外的任何东西。

1.1 数据管理

- 二、数据、信息和知识区别
  - 数据、信息（information）和知识（knowledge）这三个术语常常交迭使用。
  - 三者的主要不同是抽象的层次：数据是抽象的最低层，信息其次，知识的抽象层次最高。
  - 通过数据处理（data processing），计算机系统将数据转换为信息或知识。

1.1 数据管理

- 一、数据密集型应用与数据管理
  - **数据密集型应用（Data Intensive Applications）**
    - **特点**
      - 数据量大（>=megabyte-level, 1 megabyte = 106 bytes）
      - 持久数据（persistent data）
      - 共享数据（shared data）
    - **e.g. IS, MIS, OLTPs; OLAPs, DM&DW, etc. (*Airline Reservations Systems, Banking Systems, …*)**
  - **数据管理（Data Management）**
    - **任务**
      - 数据组织（organization）
      - 数据存储（storage）
      - 数据访问/检索/查询（access/retrieval/query）
      - 数据更新/维护（updating/ maintenance）
      - 数据安全（security）

1.1.2 数据管理技术的发展阶段

- **文件系统（File Systems）**：50年代中 ~ 60年代中后
  - **特点**
    - “数据管理”作为OS的一部分
    - 数据以“文件”（----记录的集合）的形式组织、存储
    - “文件级”的数据共享
    - 仅提供低级的文件操作（OPEN, CLOSE, READ, WRITE, …）
  - **缺点**
    - 编写应用程序很不方便；程序员负担过重；数据操作太过程化（procedural）
    - 数据冗余（redundancy）& 不一致性（inconsistency）难以避免
    - 数据独立性（independence）不好；应用可维护性差
    - 数据共享性差；不支持并发访问（concurrent access）
    - 由于缺少统一管理，规范化&标准化程度低；安全性（security）差

1.1.2 数据管理技术的发展阶段

- **数据库系统（Database Systems）：60年代后 ~ 至今**
  - **特点**
    - “数据管理”有专门的**数据库管理系统（Database Management System, DBMS）**统一管理
    - 数据以一定的**模式（schema）**组织、存储
    - **“记录/数据项级”**的数据共享
    - 提供高级的、可并发的数据操作（查询，操纵，控制）

1.1.2 数据管理技术的发展阶段

- **数据库系统（Database Systems）**
  - **优点**
    - 提供了专门的数据库语言（Database Languages）用以：
      - 定义数据模式、查询数据、操纵数据、控制数据
      - 而且，数据的逻辑形式与物理形式相分离，用户只面对其逻辑形式；数据的查询与操纵是高度非过程化的（non-procedural）；应用程序的编写很方便。
    - 尽可能地避免了数据冗余、保持数据的一致性
    - 数据独立性好；应用可维护性高
    - 数据共享性高；支持并发访问
    - 由于统一管理，规范化&标准化程度高；保证了数据安全性、具备数据恢复（recovery）能力

1.1.2 数据管理技术的发展阶段

- **二、数据库技术的发展历史**
  - 以采用的**数据模型(Data Model)**来划分：
  - **第一代：层次(hierarchical)数据库 & 网状(network)数据库**
    - 第一个DBMS：1964年，美国通用电器公司Bachman等人开发成功的IDS (Integrated Data Store)奠定了网状数据库的基础
    - 第一个商品化的层次DBMS：60年代末，IBM公司推出的层次数据库管理系统IMS (Information Management System)

1.1.2 数据管理技术的发展阶段

- **第二代：关系(Relational)数据库**
  - **理论：**1970年，Codd E.F., “A Relational Model for Large Shared Data Banks”, In: ***Communication of the ACM***, 13:6, PP.377-387.奠定了关系数据库的理论基础
    - (***ACM*** = ***A***ssociation for ***C***omputing ***M***achinery, (美国)计算机协会)
    - (***IEEE*** = ***I***nstitute of ***E***lectrical & ***E***lectronic ***E***ngineers, (美国)电气&电子工程师学会)
  - **产品：**1977年前后，IBM公司的原型系统System R→商品化的SQL/DS及DB2；
  - 加州大学的原型系统INGRES→INGRES公司将其商品化
  - 80年代后，关系数据库大发展，流行的RDBMS有：Oracle, Sybase, Informix, SQL Server, FoxPro, etc.

1.1.2 数据管理技术的发展阶段

- 第三代：后关系(Post-relational)数据库
  - 改造与扩充关系数据库，以适应新的应用领域及其应用需求，
    - e.g. CAD/CAM, CIMS, OA, GIS, S&S, OLAP&DSS, Data Mining, Data Warehousing, …
  - 采用新的数据模型，从而产生新的数据库系统，
    - e.g. Object-Relational, O-O, Logic/Deductive, …
  - Web的流行、越来越复杂的应用环境、以及硬件的飞速发展，动摇了传统数据库的基本前提假设，新一代数据库系统必将应运而生。

