数据库概论

- 基本专业术语
- 数据库技术的产生、发展与研究领域
- 数据库系统的组成
- 数据库管理系统DBMS的组成和功能
- 练习题

专业术语

- **信息**：是关于现实世界事物的存在方式或运动状态的反映和综合。
- **数据**：数据是用来记录信息的可识别的符号，是信息的具体表现形式。

❖数据和信息的区别与联系：

数据是信息的符号表示；信息是数据的内涵，是对数据的语义解释。

- **数据库DB**：是长期储存在计算机内的、有组织的、可共享的数据的集合。其对于数据的组织要便于数据查询和检索。通常，数据库是指由DBMS管理的数据的集合。
- **数据库管理系统DBMS**：是处理数据库访问的软件，提供数据库的用户接口。DBMS的目的：提供一个可以方便地、有效地存取数据库信息的环境。

data base management system

数据库应用

- 数据库应用：
  - 银行：所有事务
  - 航班：预订、时刻表
  - 大学：注册、成绩
  - 销售：客户、产品、买卖
  - 制造业：产品、库存、订单、供应链
  - 人力资源：雇员、工资、扣税
- 数据库影响我们生活的方方面面。

数据库技术的产生与发展

数据处理的中心问题是数据管理。

数据管理是指对数据的组织、分类、编码、存储、检索和维护。

随着计算机硬件和软件的发展，数据管理经历了以下三个阶段。

- 人工管理（20世纪50年代中期以前）
- 文件系统（20世纪50年代后期至60年代中期）
- 数据库系统（20世纪60年代末开始）

人工管理阶段

20世纪50年代中期以前。

这一阶段计算机主要用于科学计算。

- 硬件中的外存只有卡片、纸带、磁带，没有磁盘等直接存取设备。
- 软件只有汇编语言，没有操作系统和管理数据的软件。
- 数据处理的方式基本上是批处理。

人工管理的特点

- 1. 数据无法在计算机内保存
- 2. 系统没有专用的软件对数据进行管理

每个应用程序都要编写和规定数据的存储结构、存取方法、输入方式等，程序员负担很重。

- 3. 数据不共享

数据是面向程序的，一组数据只能对应一个程序。因此有大量的冗余数据。

- 4. 数据不具有独立性

程序与数据相互依赖，如果数据的类型、格式、或输入输出方式等逻辑结构或物理结构发生变化，必须对应用程序做出相应的修改。

文件系统的特点

文件系统的主要任务就是管理存放的数据或程序。

- 数据以文件形式可长期保存下来
- 文件系统可对数据的存取进行管理

程序员只与文件名打交道，不必明确数据的物理存储，大大减轻了程序员的负担。

- 文件形式多样化

有顺序文件、倒排文件、索引文件等，因而对文件的记录可顺序访问，也可随机访问，更便于存储和查找数据。

- 程序与数据间有一定独立性

由专门的软件（即文件系统）进行数据管理，程序和数据间由软件提供的存取方法进行转换，数据存储发生变化不一定影响程序的运行。

文件系统的缺点

- 数据冗余度大，数据一致性差。

  各数据文件之间没有有机的联系，一个文件基本上对应于一个应用程序，数据不能共享。

由于相同数据的重复存储、各自管理，在进行更新操作时，容易造成数据的不一致性。

- 数据读取困难。数据和程序相互依赖，一旦改变数据的逻辑结构，必须修改相应的应用程序。
- 数据之间相互隔绝，数据独立性低。
- 完整性差：完整性约束成为程序的一部分，难以增加或修改。
- 无法保证更新的原子性：当发生错误时，会导致只有部分数据更新。
- 多用户并发控制：无法控制，从而导致数据的不一致。

数据库系统阶段

60年代后期，计算机应用于管理的规模更加庞大，数据量急剧增加，为解决多用户、多个应用程序共享数据的需求，出现了统一管理数据的专门软件系统，即数据库管理系统。DBMS包括如下功能：

