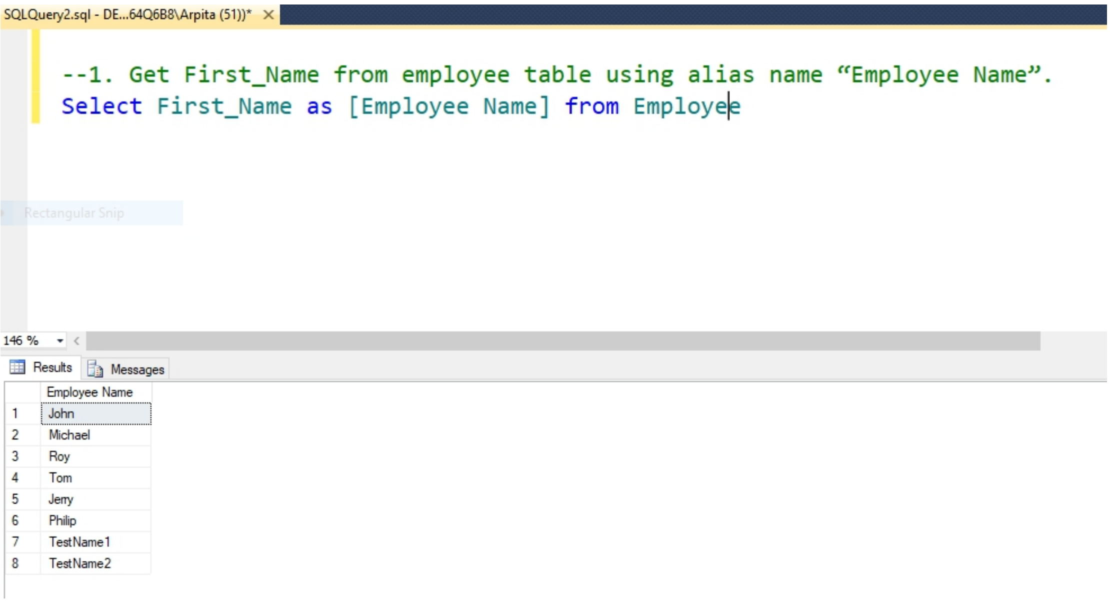
SQL Server Test

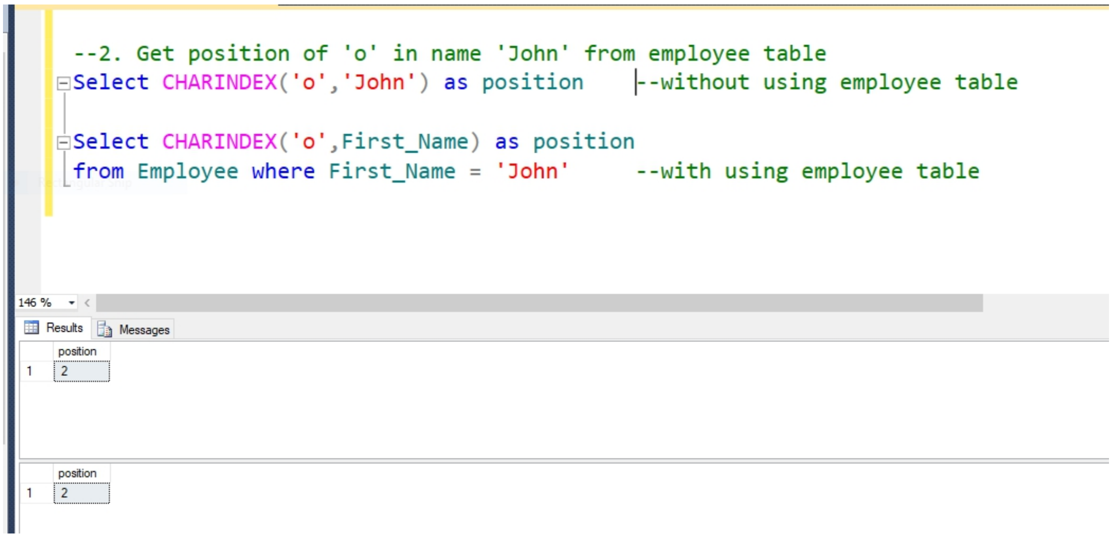
1. Get First_Name from employee table using alias name “Employee Name”.



```sql
--1. Get First_Name from employee table using alias name “Employee Name”.
Select First_Name as [Employee Name] from Employee
```

	Employee Name
1	John
2	Michael
3	Roy
4	Tom
5	Jerry
6	Philip
7	TestName1
8	TestName2

2. Get position of 'o' in name 'John' from employee table .

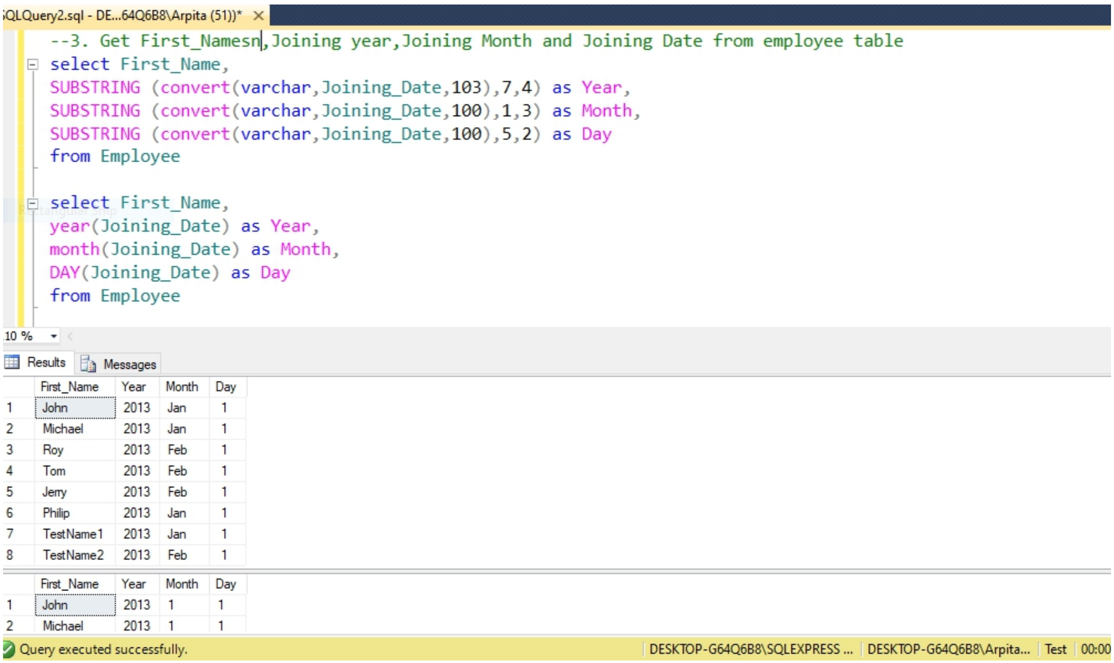


```sql
--2. Get position of 'o' in name 'John' from employee table
Select CHARINDEX('o','John') as position      --without using employee table

Select CHARINDEX('o',First_Name) as position
from Employee where First_Name = 'John'      --with using employee table
```

