

דוח ג' מיני"פ בסנ"ת

1. תיאור מילולי של התוכנית

פונקציה זו נקראת `get_trip_details` והיא מקבלת כפרמטר את מזהה מנהל הטיול (trip_manager_id). הפונקציה מחזירה ref cursor עם פרטי הטיול ופרטי הסטודנטים הקשורים לטיול.

שלבים בפונקציה:

1. שליפת פרטי הטיול:
 - הפונקציה מתחילה בשליפת תאריך הטיול, נושא הטיול והמיקומים מהטבלה `Trips` על פי מזהה מנהל הטיול.
 - אם לא נמצא טיול עם מזהה מנהל זה, הפונקציה זורקת חריגה מותאמת אישית.
2. עדכון נושא הטיול:
 - הפונקציה מעדכנת את נושא הטיול לנושא חדש (המשתמש מכניס).
 - אם לא נעשה עדכון, הפונקציה מדפיסה הודעה מתאימה.
3. פתיחת cursor להחזרת פרטי הטיול ופרטי הסטודנטים:
 - הפונקציה פותחת cursor המחזיר את פרטי הטיול ופרטי הסטודנטים הקשורים לטיול לפי מזהה מנהל הטיול.
 - הפונקציה מחזירה את ה-cursor.
4. טיפול בשגיאות:
 - הפונקציה מדפיסה הודעה אם לא נמצא טיול עם מזהה המנהל הנתון.
 - הפונקציה מדפיסה הודעה במקרה של שגיאות אחרות ומחזירה-NULL.

הקוד של הפונקציה:

```
CREATE OR REPLACE FUNCTION get_trip_details(p_trip_manager_id IN Trips.Trip_manager_id%TYPE)
RETURN SYS_REFCURSOR
IS
    trip_cursor SYS_REFCURSOR; -- Ref cursor to return trip details and student details
    v_trip_date Trips.Trip_date%TYPE; -- Variables to hold trip details
    v_trip_topic Trips.Trip_topic%TYPE;
    v_locations Trips.Locations%TYPE;
    v_updated_topic Trips.Trip_topic%TYPE := 'Diving'; -- New trip topic for update

    e_no_trip EXCEPTION; -- Custom exception for no trip found
    PRAGMA EXCEPTION_INIT(e_no_trip, -20001); -- Associate error code with exception

BEGIN
    -- Fetch trip details for the given manager ID
    BEGIN
        SELECT Trip_date, Trip_topic, Locations
        INTO v_trip_date, v_trip_topic, v_locations
        FROM Trips
        WHERE Trip_manager_id = p_trip_manager_id;
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            RAISE e_no_trip; -- Raise exception if no trip found
    END;

    -- Perform an update on the trip topic
    UPDATE Trips
    SET Trip_topic = v_updated_topic
    WHERE Trip_manager_id = p_trip_manager_id;

    -- Check if the update was successful
    IF SQL%ROWCOUNT = 0 THEN
        DBMS_OUTPUT.PUT_LINE('No rows were updated.');
```

Amir hay Tamar : 322879339

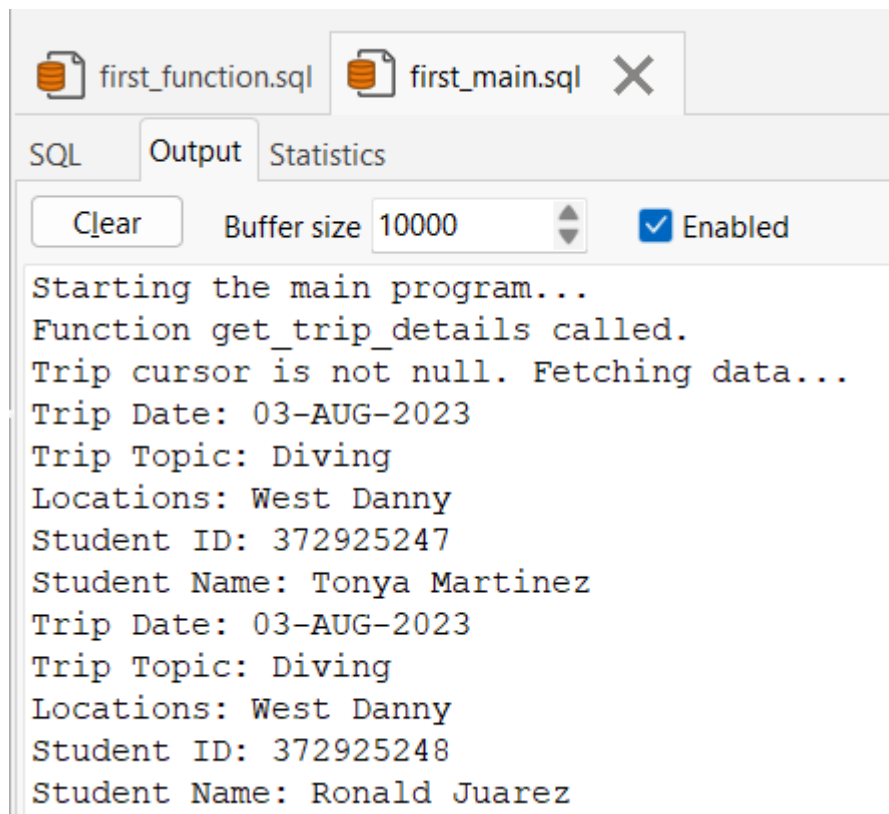
Ori Shoshana : 324891134

```
-- Open the ref cursor for trip details and associated student details
OPEN trip_cursor FOR
SELECT
    t.Trip_date AS Trip_Date,
    t.Trip_topic AS Trip_Topic,
    t.Locations AS Locations,
    s.Student_id AS Student_ID,
    s.Student_Name AS Student_Name
FROM
    Trips t
LEFT JOIN
    Going g ON t.Trip_manager_id = g.Trip_manager_id
LEFT JOIN
    Student_ s ON g.Student_id = s.Student_id
WHERE
    t.Trip_manager_id = p_trip_manager_id;

RETURN trip_cursor; -- Return the ref cursor containing trip details and associated student info

EXCEPTION
WHEN e_no_trip THEN
    DBMS_OUTPUT.PUT_LINE('No trip found with the given manager ID');
    RETURN NULL; -- Return NULL if no trip found
WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLERRM);
    RETURN NULL; -- Return NULL for any other exception
END get_trip_details;
/
```

