|  |  |
| --- | --- |
| 第10章作业 | |
| 姓名 | 潘腾凯 |
| 学号 | 37220232203786 |
| 班级 | 01班 |

**1.试述查询优化在关系数据库系统中的重要性和可能性。**

答：（1）重要性：

提升查询效率：优化可以显著减少 SQL 语句的执行时间，提升系统响应速度。

降低系统资源消耗：避免不必要的 CPU、内存和磁盘 I/O 开销。

支持高并发访问：优化后的查询有助于数据库在多用户环境下保持稳定性能。

防止系统性能瓶颈：避免由于低效查询导致数据库阻塞或崩溃。

（2）可能性：

由于关系表达式的语义级别很高，使关系系统可以从关系表达式中分析查询

语义，提供了执行查询优化的可能性

**2.假设关系 R(A,B) 和 S(B,C,D) 情况如下：R 有 20 000 个元组，S 有 1 200 个元组，一个块能装 40 个 R 的元组，能装 30 个 S 的元组。试估算下列操作需要多少次磁盘块读写。**

**① R 上没有索引，select \* from R;**

**② R 中 A 为主码，有 3 层 B+树索引，select \* from R where A = 10;**

**③ 嵌套循环连接 R▷◁S。**

**④ 排序合并连接 R▷◁S，区分 R 与 S 在 B 属性上有序和无序两种情况。**

答：关系R有20000个元组，每块装40个，需要500块。同理关系S需要1200/30=40块。

①没有索引，全表扫描，读完整个要500块。故需要500次读写。

②A是主码，唯一；且有3层B+树索引。访问路径代价为：查找索引3块+数据读取1块。快数为4.

