

# 数据库管理系统 原理与实现

杜小勇 陈红 卢卫 主编  
中国人民大学  
信息学院

# 第一章 概述

- 1.1 数据库系统概述
- 1.2 数据库技术的发展
- 1.3 数据库管理系统组成
- 1.4 我国数据库的发展历程
- 1.5 小结

# 第一章 概述

**章成源**

湖南大学-信息科学与工程学院-计算机科学系

办公室：院楼403

Email: cyzhangcse@hnu.edu.cn

# 数据库系统概述

## 内容

介绍数据库的4个基本概念，即数据、数据库、数据库管理系统和数据库系统；讲述使用数据库系统的原因。

## 目标

区分数据库的4个基本概念；

了解使用数据库系统的原因。

# 数据库系统概述

- 1.1.1 数据库系统的基本概念
- 1.1.2 为什么要使用数据库系统

## 1.1.1 数据库的四个基本概念

- 数据 (Data)
- 数据库 (Database)
- 数据库管理系统 (DBMS)
- 数据库系统 (DBS)

# 数据

- **数据（Data）** 是数据库中存储的基本对象
- **数据的定义**
  - 描述事物的符号记录
- **数据的种类**
  - 数字、文字、图形、图像、音频、视频、学生的档案记录等

# 数据举例

- 数据的含义称为数据的语义，数据与其语义是不可分的

例如：93是一个数据

语义1：学生某门课的成绩

语义2：某同学的身高

语义3：计算机系2019级学生人数

语义4：请同学给出……

## 1.1.1 数据库的四个基本概念

- 数据 (Data)
- 数据库 (Database)
- 数据库管理系统 (DBMS)
- 数据库系统 (DBS)

# 数据库

- **数据库的定义**

数据库（Database，简称DB）是长期储存在计算机内、有组织的、可共享的大量数据的集合。

- **数据库的基本特征**

- 数据按一定的数据模型组织、描述和储存
- 可为各种用户共享
- 冗余度较小
- 数据独立性较高
- 易扩展

## 1.1.1 数据库的四个基本概念

- 数据 (Data)
- 数据库 (Database)
- **数据库管理系统 (DBMS)**
- 数据库系统 (DBS)

# 数据库管理系统

- 什么是数据库管理系统?
  - 位于用户与操作系统之间的一层数据管理软件
  - 基础软件，是一个大型复杂的软件系统
- 数据库管理系统的用途
  - 科学地组织和存储数据、高效地获取和维护数据

## 1.1.1 数据库的四个基本概念

- 数据 (Data)
- 数据库 (Database)
- 数据库管理系统 (DBMS)
- 数据库系统 (DBS)

# 数据库系统

- 数据库系统是指在计算机系统中引入数据库后的系统构成
- 由四部分组成：
  - 数据库
  - 数据库管理系统（及外围的应用开发工具）
  - 应用系统
  - 数据库管理员（DBA）
- 在不引起混淆的情况下，人们常常把数据库系统简称为数据库。

# 数据库系统概述

- 1.1.1 数据库系统的基本概念
- 1.1.2 为什么要使用数据库系统

## 1.1.2 为什么要使用数据库系统？

- 用数据库系统来管理数据具有如下特点：
  - 数据结构化：**数据用**统一的模型**来描述
  - 数据高共享：**数据被多用户、多应用、不同接口、不同编程语言**共享使用**
  - 数据独立性高：**数据独立性是指应用程序与数据相分离，包括数据的**物理独立性**和数据的**逻辑独立性**
  - 数据由数据库管理系统统一管理和控制：**数据库管理系统在数据库建立、运维时对数据库进行统一控制

# 课堂测试

- 数据库核心特征是什么？

长期储存、有组织、可共享、大量。

- 数据库系统主要由什么构成？

数据库

数据库管理系统（及其应用开发工具）

应用程序

数据库管理员



# 第一章 概述

- 1.1 数据库系统概述
- 1.2 数据库技术的发展
- 1.3 数据库管理系统组成
- 1.4 我国数据库的发展历程
- 1.5 小结

## 1.2 数据库技术的发展

数据库技术是计算机领域中发展最快的领域之一，也是应用最广泛的技术之一，其中**数据模型、应用需求和计算平台**的发展是推动数据库发展的三个重要因素。下面分别以这三个维度来介绍数据库技术的发展。

# 第一章 概述

- 1.2.1 数据模型的发展
- 1.2.2 应用需求推动数据库技术发展
- 1.2.3 计算平台的进步推动数据库技术发展

## 1.2.1 数据模型的发展

**数据模型---对现实世界中实体和实体之间的联系等的抽象和表示**

- 数据库系统是基于某种数据模型实现的
- 数据模型的发展历史一定程度上也是数据库系统的发展史

# 第一章 概述

- 1. 层次模型和网状模型
- 2. 关系模型
- 3. 星型模型
- 4. 面向对象模型
- 5. NoSQL数据库
- 6. 多模数据库

# 网状模型

- 网状模型

满足下面两个条件的基本层次联系的集合：

1. 允许一个以上的结点无双亲；
2. 一个结点可以有多于一个的双亲。

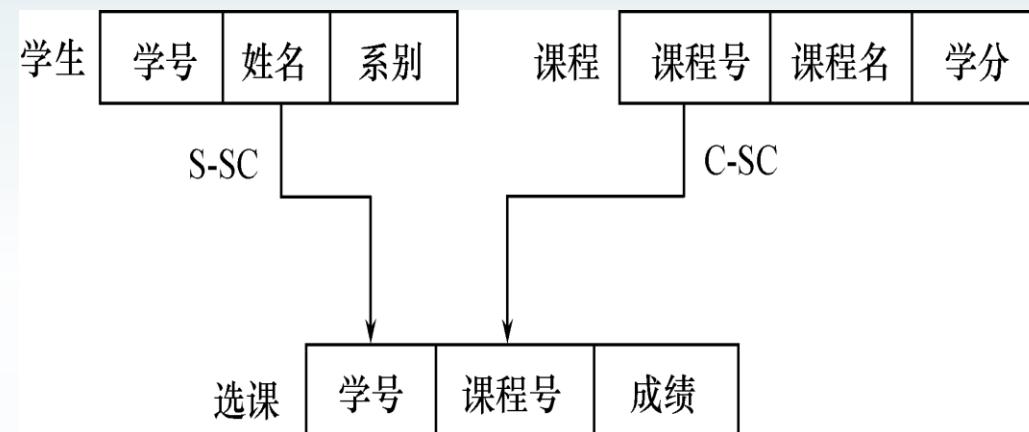


Fig. 学生/选课/课程的网状数据模型

# 网状模型

- 采用网状模型的数据库称为网状数据库
  - 网状数据库是最早出现的数据库之一
- 典型代表是**IDS (Integrated Data Storage)**：
  - 世界上第一个数据库系统
  - 将数据从应用程序中独立出来并进行集中管理，多个应用可同时共享访问数据库

# 层次模型的数据结构

- **层次模型**

满足下面两个条件的基本层次联系的集合：

1. 有且只有根结点没有双亲结点，
2. 根以外的其它结点有且只有一个双亲结点

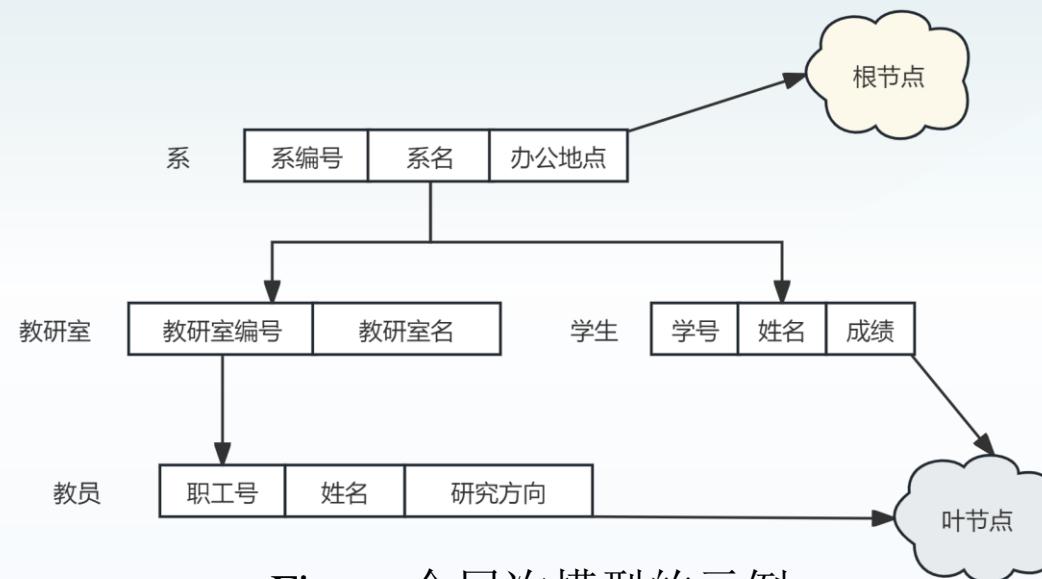


Fig. 一个层次模型的示例

# 层次模型

- **层次模型**

- 层次模型是数据库系统中**最早出现的数据模型之一**
- 采用层次模型的数据库称为**层次数据库系统**，典型代表是IBM公司的**IMS**（Information Management System）数据库管理系统
- 层次模型用**树形结构**来表示各类实体以及实体间的联系

# 层次/网状模型

## 层次/网状模型

- 在数据库发展**早期**比较流行，但是之后被淘汰，因为：
  - 数据查询和数据操纵方面都采用的是一次一个记录的导航式过程化语言
  - 需要用户明确数据的存储结构，预设数据访问路径，编程繁琐，应用程序的可移植性较差

# 第一章 概述

- 1. 层次模型和网状模型
- 2. **关系模型**
- 3. 星型模型
- 4. 面向对象模型
- 5. NoSQL数据库
- 6. 多模数据库

# 关系模型的数据结构

- 在用户观点下，关系模型中数据的逻辑结构是一张**二维表**，它由行和列组成。

学 号	姓 名	年 龄	性 别	系 名	年 级
2013004	王小明	19	女	社会学	2013
2013006	黄大鹏	20	男	商品学	2013
2013008	张文斌	18	女	法律	2013
...	...	...	...	...	...

Fig. 学生登记表

# 关系模型的数据结构

- **关系 (Relation)**
  - 一个关系对应通常说的一张表
- **元组 (Tuple)**
  - 表中的一行即为一个元组
- **属性 (Attribute)**
  - 表中的一列即为一个属性，给每一个属性起一个名称即属性名
- **主码 (Key)**
  - 也称码键。表中的某个属性组，它可以唯一确定一个元组

# 关系数据库

- **关系数据库系统**采用**关系模型**作为数据的组织方式
- **关系模型**
  - 首次提出--1970年美国IBM公司San Jose研究室的研究员E.F.Codd提出关系模型
  - 严密的数学基础--基于谓词逻辑和集合论，并提供了高级别的数据抽象层次。



INGRES

# 关系数据库

- 关系数据库的基本理论已经成熟，但是当
  - 数据规模越来越大
  - 结构越来越复杂
  - 越来越多的用户需要共享数据库时
- 如何保障数据库的高性能，高安全，高可靠成为一个具有挑战性的问题

# 第一章 概述

- 1. 层次模型和网状模型
- 2. 关系模型
- 3. 星型模型
- 4. 面向对象模型
- 5. NoSQL数据库
- 6. 多模数据库

# 数据仓库

数据仓库是一个用以更好地支持企业（或组织）**决策分析处理**的、面向主题的、集成的、不可更新的、随时间不断变化的数据集合。源数据经过**抽取、转换、装载（简称ETL）**进入数据仓库

