数据库系统第六次作业

人工智能 2103601 班 2021112845 张智雄

一、考虑在公共属性a上连接关系$R(a,b)$和$S(a,c,d)$，假设表上没有可用的索引来加速连接算法。

- ✧ 缓冲区中有$B=75$页；
- ✧ 表R包含$M=2400$个页面，每个页面包含80个元组；
- ✧ 表S包含$N=1200$个页面，每个页面包含100个元组；

请回答以下关于计算连接的 I/O 开销的问题。您可以假设最简单的开销模型，即每次只读写一个页面。你还可以假设你需要一个缓冲块来保存演变中的输出块，以及一个输入块来保存内部关系的当前输入块。你可以忽略写最终结果的成本。

A. 以S为外部关系，R为内部关系的散列连接。您可以忽略递归分区和部分填充的块。划分阶段的成本是多少?

B. 以R为外部关系，S为内部关系的块嵌套循环连接

C. 以S为外部关系，R为内部关系的块嵌套循环连接

答:因为需要一个缓冲块来保存演变中的输出块,以及一个输入块来保存内部关系的当前输入块。所以块嵌套循环连接的外部关系**最多可以使用$M-2$块**。

A. 对关系R和S进行哈希分桶时，R和S的每块读 1 次，需要$B(R)+B(S)=3600$次 I/O；将R和S的桶全部写入文件，需要$\sum_i^{M-1}\big(B(R_i)+B(S_i)\big)\approx B(R)+B(S)=3600$次 I/O；使用一趟集合差算法计算$R_i\bowtie S_i$的 I/O 代价为$B(R_i)+B(S_i)$；因此最终 I/O 代价$3B(R)+3B(S)=10800$

B. 外关系R的每个元组只读 1 次,每次产生$B(R)=2400$次 I/O;内关系S扫描$\left\lceil\frac{B(R)}{M-2}\right\rceil=\left\lceil\frac{2400}{75-2}\right\rceil$次，合计$1200\times\left\lceil\frac{2400}{75-2}\right\rceil=39600$次 I/O；则最终 I/O 代价为

$$B(R)+B(R)\left\lceil\frac{B(S)}{M-2}\right\rceil=2400+39600=42000$$

C. 外关系S的每个元组只读 1 次,合计产生$B(S)=1200$次 I/O;内关系R扫描$\left\lceil\frac{B(S)}{M-2}\right\rceil=\left\lceil\frac{1200}{75-2}\right\rceil$次，合计$2400\times\left\lceil\frac{1200}{75-2}\right\rceil=40800$次 I/O；则最终 I/O 代价为

$$B(S)+B(S)\left\lceil\frac{B(R)}{M-2}\right\rceil=1200+40800=42000$$

二、设关系$R(X,Y)$和$S(Y,Z)$，R共有1000个元组，S共有1500个元组，每个块中可容纳20个R元组或50个S元组。S中Y不同值的个数为20。

1) 若在$S.Y$上建有聚簇索引，估计R和S基于索引连接的 I/O 代价。
2) 若在$S.Y$上建有非聚簇索引，估计R和S基于索引连接的 I/O 代价。

答：由题可知，关系R的元组数为$T(R)=1000$，关系S的元组数为$T(S)=1500$，关系R的页数为$B(R)=\left\lceil\frac{1000}{20}\right\rceil=50$，关系$S$的页数为$B(S)=\left\lceil\frac{1500}{50}\right\rceil=30$，而关系$S$属性集$\{Y\}$的不同值的个数为$V(S,Y)=20$

1) R的每块只读 1 次，合计$B(R)$次 I/O。因为在$S.Y$上建有**聚簇索引**，所以对于R的每个元组r，S中能与r连接的元组一定连续存储于S的文件中，约占$\lceil\frac{B(S)}{V(S,Y)}\rceil$个块，合计$T(R)\left\lceil\frac{B(S)}{V(S,Y)}\right\rceil$次 I/O。

忽略写结果的代价，则 I/O 代价为$B(R)+T(R)\left\lceil\frac{T(S)}{V(S,Y)}\right\rceil=50+1000\times\left\lceil\frac{30}{20}\right\rceil=2050$次。

2) R的每块只读 1 次，合计$B(R)$次 I/O。对于R的每个元组r，S中平均约有$\frac{T(S)}{V(S,Y)}$个元组能与r相连。因为索引是**非聚簇索引**，所以这些元组在文件中不一定连续存储。最坏情况下，读每个元组产生 1 次 I/O，合计$\frac{T(R)T(S)}{V(S,Y)}$次 I/O。

