שלב ג – תכנות

תוכנית 1

התוכנית מתארת מקרה שמטופל נזקק לטיפול דחוף ועוקבת אחרי התרופות שהרופא נותן לו בטיפול.

שימוש בפונקציה שבודקת זמינות של תרופה במלאי.

```
CREATE OR REPLACE NONEDITIONABLE FUNCTION check_medication_availability(
    p_medication_name IN VARCHAR2,
    p_strength IN VARCHAR2,
    p_quantity_needed IN NUMBER
) RETURN BOOLEAN
IS
    v_quantity_in_stock NUMBER;
BEGIN
    SELECT QuantityInStock INTO v_quantity_in_stock
    FROM Medication
    WHERE LOWER(Name) = LOWER(p_medication_name)
    AND Strength = p_strength
    AND ROWNUM = 1;

    RETURN v_quantity_in_stock >= p_quantity_needed;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RETURN FALSE;
END;
```

שימוש בפרוצדורה שבהינתן מטופל, רופא ורשימת תרופות מנהלת את הטיפול:

1. מוצאת את החולה או מכניסה אותו למערכת אם הוא חולה חדש.
2. מוצאת לו חדר פנוי במחלקת החירום.
3. יוצרת רשומה של הטיפול ומקשרת לכל התרופות שצריך לקבל (ומעדכנת את הכמות שלהן במאגר).

```
CREATE OR REPLACE PROCEDURE handle_emergency_treatment(
    p_patient_name IN VARCHAR2,
    p_patient_dob IN DATE,
    p_doctor_id IN NUMBER,
    p_medication_list IN SYS_REFCURSOR,
    p_treatment_result OUT SYS_REFCURSOR
)
IS
    v_patient_id NUMBER;
    v_treatment_id NUMBER;
    v_room_id NUMBER;
    v_medication_id NUMBER;
    v_emergency_dept_id NUMBER;

    TYPE medication_record IS RECORD (
        med_name VARCHAR2(100),
        strength VARCHAR2(50),
        quantity NUMBER
    );
    v_medication medication_record;

    insufficient_medication EXCEPTION;
    PRAGMA EXCEPTION_INIT(insufficient_medication, -20001);
BEGIN
    -- Find or create patient
    BEGIN
        SELECT PatientID INTO v_patient_id
        FROM Patient
        WHERE Name = p_patient_name AND DateOfBirth = p_patient_dob;
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            -- Create new patient
            SELECT NVL(MAX(PatientID), 0) + 1 INTO v_patient_id FROM Patient;
            INSERT INTO Patient (PatientID, Name, DateOfBirth)
            VALUES (v_patient_id, p_patient_name, p_patient_dob);
    END;

    -- Find available room in Emergency Department
    SELECT DepartmentID INTO v_emergency_dept_id
    FROM Department
    WHERE UPPER(Name) = 'EMERGENCY';

    SELECT MIN(RoomID) INTO v_room_id
    FROM Room
    WHERE DepartmentID = v_emergency_dept_id
    AND RoomID NOT IN (SELECT RoomID FROM Patient WHERE RoomID IS NOT NULL);

    -- Assign room to patient
    UPDATE Patient
    SET RoomID = v_room_id
    WHERE PatientID = v_patient_id;

    -- Create new treatment
    SELECT NVL(MAX(TreatmentID), 0) + 1 INTO v_treatment_id FROM Treatment;
    INSERT INTO Treatment (TreatmentID, Description, PatientID, DoctorID, TreatmentDate)
    VALUES (v_treatment_id, 'Emergency Treatment', v_patient_id, p_doctor_id, SYSDATE);

```

```
    -- Process medications
    LOOP
        FETCH p_medication_list INTO v_medication;
        EXIT WHEN p_medication_list%NOTFOUND;

        IF NOT check_medication_availability(
            v_medication.med_name,
            v_medication.strength,
            v_medication.quantity
        ) THEN
            RAISE insufficient_medication;
        END IF;

        SELECT MedicationID INTO v_medication_id
        FROM Medication
        WHERE LOWER(Name) = LOWER(v_medication.med_name)
        AND Strength = v_medication.strength

        -- Update medication stock
        UPDATE Medication
        SET QuantityInStock = QuantityInStock - v_medication.quantity
        WHERE MedicationID = v_medication_id;

        -- Link medication to treatment
        INSERT INTO Medication_Treatment (MedicationID, TreatmentID, Quantity)
        VALUES (v_medication_id, v_treatment_id, v_medication.quantity);
    END LOOP;

    -- Prepare treatment result
    OPEN p_treatment_result FOR
    SELECT t.TreatmentID, p.Name AS PatientName, d.Name AS DoctorName,
           m.Name AS MedicationName, m.Strength, m.Form, m.Manufacturer, mt.Quantity
    FROM Treatment t
    JOIN Patient p ON t.PatientID = p.PatientID
    JOIN Doctor d ON t.DoctorID = d.DoctorID
    JOIN Medication_Treatment mt ON t.TreatmentID = mt.TreatmentID
    JOIN Medication m ON mt.MedicationID = m.MedicationID
    WHERE t.TreatmentID = v_treatment_id;

    COMMIT;

EXCEPTION
    WHEN insufficient_medication THEN
        ROLLBACK;
        RAISE_APPLICATION_ERROR(-20001, 'Insufficient medication in stock');
    WHEN OTHERS THEN
        ROLLBACK;
        RAISE;
END;
```

