What is normalization? Explain different levels of normalization?

Check out the article Q100139 from Microsoft knowledge base and of course, there's much more information available in the net. It'll be a good idea to get a hold of any RDBMS fundamentals text book, especially the one by C. J. Date. Most of the times, it will be okay
if you can explain till third normal form.

What is denormalization and when would you go for it?

As the name indicates, denormalization is the reverse process of normalization. It's the controlled introduction of redundancy in to the database design. It helps improve the query performance as the number of joins could be reduced.

How do you implement one-to-one, one-to-many and many-to-many relationships while designing tables?

One-to-One relationship can be implemented as a single table and rarely as two tables with primary and foreign key relationships. One-to-Many relationships are implemented by splitting the data into two tables with primary key and foreign key relationships. Many-to-Many relationships are implemented using a junction table with the keys from both the tables forming the composite primary key of the junction table.

It will be a good idea to read up a database designing fundamentals text book.

What's the difference between a primary key and a unique key?

Both primary key and unique enforce uniqueness of the column on which they are defined. But by default primary key creates a clustered index on the column, where are unique creates a nonclustered index by default. Another major difference is that, primary key doesn't allow NULLs, but unique key allows one NULL only.

What are user defined datatypes and when you should go for them?

User defined datatypes let you extend the base SQL Server datatypes by providing a descriptive name, and format to the database. Take for example, in your database, there is a column called Flight_Num which appears in many tables. In all these tables it should be varchar(8).
In this case you could create a user defined datatype called Flight_num_type of varchar(8) and use it across all your tables.

See sp_addtype, sp_droptype in books online.

What is bit datatype and what's the information that can be stored inside a bit column?

Bit datatype is used to store boolean information like 1 or 0 (true or false). Untill SQL Server 6.5 bit datatype could hold either a 1 or 0 and there was no support for NULL. But from SQL Server 7.0 onwards, bit datatype can represent a third state, which is NULL.

Define candidate key, alternate key, composite key.

A candidate key is one that can identify each row of a table uniquely. Generally a candidate key becomes the primary key of the table. If the table has more than one candidate key, one of them will become the primary key, and the rest are called alternate keys.

A key formed by combining at least two or more columns is called composite key.

What are defaults? Is there a column to which a default can't be bound?

A default is a value that will be used by a column, if no value is supplied to that column while inserting data. IDENTITY columns and timestamp columns can't have defaults bound to them. See CREATE DEFUALT in books online.

What is a transaction and what are ACID properties?

A transaction is a logical unit of work in which, all the steps must be performed or none. ACID stands for Atomicity, Consistency, Isolation, Durability. These are the properties of a transaction. For more information and explanation of these properties, see SQL Server books online or any RDBMS fundamentals text book.

Explain different isolation levels

An isolation level determines the degree of isolation of data between concurrent transactions. The default SQL Server isolation level is Read Committed. Here are the other isolation levels (in the ascending order of isolation): Read Uncommitted, Read Committed, Repeatable Read, Serializable. See SQL Server books online for an explanation of the isolation levels. Be sure to read about SET TRANSACTION ISOLATION LEVEL, which lets you customize the isolation level at the connection level.

CREATE INDEX myIndex ON myTable(myColumn)
What type of Index will get created after executing the above statement?

Non-clustered index. Important thing to note: By default a clustered index gets created on the primary key, unless specified otherwise.

What's the maximum size of a row?

8060 bytes. Don't be surprised with questions like 'what is the maximum number of columns per table'. Check out SQL Server books online for the page titled: "Maximum Capacity Specifications".

Explain Active/Active and Active/Passive cluster configurations

Hopefully you have experience setting up cluster servers. But if you don't, at least be familiar with the way clustering works and the two clusterning configurations Active/Active and Active/Passive. SQL Server books online has enough information on this topic and there is a good white paper available on Microsoft site.

Explain the architecture of SQL Server

This is a very important question and you better be able to answer it if consider yourself a DBA. SQL Server books online is the best place to read about SQL Server architecture. Read up the chapter dedicated to SQL Server Architecture.

What is lock escalation?

Lock escalation is the process of converting a lot of low level locks (like row locks, page locks) into higher level locks (like table locks). Every lock is a memory structure too many locks would mean, more memory being occupied by locks. To prevent this from happening, SQL Server escalates the many fine-grain locks to fewer coarse-grain locks. Lock escalation threshold was definable in SQL Server 6.5, but from SQL Server 7.0 onwards it's dynamically managed by SQL Server.

What's the difference between DELETE TABLE and TRUNCATE TABLE commands?

DELETE TABLE is a logged operation, so the deletion of each row gets logged in the transaction log, which makes it slow. TRUNCATE TABLE also deletes all the rows in a table, but it won't log the deletion of each row, instead it logs the deallocation of the data pages of the table, which makes it faster. Of course, TRUNCATE TABLE can be rolled back.

