

第9章

关系查询处理和查询优化

第9章 关系查询处理和查询优化

9.1 关系数据库系统的查询处理

9.2 关系数据库系统的查询优化

9.3 代数优化

9.4 物理优化

9.1 关系数据库系统的查询处理

- 9.1.1 查询处理步骤
- 9.1.2 实现查询操作的算法示例

9.1.1 查询处理步骤

- 关系数据库管理系统查询处理阶段：

1. 查询分析
2. 查询检查
3. 查询优化
4. 查询执行

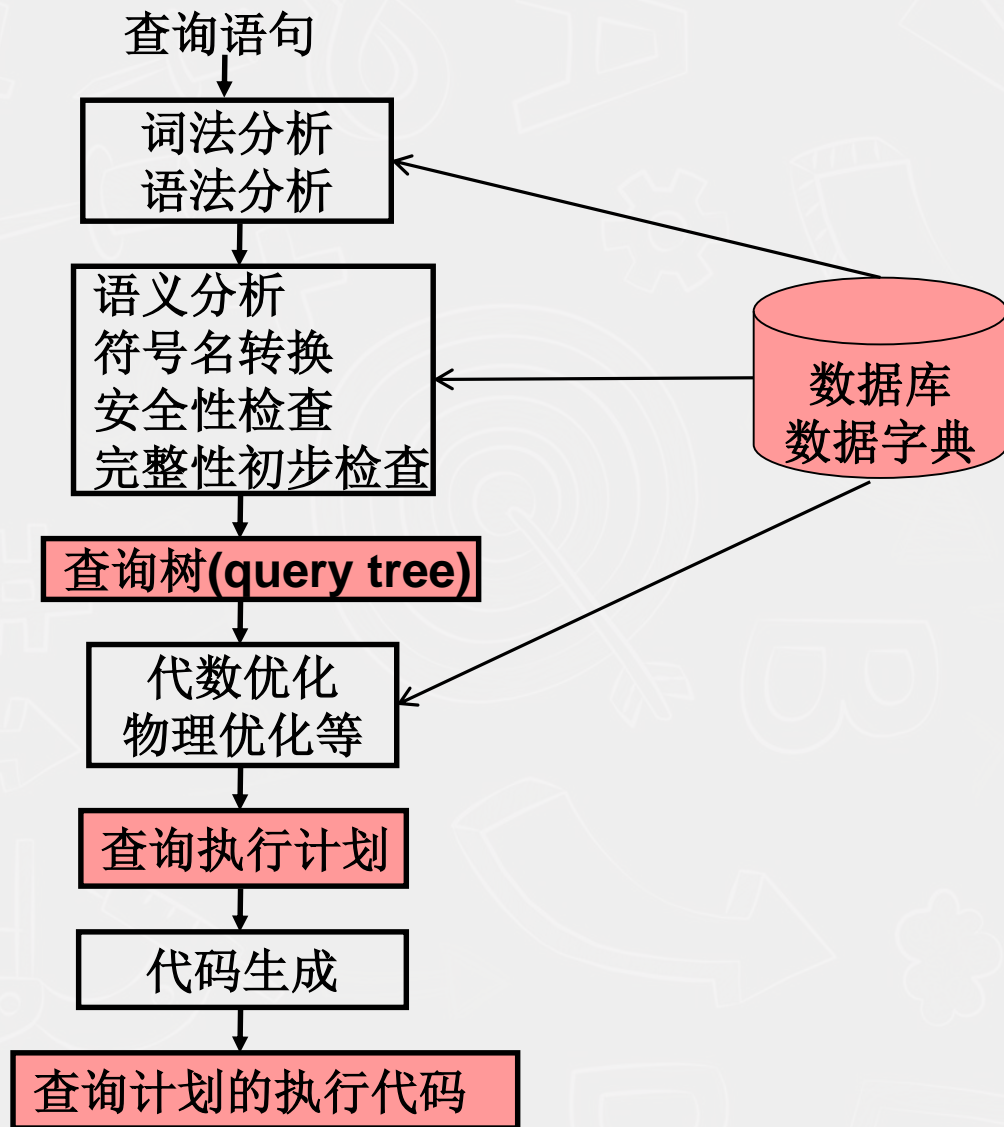
• 查询处理步骤

查询分析

查询检查

查询优化

查询执行



9.1.2 实现查询操作的算法示例

- 选择操作的实现
- 连接操作的实现

1.选择操作的实现

- 选择操作典型实现方法：

- (1) 全表扫描方法 (Table Scan)

- 对查询的基本表顺序扫描，逐一检查每个元组是否满足选择条件，把满足条件的元组作为结果输出
 - 适合小表，不适合大表

- (2) 索引扫描方法 (Index Scan)

- 适合于选择条件中的属性上有索引(例如B+树索引或Hash索引)
 - 通过索引先找到满足条件的元组主码或元组指针，再通过元组指针直接在查询的基本表中找到元组

- [例9.1] SELECT *

FROM Student

WHERE <条件表达式>

考虑<条件表达式>的几种情况：

C1：无条件；

C2：Sno = '201215121'；

C3：Sage>20；

C4：Sdept = 'CS' AND Sage>20；

2.连接操作的实现

- 连接操作是查询处理中最耗时的操作之一

- [例9.2] `SELECT *`

`FROM Student, SC`

`WHERE Student.Sno=SC.Sno;`

- 
- (1) 嵌套循环算法(nested loop join)
 - (2) 排序-合并算法(sort-merge join 或merge join)
 - (3) 索引连接(index join)算法
 - (4) Hash Join算法