	position
1	2

	position
1	2

3. Get FIRST_NAME ,Joining year,Joining Month and Joining Date from employee table.



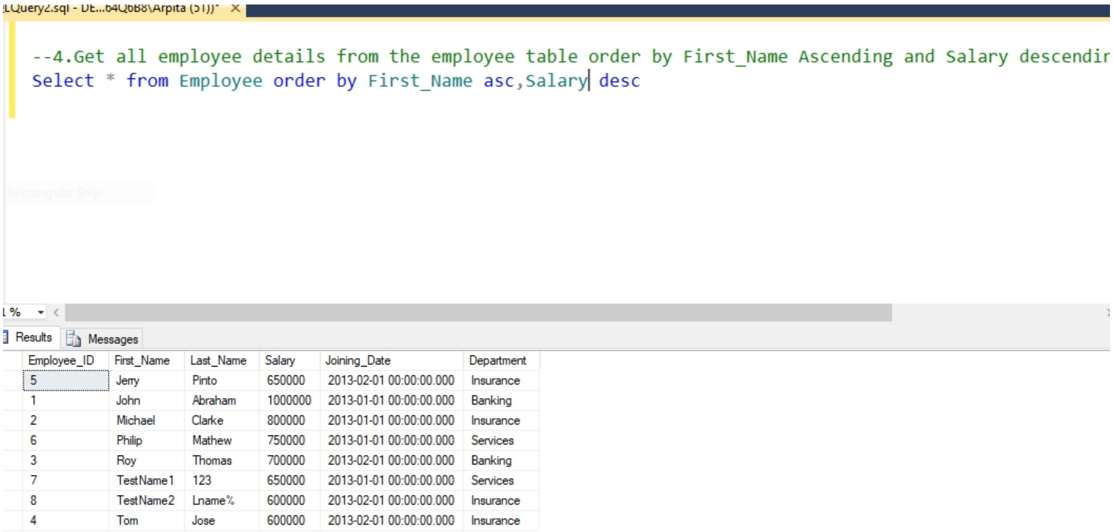
```sql
--3. Get First_Namesn,Joining year,Joining Month and Joining Date from employee table
select First_Name,
SUBSTRING (convert(varchar,Joining_Date,103),7,4) as Year,
SUBSTRING (convert(varchar,Joining_Date,100),1,3) as Month,
SUBSTRING (convert(varchar,Joining_Date,100),5,2) as Day
from Employee

select First_Name,
year(Joining_Date) as Year,
month(Joining_Date) as Month,
DAY(Joining_Date) as Day
from Employee
```

	First_Name	Year	Month	Day
1	John	2013	Jan	1
2	Michael	2013	Jan	1
3	Roy	2013	Feb	1
4	Tom	2013	Feb	1
5	Jerry	2013	Feb	1
6	Philip	2013	Jan	1
7	TestName1	2013	Jan	1
8	TestName2	2013	Feb	1

	First_Name	Year	Month	Day
1	John	2013	1	1
2	Michael	2013	1	1

Query executed successfully. | DESKTOP-G64Q6B8\SQLEXPRESS ... | DESKTOP-G64Q6B8\Arpita... | Test | 00:00

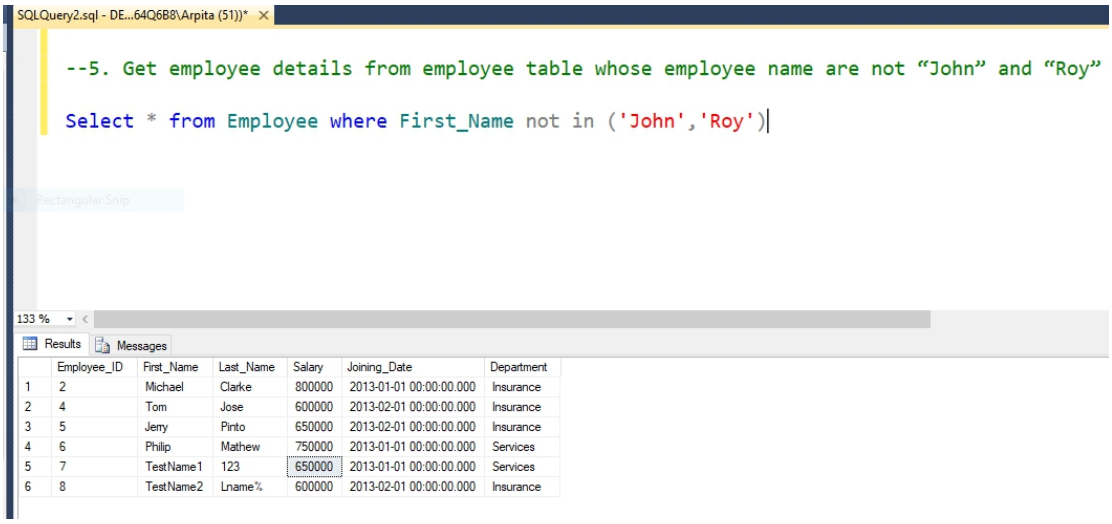
4. Get all employee details from the employee table order by First_Name Ascending and Salary descending.



```sql
--4.Get all employee details from the employee table order by First_Name Ascending and Salary descendir
Select * from Employee order by First_Name asc,Salary desc
```

Employee_ID	First_Name	Last_Name	Salary	Joining_Date	Department
5	Jerry	Pinto	650000	2013-02-01 00:00:00.000	Insurance
1	John	Abraham	1000000	2013-01-01 00:00:00.000	Banking
2	Michael	Clarke	800000	2013-01-01 00:00:00.000	Insurance
6	Philip	Mathew	750000	2013-01-01 00:00:00.000	Services
3	Roy	Thomas	700000	2013-02-01 00:00:00.000	Banking
7	TestName1	123	650000	2013-01-01 00:00:00.000	Services
8	TestName2	Lname%	600000	2013-02-01 00:00:00.000	Insurance
4	Tom	Jose	600000	2013-02-01 00:00:00.000	Insurance

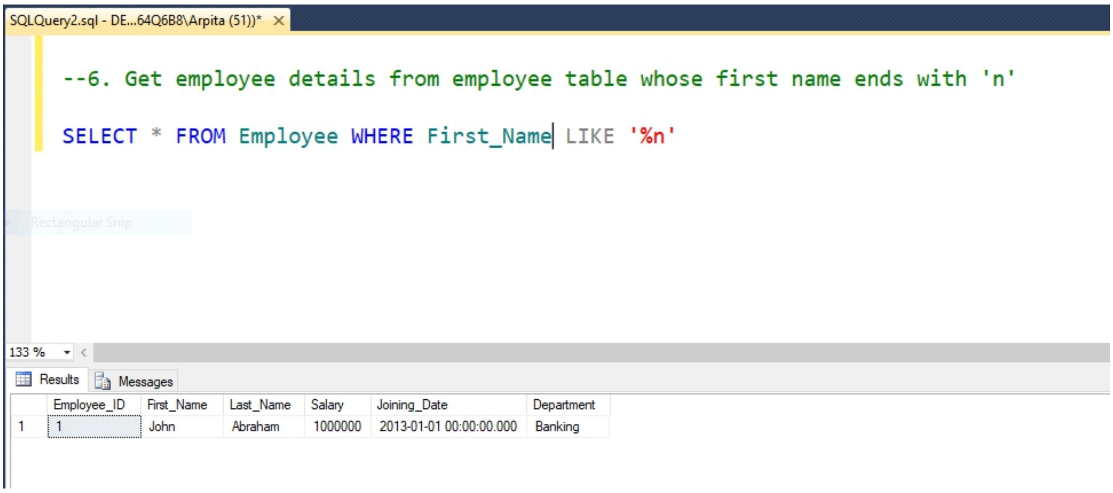
5. Get employee details from employee table whose employee name are not “John” and “Roy”.



```
--5. Get employee details from employee table whose employee name are not “John” and “Roy”
Select * from Employee where First_Name not in ('John','Roy')
```