- 允许用户用数据定义语言建立新的数据库和指定其模式（数据的逻辑结构）。
- 使用户能够用适当的语言（查询语言或数据操作语言）查询数据和更新数据。
- 支持存储大量的数据（G(10^9,十亿)字节以上），在长时间后仍保证安全，使其免遭意外或非授权的使用，同时允许对数据库查询和更新的有效访问。
- 控制多用户的同时访问，使得一个用户的访问不影响其他用户，保证同时访问不损坏数据。

数据库系统的特点

- 采用复杂的数据模型结构，描述数据本身的特点及数据之间的联系。
- 数据的独立性高，数据的物理结构与逻辑结构间差别很大，当物理结构变化时，尽量不影响用户逻辑结构。
- 共享性高、冗余少。
- 减少应用程序的开发时间，为用户提供了方便的用户接口，可使用简单的查询语言或程序方式来访问DB。
- 数据结构化，保证数据的一致性。
- 统一的数据管理和控制功能

  包括数据的安全性控制、完整性控制、并发控制、以及数据恢复。

数据库技术的发展-总结

◆数据处理发展简史

手工处理时期→文件系统时期→数据库系统时期

	手工处理	文件系统	数据库系统
计算机	科学计算	信息管理	庞大数据量
存储设备	无直接存取设备	有直接存储设备	
软件	无操作系统	有操作系统和高级语言	
数据处理	批处理	批处理和联机处理	
数据管理	无专用的软件	文件系统	数据库管理系统
数据独立	不独立	有一定独立性	独立性高
数据共享	不共享		共享性高
数据冗余与一致性	冗余度大、一致性差		冗余少，有数据控制功能

数据库系统的发展

数据模型是数据库系统的核心和基础，按照数据模型的发展历程，数据库技术的发展也经历了三个发展阶段。

❖第一代数据库系统于20世纪60年代后期研究开发的。其格式化模型包括层次模型和网状模型。

层次数据库系统和网状数据库系统的数据模型分别为层次模型和网状模型，但从本质上讲层次模型是网状模型的特例，二者从体系结构、数据库语言到数据存储管理上均具有共同的特征，都是格式化模型，属于第一代数据库系统。

1969年，美国CODASYL组织的数据库工作小组发表了DBTG报告，提出了网络数据模型的一个标准化的文本。

1969年，美国IBM公司研制了第一个层次数据库管理系统IMS。

数据库系统的发展

❖第二代数据库系统

第二代数据库系统是指支持关系数据模型的关系数据库系统。

- 1970年，IBM公司的E.F.Codd研究员发表“A Relation Model of Data for Large Shared Data Banks”，开创了数据库的关系方法和数据规范化理论的研究，奠定了关系数据库的理论基础。
- 1973年始，美国IBM公司用了五年时间研制了世界上第一个关系DBMS——System R。之后推出了DB2、DB3。
- 1985年E.F.Codd给出了完全支持关系模型的RDBMS应遵循的十二条准则，以及评价RDBMS产品的方法和规则。

其他较有影响的RDBMS还有Oracle、SQL Server、Informix等。

数据库系统的发展

- 第三代数据库系统 还处于发展阶段

主要包括面向对象DB、演绎DB、模糊DB、巨型DB、主动DB等。

1989年9月，文献“面向对象的数据库系统宣言”提出，继第一代（层次、网状）和第二代（关系）数据库系统后，新一代DBS将是OODBS。

1990年高级DBMS功能委员会发表了“第三代数据库系统宣言”的文章，提出第三代DBMS应具有三个基本特征：

第三代DBMS的基本特征

- 支持面向对象的数据模型

除提供传统的数据管理服务外，第三代数据库系统应支持数据管理、对象管理和知识管理，支持更加丰富的对象结构和规则，以提供更加强大的管理功能，支持更加复杂的数据类型。

- 保持或继承第二代数据库系统的优点

支持原有的数据管理，保持第二代数据库系统的非过程化的数据存取方式和数据独立性。

- 必须具有开放性

必须支持当前普遍承认的计算机技术标准，如支持SQL语言，支持多种网络标准协议。开放性还包括系统的可移植性、可连接性、可扩展性和可互操作性等。

第三代DBMS的发展趋势

- 分布式数据库：基于网络技术。将分散存放在不同地区的数据库组成一个逻辑上的整体数据库。例如SDD-1，DDM，POREL等。
- 演绎数据库：基于人工智能。能够根据已知的事实和规则进行推理，回答用户提出的各种问题。
- 面向对象数据库：基于面向对象的数学模型。能够封装具有相同特征的对象于一体，方便处理。
- 多媒体数据库
- 专家数据库
- 智能数据库

