

- Write a PL/SQL program using an **implicit** cursor that displays each state name, value for each race, and race-ethnicity in the following format:

The state name should be written only once. You can only use the POPULATION_COPD table. If you use any other table, 0 points.

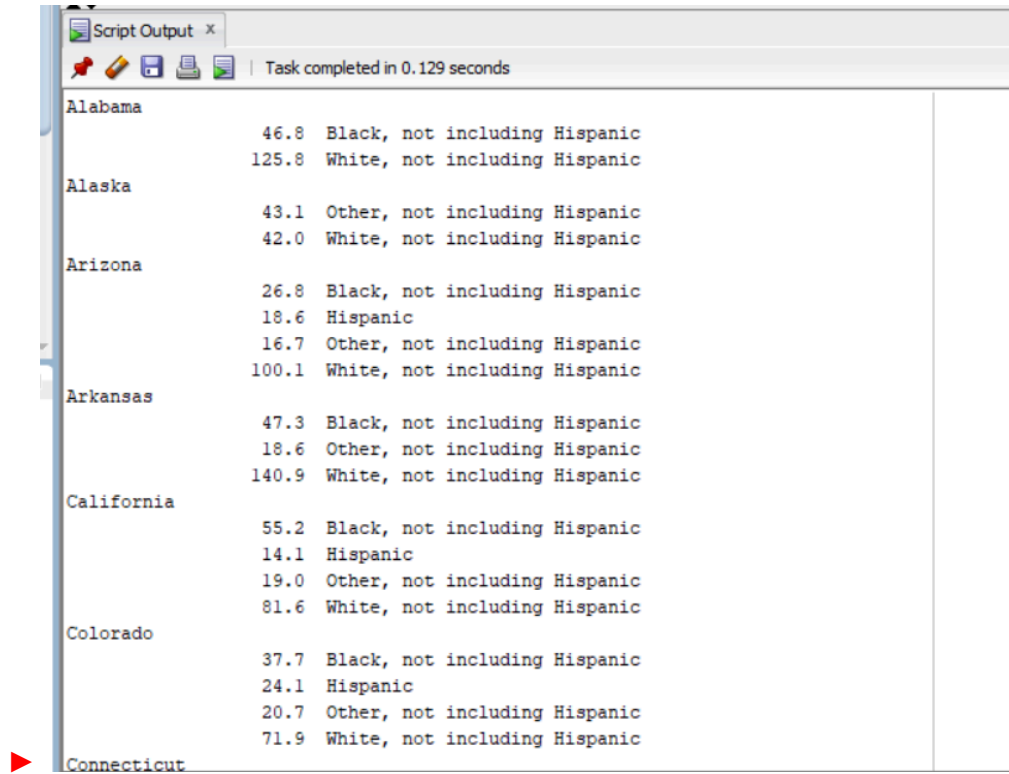
Show the program at the first red arrow.

Show the first 10 rows of the result at the second red arrow by snipping them. [10 points]

```
SET SERVEROUTPUT ON;
DECLARE
    v_CurrentState VARCHAR2(50) := NULL;
    CURSOR population_cursor IS
        SELECT State, Value, RaceEthnicity
        FROM Population_copd
        ORDER BY State, RaceEthnicity;
BEGIN
    FOR r IN population_cursor LOOP
        IF v_CurrentState IS NULL OR v_CurrentState <> r.State THEN
            DBMS_OUTPUT.PUT_LINE(r.State);
            v_CurrentState := r.State;
        END IF;

        DBMS_OUTPUT.PUT_LINE(
            RPAD(' ', 15) || TO_CHAR(r.Value, '9999.9') || ' ' || r.RaceEthnicity
        );
    END LOOP;
END;
```

```
END LOOP;  
END;  
/
```



Alabama	46.8	Black, not including Hispanic
	125.8	White, not including Hispanic
Alaska	43.1	Other, not including Hispanic
	42.0	White, not including Hispanic
Arizona	26.8	Black, not including Hispanic
	18.6	Hispanic
	16.7	Other, not including Hispanic
	100.1	White, not including Hispanic
Arkansas	47.3	Black, not including Hispanic
	18.6	Other, not including Hispanic
	140.9	White, not including Hispanic
California	55.2	Black, not including Hispanic
	14.1	Hispanic
	19.0	Other, not including Hispanic
	81.6	White, not including Hispanic
Colorado	37.7	Black, not including Hispanic
	24.1	Hispanic
	20.7	Other, not including Hispanic
	71.9	White, not including Hispanic
Connecticut		

2-a) Write a PL/SQL program using an **implicit** cursor that compares only White, not including Hispanic values to the average of White, not including Hispanic values.

