

数据库系统原理及应用 (第 4 版)

数据库系统概论

Gavin-Yi LIU

<https://github.com/songzitea>

<https://gitee.com/songzitea>

Knowledge Artificial Intelligence(KAI)



本章目录

1. 课程导论

2. 认识数据库

2.1. 数据与数据管理

2.2. 数据库技术的产生与发展

3. 数据模型

3.1. 数据模型分层

3.2. 组成要素

4. 数据库系统的结构与组成

4.1. 数据库的三级模式

4.2. 数据库的两层映像

4.3. 数据库系统组成

5. 数据库领域的新技术

5.1. 云数据库与分布式数据库

5.2. 大数据与主动数据库

5.3. 数据仓库与数据挖掘

6. 参考文献

教学内容与课时安排

- 数据库系统概论 (4)
- 关系模型与关系代数 (4)
- SQL 查询语言、数据定义、更新及数据库编程 (6)
- 数据库建模 (6)
- 关系数据理论及模式求精 (6)
- 关系数据库设计实例——网上书店 (3)
- 数据库存储结构与查询处理 (3)
- 数据库安全性与完整性 (2)
- 事务管理与恢复 (4)

考核要求

课程考核¹: 平时考核、课堂测试和期末考试。

- 平时考核: 共 40 分, 具体安排如下:
 - 平时表现(10 分): 考核学生课前自学、提问的态度和能力, 考勤以及课堂上参与讨论的态度和表现, 充分调动学生自主学习的动力。
 - 平时作业(20 分): 考核学生 SQL 语句以及 SQL 编程的实际动手操作能力; 对重要难点章节布置课外作业, 由学生独立思考并完成。课堂上由学生对作业进行解答, 老师和其他同学参与讨论, 从而加深和巩固所学的知识点。
 - 课堂测试(10 分): 根据课程进度, 安排 1-3 次单元测验, 最终得分依据平均分来计算。
- 期末考试: 闭卷笔试(60 分)。
 - 通过期末考试来检查学生是否掌握了数据库的基本知识、基本理论和基本方法; 是否达到了本门课程的教学目的。

¹专业主干课程前提

数据管理技术的体系 (核心概念)

- 数据库管理系统 (database management system, DBMS) 是由一个相互关联的数据的集合和一组用以访问、管理和控制这些数据的程序(建立在操作系统之上的系统软件) 组成。
- 这个数据集合通常称为数据库 (database, DB)，其中包含了关于某个企业信息系统的所有信息。
- 设计 DBMS 的目的是为了有效地管理大量的数据，并解决操作系统的文件处理系统中存在的问题。
- 数据的有效管理，包括定义数据存储结构、提供数据操作机制。
 - 不仅需要解决数据的共享性、独立性和数据之间的联系问题；
 - 还需要解决数据的完整性、原子性、并发控制和安全性问题。
- 数据库系统 (database system, DBS)，是指在计算机系统中引入数据库后的系统，一般由数据库、数据库管理系统(及其应用开发工具)、应用系统、数据库管理员和最终用户构成。——人机系统

数据管理技术的体系

- 模型是主线
 - 概念模型：E-R 模型
 - 逻辑模型：关系模型（关系数据结构、关系操作、关系完整性约束）
 - 物理模型：存储结构、索引技术等
- 系统是核心（DBMS、DBS、应用系统）
 - 数据库管理系统 DBMS：存储结构与索引、查询与优化、完整性与安全、事务与恢复等
 - 数据库系统 DBS：数据库、数据库管理系统（及其应用开发工具）、应用系统、数据库管理员和最终用户等
 - 应用系统：企业运营、生产、供应链、客户关系、人力资源和财务管理等
- 应用是动力
 - 需求分析：业务需求及处理流程、功能需求及数据需求分析、业务规则分析等
 - 数据库设计：数据库概念模型、逻辑模型和物理模型等
 - 数据库应用开发：数据库应用系统的体系结构、常用数据库访问技术和数据库应用开发技术等

本章导读与学习目标

本章导读

主要介绍数据库系统**最基本、最重要的概念**。

- 什么是数据、数据管理、数据库、数据模型、数据独立性、数据库模式、数据库管理系统和数据库系统。
- **数据模型是数据库的组织基础**, 根据**数据抽象**的不同级别, 可以将数据模型划分为 3 层: 概念模型、逻辑模型和物理模型。
- **数据库是最基本的概念**, 在理解数据抽象的基础上掌握什么是数据库的三级模式和两层映像。
- **数据库管理系统是数据库系统的核**心, 数据库管理系统主要有哪些组成与功能;
- **数据库系统是数据库技术的应用系统**, 要求掌握数据库系统中各部分有什么作用, 特别是 DBA 的职责。

学习目标

- 数据库和数据库管理系统这两个最基本概念入手, 引出数据库管理系统所涉及的主要问题并做概括性讨论。
- 因此, 教学目标主要有两个,
 - 一是要求读者对数据库管理系统有一个初步的认识, 并了解数据库管理系统的基本功能;
 - 二是要求掌握数据抽象、数据模型、数据库模式等核心概念, 并理解这些概念在数据库管理系统中的地位和作用。

1. 课程导论
2. 认识数据库
 - 2.1. 数据与数据管理
 - 2.2. 数据库技术的产生与发展
3. 数据模型
 - 3.1. 数据模型分层
 - 3.2. 组成要素
4. 数据库系统的结构与组成
 - 4.1. 数据库的三级模式
 - 4.2. 数据库的两层映像
 - 4.3. 数据库系统组成
5. 数据库领域的新技术
 - 5.1. 云数据库与分布式数据库
 - 5.2. 大数据与主动数据库
 - 5.3. 数据仓库与数据挖掘
6. 参考文献

数据与信息

数据：描述事物的符号记录。

- 无结构的文本形式描述：张三，男，1980 年 9 月出生，上海人，现工作于 AI 科技研究所，高级工程师，主要研究兴趣包括大数据管理、情感分析。
显然，数据的表现形式不能完全表达其内容，其含义即语义需要经过解释才能被正确理解，因此数据和关于数据的解释是不可分的。
- 表格形式（有结构的记录形式）描述：

Table 1: 员工基本信息

姓名	性别	出生年月	籍贯	工作单位/部门	职称	研究方向
张三	男	1980 年 9 月	上海	AI 科技研究所	高级工程师	大数据管理、情感分析
李四	男	1988 年 12 月	北京	CoRobot 研究所	中级工程师	数据安全管理
王五	男	1990 年 11 月	深圳	XGEN 有限公司	初级工程师	大语言模型
...

表中一行数据组织在一起便构成一条记录，其数据的语义已由其所在列的表头栏目名解释，因此表格描述的数据称为结构化数据。

数据分类-1. 结构化数据

Definition (结构化数据)

- **结构化数据 (Structured Data)** 是指按照预定义的数据模型和格式进行组织，具有高度规则性和严格架构的数据。可以被高效地存储、查询和分析，通常存储在关系型数据库(如 MySQL, Oracle, SQL Server) 中，以二维表格的形式呈现。
- 简单来说，结构化数据就像是图书馆里整理好的图书：每本书都有固定的位置（索书号）、明确的标题、作者、出版社等信息，管理员可以快速准确地找到任何一本书。

特征：

- 预定义的模式 (Schema-on-Write)，这是最核心的特征。在存入数据之前，必须先严格定义好数据的结构/模式。包括：定义有哪些表，每个表里有哪些列，每一列的数据类型是什么（如：整数、字符串、日期等），以及哪些列是主键、外键等约束。结构化数据是“模式在先” (Schema-on-Write)，写入前必须先定义结构。
- 表格形式 (Tabular Format)：数据以行和列的二维表形式存在。
- 易于查询和分析：由于其规整的结构，使用 SQL(结构化查询语言) 可以非常高效、精确地进行复杂的查询、连接、过滤、聚合和数据分析。
- 数据完整性 (Data Integrity)：数据库可以通过约束 (Constraints) 来保证数据的准确性和可靠性。

数据分类-1. 结构化数据

优缺点

优点：

- **高效查询**: SQL 语言功能强大，查询速度极快，尤其适合复杂的分析。
- **数据一致性**: 通过事务处理 (ACID) 保证数据的准确和可靠。
- **易于理解和使用**: 结构清晰，业务人员和技术人员都容易理解。

缺点：

- **灵活性差**: 模式是固定的。如果需要增加一个新字段 (例如，给员工加一个“微信号”)，通常需要修改数据库结构，这在大型系统中可能是一个昂贵且耗时的操作。
- **扩展性挑战**: 关系型数据库在处理海量数据/大数据时，通常难以进行横向扩展 (scale-out)。

数据分类-2. 非结构化数据

Definition (非结构化数)

- 非结构化数据 (Unstructured Data) 是指没有预定义的数据模型或固定格式的信息。不像数据库表中的数据那样整齐地分成行和列。其内容通常是不规则、不完整的，并且其格式和含义需要借助特定技术或工具来解析。
- 简单来说，非结构化数据就像是一个杂物间：里面堆满了各种文本、图片、视频等，它们没有被分门别类地放在贴好标签的架子上。需要亲自查看每一样东西，才能知道它是什么以及有什么价值。

