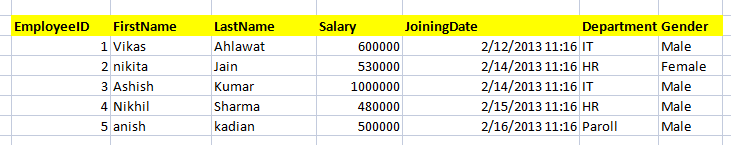


SQL Mega Project:-

Basics and Intermediate

Project Type(A)





1) Write a query to get all EmployeeDetail detail from "EmployeeDetail" table **SELECT \* FROM EMPLOYEEDETAIL**

2) Write a query to get only "FirstName" column from "EmployeeDetail" table

**SELECT FirstName FROM EMPLOYEEDETAIL**

3) Write a query to get FirstName in upper case as "First Name".

SELECT UPPER(FirstName) AS "First Name"

FROM EmployeeDetail;

4) Write a query to get FirstName in upper case as "First Name".

SELECT UPPER(FirstName) AS "First Name"

FROM EmployeeDetail;

5) Write a query for combine FirstName and LastName and display it as "Name" (also include white space between first name & last name)

SELECT CONCAT(FirstName, ' ', LastName) AS "Name"

FROM EmployeeDetail;

1. Select EmployeeDetail detail whose name is "Vikas

SELECT \*

FROM EmployeeDetail WHERE FirstName = "Vikas";

1. Get all EmployeeDetail detail from EmployeeDetail table whose "FirstName" start with latter 'a'.

SELECT \*

FROM EmployeeDetail

WHERE FirstName LIKE 'a%';

8) Get all EmployeeDetail detail from EmployeeDetail table whose "FirstName" start with latter 'a'.

SELECT \*

FROM EmployeeDetail

WHERE FirstName LIKE 'a%';

1. Get all EmployeeDetail details from EmployeeDetail table whose "FirstName" end with 'h'

SELECT \*

FROM EmployeeDetail

WHERE FirstName LIKE '%h';

Get all EmployeeDetail detail from EmployeeDetail table whose "FirstName" start with any single character between 'a-p'

SELECT \*

FROM EmployeeDetail

WHERE FirstName LIKE '[a-p]%';

1. Get all EmployeeDetail detail from EmployeeDetail table whose "FirstName" not start with any single character between 'a-p'

SELECT \*

FROM EmployeeDetail

WHERE FirstName NOT LIKE '%[a-p]';

1. Get all EmployeeDetail detail from EmployeeDetail table whose "Gender" end with 'le' and contain 4 letters. The Underscore(\_) Wildcard Character represents any single character

SELECT \*

FROM EmployeeDetail

WHERE Gender LIKE '\_\_le';

1. Get all EmployeeDetail detail from EmployeeDetail table whose "FirstName" start with 'A' and contain 5 letters

SELECT \*

FROM EmployeeDetail

WHERE FirstName LIKE 'A\_\_\_\_';

1. Get all EmployeeDetail detail from EmployeeDetail table whose "FirstName" containing '%'. ex:-"Vik%as".

SELECT \*

FROM EmployeeDetail

WHERE FirstName LIKE '%\%%' ESCAPE '\';

1. Get all unique "Department" from EmployeeDetail table

SELECT DISTINCT department

FROM EmployeeDetail;

1. Get the highest "Salary" from EmployeeDetail table.

SELECT MAX(Salary)

FROM EmployeeDetail;

1. Get the lowest "Salary" from EmployeeDetail table

SELECT MIN(Salary)

FROM EmployeeDetail;

1. Show "JoiningDate" in "dd mmm yyyy" format, ex- "15 Feb 2013

SELECT CONVERT(VARCHAR(11), JoiningDate, 106)

FROM EmployeeDetail;

1. Show "JoiningDate" in "yyyy/mm/dd" format, ex- "2013/02/15"

SELECT CONVERT(VARCHAR, JoiningDate, 111) FROM EmployeeDetail;

1. Show only time part of the "JoiningDate"

SELECT FORMAT(JoiningDate, 'HH:mm:ss')

FROM EmployeeDetail;

1. Get only Year part of "JoiningDate"

SELECT YEAR(JoiningDate)FROM EmployeeDetail;

1. Get only Month part of "JoiningDate”

SELECT MONTH(JoiningDate)FROM EmployeeDetail; in number

SELECT DATENAME(MONTH,JoiningDate) FROM EmployeeDetail; in words

1. Get system date

SELECT GETDATE();

1. Get UTC date.

SELECT GETUTCDATE();

* 1. Get the first name, current date, joiningdate and diff between current date and joining date in months.

SELECT FirstName,GETDATE() AS CurrentDate,JoiningDate,DATEDIFF(MONTH, JoiningDate, GETDATE()) AS Monthdifference FROM EmployeeDetail

1. Get the first name, current date, joiningdate and diff between current date and joining date in days.

SELECT FirstName,GETDATE() AS CurrentDate,JoiningDate,DATEDIFF(DAY, JoiningDate, GETDATE()) AS Daydifference FROM EmployeeDetail

1. Get all EmployeeDetail details from EmployeeDetail table whose joining year is 2013

SELECT \*

FROM EmployeeDetail

WHERE YEAR(JoiningDate) = 2013;

1. Get all EmployeeDetail details from EmployeeDetail table whose joining month is Jan(1)

SELECT \* FROM EmployeeDetail WHERE DATENAME(MONTH,JoiningDate) = 'January'

1. Get all EmployeeDetail details from EmployeeDetail table whose joining month is Jan(1)

SELECT \* FROM EmployeeDetail WHERE DATENAME(MONTH,JoiningDate) = 'January'

1. Get all EmployeeDetail details from EmployeeDetail table whose joining month is Jan(1)

SELECT \* FROM EmployeeDetail WHERE DATENAME(MONTH,JoiningDate) = 'January'

1. Get how many EmployeeDetail exist in "EmployeeDetail" table

SELECT count(\*) FROM EmployeeDetail

1. Select only one/top 1 record from "EmployeeDetail" table

SELECT TOP 1 \*

FROM EmployeeDetail

1. Select all EmployeeDetail detail with First name "Vikas","Ashish", and "Nikhil".

SELECT \* FROM EmployeeDetail WHERE FirstName IN ('Vikas','Ashish','Nikhil')

1. Select all EmployeeDetail detail with First name not in "Vikas","Ashish", and "Nikhil"

SELECT \* FROM EmployeeDetail WHERE FirstName NOT IN ('Vikas','Ashish','Nikhil')

1. Select first name from "EmployeeDetail" table after removing white spaces from right side

SELECT RTRIM(FirstName) FROM EmployeeDetail;

1. Select first name from "EmployeeDetail" table after removing white spaces from left side

SELECT LTRIM(FirstName) FROM EmployeeDetail;

1. Display first name and Gender as M/F.(if male then M, if Female then F)

SELECT

FirstName,

CASE

WHEN Gender = 'Male' THEN 'M'

WHEN Gender = 'Female' THEN 'F'

ELSE Gender

END AS Gender

FROM EmployeeDetail;

1. Select first name from "EmployeeDetail" table prifixed with "Hello

SELECT 'Hello ' + FirstName FROM EmployeeDetail;

SELECT \* FROM EmployeeDetail WHERE Salary > 600000;

1. Get Get EmployeeDetail details from "EmployeeDetail" table whose Salary greater than 600000
2. EmployeeDetail details from "EmployeeDetail" table whose Salary less than 700000

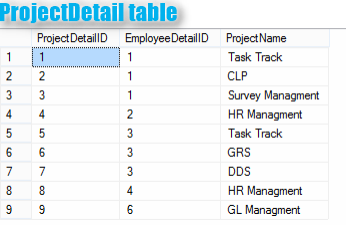
SELECT \* FROM EmployeeDetail WHERE Salary < 700000;

1. Get EmployeeDetail details from "EmployeeDetail" table whose Salary between 500000 than 600000

SELECT \* FROM EmployeeDetail WHERE Salary >= 500000 and Salary <= 600000;







1. Give records of ProjectDetail table

SELECT \* FROM ProjectDetail

1. Write the query to get the department and department wise total(sum) salary from "EmployeeDetail" table.

SELECT ed.Department, SUM(ed.Salary) AS TotalSalary

FROM EmployeeDetail ed

JOIN ProjectDetail pd ON ed.empId = pd.EmployeeDetailID

GROUP BY ed.Department;

1. Write the query to get the department and department wise total(sum) salary, display it in ascending order according to salary.

SELECT ed.Department, SUM(ed.Salary) AS TotalSalary

FROM EmployeeDetail ed

JOIN ProjectDetail pd ON ed.empId = pd.EmployeeDetailID

GROUP BY ed.Department ORDER BY TotalSalary ASC;

1. Write the query to get the department and department wise total(sum) salary, display it in descending order according to salary.

SELECT ed.Department, SUM(ed.Salary) AS TotalSalary

FROM EmployeeDetail ed

JOIN ProjectDetail pd ON ed.empId = pd.EmployeeDetailID

GROUP BY ed.Department ORDER BY TotalSalary ASC;

1. Write the query to get the department, total no. of departments, total(sum) salary with respect to department from "EmployeeDetail" table.

SELECT

Department,

COUNT(\*) AS TotalDepartment,

SUM(Salary) AS TotalSalary

FROM

EmployeeDetail

GROUP BY

Department;

1. Get department wise average salary from "EmployeeDetail" table order by salary ascending

SELECT

Department,

AVG(Salary) AS AverageSalary

FROM

EmployeeDetail

GROUP BY

Department

ORDER BY

AverageSalary ASC;

47 . Get department wise maximum salary from "EmployeeDetail" table order by salary ascending

SELECT

Department,

MAX(Salary) AS MaxSalary

FROM

EmployeeDetail

GROUP BY

Department

ORDER BY

MaxSalary ASC;

48.Get department wise minimum salary from "EmployeeDetail" table order by salary ascending.

SELECT

Department,

MIN(Salary) AS MinSalary

FROM

EmployeeDetail

GROUP BY

Department

ORDER BY

MinSalary ASC;

1. Join both the table that is EmployeeDetail and ProjectDetail based on some common parameter

SELECT

\* FROM EmployeeDetail JOIN ProjectDetail ON

EMPLOYEEDETAIL.employeeId = ProjectDetail.EmployeeDetailId ;

1. Get EmployeeDetail name, project name order by firstname from "EmployeeDetail" and "ProjectDetail" for those EmployeeDetail which have assigned project already.

SELECT

FirstName, ProjectName

FROM

[EmployeeDetail]A

INNER JOIN

[ProjectDetail]B

ON

A.employeeId = B.EmployeeDetailId

ORDER By

FirstName;

1. Get EmployeeDetail name, project name order by firstname from "EmployeeDetail" and "ProjectDetail" for all EmployeeDetail even they have not assigned project.

SELECT

FirstName, ProjectName

FROM

[EmployeeDetail]A

LEFT OUTER JOIN

[ProjectDetail]B

ON

A.employeeId = B.EmployeeDetailId

ORDER By

FirstName;

1. Get EmployeeDetail name, project name order by firstname from "EmployeeDetail" and "ProjectDetail" for all EmployeeDetail if project is not assigned then display "-No Project Assigned"

SELECT

FirstName, ISNULL(ProjectName,'-No Project Assigned')

FROM

[EmployeeDetail]A

LEFT OUTER JOIN

[ProjectDetail]B

ON

A.employeeId = B.EmployeeDetailId

ORDER By

FirstName;

54.Get all project name even they have not matching any EmployeeDetailid, in left table, order by firstname from "EmployeeDetail" and "ProjectDetail

SELECT FirstName ,

ProjectName FROM

[EmployeeDetail]A

RIGHT OUTER JOIN

[ProjectDetail]B

ON

A.employeeId = B.EmployeeDetailId

ORDER By

FirstName;

1. Get complete record (EmployeeDetailname, project name) from both tables ([EmployeeDetail],[ProjectDetail]), if no match found in any table then show NULL

SELECT FirstName ,

ProjectName FROM

[EmployeeDetail]A

FULL OUTER JOIN

[ProjectDetail]B

ON

A.employeeId = B.EmployeeDetailId

ORDER By

FirstName;

1. Get complete record (EmployeeDetailname, project name) from both tables ([EmployeeDetail],[ProjectDetail]), if no match found in any table then show NULL

SELECT FirstName ,

ProjectName FROM

[EmployeeDetail]A

FULL OUTER JOIN

[ProjectDetail]B

ON

A.employeeId = B.EmployeeDetailId

ORDER By

FirstName;

57.Get complete record (EmployeeDetailname, project name) from both tables ([EmployeeDetail],[ProjectDetail]), if no match found in any table then show NULL

SELECT FirstName ,

ProjectName FROM

[EmployeeDetail]A

FULL OUTER JOIN

[ProjectDetail]B

ON

A.employeeId = B.EmployeeDetailId

ORDER By

FirstName;

58.Write down the query to fetch EmployeeDetailName & Project who has assign more than one project

