Primary Key

- Scalar UDFs return a single value, which can be used in SELECT statements, WHERE clauses, and more.

Syntax for creating scalar-valued user define function:

```
CREATE FUNCTION dbo.FunctionName
(
   @Parameter1 DataType,
   @Parameter2 DataType
)
RETURNS DataType
AS
BEGIN
   -- Function logic here
   RETURN ResultValue;
END;
```

For modifying the existing function

```
ALTER FUNCTION dbo.FunctionName
(
   @NewParameter DataType
)
RETURNS DataType
AS
BEGIN
   -- Updated function logic here
   RETURN ResultValue;
END;
```

Syntax for executing the scalar user define function:

SQL

User define function (scalar valued)

```
Select dbo.functioname(@parameters)
```

Table valued user define function *(TVF)*

- Scalar UDFs return a single value, which can be used in SELECT statements, WHERE clauses, and more.

General syntax for creating the TVF

```
CREATE FUNCTION dbo.function_name
(
   -- Input parameters (if any)
   @param1 datatype,
   @param2 datatype
)
RETURNS TABLE
AS
RETURN
(
   -- SELECT statement to define the result set
   SELECT column1, column2, ...
   FROM your_tables_or_queries
   WHERE conditions (if any)
);
```

*Syntax for executing the function:- select * from dbo.function_name(@parameters)*

- Create a function to give the details of the employees with the help of employee id

```
create function fn_getbyemployeeid(@id int)
returns table
as
return(select * from employees where employee_id=@id)
```

employee_id	employee_name	department_id	salary
1	John Doe	1	60000.00
2	Jane Smith	2	55000.00
3	Bob Johnson	1	70000.00

```
select * from dbo.fn_getbyemployeeid(1)
```

employee_id	employee_name	department_id	salary
1	John Doe	1	60000.00

Stored procedure

- A stored procedure is a precompiled and reusable set of SQL statements and procedural logic stored in a database, which can be invoked with parameters and executed on demand, providing a way to encapsulate and manage complex database operations

Syntax

```
CREATE PROCEDURE procedure_name
   @parameter1 datatype1,
   @parameter2 datatype2,
   ...
AS
BEGIN
   -- SQL statements
END;
```

To execute procedure:- exec procedure_name @parameters

Procedure to give the details of employees with the help of department name

```
CREATE PROCEDURE GetEmployeeByDepartment
   @dept_id INT
AS
BEGIN
   SELECT * FROM employees
WHERE department_id = @dept_id;
END;
```

employee_id	employee_name	department_id	salary
1	John Doe	1	60000.00
3	Bob Johnson	1	70000.00

Executing the procedure

```
EXEC GetEmployeeByDepartment @dept_id = 1;
```

Deleting the procedure

```
drop procedure GetEmployeeByDepartment
```

IntelliPaat

SQL

UDF & SP

Cheat Sheet