קרן קיימת לישראל
Forests

קרן קימת לישראל

K K L - J N F

Shani Zegal 673692804
Rachelli Adler 213836687

שלב ג

https://github.com/RachelliAA/DBProject-6687-2804/releases/tag/v3.0.0

Table of contents:

PROGRAM #1:

Previously, the fire prevention actions table had 400 entries, including duplicate action names with different costs. We have decided to delete all duplicates and decide on a specific cost for each action according to the level of protection (i.e. for low protection - 10.3 [price is in 1000 nis per km^2]).
Secondly, we wanted to test a prevention action to see if it actually helps with preventing/stopping forest fires. To do that, we added a small planting project to an existing forest,added the action we want to test to the forest, and assigned a research station associated with said forest to study the aforementioned action.

Procedure to update prevention actions table:

```
CREATE OR REPLACE PROCEDURE updateActions IS
  low_cost CONSTANT NUMBER := 10.3;
  medium_cost CONSTANT NUMBER := 40.5;
  strong_cost CONSTANT NUMBER := 70.1;
  new_action_cost NUMBER;
BEGIN
  -- Delete duplicates based on action_name
  DELETE FROM prevention_actions
  WHERE ROWID NOT IN (
    SELECT MIN(ROWID)
    FROM prevention_actions
    GROUP BY action_name
  );

  DBMS_OUTPUT.PUT_LINE('Duplicate actions deleted');

  -- Loop through prevention_actions to update costs
  FOR action_rec IN (SELECT * FROM prevention_actions) LOOP
    -- Determine the cost based on action_duration
    new_action_cost := CASE
      WHEN action_rec.action_duration LIKE 'low%' THEN low_cost
      WHEN action_rec.action_duration LIKE 'medium%' THEN medium_cost
      WHEN action_rec.action_duration LIKE 'strong%' THEN strong_cost
      ELSE 0
    END;

    UPDATE prevention_actions
    SET action_id = rownum;
    -- Update the cost for the current action
    UPDATE prevention_actions
    SET cost = new_action_cost
    WHERE action_id = action_rec.action_id;
  END LOOP;

  DBMS_OUTPUT.PUT_LINE('Costs updated');

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

Function to add planting project and assign a research station to study action:

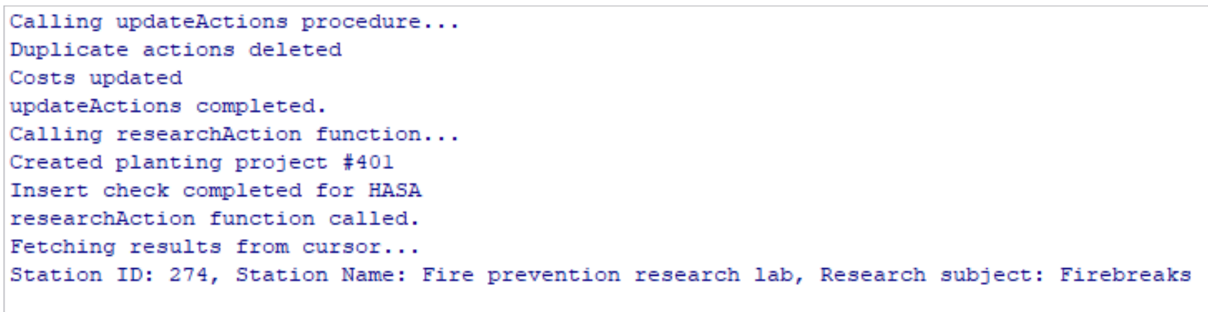
```
CREATE OR REPLACE FUNCTION researchAction(f_id forests.forest_id%type, a_id
prevention_actions.action_id%type)
RETURN SYS_REFCURSOR IS
  TYPE ref_cursor_type IS REF CURSOR;
  Result ref_cursor_type;
  action VARCHAR2(50);
  v_planting_id INTEGER;
BEGIN
  -- Find the next available planting_id
  SELECT COALESCE(MAX(planting_id), 0) + 1 INTO v_planting_id FROM
Tree_Planting_Projects;
  -- New planting project
  INSERT INTO TREE_PLANTING_PROJECTS (PLANTING_ID, START_DATE,
END_DATE, FOREST_ID)
  VALUES (v_planting_id, TO_DATE('04-Jul-27', 'DD-Mon-YY'), TO_DATE('25-Aug-24',
'DD-Mon-YY'), f_id);
  DBMS_OUTPUT.PUT_LINE('Created planting project #' || v_planting_id);
  -- Add action to the forest
  -- Use a MERGE statement to insert only if the combination does not exist
  MERGE INTO HASA t
  USING (SELECT f_id AS forest_id, a_id AS action_id FROM DUAL) s
  ON (t.forest_id = s.forest_id AND t.action_id = s.action_id)
  WHEN NOT MATCHED THEN
    INSERT (forest_id, action_id)
    VALUES (s.forest_id, s.action_id);

  DBMS_OUTPUT.PUT_LINE('Insert check completed for HASA');

  -- Select action name into action variable
  SELECT action_name INTO action
  FROM prevention_actions
  WHERE action_id = a_id;

  -- Update the forest's research station subject and name
  UPDATE research_stations
  SET research_subject = action,
      station_name = 'Fire prevention research lab'
  WHERE rowid IN (
    SELECT rowid
    FROM research_stations
    WHERE forest_id = f_id
    AND ROWNUM = 1
  );

  -- Open the cursor for the result set
  OPEN Result FOR
    SELECT *
    FROM research_stations
    WHERE research_subject = action;

  RETURN Result;
END researchAction;
```

Main:

```
DECLARE
  -- Variables
  random_forest NUMBER;
  random_action NUMBER;
  min_value INTEGER := 1;
  max_forest INTEGER := 400;
  max_action INTEGER := 10;
  v_cursor SYS_REFCURSOR;
  v_station_record research_stations%ROWTYPE;

  -- Exception to handle table lock
  e_table_busy EXCEPTION;
  PRAGMA EXCEPTION_INIT(e_table_busy, -54);

BEGIN
  DBMS_OUTPUT.PUT_LINE('Calling updateActions procedure...');
  -- Call procedure updateActions that updates prevention_actions table
  updateActions();
  DBMS_OUTPUT.PUT_LINE('updateActions completed.');

  /*
  -- Drop and recreate the HasA table
  BEGIN
    DBMS_OUTPUT.PUT_LINE('Dropping table HasA...');
    EXECUTE IMMEDIATE 'DROP TABLE HasA';
    DBMS_OUTPUT.PUT_LINE('Table HasA dropped.');
  EXCEPTION
    WHEN e_table_busy THEN
      DBMS_OUTPUT.PUT_LINE('Table HasA is busy, retrying...');
      DBMS_LOCK.SLEEP(1); -- Wait for 1 second before retrying
      EXECUTE IMMEDIATE 'DROP TABLE HasA';
      DBMS_OUTPUT.PUT_LINE('Table HasA dropped after retry.');
    WHEN OTHERS THEN
      DBMS_OUTPUT.PUT_LINE('Error dropping table HasA: ' || SQLERRM);
  END;

  EXECUTE IMMEDIATE 'CREATE TABLE HasA (
                      forest_ID INT NOT NULL,
                      action_ID INT NOT NULL,
                      PRIMARY KEY (forest_ID, action_ID),
                      FOREIGN KEY (forest_ID) REFERENCES Forests(forest_ID),
                      FOREIGN KEY (action_ID) REFERENCES Prevention_Actions(action_ID)
                  )';
  DBMS_OUTPUT.PUT_LINE('Table HasA created successfully.');
  */

  -- Seed the random number generator (optional, but recommended for better randomness)
  DBMS_RANDOM.SEED(TO_NUMBER(TO_CHAR(SYSDATE, 'SSSSS')));

  -- Insert random data into HasA table
  FOR i IN 1..400 LOOP
    -- Generate random numbers within specified ranges
    random_forest := TRUNC(DBMS_RANDOM.VALUE(min_value, max_forest));
    random_action := TRUNC(DBMS_RANDOM.VALUE(min_value, max_action));

    BEGIN
      -- Attempt to insert
      INSERT INTO HASA (forest_id, action_id) VALUES (random_forest, random_action);
    EXCEPTION
      WHEN DUP_VAL_ON_INDEX THEN
          DELETE FROM hasa
          WHERE forest_id = random_forest AND action_id = random_action;
      WHEN OTHERS THEN
        -- Handle any other exceptions
        DBMS_OUTPUT.PUT_LINE('Error inserting into HasA: ' || SQLERRM);
    END;
  END LOOP;


  -- Call function researchAction
  BEGIN
    DBMS_OUTPUT.PUT_LINE('Calling researchAction function...');
    v_cursor := researchAction(f_id => 214, a_id => 8);
    DBMS_OUTPUT.PUT_LINE('researchAction function called.');
  EXCEPTION
    WHEN OTHERS THEN
      DBMS_OUTPUT.PUT_LINE('Error calling researchAction: ' || SQLERRM);
  END;

