National University of Computer and Emerging Sciences

Lab Manual 6

“Stored Procedures”

Database Systems

Spring 2022

Department of Computer Science
FAST-NU, Lahore, Pakistan

Contents

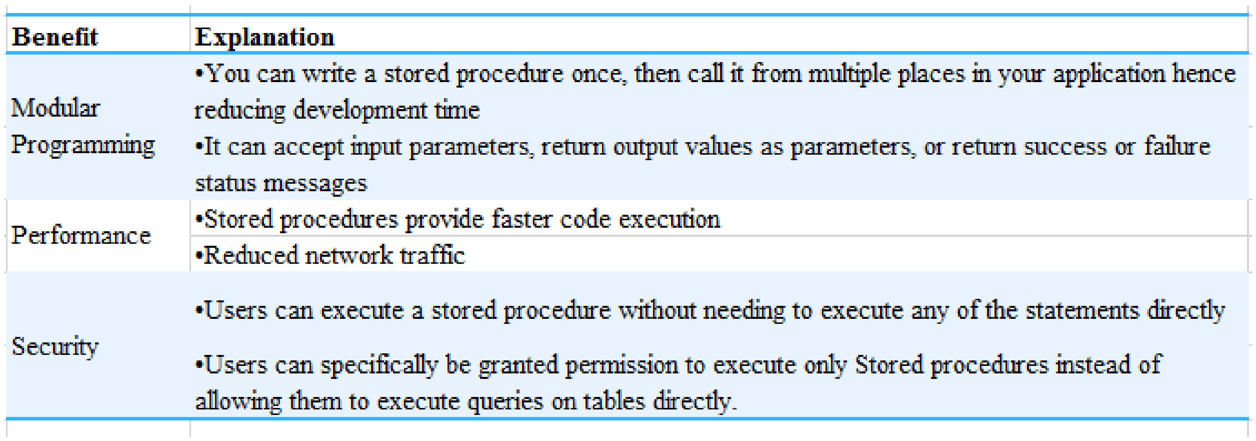
1 Objectives

The purpose of this lab is to get started with stored procedures. Why we should the stored procedure? How to create a stored procedure? Input/output parameters, if statement and while in stored procedure and procedure execution.

2 Stored Procedures

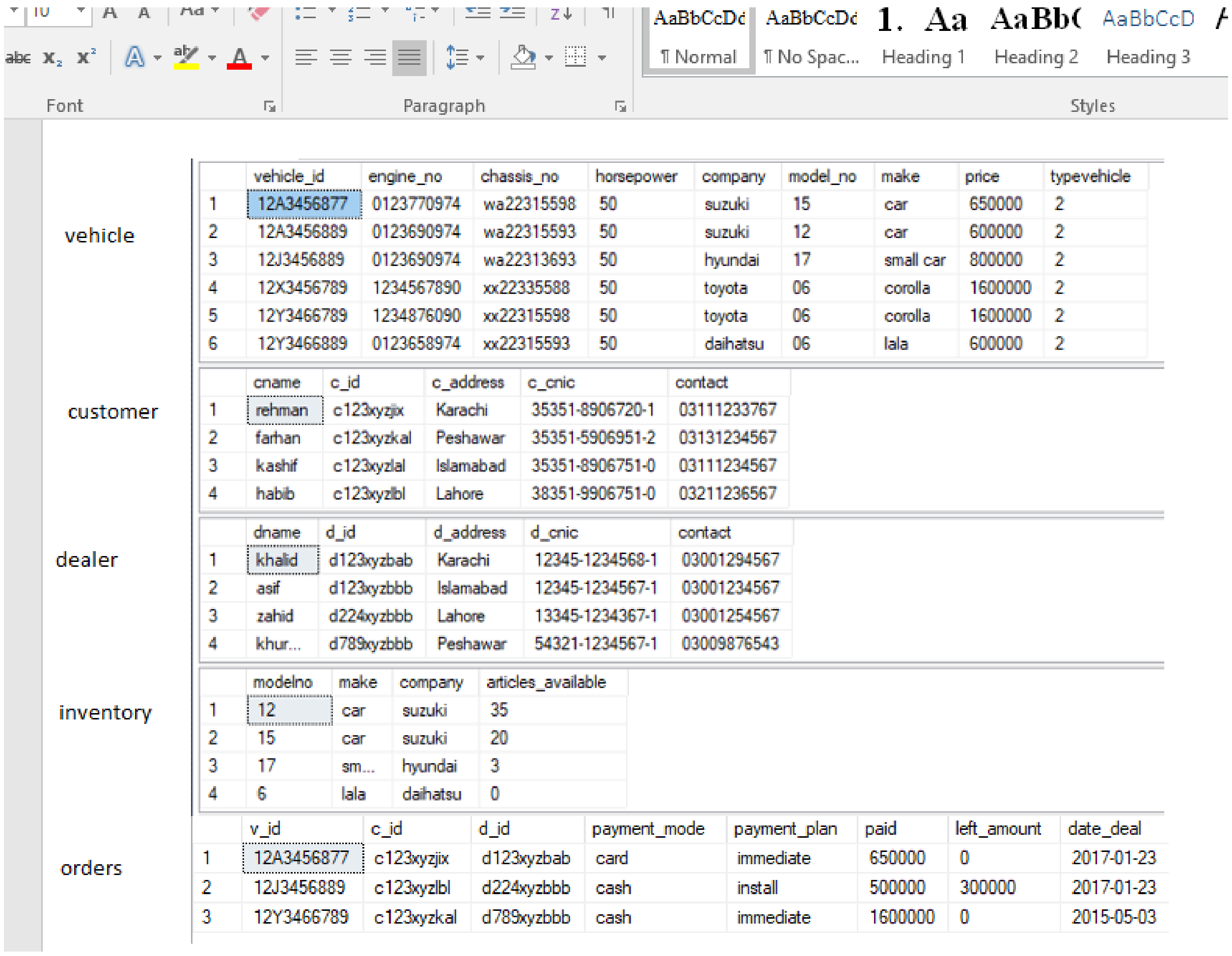
Stored Procedure in SQL server can be defined as the set of logically group of SQL statement which are grouped to perform a specific task. A stored procedure is a prepared SQL code that you save so that you can reuse the code over and over again.