Explain the storage models of OLAP

Check out MOLAP, ROLAP and HOLAP in SQL Server books online for more infomation.

What are the new features introduced in SQL Server 2000 (or the latest release of SQL Server at the time of your interview)? What changed between the previous version of SQL Server and the current version?

This question is generally asked to see how current is your knowledge. Generally there is a section in the beginning of the books online titled "What's New", which has all such information. Of course, reading just that is not enough, you should have tried those things to better answer the questions. Also check out the section titled "Backward Compatibility" in books online which talks about the changes that have taken place in the new version.

What are constraints? Explain different types of constraints.

Constraints enable the RDBMS enforce the integrity of the database automatically, without needing you to create triggers, rule or defaults.

Types of constraints: NOT NULL, CHECK, UNIQUE, PRIMARY KEY, FOREIGN KEY

For an explanation of these constraints see books online for the pages titled: "Constraints" and "CREATE TABLE", "ALTER TABLE"

What is an index? What are the types of indexes? How many clustered indexes can be created on a table? I create a separate index on each column of a table. what are the advantages and disadvantages of this approach?

Indexes in SQL Server are similar to the indexes in books. They help SQL Server retrieve the data quicker.

Indexes are of two types. Clustered indexes and non-clustered indexes. When you craete a clustered index on a table, all the rows in the table are stored in the order of the clustered index key. So, there can be only one clustered index per table. Non-clustered indexes have their own storage separate from the table data storage. Non-clustered indexes are stored as B-tree structures (so do clustered indexes), with the leaf level nodes having the index key and it's row locater. The row located could be the RID or the Clustered index key, depending up on the absence or presence of clustered index on the table.

If you create an index on each column of a table, it improves the query performance, as the query optimizer can choose from all the existing indexes to come up with an efficient execution plan. At the same t ime, data modification operations (such as INSERT, UPDATE, DELETE) will become slow, as every time data changes in the table, all the indexes need to be updated. Another disadvantage is that, indexes need disk space, the more indexes you have, more disk space is used.

What is RAID and what are different types of RAID configurations?

RAID stands for Redundant Array of Inexpensive Disks, used to provide fault tolerance to database servers. There are six RAID levels 0 through 5 offering different levels of performance, fault tolerance. MSDN has some information about RAID levels and for detailed information, check out the RAID advisory board's homepage

What are the steps you will take to improve performance of a poor performing query?

This is a very open ended question and there could be a lot of reasons behind the poor performance of a query. But some general issues that you could talk about would be: No indexes, table scans, missing or out of date statistics, blocking, excess recompilations of stored procedures, procedures and triggers without SET NOCOUNT ON, poorly written query with unnecessarily complicated joins, too much normalization, excess usage of cursors and temporary tables.

Some of the tools/ways that help you troubleshooting performance problems are: SET SHOWPLAN_ALL ON, SET SHOWPLAN_TEXT ON, SET STATISTICS IO ON, SQL Server Profiler, Windows NT /2000 Performance monitor, Graphical execution plan in Query Analyzer.

Download the white paper on performance tuning SQL Server from Microsoft web site. Don't forget to check out sql-server-performance.com

What are the steps you will take, if you are tasked with securing an SQL Server?

Again this is another open ended question. Here are some things you could talk about: Preferring NT authentication, using server, databse and application roles to control access to the data, securing the physical database files using NTFS permissions, using an unguessable SA password, restricting physical access to the SQL Server, renaming the Administrator account on the SQL Server computer, disabling the Guest account, enabling auditing, using multiprotocol encryption, setting up SSL, setting up firewalls, isolating SQL Server from the web server etc.

Read the white paper on SQL Server security from Microsoft website. Also check out My SQL Server security best practices

What is a deadlock and what is a live lock? How will you go about resolving deadlocks?

Deadlock is a situation when two processes, each having a lock on one piece of data, attempt to acquire a lock on the other's piece. Each process would wait indefinitely for the other to release the lock, unless one of the user processes is terminated. SQL Server detects deadlocks and terminates one user's process.

A livelock is one, where a request for an exclusive lock is repeatedly denied because a series of overlapping shared locks keeps interfering. SQL Server detects the situation after four denials and refuses further shared locks. A livelock also occurs when read transactions monopolize a table or page, forcing a write transaction to wait indefinitely.

Check out SET DEADLOCK_PRIORITY and "Minimizing Deadlocks" in SQL Server books online. Also check out the article Q169960 from Microsoft knowledge base.

What is blocking and how would you troubleshoot it?

Blocking happens when one connection from an application holds a lock and a second connection requires a conflicting lock type. This forces the second connection to wait, blocked on the first.

Read up the following topics in SQL Server books online: Understanding and avoiding blocking, Coding efficient transactions.

