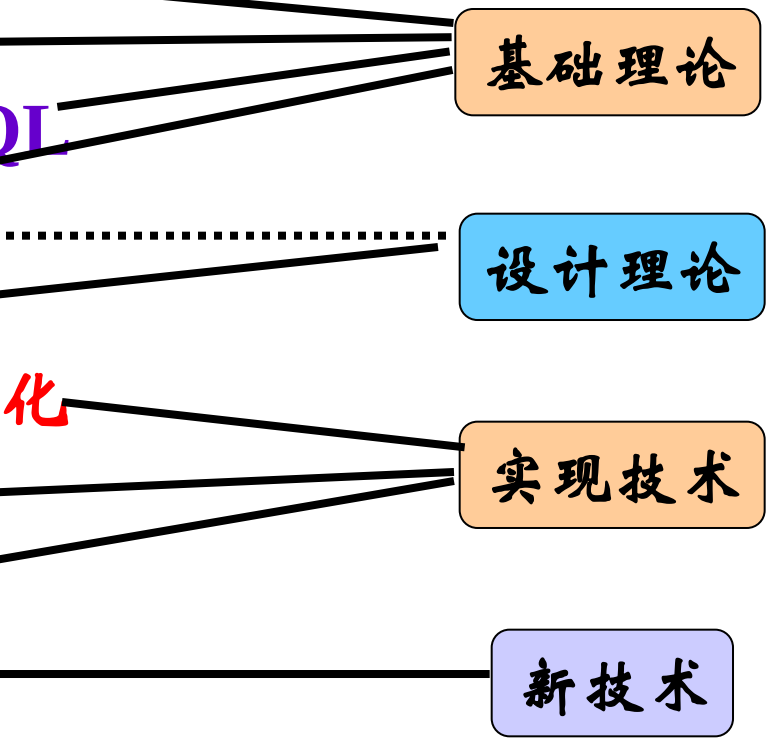
课程内容

- 数据库系统基本概念（数据模型，体系结构）
- 关系数据库
- 关系数据库标准语言SQL
- 关系数据理论
- 数据库设计
- 关系查询处理和查询优化
- 事务处理技术
- 数据库保护
- 数据库技术新发展



DBMS体系结构（1）

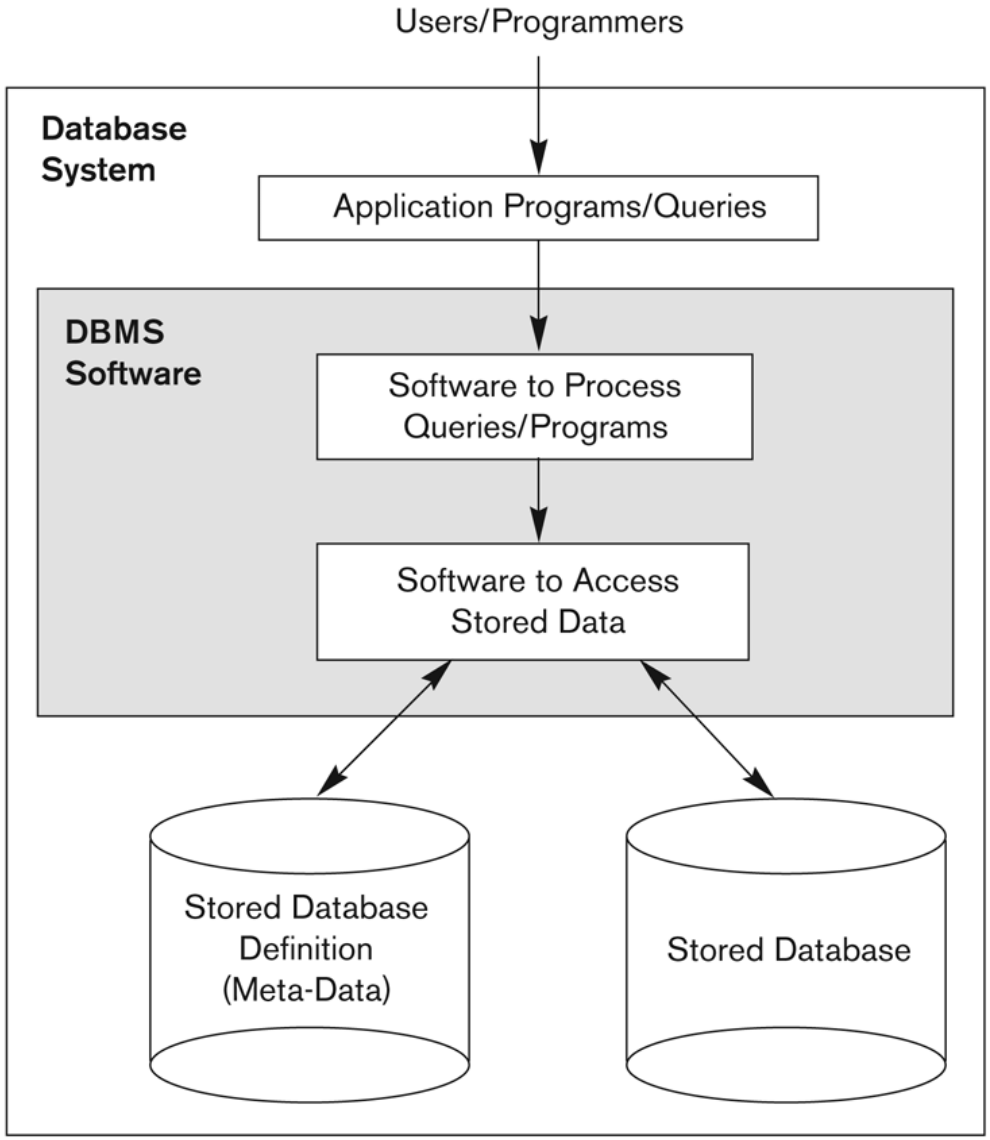
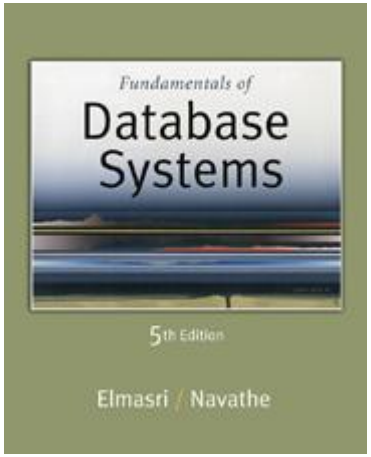


