

**NSHURO ARNAUD NELLIGAN**

**27960**

**PHASE VI**

**Objective:**

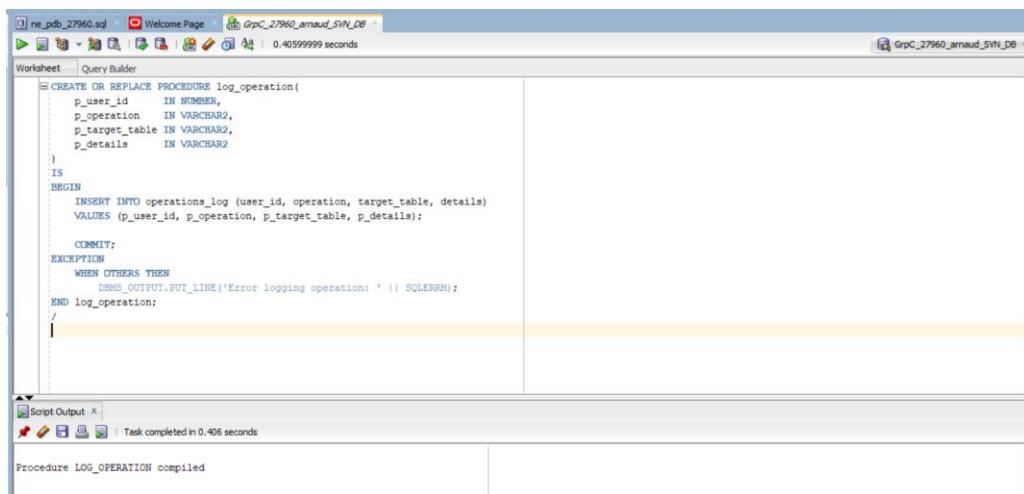
**Develop PL/SQL procedures, functions, packages, and cursors to handle:**

- **Logging database operations**
- **Detecting violations**
- **Sending alerts to administrators**
- **Exception handling**
- **Basic analytics preparation**

## **1. PL/SQL Procedures**

### **Procedure 1: Log Operation**

Logs any user operation in the database.



The screenshot shows the Oracle SQL Developer interface with a query builder window open. The code in the window is as follows:

```
CREATE OR REPLACE PROCEDURE log_operation(
    p_user_id      IN NUMBER,
    p_operation     IN VARCHAR2,
    p_target_table  IN VARCHAR2,
    p_details       IN VARCHAR2
)
IS
BEGIN
    INSERT INTO operations_log (user_id, operation, target_table, details)
    VALUES (p_user_id, p_operation, p_target_table, p_details);
    COMMIT;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error logging operation: ' || SQLERRM);
END log_operation;
```

The script output pane at the bottom shows the message "Procedure LOG\_OPERATION compiled".

### **Procedure 2: Notify Administrator**

Sends alerts to admins when a violation occurs.

```

CREATE OR REPLACE PROCEDURE notify_admin(
    pViolation_id IN NUMBER
)
IS
    v_user      VARCHAR2(50);
    v_details  VARCHAR2(400);
BEGIN
    SELECT u.username, o.details
    INTO v_user, v_details
    FROM violations v
    JOIN operations_log o ON v.log_id = o.log_id
    JOIN users u ON o.user_id = u.user_id
    WHERE v.violation_id = pViolation_id;

    DBMS_OUTPUT.PUT_LINE('ALERT: Violation detected! User: ' || v_user || ', Details: ' || v_details);

    -- Optional: email logic using UTL_MAIL package
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('No violation found for ID ' || pViolation_id);
END notify_admin;
/

```

Script Output: Task completed in 0.055 seconds

Procedure NOTIFY\_ADMIN compiled

## Procedure 3: Resolve Violation

Marks a violation as resolved.

```

CREATE OR REPLACE PROCEDURE resolve_violation(
    pViolation_id IN NUMBER
)
IS
BEGIN
    UPDATE violations
    SET status = 'RESOLVED'
    WHERE violation_id = pViolation_id;

    COMMIT;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error resolving violation: ' || SQLERRM);
END resolve_violation;
/

```

Script Output: Task completed in 0.037 seconds

Procedure RESOLVE\_VIOLATION compiled

## 2. PL/SQL Functions

### Function 1: Check Violation Status

Returns current status of a violation.

```

CREATE OR REPLACE FUNCTION getViolationStatus(
    pViolation_id IN NUMBER
) RETURN VARCHAR2
IS
    v_Status VARCHAR2(20);
BEGIN
    SELECT status INTO v_Status
    FROM violations
    WHERE violation_id = pViolation_id;

    RETURN v_Status;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RETURN 'NOT FOUND';
END getViolationStatus;
/

