# What is SQL and where does it come from?

Structured Query Language (SQL) is a language that provides an interface to relational database systems. The proper pronunciation of SQL is "ess cue ell," and not "sequel" as is commonly heard.

SQL was developed by IBM in the 1970s for use in System R, and is a de facto standard, as well as an ISO and ANSI standard.

In common usage SQL also encompasses DML (Data Manipulation Language), for INSERTs, UPDATEs, DELETEs and DDL (Data Definition Language), used for creating and modifying tables and other database structures.

The development of SQL is governed by standards. A major revision to the SQL standard was completed in 1992, called SQL2. SQL3 support object extensions and are (partially?) implemented in Oracle8 and 9i.

# What are the difference between DDL, DML and DCL commands?

**DDL**

DDL (Data Definition Language) statements are used to define the database structure or schema. Some examples:

* CREATE - to create objects in the database
* ALTER - alters the structure of the database
* DROP - delete objects from the database
* TRUNCATE - remove all records from a table, including all spaces allocated for the records are removed
* COMMENT - add comments to the data dictionary
* RENAME - rename an object

**DML**

DML (Data Manipulation Language) statements are used for managing data within schema objects. Some examples:

* SELECT - retrieve data from the a database
* INSERT - insert data into a table
* UPDATE - updates existing data within a table
* DELETE - deletes all records from a table, the space for the records remain
* MERGE - UPSERT operation (insert or update)
* CALL - call a PL/SQL or Java subprogram
* EXPLAIN PLAN - explain access path to data
* LOCK TABLE - control concurrency

**DCL**

DCL is Data Control Language statements. Some examples:

* GRANT - gives user's access privileges to database
* REVOKE - withdraw access privileges given with the GRANT command

**Transaction Control**

Manages the changes made by DML statements. These commands allow statements to be grouped together into logical transactions.

* COMMIT - save work done
* SAVEPOINT - identify a point in a transaction to which you can later roll back
* ROLLBACK - restore database to original since the last COMMIT
* SET TRANSACTION - Change transaction options like what rollback segment to use

# Difference between TRUNCATE, DELETE and DROP commands

**DELETE**

The DELETE command is used to remove rows from a table. A WHERE clause can be used to only remove some rows. If no WHERE condition is specified, all rows will be removed. After performing a DELETE operation you need to COMMIT or ROLLBACK the transaction to make the change permanent or to undo it.

SQL> SELECT COUNT(\*) FROM emp;

COUNT(\*)

----------

14

SQL> DELETE FROM emp WHERE job = 'CLERK';

4 rows deleted.

SQL> COMMIT;

Commit complete.

SQL> SELECT COUNT(\*) FROM emp;

COUNT(\*)

----------

10

**TRUNCATE**

TRUNCATE removes **all rows** from a table. The operation cannot be rolled back. As such, TRUCATE is faster and doesn't use as much undo space as a DELETE.

SQL> TRUNCATE TABLE emp;

Table truncated.

SQL> SELECT COUNT(\*) FROM emp;

COUNT(\*)

----------

0

**DROP**

The DROP command removes a table from the database. All the tables' rows, indexes and privileges will also be removed. The operation cannot be rolled back.

SQL> DROP TABLE emp;

Table dropped.

SQL> SELECT \* FROM emp;

SELECT \* FROM emp

\*

ERROR at line 1:

ORA-00942: table or view does not exist

DROP and TRUNCATE are DDL commands, whereas DELETE is a DML command. Therefore DELETE operations can be rolled back (undone), while DROP and TRUNCATE operations cannot be rolled back.

From Oracle 10g a table can be "undropped". Example:

SQL> FLASHBACK TABLE emp TO BEFORE DROP;

Flashback complete.

PS: DROP and TRUNCATE are DDL commands, whereas DELETE is a DML command. As such, DELETE operations can be rolled back (undone), while DROP and TRUNCATE operations cannot be rolled back.

# How does one escape special characters when writing SQL queries?

**Escape quotes**

Use two quotes for every one displayed. Examples:

SQL> SELECT 'Frank''s Oracle site' AS text FROM DUAL;

TEXT

--------------------

Franks's Oracle site

SQL> SELECT 'A ''quoted'' word.' AS text FROM DUAL;

TEXT

----------------

A 'quoted' word.

SQL> SELECT 'A ''''double quoted'''' word.' AS text FROM DUAL;

TEXT

-------------------------

A ''double quoted'' word.

**Escape wildcard characters**

The LIKE keyword allows for string searches. The '\_' wild card character is used to match exactly one character, while '%' is used to match zero or more occurrences of any characters. These characters can be escaped in SQL. Examples:

SELECT name FROM emp

WHERE id LIKE '%/\_%' ESCAPE '/';

SELECT name FROM emp

WHERE id LIKE '%\%%' ESCAPE '\';

**Escape SQL\*Plus special characters**

When using SQL\*Plus, the DEFINE setting can be changed to allow &'s (ampersands) to be used in text:

SET DEFINE ~

SELECT 'Lorel & Hardy' FROM dual;

# Can one select a random collection of rows from a table?

The following methods can be used to select a random collection of rows from a table:

**The SAMPLE Clause**

From Oracle 8i, the easiest way to randomly select rows from a table is to use the SAMPLE clause with a SELECT statement. Examples:

SELECT \*

FROM emp SAMPLE(10);

In the above example, Oracle is instructed to randomly return 10% of the rows in the table.

SELECT \*

FROM emp SAMPLE(5) BLOCKS;

This example will sample 5% of all formatted database blocks instead of rows.

This clause only works for single table queries on local tables. If you include the SAMPLE clause within a multi-table or remote query, you will get a parse error or "ORA-30561: SAMPLE option not allowed in statement with multiple table references". One way around this is to create an inline view on the driving table of the query with the SAMPLE clause. Example:

SELECT t1.dept, t2.emp

FROM (SELECT \* FROM dept SAMPLE(5)) t1,

emp t2

WHERE t1.dep\_id = t2.dep\_id;

If you examine the execution plan of a "Sample Table Scan", you should see a step like this:

TABLE ACCESS (SAMPLE) OF 'EMP' (TABLE)

**ORDER BY dbms\_random.value()**

This method orders the data by a by a random column number. Example:

SQL> SELECT \* FROM (SELECT ename

2 FROM emp

3 ORDER BY dbms\_random.value())

4 WHERE rownum <= 3

5 /

ENAME

----------

WARD

MILLER

TURNER

**The ORA\_HASH() function**

The following example retrieves a subset of the data in the emp table by specifying 3 buckets (0 to 2) and then returning the data from bucket 1:

SELECT \* FROM emp WHERE ORA\_HASH(empno, 2) = 1

# How does one eliminate duplicates rows from a table?

Choose one of the following queries to identify or remove duplicate rows from a   
table leaving only unique records in the table:

**Method 1:**

SQL> DELETE FROM table\_name A WHERE ROWID > (

2 SELECT min(rowid) FROM table\_name B

3 WHERE A.key\_values = B.key\_values);

**Method 2:**

SQL> create table table\_name2 as select distinct \* from table\_name1;

SQL> drop table table\_name1;

SQL> rename table\_name2 to table\_name1;

This method is usually faster. However, remember to recreate all indexes, constraints, triggers, etc on the table when done.

**Method 3:**

(contributed by [**Dennis Gurnick**](mailto:dgurnick@otb.com))

SQL> delete from my\_table t1

SQL> where exists (select 'x' from my\_table t2

SQL> where t2.key\_value1 = t1.key\_value1

SQL> and t2.key\_value2 = t1.key\_value2

SQL> and t2.rowid > t1.rowid);

Note: One can eliminate N^2 unnecessary operations by creating an index on the joined fields in the inner loop (no need to loop through the entire table on each pass by a record). This will speed-up the deletion process.

Note 2: If you are comparing NOT-NULL columns, use the NVL function. Remember that NULL is not equal to NULL. This should not be a problem as all key columns should be NOT NULL by definition.

# How does one add a day/hour/minute/second to a date value?

The SYSDATE pseudo-column shows the current system date and time. Adding 1 to SYSDATE will advance the date by 1 day. Use fractions to add hours, minutes or seconds to the date. Look at these examples:

SQL> select sysdate, sysdate+1/24, sysdate +1/1440, sysdate + 1/86400 from dual;

SYSDATE SYSDATE+1/24 SYSDATE+1/1440 SYSDATE+1/86400

-------------------- -------------------- -------------------- --------------------

03-Jul-2002 08:32:12 03-Jul-2002 09:32:12 03-Jul-2002 08:33:12 03-Jul-2002 08:32:13

The following format is frequently used with Oracle Replication:

select sysdate NOW, sysdate+30/(24\*60\*60) NOW\_PLUS\_30\_SECS from dual;

NOW NOW\_PLUS\_30\_SECS

-------------------- --------------------

03-JUL-2005 16:47:23 03-JUL-2005 16:47:53

# How does one add a column to the middle of a table?

Oracle only allows columns to be added to the end of an existing table. Example:

SQL> CREATE TABLE tab1 ( col1 NUMBER );

Table created.

SQL> ALTER TABLE tab1 ADD (col2 DATE);

Table altered.

SQL> DESC tab1

Name Null? Type

----------------------------------------- -------- ----------------------------

COL1 NUMBER

COL2 DATE

Nevertheless, some databases also allow columns to be added to an existing table after a particular column (i.e. in the middle of the table). For example, in MySQL the following syntax is valid:

ALTER TABLE tablename ADD columnname AFTER columnname;

Oracle does not support this syntax. However, it doesn't mean that it cannot be done.

**Workarounds:**

1. Create a new table and copy the data across.

SQL> RENAME tab1 TO tab1\_old;

Table renamed.

SQL> CREATE TABLE tab1 AS SELECT 0 AS col1, col1 AS col2 FROM tab1\_old;

Table created.

2. Use the DBMS\_REDEFINITION package to change the structure on-line while users are workining.

# What is the difference between VARCHAR, VARCHAR2 and CHAR data types?

Both CHAR and VARCHAR2 are used to store character string values, however, they behave differently, while VARCHAR should not be used:

**CHAR**

CHAR is used for fix length character strings. Shorter strings will be space padded before stored.

SQL> CREATE TABLE char\_test (col1 CHAR(10));

Table created.

SQL> INSERT INTO char\_test VALUES ('qwerty');

1 row created.

SQL> SELECT col1, length(col1), dump(col1) "ASCII Dump" FROM char\_test;

COL1 LENGTH(COL1) ASCII Dump

---------- ------------ ------------------------------------------------------------

qwerty 10 Typ=96 Len=10: 113,119,101,114,116,121,32,32,32,32

**VARCHAR**

Currently VARCHAR behaves exactly the same as VARCHAR2.  
However, this type should not be used as it is reserved for future usage.

**VARCHAR2**

Variable character strings.

SQL> CREATE TABLE varchar2\_test (col1 VARCHAR2(10));

Table created.

SQL> INSERT INTO varchar2\_test VALUES ('qwerty');

1 row created.

SQL> SELECT col1, length(col1), dump(col1) "ASCII Dump" FROM varchar2\_test;

COL1 LENGTH(COL1) ASCII Dump

---------- ------------ ------------------------------------------------------------

qwerty 6 Typ=1 Len=6: 113,119,101,114,116,121

# What is PL/SQL and what is it used for?

SQL is a declarative language that allows database programmers to write a SQL declaration and hand it to the database for execution. As such, SQL cannot be used to execute procedural code with conditional, iterative and sequential statements. To overcome this limitation, PL/SQL was created.

PL/SQL is Oracle's Procedural Language extension to SQL. PL/SQL's language syntax, structure and data types are similar to that of [**ADA**](http://www.orafaq.com/glossary/faqglosa.htm). Some of the statements provided by PL/SQL:

**Conditional Control Statements:**

* IF ... THEN ... ELSIF ... ELSE ... END IF;
* CASE ... WHEN ... THEN ... ELSE ... END CASE;

**Iterative Statements:**

* LOOP ... END LOOP;
* WHILE ... LOOP ... END LOOP;
* FOR ... IN [REVERSE] ... LOOP ... END LOOP;

**Sequential Control Statements:**

* GOTO ...;
* NULL;

The PL/SQL language includes object oriented programming techniques such as encapsulation, function overloading, information hiding (all but inheritance).

PL/SQL is commonly used to write data-centric programs to manipulate data in an Oracle database.

# How can one see if somebody modified any code?

The source code for stored procedures, functions and packages are stored in the Oracle Data Dictionary. One can detect code changes by looking at the LAST\_DDL\_TIME column in the USER\_OBJECTS dictionary view. Example:

SELECT OBJECT\_NAME,

TO\_CHAR(CREATED, 'DD-Mon-RR HH24:MI') CREATE\_TIME,

TO\_CHAR(LAST\_DDL\_TIME, 'DD-Mon-RR HH24:MI') MOD\_TIME,

STATUS

FROM USER\_OBJECTS

WHERE LAST\_DDL\_TIME > '&CHECK\_FROM\_DATE';

# Should one use PL/SQL or Java to code procedures and triggers?

Internally the Oracle database supports two procedural languages, namely   
PL/SQL and Java. This leads to questions like "Which of the two is the best?"   
and "Will Oracle ever desupport PL/SQL in favour of Java?".

Many Oracle applications are based on PL/SQL and it would be difficult of   
Oracle to ever desupport PL/SQL. In fact, all indications are that PL/SQL still  
has a bright future ahead of it. Many enhancements are still being made to   
PL/SQL. For example, Oracle 9iDB supports native compilation of Pl/SQL code   
to binaries. Not to mention the numerous PL/SQL enhancements made in Oracle 10g.

PL/SQL and Java appeal to different people in different job roles. The   
following table briefly describes the difference between these two language   
environments:

**PL/SQL:**

* Data centric and tightly integrated into the database
* Proprietary to Oracle and difficult to port to other database systems
* Data manipulation is slightly faster in PL/SQL than in Java
* Easier to use than Java (depending on your background)

**Java:**

* Open standard, not proprietary to Oracle
* Incurs some data conversion overhead between the Database and Java type systems
* Java is more difficult to use (depending on your background)

# Can one read/write files from PL/SQL?

The UTL\_FILE database package can be used to read and write operating system files.

A DBA user needs to grant you access to read from/ write to a specific directory before using this package. Here is an example:

CONNECT / AS SYSDBA

CREATE OR REPLACE DIRECTORY mydir AS '/tmp';

GRANT read, write ON DIRECTORY mydir TO scott;

Copy and paste these examples to get you started:

**Write File**

DECLARE

fHandler UTL\_FILE.FILE\_TYPE;

BEGIN

fHandler := UTL\_FILE.FOPEN('MYDIR', 'myfile', 'w');

UTL\_FILE.PUTF(fHandler, 'Look ma, I''m writing to a file!!!\n');

UTL\_FILE.FCLOSE(fHandler);

EXCEPTION

WHEN utl\_file.invalid\_path THEN

raise\_application\_error(-20000, 'Invalid path. Create directory or set UTL\_FILE\_DIR.');

END;

/

**Read File**

DECLARE

fHandler UTL\_FILE.FILE\_TYPE;

buf varchar2(4000);

BEGIN

fHandler := UTL\_FILE.FOPEN('MYDIR', 'myfile', 'r');

UTL\_FILE.GET\_LINE(fHandler, buf);

dbms\_output.put\_line('DATA FROM FILE: '||buf);

UTL\_FILE.FCLOSE(fHandler);

EXCEPTION

WHEN utl\_file.invalid\_path THEN

raise\_application\_error(-20000, 'Invalid path. Create directory or set UTL\_

FILE\_DIR.');

END;

/

NOTE: UTL\_FILE was introduced with Oracle 7.3. Before Oracle 7.3 the only means of writing a file was to use DBMS\_OUTPUT with the SQL\*Plus SPOOL command.

# Can one call DDL statements from PL/SQL?

One can call DDL statements like CREATE, DROP, TRUNCATE, etc. from PL/SQL by   
using the "EXECUTE IMMEDATE" statement. Users running Oracle versions below 8i   
can look at the DBMS\_SQL package (see FAQ about Dynamic SQL).

begin

EXECUTE IMMEDIATE 'CREATE TABLE X(A DATE)';

end;

NOTE: The DDL statement in quotes should not be terminated with a semicolon.

# I can SELECT from SQL\*Plus but not from PL/SQL. What is wrong?

PL/SQL respect object privileges given directly to the user, but does not   
observe privileges given through roles. The consequence is that a SQL statement   
can work in SQL\*Plus, but will give an error in PL/SQL.  
Choose one of the following solutions:

* Grant **direct** access on the tables to your user. Do not use roles!

GRANT select ON scott.emp TO my\_user;

* Define your procedures with invoker rights (Oracle 8i and higher);
* create or replace procedure proc1
* authid current\_user is
* begin

...

* Move all the tables to one user/schema.

# What is the difference between %TYPE and %ROWTYPE?

Both %TYPE and %ROWTIPE are used to define variables in PL/SQL as it is defined within the database. If the datatype or precision of a column changes, the program automically picks up the new definition from the database without having to make any code changes.

The %TYPE and %ROWTYPE constructs provide data independence, reduces maintenance costs, and allows programs to adapt as the database changes to meet new business needs.

**%TYPE**

%TYPE is used to declare a ***field*** with the same type as that of a specified table's column. Example:

DECLARE

v\_EmpName emp.ename%TYPE;

BEGIN

SELECT ename INTO v\_EmpName FROM emp WHERE ROWNUM = 1;

DBMS\_OUTPUT.PUT\_LINE('Name = ' || v\_EmpName);

END;

/

**%ROWTYPE**

%ROWTYPE is used to declare a ***record*** with the same types as found in the specified database table, view or cursor. Examples:

DECLARE

v\_emp emp%ROWTYPE;

BEGIN

v\_emp.empno := 10;

v\_emp.ename := 'XXXXXXX';

END;

/

DECLARE

v\_EmpRecord emp%ROWTYPE;

BEGIN

SELECT \* INTO v\_EmpRecord FROM emp WHERE ROWNUM = 1;

DBMS\_OUTPUT.PUT\_LINE('Name = ' || v\_EmpRecord.ename);

DBMS\_OUTPUT.PUT\_LINE('Salary = ' || v\_EmpRecord.sal);

END;

/

**DB Interview Questions**

1. **What is SQL?**
2. **What is SELECT statement?**
3. **How can you compare a part of the name rather than the entire name?**
4. **What is the INSERT statement?**
5. **How do you delete a record from a database?**
6. **How could I get distinct entries from a table?**
7. **How to get the results of a Query sorted in any order?**
8. **How can I find the total number of records in a table?**
9. **What is GROUP BY?**
10. **What is the difference among "dropping a table", "truncating a table" and "deleting all records" from a table?**
11. **What are the Large object types suported by Oracle?**
12. **Difference between a "where" clause and a "having" clause ?**
13. **What's the difference between a primary key and a unique key?**
14. **What are cursors? Explain different types of cursors. What are the disadvantages of cursors? How can you avoid cursors?**
15. **What are triggers? How to invoke a trigger on demand?**
16. **What is a join and explain different types of joins.**
17. **What is a self join?**

**Q: What is SQL?**

**A:** SQL stands for 'Structured Query Language'.

**Q:** **What is SELECT statement?**

**A:** The SELECT statement lets you select a set of values from a table in a database. The values selected from the database table would depend on the various conditions that are specified in the SQL query.

**Q:** **How can you compare a part of the name rather than the entire name?**

**A:** SELECT \* FROM people WHERE empname LIKE '%ab%'  
Would return a recordset with records consisting empname the sequence '**ab**' in empname.

| **Q:** | **What is the INSERT statement?** |
| --- | --- |
| **A:** | The INSERT statement lets you insert information into a database. |
|  | **TOPtop** |

| **Q:** | **How do you delete a record from a database?** |
| --- | --- |
| **A:** | Use the DELETE statement to remove records or any particular column values from a database. |
|  | **TOPtop** |

| **Q:** | **How could I get distinct entries from a table?** |
| --- | --- |
| **A:** | The SELECT statement in conjunction with DISTINCT lets you select a set of distinct values from a table in a database. The values selected from the database table would of course depend on the various conditions that are specified in the SQL query. Example *SELECT DISTINCT empname FROM emptable* |
|  | **TOPtop** |

| **Q:** | **How to get the results of a Query sorted in any order?** |
| --- | --- |
| **A:** | You can sort the results and return the sorted results to your program by using ORDER BY keyword thus saving you the pain of carrying out the sorting yourself. The ORDER BY keyword is used for sorting.  *SELECT empname, age, city FROM emptable ORDER BY empname* |
|  | **TOPtop** |

| **Q:** | **How can I find the total number of records in a table?** |
| --- | --- |
| **A:** | You could use the COUNT keyword , example  *SELECT COUNT(\*) FROM emp WHERE age>40* |
|  | **TOPtop** |

| **Q:** | **What is GROUP BY?** |
| --- | --- |
| **A:** | The GROUP BY keywords have been added to SQL because aggregate functions (like SUM) return the aggregate of all column values every time they are called. Without the GROUP BY functionality, finding the sum for each individual group of column values was not possible. |
|  | **TOPtop** |

| **Q:** | **What is the difference among "dropping a table", "truncating a table" and "deleting all records" from a table**. |
| --- | --- |
| **A:** | **Dropping :**  (Table structure  + Data are deleted), Invalidates the dependent objects ,Drops the indexes  **Truncating:**  (Data alone deleted), Performs an automatic commit, Faster than delete  **Delete :** (Data alone deleted), Doesn’t perform automatic commit |
|  | **TOPtop** |

| **Q:** | **What are the Large object types suported by Oracle?** |
| --- | --- |
| **A:** | Blob and Clob. |
|  | **TOPtop** |

| **Q:** | **Difference between a "where" clause and a "having" clause.** |
| --- | --- |
| **A:** | Having clause is used only with group functions whereas Where is not used with. |
|  | **TOPtop** |

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

**A:** 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.

**TOPtop**

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

**A:** 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.

**TOPtop**

**Q:** **What are triggers? How to invoke a trigger on demand?**

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

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.

| **Q:** | **What is a join and explain different types of joins.** |
| --- | --- |

**A:** 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.

**Q:** **What is a self join?**

**A:** Self join is just like any other join, except that two instances of the same table will be joined in the query.

GENERAL INTERVIEW QUESTIONS

**1.What are the various types of Exceptions ?**  
User defined and Predefined Exceptions.  
**2.Can we define exceptions twice in same block ?**  
No.  
**3.What is the difference between a procedure and a function ?**  
Functions return a single variable by value whereas procedures do not return any variable by value. Rather they return multiple variables by passing variables by reference through their OUT parameter.  
**4.Can you have two functions with the same name in a PL/SQL block ?**  
Yes.  
**5.Can you have two stored functions with the same name ?**  
Yes.  
**6.Can you call a stored function in the constraint of a table ?**  
No.  
**7.What are the various types of parameter modes in a procedure ?**  
IN, OUT AND INOUT.  
**8.What is Over Loading and what are its restrictions ?**  
OverLoading means an object performing different functions depending upon the no. of parameters or the data type of the parameters passed to it.  
**9.Can functions be overloaded ?**  
Yes.  
**10.Can 2 functions have same name & input parameters but differ only by return datatype**   
No.  
**11.What are the constructs of a procedure, function or a package ?**  
The constructs of a procedure, function or a package are :   
variables and constants   
cursors   
exceptions  
**12.Why Create or Replace and not Drop and recreate procedures ?**  
So that Grants are not dropped.  
**13.Can you pass parameters in packages ? How ?**  
Yes. You can pass parameters to procedures or functions in a package.  
**14.What are the parts of a database trigger ?**  
The parts of a trigger are:   
A triggering event or statement   
A trigger restriction   
A trigger action  
**15.What are the various types of database triggers ?**  
There are 12 types of triggers, they are combination of :  
Insert, Delete and Update Triggers.  
Before and After Triggers.  
Row and Statement Triggers.  
(3\*2\*2=12)   
**16.What is the advantage of a stored procedure over a database trigger ?**  
We have control over the firing of a stored procedure but we have no control over the firing of a trigger.  
**17.What is the maximum no. of statements that can be specified in a trigger statement ?**  
One.  
**18.Can views be specified in a trigger statement ?**  
No  
**19.What are the values of :new and :old in Insert/Delete/Update Triggers ?**  
INSERT : new = new value, old = NULL  
DELETE : new = NULL, old = old value  
UPDATE : new = new value, old = old value   
**20.What are cascading triggers? What is the maximum no of cascading triggers at a time?**  
When a statement in a trigger body causes another trigger to be fired, the triggers are said to be cascading. Max = 32.  
**21.What are mutating triggers ?**  
A trigger giving a SELECT on the table on which the trigger is written.  
**22.What are constraining triggers ?**  
A trigger giving an Insert/Updat e on a table having referential integrity constraint on the triggering table.  
**23.Describe Oracle database's physical and logical structure?**   
Physical : Data files, Redo Log files, Control file.  
Logical : Tables, Views, Tablespaces, etc.  
**24.Can you increase the size of a tablespace ? How?**   
Yes, by adding datafiles to it.  
**25.Can you increase the size of datafiles ? How?**   
No (for Oracle 7.0)  
Yes (for Oracle 7.3 by using the Resize clause ----- Confirm !!).  
**26.What is the use of Control files?**   
Contains pointers to locations of various data files, redo log files, etc.  
**27.What is the use of Data Dictionary?**   
Used by Oracle to store information about various physical and logical Oracle structures e.g. Tables, Tablespaces, datafiles, etc  
**28.What are the advantages of clusters?**   
Access time reduced for joins.

**29.What are the disadvantages of clusters?**   
The time for Insert increases.  
**30.Can Long/Long RAW be clustered?**   
No.  
**31.Can null keys be entered in cluster index, normal index?**   
Yes.  
**32.Can Check constraint be used for self referential integrity ? How ?**  
Yes. In the CHECK condition for a column of a table, we can reference some other column of the same table and thus enforce self referential integrity.  
**33.What are the min. extents allocated to a rollback extent ?**  
Two  
**34.What are the states of a rollback segment ? What is the difference between partly available and needs recovery ?**  
The various states of a rollback segment are :  
ONLINE, OFFLINE, PARTLY AVAILABLE, NEEDS RECOVERY and INVALID.  
**35.What is the difference between unique key and primary key ?**  
Unique key can be null; Primary key cannot be null.  
**36.An insert statement followed by a create table statement followed by rollback ? Will the rows be inserted ?**  
No.  
**37.Can you define multiple savepoints ?**Yes.  
**38.Can you Rollback to any savepoint ?**  
Yes.  
**40.What is the maximum no. of columns a table can have ?**  
254.  
**41.What is the significance of the & and && operators in PL SQL ?**  
The & operator means that the PL SQL block requires user input for a variable. The && operator means that the value of this variable should be the same as inputted by the user previously for this same variable.  
If a transaction is very large, and the rollback segment is not able to hold the rollback information, then will the transaction span across different rollback segments or will it terminate ?  
It will terminate (Please check ).  
**42.Can you pass a parameter to a cursor ?**  
Explicit cursors can take parameters, as the example below shows. A cursor parameter can appear in a query wherever a constant can appear.   
CURSOR c1 (median IN NUMBER) IS  
SELECT job, ename FROM emp WHERE sal > median;  
**43.What are the various types of RollBack Segments ?**  
Public Available to all instances  
Private Available to specific instance  
**44.Can you use %RowCount as a parameter to a cursor ?**  
Yes  
**45.Is the query below allowed :**  
**Select sal, ename Into x From emp Where ename = 'KING'  
(Where x is a record of Number(4) and Char(15))**  
Yes  
**46.Is the assignment given below allowed :**  
**ABC = PQR (Where ABC and PQR are records)**  
Yes  
**47.Is this for loop allowed :  
For x in &Start..&End Loop**  
Yes  
**48.How many rows will the following SQL return :  
Select \* from emp Where rownum < 10;**  
9 rows  
**49.How many rows will the following SQL return :  
Select \* from emp Where rownum = 10;**  
No rows  
**50.Which symbol preceeds the path to the table in the remote database ?**  
@  
**51.Are views automatically updated when base tables are updated ?**  
Yes  
**52.Can a trigger written for a view ?**  
No  
**53.If all the values from a cursor have been fetched and another fetch is issued, the output will be : error, last record or first record ?**  
Last Record  
**54.A table has the following data : [5, Null, 10]. What will the average function return ?**  
7.5  
**55.Is Sysdate a system variable or a system function?**  
System Function  
**56.Consider a sequence whose currval is 1 and gets incremented by 1 by using the nextval reference we get the next number 2. Suppose at this point we issue an rollback and again issue a nextval. What will the output be ?**  
3  
**56.Definition of relational DataBase by Dr. Codd (IBM)?**  
A Relational Database is a database where all data visible to the user is organized strictly as tables of data values and where all database operations work on these tables.  
**57.What is Multi Threaded Server (MTA) ?**  
In a Single Threaded Architecture (or a dedicated server configuration) the database manager creates a separate process for each database user. But in MTA the database manager can assign multiple users (multiple user processes) to a single dispatcher (server process), a controlling process that queues request for work thus reducing the databases memory requirement and resources.  
**58.Which are initial RDBMS, Hierarchical & N/w database ?**  
RDBMS - R system  
Hierarchical - IMS  
N/W - DBTG  
**59.Difference between Oracle 6 and Oracle 7**  
ORACLE 7 ORACLE 6   
Cost based optimizer · Rule based optimizer   
Shared SQL Area · SQL area allocated for each user   
Multi Threaded Server · Single Threaded Server   
Hash Clusters · Only B-Tree indexing   
Roll back Size Adjustment · No provision   
Truncate command · No provision   
Database Integrity   
Constraints · Provision at Application Level   
Stored procedures, functions   
packages & triggers · No provision   
Resource profile limit. It   
prevents user from running  
away with system resources · No provision   
Distributed Database · Distributed Query   
Table replication & snapshots· No provision   
Client/Server Tech. · No provision   
**60.What is Functional Dependency**  
Given a relation R, attribute Y of R is functionally dependent on attribute X of R if and only if each X-value has associated with it precisely one -Y value in R   
**61.What is Auditing ?**  
The database has the ability to audit all actions that take place within it.   
a) Login attempts, b) Object Accesss, c) Database Action  
Result of Greatest(1,NULL) or Least(1,NULL)  
NULL  
**62.While designing in client/server what are the 2 imp. things to be considered ?**  
Network Overhead (traffic), Speed and Load of client server  
**63.What are the disadvantages of SQL ?**  
Disadvantages of SQL are :  
· Cannot drop a field  
· Cannot rename a field  
· Cannot manage memory  
· Procedural Language option not provided  
· Index on view or index on index not provided  
· View updation problem

**64.When to create indexes ?**  
To be created when table is queried for less than 2% or 4% to 25% of the table rows.  
**65.How can you avoid indexes ?**  
TO make index access path unavailable  
· Use FULL hint to optimizer for full table scan  
· Use INDEX or AND-EQUAL hint to optimizer to use one index or set to   
indexes instead of another.  
· Use an expression in the Where Clause of the SQL.  
**66.What is the result of the following SQL :  
Select 1 from dual  
UNION  
Select 'A' from dual;**  
Error  
**67.Can database trigger written on synonym of a table and if it can be then what would be the effect if original table is accessed.**  
Yes, database trigger would fire.  
**68.Can you alter synonym of view or view ?**  
No  
**69.Can you create index on view**   
No.  
**70.What is the difference between a view and a synonym ?**  
Synonym is just a second name of table used for multiple link of database. View can be created with many tables, and with virtual columns and with conditions. But synonym can be on view.  
**71.What is the difference between alias and synonym ?**  
Alias is temporary and used with one query. Synonym is permanent and not used as alias.  
**72.What is the effect of synonym and table name used in same Select statement ?**  
Valid  
**73.What's the length of SQL integer ?**  
32 bit length  
**74.What is the difference between foreign key and reference key ?**  
Foreign key is the key i.e. attribute which refers to another table primary key.  
Reference key is the primary key of table referred by another table.  
**75.Can dual table be deleted, dropped or altered or updated or inserted ?**  
Yes  
**76.If content of dual is updated to some value computation takes place or not ?**  
Yes  
**77.If any other table same as dual is created would it act similar to dual?**  
Yes  
**78.For which relational operators in where clause, index is not used ?**  
<> , like '% ...' is NOT functions, field +constant, field || ''  
**79.Assume that there are multiple databases running on one machine. How can you switch from one to another ?**  
Changing the ORACLE\_SID  
**80.What are the advantages of Oracle ?**  
Portability : Oracle is ported to more platforms than any of its competitors, running on more than 100 hardware platforms and 20 networking protocols.  
Market Presence : Oracle is by far the largest RDBMS vendor and spends more on R & D than most of its competitors earn in total revenue. This market clout means that you are unlikely to be left in the lurch by Oracle and there are always lots of third party interfaces available.  
Backup and Recovery : Oracle provides industrial strength support for on-line backup and recovery and good software fault tolerence to disk failure. You can also do point-in-time recovery.  
Performance : Speed of a 'tuned' Oracle Database and application is quite good, even with large databases. Oracle can manage > 100GB databases.  
Multiple database support : Oracle has a superior ability to manage multiple databases within the same transaction using a two-phase commit protocol.  
**81.What is a forward declaration ? What is its use ?**  
PL/SQL requires that you declare an identifier before using it. Therefore, you must declare a subprogram before calling it. This declaration at the start of a subprogram is called forward declaration. A forward declaration consists of a subprogram specification terminated by a semicolon.  
**82.What are actual and formal parameters ?**  
Actual Parameters : Subprograms pass information using parameters. The variables or expressions referenced in the parameter list of a subprogram call are actual parameters. For example, the following procedure call lists two actual parameters named emp\_num and amount:  
Eg. raise\_salary(emp\_num, amount);

Formal Parameters : The variables declared in a subprogram specification and referenced in the subprogram body are formal parameters. For example, the following procedure declares two formal parameters named emp\_id and increase:  
Eg. PROCEDURE raise\_salary (emp\_id INTEGER, increase REAL) IS current\_salary REAL;  
**83.What are the types of Notation ?**  
Position, Named, Mixed and Restrictions.  
**84.What all important parameters of the init.ora are supposed to be increased if you want to increase the SGA size ?**  
In our case, db\_block\_buffers was changed from 60 to 1000 (std values are 60, 550 & 3500) shared\_pool\_size was changed from 3.5MB to 9MB (std values are 3.5, 5 & 9MB) open\_cursors was changed from 200 to 300 (std values are 200 & 300) db\_block\_size was changed from 2048 (2K) to 4096 (4K) {at the time of database creation}.  
The initial SGA was around 4MB when the server RAM was 32MB and The new SGA was around 13MB when the server RAM was increased to 128MB.  
**85.If I have an execute privilege on a procedure in another users schema, can I execute his procedure even though I do not have privileges on the tables within the procedure ?**  
Yes  
**86.What are various types of joins ?**  
Equijoins, Non-equijoins, self join, outer join  
**87.What is a package cursor ?**  
A package cursor is a cursor which you declare in the package specification without an SQL statement. The SQL statement for the cursor is attached dynamically at runtime from calling procedures.  
**88.If you insert a row in a table, then create another table and then say Rollback. In this case will the row be inserted ?**  
Yes. Because Create table is a DDL which commits automatically as soon as it is executed. The DDL commits the transaction even if the create statement fails internally (eg table already exists error) and not syntactically.  
**89.What are the various types of queries ?**  
Normal Queries  
Sub Queries  
Co-related queries  
Nested queries  
Compound queries  
**90.What is a transaction ?**  
A transaction is a set of SQL statements between any two COMMIT and ROLLBACK statements.

**91.What is implicit cursor and how is it used by Oracle ?**  
An implicit cursor is a cursor which is internally created by Oracle. It is created by Oracle for each individual SQL.  
**92.Which of the following is not a schema object : Indexes, tables, public synonyms, triggers and packages ?**  
Public synonyms  
**93.What is the difference between a view and a snapshot ?**  
**94.What is PL/SQL?**  
PL/SQL is Oracle's Procedural Language extension to SQL. The language includes object oriented programming techniques such as encapsulation, function overloading, information hiding (all but inheritance), and so, brings state-of-the-art programming to the Oracle database server and a variety of Oracle tools.  
**95.Is there a PL/SQL Engine in SQL\*Plus?**  
No. Unlike Oracle Forms, SQL\*Plus does not have a PL/SQL engine. Thus, all your PL/SQL are send directly to the database engine for execution. This makes it much more efficient as SQL statements are not stripped off and send to the database individually.  
**96.Is there a limit on the size of a PL/SQL block?**  
Currently, the maximum parsed/compiled size of a PL/SQL block is 64K and the maximum code size is 100K. You can run the following select statement to query the size of an existing package or procedure.  
SQL> select \* from dba\_object\_size where name = 'procedure\_name'  
**97.Can one read/write files from PL/SQL?**  
Included in Oracle 7.3 is a UTL\_FILE package that can read and write files. The directory you intend writing to has to be in your INIT.ORA file (see UTL\_FILE\_DIR=... parameter). Before Oracle 7.3 the only means of writing a file was to use DBMS\_OUTPUT with the SQL\*Plus SPOOL command.  
DECLARE  
fileHandler UTL\_FILE.FILE\_TYPE;  
BEGIN  
fileHandler := UTL\_FILE.FOPEN('/home/oracle/tmp', 'myoutput','W');  
UTL\_FILE.PUTF(fileHandler, 'Value of func1 is %s\n', func1(1));  
UTL\_FILE.FCLOSE(fileHandler);  
END;

**98.How can I protect my PL/SQL source code?**  
PL/SQL V2.2, available with Oracle7.2, implements a binary wrapper for PL/SQL programs to protect the source code. This is done via a standalone utility that transforms the PL/SQL source code into portable binary object code (somewhat larger than the original). This way you can distribute software without having to worry about exposing your proprietary algorithms and methods. SQL\*Plus and SQL\*DBA will still understand and know how to execute such scripts. Just be careful, there is no "decode" command available.  
The syntax is:   
wrap iname=myscript.sql oname=xxxx.yyy  
**99.Can one use dynamic SQL within PL/SQL? OR Can you use a DDL in a procedure ? How ?**  
From PL/SQL V2.1 one can use the DBMS\_SQL package to execute dynamic SQL statements.   
Eg: CREATE OR REPLACE PROCEDURE DYNSQL  
AS  
cur integer;  
rc integer;  
BEGIN  
cur := DBMS\_SQL.OPEN\_CURSOR;  
DBMS\_SQL.PARSE(cur,'CREATE TABLE X (Y DATE)', DBMS\_SQL.NATIVE);  
rc := DBMS\_SQL.EXECUTE(cur);  
DBMS\_SQL.CLOSE\_CURSOR(cur);  
END;  
1. What is Referential Integrity rule?  
Differentiate between   
2. Delete & Truncate command.  
3. Implicit Cursor & Explicit Cursor.  
4. Ref. key & Foreign key.  
5. Where & Having Clause.  
6. What are various kinds of Integrity Constraints in Oracle?  
7. What are various kind of joins?  
8. What is Raise\_Application\_Error?  
9. What are various kinds of exceptions in Oracle?  
10. Normal Forms

Oracle Notes :

Oracle 8i -  
It is a DB of internet computing , It changes the of information managed and accessed to meet the demandof internet age.  
-- Significant new feature for OLTP(Online trans Processing) and data ware housing Appl.  
-- To mange all types of datain web site.  
-- iFS Internet file Syatem   
-- interMedia to manage and access multimedia data,audio,video  
-- Support to java(to install JVM on server)  
-- Security enhancement(authentication and authorization,centralizing user management)  
  
Oracle 8(ORDBMS) -  
-Parrallel enhancement ,faster connection  
-Table partitioning , Connection inc to 30000 ,Table column upto 1000  
-DB size inc from few tera byte to 10 tera. , Data file inc 65,533  
-Support MTS,provides LOB   
  
Oracle Start -  
1. Oracle instance start -Allocates SGA and start BAckground processes.  
2. Mount Oracle DB-Method of Associating DB with previous started instance   
3.Opening DB-To make available.

Normalization -  
It's a technique thr. which we can design the DB.  
During normalization dependancies can be identified which can cause pbs during deletion & updation .It is used in simplifing the structure of table.  
1NF-Unnorma;ised data transfer to normalised form.  
2NF-Functional dependancies can be find out & decompose the table without loss of data.  
3NF-Transist dependancies ,Every non key attrbute is functionally dependant on just PK.  
4NF(BCNF)-The relation which has multiple candidate keys ,then we have to go for BCNF.  
Denormalization-  
At the same time when information is required from more than one table at faster rate then it is wiser to add some sort of dependancies.

Rooling Forward -To reapply to Data file to all changes that are recorded in Redo log file due to which datafile contains commited & uncommited dat.  
Forward Declaration-To declare variable and procedures before using it.  
2- Tier Arch. Disadv-When Business Rule changes.  
PL/SQL Record-To represent more than one row at time.  
PL/SQL Table -To define single variable comprises several data element.  
To delete define one more empty table and assign it.  
Tablespace -  
Profile-To control system resources ,memory ,diskspace and CPU time.  
We can find rows effected by %rowcount.  
Data Binding-Dividing the cursor in appl as per select stamt.  
Trancate -Faster than delete ,doesn't fire ny DB trigger ,Allocate space ,No roolback.  
Defered Integrity constraints-When we refere PK in the same table where we defined .  
Cascading trigger-  
Temporary Table-Delete operation table.  
Log Table-to store information abt error.  
Coordinity-  
Err Trap -To trap error use SQLERRM,SQLCODE  
Modularity-PL/SQL allows to create program module to improve software reliability and to hide complexity   
Positional and Named Notation -  
The actual arguments are associaed with formal arguments by position k/s Positional Notation.It's commonly used.  
A Trigger doesn't accept argument & have same name as table or procedure as it exist in seperate namespace.  
How we ref FK in Sql -Join Condition.  
Security/Lock-  
Shared/exclusive -When 2 transaction wants to read/write from db at the same time.  
Dead- 1trans updates emp and dep  
2 trans update dep and emp  
TO add a not null column to a table which has already some records -  
Alter table a  
Add(b number default 1 not null)  
Sequence- Start with,increment by,Cache/No cache,Order/No order,Max,Min  
ER Dia.- Entity Relation Dia.  
Set Transaction -To set a current transaction online offline   
Oracle err-  
ORA-06500 stiorage err  
ORA-00923 from keyword not found  
ORA-06501program err  
ORA-00904 Invalid Col  
ORA-00001Uk violated.  
Dynamic Sql -Which uses late binding   
File I/O-To read and write dat to and from text file thr. Oracle procedure.  
Joins-Equi,Non EQui,Self,inner joins,outer joins  
Index-16 col per table.  
Parsing-Syntax checking.  
Optimization-Use of index (HINT)  
Corelated Subquery -Which fires only once/ per row for entire stmt.  
Simple Query--Which fires everytime for entire stmt  
Packages- Encapsulation,Overloading,improve performance as I/O reduces.  
PL/SQL Signature Method- To determine when remote dependant object get invalid.  
Object Previledge - On a particular object- I/U/D/Exec  
System Previledge -Entire collection object -C/A/D  
SGA Comprises -Data Buffer, Redo Log Buffer,Shared pool Buffer.  
Shared Pool - Req to process unique SQL stmt submitted to DB.  
It contains information such as parse tree and execusion plan .  
PGA -A memry buffer that contains data and control information for a server process.  
Dedicated server - Handles request. for single user.  
Multithresd Server-Handles request. for multiple user.  
Background process -DBWR,LGWR,PMON,SMON,CKPT  
DBWR-Writes modified data blocks from DB buffer to data file.  
LGWR-  
CKPT-Responsible to check DBWR and update control file to indicate most recent CKPT.  
SMON-Instance recovery at start up,Clean Temporary. Segment.  
PMON-Responsible for process recovery and user process fails,Cleaning up cache ,freeing resources which was using process.  
Segment-Data/Index/Rollback/Temp  
Data Dictionary -V$SESSION, information abt integrity constraints,space allocated for schema object.  
USER\_TAB\_COLUMNS gives you a list of tables as per Column.  
EOD Procedure-

Mutating/Constraining Err/Table  
Diff of where and group by  
Connect,Allocate.Analyse Command.  
Queries--  
1. 3rd Max   
select distinct sal  
from emp a  
where 3=(select count(distinct sal)   
from emp b  
where a.sal=<B.SAL)  
2. Delete Duplicate rows  
Delete Emp  
where rowid not in(select max(rowid)   
from emp  
group by emp\_no)  
3. First 5 Max No  
select sal   
from (select abs(-sal) sal  
from emp   
group by -sal)  
where rownum<6  
Views--  
-No Aggr function,group by,having  
-U/D without PK but not Insert.  
-Join -No DML  
-No join-DML  
Index -are used for row selection in where and order by only if indexing on column

You can launch the DBA Studio or the individual tools directly from the Windows NT Start menu. Or, you can use the following syntax to launch them from a command line prompt:

oemapp tool\_name

where tool\_name may be dbastudio, instance, security, storage, schema, or worksheet, if installed.

DBMS\_ALERT is a Transaction Processing Package while DBMS\_PIPE is an Application Development package

Developed By Satish Shrikhande

DBA—

If to\_date(sysdate,'DAY')='Tuesday' then ..  
Buffer Cache-To improve data block recently used by user in order to improve the performance.  
Ordinality-Emp, Expences-Emp may expense sheet and Expense sheet has only one Emp. This fact k/s Referred Ordinality.  
Three Steps in creating DB.--  
-Creating physical location for data in tables and indexes to be stored in DB.  
-To create the file that still store log entries.  
-To create logical structure of data dictionary.   
This is accomplished by create DB   
1. Back up existing DB.  
2.Create or Edit the init.ora file  
3.Varify the instance name   
4. Start Application management DB tool.  
5.start instance  
6.Create and Backup the new DB.   
Control file -250K  
Oracle Administration Assistant for W-NT is a DB management tool that enables to create DB administartor, operator, Users and role. To manage Oracle DB services, DB start up, shut down, Edit registry parameter setting, views oracle process information.  
Database Configuration Assistant -To create DB  
Oracle environment-  
OLTP-Many users can read and update, hight response time.  
DSS-Read only.  
Hybrid-both OLTP & DSS App. are running with this App.  
Init.ora-is a parameter file like DB\_NAME, CONTROL\_FILE, DB\_BLOCK \_SIZE  
RowID-BlockIDRowIdDatafileId  
Cluster Segment-To support use of cluster on the DB.  
Hash Cluster-By placing data in close proximity k/s Hashing.  
Optimization-  
Decides line of execution of query.  
First apply condition and then make Cartesian product. The cost can reduce by reducing no of rows.  
Oracle ways for optimization-  
-Evaluation of expression and condition amt>500/100--amt>5  
Like convert to equal   
IN - OR condition  
Any -OR  
Between/ALL -AND  
NOT-Avoid   
Transitivity-where a.id=b.id and a.id=1 use a.id=1 and b.id =1  
Merging views  
Index column be in order by clause.

