Database Management System – cs422 DE

Lab 3 – Week 7

This Lab is based on Transact-SQL.

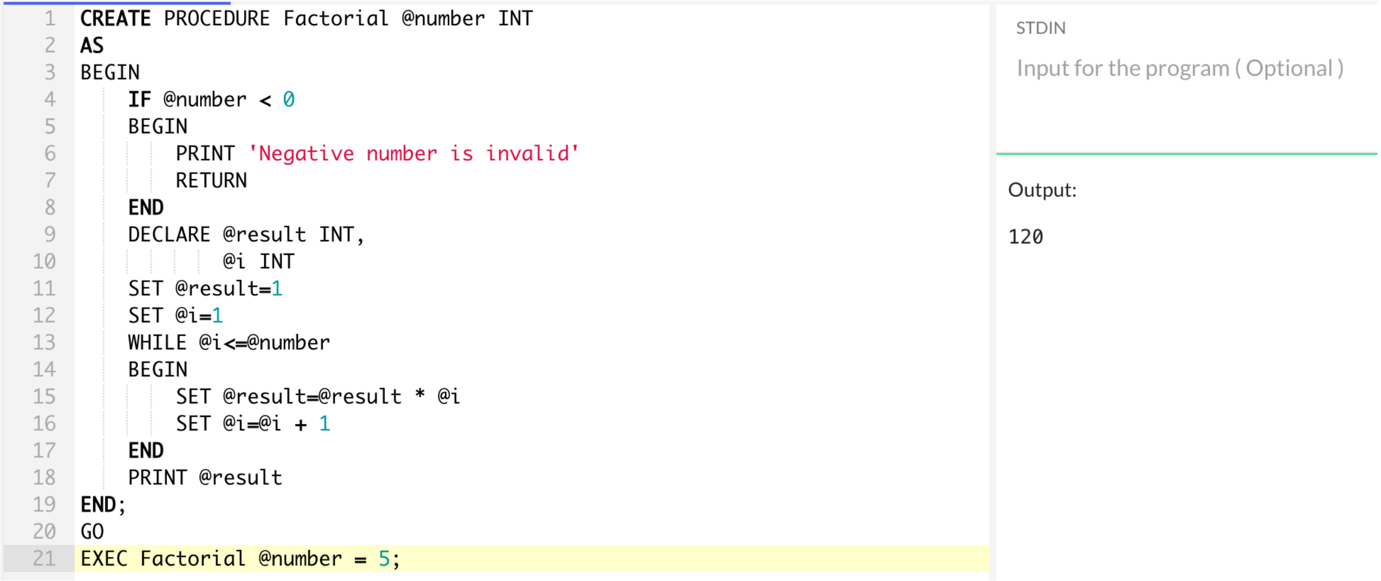
- Submit your *own work* on time. No credit will be given if the lab is submitted after the due date.
- Note that the completed lab should be submitted in .doc, .docx, .rtf, .pdf or .zip format only.

----

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:

```
CREATE PROCEDURE Factorial @number INT
AS
BEGIN
    IF @number < 0
    BEGIN
        PRINT 'Negative number is invalid'
        RETURN
    END
    DECLARE @result INT,
            @i INT
    SET @result=1
    SET @i=1
    WHILE @i<=@number
    BEGIN
        SET @result=@result * @i
        SET @i=@i + 1
    END
    PRINT @result
END;
GO
EXEC Factorial @number = 5;
```



```
CREATE PROCEDURE Factorial @number INT
AS
BEGIN
    IF @number < 0
    BEGIN
        PRINT 'Negative number is invalid'
        RETURN
    END
    DECLARE @result INT,
            @i INT
    SET @result=1
    SET @i=1
    WHILE @i<=@number
    BEGIN
        SET @result=@result * @i
        SET @i=@i + 1
    END
    PRINT @result
END;
GO
EXEC Factorial @number = 5;
```

Output:

```
120
```

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:

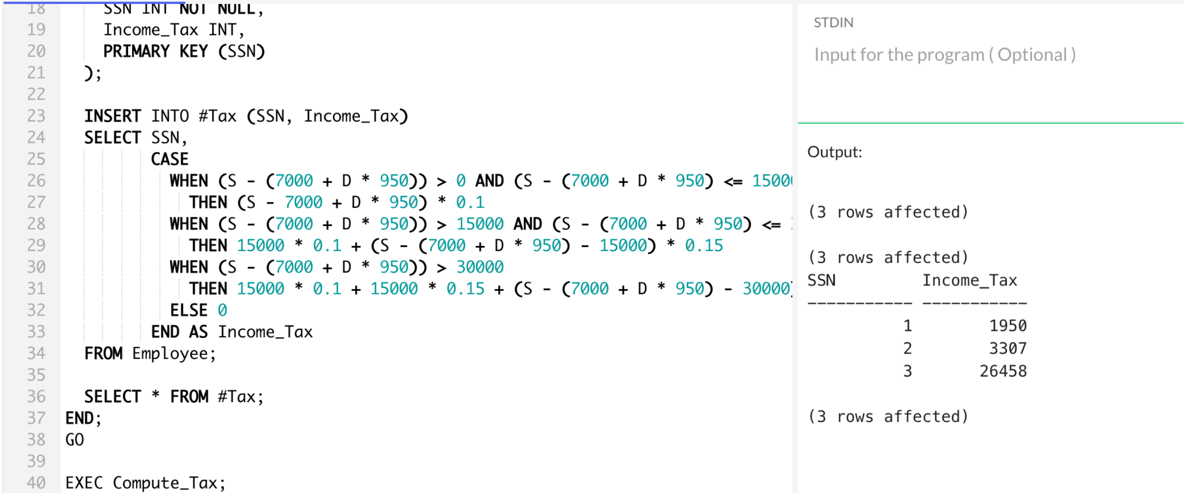
```
CREATE TABLE Employee (
    SSN INT PRIMARY KEY,
    S INT,
    D INT
);

INSERT INTO Employee (SSN, S, D) VALUES
(1, 25000, 0),
(2, 35000, 1),
(3, 120000, 2);
Go

CREATE PROCEDURE Compute_Tax
AS
BEGIN
  CREATE TABLE #Tax (
    SSN INT NOT NULL,
    Income_Tax INT,
    PRIMARY KEY (SSN)
  );

  INSERT INTO #Tax (SSN, Income_Tax)
  SELECT SSN,
         CASE
           WHEN (S - (7000 + D * 950)) > 0 AND (S - (7000 + D * 950) <= 15000)
             THEN (S - 7000 + D * 950) * 0.1
           WHEN (S - (7000 + D * 950)) > 15000 AND (S - (7000 + D * 950) <= 30000)
             THEN 15000 * 0.1 + (S - (7000 + D * 950) - 15000) * 0.15
           WHEN (S - (7000 + D * 950)) > 30000
             THEN 15000 * 0.1 + 15000 * 0.15 + (S - (7000 + D * 950) - 30000) * 0.28
           ELSE 0
         END AS Income_Tax
  FROM Employee;
  SELECT * FROM #Tax;
END;
GO

EXEC Compute_Tax;
```



```
    SSN INT NOT NULL,
    Income_Tax INT,
    PRIMARY KEY (SSN)
  );

  INSERT INTO #Tax (SSN, Income_Tax)
  SELECT SSN,
         CASE
           WHEN (S - (7000 + D * 950)) > 0 AND (S - (7000 + D * 950) <= 1500
             THEN (S - 7000 + D * 950) * 0.1
           WHEN (S - (7000 + D * 950)) > 15000 AND (S - (7000 + D * 950) <=
             THEN 15000 * 0.1 + (S - (7000 + D * 950) - 15000) * 0.15
           WHEN (S - (7000 + D * 950)) > 30000
             THEN 15000 * 0.1 + 15000 * 0.15 + (S - (7000 + D * 950) - 30000)
           ELSE 0
         END AS Income_Tax
  FROM Employee;

  SELECT * FROM #Tax;
END;
GO

EXEC Compute_Tax;
```

Output:

```
(3 rows affected)

(3 rows affected)
SSN         Income_Tax
----------- -----------
          1        1950
          2        3307
          3       26458

(3 rows affected)
```