SQL INTERVIEW QUESTIONS

&

ANSWERS

MOST FREQUENTLY ASKED SQL INTERVIEW QUESTIONS

1. What is the difference between "Stored Procedure" and "Function"?

1. A procedure can have both input and output parameters, but a function can only have input parameters.
2. Inside a procedure we can use DML (INSERT/UPDATE/DELETE) statements. But inside a function we can't use DML statements.
3. We can't utilize a Stored Procedure in a Select statement. But we can use a function in a Select statement.
4. We can use a Try-Catch Block in a Stored Procedure but inside a function we can't use a Try-Catch block.
5. We can use transaction management in a procedure but we can't in a function.
6. We can't join a Stored Procedure but we can join functions.
7. Stored Procedures cannot be used in the SQL statements anywhere in the WHERE/HAVING/SELECT section. But we can use a function anywhere.
8. A procedure can return 0 or n values (max 1024). But a function can return only 1 value that is mandatory.
9. A procedure can't be called from a function but we can call a function from a procedure.

2.What is difference between "Clustered Index" and "Non Clustered Index"?

1. A Clustered Index physically stores the data of the table in the order of the keys values and the data is resorted every time whenever a new value is inserted or a value is updated in the column on which it is defined, whereas a non-clustered index creates a separate list of key values (or creates a table of pointers) that points towards the location of the data in the data pages.
2. A Clustered Index requires no separate storage than the table storage. It forces the rows to be stored sorted on the index key whereas a non-clustered index requires separate storage than the table storage to store the index information.
3. A table with a Clustered Index is called a Clustered Table. Its rows are stored in a B-Tree structure sorted whereas a table without any clustered indexes is called a non-clustered table. Its rows are stored in a heap structure unsorted.
4. The default index is created as part of the primary key column as a Clustered Index.
5. In a Clustered Index, the leaf node contains the actual data whereas in a non-clustered index, the leaf node contains the pointer to the data rows of the table.
6. A Clustered Index always has an Index Id of 1 whereas non-clustered indexes have Index Ids > 1.
7. A Table can have **only** 1 Clustered Index whereas prior to SQL Server 2008 only 249 non-clustered indexes can be created. With SQL Server 2008 and above 999 non-clustered indexes can be created.
8. A Primary Key constraint creates a Clustered Index by default whereas A Unique Key constraint creates a non-clustered index by default.

3.What is the difference between the "DELETE" and "TRUNCATE" commands?

1. The DELETE command is used to remove rows from a table based on a WHERE condition whereas TRUNCATE removes all rows from a table.
2. So we can use a where clause with DELETE to filter and delete specific records whereas we cannot use a Where clause with TRUNCATE.
3. DELETE is executed using a row lock, each row in the table is locked for deletion whereas TRUNCATE is executed using a table lock and the entire table is locked for removal of all records.
4. DELETE is a DML command whereas TRUNCATE is a DDL command.
5. DELETE retains the identity of the column value whereas in TRUNCATE, the Identify column is reset to its seed value if the table contains any identity column.
6. To use Delete you need DELETE permission on the table whereas to use Truncate on a table you need at least ALTER permission on the table.
7. DELETE uses more transaction space than the TRUNCATE statement whereas Truncate uses less transaction space than DELETE statement.
8. DELETE can be used with indexed views whereas TRUNCATE cannot be used with indexed views.
9. The DELETE statement removes rows one at a time and records an entry in the transaction log for each deleted row whereas TRUNCATE TABLE removes the data by deallocating the data pages used to store the table data and records only the page deallocations in the transaction log.
10. Delete activates a trigger because the operation is logged individually whereas TRUNCATE TABLE can't activate a trigger because the operation does not log individual row deletions.

4.What is the difference between the "WHERE" clause and the "HAVING" clause?

1. WHERE clause can be used with a Select, Update and Delete Statement Clause but the HAVING clause can be used only with a Select statement.
2. We can't use an aggregate functions in the WHERE clause unless it is in a sub-query contained in a HAVING clause whereas we can use an aggregate function in the HAVING clause. We can use a column name in the HAVING clause but the column must be contained in the group by clause.
3. WHERE is used before the GROUP BY clause whereas a HAVING clause is used to impose a condition on the GROUP Function and is used after the GROUP BY clause in the query.
4. A WHERE clause applies to each and every row whereas a HAVING clause applies to summarized rows (summarized with GROUP BY).
5. In the WHERE clause the data that is fetched from memory depending on a condition whereas in HAVING the completed data is first fetched and then separated depending on the condition.

5.What is the difference between "Primary Key" and "Unique Key"?

1. We can have only one Primary Key in a table whereas we can have more than one Unique Key in a table.
2. The Primary Key cannot have a NULL value whereas a Unique Key may have only one null value.
3. By default, a Primary Key is a Clustered Index whereas by default, a Unique Key is a unique non-clustered index.
4. A Primary Key supports an Auto Increment value whereas a Unique Key doesn't support an Auto Increment value.

6. What is the difference between a "Local Temporary Table" and "Global Temporary Table"?

1. A Local Temporary Table is created by giving it a prefix of # whereas a Global Temporary Table is created by giving it a prefix of ##.
2. A Local Temporary Table cannot be shared among multiple users whereas a Global Temporary Table can be shared among multiple users.
3. A Local Temporary Table is only available to the current DB connection for the current user and are cleared when the connection is closed whereas a Global Temporary Table is available to any connection once created. They are cleared when the last connection is closed.

7.What are super, primary, candidate and foreign keys?
Ans: A **super key** is a set of attributes of a relation schema upon which all attributes of the schema are functionally dependent. No two rows can have the same value of super key attributes.
A **Candidate key** is minimal super key, i.e., no proper subset of Candidate key attributes can be a super key.
A **Primary Key** is one of the candidate keys. One of the candidate keys is selected as most important and becomes the primary key. There cannot be more that one primary keys in a table.
Foreign key is a field (or collection of fields) in one table that uniquely identifies a row of another table.

8.What is the difference between primary key and unique constraints?
Ans: Primary key cannot have NULL value, the unique constraints can have NULL values. There is only one primary key in a table, but there can be multiple unique constrains.

9.What is database normalization?
Ans: It is a process of analyzing the given relation schemas based on their functional dependencies and primary keys to achieve the following desirable properties:
1)Minimizing Redundancy
2) Minimizing the Insertion, Deletion, And Update Anomalies
Relation schemas that do not meet the properties are decomposed into smaller relation schemas that could meet desirable properties.

10.What is SQL?
SQL is Structured Query Language designed for inserting and modifying in a relational database management system.

11.What are the differences between DDL, DML and DCL in SQL?
Ans: Following are some details of three.
DDL stands for Data Definition Language. SQL queries like CREATE, ALTER, DROP and RENAME come under this.
DML stands for Data Manipulation Language. SQL queries like SELECT, INSERT and UPDATE come under this.
DCL stands for Data Control Language. SQL queries like GRANT and REVOKE come under this.

12.What is the difference between having and where clause?
Ans: HAVING is used to specify a condition for a group or an aggregate function used in select statement. The WHERE clause selects before grouping. The HAVING clause selects rows after grouping. Unlike HAVING clause, the WHERE clause cannot contain aggregate functions.

13.What is Join?
Ans: An SQL Join is used to combine data from two or more tables, based on a common field between them. For example, consider the following two tables.

Student Table

EnrollNo StudentName Address
1000   geek1   geeksquiz1
1001   geek2   geeksquiz2
1002   geek3   geeksquiz3

StudentCourse Table

CourseID EnrollNo

1   1000

2   1000

3   1000

1   1002

2   1003

Following is join query that shows names of students enrolled in different courseIDs.

