Shubham Modi

shubhammodi1998.smgmail.com

GitHub Link:

<https://github.com/Shubham-Modi-98/cybercom/tree/asp/asp.net/code>

SQL Server Test

Cybercom Creation

* **Employee Table Create Syntax Insert Data into Employee Table: -**

Use [DatabaseDemo]

Create Table Employee

(

Employee\_id int PRIMARY KEY Idnetity(1,1),

First\_name nvarchar(50) NOT NULL,

Last\_name nvarchar(50) NOT NULL,

Salary int NOT NULL,

Joining\_date datetime NOT NULL,

Department nvarchar(50) NOT NULL

)

GO

insert into Employee values (1,'John','Abraham',1000000,'01-01-13 12:00:00 AM','Banking')

insert into Employee values (2,'Michael','Clarke',800000,'01-01-13 12:00:00 AM','Insurance')

insert into Employee values (3,'Roy','Thomas',700000,'01-02-13 12:00:00 AM','Banking')

insert into Employee values (4,'Tom','Jose',600000,'01-02-13 12:00:00 AM','Insurance')

insert into Employee values (5,'Jerry','Pinto',650000,'01-02-13 12:00:00 AM','Insurance')

insert into Employee values (6,'Philip','Mathew',750000,'01-01-13 12:00:00 AM','Services')

insert into Employee values (7,'TestName 1','123',650000,'01-01-13 12:00:00 AM','Services')

insert into Employee values (8,'TestName 2','Lname%',600000,'01-02-13 12:00:00 AM','Insurance')

* **Incentives Table Create Syntax and Insert Data into Incentives Table: -**

Create Table Incentive

(

Employee\_ref\_id int FOREIGN KEY REFERENCES Employee(Employee\_id),

Incentive\_date date NOT NULL,

Incentive\_amount int NOT NULL

)

insert into Incentives values (1,'01-02-13',5000)

insert into Incentives values (2,'01-02-13',3000)

insert into Incentives values (3,'01-02-13',4000)

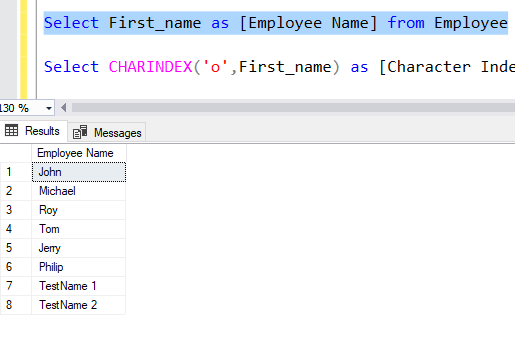
insert into Incentives values (1,'01-01-13',4500)

insert into Incentives values (2,'01-01-13',3500)

## **Get First\_Name from employee table using alias name “Employee Name”**

## **Ans.**

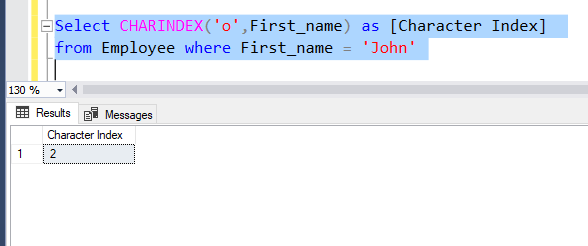
## Select First\_name as [Employee Name] from Employee



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

**Ans.**

Select CHARINDEX('o',First\_name) as [Character Index] from Employee where First\_name = 'John'

****

1. **Get FIRST\_NAME ,Joining year,Joining Month and Joining Date from employee table**

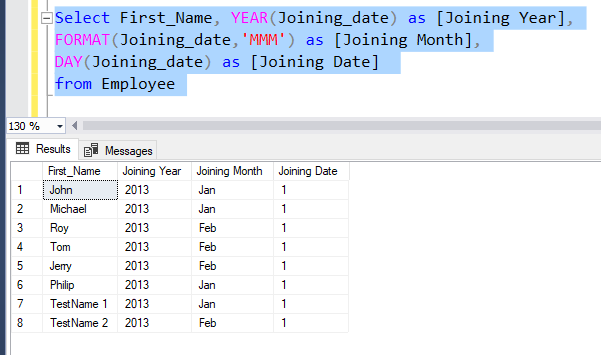
**Ans.**

Select First\_Name, YEAR(Joining\_date) as [Joining Year],

FORMAT(Joining\_date,'MMM') as [Joining Month],

DAY(Joining\_date) as [Joining Date]

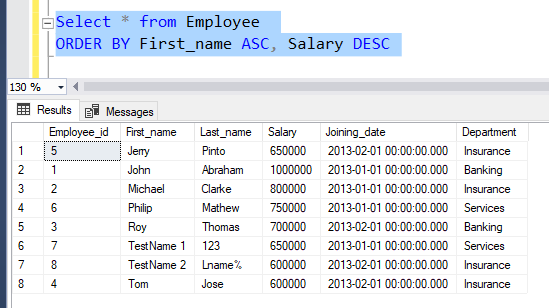
from Employee



1. **Get all employee details from the employee table order by First\_Name Ascending and Salary descending**

