**DBMS LAB**

**Lab Assignment number 09**

**Name:** Aamir Ansari **Batch:** A **Roll no.** 01

**Aim:-** Experiment to study Stored Procedures and functions.

**Theory:-**

**PL/SQL:- ( Procedural SQL)**

PL/SQL is extension of SQL.

SQL with Programming Capability.

**Variable:- Is an object that can hold data value.**

**Local variable:-**

Local variable name should begin with @ sign in SQL Server.

Local variables are used for passing values to SQL statement.

Local variables are used to store value temporary

eg. @empid

**System Global variable:-**

1. Global variable names begins with @@ sign.

2. eg. @@version

**Declaring variable:-**

General syntax:

Declare vname as datatype

Or

Declare vname datatype

Eg. Declare @eid as integer

Every variable should be defined for its datatype separately.

Begin……..end: -

Block of code is identified by using begin….end keyword.

Scope of the variable is identified by using begin and end keyword.

Eg. Begin

Declare @a as float

End

Assigning Value to the variable:-

Set or select statement is used to assign value to the variable.

Set keyword:-

Set @vname=10

Select @vname=columnname

Eg.

Begin

Declare @s as integer,@name as varchar(20)

Set @s=101

Select @name=ename from emp where ssn=@s

End

select @@version --Global Variable

--output

Microsoft SQL Server 2000

**Control structures:-**

**-- If Control structure**

begin

declare @a int,@b int,@c int General Syntax:-

set @a=10 if (condition)

set @b=20 set @c=15 begin

if(@a>@b)and(@a>@c) stmts

print @a end

else if(@b>@c) else

print @b begin

else print @c stmts

end end

--output 20

**-- While loop**

begin

declare @i int **General syntax:-**

set @i=1 while(cond)

while(@i<10) begin

begin stmts

print 'i='+cast(@i as varchar(20)) end

--print @i

set @i=@i+1

end

end

--output

i=1

i=2

i=3

i=4

i=5

i=6

i=7

i=8

i=9

**-- Case Control Structures**

begin

declare @t as varchar(20)

declare @s varchar(20) **General Syntax:-**

set @t=’o’ case

set @s=case when condn then exp

when @t='o' then 'one' when condn then exp

when @t=’t’ then ’two’ else exp

else ‘greater than two’ end

end

end

**Stored Procedures:-**

Stored procedures are precompiled database queries that improves the security, efficiency and usability of code.

Stored procedures are extremely similar to the constructs seen in other programming languages.

They accept data in the form of input parameters that are specified at execution time. These input parameters (if implemented) are utilized in the execution of a series of statements that produce some result.

This result is returned to the calling environment through the use of a [recordset](http://databases.about.com/library/glossary/bldef-record.htm), output parameters and a return code.

Stored procedures can have upto 1024 parameter.

General syntax:-

Create procedure procname (@v as datatype in|out)

As

Begin

Stmts

End

Calling Procedure:-

Execute procname value

create procedure getmonth

as

begin

select month(getdate())

end

execute getmonth

**--Procedure without output parameter**

-- create procedure which displays salary of emp when we pass ssn as parameter to the procedure

create procedure listsal(@s int)

as

begin

declare @sal int

select @sal=emp\_sal from empl1 where emp\_id=@s

print @sal

end

--**Executing Procedure without parameter**

execute listsal 1

**--Procedure with output parameter**

create procedure listsal1(@e int,@sal int output)

as

begin

select @sal=emp\_sal from empl1 where emp\_id=@e

end

--Execution

begin

declare @s int

execute listsal1 2,@s output

print @s

end

--Nesting of procedure

create procedure grosssal(@e int,@d int,@hra int)

as

begin

declare @g int,@s int

execute listsal1 @e, @s output

set @g=@s+(@s\*(@d/100))+(@s\*(@hra/100))

print 'gross sal'+ cast(@g as varchar(20))

end

execute grosssal 1,50,20

**Advantages of Stored procedure:-**

Precompiled execution. SQL Server compiles each stored procedure once and then reutilizes the execution plan. Execution speed increases.

Reduced client/server traffic.

Efficient reuse of code and programming abstraction. Stored procedures can be used by multiple users and client programs.

Enhanced security controls. You can grant users permission to execute a stored procedure independently of underlying [table](http://databases.about.com/library/glossary/bldef-table.htm) permissions.

**Function:-**

Function is precompiled set of statements which returns value explicitly to the caller of function.

Create funcname fname (@v as datatype) returns datatype

As

Begin

Stmts

Return value

End

**Execution :-**

Print username.functionname(value)

create function avgsal1(@d int)

returns int as

begin

declare @avgsal int

select @avgsal=avg(emp\_sal) from empl1 where do=@d

return @avgsal

end

**--Execution**

print jayshree.avgsal1(1)

select \*from empl1 where emp\_sal>jayshree.avgsal1(1)

**Procedures**

1. Printing numbers from 1 to 10

begin

declare @a as int;

set @a=1;

while(@a<10)

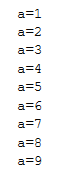
begin

print 'a='+cast(@a as varchar(20))

set @a=@a+1;

end

end



1. Print the greatest of 3 numbers

begin

declare @a as int,@b as int,@c as int;

set @a=15;

Set @b=20

set @c=10;

print 'The largest number is'

if(@a>@b)and(@a>@c)

print @a

else if(@b>@c)

print @b

else

print @c

end



1. Printing the table of 5

begin

declare @i as int;

set @i=1

declare @m as int;

set @m=5

while(@i<=10)

begin

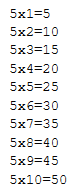
print cast(@m as varchar(20))+'x'+cast(@i as varchar(20))+'='+cast(@m\*@i as

varchar(20))

set @i=@i+1;

end

end



**Stored Procedures:**

1. Without Parameters: Create a procedure which displays details of all employees

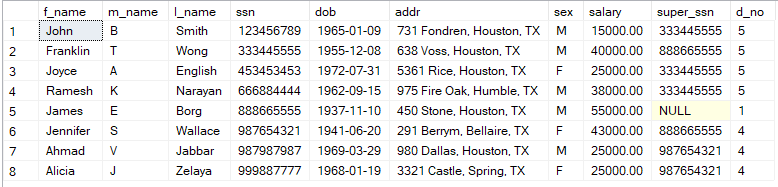
CREATE PROCEDURE EmployeeSelect

AS

SELECT \* FROM Employee

GO;

EXEC EmployeeSelect;



1. With Parameters

a. A procedure which displays details of Employee with name 'James'

CREATE PROCEDURE EmpSelect @Fname varchar(30)

AS

SELECT \* FROM Employee WHERE Fname=@Fname

GO

EXEC EmpSelect @Fname='James';



b. A procedure which displays details of Employee with name 'Ramesh' and

salary=38000

CREATE PROCEDURE selectCondition @Fname varchar(30), @Salary money

AS

SELECT \* FROM Employee WHERE Fname=@Fname AND Salary=@Salary

GO

EXEC selectCondition @Fname='Ramesh',@Salary=38000;



**Conclusion:** Hence we have successfully studied and implemented stored Procedures in DBMS