

高级数据库系统及其应用

第1部分 数据库系统基础

第1章 数据库系统导论

课程QQ群：

126524799

xshxie@ustc.edu.cn

LOGO



群名称：高级数据库

群 号：126524799

课程基本实践

1. 安装、配置和熟练应用**Mysql**系统。
2. 安装**&**应用两款主流的可视化客户端工具**workbench**和 **Navicat**，辅助操纵数据库。
3. 创建课程基本实验的数据库模式**saildb**，充分实验关系数据库的各类**SQL**操作。

课程教学内容与目标



1. 课程教学内容

(1) 数据库及其应用基础 (**ch1-3**)

- 核心概念 (数据模型/关系数据模型)
- 关系数据库模型理解、应用
- 关系数据库表设计规范理论

(2) 高级主题 (实现机制理解/高级应用)

- 存储管理 (**ch4-5**)
- 查询管理 (**ch6-7**)
- 事务管理 (**ch8-9**)

2. 课程教学/学习目标

课程开放作业

1. 撰写数据库技术的现状与发展趋势综述报告（**10-20**页）。
2. 基于诺依前后端分离框架，开发学习/考试管理系统。
 - ❑ 前端vue3.x，涉及js, typescript语言；
 - ❑ 后端SpringBoot 3.x（涉及java, MyBatis）。
 - ❑ 可以找开源系统作为基础、小组合作（小组人数不要超过**10**人）
 - ❑ 项目系统可以小组合作，但项目开发工作报告各自独立撰写。

第1章 数据库系统导论



1.1

数据模型

1.2

数据库方法特点与应用简史

1.3

数据库系统体系结构

1.4

数据库系统的发展与演化

1.1 数据模型

1.1.1 数据模型的定义与分类

1.1.2 一些典型的数据模型特点综述

什么是
模型？

模型是人类学习知识、认识世界的基本手段方法，是系统知识的抽象表示，使我们可以把精力集中在问题的主要方面——只反映系统某些选定的方面，如结构、行为（操作）和约束等特征。

什么是
数据模型
？

是可精确、抽象描述数据如何表示的一组概念集（包括描述数据结构、关系和约束等数据结构化部分），并可选地包括一组描述数据如何操纵的操作方法集。

1.1.2 数据模型分类

通常按描述**DB**结构概念的抽象层次进行分类

❖ 高级数据模型(概念数据模型)

- 所提供的概念抽象层次比较高，与领域用户理解数据的方式较接近，能更好隐藏数据存储组织和操作的细节。
- 典型代表包括：**E-R模型**、扩展**E-R模型(EER模型)**和**UML类图**等。

❖ 低级数据模型(物理数据模型)

- 所提供的概念描述数据如何在计算机上存储的具体细节。对应**DBMS**底层实现部分，没有/也不需要统一的实现
- 主要感兴趣者：**DB系统开发专家**。

❖ 逻辑数据模型

- 介于概念和物理两类数据模型之间，是**DB系统**的主要工作模型。典型代表包括关系模型、**OO模型**和**O-R数据模型**。

1.2 数据库技术的特点与发展简史



1.2.1 数据库方法特点

1.2.2 数据库技术的发展简史

1.2.1 数据库方法的技术特点

❖ 使用文件进行简单数据管理的局限：

- 数据结构、关系、冗余性处理等，都需程序员自己处理，使用和维护困难，且难以共享。

❖ 与利用**OS**文件进行简单的数据管理相比，利用**DB**进行数据管理至少具有以下优势：

- 具有更好的数据独立性，更便于数据共享
- 具有并发存取和崩溃恢复功能，能更好确保数据的完整性和安全性。
- 更有利于减小应用开发时间，提高应用的健壮性。

❖ 不适合使用**DB**的一些用例

- 只有几个严格定义的操作，关注操作存取高效性，不关心灵活查询、安全性、并发存取和崩溃恢复等性能。
- 应用可能需要以DBMS不支持的方式来查询数据。

1.2.2 数据库技术发展简史(1)

