

תוכן

[שלב 3 3](#_Toc170255915)

[3.1 קבוצת תוכניות א' – יצירת טבלת יעדים מסוכנים, למעצר 3](#_Toc170255916)

[3.1.1 – פונקציה Calculate\_Distance 4](#_Toc170255917)

[3.1.2 – פונקציה Calculate\_Danger\_Level: 5](#_Toc170255918)

[3.1.3 הפונקציה Insert\_Dangerous\_People 7](#_Toc170255919)

[3.1.4 הפונקציה הראשית Main\_Procedure 8](#_Toc170255920)

# שלב 3

## 3.1 קבוצת תוכניות א' – יצירת טבלת יעדים מסוכנים, למעצר

#### תנאים מקדימים - Alter Table

לפני הרצת השאילתה יצרנו את טבלת Dangerous\_People\_Operations, אליה יוכנסו בסופו של דבר תוצאות ההרצה של רצף התוכניות שתוארו לעיל.

*-- Create the Dangerous\_People\_Operations table*

CREATE TABLE Dangerous\_People\_Operations

(

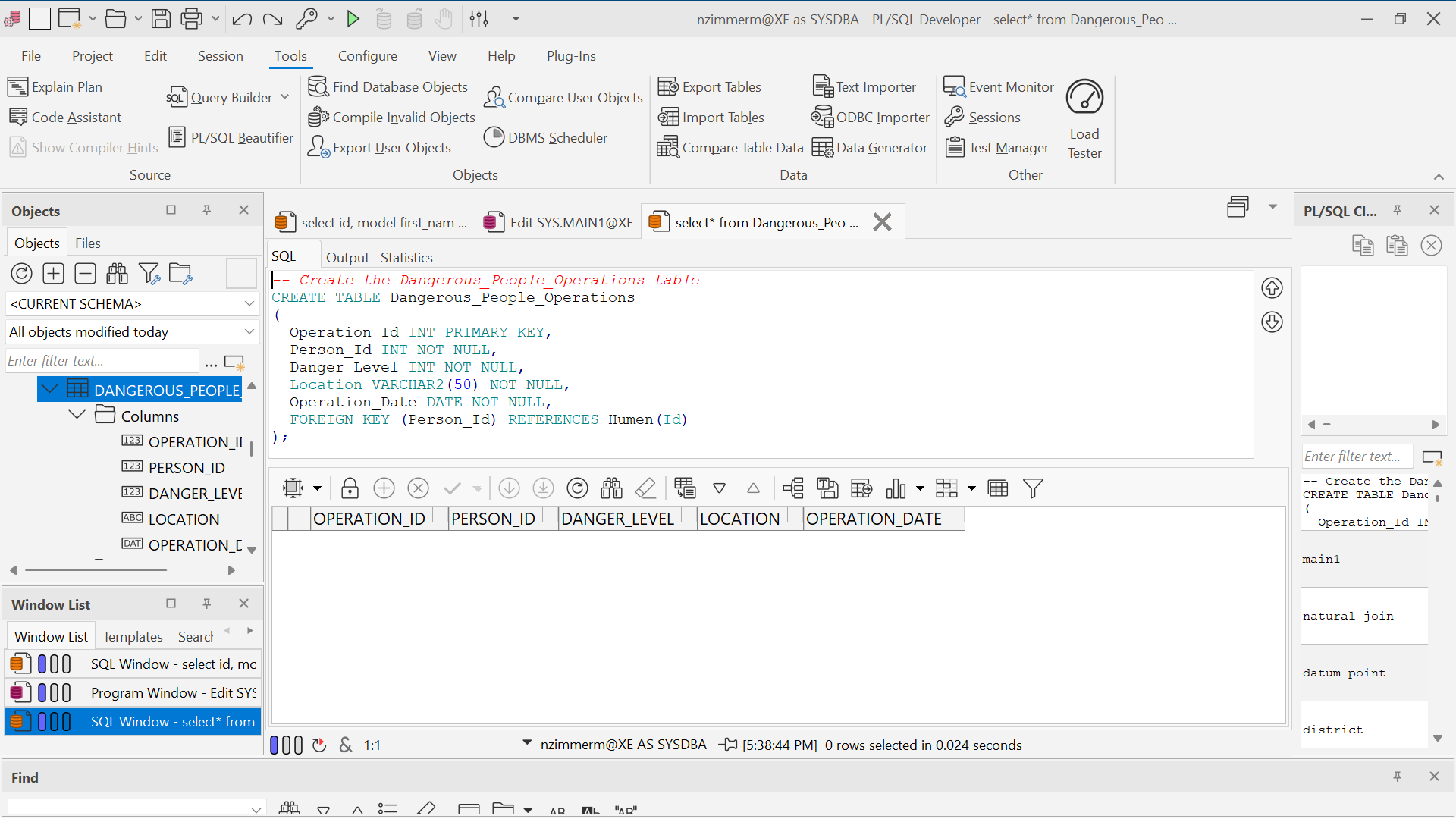
Operation\_Id INT PRIMARY KEY,

Person\_Id INT NOT NULL,

Danger\_Level INT NOT NULL,

Location VARCHAR2(50) NOT NULL,

FOREIGN KEY (Person\_Id) REFERENCES Humen(Id)

);