הוכחה שהתוכנית אכן עבדה:



```
SQL first_function.sql first_main.sql X
SQL Output Statistics
Clear Buffer size 10000 Enabled
Starting the main program...
Function get_trip_details called.
Trip cursor is not null. Fetching data...
Trip Date: 03-AUG-2023
Trip Topic: Diving
Locations: West Danny
Student ID: 372925247
Student Name: Tonya Martinez
Trip Date: 03-AUG-2023
Trip Topic: Diving
Locations: West Danny
Student ID: 372925248
Student Name: Ronald Juarez
```

2. תיאור מילולי של התוכנית

פרוצדורה זו נקראת process_trip_updates ומטרתה לעדכן את הציונים (Grades) של טיולים על פי מזהה מנהל הטיול (trip_manager_id). הפרוצדורה מבצעת מספר שלבים הכוללים שליפת פרטי הטיול, עדכון הכיתה שיוצאת לטיול, מחיקת פעילות (Activity) מסוימת וטיפול בשגיאות.

שלבים בפרוצדורה:

1. שליפת פרטי הטיול:
 - הפרוצדורה משתמשת ב cursor-על מנת לשלוף את תאריך הטיול, נושא הטיול והמיקומים מהטבלה Trips על פי מזהה מנהל הטיול.
2. עדכון הכיתה היוצאת הטיול:
 - הפרוצדורה מעדכנת את הכיתה שיוצאת לטיול ל-7 (או כל ערך אחר בין 1 ל-12).
 - אם לא נעשה עדכון, הפרוצדורה מדפיסה הודעה מתאימה.
 - אם העדכון נעשה בהצלחה, הפרוצדורה מדפיסה הודעה מתאימה.
3. מחיקת פעילות:
 - הפרוצדורה מוחקת פעילות עם מזהה 374 מהטבלה Activity.
 - במידה ויש שגיאה במהלך המחיקה, מתבצע rollback והודעה מתאימה מוצגת.
4. טיפול בשגיאות:
 - אם לא נמצא טיול עם מזהה המנהל הנתון, הפרוצדורה מדפיסה הודעה מתאימה.
 - במקרה של שגיאות אחרות, מתבצע rollback והודעה מתאימה מוצגת.

הקוד של הפרוצדורה:

```
CREATE OR REPLACE PROCEDURE process_trip_updates(p_trip_manager_id IN Trips.Trip_manager_id%TYPE)
IS
    -- Cursor declaration
    CURSOR trip_cursor IS
        SELECT Trip_date, Trip_topic, Locations
        FROM Trips
        WHERE Trip_manager_id = p_trip_manager_id;

    -- Variables to hold trip details
    v_trip_date Trips.Trip_date%TYPE;
    v_trip_topic Trips.Trip_topic%TYPE;
    v_locations Trips.Locations%TYPE;
    v_updated_grade Trips.Grade%TYPE := 8; -- Example: Set default grade to 8

    -- Exception declaration
    e_no_trip EXCEPTION;
    PRAGMA EXCEPTION_INIT(e_no_trip, -20001);

BEGIN
    -- Loop through trips for the given manager ID
    FOR trip_rec IN trip_cursor LOOP
        v_trip_date := trip_rec.Trip_date;
        v_trip_topic := trip_rec.Trip_topic;
        v_locations := trip_rec.Locations;

        -- Perform an update on the grade of the trip
        UPDATE Trips
        SET Grade = LEAST(GREATEST(v_updated_grade, 1), 12) -- Ensures grade is between 1 and 12
        WHERE Trip_manager_id = p_trip_manager_id;

        -- Check if the update was successful
        IF SQL%ROWCOUNT = 0 THEN
            DBMS_OUTPUT.PUT_LINE('No rows were updated.');
```