  -- Display result
  -- Check if cursor is not null
  IF v_cursor IS NOT NULL THEN
    DBMS_OUTPUT.PUT_LINE('Fetching results from cursor...');
    -- Fetch and print the results
    LOOP
      FETCH v_cursor INTO v_station_record;
      EXIT WHEN v_cursor%NOTFOUND;
      DBMS_OUTPUT.PUT_LINE('Station ID: ' || v_station_record.station_id
                    || ', Station Name: ' || v_station_record.station_name
```

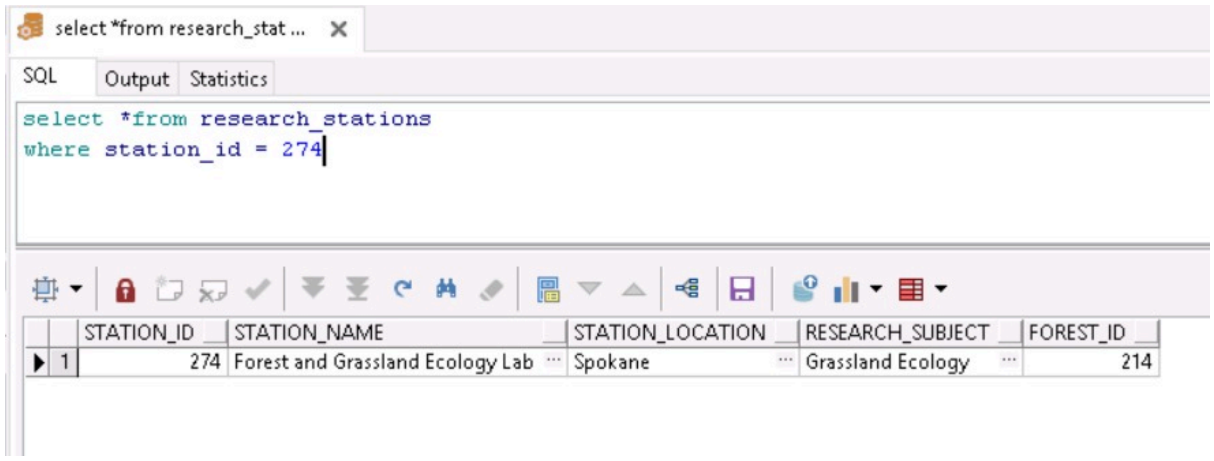
```
                    || ', Research subject: ' || v_station_record.research_subject);
    END LOOP;
    -- Close the cursor
    CLOSE v_cursor;
  ELSE
    DBMS_OUTPUT.PUT_LINE('v_cursor is null, no results to display.');
  END IF;

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

Output:



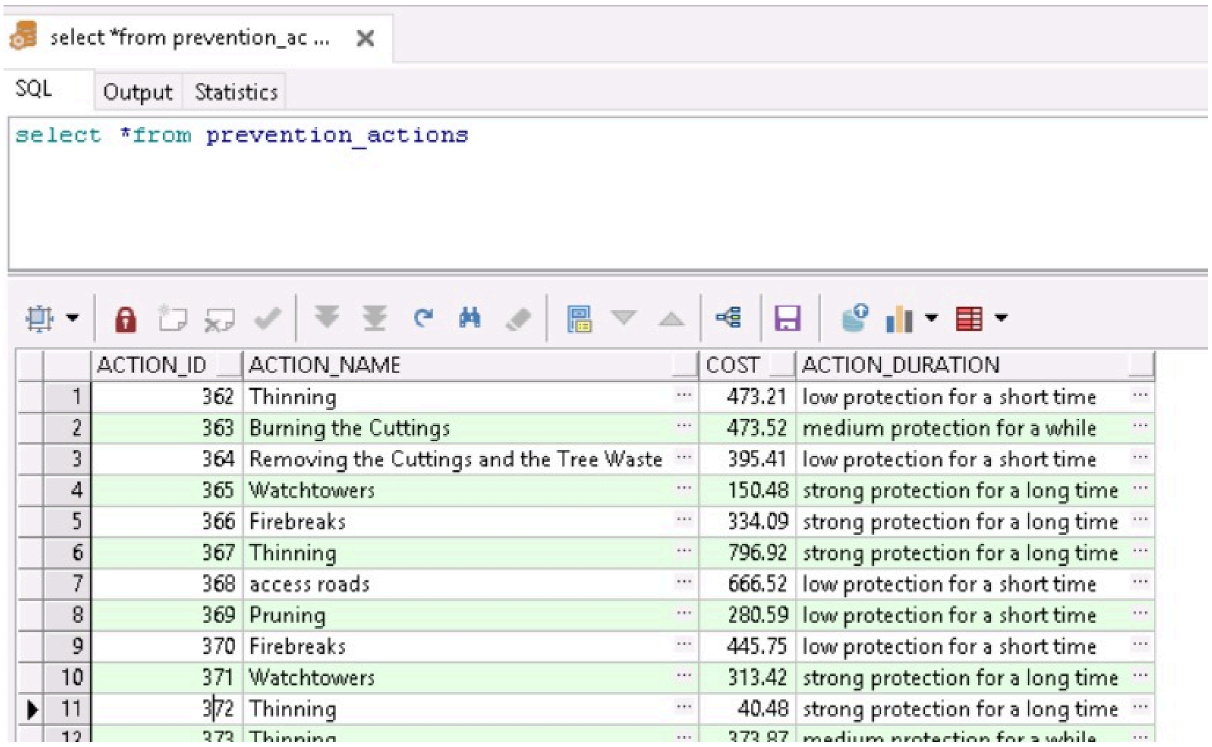
```
Calling updateActions procedure...
Duplicate actions deleted
Costs updated
updateActions completed.
Calling researchAction function...
Created planting project #401
Insert check completed for HASA
researchAction function called.
Fetching results from cursor...
Station ID: 274, Station Name: Fire prevention research lab, Research subject: Firebreaks
```

DB before:



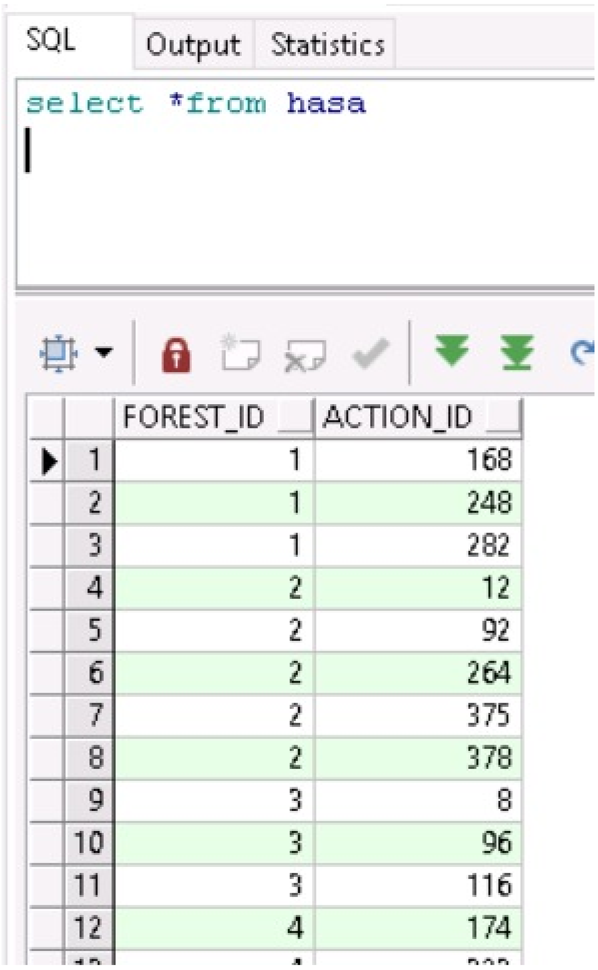
```
select *from research_stations
where station_id = 274
```

	STATION_ID	STATION_NAME	STATION_LOCATION	RESEARCH_SUBJECT	FOREST_ID
1	274	Forest and Grassland Ecology Lab	Spokane	Grassland Ecology	214



```
select *from prevention_actions
```

	ACTION_ID	ACTION_NAME	COST	ACTION_DURATION
1	362	Thinning	473.21	low protection for a short time
2	363	Burning the Cuttings	473.52	medium protection for a while
3	364	Removing the Cuttings and the Tree Waste	395.41	low protection for a short time
4	365	Watchtowers	150.48	strong protection for a long time
5	366	Firebreaks	334.09	strong protection for a long time
6	367	Thinning	796.92	strong protection for a long time
7	368	access roads	666.52	low protection for a short time
8	369	Pruning	280.59	low protection for a short time
9	370	Firebreaks	445.75	low protection for a short time
10	371	Watchtowers	313.42	strong protection for a long time
11	372	Thinning	40.48	strong protection for a long time
12	373	Thinning	373.87	medium protection for a while