**Ans.**

Select Employee\_id, First\_name, Last\_name, Salary, Joining\_date, Department from Employee ORDER BY First\_name ASC, Salary DESC

****

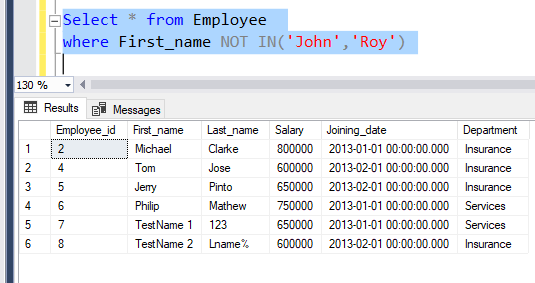
1. **Get employee details from employee table whose employee name are not “John” and “Roy”**

**Ans.**

Select \* from Employee where First\_name NOT IN('John','Roy')

OR used Sub-Query

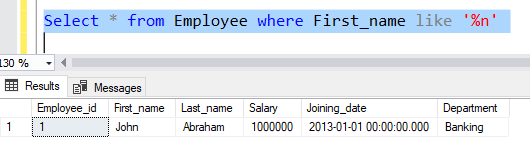
Select \* from Employee where First\_name Not In(select First\_name from Employee where First\_name like 'John%' or First\_name like 'Roy%')



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

**Ans.**

Select \* from Employee where First\_name like '%n'

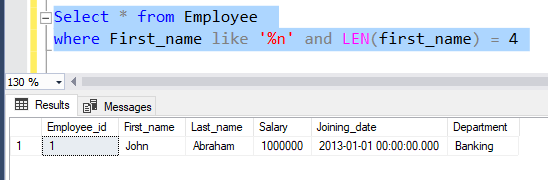


1. **Get employee details from employee table whose first name ends with 'n' and name contains 4 letters**

**Ans.**

Select \* from Employee where First\_name like '%n'

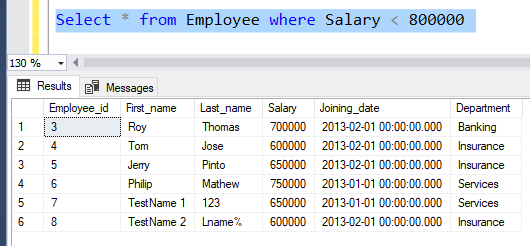
and LEN(first\_name) = 4



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

**Ans.**

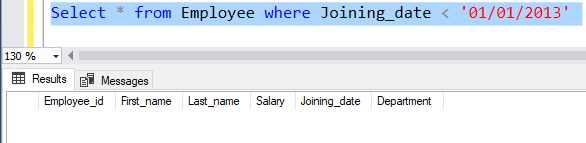
Select \* from Employee where Salary < 800000



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

**Ans.**

Select \* from Employee where Joining\_date < '01/01/2013'

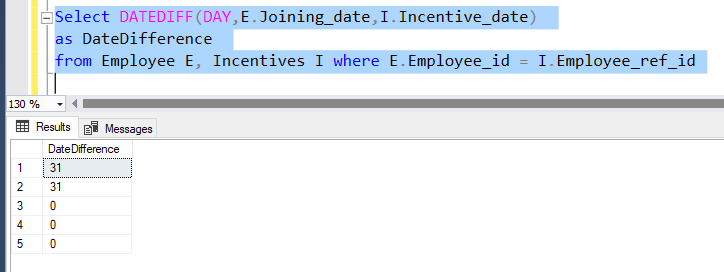
****

1. **Get difference between JOINING\_DATE and INCENTIVE\_DATE from employee and incentives table**

**Ans.**

Select DATEDIFF(DAY,E.Joining\_date,I.Incentive\_date) as DateDifference

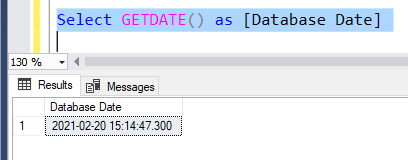
from Employee E, Incentives I where E.Employee\_id = I.Employee\_ref\_id



1. **Print database date.**

**Ans.**

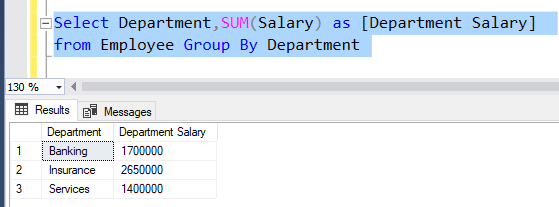
Select GETDATE() as [Database Date]

****

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

**Ans.**

Select Department,SUM(Salary) as [Department Salary] from Employee Group By Department

****

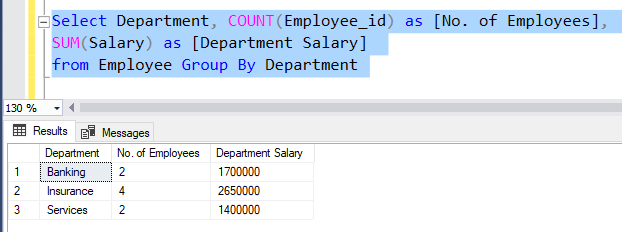
1. **Get department,no of employees in a department,total salary with respect to a department from employee table order by total salary descending .**