SELECT StudentCourse.CourseID, Student.StudentName
    FROM StudentCourse
    INNER JOIN Customers
    ON StudentCourse.EnrollNo = Student.EnrollNo
    ORDER BY StudentCourse.CourseID;

The above query would produce following result.

CourseID StudentName

1   geek1

1   geek2

2   geek1

2   geek3

3   geek1

14.What is Identity?

Ans: Identity (or AutoNumber) is a column that automatically generates numeric values. A start and increment value can be set, but most DBA leave these at 1. A GUID column also generates numbers; the value of this cannot be controlled. Identity/GUID columns do not need to be indexed.

15.What is a view in SQL? How to create one

Ans: A view is a virtual table based on the result-set of an SQL statement. We can create using create view syntax.

CREATE VIEW view_name AS
SELECT column_name(s)
FROM table_name
WHERE condition

16..What are the uses of view?

1. Views can represent a subset of the data contained in a table; consequently, a view can limit the degree of exposure of the underlying tables to the outer world: a given user may have permission to query the view, while denied access to the rest of the base table.

2. Views can join and simplify multiple tables into a single virtual table
3. Views can act as aggregated tables, where the database engine aggregates data (sum, average etc.) and presents the calculated results as part of the data
4. Views can hide the complexity of data; for example a view could appear as Sales2000 or Sales2001, transparently partitioning the actual underlying table.
5. Depending on the SQL engine used, views can provide extra security.

17.What is a Trigger?
Ans: A Trigger is a code that associated with insert, update or delete operations. The code is executed automatically whenever the associated query is executed on a table. Triggers can be useful to maintain integrity in database.

18.What is a stored procedure?
Ans: A stored procedure is like a function that contains a set of operations compiled together. It contains a set of operations that are commonly used in an application to do some common database tasks.

19.What is the difference between Trigger and Stored Procedure?
Ans: Unlike Stored Procedures, Triggers cannot be called directly. They can only be associated with queries.

20.What is a transaction? What are ACID properties?
Ans: A Database Transaction is a set of database operations that must be treated as whole, means either all operations are executed or none of them.
An example can be bank transaction from one account to another account. Either both debit and credit operations must be executed or none of them.
ACID (Atomicity, Consistency, Isolation, Durability) is a set of properties that guarantee that database transactions are processed reliably.

21.What are indexes?
Ans: A database index is a data structure that improves the speed of data retrieval operations on a database table at the cost of additional writes and the use of more storage space to maintain the extra copy of data.
Data can be stored only in one order on disk. To support faster access according to different values, faster search like binary search for different values is desired, For this purpose, indexes are created on tables. These indexes need extra space on disk, but they allow faster search according to different frequently searched values.

22.What are clustered and non-clustered Indexes?
Ans: Clustered indexes is the index according to which data is physically stored on disk. Therefore, only one clustered index can be created on a given database table.
Non-clustered indexes don't define physical ordering of data, but logical ordering. Typically, a tree is created whose leaf point to disk records. B-Tree or B+ tree are used for this purpose.

23.What are Primary Keys and Foreign Keys?
Ans: Primary keys are the unique identifiers for each row. They must contain unique values and cannot be null. Due to their importance in relational databases, Primary keys are the most fundamental aspect of all keys and constraints. A table can have only one primary key. Foreign keys are a method of ensuring data integrity and manifestation of the relationship between tables.

24.What is SQL ?

Ans: Structured Query Language(SQL) is a language designed specifically for communicating with databases. SQL is an ANSI (American National Standards Institute) standard .

25. What are the different type of SQL or different commands in SQL?
Ans: Frequently asked SQL Interview Questions

1.DDL – Data Definition Language.DDL is used to define the structure that holds the data.

2.DML– Data Manipulation Language
DML is used for manipulation of the data itself. Typical operations are Insert, Delete,Update and retrieving the data from the table

3.DCL–DataControlLanguage
DCL is used to control the visibility of data like granting database access and set privileges to create table etc.

4.TCL-TransactionControl Language
 It contains

26. What are the Advantages of SQL?

1. **SQL is not a proprietary language** used by specific database vendors. Almost every major DBMS supports SQL, so learning this one language will enable programmer to interact with any database like ORACLE, SQL ,MYSQL etc.

2. **SQL is easy to learn**. The statements are all made up of descriptive English words, and there aren't that many of them.

3. SQL is actually a very powerful language and by using its language elements you can perform very **complex and sophisticated database operations**.

27. what is a field in a database ?

A field is an area within a record reserved for a specific piece of data.

Examples: Employee Name , Employee ID etc

28.What is a Record in a database ?

A record is the collection of values / fields of a specific entity: i.e. a Employee , Salary etc.

29. What is a Table in a database ?

A table is a collection of records of a specific type. For example, employee table , salary table etc.

30.What is a database transaction?

Database transaction take database from one consistent state to another. At the end of the transaction the system must be in the prior state if transaction fails or the status of the system should reflect the successful completion if the transaction goes through.

31.What are properties of a transaction?

Properties of the transaction can be summarized as ACID Properties.
1. Atomicity
A transaction consists of many steps. When all the steps in a transaction gets completed, it will get reflected in DB or if any step fails, all the transactions are rolled back.

2. Consistency
The database will move from one consistent state to another, if the transaction succeeds and remain in the original state, if the transaction fails.

3. Isolation
Every transaction should operate as if it is the only transaction in the system

4. Durability
Once a transaction has completed successfully, the updated rows/records must be available for all other transactions on a permanent basis

32.What is a Database Lock ?

Database lock tell a transaction, if the data item in questions is currently being used by other transactions.

33. What are the type of locks ?

1. Shared Lock
When a shared lock is applied on data item, other transactions can only read the item, but can't write into it.
2. Exclusive Lock
When a exclusive lock is applied on data item, other transactions can't read or write into the data item.

34.What are the different type of normalization?

Frequently asked SQL Interview Questions

In database design , we start with one single table, with all possible columns. Lot of redundant data would be present since it's a single table. **The process of removing the redundant data, by splitting up the table in a well defined fashion is called normalization.**

1. First Normal Form (1NF)
A relation is said to be in first normal form if and only if all underlying domains contain atomic values only. After 1NF, we can still have redundant data

2. Second Normal Form (2NF)
A relation is said to be in 2NF if and only if it is in 1NF and every non key attribute is fully dependent on the primary key. After 2NF, we can still have redundant data

3. Third Normal Form (3NF)
A relation is said to be in 3NF, if and only if it is in 2NF and every non key attribute is non-transitively dependent on the primary key

35.What is a primary key?

Frequently asked SQL Interview Questions
A primary key is a column whose values **uniquely identify every row** in a table. Primary key values can never be reused. If a row is deleted from the table, its primary key may not be assigned to any new rows in the future. To define a field as primary key, following conditions had to be met :

1. No two rows can have the same primary key value.
2. Every row must have a primary key value
3. Primary key field cannot be null
4. Values in primary key columns can never be modified or updated

36.What is a Composite Key ?

A Composite primary key is a type of candidate key, which represents a set of columns whose values uniquely identify every row in a table.

For example - if "Employee_ID" and "Employee Name" in a table is combined to uniquely identifies a row its called a Composite Key.

37.What is a Composite Primary Key ?
A Composite primary key is a set of columns whose values uniquely identify every row in a table. What it means is that, table which contains composite primary key will be indexed based on columns specified in the primary key. This key will be referred in Foreign Key tables.
For example - if combined effect of columns, "Employee_ID" and "Employee Name" in a table is required to uniquely identifies a row, its called a Composite Primary Key. In this case, both the columns will be represented as primary key.

38.What is a Foreign Key ?

Frequently asked SQL Interview Questions
When a "one" table's primary key field is added to a related "many" table in order to create the common field which relates the two tables, it is called a foreign key in the "many" table.

