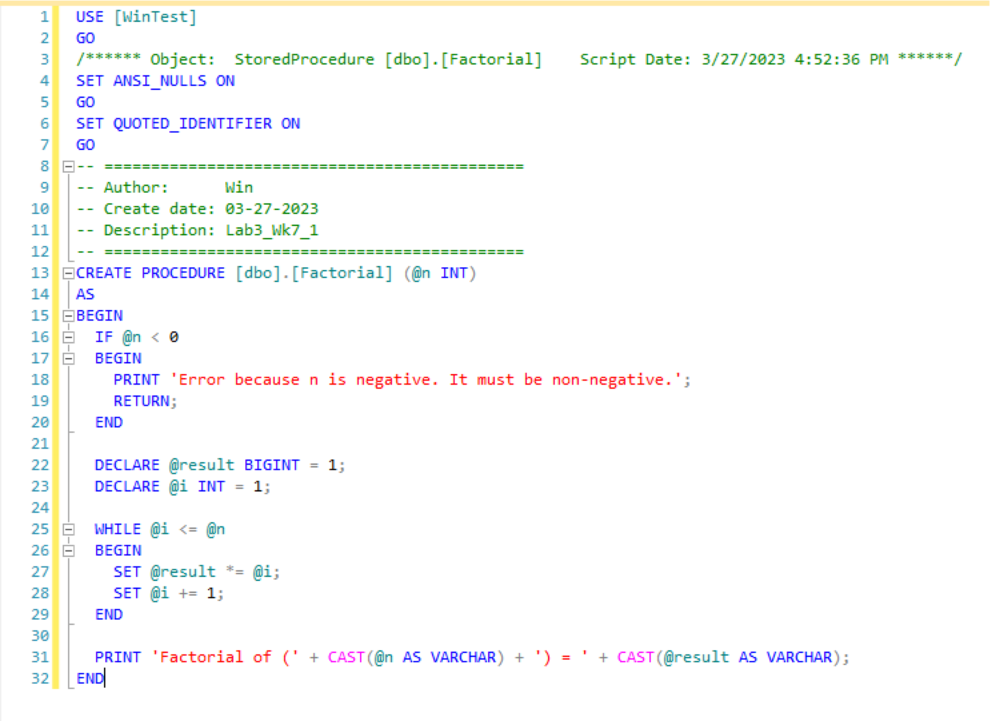
Database Management System – cs422 DE

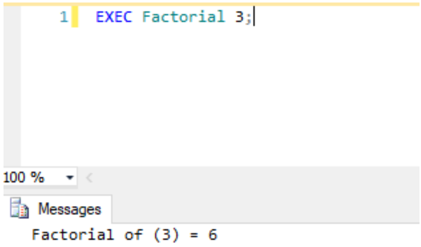
Lab 3 – Week 7

---

1) [3] Write and execute a T-SQL stored procedure *Factorial*(*n*), which computes and outputs the factorial of the input parameter *n*. If *n* is negative, then the procedure prints an error message.
Attach the screenshots of the output and the command which you used to execute the SP.
ANS:

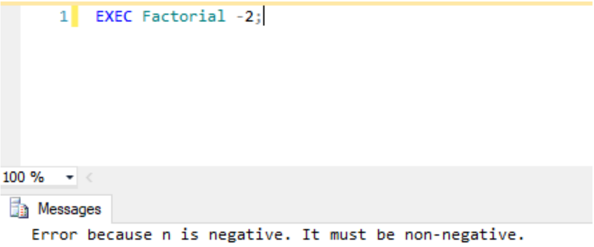


```sql
USE [WinTest]
GO
/****** Object:  StoredProcedure [dbo].[Factorial]    Script Date: 3/27/2023 4:52:36 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:      Win
-- Create date: 03-27-2023
-- Description: Lab3_Wk7_1
-- =============================================
CREATE PROCEDURE [dbo].[Factorial] (@n INT)
AS
BEGIN
  IF @n < 0
  BEGIN
    PRINT 'Error because n is negative. It must be non-negative.';
    RETURN;
  END

  DECLARE @result BIGINT = 1;
  DECLARE @i INT = 1;

  WHILE @i <= @n
  BEGIN
    SET @result *= @i;
    SET @i += 1;
  END

  PRINT 'Factorial of (' + CAST(@n AS VARCHAR) + ') = ' + CAST(@result AS VARCHAR);
END
```



```sql
EXEC Factorial 3;
```

```
Factorial of (3) = 6
```



```sql
EXEC Factorial -2;
```

```
Error because n is negative. It must be non-negative.
```

2) [7] Create a Table *Employee* with the fields: social security no. (primary key), name, position, no. of dependents, annual salary.

Write and execute a T-SQL procedure *Compute_Tax* to do the following:

- Create a new table *Tax* with fields: social security no., income tax.
- Fill the table *Tax* with data by computing the income tax for each person in the Employee Table.

The income tax is computed from the annual salary S and the number of dependents D.