Select EmployeeId,FirstName, ProjectName from [EmployeeDetail]E JOIN [ProjectDetail]P ON E.EmployeeId = P.EmployeeDetailId Where EmployeeId IN (Select EmployeeDetailId from ProjectDetail GROUP by EmployeeDetailId HAVING Count(\*) > 1);

1. Write down the query to fetch ProjectName on which more than one EmployeeDetail are working along with EmployeeDetailName

Select P.ProjectName, E.FirstName from ProjectDetail P INNER JOIN EmployeeDetail E ON P.EmployeeDetailId

= E.EmployeeID WHERE P.ProjectName IN (Select ProjectName From ProjectDetail GROUP BY ProjectName HAVING Count(1) > 1);

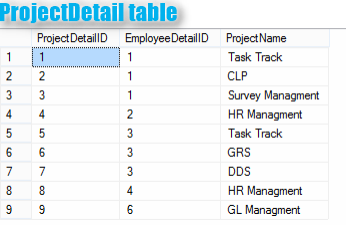
Apply Cross Join in Both the tables

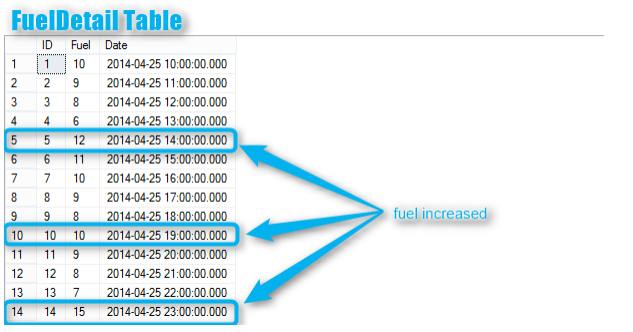
Advanced:-

Project Type(B):-









**CREATE TABLE FuelDetail (**

**ID INT PRIMARY KEY,**

**Fuel INT,**

**Date DATETIME**

**);**

**INSERT INTO FuelDetail (ID, Fuel, Date) VALUES**

**(1, 10, '2014-04-25 10:00:00'),**

**(2, 9, '2014-04-25 11:00:00'),**

**(3, 8, '2014-04-25 12:00:00'),**

**(4, 8, '2014-04-25 13:00:00'),**

**(5, 12, '2014-04-25 14:00:00'),**

**(6, 12, '2014-04-25 15:00:00'),**

**(7, 12, '2014-04-25 16:00:00'),**

**(8, 12, '2014-04-25 17:00:00'),**

**(9, 12, '2014-04-25 18:00:00'),**

**(10, 15, '2014-04-25 19:00:00'),**

**(11, 15, '2014-04-25 20:00:00'),**

**(12, 15, '2014-04-25 21:00:00'),**

**(13, 15, '2014-04-25 22:00:00'),**

**(14, 20, '2014-04-25 23:00:00');**

1. Write down the query to print first letter of a Name in Upper Case and all other letter in Lower Case.(EmployDetail table)

**SELECT**

**EmployeeID,**

**UPPER(LEFT(FirstName, 1)) + LOWER(SUBSTRING(FirstName, 2, LEN(FirstName))) AS FormattedFirstName,**

**LastName,**

**Salary,**

**JoiningDate,**

**Department,**

**Gender**

**FROM EmployeeDetail;**

1. 1. Write down the query to display all EmployeeDetail name in one cell seprated by ',' ex:-"Vikas, nikita, Ashish, Nikhil , anish"(EmployDetail table)

**SELECT STRING\_AGG(FirstName, ', ') AS EmployeeNames**

**FROM EmployeeDetail;**

1. Write down the query to get ProjectName and respective EmployeeDetailName(firstname) which are working on the project, --if more then one EmployeeDetail working on same project, then it should be in same cell seprated by comma

--for example :- Task Tracker : Vikas, Ashish

**SELECT**

**pd.ProjectName,**

**STRING\_AGG(ed.FirstName, ', ') AS EmployeeNames**

**FROM**

**ProjectDetail pd**

**JOIN**

**EmployeeDetail ed ON pd.EmployeeDetailID = ed.EmployeeID**

**GROUP BY**

**pd.ProjectName;**

1. : You have a table(FuelDetail) with ID, Fuel, And Date columns. --Fuel column is contain fuel quantity at a particular time when car start traveling.

So we need to find out that when the driver fill Petrol in his/her car.

--By FuelDetail Table image on the top of this post, you can understand the query.

--Car start driving at 10 Am on 25th April with petrol(10 liter)

--at 11 AM Petrol was 9 liters

--at 12 AM petrol was 8 liters

--at 2 PM (14) petrol was 12 liters...

--This means that he/she fill the petrol at 25th April 2014 at 2PM --Next time he fill petrol at 7PM 25th April 2014

--and Next time he fill petrol at 11PM 25th April 2014

**SELECT**

**fd.ID,**

**fd.Fuel,**

**fd.Date**

**FROM**

**(SELECT**

**ID,**

**Fuel,**

**Date,**

**LAG(Fuel) OVER (ORDER BY Date) AS PreviousFuel**

**FROM FuelDetail) fd**

**WHERE**

**fd.Fuel > fd.PreviousFuel;**

5.What would be the out-put of the following Sql query?

SELECT A.A FROM (SELECT 1 A, 2 B) A JOIN (SELECT 1 A,1 B)B ON A.A = B.B

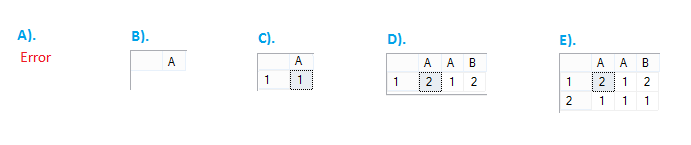
Options:



**Ans: C**

1. What would be the out-put of the following Sql query?

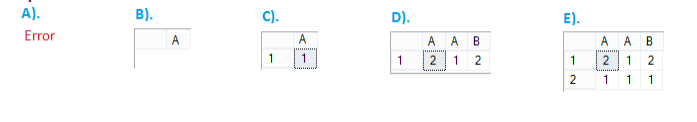
SELECT B.A FROM (SELECT 1 A) A JOIN (SELECT 1 A, 2 B)B ON A.A = B.A Options:



**Ans: C**

1. What would be the out-put of the following Sql query?

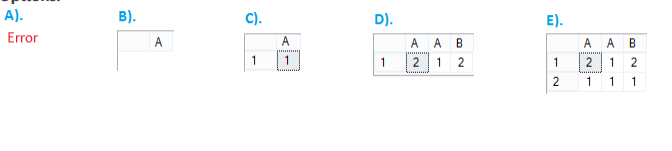
SELECT B.A FROM (SELECT 1 A) A JOIN (SELECT 1 A, 2 B)B ON A.A = B.B Options:



**Ans: B**

1. What would be the out-put of the following Sql query? SELECT \* FROM (SELECT 1 A UNION ALL SELECT 2 B) A JOIN (SELECT 1 A,2 B)B ON A.A = B.B

Options:



**Ans: D**

Advanced:-

Project Type©(Syntax Based Project):-

Write a Syntax queries of All Questions:-

1. How to select random record form a table?

**SELECT TOP 1 \* FROM EMPLOYEEDETAIL ORDER BY NEWID();**

1. Suppose that you have table EmployeeDetail with a column EName which contain Records EmployeeDetail name(EName) as A,B,A,A,B,D,C,M,A, Write a query which will change/Swap the EName A to B and B to A

UPDATE EmployeeDetail

SET EName = CASE

WHEN EName = 'A' THEN 'B'

WHEN EName = 'B' THEN 'A'

ELSE EName

END;

1. Write a query to create a clone of existing table without using Create Command

**SELECT \***

**INTO new\_table\_name**

**FROM original\_table\_name;**

1. Table Tbl1 has 100 rows, Table Tbl2 has 0 rows so number of rows returned by the below query?

**SELECT \***

**FROM Tbl1**

**INNER JOIN Tbl2**

**ON Tbl1.id = Tbl2.id;**

1. Write a query to print 1 to 100 in sql server without using loops

**WITH Numbers AS (**

**SELECT 1 AS Number**

**UNION ALL**

**SELECT Number + 1**

**FROM Numbers**

**WHERE Number < 100**

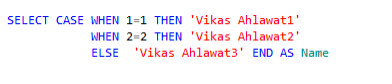
**)**

**SELECT Number**

**FROM Numbers**

**OPTION (MAXRECURSION 100);**

1. You have two tables with blank value in both table as shown in below image, Then what would be the output of the following Query based on the tables shown in image?
2. What will be the output of the following query?



**WITH Numbers AS (**

**SELECT 1 AS Number**

**UNION ALL**

**SELECT Number + 1**

**FROM Numbers**

**WHERE Number < 100**

**)**

**SELECT Number**

**FROM Numbers**

**OPTION (MAXRECURSION 100);**

1. What is the purpose of the WITH SCHEMABINDING clause and where can it be used?

The WITH SCHEMABINDING clause in SQL Server is used to bind a view or a user-defined function (UDF) to the schema of the underlying tables or objects it references. When you use this clause, it ensures that the structure of the referenced tables or objects cannot be altered in a way that would affect the view or function.

**Views**:

* When creating a view, you can use WITH SCHEMABINDING to ensure that the underlying schema is fixed and prevent schema changes.

CREATE VIEW dbo.MyView

WITH SCHEMABINDING

AS

SELECT Column1, Column2

FROM dbo.MyTable;

**User-Defined Functions (UDFs)**:

* It can be used in scalar or table-valued user-defined functions to bind the schema of any referenced objects.

CREATE FUNCTION dbo.MyFunction()

RETURNS TABLE

WITH SCHEMABINDING

AS

RETURN

SELECT Column1, Column2

FROM dbo.MyTable;

1. What is an Index?

An index in a database is a data structure that improves the speed of data retrieval operations on a table by providing quick access to rows. It is analogous to the index in a book, which helps you quickly locate specific information without flipping through every page.

Indexes are created on one or more columns of a table and store a sorted list of the data, enabling efficient lookups, sorting, and filtering.

1. Write the T-Sql statement/syntex for create and index

1. **Create a Non-Clustered Index**:

CREATE NONCLUSTERED INDEX Index\_Name

ON Table\_Name (Column1 ASC, Column2 DESC);

2. **Create a Clustered Index**:

CREATE CLUSTERED INDEX Index\_Name

ON Table\_Name (Column1 ASC);

3. **Create a Unique Index**:

CREATE UNIQUE NONCLUSTERED INDEX Index\_Name

ON Table\_Name (Column1);

4. **Create a Composite Index**:

CREATE NONCLUSTERED INDEX Index\_Name

ON Table\_Name (Column1, Column2);

5. **Create a Full-Text Index**:

CREATE FULLTEXT INDEX ON Table\_Name (Column1, Column2)

KEY INDEX PK\_Index\_Name

ON FullTextCatalog\_Name;

6. **Create a Filtered Index**:

CREATE NONCLUSTERED INDEX Index\_Name

ON Table\_Name (Column1)

WHERE Column1 IS NOT NULL;

Advanced :-

Project Type(D)(Stored Procedure Project):-

1. . What is Stored Procedure

A **stored procedure** in SQL is a precompiled collection of SQL statements and optional control-flow logic stored on the database server. It allows you to encapsulate and reuse SQL code for tasks that are performed frequently, such as querying, inserting, updating, or deleting data.

CREATE PROCEDURE Procedure\_Name

@Parameter1 DataType,

@Parameter2 DataType OUTPUT

AS

BEGIN

-- SQL Statements

SELECT @Parameter2 = COUNT(\*) FROM Table\_Name WHERE Column\_Name = @Parameter1;

END;

Stored Procedure with an output parameter

A stored procedure to calculate the total number of employees.

CREATE PROCEDURE GetEmployeeCount

@Count INT OUTPUT

AS

BEGIN

SELECT @Count = COUNT(\*)

FROM Employee;

END;

You can execute this with:

DECLARE @Total INT;

EXEC GetEmployeeCount @Total OUTPUT;

PRINT @Total;

**Executing a Stored Procedure:**

To execute a stored procedure, use the EXEC command or EXECUTE:

EXEC Procedure\_Name @Parameter1 = Value1, @Parameter2 = Value2;

**Modifying a Stored Procedure:**

To make changes to an existing stored procedure, use ALTER:

ALTER PROCEDURE Procedure\_Name

AS

BEGIN

-- New SQL Logic

END;

**Deleting a Stored Procedure:**

To remove a stored procedure:

DROP PROCEDURE Procedure\_Name;

1. . What are the uses of stored procedure
   * + 1. Automating Repetitive Tasks
       2. Performance Improvement
       3. Security
       4. Data Validation and Integrity
       5. Simplifying Complex Operations
       6. Code Reusability
       7. Centralized Business Logic
       8. Batch Processing
       9. Easier Debugging and Maintenance
       10. Dynamic Query Execution
       11. Transaction Management
       12. Simplified Application Logic
2. . What are the type of Stored procedure in SQL Server?