	Employee_ID	First_Name	Last_Name	Salary	Joining_Date	Department
1	2	Michael	Clarke	800000	2013-01-01 00:00:00.000	Insurance
2	4	Tom	Jose	600000	2013-02-01 00:00:00.000	Insurance
3	5	Jerry	Pinto	650000	2013-02-01 00:00:00.000	Insurance
4	6	Philip	Mathew	750000	2013-01-01 00:00:00.000	Services
5	7	TestName1	123	650000	2013-01-01 00:00:00.000	Services
6	8	TestName2	Lname%	600000	2013-02-01 00:00:00.000	Insurance

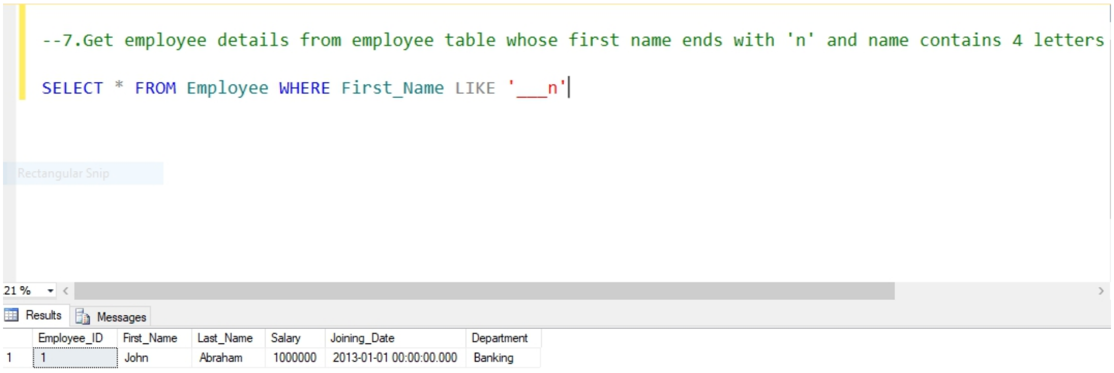
6. Get employee details from employee table whose first name ends with 'n'.



```
--6. Get employee details from employee table whose first name ends with 'n'
SELECT * FROM Employee WHERE First_Name LIKE '%n'
```

	Employee_ID	First_Name	Last_Name	Salary	Joining_Date	Department
1	1	John	Abraham	1000000	2013-01-01 00:00:00.000	Banking

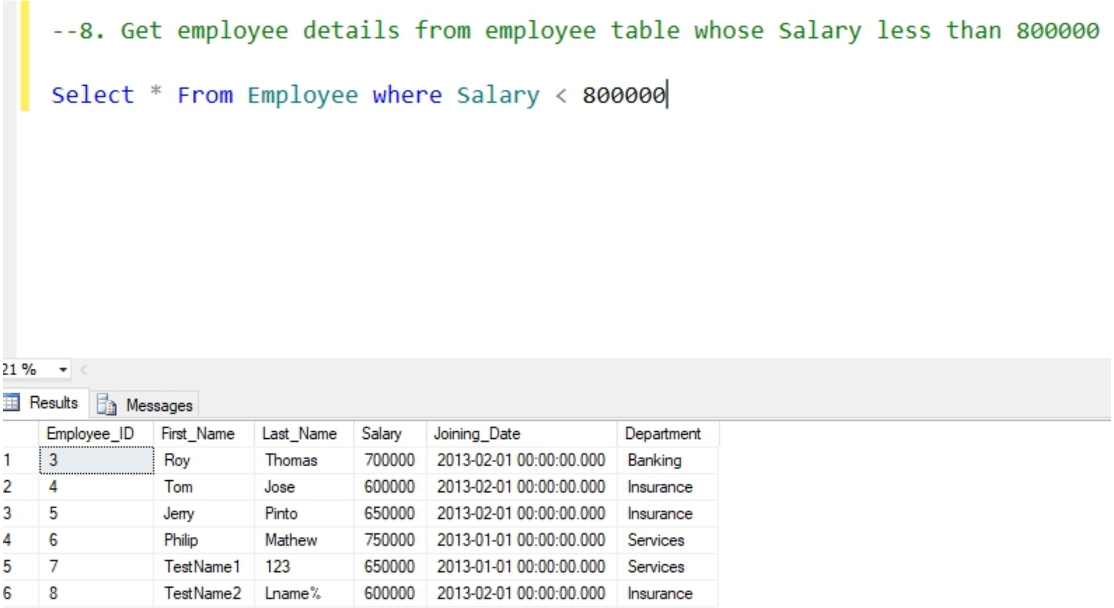
7. Get employee details from employee table whose first name ends with 'n' and name contains 4 letters.



```
--7.Get employee details from employee table whose first name ends with 'n' and name contains 4 letters
SELECT * FROM Employee WHERE First_Name LIKE '___n'
```

	Employee_ID	First_Name	Last_Name	Salary	Joining_Date	Department
1	1	John	Abraham	1000000	2013-01-01 00:00:00.000	Banking

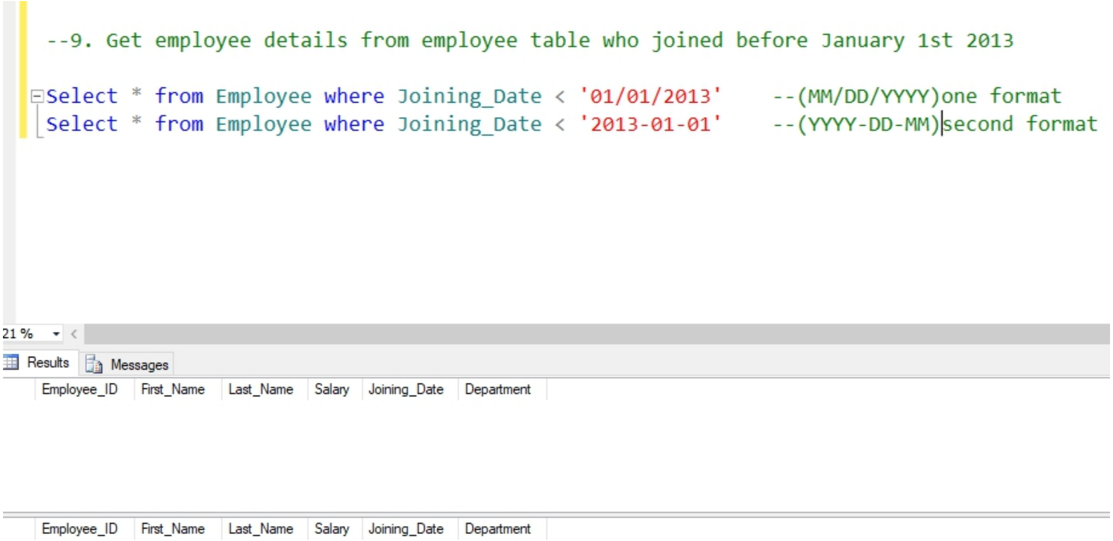
8. Get employee details from employee table whose Salary less than 800000.



```
--8. Get employee details from employee table whose Salary less than 800000

Select * From Employee where Salary < 800000
```