Explain CREATE DATABASE syntax

Many of us are used to craeting databases from the Enterprise Manager or by just issuing the command: CREATE DATABAE MyDB. But what if you have to create a database with two filegroups, one on drive C and the other on drive D with log on drive E with an initial size of 600 MB and with a growth factor of 15%? That's why being a DBA you should be familiar with the CREATE DATABASE syntax. Check out SQL Server books online for more information.

How to restart SQL Server in single user mode? How to start SQL Server in minimal configuration mode?

SQL Server can be started from command line, using the SQLSERVR.EXE. This EXE has some very important parameters with which a DBA should be familiar with. -m is used for starting SQL Server in single user mode and -f is used to start the SQL Server in minimal confuguration mode. Check out SQL Server books online for more parameters and their explanations.

As a part of your job, what are the DBCC commands that you commonly use for database maintenance?

DBCC CHECKDB, DBCC CHECKTABLE, DBCC CHECKCATALOG, DBCC CHECKALLOC, DBCC SHOWCONTIG, DBCC SHRINKDATABASE, DBCC SHRINKFILE etc. But there are a whole load of DBCC commands which are very useful for DBAs.
Check out SQL Server books online for more information.

What are statistics, under what circumstances they go out of date, how do you update them?

Statistics determine the selectivity of the indexes. If an indexed column has unique values then the selectivity of that index is more, as opposed to an index with non-unique values. Query optimizer uses these indexes in determining whether to choose an index or not while executing a query.

Some situations under which you should update statistics:
1) If there is significant change in the key values in the index
2) If a large amount of data in an indexed column has been added, changed, or removed (that is, if the distribution of key values has changed), or the table has been truncated using the TRUNCATE TABLE statement and then repopulated
3) Database is upgraded from a previous version

Look up SQL Server books online for the following commands: UPDATE STATISTICS, STATS_DATE, DBCC SHOW_STATISTICS, CREATE STATISTICS, DROP STATISTICS, sp_autostats, sp_createstats, sp_updatestats

What are the different ways of moving data/databases between servers and databases in SQL Server?

There are lots of options available, you have to choose your option depending upon your requirements. Some of the options you have are: BACKUP/RESTORE, dettaching and attaching databases, replication, DTS, BCP, logshipping, INSERT...SELECT, SELECT...INTO, creating INSERT scripts to generate data.

Explian different types of BACKUPs avaialabe in SQL Server? Given a particular scenario, how would you go about choosing a backup plan?

Types of backups you can create in SQL Sever 7.0+ are Full database backup, differential database backup, transaction log backup, filegroup backup. Check out the BACKUP and RESTORE commands in SQL Server books online. Be prepared to write the commands in your interview. Books online also has information on detailed backup/restore architecture and when one should go for a particular kind of backup.

What is database replication? What are the different types of replication you can set up in SQL Server?

Replication is the process of copying/moving data between databases on the same or different servers. SQL Server supports the following types of replication scenarios:

* Snapshot replication
* Transactional replication (with immediate updating subscribers, with queued updating subscribers)
* Merge replication

See SQL Server books online for indepth coverage on replication. Be prepared to explain how different replication agents function, what are the main system tables used in replication etc.

How to determine the service pack currently installed on SQL Server?

The global variable @@Version stores the build number of the sqlservr.exe, which is used to determine the service pack installed. To know more about this process visit SQL Server service packs and versions.

What are cursors? Explain different types of cursors. What are the disadvantages of cursors? How can you avoid cursors?

Cursors allow row-by-row prcessing of the resultsets.

Types of cursors: Static, Dynamic, Forward-only, Keyset-driven. See books online for more information.

Disadvantages of cursors: Each time you fetch a row from the cursor, it results in a network roundtrip, where as a normal SELECT query makes only one rowundtrip, however large the resultset is. Cursors are also costly because they require more resources and temporary storage (results in more IO operations). Furthere, there are restrictions on the SELECT statements that can be used with some types of cursors.

Most of the times, set based operations can be used instead of cursors. Here is an example:

If you have to give a flat hike to your employees using the following criteria:

Salary between 30000 and 40000 -- 5000 hike
Salary between 40000 and 55000 -- 7000 hike
Salary between 55000 and 65000 -- 9000 hike

In this situation many developers tend to use a cursor, determine each employee's salary and update his salary according to the above formula. But the same can be achieved by multiple update statements or can be combined in a single UPDATE statement as shown below:

```
UPDATE tbl_emp SET salary =
CASE WHEN salary BETWEEN 30000 AND 40000 THEN salary + 5000
WHEN salary BETWEEN 40000 AND 55000 THEN salary + 7000
WHEN salary BETWEEN 55000 AND 65000 THEN salary + 10000
END
```

Another situation in which developers tend to use cursors: You need to call a stored procedure when a column in a particular row meets certain condition. You don't have to use cursors for this. This can be achieved using WHILE loop, as long as there is a unique key to identify each row. For examples of using WHILE loop for row by row processing, check out the 'My code library' section of my site or search for WHILE.