System Stored Procedures

User-Defined Stored Procedures

Temporary Stored Procedures

Extended Stored Procedures (Deprecated)

Remote Stored Procedures

CLR Stored Procedures

Recursive Stored Procedures

Parameterized Stored Procedures

System Extended Stored Procedures

1. . What are the type of Stored procedure in SQL Server?

1. System Stored Procedures

2. User-Defined Stored Procedures

3. Temporary Stored Procedures

4. Extended Stored Procedures (Deprecated)

5. Remote Stored Procedures

6. CLR Stored Procedures

7. Recursive Stored Procedures

8. Parameterized Stored Procedures

9. System Extended Stored Procedures

1. Explain about recursive stored procedures?

A **recursive stored procedure** is a stored procedure that calls itself to solve a problem. This approach is useful for handling problems that can be broken into smaller, repetitive tasks, such as traversing hierarchical data or performing iterative calculations.

Example: Traversing Hierarchical Data

Table Structure

CREATE TABLE Employees (

EmployeeID INT,

EmployeeName VARCHAR(100),

ManagerID INT

);

Sample Data

INSERT INTO Employees (EmployeeID, EmployeeName, ManagerID) VALUES

(1, 'CEO', NULL),

(2, 'Manager A', 1),

(3, 'Manager B', 1),

(4, 'Employee 1', 2),

(5, 'Employee 2', 2),

(6, 'Employee 3', 3);

Recursive Stored Procedure

CREATE PROCEDURE GetHierarchy (@ManagerID INT)

AS

BEGIN

-- Retrieve direct reports

SELECT EmployeeID, EmployeeName, ManagerID

FROM Employees

WHERE ManagerID = @ManagerID;

-- Recursive call to find reports of direct reports

DECLARE @EmployeeID INT;

DECLARE EmployeeCursor CURSOR FOR

SELECT EmployeeID FROM Employees WHERE ManagerID = @ManagerID;

OPEN EmployeeCursor;

FETCH NEXT FROM EmployeeCursor INTO @EmployeeID;

WHILE @@FETCH\_STATUS = 0

BEGIN

EXEC GetHierarchy @EmployeeID; -- Recursive call

FETCH NEXT FROM EmployeeCursor INTO @EmployeeID;

END;

CLOSE EmployeeCursor;

DEALLOCATE EmployeeCursor;

END;

**Execution**

To find the hierarchy under the CEO:

EXEC GetHierarchy 1;

1. Can a stored procedure call itself or recursive stored procedure?

Yes, a **stored procedure** can call itself, and this is referred to as a **recursive stored procedure**. However, recursion in stored procedures must be implemented carefully, as improper recursion can lead to infinite loops and stack overflow errors.

A recursive stored procedure is one that calls itself in its body. This is typically done in situations where the problem can be broken down into smaller, identical subproblems. Each call to the stored procedure works on a smaller part of the problem, and the recursion continues until a defined **base case** is met, which stops the recursion.

1. How much level SP nesting is possible?

In SQL Server, stored procedure (SP) nesting refers to the ability to call one stored procedure from another, and then potentially call additional stored procedures within those procedures. SQL Server has a limit to how many levels of stored procedure nesting are allowed.

SQL Server allows **a maximum of 32 levels of stored procedure nesting**. This means that you can call a stored procedure from another stored procedure, and so on, up to 32 levels deep. By default, SQL Server sets the **maximum depth of stored procedure nesting to 32**. If a stored procedure calls another stored procedure, and so on, the system will allow up to 32 levels of nesting.

If you exceed 32 levels of stored procedure nesting, SQL Server will return the following error:

Error: 213, Severity: 16, State: 2

Procedure 'procedure\_name' has caused a stack overflow.

You can change the maximum allowed levels of nesting using the SET MAXDOP option (for parallelism control), but SQL Server does **not** allow you to directly change the nesting level limit. The **32-level** limit is hard-coded into SQL Server and cannot be increased or decreased.

1. Have you ever created or used recursive stored procedure? Give example?

Yes, recursive stored procedures are commonly used to handle problems that involve hierarchical or nested data, such as processing organizational charts, file systems, or bill-of-material structures. A recursive stored procedure calls itself to break down a problem into smaller, more manageable sub-problems.

**Example of a Recursive Stored Procedure**

Let's take an example where we want to calculate the **factorial of a number** using a recursive stored procedure. The factorial of a number n is defined as:

* factorial(n) = n \* factorial(n-1) for n > 1
* factorial(1) = 1

CREATE PROCEDURE CalculateFactorial

@Number INT, -- Input parameter: number whose factorial is to be calculated

@Result INT OUTPUT -- Output parameter: stores the factorial value

AS

BEGIN

-- Base case: if Number is 1, return 1

IF @Number = 1

BEGIN

SET @Result = 1

RETURN

END

-- Recursive case: call the procedure itself with Number-1

DECLARE @SubResult INT

EXEC CalculateFactorial @Number - 1, @SubResult OUTPUT

-- Multiply the result from the recursive call by the current number

SET @Result = @Number \* @SubResult

END

To call this stored procedure for calculating the factorial of 5, you would execute the following:

DECLARE @FactorialResult INT

EXEC CalculateFactorial @Number = 5, @Result = @FactorialResult OUTPUT

SELECT @FactorialResult AS Factorial

1. What are the disadvantages of using a Stored Procedures?

While **stored procedures** offer many advantages like improved performance, code reuse, and better data security, there are also some disadvantages and limitations associated with their use. Here are the key disadvantages:

* + - 1. **Vendor Lock-In**
      2. **Complexity in Debugging and Testing**
      3. Limited Portability
      4. Increased Load on Database Server
      5. Maintenance Overhead
      6. Version Control Challenges
      7. Reduced Flexibility in Application Logic
      8. Security Risks
      9. Performance Issues with Complex Logic
      10. Limited Scalability

1. How to Optimize Stored Procedure Optimization?

Optimizing stored procedures in SQL Server (or any other RDBMS) is crucial to improve performance, ensure scalability, and reduce resource consumption. Here are the best practices and strategies for optimizing stored procedures:

**Minimize the Use of Cursors**

Use Proper Indexing

Avoid SELECT \* (Select All Columns)

Use the EXPLAIN Plan (Query Execution Plan)

Avoid Nested Queries When Possible

Optimize Temporary Tables

Limit the Use of Transactions

Use Efficient Data Types

Optimize Joins

Optimize WHERE Clauses

Avoid Using DISTINCT and ORDER BY Without Need

Use Set-Based Operations

Optimize Error Handling

Avoid Recompilation

Use Parameterized Queries

Optimize Data Insertion

Partition Large Tables

Analyze and Reorganize Indexes Regularly

1. How you will execute the stored procedure as a different user?

In SQL Server, to execute a stored procedure as a different user, you can use the EXECUTE AS clause, which allows you to temporarily impersonate a different user. This is useful when you want to execute a stored procedure with the permissions of another user without changing the ownership or modifying permissions.

1. **Using EXECUTE AS Clause**:

-- Execute the stored procedure as a different user

EXECUTE AS USER = 'username'; -- Impersonate the target user

EXEC stored\_procedure\_name; -- Execute the stored procedure

REVERT; -- Revert back to the original user

1. **Using EXECUTE AS LOGIN (for login-based execution)**:

-- Execute the stored procedure as a different SQL Server login

EXECUTE AS LOGIN = 'login\_name'; -- Impersonate the target login

EXEC stored\_procedure\_name; -- Execute the stored procedure

REVERT; -- Revert back to the original login

1. **Permissions Required**:

-- Example: Execute stored procedure as a user named 'admin\_user'

EXECUTE AS USER = 'admin\_user';

EXEC usp\_GetCustomerData; -- Stored procedure to retrieve customer data

REVERT; -- Revert to the original user

1. What is the difference between stored procedure and view in SQL Server

In SQL Server, both stored procedures and views are important database objects that serve different purposes. Here's a detailed comparison of the two:

Here’s a comparison of **Stored Procedures** and **Views** in SQL Server in a table format:

| **Feature** | **Stored Procedure** | **View** |
| --- | --- | --- |
| **Definition** | A precompiled set of SQL statements | A virtual table created from a query |
| **Execution** | Executed explicitly using EXEC | Queried like a regular table using SELECT |
| **Return Type** | Can return result sets, status codes, and output parameters | Always returns a result set |
| **Parameters** | Can accept input and output parameters | Does not accept parameters |
| **Data Modification** | Can modify data (INSERT, UPDATE, DELETE) | Usually read-only, but can modify simple data |
| **Purpose** | Business logic, automation, data manipulation | Data abstraction, simplifying complex queries |
| **Performance** | May involve more processing due to logic and transactions | Performance depends on query complexity |
| **Transaction Support** | Can manage transactions (BEGIN, COMMIT, ROLLBACK) | Cannot handle transactions |
| **Usage** | Used for automation, complex data manipulation, error handling, and transactions | Used to simplify query access or present data in a structured way |
| **Example** | CREATE PROCEDURE GetEmployeeDetails @DeptId INT AS SELECT \* FROM Employees WHERE DeptId = @DeptId; | CREATE VIEW EmployeeView AS SELECT ID, Name FROM Employees WHERE Department = 'HR'; |
| **Flexibility** | Highly flexible, supports business logic and multiple actions | Limited to returning data, cannot include business logic |
| **Modifiability** | Can accept parameters for dynamic behavior, can also update the data | Usually read-only unless it's a simple table query |

1. How do we recompile a stored procedure at run time?

To **recompile a stored procedure at runtime** in SQL Server, you can use the sp\_recompile system stored procedure. This procedure marks the specified stored procedure for recompilation the next time it is executed.

Syntax: EXEC sp\_recompile 'stored\_procedure\_name';

Example: EXEC sp\_recompile 'GetEmployeeDetails';

1. Can we use try and catch in stored procedure and function both? give and example?

**Stored Procedures**: Yes, you can use TRY and CATCH in stored procedures to handle errors and exceptions.

**Functions**: No, you cannot use TRY and CATCH in SQL Server user-defined functions. Functions in SQL Server do not support error handling with TRY...CATCH. Functions are designed to return a value and do not allow for control-of-flow constructs such as TRY...CATCH.

Example: 1. **Using TRY and CATCH in a Stored Procedure**:

CREATE PROCEDURE GetEmployeeDetails

@EmployeeId INT

AS

BEGIN

BEGIN TRY

-- Attempting to fetch employee details

SELECT \* FROM Employees WHERE EmployeeId = @EmployeeId;

-- Intentionally causing a divide-by-zero error for demonstration

DECLARE @Result INT;

SET @Result = 10 / 0;

END TRY

BEGIN CATCH

-- Error handling

SELECT ERROR\_MESSAGE() AS ErrorMessage;

SELECT ERROR\_NUMBER() AS ErrorNumber;

END CATCH

END;

2. **Example of SQL Function (Without TRY...CATCH)**:

CREATE FUNCTION GetEmployeeName (@EmployeeId INT)

RETURNS VARCHAR(100)

AS

BEGIN

-- Function that fetches employee name based on ID

DECLARE @EmployeeName VARCHAR(100);

SELECT @EmployeeName = EmployeeName FROM Employees WHERE EmployeeId = @EmployeeId;

RETURN @EmployeeName;

END;

1. Can we create Stored Procedure without "Begin" and "End" refer the below image and try to answers?

In SQL Server, **BEGIN and END** are typically used to group multiple statements together in a stored procedure, function, or other control-of-flow constructs (like IF, WHILE, etc.). However, in a **stored procedure**, it is **possible** to omit BEGIN and END if the stored procedure contains **only one statement**.

you can create a stored procedure without using BEGIN and END **if the stored procedure contains a single statement**. For example:

CREATE PROCEDURE GetEmployeeName

@EmployeeId INT

AS

SELECT EmployeeName FROM Employees WHERE EmployeeId = @EmployeeId;

Example of a Stored Procedure with BEGIN and END:

CREATE PROCEDURE GetEmployeeDetails

@EmployeeId INT

AS

BEGIN

-- Multiple statements require BEGIN and END

SELECT EmployeeName FROM Employees WHERE EmployeeId = @EmployeeId;

SELECT DepartmentName FROM Departments WHERE EmployeeId = @EmployeeId;

END;

Project Type(E) (Theorytical Questions):-

Now all SQL Server Interview Questions and answers at one place. Here we come with all Sql server questions asked during SQL Interview. This is not matter if you are a fresher or Experienced (DBA), these questions for all, From very simple to very complex

1. What is RDBMS?

RDBMS (Relational Database Management System) is a type of database management system that organizes data into **tables** (rows and columns). Data in an RDBMS is structured based on a relational model and uses **keys** to establish relationships between tables. Examples: MySQL, SQL Server, PostgreSQL, Oracle DB.

