

□ 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

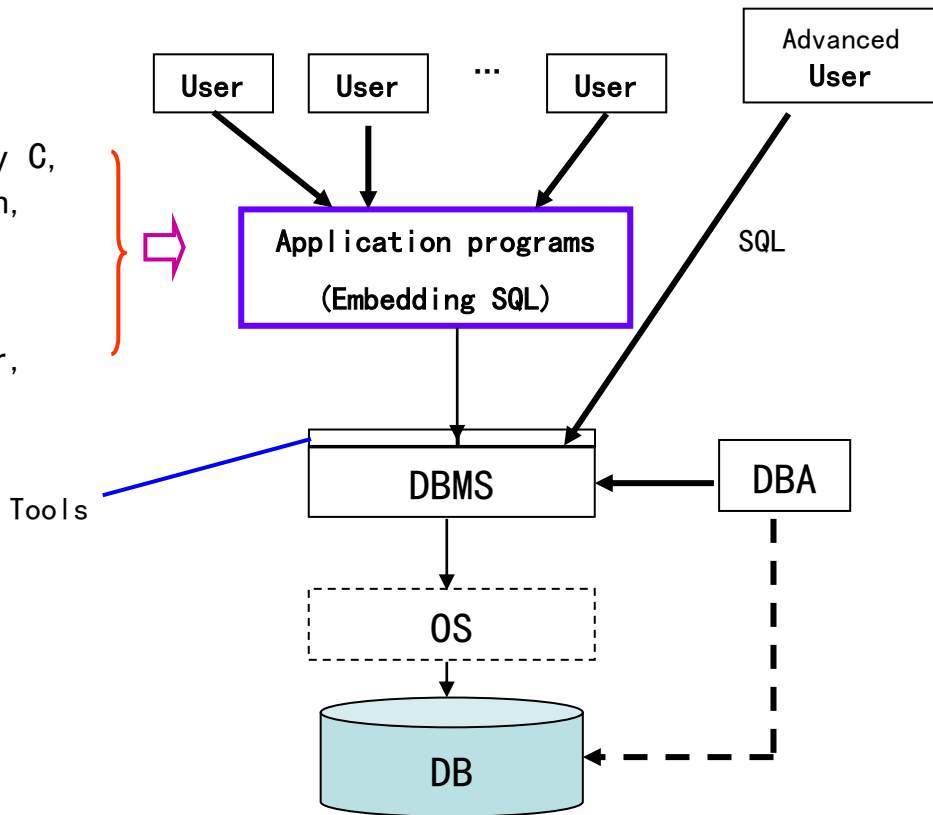
- 在宿主语言中使用嵌入式SQL



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）

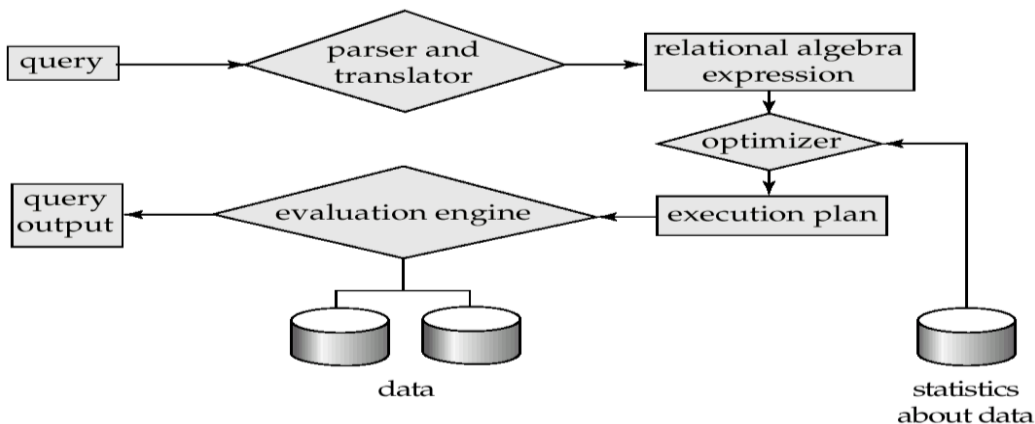
- ❑ 并发的使用很重要，但也会带来一些问题
- ❑ 事务：是在数据库应用中完成单一逻辑功能的操作集合
- ❑ 事务的要求：Atomicity（原子性），Consistence（一致性），Isolation（隔离性），Durability（持久性） / ACID
- ❑ 事务管理组件：确保系统在出现故障（例如断电或操作系统宕机），或事务失败的情况下，数据库都能保持一致性（正确性）
- ❑ 并发控制管理器：控制并发事务之间的交互