Bitmap Index- If the column has very few distinct entries  
We have to specify in init.ora Rate, Cost Choose mode based Approach   
-Avoid full table scan.  
-Access by Rowid  
-No function on Index column as it prevents the optimization.  
-Avoid IN, NOT and LIKE operator.  
-Column in where clause should be indexed.  
DATABASE-  
Profile -To control system resources like memory, diskspace, and CPU time.  
Role -Collection of privileges.  
Type of segment- Rollback, Temp, Data, Index  
Snapshot-It's a read only table, to improve efficiency of query, which referred remote db, therefore reduce remote traffic.  
DB trigger-is a PL/SQL block that are associated with given table.  
Diff bet Trigger and Procedure-  
-Trigger need not required to be call (Implicitly fire on event)  
-No TCL used  
-Proc/fun can be used in trigger  
-No use of Long raw,LOB,LONG   
-Procedure is prefered over trigger as proc stored in compile form as trigg p\_code stores.  
TO check time nbetwen 8 am and 6 pm.  
Create or replace trigger ptpt  
before insert on batch  
for each row  
declare  
A varchar2 (20);  
begin  
Select substr (to\_char (sysdate,'HH: MI: SSSS’) 1,2) into a from dual;

If (a between '08' and ‘18’) then  
Raise\_application\_error (-20001,'Invalid Time');  
End if;  
End;  
Snapshot too old-We have to refresh the snapshot  
Alter snapshot as  
Select \* from batch@dmaster.link  
Refresh after seven days.  
We can reduce network traffic-  
-By using snapshot  
-By storing related table in same tablespace  
-By avoiding Row chain.  
Oracle DB uses three types of file structure.   
Data files-store actual data for tablespace, which is a logical unit of storage. Every tablespace has one or more data file to store actual data for tables, indexes, and clusters. Data is read and write to data file as needed.  
Redo log file-Two or more redo log file make up a logical redo log, which is used to recover modifications that have not been written to data files in event of power outage.  
Control file-Used at start up to identify the DB and determine which redo log file and data file are created.  
1 data file, 1 control file, 2 redo log file.  
SET TRANSACTION-We use set transaction statement to login a read only or read-write or to assign the current transaction to specified rollback segment.  
Where date=sysdate-daily sale  
>sysdate-7 weekly sale  
>sysdate-30 monthly sale.  
A function must contain atleast one return value else PL/SQL raises predefined exception program\_error.  
Actual parameter- when call  
Formal parameter   
Parametric Cursor - The cursor in which we can pass value when it is being opened  
Sql Stmt Execu-  
-Reserves an area in memory called Private Sql Area.  
-Populate this area with app. data.  
-Process data in memory area.  
-Free the, memory area when exec is complete.  
Active set- A set of rows return by a mult-row query.  
Export-Putting data of tables in file, which can be, handles by OS.  
Auditing-  
is used for noting down user's activity and statistics abt the operations in data objects. The auditing are   
1-Stmt  
2-Preveledge  
3-Object   
1-It is done to audit stmt activity .The auditing information abt. date & time of information, nature of operation is stored in table AUD$ which is used by user sys.  
Audit select on itemmaster;  
Then app. auditing is done and stored in table .  
-To record the usage of privilege  
-To record the activity on object.  
Nature of Auditing-  
Auditing is done on   
-Per session basis-one record is generated.  
Per statement basis per session/stmt  
Audit any allows user to audit any schema object in the DB.  
Table partitioning-  
Table partitioning divides table data between two or more tablespaces and physical data file on separate disk.  
We can use it to improve transaction throughout and certain type of queries for large tables.  
Restriction-  
-A table that is a part of cluster can't be partioned.  
-A table can be partitioned based on ranges column values only.  
-Attribute of partitioned table can't include long, long raw or any lob data type.  
-Bitmap indexes can't be defined on partioned tables.  
We add partition using ALTER TABLE OR  
Create table aa (  
a date,  
B number  
C varchar2 (10))  
partion by range(a,b)  
(partition pa1 values less than ('01-jan-99', 2) tablespace tsp1,  
-----------------------------------);  
Accessing partition table-  
Select \* from aa partion(pa1);  
Drop partion   
-Alter table AA  
drop partion pa1;

SQL Language Extension-  
Oracle \* provide new built-in datatype, object datatypes, nested tables, and a no of other features that require new DDL extension.  
VARRAY  
REF  
LOBS  
Create table AA(a N (10)  
B date,  
C varchar2 (10));

Create type aa1 as varray (5) of number (5);

The UTLBSTAT and UTLESTAT script to get general overview of database 's performance over a certain period of time.  
UTLBSTAT creates table and views containing cumulative database performance summary information at the time when the script runs .All the objects create by UTLBSTAT contain word login.  
Utlbstat.sql

UTLESTAT creates table and views containing cumulative database performance summary information at the time when the script runs .All the objects create by UTLESTAT contain word end.  
UTLESTAT spools the results of these SQL statements to a file called REPORT.TXT   
Utlestat.sql

Determine the shared Pool Performance.

The shared pool is one of the memory structures in SGA .It is comprised of the data dictionary and the library cache. Check v$sgastat  
The data dictionary cache buffers data dictionary objects that contain data about tables, indexes, users and all other objects.  
The Library Cache/SQL Cache buffers previously executed queries, so that they need not be reloaded and reparsed if user calls them again.

Otherwise if the information is not in the buffer then oracle must get it from disk.

The V$LIBRAY CACHE View stores performance data for library cache and V$ROWCACHE view stores performance data for the data dictionary cache.

Sometime we may have to increase the value of initialization parameter SHARED\_POOL\_SIZE.  
To improve the performance .

Redo Log --

Oracle 8 stores all changes to the database, even uncommitted changes, in the redo log files.   
LGWR writes .

Alter database archievelog

Edit the parameter initialization file.  
Log\_archieve\_start =true -turn it on   
Log\_archieve\_dest=c:/oracle/ora81/archieve -location   
log\_archieve\_format="ARCH%S.LOG" - name format for archieve file .  
%S for log sequence number .

By querying the V$SESSION view , we can determine who is logged on ,as well as information such as the time of logon .

Kill a session - ALTER system kill session '&sid,&serial'  
Select Sid,serial#,status from V$session where username='name';

Unbalanced Index –  
if we do have lot on index on a table and we are doing I/U/D frequently then there is a problem of disk contention . To check this problem sees the BLEVEL value in DBA\_INDEXES and if it is 1,2,3,4 then it’s ok else rebuild the index .

Alter index satish.a\_satish rebuild unrecoverable ;

Comments on table and columns

--For documentation purpose .  
Comment table a is ‘table a’ ; Retrieve comment from user\_tab\_comment  
Comment column a. a is ‘column a’; user\_col\_comments

Detect the objects close to maximum extent   
Check in dba\_seqment  
Detect row chaining and row migration in tables  
Row migration occurs when a database block doesn’t contain enough free space to accommodate an update statement .In that case server moves the row to another block and maintains a pointer to to new block in the row’s original block .when pctfree is 0  
Row chaining in contrast , occurs when no single db block is larger enough to accommodate a particular row . this is common when table contain several large data types. It will reside in multiple database blocks .  
An unpleasant side effect of both chaining and migration is that the oracle \* server must read more than one db block to read a single row . solution – move rows to a temp table and then delete rows from original table and then insert it from temp table .

Execute utlchain.sql

Get information from CHAINED\_ROWS or V$SYSSTAT

* SQL: Queries
  1. Write a query to select the second highest salary from a table.  
     Answer: SELECT max(salary) AS salary2 FROM orders WHERE salary < (SELECT max(salary) AS salary1 FROM orders)
  2. Write a query to select the 5th highest salary from a table.  
     Answer: SELECT min(salary) AS high5 FROM employee WHERE salary IN(SELECT DISTINCT TOP 5 salary FROM orders ORDER BY salary DESC)
  3. How to find duplicate records with the number they are duplicated?  
     Answer: SELECT Id, count (\*) as number\_records from table group by id having count (\*) > 1.
* SQL: Questions
  1. What is the difference between Delete and Truncate command in SQL?  
     Answer: Delete command and truncate command both will delete the data, however the truncate command can not be rolled back as delete can be. The delete command can be used for selected records using the where clause but with the truncate command we have to loose data. DELETE statement is a logged operation and hence takes more time then truncate.
  2. What is Magic Table in SQL?  
     Answer: The insert and Delete commands are known as magic tables in SQL.
  3. Can Primary key is a Foreign Key on the same table?  
     Answer: Yes, Consider a category table in a e-commerce web site.  
     Category\_Id, Category\_Name, Parent\_Category\_ID. In this table all the parent categories are also categories. When we create a self join category id will be treated as foreign key to the same table.
  4. What is Normalization? What are it’s rules?  
     Answer: Normalisation is the technique in the database design wher ethe idea is to reduce the redundancy of non key data items across the table.
     + Rule 1: There should be a one-to-one relationship between the instances of an entity and the rows of the table.
     + Rule 2: A field should have the same meaning in each row of the table.
     + Rule 3: Each table should represent at most one entity.
     + Rule 4: Multiple instances of an entity should be represented by multiple rows in a table.
     + Rule 5: Joins should be based only on primary and foreign-key equality.
     + Rule 6: Make sure keys are linked correctly.
  5. What are the advantages and disadvantages of Normalization?  
     Answer: There are several advantages of normalization as under:
     + Faster sorting and index creation.
     + A larger number of clustered indexes.
     + Narrower and more compact indexes.
     + Fewer indexes per table, which improves the performance of INSERT, UPDATE, and DELETE statements
     + Fewer null values and less opportunity for inconsistency, which increase database compactness.  
         
       Beside the above benefits there are few disadvantages as well:
     + Increased amount of Normalization increases the amount of complexity of joins between tables and that hinders the performance.
  6. What are the conditions to achieve the normalization?  
     Answer: There are few conditions to achieve the normalization:
     + There should be a unique row identifier.
     + A table should store only data for a single type of entity. For e.g. details for book’s publisher and book’s author should be saved under different table.
     + A table should avoid columns which can be null-able.
     + A table should avoid duplication of data and columns.
  7. What is a Stored Procedure? State its advantage.  
     Answer: A stored procedure is a set of pre-compiled SQL commands (query statements), which are stored in the server. It is faster then the loose SQL statements processed on client, as it is pre-compiled. It can execute more then one SQL commands once as they are bundled in a single entity. We can use control statements within the stored procedure, which will allow us to repeat some SQL command. It can send return values depending upon the result. Stored procedures are used to reduce network traffic.
  8. What is a Trigger?  
     Answer: Triggers are a special type of stored procedure, which gets invoked upon a certain event. They can be performed upon an INSERT, UPDATE and DELETE.
  9. What is a Clustered Index?  
     Answer: The data rows are stored in order based on the clustered index key. Data stored is in a sequence of the index. In a clustered index, the physical order of the rows in the table is the same as the logical (indexed) order of the key values. A table can contain only one clustered index. A clustered index usually provides faster access to data than does a non-clustered index
  10. What is a Non-Clustered Index?  
      Answer: The data rows are not stored in any particular order, and there is no particular order to the sequence of the data pages. In a clustered index, the physical order of the rows in the table is not same as the logical (indexed) order of the key values.
  11. Describe the three levels of data abstraction?  
      The are three levels of abstraction:
      + Physical level: The lowest level of abstraction describes how data are stored.
      + Logical level: The next higher level of abstraction, describes what data are stored in database and what relationship among those data.
      + View level: The highest level of abstraction describes only part of entire database
  12. What is DDL (Data Definition Language)?  
      Answer: A data base schema which is specified by a set of definitions expressed by a special language is called DDL. Data Definition Language (DDL) is used to define and manage all the objects in an SQL database.
  13. What is DML?  
      Answer: It is a special language used to manipulate the Data. Data Manipulation Language (DML), which is used to select, insert, update, and delete data in the objects defined using DDL.
  14. What is a PRIMARY KEY?  
      Answer: The PRIMARY KEY is the column(s) used to uniquely identify each row of a table.
  15. What is a FOREIGN KEY?  
      Answer: A FOREIGN KEY is one or more columns whose values are based on the PRIMARY or CANDITATE KEY values from the database.
  16. What is a UNIQUE KEY?  
      Answer: A UNIQUE KEY is one or more columns that must be unique for each row of the table.
  17. What is the difference between UNIQUE and PRIMARY KEY?  
      Answer: The UNIQUE KEY column restricts entry of duplicate values but entry of NULL value is allowed. In case of PRIMARY KEY columns entry of duplicate as well as <NULL> value is also restricted.
  18. What is a VIEW?  
      Answer: A View is a database object that is a logical representation of a table. It is derived from a table but has no storage space of its own and often may be used in the same manner as a table.
  19. What is a ROWID?  
      Answer: ROWID is the logical address of a row, and it is unique within the database.
  20. What is INDEX?  
      Answer: INDEX is a general term for an SQL feature used primarily to speed up execution and impose UNIQUENESS upon data. You can use an index to gain fast access to specific information in a database table. An index is a structure that orders the values of one or more columns in a database table. The index provides pointers to the data values stored in specified columns of the table, and then orders those pointers according to the sort order you specify.
  21. What is a cursor?  
      Answer: An entity that maps over a result set and establishes a position on a single row within the result set. After the cursor is positioned on a row, operations can be performed on that row, or on a block of rows starting at that position. The most common operation is to fetch (retrieve) the current row or block of rows.
  22. The Difference between 'Count' and 'Count (\*)'?  
      Answer: 'Count': Counts the number of non-null values. 'Count (\*)': Counts the number of rows in the table, including null values and duplicates.
* 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 DEFAULT 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. Read Committed - A transaction operating at the Read Committed level cannot see changes made by other transactions until those transactions are committed. At this level of isolation, dirty reads are not possible but nonrepeatable reads and phantoms are possible. Read Uncommitted - A transaction operating at the Read Uncommitted level can see uncommitted changes made by other transactions. At this level of isolation, dirty reads, nonrepeatable reads, and phantoms are all possible. Repeatable Read - A transaction operating at the Repeatable Read level is guaranteed not to see any changes made by other transactions in values it has already read. At this level of isolation, dirty reads and nonrepeatable reads are not possible but phantoms are possible. Serializable - A transaction operating at the Serializable level guarantees that all concurrent transactions interact only in ways that produce the same effect as if each transaction were entirely executed one after the other. At this isolation level, dirty reads, nonrepeatable reads, and phantoms are not possible.
* 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’. 1024 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. TRUNCATE TABLE is functionally identical to DELETE statement with no WHERE clause: both remove all rows in the table. But TRUNCATE TABLE is faster and uses fewer system and transaction log resources than DELETE. The DELETE statement removes rows one at a time and records an entry in the transaction log for each deleted row. TRUNCATE TABLE removes the data by deallocating the data pages used to store the table’s data, and only the page deallocations are recorded in the transaction log. TRUNCATE TABLE removes all rows from a table, but the table structure and its columns, constraints, indexes and so on remain. The counter used by an identity for new rows is reset to the seed for the column. If you want to retain the identity counter, use DELETE instead. If you want to remove table definition and its data, use the DROP TABLE statement. You cannot use TRUNCATE TABLE on a table referenced by a FOREIGN KEY constraint; instead, use DELETE statement without a WHERE clause. Because TRUNCATE TABLE is not logged, it cannot activate a trigger. TRUNCATE TABLE may not be used on tables participating in an indexed view
* 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 create 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 creating 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 configuration 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, and 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.
* Explain 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 processing 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 roundtrip, however large the resultset is. Cursors are also costly because they require more resources and temporary storage (results in more IO operations). Further, 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

###### **Left Joins, Right Joins, Inner Joins, Outer Joins**

The most basic SQL statement is the SELECT statement. In it's most basic form, it goes something like this:

#### SELECT Surname, GivenName FROM Employee

The above query will return the *Surname* and *GivenName* fields of every row in the *Employee* table.

If you want to know the names of all the employees who share the surname 'Smith', then you could add a WHERE clause like so:

#### SELECT Surname, GivenName FROM Employee WHERE Surname='Smith'

Now suppose you have another table called *'Client'*, and that we're in the kind of business where each employee has their own set of clients - such as a law firm.

Now being the good database designer that you are, you have created a primary key field in the *Employee* table that links the two tables.

*Table:* ***Employee*** *=================  
Field1: EmployeeID int identity primary key  
Field2: Surname varchar(25)  
Field3: GivenNames varchar(25)*

*Table:* ***Client*** *=================  
Field1: ClientID int identity primary key  
Field2: EmployeeID int  
Field3: Surname varchar(25)  
Field4: GivenNames varchar(25)*

To see which clients belong to which employees, you need a query, which joins the two tables. Here we have a couple of options.

(i) *SELECT Client.Surname, Client.GivenNames,Employee.Surname, Employee.GivenNames*

*FROM Client INNER JOIN Employee ON Client.EmployeeID=Employee.EmployeeID*

(ii) *SELECT Client.Surname, Client.GivenNames,Employee.Surname, Employee.GivenNames*

*FROM Client LEFT JOIN Employee ON Client.EmployeeID=Employee.EmployeeID*

*(iii) SELECT Client.Surname, Client.GivenNames,Employee.Surname, Employee.GivenNames*

#### FROM Client RIGHT JOIN Employee ON Client.EmployeeID=Employee.EmployeeID

Each of these queries will return data similar to:

Surname GivenNames Surname GivenNames

---------------------------------------------------------

Jones Bob Smith Jane

Finch Rachel Smith Jane

Brown Kyle Wu Alice

Note that employees can have more than one client each, and so can appear more than once in the output.

What about clients that have not been assigned to an employee? Or employees that don't have any clients? Herein lies the difference between the types of joins.

→The INNER JOIN (i) will only return clients that have a corresponding record in the Employee –

table, and will only show employees who have at least one client.

→The LEFT JOIN (ii) will return all of the records from the table on the left side of the join,

regardless of whether there is a match in the other table.

Suppose the first set of output above, is the result of an INNER JOIN. A LEFT JOIN will return at least as many records as the INNER JOIN, plus possibly some that would have been missed by the INNER JOIN:

Surname GivenNames Surname GivenNames

---------------------------------------------------------

Jones Bob Smith Jane

Finch Rachel Smith Jane

Brown Kyle Wu Alice

Vir Jacob - -

In this example, Jacob Vir is a client who has not been assigned to an employee. The employee Surname and employee GivenNames are NULL for this row.

→The RIGHT JOIN (iii) will return all of the records from the table on the right-hand side of the

join, regardless of whether there is a match in the other table.

→A RIGHT JOIN will return at least as many records as the INNER JOIN, plus possibly some

that would have been missed by the INNER JOIN:

Surname GivenNames Surname GivenNames

---------------------------------------------------------

Jones Bob Smith Jane

Finch Rachel Smith Jane

Brown Kyle Wu Alice

- - Hall Andrew

In this example, Andrew Hall is an employee with no clients.

→The LEFT JOIN and RIGHT JOIN operators are both types of outer joins, and can also be

written LEFT OUTER JOIN and RIGHT OUTER JOIN.

"How can I create a query that will return all of the records from \*both\* tables".?

To do that, you need a FULL OUTER JOIN:

*SELECT Client.Surname, Client.GivenNames,  
Employee.Surname, Employee.GivenNames  
FROM Client FULL OUTER JOIN Employee  
ON Client.EmployeeID=Employee.EmployeeID*

1. **Difference between group functions and single row functions.**

**Group Function Single Row Function**

A group function operates A single row function

on many rows returns one and result for one row.

returns single result.

Not allowed in Pl/sql procedural Allowed in Pl/Sql

Procedural statements

Statements.

eg SUM(),AVG,MIN,MAX etc eg UPPER,LOWER,CHR...

1. **Difference between DECODE and TRANSLATE**

DECODE is value by value TRANSLATE is character by

character replacement. replacement.

Ex SELECT DECODE('ABC','A',1,'B',2,'ABC',3) eg SELECT

from dual; o/p 3 TRANSLATE('ABCGH',

'ABCDEFGHIJ', 1234567899)

FROM DUAL; o/p 12378

(DECODE command is used to bring IF,THEN,ELSE logic to SQL.It tests for the IF values(s) and then aplies THEN value(s) when true, the ELSE value(s) if not.)

1. **Difference between TRUNCATE and DELETE**

TRUNCATE deletes much faster than DELETE

**Truncate Delete**

It is a DDL statement It is a DML statement

It is a one way trip,cannot One can Rollback

ROLLBACK

Doesn't have selective features (where clause) Has

Doesn't fire database triggers Does

It requires disabling of referential Does not require

constraints.

1. **What is a CO-RELATED SUBQUERY**

A CO-RELATED SUBQUERY is one that has a correlation

name as table or view designator in the FROM clause of the outer

query and the same correlation name as a qualifier of a search

condition in the WHERE clause of the subquery.

eg

*SELECT field1 from table1 X*

*WHERE field2>(select avg(field2) from table1 Y*

*where*

*field1=X.field1);*

(The subquery in a correlated subquery is revaluated

for every row of the table or view named in the outer query.)

1. **What are various joins used while writing SUBQUERIES**

**Self join**-Its a join foreign key of a table references the same table.

**Outer Join**--Its a join condition used where One can query all the rows of one of the

tables in the join condition even though they don't satisfy the join condition.

**Equi-join**--Its a join condition that retrieves rows from one or more tables in which one

Or more columns in one table are equal to one or more columns in the second table.

1. **What are various constraints used in SQL**

*NULL*

*NOT NULL*

*CHECK*

*DEFAULT*

1. **What are different Oracle database objects**

*TABLES*

*VIEWS*

*INDEXES*

*SYNONYMS*

*SEQUENCES*

*TABLESPACES etc*

1. **What is difference between Rename and Alias**

Rename is a permanent name given to a table or column whereas Alias is a temporary

name given to a table or column which do not exist once the SQL statement is

executed.

1. **What is a view**

A view is stored procedure based on one or more tables, it’s a virtual table.

**What are various privileges that a user can grant to another user**

*SELECT*

*CONNECT*

*RESOURCES*

1. **What is difference between UNIQUE and PRIMARY KEY constraints**

A table can have only one *PRIMARY KEY* whereas there can be any number of *UNIQUE* keys.

The columns that compose PK are automatically define NOT NULL, whereas a column that compose a *UNIQUE* is not automatically defined to be mandatory must also specify the column is *NOT NULL.*

1. **Can a primary key contain more than one columns**

Yes

1. **How you will avoid duplicating records in a query**

By using *DISTINCT*

1. **What is difference between SQL and SQL\*PLUS**

SQL\*PLUS is a command line tool where as SQL and PL/SQL language interface and

reporting tool. Its a command line tool that allows user to type SQL commands to be

executed directly against an Oracle database. SQL is a language used to query the

relational database(DML,DCL,DDL). SQL\*PLUS commands are used to format query

result, Set options, Edit SQL commands and PL/SQL.

1. **Which datatype is used for storing graphics and images**

*LONG RAW* data type is used for storing BLOB's (binary large objects).

1. **How will you delete duplicating rows from a base table**

delete from table\_name where rowid not in (select max(rowid) from table group by duplicate\_values\_field\_name);

or

delete duplicate\_values\_field\_name dv from table\_name ta where rowid <(select min(rowid)  from table\_name tb where ta.dv=tb.dv);

1. **What is difference between SUBSTR and INSTR**

SUBSTR returns a specified portion of a string

eg SUBSTR('BCDEF',4) output BCDE

INSTR provides character position in which a pattern

is found in a string.

eg INSTR('ABC-DC-F','-',2) output 7 (2nd occurence of '-')

1. **There is a string '120000 12 0 .125' ,how you will find the**

**position of the decimal place**

INSTR('120000 12 0 .125',1,'.')

output 13

1. **There is a '%' sign in one field of a column. What will be**

**the query to find it.**

'\' Should be used before '%'.

1. **When you use WHERE clause and when you use HAVING clause**

*HAVING* clause is used when you want to specify a condition for a group function and it

is written after GROUP BY clause

The *WHERE* clause is used when you want to specify a condition for columns, single

row functions except group functions and it is written before GROUP BY clause if it is

used.

1. **Which is more faster - IN or EXISTS**

EXISTS is more faster than IN because EXISTS returns

a Boolean value whereas IN returns a value.

1. **What is a OUTER JOIN**

Outer Join--Its a join condition used where you can query all the rows of one of the

tables in the join condition even though they don’t satisfy the join condition.

1. **How you will avoid your query from using indexes**

SELECT \* FROM emp

Where emp\_no+' '=12345;

i.e you have to concatenate the column name with

space within codes in the where condition.

SELECT /\*+ FULL(a) \*/ ename, emp\_no from emp

where emp\_no=1234;

i.e using HINTS

1. **What is a pseudo column. Give some examples**

It is a column that is not an actual column in the

table.

eg USER, UID, SYSDATE, ROWNUM, ROWID, NULL, AND LEVEL.

1. **Suppose customer table is there having different columns**

**like customer no, payments.What will be the query to select top three max payments.**

*SELECT* customer\_no, payments from customer C1

*WHERE* 3<=(*SELECT COUNT*(\*) from customer C2

*WHERE* C1.payment <= C2.payment)

**Display the records between two range**

select rownum, empno, ename  from emp  where  rowid in   
 (select rowid from emp where rownum <=&upto   
 minus  select rowid from emp where rownum<&Start);

**Find out nth highest salary from emp table**   
 SELECT DISTINCT (a.sal) FROM EMP A WHERE &N = (SELECT COUNT (DISTINCT (b.sal)) FROM EMP B WHERE a.sal<=b.sal);

For Eg:-

Enter value for n: 2   
      SAL   
---------   
     3700

**Display the number value in Words**   
 SQL> select sal, (to\_char(to\_date(sal,'j'), 'jsp'))   
from emp;

the output like,

      SAL (TO\_CHAR(TO\_DATE(SAL,'J'),'JSP'))   
--------- -----------------------------------------------------   
      800 eight hundred   
     1600 one thousand six hundred   
     1250 one thousand two hundred fifty   
If you want to add some text like,   
Rs. Three Thousand only.   
SQL> select sal  "Salary ",   
 (' Rs. '|| (to\_char(to\_date(sal,'j'), 'Jsp'))|| ' only.'))   
 "Sal in Words" from emp   
/   
Salary  Sal in Words   
------- ------------------------------------------------------   
    800  Rs. Eight Hundred only.   
   1600  Rs. One Thousand Six Hundred only.   
   1250  Rs. One Thousand Two Hundred Fifty only.

**14.  Display Odd/ Even number of records**   
 Odd number of records:   
select \* from emp where (rowid,1) in (select rowid, mod(rownum,2) from emp);

**Output:-**   
1   
3   
5

Even number of records:   
select \* from emp where (rowid,0) in (select rowid, mod(rownum,2) from emp)

**Output:-**   
  
2   
4   
6

**'How do you find the numbert of rows in a Table ?'**

A bad answer is count them (SELECT COUNT(\*) FROM table\_name)

A good answer is :-

'By generating SQL to ANALYZE TABLE table\_name COUNT STATISTICS by querying Oracle System Catalogues (e.g. USER\_TABLES or ALL\_TABLES).

The best answer is to refer to the utility which Oracle released which makes it

unnecessary to do ANALYZE TABLE for each Table individually.

**minvalue.sql**

**Select the Nth lowest value from a table**

select level, min('col\_name') from my\_table where level = '&n' connect by prior ('col\_name') < 'col\_name')

group by level;

Example:

Given a table called emp with the following columns:

-- id number

-- name varchar2(20)

-- sal number

--

**-- For the second lowest salary:**

-- select level, min(sal) from emp

-- where level=2

-- connect by prior sal < sal

-- group by level

**maxvalue.sql**

**Select the Nth Highest value from a table**

select level, max('col\_name') from my\_table where level = '&n' connect by prior ('col\_name') > 'col\_name')

group by level;

Example:

Given a table called emp with the following columns:

-- id number

-- name varchar2(20)

-- sal number

--

**-- For the second highest salary:**

-- select level, max(sal) from emp

-- where level=2

-- connect by prior sal > sal

-- group by level

# SQL Server Interview Questions

Questions are categorized under the following sections, for your convenience:

1. Database design (8 questions)
2. SQL Server architecture (12 questions)
3. Database administration (13 questions)
4. Database programming (10 questions)
5. Database design

* 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 DEFAULT 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. Read Committed - A transaction operating at the Read Committed level cannot see changes made by other transactions until those transactions are committed. At this level of isolation, dirty reads are not possible but nonrepeatable reads and phantoms are possible. Read Uncommitted - A transaction operating at the Read Uncommitted level can see uncommitted changes made by other transactions. At this level of isolation, dirty reads, nonrepeatable reads, and phantoms are all possible. Repeatable Read - A transaction operating at the Repeatable Read level is guaranteed not to see any changes made by other transactions in values it has already read. At this level of isolation, dirty reads and nonrepeatable reads are not possible but phantoms are possible. Serializable - A transaction operating at the Serializable level guarantees that all concurrent transactions interact only in ways that produce the same effect as if each transaction were entirely executed one after the other. At this isolation level, dirty reads, nonrepeatable reads, and phantoms are not possible.
* 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'. 1024 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. TRUNCATE TABLE is functionally identical to DELETE statement with no WHERE clause: both remove all rows in the table. But TRUNCATE TABLE is faster and uses fewer system and transaction log resources than DELETE. The DELETE statement removes rows one at a time and records an entry in the transaction log for each deleted row. TRUNCATE TABLE removes the data by deallocating the data pages used to store the table's data, and only the page deallocations are recorded in the transaction log. TRUNCATE TABLE removes all rows from a table, but the table structure and its columns, constraints, indexes and so on remain. The counter used by an identity for new rows is reset to the seed for the column. If you want to retain the identity counter, use DELETE instead. If you want to remove table definition and its data, use the DROP TABLE statement. You cannot use TRUNCATE TABLE on a table referenced by a FOREIGN KEY constraint; instead, use DELETE statement without a WHERE clause. Because TRUNCATE TABLE is not logged, it cannot activate a trigger. TRUNCATE TABLE may not be used on tables participating in an indexed view
* 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 create 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 creating 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 configuration 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.
* Explain 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 processing 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 roundtrip, however large the resultset is. Cursors are also costly because they require more resources and temporary storage (results in more IO operations). Further, 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

# SQL SERVER

34.What is normalization?

Ans: Well a relational database is basically composed of tables that contain related data. So the Process of organizing this data into tables is actually referred

to as normalization.

35.What is a Stored Procedure?

Ans: Its nothing but a set of T-SQL statements combined to perform a single task of several tasks. Its basically like a Macro so when you invoke the Stored procedure, you actually run a set of statements.

36.Can you give an example of Stored Procedure?

Ans: sp\_helpdb , sp\_who2, sp\_renamedb are a set of system defined stored procedures. We can also have user defined stored procedures which can be called in

similar way

What is a trigger?

Ans: Triggers are basically used to implement business rules. Triggers is also similar to stored procedures. The difference is that it can be activated when data

is added or edited or deleted from a table in a database.

37.What is a view?

Ans: If we have several tables in a db and we want to view only specific columns from specific tables we can go for views. It would also suffice the needs of

security some times allowing specfic users to see only specific columns based on the permission that we can configure on the view. Views also reduce the effort

that is required for writing queries to access specific columns every time.

38.What is an Index?

Ans: When queries are run against a db, an index on that db basically helps in the way the data is sorted to process the query for faster and data retrievals

are much faster when we have an index.

39.What are the types of indexes available with SQL Server?

Ans: There are basically two types of indexes that we use with the SQL Server. Clustered and the Non-Clustered.

40.What is the basic difference between clustered and a non-clustered index?

Ans: The difference is that, Clustered index is unique for any given table and we can have only one clustered index on a table. The leaf level of a clustered index

is the actual data and the data is resorted in case of clustered index. Whereas in case of non-clustered index the leaf level is actually a pointer to the data

in rows so we can have as many non-clustered indexes as we can on the db.

41.What are cursors?

Ans: Well cursors help us to do an operation on a set of data that we retreive by commands such as Select columns from table. For example : If we have duplicate

records in a table we can remove it by declaring a cursor which would check the records during retrieval one by one and remove rows which have duplicate

values.

42.When do we use the UPDATE\_STATISTICS command?

Ans: This command is basically used when we do a large processing of data. If we do a large amount of deletions any modification or Bulk Copy into the tables, we need to basically update the indexes to take these changes into account. UPDATE\_STATISTICS updates the indexes on these tables accordingly.

43.SQL Server runs on which TCP/IP port?

Ans: SQL Server runs on port 1433 but we can also change it for better security.

44.From where can you change the default port?

Ans: From the Network Utility TCP/IP properties --> Port number.both on client and the server.

45.Can you tell me the difference between DELETE & TRUNCATE commands?

Ans: Delete command removes the rows from a table based on the condition that we provide with a WHERE clause. Truncate will actually remove all the rows

from a table and there will be no data in the table after we run the truncate command.

46.Can we use Truncate command on a table which is referenced by FOREIGN KEY?

**Ans:** No. We cannot use Truncate command on a table with Foreign Key because of referential integrity.

TRUNCATE TABLE removes all rows from a table, but the table structure and its columns, constraints, indexes and so on remain. The counter used by an identity for new rows is reset to the seed for the column. If you want to retain the identity counter, use DELETE instead. If you want to remove table definition and its data, use the DROP TABLE statement.

47.What is the use of DBCC commands?

Ans: DBCC stands for database consistency checker. We use these commands to check the consistency of the databases, i.e., maintenance, validation task and

status checks.

48.Can you give me some DBCC command options?(Database consistency check)

Ans: DBCC CHECKDB - Ensures that tables in the db and the indexes are correctly linked.and DBCC CHECKALLOC - To check that all pages in a db are correctly

allocated. DBCC SQLPERF - It gives report on current usage of transaction log in percentage. DBCC CHECKFILEGROUP - Checks all tables file group for any

damage.

49.What command do we use to rename a db?

Ans: sp\_renamedb 'oldname' , 'newname'

50.Well sometimes sp\_reanmedb may not work you know because if some one is using the db it will not accept this command so what do you think you can do in such

cases?

Ans: In such cases we can first bring to db to single user using sp\_dboptions and then we can rename that db and then we can rerun the sp\_dboptions command to

remove the single user mode.

51**.What is the difference between a HAVING CLAUSE and a WHERE CLAUSE?**

Ans: Having Clause is basically used only with the GROUP BY function in a query. WHERE Clause is applied to each row before they are part of the GROUP BY

function in a query.

52.What do you mean by COLLATION?

Ans: Collation is basically the sort order. There are three types of sort order Dictionary case sensitive, Dictonary - case insensitive and Binary.

53.What is a Join in SQL Server?

Ans: Join actually puts data from two or more tables into a single result set.

54.Can you explain the types of Joins that we can have with Sql Server?

Ans: There are three types of joins: Inner Join, Outer Join, Cross Join

55.When do you use SQL Profiler?

Ans: SQL Profiler utility allows us to basically track connections to the SQL Server and also determine activities such as which SQL Scripts are running,

failed jobs etc..

56.What is a Linked Server?

Ans: Linked Servers is a concept in SQL Server by which we can add other SQL Server to a Group and query both the SQL Server dbs using T-SQL Statements.

57.Can you link only other SQL Servers or any database servers such as Oracle etc..

Ans: We can link any server provided we have the OLE-DB provider from Microsoft to allow a link. For Oracle we have a OLE-DB provider for oracle that microsoft provides to add it as a linked server to the sql server group.

58.Which stored procedure will you be running to add a linked server?

Ans: sp\_addlinkedserver, sp\_addlinkedsrvlogin

59.What are the OS services that the SQL Server installation adds?

Ans: MS SQL SERVER SERVICE, SQL AGENT SERVICE, DTC

(Distribution transac co-ordinator)

60.Can you explain the role of each service?

Ans: SQL SERVER - is for running the databases

SQL AGENT - is for automation such as Jobs, DB Maintanance, Backups

DTC - Is for linking and connecting to other SQL Servers

61.How do you troubleshoot SQL Server if its running very slow?

Ans: First check the processor and memory usage to see that processor is not above 80% utilization and memory not above 40-45% utilization then check the disk

utilization using Performance Monitor, Secondly, use SQL Profiler to check for the users and current SQL activities and jobs running which might be a problem.

Third would be to run UPDATE\_STATISTICS command to update the indexes

62.Lets say due to N/W or Security issues client is not able to connect to server or vice versa. How do you troubleshoot?

Ans: First I will look to ensure that port settings are proper on server and client Network utility for connections. ODBC is properly configured at client end

for connection ------Makepipe & readpipe are utilities to check for connection. Makepipe is run on Server and readpipe on client to check for any connection issues.

63.What are the authentication modes in SQL Server?

Ans: Windows mode and mixed mode (SQL & Windows).

64.Where do you think the users names and passwords will be stored in sql server?

Ans: They get stored in master db in the sysxlogins table.

65.What is log shipping? Can we do logshipping with SQL Server 7.0

Ans: Logshipping is a new feature of SQL Server 2000. We should have two SQL Server - Enterprise Editions.

>From Enterprise Manager we can configure the logshipping. In logshipping the transactional log file from one server is automatically updated into the

backup database on the other server. If one server fails, the other server will have the same db and we can use this as the DR (disaster recovery) plan.

66.Let us say the SQL Server crashed and you are rebuilding the databases including the master database what procedure to you follow?

Ans: For restoring the master db we have to stop the SQL Server first and then from

command line we can type SQLSERVER –m which will basically bring it into

the maintenance mode after which we can restore the master db.

67.Let us say master db itself has no backup. Now you have to rebuild the db so what kind of action do you take?

Ans: (I am not sure- but I think we have a command to do it).

68.What is BCP? When do we use it?

Ans:BulkCopy is a tool used to copy huge amount of data from tables and views. But it won't copy the structures of the same.

69.What should we do to copy the tables, schema and views from one SQL Server to another?

Ans: We have to write some DTS packages for it.

70.What are the different types of joins and what dies each do?

71.What are the four main query statements?

72.What is a sub-query? When would you use one?

73.What is a NOLOCK?

Using NOLOCK politely asks SQL Server to ignore locks and read directly from the tables. This means you completely circumvent the lock system, which is a major performance and scalability improvement. However, you also completely circumvent the lock system, which means your code is living dangerously. You might read the not-necessarily-valid uncommitted modifications of a running transaction. This is a calculated risk.

74.What are three SQL keywords used to change or set someone’s permissions?

75.What is the difference between HAVING clause and the WHERE clause?

Explained in the below material

76.What is referential integrity? What are the advantages of it?

77.What is database normalization?

It is the process of organizing data to minimize data redundancy

Normalization usually involves dividing a database into two or more tables and defining a relationship between the tables

First Normal Form (1NF)

Each attribute value must be atomic

Second Normal Form (2NF)

1NF and each non-key attribute must be dependent on the entire primary key

\*Third Normal Form (3NF)

2NF and each non-key attribute depends only on the primary key

Fourth Normal Form (4NF)

3NF and there can be no more than one multi-valued attribute in the relation

Fifth Normal Form (5NF)

4NF and the table cannot be split into two or more tables without loss of information

78.Which command using Query Analyzer will give you the version of SQL server and operating system?

ANS: select @@version

79.Using query analyzer, name 3 ways you can get an accurate count of the number of records in a table?

ANS:

SELECT COUNT(\*) from TableName

SELECT rows FROM sysindexes WHERE id = OBJECT\_ID('What\_When') AND indid < 2

80.What is the purpose of using COLLATE in a query?

81.What is a trigger?

82.What is one of the first things you would do to increase performance of a query? Example “ a query ran yesterday took 30 seconds, today it takes 6 minutes”

83.What is an execution plan? When would you use it? How would you view the execution plan?

84.What is the STUFF function and how does it differ from the REPLACE function?

Deletes a specified length of characters and inserts another set of characters at a specified starting point.

STUFF **(** *character\_expression* **,** *start* **,** *length* **,** *character\_expression* **)**

# REPLACE

Replaces all occurrences of the second given string expression in the first string expression with a third expression.

##### Syntax

REPLACE **( '***string\_expression1***' , '***string\_expression2***' , '***string\_expression3***'**

85.What does it mean to have quoted identifier on? What are the implications of having it off?

Causes Microsoft® SQL Server™ to follow the SQL-92 rules regarding quotation mark delimiting identifiers and literal strings. Identifiers delimited by double quotation marks can be either Transact-SQL reserved keywords or can contain characters not usually allowed by the Transact-SQL syntax rules for identifiers.

##### Syntax

SET QUOTED\_IDENTIFIER { ON | OFF }

##### Remarks

When SET QUOTED\_IDENTIFIER is ON, identifiers can be delimited by double quotation marks, and literals must be delimited by single quotation marks. When SET QUOTED\_IDENTIFIER is OFF, identifiers cannot be quoted and must follow all Transact-SQL rules for identifiers. For more information, see **Using Identifiers**. Literals can be delimited by either single or double quotation marks.

When SET QUOTED\_IDENTIFIER is ON, all strings delimited by double quotation marks are interpreted as object identifiers. Therefore, quoted identifiers do not have to follow the Transact-SQL rules for identifiers. They can be reserved keywords and can include characters not usually allowed in Transact-SQL identifiers. Double quotation marks cannot be used to delimit literal string expressions; single quotation marks must be used to enclose literal strings. If a single quotation mark (') is part of the literal string, it can be represented by two single quotation marks ("). SET QUOTED\_IDENTIFIER must be ON when reserved keywords are used for object names in the database.

When SET QUOTED\_IDENTIFIER is OFF (default), literal strings in expressions can be delimited by single or double quotation marks. If a literal string is delimited by double quotation marks, the string can contain embedded single quotation marks, such as apostrophes.

86.What are the different types of replication? How are they used?

ANS: 3Types, Snapshot, transactional, Merge

87.What is the difference between a local and a global variable?

SET @*local\_variable*

Is the name of a variable of any type except **cursor**, **text**, **ntext**, or **image**

Variable names must begin with one at sign (**@**). Variable names must conform to the rules for identifiers we can use a local variable assigned a value with SET in a SELECT statement. It is recommended that SET *@local\_variable* be used for variable assignment rather than SELECT *@local\_variable*

88.What is the difference between a Local temporary table and a Global temporary table? How is each one used?

Creating a temporary table is virtually the same as creating a normal table. The main exception is the naming of the table. A hash ('#') character as the first character in the table name denotes that it is a temporary table.

There are two types of temporary tables, local and global.

A local tempory table has a single hash ('#') at the start of its name. A local temporary table is visible only to the user who created it and is destroyed automatically when that user disconnects.

A global temporary table is denoted by a name starting with two hashs (i.e. '##'). A global temporary table is visible to all users and is deleted automatically when the last user who has referenced the table disconnects.

An example of creating a local temporary table:

create table #foo  
(  
    CarIndex  smallint,  
    CarType   varchar(20)  
)

An example of creating a global temporary table:

create table ##baz  
(  
    CarIndex  smallint,  
    CarType   varchar(20)  
)

89.What are cursors? Name four types of cursors and when each one would be applied?

91.How do you use DBCC statements to monitor various aspects of a SQL server installation?

ANS: Updates information about the distribution of key values for one or more statistics groups (collections) in the specified table or indexed view

92.How do you load large data to the SQL server database?

93.How do you check the performance of a query and how do you optimize it?

94.How do SQL server 2000 and XML linked? Can XML be used to access data?

95.What is SQL server agent?

96.What is referential integrity and how is it achieved?

97.What is indexing?

98.What is normalization and what are the different forms of normalizations?

99.Difference between server.transfer and server.execute method?

100.What id de-normalization and when do you do it?

101.What is better- 2nd Normal form or 3rd normal form? Why?

102.Can we rewrite sub queries into simple select statements or with joins? Example?

103.What is a function? Give some example?

104.What is a stored procedure?

105.Difference between Function and Procedure-in general?

107.Can a stored procedure call another stored procedure. If yes what level and can it be controlled?

108.Can a stored procedure call itself(recursive). If yes what level and can it be controlled.?

109.How do you find the number of rows in a table?

110.Difference between Cluster and Non-cluster index?

111.What is a table called, if it does have neither Cluster nor Non-cluster Index?

112.Explain DBMS, RDBMS?

113.Explain basic SQL queries with SELECT from where Order By, Group By-Having?

114.Explain the basic concepts of SQL server architecture?

115.Explain couple pf features of SQL server 2000(Scalability, Availability, Integration with internet, etc.)?

116.Explain fundamentals of Data ware housing & OLAP?

117.Explain the new features of SQL server 2000?

118.How do we upgrade from SQL Server 6.5 to 7.0 and 7.0 to 2000?

119.What is data integrity? Explain constraints?

120.Explain some DBCC commands?

121.Explain sp\_configure commands, set commands?

Displays or changes global configuration settings for the current server

122.Explain what are db\_options used for?

123.What are the basic functions for master, msdb, tempdb databases?

124.What is a job?

125.What are tasks?

126.What are primary keys and foreign keys?

127.How would you Update the rows, which are divisible by 10, given a set of numbers in column?

128.If a stored procedure is taking a table data type, how it looks?

129.How m-m relationships are implemented?

130.How do you know which index a table is using?

131.How will you test the stored procedure taking two parameters namely first name and last name returning full name?

132.How do you find the error, how can you know the number of rows effected by last SQL statement?

134.What are sub-queries? Give example? In which case sub-queries are not feasible?

135.What are the types of joins? When do we use Outer and Self-joins?

136.Which virtual table does a trigger use?

137.How do you measure the performance of a stored procedure?

138.Questions regarding Raiseerror?

RAISERROR is a more powerful statement than PRINT for returning messages back to applications. RAISERROR can return messages in either of these forms:

* A user-defined error message that has been added to **master.dbo.sysmessages** using the **sp\_addmessage** system stored procedure.
* A message string specified in the RAISERROR statement.

RAISERROR also has these extensions to the capabilities of PRINT:

* RAISERROR can assign a specific error number, severity, and state.
* RAISERROR can request that the error be logged in the Microsoft® SQL Server™ 2000 error log and the Microsoft Windows NT® application log.

Severity level 10 messages are informational  and indicate problems caused by mistakes in the information you have entered. Severity levels from 11 through 16 are generated by the user, and can be corrected by the user.

Severity levels from 17 through 25 indicate software or hardware errors. You should inform the system administrator whenever problems that generate errors with severity levels 17 and higher occur. The system administrator must resolve these errors and track their frequency. When a level 17, 18, or 19 error occurs, you can continue working, although you might not be able to execute a particular statement.

The system administrator should monitor all problems that generate severity levels from 17 through 25 and print the error log that contains information to backtrack from the error.

If the problem affects an entire database, you can use DBCC CHECKDB (database ) to determine the extent of the damage. DBCC may identify some objects that must be removed and will optionally repair the damage. If damage is extensive, the database might have to be restored.

When specifying user-defined error messages with RAISERROR, use error message numbers greater than 50,000 and severity levels from 0 through 18. Only system administrators can issue RAISERROR with a severity level from 19 through 25.

139. Alternative way to get the table's row count?

SET SHOWPLAN\_TEXT ON

SET STATISTICS IO ON

There is another way to determine the total row count in a table. You can use the **sysindexes** system table for this purpose. There is ROWS column in the **sysindexes** table. This column contains the total row count for each table in your database. So, you can use the following select statement instead of above one:  
  
*SELECT rows FROM sysindexes WHERE id = OBJECT\_ID('table\_name') AND indid < 2*  
  
There are physical read and logical read operations. A logical read occurs if the page is currently in the cache. If the page is not currently in the cache, a physical read is performed to read the page into the cache. To see how many logical or physical read operations were made, you can use **SET STATISTICS IO ON** command.

140.If there is failure during updation of certain rows, what will be the state?

@@FETCH\_STATUS

Returns the status of the last cursor FETCH statement issued against any cursor currently opened by the connection.

| **Return value** | **Description** |
| --- | --- |
| 0 | FETCH statement was successful. |
| -1 | FETCH statement failed or the row was beyond the result set. |
| -2 | Row fetched is missing. |

##### Syntax: @@FETCH\_STATUS Return Types: integer

##### Remarks

Because @@FETCH\_STATUS is global to all cursors on a connection, use @@FETCH\_STATUS carefully. After a FETCH statement is executed, the test for @@FETCH\_STATUS must occur before any other FETCH statement is executed against another cursor. The value of @@FETCH\_STATUS is undefined before any fetches have occurred on the connection.

For example, a user executes a FETCH statement from one cursor, and then calls a stored procedure that opens and processes the results from another cursor. When control is returned from the called stored procedure, @@FETCH\_STATUS reflects the last FETCH executed in the stored procedure, not the FETCH statement executed before the stored procedure is called.

##### Examples

This example uses @@FETCH\_STATUS to control cursor activities in a WHILE loop.

DECLARE Employee\_Cursor CURSOR FOR

SELECT LastName, FirstName FROM Northwind.dbo.Employees

OPEN Employee\_Cursor

FETCH NEXT FROM Employee\_Cursor

WHILE @@FETCH\_STATUS = 0

BEGIN

FETCH NEXT FROM Employee\_Cursor

END

CLOSE Employee\_Cursor

DEALLOCATE Employee\_Cursor

# @@CONNECTIONS

Returns the number of connections, or attempted connections, since Microsoft® SQL Server™ was last started.  
**Return Types:** integer

# @@CPU\_BUSY

Returns the time in milliseconds (based on the resolution of the system timer) that the CPU has spent working since Microsoft® SQL Server™ was last started.

**Return Types:** integer

# @@CURSOR\_ROWS

Returns the number of qualifying rows currently in the last cursor opened on the connection. To improve performance, Microsoft® SQL Server™ can populate large keyset and static cursors asynchronously. @@CURSOR\_ROWS can be called to determine that the number of the rows that qualify for a cursor are retrieved at the time @@CURSOR\_ROWS is called.

**Return value: Description**

-*m*: The cursor is populated asynchronously. The value returned (-*m*) is the number of rows currently in the keyset.

-1: The cursor is dynamic. Because dynamic cursors reflect all changes, the number of rows that qualify for the cursor is constantly changing. It can never be definitely stated that all qualified rows have been retrieved.

0: No cursors have been opened, no rows qualified for the last opened cursor, or the last-opened cursor is closed or deallocated.

*n*: The cursor is fully populated. The value returned (*n*) is the total number of rows in the cursor.

