**What Is SQL Profiler?**

SQL Profiler is a tool which allows system administrator to monitor events in the SQL server. This is mainly used to capture and save data about each event of a file or a table for analysis.

**What is indexes in SQL?**

Indexes are used by queries to fiNd data from tables quickly. Indexes are created on tables and views

CREATE Index IX\_tblEmployee\_Salary ON tblEmployee (Salary ASC)

Types of Indexes,

Clustered ,Non Clustered, Uniq

Clustered and Non Clustered Index:-

A table can have only one clustered index, where as you can have more than one non clustered index

Clustered index is faster than non clustered index because the non clustered index has to refer back to the table, if the selected column is not present in the index.

Clustered index determines the storage order of rows in the table, and hence doesn’t require additional disk space, but where as the non clustered index is stored separately from the table, additional storage space is required

Unique Index:-

Unique index is used to enforce uniqueness of key value in the index.

**What is recursive stored procedure?**

SQL Server supports recursive stored procedure which calls by itself. Recursive stored procedure can be defined as a method of problem solving wherein the solution is arrived repetitively. It can nest up to 32 levels.

**What is CHECK constraint?**

A CHECK constraint can be applied to a column in a table to limit the values that can be placed in a column. Check constraint is to enforce integrity.

**What is sub query and its properties?**

A sub-query is a query which can be nested inside a main query like Select, Update, Insert or Delete statements. This can be used when expression is allowed. Properties of sub query can be defined as

* A sub query should not have order by clause
* A sub query should be placed in the right hand side of the comparison operator of the main query
* A sub query should be enclosed in parenthesis because it needs to be executed first before the main query
* More than one sub query can be included

**What are the types of sub query?**

There are three types of sub query –

* Single row sub query which returns only one row
* Multiple row sub query which returns multiple rows
* Multiple column sub query which returns multiple columns to the main query. With that sub query result, Main query will be executed.

**What is SQL server agent?**

The SQL Server agent plays a vital role in day to day tasks of SQL server administrator(DBA). Server agent's purpose is to implement the tasks easily with the scheduler engine which allows our jobs to run at scheduled date and time.

**What is CHECK constraint?**

A CHECK constraint can be applied to a column in a table to limit the values that can be placed in a column

**What are scheduled tasks in SQL Server?**

Scheduled tasks or jobs are used to automate processes that can be run on a scheduled time at a regular interval. This scheduling of tasks helps to reduce human intervention during night time and feed can be done at a particular time. User can also order the tasks in which it has to be generated.

**How exceptions can be handled in SQL Server Programming?**

Exceptions are handled using TRY----CATCH constructs and it is handles by writing scripts inside the TRY block and error handling in the CATCH block.

**What is the purpose of FLOOR function?**

FLOOR function is used to round up a non-integer value to the previous least integer. Example is given

FLOOR(6.7)

Returns 6.

**Can we check locks in database? If so, how can we do this lock check?**

Yes, we can check locks in the database. It can be achieved by using in-built stored procedure called sp\_lock.

**What is the use of SIGN function?**

SIGN function is used to determine whether the number specified is Positive, Negative and Zero. This will return +1,-1 or 0.

Example – SIGN(-35) returns -1

**What is a Trigger?**

Triggers are used to execute a batch of SQL code when insert or update or delete commands are executed against a table. Triggers are automatically triggered or executed when the data is modified. It can be executed automatically on insert, delete and update operations.

**What are the types of Triggers?**

* DDL (Data Definition Language) - A DDL trigger executes in response to a change to the structure of a database (for example, CREATE, ALTER, DROP)
* DML (Data Manipulation Language) - A DML trigger executes in response to a change in data (INSERT, UPDATE, DELETE)

**What is an IDENTITY column in insert statements?**

IDENTITY column is used in table columns to make that column as Auto incremental number or a surrogate key.

**What is the difference between Primary Key and Identity column in SQL?**

A PK is a unique identifier for a row of data. An identity column will **auto fill based on some increment each new row**. Identities are not necessarily unique or contiguous. You can make the identity column your PK

An identity is simply an auto-increasing column.

