



2026年春季学期

# 数据库系统概论

## An Introduction to Database Systems

### 第十章 关系查询处理和查询优化

中国科学技术大学  
人工智能与数据科学学院  
黄振亚, [huangzhy@ustc.edu.cn](mailto:huangzhy@ustc.edu.cn)



# 第十章 关系系统及其查询优化

2

**10.1 关系数据库系统的查询处理**

**10.2 关系数据库系统的查询优化**

**10.3 代数优化**

**10.4 物理优化**

**\*10.5 查询计划的执行**

**10.6 小结**



# 关系系统及其查询优化（续）

3

- 本章目的：
  - RDBMS的查询处理步骤
  - 查询优化的概念
  - 基本方法和技术
  
- 查询优化分类：
  - 代数优化
  - 物理优化



# 10.1 关系数据库系统的查询处理

4

- 10.1.1 查询处理步骤
- 10.1.2 实现查询操作的算法示例



## 10.1.1 查询处理步骤

5

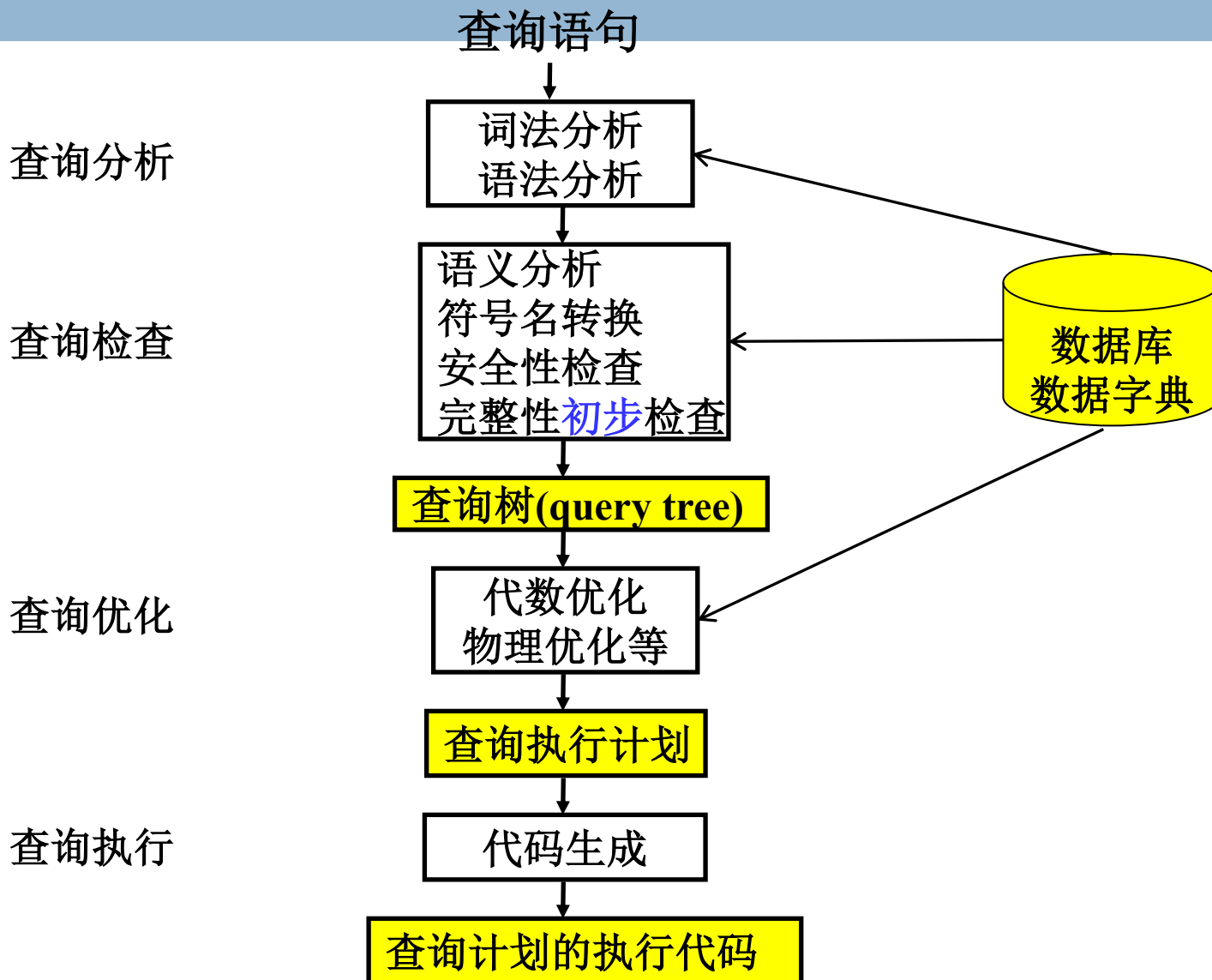
### □ RDBMS查询处理阶段：

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

6





# SQL语句

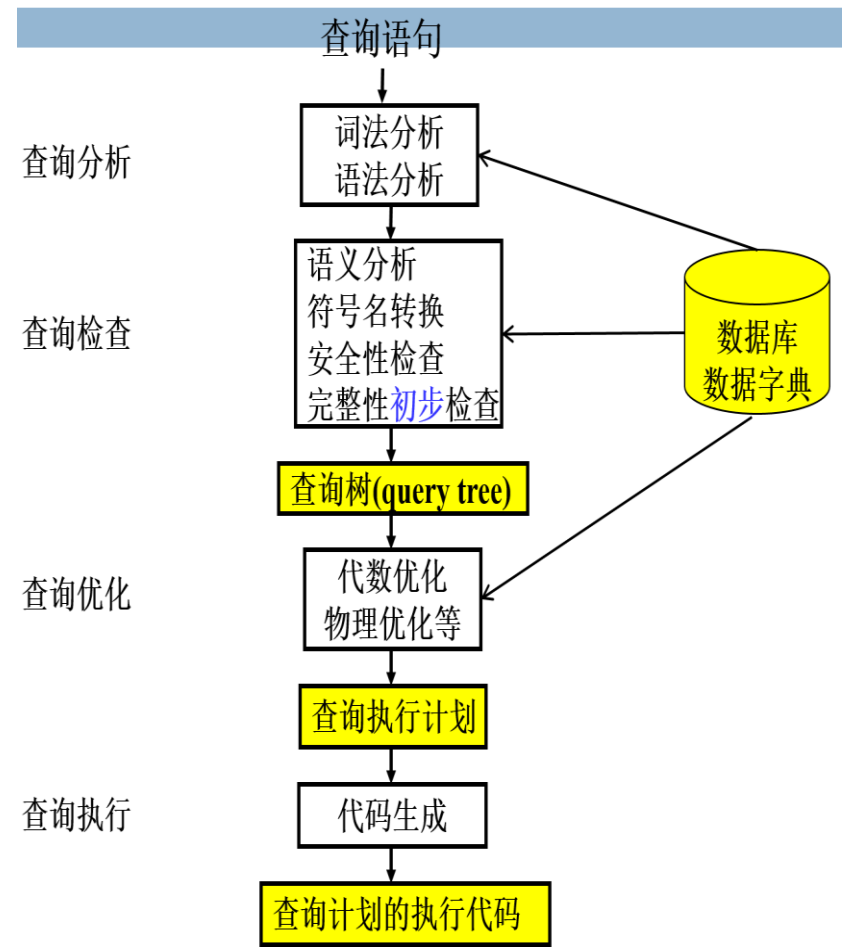
7

- Select sno, sname, cno, grade
- From Student, SC
- Where Student.sno = SC.sno and grade < 60



# 1. 查询分析

- 对查询语句进行扫描、词法分析和语法分析
- 词法分析：从查询语句中识别出语言符号
  - SQL关键字、属性名、关系名等
- 语法分析：进行语法检查和语法分析
  - 是否符合SQL语法规则