### 3.1.1 – פונקציה Calculate\_Distance

#### תיאור מילולי

הפונקציה מקבלת מיקום גיאוגרפי ומחזירה ציון לפי מרחקו מדרום הרצועה (ישמש כאחד הפרמטרים לתיעדוף האופרציה על ה"יעד"), ככל שהיעד רחוק יותר מדרום הרצועה הציון יהיה נמוך יותר.

#### הקוד בSQL

CREATE OR REPLACE FUNCTION Calculate\_Distance (datum\_point IN VARCHAR2)

RETURN NUMBER

IS

region\_code NUMBER;

BEGIN

CASE

WHEN datum\_point LIKE '%Southern Gaza%' THEN

region\_code := 100;

WHEN datum\_point LIKE '%Central Gaza%' THEN

region\_code := 50;

WHEN datum\_point LIKE '%City Gaza%' THEN

region\_code := 20;

WHEN datum\_point LIKE '%Northern Gaza%' THEN

region\_code := 10;

ELSE

region\_code := 0; *--default value*

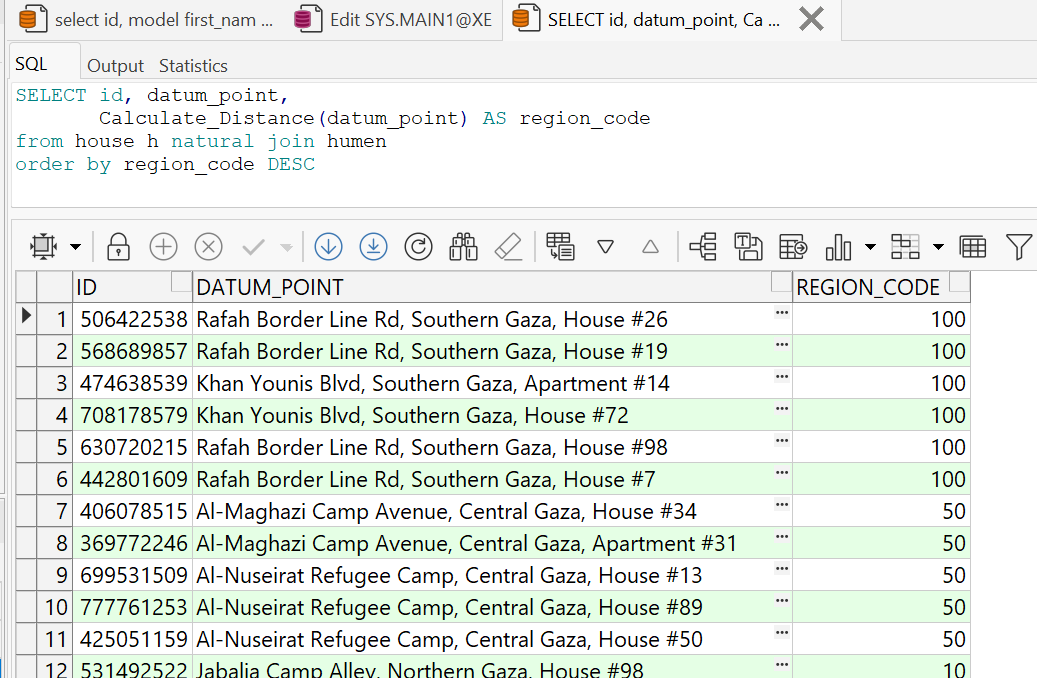
END CASE;

RETURN region\_code;

END;

/

#### מדגם תוצאות ההרצה



### 3.1.2 – פונקציה Calculate\_Danger\_Level:

#### תיאור מילולי:

הפונקציה Calculate\_Danger\_Level מבצעת תהליך של סינון ודירוג אנשים מסוכנים בהתבסס על רשימת כלי נשק, מיקום האנשים, פיגועי הטרור שביצעו ומרחקם מדרום הרצועה ( ע"י Calculate\_Distance).

