

数据库系统原理

陈岭

浙江大学计算机学院

- **教材:**

- Database Systems Concepts 6th edition
- By Abraham Silberschatz, Henry F. Korth and S. Sudarshan
- Higher Education Press, McGraw-Hill Companies

- **参考书目:**

- Database Management Systems 3rd edition
By Ramakrishnan and Gehrke
- Database Systems: The Complete Book
By Garcia-Molina, Ullman and Widom
- 数据库系统概论（第四版），萨师煊 王珊，高等教育出版社，2006
- 数据库系统原理教程，王珊 陈红，清华大学出版社，2003
- 数据库课程设计，陈根才 孙建伶 林怀中 周波，浙江大学出版社，2007
(实验参考书)

1

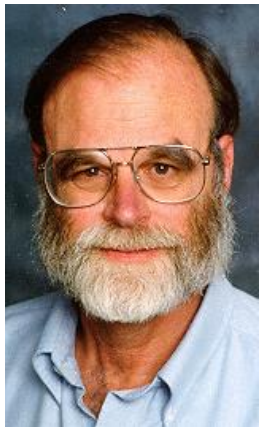
引论

- ❑ 数据库系统目的
- ❑ 数据视图
- ❑ 数据模型
- ❑ 数据库语言
- ❑ 数据库管理员
- ❑ 数据库用户
- ❑ 事务管理
- ❑ 存储管理
- ❑ 数据库体系结构

数据库系统目的

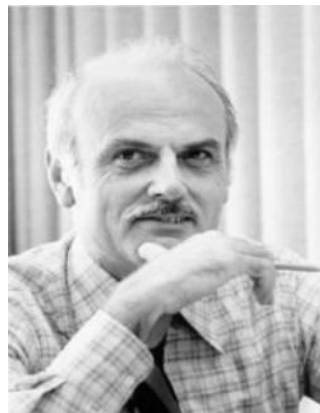
- ❑ 数据处理和管理是计算机应用最重要的领域，数据库系统的知识对于计算机学者至关重要。
- ❑ 数据库涉及社会生活的方方面面：
 - 银行业：所有交易
 - 航空公司：预订，时间表
 - 大学：注册，成绩
 - 销售：客户，产品，购买
 - 制造业：生产，库存，订单，供应链
 - 电子政务，电子商务，... ..
- ❑ 你可能学习了很多课程，但数据库系统能够让你找到一份好工作

图灵奖获得者



1998, James Gray——事务、锁、日志和二阶段提交

1981, Edgar F. Codd——关系数据库



1972, Charles W. Bachman——网状数据库

学些什么？

我们将从以下三方面学习数据库相关知识：

□ 数据库模型与设计

- 从现实生活中抽象出数据模型，再将其转换为适合目标DBMS（数据库管理系统）的形式：表、视图。

□ 编程：使用数据库

- 查询、更新数据（SQL）

□ 数据库管理系统实现

- 数据库管理系统的工作机制及设计

学生分数表的设计

学号	姓名	专业	DB平时	DB期末	DB总评成绩	OS平时	OS期末	OS总评成绩
3023001093	黄毅照	混合班	85	95	90			85
3011112340	周朝威	计算机科学与技术	80	90	85			88
3020621034	徐鑫	计算机科学与技术	90	90	90			85
3020831035	薄延嵩	计算机科学与技术	70	80	75			90
3021131123	胡俊	计算机科学与技术	70	70	70			75
3022112002	蒋永丽	计算机科学与技术	80	90	85			80
3022112003	顾娉婷	计算机科学与技术	90	90	90			85

这张表的设计好吗？为什么？

另外一种表的设计

Students

Sid	Sname	Ssex	Sage	Specialty
3023001093	黄毅照	M	21	No
3011112340	周朝威	F	20	Cs
3020621034	徐鑫	M	18	Cs
3020831035	薄延嵩	M	19	Cs
3021131123	胡俊	F	22	Cs

Courses

cid	Cname	credit
1	DB	4
2	OS	5
3	English	4
4	Math	4

Enrolled

sid	cid	grade1	grade2	grade3
3023001093	1	90		
3023001093	2	85		
3020621034	1	90		
3020831035	1	75		
3021131123	2	75		