□ 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

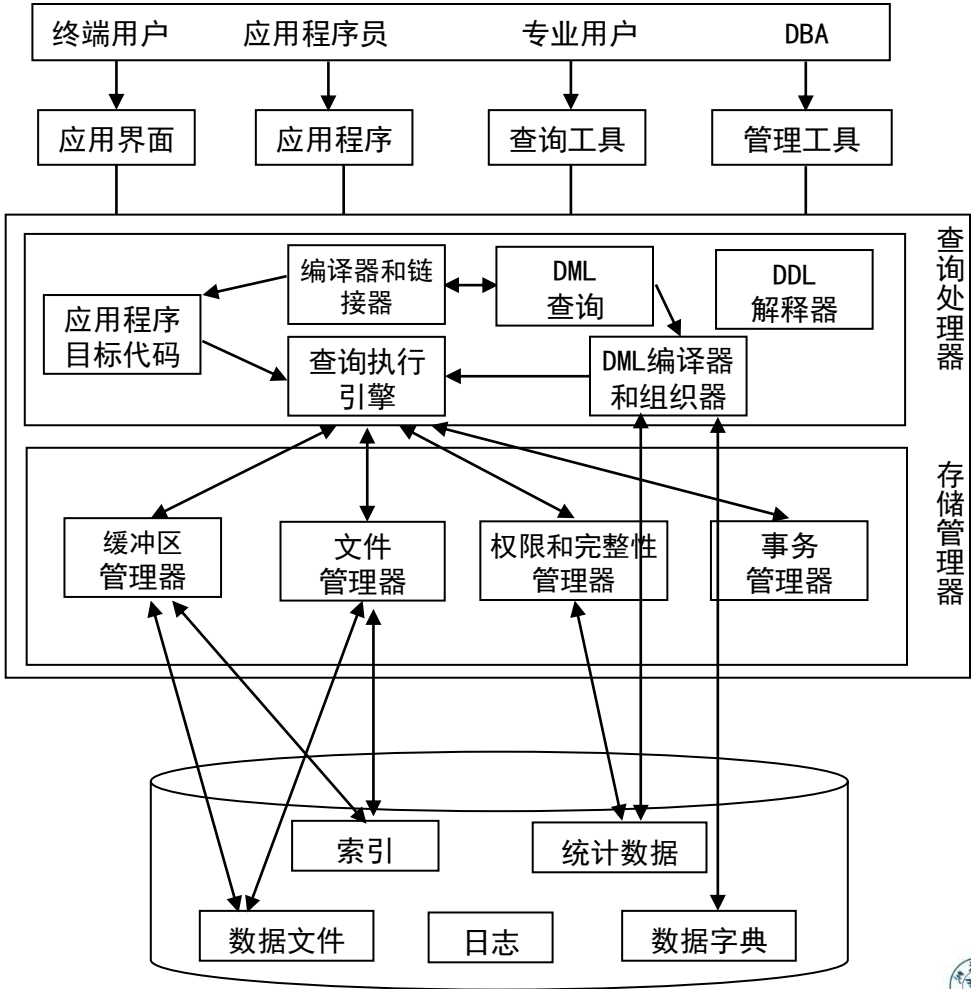
- 执行给定查询操作的方法：
 - 等价表达式
 - 每个操作有不同的实现算法

■ 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