1. What are the properties of the relational tables?

Relational tables have the following properties:

1. **Rows (Tuples):** Each row is unique and represents a single record.
2. **Columns (Attributes):** Each column has a unique name and contains data of a specific type.
3. **Unique Key:** Each table has at least one primary key to uniquely identify rows.
4. **Atomicity:** Data stored in each cell is atomic (indivisible).
5. **No Order Dependence:** Rows are unordered; the sequence doesn't matter.
6. **Null Values:** Columns can store nulls if allowed.
7. What is Normalization?

Normalization is the process of organizing database tables to reduce **data redundancy** and improve **data integrity**. It involves dividing a large table into smaller tables and defining relationships between them.  
**Common Normal Forms:**

* **1NF:** Eliminate duplicate columns; ensure atomicity.
* **2NF:** Meet 1NF and remove partial dependencies.
* **3NF:** Meet 2NF and remove transitive dependencies.

1. What is De-Normalization?

De-normalization is the process of combining tables to improve **query performance** at the cost of potential **data redundancy**. It is used when:

* Queries involve many joins.
* There is a need for faster read operations.

1. What is Self Join?

A **self join** is a join where a table is joined to itself. It is used when:

* You need to compare rows within the same table.
* Example:

SELECT A.EmployeeID, A.ManagerID, B.EmployeeName AS ManagerName

FROM Employees A

JOIN Employees B ON A.ManagerID = B.EmployeeID;

1. What is Cross Join?

A **cross join** returns the Cartesian product of two tables. Every row in the first table is paired with every row in the second table.

SELECT \* FROM Table1 CROSS JOIN Table2;

1. What are the different types of cursor?

Cursors are used to fetch data row-by-row. Types include:

1. **Static Cursor:** A copy of the data is stored; changes in the underlying data are not visible.
2. **Dynamic Cursor:** Reflects all changes in the underlying data.
3. **Forward-Only Cursor:** Can only move forward.
4. **Keyset Cursor:** Keyset of rows is fixed, but data changes are visible.
5. What is a default constraint?

A **default constraint** specifies a default value for a column when no value is provided during insertion.

* Example:

ALTER TABLE Employees ADD CONSTRAINT Default\_Salary DEFAULT 50000 FOR Salary;

1. How to strip all non-alphabetic characters from string in SQL Server?

Use the PATINDEX and STUFF functions with a loop to remove non-alphabetic characters. Example:

DECLARE @Input NVARCHAR(MAX) = '123abc!@#DEF456';

WHILE PATINDEX('%[^a-zA-Z]%', @Input) > 0

SET @Input = STUFF(@Input, PATINDEX('%[^a-zA-Z]%', @Input), 1, '');

SELECT @Input AS AlphabetOnly;

1. How do I find a value anywhere in a SQL Server Database?.

Use the below query to search a value (YourValue) across all tables and columns:

DECLARE @Search NVARCHAR(MAX) = 'YourValue';

DECLARE @SQL NVARCHAR(MAX);

SET @SQL = '';

SELECT @SQL = @SQL +

'SELECT ''' + TABLE\_NAME + ''' AS TableName, ''' + COLUMN\_NAME + ''' AS ColumnName

FROM ' + TABLE\_NAME +

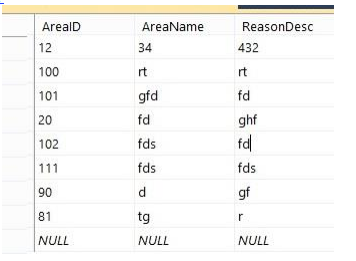
' WHERE [' + COLUMN\_NAME + '] LIKE ''%' + @Search + '%'' UNION ALL '

FROM INFORMATION\_SCHEMA.COLUMNS;

SET @SQL = LEFT(@SQL, LEN(@SQL) - 10); -- Remove the last UNION ALL

EXEC (@SQL);

1. Suppose that you have a table with AreaID but for that column you forgot set Identity. later you got it. Now you want to set identity without effecting the current records, and you only enabled Identity with seed 1. Then what would be the next value for AreaID in our below table example?



Based on the provided table, if the AreaID column is altered to include an identity property with a seed of 1, the identity column will automatically assign the next value after the highest current value in the column.

**Table Analysis:**

The current highest value in the AreaID column is 111.

**Expected Behavior:**

When the identity property is added, the next value in the AreaID column will be 112, since it increments by 1 starting from the highest existing value in the column.

**Answer:** The next value for AreaID will be **112**.

1. How to pass an array(or a table variable) into a SQL Server stored procedure?

You can pass a table-valued parameter to a stored procedure

CREATE TYPE MyTableType AS TABLE (Id INT, Name NVARCHAR(50));

CREATE PROCEDURE MyProcedure (@TableParam MyTableType READONLY)

AS

BEGIN

SELECT \* FROM @TableParam;

END;

-- Execute Procedure

DECLARE @MyTable MyTableType;

INSERT INTO @MyTable (Id, Name) VALUES (1, 'John'), (2, 'Jane');

EXEC MyProcedure @TableParam = @MyTable;

1. When should you use "with (nolock)"?
2. **"WITH (NOLOCK)"** is used to read uncommitted data (dirty reads) without locking the table. Use it:
3. When you need to prevent blocking in high-transaction environments.
4. For queries where 100% accuracy isn't critical (e.g., reporting on data changing frequently).
5. . Is it possible to insert into two tables at the same time?

No, you can't insert into two tables simultaneously in a single INSERT statement.

However, you can use a trigger on one table to insert data into another automatically.

1. Preferred Method of Storing Passwords In Database?

Use hashing algorithms like SHA-256 or bcrypt with a unique salt for each password.

Example: Use hashing functions in SQL Server such as HASHBYTES.

1. Can a view be updated/inserted/deleted? If Yes – under what conditions?

Yes, a view can be updated, inserted into, or deleted from if:

The view is created on a single table.

It doesn’t include complex joins, aggregations, or calculated columns.

There is no DISTINCT, GROUP BY, or UNION clause.

1. Where the integrity constraints are stored in data dictionary?
2. Integrity constraints are stored in the **sys.objects**, **sys.check\_constraints**, and **sys.foreign\_keys** system tables.
3. How many LONG columns are allowed in a table? Is it possible to use LONG columns in WHERE clause or ORDER BY?

Only 1 LONG column is allowed per table.

LONG columns cannot be used in WHERE or ORDER BY clauses.

1. What is Collation?

Collation determines the rules for sorting and comparing data, such as case sensitivity and accent marks.

Example: SQL\_Latin1\_General\_CP1\_CI\_AS (Case Insensitive, Accent Sensitive).

1. What is SQL Server Agent?

SQL Server Agent is a service for scheduling and automating tasks such as backups, running jobs, and alerts.

1. How to convert Rows to Columns in SQL Server?

Use **PIVOT** to transform rows into columns.

SELECT \*

FROM (

SELECT Year, Sales

FROM SalesTable

) AS SourceTable

PIVOT (

SUM(Sales)

FOR Year IN ([2021], [2022], [2023])

) AS PivotTable;

1. What is the difference between ROW\_NUMBER and Ranking function in SQL SERVER?

ROW\_NUMBER: Assigns a unique number to each row.

Ranking functions (RANK, DENSE\_RANK): Assign ranks based on order; handles ties differently.

21). What is the difference between ROW\_NUMBER and Ranking function in SQL SERVER?

ROW\_NUMBER: Assigns a unique number to each row.

Ranking functions (RANK, DENSE\_RANK): Assign ranks based on order; handles ties differently.

22). Difference between Where clause and Having clause in SQL Server?

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **WHERE Clause** | **HAVING Clause** | | Filters rows **before** grouping. | Filters groups **after** grouping. | | Used with SELECT, INSERT, UPDATE, DELETE. | Used with SELECT and GROUP BY. | |

23). What is data dictionary?

The data dictionary is a collection of metadata about database objects like tables, columns, constraints, and indexes.

24). Explain the use of SSIS in BI?

* SSIS (SQL Server Integration Services) is used for data integration, transformation, and migration in Business Intelligence solutions.

25). In Linq Query why from clause come first as select statement ?

The from clause establishes the data source to iterate over before applying filters or projections.

26). What is collation?

Collation determines the rules for sorting and comparing data, such as case sensitivity and accent marks.

Example: SQL\_Latin1\_General\_CP1\_CI\_AS (Case Insensitive, Accent Sensitive).

27). How to convert a table data in XML format in sql server?

SELECT \* FROM MyTable FOR XML AUTO, ELEMENTS;

28). What are the different types of locks?

* Shared (S)
* Exclusive (X)
* Update (U)
* Intent locks (IS, IX)
* Schema locks (Sch-S, Sch-M).

29). What are Pessimistic and optimistic Locks?

| **Pessimistic Lock** | **Optimistic Lock** |
| --- | --- |
| Locks resources for the duration of a transaction. | Assumes no conflicts; verifies changes at commit. |
| Used when high contention is expected. | Used when conflicts are rare. |

30). What is the difference between an Update and Exclusive lock?

| **Update Lock (U)** | **Exclusive Lock (X)** |
| --- | --- |
| Used when updating rows. | Prevents all access to rows. |
| Allows reading but blocks others from modifying. | Blocks both reading and writing. |

31). What is the difference between table and view in sql server?

| **Table** | **View** |
| --- | --- |
| Physical storage of data. | Virtual table; no data stored. |
| Allows DML operations. | DML operations may be limited. |

32). What is NOLOCK hint?

Allows reading uncommitted data (dirty reads) to avoid locking.

33). What is NOT NULL Constraint?

Ensures a column cannot have NULL values.

34). What is the difference between DATETIME2 and DATETIME?

| **DATETIME** | **DATETIME2** |
| --- | --- |
| Accuracy: 3.33ms. | Accuracy: 100ns. |
| Range: 1753 to 9999. | Range: 0001 to 9999. |

35). Explain Geography datatype in SQL Server?

Stores spatial data like latitude and longitude for GIS applications.

36). What is nolock hint in sql server 2008?

37). What are the different types of SQL Commands?

* **DDL:** Create, Alter, Drop.
* **DML:** Insert, Update, Delete.
* **DQL:** Select.
* **DCL:** Grant, Revoke.
* **TCL:** Commit, Rollback.

38). What is the difference between a DDL trigger and a DML trigger?

| **DDL Trigger** | **DML Trigger** |
| --- | --- |
| Triggers on schema changes. | Triggers on data modifications. |
| Example: CREATE, ALTER. | Example: INSERT, UPDATE. |

39). What are the types of transaction levels in SQL SERVER?

* **Read Uncommitted**
* **Read Committed**
* **Repeatable Read**
* **Serializable**
* **Snapshot**

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

 Default port: **1433**.

 To change: Use **SQL Server Configuration Manager**.

41). What is @@ERROR?

@@ERROR is a system function in SQL Server that returns the error number of the last Transact-SQL statement executed. If the previous statement ran successfully, @@ERROR returns 0. It is used in error-handling to capture and respond to errors in SQL scripts or stored procedures.

42). what is the difference between count(\*) and count(1)?

 **COUNT(\*):** Counts all rows in a table, including those with NULL values in any column.

 **COUNT(1):** Counts rows by evaluating the constant 1 for each row. The behavior is the same as COUNT(\*) but might vary in performance depending on the database engine.

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

 **Clustered Index:**

* Sorts and stores the rows of a table physically in the order of the index.
* Each table can have only one clustered index.
* Efficient for range queries.

 **Non-Clustered Index:**

* Maintains a separate structure from the data, with pointers to the rows in the table.
* A table can have multiple non-clustered indexes.
* Useful for queries that involve specific columns frequently.

44). What's the maximum size of a row?

The maximum size of a row in SQL Server is **8,060 bytes** for fixed-length data. However, this limit can be exceeded with variable-length data (e.g., VARCHAR(MAX)) using out-of-row storage.

45). What is HINT?

A **HINT** in SQL Server provides guidance to the query optimizer on how to execute a query. Examples include:

* **Query Hints:** e.g., OPTION (HASH JOIN)
* **Table Hints:** e.g., WITH (NOLOCK)
* **Join Hints:** e.g., INNER LOOP JOIN

46). How do you delete duplicate records?

WITH CTE AS (

SELECT

column1, column2,

ROW\_NUMBER() OVER (PARTITION BY column1, column2 ORDER BY id) AS rn

FROM TableName

)

DELETE FROM CTE WHERE rn > 1;

*Explanation:* The query uses ROW\_NUMBER() to identify duplicate rows and deletes those where the row number exceeds 1.

47). How do you delete all tables?

To delete all tables in a database:

1. Drop constraints like foreign keys first.
2. Then drop the tables.

EXEC sp\_MSforeachtable @command1 = "DROP TABLE ?";

48). What is Aggregate function?

Aggregate functions perform calculations on a set of values and return a single value. Examples:

* SUM(): Adds values.
* AVG(): Calculates the average.
* COUNT(): Counts rows.
* MAX() / MIN(): Finds the highest/lowest value.

49). What is the difference between a query and stored procedure?

 **Query:**

* A single SQL statement or a batch of statements executed at runtime.
* Requires parsing and compilation each time it runs.