③嵌套循环连接，外层表每一块都与内层表全部块匹配。设R为外表S为内表。

外表 R：500 块；

每块 R 匹配 S（40 块）-> 500 \* 40；

总计块数 = 500 + 500×40 = 20,500

④情况一：R和S都在B属性上有序：  
只需要做一次归并连接，每张表各读一次即可。故块数总计500+40=540。

情况二：无序：

需要先将R和S都按B排序，再归并。

排序阶段（外部排序）：

R 有 500 块，假设内存能放 M 块（不指定，默认两路归并）

外部排序代价约为：2 × B(R) × logM(B(R))，以粗略估算简化处理：

通常用简单公式近似：

排序 R：2 × 500 × log2(500) ≈ 2 × 500 × 9 ≈ 9,000

排序 S：2 × 40 × log2(40) ≈ 2 × 40 × 6 ≈ 480

加上归并连接：

再扫描一次 R 和 S：500 + 40 = 540

结果：排序 + 归并 ≈ 9,000 + 480 + 540 = 10,020

**3.对“学生选课管理”数据库，查询信息管理与信息系统专业学生选修的所有课程名称。**

**SELECT Cname FROM Student, Course, SC**

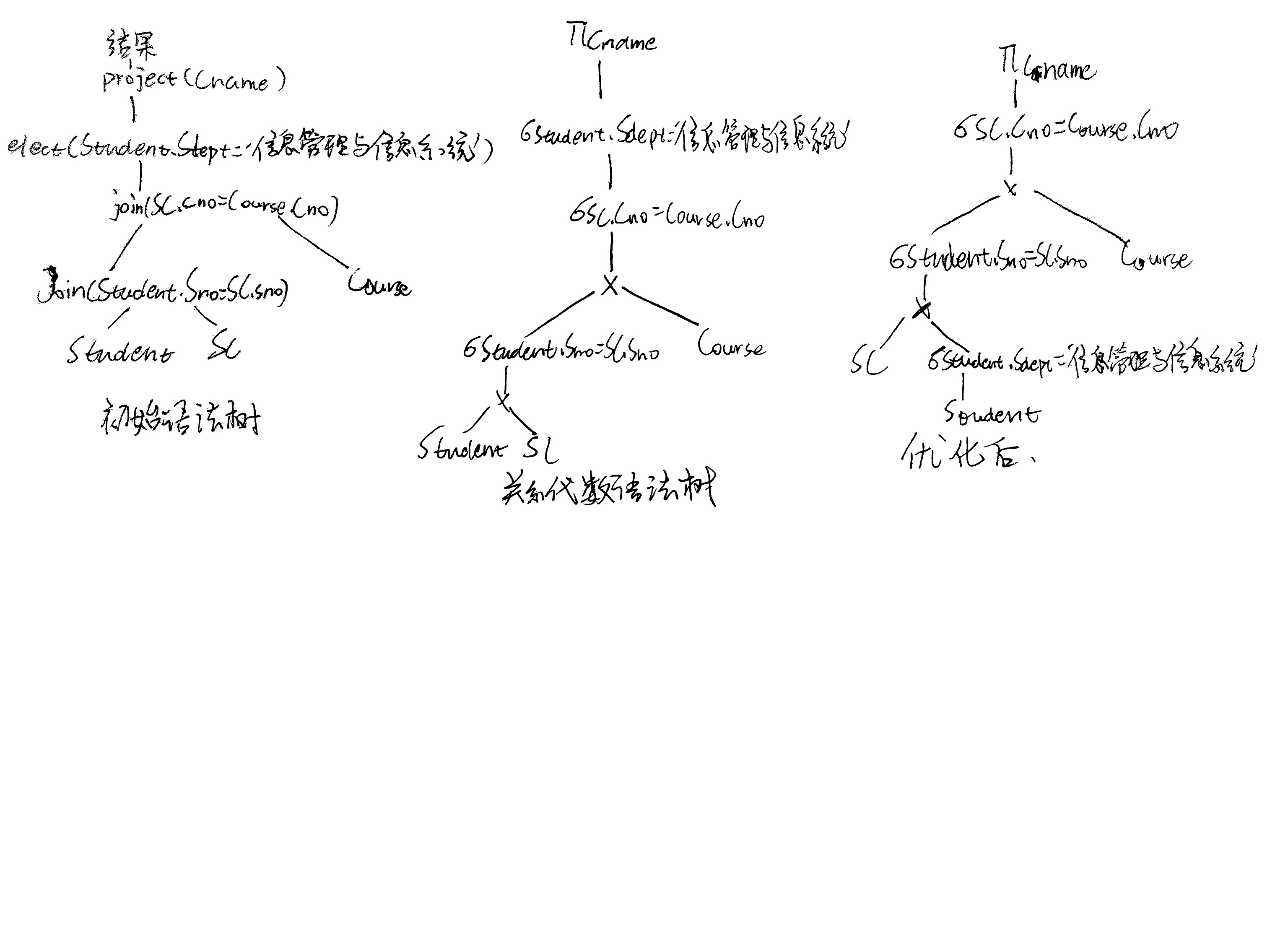
**WHERE Student.Sno = SC.Sno**

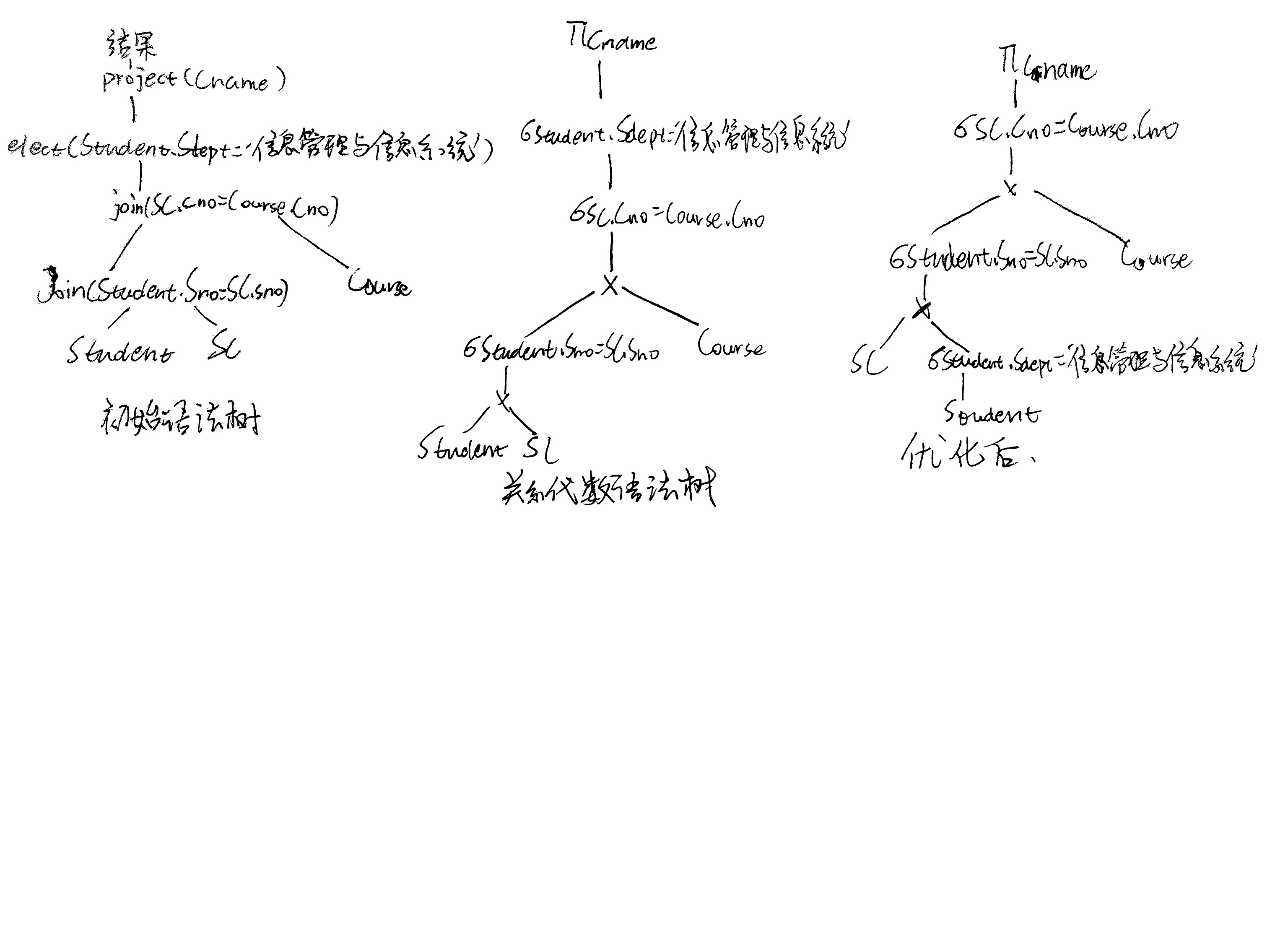
**AND SC.Cno = Course.Cno**

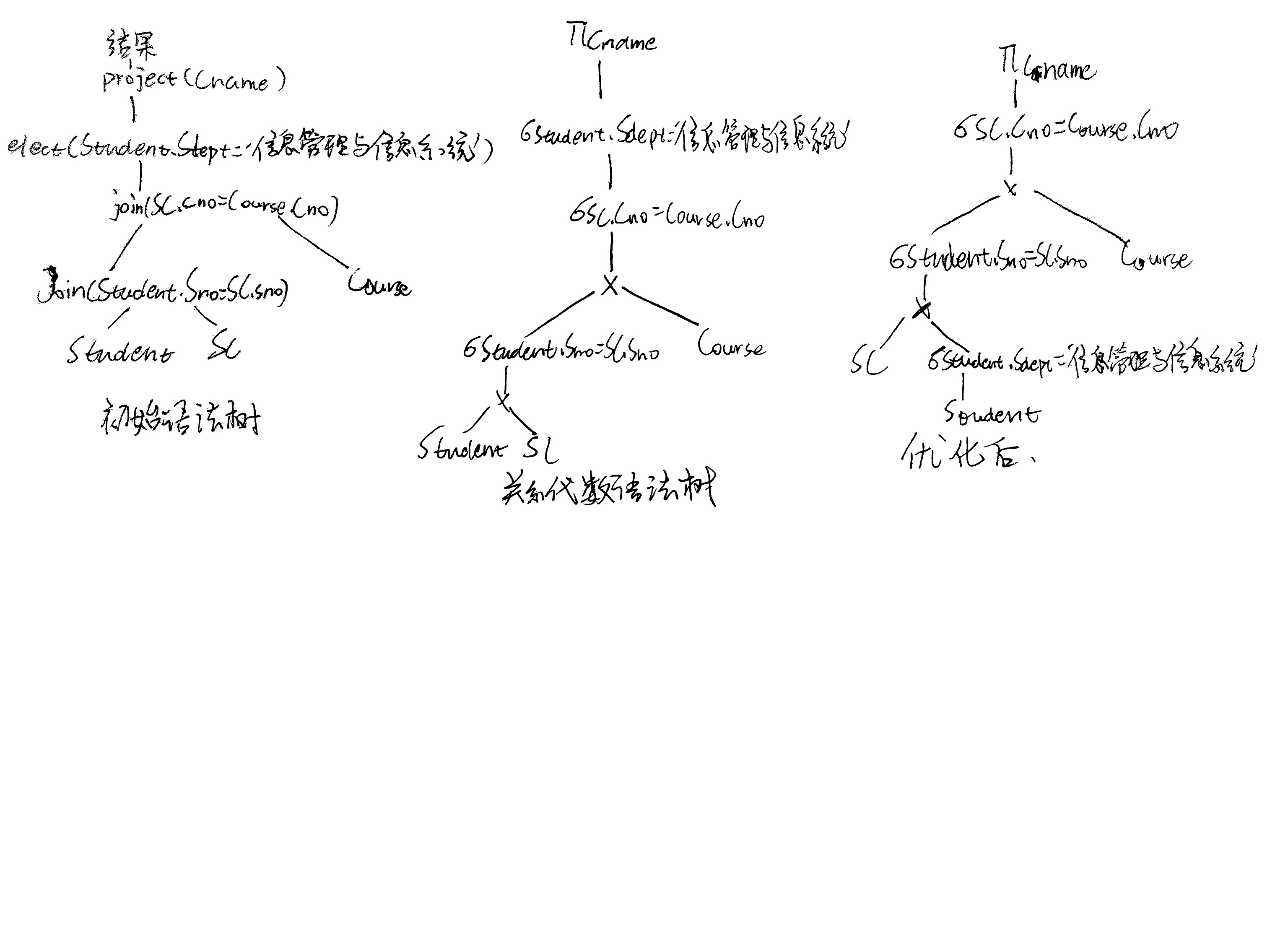
**AND Student.Smajor = '信息管理与信息系统';**

**试画出用关系代数表示的语法树，并用关系代数表达式优化算法对原始的语法树进行优化处理，画出优化后的标准语法树。**

答：如图

****

****

****

**4.对于数据库模式：**

**Teacher (Tno, Tname, Tsex, Ttitle, Tbirthdate, Dno);**

**Department (Dno, Dname, Dcontact, Dtel, Director, SHno);**

**Work (Tno, Dno, Year, Salary);**

**以上模式的语义参见本书附录。假设 Teacher 的 Tno 属性、Department 的 Dno 属性以及 Work 的 Year 属性上有 B+树索引。请说明下列查询语句的一种较优的处理方法。**

**① SELECT \* FROM Teacher WHERE Tsex = '女';**

**② SELECT \* FROM Department WHERE Dno < 301;**

**③ SELECT \* FROM Work WHERE Year <> 2000;**

**④ SELECT \* FROM Work WHERE Year > 2000 AND Salary < 5000;**

**⑤ SELECT \* FROM Work WHERE Year < 2000 OR Salary < 5000;**

答：

①对 Teacher 进行全表描，查看元组是否满足性别为女。因为男、女比例相差不会很大，使用全表扫描是合适的。

②如果满足 Dno<301元组数目较少，可以通过索引找到Dno=301的索引项，然后顺着 B+树的顺序集得到 Dno<301的索引项，通过这些索引项的指针找到 Department 中的元组。