目录 Contents

- 1.1 数据管理
- 1.2 数据库系统
- 1.3 数据抽象
- 1.4 数据库的生存周期

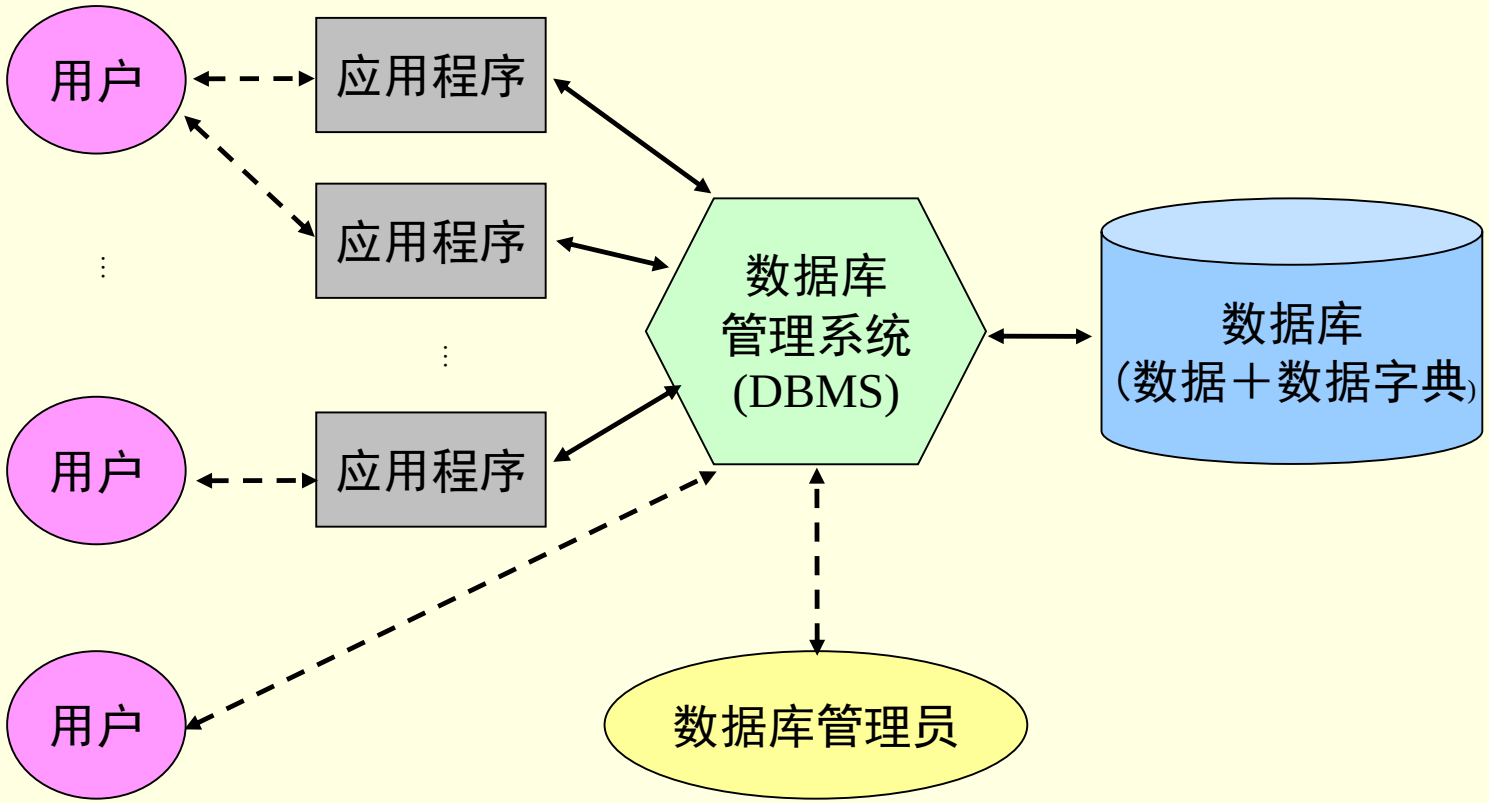
1.2 数据库系统

一、何谓数据库系统

- **数据库系统**(Database System)：由数据库管理系统(DBMS)、数据库(DB)、应用程序(applications)、及数据库管理员(Database Administrator, DBA)组成的系统。
- 其核心是DBMS（一般是通用的商品化软件）。
- DBMS的正常运行需要一个特定的环境。一个DBMS环境（DBMS Environment）通常包括五个部分：硬件、软件、数据、处理过程和人。

