Class Quiz I - Scenarios about triggers & package in PLSQL

Academic year 2025-2026, SEM II

Course Code and Name: INSY 8311 | Database development with PL/SQL

Instructor: Eric Maniraguha **Exam Duration:** 2hrs
Time: **Date:** November 24, 2025
Total max: /2pts

I. Scenario: AUCA System Access Policy

Business Rules:

1. Users **cannot access or record data** on the Sabbath (**Saturday and Sunday**).
2. Users can access the system **Monday to Friday, from 8:00 AM to 5:00 PM**.
3. Any **attempts outside these allowed days or hours** should be **blocked** and **logged** automatically.

Implementation Note:

- **Trigger 1:** Enforces access restrictions based on day and time. If a user attempts to access or modify data outside allowed hours, the action is prevented.
- **Trigger 2:** Records the attempted violations (who tried, when, and what action was attempted) into an **error logging table** for auditing purposes.

This ensures both **real-time enforcement** and **auditability** of unauthorized access attempts.

II. PL/SQL Assignment: HR Employee Management System Package

Scenario:
You are tasked with designing a PL/SQL package for an HR Employee Management System. The package should help manage employee salaries and taxes efficiently.

Requirements:

1. **Functions:**
   - Create one or more functions that calculate the RSSB tax for an employee.
   - The functions should return the employee's **net salary** after deducting the RSSB tax.
2. **Procedure:**

- Include a **dynamic procedure** that performs an operation (e.g., updating employee salary, inserting records, or generating a report).
- Use **dynamic SQL** where appropriate.

3. **Security Context:**
   - Correctly apply and demonstrate the use of **USER / CURRENT_USER** and **DEFINER / INVOKER rights** within your package.
   - Explain the difference between USER and CURRENT_USER in your comments.
4. **Deliverables:**
   - A fully compiled PL/SQL package with at least one function and one dynamic procedure.
   - Sample calls demonstrating the calculation of net salary and the execution of the dynamic procedure.
   - Clear comments explaining your code logic and the usage of USER and CURRENT_USER.

Optional Challenge:

- Extend the package to handle multiple employees in bulk using loops or cursors.

III. Your organization wants to strengthen system security by monitoring suspicious login behavior. A security policy has been introduced stating that:

“If any user attempts to log in more than two times with incorrect credentials during the same session or day, the system must immediately record the event and trigger a security alert.”

As the database developer, you are required to implement this policy.

Assignment Tasks

1. **Create a table** named login_audit to store all login attempts.
   The table should record:
   - Username
   - Attempt time
   - Status (SUCCESS / FAILED)
   - IP address or device information (optional)

2. **Create another table** named security_alerts to store alerts for suspicious activity.
   Each alert should contain:
   - Username
   - Number of failed attempts
   - Alert time
   - Alert message
   - Email or contact to notify
3. **Create a trigger** that fires *after each failed login attempt* (assume the application inserts login attempts into login_audit).
   The trigger must:
   - Count how many failed attempts the same user has made within a given period (e.g., same day).
   - If failed attempts exceed 2, insert a new record into security_alerts.
4. **(Optional – Advanced)**
   Create another trigger or stored procedure that automatically sends an email notification to the security team whenever a new alert is generated.

Expected Outcome

- A user who enters wrong credentials **once or twice** is only recorded in login_audit.
- A user who fails **three times or more** triggers a security alert, recorded in security_alerts, and (optionally) triggers an email notification.

IV. Scenario: Hospital Management Package with Bulk Processing

Background

The hospital management team wants to streamline patient management by storing patient and doctor information in the database. They also want to handle multiple patients at once efficiently and provide functionalities to display information and manage admissions.

As a database developer, you are tasked to create a **PL/SQL package** to support these operations.

Requirements

1. **Tables**:
   - patients table to store patient information (ID, name, age, gender, admitted status).

- doctors table to store doctor information (ID, name, specialty).

2. **Package Specification**:
    - Define a **collection type** to hold multiple patients for bulk processing.
    - Include procedures and functions such as:
        - **bulk_load_patients** – procedure to insert multiple patient records at once using bulk collection.
        - **show_all_patients** – function to display all patients (returns a cursor).
        - **count_admitted** – function to return the number of patients currently admitted.
        - **admit_patient** – procedure to update a patient's status as admitted.
3. **Package Body**:
    - Implement the above procedures and functions.
    - Use **bulk processing techniques** (FORALL) for efficient insertion.
    - Use **commits** appropriately for data consistency.
4. **Testing**:
    - Students must create test scripts to:
        - Load multiple patients using the bulk_load_patients procedure.
        - Display all patients using show_all_patients.
        - Admit one or more patients and verify the change using count_admitted.

Expected Outcome

- The package allows **efficient bulk insertion** of patient data.
- Students can **query patient information** through a function.
- The system tracks **admitted patients** and updates statuses correctly.
- Test scripts demonstrate that all procedures and functions work as expected.

Blessings to you all!