如果满足 Dno<301 的元组数目较多，可以采用对 Departmelnt 的全表扫描方式处理。

RDBMS的查询优化算法将计算这两种执行计划的代价，然后选择代价小的一种执行。

③ 对 Work 进行全表扫描，査看元组是否满足 Year ' 2000。

虽然在 Work 的 Year 属性上有 B+树索引，因为要查找的是除 2000年之外的所有元组，所

以使用全表扫描是合适的。

④ 一种查询执行计划是：通过 Year 的索引找到满足 Year>2000的元组，检查元组是否满足Salary<5000。因为 Work的 Year属性上有 B+树索引，可以通过索引找到 Year=2000的索引项作为入口点，然后顺着 B+树的顺序集得到 Year>2000的所有索引项，再通过这些索引项的指针找到 Work元组中满足 Salary<5000 条件的元组。

另一种查询执行计划是：对 Work 进行全表扫描，一边扫描一边查看元组是否满足 Year >2000 AND Salary< 5000。

RDBMS的查询优化算法将计算这两种执行计划的代价，然后选择代价小的一种执行

⑤ 对 Work 进行全表扫描,一边扫描一边査看元组是否满足 Year<2000 或 Salary<5000。

因为查询条件是 OR连接的析取选择条件，所以采用全表扫描是合适的。

