CSIS05I

Database Systems II

Lab (8)

INTRODUCTION TO STORED PROCEDURESII-SOLUTIONS

Exercise

1- Retrieve the customer ID, first name and the order ID of order that customer number 1 did.

```
CREATE PROCEDURE questionA1
@customerID INT
AS
SELECT CustomerID ,CustFirstName , OrderID
FROM Customer, Orderr
WHERE CID=CustomerID
and CustomerID=@customerID
Go

EXEC questionA1 1;
```

2- Retrieve the information of order 101 and its status.

```
CREATE PROCEDURE Getorder_withparametersdefault
@orderID INT= 101
AS
SELECT OrderID ,OrderStatus
FROM Orderr
WHERE OrderID=@orderID
Go

EXEC Getorder_withparametersdefault;
```

3- Once the user has entered or passed the product name (ex: tv), you should retrieve this product name and also its description. Then print them out with the alias “The name of the product is TV and its description is Smart TV”.

```
Create PROCEDURE GetDescription_Name
--Input parameters
@Pname varchar(50),
--Output parameters
@ProductDescriptionAndName varchar(100) OUT

AS
select @ProductDescriptionAndName = (select Pname +' and its
description is '+ Pdescription
FROM Product
where Pname =@Pname)

Print 'The Name of the product is ' + @ProductDescriptionAndName
GO

Execute GetDescription_Name 'TV' , OUT
```

OR you can print the output parameter outside the procedure:

```
alter PROCEDURE GetDescription_Name
--Input parameters
@Pname varchar(50),
--Output parameters
@ProductDescriptionAndName varchar(100) OUT

AS
select @ProductDescriptionAndName = (select Pname +' and its
description is '+ Pdescription
FROM Product
where Pname =@Pname)
GO


Declare @p_d varchar(100)
Execute GetDescription_Name 'TV' , @p_d OUT
Print 'The Name of the product is ' + @p_d
```

Another solution for question 3:

```
Create Procedure ProductDescription
@Pname varchar(20) ,
@PDescription varchar (30) out
AS
SELECT @PDescription = PDescription
FROM Product
WHERE PName = @Pname;
print 'The name of the product is '+ @Pname +' and its description is ' +
@PDescription ;
Go
EXEC ProductDescription 'TV', out;
```

4- Calculate the area of the circle (A=PI *r*r) note that the Pi= 3.14

```
Create PROCEDURE CalculateTheAreaOfCicrle
@Radius float,
@area float OUT
AS
SELECT @area=@Radius*@Radius*3.14;
Go

DECLARE @Area float
Execute CalculateTheAreaOfCicrle 5, @Area OUT
Print 'The area of a circle is:'
Print @Area
```

```
OR

Create OR ALTER PROCEDURE CalculateTheAreaOfCicrle
@Radius float,
@area float OUT
AS
DECLARE @PI float = 3.14
SELECT @area=@Radius*@Radius*@PI;
Print @area
Go

DECLARE @Area float
Execute CalculateTheAreaOfCicrle '5', @Area OUT
```

5- Hide all the information for the customer and his order using stored procedure.

```
CREATE PROCEDURE EncryptCustomerInfoAndTheirOrders
WITH ENCRYPTION
AS
SELECT CustomerID,CustFirstName,OrderID
from Customer, Orderr
where CID=CustomerID
Go

execute EncryptCustomerInfoAndTheirOrders

exec sp_helptext 'EncryptCustomerInfoAndTheirOrders'
```

6- Copy the records of table product into another table using stored procedure (you can specify different values of product description). *(Extra Exercise Question)*

```
create table Product2(
PID int not null,
Pname varchar(10),
Pdescription varchar(20),

constraint Prod_PK2 primary key (PID)
);

CREATE PROCEDURE ProductInfo
@ProductDesc varchar(20)
AS
insert into Product2
select *
from Product
where PDescription=@ProductDesc
GO

exec ProductInfo 'smart tv'

Select * from Product2;
```

Example 10: (EXTRA Question Solution)

Create a procedure called CustomerDataCheck that check if a customer is a new customer, it will allow the user to enter his/her data into Customer table. If not, it will update his/her data with new values. And write the execution statement.

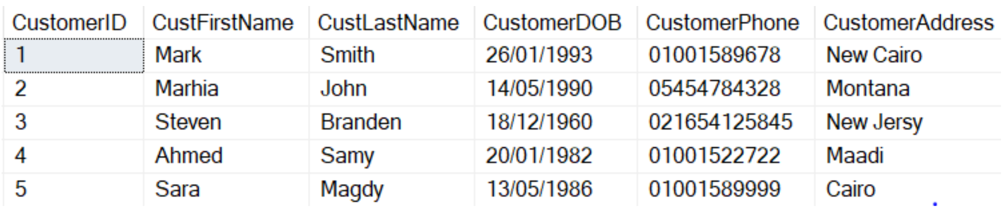
```
CREATE OR ALTER PROC CustomerDataCheck
@CustomerID int,
@FirstName varchar(30),
@LastName varchar(30),
@DOB varchar(30),
@phone varchar(30),
@CustAddress varchar(30)
AS
BEGIN
       IF (EXISTS(SELECT * FROM Customer WHERE CustomerID = @CustomerID))
       BEGIN
              UPDATE Customer
              SET CustFirstName = @FirstName , CustLastName = @LastName , CustomerDOB =
@DOB, CustomerPhone = @phone, CustomerAddress = @CustAddress
              WHERE CustomerID = @CustomerID
       END

       ELSE
       BEGIN
              insert into Customer
        (CustomerID, CustFirstName, CustLastName, CustomerDOB, CustomerPhone,
CustomerAddress)
              values (@CustomerID, @FirstName, @LastName, @DOB, @phone, @CustAddress)
       END
END
GO

-- Calling to Update an Existing Record With New Values if the Customer ID is Already
Exists--
Execute CustomerDataCheck '4', 'Ahmed', 'Samy', '20/01/1982', '01001522722', 'Maadi'

-- Calling to Add a New Record to the same table if the Customer ID not Exists --
Execute CustomerDataCheck '5', 'Sara', 'Magdy', '13/05/1986', '01001589999', 'Cairo'

-- Testing Procedure --
Select * from Customer;
```



CustomerID	CustFirstName	CustLastName	CustomerDOB	CustomerPhone	CustomerAddress
1	Mark	Smith	26/01/1993	01001589678	New Cairo
2	Marhia	John	14/05/1990	05454784328	Montana
3	Steven	Branden	18/12/1960	021654125845	New Jersy
4	Ahmed	Samy	20/01/1982	01001522722	Maadi
5	Sara	Magdy	13/05/1986	01001589999	Cairo