特征：

- 无预定义模式 (Schema-less)：这是最本质的特征。数据在生成和存储时没有任何强制性的结构要求。其结构是隐含在数据本身内部的。非结构化数据是“模式在后” (Schema-on-Read)，只有在需要读取和分析时，才去尝试理解和提取其结构。
- 格式多样且复杂：非结构化数据包含了各种形式和格式，从简单的文本文档到复杂的视频流，其内部结构千差万别，没有统一的标准。
- 不易用传统数据库存储和管理：关系型数据库的 2D 表模型无法直接有效地存储、查询非结构化数据。通常，被存储在文件系统、数据湖或专门的非结构化数据管理系统存储中。
- 需要高级技术进行分析：无法使用简单的 SQL 进行查询。提取其价值需要借助人工智能、机器学习、自然语言处理、计算机视觉和复杂的数据挖掘技术。

数据分类-2. 非结构化数据

优缺点

优点：

- **内容丰富**：包含了大量无法用表格表示的宝贵信息，如情感、意图、场景、趋势等。
- **灵活性高**：无需预先定义模式，可以容纳任何形式的信息。
- **价值潜力巨大**：是 AI 和数据分析的主要燃料，蕴含着揭示深层洞察的巨大潜力。

缺点：

- **难以管理**：存储、组织、备份和检索都比结构化数据复杂。
- **分析困难**：需要复杂且昂贵的技术才能提取价值，分析过程计算密集型。
- **数据质量不一**：内容可能不完整、不一致或有歧义，需要大量的数据清洗和预处理工作。

数据分类-3. 半结构化数据

Definition (半结构化数据)

- 半结构化数据 (Semi-structured Data) 是一种虽然不具有关系型数据库的严格表格结构，但包含标签、标记或其他格式来分隔数据元素并体现层次关系的数字信息。具有一定的结构性，但结构是灵活、自描述且可能变化的。
- 简单来说，半结构化数据就像是自助式仓库：仓库里的每个箱子大小形状不一样，但每个箱子上贴了一张清单，写明里面装了什么东西。

特征：

- 自描述性 (Self-Describing): 这是最核心的特征。数据本身携带着关于其自身结构的信息 (即元数据)。标签、键或标记与数据值紧密相连，共同存储。
- 模式在读时 (Schema-on-Read): 在存储数据时，不预先定义一个严格的、固定的模式。只有在读取或处理数据时，才根据需要去解析和理解其内在的结构。这提供了极大的灵活性。
- 层次性与嵌套性 (Hierarchical&Nested): 数据通常以树状或层级结构组织。一个对象内部可以包含另一个对象或数组。
- 结构灵活且可演化: 不同的数据记录拥有不同的属性。某些记录可能有某个字段，而其他记录可能没有，这是被允许的。可以相对容易地添加新的字段，而不会破坏现有的应用程序。

数据分类-3. 半结构化数据：典型例子与格式

最常见的半结构化数据格式包括：

- 1. JSON (JavaScript Object Notation): 当今 Web 和 API 通信的绝对主流。轻量级、易于人阅读和编写，也易于机器解析和生成。

```
json
{
  "id": 101,
  "name": "张三",
  "isActive": true,
  "department": "技术部",
  "skills": ["Java", "Python", "SQL"], // 这是一个数组
  "address": {                      // 这是一个嵌套对象
    "street": "科技路",
    "city": "北京"
  }
}
```

数据分类-3. 半结构化数据: 典型例子与格式

最常见的半结构化数据格式包括:

- 2. XML (eXtensible Markup Language):
在早期 Web 服务和配置文件领域非常流行, 比 JSON 更冗长但更严格。

```
<employee>
  <id>101</id>
  <name>张三</name>
  <department>技术部</
    ↪ department>
  <skills>
    <skill>Java</skill>
    <skill>Python</skill>
  </skills>
</employee>
```

- 3. YAML (YAML Ain't Markup Language):
常用于配置文件 (如 Docker Compose, Kubernetes), 强调可读性。

```
employee:
  id: 101
  name: 张三
  department: 技术部
  skills:
    - Java
    - Python
```

- 4. NoSQL 数据库中的文档: MongoDB、Couchbase 等文档数据库直接将 JSON/BSON 格式的文档作为基本存储单元。

数据分类对比

结构化、半结构化、非结构化数据的对比

Table 2: 数据分类对比

特性	结构化数据	半结构化数据	非结构化数据
模式	模式在先（写入前定义）	模式在后（读取时定义）	无模式
格式	严格，基于表格（行/列）	松散，自描述标签（如 JSON, XML）	原始，无固定格式
存储	关系型数据库 (RDBMS)	NoSQL 数据库、文件	文件系统、对象存储、数据湖
查询	SQL	类 SQL、特定 API	AI/ML, NLP, 计算机视觉
例子	SQL 数据库表	JSON 配置文件, XML 文档	图片、视频、PDF、邮件

数据处理与数据管理

Definition (数据处理)

从大量的、可能是杂乱无章的、难以理解的数据中抽取并推导出对于某些特定的人们来说有价值、有意义的数据。即：将数据转换成信息的过程，包括对数据进行采集、管理、加工、变换和传输等一系列活动。

数据处理的目的

- 其一是从大量的原始数据中抽取和推导出有价值的信息，作为决策的依据；
- 其二是借助计算机科学地保存和管理大量复杂的数据，便于人们能够充分地利用这些信息资源。

Definition (数据管理)

对数据进行有效的分类、组织、编码、存储、检索、维护和应用——数据处理的中心问题。

Tips:

- 数据管理是数据处理的核心。
- 对于这些数据管理操作，人们需要一个通用、高效且使用方便的管理软件，将数据有效地管理起来。
- 数据库技术正是瞄准这一目标，研究、发展并完善起来的。

数据库、数据库管理系统与数据库系统

- **数据库** (database, DB): 长期存储在计算机内，有组织的、可共享的大量数据的集合。这种集合按一定的数据模型组织、描述并长期存储，同时能够以安全可靠的方法对数据进行检索。数据库数据具有冗余度小、独立性高、延展性强、共享性好，以及结构化和永久性等特点。
- **数据库管理系统** (database management system, DBMS): 由一组相互关联的数据的集合和一组用以访问、管理和控制这些数据的程序组成；位于用户与操作系统之间的一层数据管理软件，用于科学地组织、存储和管理数据，并提供数据定义、数据操纵、数据控制、存储与管理等功能。
- **数据库系统** (Database System) 是指在计算机系统中引入数据库后的系统，它包括数据库、数据库管理系统、数据库应用程序、数据库管理员和用户等组成部分。

三者之间的关系

1. 数据库是数据的集合；
2. 数据库管理系统是管理和操作数据库的软件；
3. 数据库系统是包含数据库、数据库管理系统、应用程序和用户的整体系统。

1. 课程导论
2. 认识数据库
 - 2.1. 数据与数据管理
 - 2.2. 数据库技术的产生与发展
3. 数据模型
 - 3.1. 数据模型分层
 - 3.2. 组成要素
4. 数据库系统的结构与组成
 - 4.1. 数据库的三级模式
 - 4.2. 数据库的两层映像
 - 4.3. 数据库系统组成
5. 数据库领域的新技术
 - 5.1. 云数据库与分布式数据库
 - 5.2. 大数据与主动数据库
 - 5.3. 数据仓库与数据挖掘
6. 参考文献

人工管理阶段

- 20世纪50年代中期以前的这段时间。
- 计算机还很简陋，尚没有完整的操作系统，主要应用于科学计算。
- 软件方面只有汇编语言，没有操作系统和管理数据的软件，因此只能采用人工方式对数据进行管理。
- 数据是面向应用程序的，一个数据集只能对应于一个程序，程序与数据之间的关系如图1所示。
- 数据需要由应用程序自己定义和管理，没有相应的软件系统专门负责数据的管理工作。
- 当多个应用程序涉及某些相同的数据时，必须由各自的应用程序分别定义和管理这些数据，无法共享利用，因此存在大量冗余数据。

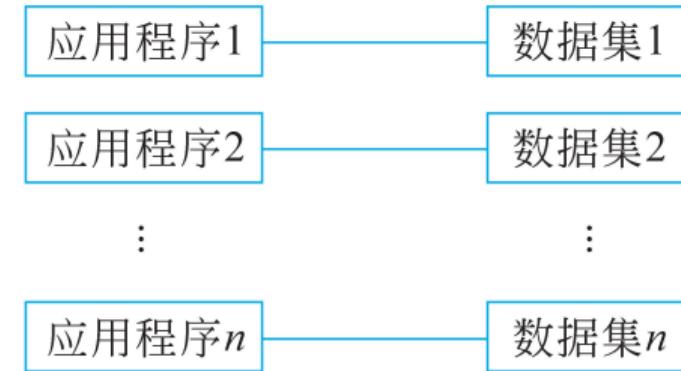


Figure 1: 人工管理阶段应用程序与数据之间的对应关系

人工管理数据的特点

数据不保存、不共享、不具独立性和无专门的数据管理软件

文件系统阶段

- 20世纪50年代后期到60年代中期的这段时间。
- 计算机除了应用于科学计算外，已开始应用于数据管理
- 文件系统阶段程序与数据之间的关系如图2所示。
- 在操作系统之上建立的文件系统已经成熟并广泛应用，数据由专门的软件进行统一管理。
- 对于一个特定的应用，数据被集中组织存放在多个数据文件（以后简称为文件或文件组）中，并针对该文件组来开发特定的应用程序。
- 利用“按文件名访问，按记录进行存取”的管理技术，可以对文件进行记录的修改、插入和删除等操作。