	Employee_ID	First_Name	Last_Name	Salary	Joining_Date	Department
1	3	Roy	Thomas	700000	2013-02-01 00:00:00.000	Banking
2	4	Tom	Jose	600000	2013-02-01 00:00:00.000	Insurance
3	5	Jerry	Pinto	650000	2013-02-01 00:00:00.000	Insurance
4	6	Philip	Mathew	750000	2013-01-01 00:00:00.000	Services
5	7	TestName1	123	650000	2013-01-01 00:00:00.000	Services
6	8	TestName2	Lname%	600000	2013-02-01 00:00:00.000	Insurance

9. Get employee details from employee table who joined before January 1st 2013

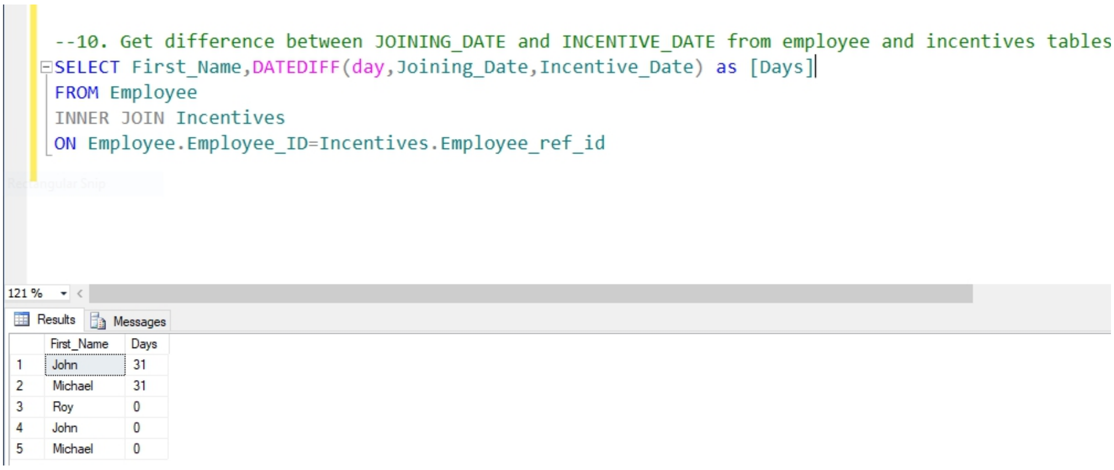


```
--9. Get employee details from employee table who joined before January 1st 2013

Select * from Employee where Joining_Date < '01/01/2013'     --(MM/DD/YYYY)one format
Select * from Employee where Joining_Date < '2013-01-01'     --(YYYY-DD-MM)second format
```

Employee_ID	First_Name	Last_Name	Salary	Joining_Date	Department

Employee_ID	First_Name	Last_Name	Salary	Joining_Date	Department

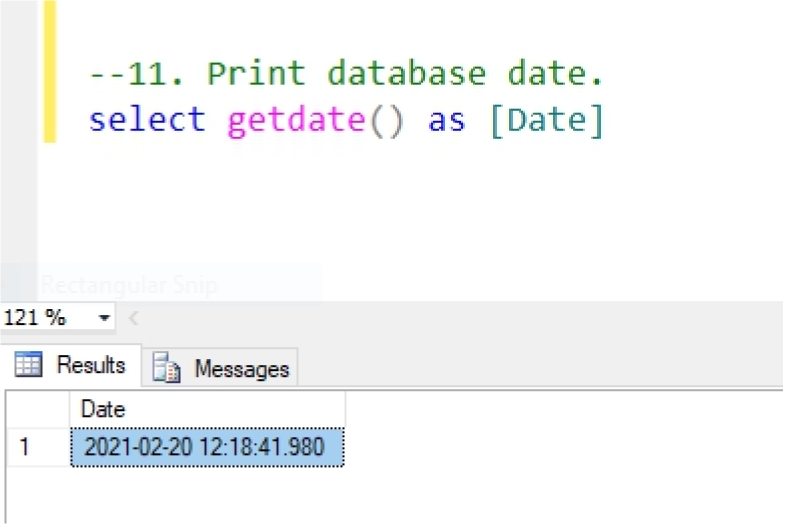
10. Get difference between JOINING_DATE and INCENTIVE_DATE from employee and incentives table



```sql
--10. Get difference between JOINING_DATE and INCENTIVE_DATE from employee and incentives tables
SELECT First_Name,DATEDIFF(day,Joining_Date,Incentive_Date) as [Days]
FROM Employee
INNER JOIN Incentives
ON Employee.Employee_ID=Incentives.Employee_ref_id
```

	First_Name	Days
1	John	31
2	Michael	31
3	Roy	0
4	John	0
5	Michael	0

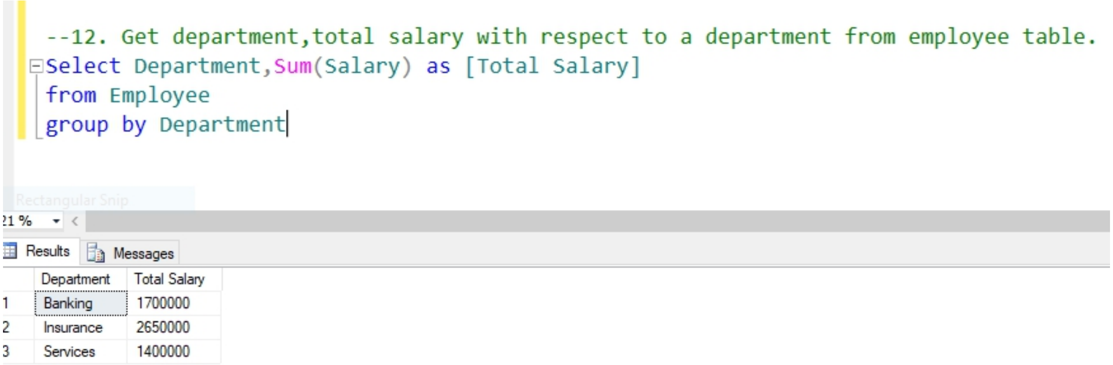
11. Print database date.



```sql
--11. Print database date.
select getdate() as [Date]
```