1.2 数据库系统

- **数据库系统构成图示**



1.2 数据库系统

二、数据库系统中各类人员

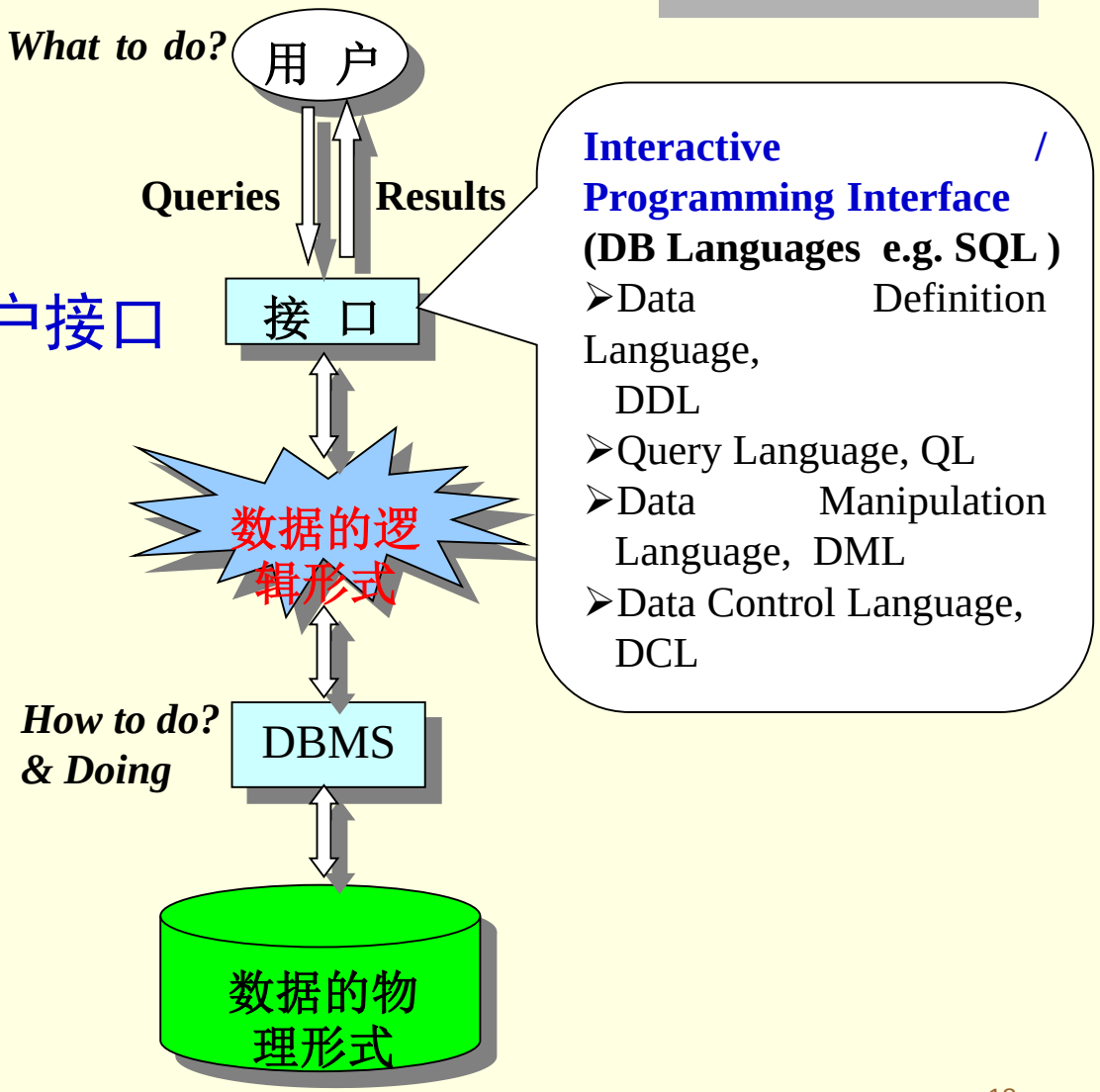
- 数据管理员（data administrator, DA）
  - DA负责一个组织的数据资源的管理，数据库规划，标准、政策、处理过程等的开发与维护，数据库的概念设计与逻辑设计等。DA确保数据库开发最终能支持组织的目标。
- 数据库管理员（database administrator, DBA）
  - DBA负责数据库的物理实现，数据库的物理设计与实现，安全与完整性（integrity）控制，运行维护，并确保数据库应用的满意性能。DBA较DA更为面向技术，通常需具备目标DBMS和系统环境的详细知识。
- 数据库最终用户（end-user）
  - 最终用户也称数据库的客户（clients），包括使用已实现的应用程序来访问数据库的一般用户，或直接使用数据库语言（如SQL）与DBMS交互来访问数据库的高级用户。

1.2 数据库系统

- 三、数据库系统中各类人员
  - 数据库设计员（database designer）
    - 此类人员的职责是数据库的设计，包括逻辑设计（标识数据、数据间关系、数据上的约束等）和物理设计（选择合适的存储结构和存取方法等）。
  - 数据库应用开发员（application developer）
    - 此类人员的职责是使用第三代编程语言（如C语言）或第四代语言（4GL，如SQL语言）来设计与实现应用程序，以便为数据库最终用户提供所需的数据访问（数据检索）和数据操纵（数据插入、更新与删除）功能。

1.2 数据库系统

- 三、初识DBMS
  - DBMS功能
    - 提供高级的用户接口



1.2 数据库系统

- **DBMS功能**
  - 查询处理&优化
    - **查询(Queries)**为广义的概念，包括：**SELECT, INSERT, DELETE, UPDATE**
    - **查询处理与优化(Query Processing & Optimization):** 语法检查，语义分析，制定执行策略，执行查询并返回结果。
  - 数据目录管理
    - **元数据(Metadate):** 数据定义信息，存储结构信息，其他管理信息
    - **数据目录/字典(Data Catalog / Directory，DD):**存放元数据的（系统）数据库。

1.2 数据库系统

- **DBMS功能**
  - 并发控制
    - 由于数据库中的数据是共享的，即用户是并发访问数据库的，所以要有**并发控制**(Concurrency Control)机制,以保证不发生“冲突”。
  - 数据库恢复
    - 系统可能会发生故障从而导致数据库**失效**(Failure)，所以要有**数据库恢复**(Database Recovery)机制，以保证DB始终处于**一致**(Consistency)状态。

1.2 数据库系统

- **DBMS功能**
  - 完整性约束检查
    - 数据库中的数据必须遵守一定的约束才能保证其正确性，并向用户提供正确的信息含义。约束可分为**语法的**(syntactical)约束和**语义的**(semantic)约束----称为**完整性约束**(Integrity Constraints)。
    - DBMS必须提供完整性约束检查功能。
  - 访问控制
    - 在共享的数据环境中，必须控制不同用户对数据库的不同访问**特权**(Privileges)，以保证数据库的**安全性**(Security)。

