1. Difference between WHERE and HAVING:
   1. When the query has a GROUP BY clause, WHERE can only be used before GROUP BY, while HAVING can only be used after GROUP BY.
   2. We cannot use HAVING without GROUP BY. However, we can use WHERE without GROUP BY.
2. CTE benefits
   1. CTE can be used to create a recursive query.
   2. CTE does not store the definition in metadata.
   3. CTE improves readability and manageability of complex SQL statements.
3. CTE vs. View
   1. CTE is the substitute for a View when the general use of a view is not required.
   2. CTE does not store the definition in metadata, while a view stores the definition in metadata.
4. Union vs. Union all
   1. Union will give the unique records but union all will not.
   2. Union will sort the data based on the first column in the first select statement.
   3. Union all is faster.
   4. Union cannot be used inside a recursive CTE.
5. Join:
   1. **inner join**: will fetch the data from both right and left table which will satisfy the join condition.
   2. **left outer join**: it will bring all the records from the left table but only those records from the right table which will satisfy the join condition. for non-matching records right table will return null value.
   3. **right join**: it will bring all the records from the right table but only those records from the left.
6. Unique constraint vs. Primary key
   1. Unique constraint allows one null value, but primary key does not.
   2. In a table only one primary key is allowed but multiple unique constraints
   3. Primary key will sort the data in asc order by default, but unique key does not do that.
   4. Primary key by default creates the clustered index but unique create non clustered index.
7. Transactions
   1. When transaction1 allows transaction2 to read the uncommitted data and after that transaction1 rollbacks then dirty reads happens
   2. when transaction1 and transaction2 read and modify the same data but transaction2 finishes it work before transaction1 then lost update happens
      1. read uncommitted and read committed isolation level.
   3. non-repeatable read concurrency problem
      1. transaction1 is reading the same data twice but in between transaction2 updates the data so in both reads by transaction1 we will get different results.
   4. Phantom read:
      1. when transaction1 reads the same data twice but transaction2 inserts new data
8. View
   1. **benefits**
      1. it can make complex queries easy.
      2. it can give different result set of the same table.
   2. **Disadvantages**
      1. it cannot accept the parameters (which can cause SQL injection)
      2. it cannot be recursive.
      3. modifying data using view does not give the desired results always if there are multiple base tables.
9. Trigger
   1. Special Tables: inserted, deleted.
10. **Index**
    1. Clustered index is created automatically when a primary key is created, non-clustered index is created when a unique constraint is applied.
    2. A table can have only one clustered index, but it can have multiple non-clustered-255.
    3. A clustered index will by default sort the data in a physical order, but non-clustered index cannot sort the data.
    4. **When to use index?**
       1. when you have multiple rows (millions) and you need to fetch up to 5%-10%
       2. Create index on a column which is frequently used in the where clause.
       3. Create index on a column which can contain multiple null values.
       4. Create index on column with foreign key relationship (those columns which participates the join condition)