

East West University

Department of Computer Science & Engineering

**Course Title:** Database System

**Course Code:** CSE301  
**Experiment No:** 09

**Semester:** Fall, 2015

**Section:** 01

**Submitted By:**

S. M. Amirul Islam

ID: 2013-1-60-048

**Introduction:**

In SQL, server can be defined as the set of logically group of sql statement which is grouped to perform a specific task. The main benefit of using a Stored Procedure is that it increases the performance of the database by reducing the amount of information sent to the database server. The other benefits of using the Stored Procedure is that compilation step is required only once when it is created, it helps in re-usability of the sql code and is helpful in enhancing the security. In this experiment, we will try to learn about Stored Procedure in database.

**Lab Examples:**

Query of all Tables with Outputs:

**Customer:**

create table Customer

(

customer\_id int ,

name varchar(20),

address varchar(30),

primary key(customer\_id)

)

create procedure Customer\_get (@a int,@b varchar(50),@c varchar(50))

as

begin

insert into Customer values (@a,@b,@c)

end

exec Customer\_get 105,'Asif','Dhaka'

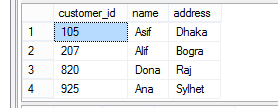
exec Customer\_get 925,'Ana','Sylhet'

exec Customer\_get 820,'Dona','Raj'

exec Customer\_get 207,'Alif','Bogra'

select\*from Customer

**Output:**



**Account:**

create table Account

(

acc\_no int not null,

balance int,

branch\_name varchar(50),

primary key(acc\_no)

)

create procedure Account\_get (@a int,@b varchar(50),@c varchar(50))

as

begin

insert into Account values (@a,@b,@c)

end

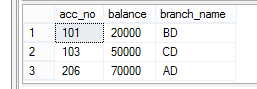
exec Account\_get 101,20000,'BD'

exec Account\_get 103,50000,'CD'

exec Account\_get 206,70000,'AD'

select \* from Account

**Output:**



**Loan:**

create table Loan

(

loan\_no int not null,

amount int not null,

branch\_name varchar(50),

primary key(loan\_no)

)

create procedure loan\_get (@loan\_no int,@amount int,@branch\_name varchar(50))

as

begin

insert into loan values (@loan\_no,@amount,@branch\_name)

end

exec loan\_get 245,10000,'AB'

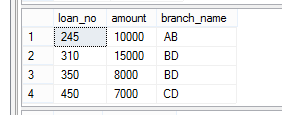
exec loan\_get 310,15000,'BD'

exec loan\_get 350,8000,'BD'

exec loan\_get 450,7000,'CD'

select\*from Loan

**Output:**

****

**Q1. Write a procedure to get the amount by giving loan no.**

Ans:

create procedure get\_amount (@loan1\_no int)

as

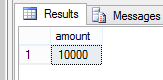
begin

select amount from Loan where loan\_no =@loan1\_no

end

exec get\_amount 245

**Output:**



**Q2. Write a procedure to get the balance of a branch.**

Ans:

create procedure get\_balance (@branch1\_name varchar (20))

as

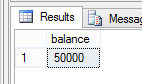
begin

select balance from Account where branch\_name=@branch1\_name

end

exec get\_balance 'CD'

**Output:**



**Q3. Write a procedure to get customer name from a customer id.**

Ans:

create procedure get\_name (@customer\_id1 int)

as

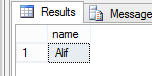
begin

select name from Customer where customer\_id=@customer\_id1

end

exec get\_name 207

**Output:**



**Q4. Write a procedure to get customer id from a customer name.**

Ans:

create procedure get\_customer\_id (@name1 varchar(20))

as

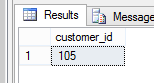
begin

select customer\_id from Customer where name=@name1

end

exec get\_customer\_id 'Asif'

**Output:**



**Q5. Write a procedure to get customer name/id from a customer address.**

Ans:

create procedure get\_customer\_id1 (@address1 varchar(20))

as

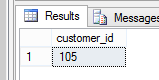
begin

select customer\_id from Customer where address=@address1

end

exec get\_customer\_id1 'Dhaka'