A primary key is the unique column or columns that define the row.

These two are often used together, but there's no requirement that this be so.

**What is Bulkcopy in SQL?**

Bulkcopy is a tool used to copy large amount of data from Tables. This tool is used to load large amount of data in SQL Server.

**What will be query used to get the list of triggers in a database?**

Query to get the list of triggers in database-

Select \* from sys.objects where type='tr'

**What is the difference between UNION and UNION ALL?**

* UNION: To select related information from two tables UNION command is used. It is similar to JOIN command.
* UNION All: The UNION ALL command is equal to the UNION command, except that UNION ALL selects all values. It will not remove duplicate rows, instead it will retrieve all rows from all tables.

**What are the differences between Stored Procedure and the dynamic SQL?**

Stored Procedure is a set of statements which is stored in a compiled form. Dynamic SQL is a set of statements that dynamically constructed at runtime and it will not be stored in a Database and it simply execute during run time.

**Joins in SQL SERVER**

* (INNER) JOIN: Returns records that have matching values in both tables
* LEFT (OUTER) JOIN: Returns all records from the left table, and the matched records from the right table
* RIGHT (OUTER) JOIN: Returns all records from the right table, and the matched records from the left table
* FULL (OUTER) JOIN: Returns all records when there is a match in either left or right table

      

**Eg.**

SELECT Orders.OrderID, Customers.CustomerName, Orders.OrderDate  
FROM Orders  
INNER JOIN Customers ON Orders.CustomerID=Customers.CustomerID;

**Write a SQL query to fetch Employee Name and Manager Name from below table**

|  |  |  |
| --- | --- | --- |
| EmployeeID | Name | ManagerID |
| 1 | Mike | 3 |
| 2 | Rob | 1 |
| 3 | Todd | NULL |
| 4 | Ben | 1 |
| 5 | Sam | 1 |

Select E.Name AS EmployeeName, M.Name AS ManagerName

FROM tblEmployee E

LEFT JOIN tblEmployee M

On E.ManagerID = M.EmployeeID

|  |  |
| --- | --- |
| EmployeeName | ManagerName |
| Mike | Todd |
| Rob | Mike |
| Todd | NULL |
| Ben | Mike |
| Sam | Mike |

**Write a SQL query to find second highest salary of an employee**

1. **Select** **Max**(Salary) **as** Salary **from** tbl\_Employees

**where** Salary < (**select** **MAX**(Salary) **from** tbl\_Employees)

1. SELECT TOP 1 SALARY

FROM ( SELECT DISTINCT TOP 2 SALARY

FROM tbl\_Employees

ORDER BY SALARY DESC

) RESULT

ORDER BY SALARY

**Write a SQL query to find third highest salary of an employee**

SELECT TOP 1 SALARY

FROM (

SELECT DISTINCT TOP 3 SALARY

FROM tbl\_Employees

ORDER BY SALARY DESC

) RESULT

ORDER BY SALARY

**What is GROUP BY query in sql**

<https://www.geeksforgeeks.org/sql-group-by/>

**Write SQL update Query**

UPDATE Customers  
SET ContactName = 'Alfred Schmidt', City= 'Frankfurt'  
WHERE CustomerID = 1;

**Write SQL Insert query**

INSERT INTO Customers (CustomerName, City, Country)  
VALUES ('Cardinal', 'Stavanger', 'Norway');

**What is IsNull Function?**

Select E.Name as Employee ISNULL(M.Name, ‘No Manager’) as Manager

FROM tblEmployee E

Left Join tblEmlpoyee M

ON E.ManagerID = M.EmplyeeID

Select E.Name as Employee COALESCE(M.Name, ‘No Manager’) as Manager

FROM tblEmployee E

Left Join tblEmlpoyee M

ON E.ManagerID = M.EmplyeeID

**What is indexing in SQL?**

Indexes are used to retrieve data from the database more quickly.