Amir hay Tamar : 322879339
Ori Shoshana : 324891134

```
-- Example of using exception handling
BEGIN
  -- Additional DML command (delete example)
  DELETE FROM Activity
  WHERE Activity_id = 500;

  -- Commit transaction
  COMMIT;
EXCEPTION
  WHEN OTHERS THEN
    ROLLBACK;
    DBMS_OUTPUT.PUT_LINE('Error in nested block: ' || SQLERRM);
END;
END LOOP;

EXCEPTION
  WHEN e_no_trip THEN
    DBMS_OUTPUT.PUT_LINE('No trip found for the given manager ID.');
```

```
WHEN OTHERS THEN
  DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLERRM);
  ROLLBACK; -- Rollback in case of any error
END process_trip_updates;
/
```

הוכחה שהתוכנית אכן עבדה:

```
Calling process_trip_updates procedure...
Grade updated successfully.
Error in nested block: ORA-02292: integrity constraint (SYS.SYS_C009127) violated - child record found
Procedure process_trip_updates called.
Main program completed.
```

(ניתן לראות שהודפסה הודעה שהכיתה עודכנה וכמו כן ניתן לראות שיש בעיה במחיקה מאחר וניסינו למחוק רשימה שמקושרת לרשימות אחרות – שזה מצוין זה מראה שהבדיקה של המקרים החריגים עובדת – בכל מקרה אם רוצים שתתאפשר מחיקה גם במקרה כזה פשוט מוסיפים את האופציה של cascade)

3. תיאור מילולי של התוכנית

תוכנית זו היא תוכנית ראשית שמבצעת את השלבים הבאים:

1. מגדירה מזהה מנהל טיול.(trip_manager_id)
2. קוראת לפונקציה get_trip_details כדי לקבל פרטי טיול ופרטי סטודנטים קשורים.
3. מציגה את פרטי הטיול והסטודנטים על המסך.
4. קוראת לפרוצדורה process_trip_updates כדי לעדכן את הכיתה שיוצאת לטיול ולבצע פעולות נוספות.
5. מטפלת בשגיאות במידה ומתעוררות.

שלבים בתוכנית:

1. הגדרת משתנים:
 - o מזהה מנהל הטיול.(trip_manager_id)

- משתנים להחזיק את פרטי הטיול (trip_date, trip_topic, locations) ופרטי הסטודנטים (student_id, student_name).
- 2. הדפסת הודעה להתחלת התוכנית:
 - התוכנית מדפיסה הודעה שמודיעה על תחילת התוכנית הראשית.
- 3. קריאה לפונקציה: get_trip_details
 - קריאה לפונקציה עם מזהה מנהל הטיול הנתון ושמירת התוצאה ב-cursor.
 - הדפסת הודעה שמודיעה שהפונקציה נקראה.
- 4. בדיקה אם ה cursor אינו NULL
 - אם ה cursor אינו NULL, התוכנית מדפיסה הודעה ומתחילה לשלוף את הנתונים מה cursor.
 - לולאה שגורפת את הנתונים מה cursor ומדפיסה אותם על המסך:
 - תאריך הטיול
 - נושא הטיול
 - מיקומים
 - מזהה הסטודנט
 - שם הסטודנט
 - סגירת ה cursor לאחר סיום השליפה.
- 5. קריאה לפרוצדורה: process_trip_updates
 - הדפסת הודעה שמודיעה שהפרוצדורה נקראת.
 - קריאה לפרוצדורה עם מזהה מנהל הטיול הנתון.
 - הדפסת הודעה שמודיעה שהפרוצדורה נקראה.
- 6. טיפול בשגיאות:
 - במקרה של NO_DATA_FOUND, הנתונים עבור מזהה מנהל הטיול הנתון.
 - במקרה של שגיאה אחרת, התוכנית מדפיסה את הודעת השגיאה ומסוגרת את ה cursor אם הוא פתוח.

