Manish Jaiswal

SQL Interview Question

MASTERING

Non-Clustered Index in SQL

swipe

Manish Jaiswal

SQL Interview Question

01

What is a Non-Clustered Index?

A **non-clustered** index is an index in SQL that improves query speed by creating a separate structure from the actual data. It contains pointers to rows in the data table instead of storing the data itself, as a clustered index does.

swipe

02

When to Use

- Ideal for columns that are frequently used in **WHERE** clauses, **JOIN** operations, or **ORDER BY** clauses.
- Helpful when working with searches on specific columns that don’t require the entire row’s data, e.g., retrieving a list of emails or names.
- It’s especially effective for tables that experience a lot of reads but fewer writes.

swipe

Manish Jaiswal

SQL Interview Question

03

How to Create

```sql
CREATE NONCLUSTERED INDEX idx_employee_name
ON Employees (LastName);
```

This index is created on the **LastName** column of the **Employees table**, which will speed up searches on this column without altering the table's structure.

swipe

Manish Jaiswal

SQL Interview Question

04

Performance Impact 🚀

- **Boosts Query Performance:** Accelerates read operations by providing quick access to row locations in the table.
- **Increases Storage Requirements:** Since non-clustered indexes are stored separately from data, they require additional disk space.
- **Potentially Slower Inserts/Updates:** Due to extra maintenance during write operations, as the index needs to be updated with each change.

swipe

Manish Jaiswal

SQL Interview Question

05

Example in Action

Suppose we want to frequently retrieve employees by their **LastName**. Creating a non-clustered index on LastName can reduce search time significantly:

```sql
SELECT EmployeeID, LastName, FirstName
FROM Employees
WHERE LastName = 'Doe';
```

The query engine will use the non-clustered index on LastName, speeding up this search by accessing only relevant rows.

swipe

Manish Jaiswal

SQL Interview Question

06

Interview Questions Variations ?

1. Difference between clustered and non-clustered indexes?
2. How does SQL Server handle multiple non-clustered indexes?
3. When would a non-clustered index hurt performance?
4. How do non-clustered indexes impact insert/update performance?
5. Explain the structure of a non-clustered index.

swipe