❖ 1960s年代---层次与网状数据库

- 网状数据库：1961年，通用电气公司开发了IDS（Integrated DataStore），通过网状结构（多对多关系）模拟复杂的数据关联。**网状模型**灵活但设计复杂，依赖指针管理，维护成本高。
- 层次数据库：IBM于1968年推出IMS（Information Management System），采用树形结构（父子关系）的**层次模型**存储数据，适合处理一对多关系（如组织架构），解决了数据冗余问题，但多对多关系处理复杂。

1.2.2 数据库技术发展简史(2)

❖ 1970s年代

- 1970年，IBM San Jose研究室的E.F.Codd提出了新一代的数据表示框架——**关系数据模型**，用二维表（关系）表示数据，并通过数学理论（集合论）描述数据操作。这一模型简化了数据管理，支持复杂的查询和事务处理。该成果被认为是数据库系统发展的一个分水岭。
- 数据库作为一个学科分支开始逐步走向成熟，关系DBMS也变得十分流行，其优点已被人们广泛认可。利用DBMS管理公司/组织的数据逐渐开始成为趋势。
- 1970年代末，Oracle、IBM DB2、Sybase等关系数据库系统相继推出，结合SQL（结构化查询语言），成为主流。

1.2.2 数据库技术发展简史(3)



❖ 1980s年代初、中期

- 关系数据模型成为DBMS的主流数据模型，并进一步巩固了其领域地位。
- SQL在1986年被美国国家标准信息委员会(ANSI)和国际标准化组织(ISO)采纳为关系数据库语言的标准。
- 该时期，最广泛使用的并发程序形式是可并发执行的、被称为“事务”的DB程序。

1.2.2 数据库技术发展简史(4)

❖ 从1980s后期到1990s年代初期

- DB领域在许多方面--从更强有力的查询语言到各种新型数据模型，都得到了广泛且深入的研究并取得了重要进展。
- 很多著名的DBMS开发商，都增加了对大型图像、文本等新数据类型的支持，增强了针对复杂数据集进行数据分析处理和回答更复杂查询的能力，增强了支持创建数据仓库的特性。
- 该时期另一个重要进展是关于“对象数据库系统(ODBMS)”研究和开发。
 - 目标：支持复杂数据类型（如图像、视频）和继承、多态等OO特性。
 - 但纯粹的ODBMS相关技术及产品由于没有统一标准，缺乏可移植性，故大都属于原型产品，没有发展起来。
 - 与关系模型的融合：部分数据库（如PostgreSQL）尝试将面向对象特性融入关系模型，而不是取代，形成对象-关系数据库。

1.2.2 数据库技术发展简史(5)

❖ 1993年，ODMG提出了第一个对象数据库管理标准ODMG 1.0。

- 该标准后续修订版包括ODMG 2.0(1997)和ODMG 3.0(2000)。ODMG标准出现，不仅改善对象数据库系统的可移植性，而且进一步促进了ODBMS的规范健康发展。
- 新一代的ORDBMS不仅能提供了更为丰富的类型系统，包括复杂数据类型和面向对象，而且还在SQL-99标准的框架下，扩展了传统的关系查询/关系数据操纵语言。

❖ 1990s后期出现的对象关系映射（ORM中间件）技术

- 有效解决了关系数据存储对象问题，使得关系数据库技术焕发了新春，巩固了其在数据管理领域的持久主流地位。

1.2.2 数据库技术发展简史(6)



❖ 分布式与并行数据库（**1990-2000年代**）

❖ 分布式数据库

- 互联网发展催生了分布式数据库（如Oracle RAC、MySQL Cluster），将数据存储在多个节点，提高可扩展性和容错性。典型应用包括银行跨区域事务处理。

❖ 并行数据库

- 通过多节点并行处理数据（如Teradata），提升大规模数据的查询和分析效率，适用于数据仓库场景。

1.2.2 数据库技术发展简史(7)

❖ NoSQL数据库（2000年代-2010年代）

- 随着Web 2.0和多媒体应用的兴起，传统关系数据库在海量**大数据**和**高并发**场景下面临瓶颈。NoSQL (Not Only SQL) 技术应运而生，采用非关系模型，如：
 - 键值数据库(Redis)； - 文档数据库(MongoDB)
 - 列族数据库(Cassandra)； - 图数据库(Neo4j)
- CAP定理与BASE原则
 - NoSQL数据库弱化一致性（CAP定理），优先保证可用性和分区容忍性（BASE原则），适合实时读写和弹性扩展。