1.2 数据库系统

- **DBMS组成**

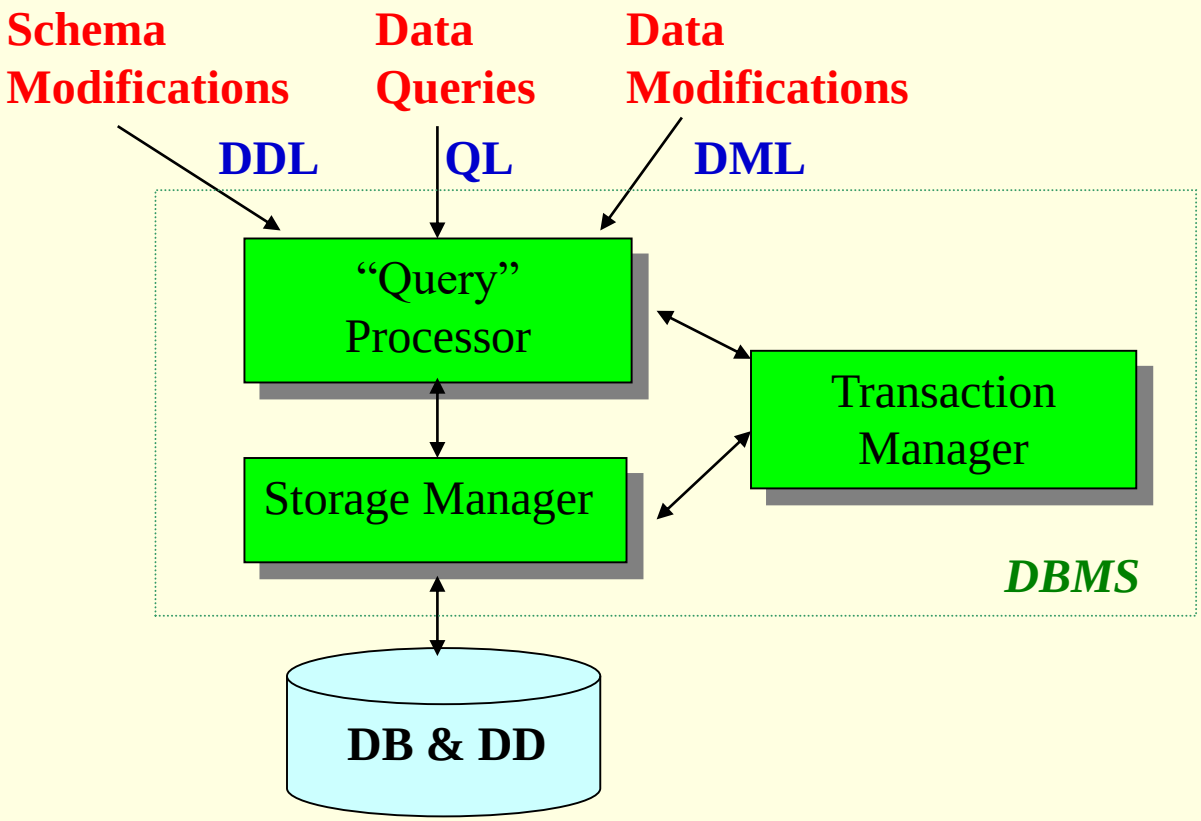
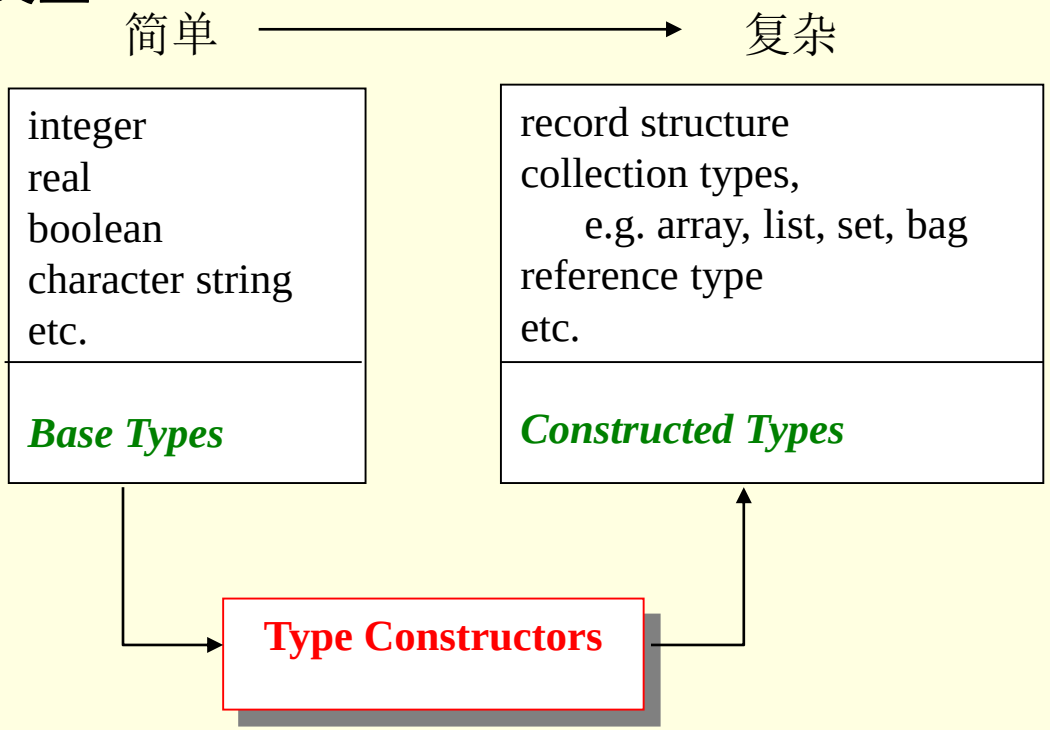