数据库技术的研究领域

- 数据库管理系统软件的研制

  包括工具软件和中间件的研制，提高系统的性能和提高用户的生产率。

- 数据库设计
  - 数据库的设计方法、设计工具和设计理论的研究；
  - 数据模型和数据建模的研究；
  - 计算机辅助数据库设计及其软件系统的研究；
  - 数据库设计规范和标准的研究等。
- 数据库理论

  包括关系规范化理论、关系数据理论等。数据库技术与人工智能技术、并行计算技术等的结合，促进了数据库理论的发展。

数据库系统的组成

数据库系统由数据（库）、用户、软件和硬件四部分组成。

- 数据（库）：它可以供用户共享，具有尽可能小的冗余度和较高的数据独立性，使得数据存储最优，数据最容易操作，并且具有完善的自我保护能力和数据恢复能力。
- 软件
  - DBMS是数据库系统的核心软件。
  - 数据库系统的各类人员对数据库的各种操作请求，都由DBMS完成。
- 硬件
  - 存储和运行数据库系统的硬件设备。包括CPU、内存、大容量的存储设备、外部设备等。

数据库系统的组成

❖用户：是指使用数据库的人，即对数据库的存储、维护和检索等操作。用户分为三类：

➢终端用户

主要是使用数据库的各级管理人员、工程技术人员、科研人员，一般为非计算机专业人员。

➢应用程序员

负责为终端用户设计和编制应用程序，以便终端用户对数据库进行存取操作。

➢数据库管理员（Database Administrator，简称DBA）

DBA是指全面负责数据库系统的管理、维护和正常使用的人员，其职责如下：

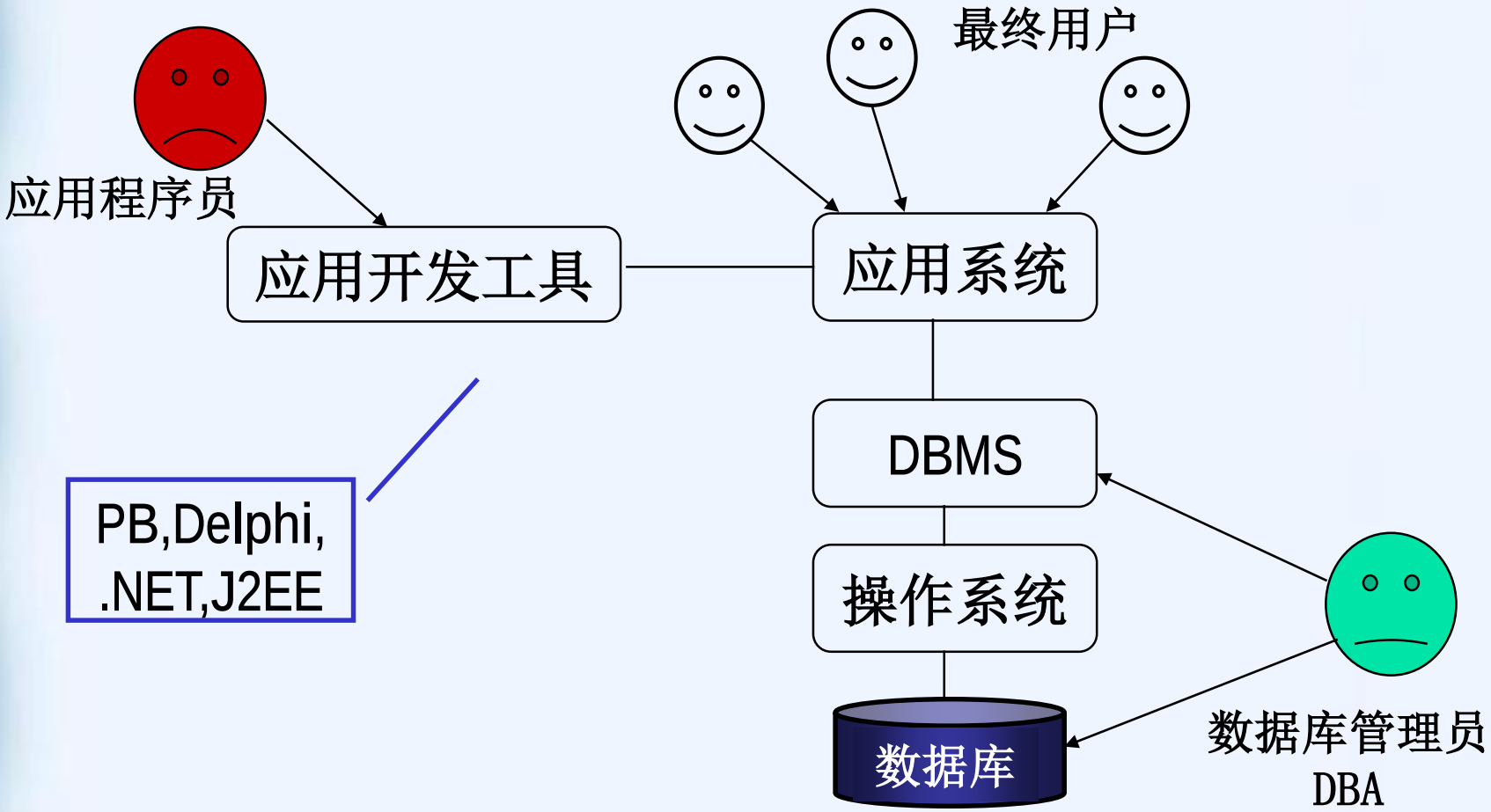
数据库管理员的职责

1. 参与数据库设计的全过程，决定数据库的结构和内容；
2. 定义数据的安全性和完整性，负责分配用户对数据库的使用权限和口令管理；
3. 监督控制数据库的使用和运行，改进和重新构造数据库系统。

