

דו"ח פרויקט שלב 3

הפרוצדורות:

פרוצדורה 1:

```

CREATE OR REPLACE PROCEDURE reorganize_platoon_squads(p_platoon_number IN
NUMBER)
IS
    TYPE squad_record IS RECORD (
        squad_number NUMBER,
        squad_name VARCHAR2(20),
        number_of_soldiers NUMBER
    );
    TYPE squad_table IS TABLE OF squad_record;
    v_squads squad_table;

    v_total_soldiers NUMBER := 0;
    v_target_squad_size NUMBER;
    v_current_squad NUMBER := 1;
    v_soldiers_in_current_squad NUMBER := 0;
BEGIN
    -- Get all squads in the platoon
    SELECT SquadNumber, SquadName, NumberOfSoldiers
    BULK COLLECT INTO v_squads
    FROM Squad
    WHERE PlatoonNumber = p_platoon_number;

    -- Calculate total number of soldiers
    SELECT SUM(NumberOfSoldiers) INTO v_total_soldiers
    FROM Squad
    WHERE PlatoonNumber = p_platoon_number;

    -- Calculate target squad size
    v_target_squad_size := CEIL(v_total_soldiers / v_squads.COUNT);

    -- Reorganize squads
    FOR i IN 1..v_squads.COUNT LOOP
        IF v_soldiers_in_current_squad + v_squads(i).number_of_soldiers <=
v_target_squad_size THEN
            -- Add soldiers to current squad
            UPDATE Soldier
            SET SquadNumber = v_squads(v_current_squad).squad_number
            WHERE SquadNumber = v_squads(i).squad_number;

            v_soldiers_in_current_squad := v_soldiers_in_current_squad +
v_squads(i).number_of_soldiers;
        ELSE
            -- Move to next squad

```

```

v_current_squad := v_current_squad + 1;
v_soldiers_in_current_squad := v_squads(i).number_of_soldiers;

UPDATE Soldier
SET SquadNumber = v_squads(v_current_squad).squad_number
WHERE SquadNumber = v_squads(i).squad_number;
END IF;
END LOOP;

-- Update squad sizes
FOR i IN 1..v_squads.COUNT LOOP
    UPDATE Squad
    SET NumberOfSoldiers = (
        SELECT COUNT(*)
        FROM Soldier
        WHERE SquadNumber = v_squads(i).squad_number
    )
    WHERE SquadNumber = v_squads(i).squad_number;
END LOOP;

COMMIT;
DBMS_OUTPUT.PUT_LINE('Platoon ' || p_platoon_number || ' reorganized
successfully.');
```

```

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

הסבר:

הפרוצדורה הזו, בשם "reorganize_platoon_squads", מארגנת מחדש את החיילים בכיתות של מחלקה נתונה. הנה השלבים העיקריים:

1. היא מקבלת מספר מחלקה כפרמטר.
2. היא אוספת מידע על כל הכיתות במחלקה זו.
3. היא מחשבת את סך כל החיילים במחלקה.
4. היא מחשבת את גודל הכיתה האידיאלי על ידי חלוקת סך החיילים במספר הכיתות.
5. היא עוברת על כל החיילים ומשייכת אותם מחדש לכיתות, כך שכל כיתה תהיה קרובה ככל האפשר לגודל האידיאלי.
6. לבסוף, היא מעדכנת את מספר החיילים בכל כיתה בטבלת Squad.

המטרה היא ליצור איזון טוב יותר בין הכיתות במחלקה, כך שכל הכיתות יהיו בגודל דומה ככל האפשר.

פרוצדורה 2:

```

CREATE OR REPLACE PROCEDURE promote_soldiers
IS
    CURSOR c_soldiers IS
```

```

        SELECT PersonalNumber, SRank, DeploymentDate
        FROM Soldier
        FOR UPDATE OF SRank;

        v_months_deployed NUMBER;
        v_new_rank VARCHAR2(20);
BEGIN
    FOR soldier IN c_soldiers LOOP
        v_months_deployed := MONTHS_BETWEEN(SYSDATE, soldier.DeploymentDate);

