2021

SQL Server

TEST OF SQL SERVER

JYOT HATHI

Index

A. Given Reference Table Creation and Data Insertion 3

1. Employee Table: 3

2. Department Table (To Normalize Employee Table): 4

3. Incentive Table: 4

B. Execution of Give Queries: 5

1. Get First_Name from employee table using alias name “Employee Name” 5

2. Get position of 'o' in name 'John' from employee table 5

3. Get FIRST_NAME ,Joining year,Joining Month and Joining Date from employee table 5

4. Get all employee details from the employee table order by First_Name Ascending and Salary descending. 6

5. Get employee details from employee table whose employee name are not “John” and “Roy”. 6

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

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

8. Get employee details from employee table whose Salary less than 800000 7

9. Get employee details from employee table who joined before January 1st 2013 7

10. Get difference between JOINING_DATE and INCENTIVE_DATE from employee and incentives table 8

11. Print database date. 8

12. Get department,total salary with respect to a department from employee table. 8

13. Get department,no of employees in a department,total salary with respect to a department from employee table order by total salary descending 9

14. Select no of employees joined with respect to year and month from employee table. 9

15. Update incentive table with employee's Incentive_amount as '12000' where employee name is 'John' 9

16. Select TOP 2 salary from employee table 10

17. Select 2nd Highest salary from employee table 10

18. Write. What is the difference between UNION and UNION ALL? 10

19. Write a syntax for CREATE Employee Table. 10

20. Write a syntax for truncate all data from Emplyee Table. 11

21. Write a syntax for CREATE Procedure to display the Employee details by passing the “Employee Id” in the procedure. 11

22. Write a syntax for CREATE SQL function, which accept three number as argument and return the highest number. 11

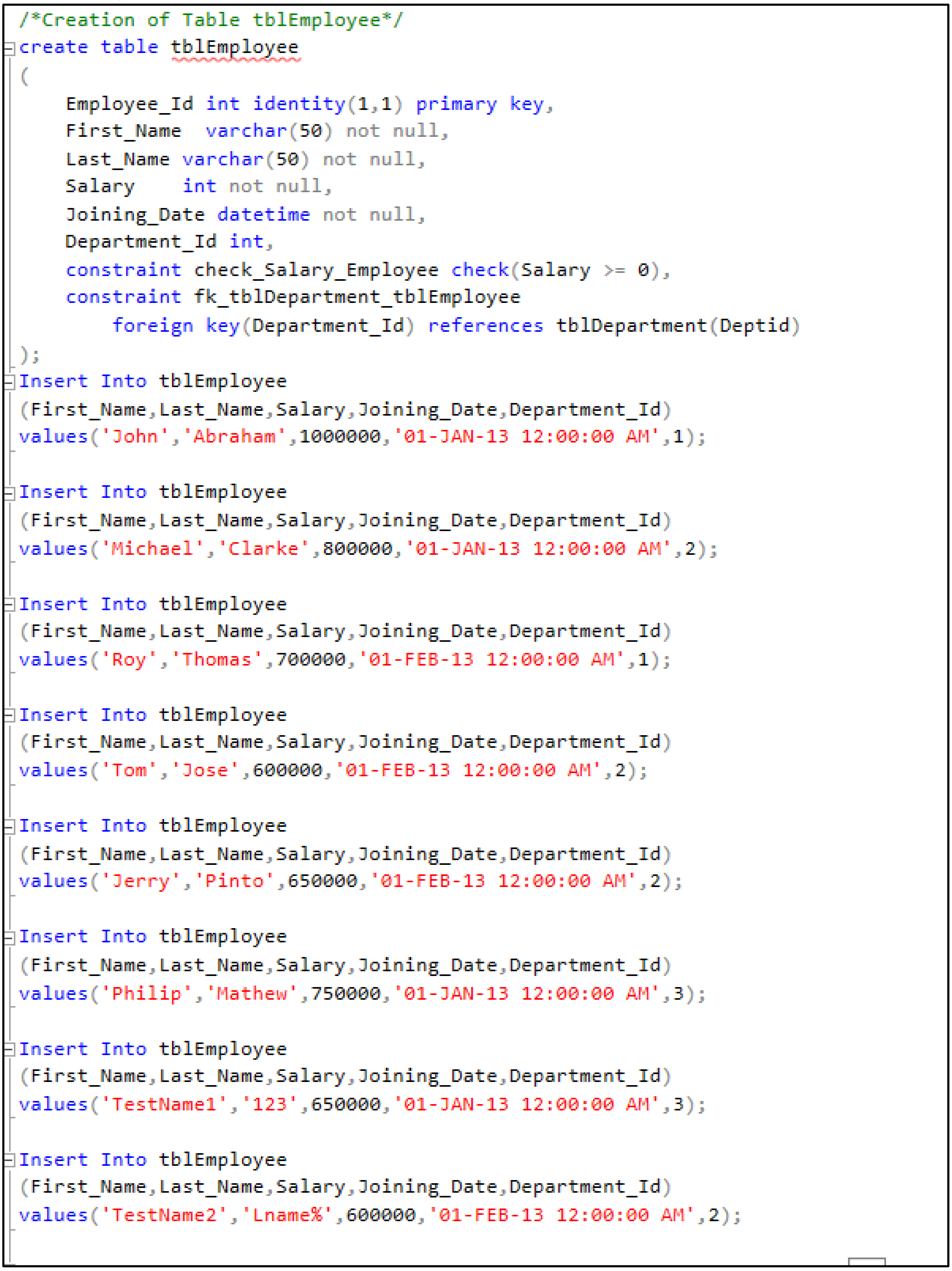
23. Write a syntax for Update the Employee's salary whose department is “Insurance”. . 12

24. State the difference between varchar and nvarchar. 12

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' } 13

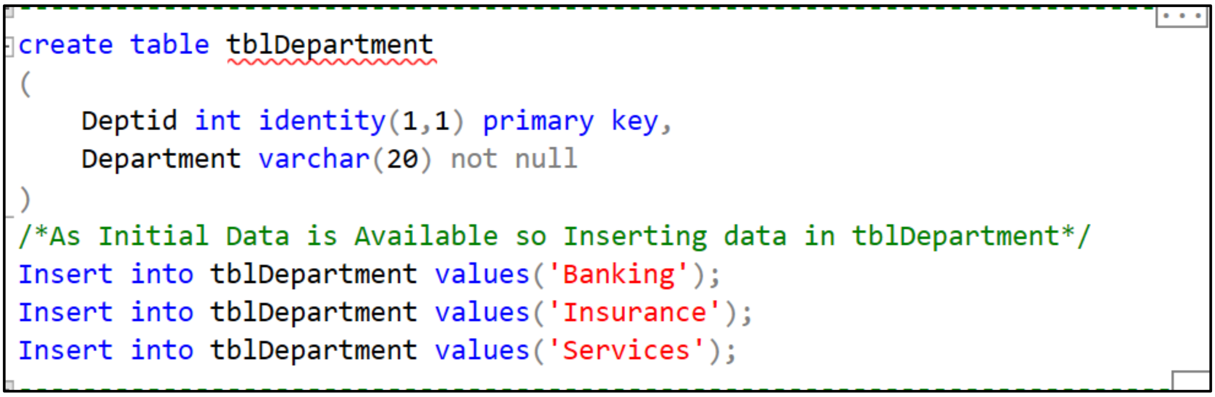
A. Given Reference Table Creation and Data Insertion