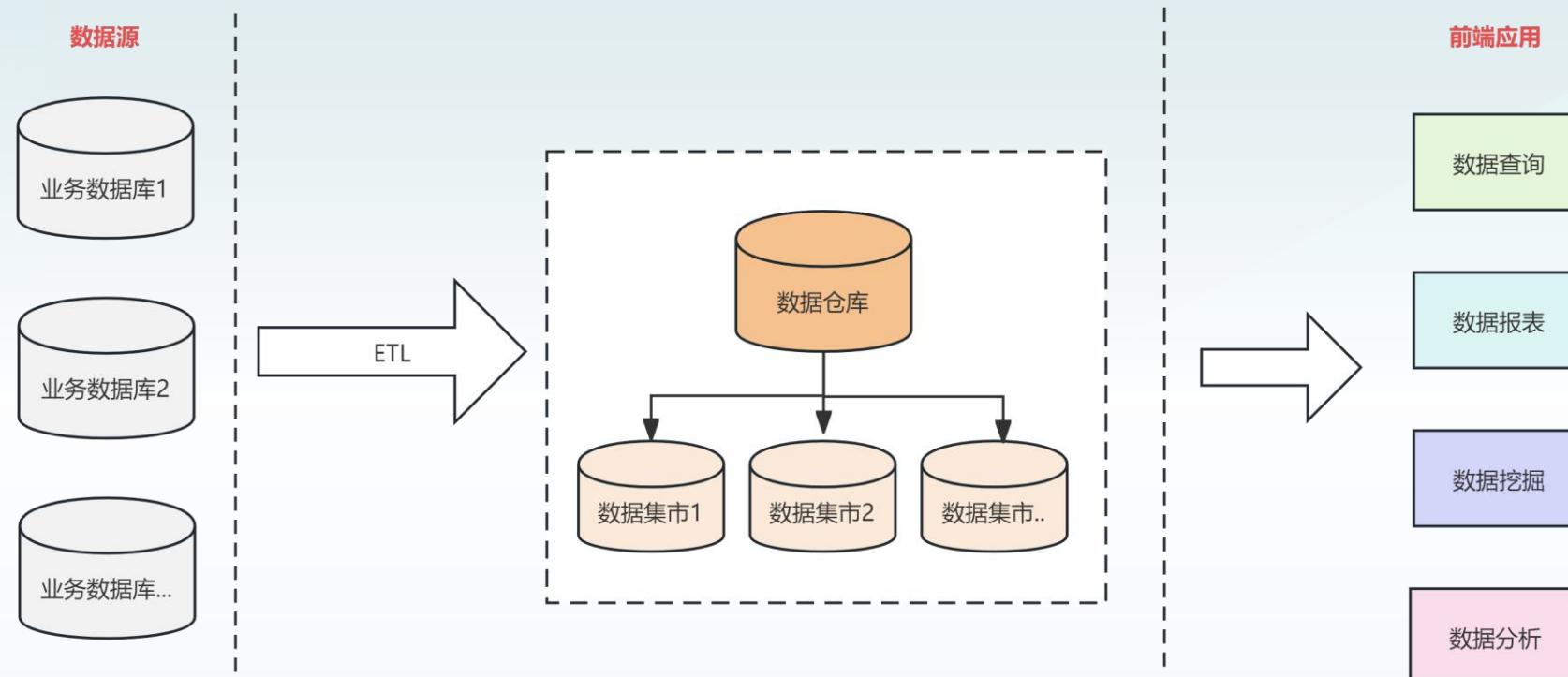


Fig. 数据仓库

# 数据仓库的数据模型

- 数据仓库最基本的数据模型是**立方体模型**，也称**星型模型、多维数据模型**
- **星型模型**
  - 一个事实表和一组维表组成。
  - **事实表记录单个事件的信息**：例如商品的销售信息。事实表的属性可以分为维属性和度量属性
    - 维属性是对度量属性进行分组和查看的维度，例如，对于销售信息
    - 度量属性存储能够执行聚集操作的量化信息，例如商品的销售额
  - **维度表**可以是顾客、产品、供应商、时间等。

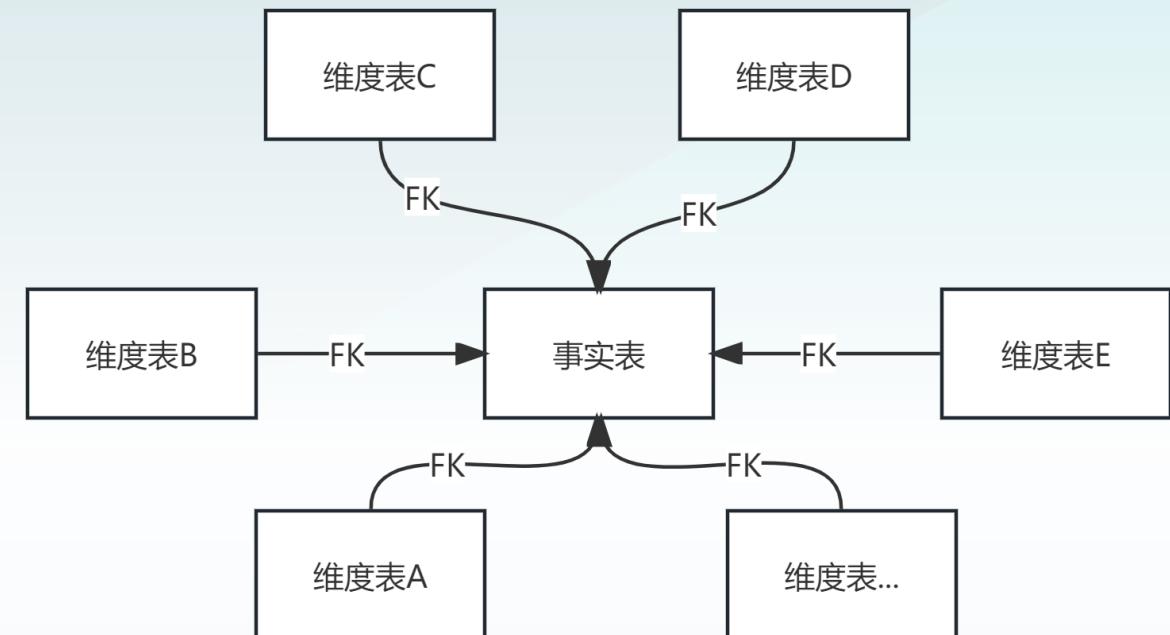


Fig. 星型模型

# 第一章 概述

- 1. 层次模型和网状模型
- 2. 关系模型
- 3. 星型模型
- 4. 面向对象模型
- 5. NoSQL数据库
- 6. 多模数据库

# 面向对象模型

## 面向对象的数据模型：

- 用面向对象的观点来描述现实世界实体的逻辑组织、对象间的关系

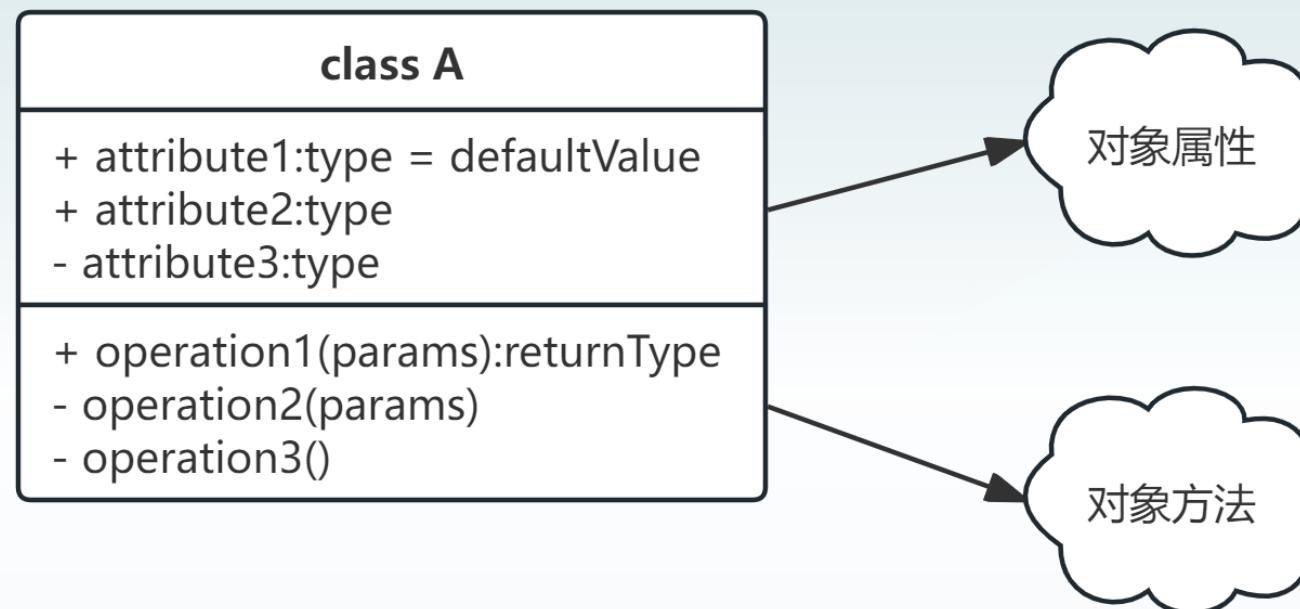


fig. 面向对象模型

# 面向对象数据库

面向对象数据的研究始于20世纪80年代

- 有许多面向对象数据库产品，例如Object Store、O2、ONTOS等
- 典型代表：对象关系数据库ORDB
  - 结合了关系数据库与面向对象数据库技术结合形成的关系数据库系统
  - 保持了关系数据库的非过程化数据存取方式和数据独立性
  - 又支持面向对象模型和对象管理

# 第一章 概述

- 1. 层次模型和网状模型
- 2. 关系模型
- 3. 星型模型
- 4. 面向对象模型
- 5. NoSQL数据库
- 6. 多模数据库

# NoSQL数据库

随着互联网的发展，数据库需要处理的数据量越来越大，数据类型也越来越多样和异构，例如HTML、XML、JSON等半结构化数据到图形图像、音频、视频等非结构化数据等等

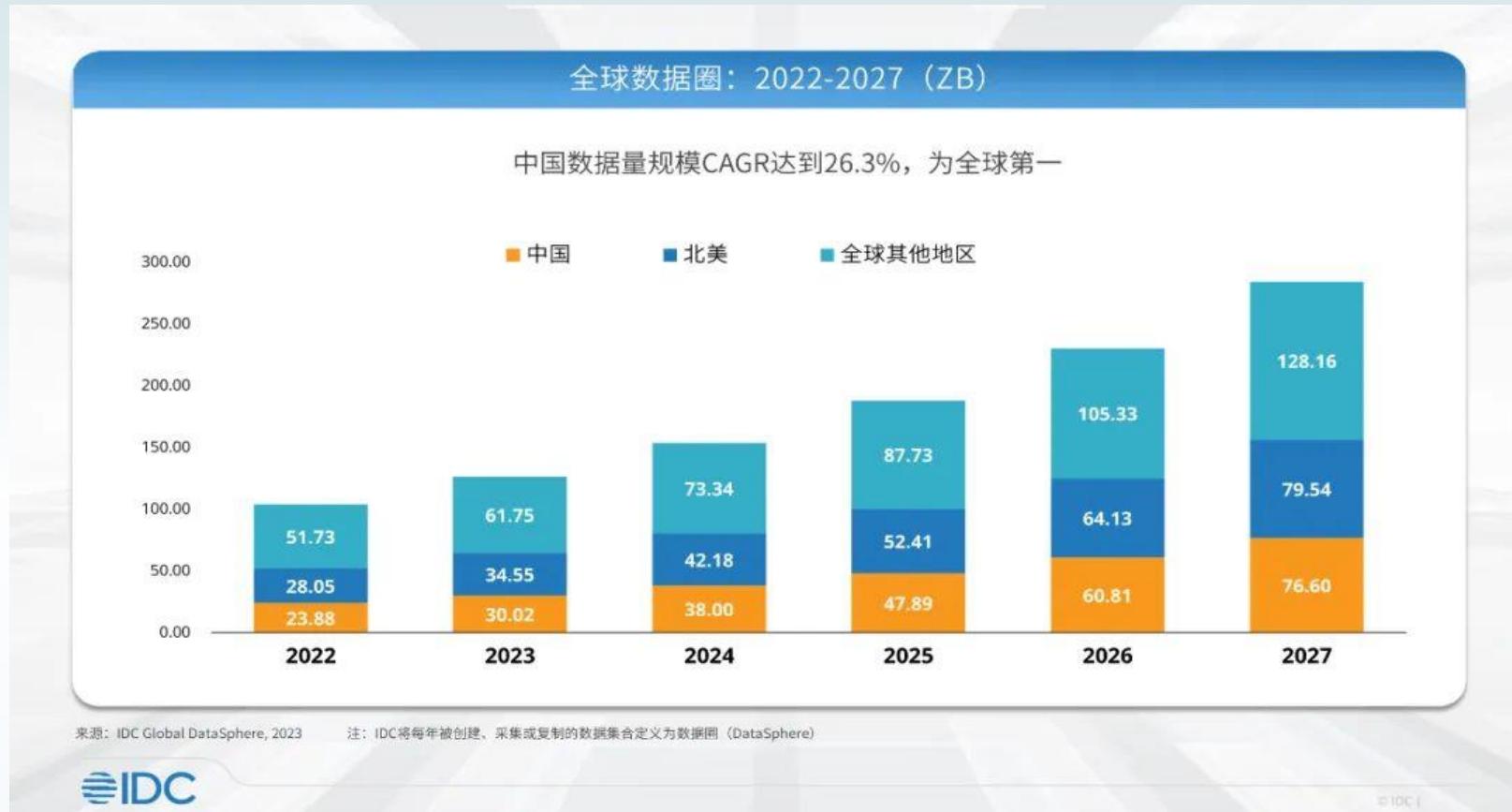


Fig. 全球数据量变化-IDC

# NoSQL数据库

- **NoSQL数据库**
  - 传统的关系数据库在系统的**伸缩性、容错性和可扩展性**等方面难以满足海量数据管理的需求，NoSQL数据库应运而生
- NoSQL有两种解释，第二种解释更容易被接受。
  - **Non-Relational**，即非关系数据库；
  - 另一种是**Not Only SQL**，即数据管理技术不仅仅是SQL，还需要一些扩展