תוכנית ראשית המוצאת רופא מתאים ואת רשימת התרופות, קוראת לפרוצדורה ומדפיסה תוצאות למסך.

```
DECLARE
    v_medication_list SYS_REFCURSOR;
    v_treatment_result SYS_REFCURSOR;
    v_patient_name VARCHAR2(100) := 'Kaela Cabena';
    v_patient_dob DATE := TO_DATE('1999-07-19', 'YYYY-MM-DD');
    v_doctor_id NUMBER;

    v_treatment_id NUMBER;
    v_patient_name_result VARCHAR2(100);
    v_doctor_name VARCHAR2(100);
    v_medication_name VARCHAR2(100);
    v_strength VARCHAR2(50);
    v_quantity NUMBER;
BEGIN
    -- Find a doctor in the Emergency Department
    SELECT DoctorID INTO v_doctor_id
    FROM Doctor d
    JOIN Department dep ON d.DepartmentID = dep.DepartmentID
    WHERE UPPER(dep.Name) = 'EMERGENCY'
    AND ROWNUM = 1;

    -- Prepare medication list
    OPEN v_medication_list FOR
    SELECT Name, Strength, 1 AS Quantity
    FROM Medication
    WHERE UPPER(Name) IN ('ASPIRIN', 'MENTHOL', 'OCTOCRYLENE')
    AND QuantityInStock > 0;
```

```
    -- Handle emergency treatment
    handle_emergency_treatment(
        p_patient_name => v_patient_name,
        p_patient_dob => v_patient_dob,
        p_doctor_id => v_doctor_id,
        p_medication_list => v_medication_list,
        p_treatment_result => v_treatment_result
    );

    -- Display treatment result
    DBMS_OUTPUT.PUT_LINE('Emergency Treatment Result:');
    LOOP
        FETCH v_treatment_result INTO
            v_treatment_id, v_patient_name_result, v_doctor_name,
            v_medication_name, v_strength, v_quantity;
        EXIT WHEN v_treatment_result%NOTFOUND;

        DBMS_OUTPUT.PUT_LINE('Treatment ID: ' || v_treatment_id);
        DBMS_OUTPUT.PUT_LINE('Patient: ' || v_patient_name_result);
        DBMS_OUTPUT.PUT_LINE('Doctor: ' || v_doctor_name);
        DBMS_OUTPUT.PUT_LINE('Medication: ' || v_medication_name ||
                             ', Strength: ' || v_strength ||
                             ' (Quantity: ' || v_quantity || ')');
        DBMS_OUTPUT.PUT_LINE('---');
    END LOOP;
    CLOSE v_treatment_result;
```

```
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLERRM);
        ROLLBACK;
END;
```

תוצאת ההרצה:

```
Emergency Treatment Result:
Treatment ID: 1005
Patient: Kaela Cabena
Doctor: Gunter Moyles
Medication: octocrylene, Strength: 1 (Quantity: 1)
---
Treatment ID: 1005
Patient: Kaela Cabena
Doctor: Gunter Moyles
Medication: Menthol, Strength: 2 (Quantity: 1)
---
Treatment ID: 1005
Patient: Kaela Cabena
Doctor: Gunter Moyles
Medication: OCTOCRYLENE, Strength: 4 (Quantity: 1)
---
Treatment ID: 1005
Patient: Kaela Cabena
Doctor: Gunter Moyles
Medication: Menthol, Strength: 5 (Quantity: 1)
---
Treatment ID: 1005
Patient: Kaela Cabena
Doctor: Gunter Moyles
Medication: Aspirin, Strength: 4 (Quantity: 1)
---
```

תוכנית 2

פונקציה שמחזירה נתונים סטטיסטיים אודות מחלקה (כרגע רק את מספר החולים, בהמשך אולי עוד נתונים):

```
-- Function to get department statistics
create or replace FUNCTION get_dept_stats(p_dept_id NUMBER) RETURN number IS
    v_result number;
BEGIN
    -- Count patients in the department
    SELECT COUNT(DISTINCT p.PatientID) INTO v_result
    FROM Patient p
    JOIN Room r ON p.RoomID = r.RoomID
    WHERE r.DepartmentID = p_dept_id;

    RETURN v_result;
END;
```