**How to choose column while creating index?**

Columns that appear in where clause or JOIN Conditions

**What are the types of indexes?**

1. Clustered Index

CREATE INDEX index\_name  
ON table\_name (column1, column2, ...);

1. Non Clustered Index

(We can create indexes using object explorer also)

**Can we have two primary keys for one table?**

No

**How to reset primary key to zero again after deleting all the records?**

ALTER TABLE tablename AUTO\_INCREMENT = 1

**What is the difference between Delete, Truncate and Drop commands?**

**Delete :-** Basically, it is a [Data Manipulation Language Command (DML)](https://www.geeksforgeeks.org/sql-ddl-dql-dml-dcl-tcl-commands/). It is used to delete one or more tuples of a table. With the help of the “DELETE” command, we can either delete all the rows in one go or can delete rows one by one. i.e., we can use it as per the requirement or the condition using the Where clause. It is comparatively slower than the TRUNCATE command. The TRUNCATE command does not remove the structure of the table.

**DROP** :- It is a Data Definition Language Command (DDL). It is used to drop the whole table. With the help of the “DROP” command we can drop (delete) the whole structure in one go i.e. it removes the named elements of the schema. By using this command the existence of the whole table is finished or say lost.

**TRUNCATE** :- It is also a Data Definition Language Command (DDL). It is used to delete all the rows of a relation (table) in one go. With the help of the “TRUNCATE” command, we can’t delete the single row as here WHERE clause is not used. By using this command the existence of all the rows of the table is lost. It is comparatively faster than the delete command as it deletes all the rows fastly.

**What is View in SQL?**

In SQL, a view is a virtual table based on the result-set of an SQL statement.

CREATE VIEW [Brazil Customers] AS  
SELECT CustomerName, ContactName  
FROM Customers  
WHERE Country = 'Brazil';

**What is Table variable in SQL?**

Table variable is a type of local variable that used to store data temporarily, and we can perform all table operations on this table

1. **Declare** @TempTable **TABLE**(
2. id **int**,
3. **Name** **varchar**(20)
4. )

**What is difference between view and table variable in sql?**

View and Table both are integral parts of a relational database, and both terms are used interchangeably. The view is a result of an SQL query and it is a virtual table, whereas a Table is formed up of rows and columns that store the information of any object and be used to retrieve that data whenever required.

**How to Handle Global Exceptions in SQL Server?**

There are following two ways to handle exceptions in SQL:-

1. Using Try and Catch
2. Using @@ERROR global variable
3. **select** 1/0;
4. **Declare** @Err\_num **int**; --Declare variable
5. **set** @Err\_Num=@@ERROR; --assigning error number to variable @Err\_Num
6. **select** @Err\_num **as** Error\_Number

**What is difference between Unique key and Primary key?**

Primary key and Unique key are the unique identities of the row just the difference is unique key can contain null values where Primary key cannot contain null values.

**What are the constraint in SQL?**

There are following constraints in SQL:

Primary Key

Foreign Key

Unique Key

Check Constraint

Not Null Constraint

**What are the types of functions in SQL?**

* Scalar Functions (Returns A Single Value)
* Inline Table Valued Functions (Contains a single TSQL statement and returns a Table Set)
* Multi-Statement Table Valued Functions (Contains multiple TSQL statements and returns Table Set)

**What is difference between Inline table valued function and Multi-Statement table value function?**

**Inline table valued function:**

A table-valued function is a [user-defined function](https://www.sqlservertutorial.net/sql-server-user-defined-functions/) that returns data of a table type. The return type of a table-valued function is a table, therefore, you can use the table-valued function just like you would use a table.

CREATE FUNCTION udfProductInYear (

@model\_year INT

)

RETURNS TABLE

AS

RETURN

SELECT

product\_name,

model\_year,

list\_price

FROM

production.products

WHERE

model\_year = @model\_year;

**Multi Statement table valued function**

A multi-statement table-valued function or MSTVF is a table-valued function that returns the result of multiple statements.

The multi-statement-table-valued function is very useful because you can execute multiple queries within the function and aggregate results into the returned table.