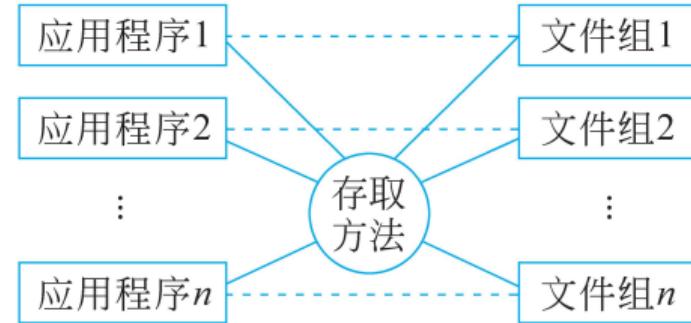


Figure 2: 文件系统阶段应用程序与数据之间的对应关系

文件系统的主要特点

- 文件系统实现了**文件内的结构性**，即一个文件内的数据是按记录进行组织的，这样的数据是有**结构的(语义的)**，数据的**语义**是明确的。
- 整体上还是**无结构的**，即**多个文件之间是相互独立的**，无法建立**全局的结构化数据管理模式**。
- 程序和数据之间由**文件系统提供的存取方法**进行转换，程序员可以不必过多地考虑数据的**物理存储细节**。
- 由于数据在**物理存储结构**上的改变不一定反映在程序上，因此应用程序与数据之间有了一定的**物理独立性**。

文件系统的弊端

- 数据共享性差，数据冗余和不一致
 - 数据冗余是指相同的数据在不同的地方（文件）重复存储
 - 文件系统中的一个（或一组）文件基本上对应于一个应用程序，不同应用程序之间很难共享相同数据。
 - 如何有效地提高不同应用共享数据的能力成为急需解决的问题
- 数据独立性差
 - 文件系统中的文件组是为某一特定应用服务的，其逻辑结构对于该特定应用程序来说是优化的，但是系统不易扩充。因此，数据与应用程序之间缺乏逻辑独立性
 - 如何有效地提高数据与应用程序之间的独立性成为急需解决的问题
- 数据孤立，数据获取困难
 - 对于数据与数据之间的联系，文件系统仍缺乏有效的管理手段
 - 如何有效地管理数据与数据之间的联系成为急需解决的问题
- 完整性问题
 - 数据的完整性是指数据的正确性、有效性和相容性，也称为一致性约束
 - 例如，一个学生需要选修某门课程，该学生必须已经修过了该课程规定的先修课程时才能选修（因为课程之间存在先修后修关系）；必须在该教学班尚未选满时才能选修（因为教室容量有限）；必须在时间上与其它已经选修的课程不冲突时才能选修
 - 如何有效地表达和实现一致性（即完整性）约束成为急需解决的问题

文件系统的弊端 (CONT.)

- 安全性问题
 - 一个系统可能有很多用户，不同用户可能只允许其访问一部分数据，即该用户只有一部分数据的访问权限
 - 如何有效地保障数据的安全性就成为急需解决的问题
- 原子性问题
 - 计算机系统有时会发生故障，一旦故障发生并被检测到，数据就应该恢复到故障发生前的状态
 - 例如，学生选课时，不仅要在选课文件中增加某学生选修某门课的记录，同时也要在该课程教学班记录中将已选课人数加 1，以便学生选课时进行容量控制
 - 因此，增加选课记录与选课人数加 1 两个操作要么都发生，要么都不发生，这就是学生选课操作的原子性要求
 - 如何有效地保障操作的原子性就成为急需解决的问题
- 并发访问异常
 - 系统应该允许多个用户同时访问数据，在这样的环境中由于并发更新操作相互影响，可能会导致数据的不一致
 - 如何有效地进行并发控制（即确保并发操作正确性）就成为急需解决的问题

数据库管理系统阶段

- 20世纪60年代后期以来
- 数据管理对象的规模越来越大，应用范围越来越广，多种应用共享数据的要求越来越强烈
- **数据库管理系统(DBMS)** 是由一个相**互关联的数据的集合**和一组用以**访问、管理和控制这些数据的程序**组成
- 这个数据集合通常称为**数据库**(database, DB)，其中包含了**关于某个企业信息系统的所有信息**。
- DBMS 是位于用户与操作系统之间的一层**数据管理软件**，它提供一个可以**方便且高效地存取、管理和控制数据库信息的环境**
- DBMS 和操作系统一样，都是计算机的基础软件(系统软件)，也是一个大型复杂的软件系统

数据库管理系统阶段

- 设计数据库管理系统的目的是为了有效地管理大量的数据，
 - 既涉及到数据存储结构的定义，
 - 又涉及到数据操作机制的提供。
- 解决文件处理系统中存在的问题：
 - 数据共享性差 (数据冗余和不一致)
 - 数据独立性差
 - 数据孤立和数据获取困难
 - 完整性问题
 - 原子性问题
 - 并发访问异常
 - 安全性问题

数据库管理系统的主要特点

- **数据结构化**。数据库管理系统实现数据的整体结构化，这是数据库的主要特征之一，也是数据库管理系统与文件系统的**本质区别**。
 - 一是指：数据不仅仅是**内部结构化**，而是将**数据以及数据之间的联系**统一管理起来，使之结构化。如图3所示。

学生文件Student的记录结构

学号	姓名	性别	出生日期	所学专业	家庭住址	联系电话
----	----	----	------	------	------	------

课程文件Course的记录结构

课程号	课程名称	学时	学分	教材名称
-----	------	----	----	------

学生成绩文件Score的记录结构

学号	课程号	学期	成绩
----	-----	----	----

Figure 3: 学生、课程、学生成绩文件结构

- 二是指：在数据库中的数据**不是仅仅针对某一个应用，而是面向全组织的所有应用**。
 - 例如，一个学校的信息系统中不仅要考虑教务处的**学生成绩管理**，还要考虑学工处的**学籍注册管理、学生奖惩管理、学生家庭成员管理**，以及财务处的**学生缴费管理**；同时还要考虑研究生

数据库管理系统的主要特点 (CONT.)

- **数据结构化**。数据库管理系统实现数据的整体结构化，这是数据库的主要特征之一，也是数据库管理系统与文件系统的**本质区别**。
 - (续) 院的**研究生管理**、科研处的**科研管理**、人事处的**教职工人事管理和工资管理**等。
 - 因此，学校信息系统中的学生数据要**面向全校各个职能部门和院系的应用**，而不仅仅是教务处的一个学生成绩管理应用，如图4所示。



Figure 4: 某校信息管理系统中的学生数据

数据库管理系统的主要特点 (CONT.)

- 数据的共享度高，冗余度底，易扩充
 - 数据库管理系统从整体角度描述和组织数据，数据不再是面向某个应用，而是面向整个系统
 - 因此，数据可以被多个用户、多个应用共享使用
 - 数据共享可以大大减少数据的冗余，避免数据之间的不一致性
- 数据独立性高
 - 数据独立是指数据的使用(即应用程序)与数据的说明(即数据的组织结构与存储方式)分离
 - ▶ 这样，应用程序只需要考虑如何使用数据，而无须关心数据库中的数据是如何构造和存储的
 - ▶ 因而，各方(在一定范围内)的变更互不影响
 - 数据独立性用来描述应用程序与数据结构之间的依赖程度，包括数据的物理独立性和数据的逻辑独立性，依赖程度越低则独立性越高。
 - 物理独立性是指用户的应用程序与数据库中数据的物理结构是相互独立的。数据在磁盘上如何组织和存储由 DBMS 负责，应用程序只关心数据的逻辑结构；当数据的物理存储结构改变时，应用程序不用修改
 - 逻辑独立性是指用户的应用程序与数据库中数据的逻辑结构是相互独立的。数据的(全局)逻辑结构由 DBMS 负责，应用程序只关心数据的局部逻辑结构(即应用视图)，数据的(全局)逻辑结构改变了，应用程序也可以不用修改

数据库管理系统的主要特点 (CONT.)