Figure: Major Components of a DBMS

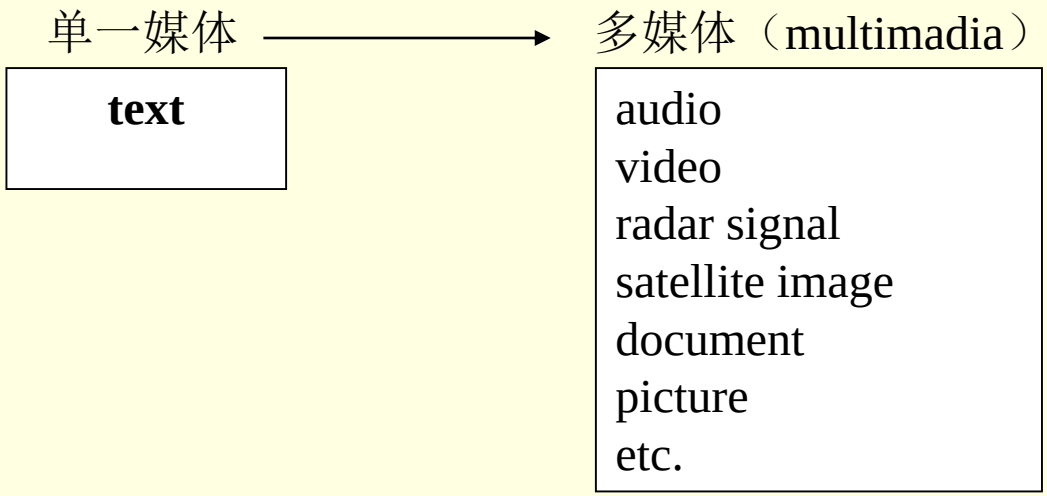
1.2 数据库系统

- **四、趋势与未来**
  - 面向对象(Object Orientation)
    - 数据→对象
      - 数据类型



1.2 数据库系统

- **面向对象(Object Orientation)**
  - **数据→对象**
    - **数据媒体**



- **对象-关系数据库（Object-relational Databases）成为主流**

1.2 数据库系统

- **四、趋势与未来**
  - **多种数据格式并存**
    - 结构化（Structured）、半结构化（Semi-Structured）甚至无结构（Unstructured）数据并存
    - XML已是Web数据表示和交换的工业标准，XML数据是半结构化数据，XML文档是结构化文档
    - 主要的商业(O-)RDBMS产品已更新换代，支持XML数据处理，成为XML-Enabled数据库产品

1.2 数据库系统

- **四、趋势与未来**
  - **大型数据库（VLDB）**
    - **数据库容量不断扩大（缘于：多媒体数据，保存大量历史数据的事务处理系统，e.g. Retail chains）：**
      - **1 bit = 0 or 1**
      - **1 byte = 8 bits**
      - **1 kilobyte = 103 bytes or 1024 bytes**
      - **1 megabyte = 103 kilobytes**
      - **1 gigabyte = 103 megabytes**
      - **1 terabyte = 103 gigabytes**
      - **1 petabyte = 103 terabytes**
    - **二级存储(Secondary Storage)→ 三级存储(Tertiary Storage)**
      - 主存 ——— 辅存 ——— 三级存储设备
      - **(内存)　　(磁盘)　　(机器人CD库)**
    - **数据查询处理及其他机制的技术与算法将发生根本变化**

1.2 数据库系统

- **三、趋势与未来**
  - **分布(Distribution)& 集成(Integration)**
    - 数据在物理上是**分布的**(Distributed)
      - 数据分散存储在不同的地理位置、不同的机器上。特别是Internet/Web的流行使的**Web数据源**（Web Data Sources）成为数据管理的主要数据源。
    - 数据源往往是**异构的**(Heterogeneous)
      - 可能有不同的DBMS、不同的数据模型、不同的数据语法&语义。

1.2 数据库系统

- 数据需要**集成(Integrating)**
  - 集成：将分散在各单元中的（软、硬件）元素集合到一个整体系统中，使整体比各单元发挥更大作用的过程。
  - 通过集成，向用户提供统一的数据视图（Unified Data View）,而且分布透明（Distribution Transparency）、场地自治（Site Autonomy）。

- 集成的方法
  - 通过统一的分布式数据库管理系统（DDBMS）：强调全局数据模式（Global Data Schema）、数据统一管理；自治性差，较适合同构数据源。（逻辑上集中）
  - 联邦式数据库系统(Federated DB System)：不强调全局数据模式、松散联合；自治性高，允许异构数据源。（逻辑上分布）

1.2 数据库系统

- **四、趋势与未来**
  - **主动数据库(Active Database)**
    - 数据库中加入**主动元素**(Active Elements)
      - 约束：语义完整性的自动维护
      - 触发器(Trigger)：事件-条件-动作规则(ECA Rule)

1.2 数据库系统

- **四、趋势与未来**
  - **支持高层次应用（如：决策分析、知识发现等）和复杂系统的数据库新技术**
    - 数据仓库化（Data Warehousing）
    - 数据挖掘（Data Mining）
    - 工作流（Workflow）应用中的数据库技术
    - 电子商务（E-Commerce or E-Business）应用中的数据库技术
    - 数据库/数据管理技术集成于其他计算机应用系统中

目录 Contents

- 1.1 数据管理
- 1.2 数据库系统
- 1.3 数据抽象
- 1.4 数据库的生存周期

1.3 数据抽象

- 一、三层抽象
  - 数据库系统是面向计算机的，而应用是面向现实世界的，两个世界存在着很大差异，要直接将现实世界中的语义映射到计算机世界是十分困难的，因此引入一个信息世界作为现实世界通向计算机实现的桥梁。
  - 一方面，信息世界是对现实世界的抽象，从纷繁的现实世界中抽取出能反映现实本质的概念和基本关系；另一方面，信息世界中的概念和关系，要以一定的方式映射到计算机世界中去，在计算机系统上最终实现。信息世界起到了承上启下的作用。