Your output should look like the following:

You can only use the COPD1 table. If you use any other table, 0 points.

Show the program at the first red arrow.

Show the first 10 rows of the result at the second red arrow by snipping them. [5 points]



```
SET SERVEROUTPUT ON;
DECLARE
    v_AverageWhite NUMBER;
BEGIN
    -- Calculate the average value for "White, not including Hispanic"
    SELECT AVG(Value)
    INTO v_AverageWhite
    FROM COPD1
    WHERE RaceEthnicity = 'White, not including Hispanic';

    -- Display the average value
    DBMS_OUTPUT.PUT_LINE('The average of all "White, not including Hispanic" value is ' ||
        TO_CHAR(v_AverageWhite, '999.99'));

    DBMS_OUTPUT.PUT_LINE('-----');

    -- Compare each state's value with the average
    FOR r IN (SELECT State, Value FROM COPD1 WHERE RaceEthnicity = 'White, not including
        Hispanic') LOOP
        DBMS_OUTPUT.PUT_LINE(
            RPAD(r.State, 20) || 'value(' || TO_CHAR(r.Value, '999.9') || ') is ' ||
            CASE
                WHEN r.Value > v_AverageWhite THEN 'greater (>) than'
                WHEN r.Value < v_AverageWhite THEN 'less (<) than'
```

```

        ELSE 'equal to'
    END ||
    ' average value ' || TO_CHAR(v_AverageWhite, '999.9')
);
END LOOP;
END;
/

```

Script Output x

Task completed in 0.144 seconds

The average of all "White, not including Hispanic" value is 87.12		

Alabama	value(125.8) is greater (>) than average value	87.1
Alaska	value(42.0) is less (<) than average value	87.1
Arizona	value(100.1) is greater (>) than average value	87.1
Arkansas	value(140.9) is greater (>) than average value	87.1
California	value(81.6) is less (<) than average value	87.1
Colorado	value(71.9) is less (<) than average value	87.1
Connecticut	value(60.4) is less (<) than average value	87.1
Delaware	value(87.5) is greater (>) than average value	87.1
District of Columbia	value(10.4) is less (<) than average value	87.1
Florida	value(106.5) is greater (>) than average value	87.1
Georgia	value(96.4) is greater (>) than average value	87.1
Hawaii	value(47.5) is less (<) than average value	87.1
Idaho	value(79.6) is less (<) than average value	87.1
Illinois	value(78.9) is less (<) than average value	87.1
Indiana	value(111.2) is greater (>) than average value	87.1
Iowa	value(85.7) is less (<) than average value	87.1
Kansas	value(95.6) is greater (>) than average value	87.1
Kentucky	value(113.8) is greater (>) than average value	87.1
Louisiana	value(95.4) is greater (>) than average value	87.1

2-b) Write a PL/SQL program using an **implicit** cursor that compares MULTIRACIAL values for all states in alphabetical order.

Your output should look like the following:

You can only use the POPULATION2 table. If you use any other table, 0 points.

Show the program at the first red arrow.