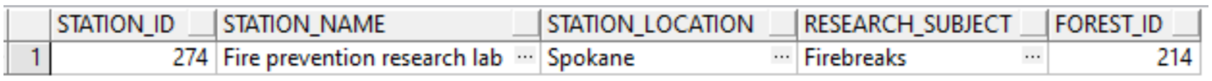


```
select *from hasa
```

	FOREST_ID	ACTION_ID
1	1	168
2	1	248
3	1	282
4	2	12
5	2	92
6	2	264
7	2	375
8	2	378
9	3	8
10	3	96
11	3	116
12	4	174
[illegible]	4	[illegible]

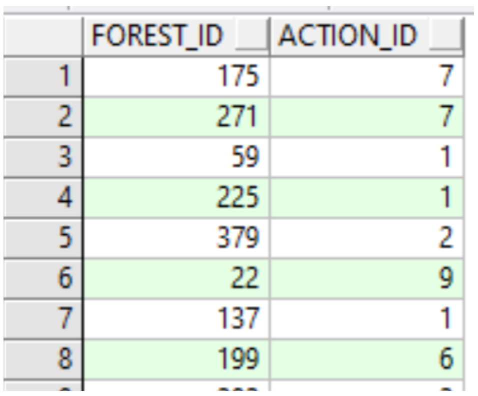
DB after:

Research station #274 now studies a fire prevention action:



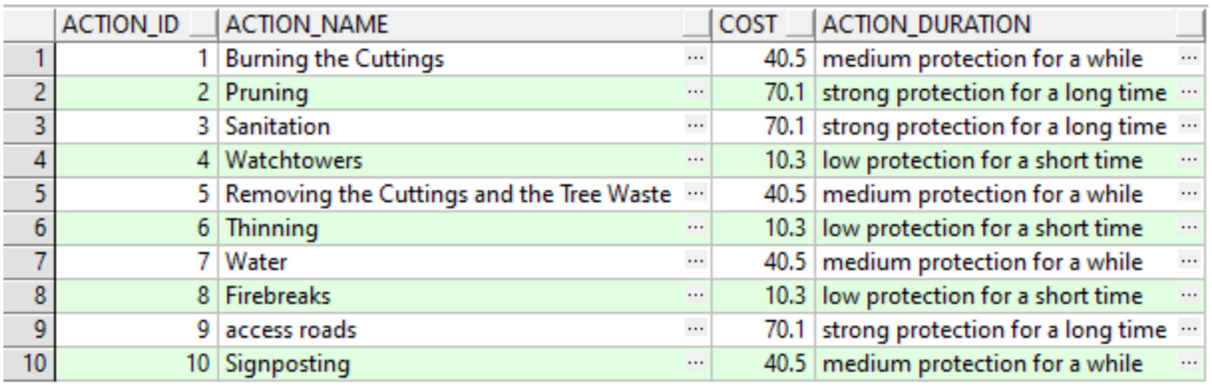
	STATION_ID	STATION_NAME	STATION_LOCATION	RESEARCH_SUBJECT	FOREST_ID
1	274	Fire prevention research lab	Spokane	Firebreaks	214

HasA table has no action_id above 10 now:



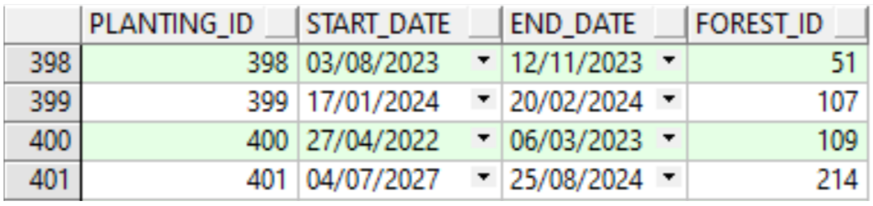
	FOREST_ID	ACTION_ID
1	175	7
2	271	7
3	59	1
4	225	1
5	379	2
6	22	9
7	137	1
8	199	6

Prevention_actions table now has no duplicate actions: (cost is in 100s dollars per km^2)



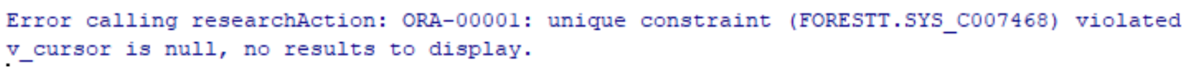
	ACTION_ID	ACTION_NAME	COST	ACTION_DURATION
1	1	Burning the Cuttings	40.5	medium protection for a while
2	2	Pruning	70.1	strong protection for a long time
3	3	Sanitation	70.1	strong protection for a long time
4	4	Watchtowers	10.3	low protection for a short time
5	5	Removing the Cuttings and the Tree Waste	40.5	medium protection for a while
6	6	Thinning	10.3	low protection for a short time
7	7	Water	40.5	medium protection for a while
8	8	Firebreaks	10.3	low protection for a short time
9	9	access roads	70.1	strong protection for a long time
10	10	Signposting	40.5	medium protection for a while

Tree planting project (for action research) added:



	PLANTING_ID	START_DATE	END_DATE	FOREST_ID
398	398	03/08/2023	12/11/2023	51
399	399	17/01/2024	20/02/2024	107
400	400	27/04/2022	06/03/2023	109
401	401	04/07/2027	25/08/2024	214

ERRORS:



```
Error calling researchAction: ORA-00001: unique constraint (FORESTT.SYS_C007468) violated
v_cursor is null, no results to display.
```

Code that raised the error:

```
-- Add action to the forest
INSERT INTO HASA (FOREST_ID, ACTION_ID) VALUES (f_id, a_id);
```

PROGRAM #2

This program has 2 sections : adding a column and planting trees for a soldier.

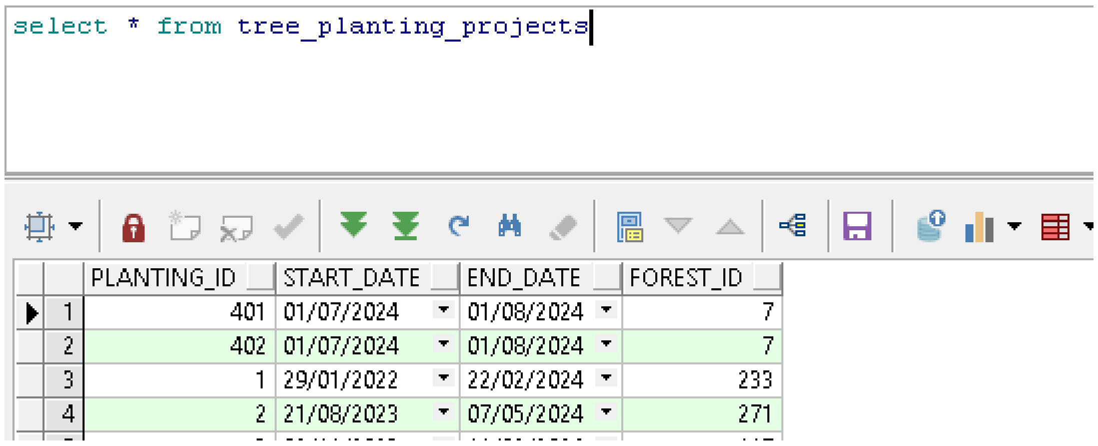
In the war swords of iron a soldier who kot killed and his family want to memorialise him by planting trees in a forest of their choice. The family gets to pick the type and origin of the tree. And how many trees to plant. And then an education program to memorialise the soldier.

Section 1

Add_column_to_projects procedure:

Adds a column “amount of trees” to table “tree planting projects”.

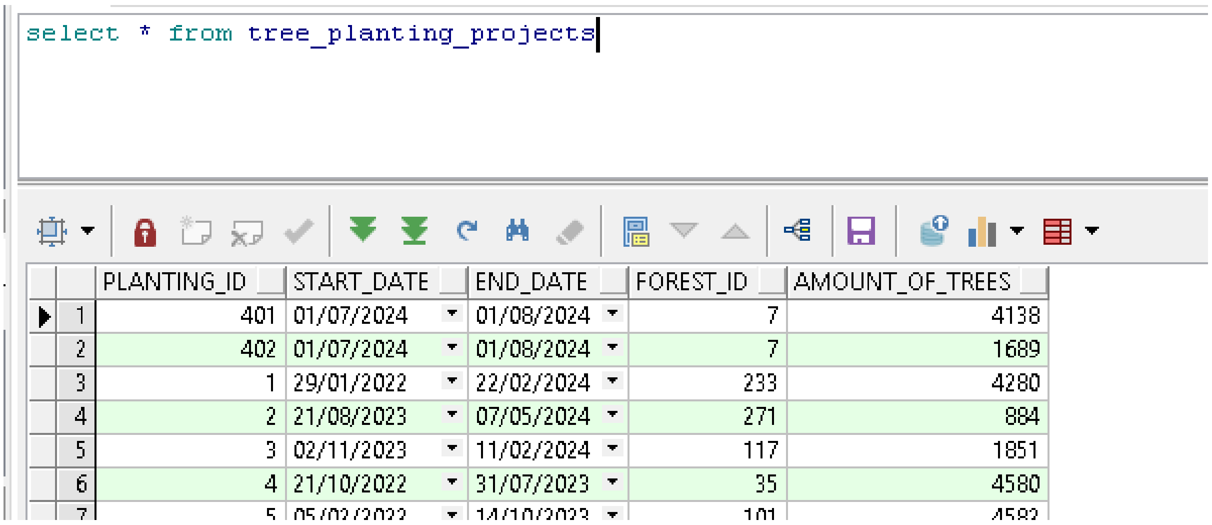
Before:
you can see that the column “amount of trees” does not exist.



```
select * from tree_planting_projects
```