For example, salary of an employee is stored in salary table. Relation is established via foreign key column "Employee_ID_Ref" which refers "Employee_ID" field in Employee table .

39. What is a Unique Key ?

Unique key is same as primary with difference being existence of null. Unique key field allows one value as NULL value.

40.Define SQL Insert Statement ?

SQL INSERT statement is used to add rows to a table. For a full row insert , SQL Query should start with "insert into " statement followed by table name and values command, followed by the values that need to be inserted into the table. Insert can be used in several ways:
1. To insert a single complete row
2. To insert a single partial row

41. Define SQL Update Statement ?

SQL Update is used to update data in a row or set of rows specified in the filter condition.

The basic format of an SQL UPDATE statement is ,Update command followed by table to be updated and SET command followed by column names and their new values followed by filter condition that determines which rows should be updated

42. Define SQL Delete Statement ?

SQL Delete is used to delete a row or set of rows specified in the filter condition.

The basic format of an SQL DELETE statement is, DELETE FROM command followed by table name followed by filter condition that determines which rows should be updated.

43.What are wild cards used in database for Pattern Matching ?

SQL Like operator is user for pattern matching. SQL 'Like' command takes more time to process. So before using like operator, consider suggestions given below on when and where to use wild card search.
1) Don't overuse wild cards. If another search operator will do, use it instead.
2) When you do use wild cards, try not to use them at the beginning of the search pattern, unless absolutely necessary. Search patterns that begin with wild cards are the slowest to process.
3) Pay careful attention to the placement of the wild card symbols. If they are misplaced, you might not return the data you intended

44. Define Join and explain different type of joins?

In order to avoid data duplication, data is stored in related tables . **Join** keyword is used to fetch data from related table. Join return rows when there is at least one match in both tables . Type of joins are

Right Join
Return all rows from the right table, even if there are no matches in the left table .
Outer Join
Left Join
Return all rows from the left table, even if there are no matches in the right table .
Full Join
Return rows when there is a match in one of the tables .

45. What is Self-Join?

Self-join is query used to **join a table to itself**. Aliases should be used for the same table comparison.

46. What is Cross Join?

Cross Join will return all records where each row from the first table is combined with each row from the second table.

SQL Interview Questions and answers on Database Views

47. What is a view?

Views are virtual tables. Unlike tables that contain data, views simply contain queries that dynamically retrieve data when used.

48. What is a materialized view?

Materialized views is also a view but are disk based. **Materialized views** get updated on specific duration, base upon the interval specified in the query definition. We can index materialized view.

49. What are the advantages and disadvantages of views in a database?

Advantages:
1. Views doesn't store data in a physical location.
2. View can be use to hide some of the columns from the table
3. Views can provide Access Restriction, since data insertion, update and deletion is not possible on the view.

Disadvantages:
1. When a table is dropped, associated view become irrelevant.
2. Since view are created when a query requesting data from view is triggered, its bit slow
3. When views are created for large tables, it occupy more memory .

SQL Interview Questions and answers on Stored Procedures and Triggers

50. What is a stored procedure?

Stored Procedure is a function which contain collection of SQL Queries. Procedure can take inputs , process them and send back output.

51.What are the advantages a stored procedure?

Stored Procedures are pre-complied and stored in database. This enable the database to execute the queries much faster. Since many queries can be included in a stored procedure, round trip time to execute multiple queries from source code to database and back is avoided.

52.What is a trigger?

Database are set of commands that get executed when an event(Before Insert, After Insert, On Update, On delete of a row) occurs on a table, views.

53. Explain the difference between DELETE , TRUNCATE and DROP commands?

Once **delete operation** is performed, Commit and Rollback can be performed to retrieve data.
Once **truncate** statement is executed, Commit and Rollback statement cant be performed. Where condition can be used along with delete statement but it cant be used with truncate statement.
Drop command is used to drop the table or keys like primary,foreign from a table.

54. What is the difference between Cluster and Non cluster Index?

A **clustered index** reorders the way records in the table are physically stored. There can be only one clustered index per table. It make data retrieval faster.

A **non clustered index** does not alter the way it was stored but creates a complete separate object within the table. As a result insert and update command will be faster.

55. What is Union, minus and Interact commands?

MINUS operator is used to return rows from the first query but not from the second query. INTERSECT operator is used to return rows returned by both the queries.

56.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 non clustered index by default. Another major difference is that, primary key doesn't allow NULLs, but unique key allows one NULL only.

57. 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.

58.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.

59.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). Until 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.

60.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.

61. 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.

62. 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.

63.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.

64.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.

65.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".

66. 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 clustering 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.

67. 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.

68. 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.

69.What is 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 de-allocation of the data pages of the table, which makes it faster. Of course, TRUNCATE TABLE can be rolled back.

70. Explain the storage models of OLAP

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

71. 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.

72. 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"

73. 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 createe 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 time, 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.

74. 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.

75. 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.

76. 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, database 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.

77. 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 live lock 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 live lock also occurs when read transactions monopolize a table or page, forcing a write transaction to wait indefinitely.

78. 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.

79. 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 file groups, 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.

80. 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.

81. 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.

82. 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

83. 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.

84. Explain different types of BACKUPs available in SQL Server? Given a particular scenario, how would you go about choosing a backup plan?
Types of backups you can create in SQL Server 7.0+ are Full database backup, differential database backup, transaction log backup, file group 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.

85.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:
o Snapshot replication
o Transactional replication (with immediate updating subscribers, with queued updating subscribers)
o Merge replication

86. 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

87.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. 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.

88. 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]]

89. 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.

90. Can you have a nested transaction?

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

91. 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_OA Create stored procedure.

92. 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().

93. 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.

94. 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.