 **Stored Procedure:**

* A precompiled collection of SQL statements stored in the database.
* Faster execution as it’s precompiled and reusable.

50). What will be the result of this query. select \* from TableName order by 1 .Will this query

throw an error?

No, it will not throw an error. ORDER BY 1 orders the result set based on the first column in the SELECT list.

50). How to get specific string/Date from a string in SQL Server using TSQL?

To extract a substring or date:

SELECT SUBSTRING(column\_name, start\_position, length) FROM TableName;

SELECT CONVERT(DATE, column\_name, 101) FROM TableName; -- For dates

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

* Single Column Index
* Composite Index (on multiple columns)
* Unique Index
* Filtered Index
* Clustered Index
* Non-Clustered Index

52). What are the different types of BACKUPs avaialabe in SQL Server 2005?

1. **Full Backup**: Backs up the entire database.
2. **Differential Backup**: Backs up only changes since the last full backup.
3. **Transaction Log Backup**: Backs up the transaction log for point-in-time recovery.
4. **File/Filegroup Backup**: Backs up individual files or filegroups.

53). What are the different locks in Sql Server?

* **Shared Lock (S):** For reading data.
* **Exclusive Lock (X):** For writing data.
* **Update Lock (U):** Prevents deadlocks during updates.
* **Intent Locks:** Reserved for future locking.
* **Schema Lock (Sch):** For operations involving metadata.

54). What is OLTP (Online Transaction Processing)?

OLTP refers to a system designed to handle a large number of short online transactions (e.g., insert, update, delete). Characteristics:

* Supports real-time data processing.
* Optimized for transactional integrity.

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

 **Primary Key:**

* Uniquely identifies each row in a table.
* Cannot contain NULL values.
* Only one primary key per table.

 **Unique Key:**

* Ensures all values in a column or combination of columns are unique.
* Can have multiple unique keys in a table.
* Allows a single NULL value.

56). What is the difference between DELETE and TRUNCATE statement?

57). What is SQL Profiler?

58). What are Data files?

Data files are physical files where SQL Server stores database objects and data. Types:

* **Primary Data File (.mdf):** Contains startup information and primary objects.
* **Secondary Data File (.ndf):** Optional, used to spread data across files.
* **Log File (.ldf):** Stores transaction logs.

59). What is difference between DELETE and TRUNCATE commands?

* **DELETE:**
  + Removes rows from a table with optional WHERE clause.
  + Logs each row deleted in the transaction log.
  + Slower than TRUNCATE.
* **TRUNCATE:**
  + Removes all rows from a table.
  + Does not log individual row deletions.
  + Faster and resets identity values.

60). When is the use of UPDATE\_STATISTICS command?

The UPDATE\_STATISTICS command is used to update the statistics for indexes or columns to ensure the query optimizer has the latest data distribution information, improving query performance.

61). What is the default port of Microsoft SQl Server?

The default port for Microsoft SQL Server is **1433** for TCP/IP connections.

62). What is Data Compression in sql server 2012?

Data compression reduces the amount of disk space used by tables and indexes. SQL Server 2012 supports:

* **Row Compression:** Eliminates unused space in fixed-length data types.
* **Page Compression:** Combines row compression with additional compression methods like prefix and dictionary compression.

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

* **WHERE CLAUSE:** Filters rows before grouping or aggregation.
* **HAVING CLAUSE:** Filters groups or aggregated data after the GROUP BY clause.

64). What is a linked server?

A linked server allows SQL Server to execute commands on another database server. It enables querying and joining data across multiple servers.

65). What database does SQL Server use for temporary tables?

SQL Server stores temporary tables in the **tempdb** system database.

66). What are different Types of Sub-Queries?

* **Single Row Subquery:** Returns one row (e.g., =, <, >).
* **Multiple Row Subquery:** Returns multiple rows (e.g., IN, ANY, ALL).
* **Correlated Subquery:** Refers to columns from the outer query.
* **Nested Subquery:** Subquery inside another subquery.

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

1. **Windows Authentication:** Uses Active Directory credentials.
2. **Mixed Mode Authentication:** Supports both Windows and SQL Server logins.

To change:

* Use SQL Server Management Studio (SSMS).
* Navigate to Security > Login Mode and switch modes.

68). Which command using Query Analyzer will give you the version of SQL server and

operating system?

SELECT @@VERSION;

69). What is service Broker?

Service Broker is a messaging framework in SQL Server that enables asynchronous communication between databases and applications. It is used for tasks like queuing, messaging, and event-driven architectures.

70). What is the difference between HAVING and WHERE clause?

71). What is a B-Tree?

A **B-Tree (Balanced Tree)** is a data structure used in indexes to organize and search data efficiently. It ensures balanced depth and provides fast lookups, inserts, and deletes.

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

73). What is an active database?

An **active database** is one that responds automatically to certain conditions or events using triggers, rules, and constraints.

74). .........?

75). Can a stored procedure call itself or recursive stored procedure? How much level SP

nesting is possible?

Yes, a stored procedure can call itself (recursion). The default nesting level is **32** in SQL Server.

76). What is Log Shipping?

Log Shipping is a disaster recovery solution that involves:

1. Backing up transaction logs from the primary server.
2. Copying them to a secondary server.
3. Restoring them on the secondary server to keep it in sync.

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

 SELECT COUNT(\*) FROM TableName;

 Using sys.dm\_db\_partition\_stats:

SELECT SUM(row\_count) FROM sys.dm\_db\_partition\_stats WHERE object\_id = OBJECT\_ID('TableName');

Use a filtered index or aggregate query for specific conditions.

78). What does it mean to have QUOTED\_IDENTIFIER ON? What are the implications of

having it OFF?

* **ON:** Allows using double quotes (") for column and table names.
* **OFF:** Double quotes are treated as string delimiters.  
  *Implication:* Turning it OFF may lead to syntax errors in queries using quoted identifiers.

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

* **Local Temporary Table (#Temp):** Available only in the session that created it.
* **Global Temporary Table (##Temp):** Accessible by all sessions until the session that created it disconnects.

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

**STUFF:** Inserts or replaces part of a string at a specific position

SELECT STUFF('abcdef', 2, 3, 'XYZ'); -- Result: aXYZef

**REPLACE:** Substitutes all occurrences of a substring

SELECT REPLACE('abcdef', 'bc', 'XYZ'); -- Result: aXYZdef

81). What is the difference between TEMP tables and Variable Tables?

* **TEMP Tables (#Temp):** Stored in tempdb, can have indexes, and are transactional.
* **Table Variables:** Stored in memory, have limited indexing capabilities, and are not transactional.

82). What are the different String Functions in SQL Server?

* LEN(), CHARINDEX(), SUBSTRING(), REPLACE(), STUFF(), LTRIM(), RTRIM(), LEFT(), RIGHT()

83). What are the different Index configurations a table can have?

* Clustered Index
* Non-Clustered Index
* Unique Index
* Filtered Index
* Full-Text Index

84). What is a Filtered Index?

A **Filtered Index** is a non-clustered index with a WHERE clause to filter the rows included.

85). What are indexed views?

Indexed views are views with a unique clustered index. They improve performance by storing the results in a materialized format.

86). What is the restriction on Indexed view?

* No DISTINCT, TOP, OUTER JOIN, or UNION.
* Base tables must be referenced with dbo.
* Schema binding is required.

87). What are Statistics? How can you find it in SQL Server?

Statistics provide data distribution information for indexes and columns, helping the optimizer create efficient query plans. Use: DBCC SHOW\_STATISTICS('TableName', 'IndexName');

88). How you can remove time part of datetime in SQL Server?

SELECT CAST(GETDATE() AS DATE);

89). NOT IN vs NOT EXISTS?

* **NOT IN:** Fails if NULL is in the subquery.
* **NOT EXISTS:** Works regardless of NULL values and is generally faster.

90). What are the different Mathematical Functions in SQL Server?

ABS(), CEILING(), FLOOR(), ROUND(), POWER(), SQRT(), LOG(), EXP()

91). What is the difference between COUNT and COUNT\_BIG?

* **COUNT:** Returns INT (max ~2 billion).
* **COUNT\_BIG:** Returns BIGINT for very large row counts.

92). What is WITH CHECK OPTION on view?

Ensures rows updated through the view satisfy the view's WHERE condition.

93). What is the difference between INSTEAD OF TRIGGER and AFTER Trigger?

* **INSTEAD OF TRIGGER:** Executes in place of the triggering action.
* **AFTER TRIGGER:** Executes after the triggering action.

94). How is the ACID property related to the database?

ACID (Atomicity, Consistency, Isolation, Durability) ensures reliable transactions:

* **Atomicity:** All or nothing.
* **Consistency:** Maintains data integrity.
* **Isolation:** Prevents concurrent issues.
* **Durability:** Changes persist after commit.

95). What are the different normalization forms?

* **1NF:** Eliminate duplicate columns.
* **2NF:** Eliminate partial dependencies.
* **3NF:** Eliminate transitive dependencies.
* **BCNF:** Higher normalization of 3NF.
* **4NF:** Eliminate multi-valued dependencies.
* **5NF:** Eliminate redundancy due to join dependencies.

96). What are different part of a SQL Page?

A SQL Page consists of the following main parts:

1. **Page Header:** Contains metadata about the page, such as page number, type, and allocation information.
2. **Data Rows:** Stores the actual data.
3. **Row Offset Array:** A collection of pointers pointing to the starting position of each row in the page.
4. **Free Space:** The unused space available for adding new rows or modifying existing ones.

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

* **Local Variable:** Declared using DECLARE within a batch or procedure and is accessible only within its scope.
* **Global Variable:** Predefined by the system (e.g., @@ERROR, @@ROWCOUNT) and is accessible throughout the SQL Server session.

98). What is PRIMARY KEY?

A **PRIMARY KEY** uniquely identifies each record in a table and:

* Cannot contain duplicate values.
* Automatically creates a clustered index unless specified otherwise.
* Cannot contain NULL values.

99). What is UNIQUE KEY constraint?

A **UNIQUE KEY** ensures that all values in a column are unique but:

* Allows a single NULL value per column.
* Can create a non-clustered index by default.

100). What is FOREIGN KEY?

A **FOREIGN KEY** establishes a relationship between two tables by referencing the **PRIMARY KEY** or **UNIQUE KEY** of another table. It ensures referential integrity by restricting invalid data insertion or deletion.

101). What is CHECK Constraint?

A **CHECK** constraint ensures that column values meet a specific condition, e.g.,: CHECK (Salary > 0)

102). What is the real time example of RIGHT Outer Join?

Retrieve employees who do not have projects assigned:

SELECT e.Name, p.ProjectName

FROM Employees e

RIGHT JOIN Projects p

ON e.ProjectID = p.ProjectID;

103). What are the types of database recovery models?

1. **Full:** Logs every transaction and allows full recovery.
2. **Bulk-logged:** Minimal logging for bulk operations but supports point-in-time recovery.
3. **Simple:** Does not log transactions and does not support point-in-time recovery.

104). What is NOT NULL Constraint?

The **NOT NULL** constraint ensures that a column cannot have NULL values.

105). What are the different data types in SQL Server?

* **Numeric:** INT, DECIMAL, FLOAT.
* **String:** CHAR, VARCHAR, TEXT.
* **Date/Time:** DATE, DATETIME, TIME.
* **Binary:** BINARY, VARBINARY.
* **Other:** XML, JSON, GEOGRAPHY.

106). What is blocking?

Blocking occurs when one transaction holds a lock on a resource, and another transaction is waiting for the lock to be released.

107). What are the different Date functions in SQL Server?

1. GETDATE(): Returns the current date and time.
2. DATEADD(): Adds a specified time interval to a date.
3. DATEDIFF(): Returns the difference between two dates.
4. FORMAT(): Formats a date.

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

Use a local variable to capture both:

DECLARE @Error INT, @RowCount INT;

UPDATE TableName SET Column = Value;

SET @Error = @@ERROR;

SET @RowCount = @@ROWCOUNT;

109). What is a Scheduled Jobs or what is a Scheduled Tasks?

A **Scheduled Job** in SQL Server is an automated task created using SQL Server Agent to run queries, backup databases, etc., on a defined schedule.

110). What are the advantages of using Stored Procedures?

1. Increased performance (precompiled).
2. Reusability and modularity.
3. Enhanced security with permissions.
4. Reduces network traffic.

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

for?

It is called a **Heap Table**, which is primarily used for temporary or unstructured data.

112). Can SQL Servers linked to other servers like Oracle?

Yes, using **Linked Servers**, SQL Server can connect to other RDBMS like Oracle, MySQL, etc.

113). What is BCP? When does it used?

**BCP (Bulk Copy Program)** is a command-line tool to import/export large data volumes between SQL Server and files.

114). How to implement one-to-one, one-to-many and many-to-many relationships while

designing tables?

1. **One-to-One:** Use a unique foreign key.
2. **One-to-Many:** Use a foreign key in the child table referencing the parent table.
3. **Many-to-Many:** Use a junction table with foreign keys referencing both tables.