@@DATEFIRST

Returns the current value of the SET DATEFIRST parameter, which indicates the specified first day of each week: 1 for Monday, 3 for Wednesday, and so on through 7 for Sunday.  
Return Types **tinyint**

# @@DBTS

Returns the value of the current **timestamp** data type for the current database. This **timestamp** is guaranteed to be unique in the database.

##### Return Types: varbinary

# @@ERROR

Returns the error number for the last Transact-SQL statement executed.

##### Return Types integer

##### Remarks

When Microsoft® SQL Server™ completes the execution of a Transact-SQL statement, @@ERROR is set to 0 if the statement executed successfully. If an error occurs, an error message is returned. @@ERROR returns the number of the error message until another Transact-SQL statement is executed. You can view the text associated with an @@ERROR error number in the **sysmessages** system table.

Because @@ERROR is cleared and reset on each statement executed, check it immediately following the statement validated, or save it to a local variable that can be checked later.

###### Use @@ERROR with @@ROWCOUNT

This example uses @@ERROR with @@ROWCOUNT to validate the operation of an UPDATE statement. The value of @@ERROR is checked for any indication of an error, and @@ROWCOUNT is used to ensure that the update was successfully applied to a row in the table.

# @@IDENTITY

Returns the last-inserted identity value.  
Return Types: **numeric**

# @@IDLE

Returns the time in milliseconds (based on the resolution of the system timer) that Microsoft® SQL Server™ has been idle since last started.  
**Return Types**: integer

# @@IO\_BUSY

Returns the time in milliseconds (based on the resolution of the system timer) that Microsoft® SQL Server™ has spent performing input and output operations since it was last started.

# @@LANGID

Returns the local language identifier (ID) of the language currently in use.  
Return Types: **smallint**

# @@LANGUAGE

Returns the name of the language currently in use.

##### Return Types: nvarchar

##### Remarks To view information about language settings (including valid official language names), run sp\_helplanguage with no parameter specified.

# @@LOCK\_TIMEOUT

Returns the current lock time-out setting, in milliseconds, for the current session.  
Return Types: **integer**

# @@MAX\_CONNECTIONS

Returns the maximum number of simultaneous user connections allowed on a Microsoft® SQL Server™. The number returned is not necessarily the number currently configured.

##### Return Types: integer

# @@MAX\_PRECISION

Returns the precision level used by **decimal**and **numeric**data types as currently set in the server.  
**Return Types:** tinyint

# @@NESTLEVEL

Returns the nesting level of the current stored procedure execution (initially 0).  
Return Types: **integer**

# @@OPTIONS

Returns information about current SET options.

##### Remarks

SET options can be modified as a whole by using the **sp\_configure user options** configuration option. Each user has an @@OPTIONS function that represents the configuration. When first logging on, all users are assigned a default configuration set by the system administrator.

You can change the language and query-processing options by using the SET statement

# @@PACK\_RECEIVED

Returns the number of input packets read from the network by Microsoft® SQL Server™ since last started.

##### Return Types integer

# @@PACK\_SENT

Returns the number of output packets written to the network by Microsoft® SQL Server™ since last started.

# @@PACKET\_ERRORS

Returns the number of network packet errors that have occurred on Microsoft® SQL Server™ connections since SQL Server was last started.

# @@PROCID

Returns the stored procedure identifier (ID) of the current procedure.

##### Return Types: integer

# @@REMSERVER

Returns the name of the remote Microsoft® SQL Server™ database server as it appears in the login record.

##### Remarks

@@REMSERVER enables a stored procedure to check the name of the database server from which the procedure is run.

@@ROWCOUNT

Returns the number of rows affected by the last statement.

##### Remarks

This variable is set to 0 by any statement that does not return rows, such as an IF statement.

If the table has more than 2 billion rows, use ROWCOUNT\_BIG(). For more information, see [**ROWCOUNT\_BIG()**](http://msdn.microsoft.com/library/en-us/tsqlref/ts_ra-rz_0f55.asp). http://msdn.microsoft.com/library/en-us/tsqlref/ts\_ra-rz\_0f55.asp

# ROWCOUNT\_BIG

Returns the number of rows affected by the last statement executed. This function operates like @@ROWCOUNT, except that the return type of ROWCOUNT\_BIG is **bigint**.

# @@SERVERNAME

Returns the name of the local server running Microsoft® SQL Server™.

##### Syntax

@@SERVERNAME

##### Return Types

**nvarchar**

##### Remarks

SQL Server Setup sets the server name to the computer name during installation. Change @@SERVERNAME by using **sp\_addserver** and then restarting SQL Server. This method, however,  is not usually required.

With multiple instances of SQL Server installed, @@SERVERNAME returns the following local server name information if the local server name has not been changed since setup.

**Instance Server information**

**-----------------------------------------------------------------------------**

Default instance '*servername*'

Named instance '*servername*\*instancename*'

Virtual server - default instance '*virtualservername*'

Virtual server - named instance '*virtualservername*\*instancename*'

Although the @@SERVERNAME function and the SERVERNAME property of SERVERPROPERTY function may return strings with similar formats, the information can be different. The SERVERNAME property automatically reports changes in the network name of the computer.

In contrast, @@SERVERNAME does not report such changes. @@SERVERNAME reports changes made to the local server name using the **sp\_addserver** or **sp\_dropserver** stored procedure.

# @@SERVICENAME

Returns the name of the registry key under which Microsoft® SQL Server™ is running. @@SERVICENAME returns MSSQLServer if the current instance is the default instance; this function returns the instance name if the current instance is a named instance.

# @@SPID

Returns the server process identifier (ID) of the current user process.

# @@TEXTSIZE

Returns the current value of the TEXTSIZE option of the SET statement, which specifies the maximum length, in bytes, of **text** or **image** data that a SELECT statement returns

# @@TIMETICKS

Returns the number of microseconds per tick.

##### Return Types

**integer**

##### Remarks

The amount of time per tick is computer-dependent. Each tick on the operating system is 31.25 milliseconds, or one thirty-second of a second.

# @@TOTAL\_ERRORS

Returns the number of disk read/write errors encountered by Microsoft® SQL Server™ since last started.

# @@TOTAL\_READ

Returns the number of disk reads (not cache reads) by Microsoft® SQL Server™ since last started.

##### Remarks

To display a report containing several SQL Server statistics, including read and write activity, run **sp\_monitor**.

# @@TOTAL\_WRITE

Returns the number of disk writes by Microsoft® SQL Server™ since last started.

##### Remarks

To display a report containing several SQL Server statistics, including read and write activity, run **sp\_monitor**.

# @@TRANCOUNT

Returns the number of active transactions for the current connection

##### Return Types

**integer**

##### Remarks

The BEGIN TRANSACTION statement increments @@TRANCOUNT by 1. ROLLBACK TRANSACTION decrements @@TRANCOUNT to 0, except for ROLLBACK TRANSACTION *savepoint\_name*, which does not affect @@TRANCOUNT. COMMIT TRANSACTION or COMMIT WORK decrement @@TRANCOUNT by 1.

# @@VERSION

Returns the date, version, and processor type for the current installation of Microsoft® SQL Server™.

**SQL Server Locks**

Relational databases like Microsoft's SQL Server use locks to prevent multiple users from making conflicting modifications to a set of data: when a set of data is locked by a user, no other users can modify that same set of data until the first user finishes modifying the data and relinquishes the lock. There are exceptions, but let's not go there.

Some databases - SQL Server included - use locks to prevent users from seeing uncommitted modifications. In these systems, if UserA is modifying some set of data, UserB and all the rest of the users must wait until UserA is done modifying that data before UserB can get a shot at even reading the data, let alone modifying it.

Databases place locks at all levels of their physical hierarchies: rows, pages (typically a few KB of rows), extents (typically a few pages), entire tables, and entire databases. Some databases (Oracle, others?) only use fine-grained row locks, others don't do row locks at all and only allow rough-grained page, extent, table, and database locks. Most databases - SQL Server included - support row locking, but often use rough-grained locks. This is because lock management is a royal pain. Locks aren't small or simple entities, so if you only do row-level locking, you can get yourself into a world of pain: a million-row update can easily swamp memory and be a bear to manage.

Databases that don't do just row-level locking often use a technique called lock escalation to achieve better performance. Unless its clear from the outset that a whole table will be modified, these databases start off using row locks, and they make plans to trade these locks in for rough-grained locks later if too many rows are modified.

Unfortunately, lock escalation introduces and amplifies a whole new problem: deadlock. If two users try to modify semantically-unrelated but physically-near data in two separate tables in reverse order, both users will start off with row locks, then try to upgrade them to page locks, and the situation will be that each user wants something the other user has, so they're stuck. This is deadlock.

For example:

* UserA modifies some rows in TableA, causing a page lock affecting not just the rows UserA modified, but many others
* UserB modifies some rows in TableB, causing a page lock affecting not just the rows UserA modified, but many others
* UserA wants to modify some rows that UserB has locked (but not modified) in TableB
* UserB wants to modify - or maybe just access - some rows that UserA has locked (but not modified) in TableA.

Something's gotta give. To deal with this problem, the database occasionally looks for deadlocks, and kills off one of the transactions so the other can finish. It usually kills the one that's made the least modifications so that it minimizes the cost of rolling back changes. Databases that use only row-level locking almost never have this problem because two users rarely want to modify the exact same row, and even more rarely do they attain locks in the perfectly poor order needed to cause deadlock.

Also, databases like this use lock timeouts to prevent users from waiting too long for a lock. Query timeouts also factor in here. You can write code to retry queries that timeout, but this only automates database congestion. Any timeout that is often reached will only serve to worsen the user experience. Things simply should not take that long.

In practice and under high load, SQL Server's locking system - which is based on lock escalation - does not perform well. Why? Lock contention. Lock contention is the problems of deadlock and waiting for locks. In a system in which many users are modifying the database at once, and many more users are trying to access the database concurrently, the locks are flying, users spend a lot of time waiting to attain locks, deadlocks are frequent, and users are far from happy.

Granted, if you've only got a few occasional users you won't have much trouble with SQL Server's out-of-the-box behavior. You'll be hard pressed to see these problems with simple in-the-office tests or deployments involving just a few users. But throw a couple hundred concurrent users at your database and a constant stream of INSERTS and UPDATES with quite a few DELETEs sprinkled in, and you'll start reading Oracle literature and eyeing your war chest. However, I've got a solution for you that will only cost you a code review, a few minor tweaks, and a system test. You do have a system test procedure in place, right?

If you used Streamload.com at all during June, July, and August, you probably got a "You were the deadlock loser" error, or a "Lock timeout" error, or an "Object required" error. These were all caused by lock contention. After scouring the documentation and talking to a few people, I learned what I summarized above and will say again here:

* SQL Server starts with row-level locks, but often escalates these to page and table locks, causing deadlocks
* SQL Server requires locks for reading from the database (SELECTs), so even folks not trying to modify the database are affected by the lock system.

Fortunately, I stumbled across some obscure keywords from the SQL Server lexicon: NOLOCK and ROWLOCK. They are used like this:

| SELECT COUNT(UserID) FROM Users WITH (NOLOCK) WHERE Username LIKE 'foobar' |
| --- |

and

| UPDATE Users WITH (ROWLOCK) SET Username = 'fred' WHERE Username = 'foobar' |
| --- |

What do these extra incantations do? We'll examine these two hints - NOLOCK and ROWLOCK - on the next page!

In the last section we discussed what row-locking was, why it was done, and its impact on database performance. In this part we'll examine two query-level hints you can provide SQL Server with to specify how you want SQL to handle locking!

**NOLOCK**  
Using NOLOCK politely asks SQL Server to ignore locks and read directly from the tables. This means you completely circumvent the lock system, which is a major performance and scalability improvement. However, you also completely circumvent the lock system, which means your code is living dangerously. You might read the not-necessarily-valid uncommitted modifications of a running transaction. This is a calculated risk.

For financial code and denormalized aggregates (those little counters of related data that you stash away and try desperately to keep accurate), you should play it safe and not use this technique. But I think you'll find that for better than 90% of your application, it would not be that big of a deal if a user (or even intermediate code) saw an uncommitted modification. In fact, you'll probably find that most of your data never or only very rarely changes, in which case the overhead of locking the data is almost always completely wasted.

For example, if I want to count all users that joined Streamload.com between June 1 and August 31 of Y2K, there's no reason for me to lock anything: that number was cast in stone the moment September 1, 2000 rolled around. Another example is the file listings you see on Streamload.com: it doesn't much matter if you don't see the exact perfect data, since either you don't own the data and it doesn't much matter what you see, or you do own the data and you know perfectly well whether you just modified the data or not and whether new files have finished uploading.

Just don't use this type of data as the basis for modifications to the database, and don't use it when it's really important that the user not see the wrong thing (an account statement or balance, for instance).

**ROWLOCK**  
Using ROWLOCK politely asks SQL Server to only use row-level locks. You can use this in SELECT, UPDATE, and DELETE statements, but I only use it in UPDATE and DELETE statements. You'd think that an UPDATE in which you specify the primary key would always cause a row lock, but when SQL Server gets a batch with a bunch of these, and some of them happen to be in the same page (depending on this situation, this can be quite likely, e.g. updating all files in a folder, files which were created at pretty much the same time), you'll see page locks, and bad things will happen. And if you don't specify a primary key for an UPDATE or DELETE, there's no reason the database wouldn't assume that a lot won't be affected, so it probably goes right to page locks, and bad things happen.

By specifically requesting row-level locks, these problems are avoided. However, be aware that if you are wrong and lots of rows are affected, either the database will take the initiative and escalate to page locks, or you'll have a whole army of row locks filling your server's memory and bogging down processing. One thing to be particularly aware of is the "Management/Current Activity" folder with Enterprise Manager. It takes a long time to load information about a lot of locks. The information is valuable, and this technique is very helpful, but don't be surprised if you see hundreds of locks in the "Locks/Processes" folder after employing this technique. Just be glad you don't have lock timeouts or deadlocks.

**Notes:**  
I get the sense that SQL Server honors NOLOCK requests religiously, but is more discretional with ROWLOCK requests. You can only use NOLOCK in SELECT statements. This includes inner queries, and the SELECT clause of the INSERT statement. You can and should use NOLOCK in joins:

| SELECT COUNT(Users.UserID) FROM Users WITH (NOLOCK)    JOIN UsersInUserGroups WITH (NOLOCK) ON        Users.UserID = UsersInUserGroups.UserID |
| --- |

**Results**  
It's difficult to quantify the performance gain had by applying these techniques to Streamload.com, and impossible to speculate as to the effects this would have on your site. Before we did it, the site was slow, often unusable, and always unreliable. After we did it, the site was fast, usable, and reliable. Truly, it was a night and day improvement. And you won't find this if you go searching through the documentation for help with lock contention. The docs recommend rewriting your app so that tables are referenced - and hence, locks are attained - in the same order throughout (yeah, right!), keeping transactions short and in one batch (a good idea, but in practice "yeah, right!"), use a low isolation level (also a good idea: NOLOCK takes this to an extreme), and use bound connections to allow processes to (share locks and) cooperate (sounds like a very complicated bad idea). I don't get the sense the consultants of the world are aware of (or comfortable with?) this technique either. But you heard it here, and it's worked great for Streamload.com. If you're having lock contention problems with SQL Server, it could work for your site, too.

**Big Disclaimer**  
Use these techniques with caution and discretion. The way I approached it was to look at all my stored procedures and ad hoc queries, and based on my understanding of where and how they were used, I decided whether it would be acceptable for the caller or user to get possibly incorrect results for NOLOCK, and whether it was likely that more than a few dozen rows would be locked with ROWLOCK. In almost all cases it was fine, but maybe for your code you should be more careful. You might need to produce separate procedures based on whether or to lock, and how to lock. There are other incantations (PAGLOCK, TABLOCK) which you might want to use when you know the UPDATE or DELETE query will affect many rows.

**Transact-SQL Optimization Tips**  
Here are fourteen little known tips that you can use to ensure your Transact-SQL queries are performing in the most efficient manner possible.

**1. Try to restrict the queries result set by using the WHERE clause.**

This can result in a performance benefit, as SQL Server will return to the client only particular rows, not all rows from the table(s). This can reduce network traffic and boost the overall performance of the query.

**2. Try to restrict the queries result set by returning only the particular columns from the table, not all the table's columns.**

This can result in a performance benefit as well, because SQL Server will return to the client only particular columns, not all the table's columns. This can reduce network traffic and boost the overall performance of the query.

**3. Use views and stored procedures instead of heavy-duty queries.**

This can reduce network traffic as your client will send to the server only stored procedures or view name (perhaps with some parameters) instead of large heavy-duty queries text. This can be used to facilitate permission management also, because you can restrict user access to table columns they should not see.

**4. Whenever possible, try to avoid using SQL Server cursors.**

SQL Server cursors can result in some performance degradation in comparison with select statements. Try to use correlated subquery or derived tables, if you need to perform row-by-row operations.

**5. If you need to return the total table's row count, you can use an alternative way instead of the SELECT COUNT(\*) statement.**

Because the SELECT COUNT(\*) statement makes a full table scan to return the total table's row count, it can take an extremely long time for large tables. There is another way to determine the total row count in a table. In this case, you can use the sysindexes system table. There is a ROWS column in the sysindexes table. This column contains the total row count for each table in your database. So, you can use the following select statement instead of SELECT COUNT(\*):

SELECT rows FROM sysindexes WHERE id = OBJECT\_ID('table\_name') AND indid < 2

This way, you can improve the speed of such queries by several times. See this article for more details: [**Alternative way to get the table's row count.**](http://www.mssqlcity.com/Articles/KnowHow/RowCount.htm)

http://www.mssqlcity.com/Articles/KnowHow/RowCount.htm

**6. Try to use constraints instead of triggers, whenever possible.**

Constraints are much more efficient than triggers and can boost performance. So, whenever possible, you should use constraints instead of triggers.

**7. Use table variables instead of temporary tables.**

Table variables require fewer locking and logging resources than temporary tables, so table variables should be used whenever possible. The table variables are available in SQL Server 2000 only.

**8. Try to avoid the HAVING clause, whenever possible.**

The HAVING clause is used to restrict the result set returned by the GROUP BY clause. When you use GROUP BY with the HAVING clause, the GROUP BY clause divides the rows into sets of grouped rows and aggregates their values, and then the HAVING clause eliminates undesired aggregated groups. In many cases, you can write your select statement so that they will contain only WHERE and GROUP BY clauses without the HAVING clause. This can improve the performance of your query.

**9. Whenever possible, try to avoid using the DISTINCT clause.**

Because using the DISTINCT clause will result in some performance degradation, you should use this clause only when it is absolutely necessary.

**10. Include SET NOCOUNT ON statement into your stored procedures to stop the message indicating the number of rows affected by a T-SQL statement.**

This can reduce network traffic, as your client will not receive the message indicating the number of rows affected by a T-SQL statement.

**11. Use select statements with the TOP keyword or the SET ROWCOUNT statement if you need to return only the first n rows.**

This can improve performance of your queries, as a smaller result set will be returned. This can also reduce the traffic between the server and the clients.

**12. Use the FAST number\_rows table hint if you need to quickly return 'number\_rows' rows.**

You can quickly get the n rows and can work with them when the query continues execution and produces its full result set.

**13. Try to use UNION ALL statement instead of UNION, whenever possible.**

The UNION ALL statement is much faster than UNION, because UNION ALL statement does not look for duplicate rows, while the UNION statement does look for duplicate rows, whether they exist or not.

**14. Do not use optimizer hints in your queries.**

Because the SQL Server query optimizer is very clever, it is highly unlikely that you can optimize your query by using optimizer hints; more often than not, this will hurt performance.

**Transaction Isolation Level**  
In this article I want to tell you about Transaction Isolation Level in SQL Server 6.5 and SQL Server 7.0, what kind of Transaction Isolation Level exists, and how you can set the appropriate Transaction Isolation Level.

There are four isolation levels:

READ UNCOMMITTED

READ COMMITTED

REPEATABLE READ

SERIALIZABLE

SQL Server 6.5 supports all of these Transaction Isolation Levels, but has only three different behaviors, because in SQL Server 6.5 REPEATABLE READ and SERIALIZABLE are synonyms. It because SQL Server 6.5 supports only page locking (there is no full support of row locking as in SQL Server 7.0) and if REPEATABLE READ isolation level was set, then another transaction cannot insert the row before the first transaction was finished, because page will be locked. So there are no phantoms in SQL Server 6.5, if REPEATABLE READ isolation level was set.

SQL Server 7.0 supports all of these Transaction Isolation Levels and can separate REPEATABLE READ and SERIALIZABLE. Let me to describe each isolation level.

### READ UNCOMMITTED

When it's used, SQL Server not issue shared locks while reading data. So, you can read an uncommitted transaction that might get rolled back later. This isolation level is also called dirty read. This is the lowest isolation level.

It ensures only that a physically corrupt data will not be read.

### READ COMMITTED

This is the default isolation level in SQL Server. When it's used, SQL Server will use shared locks while reading data. It ensures that a physically corrupt data will not be read and will never read data that another application has changed and not yet committed, but it not ensures that the data will not be changed before the end of the transaction.

### REPEATABLE READ

When it's used, then dirty reads and nonrepeatable reads cannot occur. It means that locks will be placed on all data that is used in a query, and another transactions cannot update the data.

This is the definition of nonrepeatable read from SQL Server Books Online:

| nonrepeatable read  When a transaction reads the same row more than one time, and between the  two (or more) reads, a separate transaction modifies that row. Because the  row was modified between reads within the same transaction, each read  produces different values, which introduces inconsistency. |
| --- |

### SERIALIZABLE

Most restrictive isolation level. When it's used, then phantom values cannot occur. It prevents other users from updating or inserting rows into the data set until the transaction is complete.

This is the definition of phantom from SQL Server Books Online:

| phantom  Phantom behavior occurs when a transaction attempts to select a row that  does not exist and a second transaction inserts the row before the first  transaction finishes. If the row is inserted, the row appears as a phantom  to the first transaction, inconsistently appearing and disappearing. |
| --- |

You can set the appropriate isolation level for an entire SQL Server session with the SET TRANSACTION ISOLATION LEVEL statement. This is the syntax from SQL Server Books Online:

| SET TRANSACTION ISOLATION LEVEL  {  READ COMMITTED  | READ UNCOMMITTED  | REPEATABLE READ  | SERIALIZABLE  } |
| --- |

You can use DBCC USEROPTIONS command to determine the Transaction Isolation Level currently set. This command returns the set options that are active for the current connection. This is the example:

| SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED  GO  DBCC USEROPTIONS  GO |
| --- |

These are the results:

Set Option Value

------------------------------ ------------------------------------

textsize 64512

language us\_english

dateformat mdy

datefirst 7

isolation level read uncommitted

**SQL Server 2000: Some Useful Undocumented DBCC Commands**

· DBCC BUFFER · DBCC BYTES · DBCC DBINFO · DBCC DBTABLE

· DBCC DES · DBCC HELP · DBCC IND · DBCC LOG · DBCC PAGE · DBCC PROCBUF   
· DBCC PRTIPAGE · DBCC PSS · DBCC RESOURCE · DBCC TAB

## Introduction

In this article I want to tell you about some useful undocumented DBCC commands, and how you can use these commands in SQL Server 2000 for administering and monitoring. DBCC is an abbreviation of a Database Console Command. This is the description of DBCC from SQL Server Books Online: The Transact-SQL programming language provides DBCC statements that act as Database Console Commands for Microsoft SQL Server 2000.These statements check the physical and logical consistency of a database. Many DBCC statements can fix detected problems.

## Undocumented DBCC commands

Undocumented DBCC commands

**1. DBCC BUFFER**

This command can be used to print buffer headers and pages from the

buffer cache.

Syntax:

dbcc buffer ([dbid|dbname] [,objid|objname] [,nbufs], [printopt])

where

dbid|dbname - database id|database name.

objid|objname - object id|object name

nbufs - number of buffers to examine

printopt - print option

0 - print out only the buffer header and page header

(default)

1 - print out each row separately and the offset table

2 - print out each row as a whole and the offset table

This is the example:

DBCC TRACEON (3604)

dbcc buffer(master,'sysobjects')

**2. DBCC BYTES**

This command can be used to dump out bytes from a specific address.

Syntax:

dbcc bytes ( startaddress, length )

where

startaddress - starting address to dump

length - number of bytes to dump

This is the example:

DBCC TRACEON (3604)

dbcc bytes (10000000, 100)

**3. DBCC DBINFO**

Print DBINFO structure for specified database.

Syntax:

DBCC DBINFO [( dbname )]

where

dbname - is the database name.

This is the example:

DBCC TRACEON (3604)

DBCC DBINFO (master)

**4. DBCC DBTABLE**

This command prints out the contents of the DBTABLE structure.

Syntax:

DBCC DBTABLE ({dbid|dbname})

where

dbid|dbname - database name or database ID

This is the example:

DBCC TRACEON (3604)

DBCC DBTABLE (master)

The DBTABLE structure has an output parameter called dbt\_open.

This parameter keeps track of how many users are in the database.

Look at here for more details:

FIX:

Database Usage Count Does Not Return to Zero

**5. DBCC DES**

Prints the contents of the specified DES (descriptor).

Syntax:

dbcc des [( [dbid|dbname] [,objid|objname] )]

where dbid|dbname - database id|database name.

objid|objname - object id|object name

This is the example:

DBCC TRACEON (3604)

DBCC DES

**6. DBCC HELP**

DBCC HELP returns syntax information for the specified DBCC statement.

In comparison with DBCC HELP command in version 6.5, it returns syntax

information only for the documented DBCC commands.

Syntax:

DBCC HELP ('dbcc\_statement' | @dbcc\_statement\_var | '?')

This is the example:

DBCC TRACEON (3604)

DECLARE @dbcc\_stmt sysname

SELECT @dbcc\_stmt = 'CHECKTABLE'

DBCC HELP (@dbcc\_stmt)

Look at here for more details:

DBCC HELP

(T-SQL)

**7. DBCC IND**

Shows all pages in use by indexes of the specified table.

Syntax:

dbcc ind( dbid|dbname, objid|objname, printopt = {-2|-1|0|1|2|3} )

where

dbid|dbname - database id|database name.

objid|objname - object id|object name

printopt - print option

There is change in this command in comparison with SQL Server 7.0 one:

you should specify printopt parameter (this parameter is not optional now)

This is the example:

DBCC TRACEON (3604)

DBCC IND (master, sysobjects, 0)

**8. DBCC log**

This command is used to view the transactional log for the specified

database.

Syntax:

DBCC log ( {dbid|dbname}, [, type={-1|0|1|2|3|4}] )

PARAMETERS:

Dbid or dbname - Enter either the dbid or the name of the database

in question.

type - is the type of output:

0 - minimum information (operation, context, transaction id)

1 - more information (plus flags, tags, row length, description)

2 - very detailed information (plus object name, index name,

page id, slot id)

3 - full information about each operation

4 - full information about each operation plus hexadecimal dump

of the current transaction log's row.

-1 - full information about each operation plus hexadecimal dump

of the current transaction log's row, plus Checkpoint Begin,

DB Version, Max XDESID

by default type = 0

To view the transaction log for the master database, you can run the

following command:

DBCC log (master)

**9. DBCC PAGE**

You can use this command to view the data page structure.

Syntax:

DBCC PAGE ({dbid|dbname}, pagenum [,print option] [,cache] [,logical])

PARAMETERS:

Dbid or dbname - Enter either the dbid or the name of the database

in question.

Pagenum - Enter the page number of the SQL Server page that is to

be examined.

Print option - (Optional) Print option can be either 0, 1, or 2.

0 - (Default) This option causes DBCC PAGE to print

out only the page header information.

1 - This option causes DBCC PAGE to print out the

page header information, each row of information

from the page, and the page's offset table. Each

of the rows printed out will be separated from

each other.

2 - This option is the same as option 1, except it

prints the page rows as a single block of

information rather than separating the

individual rows. The offset and header will also

be displayed.

Cache - (Optional) This parameter allows either a 1 or a 0 to be

entered.

0 - This option causes DBCC PAGE to retrieve the page

number from disk rather than checking to see if it is

in cache.

1 - (Default) This option takes the page from cache if it

is in cache rather than getting it from disk only.

Logical - (Optional) This parameter is for use if the page number

that is to be retrieved is a virtual page rather then a

logical page. It can be either 0 or 1.

0 - If the page is to be a virtual page number.

1 - (Default) If the page is the logical page number.

This is the example:

DBCC TRACEON (3604)

DBCC PAGE (master, 1, 1)

Look at here for more details:

Data

page structure in MS SQL 6.5

10. DBCC procbuf

This command prints procedure buffer headers and proc-headers from

the procedure cache.

Syntax:

DBCC procbuf( [dbid|dbname], [objid|objname], [nbufs], [printopt = {0|1}] )

where

dbid|dbname - database id|database name.

objid|objname - object id|object name

nbufs - number of buffers to print

printopt - print option

(0 print out only the proc buff and proc header (default)

1 print out proc buff, proc header and contents of buffer)

This is the example:

DBCC TRACEON (3604)

DBCC procbuf(master,'sp\_help',1,0)

**11. DBCC prtipage**

This command prints the page number pointed to by each row on the

specified index page.

Syntax:

DBCC prtipage( dbid, objid, indexid, indexpage )

where

dbid - database ID

objid - object ID

indexid - index ID

indexpage - the logical page number of the index page to dump

This is the example:

DBCC TRACEON (3604)

DECLARE @dbid int, @objectid int

SELECT @dbid = DB\_ID('master')

SELECT @objectid = object\_id('sysobjects')

DBCC prtipage(@dbid,@objectid,1,0)

**12. DBCC pss**

This command shows info about processes currently connected to the

dataserver. Structure is the source of data contained in the sysprocesses

table.

Syntax:

DBCC pss( suid, spid, printopt = { 1 | 0 } )

where

suid - server user ID

spid - server process ID

printopt - print option

(0 standard output,

1 all open DES's and current sequence tree)

This is the example:

DBCC TRACEON (3604)

dbcc pss

**13. DBCC resource**

This command shows dataserver level RESOURCE, PERFMON and DS\_CONFIG

info. RESOURCE shows addresses of various data structures used by

the server. PERFMON structure contains master..spt\_monitor

field info. DS\_CONFIG structure contains master..syscurconfigs

field info.

Syntax:

DBCC resource

This is the example:

DBCC TRACEON (3604)

DBCC resource

**14. DBCC TAB**

You can use the following undocumented command to view the data

pages structure (in comparison with DBCC PAGE, this command will

return information about all data pages for viewed table, not

only for particular number)

Syntax:

DBCC tab (dbid, objid)

where

dbid - is the database id

objid - is the table id

This is the example:

DBCC TRACEON (3604)

DECLARE @dbid int, @objectid int

SELECT @dbid = DB\_ID('master')

SELECT @objectid = object\_id('sysdatabases')

DBCC TAB (@dbid,@objectid)

[**Database**](http://www.techinterviews.com/index.php?cat=12): http://www.techinterviews.com/index.php?cat=12

# [**Interview questions for tech companies**](http://www.techinterviews.com)

Administration

***How do I administer SQL Server remotely?***

You can install Enterprise Manager utility and then administer remote server just as you administer a local server. If you need to administer SQL Server over the internet, you should specify the TCP/IP address instead of the remote SQL Server's name under the 'New SQL Server Registation' window.

***How do I go about using/changing my settings to use the SANs protocol?***

SAN - is the System Area Network protocols, which supported under SQL Server 2000 Enterprise Edition only and built using the Virtual Interface Architecture (VIA). SANs are intended to support the high communications bandwidth between servers.

To use SANs protocol, you can specify Giganet VIA SAN Net-Libraries on the client's computer:

1. Run SQL Server Client Network Utility.

2. On the General tab enable VIA protocol.

3. Choose VIA and click Properties button.

4. Specify Giganet as Vendor and close Client Network Utility.

***How do I mark the stored procedure to automatic execution?***

You can use the sp\_procoption system stored procedure to mark the stored procedure to automatic execution when the SQL Server will start.

Note. Only objects in the master database owned by dbo can have the startup setting changed and this option is restricted to objects that have no parameters.

This is the example to set the startup option for the indRebuild stored procedure in the master database:

| USE master  EXEC sp\_procoption 'indRebuild', 'startup', 'true' |
| --- |

***How do I report information about current users and processes?***

To report information about current users and processes, you can use the **sp\_who** and **sp\_who2** system stored procedures. The **sp\_who** system stored procedure described in SQL Server Books Online, the **sp\_who2** is unducumented stored procedure. **sp\_who2** provides more detailed information about current users and processes than **sp\_who**. The **sp\_who2** syntax is equal to **sp\_who**.

Syntax:

sp\_who2 [[@login\_name =] 'login']

Arguments:

[@login\_name =] 'login'

Is a user login name on SQL Server. login is sysname, with a default of NULL. If no name is specified, the procedure reports all active users of SQL Server.

***How do I report information about the active locks?***

To report information about the active locks, you can use the **sp\_lock** system stored procedure.

***How can I add a user-defined error message?***

You can use the **sp\_addmessage** system stored procedure to add a user-defined error message.

You can specify the id of the message (value greater than 50000), severity level from 0 through 25 (to add a message with a severity level from 19 through 25, you must be a member of the sysadmin fixe server role) and the message text (up to 255 characters).

By the way, only members of the sysadmin and serveradmin fixed server roles can execute the **sp\_addmessage** system stored procedure.

This is the example:

| USE master  EXEC sp\_addmessage @msgnum = 50001,  @severity = 18,  @msgtext = 'Custom error message' |
| --- |

***How can I change the database owner?***

You can use the **sp\_changedbowner** system stored procedure to change the database owner.

This is the example to make the user John the owner of the pubs database:

| USE pubs  GO  EXEC sp\_changedbowner 'John'  GO |
| --- |

***How can I change the object owner?***

You can use the **sp\_changeobjectowner** system stored procedure to change the object owner.

This is the example to change the owner of the authors table in the pubs database to John:

| USE pubs  GO  EXEC sp\_changeobjectowner 'authors', 'John'  GO |
| --- |

***How can I change the sort order and character set?***

You cannot make it without rebuilding your databases.

This is from SQL Server Books Online:

| It is critical that you select the correct sort order when you install SQL Server.  If you need to change sort orders after installation, you must rebuild your databases and reload your data. |
| --- |

and

| It is critical that you select the correct character set when you install SQL Server.  To change character sets after installing SQL Server, you must rebuild the databases and reload the data. |
| --- |

***How can I compare the data in two tables with identical structure?***

1. Compare the row counts in these tables by using the COUNT(\*) function:

*SELECT COUNT(\*) FROM tb1*

*SELECT COUNT(\*) FROM tb2*

2. If the row counts in the tables is identical, you can download the data from these tables into files on the disk by using the bcp utility and compare the size of these files.

To get more information about what rows are differ, you can compare these files by using the comp.exe utility (C:\WINNT\system32 - path by default for Windows NT).

3. You can use red-gate SQL compare to do this. It can compare tables, constraints, rules, stored procedures and creates a script for applying the changes from one database to another.

***How can I compare two databases?***

You can use red-gate SQL compare to do this. It can compare tables, constraints, rules, stored procedures and creates a script for applying the changes from one database to another.

You can download the 14 days trial version at here:

http://www.red-gate.com

You can use the following scripts from the swynk.com:

http://www.swynk.com/downloads/objcompare.zip

| This program is similar to Sqlcomp from Backoffice Resource Kit. With this program your can compare Stored Procedures and Views from different servers (example: Production and Developer) and versions (6.5 and 7.0). |
| --- |

http://www.swynk.com/downloads/sp\_db\_comp.sql

| This script will compare all tables in two databases including column names, datatypes and NULL options. It also lists tables that exist only in one of databases. |
| --- |

http://www.swynk.com/downloads/sp\_ABCompareDb.sql

| Compare a source and target database. List all procedures, tables, views, and triggers on the source that either don't exist on the target, or are later on the source than the target machine. This is useful when a series of changed objects need to be installed from a development server to a production server. |
| --- |

***How can I get detailed information about locking?***

You can use the **sp\_lock** system stored procedure.

Read about the **sp\_lock** stored procedure in SQL Server Books Online.

To get more detailed locking view, use the sp\_lock2 stored procedure from this article:

Detailed locking view: **sp\_lock2** (http://www.mssqlcity.com/Articles/Adm/LockView.htm)

***How can I grant to user the RESTORE DATABASE permission?***

The RESTORE DATABASE permission cannot be granted or revoked. If you need to grant to the user the restore permission, add the user to the dbcreator fixed server role. You can use the **sp\_addsrvrolemember** system stored procedure to add the user to a fixed server role.

This is the example of using the **sp\_addsrvrolemember** stored procedure to add **John** to the dbcreator fixed server role:

*EXEC sp\_addsrvrolemember 'John', 'dbcreator'*

See SQL Server Books Online for more details about the **sp\_addsrvrolemember** system stored procedure.

***How can I rebuild the SQL Server repository?***

This procedure is not easy.

See this link from Microsoft:

How to Rebuild the Repository in SQL Server 7.0

([**http://support.microsoft.com/support/kb/articles/Q238/8/22.asp**](http://support.microsoft.com/support/kb/articles/Q238/8/22.asp))

***How can I restore the user databases after reinstall SQL 7.0/2000?***

You can attach the user databases to a SQL Server or you can restore databases from the backup.

Read about the **sp\_attach\_db** stored procedure in SQL Server Books Online Title: **sp\_attach\_db (T-SQL)**.

***How can I show/hide system databases and system objects?***

Try this:

1. Run Enterprise Manager.

2. Right-click the server icon and select 'Edit SQL Server Registration properties'.

3. To show system databases and system objects check 'Show system databases and system objects'.

To hide system databases and system objects uncheck 'Show system databases and system objects'.

***How can I start MSSQLServer service at 6:30am and stop it at 6:30pm automatically?***

You can use AT command for this purpose.

AT - is a Windows NT file (c:\winnt\system32\at.exe - path by default).

It's full description of the AT command:

| The AT command schedules commands and programs to run on a computer at a specified time and date. The Schedule service must be running to use the AT command.  AT [\\computername] [ [id] [/DELETE] | /DELETE [/YES]]  AT [\\computername] time [/INTERACTIVE]  [ /EVERY:date[,...] | /NEXT:date[,...]] "command"  \\computername : Specifies a remote computer. Commands are scheduled on the local  computer if this parameter is omitted.  id : Is an identification number assigned to a scheduled command.  /delete : Cancels a scheduled command. If id is omitted, all the scheduled  commands on the computer are canceled.  /yes : Used with cancel all jobs command when no further confirmation is  desired.  time : Specifies the time when command is to run.  /interactive : Allows the job to interact with the desktop of the user who is logged on  at the time the job runs.  /every:date[,...] : Runs the command on each specified day(s) of the week or month.  If date is omitted, the current day of the month is assumed.  /next:date[,...] : Runs the specified command on the next occurrence of the day  (for example, next Thursday). If date is omitted, the current day of the  month is assumed.  "command"     : Is the Windows NT command, or batch program to be run. |
| --- |

To start MSSQLServer service at 6:30am and stop it at 6:30pm, run this:

at 6:30pm /EVERY:monday,tuesday,wednesday,thursday,friday,saturday,sunday "c:\stop.bat"

at 6:30am /EVERY:monday,tuesday,wednesday,thursday,friday,saturday,sunday "c:\start.bat"

Where start.bat:

**net start mssqlserver**

Stop.bat:

**net stop mssqlserver**

***I am a dbowner. Can I restrict access to my database for the 'sa'?***

No, you cannot. Members of the **sysadmin** fixed server role have full access to all system and user databases.

See also this link to get more information about SQL Server roles:

Understanding SQL Server roles (http://www.mssqlcity.com/Articles/Adm/SQL70Roles.htm)

***I changed the name of NT server. How can I change the name of SQL Server?***

With SQL Server 7.0/2000 you need to rerun setup to reset the server name. Setup will detect the name conflict, resolve it, and then finish.

***I forgot the sa password. How I can reset it?***

To reset the sa password, you can make the following:

1. Login to the SQL Server box as the Administrator.

2. Run SQL Server Enterprise Manager.

3. Right-click the server name and choose 'Edit SQL Server Registration properties'.

4. Choose 'Use Windows authentication' and click OK button.

5. Expand a server, expand a Security and click Logins.

6. Double-click the sa login and specify new password on the General tab.

Backup/Restore

***Can I backup a database to a network device?***

Yes, you can. You should change the account the MSSQL Server service runs under to the account belonging to the Local Administrators group and to the Domain Users group of the appropriate domain, because the Local System account has no access to shares on the network.

***Can I restore a SQL 7.0 database backup to a SQL 2000 server and vice versa?***

You can restore only the users databases from the SQL Server 7.0 database backup to a SQL Server 2000 server, but you cannot restore system databases (such as **master**, **msdb**, **model** and **tempdb**) from the SQL Server 7.0 backup to a SQL Server 2000.

You cannot restore a SQL Server 2000 database backup to a SQL Server 7.0 server.

***How can I restore a SQL 6.5 database backup to a SQL 7.0/2000 server?***

You cannot use SQL Server 6.5 backup in SQL Server 7.0/2000 and vice versa. You should previous upgrade SQL Server 6.5 to SQL Server 7.0/2000.

***Can I restore from backup with another sort order or character set?***

Usually, you cannot. See the SQL Server Books Online. It's there from:

| Usually, the sort order and character set of two servers must match to backup on one server and restore on the other. An exception is when both servers use a binary sort order; then, the character sets can differ. When the code page or sort order differs, use the bcp utility or Data Transformation Services to move the data. |
| --- |

***How can I backup/restore a single table in SQL 7.0/2000?***

This opportunity is no longer supported. But you can create new filegroup and place a table in it. Because SQL Server can restore both files and filegroups, you can restore only that table.

***How do I restore the master database?***

Yes, there is no such database option for the **master** database. There is SQL Server configuration option. You need to run SQL Server in the single-user mode. Try the following:

1. Stop MSSQLServer and SQLServerAgent services.

2. From a command prompt, enter this command:

*sqlservr.exe -m*

3. Run Enterprise Manager to restore the **master** database from the backup.

***Is it possible to restore a particular transaction from a SQL Server backup?***

No, it is not possible. The only thing you can do is, using transaction log backups, restore up to a given point in time.

***We have no any backup, can we restore deleted data?***

Very popular question in newsgroups : No, you cannot. After transaction was committed, you cannot rollback it. You should have backup to restore deleted data.

Connectivity

***Can I connect to SQL Server over the internet?***

Yes, you can.

This is from SQL Server Books Online:

| To share data over the Internet, the client and server must be connected to the Internet. In addition, you must use the TCP/IP or Multiprotocol Net-Libraries. If you use the Multiprotocol Net-Library, ensure that TCP/IP support is enabled. If the server is registered with Domain Name System (DNS), you can connect using its registered (friendly) name. |
| --- |

***Can I use the SQL Server 7.0 tools to connect to SQL Server 2000?***

Yes, you can use the SQL Query Analyzer, osql and isql SQL Server 7.0 tools to connect to SQL Server 2000.

To connect to a named instance of SQL Server 2000, you must set up an alias using the Client Network Utility.

This is from SQL Server 2000 Books Online:

| When using the SQL Server client connectivity components from SQL Server 7.0 or earlier, you must set up an alias using the Client Network Utility before you connect to a named instance of SQL Server 2000. For example, on a SQL Server 7.0 client, to connect to a named instance of SQL Server 2000, you must add an alias that points to  \\computername\pipe\MSSQL$instancename\sql\query. If you use an alias name of computername\instancename, clients can connect by specifying this name in the same way as SQL Server 2000 clients do. For the TCP/IP Sockets and NWLink IPX/SPX Net-Libraries, you must use the Client Network Utility to define an alias on the client that specifies the port address on which the named instance is listening. |
| --- |

***How can I connect to SQL Server through a firewall?***

You should ensure that SQL Server uses a specific port number for its incoming connections. The firewall system then forwards connections from the port number to SQL Server.

***How do I connect to SQL Server through a firewall?***

Basically you have to open up the ports that SQL Server uses. If you've done this and it still doesn't work then look at the firewall logs to see what packets it is dropping or do a network trace either side of the firewall to see what packets are not getting through. (You may want to disable/allow all through the firewall during testing to see what extra packets are allowed through).

Which ports to open depends on the net-lib you are using :-

For tcp-ip sockets the default port for SQL Server is 1433.

For multi-protocol (rpc) the ports are normally variable, but you can fix them. See Q164667 in the Microsoft knowledgebase for details.

For named-pipes over ip 137/138/139 are used. As these are the same ones used for file/print it is not recommended you allow these through the firewall.

***What sql net-lib should I use to enable encryption?***

You should use the multi-protocol network library to enable encryption.

1. Run **Server Network Utility**.

2. Add **Multiprotocol** if it does not appear in the **Server network** library list.

3. On the General tab, select **Multiprotocol**, click **Edit** button, and select **Enable multi-protocol encryption**.

**Note:** You do not need to use the multi-protocol network library to enable encryption in SQL Server 2000. SQL Server 2000 can use the Secure Sockets Layer (SSL) to encrypt all data transmitted between an application computer and a SQL Server instance on a database computer.

***What sql net-lib should I use to improve the speed of my network connections?***

The TCP/IP Sockets is the most fastest sql net-lib.

You can see the difference in responses times between any of the net-libs only on the slow network connections, on the normal LAN, you cannot see the difference in responses times.

This is from SQL Server Books Online:

| In general, sockets are preferred in a slow LAN, WAN, or dial-up network, whereas named pipes can be a better choice when network speed is not the issue, as it offers more functionality, ease of use, and configuration options. |
| --- |

***Why dialup networking invoked when database is opened?***

Read about "Appearance of Internet Connection Dialog Box at Startup of Windows 95/98" in SQL Server Books Online (for both SQL Server 7.0/2000).

It's therefrom:

| This behavior can be disabled by setting the registry key EnableRemoteConnect to N.  To do this, create a text file named DisbleAutoConnect.reg with the following three lines:  REGEDIT4  [HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\OLE]  "EnableRemoteConnect"="N" |
| --- |

Development

***Can I work with registry from within T-SQL?***

Yes, you can. There are many extended stored procedures shipped with SQL Server and not all of them have been described in the SQL Server manuals. There are four very useful extended stored procedures to work with registry. There are: **xp\_regread**, **xp\_regwrite**, **xp\_regdeletevalue** and **xp\_regdeletekey**.

See this article for more details: Useful undocumented extended stored procedures

(http://www.mssqlcity.com/Articles/Undoc/UndocExtSP.htm)

***How can I check from the SQL Server whether the file exists or not?***

You can use the **xp\_fileexist** undocumented extended stored procedure to determine whether the particular file exists on the disk or not.

Syntax:

*EXECUTE xp\_fileexist filename [, file\_exists INT OUTPUT]*

To check whether the file boot.ini exists on the disk C: or not, run:

*EXEC master..xp\_fileexist 'c:\boot.ini'*

There are many extended stored procedures shipped with SQL Server and not all of them have been described in the SQL Server manuals. See this article to get the useful undocumented extended stored procedures description: Useful undocumented extended stored procedures

(http://www.mssqlcity.com/Articles/Undoc/UndocExtSP.htm)

***How can I copy text or image into or out of SQL Server?***

You can use textcopy utility.

See this article for more information about the textcopy utility: Copy text or image into or out of SQL Server (http://www.mssqlcity.com/Articles/KnowHow/Textcopy.htm)

***How can I create extended stored procedures?***

To create your own extended stored procedure, you can use Visual C++ or Delphi.

See these links to create an extended stored procedure by using Visual C++:

Creating Extended Stored Procedures

(http://archive.devx.com/free/mgznarch/vcdj/1998/mar98/sql1.asp)

Enhance SQL Server with your own Extended Stored Procedures

See this link to create an extended stored procedure by using Delphi:

Extended Stored Procedure component for Delphi (http://www.bramc.ru/soft/xproc.html)

After you create your own extended stored procedure, you should register the name of a new extended stored procedure to SQL Server by using the **sp\_addextendedproc** system stored procedure.

Read about the **sp\_addextendedproc** system stored procedure in SQL Server Books Online.