这种表的设计好吗？

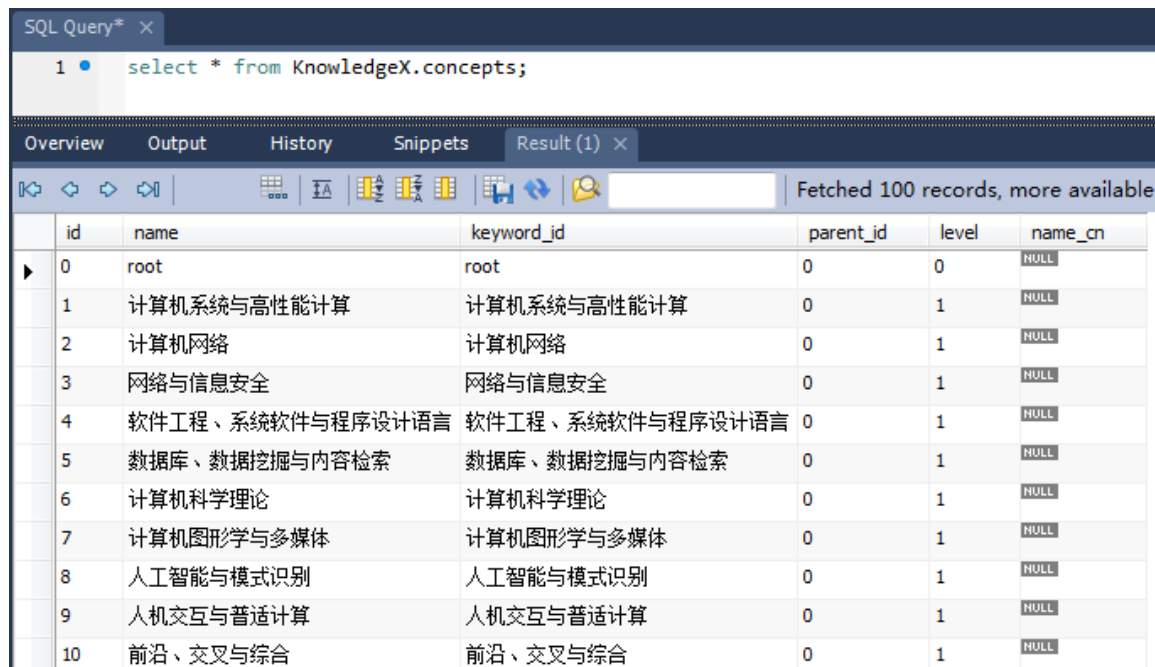
为什么？



数据库访问

方法1：利用数据库管理系统提供的交互工具访问数据库

如：MySQL WorkBench, SQL Server查询分析器, ORACLE Sql*Plus, Work Sheet



The screenshot shows a SQL query tool interface. At the top, a query window titled 'SQL Query*' contains the query: `1 • select * from KnowledgeX.concepts;`. Below the query window, a tabbed interface shows 'Result (1)' selected. The result is displayed as a table with 10 records. The table has columns: id, name, keyword_id, parent_id, level, and name_cn. The first record is the root node. The subsequent records are children of the root, each with a parent_id of 0. The table is styled with alternating light and dark gray rows. A status bar at the bottom right indicates 'Fetched 100 records, more available'.

id	name	keyword_id	parent_id	level	name_cn
0	root	root	0	0	NULL
1	计算机系统与高性能计算	计算机系统与高性能计算	0	1	NULL
2	计算机网络	计算机网络	0	1	NULL
3	网络与信息安全	网络与信息安全	0	1	NULL
4	软件工程、系统软件与程序设计语言	软件工程、系统软件与程序设计语言	0	1	NULL
5	数据库、数据挖掘与内容检索	数据库、数据挖掘与内容检索	0	1	NULL
6	计算机科学理论	计算机科学理论	0	1	NULL
7	计算机图形学与多媒体	计算机图形学与多媒体	0	1	NULL
8	人工智能与模式识别	人工智能与模式识别	0	1	NULL
9	人机交互与普适计算	人机交互与普适计算	0	1	NULL
10	前沿、交叉与综合	前沿、交叉与综合	0	1	NULL

方法2：利用开发工具设计界面、处理数据，调用ODBC等接口访问数据库，
如：ASP，JSP，VC++，PHP，PowerBuilder，Delphi



浙江大學

专业培养计划查询

必修课教学计划

院系选修课

限定性选修课

辅修课教学计划

学院

计算机科学与技术学院

 专业

计算机科学与技术

 年级

2002

 学期

全部

课程代码	课程名称	学分	周学时	考核方式	课程性质
02110010	思想道德修养	2.0	1.0-2.0	考查	必修课
02110020	法律基础	1.5	1.0-1.0	考查	必修课
02110032	毛泽东思想概论（乙）	1.5	1.0-1.0	考试	必修课
03110030	体育 I	1.0	0.0-2.0	考查	必修课
05110010	大学英语 I	3.0	2.0-2.0	考试	必修课
06110042	微积分（甲）I	4.5	4.0-1.0	考试	必修课
06110091	线性代数（甲）	3.0	3.0-0.0	考试	必修课
08110012	工程图学（乙）	2.5	2.0-1.0	考试	必修课
31110010	计算机文化	0.5	0.0-1.0		必修课
03110010	军事理论	1.5	1.0-1.0	考查	必修课
03110040	体育 II	1.0	0.0-2.0	考查	必修课
05110020	大学英语 II	3.0	2.0-2.0	考试	必修课
06110052	微积分（甲）II	4.5	4.0-1.0	考试	必修课
06110200	离散数学	4.0	4.0-0.0	考试	必修课

□ 数据库（DB）

- 与企业相关的数据集合
- 具有完整性和持久性的数据集合。[R. Ramakrishnan, J. Gehrhe]
- 长期（常常多年）存在的信息集合。[Ullman]
- 长期存储在计算机内，有组织的，可共享的数据集合。[萨师煊，王珊]

□ 数据库管理系统（DBMS）

- 数据库 + 一组用以访问、更新和管理这些数据的程序

DBMS的主要特性

- ❑ 数据访问的高效和可扩展性
- ❑ 缩短应用开发时间
- ❑ 数据独立性（物理数据独立性 / 逻辑数据独立性）
- ❑ 数据完整性和安全性
- ❑ 并发访问和鲁棒性（恢复）

DBMS的发展历史

- ❑ File processing system (1950s–1960s)
- ❑ Network and hierarchical DBMS (1960s–1970s)
 - 网状数据模型、层次数据模型 – 网状数据库、层次数据库（结构复杂、使用很困难）
- ❑ Relational database systems (RDBMS)
 - 关系模型 (1970, E. F. Codd)
 - RDBMS开始发展 (1970s)
 - RDBMS走向市场 (1980s)
 - RDBMS技术成熟 (1990s)

DBMS的发展历史

- ❑ 面向对象数据库系统: Object-oriented database system (OODBMS)
- ❑ 对象关系数据库系统: Object-relational database systems (ORDBMS)
- ❑ 面向应用数据库系统: Application-oriented database systems
 - 空间、时间、多媒体、网络数据库
- ❑ 数据仓库 (Data Warehousing)、联机分析处理 (Online Analytical Processing)、数据挖掘 (Data Mining)

□ 文件处理系统由传统操作系统所支持：

- 随着需求的增长，需要编写新的应用程序，并创建新的数据文件
- 但在相当长的时间内，数据文件可以是不同的格式。数据文件是相互独立的

□ 在文件处理系统中存储组织信息的主要弊端：

- 数据冗余和不一致
 - 多种文件格式、信息重复存储
- 数据访问困难
 - 需要编写一个新的程序来完成每一个新的任务
- 数据孤立
 - 多文件多格式，检索、共享数据困难

□ 在文件处理系统中存储组织信息的主要弊端：

■ 完整性问题

- 完整性约束（如账户余额 >0 ）成为程序代码的一部分
- 增加新的约束或更改现有的约束很困难

■ 原子性问题

- 在进行部分数据更新时，一旦发生故障，可能导致数据库处于不一致的状态
- 例如，从一个账户转移资金到另一个账户，此操作要么完成，要么根本不会发生

■ 并发访问异常

- 为了提高系统的总体性能，许多系统允许并发访问
- 不受控制的并发访问可能导致数据不一致
- 例如，两个用户读取同一账号余额，并在同一时间更新它

Database systems VS File Processing Systems

- 在文件处理系统中存储组织信息的主要弊端：
 - 安全性问题
 - 并非所有用户都可以访问所有数据
- 数据库系统为以上所有问题，提供了解决方案

- ❑ 关系数据库管理系统的公司：
 - 甲骨文（Oracle）、SAP（Sybase）：最大的数据库软件公司之一
 - IBM（DB2）：世界上最大的DBMS供应商之一
 - 微软的SQL-Server以及Access：精简、相对便宜
- ❑ 关系数据库公司也面临“面向对象DB”公司的挑战
- ❑ “对象 - 关系”系统，保留了核心的关系模型，同时允许类型扩展为面向对象的系统
- ❑ 其他数据库产品：Ingres, Paradox, Foxbase, FoxPro, dBase, ...