2.1 Benefits of Stored Procedures



Benefit	Explanation
Modular Programming	•You can write a stored procedure once, then call it from multiple places in your application hence reducing development time •It can accept input parameters, return output values as parameters, or return success or failure status messages
Performance	•Stored procedures provide faster code execution •Reduced network traffic
Security	•Users can execute a stored procedure without needing to execute any of the statements directly •Users can specifically be granted permission to execute only Stored procedures instead of allowing them to execute queries on tables directly.

Every time you execute and SQL statements syntax Check, Compilation and done before Execution and Return data.
However, Syntax check and Compilation is done while creating a procedure, and not on every execution which makes in faster than simple SQL statements.



vehicle

	vehicle_id	engine_no	chassis_no	horsepower	company	model_no	make	price	typevehicle
1	12A3456877	0123770974	wa22315598	50	suzuki	15	car	650000	2
2	12A3456889	0123690974	wa22315593	50	suzuki	12	car	600000	2
3	12J3456889	0123690974	wa22313693	50	hyundai	17	small car	800000	2
4	12X3456789	1234567890	xx22335588	50	toyota	06	corolla	1600000	2
5	12Y3466789	1234876090	xx22315598	50	toyota	06	corolla	1600000	2
6	12Y3466889	0123658974	xx22315593	50	daihatsu	06	lala	600000	2

customer

	cname	c_id	c_address	c_cnic	contact
1	rehman	c123xyzjix	Karachi	35351-8906720-1	03111233767
2	farhan	c123xyzkal	Peshawar	35351-5906951-2	03131234567
3	kashif	c123xyzlal	Islamabad	35351-8906751-0	03111234567
4	habib	c123xyzlbl	Lahore	38351-9906751-0	03211236567

dealer

	dname	d_id	d_address	d_cnic	contact
1	khalid	d123xyzbab	Karachi	12345-1234568-1	03001294567
2	asif	d123xyzbbb	Islamabad	12345-1234567-1	03001234567
3	zahid	d224xyzbbb	Lahore	13345-1234367-1	03001254567
4	khur...	d789xyzbbb	Peshawar	54321-1234567-1	03009876543

inventory

	modelno	make	company	articles_available
1	12	car	suzuki	35
2	15	car	suzuki	20
3	17	sm...	hyundai	3
4	6	lala	daihatsu	0

orders

	v_id	c_id	d_id	payment_mode	payment_plan	paid	left_amount	date_deal
1	12A3456877	c123xyzjix	d123xyzbab	card	immediate	650000	0	2017-01-23
2	12J3456889	c123xyzlbl	d224xyzbbb	cash	install	500000	300000	2017-01-23
3	12Y3466789	c123xyzkal	d789xyzbbb	cash	immediate	1600000	0	2015-05-03

Let us consider the above schema and make some stored procedures on the given schema.