See this article for more details about extended stored procedures: HOW TO: Use Extended Stored Procedures (http://support.microsoft.com/support/kb/articles/q190/9/87.asp)

***How can I decrypt a SQL Server stored procedure?***

Here is the code for SQL 6.5

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* REVISED REC'D 5/21/98 \*/

/\* sp\_decrypt\_object Tom Sager 01/26/98 \*/

/\* \*/

/\* Decrypts objects (views, procedures & trigs) \*/

/\* created with the WITH ENCRYPTION option. \*/

/\* \*/

/\* Uses the encrypt() built-in function to find \*/

/\* a plaintext string that encrypts to the same \*/

/\* value as stored in the text column of the \*/

/\* syscomments table. \*/

/\* \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

create proc sp\_decrypt\_object

(@objname varchar(30))

WITH ENCRYPTION

As

SET NOCOUNT ON

declare @errmsg varchar(80)

declare @encrtext varchar(255)

declare @decrtext varchar(255)

declare @testtext varchar(255)

declare @printline varchar(255)

declare @textlen int

declare @lup int

declare @match char(1)

declare @testchar smallint

declare @begblk smallint

declare @endblk smallint

if (select count(\*)

from sysobjects

where name = @objname) = 0

begin

select @errmsg = 'Object '

+@objname

+' not found in database '

+DB\_NAME()

print @errmsg

return 1

end

if (select count(\*) from sysobjects t1,

syscomments t2

where t1.name = @objname

and t1.id = t2.id

and t2.texttype & 4 != 0) = 0

begin

select @errmsg = 'Object '

+@objname

+' is not encrypted in database '

+DB\_NAME()

print @errmsg

return 1

end

DECLARE comments\_cursor CURSOR for

select t1.text

from syscomments t1,

sysobjects t2

where t1.id = t2.id

and t2.name = @objname

order by t1.colid

OPEN comments\_cursor

FETCH NEXT FROM comments\_cursor

INTO @encrtext

WHILE (@@fetch\_status <> -1)

BEGIN

IF (@@fetch\_status <> -2)

BEGIN

select @decrtext = REPLICATE(' ', 255)

select @textlen = DATALENGTH(@encrtext)

select @lup = 1

select @match = 'n'

while (@lup <= @textlen)

begin

select @testchar = 0

select @match = 'n'

while (@match = 'n')

begin

select @decrtext =

STUFF(@decrtext,@lup,1,CHAR(@testchar))

select @testtext = encrypt(@decrtext)

if ASCII(SUBSTRING(@testtext,@lup,1)) =

ASCII(SUBSTRING(@encrtext,@lup,1))

begin

select @match = 'y'

end

select @testchar = @testchar + 1

if (@testchar > 255)

begin

print 'Error...no match found'

return 1

end

end

select @lup = @lup + 1

end

select @begblk = 1

select @endblk = 1

while (@endblk <= @textlen)

begin

if (substring(@decrtext,@endblk,1) = 0x0a)

begin

select @printline = @printline +

SUBSTRING(@decrtext

,@begblk

,@endblk-@begblk+1)

print @printline

select @begblk = @endblk + 1

select @endblk = @begblk

select @printline = NULL

end

select @endblk = @endblk + 1

end

select @printline = @printline +

SUBSTRING(@decrtext

,@begblk

,@endblk-@begblk+1)

END

FETCH NEXT FROM comments\_cursor INTO @encrtext

END

print @printline

CLOSE comments\_cursor

DEALLOCATE comments\_cursor

GO

***How can I delete duplicate rows?***

Microsoft SQL Server tables should never contain duplicate rows, nor non-unique primary keys. For brevity, we will sometimes refer to primary keys as "key" or "PK" in this article, but this will always denote "primary key." Duplicate PKs are a violation of entity integrity, and should be disallowed in a relational system. SQL Server has various mechanisms for enforcing entity integrity, including indexes, UNIQUE constraints, PRIMARY KEY constraints, and triggers.

Despite this, under unusual circumstances duplicate primary keys may occur, and if so they must be eliminated. One way they can occur is if duplicate PKs exist in non-relational data outside SQL Server, and the data is imported while PK uniqueness is not being enforced. Another way they can occur is through a database design error, such as not enforcing entity integrity on each table.

Often duplicate PKs are noticed when you attempt to create a unique index, which will abort if duplicate keys are found. This message is:

Msg 1505, Level 16, State 1 Create unique index aborted on duplicate key.

This article discusses how to locate and remove duplicate primary keys from a table. However you should closely examine the process which allowed the duplicates to happen in order to preven a recurrence.

**MORE INFORMATION**

For this example, we will use the following table with duplicate PK values. In this table the primary key is the two columns (col1, col2). We cannot create a unique index or PRIMARY KEY constraint since two rows have duplicate PKs. This procedure illustrates how to identify and remove the duplicates.

| create table t1(col1 int, col2 int, col3 char(50))  insert into t1 values (1, 1, 'data value one')  insert into t1 values (1, 1, 'data value one')  insert into t1 values (1, 2, 'data value two') |
| --- |

The first step is to identify which rows have duplicate primary key values:

| SELECT col1, col2, count(\*)  FROM t1  GROUP BY col1, col2  HAVING count(\*) > 1 |
| --- |

This will return one row for each set of duplicate PK values in the table. The last column in this result is the number of duplicates for the particular PK value.

| col1 | col2 |  |
| --- | --- | --- |
| 1 | 1 | 2 |

If there are only a few sets of duplicate PK values, the best procedure is to delete these manually on an individual basis. For example:

| set rowcount 1  delete from t1  where col1=1 and col2=1 |
| --- |

The rowcount value should be n-1 the number of duplicates for a given key value. In this example, there are 2 duplicates so rowcount is set to 1. The col1/col2 values are taken from the above GROUP BY query result. If the GROUP BY query returns multiple rows, the "set rowcount" query will have to be run once for each of these rows. Each time it is run, set rowcount to n-1 the number of duplicates of the particular PK value.

Before deleting the rows, you should verify that the entire row is duplicate. While unlikely, it is possible that the PK values are duplicate, yet the row as a whole is not. An example of this would be a table with Social Security Number as the primary key, and having two different people (or rows) with the same number, each having unique attributes. In such a case whatever malfunction caused the duplicate key may have also caused valid unique data to be placed in the row. This data should copied out and preserved for study and possible reconciliation prior to deleting the data.

If there are many distinct sets of duplicate PK values in the table, it may be too time-consuming to remove them individually. In this case the following procedure can be used:

**1.**First, run the above GROUP BY query to determine how many sets of duplicate PK values exist, and the count of duplicates for each set.

**2.**Select the duplicate key values into a holding table. For example:

| SELECT col1, col2, col3=count(\*)  INTO holdkey  FROM t1  GROUP BY col1, col2  HAVING count(\*) > 1 |
| --- |

**3.**Select the duplicate rows into a holding table, eliminating duplicates in the process. For example:

| SELECT DISTINCT t1.\*  INTO holddups  FROM t1, holdkey  WHERE t1.col1 = holdkey.col1  AND t1.col2 = holdkey.col2 |
| --- |

**4.**At this point, the holddups table should have unique PKs, however, this will not be the case if t1 had duplicate PKs, yet unique rows (as in the SSN example above). Verify that each key in holddups is unique, and that you do not have duplicate keys, yet unique rows. If so, you must stop here and reconcile which of the rows you wish to keep for a given duplicate key value. For example, the query:

| SELECT col1, col2, count(\*)  FROM holddups  GROUP BY col1, col2 |
| --- |

should return a count of 1 for each row. If yes, proceed to step 5 below. If no, you have duplicate keys, yet unique rows, and need to decide which rows to save. This will usually entail either discarding a row, or creating a new unique key value for this row. Take one of these two steps for each such duplicate PK in the holddups table.

**5.**Delete the duplicate rows from the original table. For example:

| DELETE t1  FROM t1, holdkey  WHERE t1.col1 = holdkey.col1  AND t1.col2 = holdkey.col2 |
| --- |

**6.**Put the unique rows back in the original table. For example:

| INSERT t1 SELECT \* FROM holddups |
| --- |

**The information in this article applies to:**

* Microsoft SQL Server 6.0
* Microsoft SQL Server 6.5
* Microsoft SQL Server 7.0
* Microsoft SQL Server 2000 (all editions)

***How can I edit the system tables manually?***

It is not recommended to do, but it is possible. You should set the **allow updates** option to 1 by using the **sp\_configure** system stored procedure.

This is the example:

| sp\_configure 'allow updates', 1  GO  RECONFIGURE WITH OVERRIDE  GO |
| --- |

***How can I enforce to use particular index?***

You can use index hint (index=<index\_name>) after the table name. This is the example to use the aunmind index to get the last names of each author from the authors table:

| USE pubs  SELECT au\_lname FROM authors (index=aunmind) |
| --- |

***How can I export a query result set into file?***

There are several ways to do it.

1. You can use bcp utility.

Read about bcp utility in SQL Server Books online. This is the example to export data from the authors table in the pubs database into authors.txt file on the drive C:

| EXEC master..xp\_cmdshell  'bcp "select \* from pubs..authors" queryout c:\authors.txt -c -Usa -P' |
| --- |

2. You can use osql or isql utility.

This is the example to export data from the authors table in the pubs database into authors.txt file on the drive C:

| EXEC master..xp\_cmdshell  'osql -S ServerName -U sa -P -q "select \* from pubs..authors" -o c:\authors.txt' |
| --- |

3. You can use DTS Export Wizard.

4. You can use SQL-DMO.

(**sp\_OACreate, sp\_OAMethod, sp\_OAGetProperty, sp\_OASetProperty** and so on).

***How can I get only the specified number of rows from my query?***

You can use the **SET ROWCOUNT** statement or the TOP clause of the select statement (there is no TOP clause in SQL Server 6.5).

There are examples to return only the 10 top rows from the authors table in the pubs database:

| SET ROWCOUNT 10  GO  SELECT \* FROM pubs.dbo.authors  GO |
| --- |

or

*SELECT TOP 10 \* FROM pubs.dbo.authors*

***How can I remove foreign keys through SQL Server 7.0/2000 Enterprise Manager?***

To drop foreign keys, you can do the following:

1. Run SQL Server Enterprise Manager.

2. Expand a server group; then expand a server.

3. Expand Databases; then expand appropriate database.

4. Click Tables; then right-click the table and choose 'Design Table'.

5. Click the 'Table and Index Properties' button on the top menu.

6. On the 'Relationships' tab choose appropriate relationship and click 'Delete' button.

You can also drop foreign keys from the diagrams in the Enterprise Manager (make diagram and drop foreign keys).

***How can I return from within a T-SQL the list of all available OLE DB providers?***

You can use the **xp\_enum\_oledb\_providers** undocumented extended stored procedure to return the list of all available server's OLE DB providers. This extended stored procedure does not require any parameters.

This is the example to run:

*EXEC master..xp\_enum\_oledb\_providers*

***How can I return from within a T-SQL the list of all character sets and code pages?***

You can use the **xp\_enumcodepages** undocumented extended stored procedure to return the list of all available character sets and code pages. This extended stored procedure does not require any parameters.

This is the example to run:

*EXEC master..xp\_enumcodepages*

***How can I return from within a T-SQL the list of server's DSNs?***

You can use the **xp\_enumdsn** undocumented extended stored procedure to return the list of server's DSNs. This extended stored procedure does not require any parameters.

This is the example to run:

*EXEC master..xp\_enumdsn*

***How can I return from within a T-SQL the list of server's hard drives?***

You can use the **xp\_fixeddrives** undocumented extended stored procedure to return the list of all hard drives and the amount of free space in Mb for each hard drive. This extended stored procedure does not require any parameters.

This is the example to run:

*EXEC master..xp\_fixeddrives*

***How can I return from within a T-SQL the list of Windows NT/2000 groups?***

You can use the **xp\_enumgroups** undocumented extended stored procedure to return the list of Windows NT/2000 groups and their descriptions. This extended stored procedure does not require any parameters.

This is the example to run:

*EXEC master..xp\_enumgroups*

***How can I select only date/time part of a datetime value?***

--to select date part only

SELECT CONVERT(char(10),GetDate(),101)

--to select time part only

SELECT right(GetDate(),7)

***How can I send a message to user from the SQL Server?***

You can use the **xp\_cmdshell** extended stored procedure to run **net send** command. This is the example to send the 'Hello' message to JOHN:

*EXEC master..xp\_cmdshell "net send JOHN 'Hello'"*

To get **net send** message on the Windows 9x machines, you should run the WinPopup utility. You can place WinPopup in the Startup group under Program Files.

***How can I set database to a single user mode?***

To set database to a single user mode, you can use the Enterprise Manager or the **sp\_dboption** system stored procedure.

For example, to set pubs database to a single user mode, you can use the following statement:

*EXEC sp\_dboption 'pubs', 'single user', 'TRUE'*

Read SQL BOL to get more information about the **sp\_dboption** system stored procedure.

***How can I use a variable for the columns/tables in the select statements?***

SQL Server can contain both static sql statements and dynamic sql statements. In comparison with static sql statements, dynamic sql statements are not completely embedded in the source code; instead, portions are stored in program variables that can be modified at run time.

So, you can use dynamic sql statements (using EXECUTE command) to execute a character string which contain a variable for the columns/tables in the select statements. Read about the EXECUTE command in SQL Server Books Online.

This is the example to make select from the table, which name was passed into the stored procedure as variable:

| CREATE PROCEDURE GetRec  ( @tableName sysname )  AS  EXEC ('SELECT \* from ' + @tableName) |
| --- |

***How do I add 'n' working days to the given date and get the result date?***

To add 'n' working days to the given date and return the result date, you can use AddWorkingDays user-defined function. The AddWorkingDays udf works like DATEADD, but adds the working days.

**Syntax**

AddWorkingDays ( StartDate, WorkDays )

**Arguments**

**StartDate**

Is the datetime value (start date).

**WorkDays**

Is the integer value (number of working days).

**Return Types**

datetime

**The function's text:**

| CREATE FUNCTION dbo.ADDWorkingDays  ( @StartDate datetime,  @WorkDays int )  RETURNS datetime  AS  BEGIN  DECLARE @TotalDays int, @FirstPart int  DECLARE @EndDate datetime  DECLARE @LastNum int, @LastPart int  IF @WorkDays < 0  BEGIN  SELECT @FirstPart = CASE DATENAME(weekday, @StartDate)  WHEN 'Sunday' THEN 0  WHEN 'Monday' THEN 1  WHEN 'Tuesday' THEN 2  WHEN 'Wednesday' THEN 3  WHEN 'Thursday' THEN 4  WHEN 'Friday' THEN 5  WHEN 'Saturday' THEN 6  END  IF ABS(@WorkDays) < @FirstPart  SELECT @EndDate = DATEADD(dd, @WorkDays, @StartDate)  ELSE  BEGIN  SELECT @TotalDays = (ABS(@WorkDays) - @FirstPart) / 5  SELECT @LastPart = (ABS(@WorkDays) - @FirstPart) % 7  SELECT @LastNum = CASE  WHEN (@LastPart < 7) AND (@LastPart > 0) THEN @LastPart - 1  ELSE 0  END  SELECT @TotalDays = - 2 \* (@TotalDays + 1) + @WorkDays  SELECT @EndDate = DATEADD(dd, @TotalDays, @StartDate)  END  END  ELSE  BEGIN  SELECT @FirstPart = CASE DATENAME(weekday, @StartDate)  WHEN 'Sunday' THEN 6  WHEN 'Monday' THEN 5  WHEN 'Tuesday' THEN 4  WHEN 'Wednesday' THEN 3  WHEN 'Thursday' THEN 2  WHEN 'Friday' THEN 1  WHEN 'Saturday' THEN 0  END  IF @WorkDays < @FirstPart  SELECT @EndDate = DATEADD(dd, @WorkDays, @StartDate)  ELSE  BEGIN  SELECT @TotalDays = (@WorkDays - @FirstPart) / 5  SELECT @LastPart = (@WorkDays - @FirstPart) % 7  SELECT @LastNum = CASE  WHEN (@LastPart < 7) AND (@LastPart > 0) THEN @LastPart - 1  ELSE 0  END  SELECT @TotalDays = 2 \* (@TotalDays + 1) + @WorkDays  SELECT @EndDate = DATEADD(dd, @TotalDays, @StartDate)  END  END  RETURN ( @EndDate )  END  GO |
| --- |

**Example**

Adds 9 working days to '11/13/2002' and returns the result date:

| SELECT dbo.AddWorkingDays ('11/13/2002', 9)  GO |
| --- |

Here is the result set:

| ------------------------------------------------------  2002-11-26 00:00:00.000  (1 row(s) affected) |
| --- |

**Note.** The user-defined functions are only available with SQL Server 2000, SQL Server 7.0 does not support user-defined functions.

***How do I delete a particular value for a key in the registry via Transact-SQL?***

You can use the **xp\_regdeletevalue** undocumented extended stored procedure to delete a particular value for a key in the registry via Transact-SQL.

This is the syntax of the **xp\_regdeletevalue**:

| EXECUTE xp\_regdeletevalue [@rootkey=]'rootkey',  [@key=]'key',  [@value\_name=]'value\_name' |
| --- |

For example, to delete the value 'TestValue' for the key 'SOFTWARE\Test' from 'HKEY\_LOCAL\_MACHINE', run:

| EXEC master..xp\_regdeletevalue  @rootkey='HKEY\_LOCAL\_MACHINE',  @key='SOFTWARE\Test',  @value\_name='TestValue' |
| --- |

***How do I delete an entire key from the registry via Transact-SQL?***

You can use the **xp\_regdeletekey** undocumented extended stored procedure to delete an entire key from the registry via Transact-SQL.

This is the syntax of the **xp\_regdeletekey**:

| EXECUTE xp\_regdeletekey [@rootkey=]'rootkey',  [@key=]'key' |
| --- |

For example, to delete the key 'SOFTWARE\Test' from 'HKEY\_LOCAL\_MACHINE', run:

| EXEC master..xp\_regdeletekey  @rootkey='HKEY\_LOCAL\_MACHINE',  @key='SOFTWARE\Test' |
| --- |

***How do I execute an operating-system command via Transact-SQL?***

You can use the **xp\_cmdshell** extended stored procedure to execute a given command string as an operating-system command shell and return any output as rows of text.

This is the syntax of the **xp\_cmdshell**:

xp\_cmdshell {'command\_string'} [, no\_output]

where

'command\_string' is varchar(255) or nvarchar(4000), with no default.

no\_output - when used no output will be returned.

When executing **xp\_cmdshell** with the Windows 9x operating systems, the return code will always be 0.

Executing this **xp\_cmdshell** statement sends the 'Hello' message to JOHN:

EXEC master..**xp\_cmdshell** "net send JOHN 'Hello'"

***How do I get the date the object was created?***

To get the date the object was created, you can query the **sysobjects** system table to see the crdate column. This is the example to get the date the authors table was created:

*SELECT crdate FROM sysobjects WHERE name = 'authors'*

***How do I get working days between two dates?***

To return the number of working days between two dates (not including these dates), you can use GetWorkingDays user-defined function.

**Syntax**

GetWorkingDays (StartDate, EndDate)

**Arguments**

**StartDate**

Is the datetime value (start date).

**WorkDays**

Is the datetime value (end date).

**Return Types**

int

**The function's text:**

| CREATE FUNCTION dbo.GetWorkingDays  ( @StartDate datetime,  @EndDate datetime )  RETURNS INT  AS  BEGIN  DECLARE @WorkDays int, @FirstPart int  DECLARE @FirstNum int, @TotalDays int  DECLARE @LastNum int, @LastPart int  IF (DATEDIFF(day, @StartDate, @EndDate) < 2)  BEGIN  RETURN ( 0 )  END  SELECT  @TotalDays = DATEDIFF(day, @StartDate, @EndDate) - 1,  @FirstPart = CASE DATENAME(weekday, @StartDate)  WHEN 'Sunday' THEN 6  WHEN 'Monday' THEN 5  WHEN 'Tuesday' THEN 4  WHEN 'Wednesday' THEN 3  WHEN 'Thursday' THEN 2  WHEN 'Friday' THEN 1  WHEN 'Saturday' THEN 0  END,  @FirstNum = CASE DATENAME(weekday, @StartDate)  WHEN 'Sunday' THEN 5  WHEN 'Monday' THEN 4  WHEN 'Tuesday' THEN 3  WHEN 'Wednesday' THEN 2  WHEN 'Thursday' THEN 1  WHEN 'Friday' THEN 0  WHEN 'Saturday' THEN 0  END  IF (@TotalDays < @FirstPart)  BEGIN  SELECT @WorkDays = @TotalDays  END  ELSE  BEGIN  SELECT @WorkDays = (@TotalDays - @FirstPart) / 7  SELECT @LastPart = (@TotalDays - @FirstPart) % 7  SELECT @LastNum = CASE  WHEN (@LastPart < 7) AND (@LastPart > 0) THEN @LastPart - 1  ELSE 0  END  SELECT @WorkDays = @WorkDays \* 5 + @FirstNum + @LastNum  END  RETURN ( @WorkDays )  END  GO |
| --- |

**Example**

Returns the number of working days between '11/13/2000' and '12/27/2000':

| SELECT dbo.GetWorkingDays ('11/13/2000', '12/27/2000')  GO |
| --- |

Here is the result set:

| -----------  31  (1 row(s) affected) |
| --- |

**Note.** The user-defined functions are only available with SQL Server 2000, SQL Server 7.0 does not support user-defined functions.

***How do I move tables from one filegroup to another?***

To move tables from one filegroup to another, you can do the following:

1. Run SQL Server Enterprise Manager.

2. Expand a server group; then expand a server.

3. Expand Databases; then expand appropriate database.

4. Click Tables; then right-click the table and choose 'Design Table'.

5. Click 'Table and Index Properties' on the top menu and change the table filegroup.

***How do I perform the same actions for all databases?***

If you need to perform the same actions for all databases, you can create cursor or you can use the **sp\_MSforeachdb** undocumented stored procedure to accomplish the same goal with less work.

For example, you can use the **sp\_MSforeachdb** stored procedure to run the **DBCC CHECKDB** statement for all the databases on your server:

*EXEC sp\_MSforeachdb @command1="print '?' DBCC CHECKDB ('?')"*

***How do I perform the same actions for all tables in a database?***

If you need to perform the same actions for all tables in a database, you can create cursor or you can use the **sp\_MSforeachtable** undocumented stored procedure to accomplish the same goal with less work.

For example, you can use the **sp\_MSforeachtable** stored procedure to rebuild all indexes in a database:

*EXEC sp\_MSforeachtable @command1="print '?' DBCC DBREINDEX ('?')"*

***How do I read from the registry via Transact-SQL?***

You can use the **xp\_regread** undocumented extended stored procedure to read from the registry via Transact-SQL.

This is the syntax of the **xp\_regread**:

| EXECUTE xp\_regread [@rootkey=]'rootkey',  [@key=]'key'  [, [@value\_name=]'value\_name']  [, [@value=]@value OUTPUT] |
| --- |

For example, to read into the variable @test from the value 'TestValue' from the key 'SOFTWARE\Test' from the 'HKEY\_LOCAL\_MACHINE', run:

| DECLARE @test varchar(20)  EXEC master..xp\_regread @rootkey='HKEY\_LOCAL\_MACHINE',  @key='SOFTWARE\Test',  @value\_name='TestValue',  @value=@test OUTPUT  SELECT @test |
| --- |

***How do I return the first day of the month for the given date?***

To return the first day of the month for the given date, you can use the FirstMonthDay user-defined function.

**Syntax**

FirstMonthDay (date)

**Arguments**

**date**

Is the datetime value.

**Return Types**

datetime

**The function's text:**

| CREATE FUNCTION dbo.FirstMonthDay  ( @Date datetime )  RETURNS datetime  AS  BEGIN  RETURN (CAST(STR(MONTH(@Date))+'/'+STR(01)+'/'+STR(YEAR(@Date)) AS DateTime))  END  GO |
| --- |

**Example**

Returns the first day for the '06/15/99' date:

| SELECT dbo.FirstMonthDay('06/15/99')  GO |
| --- |

Here is the result set:

| ------------------------------------------------------  1999-06-01 00:00:00.000  (1 row(s) affected) |
| --- |

**Note.** The user-defined functions are only available with SQL Server 2000, SQL Server 7.0 does not support user-defined functions.

***How do I reset the identity column?***

You can use the DBCC CHECKIDENT statement, if you want to reset or reseed the identity column. For example, if you need to force the current identity value in the jobs table to a value of 100, you can use the following:

| USE pubs  GO  DBCC CHECKIDENT (jobs, RESEED, 100)  GO |
| --- |

***How do I return the last day of the month for the given date?***

To return the last day of the month for the given date, you can use the LastMonthDay user-defined function.

**Syntax**

LastMonthDay (date)

**Arguments**

**date**

Is the datetime value.

**Return Types**

datetime

**The function's text:**

| CREATE FUNCTION dbo.LastMonthDay  ( @Date datetime )  RETURNS datetime  AS  BEGIN  RETURN (CASE WHEN MONTH(@Date)= 12  THEN DATEADD(day, -1, CAST('01/01/' + STR(YEAR(@Date)+1) AS DateTime))  ELSE DATEADD(day, -1, CAST(STR(MONTH(@Date)+1) + '/01/' + STR(YEAR(@Date)) AS DateTime))  END)  END  GO |
| --- |

**Example**

Returns the last day for the '06/15/99' date:

| SELECT dbo.LastMonthDay('06/15/99')  GO |
| --- |

Here is the result set:

| ------------------------------------------------------  1999-06-30 00:00:00.000  (1 row(s) affected) |
| --- |

**Note.** The user-defined functions are only available with SQL Server 2000, SQL Server 7.0 does not support user-defined functions

***How do I run a DTS package from the Query Analyzer or stored procedure?***

There are two ways to run a DTS package from the Query Analyzer or stored procedure:

1. By using OLE Automation objects

(run **sp\_OACreate**, **sp\_OAGetProperty**, **sp\_OAMethod** and so on).

2. By using **xp\_cmdshell** stored procedure to run dtsrun.exe file.

This is the example to execute NewPack.dts package from the disk C:

| DECLARE @object int  DECLARE @hr int  --create a package object  EXEC @hr = sp\_OACreate 'DTS.Package', @object OUTPUT  if @hr <> 0  BEGIN  print 'error create DTS.Package'  RETURN  END  EXEC @hr = sp\_OAMethod @object, 'LoadFromStorageFile',  NULL, 'C:\NewPack.dts', ''  IF @hr <> 0  BEGIN  print 'error LoadFromStorageFile'  RETURN  END  EXEC @hr = sp\_OAMethod @object, 'Execute'  IF @hr <> 0  BEGIN  print 'Execute failed'  RETURN  END |
| --- |

***How do I run a T-SQL script through SQL-DMO?***

You can use **ExecuteImmediate** or **CommandShellImmediate** methods of SQL-DMO. This is the example to execute T-SQL script from the c:\trig.sql file:

| DECLARE @object int  DECLARE @hr int  DECLARE @property varchar(255)  DECLARE @return varchar(255)  EXEC @hr = sp\_OACreate 'SQLDMO.SQLServer', @object OUT  IF @hr <> 0  BEGIN  print 'error create SQLDMO.SQLServer'  RETURN  END  EXEC @hr = sp\_OAMethod  @object, 'Connect', NULL, 'ServerName', 'Login', 'Password'  IF @hr <> 0  BEGIN  print 'error with Connect'  RETURN  END  EXEC @hr = sp\_OAMethod @object, 'VerifyConnection', @return OUT  IF @hr <> 0  BEGIN  PRINT 'error with VerifyConnection'  RETURN  END  select @property = 'EXEC master..xp\_cmdshell "osql -S ServerName ' +  '-U LoginName -P password -i c:\trig.sql"'  EXEC @hr = sp\_OAMethod @object, 'ExecuteImmediate', Null , @property  IF @hr <> 0  BEGIN  print 'error with ExecuteImmediate'  RETURN  END  -- ahother way to run a T-SQL script  select @property = 'osql -S ServerName -U LoginName -P password -i c:\trig.sql'  EXEC @hr = sp\_OAMethod @object, 'CommandShellImmediate', Null , @property  IF @hr <> 0  BEGIN  print 'error with CommandShellImmediate'  RETURN  END |
| --- |

***How do I run SQL script located on the disk file, using Transact-SQL?***

You can use osql or isql utility to run SQL script located on the disk file. This is the example to run SQL script located in the C:\SQLScript.sql file:

EXEC master..xp\_cmdshell 'osql -S Servername -U sa -P -ic:\SQLScript.sql'

See SQL Server Books Online for more information about osql or isql utility and about xp\_cmdshell extended stored procedure.

***How do I write to the registry via Transact-SQL?***

You can use the **xp\_regwrite** undocumented extended stored procedure to write to the registry via Transact-SQL.

This is the syntax of the **xp\_regwrite**:

| EXECUTE xp\_regwrite [@rootkey=]'rootkey',  [@key=]'key',  [@value\_name=]'value\_name',  [@type=]'type',  [@value=]'value' |
| --- |

For example, to write the variable 'Test' to the 'TestValue' value, key 'SOFTWARE\Test', 'HKEY\_LOCAL\_MACHINE', run:

| EXEC master..xp\_regwrite  @rootkey='HKEY\_LOCAL\_MACHINE',  @key='SOFTWARE\Test',  @value\_name='TestValue',  @type='REG\_SZ',  @value='Test' |
| --- |

***How to install SQL scripts and import data to SQL Server from the command line?***

You can create \*.bat file which include something like this:

osql -S ServerName -U LoginName -P password -i c:\script.sql

where c:\script.sql - the script to create metadata and load data. See instpubs.sql file in the Install subdirectory of your SQL Server installation as example of creating such script.

***Is it possible to work with COM objects from within T-SQL?***

You can use OLE Automation Stored Procedures (extended stored procedures with sp\_OA prefix) to create a COM object in T-SQL and use the object's methods and properties.

***Why SELECT INTO statement does not work?***

First of all, chect the syntax of your SELECT INTO statement. If the syntax is correct, check that the **select into/bulkcopy** database option is turned on.

To check that the **select into/bulkcopy** database option is turned on, you can use the Enterprise Manager or the **sp\_dboption** system stored procedure.

For example, to check that the **select into/bulkcopy** database option is turned on for the pubs database, you can use the following statement:

*EXEC sp\_dboption pubs, 'select into/bulkcopy'*

To turn on the **select into/bulkcopy** database option, you can use the following statement:

*EXEC sp\_dboption 'pubs', 'select into/bulkcopy', 'TRUE'*

Read SQL BOL to get more information about the **sp\_dboption** system stored procedure.

General

***Are there arrays in SQL Server? Or what is the array analog?***

There are no arrays in SQL Server. You can use the character string to store the comma-separated values and then use PATINDEX function to extract the values, or you can store the values in separate table.

***Can I set tempdb database to be in RAM in SQL 7.0/2000?***

No, you cannot set **tempdb** database to be in RAM in SQL Server 7.0/2000. This option is no longer supported.

***Could you give me the comparison of MSDE with SQL Server?***

### Introduction

Often people in newsgroups ask about some comparison of MSDE and SQL Server 7.0. In this article, I want to describe the differences between the Microsoft Data Engine and SQL Server 7.0 in cost, licensing, hardware and software requirements and restrictions.

### The comparison

MSDE stands for the Microsoft Data Engine. Microsoft Data Engine is a client/server data engine that is compatible with SQL Server 7.0, but has some restrictions and is intended for a single user or for a small workgroup with up to five computers. You can use MSDE as a back-end database in a workgroup with more than five computers, but it can be very slowly.

### Hardware and Software Requirements

To install SQL Server 7.0, you should have the following hardware:

* Alpha AXP, Intel or compatible platform.
* Pentium 166 MHz or higher (Pentium Pro, Pentium II and compatible).
* 32MB RAM (minimum), 64MB RAM and more (recommended).
* 180MB (full installation), 170MB (typical), 65MB (minimum), 90MB (management tools only) plus 50MB for OLAP Services and 12MB for the English Query.

SQL Server 7.0 can work under Windows 95, Windows 95 OSR2, Windows 98, Windows NT Server/Workstation 4.0 with Service Pack 4 and Windows 2000 (all editions).

SQL Server 7.0 also requires Internet Explorer 4.01 with Service Pack 1 or later, or Internet Explorer 5.0 or later.

To install MSDE 1.0, you should have the following hardware:

* Alpha AXP, Intel or compatible platform.
* Pentium 166 MHz or higher (Pentium Pro, Pentium II and compatible).
* 32MB RAM.
* 71-79MB hard disk space.

The MSDE 1.0 works under Windows 95, Windows 95 OSR2, Windows 98, Windows NT Server/Workstation 4.0 with Service Pack 4 and Windows 2000 (all editions).

Unlike SQL Server 7.0, Microsoft Data Engine does not require Internet Explorer.

### MSDE Restrictions

Because Microsoft Data Engine is intended for a single user or for a small workgroup with up to five computers, it has some restrictions in comparison with SQL Server 7.0.

There are:

* Supports up to 2Gb RAM.
* 2GB database size limit.
* Five concurrent users or less is recommended.
* Supports up to 2 CPU on Windows NT or Windows 2000 box.
* No publishing for transaction replication.
* No Database Server Failover Support.
* No Full-text search.
* No GUI interface (there are no SQL Server Enterprise Manager, SQL Server Profiler, Query Analyzer, Database Upgrade Wizard, Index Tuning Wizard, Import and Export Wizards, and so on).
* No OLAP.
* No English Query.
* No SQL Books Online.

### Pricing and Licensing

Microsoft Data Engine does not sold as a separate product. MSDE is available to any licensed customer of Office 2000 Professional, Office 2000 Premium edition, or licensed customer of any Visual Studio 6.0 edition or any Professional or Enterprise edition of any Visual Studio 6.0 language tool (such as Visual Basic 6.0, Visual C++ 6.0, Visual InterDev 6.0, Visual J++ 6.0, or Visual FoxPro 6.0).

Microsoft SQL Server 7.0 does not sold now. Now, Microsoft promotes SQL Server 2000 as back-end database server, but if you have existing solutions on SQL Server 7.0, you may still acquire additional licenses by taking advantage of the SQL Server 2000 "downgrade rights". It enables to buy SQL Server 2000 licenses, but install and use the previous version for a reasonable period of time.

Unlike SQL Server 7.0, MSDE solutions can be distributed royalty-free, i.e. you don't need to buy Client Access License, if your application uses Microsoft Data Engine as a back-end.

### Upgrading from MSDE to SQL Server

MSDE data engines are the same as SQL Server 7.0 data engines. So, all database objects created in MSDE (such as tables, views, stored procedures and so on) will operate without any modifications in a SQL Server 7.0 database.

There are two ways to upgrade an MSDE solution to SQL Server 7.

* You can upgrade an MSDE to SQL Server 7.0 by installing SQL Server. Just run SQL Server installation, and the installation program will recognize an MSDE installation and will accompany data files and transaction logs.
* You can upgrade an MSDE to SQL Server 7.0 by using SQL Server 7.0 Enterprise Manager from a SQL Server machine to connect to the MSDE. Once connected, MSDE databases can be transferred to SQL Server (by using sp\_detach\_db and sp\_attach\_db stored procedures, by using backup/restore or by transfering data with DTS). This method is used when upgrading to SQL Server using a different machine.

When you upgrade MSDE to SQL Server 7.0, you can receive the following error:

| You cannot install a version which is older (7.00.623) than the  version on your machine (7.00.677). Uninstall the older version. |
| --- |

This error occurs when you tried to upgrade MSDE that came with the Office 2000 developer edition. You should edit some registry entries, in this case.

You should amend

*HKEY\_LOCAL\_MACHINE/Software/Microsoft/MSSQLServer/MSSQLServer/CurrentVersion*

to have a value of 7.00.623.

### Creating and designing MSDE databases

Microsoft Data Engine does not have its own GUI interface for database design. MSDE is shipped with only osql utility, but it's difficult to create and manage databases from the command prompt.

So, you can create and manage MSDE databases via the Access 2000 User Interface or via the Visual Studio development environment.

If you are a licensed customer of any Visual Studio 6.0 edition or any Professional or Enterprise edition of any Visual Studio 6.0 language tool (such as Visual Basic 6.0, Visual C++ 6.0, Visual InterDev 6.0, Visual J++ 6.0, or Visual FoxPro 6.0), you can have SQL Server 7.0 Developer Edition to manage MSDE databases.

The SQL Server 7.0 Developer Edition is shipped with MSDE for Visual Studio 6.0, and is provided on the CDs that can be ordered from the [**http://msdn.microsoft.com/vstudio/msde/**](http://msdn.microsoft.com/vstudio/msde/)

With SQL Server 7.0 Developer Edition, you can manage MSDE databases by using Enterprise Manager.

You can create you own administration tool for MSDE, because Microsoft provides SQL Distributed Management Objects (SQL-DMO) and SQL Namespace (SQL-NS) COM libraries with Microsoft Data Engine.

***Could you give me the comparison of SQL Server with Access?***

|  |
| --- |

***Could you give me the comparison of SQL 7.0 with SQL 2000?***

|  |
| --- |

***How can I add SQL Server licenses?***

|  |
| --- |

***How can I get the column's description from my stored procedure?***

|  |
| --- |

***How can I perform a nonlogged bulk copy?***

|  |
| --- |

***How can I view the SQL Server transaction log?***

|  |
| --- |

***How do I migrate an Access database to SQL 7.0?***

|  |
| --- |

***How do I migrate an Oracle database to SQL 7.0?***

|  |
| --- |

***How is SQL Server licensed?***

|  |
| --- |

***How many SQL 7.0 editions exist and what is the difference between them?***

|  |
| --- |

***How many SQL 2000 editions exist and what is the difference between them?***

|  |
| --- |

***How to detect if the SQL Server is installed on the machine?***

|  |
| --- |

***What are good SQL Server web resources?***

|  |
| --- |

***What is the difference between char and varchar data types?***

|  |
| --- |

***What is the difference between clustered and nonclustered indexes?***

|  |
| --- |

***What is the difference between Oracle and SQL Server?***

|  |
| --- |

***Where can I download free case tools?***

|  |
| --- |

***Where can I download SQL Server 2000 Books Online?***

|  |
| --- |

***Where can I find MSSQL test exams?***

|  |
| --- |

***Where can I get the benchmark tests for SQL Server and other RDBMS?***

|  |
| --- |

***Where can I get the evaluation copy of SQL Server?***

|  |
| --- |

***Where can I see the price-list for SQL Server?***

|  |
| --- |

Installation/Upgrade

***Can you give me the links to upgrade from SQL 7.0 to SQL 2000?***

***How can I check what SQL service pack am I running?***

***How do I install the SQL Server client?***

***How do I test an installation of SQL Server?***

***Should I apply the SQL service packs in order or can I just apply the latest one?***

***What is the current service pack for SQL Server 7.0?***

***What is the current service pack for SQL Server 2000?***

***Where can I download the SQL Server service packs?***

***Where can I find the list of bugs fixed by SQL 7.0 service packs?***

***Where can I find the list of bugs fixed by SQL 2000 service packs?***

OLAP

***Can I run OLAP Services 7.0 and Analysis Services 2000 on the same computer?***

***How can I connect to OLAP cubes from the client's computer?***

***What hardware and software requirements should I have for installing Analysis Services?***

***What SQL Server editions support Custom Rollups?***

***What SQL Server editions support Data Mining?***

***What SQL Server editions support Linked OLAP Cubes?***

***What SQL Server editions support user-defined OLAP partitions?***

Replication

***Can I replicate data to Access or Oracle subscribers?***

***Can I setup replication on a single computer?***

***How can I replicate objects permissions?***

***How can I start replication agents via Transact-SQL?***

***How can I validate replicated data for transaction or merge replication?***

***How do we handle duplicate primary keys for transaction/merge replication?***

***Some of my text/image values were not replicated. What is the problem?***

***What SQL Server editions support replication?***

***Where can I get tutorials about setting up snapshot/transaction/merge replication?***

***Where does snapshot replication store replicated data?***

Transfer/move

***How can I transfer SQL Server databases to a new location?***

***How can I transfer the particular database diagram from one SQL Server to another?***

***How can I transfer the particular DTS package from one SQL Server to another?***

***How do I transfer data from/to another heterogeneous sources to/from SQL Server?***

Troubleshooting

***How can I fix lost users after attaching database with sp\_attach\_db?***

***How can I resolve the error 259?***

***How can I resolve the error 605?***

***How can I resolve the error 1069?***

***How can I resolve the error 80004005?***

***How do I resolve deadlocks?***

***How to troubleshoot SQL Server installation problems?***

***I am getting an error 1205 - what can I do?***

***I can run DTS package myself, but cannot schedule it.***

***I get the full-text search timeout error. How can I fix this?***

***I got an error that RPC is not enabled. How can I enable RPC?***

***My database has been marked suspect. How can I fix this?***

***My DTS package is stored inside SQL Server. Now I cannot open it.***

***The upgrade MSDE to SQL Server fails with the error about 'old version'.***

***Why cannot I see the SQL performance counters in PerfMon?***

***Why cannot I truncate the transaction log?***

***Why DTS Import Wizard fails when importing table with a timestamp column?***

SQL Server 6.5

***Can we mirror data from one SQL 6.5 server to another?***

***Could you give me the comparison of SQL 6.5 with SQL 7.0?***

***I am getting an error 1105 - what can I do?***

***I am having problems installing the service pack 5a (cfgchar problem).***

***I have duplicate identity value error. What can I do?***

***I reinstall SQL 6.5. How can I restore the user databases?***

***Is SQL 6.5 year 2000 compliant?***

***How can I delete/edit columns in SQL 6.5?***

***How can I exceed the 16 tables restriction for one query?***

***How can I install SQL 6.5 on my home WinNT WorkStation?***

***How can I move SQL 6.5 databases to a new location?***

***How do I remove the tempdb database from the master device?***

***My database has been marked loading. How can I fix this?***

***What are the useful undocumented DBCC commands for SQL 6.5?***

***Why do my device have negative size?***

**How do I administer SQL Server remotely?**

**Answer:**  
  
You can install Enterprise Manager utility and then administer remote server just as you administer a local server. If you need to administer SQL Server over the internet, you should specify the TCP/IP address instead of the remote SQL Server's name under the 'New SQL Server Registation' window.

**How do I go about using/changing my settings to use the SANs protocol?**

**Answer:**  
  
SAN - is the System Area Network protocols, which supported under SQL Server 2000 Enterprise Edition only and built using the Virtual Interface Architecture (VIA). SANs are intended to support the high communications bandwidth between servers.  
To use SANs protocol, you can specify Giganet VIA SAN Net-Libraries on the client's computer:  
  
1. Run SQL Server Client Network Utility.  
2. On the General tab enable VIA protocol.  
3. Choose VIA and click Properties button.  
4. Specify Giganet as Vendor and close Client Network Utility.

**How do I mark the stored procedure to automatic execution?**

**Answer:**  
  
You can use the **sp\_procoption** system stored procedure to mark the stored procedure to automatic execution when the SQL Server will start.  
  
**Note**. Only objects in the **master** database owned by dbo can have the startup setting changed and this option is restricted to objects that have no parameters.  
  
This is the example to set the startup option for the indRebuild stored procedure in the **master** database:

| USE master  EXEC sp\_procoption 'indRebuild', 'startup', 'true' |
| --- |

See SQL Server Books Online for more details about the **sp\_procoption** system stored procedure.

**How do I report information about current users and processes?**

**Answer:**  
  
To report information about current users and processes, you can use the **sp\_who** and **sp\_who2** system stored procedures. The **sp\_who** system stored procedure described in SQL Server Books Online, the **sp\_who2** is unducumented stored procedure. **sp\_who2** provides more detailed information about current users and processes than **sp\_who**. The **sp\_who2** syntax is equal to **sp\_who**.  
  
**Syntax**  
  
*sp\_who2 [[@login\_name =] 'login']***Arguments**  
*[@login\_name =] 'login'*Is a user login name on SQL Server. login is sysname, with a default of NULL. If no name is specified, the procedure reports all active users of SQL Server.  
  
See also these links:  
[**SQL Server 7.0 Useful undocumented stored procedures**](http://www.mssqlcity.com/Articles/Undoc/SQL70UndocSP.htm)  
[**SQL Server 2000 Useful undocumented stored procedures**](http://www.mssqlcity.com/Articles/Undoc/SQL2000UndocSP.htm)

**How do I report information about the active locks?**

**Answer:**  
  
To report information about the active locks, you can use the **sp\_lock** system stored procedure.  
Read about the **sp\_lock** stored procedure in SQL Server Books Online.  
  
To get more detailed locking view, use the **sp\_lock2** stored procedure from this article:  
[**Detailed locking view: sp\_lock2**](http://www.mssqlcity.com/Articles/Adm/LockView.htm)

**How can I add a user-defined error message?**

**Answer:**  
  
You can use the **sp\_addmessage** system stored procedure to add a user-defined error message.  
  
You can specify the id of the message (value greater than 50000), severity level from 0 through 25 (to add a message with a severity level from 19 through 25, you must be a member of the sysadmin fixe server role) and the message text (up to 255 characters).  
  
By the way, only members of the sysadmin and serveradmin fixed server roles can execute the **sp\_addmessage** system stored procedure.  
  
This is the example:

| USE master  EXEC sp\_addmessage @msgnum = 50001,  @severity = 18,  @msgtext = 'Custom error message' |
| --- |

See SQL Server Books Online for more details about the **sp\_addmessage** system stored procedure.

**How can I change the database owner?**

**Answer:**  
  
You can use the **sp\_changedbowner** system stored procedure to change the database owner.  
Read about the **sp\_changedbowner** stored procedure in SQL Server Books Online.  
This is the example to make the user John the owner of the pubs database:

| USE pubs  GO  EXEC sp\_changedbowner 'John'  GO |
| --- |

See also this link:  
[**How can I change the object owner?**](http://www.mssqlcity.com/FAQ/Admin/ObjOwner.htm)

**How can I change the sort order and character set?**

**Answer:**  
  
You cannot make it without rebuilding your databases.  
This is from SQL Server Books Online:

| It is critical that you select the correct sort order when you install SQL Server.  If you need to change sort orders after installation, you must rebuild your  databases and reload your data. |
| --- |

and

| It is critical that you select the correct character set when you install SQL Server.  To change character sets after installing SQL Server, you must rebuild the databases  and reload the data. |
| --- |

**How can I change the sort order and character set?**

**Answer:**  
  
You cannot make it without rebuilding your databases.  
This is from SQL Server Books Online:

| It is critical that you select the correct sort order when you install SQL Server.  If you need to change sort orders after installation, you must rebuild your  databases and reload your data. |
| --- |

and

| It is critical that you select the correct character set when you install SQL Server.  To change character sets after installing SQL Server, you must rebuild the databases  and reload the data. |
| --- |

**How can I compare the data in two tables with identical structure?**

**Answer:**  
  
1. Compare the row counts in these tables by using the COUNT(\*) function:  
  
*SELECT COUNT(\*) FROM tb1  
SELECT COUNT(\*) FROM tb2*2. If the row counts in the tables is identical, you can download the data from these tables into files on the disk by using the bcp utility and compare the size of these files.  
  
To get more information about what rows are differ, you can compare these files by using the comp.exe utility (C:\WINNT\system32 - path by default for Windows NT).  
  
3. You can use red-gate SQL compare to do this. It can compare tables, constraints, rules, stored procedures and creates a script for applying the changes from one database to another.  
You can download the 14 days trial version at here:  
[**http://www.red-gate.com**](http://www.red-gate.com)  
  
See also this link:  
[**How can I compare two databases?**](http://www.mssqlcity.com/FAQ/Admin/CompareDB.htm)

**How can I compare two databases?**

**Answer:**  
  
You can use red-gate SQL compare to do this. It can compare tables, constraints, rules, stored procedures and creates a script for applying the changes from one database to another.  
You can download the 14 days trial version at here:  
[**http://www.red-gate.com**](http://www.red-gate.com)  
  
You can use the following scripts from the swynk.com:  
  
[**http://www.swynk.com/downloads/objcompare.zip**](http://www.swynk.com/downloads/objcompare.zip)

| This program is similar to Sqlcomp from Backoffice Resource Kit. With this program your can compare Stored Procedures and Views from different servers (example: Production and Developer) and versions (6.5 and 7.0). |
| --- |

[**http://www.swynk.com/downloads/sp\_db\_comp.sql**](http://www.swynk.com/downloads/sp_db_comp.sql)

| This script will compare all tables in two databases including column names, datatypes and NULL options. It also lists tables that exist only in one of databases. |
| --- |

[**http://www.swynk.com/downloads/sp\_ABCompareDb.sql**](http://www.swynk.com/downloads/sp_ABCompareDb.sql)

| Compare a source and target database. List all procedures, tables, views, and triggers on the source that either don't exist on the target, or are later on the source than the target machine. This is useful when a series of changed objects need to be installed from a development server to a production server. |
| --- |

See also this link:  
[**How can I compare the data in two tables with identical structure?**](http://www.mssqlcity.com/FAQ/Admin/CompareTb.htm) http://www.mssqlcity.com/FAQ/Admin/CompareTb.htm

**How can I get detailed information about locking?**

**Answer:**  
  
You can use the **sp\_lock** system stored procedure.  
Read about the **sp\_lock** stored procedure in SQL Server Books Online.  
  
