Gretel Rajamoney
rajamong@oregonstate.edu
933188305

Kaavya Subramanian
subramka@oregonstate.edu
933291513

1: Hash Indexing

a. Source code submitted separately through Canvas!

2: Query Processing Algorithms

a. From the given data,
Natural Join Relation R(A, B) = 80,000 blocks
Natural Join Relation S(A, C) = 20,000 blocks
Buffer Blocks available in the Main Memory = 10 buffer blocks
In this case, sort-merge join algorithm with general multi-way merge sort is the fastest join algorithm by utilizing a block based nested loop.
The improved cost of block based nested loop join can be calculated by,
= (B(R) B(S)) / (M)
= (80,000 × 20,000) / (10)
= (1,600,000,000) / (10)
= 160,000,000 I/O Accesses

b. From the given data,
Natural Join Relation R(A, B) = 80,000 blocks
Natural Join Relation S(A, C) = 20,000 blocks
Buffer Blocks available in the Main Memory = 350 buffer blocks
In this case the memory buffer is 350 blocks because,
$B(R) + B(S) < M^2$
$80,000 + 20,000 < 350^2$
80,000 + 20,000 < 122,500
Therefore, the cost of a join is the number of its block I/O Accesses is equal to,
= 3 [B(R) + B(S)]
= 3 (80,000 + 20,000)
= 300,000 I/O Accesses

c. From the given data,
Natural Join Relation R(A, B) = 80,000 blocks

Natural Join Relation S(A, C) = 20,000 blocks
Buffer Blocks available in the Main Memory = 200 buffer blocks
In this case the memory buffer is 200 blocks because,

B(R) + B(S) > M^2

80,000 + 20,000 > 200^2

80,000 + 20,000 > 40,000

Therefore, the cost of a join is the number of its block I/O Accesses is equal to,
= 5 [B(R) + B(S)]
= 5 (80,000 + 20,000)
= 500,000 I/O Accesses

3: Query Processing

a. In this instance, an index nested loop join would be used due to the fact that the cost is only the size of S and is capable of handling both clustered as well as non-clustered indexes.