1. Employee Table:



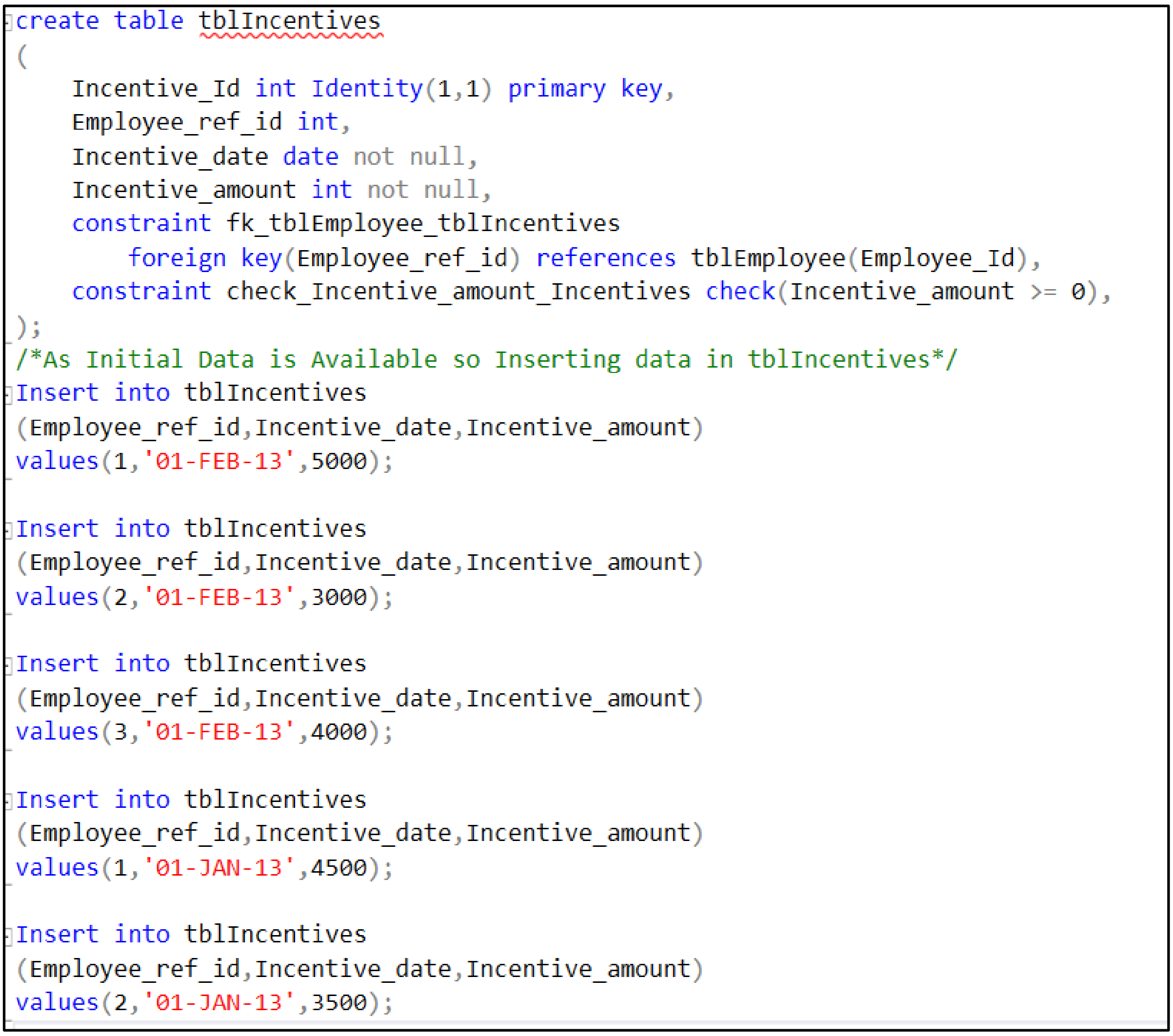
```
/*Creation of Table tblEmployee*/
create table tblEmployee
(
    Employee_Id int identity(1,1) primary key,
    First_Name  varchar(50) not null,
    Last_Name varchar(50) not null,
    Salary    int not null,
    Joining_Date datetime not null,
    Department_Id int,
    constraint check_Salary_Employee check(Salary >= 0),
    constraint fk_tblDepartment_tblEmployee
        foreign key(Department_Id) references tblDepartment(Deptid)
);
Insert Into tblEmployee
(First_Name,Last_Name,Salary,Joining_Date,Department_Id)
values('John','Abraham',1000000,'01-JAN-13 12:00:00 AM',1);

Insert Into tblEmployee
(First_Name,Last_Name,Salary,Joining_Date,Department_Id)
values('Michael','Clarke',800000,'01-JAN-13 12:00:00 AM',2);

Insert Into tblEmployee
(First_Name,Last_Name,Salary,Joining_Date,Department_Id)
values('Roy','Thomas',700000,'01-FEB-13 12:00:00 AM',1);

Insert Into tblEmployee
(First_Name,Last_Name,Salary,Joining_Date,Department_Id)
values('Tom','Jose',600000,'01-FEB-13 12:00:00 AM',2);

Insert Into tblEmployee
(First_Name,Last_Name,Salary,Joining_Date,Department_Id)
values('Jerry','Pinto',650000,'01-FEB-13 12:00:00 AM',2);

Insert Into tblEmployee
(First_Name,Last_Name,Salary,Joining_Date,Department_Id)
values('Philip','Mathew',750000,'01-JAN-13 12:00:00 AM',3);

Insert Into tblEmployee
(First_Name,Last_Name,Salary,Joining_Date,Department_Id)
values('TestName1','123',650000,'01-JAN-13 12:00:00 AM',3);

Insert Into tblEmployee
(First_Name,Last_Name,Salary,Joining_Date,Department_Id)
values('TestName2','Lname%',600000,'01-FEB-13 12:00:00 AM',2);
```

