**SQL**

**Clustered vs Non Clustered Index**

|  |  |
| --- | --- |
| Clustered Index | Non Clustered Index |
| It defines the physical sorting of data on disk. | It defines the logical sorting of data. |
| There can be only 1 CI per table. | There can be multiple NCI per table. |
| They are faster in fetching data. | They are slower. |
| They do not require any additional memory. | They require additional memory to store the index. |
| Size of CI is large | Small size |
| Primary key automatically creates CI | We need to create this manually. |

**Table Variable vs Temp Table**

|  |  |
| --- | --- |
| Table Variable | Temp Table |
| TV is created in memory. However if the number of rows is large, it may be pushed to tempdb. | TT is created on disk in tempdb. |
| Lifetime of TV is limited to the batch of query execution. They do not exist outside that. | They exist even outside the batch of execution. |
| TV usually don’t have indexes. They can have them by having a primary key or unique constraint. | TT can have indexes |
| They are suitable for small to med size tables. | They are suitable for large tables. |
| TV are faster as they are not involved in transaction, locking and logging. | They are slower. |
| TV are visible only in the routine in which they are created. | TT are visible in child routine too. |
| They do not allow schema modification. | They allow. |
| They can be passed as a parameter to a function or SP. | They cannot be passed. |

**Truncate vs Delete**

|  |  |
| --- | --- |
| Delete | Truncate |
| It is a DML | It is a DDL |
| It is used to delete specific rows | It is used to delete all rows from the table |
| We can rollback data after running delete | We cannot rollback data after truncate |
| It is slower | It is faster |
| In Delete, a tuple is locked before removing it | In truncate, entire table is locked |