1.3 数据抽象

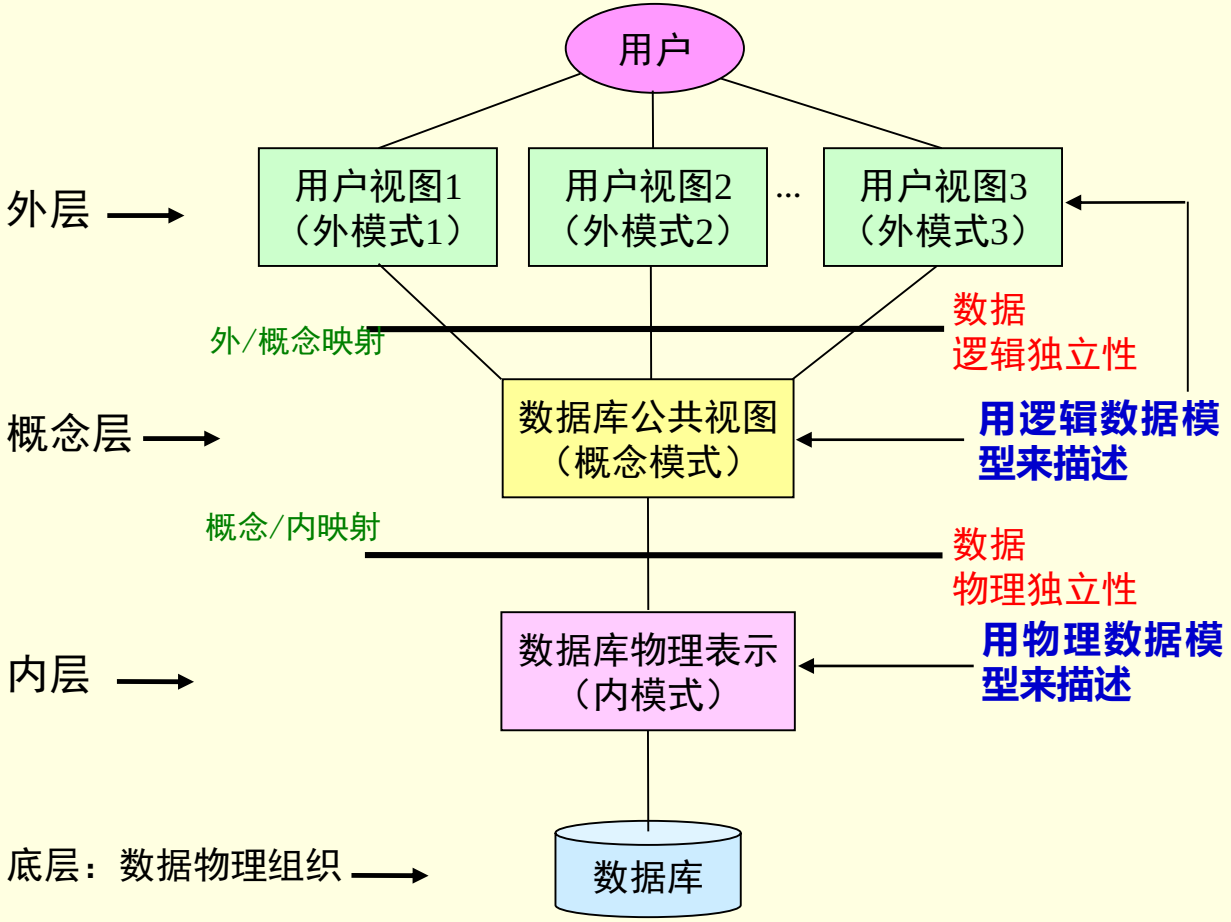
- 理想的数据模型最好
  - 既能真实、自然地模拟现实世界，便于人们理解、交流----面向现实世界/用户；
  - 又便于在计算机上实现----面向机器世界/实现
  - 但这往往是一对矛盾。
- 解决的办法是采用多层数据抽象。

1.3 数据抽象

- 1975年，美国国家标准协会（American National Standards Institute, ANSI）下属的标准规划与需求委员会（Standards Planning And Requirements Committee Architecture, SPARC）提出了数据库管理系统（DBMS）的抽象设计标准ANSI-SPARC体系结构（ANSI-SPARC Architecture）

1.3 数据抽象

- **ANSI-SPARC三层体系结构**



1.3 数据抽象

- ANSI-SPARC体系结构中三层抽象
  - 外层（external level）
    - 即数据库的用户视图（user's view）。该层描述数据库中与特定用户相关的部分。
  - 概念层（conceptual level）
    - 即数据库公共视图（community view）。该层描述数据库中包含什么（what）数据（以及数据间关系）。
  - 内层（internal level）
    - 即数据库物理表示（physical representation）。该层描述数据库中数据是如何（how）存储的。

1.3 数据抽象

- 三层体系结构的目标是将数据库物理表示和组织方式与数据库的用户视图进行分离，即提供数据独立性（data independence）。

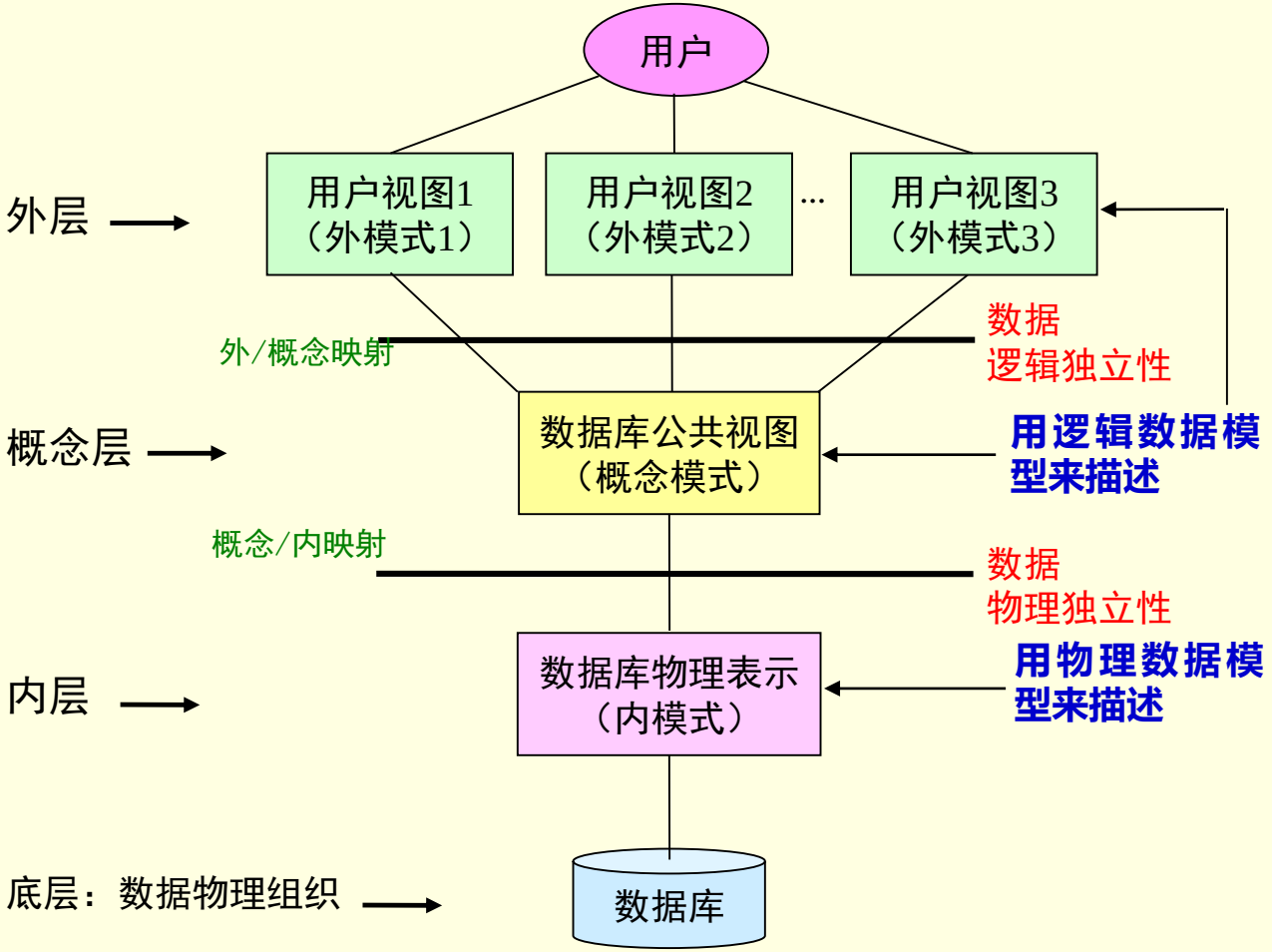
- 这种做法是希望：
  - 为不同用户提供独立的定制视图；
  - 为用户隐藏数据的物理存储细节；
  - 存储的物理方面的改变应不影响数据库的内部结构；
  - 数据库管理员（DBA）改变数据库的存储结构不影响用户视图；
  - DBA改变数据库的概念或全局结构不对用户产生影响。