2. Department Table (To Normalize Employee Table):



```
create table tblDepartment
(
    Deptid int identity(1,1) primary key,
    Department varchar(20) not null
)
/*As Initial Data is Available so Inserting data in tblDepartment*/
Insert into tblDepartment values('Banking');
Insert into tblDepartment values('Insurance');
Insert into tblDepartment values('Services');
```

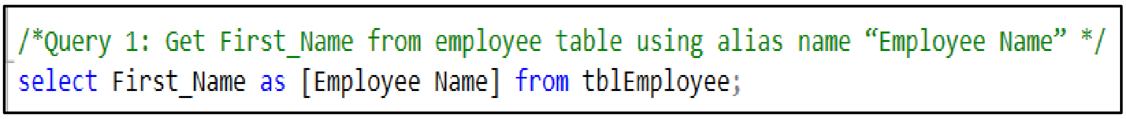
3. Incentive Table:



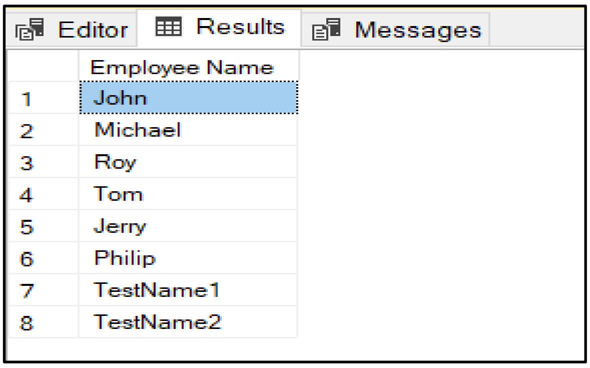
```
create table tblIncentives
(
    Incentive_Id int Identity(1,1) primary key,
    Employee_ref_id int,
    Incentive_date date not null,
    Incentive_amount int not null,
    constraint fk_tblEmployee_tblIncentives
        foreign key(Employee_ref_id) references tblEmployee(Employee_Id),
    constraint check_Incentive_amount_Incentives check(Incentive_amount >= 0),
);
/*As Initial Data is Available so Inserting data in tblIncentives*/
Insert into tblIncentives
(Employee_ref_id,Incentive_date,Incentive_amount)
values(1,'01-FEB-13',5000);

Insert into tblIncentives
(Employee_ref_id,Incentive_date,Incentive_amount)
values(2,'01-FEB-13',3000);

Insert into tblIncentives
(Employee_ref_id,Incentive_date,Incentive_amount)
values(3,'01-FEB-13',4000);

Insert into tblIncentives
(Employee_ref_id,Incentive_date,Incentive_amount)
values(1,'01-JAN-13',4500);

Insert into tblIncentives
(Employee_ref_id,Incentive_date,Incentive_amount)
values(2,'01-JAN-13',3500);
```

B. Execution of Give Queries:

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

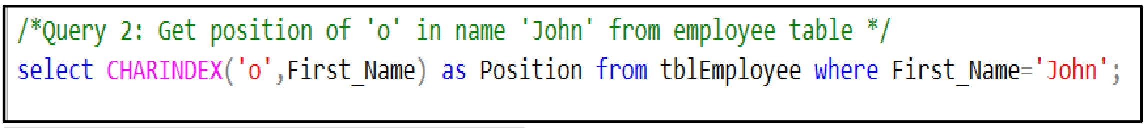


```
/*Query 1: Get First_Name from employee table using alias name “Employee Name” */
select First_Name as [Employee Name] from tblEmployee;
```

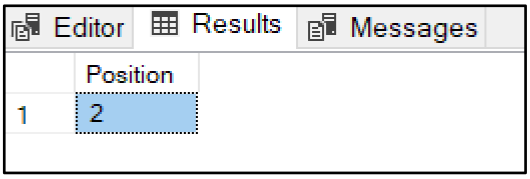


	Employee Name
1	John
2	Michael
3	Roy
4	Tom
5	Jerry
6	Philip
7	TestName1
8	TestName2

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

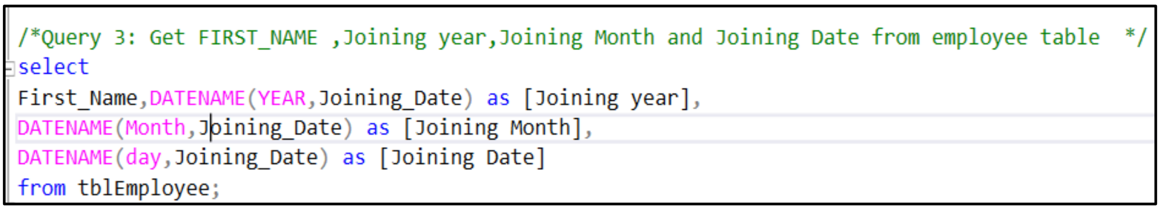


```
/*Query 2: Get position of 'o' in name 'John' from employee table */
select CHARINDEX('o',First_Name) as Position from tblEmployee where First_Name='John';
```

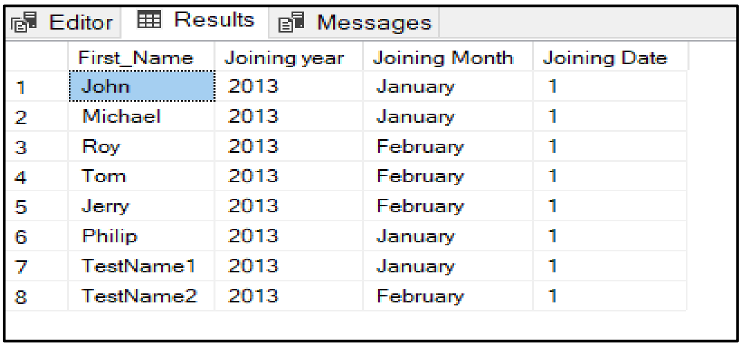


	Position
1	2

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

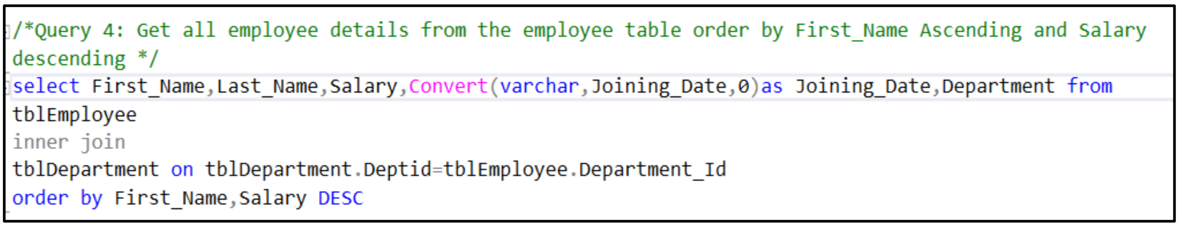


```
/*Query 3: Get FIRST_NAME ,Joining year,Joining Month and Joining Date from employee table  */
select
First_Name,DATENAME(YEAR,Joining_Date) as [Joining year],
DATENAME(Month,Joining_Date) as [Joining Month],
DATENAME(day,Joining_Date) as [Joining Date]
from tblEmployee;
```