忽略写结果的代价，则最终 I/O 代价为$B(R)+\frac{T(R)T(S)}{V(S,Y)}=50+1000\times1500/20=75050$次。

三、已知2个关系$R(A,B)$和$S(B,C)$，其主键分别为$R.A$和$S.B$。R有40000个元组，S有60000个元组，一块中可以容纳20个R元组或30个S元组。设2个关系均采用聚簇存储，且每个关系中的元组均已按照其主键值递增排序。现在要执行自然连接操作$R\bowtie S$。设缓冲区中可用内存页数为$M=41$，回答下列问题：

1) 采用嵌套循环连接算法执行$R\bowtie S$分别需要进行多少次 I/O？给出具体分析过程。

2) 采用归并连接算法执行$R\bowtie S$分别需要进行多少次 I/O？给出具体分析过程。

3) 设$R.B$是关系R的外键，参照$S.B$。如果$R\bowtie S$的结果中元组的平均大小是R中元组平均大小的1.2倍，$R\bowtie S$的结果中元组的平均大小是S中元组平均大小的2倍，那么在外存中存储$R\bowtie S$的结果需要占用多少个块（页）？给出具体分析过程

答：由题可知，关系R的元组数$T(R)=40000$，关系S的元组数$T(S)=60000$，关系R的页数为$B(R)=\left\lceil\frac{40000}{20}\right\rceil=2000$，关系$S$的页数为$B(S)=\left\lceil\frac{60000}{30}\right\rceil=2000$，缓冲区可用的内存页数为$M=41$。

1) 嵌套循环连接算法可分为基于块和基于元组的：不妨设置R为外关系，S为内关系。

- 使用基于块的嵌套循环连接算法：外关系R的每个元组只读 1 次，合计产生$B(R)$次 I/O，内关系S扫描$\frac{B(R)}{M-1}$次，合计次 I/O。忽略写结果的 I/O 代价，共$B(R)+B(R)\left\lceil\frac{B(S)}{M-1}\right\rceil=2000+2000\times\left\lceil\frac{2000}{40}\right\rceil=102000$次 I/O。
- 使用基于元组的嵌套循环连接：外关系R的每个元组只读 1 次，每次产生 1 个 I/O，合计$T(R)$次 I/O。内关系R的每个元组只读$T(S)$次，每次产生 1 个 I/O，合计$T(S)T(R)$次 I/O。忽略写结果的 I/O 代价，共$T(S)(T(R)+1)=2400040000$次 I/O。

而如果内外关系颠倒，S为外关系，R为内关系，则为$T(R)(T(S)+1)=2400060000$次 I/O。

2) 在对R创建归并段时，R的每块只读 1 次，合计$B(R)$次 I/O；而将R的归并段全部写入文件，还需$B(R)$次 I/O；因为S已经按照$S.B$排序所以无需构建归并段

在归并阶段，对 R 和 S 的每个归并段各扫描 1 次，合计$B(R)+B(S)$次 I/O。忽略写结果的 I/O 代价，共$3B(R)+B(S)=3\times2000+2000=8000$次 I/O。

3) $R.B$是关系R的外键，参照$S.B$，则$R \bowtie S$的元组数为$T(R \bowtie S) = 40000$个。

✧ $R \bowtie S$结果中元组的平均大小为R中元组的1.2倍，则一页可以容纳$[20/1.2] = 16$个元组。

✧ $R \bowtie S$结果中元组的平均大小为S中元组的2倍，则一页中可以容纳$[30/2] = 15$个元组。

选取两个情况的最小值，即一页中容纳15个$R \bowtie S$元组。因此在外存中存储$R \bowtie S$的结果需要占用$\left\lceil \frac{40000}{15} \right\rceil = 2667$个块（页）。

四、设教学管理数据库有如下 3 个关系模式：

S(S#, SNAME, AGE, SEX)

C(C#, CNAME, TEACHER)

SC(S#, C#, GRADE)

其中 S 为学生信息表、SC 为选课表、C 为课程信息表；S#、C#分别为 S、C 表的主码，(S#,C#)是 SC 表的主码，也分别是参照 S、C 表的外码。用户有一查询语句：

Select SNAME

From S, SC, C

Where SC.S#=S.S# and SC.C#=C.C# and CNAME="数据库";