- 数据由数据库管理系统 (DBMS) 统一管理和控制

- **数据的安全性保护**: 保护数据以防止不合法的使用造成数据的泄密和破坏
- **数据的完整性检查**: 将数据控制在有效的范围内, 或保证数据之间满足一定的关系
- **并发控制**: 对多个用户或应用同时访问同一个数据的并发操作加以控制和协调, 确保得到正确的修改结果或数据库的完整性不遭到破坏
- **数据库恢复**: 当计算机系统发生硬件或软件故障时, 需要将数据库从错误状态恢复到某一已经正确状态

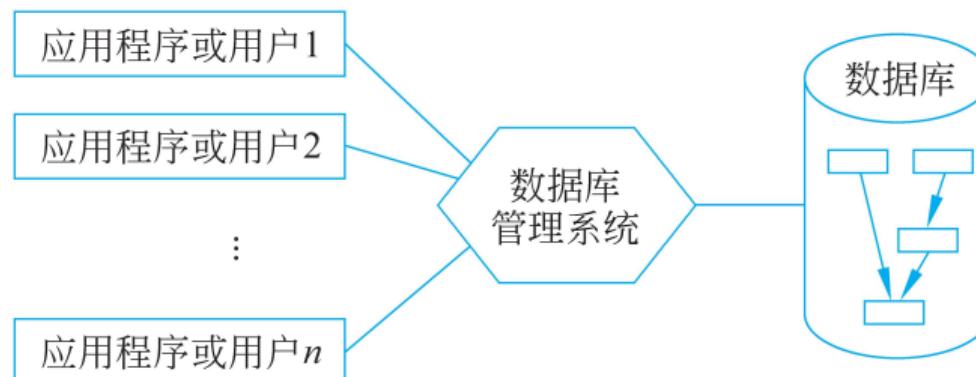


Figure 5: 数据库管理系统应用程序与数据之间的对应关系

- 图书馆管理: 用于存储图书馆的馆藏资料(图书、期刊等)、读者(教师、学生等)信息, 以及图书和期刊的借阅、归还记录等, 方便读者查找资料, 方便管理人员办理图书和期刊的借阅、归还和催还等手续, 提高图书馆管理水平
- 科研管理: 用于存储教师信息、科研成果记录等, 方便科研成果的考核、检索和统计工作
- 银行管理: 用于存储客户信息、存款账户和贷款账户记录以及银行之间的转账交易记录等, 提高存款、贷款管理水平, 加速资金流转和银行结算
- 固定资产管理: 用于存储客户信息、部门信息和员工信息, 固定资产的采购记录、领用记录和报废记录等, 自动计提固定资产折旧, 提供各种固定资产报表
- 人力资源管理: 用于存储部门信息、员工信息, 以及出勤记录、计件记录等, 自动计算员工的工资、所得税和津贴, 产生工资单

数据模型的分类

- 数据库结构的基础是**数据模型**(data model)
- **数据模型**是一个描述**数据结构**、**数据操作**以及**数据约束**的**数学形式体系**(即**概念及其符号表示系统**)
 - **数据结构**用于刻画数据、数据语义以及数据与数据之间的联系
 - **数据约束**是对数据结构和数据操作的一致性、完整性约束，亦称为**数据完整性约束**
- 通过**数据模型**可以对现实世界的数据特征进行抽象。
- 根据**数据抽象**的不同级别，将**数据模型**划分为 3 类：
 - **概念模型**：概念层次的数据模型，也称为**信息模型**
 - **逻辑模型**：用于描述数据库数据的整体逻辑结构
 - **物理模型**：用来描述数据的**物理存储结构**和**存取方法**

1. 课程导论
2. 认识数据库
 - 2.1. 数据与数据管理
 - 2.2. 数据库技术的产生与发展
3. 数据模型
 - 3.1. 数据模型分层
 - 3.2. 组成要素
4. 数据库系统的结构与组成
 - 4.1. 数据库的三级模式
 - 4.2. 数据库的两层映像
 - 4.3. 数据库系统组成
5. 数据库领域的新技术
 - 5.1. 云数据库与分布式数据库
 - 5.2. 大数据与主动数据库
 - 5.3. 数据仓库与数据挖掘
6. 参考文献

概念模型

- 按用户的观点或认识对现实世界的数据和信息进行建模
- 主要用于数据库设计——强调语义表达功能
- 常用的概念模型有实体-联系模型(E-R 模型) 和面向对象模型(OO 模型)
- E-R 模型基于对现实世界的如下认识：现实世界是由一组称作实体的基本对象以及这些对象间的联系构成
 - 实体是现实世界中可区别于其他对象的一件“事情”或一个“物体”
 - 如，一个学生、一个部门、一个教室、一种商品、一本书、一门课程，以及一次选课、采购、销售、存款业务（记录）等都是实体
- OO 模型是用面向对象观点来描述现实世界实体(对象)的逻辑组织、对象间限制和联系等的模型
 - 对象是由一组数据结构和在这组数据结构上操作的程序代码封装起来的基本单位

逻辑模型

- 逻辑模型是用户通过数据库管理系统看到的现实世界，是按计算机系统的观点对数据建模，即数据的计算机实现形式
- 主要用于DBMS 的实现。它既要考虑用户容易理解，又要考虑便于 DBMS 实现
- 不同的 DBMS 提供不同的逻辑数据模型
 - 层次模型 (hierarchical model)
 - 网状模型 (network model)
 - 关系模型 (relational model)
 - 面向对象模型 (即 OO 模型)
 - XML 模型
 - 对象关系模型 (object relational model)

物理模型

- 物理层是数据抽象的最低层，用来描述数据的物理存储结构和存取方法。
- 例如，一个数据库中的数据和索引是存放在不同的数据段上还是相同的数据段上；数据的物理记录格式是变长的还是定长的；数据是否压缩存储；索引结构是 B+ 树还是 Hash 结构等
- 物理模型的具体实现是 DBMS 的任务，数据库设计人员要了解和选择物理模型，一般用户则不必考虑物理层细节。

物理模型

适用对象

- 从现实世界到概念模型的转换是由**数据库设计人员**完成
- 从概念模型到逻辑模型的转换可以由**数据库设计人员**完成，也可以用**数据库设计工具协助**设计人员完成
- 从逻辑模型到物理模型的转换一般由**DBMS**来完成

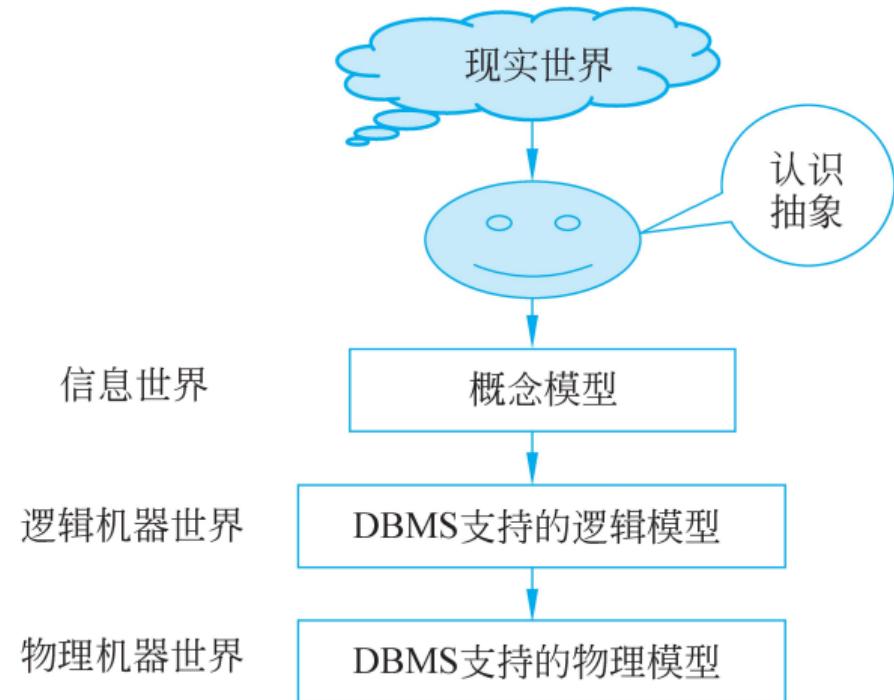


Figure 6: 数据库管理系统应用程序与数据之间的对应关系

1. 课程导论
2. 认识数据库
 - 2.1. 数据与数据管理
 - 2.2. 数据库技术的产生与发展
3. 数据模型
 - 3.1. 数据模型分层
 - 3.2. 组成要素
4. 数据库系统的结构与组成
 - 4.1. 数据库的三级模式
 - 4.2. 数据库的两层映像
 - 4.3. 数据库系统组成
5. 数据库领域的新技术
 - 5.1. 云数据库与分布式数据库
 - 5.2. 大数据与主动数据库
 - 5.3. 数据仓库与数据挖掘
6. 参考文献

数据模型的组成要素

- 数据模型是一个描述数据结构、数据操作以及数据完整性约束的数学形式体系（即概念及其符号表示系统）。
- 这些概念及其符号表示系统精确地描述了系统的静态特性、动态特性和完整性规则。
- 数据模型的组成要素有：
 - 数据结构：描述数据库的组成对象以及对象之间的联系。
 - 数据操作：指对数据库中各种对象（型）的实例（值）允许执行的操作集合，包括操作及有关的操作规。
 - 数据完整性约束：一组数据完整性规则，是数据、数据语义和数据联系所具有的制约和依存规则，包括数据结构完整性规则和数据操作完整性规则，用以限定符合数据模型的数据库状态以及状态的变化，以保证数据库中数据的正确、有效和相容。