1.2.2 数据库技术发展简史(8)

❖ NewSQL数据库（2010年代至今）

- 融合关系与NoSQL优势
- NewSQL数据库（如TiDB、CockroachDB）结合关系数据库的ACID特性和NoSQL的水平扩展能力，支持高并发、分布式事务。

❖ 现代趋势：AI与云原生

- **AI赋能数据库**：生成式AI推动数据库智能化，如某行的数据库安全审计系统利用机器学习检测异常行为，自然语言处理优化SQL查询效率。
- **云原生数据库**：云计算普及催生云数据库（如AWS Aurora、阿里云PolarDB），支持按需扩展、自动化运维，并集成Serverless架构，降低企业成本。

1.2.3 SQL标准及其发展简史(1)

❖ SQL标准 (Structured Query Language)

- **核心标准**：包含最基本和最常用的SQL功能，如数据查询、插入、更新和删除等操作。这些功能是所有SQL数据库管理系统共同支持的基本功能。
- **SQL扩展标准 I**：在核心标准的基础上增加了一些额外的功能和特性，如存储过程、触发器、视图、事务控制等。这些功能可以根据不同的数据库管理系统进行扩展和实现。
- **SQL扩展标准 II**：在扩展标准 I 之外的进一步其他技术扩展。

❖ SQL标准的优势

1.2.3 SQL标准及其发展简史(2)

❖ SQL标准

❖ SQL标准的优势

- 跨平台性：使得在不同的数据库管理系统之间迁移数据和操作更加容易。
- 易学易用：SQL具有相对简单的语法结构。
- 灵活性：SQL标准提供了丰富的操作和查询功能，可以进行各种复杂的数据操作。

不同的数据库管理系统（如MySQL、PostgreSQL、Oracle、SQL Server等）通常对SQL扩展标准 I，特别是标准扩展 II 的支持程度有所不同。

1.2.3 SQL标准及其发展简史(3)

❖ SQL标准

❖ SQL标准的优势

❖ SQL标准发展的重要时间线

- **SQL-86**：是SQL的最早版本，也被称为SQL1。定义了基本的数据库操作和查询语言，为后续SQL标准的发展奠定了基础。
- **SQL-92**，也称为SQL2。引入了视图（VIEW）、存储过程和触发器的概念；引入了更多的数据类型和操作符；扩展了连接（JOIN）语法及子查询/嵌套查询，引入了更多的控制和事务处理语句；定义了标准的权限管理语法。它使得关系数据库走向成熟和全面，对于数据库领域的发展具有重要的影响。

1.2.3 SQL标准及其发展简史(3)

- **SQL1999**，在92基础上，进一步增强了对OO特新的支持，
 - 比如，引入了数组、多集和多值数据类型相关的一些新数据类型和操作符，增强了SQL的表达能力和功能。增强了对大对象数据类型（LOB）的支持。
 - 并不是所有数据库产品都完全支持SQL:1999标准的所有特性。
- **SQL2003**:
 - 序列、自增字段、增强的XML支持、标量子查询
- **SQL2008**:
 - 分析函数、更多数据类型、增强的外部数据访问。
- **SQL2016**:
 - 行模式识别、JSON支持的扩展、多值插入、安全特性增强
- **SQL2019**:
 - 多值插入增强、更多JSON支持、安全特性增强。