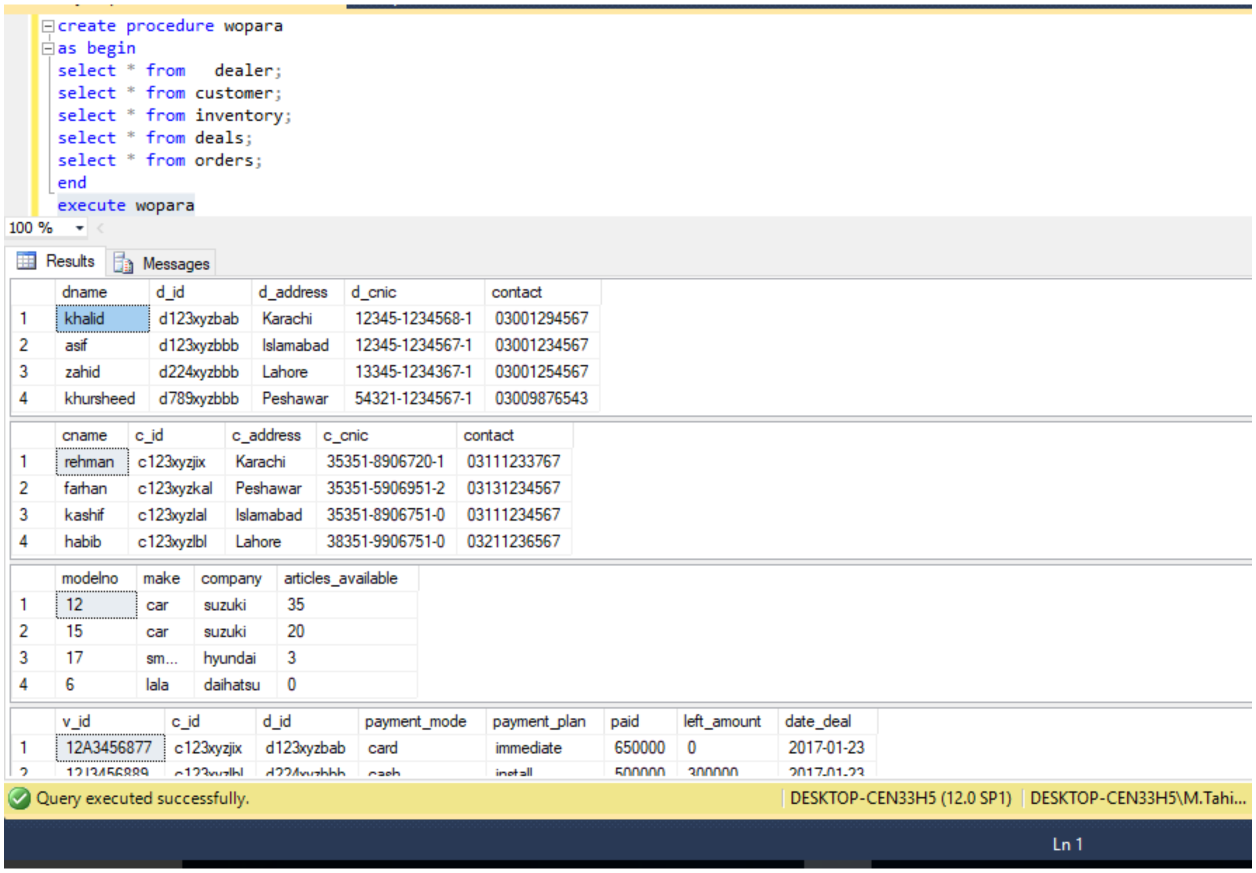
3 Types of stored procedures:

Stored procedures can be characterized on the basis of the types of arguments that can be sent to them. Stored procedures are quiet similar to the functions and methods that are used in C++ and other languages.
Before moving to the types of stored procedures let us first look at the general syntax of the stored procedures