层次模型

- 典型代表是 1968 年 IBM 公司推出的第一个大型商用数据库管理系统——IMS(information management system)
- 层次模型用**树形结构**来表示各类实体以及实体间的联系。实体用记录来表示，实体间的联系用链接(可看作指针)来表示。
- 满足如下**两个条件**的基本层次联系的集合为层次模型：
 - 有且只有一个结点没有双亲结点，这个结点称为根结点。
 - 根以外的其他结点有且只有一个双亲结点。
- 在层次模型中，**每个结点表示一个记录型**，记录(型)之间的联系用结点之间的连线(有向边)表示，这种联系是父子之间的一对多的联系
- 每个**记录型**由若干个**字段**组成，**记录型**描述的是**实体**，**字段**描述的是**实体的属性**。每个记录型可以定义一个排序字段，也称为码字段，如果所定义的排序字段的值唯一，则它也可以用来唯一标识一个记录值

层次模型

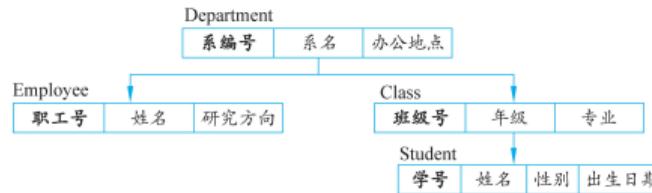


Figure 7: 简单的教学管理系统的层次数据模型

The diagram shows the actual data instances for the hierarchical model. The 'Department' table has two entries: D05 (计算机系, 信息大楼) and D08 (会计系, 管理大楼). The 'Employee' table has three entries: E0501 (万家乐, 数据库), E0502 (吴文君, 信息检索), and E0503 (廖兴旺, 操作系统). The 'Class' table has two entries: CS2101 (2021, 计算机) and IS2202 (2022, 信息系统). The 'Student' table has four entries: 2101001 (李小勇, 男, 12/21/2003), 2101008 (王红, 男, 04/26/2005), 2202002 (刘方晨, 女, 11/22/2003), and 2202005 (王红敏, 女, 10/01/2003). A separate 'Student' table at the bottom right shows entries for 2202014 (刘宏昊, 男, 09/16/2004).

Department			
D05	计算机系	信息大楼	
	D08	会计系	管理大楼
Employee			
E0501	万家乐	数据库	
E0502	吴文君	信息检索	
E0503	廖兴旺	操作系统	
Class			
CS2101	2021	计算机	
IS2202	2022	信息系统	
Student			
2202002	刘方晨	女	11/22/2003
2202005	王红敏	女	10/01/2003
2202014	刘宏昊	男	09/16/2004
Student			
2101001	李小勇	男	12/21/2003
2101008	王红	男	04/26/2005

Figure 8: 简单的教学管理系统的实例值

层次模型优缺点

优点：

- 数据结构比较简单清晰
- 查询效率高
- 提供了良好的数据完整性支持

缺点：

- 现实世界中很多联系是非层次的（如多对多联系），层次模型在表示这类联系时，解决的办法：
 - 一是通过引入冗余数据（易产生不一致性），
 - 二是创建非自然的数据结构（引入虚拟结点）。对插入和删除操作的限制比较多，因此应用程序的编写比较复杂
- 查询孩子结点必须通过双亲结点
- 由于结构严密，层次命令趋于程序化

网状模型

- 典型代表是DBTG 系统，亦称为CODASYL 系统，它是 20 世纪 70 年代由数据系统语言研究会 (conference on data system language, CODASYL) 下属的数据库任务组 (data base task group, DBTG) 提出的一个系统方案
- 满足如下两个条件的基本层次联系的集合称为网状模型
 - 允许一个以上的结点无双亲。
 - 一个结点可以有多个双亲。
- 网状模型是一种比层次模型更具普遍性的结构，它去掉了层次模型的两个限制，还允许两个结点之间有多种联系 (称为复合联系)。因此，网状模型可以更直接地去描述现实世界。

网状模型的主要优点：

- 能够更为直接地描述现实世界
- 具有良好的性能，存取效率较高

网状模型的主要缺点：

- 结构比较复杂，而且随着应用规模的扩大，数据库的结构会变得越来越复杂，不利于最终用户掌握
- 操作语言比较复杂

关系模型

- 1970 年美国 IBM 公司 San Jose 研究室的研究员 E. F. Codd 首次提出了数据库管理系统的
关系模型，开创了数据库关系方法和关系数据理论的研究，为数据库技术奠定了理论基
础。
- 由于 E. F. Codd 的杰出工作，他于 1981 年获得 ACM 图灵奖
- 20 世纪 80 年代以来，计算机厂商新推出的数据库管理系统几乎都支持关系模型，数据
库领域当前的研究工作也都是以关系方法为基础

关系数据模型的数据结构

关系模型中的常用术语：

- **关系**(relation)：一个关系对应一张二维表，每一个关系有一个名称，即关系名；
- **元组**(tuple)：表中的一行称为一个元组；
- **属性**(attribute)：表中的一列称为一个属性，每一个属性有一个名称，即属性名；
- **码**(key)：也称为码键或键。表中的某个属性或属性组，**它可以唯一地确定关系中的一个元组**，如关系 Student 中的**学号**，它可以唯一地标识一个学生；
- **域**(domain)：属性的取值范围；
- **分量**(component)：元组中的一个属性值；
- **外码**(foreign key)：表中的**某个属性或属性组**，用来描述**本关系中的元组(实体)**与**另一关系中的元组(实体)**之间的联系
 - 外码的取值范围对应于另一个关系的码的取值范围的子集。
 - 如关系 Score 中的学号，它描述了关系 Score 与关系 Student 的联系（即哪个学生选修了课程），因此学号是关系 Score 的外码。
 - 同理，课程号也是关系 Score 的外码，它描述了关系 Score 与关系 Course 的联系（即哪门课程被学生选修了）。

关系数据模型的数据结构

- **关系模式**(relational schema): 通过**关系名**和**属性名列表**对关系进行描述，即二维表的**表头部分**(表格的描述部分)

- 关系模式的一般形式：

关系名 (属性名 1, 属性名 2, …, 属性名 n)

- 关系模式 Student、Course 和 Score 可分别描述为：

- Student(学号, 姓名, 性别, 出生日期, 所学专业)
- Course(课程号, 课程名称, 学时, 学分)
- Score(学号, 课程号, 学期, 成绩)

- **关系模型要求关系必须是规范化的**，即要求**关系必须满足一定的规范条件**。

- 最基本的**规范条件**是：

- 关系的每一个元组必须是**可分区的**，即**存在码属性**。
- 关系的每一个属性(即元组的分量)必须是**一个不可分的数据项**，即**不允许表中有表**。

关系模型

Student关系

学号	姓名	性别	出生日期	所学专业
2101001	李小勇	男	2003-12-21	计算机
2101008	王红	男	2005-04-26	计算机
2103010	李宏冰	女	2005-03-09	会计学
2103045	王红	男	2005-04-26	会计学
2202002	刘方晨	女	2003-11-11	信息系统
2202005	王红敏	女	2003-10-01	信息系统
⋮	⋮	⋮	⋮	⋮

Course关系

课程号	课程名称	学时	学分
AC001	基础会计	48	3
CS012	操作系统	80	5
CS015	数据库系统概论	64	4
⋮	⋮	⋮	⋮

Score关系

学号	课程号	学期	成绩
2101001	CS012	22231	88
2101001	CS015	22232	92
2101008	AC001	21222	78
2101008	CS012	22231	93
2101008	CS015	22232	86
2103010	AC001	21221	92
2103045	AC001	21221	84
2202002	AC001	22232	95
2202002	CS012	23241	85
2202005	AC001	22232	88
2202005	CS012	23241	72
⋮	⋮	⋮	⋮

Figure 9: 关系数据模型的数据结构

关系数据模型的操作

- 关系数据模型的操作

- 关系数据模型的操作主要包括查询和更新（插入、删除和修改）

- 关系：元组的集合

- ▶ 关系模型的数据操作是集合操作，
 - ▶ 操作对象和操作结果都是关系（元组的集合）

- 关系模型：存取路径是透明的

- ▶ 用户只要指出“干什么”或“找什么”，
 - ▶ 不必说明“怎么干”或“怎么找”，
 - ▶ 从而大大提高了数据的独立性，提高了软件的开发和维护效率

- 关系数据模型的完整性约束

- 实体完整性、参照完整性和用户自定义完整性

关系数据模型的优点

优点：

- **严格的数学基础**: 有关系代数作为语言模型, 有关系数据理论作为理论基础
- **概念单一**
 - 无论实体还是实体之间的联系都是用关系来表示,
 - 对数据(关系)的操作(查询和更新)结果还是关系。
 - 所以其数据结构简单、清晰, 用户易懂易用
- **存取路径透明**: 具有更高的数据独立性、更好的安全保密性, 简化了程序员的工作, 提高了软件的开发和维护效率

缺点：

- 由于存取路径对用户透明, **查询效率**往往不如非关系数据模型
- 为了提高性能, DBMS 必须对用户的查询请求进行**查询优化**, 这样就增加了 DBMS 的开发难度

1. 课程导论
2. 认识数据库
 - 2.1. 数据与数据管理
 - 2.2. 数据库技术的产生与发展
3. 数据模型
 - 3.1. 数据模型分层
 - 3.2. 组成要素
4. **数据库系统的结构与组成**
 - 4.1. **数据库的三级模式**
 - 4.2. 数据库的两层映像
 - 4.3. 数据库系统组成
5. **数据库领域的新技术**
 - 5.1. 云数据库与分布式数据库
 - 5.2. 大数据与主动数据库
 - 5.3. 数据仓库与数据挖掘
6. 参考文献