	Date
1	2021-02-20 12:18:41.980

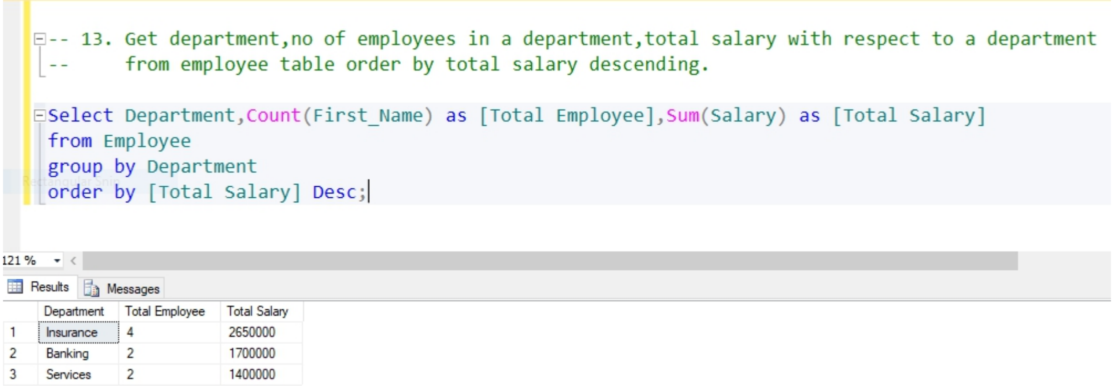
12. Get department,total salary with respect to a department from employee table.



```sql
--12. Get department,total salary with respect to a department from employee table.
Select Department,Sum(Salary) as [Total Salary]
from Employee
group by Department
```

	Department	Total Salary
1	Banking	1700000
2	Insurance	2650000
3	Services	1400000

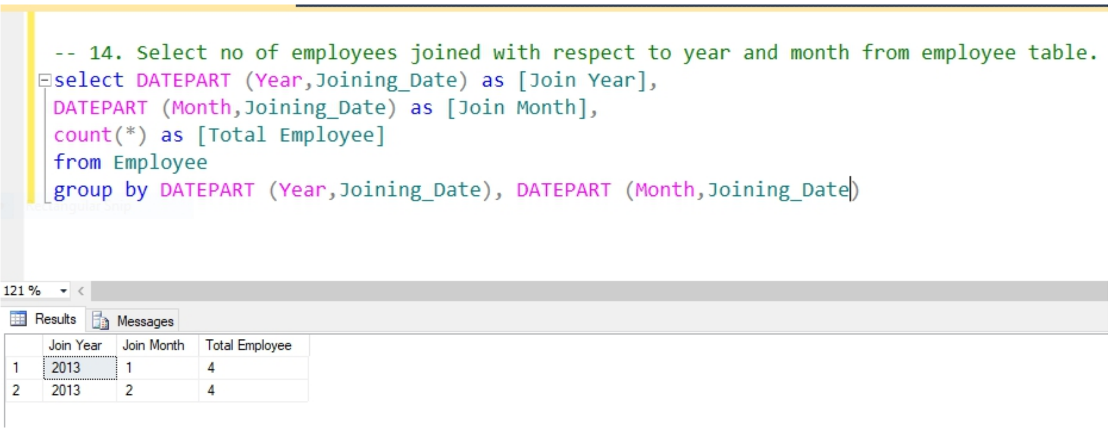
13. Get department,no of employees in a department,total salary with respect to a department from employee table order by total salary descending.



```
-- 13. Get department,no of employees in a department,total salary with respect to a department
--     from employee table order by total salary descending.

Select Department,Count(First_Name) as [Total Employee],Sum(Salary) as [Total Salary]
from Employee
group by Department
order by [Total Salary] Desc;
```

	Department	Total Employee	Total Salary
1	Insurance	4	2650000
2	Banking	2	1700000
3	Services	2	1400000

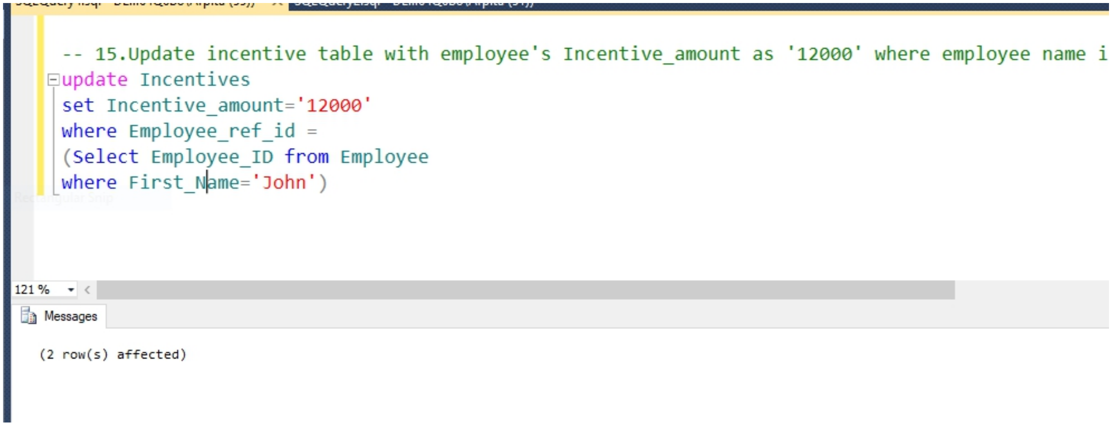
14. Select no of employees joined with respect to year and month from employee table.



```
-- 14. Select no of employees joined with respect to year and month from employee table.
select DATEPART (Year,Joining_Date) as [Join Year],
DATEPART (Month,Joining_Date) as [Join Month],
count(*) as [Total Employee]
from Employee
group by DATEPART (Year,Joining_Date), DATEPART (Month,Joining_Date)
```

	Join Year	Join Month	Total Employee
1	2013	1	4
2	2013	2	4

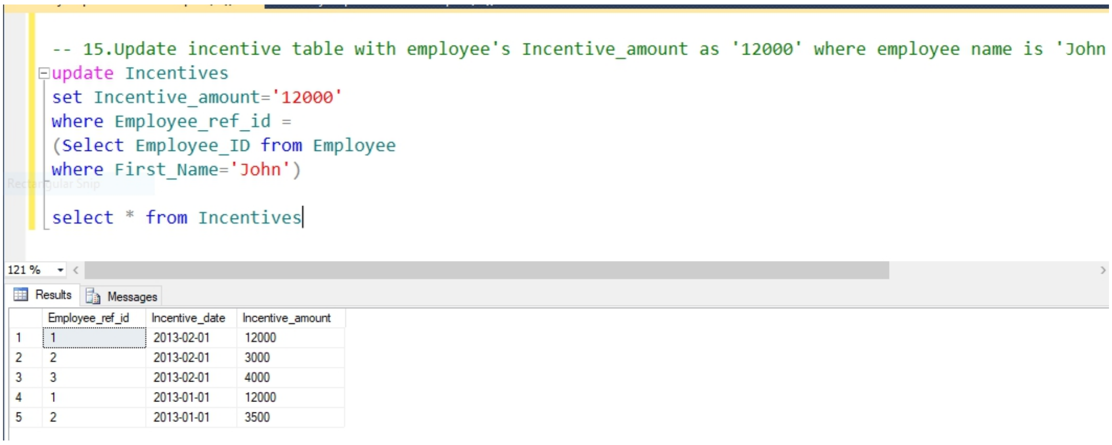
15. Update incentive table with employee's Incentive_amount as '12000' where employee name is 'John'.



```
-- 15.Update incentive table with employee's Incentive_amount as '12000' where employee name i
update Incentives
set Incentive_amount='12000'
where Employee_ref_id =
(Select Employee_ID from Employee
where First_Name='John')
```

```
(2 row(s) affected)
```