Figure 1.1
A simplified database system environment.



DBMS体系结构（2）

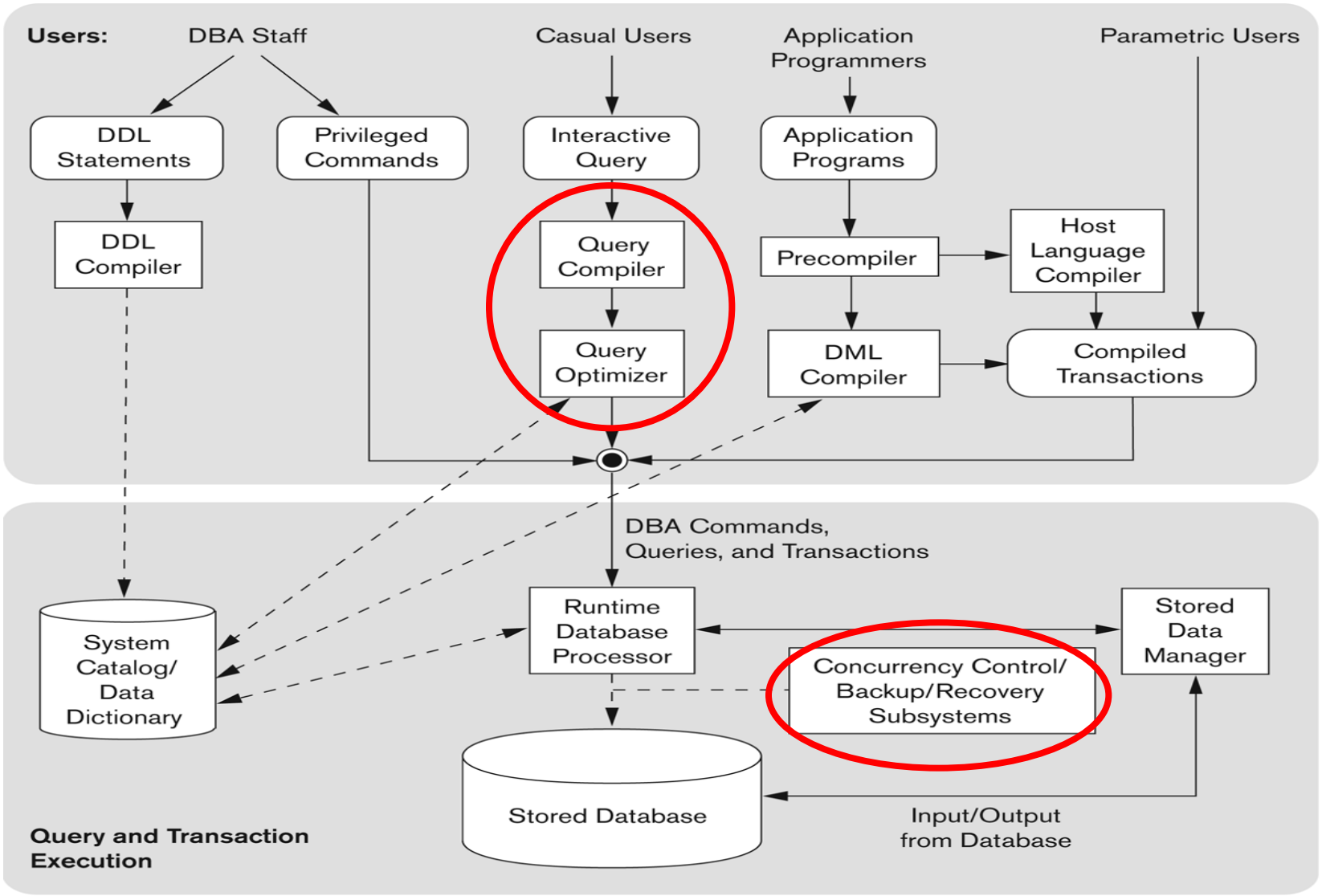


Figure 2.3
Component modules of a DBMS and their interactions.

