第九章 拓展作业 9-2

李梓童 2017202121
提交时间：2019.10.29.

一. 编程要求

给定关系表 $R_i(A_{i1},A_{i2},\cdots)(i<=3)$及 SQL 查询语句，请**编程**构造一个查询优化器，**输入为下述参数，输出为使给定查询代价最小的查询策略及相应查询代价**。

数据字典：

1. 关系统计信息：n_R，f_R，$b_R=n_R/f_R$，$Dist(A_{i1},R)$，$Dist(A_{i2},R)$
2. 索引统计信息：$I(A_{i1})$：主索引/辅索引/无索引，$HT_i(A_{i1})$，$LB_i(A_{i1})$
3. 中间结果信息：$SF_C(R_i)$,
4. 内存：M（块数）

SQL 查询语句：

1) Select * from R_i where $C_1 \wedge C_2 \wedge \cdots \wedge C_m$
2) Select * from $R_1, R_2 \cdots, R_n$ where C_1 and $\cdots C_m$ and J_1 and $\cdots J_q$ (n<=3)

符号说明：C 表示选择条件，J 表示连接条件。

二. 程序说明

① 输入数据时，各选择条件和连接条件语句（如“A1=20”,“A11=A21”）中，**不等号与等号前后不可出现空格，如请不要输入“A1 = 20”**。From 和 where 之间的**表格名字中，请不要添加空格**，如“R1,R2”请不要输入为“R1 , R2”。From、where、and 等关键词请按照**小写**输入。Sql 语句中**连续的空格数请不要超过两个**。
② 程序默认每个索引结点**可以存放 20 个指针对、50 个不同的数据值**。
③ 在排序-归并连接中，在连接之外、采用的排序代价计算公式为：b(table1) * log(2,b(table1))+b(table2)*log(2,b(table2))。
④ 假设缓冲区只能容纳每个关系的一个块。
⑤ 计算块嵌套循环连接代价时，直接选择以小表作为外表。
⑥ 程序在输出不同表格的选择策略时，会按照输入 sql 语句中“from”和“where”之间各个表格的顺序输出选择策略，详情请见测试用例⑤中说明。

三. 测试说明

*** 所有测试用例均见于附件 txt 文档，可复制到程序中进行测试。**

① Select * from R1 where A1=20 and A2=1200

该测试用例模拟 DB04-1 ppt 中 p47 页中所给的代价估算举例数值。R1 相当于 account-schema，A1 相当于 branch-name，A2 相当于 balance，**由于对表格、属性名的长度有限制**，故简化为 R1,A1 和 A2。

为了计算选择率 SF，查询时只支持以数字界定范围的查询，因此 ppt 样例中的 select account-number from account where branch-name="Perryridge" and balance=120 转化为 Select * from R1 where A1=20 and A2=1200，由两个等值选择构成。

综上，我们在输入数据时输入如下数据：n(R1)=10000, f(R1)=20, Dist(A1,R1)=50, Dist(A2,R1)=500。由于都是等值查询，所以 A1、A2 的上界和下界与结果无关，此处不妨设 A1 下界为 0，上界为 200；A2 下界为 0，上界为 2000。

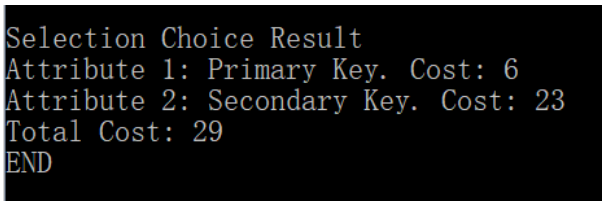
```
Please enter the complete SQL query:
Select * from R1 where A1=20 and A2=1200

Relations and attributes are numbered by the input query order, please enter more infomation.

SET RELATION 1 details:
please enter 'n f' in order:
10000 20
SET ATTRIBUTE 1 details:
please enter Dist of this attribute:
50
please enter Lowest and Highest of this attribute:
0 200
SET ATTRIBUTE 2 details:
please enter Dist of this attribute:
500
please enter Lowest and Highest of this attribute:
0 2000
```