检索选学“数据库”课程的学生的姓名。

1) 写出以上 SQL 语句所对应的关系代数表达式。

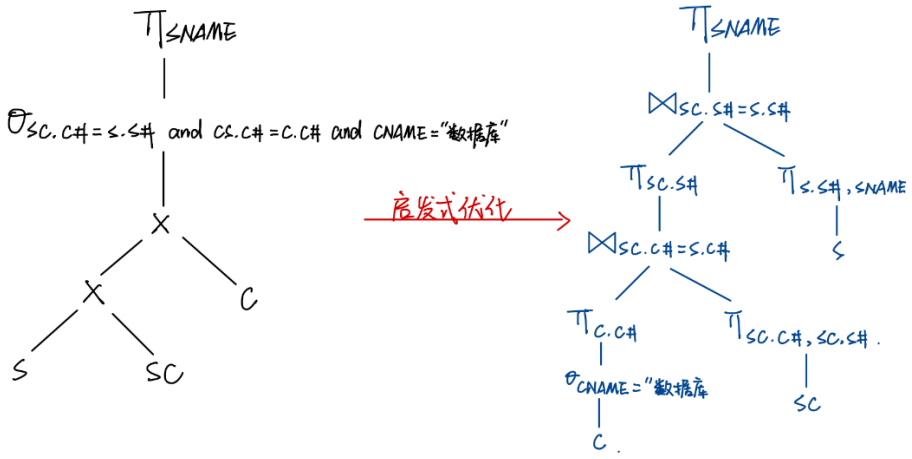
2) 画出上述关系代数表达式所对应的查询计划树。使用启发式查询优化算法，对以上查询计划树进行优化，并画出优化后的查询计划树。

3) 设 SC 表有 10000 条元组，C 表有 50 条元组，S 表中有 1000 条元组，SC 中满足选修数据库课程的元组数为 150，计算优化前与优化后的查询计划中每一步所产生的中间结果大小。

答：解答如下：

1) $\Pi_{SNAME}\left(\sigma_{SC.S\#=S.S\#\ and\ CS.C\#=C.C\#\ and\ CNAME="数据库"}((S \times SC) \times C)\right)$

2) 上述关系代数表达式对应**优化前和优化后**的查询计划树如下：



3) 计算优化前与优化后的查询计划中每一步所产生的中间结果大小如下：

优化前：

① S 与 SC 的笛卡尔积$S \times SC$的元组数为$10000 \times 1000 = 10^7$条*(S#, SNAME, AGE, SEX, C#, GRADE)*；

② 上述①的结果与 C 的笛卡尔积$(S \times SC) \times C$的元组数为$10^7 \times 50 = 5 \times 10^8$条*(S#, SNAME, AGE, SEX, C#, GRADE, CNAME, TEACHER)*；

③ 然后进行选择σ，得到 150 条满足条件的元组*(S#, SNAME, AGE, SEX, C#, GRADE, CNAME, TEACHER)*；

④ 最后对上述③的结果中 SNAME 属性进行投影Π，得到 150 条元组*(SNAME)*

优化后：

① SC 经过投影Π得到 10000 条元组*(S#, C#)*；

② C 经过选择σ和投影Π后得到 1 条元组*("数据库的课程编号 C#")*；

③ S 经过投影Π得到 1000 条元组*(S#, SNAME)*；

④ 上述①和②的结果自然连接⋈得到 150 条元组*(S#, C#)*，再经投影Π后同样是 150 条元组*(S#)*；

⑤ 上述④和③的结果自然连接⋈得到 150 条元组*(S#, SNAME)*；

⑥ 最后对上述⑤的结果中 SNAME 属性进行投影Π，得到 150 条元组*(SNAME)*

五、给定以下关系模式，

Student (sid, sname, major)

Course (cid, cname, credit)

Enrollment (sid, cid, grade)

1) 考虑以下的 SQL 查询语句，绘制其查询计划树。

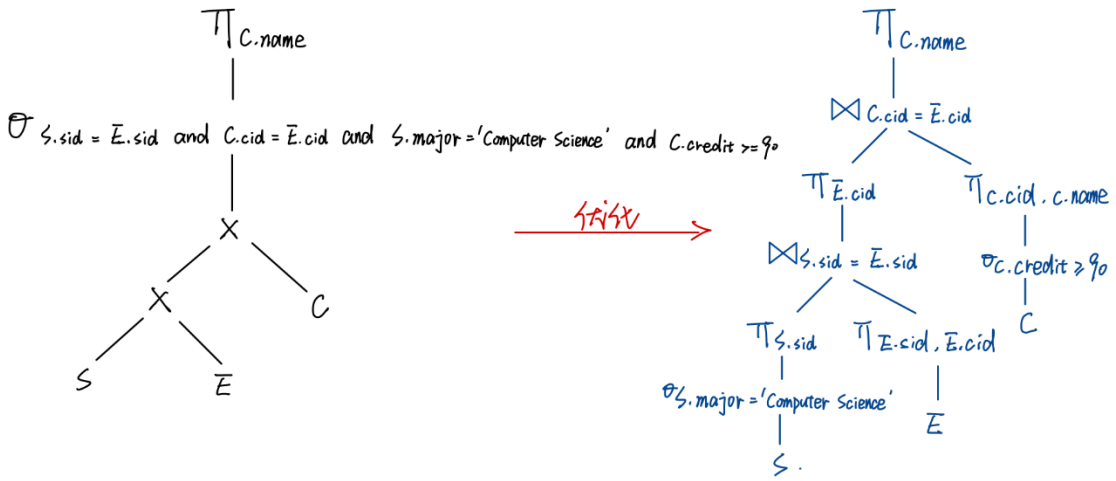
```
SELECT C.name
FROM Student S, Course C, Enrollment E
WHERE S.sid = E.sid AND C.cid = E.cid AND S.major = 'Computer Science' AND C.credit >= 90;
```