# NoSQL数据库分类

NoSQL数据库支持的数据模型通常分为以下四类：

- **键值** (Key-Value) 模型
- **宽表** (Big Table/Column-Family) 模型
- **文档** (Document) 模型
- **图** (Graph) 模型

# 键值模型

## 键值模型

- 存储KV对 (Key, Value) 数据。每一个Key值唯一对应一个Value

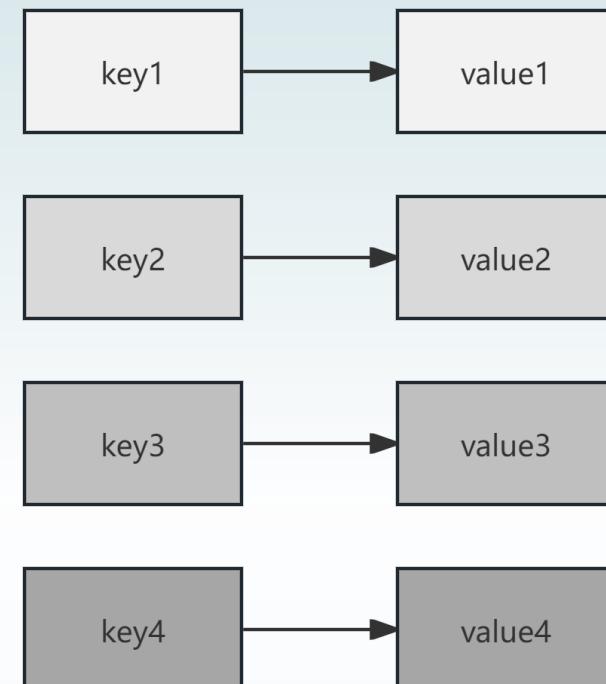


fig. key-value模型举例

# 键值数据库

## 键值数据库

- 在不涉及过多数据关系业务的需求中，使用键值模型可以有效地减少磁盘I/O的次数，比关系数据库具有更多的性能和扩展性
- 常见的键值数据库包括RocksDB、LevelDB、Redis等

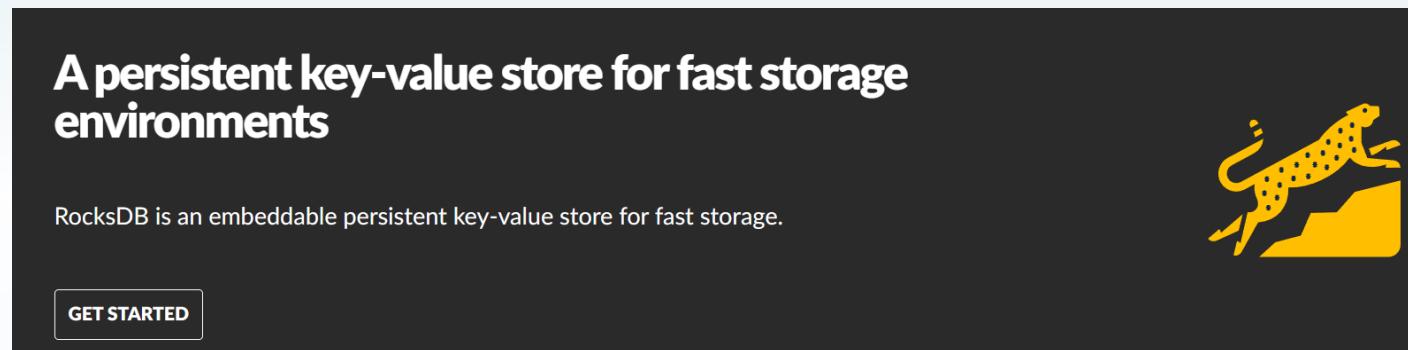


Fig. RocksDB



Fig. Redis

# 宽表/列簇

## 宽表/列簇

- 宽表，顾名思义是很宽的表，即有许多的列，列的集合称为列簇。
- 类似键值数据库，一般来说，每一行数据由一个Row key唯一标识



cf: user info			cf: address		
name	age	sex	prov	city	street
user1	22	1	HN	cs	street1
user2	23	1	HB	wh	street2
user3	24	0	SC	cd	street3
user4	25	0	BJ	hd	street4

fig.列簇举例

# 宽表数据库

宽表数据库，也称列簇式数据库。

- 按列存储：数据的各项存储在不同的列中，这些列的集合称为列簇。
- 多版本：每个数据项都包含一个时间戳属性，以便保存同一个数据项的多个版本。
- 优势：传统关系数据库有列不能分的限制，而宽表通过列簇的概念缓解了这一限制。

典型的列簇式数据库包括Hbase、Cloudera和Cassandra等



# 文档数据库

## 文档数据库

- 相比于键值数据库，它的Value值支持复杂的结构定义，例如XML或JSON等半结构化数据或者PDF或Office文档等二进制格式
- 常见的文档数据库有MongoDB、Apache CouchDB以及亚马逊的DocumentDB等

Document A
{ "id": "3", "fullName": { "first": "Adam", "last": "Stark" }, "isActive": true, "dob": "2015-04-19" }

fig. Json文档示例



# 图模型

## 图数据库

- 采用图模型G (V,E) 存储数据。其中V是点集，E是边集。
- 每个结点具有若干属性，每条边也可以具有多个属性

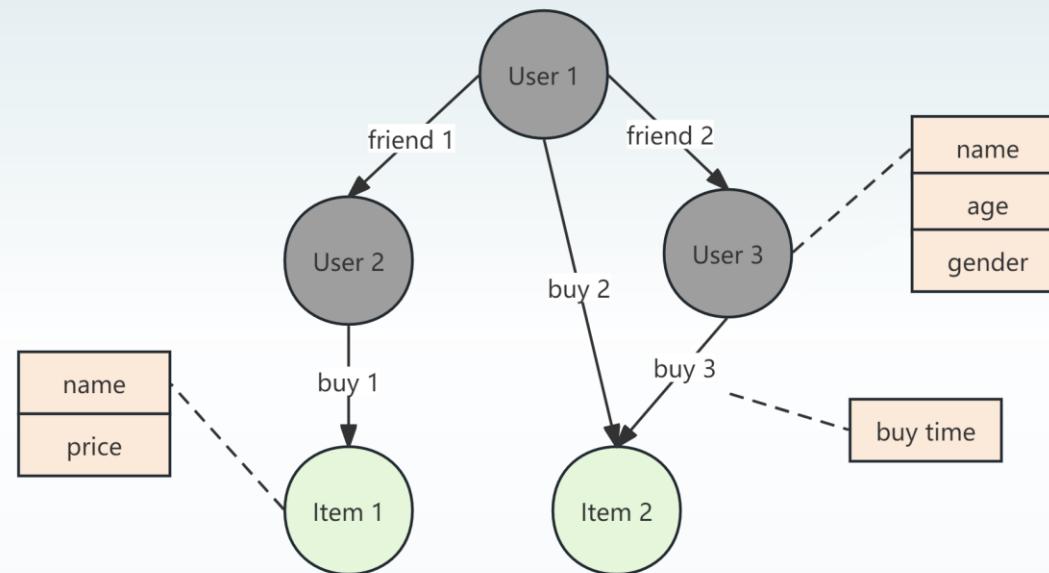


fig. 基于电商的图模型示例

# 图数据库

## 图数据库

- 图模型可以直观地表达和展示数据之间的联系，由于图数据库支撑了知识图谱、社交网络分析等新型应用，各种各样的图数据库越来越受到产业界的重视
- 常见的图数据库有Neo4j、Orient DB、ArangoDB等



# NoSQL数据库特点

通过上述的介绍，可以发现，NoSQL数据库有以下特点：

- **数据模型简单：**容易支持海量数据存储和高并发读写。
- **多副本技术：**提高系统的可用性和可扩展性
- **通常不支持SQL：**只支持简单的查询操作，而将复杂操作留给应用层实现
- **不保证数据事务处理的ACID特性**

# 第一章 概述

- 1. 层次数据库和网状数据库
- 2. 关系数据库
- 3. 数据仓库和数据湖
- 4. 面向对象数据库
- 5. NoSQL数据库
- 6. 多模数据库

# 多模数据库

## 多模数据库

- 支持多种数据模型的数据库，包括结构化数据、半结构化数据和非结构化数据，例如关系、键值、图、XML/JSON等。
- 挑战：多模数据管理在**统一建模、统一存储、查询优化、并发控制**等方面还存在着众多挑战性问题



Azure Cosmos DB

# 第一章 概述

- 1.2.1 数据模型的发展
- 1.2.2 应用需求推动数据库技术发展
- 1.2.3 计算平台的进步推动数据库技术发展

## 1.2.2 应用需求推动数据库技术发展

应用需求是推动数据库技术发展的直接动力，需求变化分为两方面：

- 数据处理领域从OLTP为代表的事务处理扩展到OLAP分析处理
- 特定领域数据库：空间数据库、时序数据库，科学数据库

## 1.2.2 应用需求推动数据库技术发展

- 1. OLTP VS. OLAP
- 2. 特定领域数据库

# OLTP数据库

## OLTP数据库

- **需求:** 处理在线业务，称为联机事务处理系统（On-Line Transaction Processing, OLTP）。
- **例如:** 银行通存通兑系统、火车售票系统、税务征收系统等。
- **特点:** 快速响应、高并发、安全性完整性要求高、数据更新频繁

## OLAP数据库

- **需求:** 如何分析数据库中的数据，支持商业决策。对大规模数据进行分析查询的场景称为联机分析处理（On-Line Analytical Processing，OLAP）。
- **例如:** 预算预测、销售数据分析等等
- **特点:** 大部分操作只读、并发量小、数据更新不频繁

# OLTP vs. OLAP

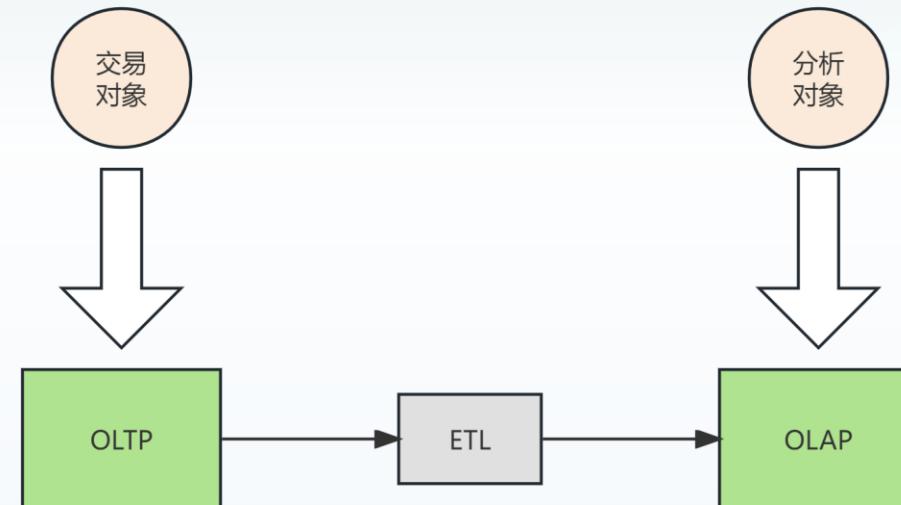
让我们来看看OLAP和OLTP系统的区别：

	OLTP系统	OLAP系统
面向的用户	面向客户，由职员或客户进行事务处理或者查询处理	面向由经理、主管和分析人员进行数据分析和决策的群体
数据内容	面向业务系统，管理当前数据	面向分析系统，管理大量历史数据，提供汇总和聚集机制，并在不同的粒度级别上存储和管理信息
操作特点	高并发且数据量级不大的查询，DML操作比较多	大部份是只读操作，主要是复杂查询
响应速度	优先级高，响应速度快	响应速度可以接受
并发访问量	大	小
事务资源消耗	小	大

# HTAP数据库

## HTAP数据库

- **需求:** 既能处理高并发的事务请求，又能够对最新数据做实时分析
- 传统数据处理流程: OLTP-> ETL (Extract-Transform-Load) ->OLAP
- 缺点: 耗时耗力、分析的数据往往已经过时