（测试用例①输入数据）

经手动计算，最小代价的查询策略为：
对 A1 采用主索引查询，此时 E(A1)=6。
对 A2 采用辅助索引，则 E(A2)=23，此时总的代价为 29。

注：ppt 第 64 页中的计算，balance 的 HT 取值为 2，故计算出的 E(balance)代价为 28；个人认为此处 balance 的 HT 应当取值为 3：[log(20,500)]=3。

程序所给结果为：

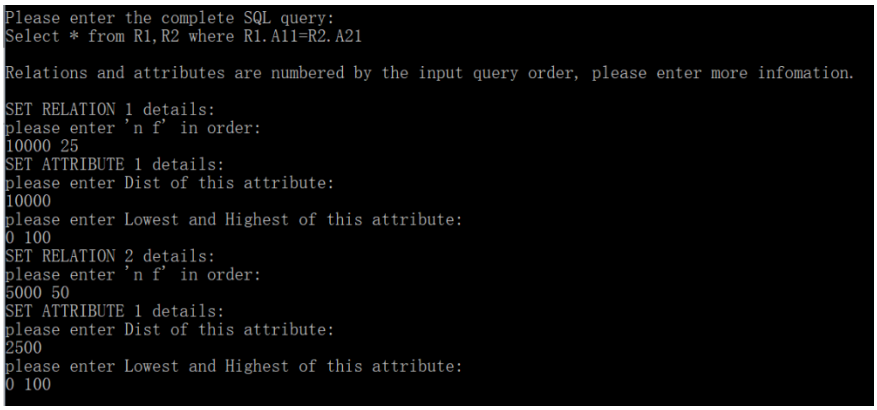


（测试用例①输出结果）

② **Select * from R1,R2 where R1.A11=R2.A21**

该测试用例模拟 Select * from customer,depositor where depositor.customer-name=customer.customer-name。同样，由于表格和属性名的长度受限，此处将名称简化为 R1,R2,A11,A21。

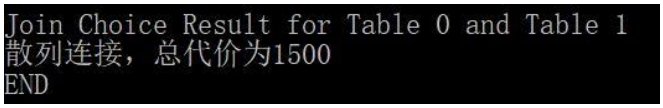
则在输入表格、属性的参数时，有 n(R1)=10000, f(R1)=25, Dist(A11,R1)=10000, A11 下界为 0，上界为 100。n(R2)=5000, f(R2) =50, Dist(A21, R2)=2500, A21 下界为 0，上界为 100。



（测试用例②输入数据）

手动计算各个代价（可见补充作业 9-2）：1) 以 R2 为外表块嵌套连接代价为 40100，2) 以 R1 为外表块嵌套连接代价为 40400，3) 以 A11 为主键进行索引嵌套连接代价为 25100，4) 以 A21 为主键进行索引嵌套连接代价为 50400，5) 排序归并连接代价为 9100，6) 散列连接代价为 **1500**，故最小代价策略为散列连接。

程序所给结果为：



（测试用例②输出结果）

符合手动计算结果。

③ **Select * from R1,R2,R3 where R3.A31=20 and R3.A32=1200 and R1.A11=R2.A21**

此条测试样例集合了①②，R1、R2 相当于②中的 R1、R2，R3 相当于①中的 R1。

输入数据:

n(R1) = 10000, f(R1)=25, D(A11,R1) = 10000, A11 范围为 0-200。

n(R2) = 5000, f(R2)=50, D(A21,R2)=2500, A21 范围为 0-200。

n(R3) = 10000, f(R3)=20, Dist(A31,R3)=50，Dist(A32,R1)=500，A31 范围为 0-200，A32 范围为 0-2000。

```
Please enter the complete SQL query:
Select * from R1,R2,R3 where R3.A31=20 and R3.A32=1200 and R1.A11=R2.A21

Relations and attributes are numbered by the input query order, please enter more infomation.

SET RELATION 1 details:
please enter 'n f' in order:
10000 25
SET ATTRIBUTE 1 details:
please enter Dist of this attribute:
10000
please enter Lowest and Highest of this attribute:
0 200
SET RELATION 2 details:
please enter 'n f' in order:
5000 50
SET ATTRIBUTE 1 details:
please enter Dist of this attribute:
2500
please enter Lowest and Highest of this attribute:
0 200
SET RELATION 3 details:
please enter 'n f' in order:
10000 20
SET ATTRIBUTE 1 details:
please enter Dist of this attribute:
50
please enter Lowest and Highest of this attribute:
0 200
SET ATTRIBUTE 2 details:
please enter Dist of this attribute:
500
please enter Lowest and Highest of this attribute:
0 2000
```