## 2. 查询检查

9

### □ 查询检查的任务

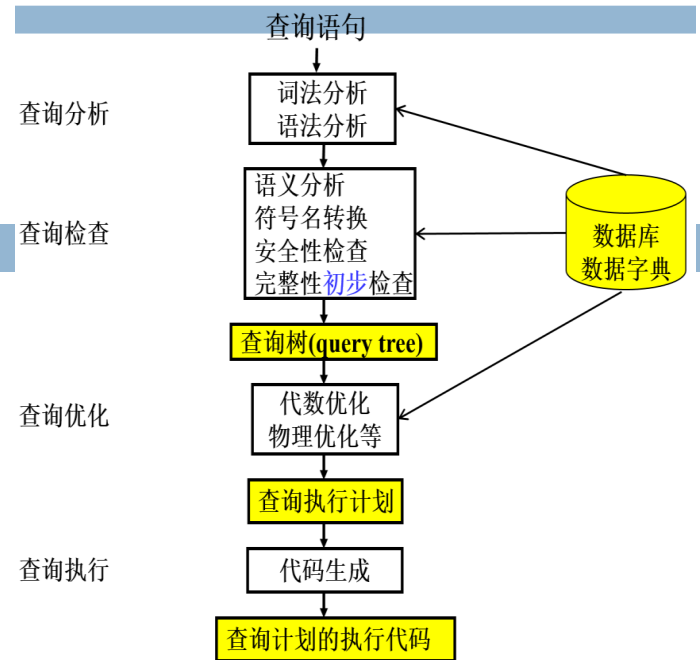
- 合法权检查
- 视图转换
- 安全性检查：用户权限
- 完整性初步检查（静态）

### □ 目标：根据数据字典对合法的查询语句进行语义检查

- 数据库对象，如关系名、属性名、视图名是否存在且有效
- 若对视图的操作，则进行视图消解，或实体化视图

### □ 根据数据字典中的用户权限和完整性约束定义对用户的存取权限进行检查

- 是否需要拒绝执行？
  - 安全性：权限
  - 完整性：如，grade是int型

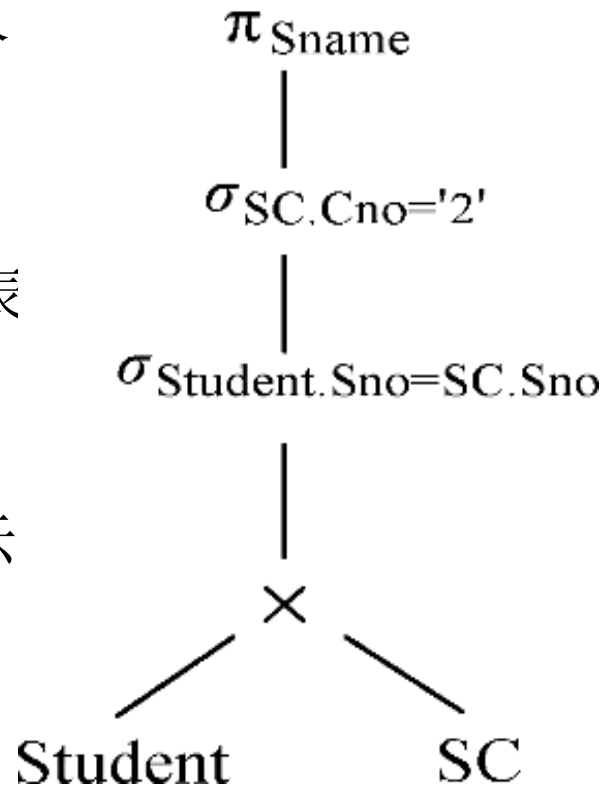




## 2. 查询检查

10

- 检查通过后，把SQL查询语句转换成等价的关系代数表达式
- RDBMS一般都用查询树(语法分析树)来表示扩展的关系代数表达式
- 把数据库对象的外部名称转换为内部表示