# HTAP数据库

## HTAP数据库

- 针对这类客户需求，提出了混合事务与分析处理 (Hybrid Transactional Analytical Processing, HTAP) 的概念
- 特点：HTAP数据库基于一站式架构混合处理 OLTP 和 OLAP 的负载，不再需要 ETL 过程

## 1.2.2 应用需求推动数据库技术发展

- 1. OLTP VS. OLAP
- 2. 特定领域数据库

# 特种数据库

- 随着人们对数据的价值的认识越来越深刻，围绕**不同类型的数据库应用**越来越丰富，出现了各种新形态的数据库。例如：
  - 空间数据库
  - 区块链数据库
  - 等等
- 下面选择几个给予介绍

# 特定领域数据库

- 空间数据库
- 时序数据库
- 区块链数据库
- AI数据库

# 空间数据库

## 空间数据库系统（Spatial Database System, SDBS）

- 描述、存储和处理空间数据及其属性数据的数据库系统。空间数据用于表示空间物体的位置、形状、大小和分布等各方面信息的数据，适用于描述所有二维、三维和多维分析的数据。
- 可以在传统的关系数据库基础上进行扩展实现

# 特定领域数据库

- 空间数据库
- 时序数据库
- 区块链数据库
- AI数据库

# 时序数据库

## 时序数据库（Time Series Database）

- 用于存储和管理时间序列数据的专业数据库。
- 举例：物联网设备监控、互联网业务监控

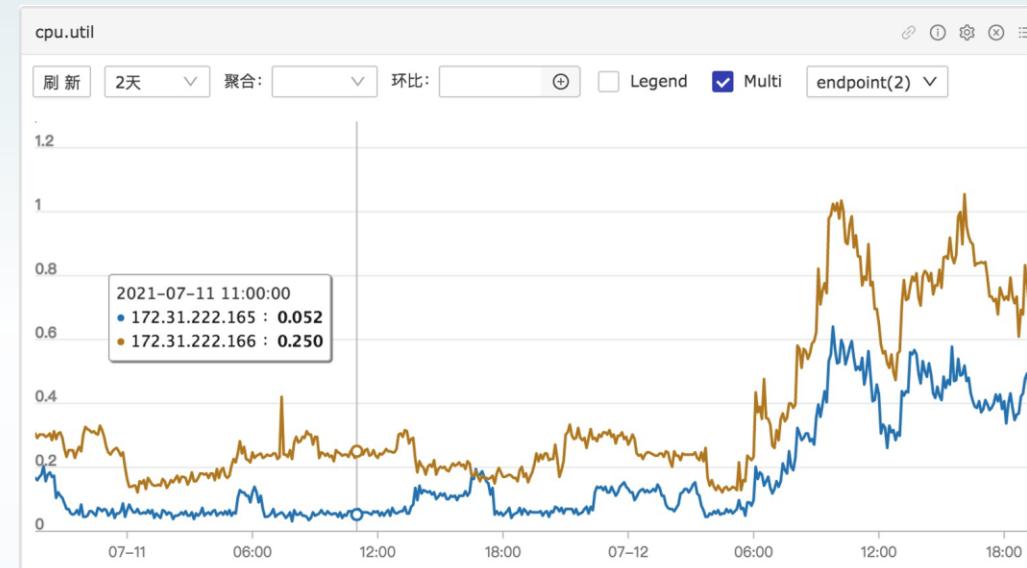


Fig. CPU时序数据

# 时序数据库

- 时序数据库主要特点
  - 持续海量数据：没有波峰波谷，例如类似哨兵的监控系统
  - 数据都是插入操作：基本没有更新或删除操作
  - 近期的数据关注度更高：时间久远的数据极少被访问，甚至可以丢弃
  - 数据存在多个维度的标签：往往需要多维度联合查询以及统计查询

# 特定领域数据库

- 空间数据库
- 时序数据库
- 区块链数据库
- AI数据库

# 区块链数据库

## 区块链数据库（Blockchain Database）

在多个主体间共享的并基于密码学技术实现可信记录的特殊分布式数据库。区块链数据库特别适用于跨不同信任域场景的数据共享和流通。

### 区块链数据库三大技术特点：

- (1) **去中心**: 不需要中心节点，所有节点地位是平等的，网络拓扑是扁平的。
- (2) **去信任**: 区块链数据库系统中可以容忍恶意节点的存在。
- (3) **极难篡改**: 采用了数字签名等密码学技术，并且通过共识协议保证所有上链的信息或对链上数据的操作都需要事先经过大多数节点认可。

# 特定领域数据库

- 空间数据库
- 时序数据库
- 区块链数据库
- AI数据库

## AI原生数据库：

直接在DBMS内部实现AI操作，不需要从数据库中导出数据。数据库和AI的结合分为两方面：

- **DB4AI:** 通过扩展SQL算子支持AI操作，实现库内的训练和推理
- **AI4DB:** 通过内置AI算法来提升数据库的智能优化和智能运维

# AI数据库

DBMS和AI的融合方案根据其融合程度可以分为三类：

**Shared-Nothing:** DBMS存储数据，分别使用SQL引擎和AI引擎进行SQL和AI操作，AI所需要的数据从DBMS中导出。该方案DB和AI无融合，需要用户进行大量的AI编程和手动的性能调整。

**Shared-data:** 通过用户自定义函数在DBMS内部实现AI操作，不需要从DBMS中导出数据，实现库内计算。该方案被很多主流的DBMS厂商采用，可以方便灵活实现AI操作，但是DB和AI仍由不同的引擎分别处理，优化少，AI操作效率低。

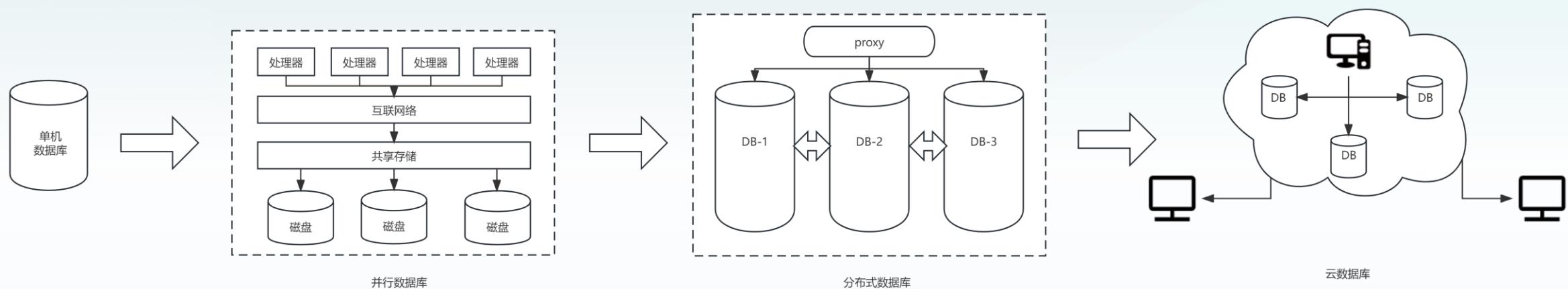
**Shared-everything:** 即统一的DB&AI融合方案，使用统一的数据模型、数据操作模型和操作优化引擎来实现DB&AI混合功能，该方案目前处于初始研究阶段，还有很多问题需要解决，例如如何设计和实现统一的数据模型、如何设计和实现统一的操作算子、如何设计和实现统一的优化引擎等。

# 第一章 概述

- 1.2.1 数据模型的发展
- 1.2.2 应用需求推动数据库技术发展
- 1.2.3 计算平台的进步推动数据库技术发展

## 1.2.3 计算平台的进步推动数据库技术发展

- **数据库是运行在计算平台之上的。**底层计算平台的变化，包括软硬件技术的发展都可能影响数据库系统技术的发展
- 从单机数据库系统，到并行数据库、分布式数据库、云数据库等



## 1.2.3 计算平台的进步推动数据库技术发展

- 并行数据库
- 分布式数据库
- 内存数据库
- 云数据库

# 并行数据库

并行计算技术利用多处理机并行处理产生的规模效益来提高系统的整体性能，根据并行粒度的不同，可以分为以下几类：

- **事务间并行性：**允许多个进程或线程同时处理多个用户的请求
- **事务内查询间并行性：**同一事务内的多个查询并行执行
- **查询内操作间并行性：**同一查询内的不同操作并行执行
- **操作内并行性：**将同一操作分解成多个独立的子操作，由不同的处理机同时执行

# 并行数据库分类

**并行数据库架构分类：**并行计算平台根据硬件架构的不同可以划分为四种体系结构：共享内存结构、共享磁盘结构、无共享结构，还有混合结构。

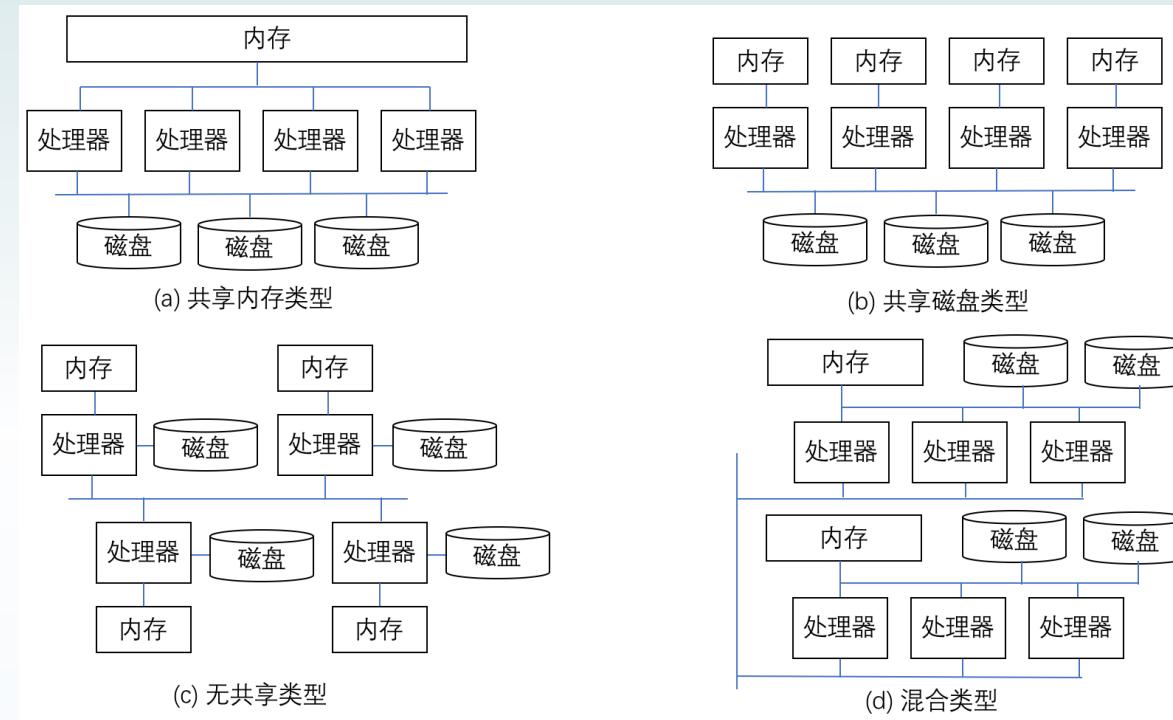


Fig. 并行数据库的四种类型

## 1.2.3 计算平台的进步推动数据库技术发展

- 并行数据库
- 分布式数据库
- 内存数据库
- 云数据库

# 分布式数据库

## 为什么要使用分布式数据库？

1. 满足在地理上分散的公司、团体和组织对数据管理的需求
2. 解决性能需求，通过增加机器数量，把数据库同时部署在多台机器上

# 分布式数据库分类

分布式数据库系统主要解决**数据规模和性能问题**。

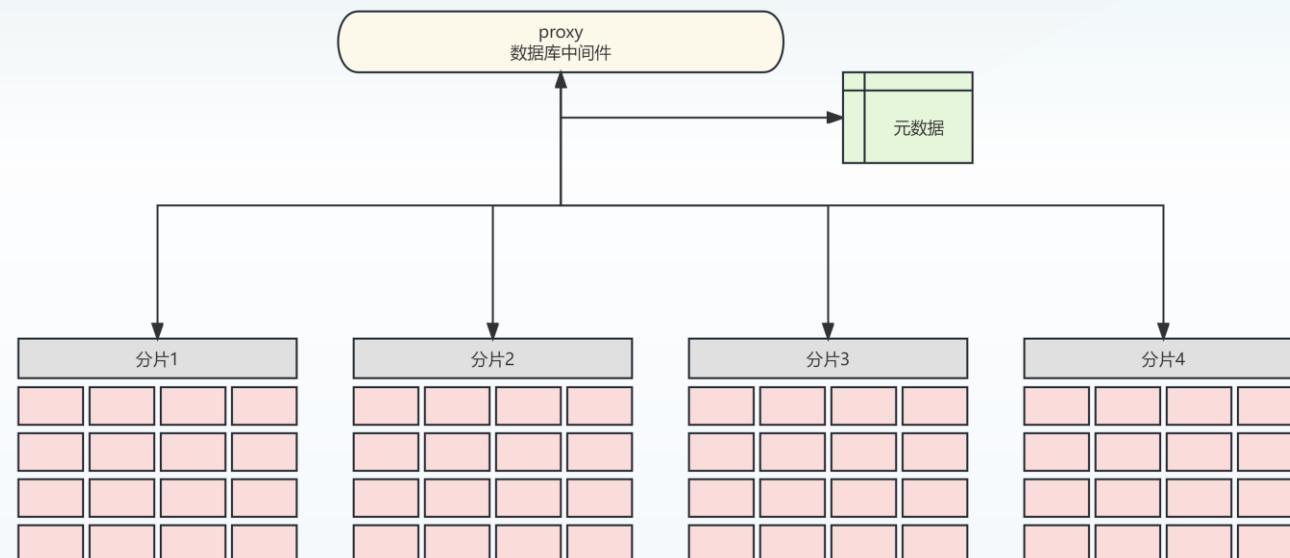
主要有以下几种类型：

- 中间件模式
- Shared-Nothing架构的并行数据库
- NoSQL分布式数据库
- NewSQL分布式数据库
- 云原生分布式数据库

# 中间件模式

## 中间件模式：

- 早期互联网应用面对高并发场景通常采用的模式
- 每个节点都是一个独立的单机数据库，通过**分库分表**的方式，把数据部署在各个节点上，通过中间件对这些节点统一管理。
- 典型的代表是阿里的DRDS、腾讯的DCDB等。