פרוצדורה שמוצאת רופאים שעובדים הרבה ומחזירה כref cursor:

```
CREATE OR REPLACE PROCEDURE find_high_workload_doctors (
      p_ref_cursor OUT SYS_REFCURSOR
) IS
    v_doctor_name Doctor.Name%TYPE;
    v_doctor_specialty Doctor.Specialty%TYPE;
    v_treatment_count NUMBER;
BEGIN
    OPEN p_ref_cursor FOR
        SELECT d.Name, d.Specialty, COUNT(t.TreatmentID) AS TreatmentCount
        FROM Doctor d
        LEFT JOIN Treatment t ON d.DoctorID = t.DoctorID
        GROUP BY d.DoctorID, d.Name, d.Specialty
        HAVING COUNT(t.TreatmentID) > (
            SELECT AVG(treatment_count)
            FROM (
                SELECT COUNT(TreatmentID) AS treatment_count
                FROM Treatment
                GROUP BY DoctorID
            )
        )
        ORDER BY TreatmentCount DESC;

    DBMS_OUTPUT.PUT_LINE('Doctors with High Workloads:');
    DBMS_OUTPUT.PUT_LINE('----------------------');

    LOOP
        FETCH p_ref_cursor INTO v_doctor_name, v_doctor_specialty, v_treatment_count;
        EXIT WHEN p_ref_cursor%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE('Dr. ' || v_doctor_name || ' (' || v_doctor_specialty ||
                             '): ' || v_treatment_count || ' treatments');
    END LOOP;

    CLOSE p_ref_cursor;
EXCEPTION
    WHEN OTHERS THEN
        IF p_ref_cursor%ISOPEN THEN
            CLOSE p_ref_cursor;
        END IF;
```

התוכנית הראשית מדפיסה את הנתונים עבור המחלקות והרופאים ע"י קריאה לפונקציה ולפרוצדורה:

```
DECLARE
    v_stats number;
    v_ref_cursor SYS_REFCURSOR;
    v_doctor_name VARCHAR2(100);
    v_doctor_specialty VARCHAR2(100);
    v_treatment_count NUMBER;
    v_total_patients NUMBER := 0;

    -- Cursor for departments
    CURSOR c_departments IS
        SELECT  DepartmentID, Name, Location
        FROM Department D
        WHERE EXISTS (
              SELECT ROOMID
              FROM ROOM R
              WHERE R.DEPARTMENTID = D.DEPARTMENTID
        )
        AND ROWNUM < 100
        ORDER BY DepartmentID
    ;

    -- Record type for department
    TYPE r_department IS RECORD (
        DepartmentID number,
        Name varchar2(100),
        Location varchar2(100)
    );
    v_department r_department;
BEGIN
    DBMS_OUTPUT.PUT_LINE('Department Analysis:');
    DBMS_OUTPUT.PUT_LINE('--------------------');

    -- Analyze each department using the function
    OPEN c_departments;
    LOOP
        FETCH c_departments INTO v_department;
        EXIT WHEN c_departments%NOTFOUND;
```

```
        v_stats := get_dept_stats(v_department.DepartmentID);
        v_total_patients := v_total_patients + v_stats;

        DBMS_OUTPUT.PUT_LINE('Department: ' || v_department.Name);
        DBMS_OUTPUT.PUT_LINE('Location: ' || v_department.Location);
        DBMS_OUTPUT.PUT_LINE('Patients: ' || v_stats);
        DBMS_OUTPUT.PUT_LINE('--------------------');
    END LOOP;
    CLOSE c_departments;

    -- Calculate and display overall statistics
    DBMS_OUTPUT.PUT_LINE('--------------------');
    DBMS_OUTPUT.PUT_LINE('Overall Statistics:');
    DBMS_OUTPUT.PUT_LINE('Total Patients: ' || v_total_patients);

    find_high_workload_doctors(v_ref_cursor);

    LOOP
        FETCH v_ref_cursor INTO v_doctor_name, v_doctor_specialty, v_treatment_count;
        EXIT WHEN v_ref_cursor%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE('Dr. ' || v_doctor_name || ' (' || v_doctor_specialty ||
                             '): ' || v_treatment_count || ' treatments');
    END LOOP;

    CLOSE v_ref_cursor;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLERRM);
END;
```

תוצאת ההרצה (חלקי):

```
Department: Intensive
Location: P-3
Patients: 2
--------------------
Department: Emergency
Location: G-5
Patients: 1
--------------------
Department: Orthopedics
Location: D-9
Patients: 3
--------------------
Department: Physical
Location: C-7
Patients: 1
--------------------
--------------------
Overall Statistics:
Total Patients: 145
Doctors with High Workloads:
--------------------
Dr. Goran Dabourne (Obstetrics): 7 treatments
Dr. Skipp Faulder (Psychiatry): 7 treatments
Dr. Robenia Carrabott (Neurology): 7 treatments
Dr. Sayre Lavalde (Obstetrics): 6 treatments
Dr. Derwin Fullun (Radiology): 6 treatments
Dr. Clayborne Shank (Obstetrics): 6 treatments
Dr. Even Beartup (Pediatrics): 6 treatments
Dr. Silvie Capron (Psychiatry): 6 treatments
Dr. Jacquenetta Eitter (Radiology): 6 treatments
Dr. Ellynn Hadingham (Psychiatry): 6 treatments
Dr. Becki Kembry (Pharmacy): 5 treatments
Dr. Berky Umbers (Laboratory): 5 treatments
Dr. Marten Elwood (Radiology): 5 treatments
```