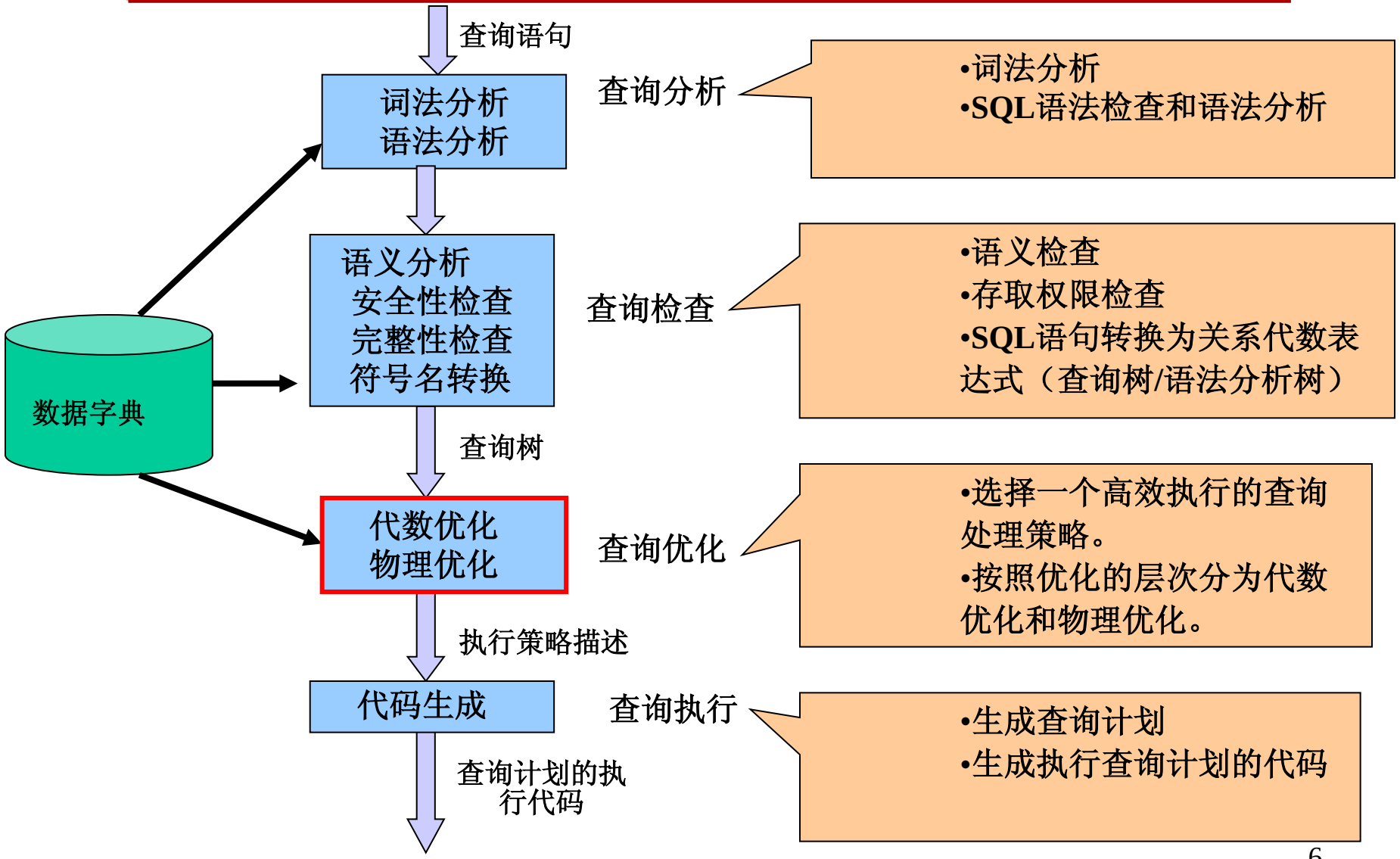
第六章 关系查询处理与查询优化

- 关系查询处理步骤
- 查询优化
  - 查询的代数优化
  - 查询的物理优化
  - 查询优化的一般步骤

关系查询处理步骤

- 关系查询处理的四个阶段:
  - 查询分析
  - 查询检查
  - 查询优化
  - 查询执行

关系查询处理步骤


查询语句
词法分析
语法分析
查询分析
•词法分析
•SQL语法检查和语法分析
语义分析
安全性检查
完整性检查
符号名转换
查询检查
•语义检查
•存取权限检查
•SQL语句转换为关系代数表达式（查询树/语法分析树）
数据字典
查询树
代数优化
物理优化
查询优化
•选择一个高效执行的查询处理策略。
•按照优化的层次分为代数优化和物理优化。
执行策略描述
代码生成
查询执行
•生成查询计划
•生成执行查询计划的代码
查询计划的执行代码

查询优化

- 查询优化目标
  - 选择一个高效执行的查询处理策略，使得查询代价最小
- 查询的执行开销（集中式数据库）
  - 总代价=I/O代价 + CPU代价 + 内存代价
  - I/O代价是最主要的
- 按照优化的层次分为代数优化和物理优化。
  - 代数优化：关系代数表达式的优化，即按照一定的规则，改变代数表达式中操作的次序和组合
  - 物理优化：存取路径和底层操作算法的选择。选择依据——基于规则的、基于代价的、基于语义的

代数优化（1）

- 通过对关系代数表达式的等价变换来提高查询效率
- 关系代数表达式的等价
  - 指用相同的关系代替两个表达式中相应的关系所得到的结果是相同的

代数优化（1）

- 关系代数表达式等价变换规则
  - 连接、笛卡尔积交换律