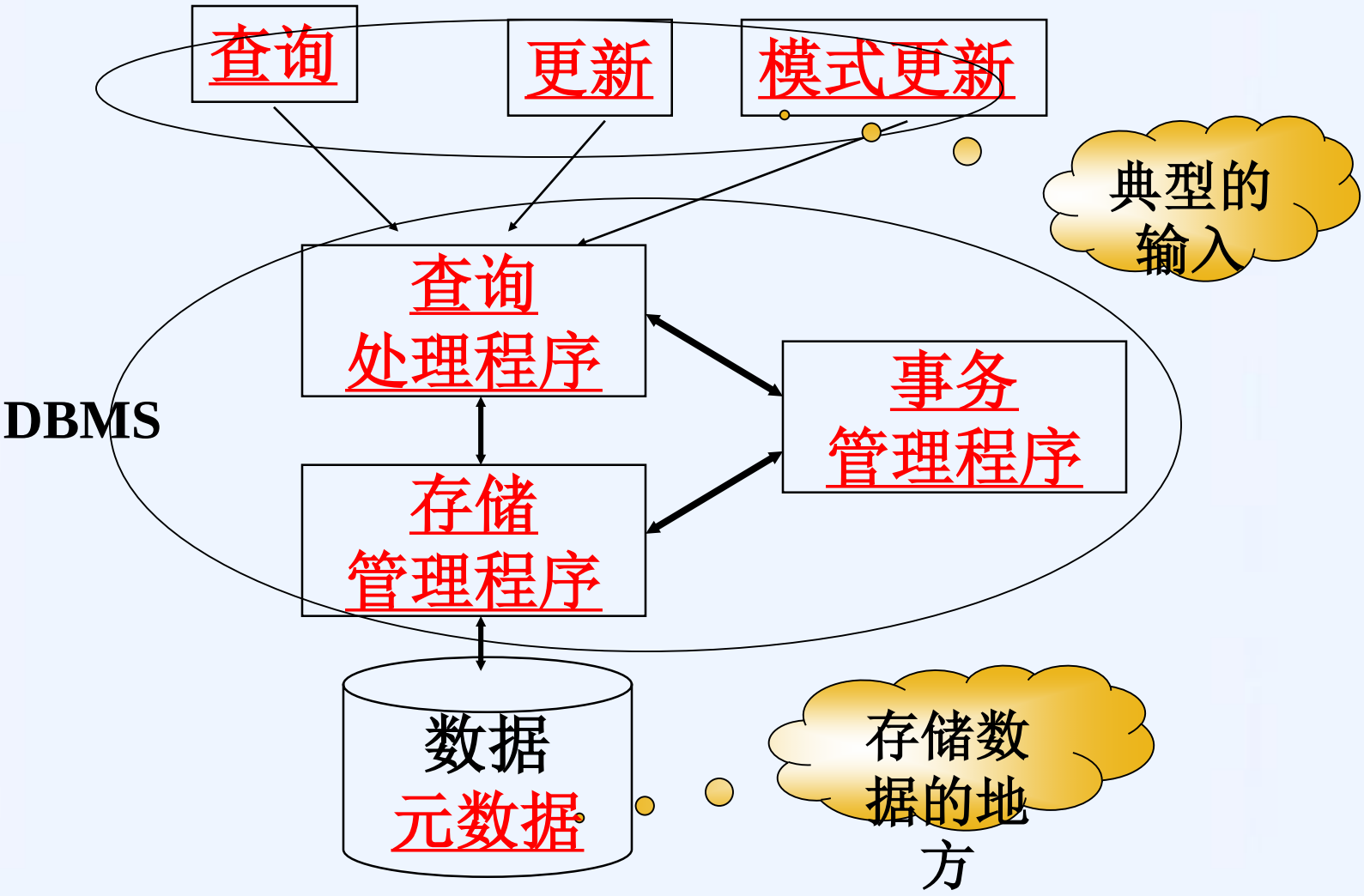
当数据库受到破坏时，应负责恢复数据库；当数据库的结构需要改变时，完成对数据结构的修改。