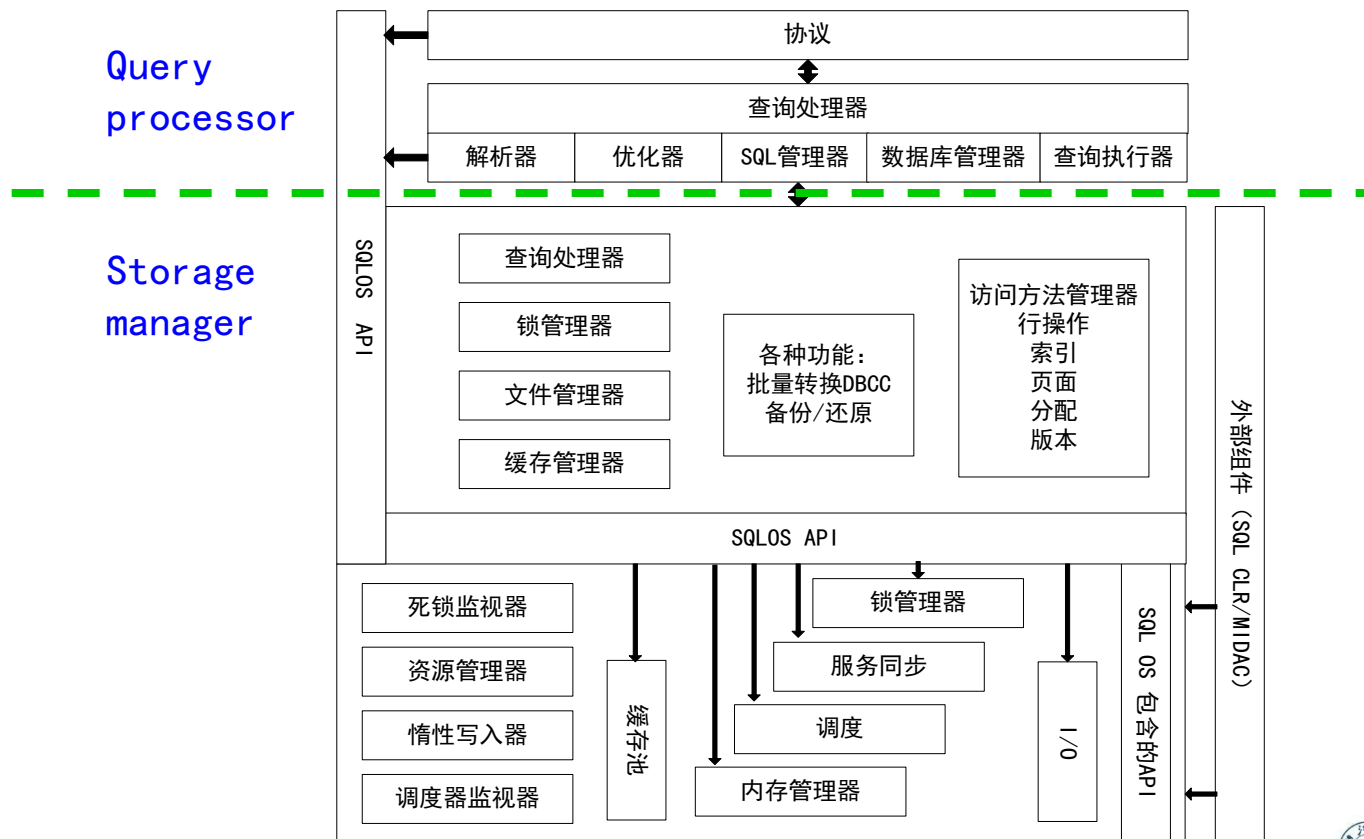
界面

(DBMS) 数据库管理系统

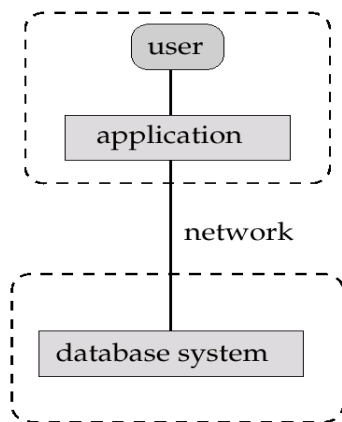
磁盘存储



SQL Server 体系结构



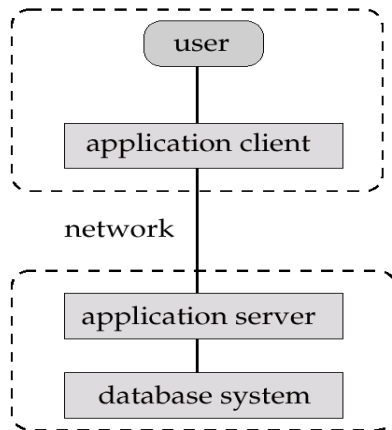
应用程序体系结构



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