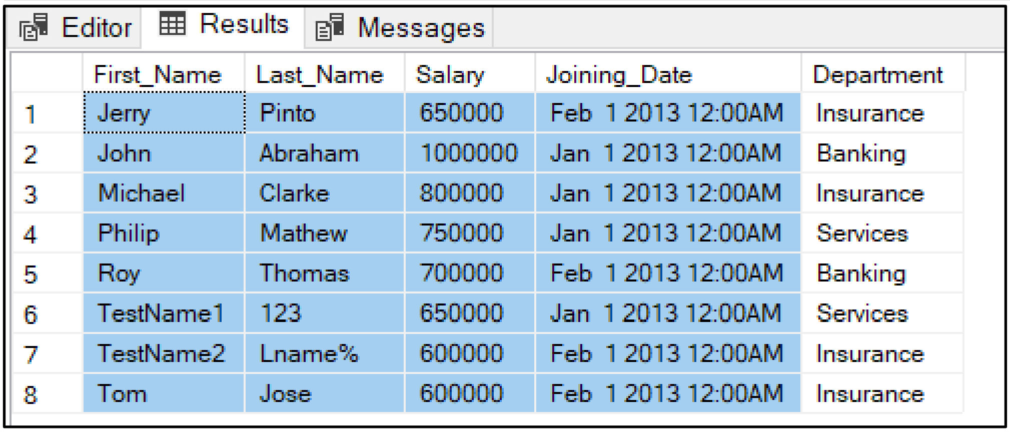


	First_Name	Joining year	Joining Month	Joining Date
1	John	2013	January	1
2	Michael	2013	January	1
3	Roy	2013	February	1
4	Tom	2013	February	1
5	Jerry	2013	February	1
6	Philip	2013	January	1
7	TestName1	2013	January	1
8	TestName2	2013	February	1

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

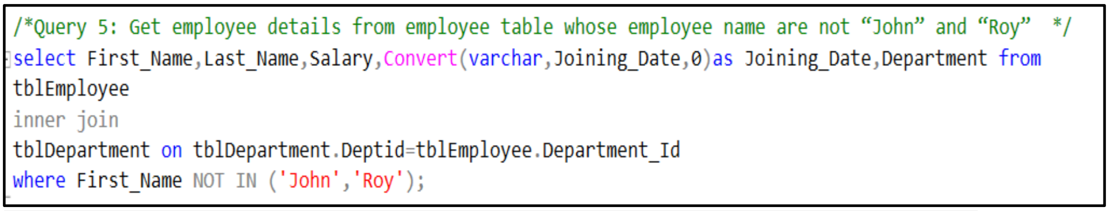


```
/*Query 4: Get all employee details from the employee table order by First_Name Ascending and Salary
descending */
select First_Name,Last_Name,Salary,Convert(varchar,Joining_Date,0)as Joining_Date,Department from
tblEmployee
inner join
tblDepartment on tblDepartment.Deptid=tblEmployee.Department_Id
order by First_Name,Salary DESC
```

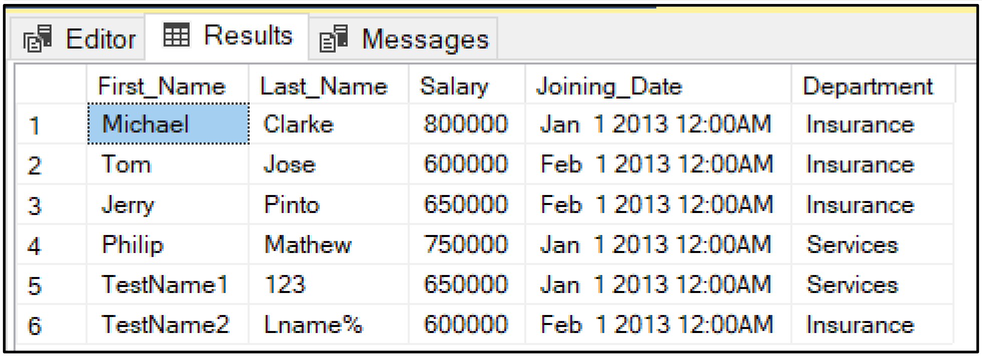


	First_Name	Last_Name	Salary	Joining_Date	Department
1	Jerry	Pinto	650000	Feb 1 2013 12:00AM	Insurance
2	John	Abraham	1000000	Jan 1 2013 12:00AM	Banking
3	Michael	Clarke	800000	Jan 1 2013 12:00AM	Insurance
4	Philip	Mathew	750000	Jan 1 2013 12:00AM	Services
5	Roy	Thomas	700000	Feb 1 2013 12:00AM	Banking
6	TestName1	123	650000	Jan 1 2013 12:00AM	Services
7	TestName2	Lname%	600000	Feb 1 2013 12:00AM	Insurance
8	Tom	Jose	600000	Feb 1 2013 12:00AM	Insurance

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

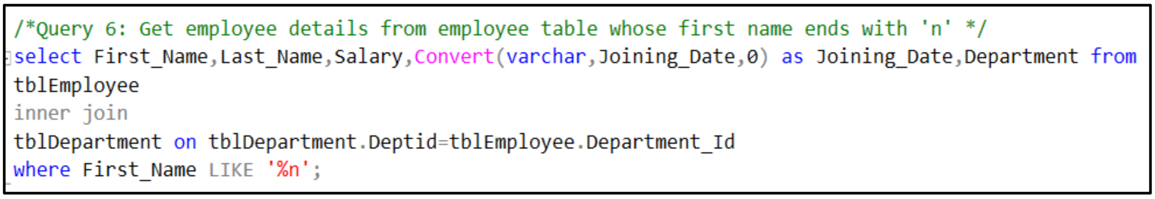


```
/*Query 5: Get employee details from employee table whose employee name are not “John” and “Roy”  */
select First_Name,Last_Name,Salary,Convert(varchar,Joining_Date,0)as Joining_Date,Department from
tblEmployee
inner join
tblDepartment on tblDepartment.Deptid=tblEmployee.Department_Id
where First_Name NOT IN ('John','Roy');
```

