

Complete SQL Preparation

Basic SQL Queries:

1. How do you retrieve all records from a table?
2. How can you fetch only distinct values in SQL?
3. What is the difference between **WHERE** and **HAVING** clauses?
4. How do you sort the results of a query in ascending or descending order?
5. Explain the **LIMIT** clause and its use.
6. How can you count the number of records in a table?
7. What is the purpose of the **GROUP BY** clause?
8. How do you filter data using multiple conditions?
9. What is the **IN** operator and how is it used?
10. How do you update a specific record in a table?

Joins

1. What are the different types of joins in SQL?
2. Explain the difference between INNER JOIN and OUTER JOIN.
3. How does a LEFT JOIN work?
4. What is a CROSS JOIN and when would you use it?
5. Can you explain the use of a SELF JOIN?
6. What is the difference between UNION and UNION ALL?
7. How can you retrieve matching and non-matching records from two tables?
8. Explain how to join more than two tables.
9. What is the importance of the ON clause in a JOIN?
10. How do you handle null values when performing joins?

Subqueries

1. What is a subquery in SQL?
2. How do you use a subquery in a WHERE clause?
3. Can you explain the difference between a correlated and a non-correlated subquery?
4. How can you return multiple values from a subquery?
5. Explain the use of subqueries in the SELECT clause.
6. What are scalar subqueries and how are they used?
7. How do you perform a DELETE operation using a subquery?
8. What are the limitations of subqueries?
9. Can subqueries be used with JOIN? How?

10. What is the impact of subqueries on performance?

Indexes

1. What is an index in SQL?
2. How do indexes improve query performance?
3. Explain the difference between a clustered and a non-clustered index.
4. How do you create an index on a table?
5. What are the advantages and disadvantages of using indexes?
6. How can you check if an index is being used in a query?
7. Explain the concept of composite indexes.
8. How does indexing affect INSERT, UPDATE, and DELETE operations?
9. When should you avoid using indexes?
10. How do you remove an index from a table?

Transactions

1. What is a transaction in SQL?
2. Explain the ACID properties of a transaction.
3. How do you start and end a transaction?
4. What is the purpose of the COMMIT and ROLLBACK commands?
5. How can you ensure data consistency in transactions?
6. Explain the concept of transaction isolation levels.
7. What is a deadlock and how do you prevent it in transactions?
8. How can you implement a save point in a transaction?
9. What is the difference between explicit and implicit transactions?
10. How do transactions work in a distributed database environment?

Stored Procedures and Functions

1. What is a stored procedure in SQL?
2. How do stored procedures differ from functions?
3. How do you create a stored procedure in SQL?
4. What are the advantages of using stored procedures?
5. How do you pass parameters to a stored procedure?
6. Explain the difference between input and output parameters in stored procedures.
7. How do you handle exceptions in a stored procedure?
8. Can you call a stored procedure within another stored procedure?
9. How do you create and use a user-defined function in SQL?
10. What are the limitations of stored procedures?

Normalization and Database Design

1. What is normalization in database design?
2. Explain the different normal forms with examples.
3. What is denormalization and when is it used?
4. How do you handle many-to-many relationships in database design?
5. What is a primary key and why is it important?
6. Explain the concept of foreign keys and referential integrity.
7. How do you design a database schema for a new application?
8. What are the common pitfalls in database design?
9. How do you ensure scalability in a database design?
10. What is the importance of indexing in database design?