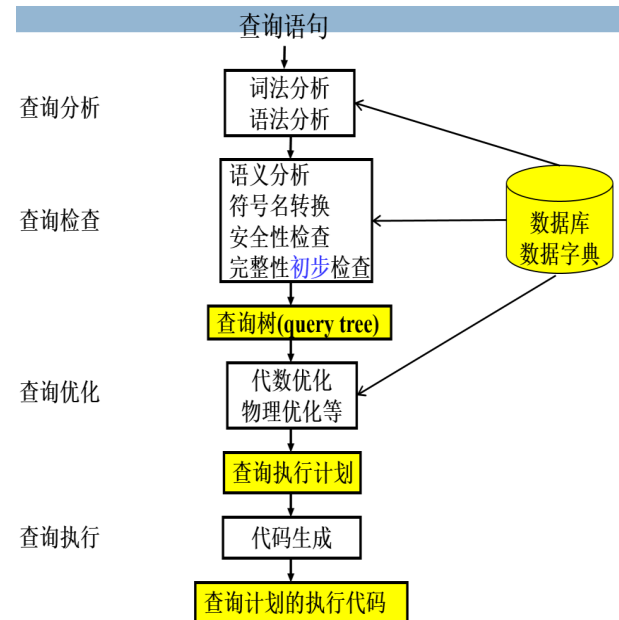
关系代数语法树



# 3. 查询优化

- 查询优化：选择一个高效执行的查询处理策略
- 查询优化层次分类：
  - 代数优化：指关系代数表达式的优化
  - 物理优化：指存取路径和底层操作算法的选择
- 查询优化方法选择的依据：
  - 基于规则(rule based)
  - 基于代价(cost based)
  - 基于语义(semantic based)

实际操作中，这些层次和方法是综合使用的。



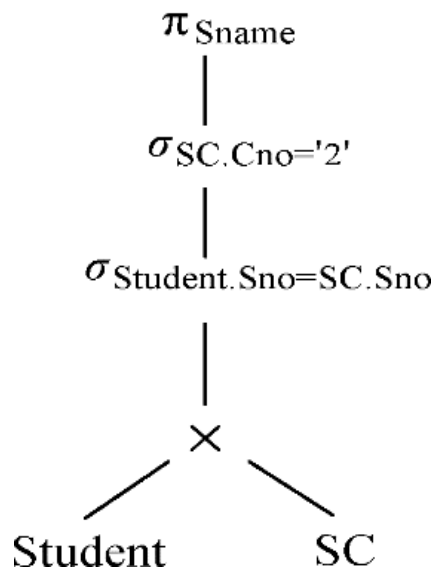
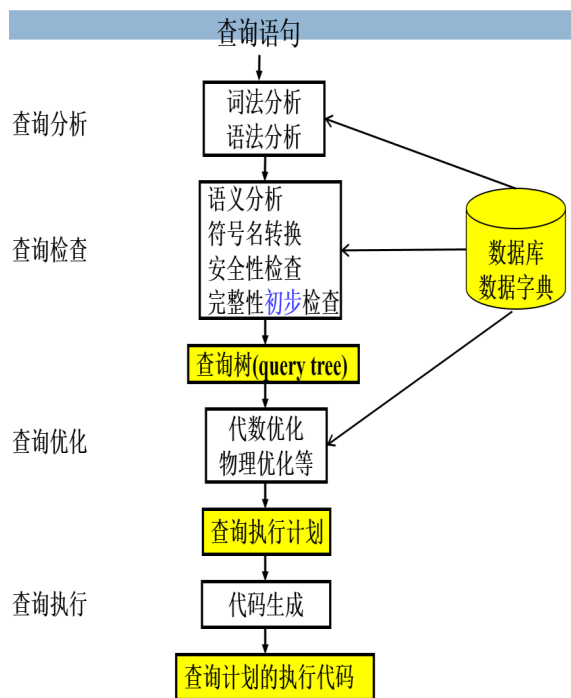


# 4. 查询执行

- 依据优化器得到的执行策略生成查询计划
- 代码生成器(code generator)生成执行查询计划的代码

## □ 两种执行方法

- 自顶向下
- 自底向上





# 10.1 关系数据库系统的查询处理

13

- 10.1.1 查询处理步骤
- 10.1.2 实现查询操作的算法示例



## 10.1.2 实现查询操作的算法示例

14

- 一、选择操作的实现
- 二、连接操作的实现



# 一、选择操作的实现

15

□ [例10.1] **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='20180003';**

**C3: Sage>20;**

**C4: Sdept='CS' AND Sage>20;**



# 选择操作的实现（续）

16

## □ 选择操作典型实现方法：

### □ 1. 简单的全表扫描方法

- 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

### □ [例10.1- C1] `Select * from student`

#### ■ Where 无条件

#### ■ 假设可以使用的内存为M块，全表扫描算法思想：

- (1) 按照物理次序读Student的M块到内存
- (2) 检查内存的每个元组t，如果满足选择条件，则输出t
- (3) 如果Student还有其他块未被处理，重复(1)和(2)