**Ans.**

Select Department, COUNT(Employee\_id) as [No. of Employees],

SUM(Salary) as [Department Salary]

from Employee Group By Department



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

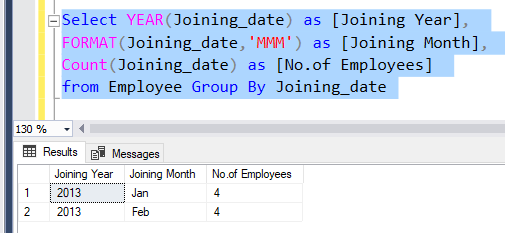
**Ans.**

Select YEAR(Joining\_date) as [Joining Year],

FORMAT(Joining\_date,'MMM') as [Joining Month],

Count(Joining\_date) as [No.of Employees]

from Employee Group By Joining\_date



1. **Update incentive table with employee's Incentive\_amount as '12000' where employee name is 'John'**

**Ans.**

Update Incentives set Incentive\_amount = 1200

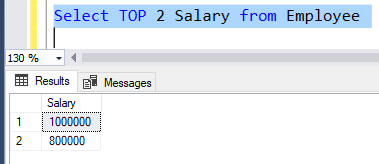
From Employee E, Incentives I

where E.Employee\_id = I.Employee\_ref\_id and E.First\_name = 'John'

1. **Select TOP 2 salary from employee table**

**Ans.**

Select TOP 2 Salary from Employee



1. **Select 2nd Highest salary from employee table**

**Ans.**

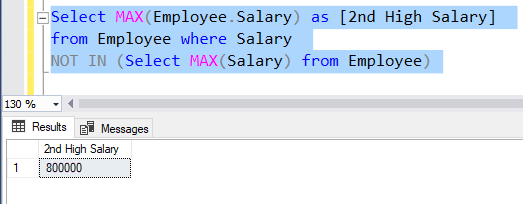
Select MAX(Employee.Salary) as [2nd High Salary]

from Employee where Salary < (Select MAX(Salary) from Employee)

OR YOU CAN USED

Select MAX(Employee.Salary) as [2nd High Salary]

from Employee where Salary NOT IN (Select MAX(Salary) from Employee)



1. **Write. What is the difference between UNION and UNION ALL ?**

**Ans.**

Union: - It first perform sorting and displays all records from Two tables and removes all duplicate records.

Union All: - It keeps all the records from more than one table and It doesn’t remove duplicate records.

1. **Write a syntax for CREATE Employee Table.**

**Ans.**

Use [DatabaseDemo]

Create Table Employee

(

Employee\_id int PRIMARY KEY Idnetity(1,1),

First\_name nvarchar(50) NOT NULL,

Last\_name nvarchar(50) NOT NULL,

Salary int NOT NULL,

Joining\_date datetime NOT NULL,

Department nvarchar(50) NOT NULL

)

GO

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

**Ans.**

Truncate Table Employee

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

**Ans.**

Create Procedure spGetEmployeeById

@id int

AS

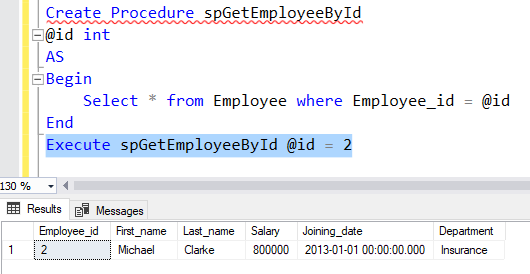
Begin

select \* from Employee where Employee\_id = @id

End

To call Procedure

Execute spGetEmployeeById @id = 2



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

**Ans.**

Create Function fnMaxNumber(@no1 int,@no2 int,@no3 int)

returns int AS

Begin

Declare @maxVal as int

If(@no1 > @no2 and @no1 > @no3)

set @maxVal = @no1

Else if (@no2 > @no1 and @no2 > @no3)

set @maxVal = @no2

ELSE

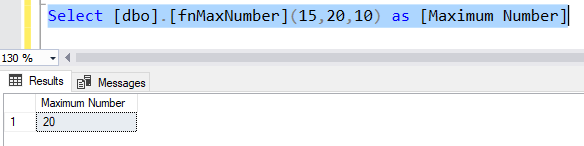
set @maxVal = @no3

return @maxVal

End

Command to Call function

Select [dbo].[fnMaxNumber](15,20,10) as [Maximum Number]



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

**Ans.**

Update Employee set Salary = new\_salary

where Department = 'Insurence'

1. **State the difference between varchar and nvarchar**

**Ans.**

varchar stored data in regular 8-bit format. So, it can hold less data as well as storage

nvarchar stored data in double of varchar as 16-bit data. So, it can be stored double data and occupied double storage in compare to varchar.

1. **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.**

Insert into Employee (First\_name,Last\_name,Salary,Joining\_date,Department)

values ('Cristiano','Ronaldo',30000,'02-01-2013 12:00:00: AM','Banking')