	PLANTING_ID	START_DATE	END_DATE	FOREST_ID
1	401	01/07/2024	01/08/2024	7
2	402	01/07/2024	01/08/2024	7
3	1	29/01/2022	22/02/2024	233
4	2	21/08/2023	07/05/2024	271

After:



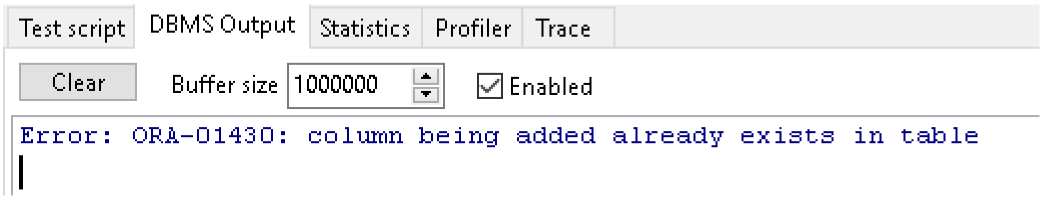
```
select * from tree_planting_projects
```

	PLANTING_ID	START_DATE	END_DATE	FOREST_ID	AMOUNT_OF_TREES
1	401	01/07/2024	01/08/2024	7	4138
2	402	01/07/2024	01/08/2024	7	1689
3	1	29/01/2022	22/02/2024	233	4280
4	2	21/08/2023	07/05/2024	271	884
5	3	02/11/2023	11/02/2024	117	1851
6	4	21/10/2022	31/07/2023	35	4580
7	5	05/03/2022	14/10/2023	101	4582

The code:

```
-- Procedure to add a column for amount of trees in tree planting project numbers and
populate it
CREATE OR REPLACE PROCEDURE add_column_to_projects IS
   v_random_number NUMBER;
BEGIN
   -- Step 1: Add a new column to store amount of trees
   EXECUTE IMMEDIATE 'ALTER TABLE tree_planting_projects ADD (amount_of_trees
NUMBER)';

   -- Step 2: Use a cursor to fetch planting_id for the loop
   FOR rec IN (SELECT planting_id FROM tree_planting_projects) LOOP
      BEGIN
         -- Generate random number between 10 and 5000
         v_random_number := ROUND(DBMS_RANDOM.VALUE(10, 5000));

         -- Use dynamic SQL to update amount_of_trees column
         EXECUTE IMMEDIATE 'UPDATE tree_planting_projects SET amount_of_trees = :1
WHERE planting_id = :2'
            USING v_random_number, rec.planting_id;

         --DBMS_OUTPUT.PUT_LINE('Updated planting_id ' || rec.planting_id || ' with
amount_of_trees: ' || v_random_number);
      END;
   END LOOP;

   -- COMMIT;

   --DBMS_OUTPUT.PUT_LINE('Random numbers added and updated successfully.');
EXCEPTION
   WHEN OTHERS THEN
      DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
END add_column_to_projects;
```

Exception:
when i try adding the column if it already exists.



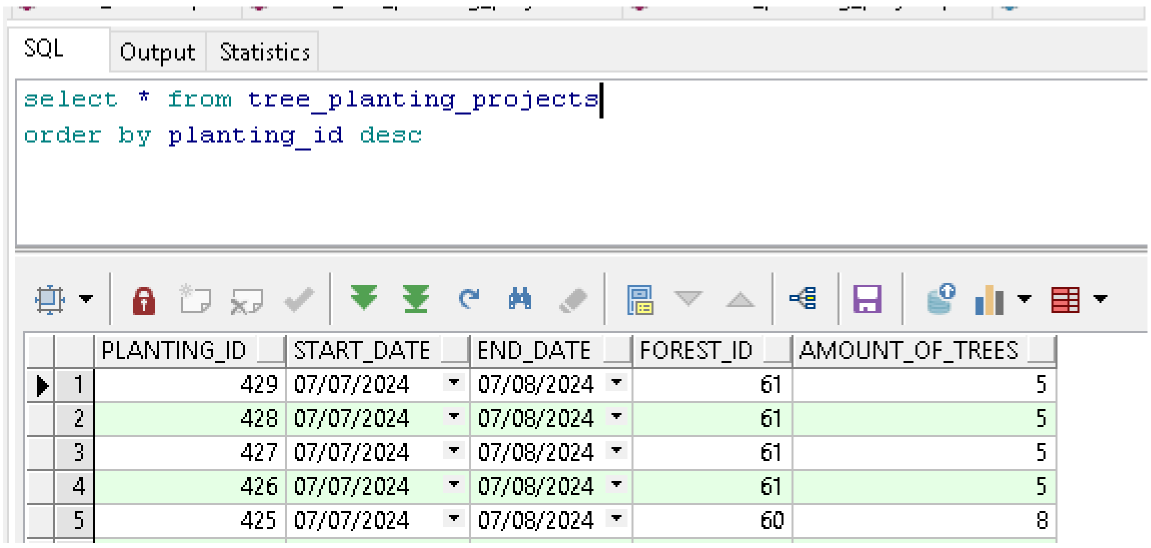
```
Error: ORA-01430: column being added already exists in table
```

Second section:

1. Add tree planting projects function:

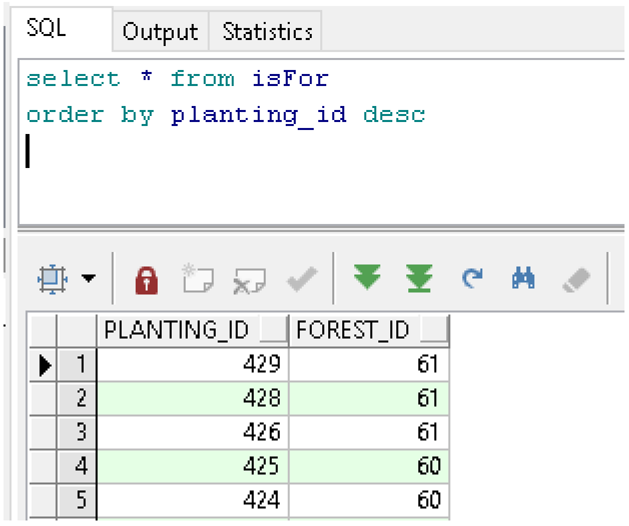
Gets forest id and amount of trees that the family want to plant.
Creates a new tree planting project and updates the isFor table.
returns the new tree planting project id.

before:



```
select * from tree_planting_projects
order by planting_id desc
```

	PLANTING_ID	START_DATE	END_DATE	FOREST_ID	AMOUNT_OF_TREES
1	429	07/07/2024	07/08/2024	61	5
2	428	07/07/2024	07/08/2024	61	5
3	427	07/07/2024	07/08/2024	61	5
4	426	07/07/2024	07/08/2024	61	5
5	425	07/07/2024	07/08/2024	60	8



```
select * from isFor
order by planting_id desc
```

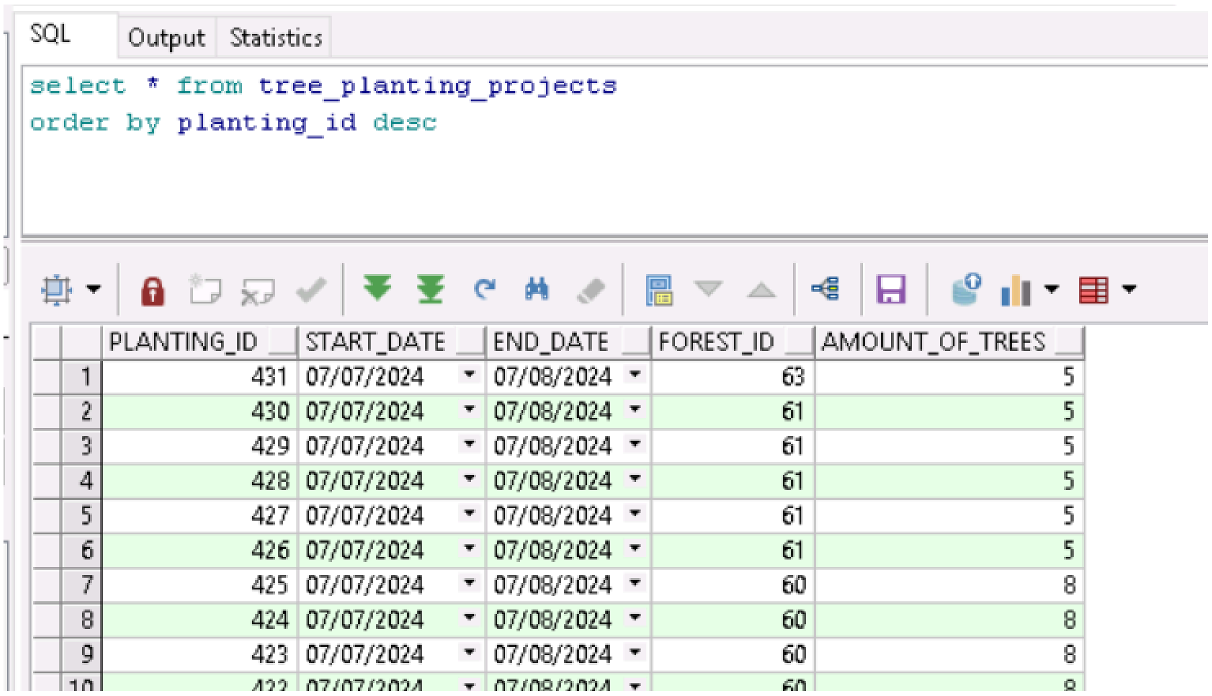
	PLANTING_ID	FOREST_ID
1	429	61
2	428	61
3	426	61
4	425	60
5	424	60