To get more detailed locking view, use the **sp\_lock2** stored procedure from this article:  
[**Detailed locking view: sp\_lock2**](http://www.mssqlcity.com/Articles/Adm/LockView.htm): http://www.mssqlcity.com/Articles/Adm/LockView.htm

**How can I get detailed information about locking?**

**Answer:**  
  
You can use the **sp\_lock** system stored procedure.  
Read about the **sp\_lock** stored procedure in SQL Server Books Online.  
  
To get more detailed locking view, use the **sp\_lock2** stored procedure from this article:  
[**Detailed locking view: sp\_lock2**](http://www.mssqlcity.com/Articles/Adm/LockView.htm)

**How can I grant to user the RESTORE DATABASE permission?**

**Answer:**  
  
The RESTORE DATABASE permission cannot be granted or revoked. If you need to grant to the user the restore permission, add the user to the dbcreator fixed server role. You can use the **sp\_addsrvrolemember** system stored procedure to add the user to a fixed server role.  
  
This is the example of using the **sp\_addsrvrolemember** stored procedure to add **John** to the dbcreator fixed server role:  
  
*EXEC sp\_addsrvrolemember 'John', 'dbcreator'*  
  
See SQL Server Books Online for more details about the **sp\_addsrvrolemember** system stored procedure.

**How can I rebuild the SQL Server repository?**

**Answer:**  
  
This procedure is not easy.  
See this link from Microsoft:  
[**How to Rebuild the Repository in SQL Server 7.0**](http://support.microsoft.com/support/kb/articles/Q238/8/22.asp)

http://support.microsoft.com/support/kb/articles/Q238/8/22.asp

**How can I restore the user databases after reinstall SQL 7.0/2000?**

**Answer:**  
  
You can attach the user databases to a SQL Server or you can restore databases from the backup.  
Read about the **sp\_attach\_db** stored procedure in SQL Server Books Online Title: **sp\_attach\_db (T-SQL)**.  
  
See also this link:  
[**How can I transfer SQL Server databases to a new location?**](http://www.mssqlcity.com/FAQ/TranMove/AttachDb.htm)

**How can I show/hide system databases and system objects?**

**Answer:**  
  
Try this:  
  
1. Run Enterprise Manager.  
2. Right-click the server icon and select 'Edit SQL Server Registration properties'.  
3. To show system databases and system objects check 'Show system databases and system objects'.  
To hide system databases and system objects uncheck 'Show system databases and system objects'.

**How can I start MSSQLServer service at 6:30am and stop it at 6:30pm**

**automatically?**

**Answer:**

You can use AT command for this purpose.

AT - is a Windows NT file (c:\winnt\system32\at.exe - path by default).

It's full description of the AT command:

| The AT command schedules commands and programs to run on a computer at  a specified time and date. The Schedule service must be running to use  the AT command.    AT [\\computername] [ [id] [/DELETE] | /DELETE [/YES]]  AT [\\computername] time [/INTERACTIVE]  [ /EVERY:date[,...] | /NEXT:date[,...]] "command"  \\computername Specifies a remote computer. Commands are scheduled on the  local computer if this parameter is omitted.  id Is an identification number assigned to a scheduled  command.  /delete Cancels a scheduled command. If id is omitted, all the  scheduled commands on the computer are canceled.  /yes Used with cancel all jobs command when no further  confirmation is desired.  time Specifies the time when command is to run.  /interactive Allows the job to interact with the desktop of the user  who is logged on at the time the job runs.  /every:date[,...] Runs the command on each specified day(s) of the week or  month. If date is omitted, the current day of the month  is assumed.  /next:date[,...] Runs the specified command on the next occurrence of the  day (for example, next Thursday). If date is omitted, the  current day of the month is assumed.  "command" Is the Windows NT command, or batch program to be run. |
| --- |

To start MSSQLServer service at 6:30am and stop it at 6:30pm, run this:

at 6:30pm /EVERY:monday,tuesday,wednesday,thursday,friday,saturday,sunday "c:\stop.bat"

at 6:30am /EVERY:monday,tuesday,wednesday,thursday,friday,saturday,sunday "c:\start.bat"

Where start.bat:

**net start mssqlserver**

Stop.bat:

**net stop mssqlserver**

**I am a dbowner. Can I restrict access to my database for the 'sa'?**

**Answer:**  
  
No, you cannot. Members of the **sysadmin** fixed server role have full access to all system and user databases.  
  
See also this link to get more information about SQL Server roles:  
[**Understanding SQL Server roles**](http://www.mssqlcity.com/Articles/Adm/SQL70Roles.htm)

**I changed the name of NT server. How can I change the name of SQL Server?**

**Answer:**  
  
With SQL Server 7.0/2000 you need to rerun setup to reset the server name. Setup will detect the name conflict, resolve it, and then finish.

**I forgot the sa password. How I can reset it?**

**Answer:**  
  
To reset the sa password, you can make the following:  
  
1. Login to the SQL Server box as the Administrator.  
2. Run SQL Server Enterprise Manager.  
3. Right-click the server name and choose 'Edit SQL Server Registration properties'.  
4. Choose 'Use Windows authentication' and click OK button.  
5. Expand a server, expand a Security and click Logins.  
6. Double-click the sa login and specify new password on the General tab.

# SQL Server FAQ

## SQL Server Backup/Restore FAQ

**Can I backup a database to a network device?**

## Answer: Yes, you can. You should change the account the MSSQLServer service runs under to the account belonging to the Local Administrators group and to the Domain Users group of the appropriate domain, because the LocalSystem account has no access to shares on the network.

**Can I restore a SQL 7.0 database backup to a SQL 2000 server and vice versa?**

## Answer: You can restore only the users databases from the SQL Server 7.0 database backup to a SQL Server 2000 server, but you cannot restore system databases (such as master, msdb, model and tempdb) from the SQL Server 7.0 backup to a SQL Server 2000. You cannot restore a SQL Server 2000 database backup to a SQL Server 7.0 server. See also this link: [**How can I restore a SQL 6.5 database backup to a SQL 7.0/2000 server?**](http://www.mssqlcity.com/FAQ/BackRest/Res65Back.htm)

**Can I restore from backup with another sort order or character set?**

**Answer:**  
  
Usually, you cannot. See the SQL Server Books Online. It's therefrom:

| Usually, the sort order and character set of two servers must match to backup  on one server and restore on the other. An exception is when both servers use  a binary sort order; then, the character sets can differ.  When the code page or sort order differs, use the bcp utility or  Data Transformation Services to move the data. |
| --- |

**How can I backup/restore a single table in SQL 7.0/2000?**

## Answer: This opportunity is no longer supported. But you can create new filegroup and place a table in it. Because SQL Server can restore both files and filegroups, you can restore only that table. See also this link: [**File/Filegroups Optimization Tips**](http://www.mssqlcity.com/Tips/tipFile.htm)

**How can I restore a SQL 6.5 database backup to a SQL 7.0/2000 server?**

**Answer:**  
  
You cannot use SQL Server 6.5 backup in SQL Server 7.0/2000 and vice versa. You should previous upgrade SQL Server 6.5 to SQL Server 7.0/2000.  
  
See also this link:  
[**Can I restore a SQL 7.0 database backup to a SQL 2000 server and vice versa?**](http://www.mssqlcity.com/FAQ/BackRest/Res70Back.htm)

http://www.mssqlcity.com/FAQ/BackRest/Res70Back.htm

**How do I restore the master database? It tell me that I should be in the single user mode, but there is no such db option for the master database.**

**Answer:**  
  
Yes, there is no such database option for the **master** database. There is SQL Server configuration option. You need to run SQL Server in the single-user mode. Try the following:  
  
1. Stop MSSQLServer and SQLServerAgent services.  
2. From a command prompt, enter this command:  
  
*sqlservr.exe -m*  
  
3. Run Enterprise Manager to restore the **master** database from the backup.

**Is it possible to restore a particular transaction from a SQL Server backup?**

**Answer:**  
  
No, it is not possible. The only thing you can do is, using transaction log backups, restore up to a given point in time.

**We have no any backup, can we restore deleted data?**

**Answer:**  
  
Very popular question in newsgroups :(  
No, you cannot. After transaction was committed, you cannot rollback it. You should have backup to restore deleted data.

# SQL Server FAQ

## SQL Server Connectivity FAQ

**Can I connect to SQL Server over the internet?**

**Answer:**  
  
Yes, you can.  
This is from SQL Server Books Online:

| To share data over the Internet, the client and server must be connected  to the Internet. In addition, you must use the TCP/IP or Multiprotocol  Net-Libraries. If you use the Multiprotocol Net-Library, ensure that TCP/IP  support is enabled. If the server is registered with Domain Name System (DNS),  you can connect using its registered (friendly) name. |
| --- |

**Why dialup networking invoked when database is opened?**

**Answer:**  
  
Read about "Appearance of Internet Connection Dialog Box at Startup of Windows 95/98" in SQL Server Books Online (for both SQL Server 7.0/2000).  
  
It's therefrom:

| This behavior can be disabled by setting the registry key  EnableRemoteConnect to N.  To do this, create a text file named DisbleAutoConnect.reg  with the following three lines:  REGEDIT4  [HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\OLE]  "EnableRemoteConnect"="N" |
| --- |

**What sql net-lib should I use to improve the speed of my network connections?**

**Answer:**  
  
The TCP/IP Sockets is the most fastest sql net-lib.  
You can see the difference in responses times between any of the net-libs only on the slow network connections, on the normal LAN, you cannot see the difference in responses times.  
  
This is from SQL Server Books Online:

| In general, sockets are preferred in a slow LAN, WAN, or dial-up  network, whereas named pipes can be a better choice when network  speed is not the issue, as it offers more functionality, ease of  use, and configuration options. |
| --- |

**What sql net-lib should I use to enable encryption?**

**Answer:**  
  
You should use the multi-protocol network library to enable encryption.  
  
1. Run **Server Network Utility**.  
2. Add **Multiprotocol** if it does not appear in the **Server network** library list.  
3. On the **General** tab, select **Multiprotocol**, click **Edit** button, and select **Enable multi-protocol encryption**.  
  
**Note.** You do not need to use the multi-protocol network library to enable encryption in SQL Server 2000. SQL Server 2000 can use the Secure Sockets Layer (SSL) to encrypt all data transmitted between an application computer and a SQL Server instance on a database computer.

**Can I use the SQL Server 7.0 tools to connect to SQL Server 2000?**

**Answer:**  
  
Yes, you can use the SQL Query Analyzer, osql and isql SQL Server 7.0 tools to connect to SQL Server 2000.  
To connect to a named instance of SQL Server 2000, you must set up an alias using the Client Network Utility.  
  
This is from SQL Server 2000 Books Online:

| When using the SQL Server client connectivity components from SQL Server 7.0  or earlier, you must set up an alias using the Client Network Utility before  you connect to a named instance of SQL Server 2000. For example, on a  SQL Server 7.0 client, to connect to a named instance of SQL Server 2000,  you must add an alias that points to  \\computername\pipe\MSSQL$instancename\sql\query. If you use an alias name  of computername\instancename, clients can connect by specifying this name  in the same way as SQL Server 2000 clients do. For the TCP/IP Sockets and  NWLink IPX/SPX Net-Libraries, you must use the Client Network Utility to  define an alias on the client that specifies the port address on which the  named instance is listening. |
| --- |

**How can I connect to SQL Server through a firewall?**

**Answer:**  
  
You should ensure that SQL Server uses a specific port number for its incoming connections. The firewall system then forwards connections from the port number to SQL Server.  
  
See this link for more details:  
[**How do I connect to SQL Server through a firewall?**](http://www.ntfaq.com/Articles/Index.cfm?ArticleID=14098)

**What sql net-lib should I use to enable encryption?**

**Answer:**  
  
You should use the multi-protocol network library to enable encryption.  
  
1. Run **Server Network Utility**.  
2. Add **Multiprotocol** if it does not appear in the **Server network** library list.  
3. On the **General** tab, select **Multiprotocol**, click **Edit** button, and select **Enable multi-protocol encryption**.  
  
**Note.** You do not need to use the multi-protocol network library to enable encryption in SQL Server 2000. SQL Server 2000 can use the Secure Sockets Layer (SSL) to encrypt all data transmitted between an application computer and a SQL Server instance on a database computer.

# SQL Server FAQ

## SQL Server Development FAQ

**Can I work with registry from within T-SQL?**

**Answer:**  
  
Yes, you can. There are many extended stored procedures shipped with SQL Server and not all of them have been described in the SQL Server manuals. There are four very useful extended stored procedures to work with registry. There are: **xp\_regread**, **xp\_regwrite**, **xp\_regdeletevalue** and **xp\_regdeletekey**.  
  
See this article for more details:  
[**Useful undocumented extended stored procedures**](http://www.mssqlcity.com/Articles/Undoc/UndocExtSP.htm)

**How can I check from the SQL Server whether the file exists or not?**

**Answer:**  
  
You can use the **xp\_fileexist** undocumented extended stored procedure to determine whether the particular file exists on the disk or not.  
  
Syntax:  
  
*EXECUTE xp\_fileexist filename [, file\_exists INT OUTPUT]*  
  
To check whether the file boot.ini exists on the disk C: or not, run:  
  
*EXEC master..xp\_fileexist 'c:\boot.ini'*  
  
There are many extended stored procedures shipped with SQL Server and not all of them have been described in the SQL Server manuals. See this article to get the useful undocumented extended stored procedures description:  
[**Useful undocumented extended stored procedures**](http://www.mssqlcity.com/Articles/Undoc/UndocExtSP.htm)

**How can I copy text or image into or out of SQL Server?**

**Answer:**  
  
You can use **textcopy** utility.  
  
See this article for more information about the **textcopy** utility:  
[**Copy text or image into or out of SQL Server**](http://www.mssqlcity.com/Articles/KnowHow/Textcopy.htm)

**How can I remove foreign keys through SQL Server 7.0/2000 Enterprise Manager?**

**Answer:**  
  
To drop foreign keys, you can do the following:  
  
1. Run SQL Server Enterprise Manager.  
2. Expand a server group; then expand a server.  
3. Expand Databases; then expand appropriate database.  
4. Click Tables; then right-click the table and choose 'Design Table'.  
5. Click the 'Table and Index Properties' button on the top menu.  
6. On the 'Relationships' tab choose appropriate relationship and click 'Delete' button.  
  
You can also drop foreign keys from the diagrams in the Enterprise Manager (make diagram and drop foreign keys).

**How can I make database to be read only?**

**Answer:**  
  
To make database to be read only, you can use the Enterprise Manager or the **sp\_dboption** system stored procedure.  
  
For example, to make the pubs database to be read only, you can use the following statement:  
  
*EXEC sp\_dboption 'pubs', 'read only', 'TRUE'*  
  
Read SQL BOL to get more information about the **sp\_dboption** system stored procedure.

**How can I get only the specified number of rows from my query?**

**Answer:**  
  
You can use the **SET ROWCOUNT** statement or the TOP clause of the select statement (there is no TOP clause in SQL Server 6.5).  
There are examples to return only the 10 top rows from the authors table in the pubs database:

| SET ROWCOUNT 10  GO  SELECT \* FROM pubs.dbo.authors  GO |
| --- |

or  
  
*SELECT TOP 10 \* FROM pubs.dbo.authors*  
  
Read about the **SET ROWCOUNT** statement in SQL Server Books Online.

**How can I export a query result set into file?**

**Answer:**  
  
There are several ways to do it.  
  
1. You can use bcp utility.  
Read about bcp utility in SQL Server Books online. This is the example to export data from the authors table in the pubs database into authors.txt file on the drive C:

| EXEC master..xp\_cmdshell  'bcp "select \* from pubs..authors" queryout c:\authors.txt -c -Usa -P' |
| --- |

2. You can use **osql** or **isql** utility.  
This is the example to export data from the authors table in the pubs database into authors.txt file on the drive C:

| EXEC master..xp\_cmdshell  'osql -S ServerName -U sa -P -q "select \* from pubs..authors" -o c:\authors.txt' |
| --- |

3. You can use DTS Export Wizard.  
  
4. You can use SQL-DMO.  
(**sp\_OACreate**, **sp\_OAMethod**, **sp\_OAGetProperty**, **sp\_OASetProperty** and so on).  
  
See this article for more details:  
[**Working with COM objects from within T-SQL**](http://www.mssqlcity.com/Articles/General/OleAutSP.htm) : http://www.mssqlcity.com/Articles/General/OleAutSP.htm

**How can I enforce to use particular index?**

**Answer:**  
  
You can use index hint (index=<index\_name>) after the table name. This is the example to use the aunmind index to get the last names of each author from the authors table:

| USE pubs  SELECT au\_lname FROM authors (index=aunmind) |
| --- |

**How can I edit the system tables manually?**

**Answer:**  
  
It is not recommended to do, but it is possible. You should set the **allow updates** option to 1 by using the **sp\_configure** system stored procedure.  
  
This is the example:

| sp\_configure 'allow updates', 1  GO  RECONFIGURE WITH OVERRIDE  GO |
| --- |

Read about the **sp\_configure** system stored procedure in SQL Server Books Online.

**How can I delete duplicate rows?**

**Answer:**  
  
Check this link to delete duplicate rows:  
[**INF: How to Remove Duplicate Rows From a Table**](http://support.microsoft.com/support/kb/articles/q139/4/44.asp) : http://support.microsoft.com/support/kb/articles/q139/4/44.asp

**How can I decrypt a SQL Server stored procedure?**

**Answer:**  
  
Check these links to decrypt stored procedure:  
[**SQL Server 7.0 Stored Procedure Decryptor**](http://www.planetsourcecode.com/vb/scripts/ShowCode.asp?lngWId=5&txtCodeId=215)  
[**How can I decrypt a SQL Server stored-procedure?**](http://www.windows2000faq.com/Articles/Index.cfm?ArticleID=14369)

**How can I create extended stored procedures?**

**Answer:**  
  
To create your own extended stored procedure, you can use Visual C++ or Delphi.  
  
See these links to create an extended stored procedure by using Visual C++:  
[**Creating Extended Stored Procedures**](http://ddart.net/mssql/sql2000/html/odssql/ods_6_con_01_22sz.htm)  
[**Enhance SQL Server with your own Extended Stored Procedures**](http://archive.devx.com/free/mgznarch/vcdj/1998/mar98/sql1.asp)  
  
See this link to create an extended stored procedure by using Delphi:  
[**Extended Stored Procedure component for Delphi**](http://www.bramc.ru/soft/xproc.html)  
  
After you create your own extended stored procedure, you should register the name of a new extended stored procedure to SQL Server by using the **sp\_addextendedproc** system stored procedure.  
  
Read about the **sp\_addextendedproc** system stored procedure in SQL Server Books Online.  
  
See this article for more details about extended stored procedures:  
[**HOW TO: Use Extended Stored Procedures**](http://support.microsoft.com/support/kb/articles/q190/9/87.asp)

**How can I return from within a T-SQL the list of all available OLE DB providers?**

**Answer:**  
  
You can use the **xp\_enum\_oledb\_providers** undocumented extended stored procedure to return the list of all available server's OLE DB providers. This extended stored procedure does not require any parameters.  
  
This is the example to run:  
  
*EXEC master..xp\_enum\_oledb\_providers*  
  
See this article to get more information about the useful undocumented extended stored procedures:  
[**Useful undocumented extended stored procedures**](http://www.mssqlcity.com/Articles/Undoc/UndocExtSP.htm)

**How can I return from within a T-SQL the list of all available character sets and code pages?**

**Answer:**  
  
You can use the **xp\_enumcodepages** undocumented extended stored procedure to return the list of all available character sets and code pages. This extended stored procedure does not require any parameters.  
  
This is the example to run:  
  
*EXEC master..xp\_enumcodepages*  
  
See this article to get more information about the useful undocumented extended stored procedures:  
[**Useful undocumented extended stored procedures**](http://www.mssqlcity.com/Articles/Undoc/UndocExtSP.htm)

**Why SELECT INTO statement does not work?**

**Answer:**  
  
First of all, chect the syntax of your SELECT INTO statement. If the syntax is correct, check that the **select into/bulkcopy** database option is turned on.  
  
To check that the **select into/bulkcopy** database option is turned on, you can use the Enterprise Manager or the **sp\_dboption** system stored procedure.  
For example, to check that the **select into/bulkcopy** database option is turned on for the pubs database, you can use the following statement:  
  
*EXEC sp\_dboption pubs, 'select into/bulkcopy'*  
  
To turn on the **select into/bulkcopy** database option, you can use the following statement:  
  
*EXEC sp\_dboption 'pubs', 'select into/bulkcopy', 'TRUE'*  
  
Read SQL BOL to get more information about the **sp\_dboption** system stored procedure.

**Is it possible to work with COM objects from within T-SQL?**

**Answer:**  
  
You can use OLE Automation Stored Procedures (extended stored procedures with sp\_OA prefix) to create a COM object in T-SQL and use the object's methods and properties.  
  
See this article for more details:  
[**Working with COM objects from within T-SQL**](http://www.mssqlcity.com/Articles/General/OleAutSP.htm)

**How to install SQL scripts and import data to SQL Server from the command line?**

**Answer:**  
  
You can create \*.bat file which include something like this:  
  
*osql -S ServerName -U LoginName -P password -i c:\script.sql*  
  
where c:\script.sql - the script to create metadata and load data. See instpubs.sql file in the Install subdirectory of your SQL Server installation as example of creating such script.

**How do I write to the registry via Transact-SQL?**

**Answer:**  
  
You can use the **xp\_regwrite** undocumented extended stored procedure to write to the registry via Transact-SQL.  
  
This is the syntax of the **xp\_regwrite**:

| EXECUTE xp\_regwrite [@rootkey=]'rootkey',  [@key=]'key',  [@value\_name=]'value\_name',  [@type=]'type',  [@value=]'value' |
| --- |

For example, to write the variable 'Test' to the 'TestValue' value, key 'SOFTWARE\Test', 'HKEY\_LOCAL\_MACHINE', run:

| EXEC master..xp\_regwrite  @rootkey='HKEY\_LOCAL\_MACHINE',  @key='SOFTWARE\Test',  @value\_name='TestValue',  @type='REG\_SZ',  @value='Test' |
| --- |

See this FAQ question:  
[**How do I read from the registry via Transact-SQL?**](http://www.mssqlcity.com/FAQ/Devel/xp_regread.htm)  
  
See this article to get more useful undocumented extended stored procedures:  
[**Useful undocumented extended stored procedures**](http://www.mssqlcity.com/Articles/Undoc/UndocExtSP.htm)

**How do I run SQL script located on the disk file, using Transact-SQL?**

**Answer:**  
  
You can use **osql** or **isql** utility to run SQL script located on the disk file. This is the example to run SQL script located in the C:\SQLScript.sql file:  
  
*EXEC master..xp\_cmdshell 'osql -S Servername -U sa -P -ic:\SQLScript.sql'*  
  
See SQL Server Books Online for more information about **osql** or **isql** utility and about **xp\_cmdshell** extended stored procedure.

**How do I run a T-SQL script through SQL-DMO?**

**Answer:**  
  
You can use **ExecuteImmediate** or **CommandShellImmediate** methods of SQL-DMO. This is the example to execute T-SQL script from the c:\trig.sql file:

| DECLARE @object int  DECLARE @hr int  DECLARE @property varchar(255)  DECLARE @return varchar(255)  EXEC @hr = sp\_OACreate 'SQLDMO.SQLServer', @object OUT  IF @hr <> 0  BEGIN  print 'error create SQLDMO.SQLServer'  RETURN  END  EXEC @hr = sp\_OAMethod  @object, 'Connect', NULL, 'ServerName', 'Login', 'Password'  IF @hr <> 0  BEGIN  print 'error with Connect'  RETURN  END  EXEC @hr = sp\_OAMethod @object, 'VerifyConnection', @return OUT  IF @hr <> 0  BEGIN  PRINT 'error with VerifyConnection'  RETURN  END  select @property = 'EXEC master..xp\_cmdshell "osql -S ServerName ' +  '-U LoginName -P password -i c:\trig.sql"'  EXEC @hr = sp\_OAMethod @object, 'ExecuteImmediate', Null , @property  IF @hr <> 0  BEGIN  print 'error with ExecuteImmediate'  RETURN  END  -- ahother way to run a T-SQL script  select @property = 'osql -S ServerName -U LoginName -P password -i c:\trig.sql'  EXEC @hr = sp\_OAMethod @object, 'CommandShellImmediate', Null , @property  IF @hr <> 0  BEGIN  print 'error with CommandShellImmediate'  RETURN  END |
| --- |

Read about the **ExecuteImmediate** or **CommandShellImmediate** methods of SQL-DMO in SQL Server Books Online.  
  
See also this link:  
[**Working with COM objects from within T-SQL**](http://www.mssqlcity.com/Articles/General/OleAutSP.htm)

**How do I run a DTS package from the Query Analyzer or stored procedure?**

**Answer:**  
  
There are two ways to run a DTS package from the Query Analyzer or stored procedure:  
  
1. By using OLE Automation objects  
(run **sp\_OACreate**, **sp\_OAGetProperty**, **sp\_OAMethod** and so on).  
  
2. By using **xp\_cmdshell** stored procedure to run dtsrun.exe file.  
  
This is the example to execute NewPack.dts package from the disk C:

| DECLARE @object int  DECLARE @hr int  --create a package object  EXEC @hr = sp\_OACreate 'DTS.Package', @object OUTPUT  if @hr <> 0  BEGIN  print 'error create DTS.Package'  RETURN  END  EXEC @hr = sp\_OAMethod @object, 'LoadFromStorageFile',  NULL, 'C:\NewPack.dts', ''  IF @hr <> 0  BEGIN  print 'error LoadFromStorageFile'  RETURN  END  EXEC @hr = sp\_OAMethod @object, 'Execute'  IF @hr <> 0  BEGIN  print 'Execute failed'  RETURN  END |
| --- |

See also this link:  
[**Working with COM objects from within T-SQL**](http://www.mssqlcity.com/Articles/General/OleAutSP.htm)

**How do I return the last day of the month for the given date?**

**Answer:**  
  
To return the last day of the month for the given date, you can use the LastMonthDay user-defined function.

#### Syntax

LastMonthDay (date)

#### Arguments

**date**  
Is the datetime value.

#### Return Types

datetime

#### The function's text:

| CREATE FUNCTION dbo.LastMonthDay  ( @Date datetime )  RETURNS datetime  AS  BEGIN  RETURN (CASE WHEN MONTH(@Date)= 12  THEN DATEADD(day, -1, CAST('01/01/' + STR(YEAR(@Date)+1) AS DateTime))  ELSE DATEADD(day, -1, CAST(STR(MONTH(@Date)+1) + '/01/' + STR(YEAR(@Date)) AS DateTime))  END)  END  GO |
| --- |

#### Example

Returns the last day for the '06/15/99' date:

| SELECT dbo.LastMonthDay('06/15/99')  GO |
| --- |

Here is the result set:

| ------------------------------------------------------  1999-06-30 00:00:00.000  (1 row(s) affected) |
| --- |

**Note.** The user-defined functions are only available with SQL Server 2000, SQL Server 7.0 does not support user-defined functions.  
  
See this article for more datetime user-defined functions:  
[**Date and Time User-Defined Functions**](http://www.mssqlcity.com/Articles/UDF/DateTimeUDF.htm)

**How do I reset the identity column?**

**Answer:**  
  
You can use the **DBCC CHECKIDENT** statement, if you want to reset or reseed the identity column. For example, if you need to force the current identity value in the jobs table to a value of 100, you can use the following:

| USE pubs  GO  DBCC CHECKIDENT (jobs, RESEED, 100)  GO |
| --- |

See **DBCC CHECKIDENT** in SQL Server Books Online for more details.

**How do I return the first day of the month for the given date?**

**Answer:**  
  
To return the first day of the month for the given date, you can use the FirstMonthDay user-defined function.

#### Syntax

FirstMonthDay (date)

#### Arguments

**date**  
Is the datetime value.

#### Return Types

datetime

#### The function's text:

| CREATE FUNCTION dbo.FirstMonthDay  ( @Date datetime )  RETURNS datetime  AS  BEGIN  RETURN (CAST(STR(MONTH(@Date))+'/'+STR(01)+'/'+STR(YEAR(@Date)) AS DateTime))  END  GO |
| --- |

#### Example

Returns the first day for the '06/15/99' date:

| SELECT dbo.FirstMonthDay('06/15/99')  GO |
| --- |

Here is the result set:

| ------------------------------------------------------  1999-06-01 00:00:00.000  (1 row(s) affected) |
| --- |

**Note.** The user-defined functions are only available with SQL Server 2000, SQL Server 7.0 does not support user-defined functions.

See this article for more datetime user-defined functions:  
[**Date and Time User-Defined Functions**](http://www.mssqlcity.com/Articles/UDF/DateTimeUDF.htm)

**How do I read from the registry via Transact-SQL?**

**Answer:**  
  
You can use the **xp\_regread** undocumented extended stored procedure to read from the registry via Transact-SQL.  
  
This is the syntax of the **xp\_regread**:

| EXECUTE xp\_regread [@rootkey=]'rootkey',  [@key=]'key'  [, [@value\_name=]'value\_name']  [, [@value=]@value OUTPUT] |
| --- |

For example, to read into the variable @test from the value 'TestValue' from the key 'SOFTWARE\Test' from the 'HKEY\_LOCAL\_MACHINE', run:

| DECLARE @test varchar(20)  EXEC master..xp\_regread @rootkey='HKEY\_LOCAL\_MACHINE',  @key='SOFTWARE\Test',  @value\_name='TestValue',  @value=@test OUTPUT  SELECT @test |
| --- |

See this article to get more useful undocumented extended stored procedures:  
[**Useful undocumented extended stored procedures**](http://www.mssqlcity.com/Articles/Undoc/UndocExtSP.htm)

**How do I perform the same actions for all tables in a database?**

**Answer:**  
  
If you need to perform the same actions for all tables in a database, you can create cursor or you can use the **sp\_MSforeachtable** undocumented stored procedure to accomplish the same goal with less work.  
  
For example, you can use the **sp\_MSforeachtable** stored procedure to rebuild all indexes in a database:  
  
*EXEC sp\_MSforeachtable @command1="print '?' DBCC DBREINDEX ('?')"*  
  
See also these links:  
[**SQL Server 2000 Useful undocumented stored procedures**](http://www.mssqlcity.com/Articles/Undoc/SQL2000UndocSP.htm)  
[**SQL Server 7.0 Useful undocumented stored procedures**](http://www.mssqlcity.com/Articles/Undoc/SQL70UndocSP.htm)

**How do I perform the same actions for all databases?**

**Answer:**  
  
If you need to perform the same actions for all databases, you can create cursor or you can use the **sp\_MSforeachdb** undocumented stored procedure to accomplish the same goal with less work.  
  
For example, you can use the **sp\_MSforeachdb** stored procedure to run the **DBCC CHECKDB** statement for all the databases on your server:  
  
*EXEC sp\_MSforeachdb @command1="print '?' DBCC CHECKDB ('?')"*  
  
See also these links:  
[**SQL Server 2000 Useful undocumented stored procedures**](http://www.mssqlcity.com/Articles/Undoc/SQL2000UndocSP.htm)  
[**SQL Server 7.0 Useful undocumented stored procedures**](http://www.mssqlcity.com/Articles/Undoc/SQL70UndocSP.htm)

**How do I move tables from one filegroup to another?**

**Answer:**  
  
To move tables from one filegroup to another, you can do the following:  
  
1. Run SQL Server Enterprise Manager.  
2. Expand a server group; then expand a server.  
3. Expand Databases; then expand appropriate database.  
4. Click Tables; then right-click the table and choose 'Design Table'.  
5. Click 'Table and Index Properties' on the top menu and change the table filegroup.

**How do I get working days between two dates?**

**Answer:**  
  
To return the number of working days between two dates (not including these dates), you can use GetWorkingDays user-defined function.

#### Syntax

GetWorkingDays (StartDate, EndDate)

#### Arguments

**StartDate**  
Is the datetime value (start date).  
  
**EndDate**  
Is the datetime value (end date).

#### Return Types

int

#### The function's text:

| CREATE FUNCTION dbo.GetWorkingDays  ( @StartDate datetime,  @EndDate datetime )  RETURNS INT  AS  BEGIN  DECLARE @WorkDays int, @FirstPart int  DECLARE @FirstNum int, @TotalDays int  DECLARE @LastNum int, @LastPart int  IF (DATEDIFF(day, @StartDate, @EndDate) < 2)  BEGIN  RETURN ( 0 )  END  SELECT  @TotalDays = DATEDIFF(day, @StartDate, @EndDate) - 1,  @FirstPart = CASE DATENAME(weekday, @StartDate)  WHEN 'Sunday' THEN 6  WHEN 'Monday' THEN 5  WHEN 'Tuesday' THEN 4  WHEN 'Wednesday' THEN 3  WHEN 'Thursday' THEN 2  WHEN 'Friday' THEN 1  WHEN 'Saturday' THEN 0  END,  @FirstNum = CASE DATENAME(weekday, @StartDate)  WHEN 'Sunday' THEN 5  WHEN 'Monday' THEN 4  WHEN 'Tuesday' THEN 3  WHEN 'Wednesday' THEN 2  WHEN 'Thursday' THEN 1  WHEN 'Friday' THEN 0  WHEN 'Saturday' THEN 0  END  IF (@TotalDays < @FirstPart)  BEGIN  SELECT @WorkDays = @TotalDays  END  ELSE  BEGIN  SELECT @WorkDays = (@TotalDays - @FirstPart) / 7  SELECT @LastPart = (@TotalDays - @FirstPart) % 7  SELECT @LastNum = CASE  WHEN (@LastPart < 7) AND (@LastPart > 0) THEN @LastPart - 1  ELSE 0  END  SELECT @WorkDays = @WorkDays \* 5 + @FirstNum + @LastNum  END  RETURN ( @WorkDays )  END  GO |
| --- |

#### Example

Returns the number of working days between '11/13/2000' and '12/27/2000':

| SELECT dbo.GetWorkingDays ('11/13/2000', '12/27/2000')  GO |
| --- |

Here is the result set:

| -----------  31  (1 row(s) affected) |
| --- |

**Note.** The user-defined functions are only available with SQL Server 2000, SQL Server 7.0 does not support user-defined functions.  
  
See this article for more datetime user-defined functions:  
[**Date and Time User-Defined Functions**](http://www.mssqlcity.com/Articles/UDF/DateTimeUDF.htm)

**How do I get the date the object was created?**

**Answer:**  
  
To get the date the object was created, you can query the **sysobjects** system table to see the crdate column. This is the example to get the date the authors table was created:  
  
*SELECT crdate FROM sysobjects WHERE name = 'authors'*  
  
Read about the **sysobjects** system table in SQL Server Books Online.

**How do I execute an operating-system command via Transact-SQL?**

**Answer:**  
  
You can use the **xp\_cmdshell** extended stored procedure to execute a given command string as an operating-system command shell and return any output as rows of text.  
  
This is the syntax of the **xp\_cmdshell**:  
  
*xp\_cmdshell {'command\_string'} [, no\_output]*  
  
where  
  
'command\_string' is varchar(255) or nvarchar(4000), with no default.  
no\_output - when used no output will be returned.  
  
When executing **xp\_cmdshell** with the Windows 9x operating systems, the return code will always be 0.  
  
Executing this **xp\_cmdshell** statement sends the 'Hello' message to JOHN:  
  
*EXEC master..xp\_cmdshell "net send JOHN 'Hello'"*  
  
See SQL BOL for more information about the **xp\_cmdshell** extended stored procedure.

**How do I delete an entire key from the registry via Transact-SQL?**

**Answer:**  
  
You can use the **xp\_regdeletekey** undocumented extended stored procedure to delete an entire key from the registry via Transact-SQL.  
  
This is the syntax of the **xp\_regdeletekey**:

| EXECUTE xp\_regdeletekey [@rootkey=]'rootkey',  [@key=]'key' |
| --- |

For example, to delete the key 'SOFTWARE\Test' from 'HKEY\_LOCAL\_MACHINE', run:

| EXEC master..xp\_regdeletekey  @rootkey='HKEY\_LOCAL\_MACHINE',  @key='SOFTWARE\Test' |
| --- |

See this FAQ question:  
[**How do I read from the registry via Transact-SQL?**](http://www.mssqlcity.com/FAQ/Devel/xp_regread.htm)  
  
See this article to get more useful undocumented extended stored procedures:  
[**Useful undocumented extended stored procedures**](http://www.mssqlcity.com/Articles/Undoc/UndocExtSP.htm)

**How do I delete a particular value for a key in the registry via Transact-SQL?**

**Answer:**  
  
You can use the **xp\_regdeletevalue** undocumented extended stored procedure to delete a particular value for a key in the registry via Transact-SQL.  
  
This is the syntax of the **xp\_regdeletevalue**:

| EXECUTE xp\_regdeletevalue [@rootkey=]'rootkey',  [@key=]'key',  [@value\_name=]'value\_name' |
| --- |

For example, to delete the value 'TestValue' for the key 'SOFTWARE\Test' from 'HKEY\_LOCAL\_MACHINE', run:

| EXEC master..xp\_regdeletevalue  @rootkey='HKEY\_LOCAL\_MACHINE',  @key='SOFTWARE\Test',  @value\_name='TestValue' |
| --- |

See this FAQ question:  
[**How do I read from the registry via Transact-SQL?**](http://www.mssqlcity.com/FAQ/Devel/xp_regread.htm)  
  
See this article to get more useful undocumented extended stored procedures:  
[**Useful undocumented extended stored procedures**](http://www.mssqlcity.com/Articles/Undoc/UndocExtSP.htm)

**How do I add 'n' working days to the given date and get the result date?**

**Answer:**  
  
To add 'n' working days to the given date and return the result date, you can use AddWorkingDays user-defined function. The AddWorkingDays udf works like DATEADD, but adds the working days.

#### Syntax

AddWorkingDays ( StartDate, WorkDays )

#### Arguments

**StartDate**  
Is the datetime value (start date).  
  
**WorkDays**  
Is the integer value (number of working days).

#### Return Types

datetime

#### The function's text:

| CREATE FUNCTION dbo.ADDWorkingDays  ( @StartDate datetime,  @WorkDays int )  RETURNS datetime  AS  BEGIN  DECLARE @TotalDays int, @FirstPart int  DECLARE @EndDate datetime  DECLARE @LastNum int, @LastPart int  IF @WorkDays < 0  BEGIN  SELECT @FirstPart = CASE DATENAME(weekday, @StartDate)  WHEN 'Sunday' THEN 0  WHEN 'Monday' THEN 1  WHEN 'Tuesday' THEN 2  WHEN 'Wednesday' THEN 3  WHEN 'Thursday' THEN 4  WHEN 'Friday' THEN 5  WHEN 'Saturday' THEN 6  END  IF ABS(@WorkDays) < @FirstPart  SELECT @EndDate = DATEADD(dd, @WorkDays, @StartDate)  ELSE  BEGIN  SELECT @TotalDays = (ABS(@WorkDays) - @FirstPart) / 5  SELECT @LastPart = (ABS(@WorkDays) - @FirstPart) % 7  SELECT @LastNum = CASE  WHEN (@LastPart < 7) AND (@LastPart > 0) THEN @LastPart - 1  ELSE 0  END  SELECT @TotalDays = - 2 \* (@TotalDays + 1) + @WorkDays  SELECT @EndDate = DATEADD(dd, @TotalDays, @StartDate)  END  END  ELSE  BEGIN  SELECT @FirstPart = CASE DATENAME(weekday, @StartDate)  WHEN 'Sunday' THEN 6  WHEN 'Monday' THEN 5  WHEN 'Tuesday' THEN 4  WHEN 'Wednesday' THEN 3  WHEN 'Thursday' THEN 2  WHEN 'Friday' THEN 1  WHEN 'Saturday' THEN 0  END  IF @WorkDays < @FirstPart  SELECT @EndDate = DATEADD(dd, @WorkDays, @StartDate)  ELSE  BEGIN  SELECT @TotalDays = (@WorkDays - @FirstPart) / 5  SELECT @LastPart = (@WorkDays - @FirstPart) % 7  SELECT @LastNum = CASE  WHEN (@LastPart < 7) AND (@LastPart > 0) THEN @LastPart - 1  ELSE 0  END  SELECT @TotalDays = 2 \* (@TotalDays + 1) + @WorkDays  SELECT @EndDate = DATEADD(dd, @TotalDays, @StartDate)  END  END  RETURN ( @EndDate )  END  GO |
| --- |

#### Example

Adds 9 working days to '11/13/2002' and returns the result date:

| SELECT dbo.AddWorkingDays ('11/13/2002', 9)  GO |
| --- |

Here is the result set:

| ------------------------------------------------------  2002-11-26 00:00:00.000  (1 row(s) affected) |
| --- |

**Note.** The user-defined functions are only available with SQL Server 2000, SQL Server 7.0 does not support user-defined functions.  
  
See this article for more datetime user-defined functions:  
[**Date and Time User-Defined Functions**](http://www.mssqlcity.com/Articles/UDF/DateTimeUDF.htm)

**How can I use a variable for the columns/tables in the select statements?**

**Answer:**  
  
SQL Server can contain both static sql statements and dynamic sql statements. In comparison with static sql statements, dynamic sql statements are not completely embedded in the source code; instead, portions are stored in program variables that can be modified at run time.  
So, you can use dynamic sql statements (using EXECUTE command) to execute a character string which contain a variable for the columns/tables in the select statements. Read about the EXECUTE command in SQL Server Books Online.  
This is the example to make select from the table, which name was passed into the stored procedure as variable:

| CREATE PROCEDURE GetRec  ( @tableName sysname )  AS  EXEC ('SELECT \* from ' + @tableName) |
| --- |

**How can I set database to a single user mode?**

**Answer:**  
  
To set database to a single user mode, you can use the Enterprise Manager or the **sp\_dboption** system stored procedure.  
  
For example, to set pubs database to a single user mode, you can use the following statement:  
  
*EXEC sp\_dboption 'pubs', 'single user', 'TRUE'*  
  
Read SQL BOL to get more information about the **sp\_dboption** system stored procedure.

**How can I send a message to user from the SQL Server?**

**Answer:**  
  
You can use the **xp\_cmdshell** extended stored procedure to run **net send** command. This is the example to send the 'Hello' message to JOHN:  
  
*EXEC master..xp\_cmdshell "net send JOHN 'Hello'"*  
  
To get **net send** message on the Windows 9x machines, you should run the WinPopup utility. You can place WinPopup in the Startup group under Program Files.

**How can I select only date/time part of a datetime value?**

**Answer:**  
  
--to select date part only  
SELECT CONVERT(char(10),GetDate(),101)  
  
--to select time part only  
SELECT right(GetDate(),7)

**How can I return from within a T-SQL the list of Windows NT/2000 groups?**

**Answer:**  
  
You can use the **xp\_enumgroups** undocumented extended stored procedure to return the list of Windows NT/2000 groups and their descriptions. This extended stored procedure does not require any parameters.  
  
This is the example to run:  
  
*EXEC master..xp\_enumgroups*  
  
See this article to get more information about the useful undocumented extended stored procedures:  
[**Useful undocumented extended stored procedures**](http://www.mssqlcity.com/Articles/Undoc/UndocExtSP.htm)

**How can I return from within a T-SQL the list of server's hard drives?**

**Answer:**  
  
You can use the **xp\_fixeddrives** undocumented extended stored procedure to return the list of all hard drives and the amount of free space in Mb for each hard drive. This extended stored procedure does not require any parameters.  
  
This is the example to run:  
  
*EXEC master..xp\_fixeddrives*  
  
See this article to get more information about the useful undocumented extended stored procedures:  
[**Useful undocumented extended stored procedures**](http://www.mssqlcity.com/Articles/Undoc/UndocExtSP.htm)

**How can I return from within a T-SQL the list of server's DSNs?**

**Answer:**  
  
You can use the **xp\_enumdsn** undocumented extended stored procedure to return the list of server's DSNs. This extended stored procedure does not require any parameters.  
  
This is the example to run:  
  
*EXEC master..xp\_enumdsn*  
  
See this article to get more information about the useful undocumented extended stored procedures:  
[**Useful undocumented extended stored procedures**](http://www.mssqlcity.com/Articles/Undoc/UndocExtSP.htm)

# SQL Server FAQ

## SQL Server General FAQ

**Are there arrays in SQL Server? Or what is the array analog?**

**Answer:**  
  
There are no arrays in SQL Server. You can use the character string to store the comma-separated values and then use PATINDEX function to extract the values, or you can store the values in separate table.  
  
See this link about sql server arrays:  
[**How can I pass an array of values to a SQL Server stored-procedure?**](http://www.ntfaq.com/Articles/Index.cfm?ArticleID=14242)

**Where can I see the price-list for SQL Server?**

**Answer:**  
  
Check this link about SQL Server prices:  
[**How to Buy**](http://www.microsoft.com/sql/howtobuy/production.asp)  
  
This is the price comparison between SQL Server 2000 and Oracle 9i:  
[**Price Comparisons**](http://www.microsoft.com/sql/evaluation/compare/pricecomparison.asp)

**Where can I get the evaluation copy of SQL Server?**

**Answer:**  
  
You can find the 120 days evaluation copy of SQL Server 2000 at here:  
[**http://www.microsoft.com/sql/evaluation/trial/2000/default.asp**](http://www.microsoft.com/sql/evaluation/trial/2000/default.asp)

**Where can I get the benchmark tests for SQL Server and other RDBMS?**

**Answer:**  
  
To get benchmark tests for SQL Server and other RDBMS, visit the Transaction Processing Performance Council Web site at [**http://www.tpc.org**](http://www.tpc.org) Here you can find the price and performance comparison of the SQL Server and other RDBMS on the different hardware and software platforms.

**Where can I find MSSQL test exams?**

**Answer:**  
  
You can find SQL Server test exams at here:  
  
[**MSSQLCity.Com**](http://www.mssqlcity.com/TestExams.htm)  
(contains 10 free questions per SQL Server test exam)  
  
[**Transcender**](http://www.transcender.com/)  
(contains 12 free questions per SQL Server test exam)  
  
[**Measureup**](http://www.measureup.com/)  
(contains 10 free questions per SQL Server test exam)  
  
[**Brainbench**](http://www.brainbench.com/)  
(does not contain free SQL Server test exams)

**Where can I download SQL Server 2000 Books Online (SQL BOL)?**

**Answer:**  
  
You can download SQL BOL here:  
[**SQL Server 2000 Books Online (SQL BOL)**](http://www.microsoft.com/sql/techinfo/productdoc/2000/books.asp)

**Where can I download free case tools?**

**Answer:**  
  
You can download free case tools at here:  
  
Free trial version of ER/Studio  
[**case tools download**](http://www.embarcadero.com/downloads/downloadERStudio.asp)  
  
DbVisualizer  
[**case tools download**](http://www.minq.se/products/dbvis/download.html)  
  
PowerDesigner 9.0 Evaluation Version (you should register before)  
[**case tools download**](http://crm.sybase.com/sybase/www/eBD/pd90_initial_contact.jsp)  
  
PowerDesigner 9.5 Beta Version (you should register before)  
[**case tools download**](http://www.sybase.com/pd_beta)

**What is the difference between PRIMARY KEY and UNIQUE constraints?**

**Answer:**  
  
Both PRIMARY KEY constraint and UNIQUE constraint uses to enforce Entity integrity (defines a row as a unique entity for a particular table), but primary keys do not allow null values.

**What is the difference between permanent and temporary tables?**

**Answer:**  
  
Temporary tables are similar to permanent tables, except temporary tables are stored in tempdb database and are deleted automatically when no longer in use.

**What is the difference between Oracle and SQL Server?**

**Answer:**  
  
Often people in newsgroups ask about some comparison of Oracle and SQL Server. In the article below, you can find the comparison of Oracle 9i Database with SQL Server 2000 regarding price, performance, platforms supported, SQL dialects and products limits:  
[**SQL Server 2000 vs Oracle 9i**](http://www.mssqlcity.com/Articles/Compare/sql_server_vs_oracle.htm)

**What is the difference between clustered and nonclustered indexes?**

**Answer:**  
  
There are clustered and nonclustered indexes. A clustered index is a special type of index that reorders the way records in the table are physically stored. Therefore table can have only one clustered index. The leaf nodes of a clustered index contain the data pages.  
  