Crete procedure <procedure name> @variable datatype, @variable2 datatype……
As begin
The code for your program
end

1) Stored Procedures without any parameters:

The syntax for the stored procedures is as follows:



```sql
create procedure wopara
as begin
select * from   dealer;
select * from customer;
select * from inventory;
select * from deals;
select * from orders;
end
execute wopara
```

100 %

Results | Messages

	dname	d_id	d_address	d_cnic	contact
1	khalid	d123xyzbab	Karachi	12345-1234568-1	03001294567
2	asif	d123xyzbbb	Islamabad	12345-1234567-1	03001234567
3	zahid	d224xyzbbb	Lahore	13345-1234367-1	03001254567
4	khursheed	d789xyzbbb	Peshawar	54321-1234567-1	03009876543

	cname	c_id	c_address	c_cnic	contact
1	rehman	c123xyzjix	Karachi	35351-8906720-1	03111233767
2	farhan	c123xyzkal	Peshawar	35351-5906951-2	03131234567
3	kashif	c123xyzlal	Islamabad	35351-8906751-0	03111234567
4	habib	c123xyzlbl	Lahore	38351-9906751-0	03211236567

	modelno	make	company	articles_available
1	12	car	suzuki	35
2	15	car	suzuki	20
3	17	sm...	hyundai	3
4	6	lala	daihatsu	0

	v_id	c_id	d_id	payment_mode	payment_plan	paid	left_amount	date_deal
1	12A3456877	c123xyzjix	d123xyzbab	card	immediate	650000	0	2017-01-23
2	12I3456889	c123xyzlbl	d224xyzbbb	cash	install	500000	300000	2017-01-23

Query executed successfully. | DESKTOP-CEN33H5 (12.0 SP1) | DESKTOP-CEN33H5\M.Tahi...

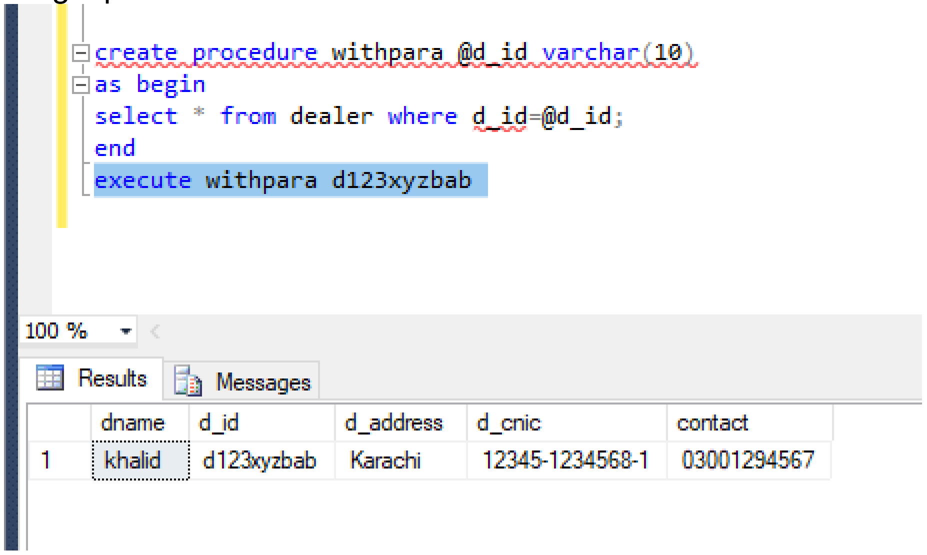
Ln 1

2) Stored Procedures with Parameters:

However if we want to send some parameters in case we want to manipulate the database according to some special values or some other certain feature we use this type of stored procedures. Let us look at the syntax for this.