- DBMS：隐藏关于数据存储和维护的某些细节，为用户提供数据在不同层次上的视图，即**数据抽象**，方便不同的使用者可以从不同的角度去观察和利用数据库中的数据
- 物理层抽象：最低层次的抽象，描述数据实际上是怎样存储的。
- 逻辑层抽象
 - 描述数据库中存储什么数据以及这些数据之间存在什么关联。
 - 提供给数据库管理员和数据库应用开发人员使用的，必须明确知道数据库中应该保存哪些信息。
- 视图层抽象
 - 最高层次的抽象，只描述整个数据库的某个部分，即**局部逻辑结构**。
 - 系统可以为同一数据库提供多个视图，每一个视图对应一个具体的应用，亦称为**应用视图**。

数据库的三级模式

- 根据数据抽象的 3 个不同级别，DBMS 也应该提供观察数据库的 3 个不同角度，以方便不同的用户使用数据库的需要。这就是数据库的三级模式结构。

外模式

- 也称子模式或用户模式，对应于视图层数据抽象
- 是数据库用户（包括应用程序员和最终用户）能够看见和使用的局部数据的逻辑结构和特征的描述。,
- 是数据库用户的[数据视图](#)，是与某一具体应用有关的数据的逻辑表示。
- 外模式是保证数据库安全性的一个有力措施**，每个用户只能看见和访问所对应的外模式中的数据，数据库中的其余数据是不可见的。

模式

- 也称为逻辑模式，对应于逻辑层数据抽象，是数据库中全体数据的逻辑结构和特征的描述，是所有用户的公共数据视图。
- 模式的一个具体值称为模式的一个实例(instance)
- 它是DBMS 模式结构的中间层，既不涉及数据的物理存储细节和硬件环境，也与具体的应用程序、所使用的应用开发工具及高级程序设计语言无关。

内模式

- 也称存储模式，对应于物理层数据抽象，
- 是数据的物理结构和存储方式的描述，
- 是数据在数据库内部的表示方式。

数据库的三级模式

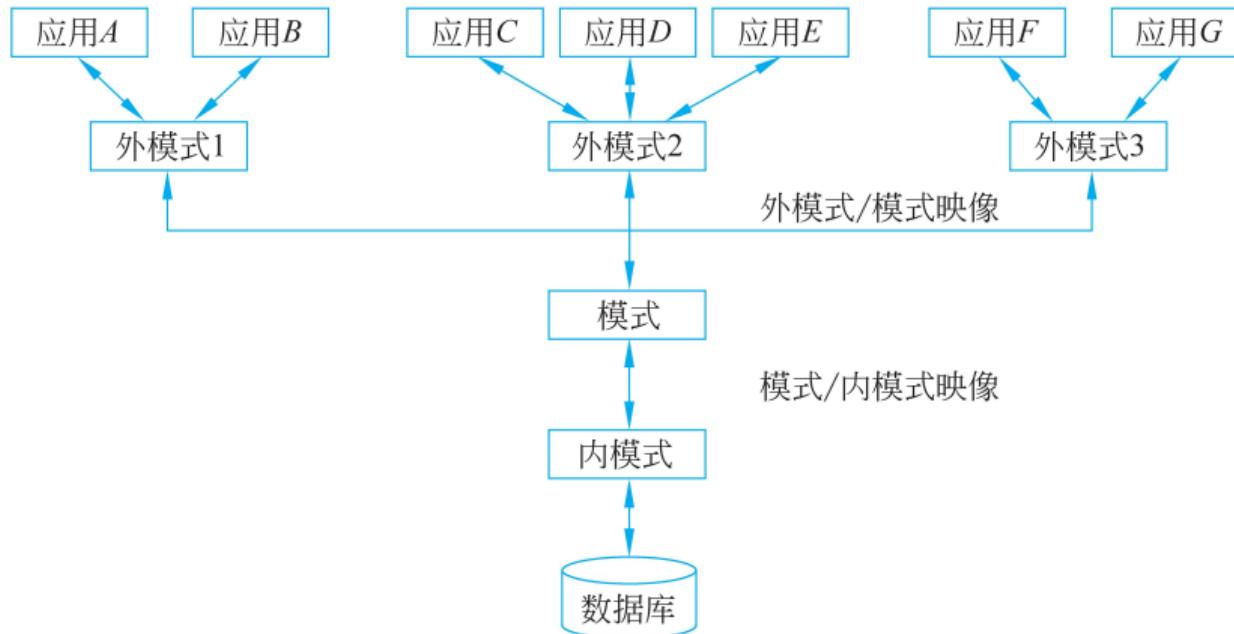


Figure 10: 数据库的三级模式结构

1. 课程导论
2. 认识数据库
 - 2.1. 数据与数据管理
 - 2.2. 数据库技术的产生与发展
3. 数据模型
 - 3.1. 数据模型分层
 - 3.2. 组成要素
4. **数据库系统的结构与组成**
 - 4.1. 数据库的三级模式
 - 4.2. **数据库的两层映像**
 - 4.3. 数据库系统组成
5. 数据库领域的新技术
 - 5.1. 云数据库与分布式数据库
 - 5.2. 大数据与主动数据库
 - 5.3. 数据仓库与数据挖掘
6. 参考文献

数据库的两层映像功能与数据独立性

- 外模式/模式映像

- 对应于一个模式可以有多个外模式。对于每一个外模式，数据库管理系统都有一个模式/外模式映像，它定义了该外模式与模式之间的对应关系。
- 在各自的外模式描述中定义外模式/模式映像。
- 保证了数据与应用程序的逻辑独立性，简称为数据的逻辑独立性

- 模式/内模式映像

- 数据库中只有一个模式，也只有一个内模式，模式/内模式映像是唯一的，它定义了数据全局逻辑结构与存储结构之间的对应关系。
- 在模式描述中定义模式/内模式映像。
- 保证了数据与应用程序的物理独立性，简称为数据的物理独立性。

- 在数据库的三级模式结构中，模式即全局逻辑结构是数据库的核心和关键，它独立于数据库的其他层次。因此，设计数据库模式结构时，应首先确定数据库的逻辑模式。
- 数据库三级模式结构是数据库管理系统 (DBMS) 的体系结构，提供外模式、模式和内模式，通过从不同抽象级别观察数据库中的数据，实现对用户屏蔽 DBMS 的复杂性、简化用户与系统的交互的目的。

数据库三级模式与三层模型的联系与区别

- 数据库三层数据模型是**数据库设计的工具和方法**，提供概念模型、逻辑模型和物理模型，通过逐层设计一应用系统的数据库，实现**从现实世界到信息世界、信息世界到逻辑机器世界、逻辑机器世界到物理机器世界的逐步转换**(对应模式与内模式要求)。
- 数据库的三级模式与三层模型之间的区别在于：作用和目的不一样。

三级模式是 DBMS 的体系结构，目的是：

1. 隐藏数据的存储和维护的细节，为**用户提供数据在不同层次上的视图**，方便不同的使用者可以从不同的角度去观察和利用数据库中的数据。
2. 支持**数据独立性的实现**。
3. 提供**全局逻辑视图（模式）**：支持整体结构化，从而实现数据共享度高、冗余度低、易扩充。
4. 部分支持**安全性的实现**。

三层模型是数据库设计的工具和方法（要满足 DBMS 体系结构的要求），目的是：

1. 较真实地**模拟现实世界、容易被人理解、便于计算机实现**。一个数据模型不可能同时满足这些要求！
2. 提供**全局逻辑模型**：支持整体结构化，从而实现数据共享度高、冗余度低。
3. 同时满足**DBMS 三级模式结构(模式与内模式)**要求。