הקוד של התוכנית

```

DECLARE
p_trip_manager_id Trips.Trip_manager_id%TYPE := 219474001; -- Replace with the appropriate trip manager ID
trip_cursor SYS_REFCURSOR;
v_trip_date Trips.Trip_date%TYPE;
v_trip_topic Trips.Trip_topic%TYPE;
v_locations Trips.Locations%TYPE;
v_student_id Student.Student_id%TYPE;
v_student_name Student.Student_Name%TYPE;

BEGIN
  DBMS_OUTPUT.PUT_LINE('Starting the main program...');

  -- Call the get_trip_details function
  trip_cursor := get_trip_details(p_trip_manager_id);

  DBMS_OUTPUT.PUT_LINE('Function get_trip_details called.');
```

```

  -- Check if the cursor is not null
  IF trip_cursor IS NOT NULL THEN
    DBMS_OUTPUT.PUT_LINE('Trip cursor is not null. Fetching data...');
```

```

    -- Fetch the trip details and associated student details from the cursor
    LOOP
      FETCH trip_cursor INTO v_trip_date, v_trip_topic, v_locations, v_student_id, v_student_name;
      EXIT WHEN trip_cursor%NOTFOUND;
```

```

      -- Display the trip details and associated student details
      DBMS_OUTPUT.PUT_LINE('Trip Date: ' || TO_CHAR(v_trip_date, 'DD-MON-YYYY'));
      DBMS_OUTPUT.PUT_LINE('Trip Topic: ' || v_trip_topic);
      DBMS_OUTPUT.PUT_LINE('Locations: ' || v_locations);
      DBMS_OUTPUT.PUT_LINE('Student ID: ' || v_student_id);
      DBMS_OUTPUT.PUT_LINE('Student Name: ' || v_student_name);
    END LOOP;
  CLOSE trip_cursor;

```

Amir hay Tamar : 322879339
Ori Shoshana : 324891134

```
-- Call the process_trip_updates procedure
DBMS_OUTPUT.PUT_LINE('Calling process_trip_updates procedure...');
process_trip_updates(p_trip_manager_id);
DBMS_OUTPUT.PUT_LINE('Procedure process_trip_updates called.');
```

```
ELSE
  DBMS_OUTPUT.PUT_LINE('No trip details found.');
```

```
END IF;
```

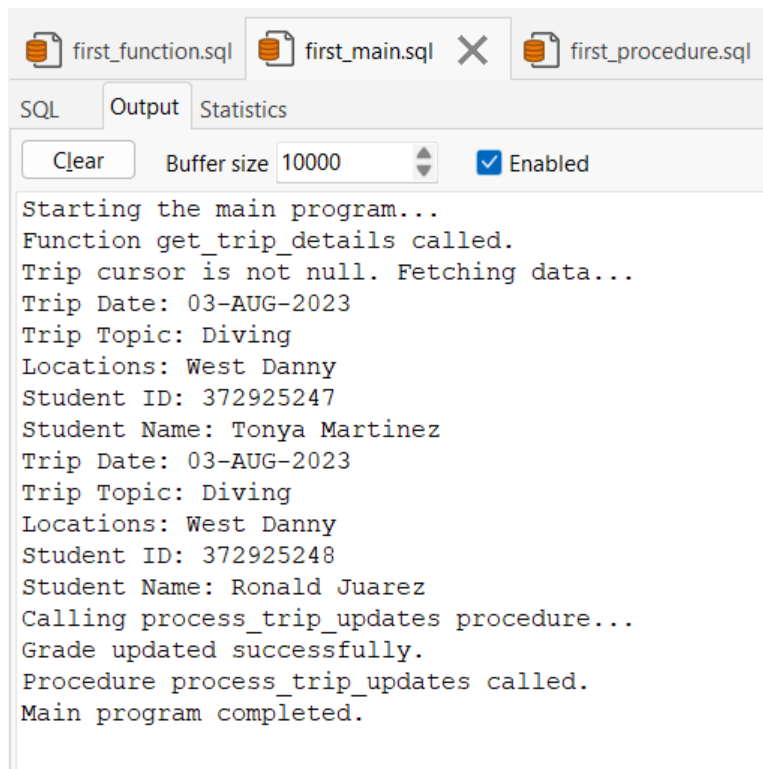
```
DBMS_OUTPUT.PUT_LINE('Main program completed.');
```

```
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE('No data found for the given manager ID.');
```

```
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLERRM);
    IF trip_cursor IS NOT NULL THEN
      CLOSE trip_cursor;
    END IF;
```

```
END;
```

```
/
```



```
first_function.sql first_main.sql first_procedure.sql
```

SQL Output Statistics

Clear Buffer size 10000 ☒ Enabled

```
Starting the main program...
Function get_trip_details called.
Trip cursor is not null. Fetching data...
Trip Date: 03-AUG-2023
Trip Topic: Diving
Locations: West Danny
Student ID: 372925247
Student Name: Tonya Martinez
Trip Date: 03-AUG-2023
Trip Topic: Diving
Locations: West Danny
Student ID: 372925248
Student Name: Ronald Juarez
Calling process_trip_updates procedure...
Grade updated successfully.
Procedure process_trip_updates called.
Main program completed.
```

הוכחה שהתוכנית אכן עבדה:

ניתן לראות שהפונקציות בעצם רצות
והדפסת הפלט שהוספנו כדי לוודא שהן
עובדות מתבצעת בהצלחה – מה שאומר
שהמייין רץ מצויין ומזמן הכל כמו שצריך.