□ 开源数据库系统：

■ MySQL：是网站上小型系统最流行的开源数据库

- MySQL是LAMP的重要组成部分（Linux, Apache, MySQL, PHP/ Perl/ Python），一个快速增长的开源企业软件堆栈
- <http://www.mysql.com>

■ PostgreSQL：是一个高度可扩展的，开放源码的对象关系型数据库管理系统。

- 最初由加州大学伯克利分校计算机系开发的“Postgres”
- <http://www.postgresql.org>

□ 如何使用数据库系统：不同的用法需要不同层次的抽象（如，学生成绩管理系统）

- 物理层：描述数据实际上是怎样存储的
- 逻辑层：描述数据库中存储什么数据及这些数据间存在什么关系

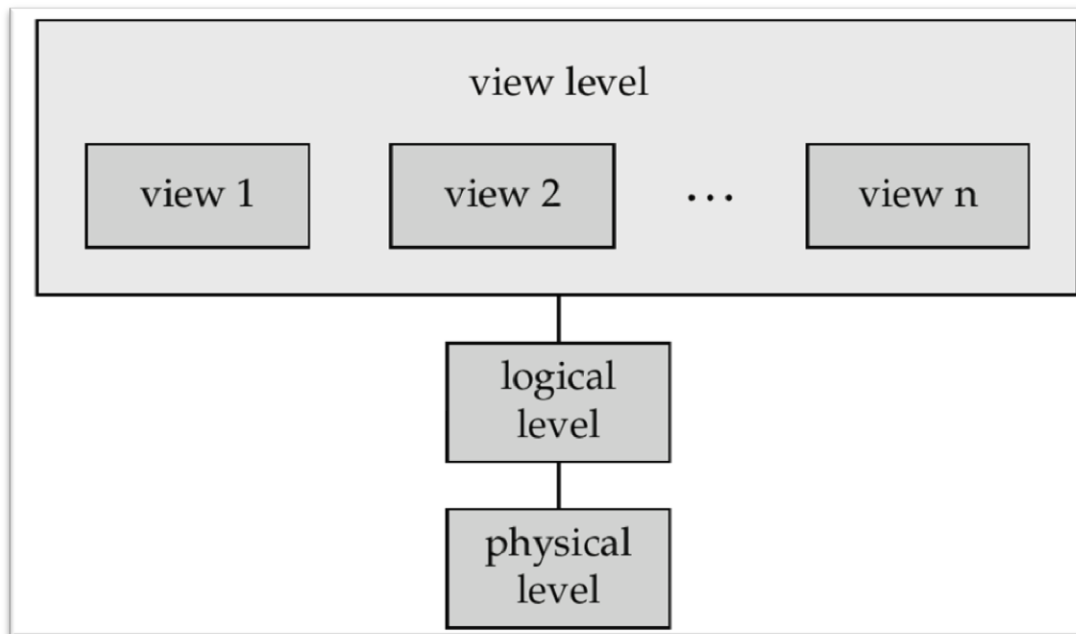
– 如, type instructor = record

```
ID : char (5);  
name : char (20);  
dept_name : char (20);  
salary : numeric (8, 2);
```

end;

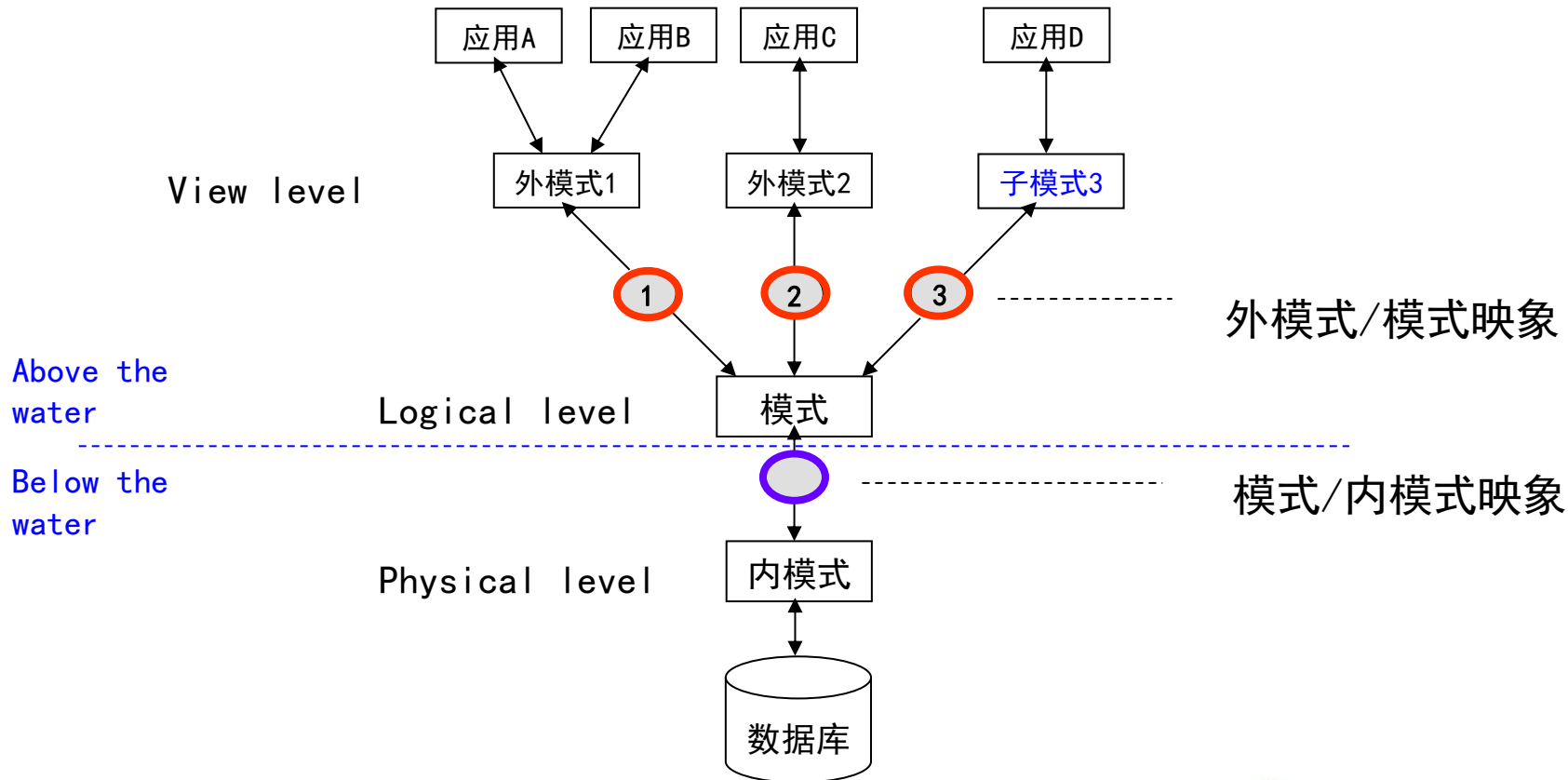
- 视图层：应用程序能够隐藏数据类型的详细信息。视图也可以出于安全目的隐藏数据信息（例如，员工的薪水）