（测试用例③输入数据）

输出结果:

```
Selection Choice Result
Attribute 1: Primary Key. Cost: 6
Attribute 2: Secondary Key. Cost: 23
Total Cost: 29
END

Join Choice Result for Table 0 and Table 1
散列连接，总代价为1500
END
```

（ 测试用例③输出结果 ）

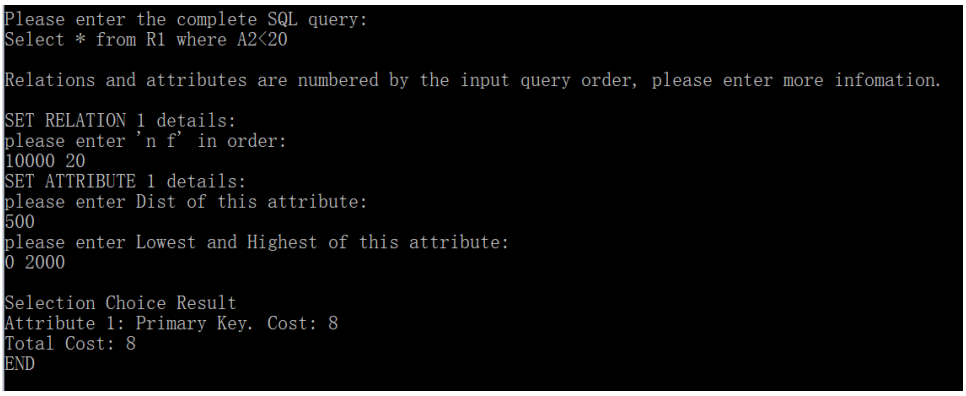
④ **Select * from R1 where A2<20**

此条测试用例为测试涉及**低选择率**情况下的代价计算设计。

参考 ppt DB04-1 第 57 页的设定，A2 相当于 account 中的 balance，在输入数据时输入如下数据：n(R1)=10000, f(R1)=20, Dist(A2,R1)=500，A2 下界为 0，上界为 2000，此时 SF=1%。