```

Script Output: Task completed in 0.048 seconds

Function GET\_VIOLATION\_STATUS compiled

### Function 2: Count Violations by User

```

ne_pdb_27960.sql | Welcome Page | GrpC_27960_armaud_SYN_DB |
Worksheet | Query Builder
CREATE OR REPLACE FUNCTION count_user_violations(
    p_user_id IN NUMBER
) RETURN NUMBER
IS
    v_count NUMBER;
BEGIN
    SELECT COUNT(*) INTO v_count
    FROM violations v
    JOIN operations_log o ON v.log_id = o.log_id
    WHERE o.user_id = p_user_id;
    RETURN v_count;
END count_user_violations;
/

```

Script Output: Task completed in 0.045 seconds

Function COUNT\_USER\_VIOLATIONS compiled

### 3. Cursors

#### Explicit Cursor: List All Violations

```

ne_pdb_27960.sql | Welcome Page | GrpC_27960_armaud_SYN_DB |
Worksheet | Query Builder
DECLARE
    CURSOR c_violations IS
        SELECT v.violation_id, u.username, v.severity, v.status_
        FROM violations v
        JOIN operations_log o ON v.log_id = o.log_id
        JOIN users u ON o.user_id = u.user_id;

    rViolation c_violations%ROWTYPE;
BEGIN
    OPEN c_violations;
    LOOP
        FETCH c_violations INTO rViolation;
        EXIT WHEN c_violations%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE('Violation ID: ' || rViolation.violation_id ||
                            ', User: ' || rViolation.username ||
                            ', Severity: ' || rViolation.severity ||
                            ', Status: ' || rViolation.status);
    END LOOP;
    CLOSE c_violations;
END;
/

```

PL/SQL procedure successfully completed.

### 4. Package Example

#### Package Specification

```

ne_pdb_27960.sql | Welcome Page | GrpC_27960_armaud_SYN_DB |
Worksheet | Query Builder
CREATE OR REPLACE PACKAGE pkg_security AS
    PROCEDURE log_operation(p_user_id NUMBER, p_operation VARCHAR2, p_target_table VARCHAR2, p_details VARCHAR2);
    PROCEDURE notify_admin(p_violation_id NUMBER);
    PROCEDURE resolveViolation(p_violation_id NUMBER);
    FUNCTION getViolationStatus(p_violation_id NUMBER) RETURN VARCHAR2;
    FUNCTION countUserViolations(p_user_id NUMBER) RETURN NUMBER;
END pkg_security;
/

```

Script Output: Task completed in 0.128 seconds

Package PKG\_SECURITY compiled

#### Package Body

ne\_db\_27960.sql | Welcome Page | Grpc\_27960\_armaud\_SVN\_DB | 0.068 seconds

Worksheet - Query Builder

```
CREATE OR REPLACE PACKAGE BODY pkg_security AS
  PROCEDURE log_operation(p_user_id NUMBER, p_operation VARCHAR2, p_target_table VARCHAR2, p_details VARCHAR2) IS
    BEGIN
      INSERT INTO operations_log (user_id, operation, target_table, details)
      VALUES (p_user_id, p_operation, p_target_table, p_details);
      COMMIT;
    END;

  PROCEDURE notify_admin(pViolation_id NUMBER) IS
    v_user VARCHAR2(50);
    v_details VARCHAR2(400);
  BEGIN
    SELECT u.username, o.details INTO v_user, v_details
    FROM violations v
    JOIN operations_log o ON v.log_id = o.log_id
    JOIN users u ON o.user_id = u.user_id
    WHERE v.violation_id = pViolation_id;
    DBMS_OUTPUT.PUT_LINE('ALERT: Violation detected! User: ' || v_user || ', Details: ' || v_details);
  END;

  PROCEDURE resolveViolation(pViolation_id NUMBER) IS
  BEGIN
    UPDATE violations
    SET status = 'RESOLVED'
    WHERE violation_id = pViolation_id;
    COMMIT;
  END;

  FUNCTION getViolationStatus(pViolation_id NUMBER) RETURN VARCHAR2 IS

```

Script Output X | Task completed in 0.068 seconds