□ 数据抽象的三层结构:



- 类似编程语言中的类型 (types) 和变量 (variables)
 - 类型 \leftrightarrow 模式, 变量 \leftrightarrow 实例
- 模式 (Schema): 数据库的总体设计
 - 类似于程序中变量的类型信息
 - 物理模式: 在物理层描述数据库的设计
 - 逻辑模式: 在逻辑层描述数据库的设计
- 实例 (Instance): 特定时刻存储在数据库中的信息的集合
 - 类似于程序中变量的值

数据库系统的模式结构



物理独立性和逻辑独立性

- ❑ 修改一层的结构定义不影响更高层的结构定义
- ❑ 物理数据独立性：修改物理结构而不需要改变逻辑结构的能力
 - 应用程序依赖于逻辑结构
 - 应用程序独立于数据的结构和存储
 - 这是使用DBMS最重要的好处
- ❑ 逻辑数据独立性：数据逻辑结构的改变不影响应用程序
 - 逻辑数据独立性一般难以实现，因为应用程序严重依赖于数据的逻辑结构

数据模型

□ 数据模型是一个概念工具的集合，用于描述：

- 数据结构
- 数据关系
- 数据语义
- 数据约束

□ 数据抽象的不同层次需要不同的数据模型来描述：

- 实体 - 关系模型
- 关系模型
- 其他模型：
 - 面向对象模型
 - 半结构化数据模型（XML）
 - 早期模型：网状模型和层次模型 ...

数据库设计过程：

- 需求分析
- 概念设计
- 逻辑设计
-



数据库设计步骤

1. 需求分析

- 需要什么样的数据、应用程序和业务

2. 概念数据库设计

- 使用 E-R 模型或类似的高层次数据模型，描述数据

3. 逻辑数据库设计

- 将概念设计转换为某个DBMS所支持的数据模型

4. 结构优化

- 关系标准化，检查冗余和相关的异常关系结构

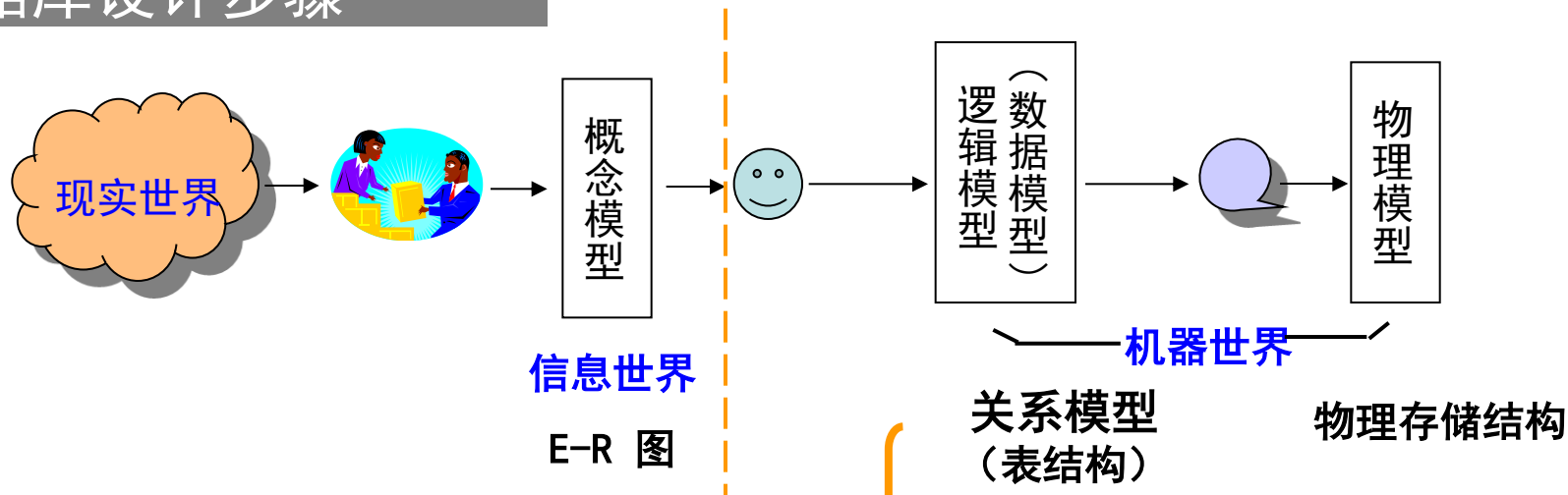
5. 物理数据库设计

- 索引，集群和数据库调优

6. 创建并初始化数据库&安全设计

- 加载初始数据，测试
- 识别不同的用户及他们的角色

数据库设计步骤



信息世界
E-R 图

机器世界

关系模型
(表结构)

物理存储结构

面向对象模型

--非关系模型--

层次模型

网状模型

DBTG模型 | CODASYL系统

网状结构 (一个结点可有多个双亲, 允许多个结点无双亲)

Honeywell IDS/2, HP image

1968, IBM IMS

树型结构

(除根结点外, 每个结点有且只有一个双亲结点)