Show the first 10 rows of the result at the second red arrow by snipping them. [5 points]



```
SET SERVEROUTPUT ON;
DECLARE
    CURSOR population_cursor IS
        SELECT State, mixed_race_multi_racial AS Multiracial
        FROM POPULATION2
        ORDER BY State;
    v_PrevState VARCHAR2(50) := NULL;
    v_PrevValue NUMBER;
    v_CurrentState VARCHAR2(50);
    v_CurrentValue NUMBER;
    v_FirstRow BOOLEAN := TRUE;
BEGIN
    OPEN population_cursor;

    -- Fetch the first row
    FETCH population_cursor INTO v_PrevState, v_PrevValue;

    -- Loop through the cursor
    LOOP
        -- Fetch the next row
        FETCH population_cursor INTO v_CurrentState, v_CurrentValue;
```

```

-- Exit the loop if there are no more rows
EXIT WHEN population_cursor%NOTFOUND;

-- Skip the comparison for the first state (Alabama)
IF v_FirstRow THEN
    v_FirstRow := FALSE;
    CONTINUE;
END IF;

-- Compare the values and print the result
DBMS_OUTPUT.PUT(v_CurrentState || ' multiracial value is (' || v_CurrentValue ||
')');

IF v_CurrentValue < v_PrevValue THEN
    DBMS_OUTPUT.PUT_LINE(' is LESS than ' || v_PrevState || ' multiracial value ('
|| v_PrevValue || ') -');
ELSIF v_CurrentValue > v_PrevValue THEN
    DBMS_OUTPUT.PUT_LINE(' is MORE than ' || v_PrevState || ' multiracial value ('
|| v_PrevValue || ') +');
ELSE
    DBMS_OUTPUT.PUT_LINE(' is EQUAL to ' || v_PrevState || ' multiracial value (' ||
v_PrevValue || ') =');
END IF;

-- Update the previous state and value
v_PrevState := v_CurrentState;
v_PrevValue := v_CurrentValue;
END LOOP;

CLOSE population_cursor;
END;
/

```

Script Output x

Task completed in 0.126 seconds

```

Arizona multiracial value is (266840) is MORE than Alabama multiracial value (184618) +
Arkansas multiracial value is (147157) is LESS than Arizona multiracial value (266840) -
California multiracial value is (1627722) is MORE than Arkansas multiracial value (147157) +
Colorado multiracial value is (260798) is LESS than California multiracial value (1627722) -
Connecticut multiracial value is (137569) is LESS than Colorado multiracial value (260798) -
Delaware multiracial value is (43023) is LESS than Connecticut multiracial value (137569) -
District of Columbia multiracial value is (29485) is LESS than Delaware multiracial value (43023) -
Florida multiracial value is (792143) is MORE than District of Columbia multiracial value (29485) +
Georgia multiracial value is (390133) is LESS than Florida multiracial value (792143) -
Idaho multiracial value is (77808) is LESS than Georgia multiracial value (390133) -
Hawaii multiracial value is (291890) is MORE than Idaho multiracial value (77808) +
Illinois multiracial value is (414855) is MORE than Hawaii multiracial value (291890) +
Indiana multiracial value is (265344) is LESS than Illinois multiracial value (414855) -
Iowa multiracial value is (108673) is LESS than Indiana multiracial value (265344) -
Kansas multiracial value is (149025) is MORE than Iowa multiracial value (108673) +
Kentucky multiracial value is (175363) is MORE than Kansas multiracial value (149025) +

```

3- This question has two steps:

STEP1: Write a PL/SQL statement to add a new column (SUMMATION) into the table POPULATION2. [3 point]

Show the PL/SQL statement at the first red arrow.

STEP2: Write a PL/SQL program using an **implicit** cursor that adds integer values of POPULATION2 in a row and then saves them in a SUMMATION column. [7 points]

Show the program at the second red arrow.

Show the first 10 rows of the result at the third red arrow by snipping them.

Show the last 10 rows of the result at the fourth red arrow by snipping them.

STEP1



```
ALTER TABLE POPULATION2
  ADD SUMMATION NUMBER;
```

STEP2



```
DECLARE
  CURSOR population_cursor IS
    SELECT State, White, black_or_african_american, mixed_race_multi_racial, other2
    FROM POPULATION2;
  v_State VARCHAR2(50);
  v_White NUMBER;
  v_Black NUMBER;
  v_Mixed NUMBER;
  v_other NUMBER;
  v_Summation NUMBER;
BEGIN
  FOR r IN population_cursor LOOP
    v_State := r.State;
    v_White := r.White;
    v_Black := r.black_or_african_american;
    v_Mixed := r.mixed_race_multi_racial;
    v_other := r.other2;
    -- Calculate the sum of integer values
    v_Summation := TRUNC(v_White) + TRUNC(v_Black) + TRUNC(v_Mixed);
```

```

-- Update the "SUMMATION" column
UPDATE POPULATION2
SET SUMMATION = v_Summation
WHERE State = v_State;
END LOOP;

COMMIT;
END;
/