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

execution plan?

An **Execution Plan** visualizes how SQL Server executes queries, showing operations like scans, seeks, joins, etc. View it using SET SHOWPLAN\_XML or Query Analyzer.

116). What is the difference between CHAR and VARCHAR?

 **CHAR:** Fixed-length storage.

 **VARCHAR:** Variable-length storage, conserving space.

117). What is the difference between VARCHAR and VARCHAR (max)?

* VARCHAR: Maximum length is 8000 bytes.
* VARCHAR(MAX): Stores up to 2 GB of data.

118). ..?

119). What are different types of Collation Sensitivity?

* **Case Sensitivity (CS):** Differentiates uppercase and lowercase.
* **Accent Sensitivity (AS):** Differentiates accented characters.
* **Kana Sensitivity (KS):** Differentiates Japanese Kana.
* **Width Sensitivity (WS):** Differentiates single-byte and double-byte characters.

120). What is a Stored Procedure?

A **Stored Procedure** is a precompiled group of SQL statements that can be executed as a single unit.

121). What are the different types of collation sensitivity?

122). What is dirty read?

A **dirty read** occurs when a transaction reads uncommitted data from another transaction, risking inconsistencies.

123). How do you check collation and compatibility level for a database?

SELECT DATABASEPROPERTYEX('DatabaseName', 'Collation') AS Collation;

SELECT compatibility\_level FROM sys.databases WHERE name = 'DatabaseName';

124). What is a covered Index?

An index that includes all columns required to satisfy a query, avoiding lookups to the base table.

125). What is the maximum row size for a table?

The maximum row size is **8,060 bytes** (excluding LOB data like VARCHAR(MAX)).

126). When I delete data from a table, Does SQL Server reduce the size of table?

No, deleting data only frees space within the table. Use DBCC SHRINKFILE to reduce size.

127). How do you rebuild master database?

 Start SQL Server in **single-user mode**.

 Use the setup.exe command with REBUILDDATABASE option.

128). What is PIVOT and UN-PIVOT?

 **PIVOT:** Converts rows to columns.

 **UNPIVOT:** Converts columns to rows.

-- Pivot Example

SELECT \* FROM Table

PIVOT (SUM(Value) FOR ColumnName IN ([Column1], [Column2])) AS PivotTable;

129). What is EXCEPT operation?

 **EXCEPT** is used to return rows from the first query that are not present in the second query.

 Both queries must have the same number of columns and compatible data types.

 Example: SELECT EmployeeID FROM Employees

EXCEPT

SELECT EmployeeID FROM Managers;

130). What are GROUPING Sets?

**GROUPING SETS** are a SQL feature that allows multiple groupings in a single query, simplifying complex queries with GROUP BY.

SELECT Department, Year, SUM(Sales) AS TotalSales

FROM Sales

GROUP BY GROUPING SETS ((Department, Year), (Department), ());

131). What are the row constructors inside SQL Server?

* Row constructors allow inserting multiple rows in a single INSERT statement.
* Example: INSERT INTO Employees (ID, Name, Department)
* VALUES (1, 'Alice', 'HR'), (2, 'Bob', 'IT');

132). What are table valued parameters?

Table-valued parameters allow you to pass a table as an input parameter to a stored procedure or function.

CREATE TYPE EmployeeTableType AS TABLE (ID INT, Name NVARCHAR(50));

CREATE PROCEDURE InsertEmployees (@EmployeeTable EmployeeTableType READONLY)

AS

BEGIN

INSERT INTO Employees SELECT \* FROM @EmployeeTable;

END;

133). How do you identify and resolve deadlock?

 **Identify:**

* Use SQL Profiler or Extended Events to capture deadlock graphs.
* Query the sys.dm\_tran\_locks and sys.dm\_exec\_requests views.

 **Resolve:**

* Implement proper indexing.
* Use NOLOCK for read-only queries.
* Minimize transaction scope.

134). What are Spare columns?

 Sparse columns are used to efficiently store NULL values in tables, reducing storage requirements.

 Example: CREATE TABLE Products (

ProductID INT PRIMARY KEY,

Description NVARCHAR(100) SPARSE NULL

);

135). How to get last inserted id?

 Use SCOPE\_IDENTITY() to retrieve the last inserted ID in the same scope.

 Example: INSERT INTO Employees (Name) VALUES ('Alice');

SELECT SCOPE\_IDENTITY();

136). Which is database shrinking?

 **Database shrinking** reclaims unused space and reduces database file size.

 Use cautiously as it can cause fragmentation.

DBCC SHRINKDATABASE ('DatabaseName');

137). What is the maximum number of columns a table can have?

A table can have up to **1,024 columns** in SQL Server.

138). What are included columns?

Included columns are non-key columns added to a non-clustered index to improve query performance without being part of the index key.

CREATE NONCLUSTERED INDEX IX\_Employee\_Name ON Employees (LastName)

INCLUDE (FirstName, HireDate);

139). What is INTERSECT operation?

**INTERSECT** returns rows that are common to both queries.

SELECT EmployeeID FROM Employees

INTERSECT

SELECT EmployeeID FROM Managers;

140). What is RAID?

* **RAID (Redundant Array of Independent Disks)** is a data storage virtualization technology for redundancy and performance:
  + **RAID 0:** Striping for performance.
  + **RAID 1:** Mirroring for redundancy.
  + **RAID 5:** Parity-based redundancy.

141). What are the limitations of Select Into clause?

* Cannot specify constraints on the new table.
* Cannot include indexes.
* Creates the table dynamically, so data types are inferred.

142). What is FileStream in SQL Server?

**FileStream** allows storage and management of unstructured data (e.g., images, documents) in the file system, with a pointer in SQL Server.

143). What do you mean by TableSample?

**TABLESAMPLE** retrieves a random sample of rows from a table.

SELECT \* FROM Employees TABLESAMPLE (10 PERCENT);

144). What are the disadvantages of using Stored Procedure?

* Hard to debug and maintain.
* Vendor lock-in (SQL Server-specific logic).
* Performance degradation if overused for logic that should be in the application layer.

145). What is the difference between COMMIT and ROLLBACK?

* **COMMIT:** Saves all changes in a transaction permanently.
* **ROLLBACK:** Undoes all changes made in a transaction.

146). What is Transaction?

* A transaction is a logical unit of work that ensures **ACID** properties:
  + **A**tomicity
  + **C**onsistency
  + **I**solation
  + **D**urability

147). How do you do Error Handling in SQL server?

Use TRY...CATCH blocks:

BEGIN TRY

INSERT INTO Employees (Name) VALUES ('Alice');

END TRY

BEGIN CATCH

PRINT ERROR\_MESSAGE();

END CATCH;

148). What is RAISEERROR?

**RAISEERROR** generates a custom error message.

RAISERROR ('Custom error message', 16, 1);

149). What is SQL INJECTION?

* SQL Injection is a security vulnerability where malicious SQL code is injected into queries. To prevent it:
  + Use parameterized queries.
  + Validate inputs.

150). How do you find list of schema names and table names from the database?

SELECT TABLE\_SCHEMA, TABLE\_NAME

FROM INFORMATION\_SCHEMA.TABLES;

151). Why can be there only one clustered index?

* A clustered index defines the physical order of rows in a table, and only one physical order is possible.

152). Can we create view on TEMP table?

Yes, but only if the temp table is a **global temporary table** (##TempTable).

153). Can we use constraints on TEMP Table?

Yes, constraints like **PRIMARY KEY** or **NOT NULL** can be applied.

154). How to recompile a stored procedure at run time?

EXEC sp\_recompile 'ProcedureName';

155). Does the order of columns in UPDATE statement matters?

No, the order does not matter.

156). What are the different types of System database inside SQL Server?

* **Master:** System-level information.
* **Model:** Template for new databases.
* **MSDB:** Used by SQL Server Agent.
* **TempDB:** Temporary objects.

157). What is the use of Transaction Log file?

Records all transactions and changes to the database for recovery purposes.

158). What is the difference between view and Materialized view?

* **View:** Logical query stored in memory.
* **Materialized View:** Physically stores the query result for faster access.

159). What is database mirroring?

Database mirroring provides high availability by maintaining a copy of the database on a mirror server.

160). How does SQL Server database engine work?

* SQL Server uses a query processor and storage engine:
  + Query processor optimizes and executes queries.
  + Storage engine manages data retrieval, storage, and indexing.

161). What is the maximum number of indices per table?

Up to **999 non-clustered** indexes and **1 clustered** index.

162). What are the purpose of Normalisation?

1. Reduce redundancy.
2. Ensure data integrity.
3. Improve query performance.

163). What does the Merge Statement do?

Combines INSERT, UPDATE, and DELETE operations in one statement.

164). What are the new data types introduced in SQL Server 2012?

1. **DATEFROMPARTS**
2. **TIMEFROMPARTS**
3. **DATETIMEOFFSET**
4. **SEQUENCE**

165). Define Hierarchyid data type?

A specialized data type used to represent hierarchical data, such as organizational charts or file systems.

166). What are synonyms?

Synonyms in SQL are database objects that serve as an alias or alternative name for other database objects like tables, views, procedures, or functions. They help to simplify complex queries or provide access to objects in different schemas or databases.

167). What is CTE?

A CTE is a temporary result set that is defined within the execution scope of a SELECT, INSERT, UPDATE, or DELETE statement. It can be referenced within the query and helps in simplifying complex joins and subqueries.

168). What are the advantages of CTE?

* Simplifies complex queries.
* Increases readability and maintainability.
* Can be used recursively.
* Can be referenced multiple times within the same query.

169). Can we write Sub Queries into Simple Select Using join and CTE?

Yes, subqueries can often be replaced by JOIN operations or CTEs, which can simplify queries, improve readability, and sometimes enhance performance.

170). Can CTE be recursive? Till what level it can be nested?

Yes, CTEs can be recursive. They are often used for hierarchical data or tree-like structures. The recursion can be limited by a MAXRECURSION option or by the system's limitations, but recursion usually goes up to 100 iterations by default unless specified otherwise.

171). What is LINQ?

Language Integrated Query (LINQ) is a set of methods that allows querying and manipulating data from different sources like collections, databases, XML, and more, directly from C# or other .NET languages.

172). What is XML?

XML (eXtensible Markup Language) is a markup language used for storing and transporting data in a structured, human-readable format. It is commonly used in data exchange between systems.

173). How can you find tables without Indexes?

You can find tables without indexes by querying the sys.tables and sys.indexes system views to identify tables that have no associated indexes.

SELECT t.name AS TableName

FROM sys.tables t

LEFT JOIN sys.indexes i ON t.object\_id = i.object\_id

WHERE i.index\_id IS NULL;

174). How do you find the size of index?

To find the size of an index, you can use the following query that utilizes the sys.dm\_db\_index\_physical\_stats DMV to get the size of an index:

SELECT

OBJECT\_NAME(IXOS.OBJECT\_ID) AS TableName,

I.name AS IndexName,

SUM(IXOS.LEAF\_INSERT\_COUNT) AS IndexSize

FROM

SYS.DM\_DB\_INDEX\_OPERATIONAL\_STATS(NULL, NULL, NULL, NULL) IXOS

INNER JOIN SYS.INDEXES I

ON I.OBJECT\_ID = IXOS.OBJECT\_ID

AND I.INDEX\_ID = IXOS.INDEX\_ID

GROUP BY

OBJECT\_NAME(IXOS.OBJECT\_ID),

I.name;

175). How do you copy data from one table to another table?

You can copy data from one table to another using the INSERT INTO ... SELECT statement:

INSERT INTO destination\_table (column1, column2, ...)

SELECT column1, column2, ...

FROM source\_table;

176). What is ROW\_NUMBER()?

ROW\_NUMBER() is a window function in SQL that assigns a unique sequential number to rows within a result set, based on the ordering specified.

177). What is ROLLUP clause?

The ROLLUP clause in SQL is used in conjunction with GROUP BY to return a result set that includes both the grouped data and subtotals (aggregate values) for each grouping level.

178). What are ranking functions?

Ranking functions are used to assign a rank to each row within a partition of a result set. Common ranking functions include:

* ROW\_NUMBER()
* RANK()
* DENSE\_RANK()
* NTILE()

179). How do you stop a log file from growing too big?

To prevent a log file from growing too large:

* Regularly backup the transaction log.
* Shrink the log file after the backup.
* Set appropriate autogrowth settings.
* Monitor and manage the size of the transaction log.

180). How do we use DBCC commands?

DBCC (Database Console Commands) are used for various maintenance tasks like checking integrity, repairing databases, and managing database consistency. For example, you can use DBCC CHECKDB to check the integrity of a database.

DBCC CHECKDB('DatabaseName');

181). What is CDC?

CDC is a SQL Server feature that tracks changes (insert, update, delete) to data in a table and stores that information in change tables for later processing.

182). What are the main performance differences between varchar and nvarchar SQL Server

data types?

