# Employee Management System implementation with Oracle PL/SQL

Advanced Database Systems with Applications

[DSCI 32012]

# Assignment I

Group ID : CS\_G20

**Group Members:**

1. CS/2020/028 - C.M.P.S. Chandrasekara
2. CS/2020/037 - H.M.B.S. Induwara

1. **Business scenario of the chosen Information System**

**Employee Management System (EMS)**

The Employee Management System (EMS) is designed to streamline and automate HR operations within an organization. It handles various functions such as storing employee information, processing payroll, calculating benefits and deductions, and maintaining employee records. The goal of the EMS is to reduce manual workload, minimize errors, and ensure consistent and transparent management of employee-related data. One of the key components of this system is payroll processing — ensuring that employees are paid correctly and on time based on their salary structure and applicable deductions and benefits. By integrating payroll functionality within a PL/SQL package, the system enables centralized logic, easy reuse, and maintainability. The EMS supports HR personnel by allowing them to generate payroll, log payroll activity, and fetch the latest salary information quickly and reliably.

1. **What each component of the package does.**

*generate\_payroll(p\_emp\_id NUMBER)* (Public Procedure)

Triggers the payroll process for a given employee. It:

Retrieves the employee’s basic salary.

Calculates total allowances, deductions, and taxes.

Computes the net salary.

Inserts a new payroll entry into the payroll table.

Logs the payroll action using the *log\_payroll\_activity* procedure.

Displays a confirmation message using DBMS\_OUTPUT.

*get\_net\_salary(p\_emp\_id NUMBER)* (Public Function)

Returns the most recent net salary for the employee by querying the payroll records sorted by payment date.

*log\_payroll\_activity(p\_emp\_id NUMBER)* (Private Procedure)

Logs a payroll-related activity in the payroll\_log table. It records successful processing or notes if the employee was not found. It also catches and logs errors for debugging purposes.

*cal\_tax(p\_basic\_salary NUMBER)* (Private Function)

Calculates the tax based on salary:

60% tax if salary > 100,000

30% if salary > 50,000

5% for others

*cal\_deduction(p\_basic\_salary NUMBER)* (Private Function)

Computes deductions such as EPF and insurance based on predefined salary brackets.

*cal\_allowance(p\_basic\_salary NUMBER)* (Private Function)

Calculates allowances, including transport, medical, and a fixed food allowance, based on salary levels.

1. **What the trigger is meant for.**

### *trg\_validate\_payroll\_entry*

**Purpose:**  
This trigger ensures data integrity in the *Payroll* table by validating critical business rules before inserting or updating payroll records.

**Functionality:**

* **Pay period validation:** Prevents entries where the *pay\_period\_end\_date* is earlier than the *pay\_period\_start\_date*, which would represent an invalid date range for salary calculation.
* **Net salary check:** Ensures that the *net\_salary* is not negative, as negative take-home pay is not logically valid in a payroll system.

This trigger is executed **before** each insert or update operation on the *Payroll* table and helps maintain consistent and meaningful payroll data.