The code:

```
CREATE OR REPLACE FUNCTION add_tree_planting_project(
    p_forest_id IN INTEGER,
    p_num_of_trees_planted IN INTEGER
) RETURN INTEGER IS
    v_planting_id INTEGER;
    start_date DATE;
    end_date DATE;
BEGIN
```

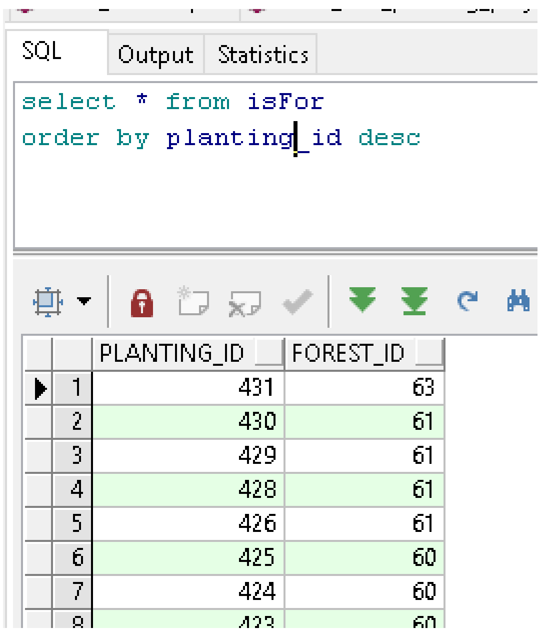
```
   -- Calculate start and end dates
   start_date := TRUNC(SYSDATE);
   end_date := ADD_MONTHS(TRUNC(SYSDATE), 1);

   -- Find the next available planting_id
   SELECT COALESCE(MAX(planting_id), 0) + 1 INTO v_planting_id FROM
Tree_Planting_Projects;

   -- Insert into Tree_Planting_Projects table
   INSERT INTO Tree_Planting_Projects (planting_id, start_date, end_date, forest_ID,
amount_of_trees)
   VALUES (v_planting_id, start_date, end_date, p_forest_id, p_num_of_trees_planted);

    DBMS_OUTPUT.PUT_LINE('Tree Planting project added successfully');

    --adds the planting id and forest id to the IsFor table
    INSERT INTO IsFor (planting_id, forest_ID)
      VALUES (v_planting_id, p_forest_id);
    DBMS_OUTPUT.PUT_LINE('added to IsFor table successfully');



   -- Return the planting_ID of the newly inserted record
   RETURN v_planting_id;

EXCEPTION
   WHEN OTHERS THEN
      -- Handle exceptions if needed
      DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM||'in planting project');
      raise;
      RETURN NULL; -- if something goes wrong returns nothing
END;
```

After:



```
select * from tree_planting_projects
order by planting_id desc
```

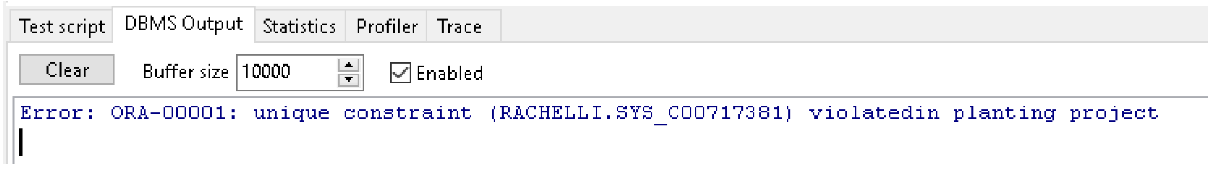
	PLANTING_ID	START_DATE	END_DATE	FOREST_ID	AMOUNT_OF_TREES
1	431	07/07/2024	07/08/2024	63	5
2	430	07/07/2024	07/08/2024	61	5
3	429	07/07/2024	07/08/2024	61	5
4	428	07/07/2024	07/08/2024	61	5
5	427	07/07/2024	07/08/2024	61	5
6	426	07/07/2024	07/08/2024	61	5
7	425	07/07/2024	07/08/2024	60	8
8	424	07/07/2024	07/08/2024	60	8
9	423	07/07/2024	07/08/2024	60	8
10	422	07/07/2024	07/08/2024	60	8



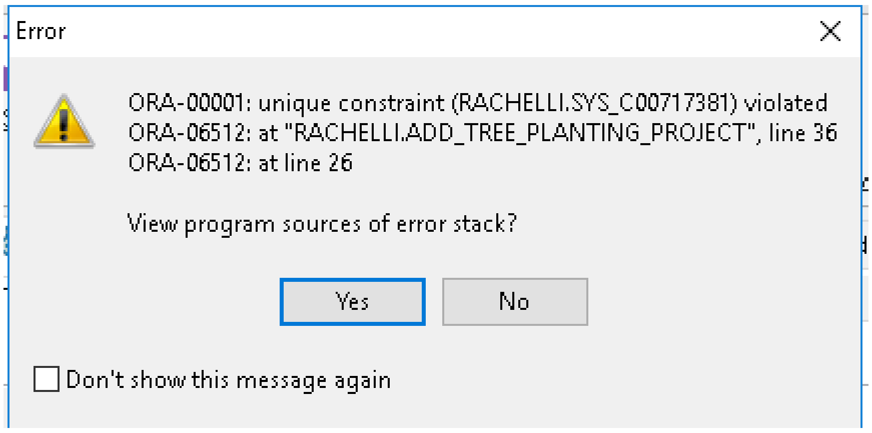
```
select * from isFor
order by planting_id desc
```

	PLANTING_ID	FOREST_ID
1	431	63
2	430	61
3	429	61
4	428	61
5	426	61
6	425	60
7	424	60
8	423	60

Exception:
Happened when I tried adding a planting project with a planting id that already existed.



```
Error: ORA-00001: unique constraint (RACHELLI.SYS_C00717381) violatedin planting project
```

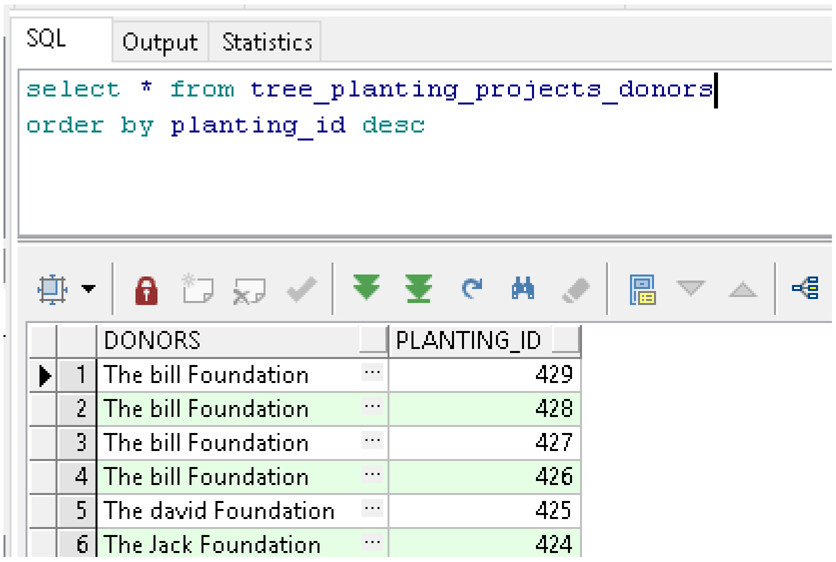


```
ORA-00001: unique constraint (RACHELLI.SYS_C00717381) violated
ORA-06512: at "RACHELLI.ADD_TREE_PLANTING_PROJECT", line 36
ORA-06512: at line 26

View program sources of error stack?
```

2. Add donors procedure:

Gets the planting id and the donors name and adds them to the tree_planting_projects_doonors table.

before:



```
select * from tree_planting_projects_donors
order by planting_id desc
```