* varchar stores non-Unicode characters (single-byte), while nvarchar stores Unicode characters (double-byte).
* nvarchar takes up more space because it uses 2 bytes per character compared to varchar's 1 byte.
* Use nvarchar when supporting multiple languages or special characters.

183). What is the difference between varchar and nvarchar?

* varchar: Stores non-Unicode data (1 byte per character).
* nvarchar: Stores Unicode data (2 bytes per character).

184). How do I insert multiple rows WITHOUT repeating the “INSERT INTO dbo.Blah” part of

the statement?

You can insert multiple rows using a single INSERT statement:

INSERT INTO dbo.Blah (column1, column2, ...)

VALUES

(value1, value2, ...),

(value3, value4, ...),

(value5, value6, ...);

185). What is the difference between Index Seek and Index Scan?

* **Index Seek**: SQL Server uses an index to quickly locate specific rows, improving performance.
* **Index Scan**: SQL Server scans the entire index to retrieve the rows, which can be slower for large datasets.

186). Can we insert data if the clustered index is disabled?

No, you cannot insert data into a table if the clustered index is disabled. The clustered index determines the physical order of the data in the table.

187). What is standby server?

A standby server is a secondary server that acts as a backup to the primary server. It can take over in case of server failure to maintain high availability.

188). How do you disable Index?

To disable an index, you can use the ALTER INDEX statement:

ALTER INDEX index\_name ON table\_name DISABLE;

189). How do you enable index?

To enable an index that has been disabled, you can use:

ALTER INDEX index\_name ON table\_name REBUILD;

190). What are the steps to create a Table Partition?

To create table partitioning:

1. Create a partition function.
2. Create a partition scheme.
3. Create a partitioned table by specifying the partition scheme.

191). What are the basics of Table Partitioning in SQL Server?

Table partitioning divides large tables into smaller, manageable pieces (partitions), usually based on a key column like a date. It helps improve query performance and maintenance.

192). How do you copy tables, schemas, and views from one sql server to another SQL

Server?

You can use SQL Server Management Studio (SSMS) to generate scripts or use the Generate Scripts feature to copy tables, schemas, and views from one server to another.

193). Where are SQL Server user names and passwords stored in?

SQL Server stores usernames and passwords in the sys.syslogins system table, but passwords are encrypted.

194). Can we have Triggers on Temp table and Variable Table?

Yes, you can create triggers on temporary tables but not on table variables.

195). What is the syntax for encrypting a column in SQL Server?

To encrypt a column, you can use the ENCRYPTBYKEY function along with a symmetric key. For example:

CREATE SYMMETRIC KEY KeyName WITH ALGORITHM = AES\_256 ENCRYPTION BY PASSWORD = 'password';

OPEN SYMMETRIC KEY KeyName DECRYPTION BY PASSWORD = 'password';

UPDATE TableName

SET EncryptedColumn = ENCRYPTBYKEY(KEY\_GUID('KeyName'), ColumnToEncrypt);

196). How do you send email on SQL Server?

You can send an email using sp\_send\_dbmail in SQL Server if Database Mail is configured:

EXEC msdb.dbo.sp\_send\_dbmail

@profile\_name = 'MailProfile',

@recipients = 'recipient@example.com',

@subject = 'Subject',

@body = 'Email body';

197). What is query optimization ?

Query optimization is the process of improving the performance of SQL queries, making them faster by using the most efficient execution plans.

198). How many maximum Identity columns we can have in a single table?

A table can have only one IDENTITY column.

199). How to find all Triggers in database?

You can find all triggers in a database by querying the sys.triggers system view:

SELECT \* FROM sys.triggers;

200). Can we add constraint on Variable Table?

No, you cannot add constraints (like PRIMARY KEY, FOREIGN KEY, CHECK, etc.) on table variables in SQL Server.

201). What is Schema?

A schema is a logical container in SQL Server that holds database objects like tables, views, and stored procedures. It helps in organizing and managing database objects.

202). Can we create multiple constraints on a single column?

Yes, you can create multiple constraints on a single column, such as NOT NULL, UNIQUE, and CHECK.

203). Can we rename table and column using ALTER command?

Yes, you can rename a table or column using ALTER commands:

* Rename table: sp\_rename 'OldTableName', 'NewTableName';
* Rename column: sp\_rename 'TableName.OldColumnName', 'NewColumnName', 'COLUMN';

204). How to rename Table and Column?

You can rename tables and columns using the sp\_rename stored procedure, as shown above.

205). How to rename Database?

To rename a database:

ALTER DATABASE OldDatabaseName MODIFY NAME = NewDatabaseName;

206). What is the disadvantage of Index?

The main disadvantage of indexes is that they slow down INSERT, UPDATE, and DELETE operations because the index needs to be updated whenever the data changes.

207). How can we find the table size?

You can find the table size using the following query:

EXEC sp\_spaceused 'TableName';

208). How to find N highest salary?

To find the N highest salaries, you can use the ROW\_NUMBER() function:

WITH RankedSalaries AS (

SELECT Salary, ROW\_NUMBER() OVER (ORDER BY Salary DESC) AS RowNum

FROM Employees

)

SELECT Salary

FROM RankedSalaries

WHERE RowNum <= N;

209). What is the difference between data mirroring and log shipping?

* **Data Mirroring**: Provides real-time redundancy by maintaining a mirror copy of the database on a standby server.
* **Log Shipping**: Involves periodically backing up and transferring transaction logs to a standby server, typically for disaster recovery.

210). What are the different backup options within SQL Server?

SQL Server offers:

* **Full backup**: Backs up the entire database.
* **Differential backup**: Backs up changes since the last full backup.
* **Transaction log backup**: Backs up the transaction log to ensure data consistency.

211). How to add DEFAULT constraint on existing column? Write query

To add a DEFAULT constraint:

ALTER TABLE TableName

ADD CONSTRAINT ConstraintName DEFAULT DefaultValue FOR ColumnName;

212). How to add NOT NULL constraint on existing column? Write query

To add a NOT NULL constraint:

ALTER TABLE TableName

ALTER COLUMN ColumnName DataType NOT NULL;

213). How do you find why query is running slow?

To find why a query is slow, you can:

* Analyze execution plans.
* Check indexes.
* Look at the query's I/O and CPU usage.
* Use SQL Server Profiler or DMVs.

214). How to create foreign key constraints on temporary table?

You can create foreign key constraints on temporary tables like this:

CREATE TABLE #TempTable (ID INT, RefID INT);

ALTER TABLE #TempTable ADD CONSTRAINT FK\_Ref FOREIGN KEY (RefID) REFERENCES AnotherTable(ID);

215). What is Dynamic SQL?

Dynamic SQL is SQL code that is generated and executed at runtime, allowing for more flexible queries based on varying conditions.

216). Can we create a table name as Table?

Yes, you can create a table with the name Table, but it is not recommended as it could conflict with SQL reserved keywords. Always use a more descriptive name.

217). What is the difference between CAST and CONVERT Function?

* CAST: ANSI standard, simpler syntax. Example: CAST(123 AS VARCHAR).
* CONVERT: SQL Server-specific, supports style codes for formatting. Example: CONVERT(VARCHAR, GETDATE(), 101).

218). What Are Binary String Data Types?

Binary string data types store binary data (e.g., images, documents) and include BINARY, VARBINARY, and IMAGE (deprecated).

219). How to execute Dynamic SQL?

Using EXEC or sp\_executesql.

EXEC('SELECT \* FROM TableName');

220). Can we use Variable table in Dynamic SQL?

Yes, by defining it before executing dynamic SQL and referring to it in the query.

221). How to execute queries stored in a table?

Use a cursor or EXEC.

Example:

DECLARE @sql NVARCHAR(MAX);

SELECT @sql = QueryColumn FROM QueriesTable;

EXEC(@sql);

222). How can you capture the length of column when it is text, image and ntext data type?

Use the DATALENGTH() function.

Example: SELECT DATALENGTH(ColumnName) FROM TableName;

223). Is it possible to import data using TSQL?

Yes, using BULK INSERT or OPENROWSET.

Example: BULK INSERT TableName FROM 'FilePath' WITH (FIELDTERMINATOR = ',', ROWTERMINATOR = '\n');

224). How can you prevent TSQL code from running on a Production server?

Implement checks like: IF @@SERVERNAME = 'ProductionServerName'

THROW 50000, 'Not allowed on Production', 1;

225). Define Unique Key?

A constraint ensuring that all values in a column (or group of columns) are unique. Allows one NULL.

226). What is the use of SP\_Helptext , SP\_HelpIndex stored procedure?

* sp\_helptext: Shows the definition of an object (e.g., views, stored procedures).
* sp\_helpindex: Shows index details of a table.

227). Can we change order of triggers?

No direct control, but you can recreate triggers with dependencies in mind.

228). What do you understand by Joins in SQL Server?

Combines rows from two or more tables:

* INNER JOIN: Matching rows only.
* LEFT JOIN: All rows from the left table.
* RIGHT JOIN: All rows from the right table.
* FULL JOIN: All rows from both tables.

229). How to disable Auto Commit in SQL Server?

SQL Server does not support disabling auto-commit but you can explicitly start a transaction:

BEGIN TRANSACTION;

230). Can we recover deleted data?

Yes, if backups or transaction logs are available.

231). How to delete Top 100 records from a table?

DELETE TOP (100) FROM TableName;

232). How to delete two tables using one Drop command?

DROP TABLE Table1, Table2;

233). Can I create CTE in Trigger?

Yes, you can use Common Table Expressions (CTEs) in triggers.

234). Can we create Variable table in Trigger?

Yes, you can use table variables inside triggers.

235). Can we use cursors in Trigger?

Yes, but it is not recommended due to performance impact.

236). Can we call Stored Procedure in Trigger?

Yes, you can execute stored procedures from triggers.

236). Can we call Stored Procedure in Trigger?

Yes, you can execute stored procedures from triggers.

236.1). Can we call Stored Procedure in a View?

No, you cannot call stored procedures directly in views.

237). Can I create Triggers on TEMP table?

No, triggers cannot be created on temporary tables.

238). Can we use PRINT Command in Triggers?

Yes, but the output is visible only in debugging scenarios.

239). How Triggers are fired?

Triggers are fired automatically after the associated INSERT, UPDATE, or DELETE operation.

240). Why do we use DECLARE for cursor and Variable table?

* Declaring ensures proper scope management and memory allocation.
* Makes the code readable and reusable.

241). How to take database online –offline?

ALTER DATABASE DatabaseName SET ONLINE;

ALTER DATABASE DatabaseName SET OFFLINE;

242). How to copy data using Bulk copy when columns data type doesn’t match?

Use FORMATFILE to define column mappings.

243). What is SP\_Configure commands and SET commands?

* sp\_configure: Configures global server settings.
* SET: Configures session-level settings (e.g., SET NOCOUNT ON).

244). Can Inserted table have multiple records?

Yes, during bulk operations.

245). Can we perform DML & DDL operation on Inserted and Deleted tables?

No, INSERTED and DELETED are virtual, read-only tables.

246). What is the advantage of Index?

Improves query performance by enabling faster lookups.

247). Which is fast UNION or UNION ALL?

UNION ALL is faster as it skips duplicate removal.

248). Can we create clustered index on view?

Yes, but the view must be indexed.

249). Can we create computed columns?

Yes, using expressions in the CREATE TABLE statement.

250). Can we change the Column Sequence order inside table?

No direct method; recreate the table with the desired order.

251). Truncate is DDL or DML?

TRUNCATE is a DDL command as it changes table structure by resetting identity and removing all rows without logging individual deletions.

252). Can we create view from view?

Yes, views can be created based on existing views.

253). What the difference between UNION and UNIONALL?

* UNION: Removes duplicates, performs sorting (slower).
* UNION ALL: Retains duplicates, no sorting (faster).

254). How to join two tables from different database?

SELECT \*

FROM Database1.dbo.Table1 AS T1

JOIN Database2.dbo.Table2 AS T2 ON T1.ID = T2.ID;

255). Can we use ORDER BY Clause in UNION?

Yes, but only at the end of the combined query.

256). What is difference between Deterministic and Non Deterministic Functions?

* **Deterministic**: Always returns the same output for the same input (LEN, ROUND).
* **Non-Deterministic**: Output depends on factors like system state (GETDATE, NEWID).

257). What is Synchronous and asynchronous function?

* **Synchronous**: Execution waits for function completion.
* **Asynchronous**: Execution continues without waiting for the function.

258). Can we add Identity column after creating the table?

No, but you can add a new table with an identity column, copy data, and rename tables.

259). Can we drop Identity column?

No, drop and recreate the table to remove it.

260). Can we store Image, MP3 and binary data in SQL Server?

Use VARBINARY(MAX) or FILESTREAM.

261). How can we disable Identity column?

Not directly possible; use SET IDENTITY\_INSERT TableName ON/OFF for manual insertions.

262). Can Foreign key column have NULL?

Yes, unless defined as NOT NULL.

263). How to find column description of a table?