        IF v_months_deployed >= 36 AND soldier.SRank = 'Private' THEN
            v_new_rank := 'Corporal';
        ELSIF v_months_deployed >= 60 AND soldier.SRank = 'Corporal' THEN
            v_new_rank := 'Sergeant';
        ELSIF v_months_deployed >= 84 AND soldier.SRank = 'Sergeant' THEN
            v_new_rank := 'Staff Sergeant';
        ELSE
            v_new_rank := soldier.SRank;
        END IF;

        IF v_new_rank != soldier.SRank THEN
            UPDATE Soldier
            SET SRank = v_new_rank
            WHERE CURRENT OF c_soldiers;

            DBMS_OUTPUT.PUT_LINE('Soldier ' || soldier.PersonalNumber || '
promoted from ' || soldier.SRank || ' to ' || v_new_rank);
        END IF;
    END LOOP;

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

הסבר:

הפרוצדורה "promote_soldiers" מקדמת חיילים בדרגה בהתבסס על זמן השירות שלהם. הנה השלבים העיקריים:

1. היא מגדירה cursor שמכיל את כל החיילים, כולל המספר האישי שלהם, הדרגה הנוכחית, ותאריך הגיוס.
2. עבור כל חייל, היא מחשבת כמה חודשים הוא שירת מאז הגיוס.
3. בהתאם לזמן השירות והדרגה הנוכחית, היא קובעת אם החייל זכאי לקידום:
 - טוראי מקודם לרב"ט אחרי 36 חודשים
 - רב"ט מקודם לסמל אחרי 60 חודשים

- סמל מקודם לסמ"ר אחרי 84 חודשים
- 4. אם החייל זכאי לקידום, היא מעדכנת את הדרגה שלו בטבלת Soldier.
- 5. היא מדפיסה הודעה על כל קידום שבוצע.
- 6. בסוף התהליך, היא מבצעת COMMIT לשמירת השינויים.

הפרוצדורה משתמשת ב cursor-עם האפשרות FOR UPDATE, מה שמאפשר לה לעדכן את הנתונים תוך כדי סריקתם. היא גם כוללת טיפול בשגיאות (exception handling) למקרה שמשומו משתבש במהלך הביצוע.

זוהי דרך יעילה לבצע קידומים אוטומטיים לכל החיילים בבסיס הנתונים בהתבסס על הקריטריונים שהוגדרו.

הפונקציות :

פונקציה 1 :

```
CREATE OR REPLACE FUNCTION get_brigade_strength(p_brigade_number IN NUMBER)
RETURN NUMBER
IS
    v_total_soldiers NUMBER := 0;
    CURSOR c_battalions IS
        SELECT BattalionNumber
        FROM Battalion
        WHERE BrigadeNumber = p_brigade_number;

    v_battalion_number Battalion.BattalionNumber%TYPE;
BEGIN
    FOR battalion_rec IN c_battalions LOOP
        v_battalion_number := battalion_rec.BattalionNumber;

        FOR company_rec IN (SELECT CompanyNumber FROM Company WHERE
BattalionNumber = v_battalion_number) LOOP
            FOR platoon_rec IN (SELECT PlatoonNumber FROM Platoon WHERE
CompanyNumber = company_rec.CompanyNumber) LOOP
                FOR squad_rec IN (SELECT SquadNumber FROM Squad WHERE
PlatoonNumber = platoon_rec.PlatoonNumber) LOOP
                    v_total_soldiers := v_total_soldiers + NVL((SELECT NumberOfSoldiers FROM Squad
WHERE SquadNumber = squad_rec.SquadNumber), 0);
                FROM Squad
                WHERE SquadNumber = squad_rec.SquadNumber;
            END LOOP;
        END LOOP;
    END LOOP;

    RETURN v_total_soldiers;
EXCEPTION
```

```

    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('No data found for Brigade Number: ' ||
p_brigade_number);
        RETURN 0;
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLERRM);
        RETURN -1;
END;

```

הסבר:

הפונקציה "get_brigade_strength" מחשבת את המספר הכולל של החיילים בחטיבה נתונה. הנה ההסבר:

1. הפונקציה מקבלת מספר חטיבה כפרמטר קלט.
2. היא מגדירה cursor בשם c_battalions שמכיל את כל מספרי הגדודים בחטיבה הנתונה.
3. הפונקציה עוברת על כל הרמות בהיררכיה הצבאית מהחטיבה ועד לכיתה:
 - היא עוברת על כל הגדודים בחטיבה
 - עבור כל גדוד, היא עוברת על כל הפלוגות
 - עבור כל פלוגה, היא עוברת על כל המחלקות
 - עבור כל מחלקה, היא עוברת על כל הכיתות
4. בשלב הכיתה, היא מוסיפה את מספר החיילים בכיתה למונה הכולל (v_total_soldiers).
5. בסוף התהליך, היא מחזירה את המספר הכולל של החיילים.
6. הפונקציה כוללת טיפול בשגיאות:
 - אם לא נמצאו נתונים עבור החטיבה, היא מחזירה 0
 - אם יש שגיאה אחרת, היא מחזירה -1

פונקציה 2:

```

CREATE OR REPLACE FUNCTION get_avg_age_in_platoon(p_platoon_number IN NUMBER)
RETURN NUMBER
IS
    v_avg_age NUMBER;
BEGIN
    SELECT AVG(MONTHS_BETWEEN(SYSDATE, s.BirthDate) / 12)
    INTO v_avg_age
    FROM Soldier s
    JOIN Squad sq ON s.SquadNumber = sq.SquadNumber
    JOIN Platoon p ON sq.PlatoonNumber = p.PlatoonNumber
    WHERE p.PlatoonNumber = p_platoon_number;

    IF v_avg_age IS NULL THEN
        RAISE NO_DATA_FOUND;
    END IF;

    RETURN ROUND(v_avg_age, 2);

```

```

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('No soldiers found in Platoon Number: ' ||
p_platoon_number);
        RETURN 0;
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLERRM);
        RETURN -1;
END;

```

הסבר:

הפונקציה "get_avg_age_in_platoon" מחשבת את הגיל הממוצע של החיילים במחלקה נתונה. הנה ההסבר:

1. הפונקציה מקבלת מספר מחלקה כפרמטר קלט.
2. היא מבצעת שאילתת SQL שמחשבת את הגיל הממוצע של החיילים במחלקה :
 - היא מחברת את טבלאות Soldier, Squad ו Platoon כדי לקשר בין החיילים למחלקה.
 - היא מחשבת את הגיל של כל חייל באמצעות MONTHS_BETWEEN(SYSDATE, s.BirthDate) / 12.
 - היא מחשבת את הממוצע של כל הגילאים הללו.
3. התוצאה נשמרת במשתנה v_avg_age.
4. אם לא נמצאו חיילים במחלקה (כלומר v_avg_age הוא NULL), היא מעלה חריגה של NO_DATA_FOUND.
5. אם נמצאו חיילים, היא מחזירה את הגיל הממוצע מעוגל לשתי ספרות אחרי הנקודה העשרונית.
6. הפונקציה כוללת טיפול בשגיאות :
 - אם לא נמצאו חיילים במחלקה, היא מדפיסה הודעה ומחזירה 0.
 - אם יש שגיאה אחרת כלשהי, היא מדפיסה את הודעת השגיאה ומחזירה -1.

ה s MAIN:

הראשון:

```

-- Test script for analyze_and_reorganize_brigade
DECLARE
    v_brigade_number NUMBER := 830539;
    v_brigade_strength NUMBER;
BEGIN
    -- Call a function
    v_brigade_strength := get_brigade_strength(v_brigade_number);
    DBMS_OUTPUT.PUT_LINE('Brigade ' || v_brigade_number || ' strength: ' ||
v_brigade_strength || ' soldiers');