数据库三级模式与三层模型的联系与区别

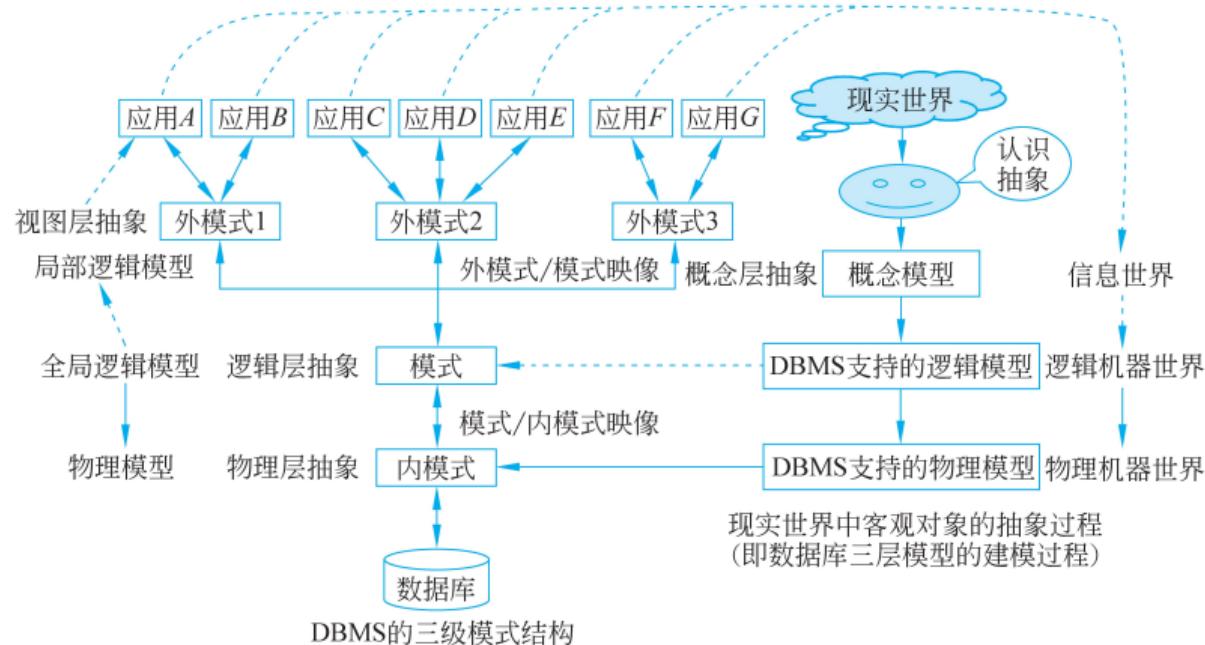


Figure 11: 数据库三级模式与三层模型之间的联系

1. 课程导论
2. 认识数据库
 - 2.1. 数据与数据管理
 - 2.2. 数据库技术的产生与发展
3. 数据模型
 - 3.1. 数据模型分层
 - 3.2. 组成要素
4. 数据库系统的结构与组成
 - 4.1. 数据库的三级模式
 - 4.2. 数据库的两层映像
 - 4.3. 数据库系统组成
5. 数据库领域的新技术
 - 5.1. 云数据库与分布式数据库
 - 5.2. 大数据与主动数据库
 - 5.3. 数据仓库与数据挖掘
6. 参考文献

数据库系统组成

- **数据库系统**(database system, DBS), 是指在计算机系统中引入数据库后的系统, 一般由**数据库**、**数据库管理系统(及其应用开发工具)**、**应用系统**、**数据库管理员**和**最终用户**构成
- **数据库管理员**(database administrator, DBA), 是指**数据库的建立、使用和维护等的工作人员**。

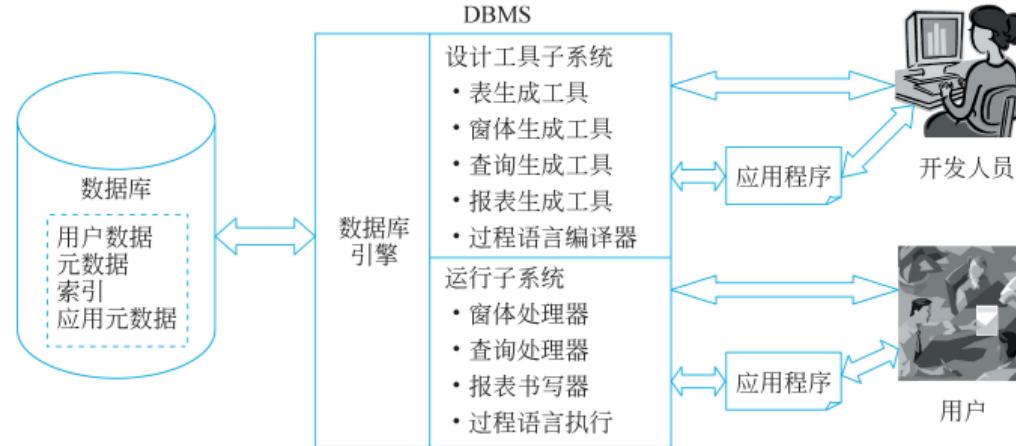


Figure 12: 数据库系统的组成

数据库管理系统——DBMS

- 数据库管理系统(DBMS) 是一组软件，负责数据库的访问、管理和控制。
- DBMS 的功能
 - 数据定义：DBMS 提供数据定义语言(DDL)
 - 数据组织、存储和管理：DBMS 要分类组织、存储和管理各种数据，包括数据字典、用户数据、数据的存取路径等
 - 数据操纵：DBMS 还提供数据操纵语言(DML)
 - 数据库的事务管理和运行管理：
 - ▶ 数据库在建立、运行和维护时，由 DBMS 统一管理、控制，以保证数据的安全性、完整性（一致性）。
 - ▶ 以及多用户对数据并发操作时的数据库正确性（称为并发控制）和系统发生故障后的数据库正确性（称为恢复与备份）
 - 数据库的建立和维护
 - 其他功能

DBMS 的组成

- **查询处理器**: 对用户请求的 SQL 操作进行查询优化，从而找到一个最优的执行策略，然后向存储管理器发出命令，使其执行。
- **存储管理器**: 根据执行策略，从数据库中获取相应的数据，或更新数据库中相应的数据。
- **事务管理器**: 负责保证系统的完整性，保证多个同时运行的事务不发生冲突操作，以及保证当系统发生故障时数据不会丢失。

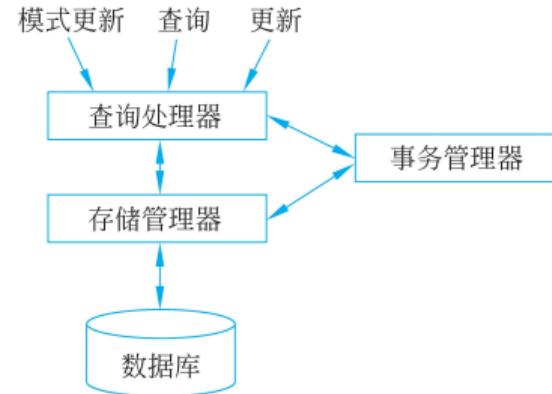


Figure 13: DBMS 的主要组成部分

数据库系统的相关人员

开发、管理和使用数据库系统的人员：

- 数据库管理员
- 系统分析员
- 数据库设计人员
- 应用程序员
- 最终用户

不同的人员涉及不同的数据抽象级别，具有不同的数据视图，如图14所示。

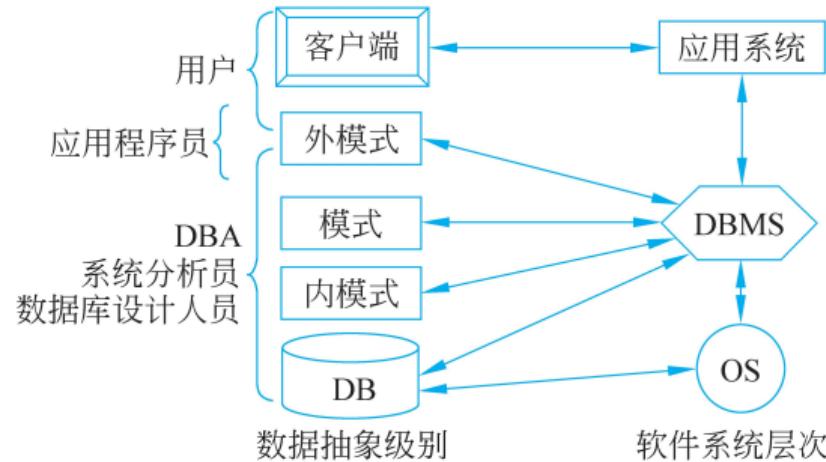


Figure 14: 数据库系统中相关人员的数据视图

数据库系统的相关人员

数据库管理员的主要职责

- 决定数据库中的信息内容和结构
- 决定数据库的存储结构和存取策略
- 定义数据的安全性要求和完整性约束条件
- 监控数据库的使用和运行
- 数据库的改进和重组重构

1. 课程导论
2. 认识数据库
 - 2.1. 数据与数据管理
 - 2.2. 数据库技术的产生与发展
3. 数据模型
 - 3.1. 数据模型分层
 - 3.2. 组成要素
4. 数据库系统的结构与组成
 - 4.1. 数据库的三级模式
 - 4.2. 数据库的两层映像
 - 4.3. 数据库系统组成
5. 数据库领域的新技术
 - 5.1. 云数据库与分布式数据库
 - 5.2. 大数据与主动数据库
 - 5.3. 数据仓库与数据挖掘
6. 参考文献

随着计算机软硬件技术的进步，特别是大数据、云计算的出现，数据库技术获得了快速发展，新技术和新系统层出不穷。

Definition (云数据库)

云数据库(Cloud DataBase) 是指**被优化或部署到一个虚拟计算环境中的数据库**。具有按需付费、按需扩展、高可用性及存储整合等优势。

将一个现有的数据库优化到云环境，有以下好处：

- 可以使用户按照存储容量和带宽的需求付费：因为云数据库基本采用多租户的形式，能够以共享资源的形式节省用户的开销。
- 云的可移植性：可移植性是指可以将数据库从一个地方移到另一个地方。用户不必控制运行原始数据库，只需要一个有效的连接字符串就可以使用云数据库。
- 可实现按需扩展：理论上云数据库具有无限的可扩展性，具有良好的弹性。
- 高可用性：不存在单点失效的问题，一个节点失效，剩余的节点会接管任务。