# Shared-Nothing架构

## Shared-Nothing架构：

- 解决海量数据的复杂查询的性能问题，通过把数据分布到多台机器中，通过大规模并行处理技术提高复杂查询的性能
- 典型的代表是Teradata、GreenPlum等商业或开源数据库产品。



# NoSQL分布式数据库

## NoSQL分布式数据库：

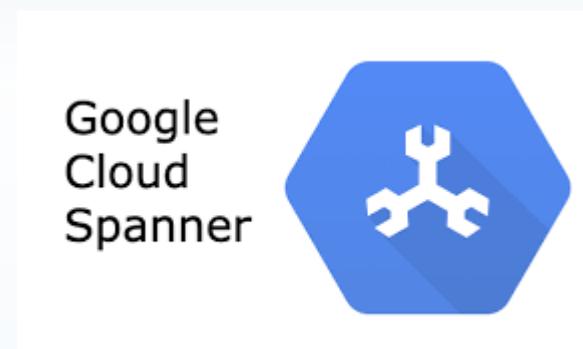
- 主要是要解决大数据环境下数据规模膨胀，单机无法保存全部数据的问题。
- 典型的代表有HBase、Cassandra和MongoDB等



# NewSQL分布式数据库

## NewSQL分布式数据库：

- 主要是要解决互联网应用高并发事务处理的性能、高可用和弹性伸缩问题，希望提供完整的SQL和事务处理能力
- 典型的代表有Google的Spanner/F1系统，开源的TiDB、CockroachDB和OceanBase。



# 云原生分布式数据库

## 云原生分布式数据库：

- 主要解决在云环境下资源的充分利用问题，提出**计算与存储分离**，分别使用不同的资源，单独扩展，并采用“**日志即数据库（Log is database）**”技术，减少计算节点和存储节点的通信，更适合云环境。
- 典型的代表有AWS Aurora和阿里云的PolarDB系统。



## 1.2.3 计算平台的进步推动数据库技术发展

- 并行数据库
- 分布式数据库
- 内存数据库
- 云数据库

# 内存数据库

## 内存数据库

- 磁盘数据库是将磁盘作为主存储设备，将内存作为缓冲区使用；
- 而内存数据库是将内存作为主存储设备的数据库系统，目的是充分利用内存的优势提高系统的性能。

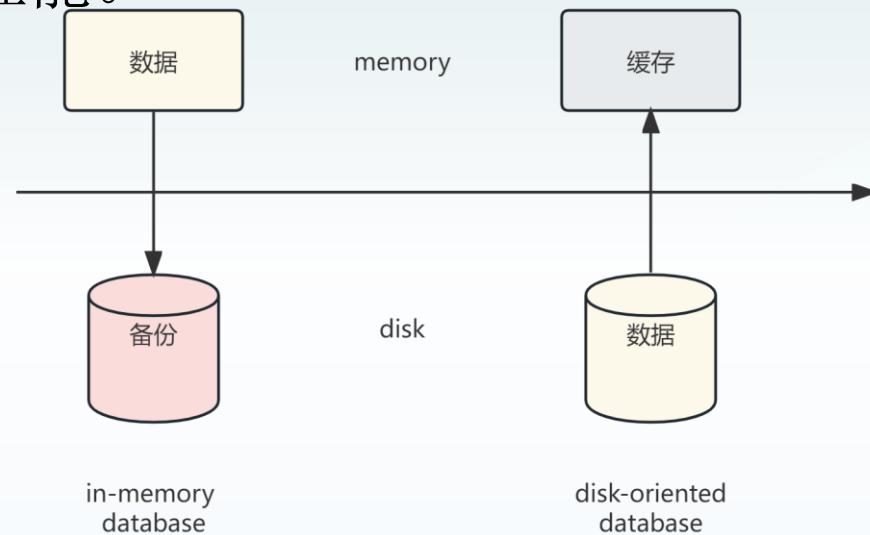


Fig. disk vs. in-memory数据库

# 内存数据库

## 内存数据库 vs. 将磁盘数据全部load到内存的磁盘数据库

- **针对内存优化：** 内存数据库中，数据的存储结构、访问算法以内存访问特性为基础进行设计优化、
- **持久化机制：** 由于内存是易失性存储介质，事务持久性的满足需要借助特殊的硬件设备、系统设计和实现机制来完成，例如日志技术、数据库复制技术。
- **性能高：** 跟数据全部缓存到内存的磁盘数据库相比，内存数据库的性能仍高出数倍

## 1.2.3 计算平台的进步推动数据库技术发展

- 并行数据库
- 分布式数据库
- 内存数据库
- 云数据库

## 什么是云计算？

- 云计算是利用**虚拟化、容器等技术**在多样化硬件上建立一个庞大的操作系统。
- **弹性服务：**利用资源池化的能力，根据用户的算力需求，为用户提供一个弹性的资源服务。
- **优势：**用户不再需要关注底层硬件差异化、网络、负载均衡等细节。

# 云数据库

随着云计算的发展，云数据库近十年也得到蓬勃发展，大体可以分为云托管、云服务和云原生三种模式。

# 云数据库

## 云托管模式：

- 将原本部署于本地物理服务器上的传统数据库部署到云主机上。
- 云厂商仅提供云主机，而不提供数据库的运维服务，由用户负责运维
- 使用传统数据库，没有针对云计算平台改造。

# 云数据库

## 云服务模式：

- 云厂商将传统数据库部署到云基础设施上，以云服务的模式给用户提供多种传统的数据库服务，例如，MySQL、SQL Server、PostgreSQL等。
- 用户可以直接使用云厂商提供的数据库服务，不用关心数据库的安装、运维等问题。
- 在这种模式下，只需要对原来的DBMS进行微改造，可以运行在云上，并提供对云上数据库管理即可

## 云原生数据库

- 充分利用云计算的优势，例如按需使用，弹性伸缩的特点
- 云原生数据库最核心的挑战是如何实现弹性伸缩，让资源按需按量使用，发挥资源的最大效能

# 云数据库

## 云原生核心技术：

- (1) 存算分离：将计算和存储解耦分离，实现了独立的计算节点和存储节点的弹性伸缩。
- (2) 日志即数据：数据脏页面无须写回存储节点，只需要将日志写回存储节点，存储节点通过日志回放来获取最新数据，从而减少了计算节点和存储节点的通信量，降低了云基础设施的网络压力；
- (3) 一写多读：支持多个节点的一致性读

# 云数据库

目前的云数据库还有很多问题需要解决

- 一写多读的特性使得数据库的扩展性和高可用性都受到限制
- 云数据库的数据安全、访问性能、HTAP混合负载的支持等问题都需要解决，  
目前云数据库也是研究热点和发展方向。

# 课堂思考

- 层次模型与网状模型存在的问题?
  - 数据查询和数据操纵方面都采用的是一次一个记录的导航式过程化语言。
  - 需要用户明确数据的存储结构，预设数据访问路径，编程繁琐，应用程序的可移植性较差。



# 第一章 概述

- 1. 1 数据库系统概述
- 1. 2 数据库技术的发展
- 1. 3 数据库管理系统组成
- 1. 4 我国数据库的发展历程
- 1. 5 小结

# 1.3 数据库管理系统的组成

本节从数据库管理系统的功能结构、模式结构、体系结构和运行时系统等多个视角介绍数据库系统的组成。

# 数据库管理系统组成

- 1.3.1 数据库管理系统的功能结构
- 1.3.2 数据库管理系统的模式结构
- 1.3.3 数据库管理系统的体系结构
- 1.3.4 数据库管理系统的执行流程

# 1.3.1 数据库管理系统的功能结构

数据库管理系统包括数据库查询语言处理器、事务管理器、存储管理器等



Fig.数据库管理系统的功能结构图

# 查询语言处理器

查询语言处理器：

- 将用户的数据库查询语句（SQL）解释形成执行计划。
- 具体分为以下几步：
  - **语法解释**: 词法解析和语法检查
  - **语义检查**: 数据类型检查，约束条件检查（需要借助数据字典）
  - **安全性检查**: 访问权限检查
  - **查询优化**: 逻辑优化和物理优化

# 数据库事务管理器

## 事务管理器

- **什么是事务？** 事务是用户定义的一个操作序列，并满足以下4个特性：原子性、一致性、隔离性和持久性，合称为ACID特性。
- **事务管理器的工作就是保证事务的ACID特性：** 具体的说，事务管理器需要处理多个事务的并发调度并提供一定的故障恢复能力

# 数据库存储管理器

## 数据库存储管理器

- 组织数据在存储器上的存放方式，尽力取得在存储空间和存取时间上的最优
- 具体地说，对于磁盘数据库，由于磁盘I/O的时间相较于CPU的处理时间要慢得多，为了匹配，内存作为处理器和外存之间的缓存。这就需要对缓存进行管理。

# 数据库管理系统组成

- 1. 3. 1 数据库管理系统的功能结构
- 1. 3. 2 **数据库管理系统的模式结构**
- 1. 3. 3 数据库管理系统的体系结构
- 1. 3. 4 数据库管理系统的执行流程

## 1.3.2 数据库管理系统的模式结构

首先介绍“型”和“值”的概念

- 型 (Type) 对某一类数据的结构和属性的说明

- 值 (Value) 是型的一个具体赋值

- 例如学生记录:

- (学号, 姓名, 性别, 系别, 年龄, 籍贯)

- 一个记录值:

- (201915130, 李明, 男, 计算机系, 19, 江苏南京市)

## 1.3.2 数据库管理系统的模式结构

- 模式 (**Schema**)
  - 数据库逻辑结构和特征的描述
  - 是型的描述，不涉及具体值
  - 反映的是数据的结构及其联系
  - 模式是相对稳定的
- 实例 (**Instance**)
  - 模式的一个**具体值**
  - 反映数据库某一时刻的状态
  - 同一个模式可以有很多实例
  - 实例随数据库中的数据的更新而变动

# 数据库系统的三级模式结构

- 模式 (Schema)
  - 外模式 (External Schema)
  - 内模式 (Internal Schema)