3.1.1

1. Single parameters



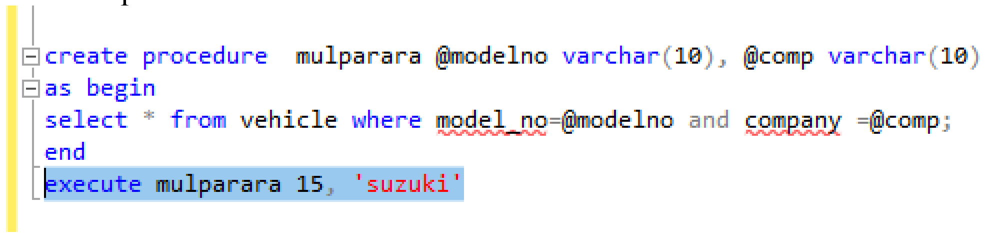
```sql
create procedure withpara @d_id varchar(10)
as begin
select * from dealer where d_id=@d_id;
end
execute withpara d123xyzbab
```

100 %

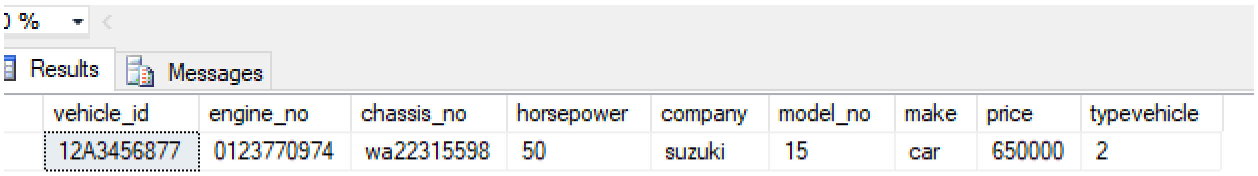
Results | Messages

	dname	d_id	d_address	d_cnic	contact
1	khalid	d123xyzbab	Karachi	12345-1234568-1	03001294567

2. Multiple Parameters:



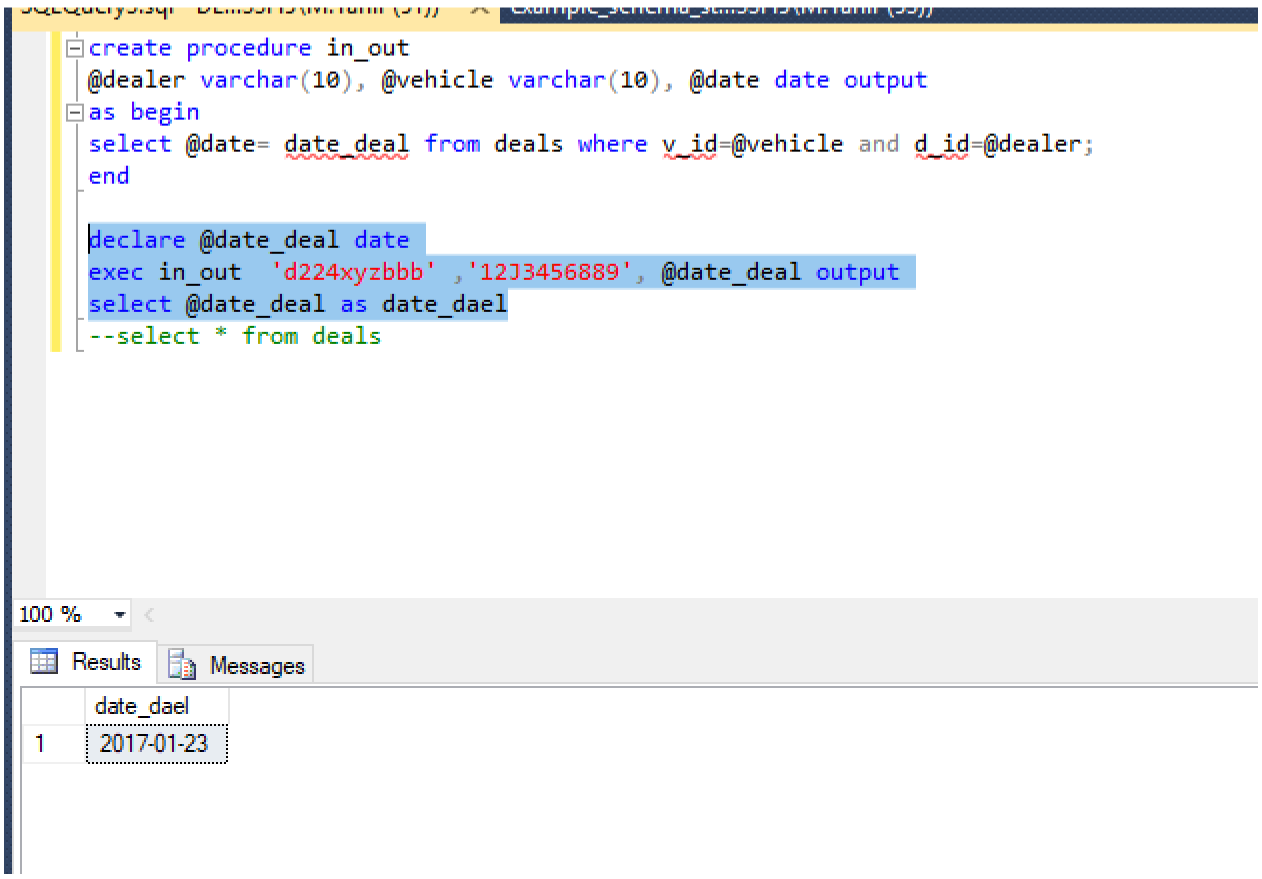
```
create procedure  mulparara @modelno varchar(10), @comp varchar(10)
as begin
select * from vehicle where model_no=@modelno and company =@comp;
end
execute mulparara 15, 'suzuki'
```