$$E_1 \times E_2 \equiv E_2 \times E_1$$
$$E_1 \bowtie E_2 \equiv E_2 \bowtie E_1$$
$$E_1 \underset{F}{\bowtie} E_2 \equiv E_2 \underset{F}{\bowtie} E_i$$

  - 连接、笛卡尔积的结合律

$$(E_1 \times E_2) \times E_3 \equiv E_1 \times (E_2 \times E_3)$$
$$(E_1 \bowtie E_2) \bowtie E_3 \equiv E_1 \bowtie (E_2 \bowtie E_3)$$
$$(E_1 \underset{F_1}{\bowtie} E_2) \underset{F_2}{\bowtie} E_3 \equiv E_1 \underset{F_1}{\bowtie} (E_2 \underset{F_2}{\bowtie} E_3)$$

代数优化（2）

- 投影的串接定律

$$\pi_{A_1,A_2,\cdots,A_n}(\pi_{B_1,B_2,\cdots,B_m}(E)) \equiv \pi_{A_1,A_2,\cdots,A_n}(E)$$

其中{A1,A2,…,An}是{B1,B2,…,Bm}的子集

- 选择的串接定律

$$\sigma_{F_1}(\sigma_{F_2}(E)) \equiv \sigma_{F_1 \wedge F_2}(E)$$

- 选择与投影的交换律

$$\sigma_F(\pi_{A_1,A_2,\cdots,A_n}(E)) \equiv \pi_{A_1,A_2,\cdots,A_n}(\sigma_F(E))$$