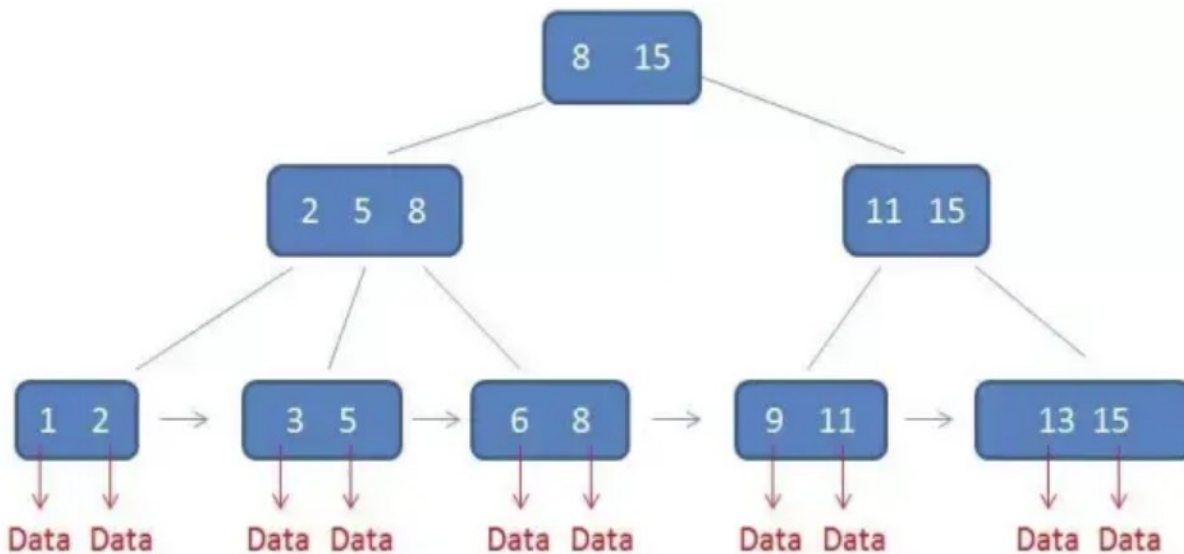


# 选择操作的实现（续）

## 选择操作典型实现方法：

### 索引(或散列)扫描方法

- 适合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





# 选择操作的实现（续）

18

## □ 索引扫描方法

[例10.1] `Select * from student where Sno='20180003';`

## □ [例10.1-C2] Sno上有索引(或Sno是散列码)

- 使用索引(或散列)得到Sno为 '20180003' 元组的指针
- 通过元组指针在student表中检索到该学生

[例9.1] `Select * from student where Sage>20;`

## □ [例10.1-C3] Sage>20, 并且Sage 上有B+树索引

- 使用B+树索引找到Sage=20的索引项, 以此为入口点在B+树的顺序集上得到Sage>20的所有元组指针
- 通过这些元组指针到student表中检索到所有年龄大于20的学生



## 选择操作的实现（续）

19

### □ 索引扫描方法

[例10.1] `SELECT * FROM Student WHERE Sdept='CS' AND Sage>20;`

### □ [例10.1-C4] 如果Sdept和Sage上都有索引：

- 算法一：分别用上面两种方法分别找到Sdept= ‘CS’的一组元组指针和Sage>20的另一组元组指针
  - 求这2组指针的交集
  - 到student表中检索
  - 得到计算机系年龄大于20的学生
  
- 算法二：找到Sdept= ‘CS’的一组元组指针，
  - 通过这些元组指针到student表中检索
  - 对得到的元组检查另一些选择条件(如Sage>20)是否满足
  - 把满足条件的元组作为结果输出。



# 一、选择操作的实现

20

□ [例10.1] **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; **全表扫描**

C2: **Sno='20180003'**; **索引扫描**

C3: **Sage>20**; **索引扫描, 或全表扫描**

C4: **Sdept='CS' AND Sage>20**; **综合考虑**