National University of Computer and Emerging Sciences, Lahore Campus

	Course:	**Data Warehousing and Business Intelligence**	**Course Code:**	**DS3003**
	Program:	**BS (Data Science)**	**Semester:**	**Fall 2023**
	Date:	**Thu 26-Oct-2023**	**Total Marks:**	**10**
	Section:	**BDS-5A**	**Max. Time:**	**15 min.**
	Quiz:	**4 (Joining Techniques) - SOLUTION**		

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 *one million and 100 million* rows respectively. Each table row and each index entry take *100 bytes* and 10 *bytes* space respectively. Data block size is 32 *KB* and available memory size is 25*0 blocks*. Suppose selectivity of Dept 10=17%, Dept 20= 8%, BatchID 2020= 5%, and BatchID 2021= 3%.

Calculate the total I/O cost for the Query using the following joining techniques. Show all steps clearly. *Assume there is an index on RollNo column of attendance table.*

Query: *SELECT student.RollNo, Name, gpa, DeptID*
FROM student JOIN attendance ON student.RollNo=attendance.RollNo
WHERE DeptID IN (10, 20) AND BatchID IN (2020, 2021);

a. Block Nested Loop Join
b. Sort Merge Join
OR
a. Indexed Nested Loop Join
b. Hash Join

Answer: ***Combine selectivity of student is 25% of (12% off 1,000,000) = 20,000 rows.***

K=250; **B**=32k (i.e. 32,768); **R**=100; $\mathbf{R_i}$=10; **bfr**=328 (i.e. B/R=32k/100); $\mathbf{bfr_i}$=3277 (i.e. B/R_i=32k/10); $\mathbf{r_C}$=1m; $\mathbf{r_S}$=100m;

$\mathbf{b_S}$=3049 (i.e. r_S/bfr= 1m/328); $\mathbf{b_A}$=304,879 (i.e. r_A/bfr= 100m/328);

$\mathbf{b_{Si}}$=305 (i.e. r_S/bfr_i= 1m/3277); $\mathbf{b_{Ai}}$=30,516 (i.e. r_A/bfr_i= 100m/3277);

Block Nested Loop Join
student's filter + (qualifying blocks * base table access cost)
= 3049 + (61 * 304,879) = 18,600,668 I/Os.

Indexed Nested Loop Join
student's filter + (qualifying rows * index access cost)
= 3049 + (20000 * 1) = 23,049 I/Os.
[**Note:** Only index will be access to match corresponding rows of students, so Attendance table access is not required.]

Other option for partial credit:
student's filter + (qualifying rows * (index access cost + attendance table access cost))
= 3049 + (20000 * (1+100)) = 2,023,049 I/Os.

b) Sort Merge Join
student's filter cost + (sort student) + (merge cost) = 3049 + (61) + (61 + **30516**) = **33,687** I/Os.
[**Note**: Index on attendance table (RollNo) will be scan, instead of table scan to merge with student table, which is already sorted.]

Other option for partial credit:

student's filter cost + (sort student) + (sort attendance) + (merge cost)
= 3049 + (61) + (304879 * ceil(log 304879/250)) + (61 + 304879)
= 3049 + (61) + (304879 * 11) + (61 + 304879) = 3,661,719 I/Os.

Hash Join
student's filter cost + hashing cost = 3049 + (61 + 30516) = **33,626** I/Os.
Other option for partial credit:
student's filter cost + hashing cost = 3049 + (61 + 304879) = 307,989 I/Os.