**5.对于第4题中的数据库模式，有如下的查询:**

**SELECT Tname FROM Teacher, Department, Work**

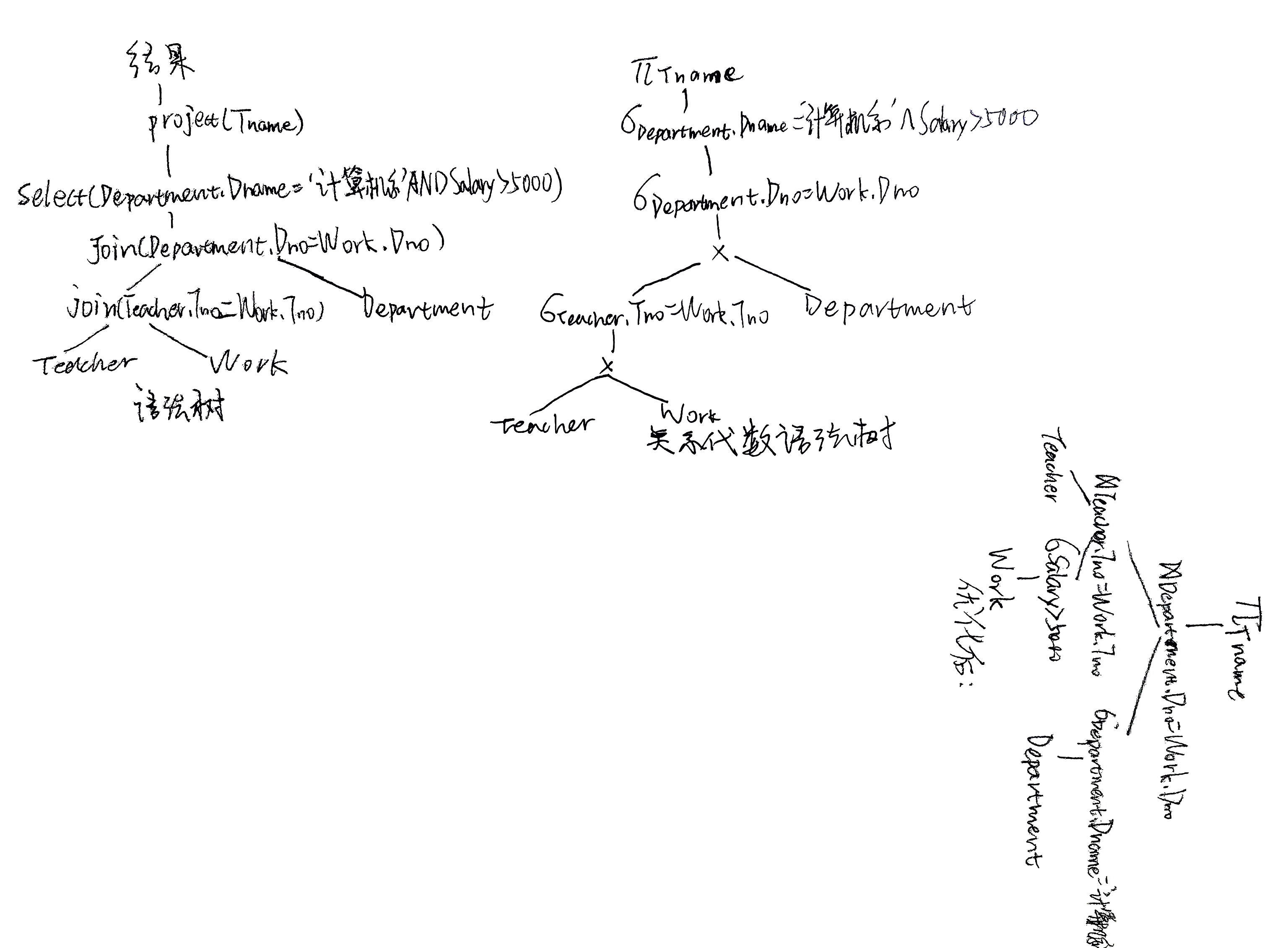
**WHERE Teacher.Tno = Work.Tno**

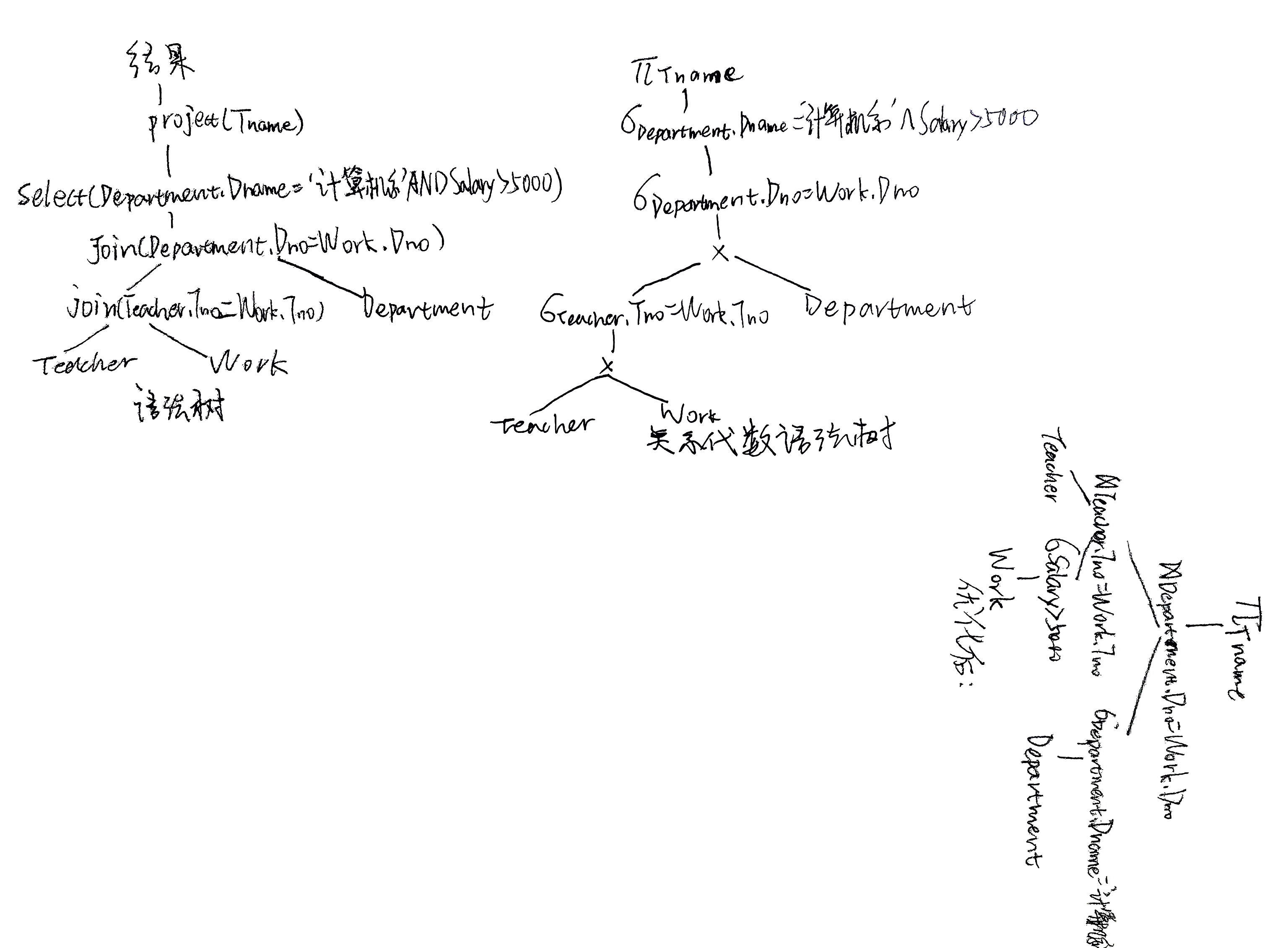
**AND Department.Dno = Work.Dno**

**AND Department.Dname = '计算机系' AND Salary > 5000**

**画出语法树以及用关系代数表示的语法树，并对关系代数语法树进行优化，画出优化后的语法树。**

答：

****

****

**6.试述关系数据库管理系统查询优化的一般准则。**

答：（1）优先使用选择性高的条件：尽量让选择操作（如 WHERE）尽早执行，尤其是那些能大幅度减少结果集的条件。

（2）避免全表扫描：使用索引加快查询，特别是在 WHERE、JOIN、ORDER BY、GROUP BY 中涉及的字段上。

（3）选择高效的连接顺序和连接方式：将小表放在连接的外层（如嵌套循环），并选择成本较低的连接算法（如哈希连接、归并连接）。

（4）减少中间结果集的大小：通过早期投影（SELECT 指定列）和选择来减少中间结果的存储和传输成本。