1. **Sample output & screenshots**

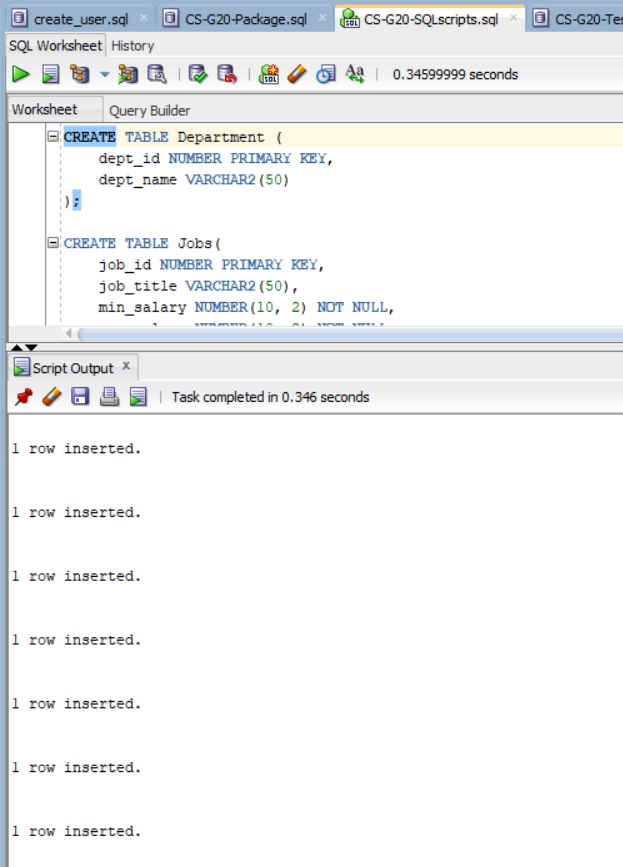
**Create user:**

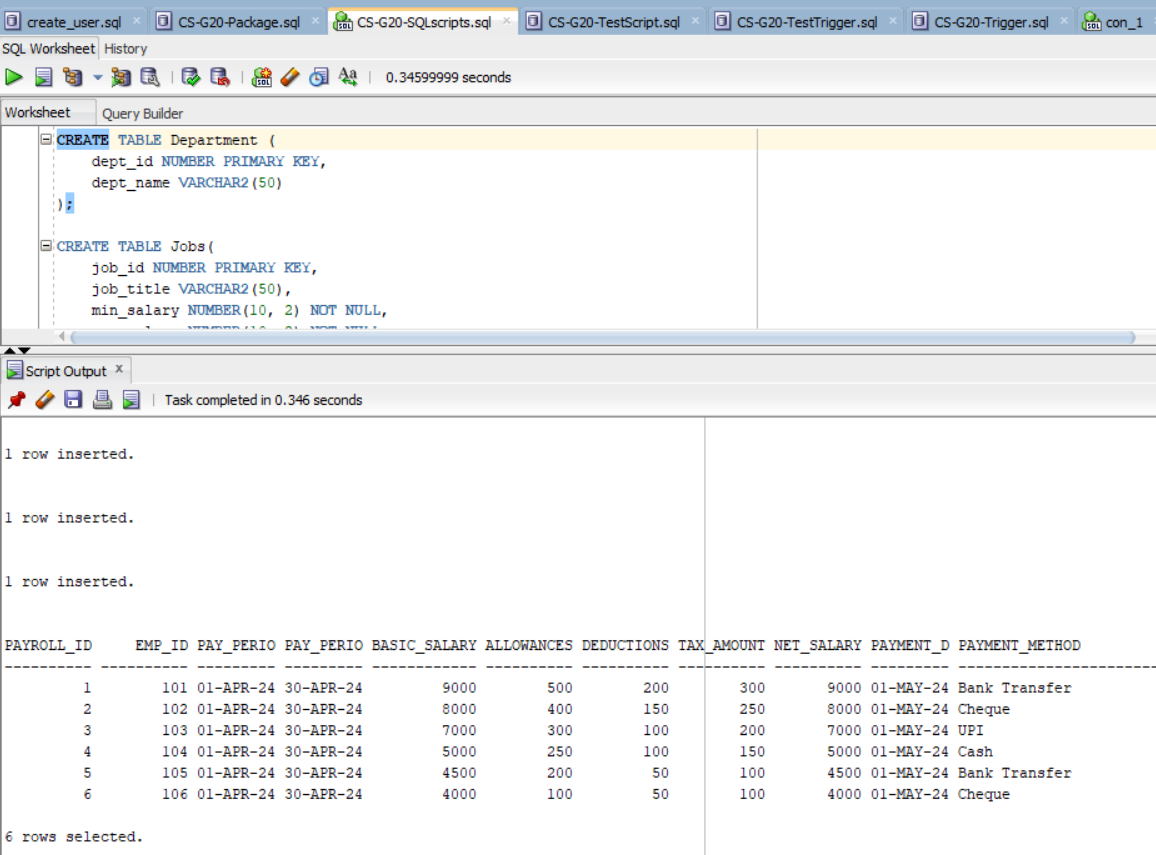
**A screenshot of a computer

AI-generated content may be incorrect.