95. 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
```

96.Define Fact tables and dimension tables.

Fact tables are central tables in data warehousing. They contain the aggregate values that are used in business process.

97.Explain the ETL process in Data warehousing.

Extraction, Transformation and loading are different stages in data warehousing.

98.What is Data mining?

Data mining is a process of analyzing current data and summarizing the information in more useful manner.

99.What are indexes?

Index can be thought as index of the book that is used for fast retrieval of information. Index uses one or more column index keys and pointers to the record to locate record.

100.Explain the types of indexes?

Clustered index
Non-clustered.

101.Define SQL.

SQL stands for Structured Query Language. It allows access, insert/update/delete records and retrieve data from the database.

102.What is RDBMS? Explain its features.

RDBMS stands for Relational Database Management System. It organizes data into related rows and columns.

103.What is an Entity-Relationship diagram?

It is a graphical representation of tables with the relationship between them.
Read answer

104.Define referential integrity.

It is the rules that are applied when the relationships are created. It ensures integrity of data and prevents inconsistent data into the tables.

105.Define Primary key and Foreign key.

A column or combination of columns that identify a row of data in a table is Primary Key.

106.Define alternate key.

There can be a key apart from primary key in a table that can also be a key. This key may or may not be a unique key.

107.Delete vs. Truncate table.

Delete logs the deletion of each row whereas Truncate doesn't log deleted rows in the transaction log. This makes truncate command is bit faster than Delete command.

108.Define constraints.

Constraints enforce integrity of the database. Constraints can be of following types

Not Null
Check
Unique
Primary key
Foreign key

109.Define stored procedure. Stored procedure is a set of pre-compiled SQL statements, executed when it is called in the program.

110.Define Trigger.

Triggers are similar to stored procedure except it is executed automatically when any operations are occurred on the table.

111. What is RDBMS?

Relational Data Base Management Systems (RDBMS) are database management systems that maintain data records and indices in tables. Relationships may be created and maintained across and among the data and tables. In a relational database, relationships between data items are expressed by means of tables. Interdependencies among these tables are expressed by data values rather than by pointers. This allows a high degree of data independence. An RDBMS has the capability to recombine the data items from different files, providing powerful tools for data usage.

112. What are the properties of the Relational tables?

Relational tables have six properties:

1. Values are atomic.
2. Column values are of the same kind.
3. Each row is unique.
4. The sequence of columns is insignificant.
5. The sequence of rows is insignificant.
6. Each column must have a unique name.

113. What is Normalization?

Database normalization is a data design and organization process applied to data structures based on rules that help building relational databases. In relational database design, the process of organizing data to minimize redundancy is called normalization. Normalization usually involves dividing a database into two or more tables and defining relationships between the tables. The objective is to isolate data so that additions, deletions, and modifications of a field can be made in just one table and then propagated through the rest of the database via the defined relationships.

114. What is De-normalization?

De-normalization is the process of attempting to optimize the performance of a database by adding redundant data. It is sometimes necessary because current DBMSs implement the relational model poorly. A true relational DBMS would allow for a fully normalized database at the logical level, while providing physical storage of data that is tuned for high performance. De-normalization is a technique to move from higher to lower normal forms of database modeling in order to speed up database access.

115. What are different normalization forms?

1. **1NF: Eliminate Repeating Groups** Make a separate table for each set of related attributes, and give each table a primary key. Each field contains at most one value from its attribute domain.

2. **2NF: Eliminate Redundant Data** If an attribute depends on only part of a multi-valued key, remove it to a separate table.
3. **3NF: Eliminate Columns Not Dependent On Key** If attributes do not contribute to a description of the key, remove them to a separate table. All attributes must be directly dependent on the primary key.
4. **BCNF: Boyce-Codd Normal Form** If there are non-trivial dependencies between candidate key attributes, separate them out into distinct tables.
5. **4NF: Isolate Independent Multiple Relationships** No table may contain two or more 1:n or n:m relationships that are not directly related.
6. **5NF: Isolate Semantically Related Multiple Relationships** There may be practical constrains on information that justify separating logically related many-to-many relationships.
7. **ONF: Optimal Normal Form** A model limited to only simple (elemental) facts, as expressed in Object Role Model notation.
8. **DKNF: Domain-Key Normal Form** A model free from all modification anomalies is said to be in DKNF.

Remember, these normalization guidelines are cumulative. For a database to be in 3NF, it must first fulfill all the criteria of a 2NF and 1NF database.

116. What is Stored Procedure?

A stored procedure is a named group of SQL statements that have been previously created and stored in the server database. Stored procedures accept input parameters so that a single procedure can be used over the network by several clients using different input data. And when the procedure is modified, all clients automatically get the new version. Stored procedures reduce network traffic and improve performance. Stored procedures can be used to help ensure the integrity of the database.
e.g. sp_helpdb, sp_renamedb, sp_depends etc.

117. What is Trigger?

A trigger is a SQL procedure that initiates an action when an event (INSERT, DELETE or UPDATE) occurs. Triggers are stored in and managed by the DBMS. Triggers are used to maintain the referential integrity of data by changing the data in a systematic fashion. A trigger cannot be called or executed; DBMS automatically fires the trigger as a result of a data modification to the associated table. Triggers can be viewed as similar to stored procedures in that both consist of procedural logic that is stored at the database level. Stored procedures, however, are not event-drive and are not attached to a specific table as triggers are. Stored procedures are explicitly executed by invoking a CALL to the procedure while triggers are implicitly executed. In addition, triggers can also execute stored procedures.

118. What is Nested Trigger?

A trigger can also contain INSERT, UPDATE and DELETE logic within itself, so when the trigger is fired because of data modification it can also cause another data modification, thereby firing another trigger. A trigger that contains data modification logic within itself is called a nested trigger.

119. What is View?

A simple view can be thought of as a subset of a table. It can be used for retrieving data, as well as updating or deleting rows. Rows updated or deleted in the view are updated or deleted in the table the view was created with. It should also be noted that as data in the original table changes, so does data in the view, as views are the way to look at part of the original table. The results of using a view are not permanently stored in the database. The data accessed through a view is actually constructed using standard T-SQL select command and can come from one to many different base tables or even other views.

120. What is Index?

An index is a physical structure containing pointers to the data. Indices are created in an existing table to locate rows more quickly and efficiently. It is possible to create an index on one or more columns of a table, and each index is given a name. The users cannot see the indexes; they are just used to speed up queries. Effective indexes are one of the best ways to improve performance in a database application. A table scan happens when there is no index available to help a query. In a table scan SQL Server examines every row in the table to satisfy the query results. Table scans are sometimes unavoidable, but on large tables, scans have a terrific impact on performance.

121. What is a Linked Server?

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. With a linked server, you can create very clean, easy to follow, SQL statements that allow remote data to be retrieved, joined and combined with local data. Stored Procedure sp_addlinkedserver, sp_addlinkedserverlogin will be used add new Linked Server.

122. What is Cursor?

Cursor is a database object used by applications to manipulate data in a set on a row-by-row basis, instead of the typical SQL commands that operate on all the rows in the set at one time.
In order to work with a cursor we need to perform some steps in the following order:

1. Declare cursor
2. Open cursor
3. Fetch row from the cursor

4. Process fetched row
5. Close cursor
6. De-allocate cursor

123. What is Collation?

Collation refers to a set of rules that determine how data is sorted and compared. 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.

124. What is Difference between Function and Stored Procedure?

UDF can be used in the SQL statements anywhere in the WHERE/HAVING/SELECT section where as Stored procedures cannot be. UDFs that return tables can be treated as another row set. This can be used in JOINs with other tables. Inline UDF's can be thought of as views that take parameters and can be used in JOINs and other Row set operations.

125. What is sub-query? Explain properties of sub-query?

Sub-queries are often referred to as sub-selects, as they allow a SELECT statement to be executed arbitrarily within the body of another SQL statement. A sub-query is executed by enclosing it in a set of parentheses. Sub-queries are generally used to return a single row as an atomic value, though they may be used to compare values against multiple rows with the IN keyword.