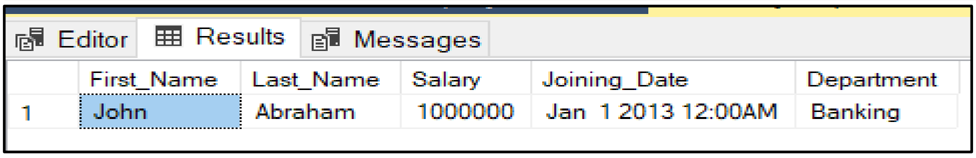


	First_Name	Last_Name	Salary	Joining_Date	Department
1	Michael	Clarke	800000	Jan 1 2013 12:00AM	Insurance
2	Tom	Jose	600000	Feb 1 2013 12:00AM	Insurance
3	Jerry	Pinto	650000	Feb 1 2013 12:00AM	Insurance
4	Philip	Mathew	750000	Jan 1 2013 12:00AM	Services
5	TestName1	123	650000	Jan 1 2013 12:00AM	Services
6	TestName2	Lname%	600000	Feb 1 2013 12:00AM	Insurance

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

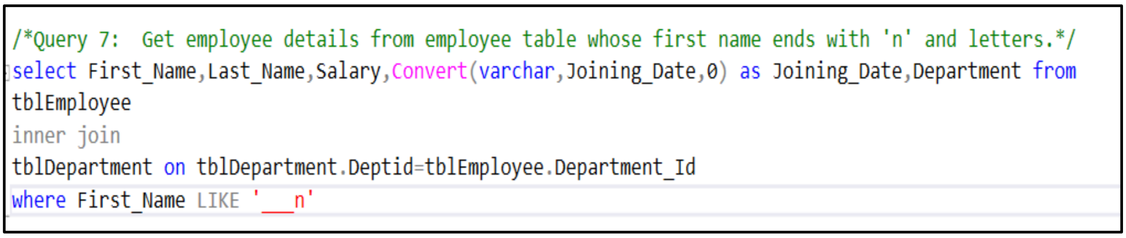


```
/*Query 6: Get employee details from employee table whose first name ends with 'n' */
select First_Name,Last_Name,Salary,Convert(varchar,Joining_Date,0) as Joining_Date,Department from
tblEmployee
inner join
tblDepartment on tblDepartment.Deptid=tblEmployee.Department_Id
where First_Name LIKE '%n';
```

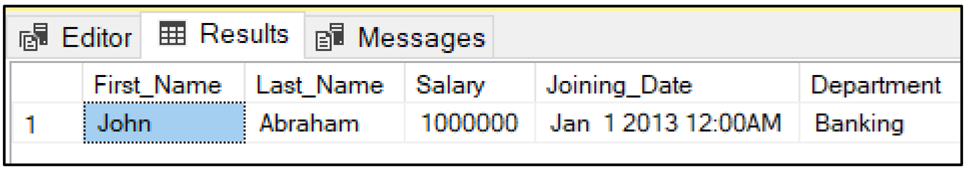


	First_Name	Last_Name	Salary	Joining_Date	Department
1	John	Abraham	1000000	Jan 1 2013 12:00AM	Banking

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

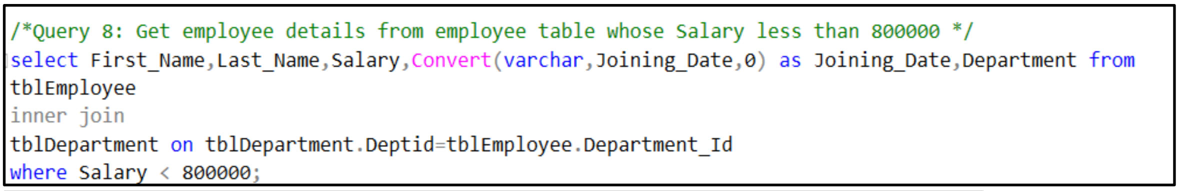


```
/*Query 7:  Get employee details from employee table whose first name ends with 'n' and letters.*/
select First_Name,Last_Name,Salary,Convert(varchar,Joining_Date,0) as Joining_Date,Department from
tblEmployee
inner join
tblDepartment on tblDepartment.Deptid=tblEmployee.Department_Id
where First_Name LIKE '___n'
```

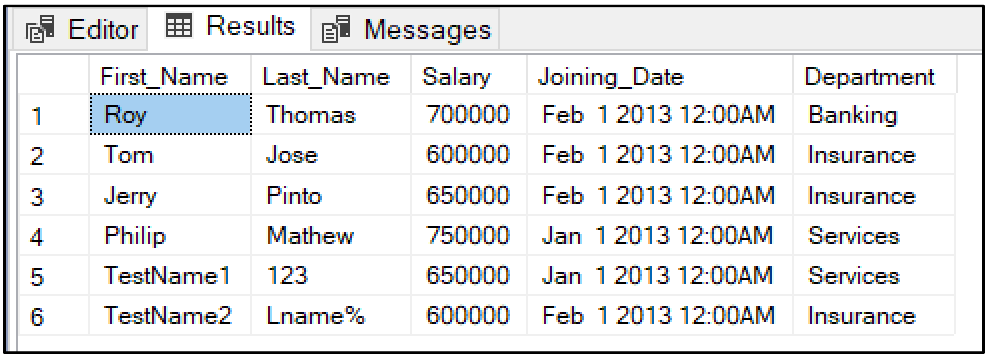


	First_Name	Last_Name	Salary	Joining_Date	Department
1	John	Abraham	1000000	Jan 1 2013 12:00AM	Banking

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

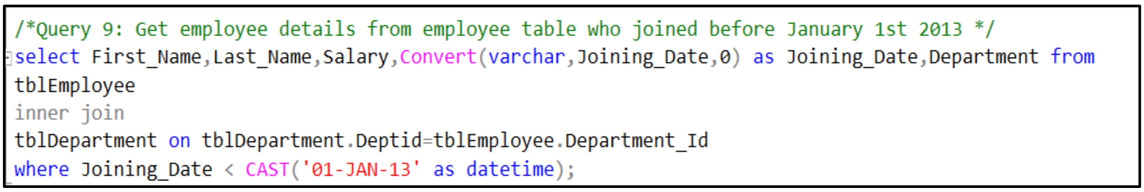


```
/*Query 8: Get employee details from employee table whose Salary less than 800000 */
select First_Name,Last_Name,Salary,Convert(varchar,Joining_Date,0) as Joining_Date,Department from
tblEmployee
inner join
tblDepartment on tblDepartment.Deptid=tblEmployee.Department_Id
where Salary < 800000;
```

