#	Question/Problem	Answer/Solution
76	What are the places one has to look for diagnosing performance issues	➤ Check if somebody has done index maintenance with a Fill Factor of the previous index maintenance activity, which was only suitable at that time. ➤ Check the processes running using sp_who2 ➤ Check slow running queries, queries in blocking sessions , check the isolation level used in those queries. ➤ Check Index fragmentation ➤ Check page splits (especially reads get slowed down due to them) ➤ Index structure (like absence of INCLUDE on the non-clustered index) ➤ Database partitioning on that table(s) ➤ Check any network issues by involving Infrastructure team ➤ Check the database files and file groups are created properly (especially for large databases) ➤ If everything is fine , then check the Setting Collation s , which are most likely to be different from the legacy systems (if migration has been recently done)
77	What are the solutions that can be applied for	Query Tuning ➤ Index Defragmentation ➤ Database Partitioning

	improving performance.	➢ Choosing appropriate 'isolation level' in the T-SQL code. ➢ Introducing Fillfactor to reduce page splits ➢ Introduce Identity column to almost totally remove page splits. ➢ Index Rebuild or Reorganize ➢ Make the collation same as the old system (if migration has been done) ➢ Remove indexes that are not required
78	What is isolation level ?	An isolation level is a T-SQL-transacional-level data read-consistency control mechanism. It spans the T-SQL session unless changed to a different isolation level. Command to set an isolation level in a query session **SET TRANSACTION ISOLATION LEVEL** Types of Isolation levels : **READ UNCOMMITTED** : A query in current transaction can read data that is written by another transaction even though not yet committed. Causes **dirty reads** as the current transaction can read data that is updated, deleted but never committed. As the data being accessed by the current transaction can be modified by the other transaction, it can cause **non-repeatable reads or phantom reads.**

Causes **dirty reads, non-repeatable reads or phantom reads.**

READ COMMITTED : A query in current transaction cannot read data being modified by another transaction that is not yet committed.Can see only committed data.

Avoids - dirty reads

Non-repeatable reads, phantom reads still possible as other transactions between issuing statements in the current transaction can still modify the data committed.

READ COMMITTED is the default isolation used by SQL server.

REPEATABLE READ

Query in current transaction cannot read data being modified by other transaction which is not yet committed which **avoids dirty reads**. Moreover, no other transaction can modify the data being read by the current transaction , **eliminating the non-repeatable reads.**However , if a query in between the issuing statements in the current transaction has to do a second read with a matching search condition , **it may cause phantom reads .**

SERIALIZABLE

Query in current transaction cannot read data being modified by other transaction which is not yet committed which **avoids dirty reads**. Moreover, no other transaction can modify the data being read by the current transaction ,

		eliminating the non-repeatable reads. Also no other transaction , that has a matching search condition same as that of the current transaction , can modify the data until the current transaction completes , **eliminating the phantom reads** . This isolation level has biggest impact on performance than any other level. **SNAPSHOT** Snapshot isolation works with a snapshot of the data set Being read by the current transaction , by keeping completely isolated from other until completion. Other transactions can modify the original data and cannot work with data set of the current transaction and neither can the current transaction read the modifications on the original recordset . This too eliminates all the three issues above. However, it is susceptible to concurrent update errors. (not ANSI/ISO SQL standard) Suggested reading https://www.simple-talk.com/sql/t-sql-programming/questions-about-t-sql-transaction-isolation-levels-you-were-too-shy-to-ask/
79	What is a resource database ?	From SQL server 2005 onwards, Microsoft introduced an **invisible , read-only system database called Resource Database.** Resource database physically contains all system objects like sys objects table , but these objects are

only logically present in the sys schema of all the other databases. It's a very critical system database as SQL server service is dependent on it.

Advantages

1. Enables rollback of system objects upgrades. Prior to 2005 , if upgrades to system objects had to be rolled back , the only was to uninstall SQL server and reinstall it , and then recover the databases.From 2005 onwards, the backup of the resource database's data (mssqlsystemresource.mdf) and log files (mssqlsystemresource.ldf)are enough to rollback the upgrade.Just replicate the copied versions prior to the upgrades will do the job.
2. Enables fast upgrade . If many instances of SQL have to be upgraded with the same service pack , the DBA has to upgrade one instance and then copy the latest files of the resource database from this instance and simply replace the older files in all other instances to upgrade all the instances.

Other information

sp_helpdb doesn't list the resource database

It cannot be backed up using the backup database command . only way is to copy its mdf and ldf and use if required.

If the resource db files are missing or renamed , sql server will not start.

| | | In clustered environment , resource DB files are on /Data folder in a shared drive.

Command to check version of the resource DB

```sql
SELECT
SERVERPROPERTY('ResourceVersion')
Resource_DB_Version,
SERVERPROPERTY('ResourceLastUpdateDateTime') Resource_DB_LastUpdate_DateTime

GO
``` |
|---|---|---|
| 80 | Difference between restore and recover DB? | Restoration is a physical act of copying backup files to the system and registering them with sql server .Example , restoring a database from a fullbackup , and apply any trans logs and restore to a point in time.