2) 假设在 *Student.major* 和 *Enrollment.sid* 上建有索引，绘制优化后的查询计划树。

答：此 SQL 查询语句对应关系代数表达式为：

$$\Pi_{C.name}\left(\sigma_{S.sid=E.sid\ AND\ C.cid=E.cid\ AND\ S.major='Computer\ Science'\ AND\ C.credit\geq 90}\left((S \times E) \times C\right)\right)$$

对应**优化前和优化后**的查询计划树如下：



六、已知一个关系数据库的模式如下：

关系 *B(bno, bname, author)*为图书表，其中 *bno* 为书号，*bname* 为书名，*author* 为作者；

关系 *S(sno, sname, dept)* 为学生表，其中 *sno* 为学号，*sname* 为姓名，*dept* 为学生所在系；

关系 *L(sno, bno, date)*为借书表，其中 *sno* 为学号，*bno* 为书号，*date* 为借书时间。

回答下列问题：

1) 绘制下面的 SQL 查询语句的逻辑查询计划树。

SELECT author FROM B NATURAL JOIN S NATURAL JOIN L

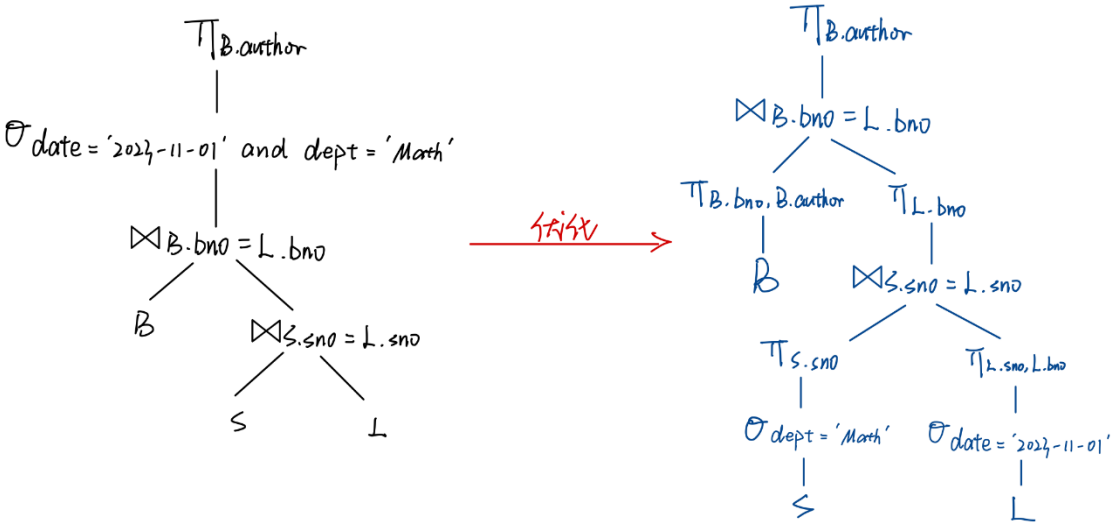
WHERE date = '2023-11-01' AND dept = 'Math';

2) 使用启发式查询优化方法对上面的逻辑查询计划树进行优化，绘制优化后得到的逻辑查询计划树，具体说明你进行这些优化的理由。

答：此 SQL 查询语句对应关系代数表达式为：

$$\prod_{B.author}\left(\sigma_{date='2023-11-01'\ AND\ dept='Math'}\left(B \bowtie (S \bowtie L)\right)\right)$$

对应**优化前和优化后**的查询计划树如下：



将选择操作$\sigma_{dept='Math'}$和$\sigma_{date='2023-11-01'}$以及投影操作$\prod_{B.author}$进行下推可以尽早减少中间结果的大小。具体来说，

a) 将选择操作移到尽可能靠近叶节点，可以减少中间结果的元组数

b) 将投影操作移到尽可能靠近叶节点，可以减少中间结果的属性

c) 先选择投影再进行连接，这样可以减少中间结果的大小，加快连接操作的处理。