# 数据库系统的三级模式结构

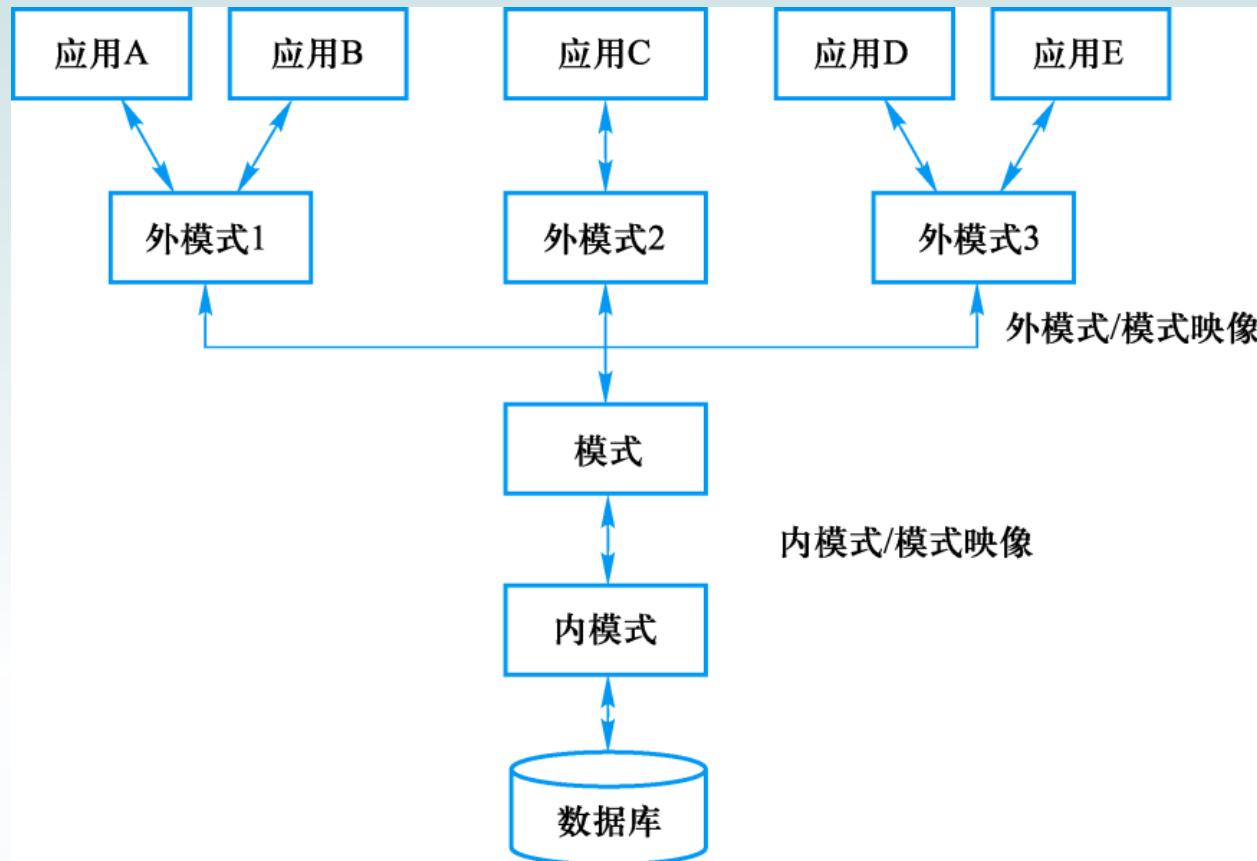


Fig. 数据库系统的三级模式结构

# 模式 (Schema)

- 模式 (也称逻辑模式)
  - 数据库中全体数据的逻辑结构和特征的描述
  - 所有用户的公共数据视图
- 一个数据库只有一个模式
- 模式的地位:
  - 是数据库系统模式结构的中间层
  - 与数据的物理存储细节和硬件环境无关
  - 与具体的应用程序、开发工具及高级程序设计语言无关

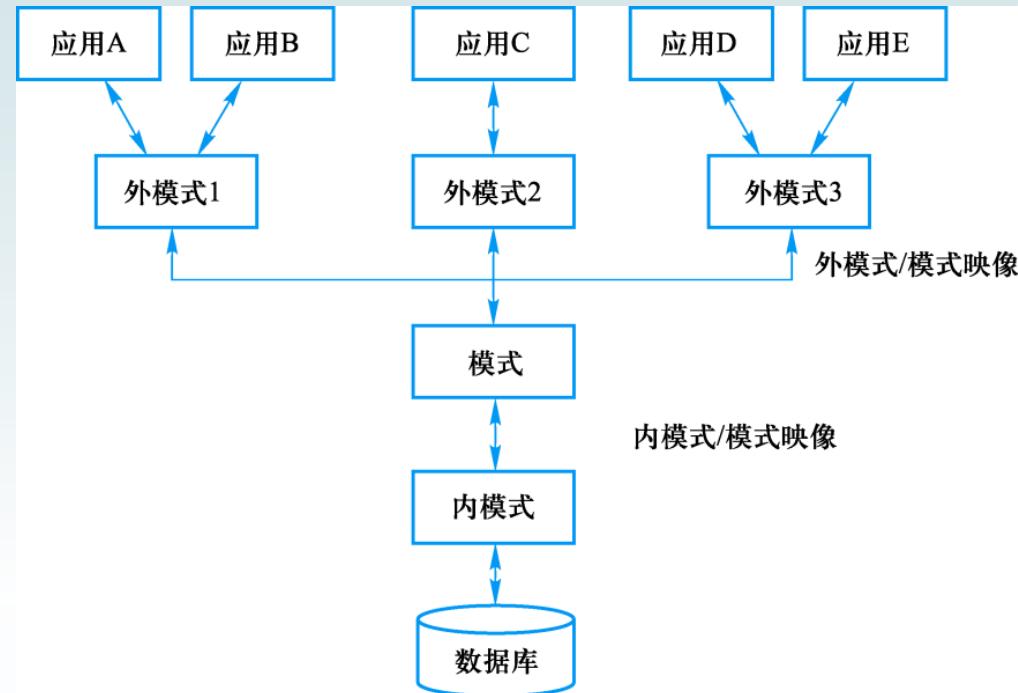


Fig. 数据库系统的三级模式结构

# 模式 (Schema)

## 模式定义

- 数据的逻辑结构（数据项的名字、类型、取值范围等）
- 数据之间的联系
- 数据有关的安全性、完整性要求

# 外模式 (External Schema)

## 外模式 (也称子模式或用户模式)

- 数据库用户（包括应用程序员和最终用户）使用的**局部数据的逻辑结构和特征的描述**
- 数据库用户的数据视图，是与某一应用有关的数据的逻辑表示

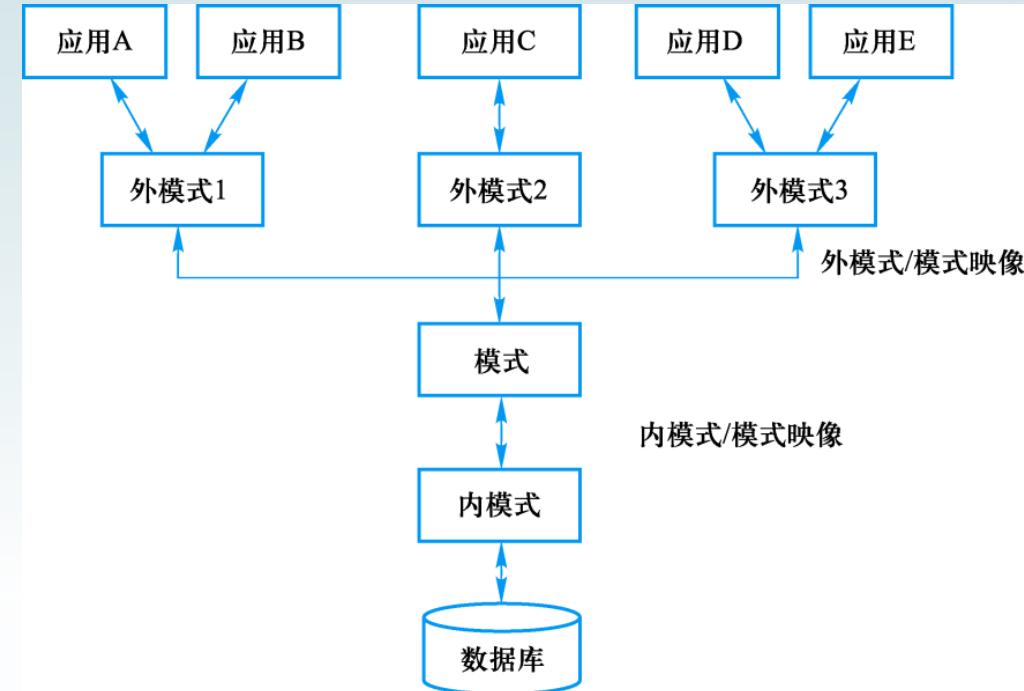


Fig. 数据库系统的三级模式结构

# 外模式 (External Schema)

- 外模式的地位：介于模式与应用之间
- 模式与外模式的关系：一对多
  - 外模式通常是模式的子集
  - 一个数据库可以有多个外模式。反映了不同的用户的应用需求、看待数据的方式、对数据保密的要求
  - 对模式中同一数据，在外模式中的结构、类型、长度、保密级别等都可以不同
- 外模式与应用的关系：一对多
  - 同一外模式也可以为某一用户的多个应用系统所使用
  - 但一个应用程序只能使用一个外模式

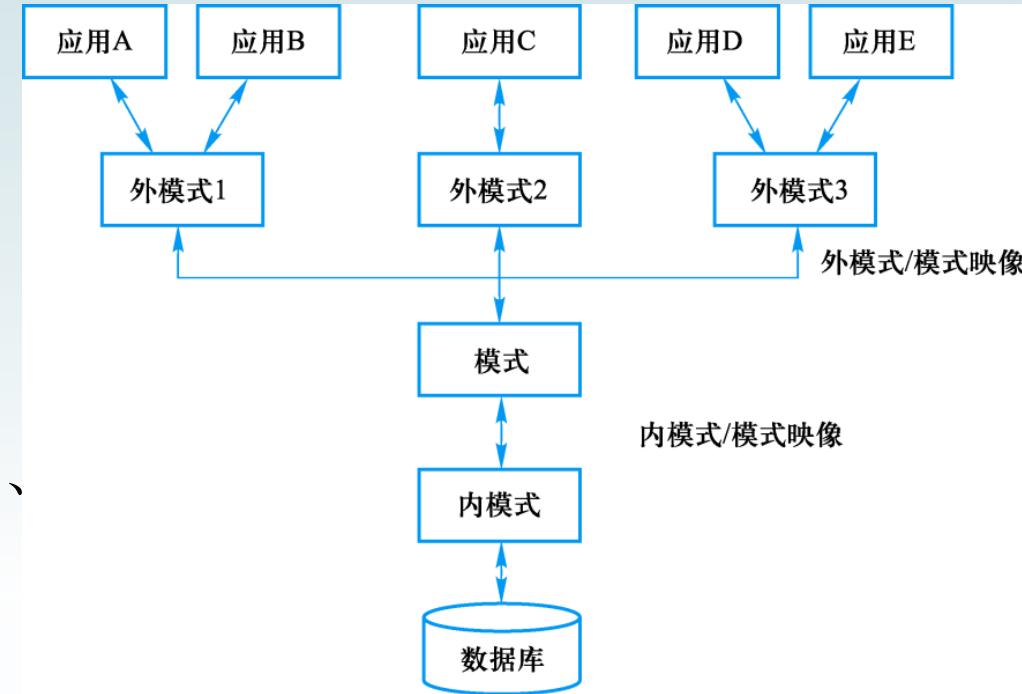


Fig. 数据库系统的三级模式结构

# 外模式 (External Schema)

## 外模式的用途

- 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

# 内模式 (Internal Schema)

## 内模式 (也称存储模式)

- 是数据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
  - 记录的存储方式（例如，顺序存储，按照B树结构存储，按hash方法存储等）
  - 索引的组织方式
  - 数据是否压缩存储
  - 数据是否加密
  - 数据存储记录结构的规定
- 一个数据库只有一个内模式