1.3 数据库系统体系结构

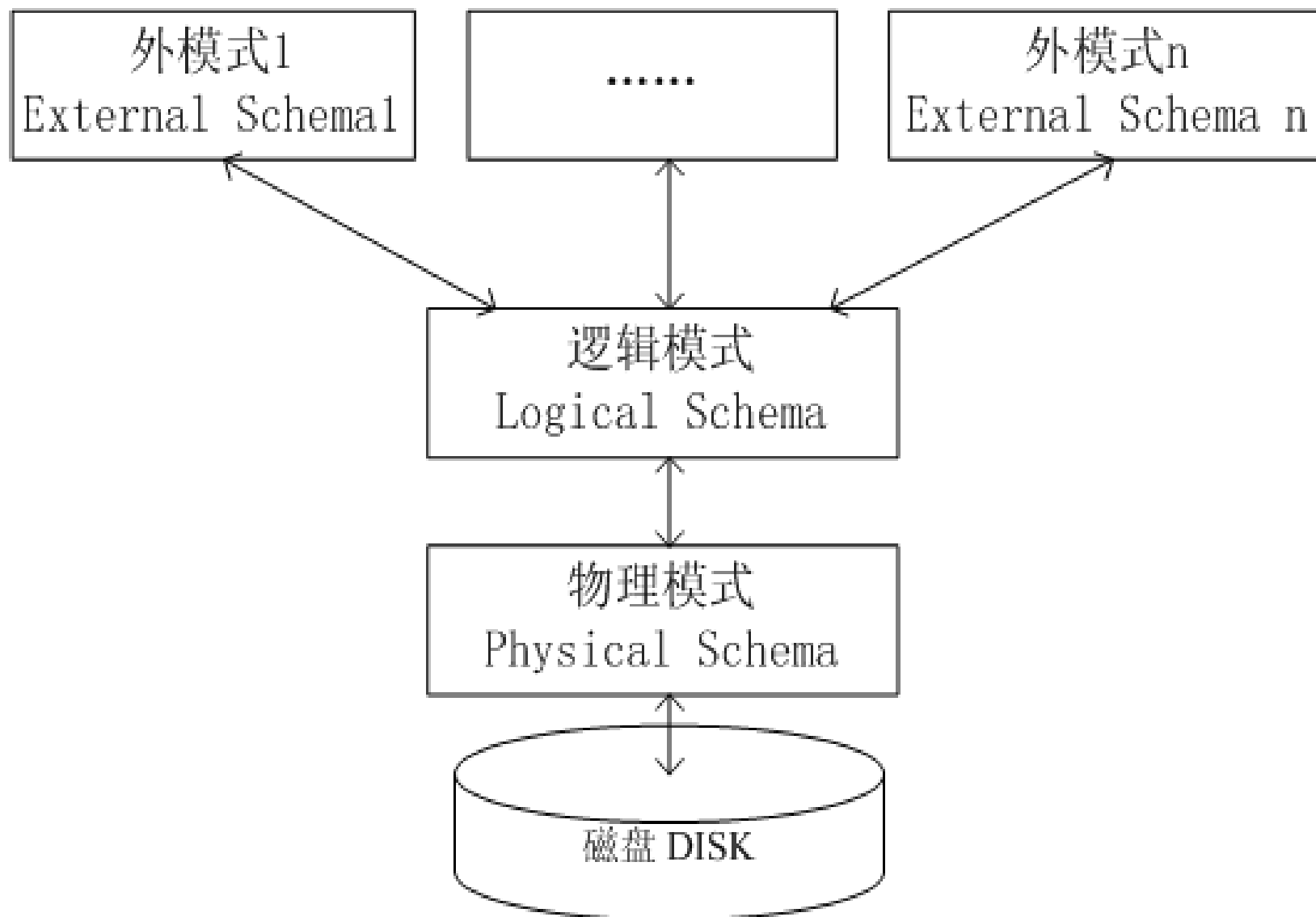


1.3.1 DB数据的三层抽象模型

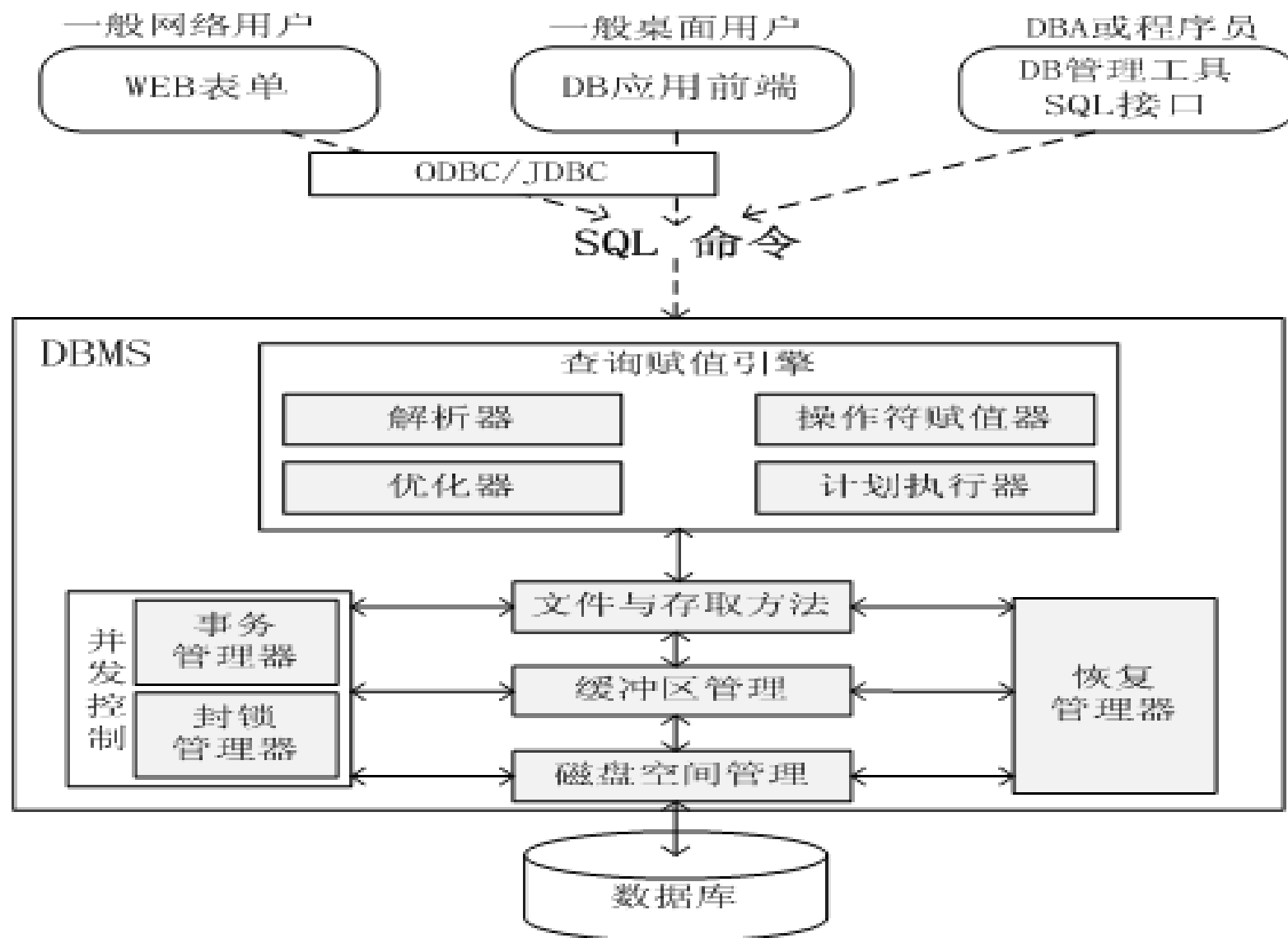
1.3.2 一种典型的DBMS实现体系结构

1.3.3 五层DBMS体系结构模型

1.3.1 DB数据的三层抽象模型



1.3.2 一种典型的DBMS实现体系结构



1.3.3 五层模型体系结构

层次	抽象层名称 (与层相关组件名称)	层中所含主要对象	辅助映射元数据
L5	非过程化或代数存取 (解释编译器、优化器)	tables, views, tuples 等数据模型概念	逻辑模式描述 (logical schema description)
L4	导航层: 面向逻辑记录的导航存取 (查询执行器、操作符赋值器)	层次结构的逻辑记录集 网络的逻辑记录集	逻辑和物理模式描述 (logical and physical schema description)
L3	记录和存取路径管理层 (供操作符赋值器调用)	逻辑记录(logical records), 存取路径(Access paths) 物理记录(Physical records)	堆文件的页链表或目录项表、 DB-键映射表(索引文件)
L2	传播控制 (缓冲区管理器)	页(pages)、DB 缓冲池、页面框 (frame)	DB 缓冲池管理的相关辅助数据结构
L1	磁盘空间管理 (磁盘空间管理器)	逻辑磁盘页(page)、OS 层次的流式文件、磁盘物理块 (blocks)	线性逻辑磁盘页序列、磁盘自由块链表, 或磁盘块使用位图表等