

Chapter 1 Introduction

什么是数据库

数据库 (Database) 是一个组织和存储数据的集合。它是在计算机系统中使用的一种结构化方式，用于有效地管理和访问数据。数据库可以存储各种类型的数据，包括文本、数字、图像、音频等。它提供了一种机制来组织数据，并通过查询和操作来检索和修改数据。

- 一个非常大且集成的数据集合
- 建模真实世界 (实体+关系)

什么是DBMS

数据库管理系统 (Database Management System, DBMS) 是一种软件系统，用于管理和操作数据库。它提供了一系列功能和工具，用于创建、存储、查询和更新数据库中的数据。DBMS负责处理数据的物理存储、安全性、并发控制、备份和恢复等管理任务。

什么是关系型数据库

关系型数据库管理系统 (Relational DBMS) 是一种特定类型的DBMS，基于关系模型来组织和管理数据。关系模型使用表 (表格) 来表示数据，其中每个表由行和列组成。关系型DBMS使用结构化查询语言 (Structured Query Language, SQL) 来操作和查询数据库中的数据。

文件与数据库的比较

- 在大型数据集的应用程序中，必须将数据在主存储器和辅助存储器之间进行阶段 (例如缓冲区，基于页的访问，32位地址寻址等)。
- 需要针对不同查询编写特殊代码。
- 必须防止由于多个并发用户导致数据不一致，需要保护数据。
- 崩溃恢复。
- 安全性与访问控制。

数据库系统的目的 (原来的问题)

- 数据冗余和不一致：数据存储在各种文件格式中，导致不同文件中的信息重复
- 数据访问困难：每次执行新任务都需要编写新程序
- 数据隔离：多个文件和格式
- 完整性问题：完整性约束（例如账户余额大于零）被“埋没”在程序代码中，而不是明确提出
- 难以添加新约束或更改现有约束。

为什么使用数据库管理系统（DBMS）

- 数据独立性和高效的访问
- 减少应用程序的开发时间
- 数据完整性和安全性
- 统一的数据管理
- 并行访问、崩溃恢复

数据（Data）、数据模型（DataModel）和数据架构（DataSchema）

- **数据**是对现实世界事物的描述，是信息的存在形式。
- **数据模型**是用于描述数据的概念和定义的集合。Cursor
- **架构**是对特定数据集的描述，使用给定的数据模型
- 目前最广泛使用的是关系模型的数据（relational model of data）：
 - 主要概念：关系relation，基本上是一个带有行和列的表格。
 - 每个关系都有一个架构，它描述列或字段。

数据是原始的符号表示，数据模型是对数据的抽象和组织方式，数据架构是数据模型在特定系统中的具体实现。

数据模型提供了对数据的概念化描述，而数据架构则将这些概念转化为实际的存储结构和操作方式。

ANSI-SPARC架构

也被称为三层架构，为组织和管理数据库系统中的数据提供了一个框架。它定义了三个抽象层级：外部或视图层views，概念或逻辑层conceptual (logical) schema，以及内部或物理层physical schema

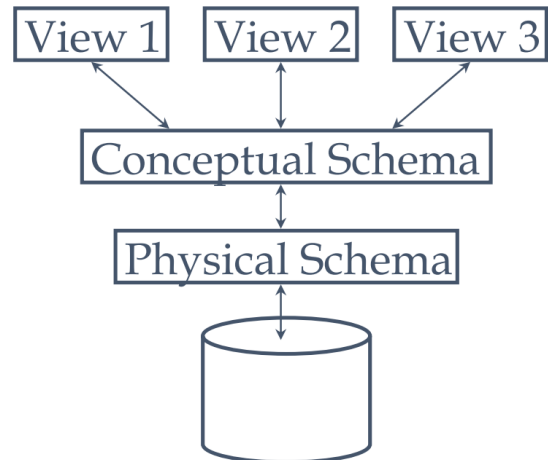
- **外部层级**：创建多个视图来描述不同用户或用户组如何看待数据。每个视图代表数据的特定子集，