**Output:**



**Q6. Write a procedure to insert value in customer table.**

Ans:

create procedure Customer\_add (@a int,@b varchar(50),@c varchar(50))

as

begin

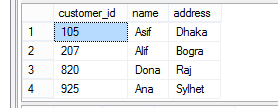
insert into Customer values (@a,@b,@c)

end

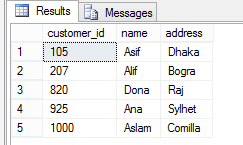
exec Customer\_add 1000,'Aslam','Comilla'

**Output:**

**Before Insertion:**

****

**After Insertion:**



**Q7. Write a procedure to update the amount by twice for given loan no.**

Ans:

create procedure uxyz (@xloan\_no int)

as

begin

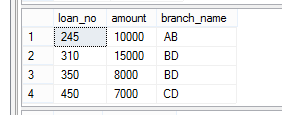
update loan set amount=amount\*2 where loan\_no=@xloan\_no

end

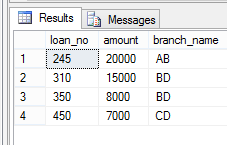
exec uxyz 245

**Output:**

**Before Update:**



After Update:



**Lab Report Problems:**

**Q1. Write a procedure to insert the values in the dept table and instructor table.**

Ans:

**Department:**

create table dept(

deptid int not null,

deptname varchar(20),

primary key(deptid)

)

create procedure dept\_add (@deptid int,@deptname varchar(20))

as

begin

insert into dept values (@deptid,@deptname)

end

exec dept\_add 60,'CSE'

exec dept\_add 50,'ECE'

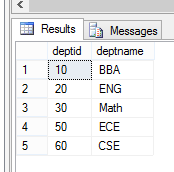
exec dept\_add 30,'Math'

exec dept\_add 20,'ENG'

exec dept\_add 10,'BBA'

select\*from dept

Output:

****

**Instructor:**

create table instructor(

insid int not null,

insname varchar(20),

deptid int not null,

salary int,

primary key(insid)

)

create procedure ins\_add (@insid int,@insname varchar(20),@salary int,@deptid int)

as

begin

insert into instructor values (@insid,@insname,@salary,@deptid)

end

exec ins\_add 101,'asif',12000,60

exec ins\_add 123,'ana',15000,60

exec ins\_add 125,'asif',16000,30

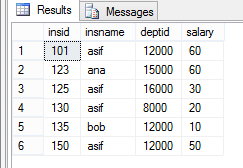
exec ins\_add 130,'asif',8000,20

exec ins\_add 135,'bob',12000,10

exec ins\_add 150,'asif',12000,50

select\*from instructor

Output:

****

**Q2. Write a function to take the id of two employees and compare their salaries and return the id of the employee whose salary is highest among them.**

Ans:

create function highestSalary (@ins1 int,@ins2 int)

returns int

as

begin

declare @id int;

declare @salary1 int;

declare @salary2 int;

select @salary1=salary from instructor where insid=@ins1

select @salary2=salary from instructor where insid=@ins2

if @salary1>@salary2

set @id=@ins1

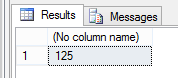
else

set @id=@ins2

return @id

end

Output:



**Q3. If there is more than 3 employees named asif then print that there are more than three asif in the university. Else print that there are three or less asif in the university**

Ans:

if(select count (customer\_id) from Customer where name like 'Asif')>3

begin

print 'there are more than three asif in our university.'

end

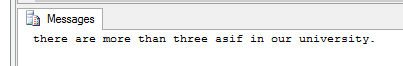
else

begin

print 'there are three or less asif in our university.';

end

**Output:**



**Q4. Declare a variable and set a number in that. If the number is more than 100, then print that the number is large, if the number is less than 10 then print that the number is small. If the number is in the range 10-100 then print that the number is medium.**

Ans:

declare @number int;

set @number=50;

if @number>100

print 'the number is large.';

else

begin

if @number<10

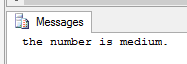
print 'the number is small.';

else

print 'the number is medium.';

end

**Output:**



**Conclusion:**

After doing this experiment, we came to know how to write Set Operation commands and have seen the outputs after they are executed.