(1)

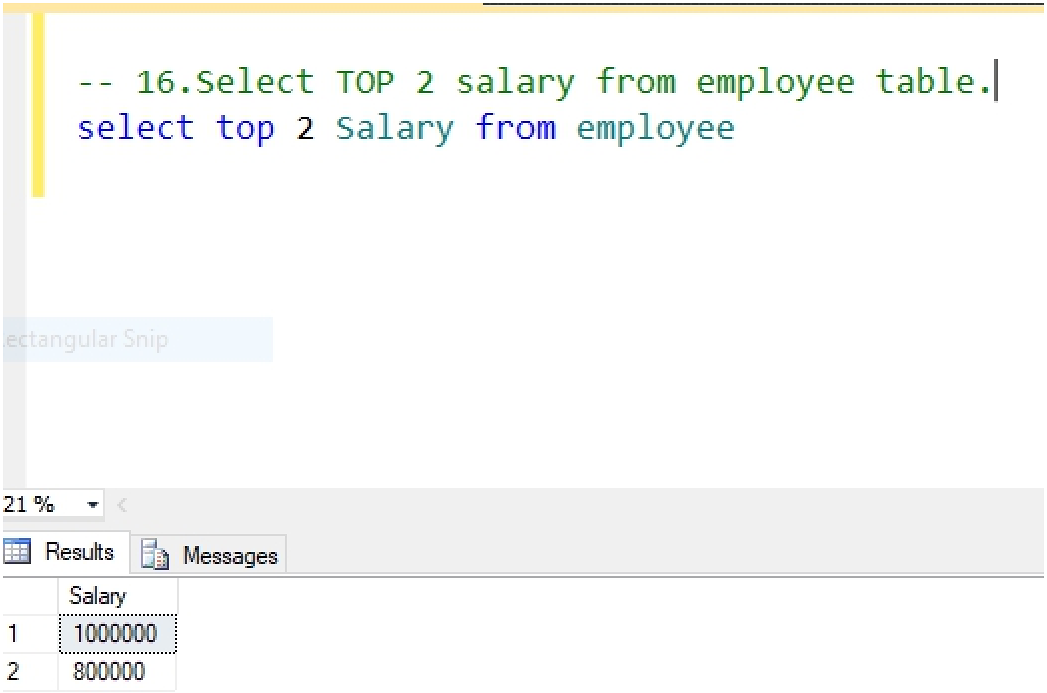


```
-- 15.Update incentive table with employee's Incentive_amount as '12000' where employee name is 'John
update Incentives
set Incentive_amount='12000'
where Employee_ref_id =
(Select Employee_ID from Employee
where First_Name='John')

select * from Incentives
```

	Employee_ref_id	Incentive_date	Incentive_amount
1	1	2013-02-01	12000
2	2	2013-02-01	3000
3	3	2013-02-01	4000
4	1	2013-01-01	12000
5	2	2013-01-01	3500

(2)

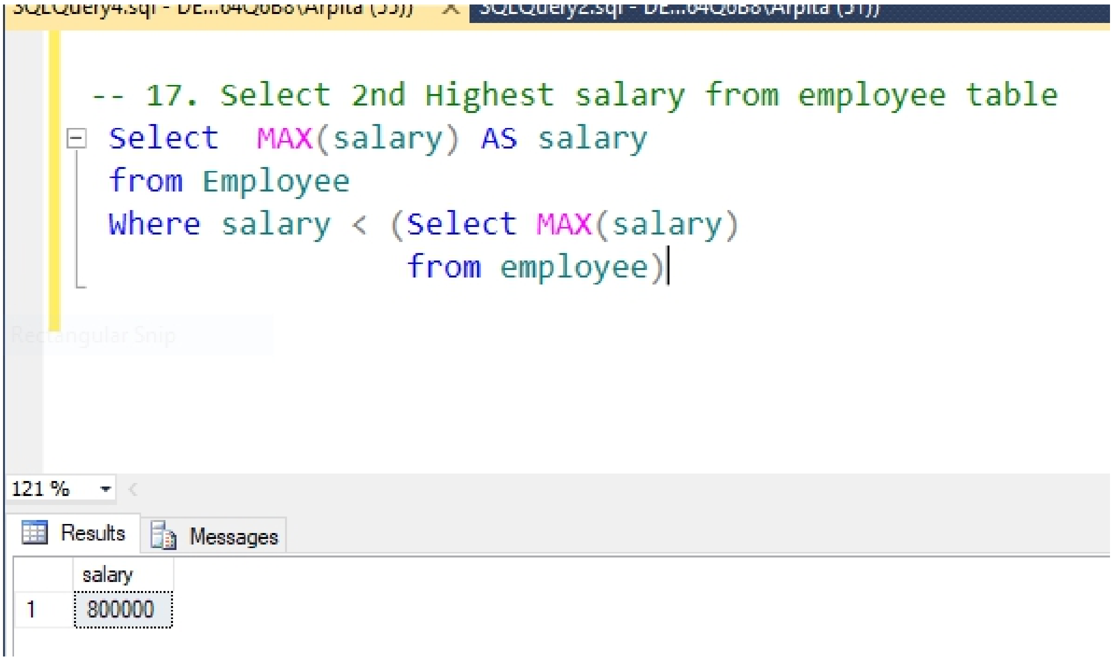
16. Select TOP 2 salary from employee table.



```
-- 16.Select TOP 2 salary from employee table.
select top 2 Salary from employee
```

	Salary
1	1000000
2	800000

17. Select 2nd Highest salary from employee table.


SQLQuery4.sql - DE...04Q0B8\Arpita (55))
SQLQuery2.sql - DE...04Q0B8\Arpita (51))
121 %
Results
Messages

```
-- 17. Select 2nd Highest salary from employee table
Select  MAX(salary) AS salary
from Employee
Where salary < (Select MAX(salary)
                from employee)
```

```
   salary
1  800000
```

18. Write. What is the difference between UNION and UNION ALL ?

- UNION and UNION ALL both are used to combine the results of two similar queries.
- The only difference between these two is : UNION will eliminate duplicate records in the result set while UNION ALL will include all duplicate rows too.

