**Scalar valued functions**

1. **Write a function to print "Hello World".**

CREATE FUNCTION fn\_PrintHello()

RETURNS VARCHAR(50)

AS

BEGIN

DECLARE @Str AS VARCHAR(50)

SET @Str='Hello World'

RETURN @Str

END

1. **Write a function which returns addition of two numbers.**

CREATE FUNCTION fn\_Addition(@No1 AS INT,@No2 AS INT)

RETURNS INT

AS

BEGIN

RETURN(@No1+@No2)

END

1. **Write a function to print cube of given number.**

CREATE FUNCTION fn\_Cube(@No AS INT)

RETURNS INT

AS

BEGIN

RETURN(@No\*@No\*@No)

END

1. **Write a function to check where given number is ODD or EVEN.**

CREATE FUNCTION fn\_CheckEvenOdd(@No AS INT)

RETURNS VARCHAR(50)

AS

BEGIN

DECLARE @Str AS VARCHAR(50)

IF (@No%2=0)

SET @Str='NO IS EVEN'

ELSE

SET @Str='NO IS ODD'

RETURN @Str

END

1. **Write a function to compare two integers and returns the comparison result. (Using Case statement)**

CREATE FUNCTION fn\_Compare(@a AS INT,@b AS INT)

RETURNS VARCHAR(50)

AS

BEGIN

DECLARE @Str AS VARCHAR(50)

SET @Str=

CASE

WHEN @a>@b THEN 'a is greater then b'

WHEN @a<@b THEN 'a is less then b'

ELSE 'a is equal to b'

END

RETURN @Str

END

1. **Write a function to print number from 1 to N. (Using while loop)**

CREATE FUNCTION fn\_Print1toN(@No AS INT)

RETURNS VARCHAR(MAX)

AS

BEGIN

DECLARE @Str AS VARCHAR(MAX)

SET @Str=' '

DECLARE @i AS INT

SET @i=1

WHILE @i<=@No

BEGIN

SET @Str=@Str+CAST(@i AS VARCHAR)+' '

SET @i=@i+1

END

RETURN @Str

END

1. **Write a function to print sum of even numbers between 1 to 20.**

CREATE FUNCTION fn\_SumOf1to20()

RETURNS INT

AS

BEGIN

DECLARE @i AS INT SET @i=1

DECLARE @Sum AS INT SET @Sum=0

WHILE (@i<=20)

BEGIN

IF (@i%2=0)

SET @Sum=@Sum+@i

SET @i=@i+1

END

RETURN @Sum

END

1. **Write a function to check weather given number is prime or not.**

CREATE FUNCTION fn\_IsPrime(@No AS INT)

RETURNS VARCHAR(50)

AS

BEGIN

DECLARE @flag AS BIT

SET @flag=1

DECLARE @i AS INT

SET @i=2

DECLARE @Str as VARCHAR(50)

WHILE (@i<@No)

BEGIN

IF (@No % @i = 0)

BEGIN

SET @flag = 0

BREAK

END

SET @i = @i + 1

END

IF (@flag=0)

SET @Str='No is not Prime'

ELSE

SET @Str='No is Prime'

RETURN @Str

END

1. **Write a function which accepts two parameters start date & end date, and returns a difference in days.**

CREATE FUNCTION fn\_DayDiff(@StartDate AS DATE,@EndDate AS DATE)

RETURNS INT

AS

BEGIN

DECLARE @Day AS INT

SET @Day=DATEDIFF(DAY,@StartDate,@EndDate)

RETURN @Day

END

1. **Write a function which accepts year & month in integer and returns total days in given month & year.**

CREATE FUNCTION fn\_NoOfDaysInMonthYear(@Year AS INT,@Month AS INT)

RETURNS INT

AS

BEGIN

DECLARE @Convert\_To\_FirstDay AS DATE

DECLARE @LastDay\_Of\_Month AS DATE

DECLARE @Day\_Diff AS INT

SET @Convert\_To\_FirstDay=DATEFROMPARTS(@Year,@Month,1)

SET @LastDay\_Of\_Month=EOMONTH(@Convert\_To\_FirstDay)

SET @Day\_Diff=DATEDIFF(DAY,@Convert\_To\_FirstDay,@LastDay\_Of\_Month)+1

RETURN @Day\_Diff

END

**Table valued functions (Use tables of lab-2)**

1. **Write a function which returns a table with detail of person whose first name starts with B.**

CREATE FUNCTION fn\_FirstNameWithB()

RETURNS TABLE

AS

RETURN(SELECT \* FROM Person WHERE FirstName LIKE 'B%')

1. **Write a function which returns a table with unique first names from person table.**

CREATE FUNCTION fn\_UniqueName()

RETURNS TABLE

AS

RETURN(SELECT DISTINCT FirstName FROM Person)

1. **Write a function which accepts department ID as a parameter & returns a detail of the persons.**

CREATE FUNCTION fn\_GetPersonsByDepartmentID (@DepartmentId INT)

RETURNS @personsTable TABLE (

FirstName VARCHAR(100),

LastName VARCHAR(100),

Salary Decimal(8,2),

DepartmentID INT)

AS

BEGIN

INSERT INTO @personsTable (FirstName, LastName, Salary,DepartmentID)

SELECT FirstName, LastName, Salary, DepartmentID

FROM Person

WHERE DepartmentID = @departmentId

RETURN

END