DBA要有较高的技术专长和较深的资历。

数据库系统的组成

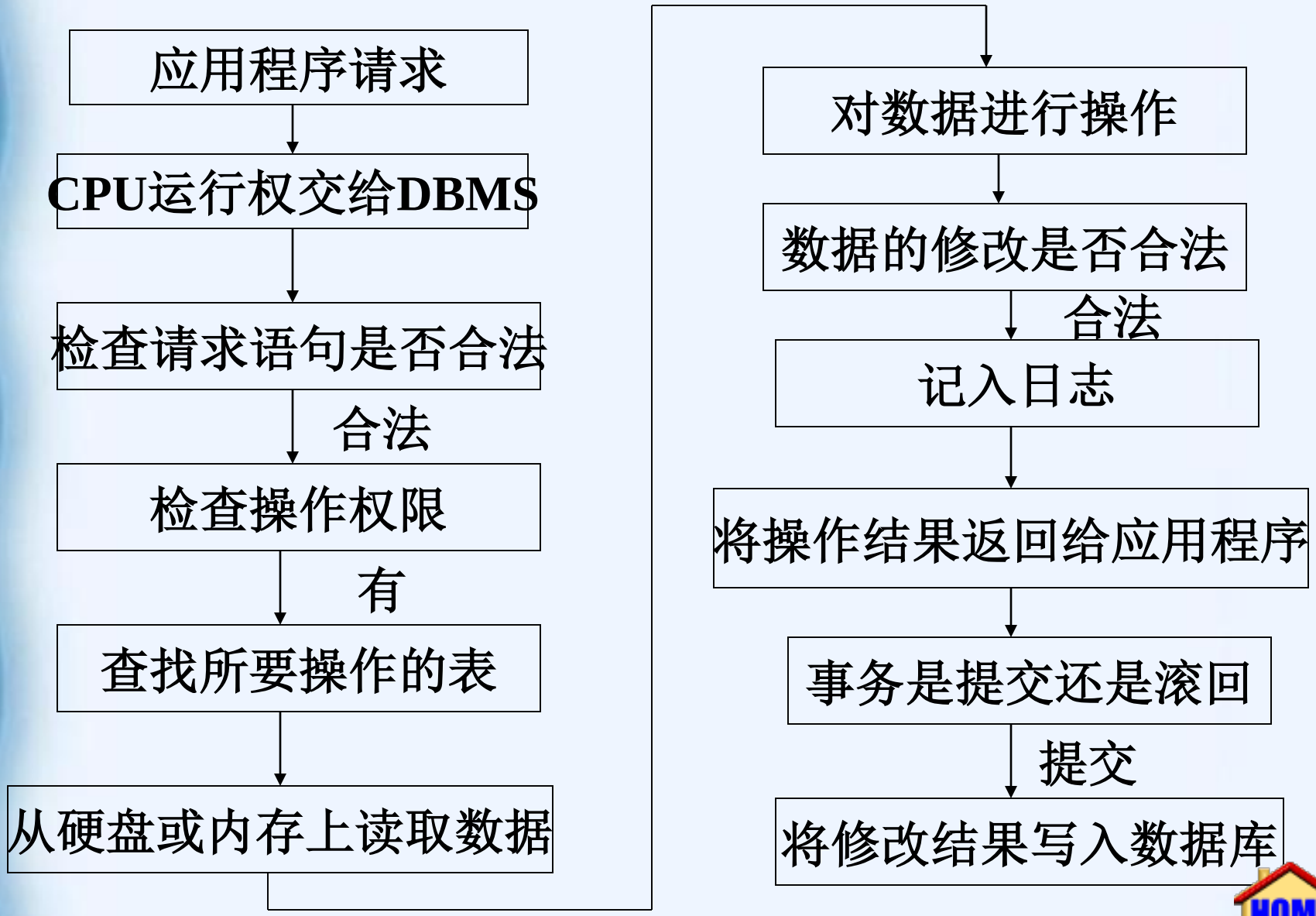

最终用户
应用程序员
应用开发工具
应用系统
DBMS
操作系统
数据库
PB,Delphi,
.NET,J2EE
数据库管理员
DBA

DBMS的组成和功能



数据库管理系统的组成

数据库系统的工作过程


应用程序请求
CPU运行权交给DBMS
检查请求语句是否合法
合法
检查操作权限
有
查找所要操作的表
从硬盘或内存上读取数据
对数据进行操作
数据的修改是否合法
合法
记入日志
将操作结果返回给应用程序
事务是提交还是滚回
提交
将修改结果写入数据库

元数据

metadata

元数据是有关数据结构的信息。

如果这个数据库管理系统是关系型的，则元数据包括关系名、关系的属性名以及属性的数据类型（整型或其它）。

查询

查询就是对数据的询问，有两种不同的生成方式:

- 通过通用的查询接口。比如，关系数据库管理系统允许用户键入SQL查询语句，然后将查询传给查询处理程序，并给出回答。

- 通过应用程序接口。典型的DBMS允许程序员通过应用程序调用DBMS来查询数据库。

比如，使用飞机订票系统的代理可以通过运行应用程序查询数据库了解航班的情况。可通过专门的接口提出查询要求，接口中也许包括填城市名和时间之类的对话框。

通过这种接口，只能进行应用程序已经定义好的查询，但对于功能性的查询，这种方式通常比直接写SQL语句更容易。

更新

更新是指更新数据的操作。

同查询一样，更新操作也可以通过通用的接口或应用程序接口来提出。

模式更新

模式更新一般由被授予了一定权限的人使用，例如数据库管理员，他们能够更改数据库模式或者建立新的数据库。

比如，假设原先学校的学生信息中没有存放学生的性别，而现在由于某种需要，必须增加这个信息，此时就需要在存放学生信息的关系（表）中加入一个新的属性来表示学生的性别。

模式的更新往往会导致应用程序的更新。

存储管理程序

存储管理程序的任务是从数据存储器获得想要查询的信息，并在接到上层的更新请求时更新相应的信息。存储管理程序包括：

- 文件管理程序 对文件在磁盘上的位置保持跟踪，并且负责取出含有缓冲区管理程序所要求的文件的一个或几个数据块。磁盘通常划分成一个个连续存储的数据块，每个数据块能容纳许多字节，从2^{12}到2^{14} (4K到16K)字节之间。

- 缓冲区管理程序 控制处理主存。它通过文件管理程序从磁盘取得数据块，并选择主存的一个页面存放其中的一块。缓冲区管理程序会把数据块在主存中保留一段时间，但当另一个数据块需要使用该页面，或者事务管理程序发出请求时，缓冲区管理程序也会把数据块写回磁盘。

查询处理程序

查询处理程序的任务是，把高级语言表示的查询或数据库操作(如SQL查询语句)转换成对存储器数据（如某个关系的特定元组或部分索引）的请求序列。

通常，查询处理任务最困难的部分是查询优化，也就是说选择好的查询计划，即对存储器系统选择好的请求序列以回答所要求的查询。

查询优化往往要利用现有的索引。

查询优化问题是数据库管理系统实现的重要方面之一。

事务管理程序

事务管理程序负责系统的完整性。它必须保证同时运行的若干个查询不互相冲突，保证系统在出现系统故障时不丢失数据。

事务管理程序要与查询处理程序互相配合，因为它必须知道当前查询将要操作的数据(以免出现冲突)，为了避免冲突的发生，也许需要延迟一些查询或操作。

事务管理程序也与存储管理程序互相配合，因为保护数据模式一般需要一个“日志”文件，记录历次数据的更新。如果操作顺序正确的话，日志文件将会记载更新的记录，从而使系统出现故障时根本没有执行的操作在其后能重新执行。

事务管理程序

典型的DBMS允许用户将一个或多个查询和/或更新组成事务(transaction)。事务，非正式地讲，是一组按顺序执行的操作单位。

数据库系统常常允许多个事务并发地执行。例如，有些事情可能在一家银行的所有ATM机器上同时执行。保证这些事务全都正确执行是DBMS中事务管理程序的任务。更详细地说，事务的“正确”执行还需要通常称为ACID的特性。

ACID取自于事务执行的四个主要需求的首字母。这四个特性是:

事务管理程序的ACID特性

- 原子性(Atomicity)

  要求整个事务都执行或者都不执行。

- 一致性(Consistency)

  要求数据符合客观世界的要求或限定条件。

- 隔离性(Isolation)

  当两个或更多的事务并发运行时，它们的作用效果必须互相分开。

  隔离性可利用加锁来实现。

- 持久性(Durability)

  如果事务已经完成，即便系统出现故障，事务的结果也不能丢失。

  持久性可利用日志和事务提交来实现。

练习题

请写出生活中需要用到数据库来管理数据和信息的场合，并考虑需要保存哪些信息。

加锁

造成事务间不独立的主要原因是两个或多个事务同时读写数据库中的同一数据项。

DBMS的事务管理程序能够对事务要访问的数据项加锁。一个事务对某数据项加锁后，其他的事务就不能访问它了。

加锁的单元称为锁的粒度。不同的**DBMS**可能规定不同大小的加锁粒度。

加锁的单元越大，一个事务必须等待另一个事务的可能性就越大，即使它们实际上并不访问同一数据。

而加锁的单元越小，加锁机制就越复杂。

日志

日志：包括每个事务的开始、每个事务所引起的数据库的更新和每个事务的结束的所有信息。

日志总是记在非易失性存储器上。

事务提交

为了保证持久性和原子性，事务一般以“试验”方式完成，事务提交时，更新操作的内容均已复制到日志中。该日志记录首先复制到磁盘上。然后才把更新的内容写入数据库本身。