UNION: only keeps unique records
UNION ALL: keeps all records, including duplicates

- UNION or UNION ALL have the same basic requirements of the data being combined:

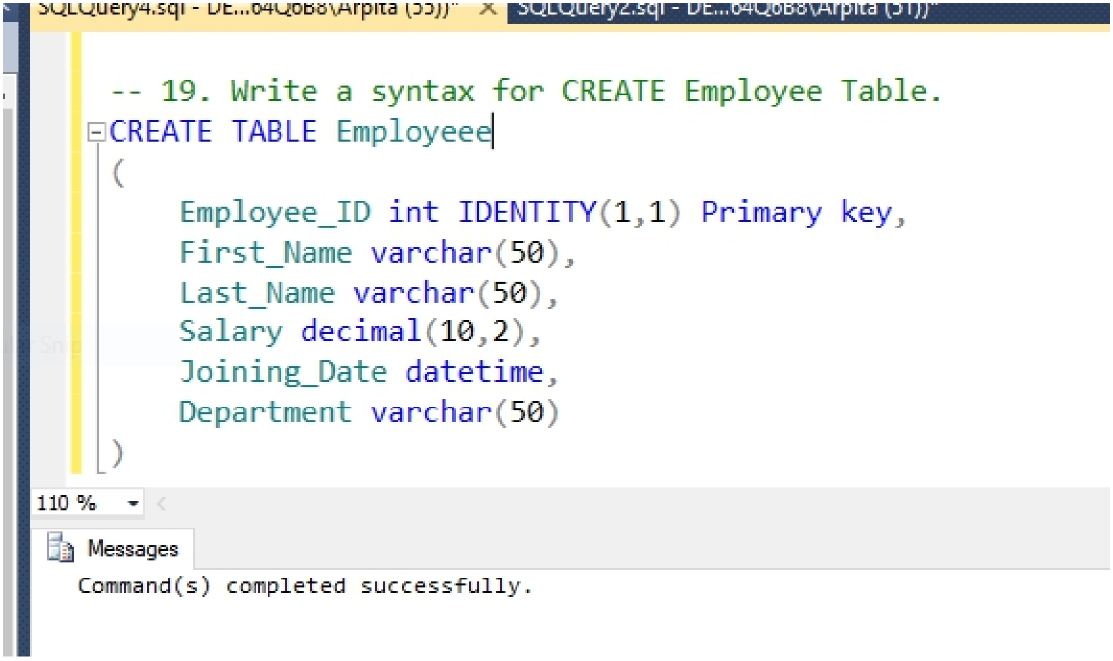
1)There must be the same number of columns retrieved in each SELECT statement to be combined.

2)The columns retrieved must be in the same order in each SELECT statement.

3)The columns retrieved must be of similar data types.

- For example, if FIRST_NAME is DOUBLE and LAST_NAME is STRING above query wont work.Since the data type of both the columns are VARCHAR, union is made possible.

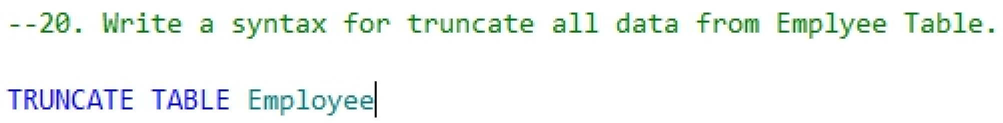
19. Write a syntax for CREATE Employee Table.


SQLQuery4.sql - DE...64Q6B8\Arpita (55))*
SQLQuery2.sql - DE...64Q6B8\Arpita (51))*
110 %
Messages

```
-- 19. Write a syntax for CREATE Employee Table.
CREATE TABLE Employeee
(
     Employee_ID int IDENTITY(1,1) Primary key,
     First_Name varchar(50),
     Last_Name varchar(50),
     Salary decimal(10,2),
     Joining_Date datetime,
     Department varchar(50)
)
```

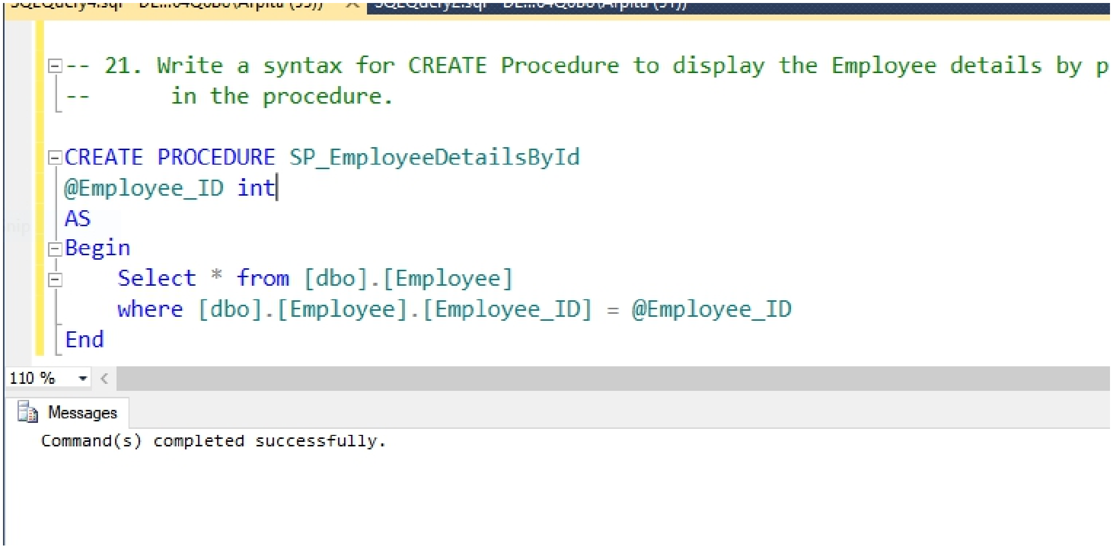
```
Command(s) completed successfully.
```

20. Write a syntax for truncate all data from Employee Table.



```
--20. Write a syntax for truncate all data from Emplyee Table.

TRUNCATE TABLE Employee
```

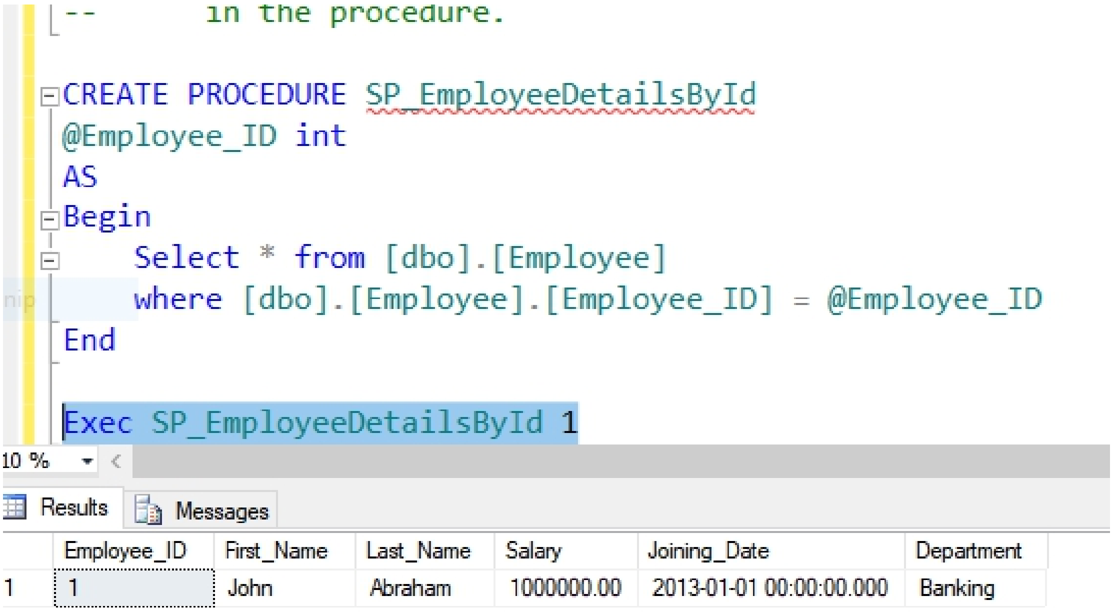
21. Write a syntax for CREATE Procedure to display the Employee details by passing the “Employee Id” in the procedure.



```
-- 21. Write a syntax for CREATE Procedure to display the Employee details by p
--      in the procedure.

CREATE PROCEDURE SP_EmployeeDetailsById
@Employee_ID int
AS
Begin
    Select * from [dbo].[Employee]
    where [dbo].[Employee].[Employee_ID] = @Employee_ID
End
```

```
Command(s) completed successfully.
```