并可以根据不同用户的需求进行定制。视图提供了对数据的简化和定制的视角，隐藏了不必要的细节，只呈现相关信息。

- **概念层级**：代表整个数据库的逻辑结构。它作为外部视图和物理存储之间的中间层。概念模式定义了管理数据的关系、约束和完整性规则。它提供了一个高层次的、抽象的数据模型表示，独立于任何具体的实现或技术。
- **内部层级**：描述了数据在底层硬件和存储设备上的物理存储和访问方式。它包括文件组织、索引技术和数据存储优化等细节。物理模式将概念模式中定义的逻辑结构映射到数据库系统中实际使用的文件、数据块和索引

Example: University Database

- Conceptual schema:
 - *Students*(sid: string, name: string, login: string, age: integer, gpa: real)
 - *Courses*(cid: string, cname: string, credits: integer)
 - *Enrolled*(sid: string, cid: string, grade: integer)
- Physical schema:
 - Relations stored as unordered files.
 - Index on first column of Students.
- External Schema (View):
 - *Course_info*(cid: string, enrollment: integer)



通过这种三层架构，ANSI-SPARC提供了数据独立性。应用程序可以通过外部层访问数据库，而无需了解底层的物理存储细节。如果数据库的物理存储结构发生变化，只需要调整内部层，而不会影响外部层和概念层。这使得数据库的维护和演化更加灵活，同时保护了应用程序不受数据库结构变化的影响。

数据定义语言 (DDL)

- 用于定义数据库模式的规定符号
- DDL编译器生成一组存储在数据字典中的表模板
- 数据字典包含元数据（即关于数据的元信息）
 - 数据库模式
 - 完整性约束：主键
 - 授权：谁可以访问什么数据（who can access what）

数据操纵语言 (DML)

- 用于访问和更新由适当的数据模型组织的数据的语言：DML也被称为查询语言query language
- DML基本上可以分为两种类型

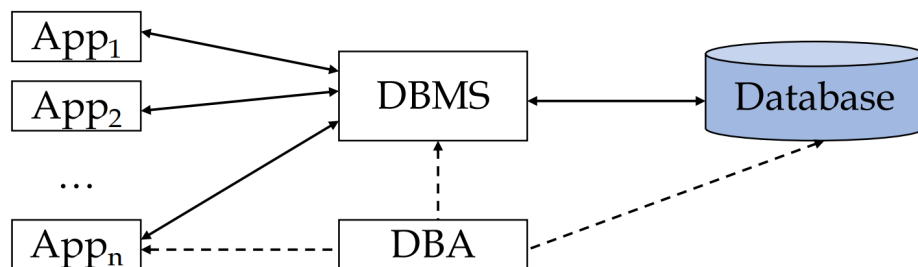
- 过程式DML (procedural) : 需要用户指定需要哪些数据以及如何获取这些数据
- 声明式DML (Declarative) : 用户只需指定需要哪些数据, 无需指定如何获取这些数据
- 声明式DML通常比过程式DML更容易学习和使用、声明式DML也被称为非过程式DML
- DML中涉及信息检索的部分被称为查询语言

SQL查询语言

- SQL查询语言是非过程性的。查询将N个表作为输入, 并且总是返回一个单一的表。
- SQL不是图灵机等价语言Turing machine equivalent language
 - 为了能够计算复杂的函数, SQL通常嵌入在某种更高层次的语言中
- 应用程序通常通过以下方式访问数据库:
 - 语言扩展允许嵌入式SQL
 - 应用程序程序接口 (例如ODBC/JDBC) , 允许将SQL查询发送到数据库

数据库系统Database System

- 应用 + DBMS + 数据库 + DBA (数据库管理员)
- DBMS 是数据库系统的核心
 - 提供高级用户接口、查询处理和优化、目录管理、并行控制和恢复、完整性约束检查、访问控制



数据库系统的生命周期

- 数据库系统规划
- 数据库设计
- 数据库建立和加载
- 数据库运行、管理和维护

- 数据库扩展和重构