```

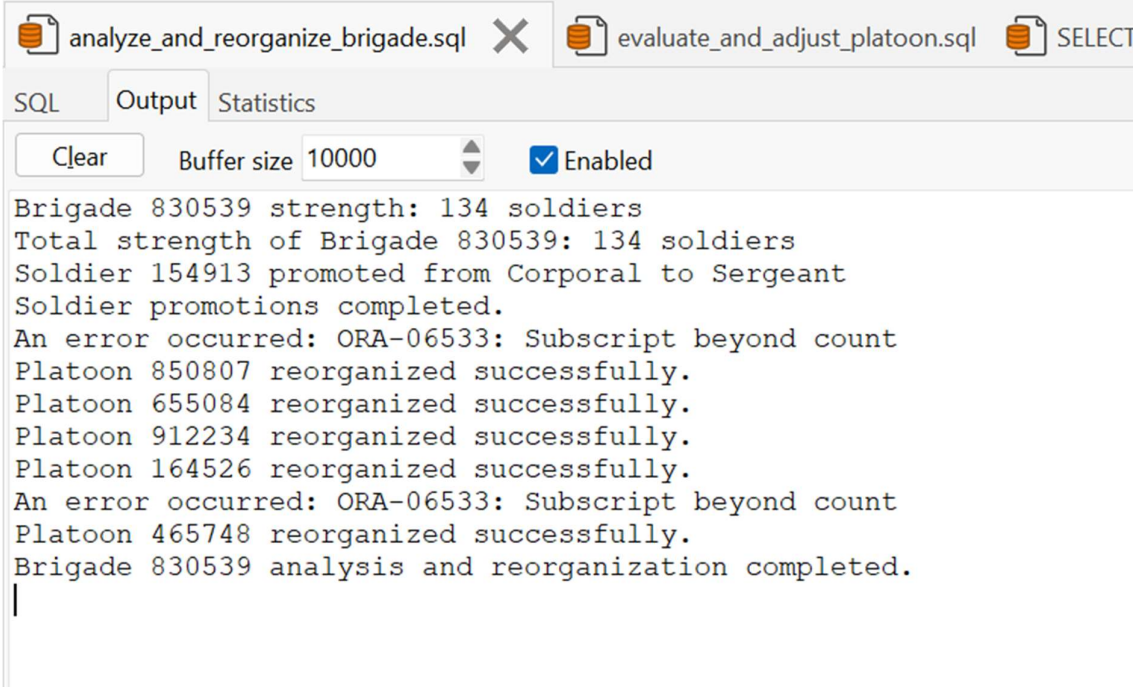
```
-- Call a procedure
analyze_and_reorganize_brigade(v_brigade_number);

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

1. הסבר:
המייך מתחיל בהגדרת שני משתנים :
v_brigade_number מספר החטיבה שאנחנו רוצים לנתח ולארגן מחדש (במקרה זה 883025).
v_brigade_strength משתנה שימש לאחסון מספר החיילים בחטיבה.
2. הוא קורא לפונקציה get_brigade_strength עם מספר החטיבה כפרמטר. התוצאה (מספר החיילים בחטיבה) נשמרת ב-v_brigade_strength.
3. הוא מדפיס את מספר החיילים בחטיבה.
4. לאחר מכן, הוא קורא לפרוצדורה analyze_and_reorganize_brigade עם אותו מספר חטיבה. פרוצדורה זו מנתחת ומארגנת מחדש את החטיבה (הפעולות המדויקות תלויות בתוכן הפרוצדורה).
5. יש טיפול בשגיאות (EXCEPTION) אם מתרחשת שגיאה כלשהי במהלך הביצוע, היא תיתפס והודעת השגיאה תודפס.

בסך הכל, המייך הזה מדגים שימוש הן בפונקציה (לקבלת מידע) והן בפרוצדורה (לביצוע פעולה) על אותה חטיבה, תוך טיפול בשגיאות אפשריות. זה מאפשר לבדוק את גודל החטיבה ואז לבצע עליה ניתוח וארגון מחדש בצורה מסודרת.

הוכחת הרצה:



The screenshot shows an SQL IDE with a script named 'analyze_and_reorganize_brigade.sql' open. The 'Output' tab is selected, displaying the results of the script execution. The output text is as follows:

```
Brigade 830539 strength: 134 soldiers
Total strength of Brigade 830539: 134 soldiers
Soldier 154913 promoted from Corporal to Sergeant
Soldier promotions completed.
An error occurred: ORA-06533: Subscript beyond count
Platoon 850807 reorganized successfully.
Platoon 655084 reorganized successfully.
Platoon 912234 reorganized successfully.
Platoon 164526 reorganized successfully.
An error occurred: ORA-06533: Subscript beyond count
Platoon 465748 reorganized successfully.
Brigade 830539 analysis and reorganization completed.
```

השני:

```
-- Declare block for procedure
DECLARE
    -- Procedure to evaluate and adjust platoon
    PROCEDURE evaluate_and_adjust_platoon(p_platoon_number IN NUMBER) IS
        v_avg_age NUMBER;
        v_ideal_age_range_min NUMBER := 20;
        v_ideal_age_range_max NUMBER := 30;