(1)

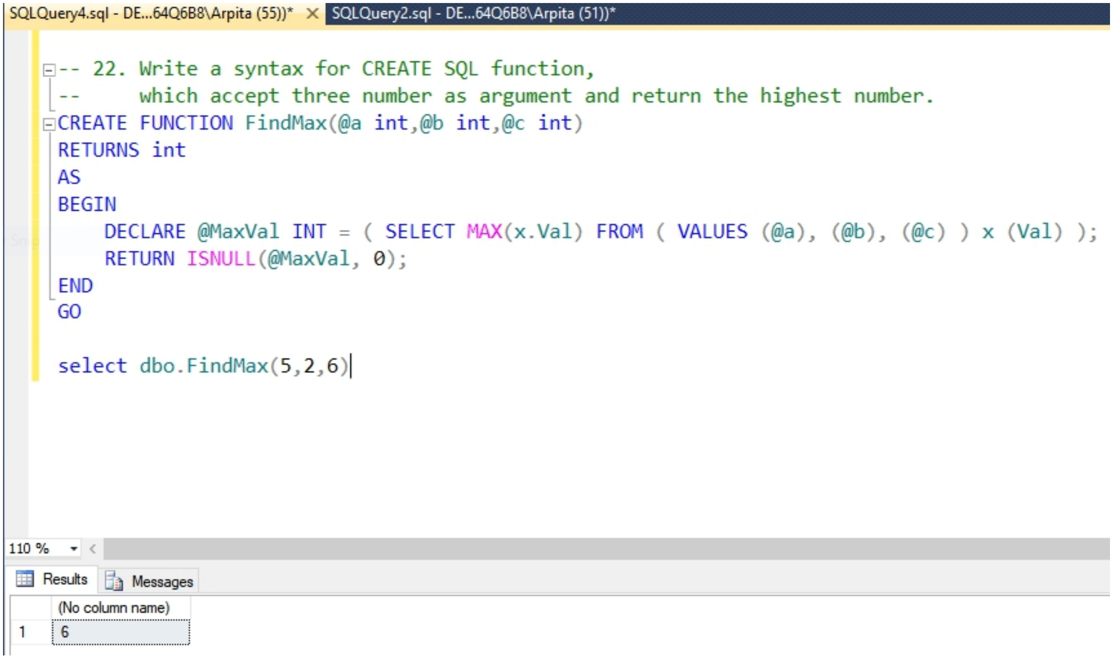


```
--        in the procedure.

CREATE PROCEDURE SP_EmployeeDetailsById
@Employee_ID int
AS
Begin
    Select * from [dbo].[Employee]
    where [dbo].[Employee].[Employee_ID] = @Employee_ID
End

Exec SP_EmployeeDetailsById 1
```

	Employee_ID	First_Name	Last_Name	Salary	Joining_Date	Department
1	1	John	Abraham	1000000.00	2013-01-01 00:00:00.000	Banking

(2)

22. Write a syntax for CREATE SQL function, which accept three number as argument and return the highest number.



SQLQuery4.sql - DE...64Q6B8\Arpita (55))*
SQLQuery2.sql - DE...64Q6B8\Arpita (51))*
110 %
Results
Messages

```
-- 22. Write a syntax for CREATE SQL function,
--     which accept three number as argument and return the highest number.
CREATE FUNCTION FindMax(@a int,@b int,@c int)
RETURNS int
AS
BEGIN
    DECLARE @MaxVal INT = ( SELECT MAX(x.Val) FROM ( VALUES (@a), (@b), (@c) ) x (Val) );
    RETURN ISNULL(@MaxVal, 0);
END
GO

select dbo.FindMax(5,2,6)
```

	(No column name)
1	6

23.Write a syntax for Update the Employee's salary whose department is“Insurance”.

Syntax:

```
Update Employee
Set Salary=<new Salary>
Where Department = ‘Insurance’
```

24.State the difference between varchar and nvarchar.

- About the Varchar:

  1) It is a variable that has a length data type.
  2) It is use to store non-Unicode characters.
  3) It Occupy 1 byte(8-bit representation) of space for each character.

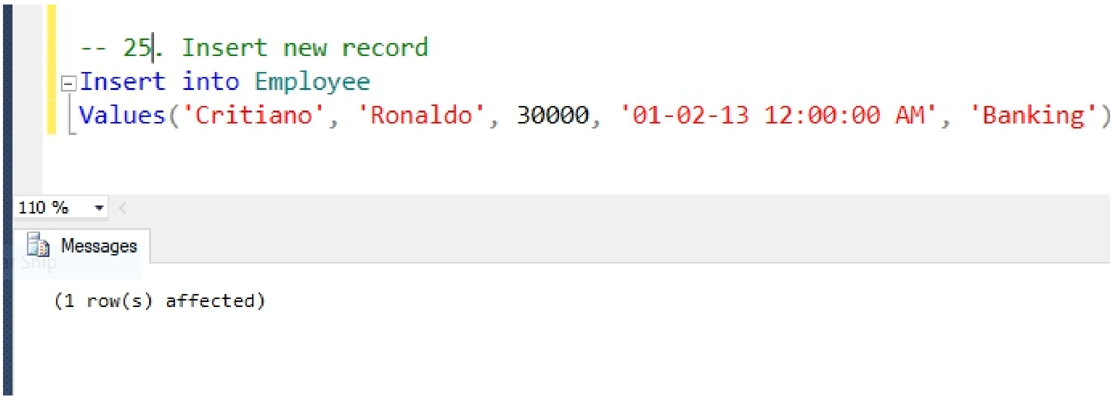
  Example:DECLARE @name VARCHAR(20)

- About the NVARCHAR data:

  1) It is a variable that has length data type.
  2) It is use to store Unicode characters.
  3) It Occupy 2 bytes(16-bit representation) of space for each character.

  Example::DECLARE @name NVARCHAR(20)

25. Write a query that insert the data into Employee table, data as mentioned. {First name : 'Critiano' , Last name : 'Ronaldo' , Salary : '30000' , Joining Date : '01-FEB-13 12.00.00 AM' , Department : 'Banking' }



```sql
-- 25. Insert new record
Insert into Employee
Values('Critiano', 'Ronaldo', 30000, '01-02-13 12:00:00 AM', 'Banking')
```

```
(1 row(s) affected)
```