

1. What is index; types of indices; pros and cons

a) index: an on-disk structure associated with a table that increase retrieval speed of rows from the table

b) clustered index, non-clustered index

c) Pros: index will help us improve retrieving speed -- improve performance of select

Cons: slow down other DML statements, extra space

--when to create index

--clustered index: always necessary, go with pk

--non clustered index: WHERE, JOIN condition, Aggregated fields

2. What's the difference between Primary key and Unique constraint?

a) unique key can accept one and only one null value; pk cannot accept any null values

b) one table can have multiple unique keys, but only one pk

c) pk will sort the data by default, but unique key will not

3. Tell me about check constraint

limit the value range that can be placed in the column

4. Difference between temp table and table variable

--1. Storage: both Temp Tables and Table Variables are stored in tempDb

--2. Scope: Temp Tables scoped local/ ; Table Variables scoped for current batch

--3. Temp Tables for large data >100 rows; Table Variables for smaller data <100 rows

--4. Usage: Temp Tables cannot be used in Stored Procedure, Functions; Table Variables can be used in SP/function. `

--5. Structure: Temp Tables can create index/ constraints except foreign key; Table Variables cannot

5. Difference between WHERE and HAVING

- a) HAVING apply only to groups as a whole, and only filters on aggregation functions; WHERE applies to individual rows
- b) WHERE goes before aggregations, but HAVING filters after the aggregations
- c) WHERE can used with SELECT UPDATE OR DELETE, but HAVING will only be used in SELECT

6. Difference between RANK() and DenseRank() — value gap

If there is a tie, RANK() function will be a gap for the next rank, while DENSE_RANK() will not skip any number

7. COUNT(*) vs. COUNT(colName)

Count(*) will include null values, but count(column name) will not include null values

8. What's the difference between left join and inner join? JOIN and Subquery, which one has a

better performance, why?

- a) INNER JOIN: return the records that have matching values in both tables, while LEFT OUTER JOIN: return all records from the left table,
and the matching records from the right table,
for the non-matching records in the right table, the result set will return us null values
- b) usually join will have a better performance than subquery. Because join has built in optimizer in SQL

9. What is correlated subquery

inner query is dependent on the outer query

10. What is a CTE, why do we need CTE?

CTE stands for common table expression, a temporary named result set that you can reference within a SELECT, INSERT, UPDATE, or DELETE statement

CTE enable users to write and maintain complex queries via increased readability and simplification more easily

11. What does SQL Profiler do?

SQL Server Profiler is an interface to create and manage traces and analyze and replay trace results.

Events are saved in a trace file that can later be analyzed or used to replay a specific series of steps when diagnosing a problem

12. What is SQL injection, how to avoid SQL injection?

SQL injection is a code injection technique that might destroy your database.

A SQL injection attack consists of insertion or “injection” of a SQL query via the input data from the client to the application

We can use Properly Constructed Stored Procedures and Allow-list Input Validation to avoid SQL injection

13. Difference between SP and user defined function? When to use SP when to use function?

- a) how to call: sp is called by its name, function will be called in SQL query
- b) output: sp may or maynot have output, but function must return some values
- c) SP can call function but function cannot call sp

Usage: sp for DML, function for calculations

14. Criteria of Union and Union all? Difference between UNION and UNION ALL

- 1) The num of columns must be the same
 - column types must be identical
 - alias must be given in the first SELECT statement
- 2) UNION remove all duplicate records, but UNION ALL will not
 - UNION sort the first column ascendingly, but UNION ALL will not
 - UNION cannot be used in recursive cte, but UNION ALL can be used in cte

15. Steps you take to improve SQL Queries

- 1. look at the execution plan; 2. choose index wisely; 3. avoid unnecessary joins; 4. avoid SELECT *; 5. JOIN to replace subquery; 6. derived table to avoid a lot of grouping by

16. concurrency problem in transaction

1. dirty reads:

t1 allows t2 to read uncommitted data and then t1 rolled back

caused by isolation level read uncommitted

solved by isolation level read committed

2. lost update

t1 and t2 read and update the same data but t2 finish its work earlier than t1, then t2 will lost their update

caused by isolation level read committed

solved by isolation level repeatable read

3. non repeatable read

t1 read the same data twice while t2 is updating the data

caused by isolation level read committed

solved by isolation level repeatable read

4. phantom read

t1 reads the same data twice while t2 is inserting records

caused by isolation level repeatable read

solved by isolation level serializable

17. what is deadlock, how to prevent

A deadlock occurs when two (or more) processes lock the separate resource.

Try to keep transactions short

Access objects in a similar logical manner in multiple transactions

Create a covering index to reduce the possibility of a deadlock

Set deadlock priorities using the SET DEADLOCK_PRIORITY session variable

Utilize the error handling using the try-catch blocks.

Change the isolation level to the READ COMMITTED SNAPSHOT ISOLATION or SNAPSHOT ISOLATION

18. what is normalization, 1NF - BCNF, benefits using normalization

Database Normalization is a process of organizing data to minimize redundancy (data duplication), which in turn ensures data consistency.

First Normal Form: One cell, one value, No repeating groups

Second Normal Form: First Normal Form + No Partial Dependency

Third Normal Form: Second Normal Form + No Transitive Dependency

BCNF: Stricter version of 3NF

19. what are the system defined databases?

Master, Model, Msdb and Tempdb database

20. composite key

A composite key in SQL can be defined as a combination of multiple columns, and these columns are used to identify all the rows that are involved uniquely

21. candidate key

Candidate key is a single key or a group of multiple keys that uniquely identify rows in a table

22. DDL vs. DML vs. DCL

Data definition language (DDL): Allows creation objects in database with: Create, Alter, Drop

Data Control Language (DCL): allows you to determine who can see or modify the data:
GRANT, DENY, REVOKE

Data Manipulation Language (DML): Allows query and modify the data: Select, Insert, Update, Delete.

23. ACID property

A: Atomicity -- work is atomic

C: Consistency -- whatever happens in the middle of the transaction, this property will never leave your db in half-completed state

I: Isolation -- two transactions will be isolated from each other by locking the resource

D: Durability -- once the transaction is completed, then the changes it has made to the db will be permanent

24. what is transaction?

-transaction: a group of logically related DML statements that will either succeed together or fail together

-three mode of transactions:

--auto commit transaction: default

--implicit transaction

--explicit transaction

25. table scan vs. index scan

Table scan means iterate over all table rows. Index scan means iterate over all index items (clustered index)

26. Difference between Union and JOIN

joins combine data into new columns, Unions combine data into new rows.

JOIN combines data from many tables based on a matched condition between them, SQL combines the result-set of two or more SELECT statements

Number of columns and datatypes of corresponding columns selected from each table may not be same,

Number of columns and datatypes of corresponding columns selected from each table should be same.