A subquery is a SELECT statement that is nested within another T-SQL statement. A subquery SELECT statement if executed independently of the T-SQL statement, in which it is nested, will return a resultset. Meaning a subquery SELECT statement can standalone and is not depended on the statement in which it is nested. A subquery SELECT statement can return any number of values, and can be found in, the column list of a SELECT statement, a FROM, GROUP BY, HAVING, and/or ORDER BY clauses of a T-SQL statement. A Subquery can also be used as a parameter to a function call. Basically a subquery can be used anywhere an expression can be used.

126. What are different Types of Join?

1. **Cross Join** A cross join that does not have a WHERE clause produces the Cartesian product of the tables involved in the join. The size of a Cartesian product result set is the number of rows in the first table multiplied by the number of rows in the second table. The common example is when company wants to combine each product with a pricing table to analyze each product at each price.
2. **Inner Join** A join that displays only the rows that have a match in both joined tables is known as inner Join. This is the default type of join in the Query and View Designer.

3. **Outer Join** A join that includes rows even if they do not have related rows in the joined table is an Outer Join. You can create three different outer join to specify the unmatched rows to be included:
    1. **Left Outer Join:** In Left Outer Join all rows in the first-named table i.e. "left" table, which appears leftmost in the JOIN clause are included. Unmatched rows in the right table do not appear.
    2. **Right Outer Join:** In Right Outer Join all rows in the second-named table i.e. "right" table, which appears rightmost in the JOIN clause are included. Unmatched rows in the left table are not included.
    3. **Full Outer Join:** In Full Outer Join all rows in all joined tables are included, whether they are matched or not.
4. **Self Join** This is a particular case when one table joins to itself, with one or two aliases to avoid confusion. A self join can be of any type, as long as the joined tables are the same. A self join is rather unique in that it involves a relationship with only one table. The common example is when company has a hierarchal reporting structure whereby one member of staff reports to another. Self Join can be Outer Join or Inner Join.

127. What are primary keys and foreign keys?

Primary keys are the unique identifiers for each row. They must contain unique values and cannot be null. Due to their importance in relational databases, Primary keys are the most fundamental of all keys and constraints. A table can have only one Primary key. Foreign keys are both a method of ensuring data integrity and a manifestation of the relationship between tables.

128. What is User Defined Functions? What kind of User-Defined Functions can be created?

User-Defined Functions allow defining its own T-SQL functions that can accept 0 or more parameters and return a single scalar data value or a table data type.
Different Kinds of User-Defined Functions created are:

1. **Scalar User-Defined Function** A Scalar user-defined function returns one of the scalar data types. Text, image and timestamp data types are not supported. These are the type of user-defined functions that most developers are used to in other programming languages. You pass in 0 to many parameters and you get a return value.
2. **Inline Table-Value User-Defined Function** An Inline Table-Value user-defined function returns a table data type and is an exceptional alternative to a view as the user-defined function can pass parameters into a T-SQL select command and in essence provide us with a parameterized, non-updateable view of the underlying tables.
3. **Multi-statement Table-Value User-Defined Function** A Multi-Statement Table-Value user-defined function returns a table and is also an exceptional alternative to a view as the function can support multiple T-SQL statements to build the final

result where the view is limited to a single SELECT statement. Also, the ability to pass parameters into a TSQL select command or a group of them gives us the capability to in essence create a parameterized, non-updateable view of the data in the underlying tables. Within the create function command you must define the table structure that is being returned. After creating this type of user-defined function, It can be used in the FROM clause of a T-SQL command unlike the behavior found when using a stored procedure which can also return record sets.

129. What is Identity?

Identity (or AutoNumber) is a column that automatically generates numeric values. A start and increment value can be set, but most DBA leave these at 1. A GUID column also generates numbers; the value of this cannot be controlled. Identity/GUID columns do not need to be indexed.

130. Which TCP/IP port does SQL Server run on? How can it be changed?

SQL Server runs on port 1433. It can be changed from the Network Utility TCP/IP properties.

131. What are the difference between clustered and a non-clustered index?

1. **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.
2. **A non clustered 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 non clustered index does not consist of the data pages. Instead, the leaf nodes contain index rows.

132. What are the different index configurations a table can have?

A table can have one of the following index configurations:

1. No indexes
2. A clustered index
3. A clustered index and many nonclustered indexes
4. A non clustered index
5. Many non clustered indexes

134. What are different types of Collation Sensitivity?

1. **Case sensitivity** - A and a, B and b, etc.
2. **Accent sensitivity**
3. **Kana Sensitivity** - When Japanese kana characters Hiragana and Katakana are treated differently, it is called Kana sensitive.

4. **Width sensitivity** - A single-byte character (half-width) and the same character represented as a double-byte character (full-width) are treated differently than it is width sensitive.

135. What is OLTP (Online Transaction Processing)?

In OLTP - online transaction processing systems relational database design use the discipline of data modeling and generally follow the Codd rules of data normalization in order to ensure absolute data integrity. Using these rules complex information is broken down into its most simple structures (a table) where all of the individual atomic level elements relate to each other and satisfy the normalization rules.

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

Both primary key and unique key enforces 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 non clustered index by default. Another major difference is that, primary key doesn't allow NULLs, but unique key allows one NULL only.

137. What is difference between DELETE and TRUNCATE commands?

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.

1. **TRUNCATE**:
    1. TRUNCATE is faster and uses fewer system and transaction log resources than DELETE.
    2. TRUNCATE 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.
    3. TRUNCATE removes all rows from a table, but the table structure, its columns, constraints, indexes and so on, remains. The counter used by an identity for new rows is reset to the seed for the column.
    4. You cannot use TRUNCATE TABLE on a table referenced by a FOREIGN KEY constraint. Because TRUNCATE TABLE is not logged, it cannot activate a trigger.
    5. TRUNCATE cannot be rolled back.
    6. TRUNCATE is DDL Command.
    7. TRUNCATE Resets identity of the table
2. **DELETE**:
    1. DELETE removes rows one at a time and records an entry in the transaction log for each deleted row.
    2. 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.
    3. DELETE Can be used with or without a WHERE clause

4. DELETE Activates Triggers.
5. DELETE can be rolled back.
6. DELETE is DML Command.
7. DELETE does not reset identity of the table.

Note: DELETE and TRUNCATE both can be rolled back when surrounded by TRANSACTION if the current session is not closed. If TRUNCATE is written in Query Editor surrounded by TRANSACTION and if session is closed, it can not be rolled back but DELETE can be rolled back.

138. When is the use of UPDATE_STATISTICS command?

This command is basically used when a large processing of data has occurred. If a large amount of deletions any modification or Bulk Copy into the tables has occurred, it has to update the indexes to take these changes into account. UPDATE_STATISTICS updates the indexes on these tables accordingly.

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

They specify a search condition for a group or an aggregate. But the difference is that HAVING can be used only with the SELECT statement. HAVING is typically used in a GROUP BY clause. When GROUP BY is not used, HAVING behaves like a WHERE clause. Having Clause is basically used only with the GROUP BY function in a query whereas WHERE Clause is applied to each row before they are part of the GROUP BY function in a query.

140. What are the properties and different Types of Sub-Queries?

1. **Properties of Sub-Query**
    1. A sub-query must be enclosed in the parenthesis.
    2. A sub-query must be put in the right hand of the comparison operator, and
    3. A sub-query cannot contain an ORDER-BY clause.
    4. A query can contain more than one sub-query.
2. **Types of Sub-Query**
    1. Single-row sub-query, where the sub-query returns only one row.
    2. Multiple-row sub-query, where the sub-query returns multiple rows,. and
    3. Multiple column sub-query, where the sub-query returns multiple columns

141. What is 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 are hampering performances by executing too slowly.

Use SQL Profiler to monitor only the events in which you are interested. If traces are becoming too large, you can filter them based on the information you want, so that only a subset of the event data is collected. Monitoring too many events adds overhead to the server and the monitoring process and can cause the trace file or trace table to grow very large, especially when the monitoring process takes place over a long period of time.

142. What are the authentication modes in SQL Server? How can it be changed?

Windows mode and Mixed Mode - SQL and Windows. To change authentication mode in SQL Server click Start, Programs, Microsoft SQL Server and click SQL Enterprise Manager to run SQL Enterprise Manager from the Microsoft SQL Server program group. Select the server then from the Tools menu select SQL Server Configuration Properties, and choose the Security page.

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

SELECT SERVERPROPERTY ('productversion'), SERVERPROPERTY ('productlevel'), SERVERPROPERTY ('edition').

144. What is SQL Server Agent?