vehicle_id	engine_no	chassis_no	horsepower	company	model_no	make	price	typevehicle
12A3456877	0123770974	wa22315598	50	suzuki	15	car	650000	2

3) Stored Procedures with input and output parameters:

Till now we have seen only the input parameters now we shall see output parameters as well

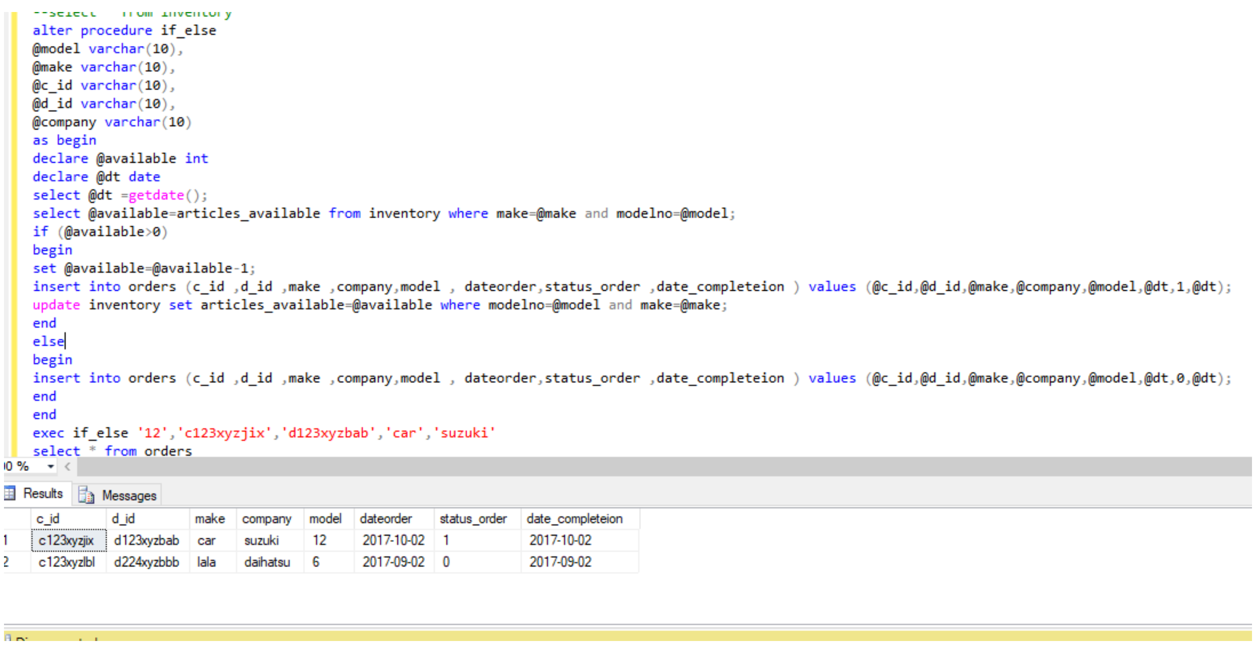

```
create procedure in_out
@dealer varchar(10), @vehicle varchar(10), @date date output
as begin
select @date= date_deal from deals where v_id=@vehicle and d_id=@dealer;
end

declare @date_deal date
exec in_out  'd224xyzbbb' ,'12J3456889', @date_deal output
select @date_deal as date_dael
--select * from deals
```

```
100 %
Results  Messages
   date_dael
1  2017-01-23
```

4 Control Structures in Stored Procedures:

4.1 If Else

Like functions in other languages stored procedures also provide the liberty of using control structures.

