1. What is an object in SQL?

   A database object is any defined object in a database that is used to store or reference data. Some examples of database objects include tables, views, clusters, sequences, indexes, and synonyms.

2. What is Index? What are the advantages and disadvantages of using Indexes?

   A SQL index is a quick lookup table for finding records users need to search frequently. An index is small, fast, and optimized for quick lookups. It is very useful for connecting the relational tables and searching large tables.

3. What are the types of Indexes?

   Clustered Index.

   Non-Clustered Index.

   Column Store Index.

   Filtered Index.

   Hash Index.

   Unique Index.

4. Does SQL Server automatically create indexes when a table is created? If yes, under which constraints?
   A unique index is automatically created when you define a primary key or unique constraint: Primary key: When you define a primary key constraint on one or more columns, SQL Server automatically creates a unique, clustered index if a clustered index does not already exist on the table or view.

5. Can a table have multiple clustered index? Why?
   There can be only one clustered index per table, because the data rows themselves can be stored in only one order. The only time the data rows in a table are stored in sorted order is when the table contains a clustered index.
6. Can an index be created on multiple columns? Is yes, is the order of columns matter?
   Yes. A composite index is an index on multiple columns. The order of the columns in the definition should not matter as they won't matter when reading the data.
7. Can indexes be created on views?
   Yes.
8. What is normalization? What are the steps (normal forms) to achieve normalization?
   Database normalization is the process of structuring a database, usually a relational database, in accordance with a series of so-called normal forms in order to reduce data redundancy and improve data integrity.

1NF: An entity type is in 1NF when it contains no repeating groups of data.
2NF: An entity type is in 2NF when it is in 1NF and when all of its non-key attributes are fully dependent on its primary key.
3NF: An entity type is in 3NF when it is in 2NF and when all of its attributes are directly dependent on the primary key.

9. What is denormalization and under which scenarios can it be preferable?

   Denormalization is a strategy used on a previously-normalized database to increase performance. In computing, denormalization is the process of trying to improve the read performance of a database, at the expense of losing some write performance, by adding redundant copies of data or by grouping data.

   Denormalization is used when the faster search is more important and to optimize the read performance.

10. How do you achieve Data Integrity in SQL Server?
    Data integrity is the maintenance of, and the assurance of, data accuracy and consistency over its entire life-cycle[1] and is a critical aspect to the design, implementation, and usage of any system that stores, processes, or retrieves data.
11. What are the different kinds of constraints that SQL Server has?

- Not Null Constraint
- Check Constraint
- Default Constraint
- Unique Constraint
- Primary Constraint
- Foreign Constraint

12. What is the difference between Primary Key and Unique Key?
    Unique key can be null, primary key cannot.

13. What is foreign key?
    A FOREIGN KEY is a field (or collection of fields) in one table that refers to the PRIMARY KEY in another table.
14. Can a table have multiple foreign keys?
    Yes. Each foriegn key can have a different parent table.
15. Does a foreign key have to be unique? Can it be null?
    Does not need to be unique. Yes, it can be null.

16. Can we create indexes on Table Variables or Temporary Tables?

    Table variables cannot be indexed by create index.

    But temp tables can .

17. What is Transaction? What types of transaction levels are there in SQL Server?

    A transaction is a logical unit of work that contains one or more SQL statements. A transaction is an atomic unit. The effects of all the SQL statements in a transaction can be either all committed (applied to the database) or all rolled back (undone from the database).

    transaction isolation levels:
    Read uncommitted
    Read committed
    Repeatable read
    Serializable