#### הקוד בSQL:

CREATE OR REPLACE FUNCTION Calculate\_Danger\_Level(p\_id INT) RETURN INT IS

v\_danger\_level INT := 0;

v\_terror\_count INT;

v\_weapon\_count INT;

v\_incrim\_count INT;

v\_datum\_point VARCHAR2(100);

v\_distance\_grade NUMBER;

BEGIN

*-- Check if the person is in a terror organization*

SELECT COUNT(\*) INTO v\_terror\_count FROM Terror\_Organization WHERE Id = p\_id;

IF v\_terror\_count > 0 THEN

v\_danger\_level := v\_danger\_level + 3;

END IF;

*-- Check if the person has a weapon*

SELECT COUNT(\*) INTO v\_weapon\_count

FROM Weapon

WHERE Id = p\_id and model in ('HK416', 'AR-15', 'Ak-47', 'Desert Eagle', 'M4 Carbine', 'Glock 19', 'MP5', 'Uzi');

IF v\_weapon\_count > 0 THEN

v\_danger\_level := v\_danger\_level + 10;

END IF;

*-- Check if the person has incriminations and sum the casualties*

SELECT SUM(Casualties) INTO v\_incrim\_count FROM Incrimination WHERE Id = p\_id AND Casualties > 0;

IF v\_incrim\_count > 0 THEN

v\_danger\_level := v\_danger\_level + v\_incrim\_count;

END IF;

*-- Fetch the datum\_point and calculate the distance grade*

BEGIN

SELECT Datum\_Point INTO v\_datum\_point

FROM House

WHERE Id = p\_id;

v\_distance\_grade := Calculate\_Distance(v\_datum\_point);

v\_danger\_level := v\_danger\_level + v\_distance\_grade;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

v\_distance\_grade := 0; *-- Default value if no datum\_point found*

END;

RETURN v\_danger\_level;

EXCEPTION

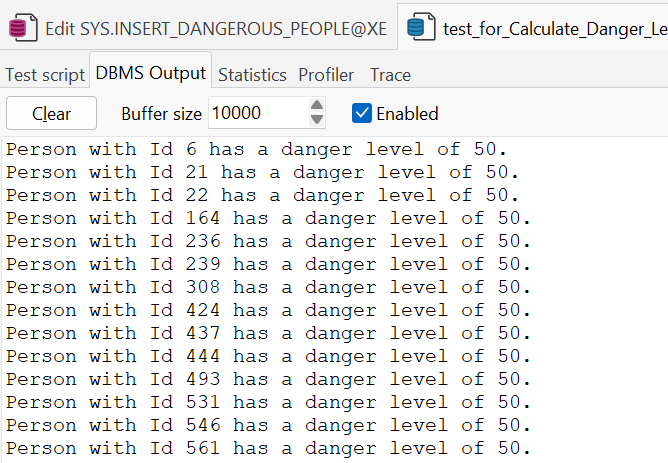
WHEN OTHERS THEN

*-- Handle unexpected errors and ensure a return value*

RETURN v\_danger\_level;

END Calculate\_Danger\_Level;

#### מדגם תוצאות ההרצה:

תמונה שמכילה טקסט, צילום מסך, תוכנה, סמל מחשב

התיאור נוצר באופן אוטומטיטסט שמוצא את כל האנשים שקיבלו ציון מסוכנות של מספר שנבחר (50 במקרה זה – מי שגר בדרום הרצועה למשל) תוך שימוש בCRUSER.

### 3.1.3 הפונקציה Insert\_Dangerous\_People

#### תיאור מילולי

הפרוצדורה Insert\_Dangerous\_People אחראית על הכנסה של 100 האנשים המסוכנים ביותר מתוך טבלת Humen לטבלת Dangerous\_People\_Operations, יחד עם רמת הסכנה והמיקום שלהם.

לאחר מכן הפונקציה תקרא לפרוצדורה RunDangerousPeopleOperation אשר תפעיל ref\_cursor למציאת כל הפעילים הרלוונטים, תחשב את הדירוג של כל אחד בלולאה ותכניס אותם לטבלת התוצאה Dangerous\_People\_Operations - טבלת היעדים המסוכנים למעצר.