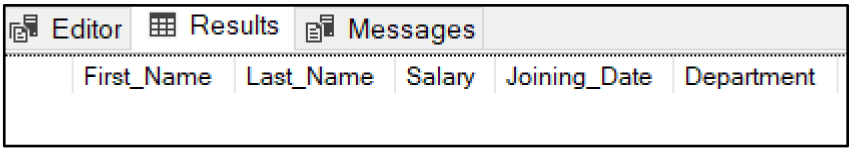


	First_Name	Last_Name	Salary	Joining_Date	Department
1	Roy	Thomas	700000	Feb 1 2013 12:00AM	Banking
2	Tom	Jose	600000	Feb 1 2013 12:00AM	Insurance
3	Jerry	Pinto	650000	Feb 1 2013 12:00AM	Insurance
4	Philip	Mathew	750000	Jan 1 2013 12:00AM	Services
5	TestName1	123	650000	Jan 1 2013 12:00AM	Services
6	TestName2	Lname%	600000	Feb 1 2013 12:00AM	Insurance

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

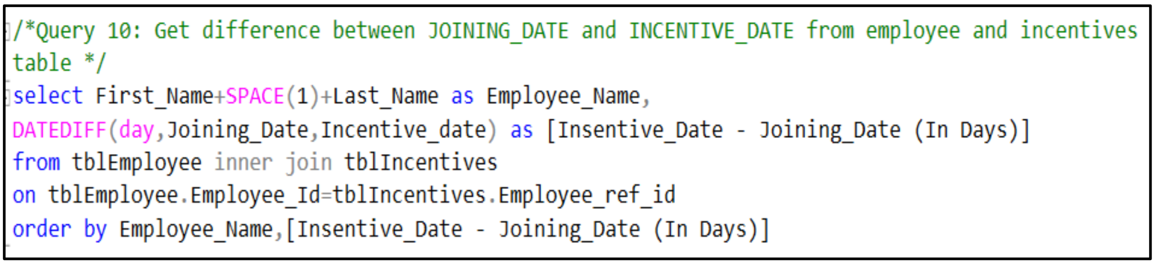


```
/*Query 9: Get employee details from employee table who joined before January 1st 2013 */
select First_Name,Last_Name,Salary,Convert(varchar,Joining_Date,0) as Joining_Date,Department from
tblEmployee
inner join
tblDepartment on tblDepartment.Deptid=tblEmployee.Department_Id
where Joining_Date < CAST('01-JAN-13' as datetime);
```

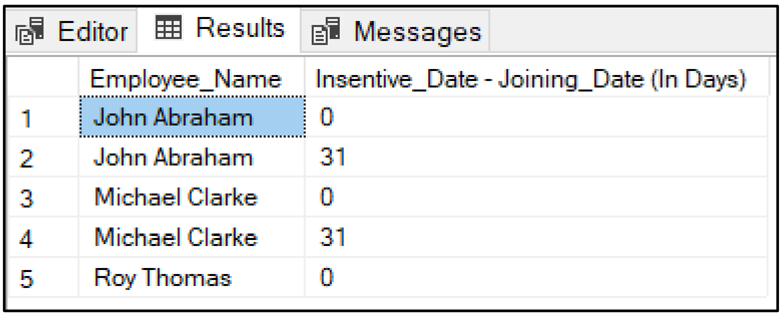


	First_Name	Last_Name	Salary	Joining_Date	Department

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

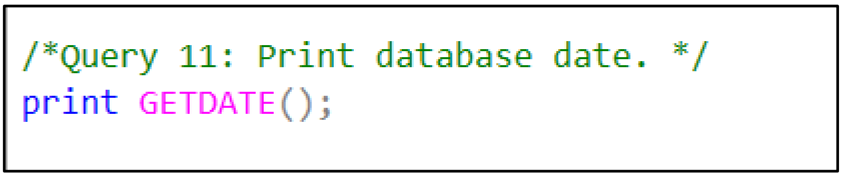


```
/*Query 10: Get difference between JOINING_DATE and INCENTIVE_DATE from employee and incentives
table */
select First_Name+SPACE(1)+Last_Name as Employee_Name,
DATEDIFF(day,Joining_Date,Incentive_date) as [Insentive_Date - Joining_Date (In Days)]
from tblEmployee inner join tblIncentives
on tblEmployee.Employee_Id=tblIncentives.Employee_ref_id
order by Employee_Name,[Insentive_Date - Joining_Date (In Days)]
```

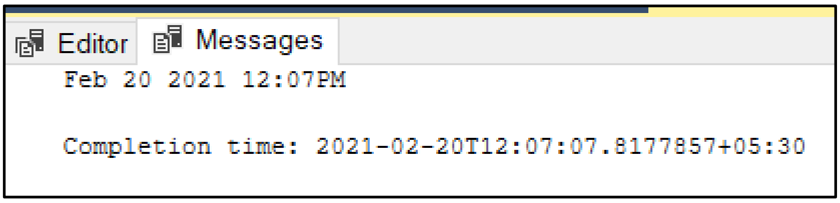


	Employee_Name	Insentive_Date - Joining_Date (In Days)
1	John Abraham	0
2	John Abraham	31
3	Michael Clarke	0
4	Michael Clarke	31
5	Roy Thomas	0

11. **Print database date.**

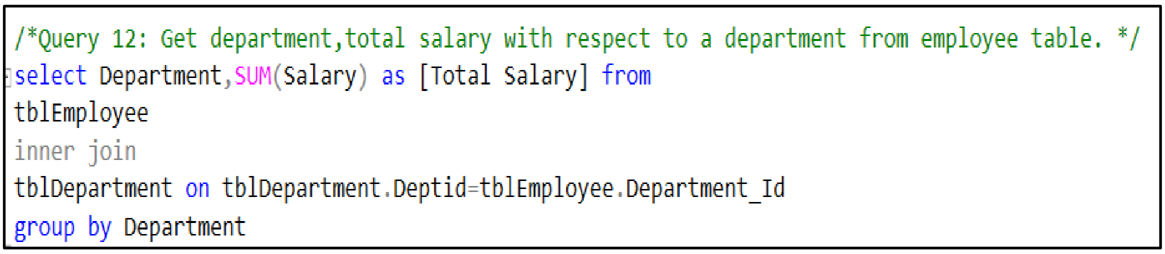


```
/*Query 11: Print database date. */
print GETDATE();
```

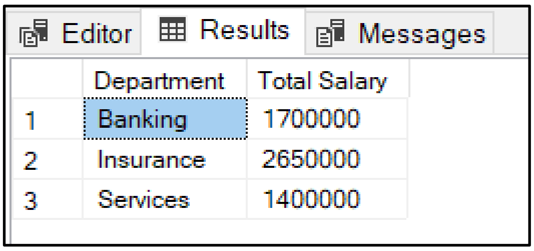


```
Feb 20 2021 12:07PM

Completion time: 2021-02-20T12:07:07.8177857+05:30
```