Write down the general syntax for a SELECT statements covering all the options.

Here's the basic syntax: (Also checkout SELECT in books online for advanced syntax).

SELECT select_list
[INTO new_table_]
FROM table_source
[WHERE search_condition]
[GROUP BY group_by__expression]
[HAVING search_condition]
[ORDER BY order__expression [ASC | DESC]]

What is a join and explain different types of joins.

Joins are used in queries to explain how different tables are related. Joins also let you select data from a table depending upon data from another table.

Types of joins: INNER JOINs, OUTER JOINs, CROSS JOINs. OUTER JOINs are further classified as LEFT OUTER JOINS, RIGHT OUTER JOINS and FULL OUTER JOINS.

For more information see pages from books online titled: "Join Fundamentals" and "Using Joins".

Can you have a nested transaction?

Yes, very much. Check out BEGIN TRAN, COMMIT, ROLLBACK, SAVE TRAN and @@TRANCOUNT

What is an extended stored procedure? Can you instantiate a COM object by using T-SQL?

An extended stored procedure is a function within a DLL (written in a programming language like C, C++ using Open Data Services (ODS) API) that can be called from T-SQL, just the way we call normal stored procedures using the EXEC statement. See books online to learn how to create extended stored procedures and how to add them to SQL Server.

Yes, you can instantiate a COM (written in languages like VB, VC++) object from T-SQL by using sp_OACreate stored procedure. Also see books online for sp_OAMethod, sp_OAGetProperty, sp_OASetProperty, sp_OADestroy. For an example of creating a COM object in VB and calling it from T-SQL, see 'My code library' section of this site.

What is the system function to get the current user's user id?

USER_ID(). Also check out other system functions like USER_NAME(), SYSTEM_USER, SESSION_USER, CURRENT_USER, USER, SUSER_SID(), HOST_NAME().

What are triggers? How many triggers you can have on a table? How to invoke a trigger on demand?

Triggers are special kind of stored procedures that get executed automatically when an INSERT, UPDATE or DELETE operation takes place on a table.

In SQL Server 6.5 you could define only 3 triggers per table, one for INSERT, one for UPDATE and one for DELETE. From SQL Server 7.0 onwards, this restriction is gone, and you could create multiple triggers per each action. But in 7.0 there's no way to control the order in which the triggers fire. In SQL Server 2000 you could specify which trigger fires first or fires last using sp_settriggerorder

Triggers can't be invoked on demand. They get triggered only when an Associated action (INSERT, UPDATE, DELETE) happens on the table on which they are defined.

Triggers are generally used to implement business rules, auditing. Triggers can also be used to extend the referential integrity checks, but wherever possible, use constraints for this purpose, instead of triggers, as constraints are much faster.

Till SQL Server 7.0, triggers fire only after the data modification operation happens. So in a way, they are called post triggers. But in SQL Server 2000 you could create pre triggers also. Search SQL Server 2000 books online for INSTEAD OF triggers.

Also check out books online for 'inserted table', 'deleted table' and COLUMNS_UPDATED()

There is a trigger defined for INSERT operations on a table, in an OLTP system. The trigger is written to instantiate a COM object and pass the newly insterted rows to it for some custom processing. What do you think of this implementation? Can this be implemented better?

Instantiating COM objects is a time consuming process and since you are doing it from within a trigger, it slows down the data insertion process. Same is the case with sending emails from triggers. This scenario can be better implemented by logging all the necessary data into a separate table, and have a job which periodically checks this table and does the needful.

What is a self join? Explain it with an example.

Self join is just like any other join, except that two instances of the same table will be joined in the query. Here is an example:
Employees table which contains rows for normal employees as well as managers. So, to find out the managers of all the employees, you need a self join.

```
CREATE TABLE emp
(
empid int,
mgrid int,
empname char(10)
)

INSERT emp SELECT 1,2,'Vyas'
INSERT emp SELECT 2,3,'Mohan'
INSERT emp SELECT 3,NULL,'Shobha'
INSERT emp SELECT 4,2,'Shridhar'
INSERT emp SELECT 5,2,'Sourabh'

SELECT t1.empname [Employee], t2.empname [Manager]
FROM emp t1, emp t2
WHERE t1.mgrid = t2.empid
```

Here's an advanced query using a LEFT OUTER JOIN that even returns the employees without managers (super bosses)

```
SELECT t1.empname [Employee], COALESCE(t2.empname, 'No manager') [Manager]
FROM emp t1
LEFT OUTER JOIN
emp t2
ON
t1.mgrid = t2.empid
```