4. תיאור מילולי של הפונקציה `get_activity_participants`.

פונקציה זו, `get_activity_participants`, נועדה להחזיר את פרטי המשתתפים בפעילות מסוימת על פי מזהה הפעילות הנתון (`activity_id`). בנוסף, היא מבצעת עדכון של מספר המשתתפים הנותרים בפעילות.

שלבים בפונקציה:

1. שליפת פרטי הפעילות:
 - הפונקציה משתמשת ב SQL- כדי לשלוף את שם הפעילות, תאריך הפעילות וקיבולת הפעילות מטבלת `Activity`.
 - במקרה שלא נמצאה פעילות עם מזהה הפעילות הנתון, הפונקציה מעלה חריג מותאם אישית.
2. עדכון קיבולת הפעילות:
 - הפונקציה מעדכנת את קיבולת הפעילות על ידי הפחתת אחד ממספר המשתתפים.
 - אם העדכון לא מתבצע (אין שורות שעודכנו), הפונקציה מדפיסה הודעה מתאימה.
3. פתיחת `SYS_REFCURSOR` לפרטי המשתתפים:
 - הפונקציה פותחת רפרנס של `cursor` שמחזיר את פרטי המשתתפים הפעילים בפעילות מהטבלאות `Student\Active`.
4. טיפול בשגיאות:
 - אם לא נמצאה פעילות עם המזהה הנתון, הפונקציה מדפיסה הודעה ומחזירה `NULL`.
 - במקרה של שגיאה כללית, הפונקציה מדפיסה את הודעת השגיאה ומחזירה `NULL`.

```
CREATE OR REPLACE FUNCTION get_activity_participants(p_activity_id IN Activity.Activity_id%TYPE)
RETURN SYS_REFCURSOR
IS
    participants_cursor SYS_REFCURSOR; -- Ref cursor to return participant details
    v_activity_name Activity.Activity_name%TYPE; -- Variables to hold activity details
    v_date_of_activity Activity.Date_of_activity%TYPE;
    v_capacity Activity.capacity%TYPE;
    v_updated_capacity Activity.capacity%TYPE; -- Updated capacity

    e_no_activity EXCEPTION; -- Custom exception for no activity found
    PRAGMA EXCEPTION_INIT(e_no_activity, -20001); -- Associate error code with exception

    TYPE participant_record IS RECORD ( -- Define a record type for participant details
        Student_id Student.Student_id%TYPE,
        Student_Name Student.Student_Name%TYPE,
        Grade Student.Grade%TYPE,
        Address Student.Address%TYPE
    );

    v_participant participant_record; -- Declare a variable of the record type

BEGIN
    -- Fetch activity details for the given activity ID
    BEGIN
        SELECT Activity_name, Date_of_activity, capacity
        INTO v_activity_name, v_date_of_activity, v_capacity
        FROM Activity
        WHERE Activity_id = p_activity_id;
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            RAISE e_no_activity; -- Raise exception if no activity found
    END;

    -- Perform an update on the activity capacity
    v_updated_capacity := v_capacity - 1;
    UPDATE Activity
    SET capacity = v_updated_capacity
    WHERE Activity_id = p_activity_id;
```

הקוד של

הפונקציה:

Amir hay Tamar : 322879339

Ori Shoshana : 324891134

```
-- Check if the update was successful
IF SQL%ROWCOUNT = 0 THEN
    DBMS_OUTPUT.PUT_LINE('No rows were updated.');
```

-- Log message if no rows were updated

```
END IF;

-- Open the ref cursor for participant details
OPEN participants_cursor FOR
SELECT
    s.Student_id AS Student_ID,
    s.Student_Name AS Student_Name,
    s.Grade AS Grade,
    s.Address AS Address
FROM
    Active a
JOIN
    Student_ s ON a.Student_id = s.Student_id
WHERE
    a.Activity_id = p_activity_id;

RETURN participants_cursor; -- Return the ref cursor containing participant details

EXCEPTION
    WHEN e_no_activity THEN
        DBMS_OUTPUT.PUT_LINE('No activity found with the given ID');
        RETURN NULL; -- Return NULL if no activity found
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLERRM);
        RETURN NULL; -- Return NULL for any other exception
END get_activity_participants;
/
```

second_function.sql second_main.sql X

SQL Output Statistics

Clear

Buffer size 10000

☒ Enabled

Student Name: Brooke Pace

Grade: 10

Address: 12679 Andrew Street, South Amberton, OH 04629

Departure time updated successfully and In time set to YES for Transportation ID 1

