National University of Computer and Emerging Sciences, Lahore Campus

Course: **Program:**	**Data Warehousing** **BS**	**Course Code:** **Semester:**	
Practice Problem:	**Joining Techniques - SOLUTION**		

Instruction/Notes:

Consider the following tables and statistics which are part of a student system:
Student (RollNo, Name, gpa, DeptID, BatchID, DegreeID, …..); Attendance (RollNo, CourseCode, Semester, AttFlag, …..);

Assume student and attendance tables containing 128,000 and 2,560,000 rows respectively (*Student:Attendance* ratio is *1:20*). Each row and each index entry takes 256 bytes and 16 bytes space respectively. Data block size is 16KB and available memory size is 100 blocks. Suppose degree= 'MS' has a selectivity of 3%, batch= ('2019' or '2018') has a selectivity of (4% + 2%), and dept= ('CS or 'EE') has a selectivity of (10% + 5%).

Query 1:

```
SELECT  COUNT(*)   FROM student  JOIN attendance ON  student.rollno=attendance.rollno
WHERE  DegreeID='MS'  AND  (BatchID='2019' OR BatchID='2018') AND  (DeptID='CS' OR DeptID='EE');
```

Query 2:

```
SELECT  *   FROM student  JOIN attendance ON  student.rollno=attendance.rollno
```

Calculate the total I/O cost (including the I/O cost to filter the condition on student table, if any) for the above Query using the following joining techniques. You are supposed to filter the condition first and then join. Show all steps clearly.

1) Nested Loop Join and variants (i.e. Block NLJ, Indexed NLJ, Clustered Indexed NLJ)
2) Sort Merge Join
3) Hash Join

Ans:

Combine selectivity of student is 3% of (6% of (15% of (128000))) = 35 rows.

K=100; **B**=16384; **R**=256; **R_i**=16; **r_S**=128,000; **r_A**=2,560,000; **bfr**=64 (i.e. B/R=16K/256); **bfr_i**=1024 (i.e. B/R_i=16K/16);

b_S=2000 (i.e. r_S/bfr= 128,000/64); **b_A**=40,000 (i.e. r_A/bfr= 2,560,000/64); **b_{Si}**=125 (i.e. r_S/bfr_i= 128,000/1024); **b_{Ai}**=2500 (i.e. r_A/bfr_i= 2,560,000/1024);

1- Nested Loop Join:

Query 1:

- Basic NLJ: student's filter & read cost + (qualifying rows * attendance blocks) = 2000 + (35 * 40,000) = **1,402,000 blocks**

- Block NLJ: student's filter & read cost + (qualifying blocks * attendance blocks) = 2000 + (1 * 40,000) = **42,000 b**

- Indexed NLJ: student's filter & read cost + (qualifying rows * Inner table index cost only) = 2000 + (35 * 1)

[Avoid inner table access cost]

- Clustered IDX NLJ: student's filter & read cost + (qualifying rows * Inner table index cost only) = 2000 + (35 * 1)

[Avoid inner table access cost]

Query 2:

- Basic NLJ: student's filter & read cost + (qualifying rows * attendance blocks) = 2000 + (128,000 * 40,000)

- Block NLJ: student's filter & read cost + (qualifying blocks * attendance blocks) = 2000 + (2000 * 40,000)

- Indexed (Hash) NLJ: student's filter & read cost + (qualifying rows * (index cost + base table cost)) = 2000 + (128,000 * (1+20))

- Clustered (Hash) IDX NLJ: student's filter & read cost + (qualifying rows * (index cost + base table cost)) = 2000 + (128,000 * (1+1))

2- Sort Merge Join:

Query 1:

student's filter & read cost + (sort student) + (sort attendance) + (merge cost)
= 2000 + (1) + (40,000 * ceil(log 40,000/100)) + (1 + 40,000)
= 2000 + (1) + (40,000 * 9) + (1 + 40,000)
= **402,002 blocks**

Query 2:

student's filter & read cost + (sort student) + (sort attendance) + (merge cost)
= 2000 + (2000 * ceil(log 2000/100)) + (40,000 * ceil(log 40,000/100)) + (2000 + 40,000)
= 2000 + (2000 * 5) + (40,000 * 9) + (2000 + 40,000)

3- Hash Join:

Query 1:

student's filter cost + hashing cost = 2000 + (**1**+ 40,000) = **42,001**

(because hash table fit in memory which requires only one block.)

Query 2:

student's filter & read cost + (partition student) + (partition attendance) + (hashing cost)
= 2000 + (2000 * ceil(log 2000/100)) + (40,000 * ceil(log 2000/100)) + (2000 + 40,000)
= 2000 + (2000 * 5) + (40,000 * 5) + (2000 + 40,000)