1.3 数据抽象

- 二、模式、实例、映射与数据独立性
  - 区分数据库的描述（元数据）与数据库本身（数据）很重要。引人了数据库模式（database schema）和数据库实例（database instance）两个不同的概念。前者相对稳定，后者经常变动。
  - 数据库的（总体）描述称为数据库模式，也称数据库的内涵（intension）。
  - 数据模式(Data Schema)
    - 运用某种数据模型*（手段）*对一个企业(Enterprise)/组织(Organization)（如：公司、学校、政府部门）的一组数据之结构、联系和约束进行描述的结果*（目的）*。
  - 数据库中特定时间点的数据称为数据库实例（database instance）或状态（state），也称数据库的外延（extension）。

1.3 数据抽象

- **ANSI-SPARC三层体系结构**



1.3 数据抽象

- 三种数据库模式
  - 对应于ANSI-SPARC体系结构的三层抽象，有：
  - 外模式（external schema）或子模式（subschema）
    - 分别描述数据的不同视图；用逻辑数据模型对一个企业中各个特定用户所涉及的那部分数据的描述。
  - 概念模式（conceptual schema）
    - 描述数据库中所有实体、属性与关系，以及完整性约束；用逻辑数据模型对一个企业中全体数据的描述。
  - 内模式（internal schema）
    - 描述数据库的内部模型，包括数据域与存储记录的定义、表示方法、索引与存储结构等。用物理数据模型对一个企业中全体数据的描述。

1.3 数据抽象

- 数据独立性
  - DBMS维护上述三种模式之间的映射（mapping），以便提供数据独立性：
  - 逻辑独立性（logical independence）
    - 指外模式（和其上运行的应用程序）对概念模式改变的抗扰性（immunity）。这通过外模式与概念模式之间的映射机制来确保。
  - 物理独立性（physical independence）
    - 指概念模式（和外模式）对内模式改变的抗扰性。这通过概念模式与内模式之间的映射机制来确保。
  - 数据独立性大大降低了数据库系统的使用与维护代价。

1.3 数据抽象

三、数据模型

数据模型(Data Model)

- 用来描述数据的一组概念和定义。这种描述包括三个要素/两个方面：
- 数据的基本结构 / 静态特性
  - 数据的逻辑/物理结构和数据间的联系。
- 数据中的约束 / 静态特性
  - 语义施加在数据上的约束（称完整性约束）。
- 数据上的操作 / 动态特性
  - 如何检索、更新（增、删、改）数据。

1.3 数据抽象

- **多级数据模型**(Multi-level Data Model)
  - 概念数据模型(Conceptual Data Model)
    - 面向现实世界/用户，与DBMS无关。
    - e.g. E-R模型、O-O模型
  - 逻辑数据模型(Logical Data Model)
    - 既面向用户有面向实现。
    - e.g. 网状模型、层次模型、关系模型、O-O模型
  - 物理数据模型(Physical Data Model)
    - 面向机器世界/实现，描述数据的存储结构。与DBMS、OS、硬件有关。