手动计算可知，此时若以 A2 为主索引，其代价为 E(A2) = HT+b(A2)*SF = 3+(10000/20)*1% = 8，小于辅助索引代价 103。

程序输出结果符合手动计算结果：



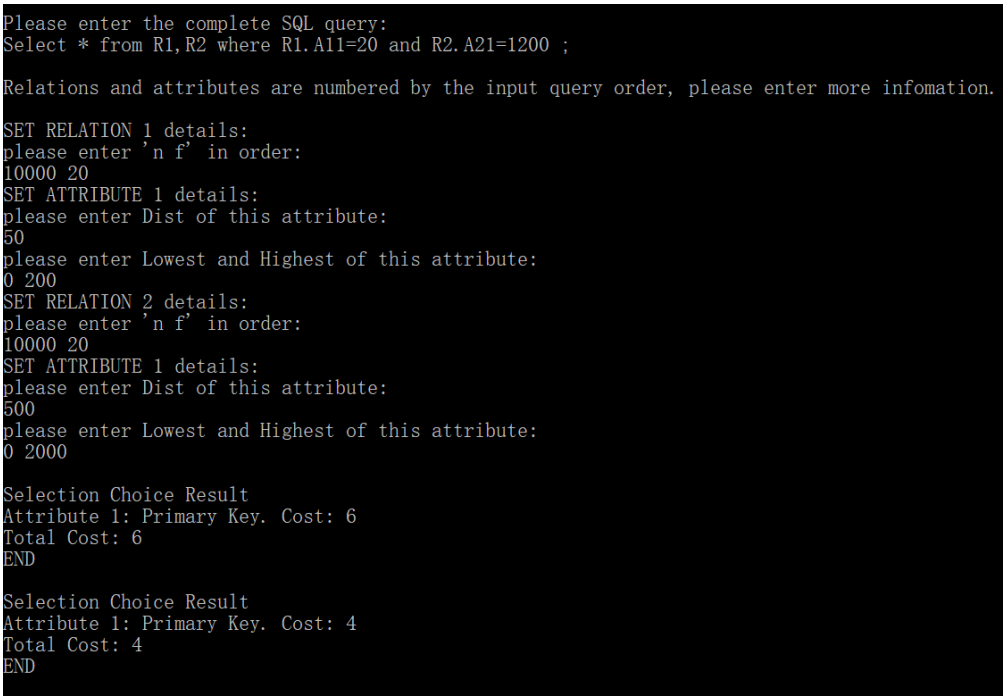
（测试用例④输入输出结果）

⑤ **Select * from R1,R2 where R1.A11=20 and R2.A21=1200 ;**

此条测试用例为了测试**涉及多个表格的选择**设置，同时测试了**对输入 sql 语句末尾有无";" 或空格**的容错性。

输入数据时输入如下数据：
n(R1)=10000, f(R1)=20, Dist(A11,R1)=50，A11 下界为 0，上界为 200。
n(R2)=10000, f(R2)=20, Dist(A21,R1)=500，A21 下界为 0，上界为 2000。

程序输入输出如下：



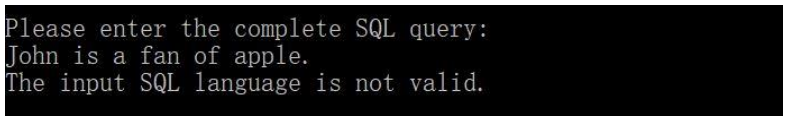
（测试用例⑤输入输出结果）

在上图中，第一个输出的 Result 对应 R1 的选择策略，第二个输出的 Result 对应 R2 的选择策略，输出顺序由 sql 语句中表格名排列顺序决定。

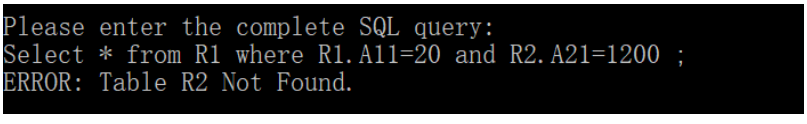
对于 R2 来说，以 A21 为主键的索引选择代价为 E(A21) = HT+SC/f = 3+[(10000/500)/50] = 4，程序输出结果符合手动计算结果。

⑥ 简单错误输入检测

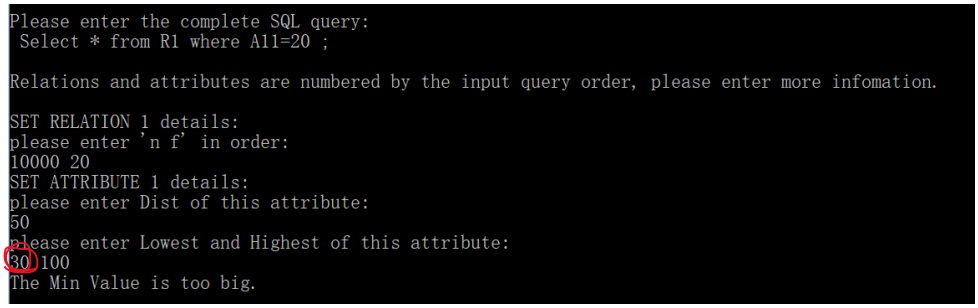
1) **John is a fan of apple.**



2) **Select * from R1 where R1.A11=20 and R2.A21=1200 ;** （R2 没有在表格中出现）



3) **Select * from R1 where A11=20 ;** （属性最大最小值合法性检测，A11=20 小于其最小值 30）


Please enter the complete SQL query:
Select * from R1 where A11=20 ;
Relations and attributes are numbered by the input query order, please enter more infomation.
SET RELATION 1 details:
please enter 'n f' in order:
10000 20
SET ATTRIBUTE 1 details:
please enter Dist of this attribute:
50
please enter Lowest and Highest of this attribute:
30 100
The Min Value is too big.