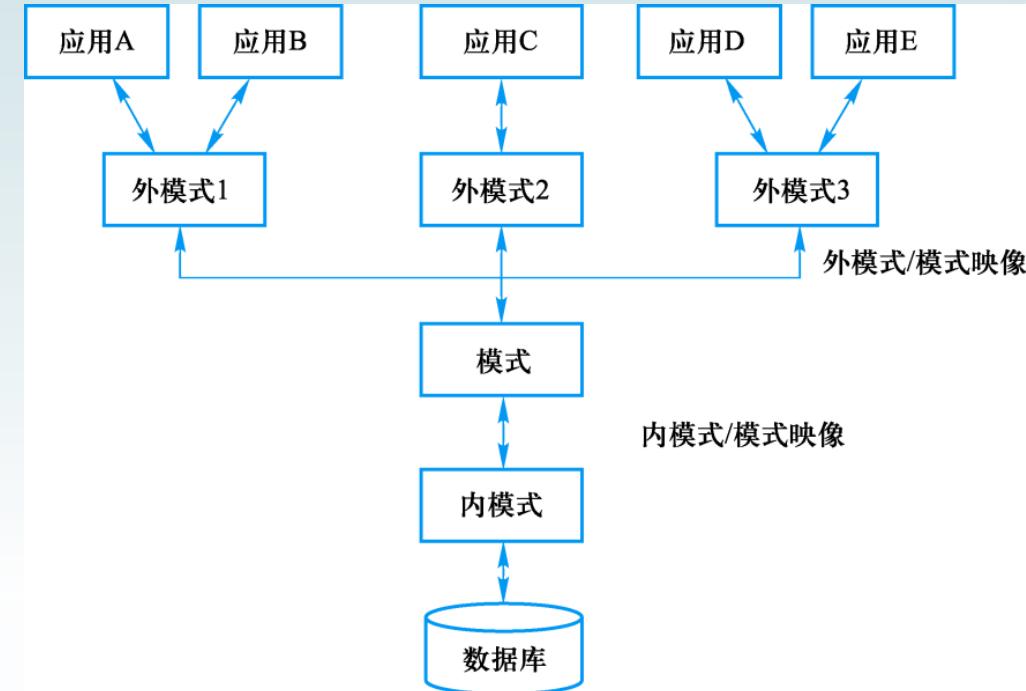


Fig. 数据库系统的三级模式结构

# 数据库的二级映像功能与数据独立性

- 三级模式是对数据的三个抽象级别
- 二级映象在数据库管理系统内部实现这三个抽象层次的联系和转换
  - 外模式 / 模式映像
  - 模式 / 内模式映像

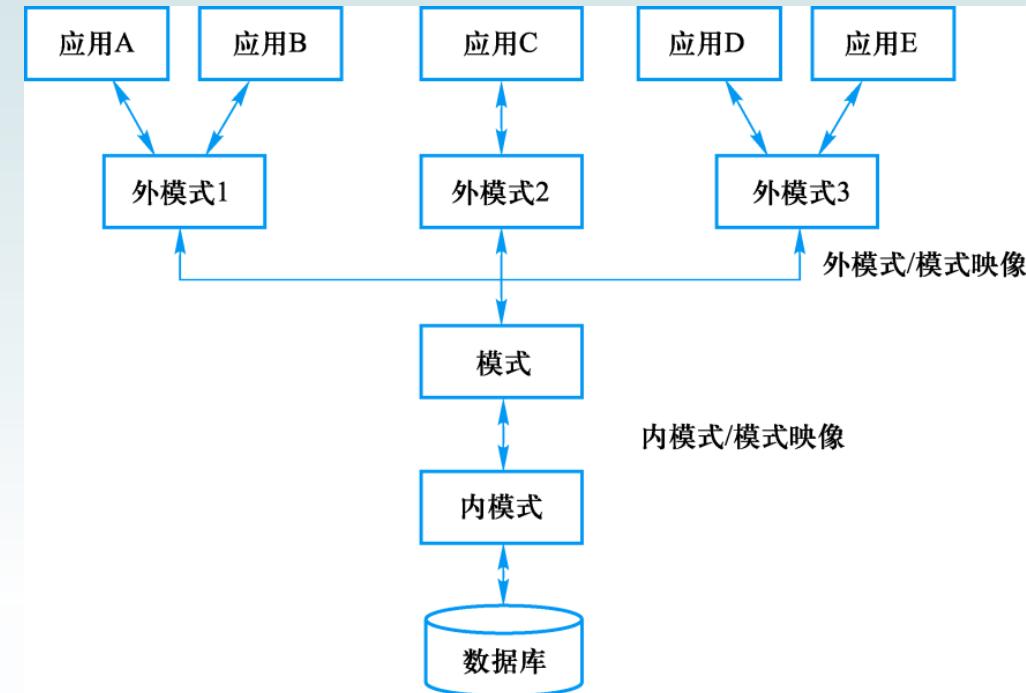


Fig. 数据库系统的三级模式结构

# 数据库的二级映像功能与数据独立性

- 外模式 / 模式映像
- 模式 / 内模式映像

# 外模式/模式映射与数据逻辑独立性

- **模式**: 描述的是数据的全局逻辑结构
- **外模式**: 描述的是数据的局部逻辑结构
- 同一个模式可以有任意多个外模式
- 每一个外模式，数据库系统都有一个外模式 / 模式映象，定义外模式与模式之间的对应关系
- 映象定义通常包含在各自外模式的描述中

# 外模式/模式映射与数据逻辑独立性

## 外模式/模式的映像保证数据的逻辑独立性

- 当模式改变时，数据库管理员对外模式 / 模式映象作相应改变，使外模式保持不变
- 应用程序是依据数据的外模式编写的，应用程序不必修改，保证了数据与程序的逻辑独立性，简称数据的逻辑独立性

# 数据库的二级映像功能与数据独立性

- 外模式 / 模式映像
- 模式 / 内模式映像

# 模式 / 内模式映像

- 模式 / 内模式映象：定义了数据全局逻辑结构与存储结构之间的对应关系。例如，说明逻辑记录和字段在内部是如何表示的
- 数据库中模式 / 内模式映象是唯一的
- 该映象定义通常包含在模式描述中

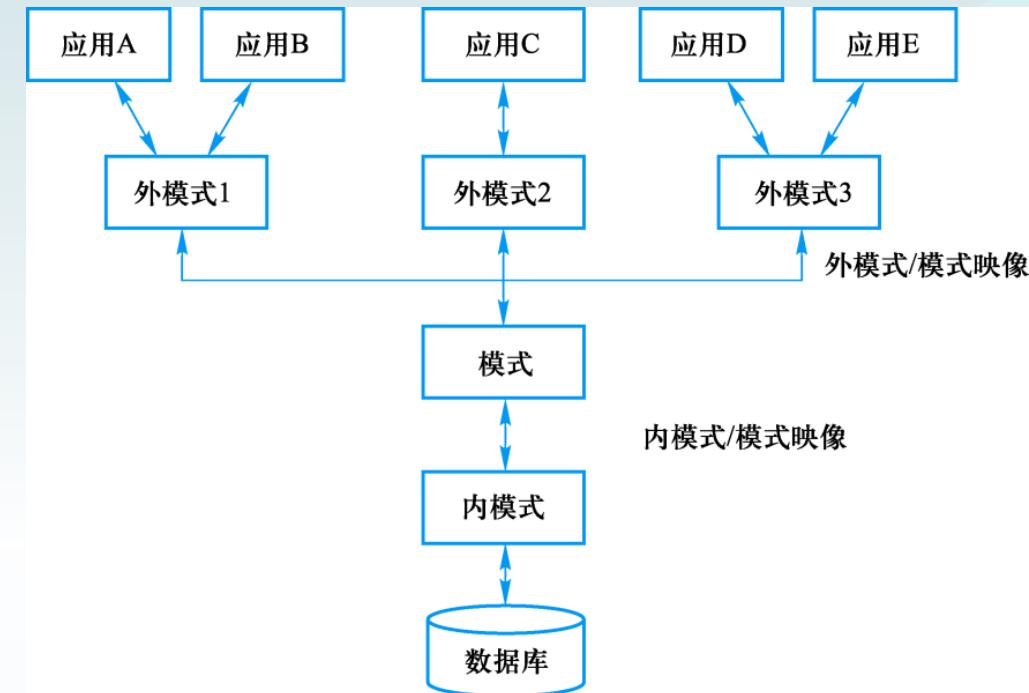


Fig. 数据库系统的三级模式结构

# 模式 / 内模式映像

## 模式/内模式映像保证数据的物理独立性

- 当数据库的存储结构改变了（例如选用了另一种存储结构），数据库管理员修改模式 / 内模式映象，使模式保持不变。
- 应用程序不受影响。保证了数据与程序的物理独立性，简称数据的物理独立性。

# 数据库管理系统组成

- 1. 3. 1 数据库管理系统的功能结构
- 1. 3. 2 数据库管理系统的模式结构
- 1. 3. 3 数据库管理系统的体系结构
- 1. 3. 4 数据库管理系统的执行流程

### 1.3.3 数据库管理系统的体系结构

数据库系统体系结构与其所依赖的计算机系统架构直接相关。

- 传统的单节点计算机系统--集中式数据库系统
- 网络出现后--客户-服务器体系结构/浏览器-服务器体系结构
- 多核时代--并行体系结构
- 大数据时代--分布式体系结构
- 云计算的出现--云计算架构

### 1.3.3 数据库管理系统的体系结构

- 集中式数据库系统
- 客户-服务器/浏览器-服务器体系结构
- 现代数据库系统

# 集中式数据库系统

## 集中式数据库系统

- 数据库管理系统、数据库和应用程序都在一台计算机上。
- 多用户系统--在小型机和大型机上的集中式数据库系统一般是多用户系统，即多个用户通过各自的终端运行不同的应用系统，共享数据库。
- 单用户系统--微型计算机上的数据库系统一般是单用户的。

### 1.3.3 数据库管理系统的体系结构

- 集中式数据库系统
- 客户-服务器/浏览器-服务器体系结构
- 现代数据库系统

# 客户机/服务器数据库系统

## 客户机/服务器数据库系统

- **服务器**---数据库管理系统和数据库驻留在服务器上
- **客户机**---应用程序放置在客户机
- **通信**---客户机和服务器通过网络进行通信

# 客户机/服务器数据库系统

在这种结构中，数据库系统可以划分为前端和后端

- 前端--SQL的用户界面、报表界面、报告生成工具、和数据挖掘与分析工具等
- 后端--访问、查询对应的数据，维护并发控制和故障恢复等
- 前端和后端通过SQL或应用程序等相连接

当客户机要存取数据库中的数据时就向服务器发出请求，服务器接收客户机的请求后进行处理，并将客户要求的数据返回给客户机。

### 1.3.3 数据库管理系统的体系结构

随着互联网技术的应用，客户机/服务器两层结构已经发展为三层或多层结构。

- 三层结构一般是指浏览器/应用服务器/数据库服务器结构。
- 用户界面采用统一的浏览器方式，应用服务器上安装应用系统或应用模块，数据库服务器上安装数据库管理系统和数据库。
- 功能再解耦，减轻数据库服务器的负担，从而使服务器有更多的能力完成事务处理和数据访问控制

### 1.3.3 数据库管理系统的体系结构

- 集中式数据库系统
- 客户-服务器/浏览器-服务器体系结构
- 现代数据库系统

# 现代数据库系统

现代计算机系统出现了**并行计算、分布式计算、云计算等新兴计算架构**:

- **并行数据库系统**: 在并行计算机上运行的具有并行处理能力的数据系统
- **分布式数据库系统**: 数据库中的数据在逻辑上是一个整体，但物理地分布在计算机网络的不同结点上。
- **云数据库系统**: 数据库部署或虚拟化在云计算环境下，通过计算机网络以服务的形式提供数据库的功能

# 数据库管理系统组成

- 1. 3. 1 数据库管理系统的功能结构
- 1. 3. 2 数据库管理系统的模式结构
- 1. 3. 3 数据库管理系统的体系结构
- 1. 3. 4 数据库管理系统的执行流程

## 1.3.4 数据库管理系统的执行过程

一条SQL语句是如何在数据库管理系统中被执行的呢？

- 粗略地说，要经历如下这几步，语法检查、语义检查、访问控制、查询优化、查询执行、事务管理、数据存取、结果输出等。
- 假设有一个查询：`SELECT * FROM Student WHERE Age < 17`（即查询“小于17岁的学生”的情况）。我们看看数据库管理系统是如何处理并获得结果的。

# 语法分析与语义检查

- DBMS接受到一条SQL语句（如右图（1））
- 词法分析和语法分析：先对SQL语句进行语法分析，是否符合SQL的语法。
- 以SELECT \* FROM Student WHERE Age < 17为例：  
词法分析阶段将SQL语句分解为SELECT、\*、  
FROM、Student、WHERE、Age、<、17等词。然  
后语法分析检查是否符合SELECT的语法

SELECT \* FROM Student  
WHERE Age < 17

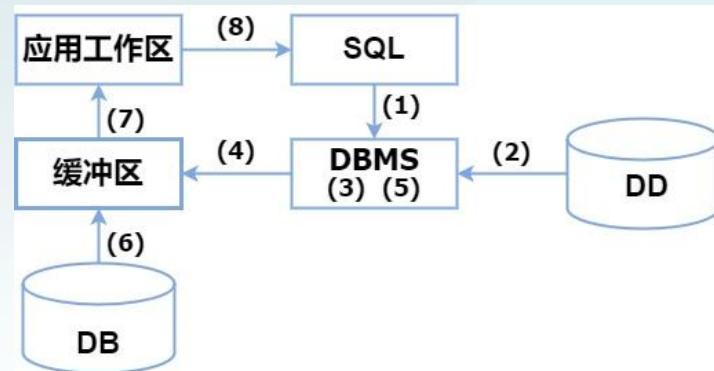


Fig. SQL执行流程

# 语法分析与语义检查

- **语义检查、完整性检查：** 检查SQL的数据库名，表名，字段等是否正确，数据类型是否匹配（如右图（3））。这一步依赖数据字典中关于表、字段、授权等描述信息（如右图（2））。
- **以SELECT \* FROM Student WHERE Age < 17为例：** 这里会检查数据库中表Student是否存在以及Age是否为表Student的字段