#### קוד הSQL

CREATE OR REPLACE PROCEDURE Insert\_Dangerous\_People IS

v\_danger\_level INT;

v\_operation\_id INT := 1;

v\_location VARCHAR2(50);

BEGIN

FOR rec IN (SELECT Id FROM (

SELECT Id, ROW\_NUMBER() OVER (ORDER BY Calculate\_Danger\_Level(Id) DESC) AS rn

FROM Humen

) WHERE rn <= 100)

LOOP

v\_danger\_level := Calculate\_Danger\_Level(rec.Id);

BEGIN

*-- Find the location of the person*

SELECT Location INTO v\_location FROM Incrimination WHERE Id = rec.Id AND ROWNUM = 1;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

v\_location := 'Unknown';

END;

*-- Insert into Dangerous\_People\_Operations table*

INSERT INTO Dangerous\_People\_Operations (Operation\_Id, Person\_Id, Danger\_Level, Location)

VALUES (v\_operation\_id, rec.Id, v\_danger\_level, v\_location);

v\_operation\_id := v\_operation\_id + 1;

END LOOP;

END Insert\_Dangerous\_People;

מדגם תוצאות ההרצה יתואר יחד עם הפונקציה הראשית (Main\_Procedure)

### 3.1.4 הפונקציה הראשית Main\_Procedure

#### תיאור מילולי

פרוצדורה זו אחראית על הכנסת הנתונים לטבלה Dangerous\_People\_Operations. הפונקציה עושה זו על ידי קריאה לפונקציות הדירוג ופרוצדורת ההכנסה לטבלת היעד שתוארו לעיל.   
בסיום הריצה הפונקציה עוברת באמצעות REF CURSORעל טבלת היעד Dangerous\_People\_Operations, לצורך הדפסת פלט רצף התוכניות וביצוע בקרה על ההכנסה לטבלה.

#### קוד הSQL

CREATE OR REPLACE PROCEDURE Main\_Procedure IS

v\_total\_inserted INT;

v\_cursor SYS\_REFCURSOR;

v\_id INT;

v\_danger\_level INT;

v\_location VARCHAR2(50);

v\_operation\_id INT := 1;

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Procedure started.');

BEGIN

*-- Insert\_Dangerous\_People קריאה לפרוצדורה*

Insert\_Dangerous\_People;

*-- Dangerous\_People\_Operations :פתיחת קורסור לקריאה מטבלת היעד*

OPEN v\_cursor FOR

SELECT Operation\_Id, Person\_Id, Danger\_Level, Location

FROM Dangerous\_People\_Operations;

*-- v\_cursor :מעבר על ערכי הטבלה אחד אחר השני בלולאה ע"י המצביע*

LOOP

FETCH v\_cursor INTO v\_operation\_id, v\_id, v\_danger\_level, v\_location;

EXIT WHEN v\_cursor%NOTFOUND;

*-- הדפסת ערכי טבלת היעד*

DBMS\_OUTPUT.PUT\_LINE('Operation ID: ' || v\_operation\_id || ', Person ID: ' || v\_id || ', Danger Level: ' || v\_danger\_level || ', Location: ' || v\_location);

END LOOP;

CLOSE v\_cursor;

EXCEPTION

WHEN OTHERS THEN

*-- טיפול בשגיאות והצגת הודעה מתאימה*

DBMS\_OUTPUT.PUT\_LINE('Error occurred: ' || SQLERRM || ' at ' || TO\_CHAR(SYSTIMESTAMP, 'YYYY-MM-DD HH24:MI:SS.FF'));

RAISE;

END;

*-- בדיקת מספר האנשים שהוכנסו*

SELECT COUNT(\*) INTO v\_total\_inserted FROM Dangerous\_People\_Operations;

DBMS\_OUTPUT.PUT\_LINE('Total number of dangerous people inserted: ' || v\_total\_inserted);

*-- הצגת הודעת סיום מוצלחת*

DBMS\_OUTPUT.PUT\_LINE('Dangerous people insertion completed successfully.');

EXCEPTION

WHEN OTHERS THEN

*-- טיפול בשגיאות כלשהן והצגת הודעה מתאימה*

DBMS\_OUTPUT.PUT\_LINE('Unexpected error occurred: ' || SQLERRM || ' at ' || TO\_CHAR(SYSTIMESTAMP, 'YYYY-MM-DD HH24:MI:SS.FF'));

RAISE;

END Main\_Procedure;

#### מדגם תוצאות ההרצה

הכנסת הפעילים אשר קיבלו את דירוג המסוכנות הגבוה ביותר

תמונה שמכילה טקסט, צילום מסך, גופן, מספר

התיאור נוצר באופן אוטומטי

תמונה שמכילה טקסט, צילום מסך, מספר, גופן

התיאור נוצר באופן אוטומטי

תמונה שמכילה טקסט, צילום מסך, מספר, גופן

התיאור נוצר באופן אוטומטי