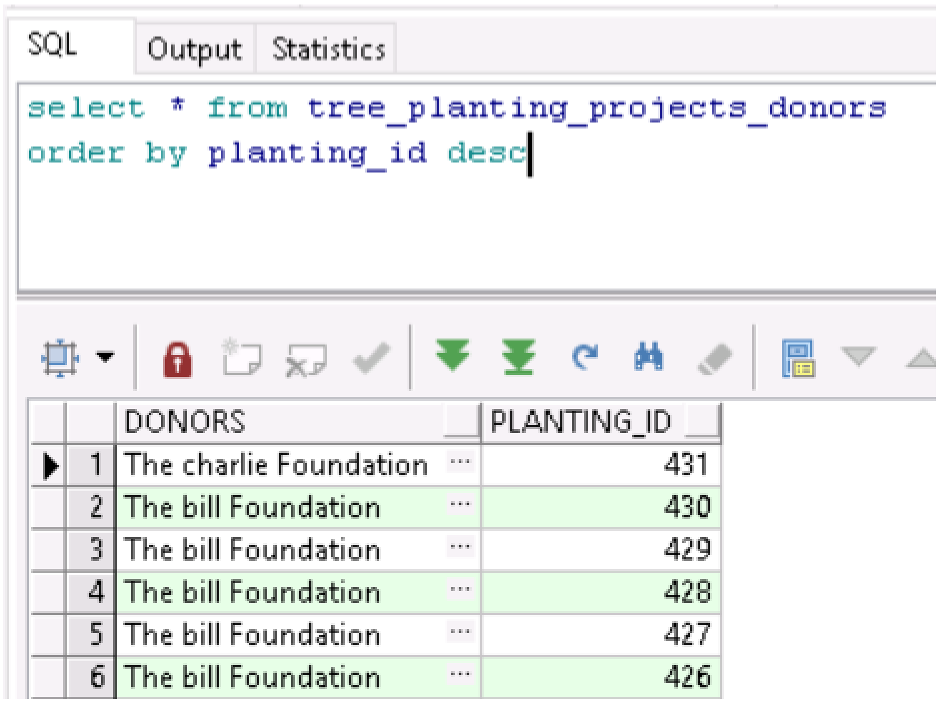
	DONORS	PLANTING_ID
1	The bill Foundation	429
2	The bill Foundation	428
3	The bill Foundation	427
4	The bill Foundation	426
5	The david Foundation	425
6	The Jack Foundation	424

The code:

```
CREATE OR REPLACE PROCEDURE add_donors (
   p_planting_id IN INTEGER,
   p_donors IN VARCHAR2
) IS
BEGIN
   -- Insert into Donors table associated with the planting project
   INSERT INTO tree_planting_projects_donors (planting_id, donors)
   VALUES (p_planting_id, p_donors);

   DBMS_OUTPUT.PUT_LINE('Donor ' || p_donors || ' added for planting ID ' ||
p_planting_id);
EXCEPTION
```

```
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM ||'in adding donors');
        raise;
END;
```

After:

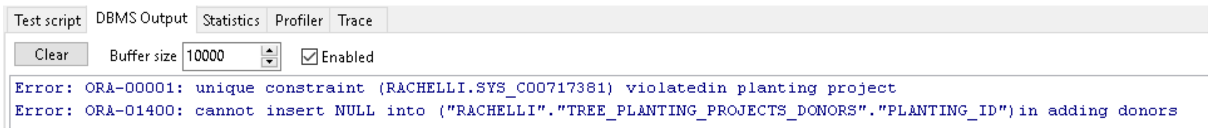


```
select * from tree_planting_projects_donors
order by planting_id desc
```

	DONORS	PLANTING_ID
1	The charlie Foundation	431
2	The bill Foundation	430
3	The bill Foundation	429
4	The bill Foundation	428
5	The bill Foundation	427
6	The bill Foundation	426

Exception:

Because i couldn't add the planting id from before it also caused a problem here.

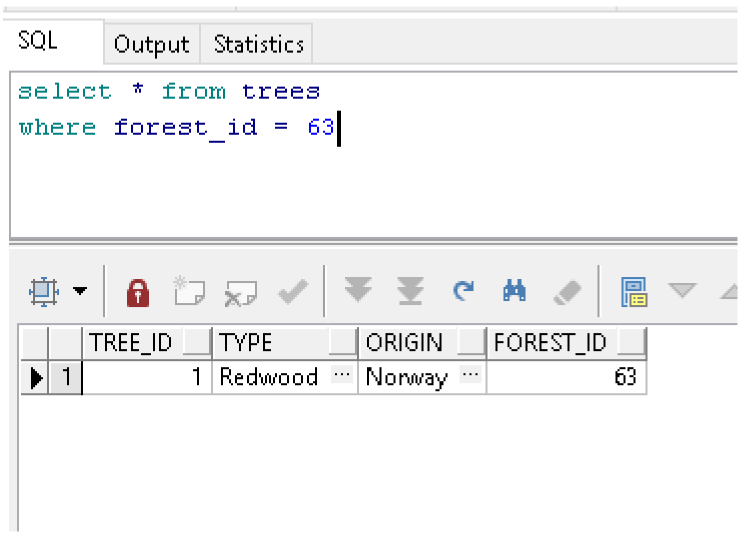


```
Error: ORA-00001: unique constraint (RACHELLI.SYS_C00717381) violatedin planting project
Error: ORA-01400: cannot insert NULL into ("RACHELLI"."TREE_PLANTING_PROJECTS_DONORS"."PLANTING_ID")in adding donors
```

3.Create trees procedure:

Gets the forest id type of tree origin of tree and num of trees. Creates the an=mount of trees that they want and add them to the trees table.

before:


SQL | Output | Statistics

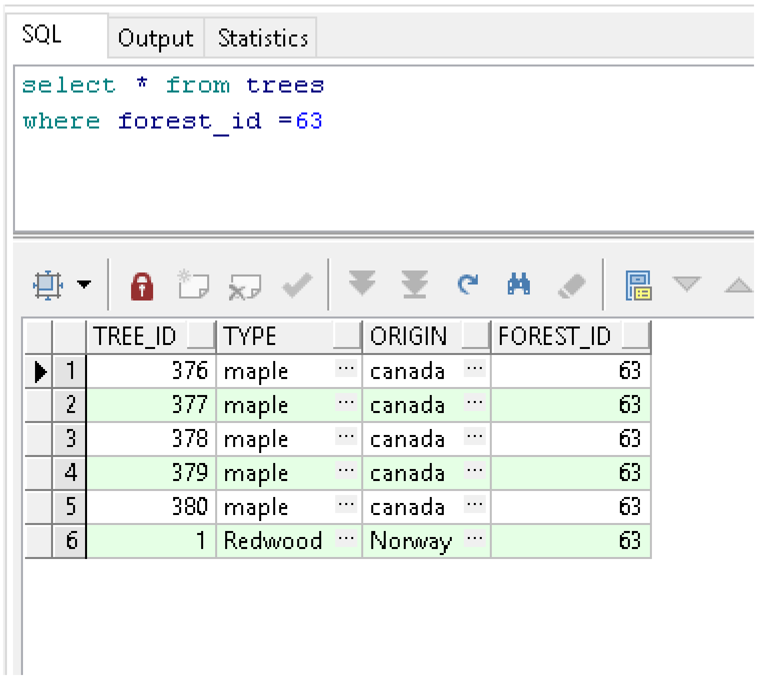
```
select * from trees
where forest_id = 63
```

```
  TREE_ID TYPE        ORIGIN    FOREST_ID
1       1 Redwood ... Norway ...       63
```

The code:

```
CREATE OR REPLACE PROCEDURE create_trees(
   p_forest_id IN INTEGER,
   p_type IN VARCHAR2,
   p_origin IN VARCHAR2,
   p_num_of_trees IN INTEGER
) IS
   v_tree_id INTEGER;
BEGIN
   FOR i IN 1..p_num_of_trees LOOP
      -- Find the next available tree_id
      SELECT COALESCE(MAX(tree_id), 0) + 1 INTO v_tree_id FROM Trees;

      -- Insert into Trees table
      INSERT INTO Trees (tree_id, forest_ID, type, origin)
      VALUES (v_tree_id, p_forest_id, p_type, p_origin);
   END LOOP;

   DBMS_OUTPUT.PUT_LINE('Successfully created ' || p_num_of_trees || ' trees.');
EXCEPTION
   WHEN OTHERS THEN
      DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM||' in create_trees');
      raise;
END;
```

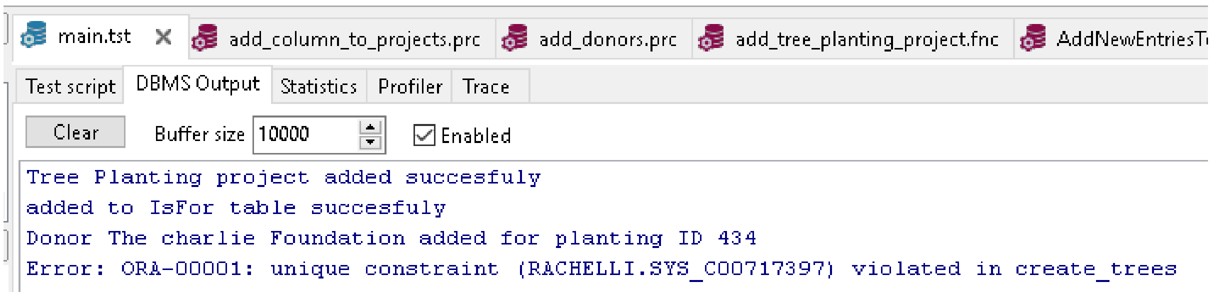
After:



```
select * from trees
where forest_id =63
```

	TREE_ID	TYPE	ORIGIN	FOREST_ID
1	376	maple	canada	63
2	377	maple	canada	63
3	378	maple	canada	63
4	379	maple	canada	63
5	380	maple	canada	63
6	1	Redwood	Norway	63

Exception:
Happened when i tried adding a non uniqu id.