A nonclustered index is a special type of index in which the logical order of the index does not match the physical stored order of the rows on disk. The leaf node of a nonclustered index does not consist of the data pages. Instead, the leaf nodes contain index rows.  
  
See this article fot more details: [**Index Optimization Tips**](http://www.mssqlcity.com/Articles/Tuning/IndexOptimTips.htm)

**What is the difference between char and varchar data types?**

**Answer:**  
  
The **char** is a fixed-length character data type, the **varchar** is a variable-length character data type.  
  
Because **char** is a fixed-length data type, the storage size of the **char** value is equal to the maximum size for this column. Because **varchar** is a variable-length data type, the storage size of the **varchar** value is the actual length of the data entered, not the maximum size for this column.  
  
You can use **char** when the data entries in a column are expected to be the same size.  
You can use **varchar** when the data entries in a column are expected to vary considerably in size.

**What are good SQL Server web resources?**

**Answer:**  
  
You can find SQL Server links at here:  
[**Microsoft SQL Server Related Links**](http://www.mssqlcity.com/Links.htm)

**How to detect if the SQL Server is installed on the machine?**

**Answer:**  
  
To check that SQL Server is installed you can check that this registry key exists:  
  
*HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\MSSQLServer\MSSQLServer\CurrentVersion*  
  
By the way, you should connect to SQL Server to guarantee that SQL Server is currently available.  
  
See also this link:  
[**How do I test an installation of SQL Server?**](http://www.mssqlcity.com/FAQ/InstUpg/TestInst.htm)

**How many SQL 2000 editions exist and what is the difference between them?**

**Answer:**  
  
SQL Server 2000 has the following editions:

Personal Edition

Standard Edition

Enterprise Edition

Developer Edition

Desktop Engine

SQL Server CE

Evaluation Edition  
  
**Personal Edition** can work on the Windows 98, Windows NT Server 4.0 with Service Pack 5 or later, Windows NT Workstation 4.0 with Service Pack 5 or later and on the all editions of Windows 2000. This edition is related to SQL Server 7.0 Desktop Edition.  
  
This edition has some restrictions:

maximum 2 CPU

no Distributed Partitioned Views

no Log Shipping

no Parallel DBCC

no Parallel index creation

no Failover clustering

no publishing for transaction replication

maximum 2Gb RAM  
  
**Standard Edition** can work on the Windows NT Server 4.0 with Service Pack 5, Windows NT Server 4.0 Enterprise Edition and on the Windows 2000 Server/Advanced Server/DataCenter.  
  
This edition has the following restrictions:

maximum 4 CPU (up to 8 CPU on the Windows NT Enterprise Edition)

no Distributed Partitioned Views

no Log Shipping

no Parallel index creation

no Failover clustering

maximum 2Gb RAM  
  
**Enterprise Edition** can work on the Windows NT Server 4.0 with Service Pack 5, Windows NT Server 4.0 Enterprise Edition and on the Windows 2000 Server/Advanced Server/DataCenter.  
  
This edition can use:

* up to 32 CPU on the Windows 2000 DataCenter up to 8 CPU on the Windows 2000 Advanced Server and on the Windows NT Server 4.0 Enterprise Edition up to 4 CPU on the Windows NT Server 4.0 and on the Windows 2000 Server
* up to 64Gb RAM on the Windows 2000 DataCenter up to 8 Gb RAM on the Windows 2000 Advanced Server up to 4 Gb RAM on the Windows 2000 Server up to 3 Gb RAM on the Windows NT Server 4.0 Enterprise Edition up to 2 Gb RAM on the Windows NT Server 4.0
* Distributed Partitioned Views
* Log Shipping
* Parallel index creation
* Failover clustering

The **Developer Edition** can be used by developers to create and debug stored procedures and triggers. This edition comes with its own compact disc and can be upgraded to SQL Server 2000 Enterprise Edition.  
  
The **Desktop Engine** has no graphical user interface and is related to the MSDE, not to the SQL Server 7.0 **Desktop Edition**. The size of **Desktop Engine** databases cannot exceed 2 GB. The **Desktop Engine** can use maximum 2 CPU.  
  
The **SQL Server CE** edition can work only on the Microsoft Windows CE, so it has all restrictions of this operation system (can use only 1 CPU, no Parallel index creation, no Full-Text Search and so on).  
  
The **Evaluation Edition** can be used only for the test purposes to learn more about the new features and enhancements and should be uninstalled after a 120-day evaluation period.  
  
See this article for more details:  
[**SQL Server 2000 Installation**](http://www.mssqlcity.com/Articles/Adm/SQL2000Inst.htm)

**How many SQL 7.0 editions exist and what is the difference between them?**

**Answer:**  
  
There are three SQL Server 7.0 editions:

Desktop Edition

Standard Edition

Enterprise Edition  
  
**Desktop Edition** can be installed on Windows 9x, Windows NT Server/Workstation 4.0 with Service Pack 4, Windows NT Server Enterprise Edition and on the all editions of Windows 2000.  
  
This edition has some restrictions:

maximum 2 CPU

no publishing for transaction replication

no SQL Server Failover Support

no Full-text search

maximum 2Gb RAM

maximum 4GB per database  
  
**Standard Edition** can be installed on Windows NT Server 4.0 with Service Pack 4, Windows NT Server Enterprise Edition and on the Windows 2000 Server/Advanced Server/Datacenter.  
  
This edition has the following restrictions:

maximum 4 CPU

no SQL Server Failover Support

maximum 2Gb RAM  
  
**Enterprise Edition** can be installed on the Windows NT Server Enterprise Edition and on the Windows 2000 Advanced Server/Datacenter.  
  
This edition can use up to 32 CPU and more than 2Gb RAM, it also supports Full-text search and SQL Server Failover Support.  
  
See this article for more details:  
[**SQL Server 7.0 Installation**](http://www.mssqlcity.com/Articles/Adm/SQL70Inst.htm)

**How is SQL Server licensed?**

**Answer:**  
  
MS SQL licenses:

Processor License

Per server Client Access License (CAL)

Per-Seat Client Access License (CAL)  
  
See this link for information about MS SQL licenses:  
[**How to Buy**](http://www.microsoft.com/sql/howtobuy/production.asp)  
  
It's therefrom:

| Processor License.  Requires a single license for each CPU in the machine running SQL Server.  This license includes unlimited client device access.    Server/Per-Seat Client Access License (CAL).  Requires a license for the machine running the Microsoft server product,  as well as Client Access Licenses (CALs) for each client device. A set  number of CALs are included with a server license and the server software. |
| --- |

See also this link:  
[**How can I add SQL Server licenses?**](http://www.mssqlcity.com/FAQ/General/AddCAL.htm)

**How do I migrate an Oracle database to SQL 7.0?**

**Answer:**  
  
Read this white paper from Microsoft about Oracle to SQL Server migration:  
[**Migrating Oracle Databases to Microsoft SQL Server 7.0 White Paper**](http://www.microsoft.com/sql/techinfo/deployment/70/oraclemigrate.asp)

**How do I migrate an Access database to SQL 7.0?**

**Answer:**  
  
See these articles from Microsoft about Access to SQL Server migration:  
[**Migrating Your Microsoft Access Database to SQL Server 7.0 White Paper**](http://www.microsoft.com/sql/techinfo/deployment/70/accessmigration.asp)  
[**Migrating Your Access Databases with the Upsizing Wizard**](http://www.microsoft.com/AccessDev/ProdInfo/AUT97dat.htm)

**How can I view the SQL Server transaction log?**

**Answer:**  
  
To view transaction log, you can query the **syslogs** system table for SQL Server 6.5, or run **DBCC LOG** statement for SQL Server 7.0/2000.  
  
See this article about how view transaction log:  
[**View Transaction Log**](http://www.mssqlcity.com/Articles/KnowHow/ViewLog.htm)

**How can I perform a nonlogged bulk copy?**

**Answer:**  
  
The following conditions should be met for nonlogged bulk copy:

The database option **select into/bulkcopy** is set to true using **sp\_dboption**.

The target table is not being replicated.

The TABLOCK hint is specified.

If the table has any indexes, then there are no existing rows in the table.

**I specified the column's description by using the SQL Server 2000 Enterprise Manager. How can I get this description from my stored procedure?**

**Answer:**  
  
The text you entered in the description edit box is implemented as a SQL Server 2000 extended property. To get its value you should use the system function **fn\_listextendedproperty**.  
See the **fn\_listextendedproperty** function in SQL Server 2000 Books Online.  
This is the example to get all extended properties for all columns in table1 table:

| SELECT \* FROM ::fn\_listextendedproperty (null, 'user', 'dbo', 'table',  'table1', 'column', default) |
| --- |

Check the 'value' column in the result set to get the column's description.

**How can I add SQL Server licenses?**

**Answer:**  
  
For SQL Server 7.0, you can use License Manager. This is the step by step description of adding licenses on Windows NT: Logon to the NT Server and run License Manager.  
(Start -> Programs -> Administrative Tools -> License Manager). Click on the Products View tab and choose "Microsoft SQL Server 7.0". Click Add licenses button to add more licenses.  
  
For SQL Server 2000 you can use SQL Server 2000 Licensing Setup utility. This is from SQL Server Books Online:

| You can add device or processor licenses after installing SQL Server,  using the SQL Server 2000 Licensing Setup utility in Control Panel.  (Do not confuse this licensing utility with the Windows Licensing utility,  also found in Control Panel.) |
| --- |

See also this link about how sql server licenses:  
[**How is SQL Server licensed?**](http://www.mssqlcity.com/FAQ/General/SQLCAL.htm)

**Could you give me the comparison of SQL 7.0 with SQL 2000?**

**Answer:**  
  
See this article about the comparison of SQL 7.0 with SQL 2000:  
[**The comparison of SQL Server 7.0 with SQL Server 2000**](http://www.mssqlcity.com/Articles/Compare/SQL70vs2000.htm)

**Could you give me the comparison of SQL Server with Access?**

**Answer:**  
  
See these articles about SQL Server vs Access:  
[**The comparison of SQL Server 2000 with Access 2000**](http://www.mssqlcity.com/Articles/Compare/sql_server_vs_access.htm)  
[**The comparison of Access 97/2000 with SQL Server 7.0/2000**](http://www.mssqlcity.com/Articles/Compare/SQLvsAccess.htm)

**Could you give me the comparison of MSDE with SQL Server?**

**Answer:**  
  
See this article about msde vs sql server:  
[**The comparison of MSDE 1.0 with SQL Server 7.0**](http://www.mssqlcity.com/Articles/Compare/SQLvsMSDE.htm)

**Can I set tempdb database to be in RAM in SQL 7.0/2000?**

**Answer:**  
  
No, you cannot set **tempdb** database to be in RAM in SQL Server 7.0/2000. This option is no longer supported.  
  
See also this link:  
[**The comparison of SQL Server 6.5 with SQL Server 7.0**](http://www.mssqlcity.com/Articles/Compare/SQL65vs70.htm)

# SQL Server FAQ

## SQL Server Troubleshooting FAQ

**How can I fix lost users after attaching database with sp\_attach\_db?**

**Answer:**  
  
You can use **sp\_change\_users\_login** system stored procedure with the Auto\_Fix action value.  
For more details, see **sp\_change\_users\_login** in SQL Server Books Online.

**How can I resolve the error 259?**

**Answer:**  
  
This is the error 259 message text:

| Ad hoc updates to system catalogs are not enabled. The system administrator  must reconfigure SQL Server to allow this. |
| --- |

You will get the error 259 when you tried to modify the system catalogs directly while the **allow updates** system configuration option is set to 0. To work around the error 259, you should set the **allow updates** option to 1 by using the **sp\_configure** system stored procedure.  
  
Check this link for more details:  
[**How can I edit the system tables manually?**](http://www.mssqlcity.com/FAQ/Devel/AmendSysTb.htm)  
**How can I resolve the error 605?**

**Answer:**  
  
It is the full error 605 message text:

| Attempt to fetch logical page %S\_PGID in database '%.\*ls'  belongs to object '%.\*ls', not to object '%.\*ls'. |
| --- |

The error 605 occurs when SQL Server detects database corruption. Read about "Error 605" in SQL Server Books Online. It's therefrom:

| Execute DBCC CHECKTABLE on the second object specified in the error message.  To determine the full extent of the corruption, execute DBCC CHECKDB as soon  as possible. Also check the error log for other errors, which often accompany  a 605 error.  If the 605 error is not transient, the problem is severe and you must run  DBCC CHECKDB with one of the repair clauses. If the error involves an index  page, use the REPAIR\_REBUILD clause. If the error involves a data page, it  may be necessary to use the REPAIR\_ALLOW\_DATA\_LOSS clause. In the likely event  that you cannot allow the loss of data, you will need to restore from a known  clean backup. If the problem persists, contact your primary support provider. |
| --- |

|  |
| --- |
|  |
|  |

**How can I resolve the error 1069?**

**Answer:**  
  
Error 1069 is a Windows NT/2000 error. The error 1069 indicates that the service cannot be started (starting a service returns "logon failure" error). For example, when you start MSSQLServer service, you can get the following error:  
  
*An error 1069 - (The service did not start due to a logon failure)*  
  
The error 1069 occurs if the service was started from an account that does not have the "Log on a service" right. To work around this problem, you should grant to this account the "Log on a service" right.  
  
See this article for more details:  
[**How to troubleshoot connection problems**](http://www.mssqlcity.com/Articles/Adm/connectivity_troubleshooting.htm)

**How can I resolve the error 80004005?**

**Answer:**  
  
Error 80004005 is a ADO/ODBC error. The error 80004005 indicates that you does not have appropriate permissions to make some work or the data you want to get cannot be available for some reason. If you cannot connect, check the user account you connected under. If you work through IIS, check the authentication properties.  
  
For example, if you get the following error 80004005 message:

| Microsoft OLE DB Provider for ODBC Drivers error '80004005'  Microsoft][ODBC Microsoft SQL Driver] Logon Failed() |
| --- |

then SQL Server does not accept or recognize the logon account and/or password you submitted.  
  
See this article for more details:  
[**How to troubleshoot connection problems**](http://www.mssqlcity.com/Articles/Adm/connectivity_troubleshooting.htm)

**Why DTS Import Wizard fails when importing table with a timestamp column?**

**Answer:**  
  
This is SQL Server 7.0 DTS Import Wizard bug. The SQL Server 7.0 DTS fails when you use the "Copy table(s) from the source database" option of the SQL Server 7.0 DTS Import Wizard to import table with a timestamp column.  
  
To work around the DTS fails problem, you can use the "Use a query to specify the data to transfer" option of the DTS Import Wizard and exclude the timestamp column; you can use "Transfer objects and data between SQL Server 7.0 databases" option of the DTS Import Wizard and specify which table you want to transfer; you can define the destination table to use binary(8) for the datatype for the column which will contain the timestamp.  
  
See also this link about DTS fails and troubleshooting DTS problems:  
[**How to troubleshoot DTS problems**](http://www.mssqlcity.com/Articles/Adm/dts_troubleshooting.htm)

**Why cannot I truncate the transaction log?**

**Answer:**  
  
You can truncate only nonactive portion of the transaction log. If your transaction log is very big and full, and you want to truncate transaction log, try the following:  
  
1. Make the full database backup.  
2. Set the **Truncate Log On Checkpoint** database option and then run **CHECKPOINT** command from the Query Analyzer.  
  
3. If the transaction log was not truncated, run the **DUMP TRANSACTION** command with **NO\_LOG** parameter.  
  
4. If the log was truncated, you can decrease the size of the log file by using the **DBCC SHRINKFILE** statement.  
  
If the transaction log was not truncated, and the database have only one data file, detach the database by using the **sp\_detach\_db** stored procedure, then attach only the data file by using the **sp\_attach\_single\_file\_db** stored procedure. The transaction log will be recreated automatically with the small size. This is the example to detach/attach the **Test** database:

| USE master  EXEC sp\_detach\_db 'Test', 'true'  EXEC sp\_attach\_single\_file\_db @dbname = 'Test',  @physname = 'c:\mssql7\data\Test\_Data.MDF' |
| --- |

You can also stop the MSSQLServer service, drop the transaction log file, start the MSSQLServer service. The transaction log will be recreated automatically with the small size. This is undocumented and not recommended way, so do not remember to make backup (see step 1).

**Why cannot I see the SQL performance counters in PerfMon?**

**Answer:**  
  
It's interesting problem. I checked sql counters on the work and everything was okay. But when I checked it on my home computer (Windows NT Workstation, MS SQL 6.5 and MS SQL 7.0) I found that native Windows NT perfmon counters were not shown, and MS SQL 7.0 SQL performance counters were not shown too.  
  
After that I tried to run this bat file:  
  
unlodctr.exe MSSQLServer  
C:\WINNT\System32\lodctr.exe "c:\mssql7\BINN\sqlctr.ini"  
  
but MS SQL 7.0 counters did not appeared.  
  
After that I checked this key in the registry  
  
*HKEY\_LOCAL\_MACHINE\Software\Microsoft\WindowsNT\CurrentVersion\Perflib*  
  
and saw that it included two subkeys: 009 and 019. 019 - it is my native regional settings, 009 - U.S. English. After that I deleted from c:\winnt\system32 (path for my Windows NT installation) two files: perfc019.dat and perfh019.dat. Now I can see native Windows NT counters, but MS SQL 7.0 counters were not still visible.  
  
After that I run this bat file again:  
  
unlodctr.exe MSSQLServer  
C:\WINNT\System32\lodctr.exe "c:\mssql7\BINN\sqlctr.ini"  
  
and now everything is okay.  
  
Check this link also:  
[**http://support.microsoft.com/support/kb/articles/Q196/6/57.ASP**](http://support.microsoft.com/support/kb/articles/Q196/6/57.ASP)

**The upgrade MSDE to SQL Server fails with the error about "old version".**

**Answer:**  
  
If you received the following error:

| *You cannot install a version which is older (7.00.623) than the*  *version on your machine (7.00.677). Uninstall the older version.* |
| --- |

then you should amend  
  
*HKEY\_LOCAL\_MACHINE/Software/Microsoft/MSSQLServer/MSSQLServer/CurrentVersion*  
to have a value of 7.00.623.  
  
For more details see:  
[**SQL Server 7.0 Installation problems**](http://www.mssqlcity.com/Articles/Adm/SQL70InstProbl.htm)

**My DTS package is stored inside SQL Server. Now I cannot open it.**

**Answer:**  
  
The DTS packages are stored in the **sysdtspackages** system table in the **msdb** database. Check the DTS package presents on the server:

| USE msdb  select name from sysdtspackages |
| --- |

The text of the DTS package is stored in the PackageData image field of the **sysdtspackages** system table in the **msdb** database.  
  
To save DTS package in file, you can use textcopy.exe utility from the C:\MSSQL7\BINN (path by default).  
  
This is the example to save DTSName package from the **sysdtspackages** table into DTSText.dts file:

| textcopy /S ServerName /U sa /P /D msdb /T sysdtspackages /C packagedata  /W "where name='DTSName'" /F DTSText.dts /O |
| --- |

Now you can open the DTSName package from the DTSText.dts file and run it.  
  
See also this link about troubleshooting DTS package errors:  
[**How to troubleshoot DTS problems**](http://www.mssqlcity.com/Articles/Adm/dts_troubleshooting.htm)

**My database has been marked suspect. How can I fix this?**

**Answer:**  
  
A database can be marked suspect for one of the following reasons (this is from SQL Server Books Online):

| If one or more database files are not available.  If the entire database is not available.  If one or more database files are corrupted.  If a database resource is being held by the operating system. |
| --- |

To resolve a suspect status, you can use the **sp\_resetstatus** stored procedure. This is the example to turn off the suspect status on the pubs database:  
  
*sp\_resetstatus pubs*  
  
Read about the **sp\_resetstatus** stored procedure in SQL Server Books Online.

**I got an error that RPC is not enabled. How can I enable RPC?**

**Answer:**  
  
You can enable RPC by using the **sp\_serveroption** system stored procedure. To enable RPC from the ProductSQL server, you can use the following:

| USE master  EXEC sp\_serveroption 'ProductSQL', 'rpc', 'true' |
| --- |

To enable RPC to the ProductSQL server, you can use the following:

| USE master  EXEC sp\_serveroption 'ProductSQL', 'rpc out', 'true' |
| --- |

**I get the full-text search timeout error. How can I fix this?**

**Answer:**  
  
If you encountered full-text search timeout error (error indicating that the full-text query is timed out), try to reduce the size of the result set, or increase the 'remote query timeout' setting, or insert the full-text query result set into a temporary table instead of streaming the results directly to the client.  
  
See also this link:  
[**How to troubleshoot full-text search problems**](http://www.mssqlcity.com/Articles/Adm/FTSProbl.htm)

**I get the full-text search timeout error. How can I fix this?**

**Answer:**  
  
If you encountered full-text search timeout error (error indicating that the full-text query is timed out), try to reduce the size of the result set, or increase the 'remote query timeout' setting, or insert the full-text query result set into a temporary table instead of streaming the results directly to the client.  
  
See also this link:  
[**How to troubleshoot full-text search problems**](http://www.mssqlcity.com/Articles/Adm/FTSProbl.htm)

**I get the full-text search timeout error. How can I fix this?**

**Answer:**  
  
If you encountered full-text search timeout error (error indicating that the full-text query is timed out), try to reduce the size of the result set, or increase the 'remote query timeout' setting, or insert the full-text query result set into a temporary table instead of streaming the results directly to the client.  
  
See also this link:  
[**How to troubleshoot full-text search problems**](http://www.mssqlcity.com/Articles/Adm/FTSProbl.htm)

**I can run DTS package myself, but cannot schedule it.**

**Answer:**  
  
It looks like you use LocalSystem account. This account has no network access. Try to use another account with the access to the network resources.  
  
When you tried to schedule DTS package, the appropriate job will run under the user account that the SQLServerAgent service is running under. For example, if SQLServerAgent service runs under the LocalSystem account, you cannot schedule DTS package, because the LocalSystem account does not have network access.  
  
So, to schedule DTS package, you should set the SQLServerAgent service to run under account, which has access to the network resources.  
  
**Note.** Before you schedule DTS package, the Schedule service must be started.  
  
To start schedule service on the Windows NT choose:  
Start -> Settings -> Control panel -> Services (choose Startup "Automatic" type) and Start.  
  
To start schedule service on the Windows 2000 choose:  
Start -> Settings -> Control panel -> Administration -> Services (choose Startup "Automatic" type) and Start.  
  
See also this link:  
[**How to troubleshoot DTS problems**](http://www.mssqlcity.com/Articles/Adm/dts_troubleshooting.htm)

**I am getting an error 1205 - what can I do?**

**Answer:**  
  
Error 1205 is a deadlock error.  
It is the full error 1205 message text:

| Your transaction (process ID #%d) was deadlocked with another process  and has been chosen as the deadlock victim. Rerun your transaction. |
| --- |

Your application should intercept the error 1205 and respond by resubmitting the transaction.  
  
See this question also:  
[**How do I resolve deadlocks?**](http://www.mssqlcity.com/FAQ/Trouble/DeadlockVic.htm)

**How to troubleshoot SQL Server installation problems?**

**Answer:**  
  
See these articles about SQL Server installation problems:  
[**SQL Server 7.0 Installation problems**](http://www.mssqlcity.com/Articles/Adm/SQL70InstProbl.htm)  
[**Troubleshooting SQL Server 2000 installation issues**](http://www.mssqlcity.com/Articles/Adm/installation_troubleshooting.htm)

**How do I resolve deadlocks?**

**Answer:**  
  
To minimize deadlock, you should (this is from SQL Server Books Online):

| Access objects in the same order.  Avoid user interaction in transactions.  Keep transactions short and in one batch.  Use as low an isolation level as possible.  Use bound connections. |
| --- |

Use trace flag 1204 to get more information about locking. You can use SQL Profiler to monitor and identify the cause of a deadlock.  
  
See also these links:  
[**SQL Server 6.5: Some useful trace flags**](http://www.mssqlcity.com/Articles/SQL65/SQL65TF.htm)  
[**SQL Server 7.0: Some useful trace flags**](http://www.mssqlcity.com/Articles/General/SQL70TF.htm)  
[**SQL Server 2000: Some useful trace flags**](http://www.mssqlcity.com/Articles/General/SQL2000TF.htm)

**How do I resolve deadlocks?**

**Answer:**  
  
To minimize deadlock, you should (this is from SQL Server Books Online):

| Access objects in the same order.  Avoid user interaction in transactions.  Keep transactions short and in one batch.  Use as low an isolation level as possible.  Use bound connections. |
| --- |

Use trace flag 1204 to get more information about locking. You can use SQL Profiler to monitor and identify the cause of a deadlock.  
  
See also these links:  
[**SQL Server 6.5: Some useful trace flags**](http://www.mssqlcity.com/Articles/SQL65/SQL65TF.htm)  
[**SQL Server 7.0: Some useful trace flags**](http://www.mssqlcity.com/Articles/General/SQL70TF.htm)  
[**SQL Server 2000: Some useful trace flags**](http://www.mssqlcity.com/Articles/General/SQL2000TF.htm)

**How can I resolve the error 8007000e?**

**Answer:**  
  
The error 8007000e indicates that not enough storage is available to complete the operation. In other words, the error 8007000e is a "out of memory" error. Release memory and try again.

**How can I resolve the error 80040e21?**

**Answer:**  
  
Error 80040e21 is an OLE DB error. The error 80040e21 indicates that the field is too small to accept the amount of data you attempted to add. For example, the error 80040e21 may be caused by trying to insert 12 characters into a SQL Server field that is formatted to accept only 10 characters. In this case, you should insert less data.

**SQL Server programming FAQ. http://www.tkdinesh.com/faq/ans/duplicate.html**

1. [**How to find and delete duplicate records from a table?**](http://www.tkdinesh.com/faq/ans/duplicate.html)
2. [**How to generate the user defined store procedures list in a database?**](http://www.tkdinesh.com/faq/ans/userstorprocs.html)
3. [**How to obtain the description of a column?**](http://www.tkdinesh.com/faq/ans/coldescrip.html)
4. [**How to schedule the scripting of database objects?**](http://www.tkdinesh.com/faq/ans/schscript.html)
5. [**How to programmatically find the installation directory of SQL Server?**](http://www.tkdinesh.com/faq/ans/instdir.html)
6. [**How to store images(JPEG,GIF...etc) in SQL server?**](http://www.tkdinesh.com/faq/ans/storeimages.html)
7. [**How to check if a temp table exists already?**](http://www.tkdinesh.com/faq/ans/temptable.html)
8. [**How to find the number of rows affected by bcp?**](http://www.tkdinesh.com/faq/ans/bcprows.html)
9. [**How to add a default constraint to an existing column?**](http://www.tkdinesh.com/faq/ans/adddef.html)
10. [**Records in one table but not in another?**](http://www.tkdinesh.com/faq/ans/missingrecords.html)
11. [**How to fire triggers in SQL 7 when using bcp?**](http://www.tkdinesh.com/faq/ans/bcptrigg.html)
12. [**How to break a column to two using SUBSTRING()?**](http://www.tkdinesh.com/faq/ans/breakup.html)
13. [**How to define constraints across databases?**](http://www.tkdinesh.com/faq/ans/acrossdbs.html)
14. [**How to write a script that enumerates all user tables in a SQL Server database and then set the primary key in each table?**](http://www.tkdinesh.com/faq/ans/powerofsql.html)
15. [**How to prefix zeroes to a integer?**](http://www.tkdinesh.com/faq/ans/prefixzeros.html)
16. [**How to specify 'order by' on specific columns in a view.?**](http://www.tkdinesh.com/faq/ans/viewoderby.html)
17. [**How to pass an array to a stored procedure?**](http://www.tkdinesh.com/faq/ans/passarray.html)
18. [**How to get the current date without the time?**](http://www.tkdinesh.com/faq/ans/filtertime.html)
19. [**How to find the execution time of a stored procedure?**](http://www.tkdinesh.com/faq/ans/exectime.html)
20. [**How to write a alphanumeric query?**](http://www.tkdinesh.com/faq/ans/alphanum.html)
21. [**How to create a system stored procedure that has type set to SYSTEM ?**](http://www.tkdinesh.com/faq/ans/sysstorproc.html)
22. [**How to find out the columns having index?**](http://www.tkdinesh.com/faq/ans/indexcolumns.html)
23. [**How to do a SELECT inside a CASE expression?**](http://www.tkdinesh.com/faq/ans/selctincase.html)
24. [**How to to find the last product purchased from each vendor?**](http://www.tkdinesh.com/faq/ans/usemax.html)
25. [**How to sort data inside a string?**](http://www.tkdinesh.com/faq/ans/sortdata.html)
26. [**How to use PATINDEX to compare data?**](http://www.tkdinesh.com/faq/ans/patindx.html)
27. [**How to catch a OUTPUT parameter?**](http://www.tkdinesh.com/faq/ans/paramout.html)
28. [**How to round up a datetime value to the nearest hour?**](http://www.tkdinesh.com/faq/ans/rounddate.html)
29. [**How to format a datetime value like " dd-mon-yy "?**](http://www.tkdinesh.com/faq/ans/datefmt.html)
30. [**Which one is faster -- IN or OR?**](http://www.tkdinesh.com/faq/ans/INorOR.html)
31. [**How to manually insert a IDENTITY value?**](http://www.tkdinesh.com/faq/ans/identinsert.html)
32. [**How to display count of rows as different set of columns?**](http://www.tkdinesh.com/faq/ans/cntrowsascols.html)
33. [**How to use the different DATE and TIME functions?**](http://www.tkdinesh.com/faq/ans/datefunc.html)
34. [**How and when to use SELF JOIN?**](http://www.tkdinesh.com/faq/ans/selfjoinuse.html)
35. [**How to generate a rownumber for the query?**](http://www.tkdinesh.com/faq/ans/rownumber.html)
36. [**How to insert carriage return?**](http://www.tkdinesh.com/faq/ans/carrret.html)
37. [**How to use dynamic sql for a nvarchar of more than 4000 characters?**](http://www.tkdinesh.com/faq/ans/concatfords.html)
38. [**How to avoid the appearance of 'n rows affected'?**](http://www.tkdinesh.com/faq/ans/nocounton.html)
39. [**How to convert a number, stored in seconds, to hours:minutes?**](http://www.tkdinesh.com/faq/ans/sectodateformat.html)
40. [**How to reset/clear a IDENTITY value?**](http://www.tkdinesh.com/faq/ans/resetidentity.html)
41. [**SET or SELECT -- is there any performance difference?**](http://www.tkdinesh.com/faq/ans/setorselect.html)
42. [**How to SELECT nth row from a table?**](http://www.tkdinesh.com/faq/ans/selnthrow.html)
43. [**Is there a performance difference between CHAR and VARCHAR?**](http://www.tkdinesh.com/faq/ans/charorvarchar.html)
44. [**How to SELECT combined DISTINCT values from two tables?**](http://www.tkdinesh.com/faq/ans/seldistinct.html)
45. [**How to use CASE for ORDER BY clause?**](http://www.tkdinesh.com/faq/ans/orderbycase.html)

| Database design http://vyaskn.tripod.com/iq.htm#top#top | **(top)** |
| --- | --- |

| 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.

**Back to top**

| SQL Server architecture | **(top)** |
| --- | --- |

| 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"

| Whar 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.

**Back to top**

| Database administration | **(top)** |
| --- | --- |

| 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**](http://www.raid-advisory.com/)

| 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**](http://www.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**](http://vyaskn.tripod.com/sql_server_security_best_practices.htm)

| 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 replicaion? 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.**](http://vyaskn.tripod.com/sqlsps.htm)

**Back to top**

| Database programming | **(top)** |
| --- | --- |

| 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**](http://vyaskn.tripod.com/code.htm)' section of my site or [**search**](http://vyaskn.tripod.com/search.htm) 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**](http://vyaskn.tripod.com/code.htm)' 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

| Given an employee table, how would you find out the second highest salary? |
| --- |

* 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 DEFAULT 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. Read Committed - A transaction operating at the Read Committed level cannot see changes made by other transactions until those transactions are committed. At this level of isolation, dirty reads are not possible but nonrepeatable reads and phantoms are possible. Read Uncommitted - A transaction operating at the Read Uncommitted level can see uncommitted changes made by other transactions. At this level of isolation, dirty reads, nonrepeatable reads, and phantoms are all possible. Repeatable Read - A transaction operating at the Repeatable Read level is guaranteed not to see any changes made by other transactions in values it has already read. At this level of isolation, dirty reads and nonrepeatable reads are not possible but phantoms are possible. Serializable - A transaction operating at the Serializable level guarantees that all concurrent transactions interact only in ways that produce the same effect as if each transaction were entirely executed one after the other. At this isolation level, dirty reads, nonrepeatable reads, and phantoms are not possible.
* 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’. 1024 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. TRUNCATE TABLE is functionally identical to DELETE statement with no WHERE clause: both remove all rows in the table. But TRUNCATE TABLE is faster and uses fewer system and transaction log resources than DELETE. The DELETE statement removes rows one at a time and records an entry in the transaction log for each deleted row. TRUNCATE TABLE removes the data by deallocating the data pages used to store the table’s data, and only the page deallocations are recorded in the transaction log. TRUNCATE TABLE removes all rows from a table, but the table structure and its columns, constraints, indexes and so on remain. The counter used by an identity for new rows is reset to the seed for the column. If you want to retain the identity counter, use DELETE instead. If you want to remove table definition and its data, use the DROP TABLE statement. You cannot use TRUNCATE TABLE on a table referenced by a FOREIGN KEY constraint; instead, use DELETE statement without a WHERE clause. Because TRUNCATE TABLE is not logged, it cannot activate a trigger. TRUNCATE TABLE may not be used on tables participating in an indexed view
* 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 create 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 creating 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 configuration 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.
* Explain 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 processing 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 roundtrip, however large the resultset is. Cursors are also costly because they require more resources and temporary storage (results in more IO operations). Further, 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

**Transact-SQL Optimization Tips**

* **Use views and stored procedures instead of heavy-duty queries.**  
  This can reduce network traffic, because your client will send to server only stored procedure or view name (perhaps with some parameters) instead of large heavy-duty queries text. This can be used to facilitate permission management also, because you can restrict user access to table columns they should not see.
* **Try to use constraints instead of triggers, whenever possible.**  
  Constraints are much more efficient than triggers and can boost performance. So, you should use constraints instead of triggers, whenever possible.
* **Use table variables instead of temporary tables.**  
  Table variables require less locking and logging resources than temporary tables, so table variables should be used whenever possible. The table variables are available in SQL Server 2000 only.
* **Try to use UNION ALL statement instead of UNION, whenever possible.**  
  The UNION ALL statement is much faster than UNION, because UNION ALL statement does not look for duplicate rows, and UNION statement does look for duplicate rows, whether or not they exist.
* **Try to avoid using the DISTINCT clause, whenever possible.**  
  Because using the DISTINCT clause will result in some performance degradation, you should use this clause only when it is necessary.
* **Try to avoid using SQL Server cursors, whenever possible.**  
  SQL Server cursors can result in some performance degradation in comparison with select statements. Try to use correlated sub-query or derived tables, if you need to perform row-by-row operations.
* **Try to avoid the HAVING clause, whenever possible.**  
  The HAVING clause is used to restrict the result set returned by the GROUP BY clause. When you use GROUP BY with the HAVING clause, the GROUP BY clause divides the rows into sets of grouped rows and aggregates their values, and then the HAVING clause eliminates undesired aggregated groups. In many cases, you can write your select statement so, that it will contain only WHERE and GROUP BY clauses without HAVING clause. This can improve the performance of your query.
* **If you need to return the total table's row count, you can use alternative way instead of SELECT COUNT(\*) statement.**  
  Because SELECT COUNT(\*) statement make a full table scan to return the total table's row count, it can take very many time for the large table. There is another way to determine the total row count in a table. You can use sysindexes system table, in this case. There is ROWS column in the sysindexes table. This column contains the total row count for each table in your database. So, you can use the following select statement instead of SELECT COUNT(\*): SELECT rows FROM sysindexes WHERE id = OBJECT\_ID('table\_name') AND indid < 2 So, you can improve the speed of such queries in several times.
* **Include SET NOCOUNT ON statement into your stored procedures to stop the message indicating the number of rows affected by a T-SQL statement.**  
  This can reduce network traffic, because your client will not receive the message indicating the number of rows affected by a T-SQL statement.
* **Try to restrict the queries result set by using the WHERE clause.**  
  This can results in good performance benefits, because SQL Server will return to client only particular rows, not all rows from the table(s). This can reduce network traffic and boost the overall performance of the query.
* **Use the select statements with TOP keyword or the SET ROWCOUNT statement, if you need to return only the first n rows.**  
  This can improve performance of your queries, because the smaller result set will be returned. This can also reduce the traffic between the server and the clients.
* **Try to restrict the queries result set by returning only the particular columns from the table, not all table's columns.**  
  This can results in good performance benefits, because SQL Server will return to client only particular columns, not all table's columns. This can reduce network traffic and boost the overall performance of the query.

1.Indexes  
2.avoid more number of triggers on the table  
3.unnecessary complicated joins  
4.correct use of Group by clause with the select list  
5.in worst cases Denormalization  
  
**Index Optimization tips**

* Every index increases the time in takes to perform INSERTS, UPDATES and DELETES, so the number of indexes should not be very much. Try to use maximum 4-5 indexes on one table, not more. If you have read-only table, then the number of indexes may be increased.
* Keep your indexes as narrow as possible. This reduces the size of the index and reduces the number of reads required to read the index.
* Try to create indexes on columns that have integer values rather than character values.
* If you create a composite (multi-column) index, the order of the columns in the key are very important. Try to order the columns in the key as to enhance selectivity, with the most selective columns to the leftmost of the key.
* If you want to join several tables, try to create surrogate integer keys for this purpose and create indexes on their columns.
* Create surrogate integer primary key (identity for example) if your table will not have many insert operations.
* Clustered indexes are more preferable than nonclustered, if you need to select by a range of values or you need to sort results set with GROUP BY or ORDER BY.
* If your application will be performing the same query over and over on the same table, consider creating a covering index on the table.
* You can use the SQL Server Profiler Create Trace Wizard with "Identify Scans of Large Tables" trace to determine which tables in your database may need indexes. This trace will show which tables are being scanned by queries instead of using an index.
* You can use sp\_MSforeachtable undocumented stored procedure to rebuild all indexes in your database. Try to schedule it to execute during CPU idle time and slow production periods.  
  sp\_MSforeachtable @command1="print '?' DBCC DBREINDEX ('?')"

**T-SQL Queries**

1. 2 tables

**Employee**

empid  
empname  
salary  
mgrid

**Phone**

empid

phnumber

1. Select all employees who doesn't have phone?  
   SELECT empname  
   FROM Employee  
   WHERE (empid NOT IN  
   (SELECT DISTINCT empid  
   FROM phone))
2. Select the employee names who is having more than one phone numbers.  
   SELECT empname  
   FROM employee  
   WHERE (empid IN  
   (SELECT empid  
   FROM phone  
   GROUP BY empid  
   HAVING COUNT(empid) > 1))
3. Select the details of 3 max salaried employees from employee table.  
   SELECT TOP 3 empid, salary  
   FROM employee  
   ORDER BY salary DESC
4. Display all managers from the table. (manager id is same as emp id)  
   SELECT empname  
   FROM employee  
   WHERE (empid IN  
   (SELECT DISTINCT mgrid  
   FROM employee))
5. Write a Select statement to list the Employee Name, Manager Name under a particular manager?  
   SELECT e1.empname AS EmpName, e2.empname AS ManagerName  
   FROM Employee e1 INNER JOIN  
   Employee e2 ON e1.mgrid = e2.empid  
   ORDER BY e2.mgrid
6. 2 tables emp and phone.  
   emp fields are - empid, name  
   Ph fields are - empid, ph (office, mobile, home). Select all employees who doesn't have any ph nos.  
   SELECT \*  
   FROM employee LEFT OUTER JOIN  
   phone ON employee.empid = phone.empid  
   WHERE (phone.office IS NULL OR phone.office = ' ')  
   AND (phone.mobile IS NULL OR phone.mobile = ' ')  
   AND (phone.home IS NULL OR phone.home = ' ')
7. Find employee who is living in more than one city.   
   Two Tables:

| **Emp** | **City** |
| --- | --- |
| Empid empName Salary | Empid City |

1. SELECT empname, fname, lname  
   FROM employee  
   WHERE (empid IN  
   (SELECT empid  
   FROM city  
   GROUP BY empid  
   HAVING COUNT(empid) > 1))
2. Find all employees who is living in the same city. (table is same as above)  
   SELECT fname  
   FROM employee  
   WHERE (empid IN  
   (SELECT empid  
   FROM city a  
   WHERE city IN  
   (SELECT city  
   FROM city b  
   GROUP BY city  
   HAVING COUNT(city) > 1)))
3. There is a table named MovieTable with three columns - moviename, person and role. Write a query which gets the movie details where Mr. Amitabh and Mr. Vinod acted and their role is actor.  
   SELECT DISTINCT m1.moviename  
   FROM MovieTable m1 INNER JOIN  
   MovieTable m2 ON m1.moviename = m2.moviename  
   WHERE (m1.person = 'amitabh' AND m2.person = 'vinod' OR  
   m2.person = 'amitabh' AND m1.person = 'vinod') AND (m1.role = 'actor') AND (m2.role = 'actor')  
   ORDER BY m1.moviename
4. There are two employee tables named emp1 and emp2. Both contains same structure (salary details). But Emp2 salary details are incorrect and emp1 salary details are correct. So, write a query which corrects salary details of the table emp2  
   update a set a.sal=b.sal from emp1 a, emp2 b where a.empid=b.empid
5. Given a Table named “Students” which contains studentid, subjectid and marks. Where there are 10 subjects and 50 students. Write a Query to find out the Maximum marks obtained in each subject.
6. In this same tables now write a SQL Query to get the studentid also to combine with previous results.
7. Three tables – student , course, marks – how do go at finding name of the students who got max marks in the diff courses.  
   SELECT student.name, course.name AS coursename, marks.sid, marks.mark  
   FROM marks INNER JOIN  
   student ON marks.sid = student.sid INNER JOIN  
   course ON marks.cid = course.cid  
   WHERE (marks.mark =  
   (SELECT MAX(Mark)  
   FROM Marks MaxMark  
   WHERE MaxMark.cID = Marks.cID))
8. There is a table day\_temp which has three columns dayid, day and temperature. How do I write a query to get the difference of temperature among each other for seven days of a week?  
   SELECT a.dayid, a.dday, a.tempe, a.tempe - b.tempe AS Difference  
   FROM day\_temp a INNER JOIN  
   day\_temp b ON a.dayid = b.dayid + 1  
   OR  
   Select a.day, a.degree-b.degree from temperature a, temperature b where a.id=b.id+1
9. There is a table which contains the names like this. a1, a2, a3, a3, a4, a1, a1, a2 and their salaries. Write a query to get grand total salary, and total salaries of individual employees in one query.  
   SELECT empid, SUM(salary) AS salary  
   FROM employee  
   GROUP BY empid WITH ROLLUP  
   ORDER BY empid
10. **How to know how many tables contains empno as a column in a database?**SELECT COUNT(\*) AS Counter  
    FROM syscolumns  
    WHERE (name = 'empno')
11. **Find duplicate rows in a table? OR I have a table with one column which has many records which are not distinct. I need to find the distinct values from that column and number of times it’s repeated.**  
    SELECT sid, mark, COUNT(\*) AS Counter  
    FROM marks  
    GROUP BY sid, mark  
    HAVING (COUNT(\*) > 1)
12. **How to delete the rows which are duplicate (don’t delete both duplicate records).**  
    SET ROWCOUNT 1  
    DELETE yourtable  
    FROM yourtable a  
    WHERE (SELECT COUNT(\*) FROM yourtable b WHERE b.name1 = a.name1 AND b.age1 = a.age1) > 1  
    WHILE @@rowcount > 0  
      DELETE yourtable  
      FROM yourtable a  
      WHERE (SELECT COUNT(\*) FROM yourtable b WHERE b.name1 = a.name1 AND b.age1 = a.age1) > 1  
    SET ROWCOUNT 0
13. **How to find 6th highest salary**SELECT TOP 1 salary  
    FROM (SELECT DISTINCT TOP 6 salary  
    FROM employee  
    ORDER BY salary DESC) a  
    ORDER BY salary
14. **Find top salary among two tables**SELECT TOP 1 sal  
    FROM (SELECT MAX(sal) AS sal  
    FROM sal1  
    UNION  
    SELECT MAX(sal) AS sal  
    FROM sal2) a  
    ORDER BY sal DESC
15. **Write a query to convert all the letters in a word to upper case**SELECT UPPER('test')
16. **Write a query to round up the values of a number. For example even if the user enters 7.1 it should be rounded up to 8.**SELECT CEILING (7.1)
17. **Write a SQL Query to find first day of month?**SELECT DATENAME(dw, DATEADD(dd, - DATEPART(dd, GETDATE()) + 1, GETDATE())) AS FirstDay

**Datepart Abbreviations**

year yy, yyyy

quarter qq, q

month mm, m

dayofyear dy, y

day dd, d

week wk, ww

weekday dw

hour hh

minute mi, n

second ss, s

millisecond ms

1. Table A contains column1 which is primary key and has 2 values (1, 2) and Table B contains column1 which is primary key and has 2 values (2, 3). Write a query which returns the values that are not common for the tables and the query should return one column with 2 records.  
   SELECT tbla.a  
   FROM tbla, tblb  
   WHERE tbla.a <>  
   (SELECT tblb.a  
   FROM tbla, tblb  
   WHERE tbla.a = tblb.a)  
   UNION  
   SELECT tblb.a  
   FROM tbla, tblb  
   WHERE tblb.a <>  
   (SELECT tbla.a  
   FROM tbla, tblb  
   WHERE tbla.a = tblb.a)  
     
   OR (better approach)  
     
   SELECT a  
   FROM tbla  
   WHERE a NOT IN  
   (SELECT a  
   FROM tblb)  
   UNION ALL  
   SELECT a  
   FROM tblb  
   WHERE a NOT IN  
   (SELECT a  
   FROM tbla)
2. There are 3 tables Titles, Authors and Title-Authors (check PUBS db). Write the query to get the author name and the number of books written by that author, the result should start from the author who has written the maximum number of books and end with the author who has written the minimum number of books.  
   SELECT authors.au\_lname, COUNT(\*) AS BooksCount  
   FROM authors INNER JOIN  
   titleauthor ON authors.au\_id = titleauthor.au\_id INNER JOIN  
   titles ON titles.title\_id = titleauthor.title\_id  
   GROUP BY authors.au\_lname  
   ORDER BY BooksCount DESC
3. UPDATE emp\_master  
   SET emp\_sal =   
   CASE  
   WHEN emp\_sal > 0 AND emp\_sal <= 20000 THEN (emp\_sal \* 1.01)   
   WHEN emp\_sal > 20000 THEN (emp\_sal \* 1.02)   
   END
4. List all products with total quantity ordered, if quantity ordered is null show it as 0.  
   SELECT name, CASE WHEN SUM(qty) IS NULL THEN 0 WHEN SUM(qty) > 0 THEN SUM(qty) END AS tot  
   FROM [order] RIGHT OUTER JOIN  
   product ON [order].prodid = product.prodid  
   GROUP BY name  
   Result:  
   coke 60  
   mirinda 0  
   pepsi 10
5. ANY, SOME, or ALL?  
   ALL means greater than every value--in other words, greater than the maximum value. For example, >ALL (1, 2, 3) means greater than 3.  
   ANY means greater than at least one value, that is, greater than the minimum. So >ANY (1, 2, 3) means greater than 1. SOME is an SQL-92 standard equivalent for ANY.
6. IN & = (difference in correlated sub query)  
     
   **INDEX**
7. **What is Index? It’s purpose?**Indexes in databases are similar to indexes in books. In a database, an index allows the database program to find data in a table without scanning the entire table. An index in a database is a list of values in a table with the storage locations of rows in the table that contain each value. Indexes can be created on either a single column or a combination of columns in a table and are implemented in the form of B-trees. An index contains an entry with one or more columns (the search key) from each row in a table. A B-tree is sorted on the search key, and can be searched efficiently on any leading subset of the search key. For example, an index on columns **A**, **B**, **C** can be searched efficiently on **A**, on **A**, **B**, and **A**, **B**, **C**.
8. **Explain about Clustered and non clustered index? How to choose between a Clustered Index and a Non-Clustered Index?**There are clustered and nonclustered indexes. A clustered index is a special type of index that reorders the way records in the table are physically stored. Therefore table can have only one clustered index. The leaf nodes of a clustered index contain the data pages.   
   A nonclustered index is a special type of index in which the logical order of the index does not match the physical stored order of the rows on disk. The leaf nodes of a nonclustered index does not consist of the data pages. Instead, the leaf nodes contain index rows.  
   Consider using a clustered index for:
   * Columns that contain a large number of distinct values.
   * Queries that return a range of values using operators such as BETWEEN, >, >=, <, and <=.
   * Columns that are accessed sequentially.
   * Queries that return large result sets.  
     Non-clustered indexes have the same B-tree structure as clustered indexes, with two significant differences:
   * The data rows are not sorted and stored in order based on their non-clustered keys.
   * The leaf layer of a non-clustered index does not consist of the data pages. Instead, the leaf nodes contain index rows. Each index row contains the non-clustered key value and one or more row locators that point to the data row (or rows if the index is not unique) having the key value.
   * Per table only 249 non clustered indexes.
9. **Disadvantage of index?**Every index increases the time in takes to perform INSERTS, UPDATES and DELETES, so the number of indexes should not be very much.
10. **Given a scenario that I have a 10 Clustered Index in a Table to all their 10 Columns. What are the advantages and disadvantages?**A: Only 1 clustered index is possible.
11. **How can I enforce to use particular index?**You can use index hint (index=<index\_name>) after the table name.  
    SELECT au\_lname FROM authors (index=aunmind)
12. **What is Index Tuning?**One of the hardest tasks facing database administrators is the selection of appropriate columns for non-clustered indexes. You should consider creating non-clustered indexes on any columns that are frequently referenced in the WHERE clauses of SQL statements. Other good candidates are columns referenced by JOIN and GROUP BY operations.  
    You may wish to also consider creating non-clustered indexes that cover all of the columns used by certain frequently issued queries. These queries are referred to as “covered queries” and experience excellent performance gains.  
    Index Tuning is the process of finding appropriate column for non-clustered indexes.  
    SQL Server provides a wonderful facility known as the Index Tuning Wizard which greatly enhances the index selection process.
13. **Difference between Index defrag and Index rebuild?**When you create an index in the database, the index information used by queries is stored in index pages. The sequential index pages are chained together by pointers from one page to the next. When changes are made to the data that affect the index, the information in the index can become scattered in the database. Rebuilding an index reorganizes the storage of the index data (and table data in the case of a clustered index) to remove fragmentation. This can improve disk performance by reducing the number of page reads required to obtain the requested data  
    DBCC INDEXDEFRAG - Defragments clustered and secondary indexes of the specified table or view.  
    \*\*
14. **What is sorting and what is the difference between sorting & clustered indexes?**The ORDER BY clause sorts query results by one or more columns up to 8,060 bytes. This will happen by the time when we retrieve data from database. Clustered indexes physically sorting data, while inserting/updating the table.
15. **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
16. **What is fillfactor? What is the use of it ? What happens when we ignore it? When you should use low fill factor?**When you create a clustered index, the data in the table is stored in the data pages of the database according to the order of the values in the indexed columns. When new rows of data are inserted into the table or the values in the indexed columns are changed, Microsoft® SQL Server™ 2000 may have to reorganize the storage of the data in the table to make room for the new row and maintain the ordered storage of the data. This also applies to nonclustered indexes. When data is added or changed, SQL Server may have to reorganize the storage of the data in the nonclustered index pages. When a new row is added to a full index page, SQL Server moves approximately half the rows to a new page to make room for the new row. This reorganization is known as a page split. Page splitting can impair performance and fragment the storage of the data in a table.When creating an index, you can specify a fill factor to leave extra gaps and reserve a percentage of free space on each leaf level page of the index to accommodate future expansion in the storage of the table's data and reduce the potential for page splits. The fill factor value is a percentage from 0 to 100 that specifies how much to fill the data pages after the index is created. A value of 100 means the pages will be full and will take the least amount of storage space. This setting should be used only when there will be no changes to the data, for example, on a read-only table. A lower value leaves more empty space on the data pages, which reduces the need to split data pages as indexes grow but requires more storage space. This setting is more appropriate when there will be changes to the data in the table.  
    **DATA TYPES**
17. **What are the data types in SQL**

bigint

Binary

bit

char

cursor

datetime

Decimal

float

image

int

money

Nchar

ntext

nvarchar

real

smalldatetime

Smallint

smallmoney

text

timestamp

tinyint

Varbinary

Varchar

uniqueidentifier

1. **Difference between char and nvarchar / char and varchar data-type?**  
   char[(n)] - Fixed-length non-Unicode character data with length of n bytes. n must be a value from 1 through 8,000. Storage size is n bytes. The SQL-92 synonym for char is character.  
   nvarchar(n) - Variable-length Unicode character data of n characters. n must be a value from 1 through 4,000. Storage size, in bytes, is two times the number of characters entered. The data entered can be 0 characters in length. The SQL-92 synonyms for nvarchar are national char varying and national character varying.   
   varchar[(n)] - Variable-length non-Unicode character data with length of n bytes. n must be a value from 1 through 8,000. Storage size is the actual length in bytes of the data entered, not n bytes. The data entered can be 0 characters in length. The SQL-92 synonyms for varchar are char varying or character varying.
2. **GUID datasize?**128bit
3. **How GUID becoming unique across machines?**To ensure uniqueness across machines, the ID of the network card is used (among others) to compute the number.
4. **What is the difference between text and image data type?**Text and image. Use text for character data if you need to store more than 255 characters in SQL Server 6.5, or more than 8000 in SQL Server 7.0. Use image for binary large objects (BLOBs) such as digital images. With text and image data types, the data is not stored in the row, so the limit of the page size does not apply.All that is stored in the row is a pointer to the database pages that contain the data.Individual text, ntext, and image values can be a maximum of 2-GB, which is too long to store in a single data row.  
     