**

**Create tables and add dummy data: A screenshot of a computer

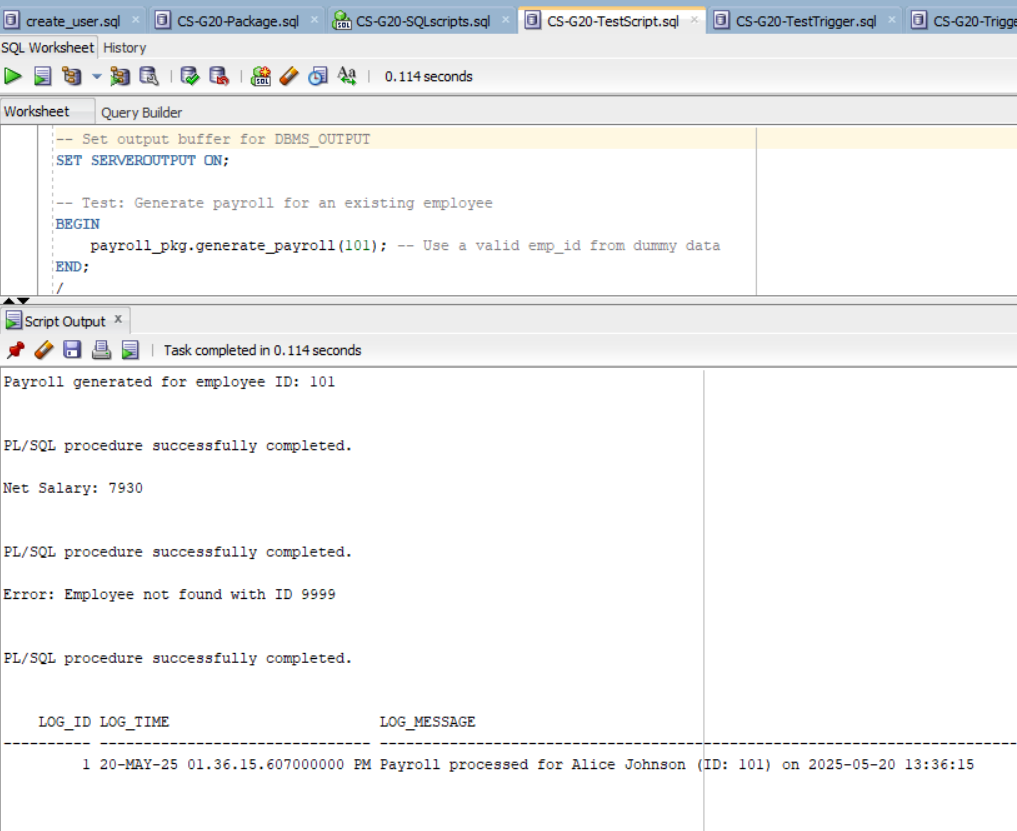
AI-generated content may be incorrect.**

