

- 1) **Explain the process of Hash Join.** How is the hash function used in the join process? What are the phases of hash join?
- 2) **When is Hash Join preferred over Nested Loop Join?** Under what conditions is Hash Join more efficient?
- 3) **What happens in a Hash Join when the smaller relation does not fit into memory?** Explain how **grace hash join** and **recursive partitioning** work.
- 4) **Explain the working of Index Nested Loop Join.** How does the presence of an index improve performance?
- 5) **When should we use Index Nested Loop Join instead of Hash Join?** Compare performance based on data size and indexing availability.
- 6) **How does an Indexed Nested Loop Join differ from a regular Nested Loop Join?** Explain how index lookup reduces the number of comparisons.
- 7) **When does a Nested Loop Join perform better than a Hash Join?** Discuss scenarios where Nested Loop Join is preferable.
- 8) **Why is Nested Loop Join considered inefficient for large datasets?** How does it compare to Sort-Merge and Hash Join?
- 9) **Hash Join Numerical Problems.** Consider two relations R and S: Relation R has 1,000 pages. Relation S has 500 pages. The system has 50 buffer pages. Calculate the I/O cost of performing a **Grace Hash Join** assuming uniform distribution.
- 10) **Consider a Hash Join between two tables:** Table A has 1,000,000 tuples, each 200 bytes. Table B has 500,000 tuples, each 250 bytes. The block size is 8 KB. Estimate the number of disk I/Os required for the join.
- 11) **Index Nested Loop Join Numerical Problems.** Consider a table with the following statistics: Outer relation RRR has 10,000 tuples. Inner relation SSS has 1,000,000 tuples. The index lookup cost is 2 I/Os per tuple. Compute the total cost of Index Nested Loop Join.
- 12) **A database contains two tables:** Table A has 10,000 pages. Table B has 2,000 pages. The database system has 20 buffer pages. Compute the I/O cost for **Block Nested Loop Join**. Compare with the cost of **Simple Nested Loop Join** **Index Nested Loop Join** with **Clustered vs. Unclustered Index**
- 13) Suppose an **Index Nested Loop Join** is being performed on the following dataset:
 - Outer Relation RRR: 100,000 tuples
 - Inner Relation SSS: 500,000 tuples
 - SSS is indexed using:
 - **Clustered index:** Each lookup requires 1 I/O.
 - **Unclustered index:** Each lookup requires 5 I/Os.

Question:

- Compute the I/O cost for both **clustered** and **unclustered index-based joins**.
- Which approach is more efficient?