   **JOINS**
5. **What are joins?**Sometimes we have to select data from two or more tables to make our result complete. We have to perform a join.
6. **How many types of Joins?**Joins can be categorized as:
   * Inner joins (the typical join operation, which uses some comparison operator like = or <>). These include equi-joins and natural joins.   
     Inner joins use a comparison operator to match rows from two tables based on the values in common columns from each table. For example, retrieving all rows where the student identification number is the same in both the **students** and **courses** tables.
   * Outer joins. Outer joins can be a left, a right, or full outer join.   
     Outer joins are specified with one of the following sets of keywords when they are specified in the FROM clause:
     + LEFT JOIN or LEFT OUTER JOIN -The result set of a left outer join includes all the rows from the left table specified in the LEFT OUTER clause, not just the ones in which the joined columns match. When a row in the left table has no matching rows in the right table, the associated result set row contains null values for all select list columns coming from the right table.
     + RIGHT JOIN or RIGHT OUTER JOIN - A right outer join is the reverse of a left outer join. All rows from the right table are returned. Null values are returned for the left table any time a right table row has no matching row in the left table.
     + FULL JOIN or FULL OUTER JOIN - A full outer join returns all rows in both the left and right tables. Any time a row has no match in the other table, the select list columns from the other table contain null values. When there is a match between the tables, the entire result set row contains data values from the base tables.
   * Cross joins - Cross joins return all rows from the left table, each row from the left table is combined with all rows from the right table. Cross joins are also called **Cartesian products.** (A Cartesian join will get you a Cartesian product. A Cartesian join is when you join every row of one table to every row of another table. You can also get one by joining every row of a table to every row of itself.)
7. **What is self join?**A table can be joined to itself in a self-join.
8. **What are the differences between UNION and JOINS?**A join selects columns from 2 or more tables. A union selects rows.
9. **Can I improve performance by using the ANSI-style joins instead of the old-style joins?**Code Example 1:  
   select o.name, i.name  
   from sysobjects o, sysindexes i  
   where o.id = i.id  
   Code Example 2:  
   select o.name, i.name  
   from sysobjects o inner join sysindexes i  
   on o.id = i.id  
   You will not get any performance gain by switching to the ANSI-style JOIN syntax.  
   Using the ANSI-JOIN syntax gives you an important advantage: Because the join logic is cleanly separated from the filtering criteria, you can understand the query logic more quickly.  
   The SQL Server old-style JOIN executes the filtering conditions before executing the joins, whereas the ANSI-style JOIN reverses this procedure (join logic precedes filtering).  
   Perhaps the most compelling argument for switching to the ANSI-style JOIN is that Microsoft has explicitly stated that SQL Server will not support the old-style OUTER JOIN syntax indefinitely. Another important consideration is that the ANSI-style JOIN supports query constructions that the old-style JOIN syntax does not support.
10. **What is derived table?**Derived tables are SELECT statements in the FROM clause referred to by an alias or a user-specified name. The result set of the SELECT in the FROM clause forms a table used by the outer SELECT statement. For example, this SELECT uses a derived table to find if any store carries all book titles in the **pubs** database:   
    SELECT ST.stor\_id, ST.stor\_name  
    FROM stores AS ST,  
         (SELECT stor\_id, COUNT(DISTINCT title\_id) AS title\_count  
          FROM sales  
          GROUP BY stor\_id  
         ) AS SA  
    WHERE ST.stor\_id = SA.stor\_id  
    AND SA.title\_count = (SELECT COUNT(\*) FROM titles)  
    **STORED PROCEDURE**
11. **What is Stored procedure?**A stored procedure is a set of Structured Query Language (SQL) statements that you assign a name to and store in a database in compiled form so that you can share it between a number of programs.
    * They allow modular programming.
    * They allow faster execution.
    * They can reduce network traffic.
    * They can be used as a security mechanism.
12. **What are the different types of Storage Procedure?**
    1. Temporary Stored Procedures - SQL Server supports two types of temporary procedures: local and global. A local temporary procedure is visible only to the connection that created it. A global temporary procedure is available to all connections. Local temporary procedures are automatically dropped at the end of the current session. Global temporary procedures are dropped at the end of the last session using the procedure. Usually, this is when the session that created the procedure ends. Temporary procedures named with # and ## can be created by any user.
    2. System stored procedures are created and stored in the **master** database and have the **sp\_** prefix.(or xp\_) System stored procedures can be executed from any database without having to qualify the stored procedure name fully using the database name **master**. (If any user-created stored procedure has the same name as a system stored procedure, the user-created stored procedure will never be executed.)
    3. Automatically Executing Stored Procedures - One or more stored procedures can execute automatically when SQL Server starts. The stored procedures must be created by the system administrator and executed under the **sysadmin** fixed server role as a background process. The procedure(s) cannot have any input parameters.
    4. User stored procedure
13. **How do I mark the stored procedure to automatic execution?**You can use the sp\_procoption system stored procedure to mark the stored procedure to automatic execution when the SQL Server will start. Only objects in the master database owned by dbo can have the startup setting changed and this option is restricted to objects that have no parameters.  
    USE master  
    EXEC sp\_procoption 'indRebuild', 'startup', 'true')
14. **How can you optimize a stored procedure?**
15. **How will know whether the SQL statements are executed?**When used in a stored procedure, the RETURN statement can specify an integer value to return to the calling application, batch, or procedure. If no value is specified on RETURN, a stored procedure returns the value 0.  The stored procedures return a value of 0 when no errors were encountered. Any nonzero value indicates an error occurred.
16. **Why one should not prefix user stored procedures with sp\_?**It is strongly recommended that you do not create any stored procedures using sp\_ as a prefix. SQL Server always looks for a stored procedure beginning with sp\_ in this order:
    1. The stored procedure in the master database.
    2. The stored procedure based on any qualifiers provided (database name or owner).
    3. The stored procedure using dbo as the owner, if one is not specified.

Therefore, although the user-created stored procedure prefixed with sp\_ may exist in the current database, the master database is always checked first, even if the stored procedure is qualified with the database name.

1. **What can cause a Stored procedure execution plan to become invalidated and/or fall out of cache?**
   1. Server restart
   2. Plan is aged out due to low use
   3. DBCC FREEPROCCACHE (sometime desired to force it)
2. **When do one need to recompile stored procedure?**if a new index is added from which the stored procedure might benefit, optimization does not automatically happen (until the next time the stored procedure is run after SQL Server is restarted).
3. **SQL Server provides three ways to recompile a stored procedure:** 
   * The **sp\_recompile** system stored procedure forces a recompile of a stored procedure the next time it is run.
   * Creating a stored procedure that specifies the WITH RECOMPILE option in its definition indicates that SQL Server does not cache a plan for this stored procedure; the stored procedure is recompiled each time it is executed. Use the WITH RECOMPILE option when stored procedures take parameters whose values differ widely between executions of the stored procedure, resulting in different execution plans to be created each time. Use of this option is uncommon, and causes the stored procedure to execute more slowly because the stored procedure must be recompiled each time it is executed.
   * You can force the stored procedure to be recompiled by specifying the WITH RECOMPILE option when you execute the stored procedure. Use this option only if the parameter you are supplying is atypical or if the data has significantly changed since the stored procedure was created.
4. **How to find out which stored procedure is recompiling? How to stop stored procedures from recompiling?**
5. **I have Two Stored Procedures SP1 and SP2 as given below. How the Transaction works, whether SP2 Transaction succeeds or fails?**CREATE PROCEDURE SP1 AS  
   BEGIN TRAN  
   INSERT INTO MARKS (SID,MARK,CID) VALUES (5,6,3)  
   EXEC SP2  
   ROLLBACK  
   GO  
     
   CREATE PROCEDURE SP2 AS  
   BEGIN TRAN  
   INSERT INTO MARKS (SID,MARK,CID) VALUES (100,100,103)  
   commit tran  
   GO  
   Both will get roll backed.
6. CREATE PROCEDURE SP1 AS  
   BEGIN TRAN  
       INSERT INTO MARKS (SID,MARK,CID) VALUES (5,6,3)  
       BEGIN TRAN  
           INSERT INTO STUDENT (SID,NAME1) VALUES (1,'SA')  
       commit tran  
   ROLLBACK TRAN  
   GO  
   Both will get roll backed.
7. **How will you handle Errors in Sql Stored Procedure?**INSERT NonFatal VALUES (@Column2)  
   IF @@ERROR <>0  
    BEGIN  
     PRINT 'Error Occured'  
    END   
   [**http://www.sqlteam.com/item.asp?ItemID=2463**](http://www.sqlteam.com/item.asp?ItemID=2463)
8. **How will you raise an error in sql?**RAISERROR - Returns a user-defined error message and sets a system flag to record that an error has occurred. Using RAISERROR, the client can either retrieve an entry from the sysmessages table or build a message dynamically with user-specified severity and state information. After the message is defined it is sent back to the client as a server error message.
9. I have a stored procedure like  
   commit tran  
   create table a()  
   insert into table b  
   --  
   --  
   rollback tran  
   what will be the result? Is table created? data will be inserted in table b?
10. **What do you do when one procedure is blocking the other?  
    \*\***
11. **How you will return XML from Stored Procedure?**You use the FOR XML clause of the SELECT statement, and within the FOR XML clause you specify an XML mode: RAW, AUTO, or EXPLICIT.
12. **What are the differences between RAW, AUTO and Explicit modes in retrieving data from SQL Server in XML format?  
    \*\***
13. **Can a Stored Procedure call itself (recursive). If so then up to what level and can it be control?**Stored procedures are nested when one stored procedure calls another. You can nest stored procedures up to 32 levels. The nesting level increases by one when the called stored procedure begins execution and decreases by one when the called stored procedure completes execution. Attempting to exceed the maximum of 32 levels of nesting causes the whole calling stored procedure chain to fail. The current nesting level for the stored procedures in execution is stored in the @@NESTLEVEL function.  
    eg:  
    SET NOCOUNT ON  
    USE master  
    IF OBJECT\_ID('dbo.sp\_calcfactorial') IS NOT NULL  
    DROP PROC dbo.sp\_calcfactorial  
    GO  
    CREATE PROC dbo.sp\_calcfactorial  
    @base\_number int, @factorial int OUT  
    AS  
    DECLARE @previous\_number int  
    IF (@base\_number<2) SET @factorial=1 -- Factorial of 0 or 1=1  
    ELSE BEGIN  
    SET @previous\_number=@base\_number-1  
    EXEC dbo.sp\_calcfactorial @previous\_number, @factorial OUT -- **Recursive** call  
    IF (@factorial=-1) RETURN(-1) -- Got an error, return  
    SET @factorial=@factorial\*@base\_number  
    END  
    RETURN(0)  
    GO  
      
    calling proc.  
    DECLARE @factorial int   
    EXEC dbo.sp\_calcfactorial 4, @factorial OUT   
    SELECT @factorial
14. **Nested Triggers**Triggers are nested when a trigger performs an action that initiates another trigger, which can initiate another trigger, and so on. Triggers can be nested up to 32 levels, and you can control whether triggers can be nested through the nested triggers server configuration option.
15. **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.
16. **Difference between view and stored procedure?**Views can have only select statements (create, update, truncate, delete statements are not allowed) Views cannot have “select into”, “Group by” “Having”, ”Order by”
17. **What is a Function & what are the different user defined functions?**Function is a saved Transact-SQL routine that returns a value. User-defined functions cannot be used to perform a set of actions that modify the global database state. User-defined functions, like system functions, can be invoked from a query. They also can be executed through an EXECUTE statement like stored procedures.
    1. Scalar Functions  
       Functions are scalar-valued if the RETURNS clause specified one of the scalar data types
    2. Inline Table-valued Functions  
       If the RETURNS clause specifies TABLE with no accompanying column list, the function is an inline function.
    3. Multi-statement Table-valued Functions  
       If the RETURNS clause specifies a TABLE type with columns and their data types, the function is a multi-statement table-valued function.
18. **What are the difference between a function and a stored procedure?**
    1. Functions can be used in a select statement where as procedures cannot
    2. Procedure takes both input and output parameters but Functions takes only input parameters
    3. Functions cannot return values of type text, ntext, image & timestamps where as procedures can
    4. Functions can be used as user defined datatypes in create table but procedures cannot   
       \*\*\*Eg:-create table <tablename>(name varchar(10),salary getsal(name))  
       Here getsal is a user defined function which returns a salary type, when table is created no storage is allotted for salary type, and getsal function is also not executed, But when we are fetching some values from this table, getsal function get’s executed and the return   
       Type is returned as the result set.
19. How to debug a stored procedure?  
      
    **TRIGGER**
20. **What is Trigger? What is its use? What are the types of Triggers? What are the new kinds of triggers in sql 2000?**Triggers are a special class of stored procedure defined to execute automatically when an UPDATE, INSERT, or DELETE statement is issued against a table or view. Triggers are powerful tools that sites can use to enforce their business rules automatically when data is modified.  
    The CREATE TRIGGER statement can be defined with the FOR UPDATE, FOR INSERT, or FOR DELETE clauses to target a trigger to a specific class of data modification actions. When FOR UPDATE is specified, the IF UPDATE (column\_name) clause can be used to target a trigger to updates affecting a particular column.  
    You can use the FOR clause to specify when a trigger is executed:
    * AFTER (default) - The trigger executes after the statement that triggered it completes. If the statement fails with an error, such as a constraint violation or syntax error, the trigger is not executed. AFTER triggers cannot be specified for views.
    * INSTEAD OF -The trigger executes in place of the triggering action. INSTEAD OF triggers can be specified on both tables and views. You can define only one INSTEAD OF trigger for each triggering action (INSERT, UPDATE, and DELETE). INSTEAD OF triggers can be used to perform enhance integrity checks on the data values supplied in INSERT and UPDATE statements. INSTEAD OF triggers also let you specify actions that allow views, which would normally not support updates, to be updatable.  
      An INSTEAD OF trigger can take actions such as:
      + Ignoring parts of a batch.
      + Not processing a part of a batch and logging the problem rows.
      + Taking an alternative action if an error condition is encountered.

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.  
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.

1. **When should one use "instead of Trigger"? Example**CREATE TABLE BaseTable  
   (  
   PrimaryKey int IDENTITY(1,1),  
   Color nvarchar(10) NOT NULL,  
   Material nvarchar(10) NOT NULL,  
   ComputedCol AS (Color + Material)  
   )  
   GO  
     
   --Create a view that contains all columns from the base table.  
   CREATE VIEW InsteadView  
   AS SELECT PrimaryKey, Color, Material, ComputedCol  
   FROM BaseTable  
   GO  
     
   --Create an INSTEAD OF INSERT trigger on tthe view.  
   CREATE TRIGGER InsteadTrigger on InsteadView  
   INSTEAD OF INSERT  
   AS  
   BEGIN  
   --Build an INSERT statement ignoring inserrted.PrimaryKey and   
   --inserted.ComputedCol.  
   INSERT INTO BaseTable  
   SELECT Color, Material  
   FROM inserted  
   END  
   GO  
     
   -- can insert value to basetable by this insert into basetable(color,material) values ('red','abc')  
     
   -- insert into InsteadView(color,material)) values ('red','abc') can't do this.  
   -- It will give error "'PrimaryKey' iin table 'InsteadView' cannot be null."  
     
   -- can insert value through table by this<  
   insert into InsteadView values (1,'red','abc',1) --PrimaryKey, ComputedCol wont take values from here
2. **Difference between trigger and stored procedure?**Trigger will get execute automatically when an UPDATE, INSERT, or DELETE statement is issued against a table or view.  
   We have to call stored procedure manually, or it can execute automatic when the SQL Server starts (You can use the sp\_procoption system stored procedure to mark the stored procedure to automatic execution when the SQL Server will start.
3. **The following trigger generates an e-mail whenever a new title is added.**CREATE TRIGGER reminder  
   ON titles  
   FOR INSERT  
   AS  
   EXEC master..xp\_sendmail 'MaryM', 'New title, mention in the next report to distributors.'
4. **Drawback of trigger? Its alternative solution?**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.  
     
   **LOCK**
5. **What are locks?**Microsoft® SQL Server™ 2000 uses locking to ensure transactional integrity and database consistency. Locking prevents users from reading data being changed by other users, and prevents multiple users from changing the same data at the same time. If locking is not used, data within the database may become logically incorrect, and queries executed against that data may produce unexpected results.
6. **What are the different types of locks?**SQL Server uses these resource lock modes.

**Lock mode** **: Description**

Shared (S) : Used for operations that do not change or update data (read-only operations), such as a SELECT statement.

Update (U) : Used on resources that can be updated. Prevents a common form of deadlock that occurs when multiple sessions are reading, locking, and potentially updating resources later.

Exclusive (X) : Used for data-modification operations, such as INSERT, UPDATE, or DELETE. Ensures that multiple updates cannot be made to the same resource at the same time.

Intent : Used to establish a lock hierarchy. The types of intent locks are: intent shared (IS), intent exclusive (IX), and shared with intent exclusive (SIX).

Schema : Used when an operation dependent on the schema of a table is executing. The types of schema locks are: schema modification (Sch-M) and schema stability (Sch-S).

Bulk Update (BU) : Used when bulk-copying data into a table and the TABLOCK hint is specified.

1. **What is a dead lock? Give a practical sample? How you can minimize the deadlock situation? 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.)
2. **What is isolation level?**An isolation level determines the degree of isolation of data between concurrent transactions. The default SQL Server isolation level is Read Committed. A lower isolation level increases concurrency, but at the expense of data correctness. Conversely, a higher isolation level ensures that data is correct, but can affect concurrency negatively. The isolation level required by an application determines the locking behavior SQL Server uses.  
   SQL-92 defines the following isolation levels, all of which are supported by SQL Server:
   * Read uncommitted (the lowest level where transactions are isolated only enough to ensure that physically corrupt data is not read).
   * Read committed (SQL Server default level).
   * Repeatable read.
   * Serializable (the highest level, where transactions are completely isolated from one another).

**Isolation level:Dirty read :Nonrepeatable read :Phantom**

Read uncommitted :Yes : Yes : Yes

Read committed : No : Yes : Yes

Repeatable read : No : No : Yes

Serializable : No : No : No

1. Uncommitted Dependency (Dirty Read) - Uncommitted dependency occurs when a second transaction selects a row that is being updated by another transaction. The second transaction is reading data that has not been committed yet and may be changed by the transaction updating the row. For example, an editor is making changes to an electronic document. During the changes, a second editor takes a copy of the document that includes all the changes made so far, and distributes the document to the intended audience.  
   Inconsistent Analysis (Nonrepeatable Read) Inconsistent analysis occurs when a second transaction accesses the same row several times and reads different data each time. Inconsistent analysis is similar to uncommitted dependency in that another transaction is changing the data that a second transaction is reading. However, in inconsistent analysis, the data read by the second transaction was committed by the transaction that made the change. Also, inconsistent analysis involves multiple reads (two or more) of the same row and each time the information is changed by another transaction; thus, the term nonrepeatable read. For example, an editor reads the same document twice, but between each reading, the writer rewrites the document. When the editor reads the document for the second time, it has changed.  
   Phantom Reads Phantom reads occur when an insert or delete action is performed against a row that belongs to a range of rows being read by a transaction. The transaction's first read of the range of rows shows a row that no longer exists in the second or succeeding read, as a result of a deletion by a different transaction. Similarly, as the result of an insert by a different transaction, the transaction's second or succeeding read shows a row that did not exist in the original read. For example, an editor makes changes to a document submitted by a writer, but when the changes are incorporated into the master copy of the document by the production department, theyfind that new unedited material has been added to the document by the author. This problem could be avoided if no one could add new material to the document until the editor and production department finish working with the original document.
2. **nolock? What is the difference between the REPEATABLE READ and SERIALIZE isolation levels?  
   Locking Hints -** A range of table-level locking hints can be specified using the SELECT, INSERT, UPDATE, and DELETE statements to direct Microsoft® SQL Server 2000 to the type of locks to be used. Table-level locking hints can be used when a finer control of the types of locks acquired on an object is required. These locking hints override the current transaction isolation level for the session.

**Locking hint:Description**

HOLDLOCK:Hold a shared lock until completion of the transaction instead of releasing the lock as soon as the required table, row, or data page is no longer required. HOLDLOCK is equivalent to SERIALIZABLE.

NOLOCK:Do not issue shared locks and do not honor exclusive locks. When this option is in effect, it is possible to read an uncommitted transaction or a set of pages that are rolled back in the middle of a read. Dirty reads are possible. Only applies to the SELECT statement.

PAGLOCK:Use page locks where a single table lock would usually be taken.

READCOMMITTED:Perform a scan with the same locking semantics as a transaction running at the READ COMMITTED isolation level. By default, SQL Server 2000 operates at this isolation level.

READPAST:Skip locked rows. This option causes a transaction to skip rows locked by other transactions that would ordinarily appear in the result set, rather than block the transaction waiting for the other transactions to release their locks on these rows. The READPAST lock hint applies only to transactions operating at READ COMMITTED isolation and will read only past row-level locks. Applies only to the SELECT statement.

READUNCOMMITTED:Equivalent to NOLOCK.

REPEATABLEREAD:Perform a scan with the same locking semantics as a transaction running at the REPEATABLE READ isolation level.

ROWLOCK:Use row-level locks instead of the coarser-grained page- and table-level locks.

SERIALIZABLE:Perform a scan with the same locking semantics as a transaction running at the SERIALIZABLE isolation level. Equivalent to HOLDLOCK.

TABLOCK:Use a table lock instead of the finer-grained row- or page-level locks. SQL Server holds this lock until the end of the statement. However, if you also specify HOLDLOCK, the lock is held until the end of the transaction.

TABLOCKX:Use an exclusive lock on a table. This lock prevents others from reading or updating the table and is held until the end of the statement or transaction.

UPDLOCK:Use update locks instead of shared locks while reading a table, and hold locks until the end of the statement or transaction. UPDLOCK has the advantage of allowing you to read data (without blocking other readers) and update it later with the assurance that the data has not changed since you last read it.

XLOCK:Use an exclusive lock that will be held until the end of the transaction on all data processed by the statement. This lock can be specified with either PAGLOCK or TABLOCK, in which case the exclusive lock applies to the appropriate level of granularity.

1. For example, if the transaction isolation level is set to SERIALIZABLE, and the table-level locking hint NOLOCK is used with the SELECT statement, key-range locks typically used to maintain serializable transactions are not taken.  
   USE pubs  
   GO  
   SET TRANSACTION ISOLATION LEVEL SERIALIZABLE  
   GO  
   BEGIN TRANSACTION  
   SELECT au\_lname FROM authors WITH (NOLOCK)  
   GO
2. **What is escalation of locks?**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.  
     
   **VIEW**
3. **What is View? Use? Syntax of View?**A view is a virtual table made up of data from base tables and other views, but not stored separately.
   * Views simplify users perception of the database (can be used to present only the necessary information while hiding details in underlying relations)
   * Views improve data security preventing undesired accesses
   * Views facilite the provision of additional data independence
4. **Does the View occupy memory space?**No
5. **Can u drop a table if it has a view?**Views or tables participating in a view created with the SCHEMABINDING clause cannot be dropped. If the view is not created using SCHEMABINDING, then we can drop the table.
6. **Why doesn't SQL Server permit an ORDER BY clause in the definition of a view?**SQL Server excludes an ORDER BY clause from a view to comply with the ANSI SQL-92 standard. Because analyzing the rationale for this standard requires a discussion of the underlying structure of the structured query language (SQL) and the mathematics upon which it is based, we can't fully explain the restriction here. However, if you need to be able to specify an ORDER BY clause in a view, consider using the following workaround:  
   USE pubs  
   GO   
   CREATE VIEW AuthorsByName  
   AS  
   SELECT TOP 100 PERCENT \*  
   FROM authors   
   ORDER BY au\_lname, au\_fname  
   GO  
   The TOP construct, which Microsoft introduced in SQL Server 7.0, is most useful when you combine it with the ORDER BY clause. The only time that SQL Server supports an ORDER BY clause in a view is when it is used in conjunction with the TOP keyword. (Note that the TOP keyword is a SQL Server extension to the ANSI SQL-92 standard.)  
     
   **TRANSACTION**
7. **What is Transaction?**A transaction is a sequence of operations performed as a single logical unit of work. A logical unit of work must exhibit four properties, called the ACID (Atomicity, Consistency, Isolation, and Durability) properties, to qualify as a transaction:
   * **Atomicity -** A transaction must be an atomic unit of work; either all of its data modifications are performed or none of them is performed.
   * **Consistency -** When completed, a transaction must leave all data in a consistent state. In a relational database, all rules must be applied to the transaction's modifications to maintain all data integrity. All internal data structures, such as B-tree indexes or doubly-linked lists, must be correct at the end of the transaction.
   * **Isolation -** Modifications made by concurrent transactions must be isolated from the modifications made by any other concurrent transactions. A transaction either sees data in the state it was in before another concurrent transaction modified it, or it sees the data after the second transaction has completed, but it does not see an intermediate state. This is referred to as serializability because it results in the ability to reload the starting data and replay a series of transactions to end up with the data in the same state it was in after the original transactions were performed.
   * **Durability -** After a transaction has completed, its effects are permanently in place in the system. The modifications persist even in the event of a system failure.
8. **After one Begin Transaction a truncate statement and a RollBack statements are there. Will it be rollbacked? Since the truncate statement does not perform logged operation how does it RollBack?**It will rollback.  
   \*\*
9. Given a SQL like   
   Begin Tran  
      Select @@Rowcount  
   Begin Tran  
      Select @@Rowcount  
   Begin Tran  
      Select @@Rowcount  
   Commit Tran  
      Select @@Rowcount  
   RollBack  
      Select @@Rowcount  
   RollBack  
      Select @@Rowcount  
   What is the value of @@Rowcount at each stmt levels?  
   Ans : 0 – zero.  
   @@ROWCOUNT - Returns the number of rows affected by the last statement.  
   @@TRANCOUNT - Returns the number of active transactions for the current connection.  
   Each Begin Tran will add count, each commit will reduce count and ONE rollback will make it 0.  
     
   OTHER
10. **What are the constraints for Table Constraints define rules regarding the values allowed in columns and are the standard mechanism for enforcing integrity. SQL Server 2000 supports five classes of constraints.**NOT NULL  
    CHECK  
    UNIQUE  
    PRIMARY KEY  
    FOREIGN KEY
11. **There are 50 columns in a table. Write a query to get first 25 columns**Ans: Need to mention each column names.
12. **How to list all the tables in a particular database?**USE pubs  
    GO  
    sp\_help
13. **What are cursors? Explain different types of cursors. What are the disadvantages of cursors? How can you avoid cursors?**Cursors allow row-by-row processing of the result sets.  
    Types of cursors: Static, Dynamic, Forward-only, Keyset-driven.  
    Disadvantages of cursors: Each time you fetch a row from the cursor, it results in a network roundtrip. Cursors are also costly because they require more resources and temporary storage (results in more IO operations). Further, there are restrictions on the SELECT statements that can be used with some types of cursors.  
    How to avoid cursor:
    1. 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
    2. 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.
14. **What is Dynamic Cursor? Suppose, I have a dynamic cursor attached to table in a database.  I have another means by which I will modify the table.  What do you think will the values in the cursor be?**Dynamic cursors reflect all changes made to the rows in their result set when scrolling through the cursor. The data values, order, and membership of the rows in the result set can change on each fetch. All UPDATE, INSERT, and DELETE statements made by all users are visible through the cursor. Updates are visible immediately if they are made through the cursor using either an API function such as SQLSetPos or the Transact-SQL WHERE CURRENT OF clause. Updates made outside the cursor are not visible until they are committed, unless the cursor transaction isolation level is set to read uncommitted.
15. **What is DATEPART?**Returns an integer representing the specified datepart of the specified date.
16. **Difference between Delete and Truncate?**TRUNCATE TABLE is functionally identical to DELETE statement with no WHERE clause: both remove all rows in the table.  
    (1) But TRUNCATE TABLE is faster and uses fewer system and transaction log resources than DELETE. The DELETE statement removes rows one at a time and records an entry in the transaction log for each deleted row. TRUNCATE TABLE removes the data by deallocating the data pages used to store the table's data, and only the page deallocations are recorded in the transaction log.  
    (2) Because TRUNCATE TABLE is not logged, it cannot activate a trigger.  
    (3) The counter used by an identity for new rows is reset to the seed for the column. If you want to retain the identity counter, use DELETE instead.  
    Of course, TRUNCATE TABLE can be rolled back.
17. **Given a scenario where two operations, Delete Stmt and Truncate Stmt, where the Delete Statement was successful and the truncate stmt was failed. – Can u judge why?  
    \*\***
18. **What are global variables? Tell me some of them?**Transact-SQL global variables are a form of function and are now referred to as functions.  
    ABS - Returns the absolute, positive value of the given numeric expression.  
    SUM  
    AVG  
    AND
19. **What is DDL?**Data definition language (DDL) statements are SQL statements that support the definition or declaration of database objects (for example, CREATE TABLE, DROP TABLE, and ALTER TABLE).  
    You can use the ADO Command object to issue DDL statements. To differentiate DDL statements from a table or stored procedure name, set the CommandType property of the Command object to adCmdText. Because executing DDL queries with this method does not generate any recordsets, there is no need for a Recordset object.
20. **What is DML?**Data Manipulation Language (DML), which is used to select, insert, update, and delete data in the objects defined using DDL
21. **What are keys in RDBMS? What is a primary key/ foreign key?**There are two kinds of keys.  
    A primary key is a set of columns from a table that are guaranteed to have unique values for each row of that table.  
    Foreign keys are attributes of one table that have matching values in a **primary key** in another table, allowing for relationships between tables.
22. **What is the difference between Primary Key and Unique Key?**Both primary key and unique key 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.
23. **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.
24. **What is the Referential Integrity?**Referential integrity refers to the consistency that must be maintained between primary and foreign keys, i.e. every foreign key value must have a corresponding primary key value.
25. **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.
26. **What is Query optimization? How is tuning a performance of query done?**
27. **What is the use of trace utility?  
    \*\***
28. **What is the use of shell commands? xp\_cmdshell**Executes a given command string as an operating-system command shell and returns any output as rows of text. Grants nonadministrative users permissions to execute **xp\_cmdshell**.
29. **What is use of shrink database?**Microsoft® SQL Server 2000 allows each file within a database to be shrunk to remove unused pages. Both data and transaction log files can be shrunk.
30. **If the performance of the query suddenly decreased where you will check?**
31. **What is a pass-through query?**Microsoft® SQL Server 2000 sends pass-through queries as un-interpreted query strings to an OLE DB data source. The query must be in a syntax the OLE DB data source will accept. A Transact-SQL statement uses the results from a pass-through query as though it is a regular table reference.  
    This example uses a pass-through query to retrieve a result set from a Microsoft Access version of the Northwind sample database.  
    SELECT \*  
    FROM OpenRowset('Microsoft.Jet.OLEDB.4.0',   
    'c:\northwind.mdb';'admin'; '',   
    'SELECT CustomerID, CompanyName  
    FROM Customers  
    WHERE Region = ''WA'' ')
32. **How do you differentiate Local and Global Temporary table?**You can create local and global temporary tables. Local temporary tables are visible only in the current session; global temporary tables are visible to all sessions. Prefix local temporary table names with single number sign (#*table\_name*), and prefix global temporary table names with a double number sign (##*table\_name*). SQL statements reference the temporary table using the value specified for *table\_name* in the CREATE TABLE statement:  
    CREATE TABLE #MyTempTable (cola INT PRIMARY KEY)  
    INSERT INTO #MyTempTable VALUES (1)
33. **How the Exists keyword works in SQL Server?**USE pubs  
    SELECT au\_lname, au\_fname  
    FROM authors  
    WHERE exists  
       (SELECT \*  
       FROM publishers  
       WHERE authors.city = publishers.city)  
    When a subquery is introduced with the keyword EXISTS, it functions as an existence test. The WHERE clause of the outer query tests for the existence of rows returned by the subquery. The subquery does not actually produce any data; it returns a value of TRUE or FALSE.
34. **ANY?**USE pubs  
    SELECT au\_lname, au\_fname  
    FROM authors  
    WHERE city = ANY  
    (SELECT city  
    FROM publishers)
35. **to select date part only**SELECT CONVERT(char(10),GetDate(),101)  
    --to select time part only  
    SELECT right(GetDate(),7)
36. **How can I send a message to user from the SQL Server?**You can use the xp\_cmdshell extended stored procedure to run net send command. This is the example to send the 'Hello' message to JOHN:  
    EXEC master..xp\_cmdshell "net send JOHN 'Hello'"  
    To get net send message on the Windows 9x machines, you should run the WinPopup utility. You can place WinPopup in the Startup group under Program Files.
37. **What is normalization? Explain different levels of normalization? Explain Third normalization form with an example?**The process of refining tables, keys, columns, and relationships to create an efficient database is called *normalization*. This should eliminates unnecessary duplication and provides a rapid search path to all necessary information.  
    Some of the benefits of normalization are:
    * Data integrity (because there is no redundant, neglected data)
    * Optimized queries (because normalized tables produce rapid, efficient joins)
    * Faster index creation and sorting (because the tables have fewer columns)
    * Faster UPDATE performance (because there are fewer indexes per table)
    * Improved concurrency resolution (because table locks will affect less data)
    * Eliminate redundancy

There are a few rules for database normalization. Each rule is called a "normal form." If the first rule is observed, the database is said to be in "first normal form." If the first three rules are observed, the database is considered to be in "third normal form." Although other levels of normalization are possible, third normal form is considered the highest level necessary for most applications.

* 1. **First Normal Form (1NF)**
     + Eliminate repeating groups in individual tables
     + Create a separate table for each set of related data.
     + Identify each set of related data with a primary key.

Do not use multiple fields in a single table to store similar data.   
Example

**Subordinate1**  **Subordinate2**  **Subordinate3**  **Subordinate4**

Bob Jim Mary Beth

Mary Mike Jason Carol Mark

Jim Alan

Eliminate duplicative columns from the same table.  Clearly, the Subordinate1-Subordinate4 columns are duplicative. What happens when we need to add or remove a subordinate?

**Subordinates**

Bob Jim, Mary, Beth

Mary Mike, Jason, Carol, Mark

Jim Alan

This solution is closer, but it also falls short of the mark. The subordinates column is still duplicative and non-atomic. What happens when we need to add or remove a subordinate? We need to read and write the entire contents of the table. That’s not a big deal in this situation, but what if one manager had one hundred employees? Also, it complicates the process of selecting data from the database in future queries.   
Solution:

**Subordinate**

Bob Jim

Bob Mary

Bob Beth

Mary Mike

Mary Jason

Mary Carol

Mary Mark

Jim Alan

* 1. **Second Normal Form (2NF)**
     + Create separate tables for sets of values that apply to multiple records.
     + Relate these tables with a foreign key.

Records should not depend on anything other than a table's primary key (a compound key, if necessary).   
For example, consider a customer's address in an accounting system. The address is needed by the Customers table, but also by the Orders, Shipping, Invoices, Accounts Receivable, and Collections tables. Instead of storing the customer's address as a separate entry in each of these tables, store it in one place, either in the Customers table or in a separate Addresses table.

* 1. **Third Normal Form (3NF)**
     + Eliminate fields that do not depend on the key.

Values in a record that are not part of that record's key do not belong in the table. In general, any time the contents of a group of fields may apply to more than a single record in the table, consider placing those fields in a separate table.  
For example, in an Employee Recruitment table, a candidate's university name and address may be included. But you need a complete list of universities for group mailings. If university information is stored in the Candidates table, there is no way to list universities with no current candidates. Create a separate Universities table and link it to the Candidates table with a university code key.  
Another Example :

**MemberId** **Name** **Company** **CompanyLoc**

1 John Smith ABC Alabama

2 Dave Jones MCI Florida

The Member table satisfies first normal form - it contains no repeating groups. It satisfies second normal form - since it doesn't have a multivalued key. But the key is MemberID, and the company name and location describe only a company, not a member. To achieve third normal form, they must be moved into a separate table. Since they describe a company, CompanyCode becomes the key of the new "Company" table.

The motivation for this is the same for second normal form: we want to avoid update and delete anomalies. For example, suppose no members from the IBM were currently stored in the database. With the previous design, there would be no record of its existence, even though 20 past members were from IBM!  
Member Table

**MemberId** **Name** **CID**

1 John Smith 1

2 Dave Jones 2

Company Table

**CId** **Name** **Location**

1 ABC Alabama

2 MCI Florida

* 1. **Boyce-Codd Normal Form (BCNF)**A relation is in Boyce/Codd normal form if and only if the only determinants are candidate key. Its a different version of 3NF, indeed, was meant to replace it. [A determinant is any attribute on which some other attribute is (fully) functionally dependent.]
  2. **4th Normal Form (4NF)**A table is in 4NF if it is in BCNF and if it has no multi-valued dependencies. This applies primarily to key-only associative tables, and appears as a ternary relationship, but has incorrectly merged 2 distinct, independent relationships.  
     Eg: This could be any 2 M:M relationships from a single entity. For instance, a member could know many software tools, and a software tool may be used by many members. Also, a member could have recommended many books, and a book could be recommended by many members.

Software   member   Book

* 1. The correct solution, to cause the model to be in 4th normal form, is to ensure that all M:M relationships are resolved independently if they are indeed independent.

Software   membersoftware member memberBook book

* 1. **5th Normal Form (5NF)(PJNF)**A table is in 5NF, also called "Projection-Join Normal Form", if it is in 4NF and if every join dependency in the table is a consequence of the candidate keys of the table.
  2. **Domain/key normal form (DKNF)**. A key uniquely identifies each row in a table. A domain is the set of permissible values for an attribute. By enforcing key and domain restrictions, the database is assured of being freed from modification anomalies. DKNF is the normalization level that most designers aim to achieve.

\*\*  
Remember, these normalization guidelines are cumulative.  For a database to be in 2NF, it must first fulfill all the criteria of a 1NF database.

1. **If a database is normalized by 3 NF then how many number of tables it should contain in minimum? How many minimum if 2NF and 1 NF?**
2. **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.
3. **How can I randomly sort query results?**To randomly order rows, or to return *x* number of randomly chosen rows, you can use the RAND function inside the SELECT statement. But the RAND function is resolved only once for the entire query, so every row will get same value. You can use an ORDER BY clause to sort the rows by the result from the NEWID function, as the following code shows:  
   SELECT \*  
   FROM Northwind..Orders   
   ORDER BY NEWID()
4. **sp\_who**Provides information about current Microsoft® SQL Server™ users and processes. The information returned can be filtered to return only those processes that are not idle.
5. **Have you worked on Dynamic SQL? How will You handled “ (Double Quotes) in Dynamic SQL?**
6. **How to find dependents of a table?**Verify dependencies with **sp\_depends** before dropping an object
7. **What is the difference between a CONSTRAINT AND RULE?**Rules are a backward-compatibility feature that perform some of the same functions as CHECK constraints. CHECK constraints are the preferred, standard way to restrict the values in a column. CHECK constraints are also more concise than rules; there can only be one rule applied to a column, but multiple CHECK constraints can be applied. CHECK constraints are specified as part of the CREATE TABLE statement, while rules are created as separate objects and then bound to the column.
8. **How to call a COM dll from SQL Server 2000?**sp\_OACreate - Creates an instance of the OLE object on an instance of Microsoft® SQL Server  
   **Syntax  
   sp\_OACreate** *progid***,** | *clsid***,**   
       *objecttoken* **OUTPUT**   
       [ **,** *context* ]

*context -* Specifies the execution context in which the newly created OLE object runs. If specified, this value must be one of the following: **1** = In-process (.dll) OLE server only  
**4** = Local (.exe) OLE server only  
**5** = Both in-process and local OLE server allowed

##### Examples

###### A. Use Prog ID - This example creates a SQL-DMO SQLServer object by using its ProgID.

DECLARE @object int

DECLARE @hr int

DECLARE @src varchar(255), @desc varchar(255)

EXEC @hr = sp\_OACreate 'SQLDMO.SQLServer', @object OUT

IF @hr <> 0

BEGIN

EXEC sp\_OAGetErrorInfo @object, @src OUT, @desc OUT

SELECT hr=convert(varbinary(4),@hr), Source=@src, Description=@desc

RETURN

END

B. Use CLSID - This example creates a SQL-DMO SQLServer object by using its CLSID.

DECLARE @object int

DECLARE @hr int

DECLARE @src varchar(255), @desc varchar(255)

EXEC @hr = sp\_OACreate '{00026BA1-0000-0000-C000-000000000046}',

@object OUT

IF @hr <> 0

BEGIN

EXEC sp\_OAGetErrorInfo @object, @src OUT, @desc OUT

SELECT hr=convert(varbinary(4),@hr), Source=@src, Description=@desc

RETURN

END

1. **Difference between sysusers and syslogins?**sysusers - Contains one row for each Microsoft® Windows user, Windows group, Microsoft SQL Server™ user, or SQL Server role in the database.  
   syslogins - Contains one row for each login account.
2. **What is the row size in SQL Server 2000?**8060 bytes.
3. **How will you find structure of table, all** tables/views **in one db, all dbs?**//structure of tablesp\_helpdb tbl\_emp  
    **//**list of all databasessp\_helpdb  
   OR  
   SELECT \* FROM master.dbo.sysdatabases  
     
   //details about database pubs. .mdf, .ldf file locations, size of database  
   sp\_helpdb pubs  
     
   //lists all tables under current database  
   sp\_tables  
   OR  
   SELECT \* FROM information\_schema.tables WHERE (table\_type = 'base table')  
   OR  
   SELECT \* FROM sysobjects WHERE type = 'U' //faster
4. **B-tree indexes or doubly-linked lists?**
5. **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().
6. **What are the series of steps that happen on execution of a query in a Query Analyzer?**1) Syntax checking 2) Parsing 3) Execution plan
7. **Which event (Check constraints, Foreign Key, Rule, trigger, Primary key check) will be performed last for integrity check?**Identity Insert Check  
   Nullability constraint  
   Data type check  
   Instead of trigger  
   Primary key  
   Check constraint  
   Foreign key  
   DML Execution (update statements)  
   After Trigger **\*\***
8. **How will you show many to many relation in sql?**Create 3rd table with 2 columns which having one to many relation to these tables.
9. **When a query is sent to the database and an index is not being used, what type of execution is taking place?**A table scan.
10. **What is #, ##, @, @@ means?**@@ - System variables  
    @ - user defined variables
11. **What is the difference between a Local temporary table and a Global temporary table? How is each one denoted?**Local temporary table will be accessible to only current user session, its name will be preceded with a single hash (#mytable)  
    Global temporary table will be accessible to all users, & it will be dropped only after ending of all active connections, its name will be preceded with double hash (##mytable)
12. **What is covered queries in SQL Server?**
13. **What is HASH JOIN, MERGE JOIN?**  
    **TOOLS**
14. **Have you ever used DBCC command? Give an example for it.**The Transact-SQL programming language provides DBCC statements that act as Database Console Commands for Microsoft® SQL Serve 2000. These statements check the physical and logical consistency of a database. Many DBCC statements can fix detected problems. Database Console Command statements are grouped into these categories.