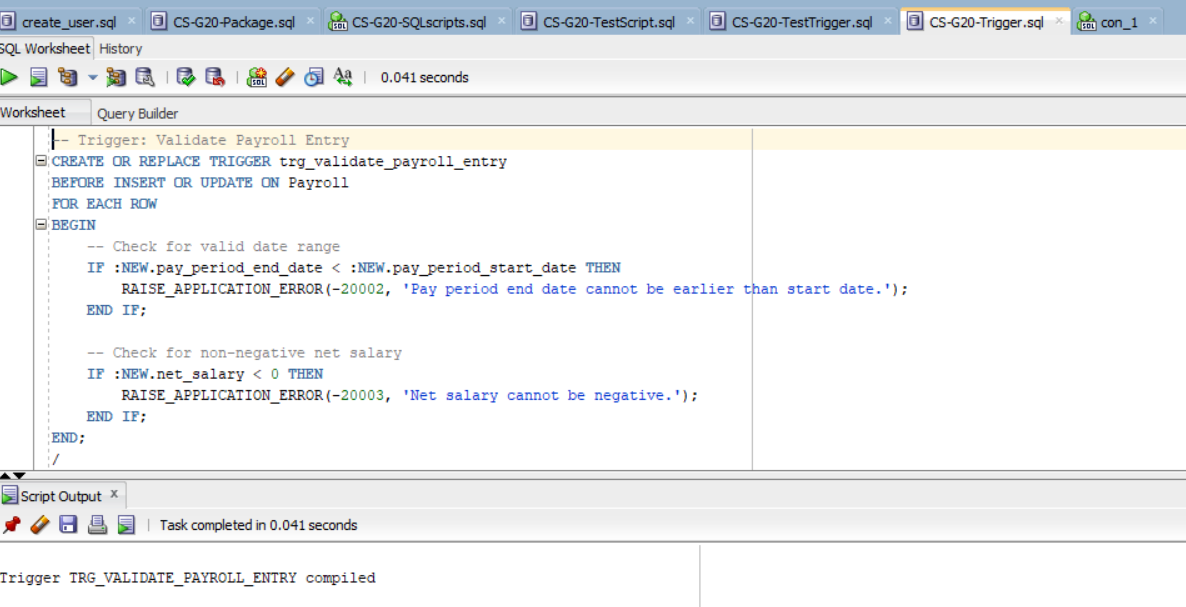
****

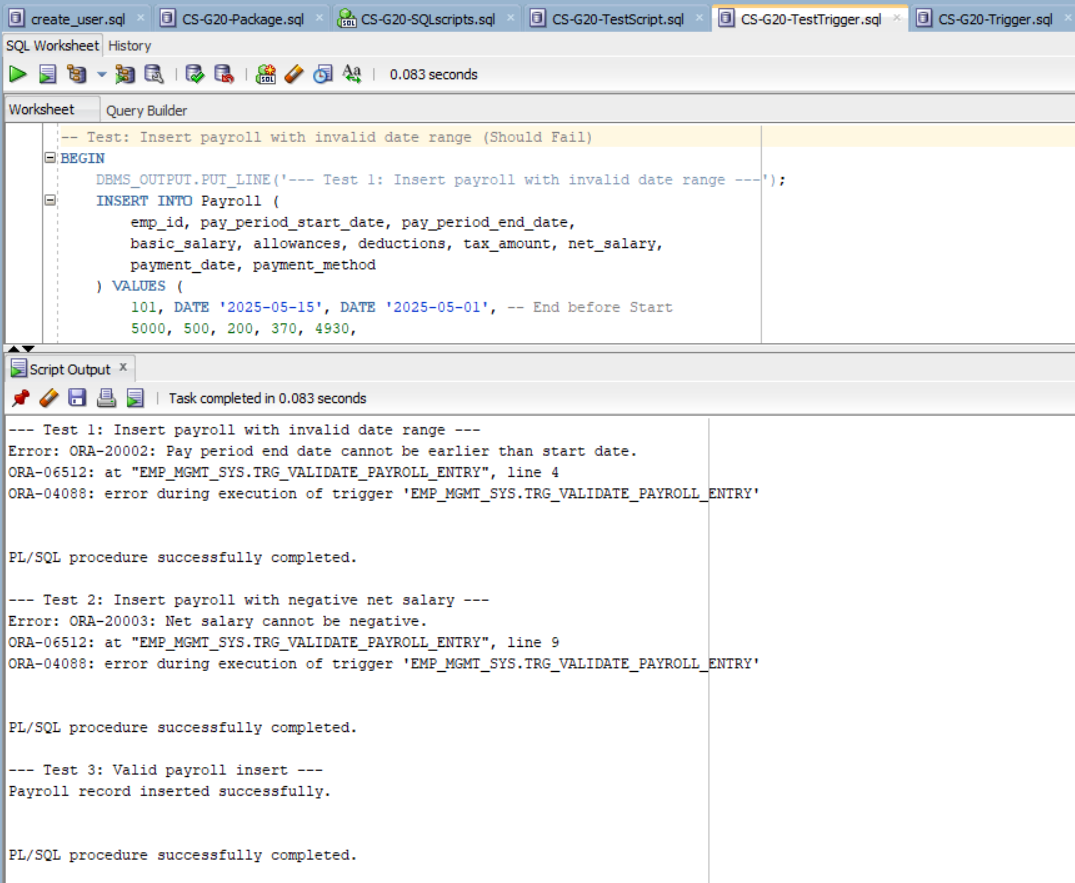
****

**Compile the package: A screenshot of a computer

AI-generated content may be incorrect.**

**Test the package:**

**Create trigger:**

**Test the trigger:**