实体 – 联系模型

□ E-R(Entity - Relationship) 模型

□ 实体（对象）

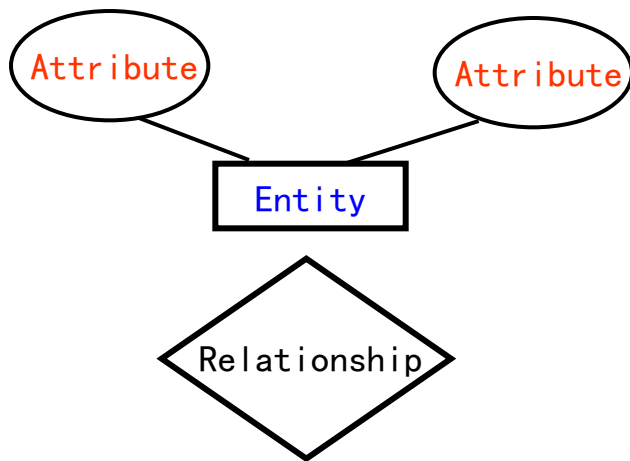
- 例如，客户、帐户、银行分支机构
- 实体由属性描述

□ 联系：是几个实体之间的关联

- 例如，帐号 A-101 是由客户 Johnson 拥有，联系设定存款关联客户的帐户

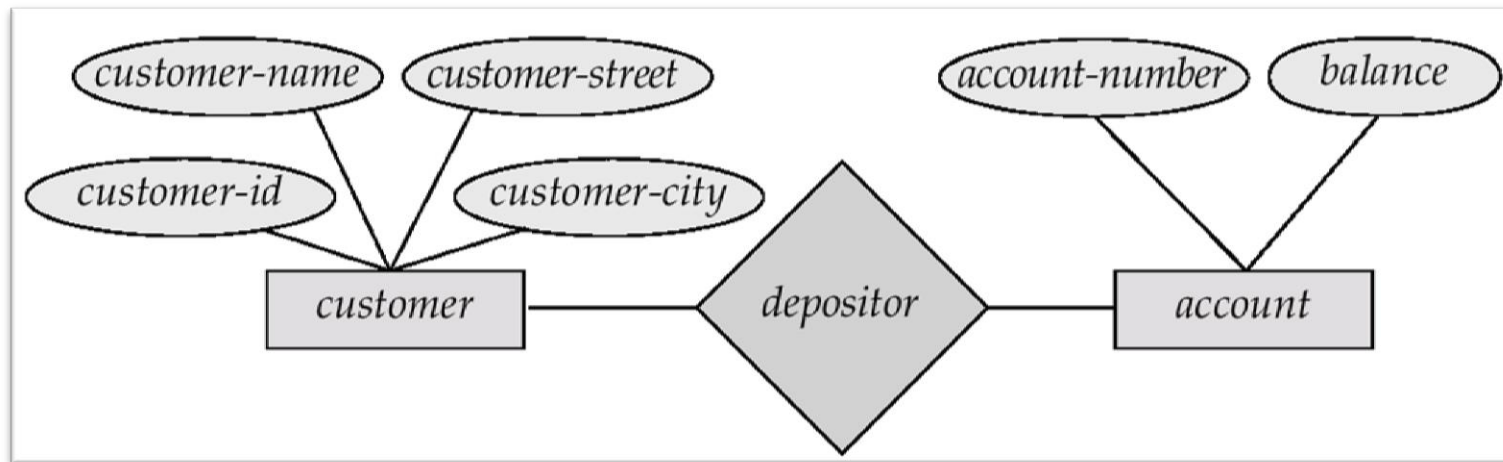
□ E-R模型数据库设计中使用广泛

- ER模型通常将数据库设计转化为关系模型的设计
- 最早由Peter Chen提出



实体 – 联系模型

□ 实体-联系模型示例：



关系模型

- ❑ 将E-R图转换为关系模式
- ❑ 在关系模型中，表格数据的示例：

Customer:

<i>customer-id</i>	<i>customer-name</i>	<i>customer-street</i>	<i>customer-city</i>	<i>account-number</i>
192-83-7465	Johnson	Alma	Palo Alto	A-101
019-28-3746	Smith	North	Rye	A-215
192-83-7465	Johnson	Alma	Palo Alto	A-201
321-12-3123	Jones	Main	Harrison	A-217
019-28-3746	Smith	North	Rye	A-201

Attributes

schema

Tuple
元组

关系数据库示例:

<i>customer-id</i>	<i>customer-name</i>	<i>customer-street</i>	<i>customer-city</i>
192-83-7465	Johnson	12 Alma St.	Palo Alto
019-28-3746	Smith	4 North St.	Rye
677-89-9011	Hayes	3 Main St.	Harrison
182-73-6091	Turner	123 Putnam Ave.	Stamford
321-12-3123	Jones	100 Main St.	Harrison
336-66-9999	Lindsay	175 Park Ave.	Pittsfield
019-28-3746	Smith	72 North St.	Rye

(a) The *customer* table

<i>account-number</i>	<i>balance</i>
A-101	500
A-215	700
A-102	400
A-305	350
A-201	900
A-217	750
A-222	700

(b) The *account* table

<i>customer-id</i>	<i>account-number</i>
192-83-7465	A-101
192-83-7465	A-201
019-28-3746	A-215
677-89-9011	A-102
182-73-6091	A-305
321-12-3123	A-217
336-66-9999	A-222
019-28-3746	A-201

(c) The *depositor* table



关系数据库示例：University Database

Students

Sid	Sname	Ssex	Sage	sdept
3023001093	Tom	M	21	Cs
3011112340	Mary	F	20	Cs
3020621034	Jack	M	18	Cs
3020831035	Smith	M	19	Ma
3021131123	Alane	F	22	Is

Course

cid	Cname	credit
1	DB	4
2	OS	5
3	English	4
4	Math	4

Enrolled

sid	cid	grade
3023001093	1	92
3023001093	2	88
3020621034	1	70
3020831035	1	85
3021131123	2	95



关系模型

成绩登记表

Sid	Sname	Cname	credit	grade
3023001093	Tom	DB	4	92
3023001093	Tom	OS	5	88
3020621034	Jack	DB	4	70
3020831035	Smith	DB	4	85
3021131123	Alane	OS	5	95

view level
(子模式)

Logical level
(模式)

Sid	Sname	Ssex	Sage	sdept
3023001093	Tom	M	21	Cs
3011112340	Mary	F	20	Cs
3020621034	Jack	M	18	Cs
3020831035	Smith	M	19	Ma
3021131123	Alane	F	22	Is