**Statement category:Perform**

Maintenance statements:Maintenance tasks on a database, index, or filegroup.

Miscellaneous statements:Miscellaneous tasks such as enabling row-level locking or removing a dynamic-link library (DLL) from memory.

Status statements:Status checks.

Validation statements:Validation operations on a database, table, index, catalog, filegroup, system tables, or allocation of database pages.

DBCC CHECKDB, DBCC CHECKTABLE, DBCC CHECKCATALOG, DBCC CHECKALLOC, DBCC SHOWCONTIG, DBCC SHRINKDATABASE, DBCC SHRINKFILE etc.

1. **How do you use DBCC statements to monitor various aspects of a SQL server installation?  
   \*\***
2. **What is the output of DBCC Showcontig statement?**Displays fragmentation information for the data and indexes of the specified table.
3. **How do I reset the identity column?**You can use the DBCC CHECKIDENT statement, if you want to reset or reseed the identity column. For example, if you need to force the current identity value in the jobs table to a value of 100, you can use the following:  
   USE pubs  
   GO  
   DBCC CHECKIDENT (jobs, RESEED, 100)  
   GO
4. **About SQL Command line executables**

**Utilities**

bcp  
console  
isql  
sqlagent  
sqldiag  
sqlmaint  
sqlservr  
vswitch

dtsrun  
dtswiz  
isqlw  
itwiz  
odbccmpt  
osql  
rebuildm  
sqlftwiz

distrib  
logread  
replmerg  
snapshot

scm

regxmlss

1. **What is DTC?**The Microsoft Distributed Transaction Coordinator (MS DTC) is a transaction manager that allows client applications to include several different sources of data in one transaction. MS DTC coordinates committing the distributed transaction across all the servers enlisted in the transaction.
2. **What is DTS? Any drawbacks in using DTS?**Microsoft® SQL Server™ 2000 Data Transformation Services (DTS) is a set of graphical tools and programmable objects that lets you extract, transform, and consolidate data from disparate sources into single or multiple destinations.
3. **What is BCP?**The **bcp** utility copies data between an instance of Microsoft® SQL Server™ 2000 and a data file in a user-specified format.  
   C:\Documents and Settings\sthomas>bcp  
   usage: bcp {dbtable | query} {in | out | queryout | format} datafile  
   [-m maxerrors] [-f formatfile] [-e errfile]  
   [-F firstrow] [-L lastrow] [-b batchsize]  
   [-n native type] [-c character type] [-w wide character type]  
   [-N keep non-text native] [-V file format version] [-q quoted identifier]  
   [-C code page specifier] [-t field terminator] [-r row terminator]  
   [-i inputfile] [-o outfile] [-a packetsize]  
   [-S server name] [-U username] [-P password]  
   [-T trusted connection] [-v version] [-R regional enable]  
   [-k keep null values] [-E keep identity values]  
   [-h "load hints"]
4. **How can I create a plain-text flat file from SQL Server as input to another application?**One of the purposes of Extensible Markup Language (XML) is to solve challenges like this, but until all applications become XML-enabled, consider using our faithful standby, the bulk copy program (bcp) utility. This utility can do more than just dump a table; bcp also can take its input from a view instead of from a table. After you specify a view as the input source, you can limit the output to a subset of columns or to a subset of rows by selecting appropriate filtering (WHERE and HAVING) clauses.  
   More important, by using a view, you can export data from multiple joined tables. The only thing you cannot do is specify the sequence in which the rows are written to the flat file, because a view does not let you include an ORDER BY clause in it unless you also use the TOP keyword.  
   If you want to generate the data in a particular sequence or if you cannot predict the content of the data you want to export, be aware that in addition to a view, bcp also supports using an actual query. The only "gotcha" about using a query instead of a table or view is that you must specify **queryout** in place of **out** in the bcp command line.  
   For example, you can use bcp to generate from the **pubs** database a list of authors who reside in California by writing the following code:  
   bcp "SELECT \* FROM pubs..authors WHERE state = 'CA'" queryout c:\CAauthors.txt -c -T -S
5. **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, detaching and attaching databases, replication, DTS, BCP, logshipping, INSERT...SELECT, SELECT...INTO, creating INSERT scripts to generate data.
6. **How will I export database?**Through DTS - Import/Export wizard  
   Backup - through Complete/Differential/Transaction Log
7. **How to export database at a particular time, every week?**Backup - Schedule  
   DTS - Schedule  
   Jobs - create a new job
8. **How do you load large data to the SQL server database?**bcp
9. **How do you transfer data from text file to database (other than DTS)?**bcp
10. **What is OSQL and ISQL utility?**The **osql** utility allows you to enter Transact-SQL statements, system procedures, and script files. This utility uses ODBC to communicate with the server.   
    The **isql** utility allows you to enter Transact-SQL statements, system procedures, and script files; and uses DB-Library to communicate with Microsoft® SQL Server™ 2000.  
    All DB-Library applications, such as isql, work as SQL Server 6.5–level clients when connected to SQL Server 2000. They do not support some SQL Server 2000 features.  
    The osql utility is based on ODBC and does support all SQL Server 2000 features. Use osql to run scripts that isql cannot run.
11. **What Tool you have used for checking Query Optimization? What is the use of profiler in sql server?  What is the first thing u look at in a SQL Profiler?**SQL Profiler is a graphical tool that allows system administrators to monitor events in an instance of Microsoft® SQL Server™. You can capture and save data about each event to a file or SQL Server table to analyze later. For example, you can monitor a production environment to see which stored procedures is hampering performance by executing too slowly.   
    Use SQL Profiler to:
    * Monitor the performance of an instance of SQL Server.
    * Debug Transact-SQL statements and stored procedures.
    * Identify slow-executing queries.
    * Test SQL statements and stored procedures in the development phase of a project by single-stepping through statements to confirm that the code works as expected.
    * Troubleshoot problems in SQL Server by capturing events on a production system and replaying them on a test system. This is useful for testing or debugging purposes and allows users to continue using the production system without interference.

Audit and review activity that occurred on an instance of SQL Server. This allows a security administrator to review any of the auditing events, including the success and failure of a login attempt and the success and failure of permissions in accessing statements and objects.  
  
**Permissions**

1. **A user is a member of Public role and Sales role. Public role has the permission to select on all the table, and Sales role, which doesn’t have a select permission on some of the tables. Will that user be able to select from all tables?  
   \*\***
2. **If a user does not have permission on a table, but he has permission to a view created on it, will he be able to view the data in table?**Yes.
3. **Describe Application Role and explain a scenario when you will use it?  
   \*\***
4. **After removing a table from database, what other related objects have to be dropped explicitly?**(view, SP)
5. **You have a SP names YourSP and have the a Select Stmt inside the SP. You also have a user named YourUser. What permissions you will give him for accessing the SP.  
   \*\***
6. **Different Authentication modes in Sql server? If a user is logged under windows authentication mode, how to find his userid?**There are Three Different authentication modes in sqlserver.
   1. Windows Authentication Mode
   2. SqlServer Authentication Mode
   3. Mixed Authentication Mode

“system\_user” system function in sqlserver to fetch the logged on user name.

1. **Give the connection strings from front-end for both type logins(windows,sqlserver)?**This are specifically for sqlserver not for any other RDBMS  
   Data Source=MySQLServer;Initial Catalog=NORTHWIND;Integrated Security=SSPI (windows)  
   Data Source=MySQLServer;Initial Catalog=NORTHWIND;Uid=” ”;Pwd=” ”(sqlserver)
2. **What are three SQL keywords used to change or set someone’s permissions?**  
   Grant, Deny and Revoke**Administration**
3. **Explain the architecture of SQL Server?  
   \*\***
4. **Different types of Backups?**
   * A full database backup is a full copy of the database.
   * A transaction log backup copies only the transaction log.
   * A differential backup copies only the database pages modified after the last full database backup.
   * A file or filegroup restore allows the recovery of just the portion of a database that was on the failed disk.
5. **What are ‘jobs’ in SQL Server? How do we create one? What is tasks?**Using SQL Server Agent jobs, you can automate administrative tasks and run them on a recurring basis.  
   \*\*
6. **What is database replication? What are the different types of replication you can set up in SQL Server? How are they used? What is snapshot replication how is it different from Transactional replication?**  
   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:
   1. Snapshot replication - It distributes data exactly as it appears at a specific moment in time and doesn’t monitor for updates. It can be used when data changes are infrequent. It is often used for browsing data such as price lists, online catalog, or data for decision support where the current data is not required and data is used as read only.
   2. Transactional replication (with immediate updating subscribers, with queued updating subscribers) - With this an initial snapshot of data is applied, and whenever data modifications are made at the publisher, the individual transactions are captured and propagated to the subscribers.
   3. Merge replication - It is the process of distributing the data between publisher and subscriber, it allows the publisher and subscriber to update the data while connected or disconnected, and then merging the updates between the sites when they are connected.
7. **How can u look at what are the process running on SQL server? How can you kill a process in SQL server?**
   * Expand a server group, and then expand a server.
   * Expand **Management**, and then expand **Current Activity**.
   * Click **Process Info**. The current server activity is displayed in the details pane.

In the details pane, right-click a Process ID, and then click **Kill Process**.

1. **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.
2. 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.
3. **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.  
   eg: Microsoft SQL Server 2000 - 8.00.760 (Intel X86) Dec 17 2002 14:22:05 Copyright (c) 1988-2003 Microsoft Corporation Enterprise Edition on Windows NT 5.0 (Build 2195: Service Pack 3)
4. **What is the purpose of using COLLATE in a query?**  
   The term, collation, refers to a set of rules that determine how data is sorted and compared. In Microsoft® SQL Server 2000, it is not required to separately specify code page and sort order for character data, and the collation used for Unicode data. Instead, specify the collation name and sorting rules to use. Character data is sorted using rules that define the correct character sequence, with options for specifying case-sensitivity, accent marks, kana character types, and character width. Microsoft SQL Server 2000 collations include these groupings:
   * Windows collations - Windows collations define rules for storing character data based on the rules defined for an associated Windows locale. The base Windows collation rules specify which alphabet or language is used when dictionary sorting is applied, as well as the code page used to store non-Unicode character data. For Windows collations, the **nchar**, **nvarchar**, and **ntext** data types have the same sorting behavior as **char**, **varchar**, and **text** data types
   * SQL collations - SQL collations are provided for compatibility with sort orders in earlier versions of Microsoft SQL Server.

**Sort Order**  
Binary is the fastest sorting order, and is case-sensitive. If **Binary** is selected, the **Case-sensitive**, **Accent-sensitive**, **Kana-sensitive**, and **Width-sensitive** options are not available.

**Sort order:Description**

**Binary**:Sorts and compares data in Microsoft® SQL Server™ tables based on the bit patterns defined for each character. Binary sort order is case-sensitive, that is lowercase precedes uppercase, and accent-sensitive. This is the fastest sorting order.   
If this option is not selected, SQL Server follows sorting and comparison rules as defined in dictionaries for the associated language or alphabet.

**Case-sensitive**:Specifies that SQL Server distinguish between uppercase and lowercase letters.   
If not selected, SQL Server considers the uppercase and lowercase versions of letters to be equal. SQL Server does not define whether lowercase letters sort lower or higher in relation to uppercase letters when Case-sensitive is not selected.

**Accent-sensitive**:Specifies that SQL Server distinguish between accented and unaccented characters. For example, 'a' is not equal to 'á'.   
If not selected, SQL Server considers the accented and unaccented versions of letters to be equal.

**Kana-sensitive**:Specifies that SQL Server distinguish between the two types of Japanese kana characters: Hiragana and Katakana.   
If not selected, SQL Server considers Hiragana and Katakana characters to be equal.

**Width-sensitive**:Specifies that SQL Server distinguish between a single-byte character (half-width) and the same character when represented as a double-byte character (full-width).   
If not selected, SQL Server considers the single-byte and double-byte representation of the same character to be equal.

Windows collation options:

* + Use **Latin1\_General** for the U.S. English character set (code page 1252).
  + Use **Modern\_Spanish** for all variations of Spanish, which also use the same character set as U.S. English (code page 1252).
  + Use **Arabic** for all variations of Arabic, which use the Arabic character set (code page 1256).
  + Use **Japanese\_Unicode** for the Unicode version of Japanese (code page 932), which has a different sort order from **Japanese**, but the same code page (932).

1. **What is the STUFF Function and how does it differ from the REPLACE function?**STUFF - Deletes a specified length of characters and inserts another set of characters at a specified starting point.   
   SELECT STUFF('abcdef', 2, 3, 'ijklmn')  
   GO  
   Here is the result set:  
   ---------   
   aijklmnef

REPLACE - Replaces all occurrences of the second given string expression in the first string expression with a third expression.  
SELECT REPLACE('abcdefghicde','cde','xxx')  
GO   
Here is the result set:   
------------  
abxxxfghixxx

1. **What does it mean to have quoted\_identifier on? What are the implications of having it off?**When SET QUOTED\_IDENTIFIER is OFF (default), literal strings in expressions can be delimited by single or double quotation marks.  
   When SET QUOTED\_IDENTIFIER is ON, all strings delimited by double quotation marks are interpreted as object identifiers. Therefore, quoted identifiers do not have to follow the Transact-SQL rules for identifiers.   
   SET QUOTED\_IDENTIFIER must be ON when creating or manipulating indexes on computed columns or indexed views. If SET QUOTED\_IDENTIFIER is OFF, CREATE, UPDATE, INSERT, and DELETE statements on tables with indexes on computed columns or indexed views will fail.   
   The SQL Server ODBC driver and Microsoft OLE DB Provider for SQL Server automatically set QUOTED\_IDENTIFIER to ON when connecting.  
   When a stored procedure is created, the SET QUOTED\_IDENTIFIER and SET ANSI\_NULLS settings are captured and used for subsequent invocations of that stored procedure. When executed inside a stored procedure, the setting of SET QUOTED\_IDENTIFIER is not changed.  
   SET QUOTED\_IDENTIFIER OFF  
   GO  
   -- Attempt to create a table with a reserved keyword as a name  
   -- should fail.  
   CREATE TABLE "select" ("identity" int IDENTITY, "order" int)  
   GO  
     
   SET QUOTED\_IDENTIFIER ON  
   GO  
   -- Will succeed.  
   CREATE TABLE "select" ("identity" int IDENTITY, "order" int)  
   GO
2. **What is the purpose of UPDATE STATISTICS?**Updates information about the distribution of key values for one or more statistics groups (collections) in the specified table or indexed view.
3. **Fundamentals of Data warehousing & olap?**
4. **What do u mean by OLAP server? What is the difference between OLAP and OLTP?**
5. **What is a tuple?**A **tuple** is an instance of data within a relational database.
6. **Services and user Accounts maintenance**
7. **sp\_configure commands?**Displays or changes global configuration settings for the current server.
8. **What is the basic functions for master, msdb, tempdb databases?**  
   Microsoft® SQL Server 2000 systems have four system databases:
   * **master** - The **master** database records all of the system level information for a SQL Server system. It records all login accounts and all system configuration settings. **master** is the database that records the existence of all other databases, including the location of the database files.
   * **tempdb** - **tempdb** holds all temporary tables and temporary stored procedures. It also fills any other temporary storage needs such as work tables generated by SQL Server. **tempdb** is re-created every time SQL Server is started so the system starts with a clean copy of the database.   
     By default, **tempdb** autogrows as needed while SQL Server is running. If the size defined for **tempdb** is small, part of your system processing load may be taken up with autogrowing **tempdb** to the size needed to support your workload each time to restart SQL Server. You can avoid this overhead by using ALTER DATABASE to increase the size of **tempdb**.
   * **model** - The **model** database is used as the template for all databases created on a system. When a CREATE DATABASE statement is issued, the first part of the database is created by copying in the contents of the **model** database, then the remainder of the new database is filled with empty pages. Because **tempdb** is created every time SQL Server is started, the **model** database must always exist on a SQL Server system.
   * **msdb** - The **msdb** database is used by SQL Server Agent for scheduling alerts and jobs, and recording operators.
9. **What are sequence diagrams? What you will get out of this sequence diagrams?**Sequence diagrams document the interactions between classes to achieve a result, such as a use case. Because UML is designed for object-oriented programming, these communications between classes are known as messages. The sequence diagram lists objects horizontally, and time vertically, and models these messages over time.
10. **What are the new features of SQL 2000 than SQL 7? What are the new datatypes in sql?**
    * XML Support - The relational database engine can return data as Extensible Markup Language (XML) documents. Additionally, XML can also be used to insert, update, and delete values in the database. (for xml raw - to retrieve output as xml type)
    * User-Defined Functions - The programmability of Transact-SQL can be extended by creating your own Transact-SQL functions. A user-defined function can return either a scalar value or a table.
    * Indexed Views - Indexed views can significantly improve the performance of an application where queries frequently perform certain joins or aggregations. An indexed view allows indexes to be created on views, where the result set of the view is stored and indexed in the database.
    * New Data Types - SQL Server 2000 introduces three new data types. **bigint** is an 8-byte integer type. **sql\_variant** is a type that allows the storage of data values of different data types. **table** is a type that allows applications to store results temporarily for later use. It is supported for variables, and as the return type for user-defined functions.
    * INSTEAD OF and AFTER Triggers - INSTEAD OF triggers are executed instead of the triggering action (for example, INSERT, UPDATE, DELETE). They can also be defined on views, in which case they greatly extend the types of updates a view can support. AFTER triggers fire after the triggering action. SQL Server 2000 introduces the ability to specify which AFTER triggers fire first and last.
    * Multiple Instances of SQL Server - SQL Server 2000 supports running multiple instances of the relational database engine on the same computer. Each computer can run one instance of the relational database engine from SQL Server version 6.5 or 7.0, along with one or more instances of the database engine from SQL Server 2000. Each instance has its own set of system and user databases.
    * Index Enhancements - You can now create indexes on computed columns. You can specify whether indexes are built in ascending or descending order, and if the database engine should use parallel scanning and sorting during index creation.
11. **How do we open SQL Server in single user mode?**We can accomplish this in any of the three ways given below :-
    1. From Command Prompt :-  
       sqlservr -m
    2. From Startup Options :-  
       Go to SQL Server Properties by right-clicking on the Server name in the Enterprise manager.   
       Under the 'General' tab, click on 'Startup Parameters'.   
       Enter a value of -m in the Parameter.
    3. From Registry :-  
       Go to HKEY\_LOCAL\_MACHINE\Software\Microsoft\MSSQLServer\MSSQLServer\Parameters.   
       Add new string value.   
       Specify the 'Name' as SQLArg(n) & 'Data' as -m.   
       Where n is the argument number in the list of arguments.
12. **Difference between clustering and NLB (Network Load Balancing)?**\*\*
13. **Explain Active/Active and Active/Passive cluster configurations?  
    \*\***
14. **What is Log Shipping?**In Microsoft® SQL Server™ 2000 Enterprise Edition, you can use log shipping to feed transaction logs from one database to another on a constant basis. Continually backing up the transaction logs from a source database and then copying and restoring the logs to a destination database keeps the destination database synchronized with the source database. This allows you to have a backup server and also provides a way to offload query processing from the main computer (the source server) to read-only destination servers.
15. **What are the main steps you take care for enhancing SQL Server performance?  
    \*\***
16. **You have to check whether any users are connected to sql server database and if any user is connected to database, you have to disconnect the user(s) and run a process in a job. How do you do the above in a job?**\*\*  
      
    **XML**
17. **How can I convert data in a Microsoft Access table into XML format?**The following applications can help you convert Access data into XML format: Access 2002, ADO 2.5, and SQLXML. Access 2002 (part of Microsoft Office XP) enables you to query or save a table in XML format. You might be able to automate this process. ADO 2.5 and later enables you to open the data into a recordset, then persist the recordset in XML format, as the following code shows:  
    rs.Save "c:\rs.xml", adPersistXML  
    You can use linked servers to add the Access database to your SQL Server 2000 database so you can run queries from within SQL Server to retrieve data. Then, through HTTP, you can use the SQLXML technology to extract the Access data in the XML format you want. **NEW**
18. **@@IDENTITY ?**Ans: Returns the last-inserted identity value.
19. **If a job is fail in sql server, how do find what went wrong?**
20. **Have you used Error handling in DTS?**

**1)How to find and delete duplicate records from a table?**

A:We may need the table design( DDL ) to identify that.But the general query to find out the duplicates is:

SELECT columnname, count(\*)

FROM dbo.tablename

GROUP BY columnname

HAVING count(\*) > 1

Different methods for identifying duplicates are discussed with examples in:

http://support.microsoft.com/default.aspx?scid=kb;en-us;Q139444

http://support.microsoft.com/default.aspx?scid=kb;EN-US;q70956

This article by Neil Boyle at swynk.com

This article at 15seconds.com

**2)How to generate the user defined store procedures list in a database?**

A:For SQL 2000 execute

select routine\_name as 'Stored Procedure Name'

from INFORMATION\_SCHEMA.ROUTINES

where routine\_type='procedure'

and objectproperty(object\_id(routine\_name),'IsMsShipped')=0

For SQL7.0 and below

SELECT name, USER\_NAME(uid) AS [Owner]

FROM sysobjects

WHERE type = 'P'

AND OBJECTPROPERTY(id, 'IsMSShipped') = 0

Note:This is needed only if you want the sql query, else Sqlserver Enterprise manager(stored procedures node) can be used.

**3)How to obtain the description of a column?**

A:When creating a table in the SQL2000 Enterprise manager its possible to add a description to each column of the table.To obtain that description we can use "fn\_listextendedproperty".

Column attributes are not supported by ANSI and since the structure of the INFORMATION\_SCHEMA views are defined by ANSI, the INFORMATION\_SCHEMA views do not show the column description . But you can join them with INFORMATION\_SCHEMA.COLUMNS as illustrated in this example....

CREATE TABLE testtable

(

column1 int,

column2 varchar(10)

)

EXEC sp\_addextendedproperty 'desc', 'column1 description',

'user', dbo, 'table', 'testtable', 'column', column1

EXEC sp\_addextendedproperty 'desc', 'column2 description',

'user', dbo, 'table', 'testtable', 'column', column2

SELECT \* FROM ::fn\_listextendedproperty (NULL, 'user', 'dbo', 'table', 'testtable',

'column', default)

SELECT

COLUMN\_NAME,

ORDINAL\_POSITION,

DATA\_TYPE,

name AS PROPERTY\_NAME,

value AS PROPERTY\_VALUE

FROM INFORMATION\_SCHEMA.COLUMNS AS C

JOIN ::fn\_listextendedproperty (NULL, 'user', 'dbo', 'table', 'testtable',

'column', default) AS E

ON C.COLUMN\_NAME = E.objname

WHERE TABLE\_NAME = 'testtable'

Also, fn\_listextendedproperty isnt exposed via SQL-DMO. Since description isnt a basic SQL Server shipping feature, DMO does not model that in the object. "Description" is added by the table editor using extend properties.Since its possible to add properties like that, there isnt any support for description nor are Extended Properties in general supported in SQL-DMO.

**4) How to schedule the scripting of database objects?**

A:We can use the script method of SQL-DMO...here is a sample....

Dim oSS As SQLDMO.SQLServer

Dim oDb As SQLDMO.Database

Dim oT As SQLDMO.Transfer

Dim sS As String

Sub Script()

Set oSS = New SQLDMO.SQLServer

Set oT = New SQLDMO.Transfer

oSS.Connect "server", "user", "password" 'Connect to the server

Set oDb = oSS.Databases("pubs") 'Use a DB

oT.CopyAllTables = True

oDb.ScriptTransfer oT, SQLDMOXfrFile\_SingleFile, "C:\pubs.sql"

End Sub

More information about SQL-DMO at:

http://msdn.microsoft.com/msdnmag/issues/01/05/sqldmo/sqldmo.asp

If its a SQL2000 ,we can also use SCPTXFR.exe utility in the upgrade folder to script things out.

The usage is explained in: DTS Does Not Copy Identity, Indexes, Primary Key or Other Constraints (Q220163)

**5) How to programmatically find the installation directory of SQL Server?**

A:There isnt a documented approach for that.Since these details are stored in the registry we can read using xp\_regread.

exec master..xp\_regread

'HKEY\_LOCAL\_MACHINE','SOFTWARE\Microsoft\MSSQLServer\Setup','SQLPath'

(or xp\_instance\_regread if on SQL Server 2000)

Note: xp\_regread and xp\_instance\* are undocumented calls...)

To get the path into a variable...do:

DECLARE @Installpath varchar(50)

exec master.dbo.xp\_regread @rootkey='HKEY\_LOCAL\_MACHINE',

@key='SOFTWARE\Microsoft\MSSQLServer\Setup',

@value\_name='SQLPath',@value=@Installpath output

select @Installpath

Also...in SQL2000 you can do:

declare @sqlpath sysname,

@datapath sysname

exec master..sp\_msget\_setup\_paths @sqlpath out, @datapath out

select @sqlpath, @datapath

Note:sp\_msget\_setup\_paths is a undocumented call.

**6) How to store images(JPEG,GIF...etc) in SQL server?**

A:We have to use the image datatype.To insert images we can use textcopy.exe, which is a utility to input or output image data into a table. It can be found in the Binn Directory of SQL Server installation. Of course, special functions can also be used to input data using the Text Functions and Statements in SQL Server, i.e. READTEXT, WRITETEXT, PTRVALID. There is a BII (Bulk Insert Image) utility in SQL2000 which is for bulk insert. We can also use ADO 2.6 stream object. There's a related article titled "Who's afraid of the big bad blob?" by Michael Otey in www.sqlmag.com.

Also...You might want to look at the LoadBlobPro Utility from TECA, Inc. This utility can import and export just about any PC file including: pictures and images,word docs, executable files, and more to and from the SQL Server's image field. It has an interactive viewer to preview files and images. Download available at teca.com

That said...if possible try to store the images in the filesystem than in the database .In the related column just store the file path which can be then used by the application to locate the file.This considerably reduces the database overhead the only disadvantage is that the application should have required permission to the local filesystem.

**7) How to check if a temp table exists already?**

A: We can do :

if object\_id('tempdb..#temptablename') is not null

begin

print '#temptablename exists!'

end

else

begin

print '#temptablename not exists!'

end

Or even the undocumented second arguement for OBJECT\_ID() as in:

if object\_id('tempdb..#temptablename','u') is not null

begin

print '#temptablename exists!'

end

else

begin

print '#temptablename not exists!'

end

**8) How to find the number of rows affected by bcp?**

A:Capture the bcp output to a temp table and parse it.

CREATE TABLE #cmdout (lin nvarchar(255))

INSERT INTO #cmdout (lin)

EXEC master..xp\_cmdshell 'bcp ...'

SELECT bcp\_rows = convert(int, substring(lin, 1, charindex(' ', lin)-1))

FROM #cmdout WHERE lin like '% rows copied.'

DROP TABLE #cmdout

**9) How to add a default constraint to an existing column**

**A:To add a default constraint to an existing column, use ALTER TABLE. For example:**

ALTER TABLE [TableName] ADD

CONSTRAINT [DefaultConstraintName] DEFAULT ('DefaultValue') FOR

[ColumnName]

**10) Records in one table but not in another?**

**A**:Consider this scenario....

In order to get all the employees in my employees table that are not in my layoff table. Empid is the common column.

Two methods...check the execution plan and adopt the best one.

SELECT E.\* FROM employees AS E WHERE

NOT EXISTS (SELECT L.\* FROM layoff AS L WHERE

E.EmpId = L.EmpId )

And here's an even faster way:

Select A.\* From employees E Left Outer Join layoff L On E.EmpId = L.EmpId

Where E.(Primarykey or other non-null column) Is Null

11)How to fire triggers in SQL 7 when using bcp?

A:SQL 7 BOL says:

' The DBCC statements have been redesigned to provide substantially improved performance. In addition, bulk copy operations now have the option to validate constraints and fire triggers as the data is loaded.'

It is an error in the documentation. MS had plans to support that in 7.0, but that didn't happen.

BOL for SQL Server 2000 suggests that you can use the FIRE\_TRIGGERS -h hint.

**12) How to break a column to two using SUBSTRING()?**

A:Consider this scenario...

A column called Empname in the format of 'last name space first name' (example. Smith John).

and you need to break it up in the form last name or first name.

If your empname column is >always< in this format, you can try:

SELECT lname = SUBSTRING(empname, 1, charindex(' ', empname) - 1),

fname = SUBSTRING(empname, charindex(' ', empname) + 1, 8000)

FROM tablename

**13) How to define constaints across databases?**

A:This functionality is not possible directly. You may need to write triggers to validate the data on inserts, updates, and deletes across databases.

**14) How to write a script that enumerates all user tables in a SQL Server database and then set the primary key in each table?**

A:For requirements of similar kind, you can generate the scripts by using T-SQL.Consider this example:

create table testtable1

(

ide int not null,

nme varchar(20)

)

create table testtable2

(

ide int not null,

nme varchar(20)

)

and so on, but with common column 'ide' defined as NOT NULL.Now you need to create a primary key on ide for all those tables.

One way would be of course to do that individually for all tables like:

alter table testtable1 add constraint pk\_testtable1 primary key clustered(ide)

But for a number of tables you can very well do:

select 'alter table '+name+' add constraint pk\_'+name+' primary key clustered(ide)'+char(13)+'GO'

from sysobjects

where type='U'

and objectproperty(id,'ismsshipped')=0

The output would be a script like:

alter table testtable1 add constraint pk\_testtable1 primary key clustered(ide)

GO

alter table testtable2 add constraint pk\_testtable2 primary key clustered(ide)

GO

------

------

Now you can just copy the script above, make the ncessary modifications, if needed and execute in Query analyzer.

**15) How to prefix zeroes to a integer?**

A:Supposing there is a integer field which stores values from 1 to let's say 99999.And you need to prefix zeroes like 00001,00011,00111,01111 etc then you can try :

declare @int int

set @int=1111

select RIGHT( '00000' + CAST( @int AS VARCHAR ), 5 )

**16) How to specify 'order by' on specific columns in a view.?**

A:Its not possible to use 'order by' in a view , however you can do :

Select \* from Viewname order by column1,column2

In SQL 7 and SQL 2000, you can use the TOP query:

create view Viewname

as

select top 100 percent

\* from tablename order by column1,column2

go

**17) How to pass an array to a stored procedure?**

A:Simple answer is you cant since SQL server doesnt have an array datatype.A workaround would be to use dynamic sql as illustrated in this example:

EXEC batch\_update '42,25,23,19,15,14,12,11,9,8,7,6', 2

Create Procedure batch\_update

@update\_list varchar,

@status INT

AS

UPDATE table1

SET table1\_status = @status

WHERE table1\_pk IN (@update\_list)

.

The above wont work, so you may have to use dynamic sql as in

ALTER PROCEDURE batch\_update

@update\_list VARCHAR,

@status INT

AS

BEGIN

EXEC('UPDATE table1 SET table1\_status='+@status+'WHERE table1\_pk

IN('+@update\_list+')')

END

There are many caveats associated with this approach.Please refer this brilliant article by Erland Sommarskog for details and other workarounds.

You can also investigate the use of XML for this functionality.I will shortly update this page with a XML case study.

**18) How to get the current date without the time?**

A:Basically you need to use the third parameter of CONVERT.See this example..

SELECT CONVERT(varchar(12), getdate() , 112)

You can also use 103,101 etc depending on the needs.Please refer 'CAST and CONVERT' topic in BooksOnLine.

**19) How to find the execution time of a stored procedure?**

A:The best way is to use SQL profiler. Just choose the required events.Else if you dont have the permission to access SQL Profiler or want to determine that programatically, then try:

DECLARE @StartDate datetime, @EndDate datetime

SET @StartDate = getdate()

/\*

--stored procedure code--

\*/

SET @EndDate = getdate()

SELECT datediff(ms,@StartDate, @EndDate) AS 'Milliseconds'

**20) How to write a alphanumeric query?**

A:Its possible to combine the two queries:

SELECT \* FROM tableName

WHERE colName LIKE '[Aa-Zz][Aa-Zz][Aa-Zz]'

SELECT \* FROM tablename

WHERE colName LIKE '[0-9][0-9][0-9]'

as in

SELECT \* FROM tablename

WHERE columnname LIKE '[A-Za-z0-9][A-Za-z0-9][A-Za-z0-9]'

21) How to create a system stored procedure that has type set to SYSTEM ?

A:Generally you may need to create some stored procedures and want them to behave just like system stored procedures.For example, a stored procedure to give more information about locks. Since you need to execute them from any database,just create that stored procedure in the master database with the name starting with sp\_ and make sure you assign correct permissions in the master database.

If you actually want it to show up as type set to SYSTEM in the Enterprise Manager (although there is really no added benefit of this) you can use the undocumented call sp\_MS\_ marksystemobject.

**22) How to find out the columns having index?**

A:There isnt any need of joins between sysobjects ,sysindexes and syscolumns system tables.

You can very well do:

sp\_helpindex tablename

In case you need to generate a listing of all indices for all tables then do

EXEC sp\_msforeachtable @command1= "PRINT '?' EXEC sp\_helpindex @objname =

'?'"

Please do note that sp\_msforeachtable is a undocumented call.

**23) How to do a SELECT inside a CASE expression?**

A: Consider this syntax:

SELECT column1, (CASE column2 WHEN 1

THEN (SELECT col3 FROM dbo.Table2 WHERE (condition))

ELSE 0 END)

FROM dbo.table1 WHERE (condition)

**24) How to to find the last product purchased from each vendor**?

A:Try these options:

select ProductName, VendorName, max(Purchasedate) as Purchasedate

from viewofproductandvendor

group by ProductName, VendorName

order by ProductName, VendorName

SELECT V1.\* FROM viewofproductandvendor AS V1

WHERE purchasedate = (SELECT MAX (purchasedate)

FROM viewofproductandvendor AS V2 WHERE V1.productid = V2.productid)

**25) How to sort data inside a string?**

**A:Supposing that one of** the columns contains data like:

0100

01A0

01AA

01Z0

and you need to sort it to

0100

01A0

01Z0

01AA

then you can write the query as:

select \*

from dbo.tablename

order by

substring (columnname, 3, 1),

substring (columnname, 4, 1)

**26) How to use PATINDEX to compare data?**

A:Consider this example...

CREATE TABLE testtabletesttablecompare

(

keyword VARCHAR(255)

)

GO

CREATE TABLE testtabletesttablecompare2

(

keyword VARCHAR(255),

status TINYINT DEFAULT 0

)

GO

INSERT INTO dbo.testtabletesttablecompare values('jan')

INSERT INTO dbo.testtabletesttablecompare values('janFeb')

INSERT INTO dbo.testtablecompare1 values('janFebMar')

INSERT INTO dbo.testtablecompare1 values('janFebMarApr')

INSERT INTO dbo.testtablecompare1 values('janFebMarAprMay')

INSERT INTO dbo.testtablecompare1 values('janFebMaraprMayJun')

GO

INSERT INTO dbo.testtablecompare2 values('Jan',default)

INSERT INTO dbo.testtablecompare2 values('Feb',default)

INSERT INTO dbo.testtablecompare2 values('Mar',default)

INSERT INTO dbo.testtablecompare2 values('Apr',default)

INSERT INTO dbo.testtablecompare2 values('May',default)

You need to update the ' status ' column in the table testtablecompare2 to 1 if any of its ' keyword ' column values is present in(part or whole) ' keyword ' column of the table testtablecompare1.

For that,we can write the query as....

UPDATE testtablecompare2

SET status = 1

FROM testtablecompare2 c2, testtablecompare1 c1

WHERE PATINDEX('%'+c2.keyword+'%', c1.keyword) > 0

**27) How to catch a OUTPUT parameter?**

A:Consider this example...

create procedure sumtestproc

(@number1 int, @number2 int, @result int output)

as

select @result = @number1 + @number2

return

Now while executing the stored procedure,you do something like

declare @total int

exec adder @num1 = 5, @num2 = 6, @result = @total

select @total

Here @total will return a value of NULL since you need to explicitly mention OUT parameter in the calling code also, as in:

declare @total int

exec dbo.sumtestproc @number1 = 5, @number2 = 6, @result = @total OUT

select @total

**28) How to round up a datetime value to the nearest hour?**

A: Couple of options...

declare @hour int

set @hour = datepart(hh, dateadd(ss, 3600-1, DateTimeColumn) )

Or

SELECT CASE WHEN DATEPART(mi,CONVERT(datetime,'09:12 AM')) != 0

THEN DATEPART(hh,DATEADD(hh,1,CONVERT(datetime, '09:12 AM')))

ELSE

DATEPART(hh,CONVERT(datetime,'09:12 AM'))

END AS 'closestHour'

**29) How to format a datetime value like " dd-mon-yy "?**

**A:Heres one way :-**

Select Replace(Convert(Varchar, GETDATE(), 6 ), ' ', '-')

**30) Which one is faster -- IN or OR?**

A:Typically there is a confusion regarding which is faster:

WHERE columnname IN(value1,value2,value3)

or

WHERE (columnname = value1 OR columnname = value2 OR columnname = value3)

Generally, SQL Server creates a same query plan for both. But it depends on the scenario of development and you may have to test it.

**31) How to manually insert a IDENTITY value?**

A:Consider this scenario...

CREATE TABLE identtest

( col1 INT IDENTITY

, col2 INT )

GO

INSERT identtest (col2) SELECT 1

INSERT identtest (col2) SELECT 3

INSERT identtest (col2) SELECT 4

GO

Now,if you need to manually insert a value of '0' for col1,you may need to do:

SET IDENTITY\_INSERT identtest ON

INSERT identtest (col1,col2 ) SELECT 0,2

SET IDENTITY\_INSERT identtest OFF

GO

You can also use DBCC CHECKIDENT to manually reseed the identity value.An example would be:

USE pubs

GO

DBCC CHECKIDENT (jobs, RESEED, 30)

GO

This example forces the current identity value in the jobs table to a value of 30.

**32) How to display count of rows as different set of columns?**

A:Lets take this example to illustrate the requirement....

CREATE TABLE Partstudy

(

Partnumber CHAR(5),

Teststatus CHAR (10),

Indate DATETIME

)

GO

INSERT INTO Partstudy VALUES ('A001', 'Pass', '01/20/02')

INSERT INTO Partstudy VALUES ('A002', 'Pass', '01/20/02')

INSERT INTO Partstudy VALUES ('A001', 'Fail', '01/20/02')

INSERT INTO Partstudy VALUES ('A004', 'Fail', '01/20/02')

INSERT INTO Partstudy VALUES ('A001', 'Okay', '02/11/02')

INSERT INTO Partstudy VALUES ('A002', 'Okay', '01/20/02')

INSERT INTO Partstudy VALUES ('A002', 'Fail', '02/11/02')

Now, you need a query to show count of each teststatus which is in turn grouped by entrydate, and each set of teststatus should be in seperate column.In that case, we can use CASE as in:

SELECT partNumber,

Passedones = SUM(CASE WHEN Teststatus = 'Pass' THEN 1 ELSE 0

end),

Failedones = SUM(CASE WHEN Teststatus = 'Fail' THEN 1 ELSE 0 end),

Okayones = SUM(CASE WHEN Teststatus = 'Okay' THEN 1 ELSE 0 end),

Indate

FROM dbo.Partstudy

GROUP BY partnumber, Indate

ORDER BY partnumber, Indate

**33) How to use the different DATE and TIME functions?**

A: Some common usages are discussed here....

To get name of the day from date :

SELECT DATENAME(weekday, getdate()) AS 'Day Name'

To get the week number :

SELECT DATEPART(wk,getdate()) AS 'Week Number'

To get date 7 days back :

SELECT DATEADD(dd, -7, GetDate())

**34) How and when to use SELF JOIN?**

A:Consider this example...

CREATE TABLE candytest

(kidId char(2),

candycolor varchar(10)

)

GO

INSERT INTO candytest SELECT 'K1', 'Yellow'

INSERT INTO candytest SELECT 'K1', 'Red'

INSERT INTO candytest SELECT 'K2', 'Red'

INSERT INTO candytest SELECT 'K2', 'Blue'

INSERT INTO candytest SELECT 'K3', 'White'

INSERT INTO candytest SELECT 'K3', 'Red'

INSERT INTO candytest SELECT 'K3', 'Yellow'

Go

Now to find those kidid's which has both yellow and red colored candies, we can write query as:

SELECT c1.kidid

FROM candytest AS c1 JOIN candytest AS c2

ON c1.candycolor = 'Red' AND c2.candycolor = 'Yellow'

AND c1.kidid = c2.kidid

**35) How to generate a rownumber for the query?**

A:If you need to generate row numbers like 1, 2, 3... in the result set retuned by query then you may have to construct the sequential

column in the query itself.An example:

SELECT \*

FROM (SELECT table1.\*, RowNumber = (SELECT COUNT(\*)

FROM dbo.tablename AS table2

WHERE table2.PrimKeycolumn <= table1.PrimKeycolumn)

FROM dbo.tablename AS table1) AS table3

Its also possible to create a temporary table with an IDENTITY column as in:

SELECT \*, RowNumber = IDENTITY (INT, 1, 1)

INTO #temptable

FROM dbo.tablename

But the above method shouldnt gurantee sequential entry always...so please note that.

There is also good information in this mskb article,

How to Dynamically Number Rows in a Select Statement (Q186133)

**36) How to insert carriage return?**

A:See this example...

CREATE TABLE testcarrreturn

(

ide INT,

val VARCHAR(20)

)

GO

INSERT testcarrreturn

SELECT 1,'block1 \

block2 \

block3 '

If you'd expect to output to be in the format:

ide val

----------- --------------------

1 block1 \

block2 \

block3

then you may to do the INSERT in this manner

INSERT testcarrreturn

SELECT 1, 'block1 \' + char(13) + char(10) + 'block2 \' + char(13) + char(10) +

'block3'

**37) How to use dynamic sql for a nvarchar of more than 4000 characters?**

A:There may be cases when you are executing a lengthy SQL statement via dynamic sql.Since varchar and nvarchar have limit on the number of characters,dynamic sql may fail. There is a workaround by means of concatenating the nvarchar or varchar variables as in:

DECLARE @string1 nvarchar(4000), @string2 nvarchar(4000)

SET @string1=REPLICATE('a',4000)

SET @string2='SELECT '

EXEC (@string2 + ''''+@string1+'''')

**38) How to avoid the appearance of 'n rows affected'?**

A:Though basically meant as informational, the apperance of this message after execution of sql queries can be troublesome.It can break some ASP scripts and also cause relevant sql jobs to run slowly.

You can avoid the appearance of that message by including SET NOCOUNT ON on the beginning as in:

SET NOCOUNT ON

SELECT \* FROM dbo.tablename

**39) How to convert a number, stored in seconds, to hours:minutes.?**

A: Couple of options, example, for 17600 seconds...

SELECT CONVERT( varchar , DATEADD( second , 17600 , '' ) , 108 )

If you don't care about seconds, then do:

SELECT CONVERT( varchar( 5 ) , DATEADD( second , 17600 , '' ) , 108 )

declare @ittime int

declare @ithour int

declare @itmin int

declare @itsec int

set @ittime = 17600

select @ithour = @ittime / 3600

select @itmin = (@ittime - (@ithour \* 3600)) / 60

select @itsec = (@ittime - (@ithour \* 3600)) - (@itmin \* 60)

select @ithour as hours ,@itmin as minutes ,@itsec as seconds

DECLARE @sec int

SET @sec = 17600

SELECT CONVERT(varchar,@sec / 60 / 60) + ':' +

CONVERT(varchar, ( @sec / 60 ) - ( @sec / 60 ) / 60 \* 60 ) AS

'Hours:Minutes'

**40) How to reset/clear a IDENTITY value?**

A: You can use either TRUNCATE TABLE or DBCC CHECKIDENT. Lets check this example...

SET NOCOUNT ON

CREATE TABLE identest

(

ide INT IDENTITY(1,1),

nme VARCHAR(20)

)

GO

INSERT identest SELECT 'name1'

INSERT identest SELECT 'name1'

INSERT identest SELECT 'name1'

INSERT identest SELECT 'name1'

INSERT identest SELECT 'name1'

GO

SELECT \* FROM identest

GO

The output for the SELECT is:

ide nme

----------- --------------------

1 name1

2 name1

3 name1

4 name1

5 name1

Now do a TRUNCATE TABLE and again INSERT some rows...

TRUNCATE TABLE identest

GO

INSERT identest SELECT 'name1'

INSERT identest SELECT 'name1'

GO

SELECT \* FROM identest

GO

Since the TRUNCATE is done, it automatically resets the IDENTITY value to start with 1.The second SELECT produces:

ide nme

----------- --------------------

1 name1

2 name1

You can also do DBCC CHECKIDENT as in:

DBCC CHECKIDENT (identest, RESEED,1)

GO

INSERT identest SELECT 'name1'

INSERT identest SELECT 'name1'

GO

That means the IDENTITY which was at 2 is reset to 1 and hence the next IDENTITY got a value of 2 and so on.

**41) SET or SELECT -- is there any performance difference?**

A:There isnt any performance difference.But SET is the standard assignment syntax.

For example, consider this subquery:

SET @variable = (SELECT columnvalue FROM dbo.tablename)

Here, if the results of the subquery expression is empty then @variable has a value of NULL but

if results of the subquery are more than one row then you get a error.And thus it makes sense while

SELECT @variable = columnvalue FROM dbo.tablename

returns a value from a number of rows and we are not sure whats the criteria in selecting that row!

Also, for ease of programming SELECT is anyways used as SQL syntax to retrieve rows.Thus it makes it easier to have SET for assigning.One major difference in assigning using SET and SELECT is:

SET @variable1 = 'somevalue', @variable2 = 'someothervalue'

This is NOT possible but the same is possible with SELECT as in:

SELECT @variable1 = 'somevalue', @variable2 = 'someothervalue'

42) How to SELECT nth row from a table?

A:On all Relational databases there isnt any ordering for rows returned.From SQL Server7

above you can use the TOP operator.Assuming you want to select the sixth row from the resultset, you can do something like....

USE pubs

GO

SELECT TOP 1 au\_id, au\_lname

FROM authors

WHERE au\_id not in (SELECT TOP 5 au\_id FROM authors ORDER BY au\_id)

ORDER BY au\_id

**43) Is there a performance difference between CHAR and VARCHAR?**

**A:Potentially yes**, but it is difficult to predict without analysing the character data distribution in the table.But that shouldnt be of a big concern since even if there is a difference between the two, it shouldnt matter much with respect to the performance.

The space utilization can be managed well if the use of CHAR and VARCHAR are managed properly. Consistently if you find that a column of CHAR(20) is having a maximum data length of only 5 bytes ,then it makes sense to declare it as VARCHAR(20) so that only 5 bytes are stored.Though performance wise, internal data storage has a least preference compared to how the coding or the database design is done.

Another area where CHAR and VARCHAR differs is in the treatment of trailing blanks.

**44) How to SELECT combined DISTINCT values from two tables?**

A:Consider this example...

CREATE TABLE #dist\_test1

(

ide INT,

nme VARCHAR(20)

)

GO

CREATE TABLE #dist\_test2

(

ide2 INT,

nme2 VARCHAR(20)

)

GO

INSERT #dist\_test1 VALUES(1,'name1')

INSERT #dist\_test1 VALUES(2,'name1')

INSERT #dist\_test1 VALUES(3,'name1')

INSERT #dist\_test1 VALUES(3,'name3')

INSERT #dist\_test1 VALUES(1,'name4')

GO

INSERT #dist\_test2 VALUES(1,'name1')

INSERT #dist\_test2 VALUES(2,'name2')

INSERT #dist\_test2 VALUES(3,'name1')

INSERT #dist\_test2 VALUES(3,'name1')

INSERT #dist\_test2 VALUES(1,'name4')

GO

Now you need a SELECT query which returns the DISTINCT values from the two tables as in:

nme

--------------------

name1

name2

name3

name4

Here you can use the UNION to combine the DISTINCT values

SELECT nme

FROM #dist\_test1

UNION

SELECT nme2

FROM #dist\_test2

**45) How to use CASE for ORDER BY clause?**

A:Consider this situation where you need to ORDER BY a SELECT statement ,but the ORDER BY columns may vary and is supplied as a parameter to a stored procedure.See this example:

CREATE PROCEDURE dbo.titleinfo ( @colmn\_name SYSNAME)

AS

SELECT \*

FROM pubs.dbo.titles

ORDER BY

CASE @colmn\_name

WHEN 'title\_id' THEN title\_id

WHEN 'type' THEN type

ELSE Null

END

To execute the stored procedure supply the relevant column name as parameter and the results appear as ORDERed BY that column.

EXEC dbo.titleinfo 'type'