Database Management System – cs422 DE

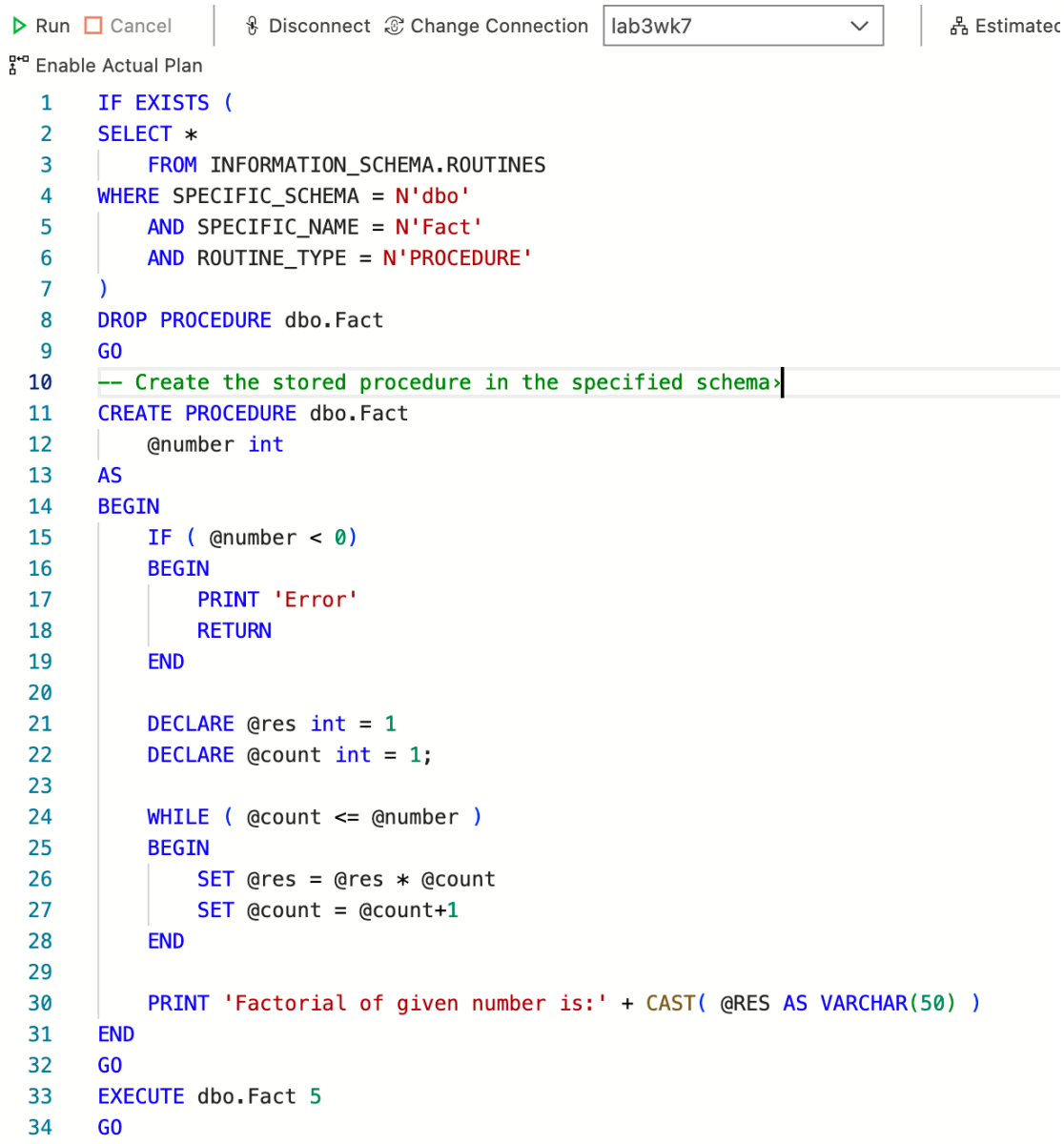
Lab 3 – Week 7

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:



Messages

```
5:58:38 PM    Started executing query at Line 1
              Commands completed successfully.
5:58:38 PM    Started executing query at Line 10
              Commands completed successfully.
5:58:38 PM    Started executing query at Line 33
              Factorial of given number is:120
              Total execution time: 00:00:00.019
```

Output

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:

Results Messages

	social	incomeTax
1	012-34-5678	20018
2	123-45-6789	25660.000000000004
3	234-56-7890	11548
4	345-67-8901	132858
5	456-78-9012	15062.000000000002
6	567-89-0123	11198
7	678-90-1234	11086
8	789-01-2345	18870
9	890-12-3456	209662.00000000003
1…	901-23-4567	405

Tax Table

// Here is the code for Compute_Tax Stored Procedure

```
-- Create the stored procedure in the specified schema
CREATE PROCEDURE dbo.Compute_Tax
AS
BEGIN
-- add more stored procedure parameters here
    DECLARE @social VARCHAR(50)
    DECLARE @dcount INT
    DECLARE @salary FLOAT
    DECLARE employee_data CURSOR FOR SELECT social,dependentCount,annualSalary FROM
Employee;

    DROP TABLE IF EXISTS Tax;
    CREATE TABLE Tax (social VARCHAR(50), incomeTax FLOAT);


    OPEN employee_data
    FETCH NEXT FROM employee_data INTO @social, @dcount, @salary
    WHILE @@FETCH_STATUS = 0
    BEGIN
    -- body of the stored procedure
        DECLARE @net_salary FLOAT =  @salary-(7000+@dcount*950);
        DECLARE @incomTax FLOAT
        IF( @net_salary <= 15000 )
            BEGIN
                SET @incomTax = @net_salary * 0.1;
            END
        ELSE IF ( @net_salary > 15000 AND @net_salary <= 30000  )
            BEGIN
                SET @incomTax = 15000 * 0.1 + ((@net_salary - 15000) * 0.15)
            END
        ELSE
            BEGIN
                SET @incomTax = 15000 * 0.1 + (15000 * 0.15) + ((@net_salary - 30000)
* 0.28)
            END
        INSERT INTO Tax ( social, incomeTax) VALUES ( @social, @incomTax )
        FETCH NEXT FROM employee_data INTO @social, @dcount, @salary
    END
    CLOSE employee_data
    DEALLOCATE employee_data
END
GO
-- example to execute the stored procedure we just created
EXECUTE dbo.Compute_Tax
GO
```

Results Messages

	social	name	position	dependentCount	annualSalary
1	012-34-5678	Kim Grace	Developer	2	97000
2	123-45-6789	John Jane	Manager	5	120000
3	234-56-7890	Smith Doe	Assistant Manager	7	71500
4	345-67-8901	Sara Johnson	Team Leader	2	500000
5	456-78-9012	Bob Lee	Developer	8	85000
6	567-89-0123	Alice Jones	Developer	12	75000
7	678-90-1234	Tom Johnson	Team Leader	4	67000
8	789-01-2345	Lee Emily	Developer	0	91000
9	890-12-3456	David Rodriguez	Assistant Manager	8	780000
1…	901-23-4567	Frank Chen	Manager	1	12000