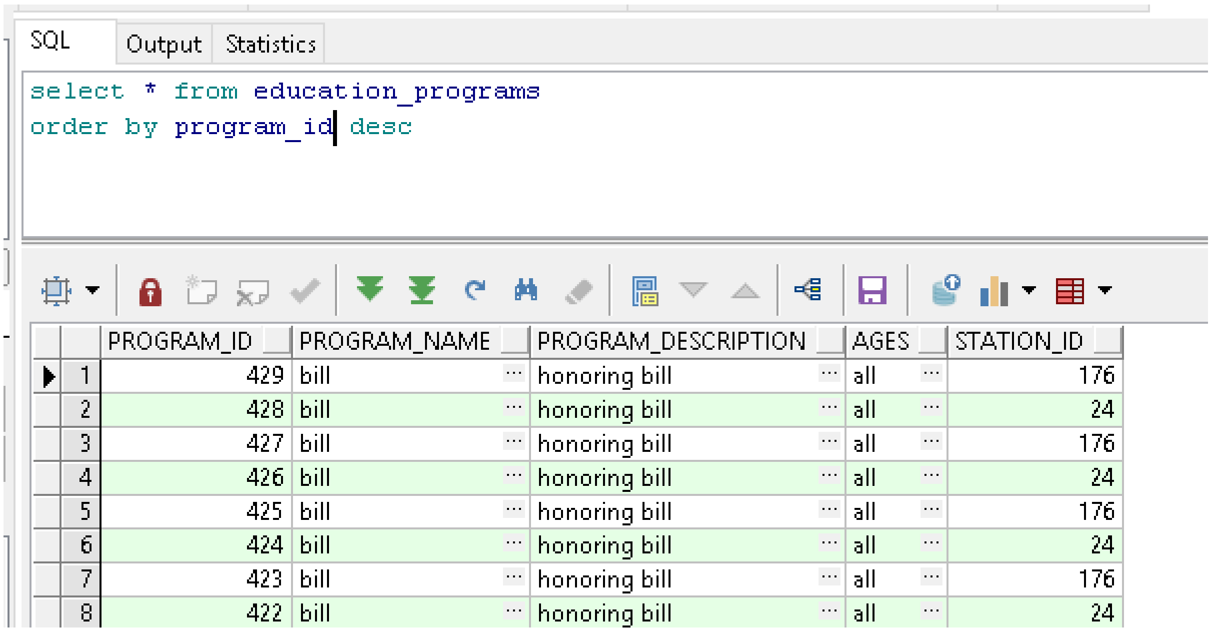


```
Tree Planting project added succesfuly
added to IsFor table succesfuly
Donor The charlie Foundation added for planting ID 434
Error: ORA-00001: unique constraint (RACHELLI.SYS_C00717397) violated in create_trees
```

4. Create education programs function:

Gets the forest id and the name of the soldier.
For every research station in the given forest it creates a new education program learning about the soldier. And returns a list of all the education programs that were created and in the main prints them out.

before:



```
select * from education_programs
order by program_id desc
```

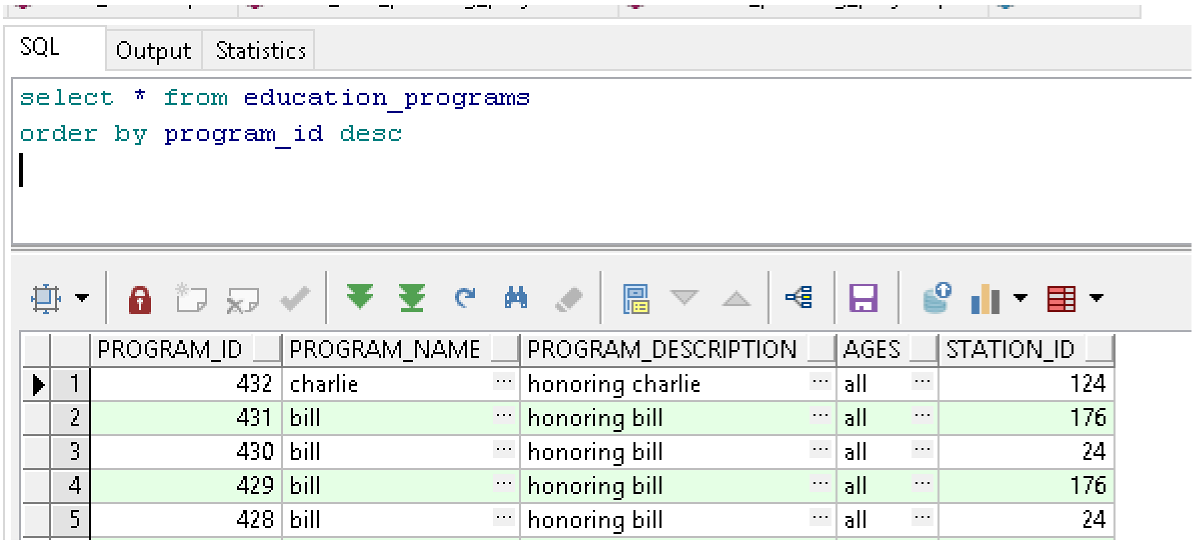
	PROGRAM_ID	PROGRAM_NAME	PROGRAM_DESCRIPTION	AGES	STATION_ID
1	429	bill	honoring bill	all	176
2	428	bill	honoring bill	all	24
3	427	bill	honoring bill	all	176
4	426	bill	honoring bill	all	24
5	425	bill	honoring bill	all	176
6	424	bill	honoring bill	all	24
7	423	bill	honoring bill	all	176
8	422	bill	honoring bill	all	24

The code:

```
CREATE OR REPLACE FUNCTION create_education_programs_func(
    p_forest_id IN INTEGER,
    p_name_of_soldier IN VARCHAR2
) RETURN SYS.ODCINUMBERLIST
IS
    l_added_stations SYS.ODCINUMBERLIST := SYS.ODCINUMBERLIST();
    CURSOR research_stations_cur IS
        SELECT station_ID FROM Research_Stations WHERE forest_ID = p_forest_id;
    v_station_id Research_Stations.station_ID%TYPE;
    p_program_description VARCHAR2(1000) := 'honoring '|| p_name_of_soldier;
    p_ages VARCHAR2(100) := 'all';
    v_program_id INTEGER;

BEGIN

    OPEN research_stations_cur;
    LOOP
        -- Find the next available program id
        SELECT COALESCE(MAX(program_id), 0) + 1 INTO v_program_id FROM
education_programs;

        FETCH research_stations_cur INTO v_station_id;
        EXIT WHEN research_stations_cur%NOTFOUND;

        -- Insert education program record
        INSERT INTO Education_Programs (program_id, program_name,
program_description, ages, station_ID)
        VALUES (v_program_id, p_name_of_soldier, p_program_description, p_ages,
v_station_id);
```

```
        -- Collect station ID where the program was added
        l_added_stations.EXTEND;
        l_added_stations(l_added_stations.LAST) := v_station_id;
    END LOOP;
    CLOSE research_stations_cur;

    DBMS_OUTPUT.PUT_LINE('education program added with id '|| v_program_id );

    -- Return the list of station IDs
    RETURN l_added_stations;

EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLERRM || 'in education
program');
        RETURN l_added_stations; -- Return whatever stations were added before the error
END;
```

After:

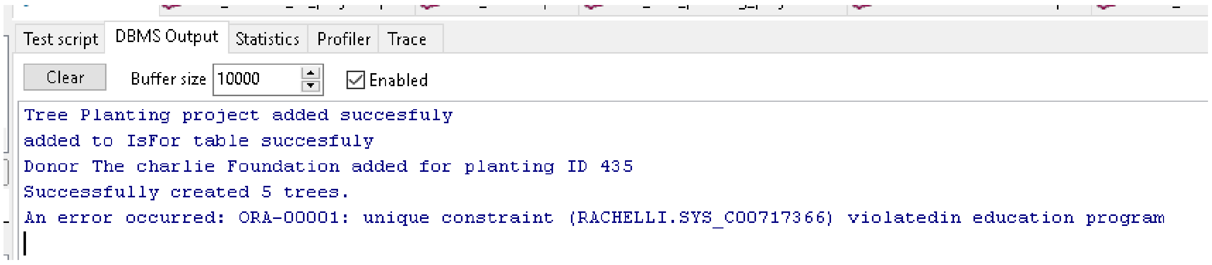


```
select * from education_programs
order by program_id desc
```

	PROGRAM_ID	PROGRAM_NAME	PROGRAM_DESCRIPTION	AGES	STATION_ID
1	432	charlie	honoring charlie	all	124
2	431	bill	honoring bill	all	176
3	430	bill	honoring bill	all	24
4	429	bill	honoring bill	all	176
5	428	bill	honoring bill	all	24

Exception:

Happened when I tried to add a non unique id.