SQL Server agent plays an important role in the day-to-day tasks of a database administrator (DBA). It is often overlooked as one of the main tools for SQL Server management. Its purpose is to ease the implementation of tasks for the DBA, with its full-function scheduling engine, which allows you to schedule your own jobs and scripts.

145. Can a stored procedure call itself or recursive stored procedure? How much level SP nesting is possible?

Yes. Because Transact-SQL supports recursion, you can write stored procedures that call themselves. Recursion can be defined as a method of problem solving wherein the solution is arrived at by repetitively applying it to subsets of the problem. A common application of recursive logic is to perform numeric computations that lend themselves to repetitive evaluation by the same processing steps. Stored procedures are nested when one stored procedure calls another or executes managed code by referencing a CLR routine, type, or aggregate. You can nest stored procedures and managed code references up to 32 levels.

146. What is Log Shipping?

Log shipping is the process of automating the backup of database and transaction log files on a production SQL server, and then restoring them onto a standby server. Enterprise Editions only supports log shipping. In log shipping 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 can be used this as the Disaster Recovery plan. The key feature of log shipping is that it will automatically backup transaction logs throughout the day and automatically restore them on the standby server at defined interval.

147. Name 3 ways to get an accurate count of the number of records in a table?

SELECT * FROM table1
SELECT COUNT(*) FROM table1
SELECT rows FROM sysindexes WHERE id = OBJECT_ID(table1) AND indid < 2

148. What does it mean to have QUOTED_IDENTIFIER ON? What are the implications of having it OFF?

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.

149. What is the difference between a Local and a Global temporary table?

1. **A local temporary** table exists only for the duration of a connection or, if defined inside a compound statement, for the duration of the compound statement.
2. **A global temporary** table remains in the database permanently, but the rows exist only within a given connection. When connection is closed, the data in the global temporary table disappears. However, the table definition remains with the database for access when database is opened next time.

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

STUFF function is used to overwrite existing characters. Using this syntax, STUFF (string_expression, start, length, replacement_characters), string_expression is the string that will have characters substituted, start is the starting position, length is the number of characters in the string that are substituted, and replacement_characters are the new characters interjected into the string. REPLACE function to replace existing characters of all occurrences. Using the syntax REPLACE (string_expression, search_string, replacement_string), where every incidence of search_string found in the string_expression will be replaced with replacement_string.

151. What is PRIMARY KEY?

A **PRIMARY KEY** constraint is a unique identifier for a row within a database table. Every table should have a primary key constraint to uniquely identify each row and only one primary key constraint can be created for each table. The primary key constraints are used to enforce entity integrity.

152. What is UNIQUE KEY constraint?

A **UNIQUE** constraint enforces the uniqueness of the values in a set of columns, so no duplicate values are entered. The unique key constraints are used to enforce entity integrity as the primary key constraints.

153. What is FOREIGN KEY?

A **FOREIGN KE**Y constraint prevents any actions that would destroy links between tables with the corresponding data values. A foreign key in one table points to a primary key in another table. Foreign keys prevent actions that would leave rows with foreign key values when there are no primary keys with that value. The foreign key constraints are used to enforce referential integrity.

154. What is CHECK Constraint?

A **CHECK** constraint is used to limit the values that can be placed in a column. The check constraints are used to enforce domain integrity.

155. What is NOT NULL Constraint?

A **NOT NULL** constraint enforces that the column will not accept null values. The not null constraints are used to enforce domain integrity, as the check constraints.

156. How to get @@ERROR and @@ROWCOUNT at the same time?

If @@Rowcount is checked after Error checking statement then it will have 0 as the value of @@Recordcount as it would have been reset. And if @@Recordcount is checked before the error-checking statement then @@Error would get reset. To get @@error and @@rowcount at the same time do both in same statement and store them in local variable.

SELECT @RC = @@ROWCOUNT, @ER = @@ERROR

157. What is a Scheduled Jobs or What is a Scheduled Tasks?

Scheduled tasks let user automate processes that run on regular or predictable cycles. User can schedule administrative tasks, such as cube processing, to run during times of slow business activity. User can also determine the order in which tasks run by creating job steps within a SQL Server Agent job. E.g. back up database, Update Stats of Tables. Job steps

give user control over flow of execution. If one job fails, user can configure SQL Server Agent to continue to run the remaining tasks or to stop execution.

158. What are the advantages of using Stored Procedures?

1. Stored procedure can reduced network traffic and latency, boosting application performance.
2. Stored procedure execution plans can be reused, staying cached in SQL Server's memory, reducing server overhead.
3. Stored procedures help promote code reuse.
4. Stored procedures can encapsulate logic. You can change stored procedure code without affecting clients.
5. Stored procedures provide better security to your data.

159. What is a table called, if it has neither Cluster nor Non-cluster Index? What is it used for?

Unindexed table or Heap. Microsoft Press Books and Book on Line (BOL) refers it as Heap. A heap is a table that does not have a clustered index and, therefore, the pages are not linked by pointers. The IAM pages are the only structures that link the pages in a table together. Unindexed tables are good for fast storing of data. Many times it is better to drop all indexes from table and then do bulk of inserts and to restore those indexes after that.

160. Can SQL Servers linked to other servers like Oracle?

SQL Server can be linked to any server provided it has OLE-DB provider from Microsoft to allow a link. E.g. Oracle has an OLE-DB provider for oracle that Microsoft provides to add it as linked server to SQL Server group.

161. How to 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.

162. What are the basic functions for master, msdb, model, tempdb and resource databases?

1. The **master** database holds information for all databases located on the SQL Server instance and is theglue that holds the engine together. Because SQL Server cannot start without a functioning masterdatabase, you must administer this database with care.

2. The **msdb** database stores information regarding database backups, SQL Agent information, DTS packages, SQL Server jobs, and some replication information such as for log shipping.
3. The **tempdb** holds temporary objects such as global and local temporary tables and stored procedures.
4. The **model** is essentially a template database used in the creation of any new user database created in the instance.
5. The **resoure** Database is a read-only database that contains all the system objects that are included with SQL Server. SQL Server system objects, such as sys.objects, are physically persisted in the Resource database, but they logically appear in the sys schema of every database. The Resource database does not contain user data or user metadata.

163. What is Service Broker?

Service Broker is a message-queuing technology in SQL Server that allows developers to integrate SQL Server fully into distributed applications. Service Broker is feature which provides facility to SQL Server to send an asynchronous, transactional message. it allows a database to send a message to another database without waiting for the response, so the application will continue to function if the remote database is temporarily unavailable.

164. Where SQL server user names and passwords are stored in SQL server?

They get stored in System Catalog Views sys.server_principals and sys.sql_logins.

165. What is Policy Management?

Policy Management in SQL SERVER 2008 allows you to define and enforce policies for configuring and managing SQL Server across the enterprise. Policy-Based Management is configured in SQL Server Management Studio (SSMS). Navigate to the Object Explorer and expand the Management node and the Policy Management node; you will see the Policies, Conditions, and Facets nodes.

166. What is Replication and Database Mirroring?

Database mirroring can be used with replication to provide availability for the publication database. Database mirroring involves two copies of a single database that typically reside on different computers. At any given time, only one copy of the database is currently available to clients which are known as the principal database. Updates made by clients to the principal database are applied on the other copy of the database, known as the mirror database. Mirroring involves applying the transaction log from every insertion, update, or deletion made on the principal database onto the mirror database.

167. What are Sparse Columns?

A **sparse** column is another tool used to reduce the amount of physical storage used in a database. They are the ordinary columns that have an optimized storage for null values. Sparse columns reduce the space requirements for null values at the cost of more overhead to retrieve nonnull values.

168. What does TOP Operator Do?

The **TOP** operator is used to specify the number of rows to be returned by a query. The TOP operator has new addition in SQL SERVER 2008 that it accepts variables as well as literal values and can be used with INSERT, UPDATE, and DELETES statements.

169. What is CTE?

CTE is an abbreviation Common Table Expression. A Common Table Expression (CTE) is an expression that can be thought of as a temporary result set which is defined within the execution of a single SQL statement. A CTE is similar to a derived table in that it is not stored as an object and lasts only for the duration of the query.

170. What is MERGE Statement?