- 选择与笛卡尔积交换律
  - F中只有E1的属性：$\sigma_F(E_1 \times E_2) \equiv \sigma_F(E_1) \times E_2$
  - F=F1∧F2，且F1只有E1的属性、F2中只有E2的属性：$\sigma_F(E_1 \times E_2) \equiv \sigma_{F_1}(E_1) \times \sigma_{F_2}(E_2)$
  - F1只有E1的属性，F2有E1和E2的属性：$\sigma_F(E_1 \times E_2) \equiv \sigma_{F_2}(\sigma_{F_1}(E_1) \times E_2)$

代数优化（2）

- 选择与并的分配律

$$\sigma_F(E_1 \cup E_2) \equiv \sigma_F(E_1) \cup \sigma_F(E_2)$$

- 选择与差的分配律

$$\sigma_F(E_1 - E_2) \equiv \sigma_F(E_1) - \sigma_F(E_2)$$

- 选择对自然连接的分配律

$$\sigma_F(E_1 \bowtie E_2) \equiv \sigma_F(E_1) \bowtie \sigma_F(E_2)$$

- 投影与笛卡尔积的分配律

$$\pi_{A_1,A_2,\cdots,A_n,B_1,B_2,\cdots,B_m}(E_1 \times E_2) \equiv \pi_{A_1,A_2,\cdots,A_n}(E_1) \times \pi_{B_1,B_2,\cdots,B_m}(E_2)$$

- 投影与并的分配律

$$\pi_{A_1,A_2,\cdots,A_n}(E_1 \cup E_2) \equiv \pi_{A_1,A_2,\cdots,A_n}(E_1) \cup \pi_{A_1,A_2,\cdots,A_n}(E_2)$$

代数优化（3）

- 查询树的启发式优化（优化的一般准则）
  - 选择运算尽可能先做。是优化策略中最重要、最基本的一条。（减小中间关系）
  - 把投影运算和选择运算同时进行；把投影同其前或其后的双目运算结合起来。（减少扫描关系的次数）
  - 把某些选择同在它前面要执行的笛卡尔积结合起来成为一个连接运算。（把笛卡尔积与选择转换为连接）
  - 找出公共子表达式，把公共子表达式的结果写入中间文件，重复使用。（中间结果复用）

物理优化

- 选择高效合理的操作算法或存取路径，得到优化的查询计划。
- 常用方法
  - 基于规则的启发式优化方法
  - 基于代价估算的优化方法
  - 两者结合的优化方法

实现查询操作的算法

- 选择(选取)操作的实现算法示例
  - 全表扫描法
  - 索引扫描法
    - 如果在选择条件的属性上有索引，先通过索引找到目标索引项，再通过索引项找到元组

实现查询操作的算法

查询示例：

```
SELECT   S#,SN, C#, G
FROM   S,  SC
WHERE  S.S#=SC.S#
```

- 连接操作的实现算法示例
  - 嵌套循环法
    - 两个连接的表，第一个表为外循环，另一个为内循环
  - 排序-合并法
    - 两个表都按照连接属性排序，取第一个表元组的连接属性与第二个表元组的比较
  - 索引连接法
    - 第二个表按照连接属性建索引，取第一个表元组的连接属性与第二个表元组的连接属性比较
  - Hash join法
    - 连接属性作为hash码，用同一个hash函数把两个连接表的元组散列到同一个hash文件

基于启发式规则的存取路径选择优化

- 选择操作的启发式规则
  - 对于小关系，使用全表顺序扫描。
  - 对于大关系：可以采用索引扫描法（如结果的元组数目较小），全表顺序扫描。
- 连接操作的启发式规则
  - 如果两个表都已经按照连接属性排序——排序-合并法；
  - 如果一个表在连接属性上有索引——索引连接法；
  - 如果连接属性上未排序且未建索引，且其中一个表较小——Hash join法；
  - 最后可选用嵌套循环法，并选择较小的表为外循环表。

查询优化的一般步骤

- 把查询转换成语法树，如关系代数语法树。
- 把语法树利用代数优化转换成优化后的标准形式。
- 利用基于启发式规则的物理优化，选择底层的存取路径。
- 生成查询计划，利用基于代价的物理优化，选择代价最小的。

小结

- 查询处理的步骤
- 查询操作的实现算法
- 查询的代数优化和物理优化的概念和原则
- 查询优化的一般步骤