（5）避免不必要的排序和重复：除非结果需要排序或去重，应避免使用 ORDER BY 和 DISTINCT，以减少排序开销。

（6）使用合适的索引结构：如 B+ 树索引适用于范围查询，哈希索引适用于等值查询。

（7）利用统计信息和执行计划：优化器根据数据分布、索引情况选择执行计划，使用 EXPLAIN 工具可以帮助分析和调优查询。

**7.试述关系数据库管理系统查询优化的一般步骤。**

答：（1） 语法与语义分析：

检查 SQL 查询的语法是否正确；

验证表名、字段名是否存在；

类型检查、权限验证等。

（2） 查询重写（逻辑优化）：

对原始查询进行逻辑等价变换，生成更高效的表达式形式。比如：将子查询改写为连接；推动选择操作（谓词下推）；删除冗余子句；合并嵌套查询。

（3） 生成逻辑查询计划：

将重写后的查询语句转换为关系代数表达式；

每个操作（选择、投影、连接等）形成一个操作树。

（4） 代价估算与物理查询计划选择：

查询优化器根据表的统计信息（如行数、索引、数据分布等），估算每种执行计划的代价（I/O、CPU）；

选择代价最低的物理执行计划（如索引扫描 vs 全表扫描，哈希连接 vs 嵌套连接）。

（5） 执行计划生成与执行：

将选定的物理执行计划转为低层次操作（如磁盘访问、缓存处理）；数据库执行计划，返回结果。