MERGE is a new feature that provides an efficient way to perform multiple DML operations. In previous versions of SQL Server, we had to write separate statements to INSERT, UPDATE, or DELETE data based on certain conditions, but now, using MERGE statement we can include the logic of such data modifications in one statement that even checks when the data is matched then just update it and when unmatched then insert it. One of the most important advantages of MERGE statement is all the data is read and processed only once.

171. What is Filtered Index?

Filtered Index is used to index a portion of rows in a table that means it applies filter on INDEX which improves query performance, reduce index maintenance costs, and reduce index storage costs compared with full-table indexes. When we see an Index created with some where clause then that is actually a FILTERED INDEX.

172. Which are new data types introduced in SQL SERVER 2008?

1. **The GEOMETRY Type:** The GEOMETRY data type is a system .NET common language runtime (CLR) data type in SQL Server. This type represents data in a two-dimensional Euclidean coordinate system.
2. **The GEOGRAPHY Type:** The GEOGRAPHY datatype's functions are the same as with GEOMETRY. The difference between the two is that when you specify GEOGRAPHY, you are usually specifying points in terms of latitude and longitude.

3. **New Date and Time Datatypes:** SQL Server 2008 introduces four new datatypes related to date and time: DATE, TIME, DATETIMEOFFSET, and DATETIME2.
    1. **DATE:** The new DATE type just stores the date itself. It is based on the Gregorian calendar and handles years from 1 to 9999.
    2. **TIME:** The new TIME (n) type stores time with a range of 00:00:00.0000000 through 23:59:59.9999999. The precision is allowed with this type. TIME supports seconds down to 100 nanoseconds. The n in TIME (n) defines this level of fractional second precision, from 0 to 7 digits of precision.
    3. **The DATETIMEOFFSET Type:** DATETIMEOFFSET (n) is the time-zone-aware version of a datetime datatype. The name will appear less odd when you consider what it really is: a date + a time + a time-zone offset. The offset is based on how far behind or ahead you are from Coordinated Universal Time (UTC) time.
    4. **The DATETIME2 Type:** It is an extension of the datetime type in earlier versions of SQL Server. This new datatype has a date range covering dates from January 1 of year 1 through December 31 of year 9999. This is a definite improvement over the 1753 lower boundary of the datetime datatype. DATETIME2 not only includes the larger date range, but also has a timestamp and the same fractional precision that TIME type provides

173. What are the Advantages of using CTE?

1. Using CTE improves the readability and makes maintenance of complex queries easy.
2. The query can be divided into separate, simple, logical building blocks which can be then used to build more complex CTEs until final result set is generated.
3. CTE can be defined in functions, stored procedures, triggers or even views.
4. After a CTE is defined, it can be used as a Table or a View and can SELECT, INSERT, UPDATE or DELETE Data.

174. How would you handle error in SQL SERVER 2008?

SQL Server now supports the use of TRY...CATCH con handling. TRY...CATCH lets us build error handling at the level we need, in the way w to, by setting a region where if any error occurs, it will break out of the region and head to an error handler.

The basic structure is as follows:
BEGIN TRY
stmts..
END TRY
BEGIN CATCH
stmts..
END CATCH

175. What is Aggregate Functions?

Aggregate functions perform a calculation on a set of values and return a single value. Aggregate functions ignore NULL values except COUNT function. HAVING clause is used, along with GROUP BY, for filtering query using aggregate values.

Following functions are aggregate functions.
AVG, MIN CHECKSUM_AGG, SUM, COUNT, STDEV, COUNT_BIG, STDEVP, GROUPING, VAR, MAX. VARP

176. What do you mean by Table Sample?

TABLESAMPLE allows you to extract a sampling of rows from a table in the FROM clause. The rows retrieved are random and they are not in any order. This sampling can be based on a percentage of number of rows. You can use TABLESAMPLE when only a sampling of rows is necessary for the application instead of a full result set.

177. What is the difference between UNION and UNION ALL?

1. **UNION** The UNION command is used to select related information from two tables, much like the JOIN command. However, when using the UNION command all selected columns need to be of the same data type. With UNION, only distinct values are selected.
2. **UNION ALL** The UNION ALL command is equal to the UNION command, except that UNION ALL selects all values.

The difference between Union and Union all is that Union all will not eliminate duplicate rows, instead it just pulls all rows from all tables fitting your query specifics and combines them into a table.

178. What is B-Tree?

The database server uses a B-tree structure to organize index information. B-Tree generally has following types of index pages or nodes:

1. **root node:** A root node contains node pointers to branch nodes which can be only one.
2. **branch node:** A branch node contains pointers to leaf nodes or other branch nodes which can be two or more.
3. **leaf nodes:** A leaf node contains index items and orizantal pointers to other leaf nodes which can be many.

179. What is a foreign key, and what is it used for?

A foreign key is used to establish relationships among relations in the relational model. Technically, a foreign key is a column (or columns) appearing in one relation that is (are) the primary key of another table. Although there may be exceptions, the values in the foreign key columns usually must correspond to values existing in the set of primary key values. This correspondence requirement is created in a database using a referential integrity constraint on the foreign key.

180. What does it mean when we say that a relation is in Boyce-Codd Normal Form (BCNF)?

A relation is in BCNF when every determinant in the relation is a candidate key. This means that any possible primary key can determine all other attributes in the relation. Attributes may not be determined by non-candidate key attributes or part of a composite candidate key. Thus it is said "I swear to construct my tables so that all nonkey columns are dependent on the key, the whole key and nothing but the key, so help me Codd!"

181. You have been given a set of tables with data and asked to create a new database to store them. When you examine the data values in the tables, what are you looking for?

(1) Multivalued dependencies, (2) Functional dependencies, (3) Candidate keys, (4) Primary keys and (5) Foreign keys.

182. Explain the difference between attributes and identifiers.

Entities have attributes. Attributes are properties that describe the entity's characteristics. Entity instances have identifiers. Identifiers are attributes that name, or identify, entity instances.

183. Name and describe three types of binary relationships.

1:1 - a single entity instance of one type is related to a single-entity instance of another type.

1:N - a single entity instance of one type is related to many-entity instances of another type.

M:N - many-entity instances of one type relate to many-entity instances of another type.

184. What are stored procedures, and how do they differ from triggers?

A stored procedure is a program that is stored within the database and is compiled when used. They can receive input parameters and they can return results. Unlike triggers, their scope is database-wide; they can be used by any process that has permission to use the database stored procedure.

185. What are the advantages of using stored procedures?

The advantages of stored procedures are (1) greater security, (2) decreased network traffic, (3) the fact that SQL can be optimized and (4) code sharing which leads to less work, standardized processing, and specialization among developers.

186. What is the relationship of ODBC, OLE DB, and ADO?

Developed first, the ODBC standard is for relational databases; while the OLE DB standard provides functionality for both relational and other databases. Finally, ADO was developed to provide easier access to OLE DB data for the non-object-oriented programmer.

187. Explain the differences between structured data and unstructured data.

Structured data are facts concerning objects and events. The most important structured data are numeric, character, and dates. Structured data are stored in tabular form. Unstructured data are multimedia data such as documents, photographs, maps, images, sound, and video clips. Unstructured data are most commonly found on Web servers and Web-enabled databases.

188.What are dimension tables and definition of Fact tables?

These two questions are most commonly asked **database interview questions**. Fact tables are mainly central tables that are an integral part of data warehousing and dimension tables are used for describing the attributes of the fact tables. Both of these tables are important and play an important role in maintaining the database management system.

189.What is Data Warehouse?

A data warehouse is a subject-oriented, integrated, time-variant and non-volatile collection of data in support of management's decision making process.

Subject-Oriented: A data warehouse can be used to analyze a particular subject area. For example, "sales" can be a particular subject.

Integrated: A data warehouse integrates data from multiple data sources. For example, source A and source B may have different ways of identifying a product, but in a data warehouse, there will be only a single way of identifying a product.

Time-Variant: Historical data is kept in a data warehouse. For example, one can retrieve data from 3 months, 6 months, 12 months, or even older data from a data warehouse. This contrasts with a transactions system, where often only the most recent data is kept. For example, a transaction system may hold the most recent address of a customer, where a data warehouse can hold all addresses associated with a customer.

Non-volatile: Once data is in the data warehouse, it will not change. So, historical data in a data warehouse should never be altered.

190.What is Data Mining?

Data mining is a term from computer science. Sometimes it is also called **knowledge discovery in databases** (KDD). Data mining is about finding new information in a lot of data. The information obtained from data mining is hopefully both new and useful.

MOST FREQUENTLY ASKED SQL QURIES

1.SQL Query to find second highest salary of Employee
Answer : There are many ways to find second highest salary of Employee in SQL, you can either use SQL Join or Subquery to solve this problem. Here is SQL query using Subquery :

