## Joins

| Sr. No. | Question | Pg. No. |
|---|---|---|
| 1. | What are the different types of SQL joins (INNER, LEFT, RIGHT, FULL, CROSS, etc.) and when would you use each? | |
| 2. | What is the difference between a CROSS JOIN and a FULL OUTER JOIN? | |
| 3. | Write a SQL query to retrieve the first and last names of employees along with the names of their managers (given Employees and Managers tables). | |
| 4. | Write a SQL query to find the average salary for each department, given tables Employees (with DepartmentID) and Departments (with DepartmentName). | |
| 5. | Write a SQL query to list all products that have never been ordered (products in a Product table with no matching rows in the Orders table). | |
| 6. | Write a SQL query to list all employees who are also managers (for example, employees who appear as managers in the same table). | |
| 7. | What is a self-join, and when might you use it? Provide an example scenario. | |
| 8. | How would you join more than two tables in a single SQL query? What factors affect the performance when joining multiple tables? | |
| 9. | Explain how an OUTER JOIN works when one side has no matching rows. How does this differ from an INNER JOIN in practice? | |

## Indexing and Keys

| Sr. No. | Question | Pg. No. |
|---|---|---|
| 1. | What is a SQL index and what are different types of indexes (clustered, non-clustered, unique, etc.)? | 1 |
| 2. | What is the difference between a heap (no clustered index) and a table with a clustered index, and how can you identify a heap table? | 2 |
| 3. | What is the difference between a PRIMARY KEY and a UNIQUE KEY (or unique index) in SQL? | |
| 4. | What are index “forwarding pointers” in a heap table, and how do they affect query performance? | |
| 5. | What is a composite index, and how do you choose the order of columns in it for optimal performance? | |
| 6. | When should you use a covering index, and how does it improve the performance of a query? | |

| Sr. No. | Question | Pg. No. |
|---|---|---|
| 7. | How does the existence of an index on a column affect INSERT, UPDATE, and DELETE performance on a table? | |
| 8. | What is index selectivity, and why is it important for query optimization? | |
| 9. | How many clustered indexes can a table have, and why? | |
| 10. | What is index fragmentation, and how can it be resolved or mitigated in a large database? | |

## Normalization and Schema Design

| Sr. No. | Question | Pg. No. |
|---|---|---|
| 1. | What is database normalization and what are the normal forms (1NF, 2NF, 3NF, BCNF)? | |
| 2. | Given a table with repeating groups of data (e.g. Customer and multiple phone numbers), how would you normalize the table to 3NF? | |
| 3. | Explain how denormalization can be used for performance, and what the trade-offs are (e.g. redundancy vs. speed). | |
| 4. | What kinds of data anomalies (insertion, update, deletion anomalies) are prevented by normalization? | |
| 5. | How would you design a table schema to handle a many-to-many relationship, for example between Students and Courses? | |
| 6. | Explain the concept of denormalization with an example scenario in a large database. | |
| 7. | What is a surrogate key vs a natural key, and when would you choose each in table design? | |
| 8. | How do you implement a 1-to-1 relationship versus a 1-to-many relationship in table design? | |
| 9. | How do you enforce referential integrity without foreign key constraints? (e.g. using triggers) | |
| 10. | How would you handle schema migrations or changes in a production database environment? | |