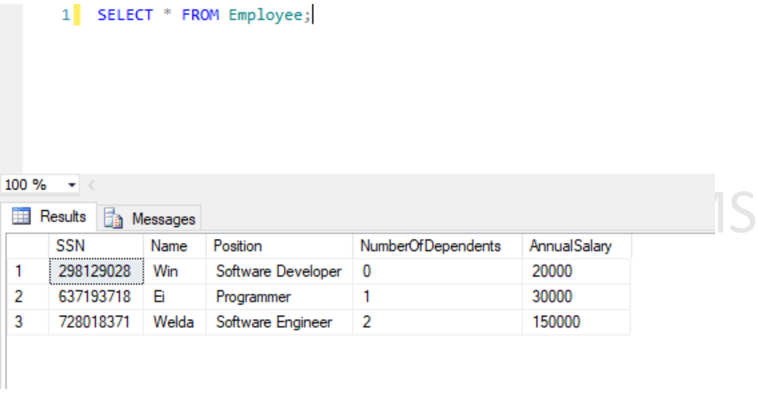
Net Salary: S - (7000 + D*950)

Tax Computed as follows:

- 10% of the first 15,000 of net salary;
- plus 15% of the next 15,000 of net salary;
- plus 28% of any net salary over 30,000.

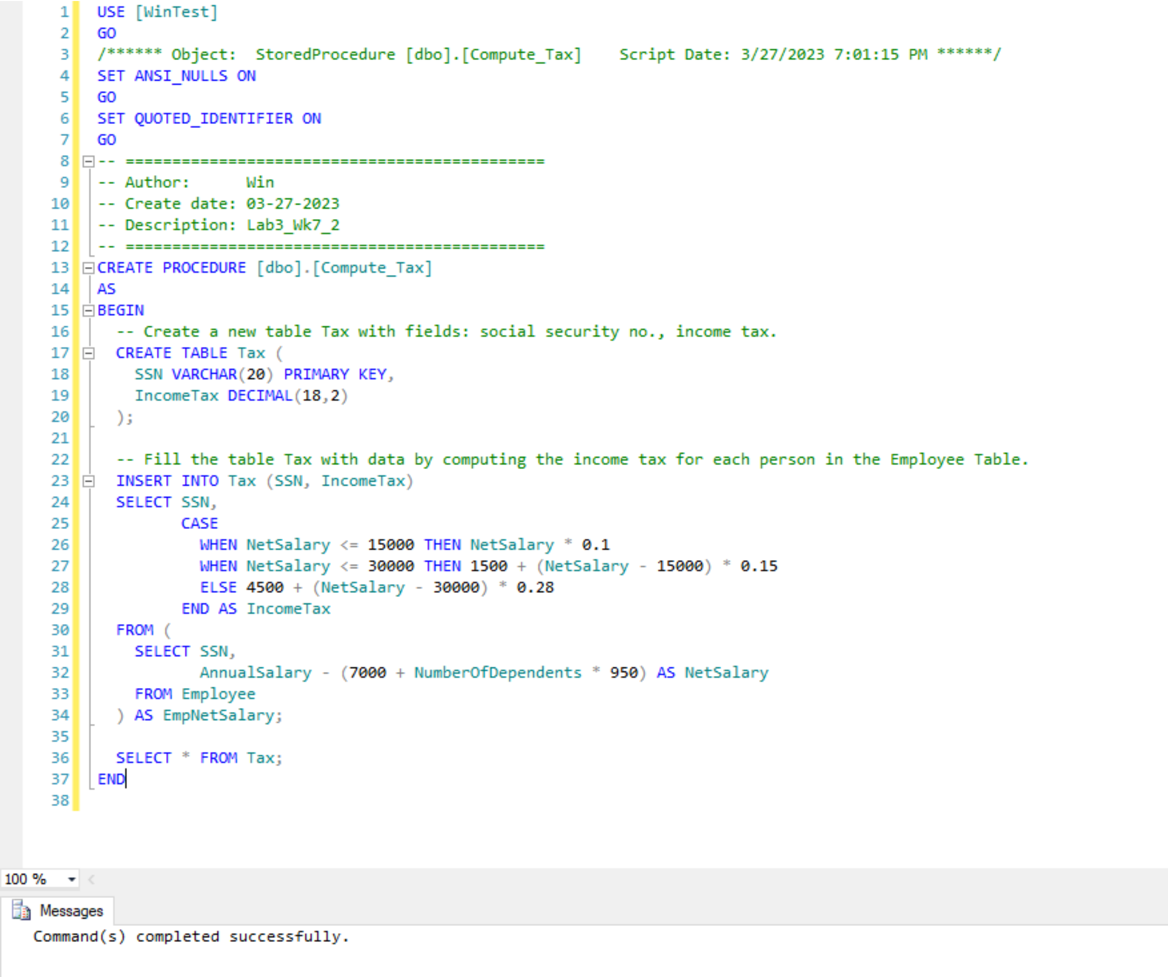
For getting full credit for this problem, you need to show me the complete code for the *Compute_Tax* SP. Also attach the screenshots of the *Employee* and the new *Tax* table.