Student

Course

cid	Cname	credit
1	DB	4
2	OS	5
3	English	4
4	Math	4

sid	cid	grade
3023001093	1	92
3023001093	2	88
3020621034	1	70
3020831035	1	85
3021131123	2	95

Enrolled



□ 数据库语言：

- Data Definition Language (DDL, 数据定义语言)
- Data Manipulation Language (DML, 数据操纵语言)
- Data Control Language (DCL, 数据控制语言)

□ 数据定义语言 (DDL)

- 指定一个数据库模式作为一组关系模式的定义
- 指定存储结构, 访问方法和一致性约束
- DDL语句经过编译, 得到一组存储在一个特殊文件中的表, 特殊文件即数据字典(data dictionary), 其中包含元数据(metadata)
- 例如, `CREATE TABLE account (account_number char(10),
balance integer);`

该SQL语句创建了表account



□ 数据定义语言 (DDL)

■ 数据字典 (data dictionary) 包含元数据 (metadata), 包括:

- 数据库模式
- 数据存储结构
- 访问方法和约束
- 统计信息
- 授权

2. 数据操纵语言 (DML)

- 从数据库中检索数据
- 插入/删除/更新数据
- DML 也称为查询语言

□ 数据操纵语言 (DML)

■ 两类基本的数据操作语言：

- **过程化DML**：要求用户指定需要什么数据，以及如何获得这些数据（C, Pascal, Java, ...）
- **声明式DML**：也称为**非过程化DML**，只要求用户指定需要什么数据，而不指明如何获得这些数据（SQL, Prolog）

3. SQL

■ SQL = DDL + DML + DCL

■ SQL已被广泛使用

– SQL (Structured Query Language, 结构化查询语言), 来源于1975年IBM System R中的“SEQUEL” (Structured English QUery Language)。

– 例1, 根据用户的 customer-id (192-83-7465) 找到用户:

```
SELECT customer-name  
FROM customer  
WHERE customer-id = '192-83-7465'
```

– 例2, 找到客户(192-83-7465)持有的所有账户的余额:

```
SELECT account.balance  
FROM depositor, account  
WHERE depositor.customer-id = '192-83-7465' and  
depositor.account-number = account.account-number
```



3. SQL

■ SQL查询示例:

<i>customer-id</i>	<i>customer-name</i>	<i>customer-street</i>	<i>customer-city</i>
192-83-7465	Johnson	12 Alma St.	Palo Alto
019-28-3746	Smith	4 North St.	Rye
677-89-9011	Hayes	3 Main St.	Harrison
182-73-6091	Turner	123 Putnam Ave.	Stamford
321-12-3123	Jones	100 Main St.	Harrison
336-66-9999	Lindsay	175 Park Ave.	Pittsfield
019-28-3746	Smith	72 North St.	Rye

(a) The *customer* table

<i>account-number</i>	<i>balance</i>
A-101	500
A-215	700
A-102	400
A-305	350
A-201	900
A-217	750
A-222	700

(b) The *account* table

<i>customer-id</i>	<i>account-number</i>
192-83-7465	A-101
192-83-7465	A-201
019-28-3746	A-215
677-89-9011	A-102
182-73-6091	A-305
321-12-3123	A-217
336-66-9999	A-222
019-28-3746	A-201

(c) The *depositor* table



3. SQL

■ SQL是使用最广泛的查询语言。有三种用法：

- 直接在交互环境中使用：

 - SQL Server：查询分析器

 - Oracle：SQL*Plus、Work Sheet

 - MySQL：命令行客户端

- 在宿主语言中，通过ODBC（开放式数据库连接）、JDBC使用

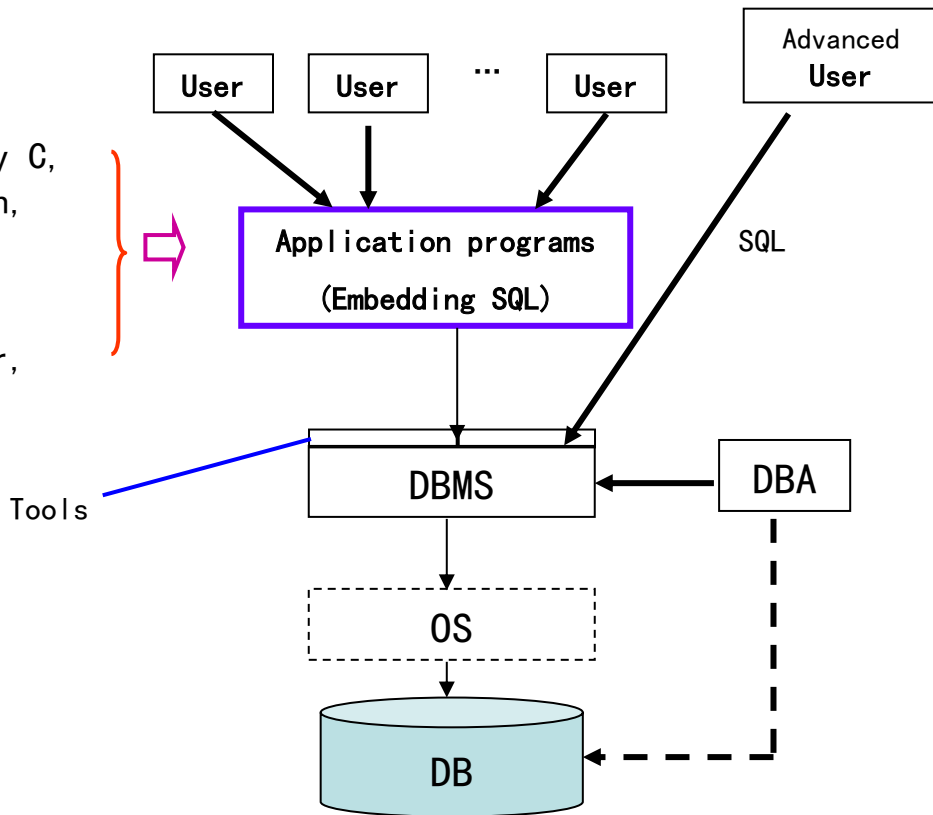
- 在宿主语言中使用嵌入式SQL



4. 数据库使用

Developed by C,
C++, Fortran,
Cobol, Java

Delphi, VB,
PowerBuilder,



根据所期望的与系统交互方式的不同，数据库系统的用户可以分为四类：

- **无经验的用户：**他们通过激活事先已经写好的应用程序同系统进行交互（普通用户）
 - 例如，人们通过网络、银行出纳员、文员访问数据库
- **应用程序员：**通过SQL调用与系统进行交互
- **富有经验的用户：**用数据库查询语言或数据分析软件等工具来表达他们的要求。例如，联机分析处理（OLAP）、数据挖掘。
- **特殊用户：**编写专门的，不适合于传统数据处理框架的数据库应用。例如计算机辅助设计系统（CAD）、知识库系统（KDB），专家系统（ES）。

