#This is Assignment 4 because of you attach assignment 4 Question in place of assignment 1

Assignment 4

1. **Explain different types of views. Demonstrate with suitable examples.**

Simple view

CREATE VIEW EmployeeView AS

SELECT EmployeeID, FirstName, LastName

FROM Employees

WHERE Department = 'IT';

Complex view

CREATE VIEW SalesSummary AS

SELECT ProductID, SUM(Quantity) AS TotalQuantity, AVG(Price) AS AvgPrice

FROM Sales

GROUP BY ProductID;

Index view

CREATE VIEW ProductSalesView WITH SCHEMABINDING AS

SELECT ProductID, SUM(Quantity) AS TotalQuantity, AVG(Price) AS AvgPrice

FROM dbo.Sales

GROUP BY ProductID;

CREATE UNIQUE CLUSTERED INDEX IX\_ProductSalesView\_ProductID

ON ProductSalesView (ProductID);

Inline view

SELECT \*

FROM (

SELECT EmployeeID, FirstName, LastName

FROM Employees

WHERE Department = 'HR'

) AS HR\_Employees;

1. **What is the difference between function and stored procedure? Write syntax for creating functions and stored procedures.**

Both functions and stored procedures are database objects in SQL that allow you to encapsulate a set of SQL statements into a single, reusable unit. However, there are key differences between them in terms of their purpose, return values, and usage.

For function

CREATE FUNCTION function\_name

( [ { IN | OUT | INOUT } [parameter\_name] data\_type [, ...] ] )

RETURNS return\_data\_type

AS

-- SQL statements

For procedure

CREATE PROCEDURE procedure\_name

[ { IN | OUT | INOUT } [parameter\_name] data\_type [, ...] ]

AS

-- SQL statements

1. **What is an index in SQL? What are the different types of indexes in SQL?**

In SQL, an index is a database object that provides a quick and efficient way to look up records in a table based on the values in one or more columns. Indexes are used to improve the speed of data retrieval operations on a database table by reducing the number of data pages that need to be read from disk.

Types - clustered, non cluster, composite, unique, full text & covering index

1. **Showcase an example of exception handling in SQL stored procedure.**

CREATE PROCEDURE DivideNumbers

@Dividend INT,

@Divisor INT

AS

BEGIN

BEGIN TRY

DECLARE @Result FLOAT;

-- Check if the divisor is zero

IF @Divisor = 0

BEGIN

THROW 50000, 'Division by zero is not allowed.', 1;

END

-- Perform the division

SET @Result = @Dividend / @Divisor;

-- Display the result

SELECT 'Result: ' + CAST(@Result AS NVARCHAR(50)) AS ResultMessage;

END TRY

BEGIN CATCH

-- Handle the exception

SELECT

ERROR\_NUMBER() AS ErrorNumber,

ERROR\_MESSAGE() AS ErrorMessage,

ERROR\_SEVERITY() AS ErrorSeverity,

ERROR\_STATE() AS ErrorState;

END CATCH

END;

-- Execute the stored procedure with a valid divisor

EXEC DivideNumbers @Dividend = 10, @Divisor = 2;

-- Execute the stored procedure with a divisor of zero

EXEC DivideNumbers @Dividend = 10, @Divisor = 0;

1. **Create a SQL function to split strings into rows on a given character?**

**Input String: Stephen;peter;berry;Olivier;caroline;**

| **Stephen** |
| --- |
| **Peter** |
| **Berry** |
| **Oliver** |
| **Caroline** |

CREATE FUNCTION dbo.SplitStringIntoRows

(

@InputString NVARCHAR(MAX),

@Delimiter NVARCHAR(1)

)

RETURNS TABLE

AS

RETURN

(

SELECT

Value AS SplitValue

FROM STRING\_SPLIT(@InputString, @Delimiter)

WHERE RTRIM(LTRIM(Value)) <> '' -- Exclude empty strings

);

1. **What is a temporary and a variable table? Write suitable syntax to create temporary tables and variable tables.**

In SQL, both temporary tables and table variables are used to store temporary data. However, there are some key differences between them, mainly related to scope, behaviour, and usage.