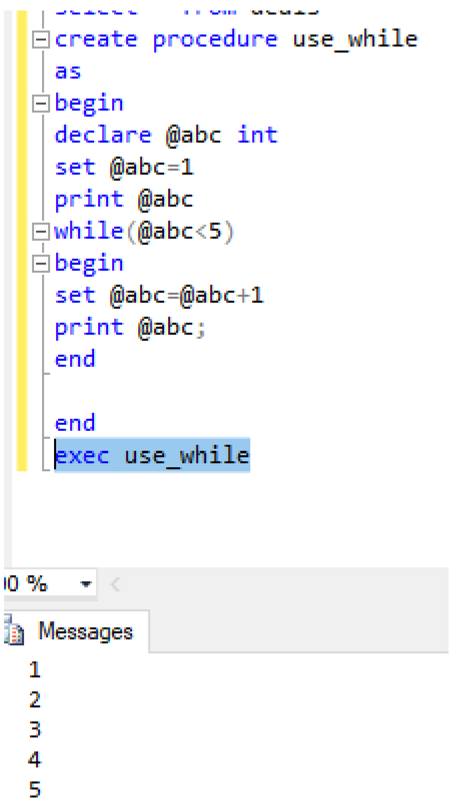


```
alter procedure if_else
@model varchar(10),
@make varchar(10),
@c_id varchar(10),
@d_id varchar(10),
@company varchar(10)
as begin
declare @available int
declare @dt date
select @dt =getdate();
select @available=articles_available from inventory where make=@make and modelno=@model;
if (@available>0)
begin
set @available=@available-1;
insert into orders (c_id ,d_id ,make ,company,model , dateorder,status_order ,date_completeion ) values (@c_id,@d_id,@make,@company,@model,@dt,1,@dt);
update inventory set articles_available=@available where modelno=@model and make=@make;
end
else
begin
insert into orders (c_id ,d_id ,make ,company,model , dateorder,status_order ,date_completeion ) values (@c_id,@d_id,@make,@company,@model,@dt,0,@dt);
end
end
exec if_else '12','c123xyzjix','d123xyzbab','car','suzuki'
select * from orders
```

	c_id	d_id	make	company	model	dateorder	status_order	date_completeion
1	c123xyzjix	d123xyzbab	car	suzuki	12	2017-10-02	1	2017-10-02
2	c123xyzlbl	d224xyzbbb	lala	daihatsu	6	2017-09-02	0	2017-09-02

In this example it can be easily seen that we can do anything we want in a stored procedure.

4.2 While



```
create procedure use_while
as
begin
declare @abc int
set @abc=1
print @abc
while(@abc<5)
begin
set @abc=@abc+1
print @abc;
end

end
exec use_while
```

Messages

```
1
2
3
4
5
```

This is a very simple example while loops are used rarely in real world scenarios but you must have a little know how about it.

5 Variables.

Like in any other programing language SQL also provides scalar variables, which are very useful when creating stored procedures. We have seen almost all of these in the examples given earlier
However let us take a closer look:

- Variable in SQL start with @ symbol
- Variable is declared using DECLARE keyword as follow
  - *DECLARE @variableName datatype;*

    Or to declare multiple variables in one statement.
  - *DECLARE @variable1Name Datatype,@variable2Name datatype;*
- Variable can be assigned a constant scalar value as follow
  - *SET @ variableName   = value;*

    Or To assign values to multiple variables in one statement
  - *select @ variable1Name   = value, @variable2Name =value;*
- Variable can be assigned a scalar value thought SQL statement as well
  - *SELECT @vairableName = columnName FROM Table WHERE <condition>*

    If SQL query returns more than one row, 1st value will be assigned to variable
- You can retrieve the value of variable as follow
  - *Select @variableName*
- You can perform operations on variables like addition, concatenation, substring etc

References

- Chapter 5 Lesson 1 and Lesson 4, MCTS 70-433 SQLServer 2008 Database Development.
- Chapter 5 Elmasri