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

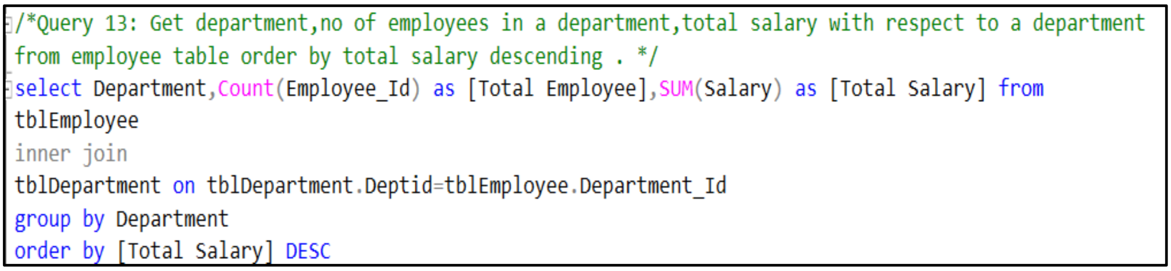


```
/*Query 12: Get department,total salary with respect to a department from employee table. */
select Department,SUM(Salary) as [Total Salary] from
tblEmployee
inner join
tblDepartment on tblDepartment.Deptid=tblEmployee.Department_Id
group by Department
```

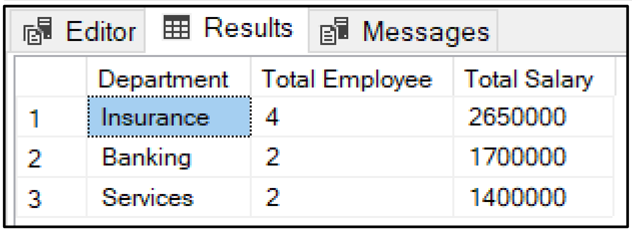


	Department	Total Salary
1	Banking	1700000
2	Insurance	2650000
3	Services	1400000

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

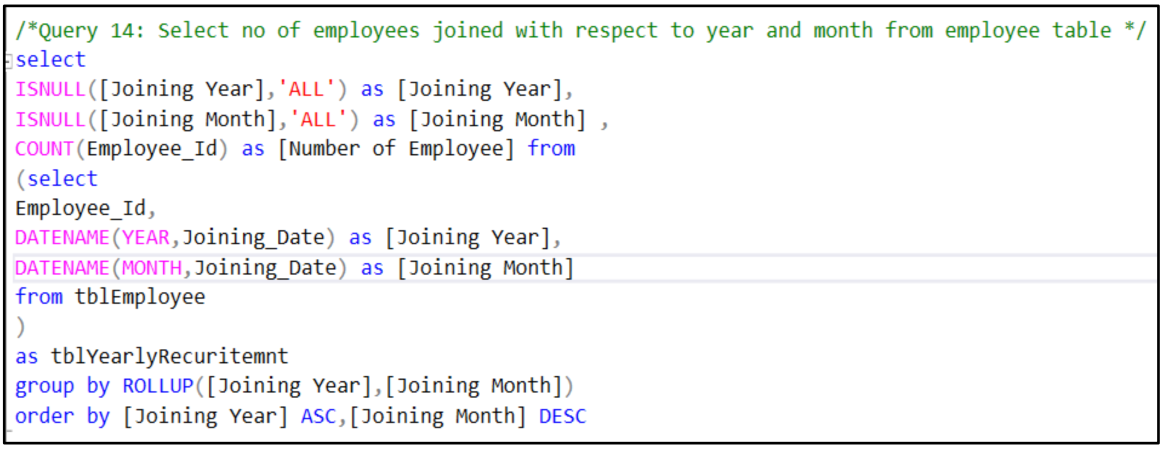


```
/*Query 13: Get department,no of employees in a department,total salary with respect to a department
from employee table order by total salary descending . */
select Department,Count(Employee_Id) as [Total Employee],SUM(Salary) as [Total Salary] from
tblEmployee
inner join
tblDepartment on tblDepartment.Deptid=tblEmployee.Department_Id
group by Department
order by [Total Salary] DESC
```

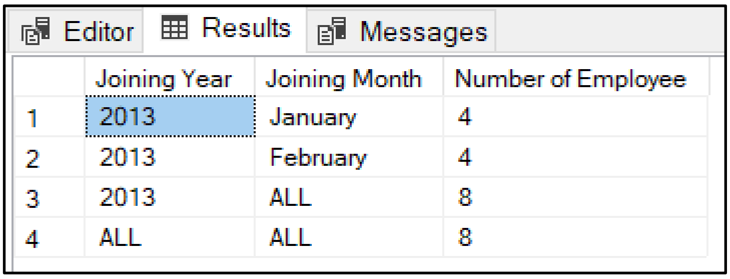


	Department	Total Employee	Total Salary
1	Insurance	4	2650000
2	Banking	2	1700000
3	Services	2	1400000

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

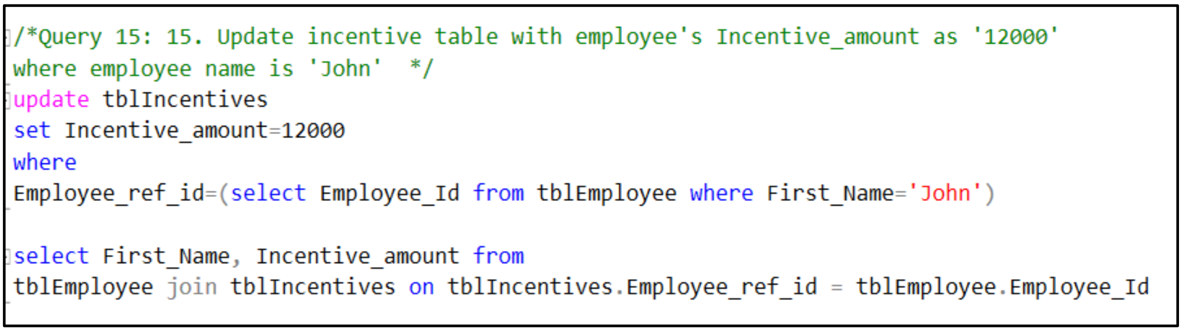


```
/*Query 14: Select no of employees joined with respect to year and month from employee table */
select
ISNULL([Joining Year],'ALL') as [Joining Year],
ISNULL([Joining Month],'ALL') as [Joining Month] ,
COUNT(Employee_Id) as [Number of Employee] from
(select
Employee_Id,
DATENAME(YEAR,Joining_Date) as [Joining Year],
DATENAME(MONTH,Joining_Date) as [Joining Month]
from tblEmployee
)
as tblYearlyRecuritemnt
group by ROLLUP([Joining Year],[Joining Month])
order by [Joining Year] ASC,[Joining Month] DESC
```