        CURSOR c_young_soldiers IS
            SELECT s.PersonalNumber, s.SName, s.BirthDate
            FROM Soldier s
            JOIN Squad sq ON s.SquadNumber = sq.SquadNumber
            WHERE sq.PlatoonNumber = p_platoon_number
            AND TRUNC(MONTHS_BETWEEN(SYSDATE, s.BirthDate) / 12) <
v_ideal_age_range_min;

        CURSOR c_old_soldiers IS
            SELECT s.PersonalNumber, s.SName, s.BirthDate
            FROM Soldier s
            JOIN Squad sq ON s.SquadNumber = sq.SquadNumber
            WHERE sq.PlatoonNumber = p_platoon_number
            AND TRUNC(MONTHS_BETWEEN(SYSDATE, s.BirthDate) / 12) >
v_ideal_age_range_max;
    BEGIN
        -- Get average age of soldiers in the platoon
        v_avg_age := get_avg_age_in_platoon(p_platoon_number);
        DBMS_OUTPUT.PUT_LINE('Average age in Platoon ' || p_platoon_number || ': '
|| v_avg_age || ' years');

        -- Evaluate age composition
        IF v_avg_age < v_ideal_age_range_min THEN
            DBMS_OUTPUT.PUT_LINE('Platoon is too young on average. Consider
transferring in some older soldiers.');
```



```

        TRUNC(MONTHS_BETWEEN(SYSDATE, young_soldier.BirthDate) / 12) || '
years)');
    END LOOP;

    -- List old soldiers
    DBMS_OUTPUT.PUT_LINE('Soldiers older than ' || v_ideal_age_range_max || '
years:');
    FOR old_soldier IN c_old_soldiers LOOP
        DBMS_OUTPUT.PUT_LINE(' ' || old_soldier.SName || ' (Age: ' ||
            TRUNC(MONTHS_BETWEEN(SYSDATE, old_soldier.BirthDate) / 12) || '
years)');
    END LOOP;

    -- Reorganize platoon
    reorganize_platoon_squads(p_platoon_number);
    DBMS_OUTPUT.PUT_LINE('Platoon ' || p_platoon_number || ' has been
reorganized.');
```

```

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

BEGIN
    -- Test variables
    DECLARE
        v_test_platoon_number NUMBER := 248379; -- Use a valid platoon number
    BEGIN
        -- Call the procedure
        evaluate_and_adjust_platoon(v_test_platoon_number);
    END;

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

הסבר:

שבודקת את הרכב evaluate_and_adjust_platoon והזו מגדירה פרוצדורה בשם הגילאים של החיילים בפלוגה מסוימת ומציעה שינויים במידת הצורך

1. מחזירה את הגיל get_avg_age_in_platoon הפונקציה: **חישוב גיל ממוצע** הממוצע של החיילים בפלוגה
2. אם הגיל הממוצע נמוך מ-20 או גבוה מ-30, מוצגת הודעה: **הערכת הרכב גילאים** שממליצה על העברת חיילים מבוגרים או צעירים לפלוגה

3. **רשימת חיילים צעירים ומבוגרים**: ו-c_old_soldiers ו-c_young_soldiers המצביעים על רשימת החיילים הצעירים והמבוגרים בהתאמה.
4. **מבצעת ארגון** reorganize_platoon_squads הפונקציה: **ארגון מחדש של הפלוגה**. מחדש של הפלוגה לפי הגילאים.

בסוף, התוכנית מדפיסה את הגילאים ואת החיילים בכל קטגוריה ומבצעת ארגון מחדש של הפלוגה.

הרצה:

The screenshot shows a SQL query execution window with three tabs: "SELECT DISTINCT p.PlatoonN ...", "SELECT sq.SquadNumber, sq. ...", and "evaluate_and_adjust_platoon.sql". The "Output" tab is active, displaying the following text:

```

Average age in Platoon 248379: 35.73 years
Platoon is too old on average. Consider transferring in some younger soldiers.
Soldiers younger than 20 years:
Soldiers older than 30 years:
    Hayley Ellison (Age: 35 years)
Platoon 248379 reorganized successfully.
Platoon 248379 has been reorganized.

```