CREATE FUNCTION udfContacts()

RETURNS @contacts TABLE (

first\_name VARCHAR(50),

last\_name VARCHAR(50),

email VARCHAR(255),

phone VARCHAR(25),

contact\_type VARCHAR(20)

)

AS

BEGIN

INSERT INTO @contacts

SELECT

first\_name,

last\_name,

email,

phone,

'Staff'

FROM

sales.staffs;

INSERT INTO @contacts

SELECT

first\_name,

last\_name,

email,

phone,

'Customer'

FROM

sales.customers;

RETURN;

END;

**What is difference between function and stored procedure?**

The function must return a value but in Stored Procedure it is optional. Even a procedure can return zero or n values.

Functions can have only input parameters for it whereas Procedures can have input or output parameters

Functions can be called from Procedure whereas Procedures cannot be called from a Function.

The procedure allows SELECT as well as DML(INSERT/UPDATE/DELETE) statement in it whereas Function allows only SELECT statement in it.

An exception can be handled by try-catch block in a Procedure whereas try-catch block cannot be used in a Function.

**Write a query to delete duplicate records from table**

Delete E1

From EmplyeeE1, EmployeeE2

Where E1.email = E2.Email

AND E1.name = E2.name

AND E1.ID > E2.ID

**Write a SQL query to find duplicate rows in a table**

Select \*, COUNT(empID)

From Employee

GROUP BY empID

HAVING COUNT (empID) >1;

**How to Optimise SQL Query?**

**SELECT fields instead of using SELECT \***

**Avoid SELECT DISTINCT**

SELECT DISTINCT is a handy way to remove duplicates from a query. SELECT DISTINCT works by GROUPing all fields in the query to create distinct results. To accomplish this goal however, a large amount of processing power is required. Additionally, data may be grouped [to the point of being inaccurate](https://www.sisense.com/blog/understanding-simpsons-paradox-to-avoid-faulty-conclusions/). To avoid using SELECT DISTINCT, select more fields to create unique results.

**Create joins with INNER JOIN (not WHERE)**

Some SQL developers prefer to make joins with WHERE clauses, such as the following:

SELECT Customers.CustomerID, Customers.Name, Sales.LastSaleDate  
FROM Customers, Sales  
WHERE Customers.CustomerID = Sales.CustomerID

This type of join creates a Cartesian Join, also called a Cartesian Product or CROSS JOIN.

**Use WHERE instead of HAVING to define filters**

**Use wildcards at the end of a phrase only**

Consider this query to pull cities beginning with ‘Char’:

SELECT City FROM Customers  
WHERE City LIKE ‘%Char%’

This query will pull the expected results of **Char**leston, **Char**lotte, and **Char**lton. However, it will also pull unexpected results, such as Cape **Char**les, Crab Or**char**d, and **Rich**ardson.

A more efficient query would be:

SELECT City FROM Customers  
WHERE City LIKE ‘Char%’

This query will pull only the expected results of **Char**leston, **Char**lotte, and **Char**lton.

**Use LIMIT to sample query results**

Before running a query for the first time, ensure the results will be desirable and meaningful by using a **LIMIT** statement. (In some DBMS systems, the word TOP is used interchangeably with LIMIT.) The LIMIT statement returns only the number of records specified. Using a **LIMIT** statement prevents taxing the production database with a large query, only to find out the query needs editing or refinement.

In the 2016 sales query from above, we will examine a limit of 10 records:

SELECT Customers.CustomerID, Customers.Name, Count(Sales.SalesID)  
FROM Customers  
  INNER JOIN Sales  
  ON Customers.CustomerID = Sales.CustomerID  
WHERE Sales.LastSaleDate BETWEEN #1/1/2016# AND #12/31/2016#  
GROUP BY Customers.CustomerID, Customers.Name  
LIMIT 10

We can see by the sample whether we have a useable data set or not.

**What is SQL Injection?**

SQL injection is a code injection technique that might destroy your database.

SQL injection is one of the most common web hacking technique.

SQL injection is the placement of malicious code in SQL statements, via web page input.