- 数据库管理员（DBA）：对数据库系统进行集中控制的特殊用户
 - DBA拥有管理数据库的最高权限
 - DBA协调数据库系统的所有活动
 - DBA控制所有用户访问数据库的权限
 - DBA对企业的信息资源和需求有很好的理解
- 数据库管理员的工作包括：
 - 模式定义
 - 存储结构与存取方法定义
 - 模式及物理组织的修改
 - 数据访问授权
 - 日常维护
 - 监视数据库的运行，确保数据库的性能
 - 数据库安全（如，定期备份数据库，数据库恢复）

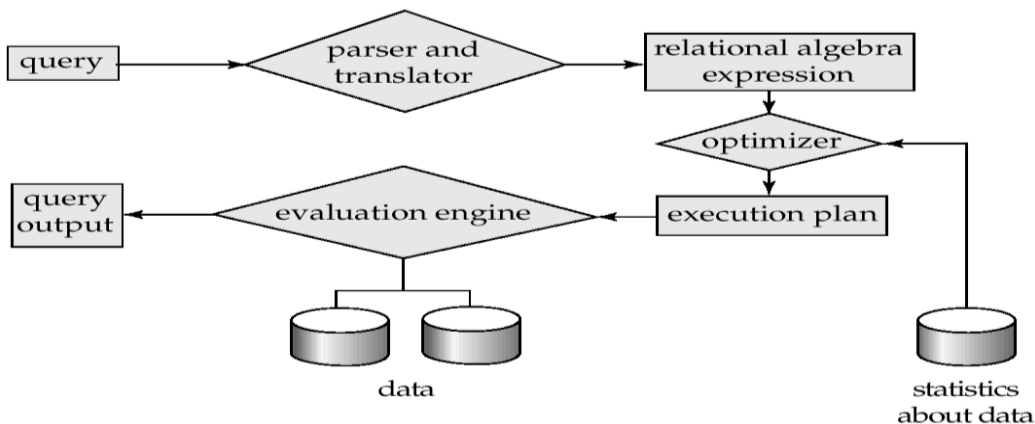
- ❑ 并发的使用很重要，但也会带来一些问题
- ❑ 事务：是在数据库应用中完成单一逻辑功能的操作集合
- ❑ 事务的要求：Atomicity（原子性），Consistence（一致性），Isolation（隔离性），Durability（持久性） / ACID
- ❑ 事务管理组件：确保系统在出现故障（例如断电或操作系统宕机），或事务失败的情况下，数据库都能保持一致性（正确性）
- ❑ 并发控制管理器：控制并发事务之间的交互

□ 存储管理器

- 在底层数据存储与应用程序及查询之间，提供接口
- 对数据库中的数据进行高效存储，检索与更新
- 包括：
 - 事务管理
 - 授权和完整性管理
 - 文件管理（管理文件系统与数据文件，数据字典，索引文件之间的交互）
 - 缓存管理