**SELECT \* FROM Student  
WHERE Age < 17**

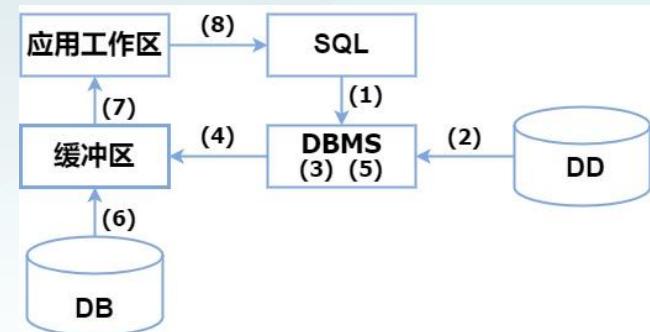


Fig. SQL执行流程

# 语法分析与语义检查

- 什么是数据字典？

**数据字典：**可以理解为系统表，记录的是关于数据对象的表述。比如，对“关系表”的描述，可以包括表名、表的创建者、表的属性个数、表的记录长度、表的记录个数等

SELECT \* FROM Student  
WHERE Age < 17

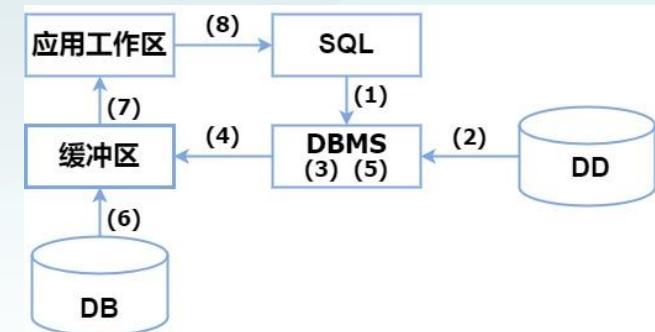


Fig. SQL执行流程

# 访问权限检查

访问控制：检查用户是否有对数据的访问权限

- 有关的信息也是放在数据字典里。
- 这类检查有些是静态就可以完成的，有些还需要等到获得具体数据才能决定。例如，对于SQL语句：**SELECT \* FROM Student WHERE Age < 17**，如果用户对student表没有访问权限，这个时候就可以检查出来，并反馈错误信息。

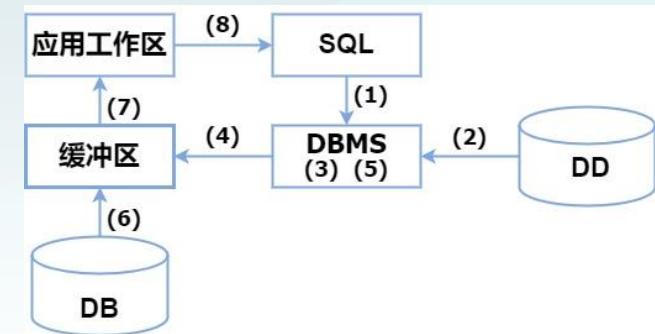


Fig. SQL执行流程

# 查询优化

## 查询计划

- **查询计划**--DBMS需要将SQL语句转换成可执行的代码，即查询执行计划
- **基本算子**--DBMS需要预先实现一些“基本算子”，也就是最基础的一些操作。例如，关系的选择操作，投影操作，连接操作等。
- **语法树结构**--DBMS需要将SQL语句转换成内部的表达，比如语法树结构，其中的树节点就是一些基本算子或数据库对象构成。

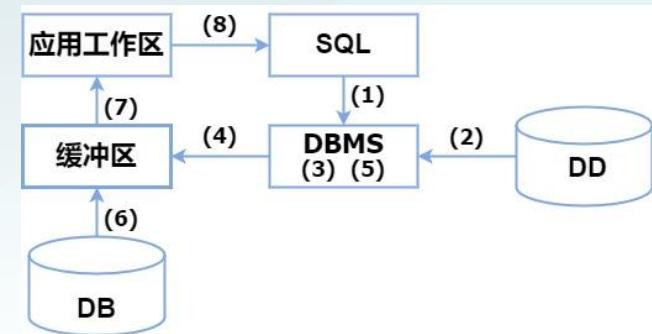


Fig. SQL执行流程

# 查询优化

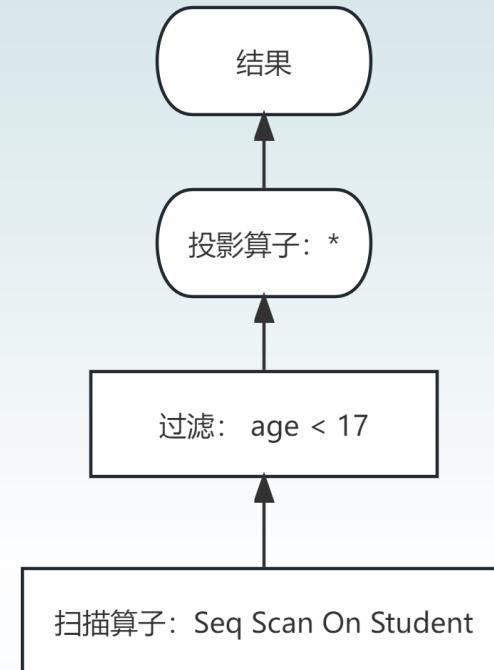


Fig.查询计划示例

# 查询优化

由于SQL语句的抽象性和数据库执行环境的复杂性，同一个SQL语句存在多个“结果等价、但效率差异巨大”的执行计划，这就为查询优化提供了空间。如何为一个SQL语句找出执行效率最高的执行计划是DBMS的主要挑战之一。

# 查询执行

查询执行：DBMS按照查询执行计划逐一完成对基本算子的执行

- **执行算子：**以选择算子为例，要从student表中选择出年龄小于17岁的学生，需要顺序扫描student表，并逐一检查每一个记录是否满足条件。
- **中间结果：**基本算子执行的结果形成一个临时文件，即可以缓存在内存中，必要的时候也可以保存在外存磁盘上。这个临时文件将作为执行计划中下一个基本算子的输入，参与下一个基本算子的执行。

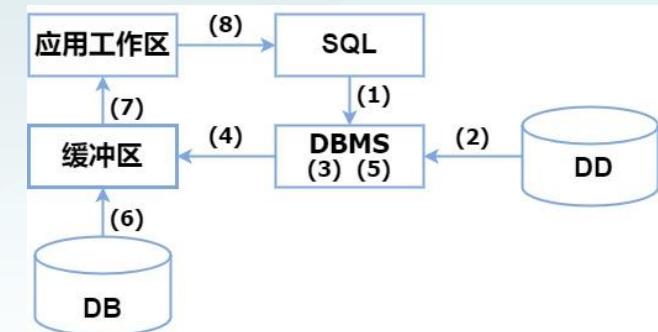


Fig. SQL执行流程

# 查询执行

查询执行：DBMS按照查询执行计划逐一完成对基本算子的执行

- 输出结果：最后，整个查询的结果将输出到用户的工作区中（如右图（7）），并以合适的方式反馈给用户（如右图（8））

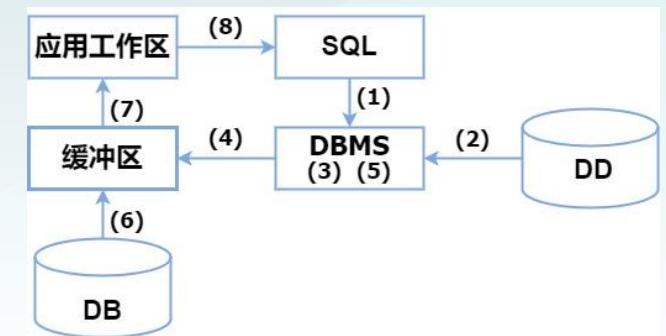


Fig. SQL执行流程

# 事务管理

## 事务管理

- 事务的并发控制---在执行基本算子，需要对数据进行读取的时候，还要考虑其他用户的干扰
- 事务的故障恢复---DBMS还要负责处理在查询执行的过程中，系统突然崩溃的异常情况，确保数据库不被破坏

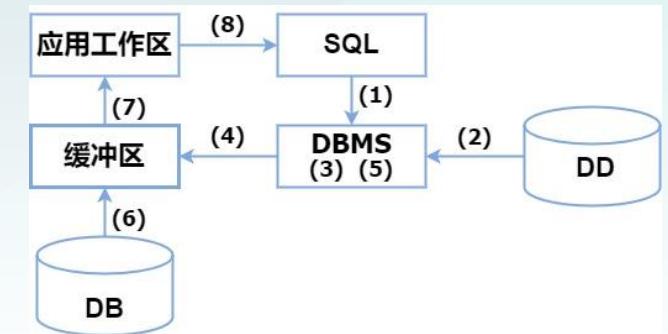


Fig. SQL执行流程

# 第一章 概述

- 1.1 数据库系统概述
- 1.2 数据库技术的发展
- 1.3 数据库管理系统组成
- 1.4 我国数据库的发展历程
- 1.5 小结

# 1.3 我国数据库的发展历程

## 国产数据库

- 20世纪70年代---引进
- 20世纪80年代---跟踪
- 20世纪90年代---原型开发和应用
- 21世纪前10年---苦苦追赶
- 21世纪后10年---快速发展的过程。特别是在“十五”、“十一五”和“十二五”期间，国家设立重大科技专项，将数据库管理系统的產品化作为信息技术的基础软件产品加以支持，使得国产数据库进步显著，在许多关键领域获得广泛应用。

# 起步期

## 起步期（1977-1999）

- 我国的数据库研究和开发工作始于20世纪70年代。
- 标志性的事件---1977年黄山数据库学术交流会，这次会议后来被学术界“确认”为全国第一次数据库学术会议



# 起步期

## 起步期

- 20世纪80年代初期，是我国引进数据库产品并开始实施应用的时期。
- 80年代初期，dBASE II进入中国：对于我国数据库的普及发挥了重要的作用，基于dBASE的数据库应用系统也开始在全国迅速展开。
- 80年代中期，ORACLE数据库开始进入中国：通过一段时间数据库应用的实践，人们也逐渐认识到了dBASE数据库的不足，在一些比较重要的信息系统中，开始放弃dBASE而采用ORACLE系统。

# 起步期

“九五”计划期间，虽然没有大规模的数据库研制活动，但是学术界仍然在艰苦地探索，国家863计划也继续支持国产数据库的产品研发，华中科技大学研制关系数据库系统，中国人民大学研制并行数据库系统，还有其它单位研制安全数据库等

# 追趕期

## 追趕期

- 进入到21世纪，中国加入WTO，市场更加国际化。
- 全球的数据库厂商也纷纷进入中国：其中包括IBM的DB2，Sybase公司的Sybase，微软公司的SQL Server等，也包括开源数据库PostgreSQL和MySQL等。
- 掀起开发国产数据库的热潮
  - 中国是一个大国，国家的信息安全是一直悬在我们头上的一把利剑；
  - 中国的信息化建设成本居高不下，严重影响了信息化的进程。

这两条理由，让国家下决心再一次开发国产数据库。

# 追赶期

## 追赶期

- 2002年，国家科技部在863计划中设立了“数据库管理系统及其应用”重大专项
- 2008年，国家发布了“核心电子器件、高端通用芯片及基础软件产品”科技重大专项，安排了“大型通用数据库管理系统与套件研发及产业化”和“非结构化数据管理系统”两类课题。
- 近10年的接续努力，我国数据管理软件核心技术和产品应用取得了重要进展。金仓、达梦为代表的国内厂商在规模上均有较大发展，形成了一支有技术能力的研发队伍以及市场推广队伍，成为国产数据库的生力军。

# 发展期

## 发展期（2010-2020）

- **大数据和云计算的出现：**给我国的数据库技术发展带来了前所未有的机会。中国政府发布了大数据行动纲要等文件，积极推动大数据的发展和应用，已经成为新一轮信息化建设的代名词。
- **信息技术的应用前沿：**
  - 以前，我们和国外的信息化程度相比差距大，我们的应用需求都是国外几年前的需求，因此，技术上大多是“跟风”，难以有创新。
  - 阿里巴巴、腾讯、蚂蚁等大型互联网企业，以及华为等技术驱动型企业，紧紧抓住了这个机会，自主创新，实现跨越发展。

# 第一章 概述

- 1.1 数据库系统概述
- 1.2 数据库技术的发展
- 1.3 数据库管理系统组成
- 1.4 我国数据库的发展历程
- 1.5 小结

# 1.5 小结

- **四个概念---**数据、数据库、数据库管理系统和数据库系统
- **数据库的发展历程---**从数据模型演化、数据库应用发展和计算平台迭代三个维度，形成了诸多的数据库类型，展示了繁荣的数据库大家族盛况
- **从多个视角介绍数据库内涵---**包括数据库管理系统的组成、数据库三级模式结构、数据库体系结构、数据库管理系统的执行过程。
- **国产数据库的发展历程---**我国数据库工作者的不懈努力和取得的成果。