Recovery is the operation of Rolling Forward or Rolling Back any transactions to get the database to a consistent state from No-Recovery mode to a Recovery mode in which the database becomes online and accepts connection.

Recovery is also done by SQL server instance when it restarts after a brutal shutdown and then recovers the database to a consistent state by rolling forward committed transactions and rolling back those uncommitted. |
| 81 | Locks in SQL Server | Locks are used to implement isolation and regulate read/write concurrency among transactions in SQL server. |

		Lock Properties

Lock Granularity (scope)

  Row lock - Data row

  Key - Index key

  Page – Entire 8k page

Extent lock - extent lock (64K = 8 pages)

Table - Entire table

Database locks

Lock Manager – controls , monitors , adjust regulates locks

Lock Modes

Shared :

Update

Exclusive

Intent Lock

Lock Duration

The duration of a Lock is influenced by the isolation level set for the transaction. |
| | | Programming in SQL Servers – Strongly suggested |

While loops

Ex:

```
USE AdventureWorks2012;

GO

WHILE (SELECT AVG(ListPrice) FROM
Production.Product) < $300

BEGIN

   UPDATE Production.Product

      SET ListPrice = ListPrice * 2

   SELECT MAX(ListPrice) FROM
Production.Product

   IF (SELECT MAX(ListPrice) FROM
Production.Product) > $500

      BREAK

   ELSE

      CONTINUE

END
```

```
PRINT 'Too much for the market to
bear';
```

IF statements

Table variables

Table Variable is a declared variable in SQL server, of type Table and have well defined scope after which they are automatically cleared by SQL server , where as temp tables have storage structure in the tempdb(hence physical IO) and have a life span of the session.

https://www.youtube.com/watch?v=MdVd0fI1s-A

User Defined Functions

Those defined by user . Also return scalar values.Cannot call a stored procedure.

https://www.youtube.com/watch?v=6BslHItOTjU

Table Valued functions

Functions that return a result-set of query to the calling statement.

Types :

Inline TVF : returns the result of a simple T-SQL select.

MultiStatementTVF : Has multiple insert stmts in its definition.

https://www.youtube.com/watch?v=nCAEgNxC7nU

Common Table Expressions (CTE)

A CTE is a collection of one or more **named Select** statements , the result set of which can used by other queries.

https://www.youtube.com/watch?v=U0wXjUi2v_U

Inner Joins and Outer Joins

https://www.youtube.com/watch?v=MJv6ZQlK_ek

Inner Joins ,

Join between two tables on a common column , to get the intersectional-area data as in a venn diagram.

Outer Join

Join to get that data also which doesn't meet criteria on join condition.

Left Outer join gets all the data from the LHS table and only the join-condition-criteria values from RHS table. For the non-existent cell values, it assigns NULL values in the result set as no value exists for them in the RHS table.

Right Outer Join

Exactly opposite of the Left Outer Join.

82	Does the ordering of columns in table definition matter ?	Yes , it depends, in rare cases. Column order can matter in rare corner cases. For example, if you have a table with 3 columns A, B, and C, each 3kb bytes long. SQL Server pages are 8kb, so C doesn't fit, and goes into its own extended page. So select A, B from YourTable` requires only half the page reads of select A, C from YourTable. Best order : Fixed-Length(Numerical) /Variable Length / Null Columns. Doesn't matter if there is no cluster on the table (Heap Table).
83	Can we run profiler on production system ?	Not recommended by Microsoft as its resource consuming. The key is to know what you are looking for: we don't tend to leave it running and trap everything. Filtering has to be done carefully when running on a highly loaded server , in which case running server side tracing is strongly recommended (see sp_trace_XXX procedures). Recommended reading http://technet.microsoft.com/en-us/library/bb630354(v=sql.105).aspx http://sqlblog.com/blogs/linchi_shea/archive/2007/08/01/trace-profiler-test.aspx

		Automating server side tracing
		http://vyaskn.tripod.com/server_side_tracing_in_sql_server.htm
		Avoiding problems with SQL profiler
		http://weblogs.sqlteam.com/dang/archive/2007/12/16/Avoid-Causing-Problems-with-Profiler.aspx
84	Can an explicit value be insert in identity column	Yes , if the IDENTITY_INSERT is set to ON.
85	What is the difference between Table_Variable and Temporary Table	Table Variable is a declared variable in SQL server, of type Table and have well defined scope after which they are automatically cleared by SQL server , where as temp tables have storage structure in the tempdb and have a life span of the session. Advantages As the Table Variables last till the duration of the update on them , they

		required less locking and logging resources. As the Table variable doesn't belong to persistent database and have very limited scope, transaction rollbacks do not affect them. Disadvantages of Table Var Cannot use a Table Var directly in 'Select into Stmt…' which is possible in case of Temp tables. Non-Clustered Indexes cannot be created on Table Var. Unlike Temp tables , table var do not maintain statistics . This will make the query optimizer difficult to create the best query execution plan which can impact the performance. Check_constraints,Default types in the table definition cannot call User_Defined_Functions. Note : Table Var IS NOT a memory-only structure. Though it resides and processed in the memory like temp tables and other variables , it might need space on the disk for accommodating the records in it.