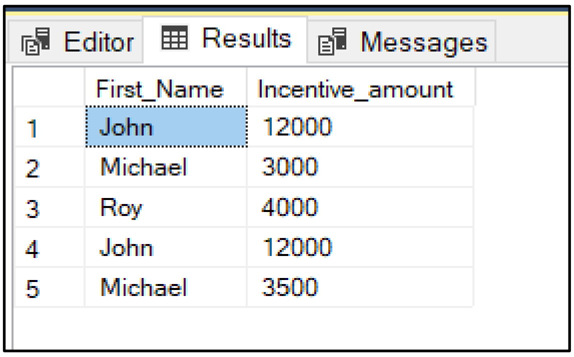


	Joining Year	Joining Month	Number of Employee
1	2013	January	4
2	2013	February	4
3	2013	ALL	8
4	ALL	ALL	8

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

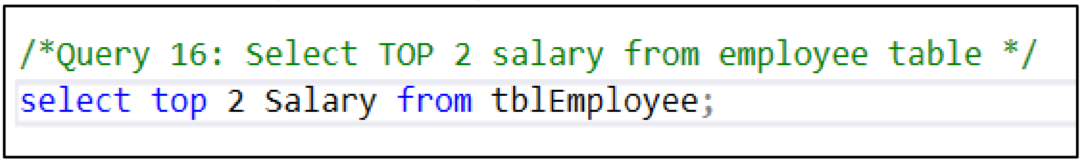


```
/*Query 15: 15. Update incentive table with employee's Incentive_amount as '12000'
where employee name is 'John'  */
update tblIncentives
set Incentive_amount=12000
where
Employee_ref_id=(select Employee_Id from tblEmployee where First_Name='John')

select First_Name, Incentive_amount from
tblEmployee join tblIncentives on tblIncentives.Employee_ref_id = tblEmployee.Employee_Id
```

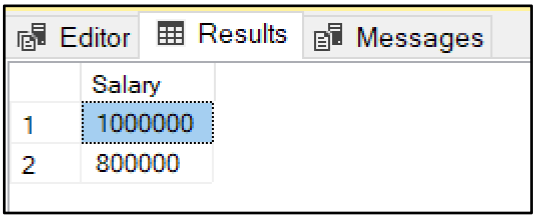


	First_Name	Incentive_amount
1	John	12000
2	Michael	3000
3	Roy	4000
4	John	12000
5	Michael	3500

16. Select TOP 2 salary from employee table

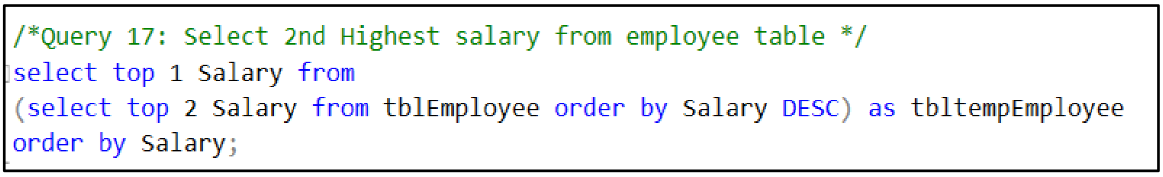


```
/*Query 16: Select TOP 2 salary from employee table */
select top 2 Salary from tblEmployee;
```

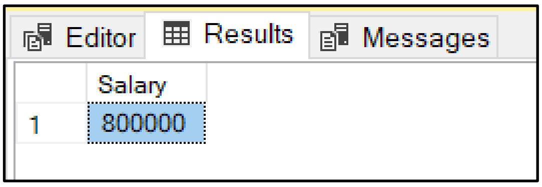


	Salary
1	1000000
2	800000

17. Select 2nd Highest salary from employee table



```
/*Query 17: Select 2nd Highest salary from employee table */
select top 1 Salary from
(select top 2 Salary from tblEmployee order by Salary DESC) as tbltempEmployee
order by Salary;
```



	Salary
1	800000

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

Ans:

1. Union All Not Remove the Duplication of Records
2. Whereas UNION First Remove Duplication and Then Display Data.

19. Write a syntax for CREATE Employee Table.

Ans:

Create table tblEmployee

(

EmployeeId int Identity(1,1) primary key,

First_Name varchar(50) not null,

```
    Last Name varchar(50) not null,
    Salary int not null,
    Joining_Date datetime not null,
    Department varchar(50),
    Constraint ck_Salary_tblEmployee CHECK(Salary >=0)
  );
```

Note: Here We Can Normalize the table By Create Department Table and then have to set reference of that to employee table instead of Department Manually.

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

Ans.

```
TRUNCATE table tblEmployee;
```

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

Ans.

```
Create proc sp_tblEmployee
@EmployeeId int =0
 As
 Begin
          Select * from tblEmployee where [Employee Id] = @ EmployeeId;
End
```

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

Ans.

```
Create function fc_HighestNumber(@number1 int, @number2 int ,@number3 int)
Returns int
As
Begin
Declare @highest int;
```

```
        If @number1 > @number2 AND @number1 > @number3
          Set @highest= @number1;
      Else if @number2 > @number3
        Set @highest= @number2;
    Else
       Set @highest= @number3;
  Return @highest;
 End
```

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

Ans:

1. If Employee Is Not Normalized by creating Separate Department table

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

2. If Employee Is Normalized by creating Separate Department table

```
Update tblEmployee
Set Salary=<new Salary>
Where DepartmentId = (select DeptId from tblDepartment where
Department = ‘Insurance’);
```

24. State the difference between varchar and nvarchar.

Ans:

1. varchar store data as 8-bit representation so it requires less size to store
2. nvarchar store data as Unicode Format (16-Bit Representation) so It Take More Amount of storage in compare of varchar.

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' }

Ans:

1. If Employee Is Not Normalized by creating Separate Department table

```
Insert into tblEmployee
(First_Name,Last_Name,Salary,Joining_Date,Department)

Values('Critiano', 'Ronaldo', 30000, '01-FEB-13 12:00:00 AM',
'Banking');
```

2. If Employee Is Normalized by creating Separate Department table

```
Insert into tblEmployee
(First_Name,Last_Name,Salary,Joining_Date,Department_Id)
Values('Critiano', 'Ronaldo', 30000,'01-FEB-13 12:00:00 AM',1);
```

Note: Here 1 is Department id from tblDepartment whose Department Value is Banking