□ 查询处理器

- 接收数据库语言输入，经过解析、优化、执行，输出相应结果给用户
- 包括：
 - 解析和翻译
 - 优化
 - 执行



□ 查询处理器

■ 查询处理优化

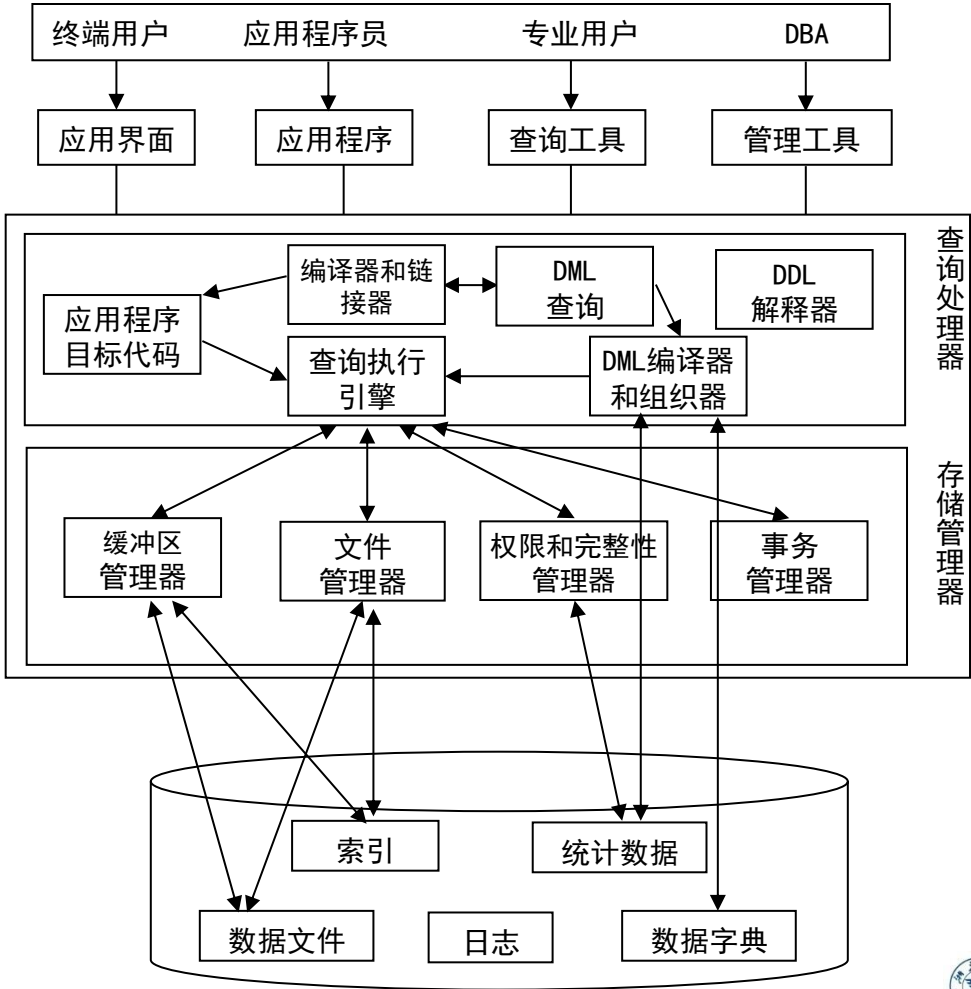
- 执行给定查询操作的方法：
 - 等价表达式
 - 每个操作有不同的实现算法

■ 不同执行方法之间的开销差可能是巨大的

■ 需要预估操作的开销

- 关键取决于数据库中所维持关系的统计信息
- 需要预估中间结果的统计信息，这些统计信息将用于计算复杂表达式的开销

数据库体系结构



用户

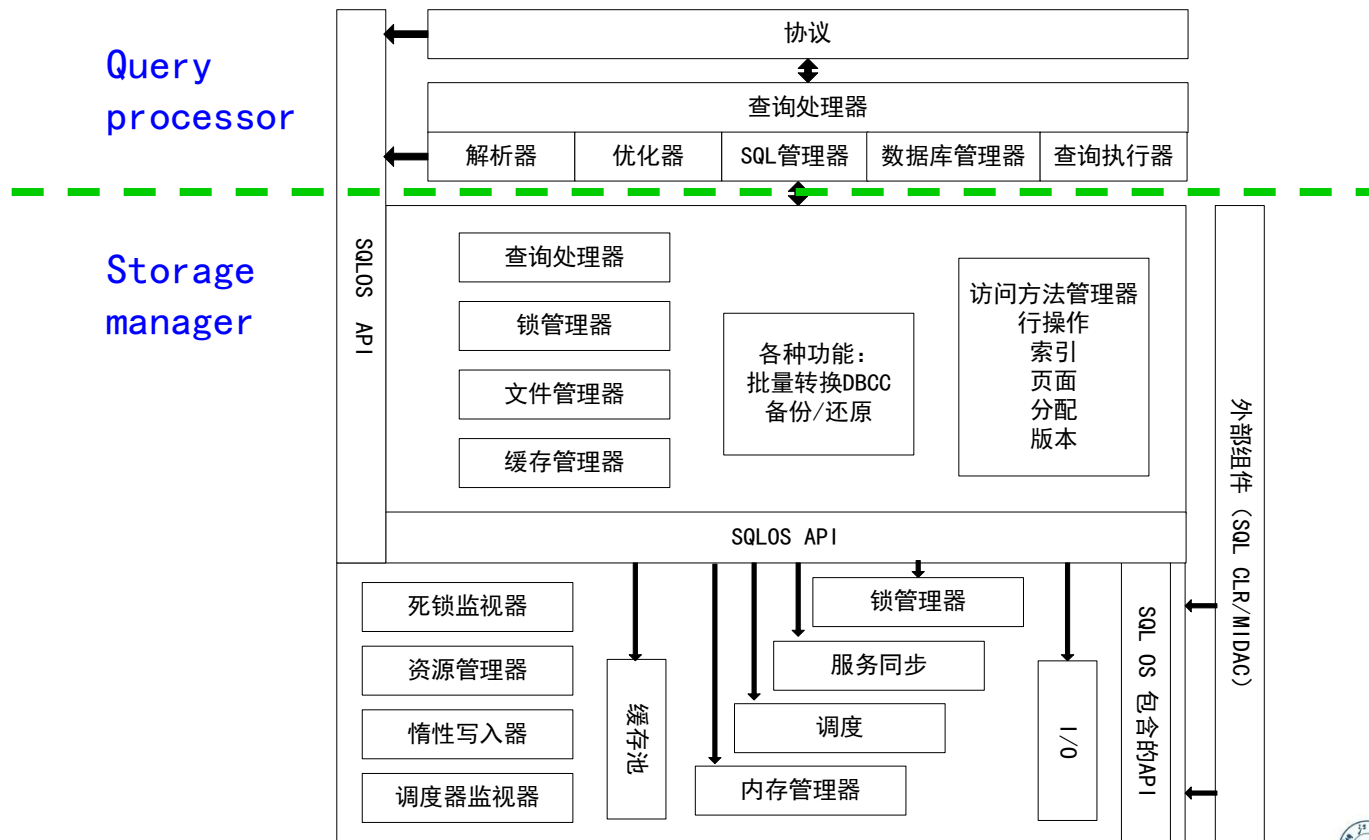
界面

(DBMS) 数据库管理系统

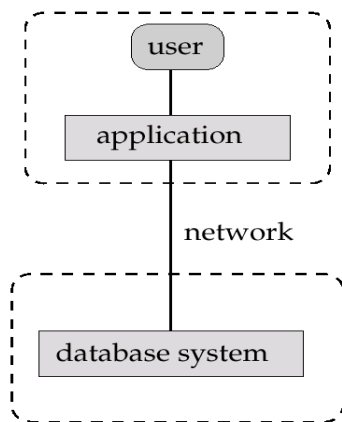
磁盘存储



SQL Server 体系结构



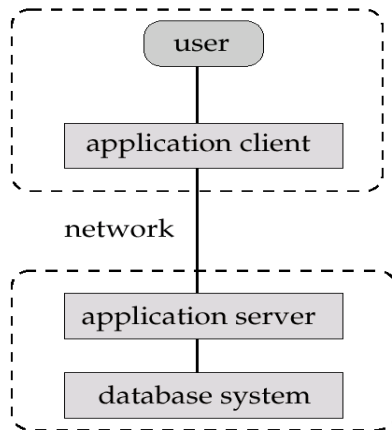
应用程序体系结构



a. two-tier architecture

client

server



b. three-tier architecture

- ❑ **两层体系结构：**像ODBC和JDBC这样的应用程序接口标准被用于进行客户端和服务器的交互
- ❑ **三层体系结构：**如基于web的应用程序及采用“中间件”构建的应用程序

- ❑ 数据库管理系统用于维护和查询大量的数据集
- ❑ 拥有故障恢复、并发访问、快速应用开发以及数据集成和数据安全的优点
- ❑ 抽象使得数据具有独立性
- ❑ E-R模型，关系模型
- ❑ DDL, DML, SQL
- ❑ 数据库管理员的职责
- ❑ DBS经典体系结构
- ❑ DBMS R&D是计算机科学领域一个最广泛，最令人兴奋的领域