Procedure update_transportation_departure_time called for Transportation ID 1

Student ID: 372925001

Student Name: Andrea Herrera

Grade: 11

Address: 720 Joseph Landing Suite 605, Debbiewview, WI 51229

Departure time updated successfully and In time set to YES for Transportation ID 1

Procedure update_transportation_departure_time called for Transportation ID 1

Student ID: 372925003

Student Name: Michelle Blake

Grade: 9

Address: USNS Lawrence, FPO AP 14435

Departure time updated successfully and In time set to YES for Transportation ID 1

Procedure update_transportation_departure_time called for Transportation ID 1

Student ID: 372925004

Student Name: Chelsea Cain

Grade: 2

Address: PSC 0753, Box 3698, APO AA 76482

Departure time updated successfully and In time set to YES for Transportation ID 1

Procedure update_transportation_departure_time called for Transportation ID 1

Student ID: 372925005

Student Name: Jonathan Jackson

Grade: 3

Address: 289 Samantha Vista, Tylerburgh, PW 06359

Departure time updated successfully and In time set to YES for Transportation ID 1

הוכחה שהתוכנית אכן עבדה:

5. תיאור מילולי של הפרוצדורה update_transportation_departure_time

פרוצדורה זו, update_transportation_departure_time, נועדה לעדכן את זמן היציאה של אמצעי התחבורה על פי מזהה אמצעי התחבורה הנתון. (p_transportation_id) בנוסף, היא מסמנת את זמן היציאה כ-"כן" בטבלת Transportation.

שלבים בפרוצדורה:

1. שליפת זמן היציאה הנוכחי:
 - הפרוצדורה משתמשת ב SQL-כדי לשלוף את זמן היציאה הנוכחי (departure_time) מטבלת Transportation עבור מזהה אמצעי התחבורה הנתון.
 - במקרה שלא נמצאה רשומת אמצעי תחבורה עם המזהה הנתון, הפרוצדורה מעלה חריג מותאם אישית.
2. חישוב זמן היציאה המעודכן:
 - הפרוצדורה מחשבת את זמן היציאה המעודכן על ידי הוספת שעה אחת לזמן היציאה הנוכחי.
3. עדכון זמן היציאה וסימון הזמן בטבלה:
 - הפרוצדורה מעדכנת את זמן היציאה (departure_time) ואת סימון הזמן (In_time) ל-"כן" בטבלת Transportation עבור מזהה אמצעי התחבורה הנתון.
 - אם העדכון לא מתבצע (אין שורות שעודכנו), הפרוצדורה מדפיסה הודעה מתאימה.
 - אם העדכון מתבצע בהצלחה, הפרוצדורה מדפיסה הודעה שמודיעה על כך.
4. טיפול בשגיאות:
 - אם לא נמצאה רשומת אמצעי תחבורה עם המזהה הנתון, הפרוצדורה מעלה שגיאה מותאמת אישית ומדפיסה הודעה.
 - במקרה של שגיאה כללית, הפרוצדורה מדפיסה את הודעת השגיאה.

הקוד של הפרוצדורה

```
CREATE OR REPLACE PROCEDURE update_transportation_departure_time (
    p_transportation_id IN INT
)
IS
    v_current_departure_time Transportation.departure_time%TYPE;
    v_updated_departure_time DATE;


    e_no_transportation EXCEPTION;
    PRAGMA EXCEPTION_INIT(e_no_transportation, -20001);
BEGIN
    -- Fetch current departure time for the given transportation ID
    SELECT departure_time
    INTO v_current_departure_time
    FROM Transportation
    WHERE Transportation_id = p_transportation_id;

    -- Example: Calculate updated departure time (e.g., add 1 hour to current time)
    v_updated_departure_time := v_current_departure_time + INTERVAL '1' HOUR;

    -- Update the departure time and set In_time to 'YES' in the Transportation table
    UPDATE Transportation
    SET departure_time = v_updated_departure_time,
        In_time = 'Yes'
    WHERE Transportation_id = p_transportation_id;

    -- Check if any rows were updated
    IF SQL%ROWCOUNT = 0 THEN
        DBMS_OUTPUT.PUT_LINE('No rows were updated for Transportation ID ' || p_transportation_id);
    ELSE
        DBMS_OUTPUT.PUT_LINE('Departure time updated successfully and In time set to YES for Transportation ID ' || p_transportation_id);
    END IF;

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RAISE_APPLICATION_ERROR(-20001, 'No transportation found for ID ' || p_transportation_id);
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
END update_transportation_departure_time;
/
```

הוכחה שהתוכנית אכן עובדת: 

```

Departure time updated successfully and In time set to YES for Transportation ID 1
Procedure update_transportation_departure_time called for Transportation ID 1
Main program for get_activity_participants completed.