Definition (分布式数据库)

分布式数据库(Distributed DataBase) 是指数据分别存储在计算机网络中的各台计算机上的数据库。

- 分布式数据库系统通常使用较小的计算机系统，每台计算机可单独放在一个地方，其中都可能有 DBMS 的一份完整拷贝副本，或者部分拷贝副本，并具有自己局部的数据库，位于不同地点的许多计算机通过网络互相连接，共同组成一个完整的、全局的、逻辑上集中且物理上分布的大型数据库。
- 分布式数据库相对传统集中式数据库具有更高的数据访问速度、更强的可扩展性和更高的并发访问量等优点。

1. 课程导论
2. 认识数据库
 - 2.1. 数据与数据管理
 - 2.2. 数据库技术的产生与发展
3. 数据模型
 - 3.1. 数据模型分层
 - 3.2. 组成要素
4. 数据库系统的结构与组成
 - 4.1. 数据库的三级模式
 - 4.2. 数据库的两层映像
 - 4.3. 数据库系统组成
5. 数据库领域的新技术
 - 5.1. 云数据库与分布式数据库
 - 5.2. 大数据与主动数据库
 - 5.3. 数据仓库与数据挖掘
6. 参考文献

Definition (大数据)

大数据 (Big Data) 也称海量数据或巨量数据，是指数据量大到无法利用传统数据处理技术在合理的时间内获取、存储、管理和分析的数据集合。“大数据”一词除用来描述信息时代产生的海量数据外，也用来命名与之相关的技术、创新与应用。

大数据的特征 (4V)

- 海量的数据规模 (Volume)
- 快速的数据流转 (Velocity)
- 多样的数据类型 (Variety)
- 价值密度低 (Value)

Definition (大数据技术)

大数据技术是指用非传统的方式对大量结构化和非结构化数据进行处理，以挖掘出数据中蕴含的价值的技术。

大数据的处理流程

根据大数据的处理流程，可以将其关键技术分为

- **大数据采集**

- 对于网络上各种来源的数据，包括社交网络数据、电子商务交易数据、网上银行交易数据、搜索引擎点击数据、物联网传感器数据等，在被采集前都是零散的，没有任何意义。大数据采集就是将这些数据写入数据仓库，整合在一起，以便对数据进行综合分析。
- 大数据采集包括网络日志采集、网络文件采集（提取网页中的图片、文本等）、关系型数据库的接入等，常用的工具有 Flume, Kafka, Sqoop 等。

- **大数据预处理**

- 由于大数据的来源和种类繁多，这些数据有残缺的、有虚假的、有过时的，因此，想要获得高质量的数据分析结果，必须在数据准备阶段提高数据的质量，即对大数据进行预处理。
- 大数据预处理是指将杂乱无章的数据转化为相对单一且便于处理的结构（数据抽取），或去除没有价值甚至会对分析造成干扰的数据（数据清洗），从而为后期的数据分析奠定基础。

大数据的处理流程

根据大数据的处理流程，可以将其关键技术分为

- **大数据存储与管理**

- 大数据存储是指用存储器把采集到的数据存储起来，并建立相应的数据库，以便对数据进行管理和调用。
- 目前，主要采用 HDFS 分布式文件系统（Hadoop Distributed File System）和非关系型分布式数据库（NoSQL）来存储和管理大数据。常用的 NoSQL 数据库包括 HBase, Redis, Cassandra, MongoDB, Neo4j 等。

- **大数据分析与挖掘**

- 大数据分析与挖掘是指通过各种算法从大量的数据中找出潜在的有用信息，并研究数据的内在规律和相互间的关系。
- 常用的大数据分析与挖掘技术包括 Spark, MapReduce, Hive, Pig, Flink, Impala, Kylin, Tez, Akka, Storm, S4, Mahout, MLlib 等。

- **大数据可视化展现**

- 大数据可视化展现是指利用可视化手段对数据进行分析，并将分析结果用图表或文字等形式展现出来，从而使读者对数据的分布、发展趋势、相关性和统计信息等一目了然。
- 目前，常用的大数据可视化工具 Echarts 和 Tableau 等。

主动数据库

- 传统数据库一般只根据应用程序的要求对数据库进行基本操作，仅作为一种被动的数据仓库存在。
- **主动数据库**(Active DataBase) 是指**在没有用户干预的情况下，能够主动地对系统内部或外部所发生的事件做出反应的数据库**，是**数据库技术与人工智能技术相结合的产物**。

特点

- 让数据库系统具有主动服务的功能，并以一种统一的机制来实现各种主动服务需求。
- 系统提供一个自动监视模块，不时地检查着规划中包含的各种事件是否发生，一旦发现某事件发生，就主动触发执行某个动作。
- 数据库管理系统就可以自动执行由用户预先设定的动作，诸如完整性约束、存取控制、例外处理、监督和警告、状态开关自动切换及检索策略的切换，乃至复杂的演绎推理和实时处理等功能以一种统一的机制实施。
- 虽然主动数据库还有待发展，但其已经在计算机集成制造、网络管理和办公自动化等领域有了广泛的应用。

1. 课程导论
2. 认识数据库
 - 2.1. 数据与数据管理
 - 2.2. 数据库技术的产生与发展
3. 数据模型
 - 3.1. 数据模型分层
 - 3.2. 组成要素
4. 数据库系统的结构与组成
 - 4.1. 数据库的三级模式
 - 4.2. 数据库的两层映像
 - 4.3. 数据库系统组成
5. 数据库领域的新技术
 - 5.1. 云数据库与分布式数据库
 - 5.2. 大数据与主动数据库
 - 5.3. 数据仓库与数据挖掘
6. 参考文献

Definition (数据仓库)

数据仓库(Data Warehouse) 是面向主题的、集成的、相对稳定的、反映历史变化的数据集合，通常用于辅助决策支持。

- **面向主题**: 数据仓库中的数据按照一定的主题域进行组织，它们划分为各自独立的领域，每个领域都有自己的逻辑内涵且互补不交叉。
- **集成性**: 数据仓库中的数据是对原有分散的数据库数据做抽取、清理后经过加工汇总得到的，源数据经统一与综合之后才能进入数据仓库。
- **相对稳定**: 数据一旦加载到数据仓库，一般情况下不会再修改或删除，而是作为数据档案长期保存。
- **反映历史变化**: 数据仓库系统通常记录一个单位从过去某一时间点到目前时间点所有时期的信息，可通过这些信息对这一单位的发展历程和未来趋势做出分析和预测。

Definition (数据挖掘)

数据挖掘(Data Mining)是从大量数据中获取有效的、新颖的、潜在有用的、最终可理解的模式的非平凡过程，也称为知识发现。简单来讲，数据挖掘就是从大量数据中提取或“挖掘”知识。

- **直接挖掘**: 利用可用的数据建立一个模型，这个模型是对一个特定属性的描述，如分类、估值和预测等
- **间接挖掘**: 在所有的属性中寻找某种关系，如关联规则和聚类等。

数据挖掘是交叉性学科。

- 是数据库技术、机器学习、统计学、人工智能、可视化分析和模式识别等多门学科的融合。
- 把现代企业中的原始数据转换为人工智能的来源，对数据进行操纵，提供可靠的、可以用来决策的信息。

本章小结

1. 从数据这个最基本的概念入手，引出了数据管理的相关概念，论述了数据管理技术的 3 个发展阶段，着重说明了数据库管理系统和文件系统在数据管理上的本质区别。
2. 数据模型是一个描述数据语义、数据与数据之间联系（数据结构），数据操作，以及一致性（完整性）约束的概念和工具的集合。根据数据抽象的不同级别，可以将数据模型划分为 3 类：概念模型、逻辑模型和物理模型。
3. 数据库管理系统的数据抽象包括物理层抽象、逻辑层抽象和视图层抽象。对于数据抽象的 3 个级别，数据库管理系统一般也提供观察数据库的 3 个不同角度，以屏蔽一些数据组织的细节，方便不同的用户使用数据库，这就是数据库的三级模式结构：内模式、模式和外模式。
4. 数据库管理系统在数据库的三级模式之间提供了两层映像：外模式/模式映像、模式/内模式映像。三级模式和两层映像可以较好地实现数据独立性（数据的逻辑独立性和数据的物理独立性）的要求。
5. 数据库系统的组成。

本章难点问题

讨论

- DBMS 的特点。难点是对数据整体结构化、数据独立性、数据统一管理和控制的深入理解。例如，为什么需要数据的整体结构化？如何获取和如何保障数据的整体结构化？为什么需要数据独立性？为什么需要对数据进行统一管理和控制？
- 数据模型。难点是对数据模型的作用的深入理解。例如，为什么需要数据模型？关系数据模型的优点是什么？

思考与创新

- 讨论数据库管理系统和文件系统在数据管理上的区别。
- 讨论数据库三级模式、二层映像与数据独立性的关系。
- “数据库三级模式”的概念与“数据库三层模型”概念之间的异同？

参考文献 |

-  CODD, B. E. F.
A relational model of data for large shared data banks.
M.D. computing : computers in medical practice 15-3 (1970), 162–166.

Thank you for your attention!
Q&A