Buffer size 10000 ☑ Enabled

```
Tree Planting project added succesfuly
added to IsFor table succesfuly
Donor The charlie Foundation added for planting ID 435
Successfully created 5 trees.
An error occurred: ORA-00001: unique constraint (RACHELLI.SYS_C00717366) violatedin education program
```

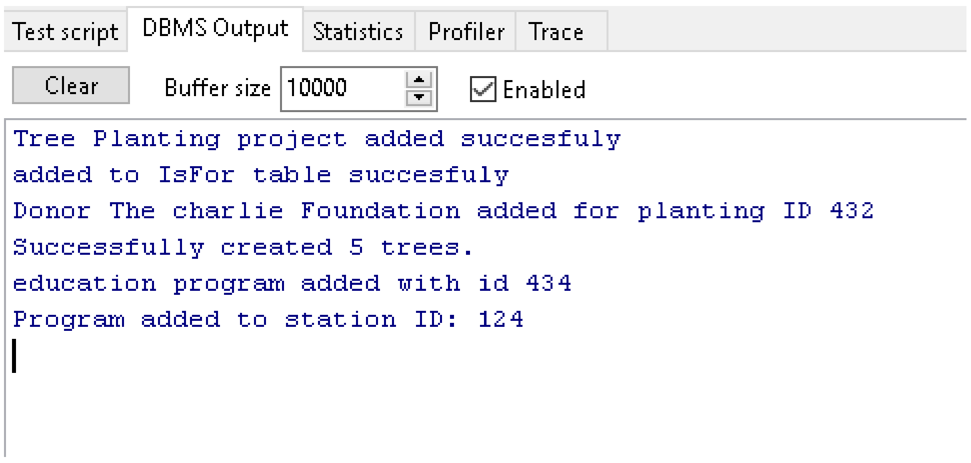
Main:

First we create anew column to tree planting projects but we only do it once.(we can drop the table first if we want) then we add the soldier memorial education programs.

The code:

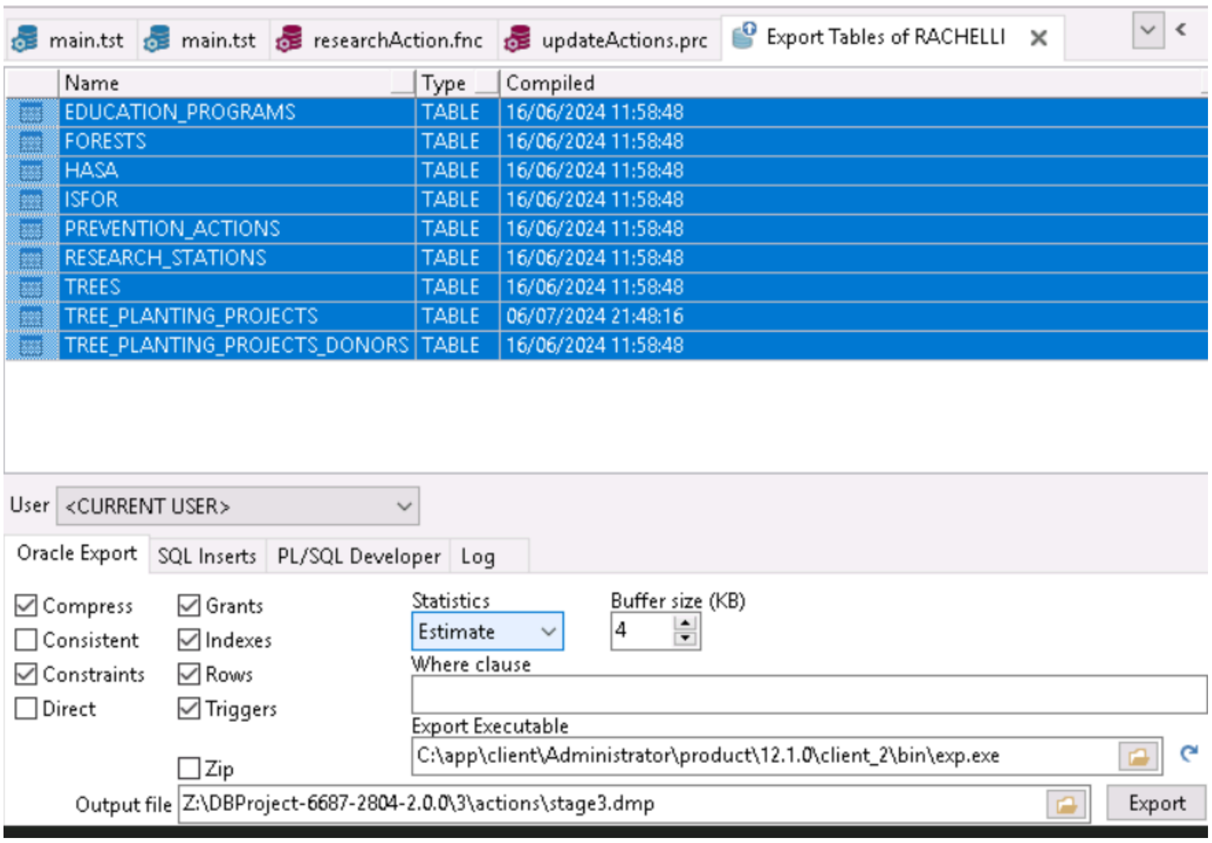
```
-- Created on 30/06/2024 by RACHELLI

DECLARE

  v_forest_id INTEGER := 63;
  v_num_of_trees INTEGER := 5;
  v_name_of_donor VARCHAR2(1000) := 'The charlie Foundation';
  v_name_of_soldier VARCHAR2(1000) := 'charlie';
  v_type VARCHAR2(1000) := 'maple';
  v_origin VARCHAR2(1000) := 'canada';
  v_planting_id INTEGER;

  stations SYS.ODCINUMBERLIST;

begin

    --dops the column amount_of_trees
    --EXECUTE IMMEDIATE 'ALTER TABLE tree_planting_projects DROP COLUMN
amount_of_trees';

    --adds a column in tree planting project representing how much trees we are going to
plant
    --add_column_to_projects();



 -- Step 1: Add a new tree planting project
    v_planting_id := add_tree_planting_project(
        p_forest_id => v_forest_id,
        p_num_of_trees_planted => v_num_of_trees
    );

    -- Step 2: Add donors for the project
    add_donors(
       p_planting_id => v_planting_id,
       p_donors => v_name_of_donor
    );

    -- Step 3: Create trees for the project
    create_trees(
        p_forest_id => v_forest_id,
        p_type => v_type,
        p_origin => v_origin,
        p_num_of_trees => v_num_of_trees
    );


  -- Step 4: Create education programs for research stations associated with the forest
  stations := create_education_programs_func(
        p_forest_id => v_forest_id,
        p_name_of_soldier => v_name_of_soldier);

   FOR i IN 1..stations.COUNT LOOP
        DBMS_OUTPUT.PUT_LINE('Program added to station ID: ' || stations(i));
   END LOOP;
commit;
end;
```

Example of output window:



```
Tree Planting project added succesfuly
added to IsFor table succesfuly
Donor The charlie Foundation added for planting ID 432
Successfully created 5 trees.
education program added with id 434
Program added to station ID: 124
```

Backup:



main.tst | main.tst | researchAction.fnc | updateActions.prc | Export Tables of RACHELLI

Name	Type	Compiled
EDUCATION_PROGRAMS	TABLE	16/06/2024 11:58:48
FORESTS	TABLE	16/06/2024 11:58:48
HASA	TABLE	16/06/2024 11:58:48
ISFOR	TABLE	16/06/2024 11:58:48
PREVENTION_ACTIONS	TABLE	16/06/2024 11:58:48
RESEARCH_STATIONS	TABLE	16/06/2024 11:58:48
TREES	TABLE	16/06/2024 11:58:48
TREE_PLANTING_PROJECTS	TABLE	06/07/2024 21:48:16
TREE_PLANTING_PROJECTS_DONORS	TABLE	16/06/2024 11:58:48

User: <CURRENT USER>

Oracle Export | SQL Inserts | PL/SQL Developer | Log

- ✓ Compress
- ☐ Consistent
- ✓ Constraints
- ☐ Direct
- ✓ Grants
- ✓ Indexes
- ✓ Rows
- ✓ Triggers
- ☐ Zip

Statistics: Estimate

Buffer size (KB): 4

Where clause:

Export Executable: C:\app\client\Administrator\product\12.1.0\client_2\bin\exp.exe

Output file: Z:\DBProject-6687-2804-2.0.0\3\actions\stage3.dmp

Export