```

000 & 35:1

ori@XE AS SYSDBA [15:55:37] Done in 0.044 seconds

6. תיאור מילולי של התוכנית הראשית

תוכנית זו נועדה לקרוא את הפונקציה `get_activity_participants` ולקבל את פרטי המשתתפים בפעילות נתונה, ולאחר מכן לקרוא לפרוצדורה `update_transportation_departure_time` ולעדכן את זמן היציאה של אמצעי התחבורה עבור כל משתתף שנמצא.

שלבים בתוכנית:

1. הכרזת משתנים:
 - מזהה הפעילות (`activity_id`), משתנים להחזקת פרטי המשתתפים (כגון מזהה סטודנט, שם סטודנט, כיתה וכתובת), ומשתנה עבור מזהה אמצעי התחבורה (`transportation_id`).
2. התחלת התוכנית:
 - הודעה על תחילת התוכנית הראשית באמצעות `DBMS_OUTPUT.PUT_LINE`.
3. קריאה לפונקציה `get_activity_participants`:
 - קריאה לפונקציה עם מזהה הפעילות הנתון.
 - הדפסת הודעה לאחר קריאת הפונקציה.
4. בדיקת הרפרנס של ה-cursor:
 - בדיקה אם הרפרנס של ה-cursor אינו NULL.
 - אם ה-cursor אינו NULL, לולאת LOOP לשליפת פרטי המשתתפים והדפסתם.
 - קריאה לפרוצדורה `update_transportation_departure_time` עבור כל משתתף שנמצא והדפסת הודעה מתאימה.
5. סגירת ה-cursor:
 - סגירת ה-cursor לאחר השלמת הלולאה.
6. הודעות שגיאה:
 - טיפול בשגיאות שונות והדפסת הודעות שגיאה מתאימות.
 - סגירת ה-cursor במקרה של שגיאה.

הקוד של התוכנית הראשית

```

DECLARE
    p_activity_id Activity.Activity_id%TYPE := 1; -- Replace with the appropriate activity ID
    participants_cursor SYS_REFCURSOR;
    v_student_id Student.Student_id%TYPE;
    v_student_name Student.Student_Name%TYPE;
    v_grade Student.Grade%TYPE;
    v_address Student.Address%TYPE;
    v_activity_name Activity.Activity_name%TYPE;
    v_date_of_activity Activity.Date_of_activity%TYPE;
    v_transportation_id Transportation.Transportation_id%TYPE := 1; -- Replace with appropriate Transportation_id

BEGIN
    DBMS_OUTPUT.PUT_LINE('Starting the main program for get_activity_participants...');

    -- Call the get_activity_participants function
    participants_cursor := get_activity_participants(p_activity_id);

    DBMS_OUTPUT.PUT_LINE('Function get_activity_participants called.');
```