```
1.  select MAX(Salary) from Employee WHERE Salary NOT IN (select MAX(Salary) from
    Employee );
```

2. SQL Query to find Max Salary from each department.
Answer :

```
SELECT DeptID, MAX(Salary) FROM Employee  GROUP BY DeptID.
```

3.Write SQL Query to display current date.
Ans:SQL has built in function called GetDate() which returns current timestamp.

```
SELECT GetDate();
```

4.Write an SQL Query to check whether date passed to Query is date of given format or not.
Ans: SQL has IsDate() function which is used to check passed value is date or not of specified format ,it returns 1(true) or 0(false) accordingly.

```
SELECT  ISDATE('1/08/13') AS "MM/DD/YY";
```

It will return 0 because passed date is not in correct format.

5. Write a SQL Query to print the name of distinct employee whose DOB is between 01/01/1960 to 31/12/1975.
Ans:

```
SELECT DISTINCT EmpName FROM Employees WHERE DOB  BETWEEN '01/01/1960'
AND '31/12/1975';
```

6.Write an SQL Query find number of employees according to gender whose DOB is between 01/01/1960 to 31/12/1975.
Answer : SELECT COUNT(*), sex from Employees WHERE DOB BETWEEN '01/01/1960 ' AND '31/12/1975' GROUP BY sex;

7.Write an SQL Query to find employee whose Salary is equal or greater than 10000.
Answer : SELECT EmpName FROM Employees WHERE Salary>=10000;

8.Write an SQL Query to find name of employee whose name Start with 'M'
Ans: SELECT * FROM Employees WHERE EmpName like 'M%';

9. find all Employee records containing the word "Joe", regardless of whether it was stored as JOE, Joe, or joe.
Answer : SELECT * from Employees WHERE upper(EmpName) like upper('joe%');

10. Write a SQL Query to find year from date.
Answer : SELECT YEAR(GETDATE()) as "Year";

11.To fetch ALTERNATE records from a table. (EVEN NUMBERED)
select * from emp where rowid in (select decode(mod(rownum,2),0,rowid, null) from emp);

12.To select ALTERNATE records from a table. (ODD NUMBERED)
select * from emp where rowid in (select decode(mod(rownum,2),0,null ,rowid) from emp);

13.Find the 3rd MAX salary in the emp table.
select distinct sal from emp e1 where 3 = (select count(distinct sal) from emp e2 where e1.sal <= e2.sal);

14.Find the 3rd MIN salary in the emp table.
select distinct sal from emp e1 where 3 = (select count(distinct sal) from emp e2where e1.sal >= e2.sal);

15.Select FIRST n records from a table.
select * from emp where rownum <= &n;

16.Select LAST n records from a table
select * from emp minus select * from emp where rownum <= (select count(*) - &n from emp);

17.List dept no., Dept name for all the departments in which there are no employees in the department.
select * from dept where deptno not in (select deptno from emp);
alternate solution: select * from dept a where not exists (select * from emp b where a.deptno = b.deptno);

altertnate solution: select empno,ename,b.deptno,dname from emp a, dept b where a.deptno(+) = b.deptno and empno is null;

18.How to get 3 Max salaries ?
select distinct sal from emp a where 3 >= (select count(distinct sal) from emp b where a.sal <= b.sal) order by a.sal desc;

19.How to get 3 Min salaries ?
select distinct sal from emp a where 3 >= (select count(distinct sal) from emp b where a.sal >= b.sal);

20.How to get nth max salaries ?
select distinct hiredate from emp a where &n = (select count(distinct sal) from emp b where a.sal >= b.sal);

21.Select DISTINCT RECORDS from emp table.
select * from emp a where rowid = (select max(rowid) from emp b
where a.empno=b.empno);

22.How to delete duplicate rows in a table?
delete from emp a where rowid != (select max(rowid) from emp b
where a.empno=b.empno);

23.Count of number of employees in department wise.
select count(EMPNO), b.deptno, dname from emp a, dept b where a.deptno(+)=b.deptno group by b.deptno,dname;

24. Suppose there is annual salary information provided by emp table. How to fetch monthly salary of each and every employee?

select ename,sal/12 as monthlysal from emp;

25.Select all record from emp table where deptno =10 or 40.

select * from emp where deptno=30 or deptno=10;

26.Select all record from emp table where deptno=30 and sal>1500.

select * from emp where deptno=30 and sal>1500;

27.Select all record from emp where job not in SALESMAN or CLERK.

select * from emp where job not in ('SALESMAN','CLERK');

28.Select all record from emp where ename in 'BLAKE','SCOTT','KING'and'FORD'.

select * from emp where ename in('JONES','BLAKE','SCOTT','KING','FORD');

29.Select all records where ename starts with 'S' and its lenth is 6 char.

select * from emp where ename like'S___';

30.Select all records where ename may be any no of character but it should end with 'R'.

select * from emp where ename like'%R';

31.Count MGR and their salary in emp table.

select count(MGR),count(sal) from emp;

32.In emp table add comm+sal as total sal .

select ename,(sal+nvl(comm,0)) as totalsal from emp;

33.Select any salary <3000 from emp table.

select * from emp where sal> any(select sal from emp where sal<3000);

34.Select all salary <3000 from emp table.

select * from emp where sal> all(select sal from emp where sal<3000);

35.Select all the employee group by deptno and sal in descending order.

select ename,deptno,sal from emp order by deptno,sal desc;

36.How can I create an empty table emp1 with same structure as emp?

Create table emp1 as select * from emp where 1=2;

37.How to retrive record where sal between 1000 to 2000?
Select * from emp where sal>=1000 And sal<2000

38.Select all records where dept no of both emp and dept table matches.
select * from emp where exists(select * from dept where emp.deptno=dept.deptno)

39.If there are two tables emp1 and emp2, and both have common record. How can I fetch all the recods but common records only once?
(Select * from emp) Union (Select * from emp1)

40.How to fetch only common records from two tables emp and emp1?
(Select * from emp) Intersect (Select * from emp1)

41. How can I retrive all records of emp1 those should not present in emp2?
(Select * from emp) Minus (Select * from emp1)

42.Count the totalsa deptno wise where more than 2 employees exist.
SELECT deptno, sum(sal) As totalsal
FROM emp
GROUP BY deptno
HAVING COUNT(empno) > 2

43.Display the names of employees who are working in the company for the past 5 years.
select ename from emp where sysdate-hiredate>5*365;

44.Display the list of employees who have joined the company before 30th June 90 or after 31st dec 90.
select * from emp where hiredate between '30-jun-1990' and '31-dec-1990';

45.Display the names of employees working in department number 10 or 20 or 40 or employees working as clerks, salesman or analyst.
select ename from emp where deptno in (10,20,40) or job in
('CLERK','SALESMAN','ANALYST');

46.Display the names of employees whose name starts with alphabet S.
select ename from emp where ename like 'S%';

47.Display employee names for employees whose name ends with alphabet.
select ename from emp where ename like '%S';