ANS:



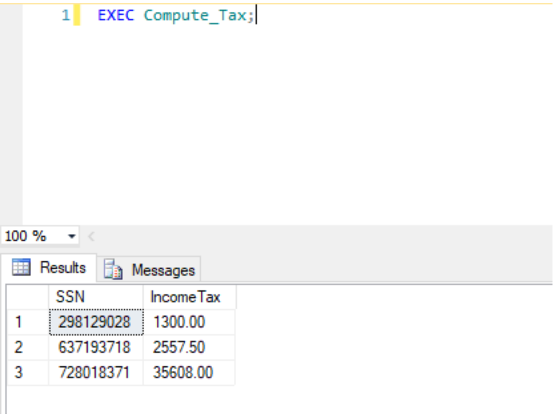
```
SELECT * FROM Employee;
```

	SSN	Name	Position	NumberOfDependents	AnnualSalary
1	298129028	Win	Software Developer	0	20000
2	637193718	Ei	Programmer	1	30000
3	728018371	Welda	Software Engineer	2	150000



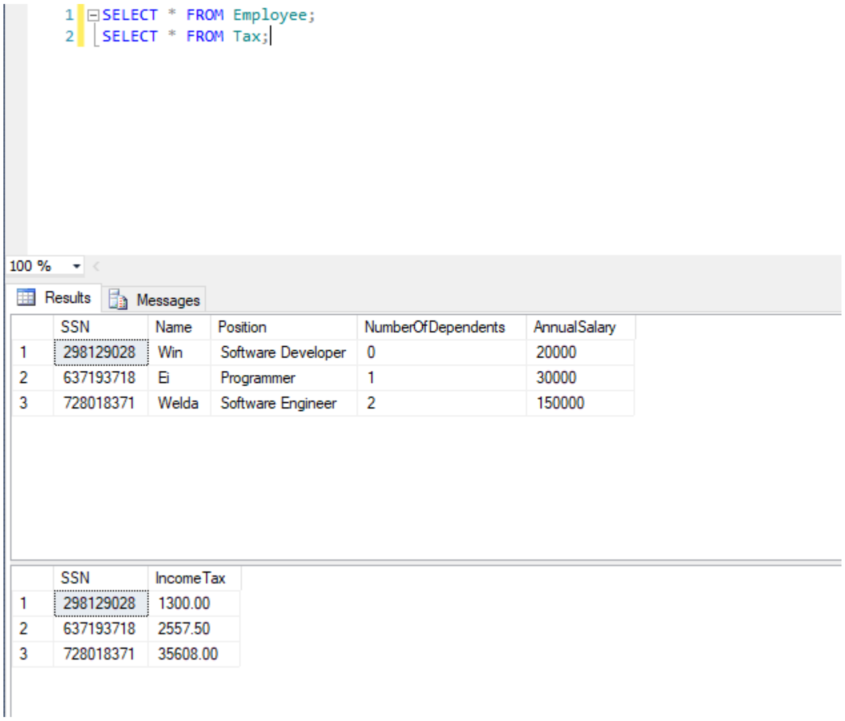
```sql
USE [WinTest]
GO
/****** Object:  StoredProcedure [dbo].[Compute_Tax]    Script Date: 3/27/2023 7:01:15 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:      Win
-- Create date: 03-27-2023
-- Description: Lab3_Wk7_2
-- =============================================
CREATE PROCEDURE [dbo].[Compute_Tax]
AS
BEGIN
  -- Create a new table Tax with fields: social security no., income tax.
  CREATE TABLE Tax (
    SSN VARCHAR(20) PRIMARY KEY,
    IncomeTax DECIMAL(18,2)
  );

  -- Fill the table Tax with data by computing the income tax for each person in the Employee Table.
  INSERT INTO Tax (SSN, IncomeTax)
  SELECT SSN,
         CASE
           WHEN NetSalary <= 15000 THEN NetSalary * 0.1
           WHEN NetSalary <= 30000 THEN 1500 + (NetSalary - 15000) * 0.15
           ELSE 4500 + (NetSalary - 30000) * 0.28
         END AS IncomeTax
  FROM (
    SELECT SSN,
           AnnualSalary - (7000 + NumberOfDependents * 950) AS NetSalary
    FROM Employee
  ) AS EmpNetSalary;

  SELECT * FROM Tax;
END
```

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



```sql
EXEC Compute_Tax;
```

	SSN	IncomeTax
1	298129028	1300.00
2	637193718	2557.50
3	728018371	35608.00



```
SELECT * FROM Employee;
SELECT * FROM Tax;
```

	SSN	Name	Position	NumberOfDependents	AnnualSalary
1	298129028	Win	Software Developer	0	20000
2	637193718	Ei	Programmer	1	30000
3	728018371	Welda	Software Engineer	2	150000

	SSN	IncomeTax
1	298129028	1300.00
2	637193718	2557.50
3	728018371	35608.00