Amir hay Tamar : 322879339

Ori Shoshana : 324891134

```
-- Fetch the participant details from the cursor
LOOP
  FETCH participants_cursor INTO v_student_id, v_student_name, v_grade, v_address;
  EXIT WHEN participants_cursor%NOTFOUND;

  -- Display the participant details
  DBMS_OUTPUT.PUT_LINE('Student ID: ' || v_student_id);
  DBMS_OUTPUT.PUT_LINE('Student Name: ' || v_student_name);
  DBMS_OUTPUT.PUT_LINE('Grade: ' || v_grade);
  DBMS_OUTPUT.PUT_LINE('Address: ' || v_address);

  -- Call the procedure to update transportation departure time
  update_transportation_departure_time(v_transportation_id);
  DBMS_OUTPUT.PUT_LINE('Procedure update_transportation_departure_time called for Transportation ID ' || v_transportation_id);
END LOOP;
CLOSE participants_cursor;
ELSE
  DBMS_OUTPUT.PUT_LINE('No participant details found.');
```

```
END IF;

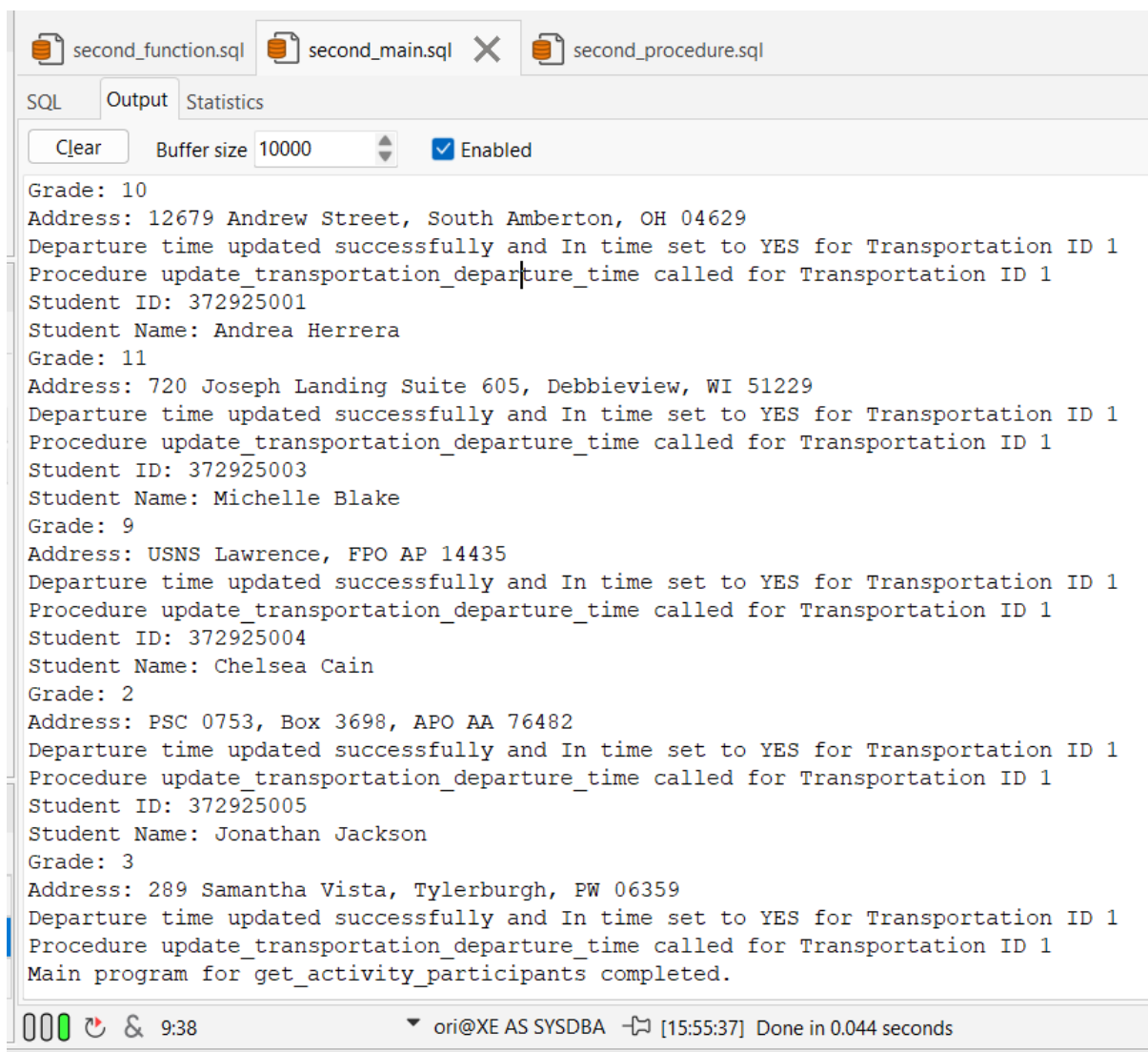
DBMS_OUTPUT.PUT_LINE('Main program for get_activity_participants completed.');
```

```
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE('No data found for the given activity ID.');
```

```
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLERRM);
    IF participants_cursor IS NOT NULL THEN
      CLOSE participants_cursor;
    END IF;
END;
```

```
/
```

הוכחה שהתוכנית עבדה: (הריצה בהצלחה את הפונקציה והפרוצדורה והוציאה פלט ראוי)



```
second_function.sql second_main.sql second_procedure.sql
```

```
SQL Output Statistics
```

Clear Buffer size 10000 Enabled

```
Grade: 10
Address: 12679 Andrew Street, South Amberton, OH 04629
Departure time updated successfully and In time set to YES for Transportation ID 1
Procedure update_transportation_departure_time called for Transportation ID 1
Student ID: 372925001
Student Name: Andrea Herrera
Grade: 11
Address: 720 Joseph Landing Suite 605, Debbievew, WI 51229
Departure time updated successfully and In time set to YES for Transportation ID 1
Procedure update_transportation_departure_time called for Transportation ID 1
Student ID: 372925003
Student Name: Michelle Blake
Grade: 9
Address: USNS Lawrence, FPO AP 14435
Departure time updated successfully and In time set to YES for Transportation ID 1
Procedure update_transportation_departure_time called for Transportation ID 1
Student ID: 372925004
Student Name: Chelsea Cain
Grade: 2
Address: PSC 0753, Box 3698, APO AA 76482
Departure time updated successfully and In time set to YES for Transportation ID 1
Procedure update_transportation_departure_time called for Transportation ID 1
Student ID: 372925005
Student Name: Jonathan Jackson
Grade: 3
Address: 289 Samantha Vista, Tylerburgh, PW 06359
Departure time updated successfully and In time set to YES for Transportation ID 1
Procedure update_transportation_departure_time called for Transportation ID 1
Main program for get_activity_participants completed.
```

9:38 ori@XE AS SYSDBA [15:55:37] Done in 0.044 seconds