目录 Contents

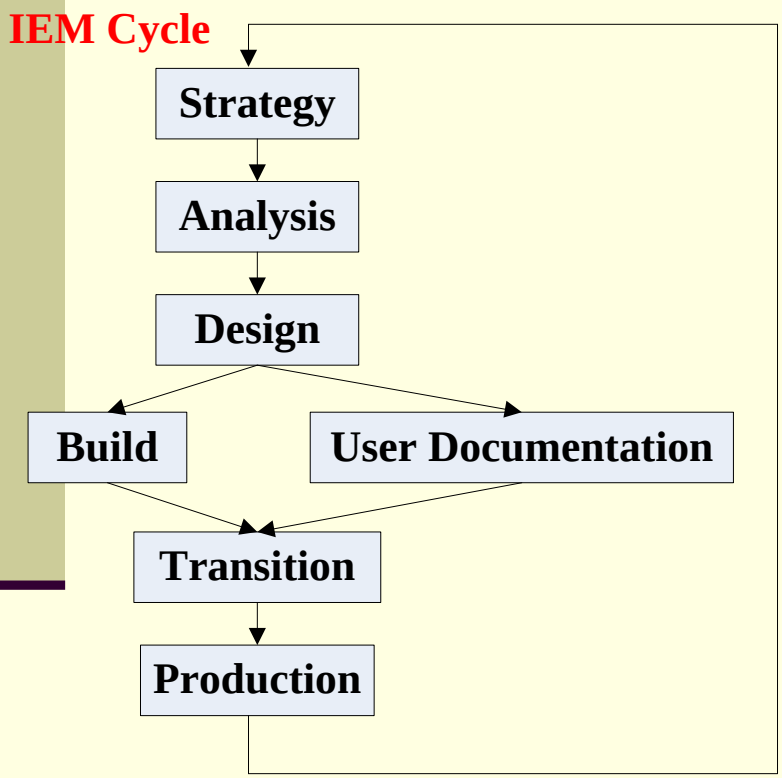
- 1.1 数据管理
- 1.2 数据库系统
- 1.3 数据抽象
- 1.4 数据库的生存周期

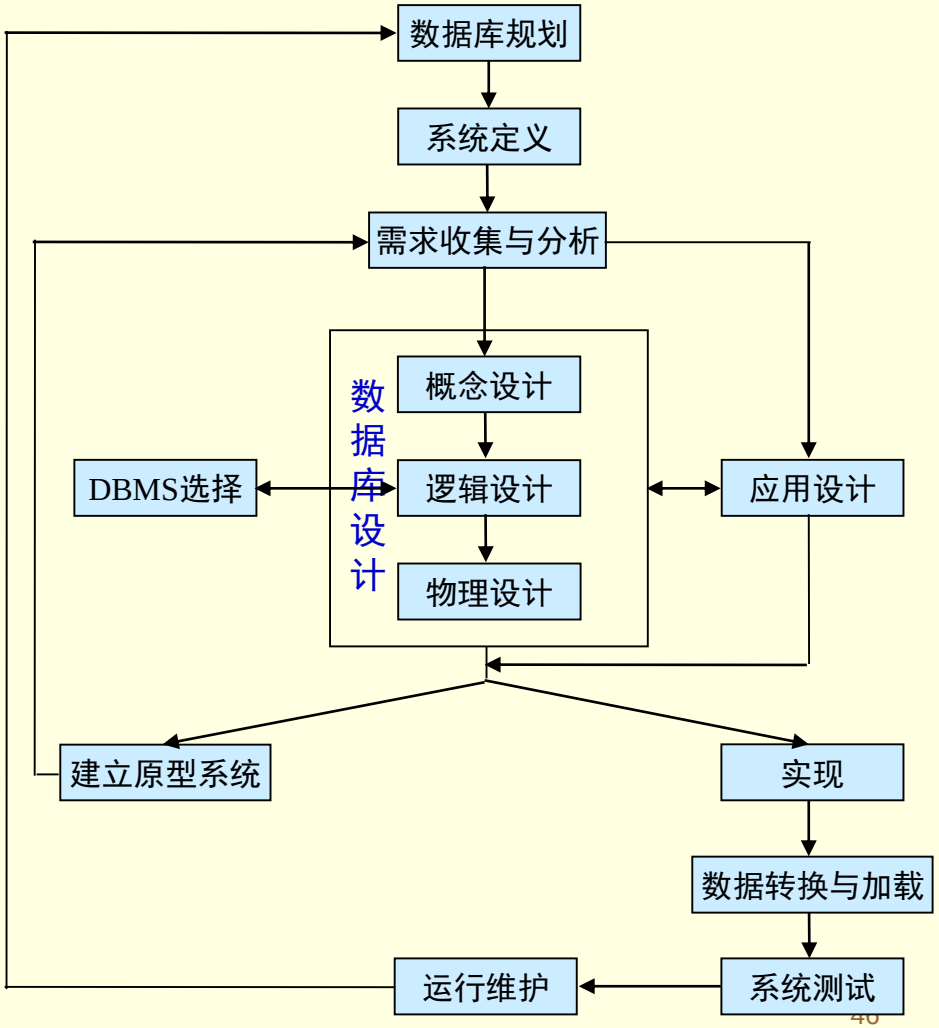
1.4 数据库的生命周期

- 在建设一个数据密集型应用系统(e.g. IS)的过程中，DB的建设是核心。
- 信息系统的建设常用**信息工程方法**(IEM)，它是一种面向数据的方法(Data-oriented Approach)，即以数据为中心，注重对业务目标的理解，在对业务过程进行分析和数据建模的基础上分阶段开发的方法。在运用IEM方法进行信息系统建设的过程中，数据库也有相应的**生命周期**(Life Cycle)。

1.4 数据库的生命周期

- **数据库系统开发生命周期的各阶段**





1.4 数据库的生命周期

- 数据库规划——规划如何最有效和高效地实现生命周期的各阶段。
- 系统定义——规定数据库系统的范围与边界，包括用户、用户视图和应用领域。
- 需求收集与分析——为新的数据库系统收集与分析需求。
- 数据库设计——数据库的概念、逻辑与物理设计。
- DBMS选择——为数据库系统选择一个合适的DBMS。
- 应用设计——设计访问与操纵数据库的用户接口与应用逻辑。
- 建立原型系统——为将实现的数据库系统构造一个原型，以便用户和设计人员评价。
- 实现——建立物理数据库定义与应用程序。
- 数据转换与加载——从旧系统加载数据到新系统，尽可能将现有应用转换到新数据库。
- 系统测试——数据库系统错误测试，用户需求可满足性验证。
- 运行维护——监控与维护运行中的数据库；可能的数据库重构以满足新的需求。