Use sp\_help or query system tables:

SELECT COLUMN\_NAME, COLUMN\_DEFAULT, IS\_NULLABLE

FROM INFORMATION\_SCHEMA.COLUMNS WHERE TABLE\_NAME = 'TableName';

264). How to delete Duplicate records?

Using ROW\_NUMBER():

WITH CTE AS (

SELECT \*, ROW\_NUMBER() OVER(PARTITION BY ColumnName ORDER BY ID) AS RN

FROM TableName

)

DELETE FROM CTE WHERE RN > 1;

265). How to find EmployeeDetails hired in last month?

SELECT \*

FROM EmployeeDetails

WHERE HireDate >= DATEADD(MONTH, -1, GETDATE()) AND HireDate < GETDATE();

266). How to find all rows that contains only numeric data?

SELECT \*

FROM TableName

WHERE ColumnName NOT LIKE '%[^0-9]%';

267). How to find primary key name if not given by the user for a particular column?

SELECT CONSTRAINT\_NAME

FROM INFORMATION\_SCHEMA.TABLE\_CONSTRAINTS

WHERE TABLE\_NAME = 'TableName' AND CONSTRAINT\_TYPE = 'PRIMARY KEY';

268). Can we add two columns using ALTER command?

ALTER TABLE TableName

ADD Column1 DataType, Column2 DataType;

269). How to get row number without ROW\_NUMBER function?

SELECT @RowNum := @RowNum + 1 AS RowNumber, ColumnName

FROM TableName, (SELECT @RowNum := 0) AS R;

270). What is Partitioned View?

Combines data from multiple tables to act as a single table for querying. Used for scaling and performance.

271). What is the difference between UNIQUE Key and Primary Key?

* **UNIQUE Key**: Allows one NULL, ensures uniqueness.
* **Primary Key**: No NULLs, ensures uniqueness, single per table.

272). How to find who deleted/ dropped from Transaction log?

Use SQL Server Audit or third-party tools. Check system functions like fn\_dblog.

273). Can we ALTER two columns using ALTER command?

ALTER TABLE TableName

ALTER COLUMN Column1 DataType, ALTER COLUMN Column2 DataType;

274). How to clean Buffer in SQL Server?

DBCC DROPCLEANBUFFERS;

275). How to clear Execution Plan cache?

DBCC FREEPROCCACHE;

276). How can we check for existence of any object in database?

IF OBJECT\_ID('TableName', 'U') IS NOT NULL

PRINT 'Exists';

277). What is meant by differed name resolution in SQL Server?

Allows SQL Server to defer checking object existence until runtime (e.g., for stored procedures).

278). How to find Organization EmployeeDetail Hierarchy using SQL?

WITH CTE AS (

SELECT EmployeeID, ManagerID FROM Employees WHERE ManagerID IS NULL

UNION ALL

SELECT E.EmployeeID, E.ManagerID FROM Employees E INNER JOIN CTE C ON E.ManagerID = C.EmployeeID

)

SELECT \* FROM CTE;

279). How does a recursive CTE works?

Defining an anchor query (base case) and a recursive query. Repeats until no more results.

280). What is Auditing inside SQL Server?

Tracks events (e.g., login, data changes) using SQL Server Audit.

281). What is the difference between GETDATE() and SYSDATETIME()?

* GETDATE(): Returns current date/time with less precision.
* SYSDATETIME(): More precision (up to nanoseconds).

282). How do you check if Automatic Statistic Update is enabled for a database?

Use the is\_auto\_update\_stats\_on column in sys.databases:

SELECT name, is\_auto\_update\_stats\_on

FROM sys.databases

WHERE name = 'YourDatabaseName';

283). What are the limitations of view?

* Cannot pass parameters.
* Cannot include ORDER BY unless using TOP.
* No support for temporary tables.
* Updates on non-updatable views fail.
* Performance issues with complex queries.

284). How to find department with highest number of EmployeeDetails?

SELECT DepartmentID, COUNT(\*) AS EmployeeCount

FROM EmployeeDetails

GROUP BY DepartmentID

ORDER BY EmployeeCount DESC

LIMIT 1;

285). What are different operation available on ONDELETE and ONUPDATE?

* **CASCADE**: Propagates changes.
* **SET NULL**: Sets the foreign key to NULL.
* **SET DEFAULT**: Sets the foreign key to a default value.
* **NO ACTION**: Rejects the change.
* **RESTRICT**: Prevents the action.

286). What are the uses of System tables?

System tables (e.g., sys.objects, sys.tables) store metadata about database objects, such as schema, structure, dependencies, and permissions.

287). What are WAIT Types?

Indicate reasons for query delays, such as:

* **PAGEIOLATCH\_SH**: Waiting for a data page.
* **ASYNC\_NETWORK\_IO**: Waiting for client to process data.
* **LCK\_M\_IX**: Waiting for an incompatible lock.

288). What is Data Page?

A unit of storage in SQL Server (8KB). Contains rows from a table or index.

289). What is FILL Factor?

Specifies the percentage of space to fill in a page during index creation, leaving space for future growth.

290). Sql server difference between view and stored procedure?

* **View**: Predefined query, no procedural logic, read-only.
* **Stored Procedure**: Procedural logic, can accept parameters, perform DML.

291). Sql server difference between unique and nonunique index?

* **Unique Index**: Ensures unique values, improves search performance.
* **Non-Unique Index**: Allows duplicates, aids performance without uniqueness constraint.

292). Sql server difference between update lock and exclusive lock?

* **Update Lock (U)**: Prevents other updates but allows reads.
* **Exclusive Lock (X)**: Blocks all operations.

293). Sql server difference between windows authentication and sql server authentication?

* **Windows Authentication**: Uses domain credentials, more secure.
* **SQL Server Authentication**: Requires username/password, independent of OS.

294). Sql server difference between sysobjects and sys.objects?

* **sysobjects**: Legacy system table for metadata.
* **sys.objects**: Newer catalog view with additional metadata.

295). Sql server difference between session and connection?

* **Session**: Individual user interaction with the server.
* **Connection**: Underlying communication link for the session.

296). Sql server difference between set and select variable?

* **SET**: Assigns single value, preferred for clarity.
* **SELECT**: Assigns multiple values, faster in batch.

297). Sql server difference between shrink database and shrink files?

* **Shrink Database**: Reduces database size, affects all files.
* **Shrink Files**: Reduces individual file size.

298). Sql server difference between revoke and deny?

* **REVOKE**: Removes granted/denied permission.
* **DENY**: Explicitly denies permission, overriding GRANT.

299). Sql server difference between rank and dense\_rank?

* **RANK**: Skips ranks for ties.
* **DENSE\_RANK**: Does not skip ranks.

300). Sql server difference between gdr and qfe?

* **GDR**: General Distribution Release, for critical fixes.
* **QFE**: Quick Fix Engineering, for specific issues.

301). Sql server difference between db\_owner and dbo?

* **db\_owner**: Role with full control of the database.
* **dbo**: Database owner, linked to schema.

302). Sql server difference between log shipping and mirroring?

* **Log Shipping**: Asynchronous, uses transaction log backups.
* **Mirroring**: Synchronous or asynchronous, real-time replication.

303). Sql server difference between isnull and coalesce?

* **ISNULL**: Two arguments, replaces null.
* **COALESCE**: Multiple arguments, returns first non-null.

304). Differences between ISNULL and IS NULL?

* **ISNULL**: Function to replace null.
* **IS NULL**: Operator to check null.

305). Difference between hadoop and sql server?

* **Hadoop**: Distributed framework for large-scale data storage/processing.
  + Handles unstructured data.
  + Scales horizontally.
  + Schema-on-read.
* **SQL Server**: RDBMS for structured data.
  + Optimized for OLTP and OLAP.
  + Schema-on-write.
  + Scales vertically.

306). Difference between PatIndex and CharIndex function in SQL SERVER?

**PATINDEX**: Supports wildcard search, returns position of a pattern.

SELECT PATINDEX('%pattern%', column\_name);

**CHARINDEX**: Searches for exact substring.

SELECT CHARINDEX('substring', column\_name);

307). What is the difference between SQL Server standard and web edition?

* **Standard Edition**: Full RDBMS capabilities, supports enterprise features.
* **Web Edition**: Optimized for web hosting, limited scalability.

308). Difference between numeric,float and decimal in sql server?

* **NUMERIC/DECIMAL**: Fixed precision and scale.
* **FLOAT**: Approximate values, better for large ranges, less precision.

309). Difference between binary and varbinary datatype in Sql server?

**BINARY**: Fixed-length binary data

BINARY(10); -- Always stores 10 bytes

**VARBINARY**: Variable-length binary data.

VARBINARY(10); -- Stores up to 10 bytes

310). Sql server difference between count(\*) and count(1)

* **COUNT(\*)**: Counts all rows, including nulls.
* **COUNT(1)**: Same as COUNT(\*), evaluates a constant.

311). Sql server difference between exec and sp\_executesql?

* **EXEC**: Executes a string as SQL, no parameterization.
* **sp\_executesql**: Supports parameterized queries, better performance.

312). Difference between blocking and deadlock sql server?

* **Blocking**: One query holds a lock, causing another to wait.
* **Deadlock**: Two queries wait for each other's resources indefinitely.

313). Difference between detach and take offline in sql server?

* **Detach**: Removes database from SQL Server instance, keeps files.
* **Take Offline**: Makes database inaccessible, remains attached.

314). Difference between Index Scan / Index Seek in sql server?

* **Index Scan**: Reads entire index, slower for large datasets.
* **Index Seek**: Searches specific rows, uses index efficiently.

315). What is the difference between TRY\_CONVERT and Convert?

* **TRY\_CONVERT**: Returns NULL for invalid conversions.
* **CONVERT**: Throws error for invalid conversions.

316). Write down the query to get the list of tables changed with in the database in last 5

days?

SELECT name, modify\_date

FROM sys.tables

WHERE DATEDIFF(DAY, modify\_date, GETDATE()) <= 5;

317). Write a query to insert your name 1000 times without Using While Loop in SQL

Server?

WITH Numbers AS (

SELECT 1 AS n

UNION ALL

SELECT n + 1 FROM Numbers WHERE n < 1000

)

INSERT INTO YourTable (NameColumn)

SELECT 'YourName'

FROM Numbers;

318). SQL Server 2016 came with new way to drop object if exist, Explain it.(DROP IF EXIST)

DROP TABLE IF EXISTS YourTable;

319). Can you drop the database on which you are using/working currently, in same

session?

USE master;

DROP DATABASE YourDatabase;

320). Write down the T-SQL Script to print then number from 1 to 10?

WITH Numbers AS (

SELECT 1 AS n

UNION ALL

SELECT n + 1 FROM Numbers WHERE n < 10

)

SELECT n FROM Numbers;

321). What is the difference between Wide and Nonwide tables in SQL Server?

* **Wide Tables**: Up to 30,000 columns using sparse columns.
* **Nonwide Tables**: Limited to 1,024 columns, no sparse columns.

322). What do you understand by index\_id = 0 and Index\_id = 1, related

to sys.indexes table? What they represent?

* **index\_id = 0**: Heap (table without clustered index).
* **index\_id = 1**: Clustered index.

323). What are NDF files in SQL Server?

Secondary database files used for extending storage beyond the primary file (MDF).

324). What is the difference between DateTime and DateTimeOffset data types in SQL

Server?

* **DATETIME**: No timezone information.
* **DATETIMEOFFSET**: Includes timezone offset.

325). What is Slot Array in Sql Server? How it is related to Database Page?

Part of a data page storing row offsets for quick lookup. Used in indexing and storage optimization.

326). What is the extension for trace file in SQL Server Profiler?

The file extension is .trc.

327). How to pass an array(or a table variable) into a SQL Server stored procedure?

CREATE TYPE MyTableType AS TABLE (ID INT);

GO

CREATE PROCEDURE MyProcedure (@MyTable MyTableType READONLY)

AS

BEGIN

SELECT \* FROM @MyTable;

END;

GO

328). What is forwarding pointer in SQL Server? How it is helpful in performance

optimization?

 **Forwarding Pointer**:

* Used in **heap tables** when a row is updated and moved to a new page due to lack of space on the original page.
* A pointer is left at the original location to reference the new location.

 **Performance Impact**:

* Reduces overhead of maintaining indexes but can lead to slower reads due to pointer lookup.

329). How you will trace all event only from a selected Database or Procedure using SQL

Server Profiler?

1. Open SQL Server Profiler.
2. Start a new trace, and select "TSQL\_SPs" or "Standard" template.
3. Add a filter under the "Events Selection" tab:
   * Database Name = <Database\_Name>
   * Object Name = <Procedure\_Name>
4. Run the trace.

330). What is the difference between SQL, PL-SQL and T-SQL

 **SQL**:

* Standard query language for managing databases.
* Syntax used across most RDBMS.

 **PL/SQL** (Oracle):

* Procedural language extension for SQL.
* Supports loops, conditions, and error handling.

 **T-SQL** (SQL Server):

* Microsoft's procedural extension for SQL.
* Includes transaction control, error handling, and procedural programming features.