```

► First 10 Rows:

STATE	WHITE	BLACK_OR_AFRICAN_AMERICAN	MIXED_RACE_MULTI_RACIAL	OTHER2	SUMMATION
1 Alabama	3171351	1288159	184618	116104	4644128
2 Alaska	421758	20731	71761	169317	514250
3 Arizona	3816547	317161	266840	558701	4400548
4 Arkansas	2063550	449884	147157	94086	2660591
5 California	13714587	2119286	1627722	6496976	17461595
6 Colorado	3760663	221310	260798	267553	4242771
7 Connecticut	2279232	360937	137569	204913	2777738
8 Delaware	579851	212960	43023	49824	835834
9 District of Columbia	261771	282066	29485	38571	573322
10 Florida	11100503	3127052	792143	821249	15019698

► Last 10 Rows:

41 South Carolina	3178552	1269031	189580	128424	4637163
42 South Dakota	705583	17441	34432	90470	757456
43 Tennessee	4900246	1083772	270223	177412	6254241
44 Texas	11584597	3444712	886095	1788384	15915404
45 Utah	2465355	37192	120452	155705	2622999
46 Vermont	573201	8649	29549	16174	611399
47 Virginia	5058363	1578090	404910	681281	7041363
48 Washington	4918820	296170	511114	919964	5726104
49 West Virginia	1598834	64749	72135	23171	1735718
50 Wisconsin	4634018	366508	203746	242156	5204272
51 Wyoming	469664	4735	23674	19732	498073

Dbms Output

+ | Buffer Size: 20000

4- Create a trigger **NewPopulation** that will guarantee any time the table POPULATION1's NATIVEAMERICAN or ASIAN or PACIFICISLANDER or OTHERRACE population status is updated, the trigger writes the following message to the output window.

++ Population may increase any number.

Show the program at the first red arrow.

Show the outputs of the trigger at the second red arrow by snipping them. [10]



```
CREATE OR REPLACE TRIGGER NewPopulation
BEFORE UPDATE OF NATIVE_AMERICAN_OR_ALASKA_NATIVE, ASIAN, PACIFIC_ISLANDER, SOME_OTHER_RACE
ON POPULATION1
FOR EACH ROW
BEGIN
    -- Check if any of the specified columns are updated
```

```

IF :NEW.NATIVE_AMERICAN_OR_ALASKA_NATIVE != :OLD.NATIVE_AMERICAN_OR_ALASKA_NATIVE OR
:NEW.ASIAN != :OLD.ASIAN OR
:NEW.PACIFIC_ISLANDER != :OLD.PACIFIC_ISLANDER OR
:NEW.SOME_OTHER_RACE != :OLD.SOME_OTHER_RACE THEN
  DBMS_OUTPUT.PUT_LINE('STATE: ' || :NEW.STATE);
  DBMS_OUTPUT.PUT_LINE('New Population Race:');
  IF :NEW.NATIVE_AMERICAN_OR_ALASKA_NATIVE != :OLD.NATIVE_AMERICAN_OR_ALASKA_NATIVE THEN
    DBMS_OUTPUT.PUT_LINE('NATIVE_AMERICAN_OR_ALASKA_NATIVE - Old Population Value: ' ||
:OLD.NATIVE_AMERICAN_OR_ALASKA_NATIVE);
    DBMS_OUTPUT.PUT_LINE('NATIVE_AMERICAN_OR_ALASKA_NATIVE - New Population Value: ' ||
:NEW.NATIVE_AMERICAN_OR_ALASKA_NATIVE);
  END IF;
  IF :NEW.ASIAN != :OLD.ASIAN THEN
    DBMS_OUTPUT.PUT_LINE('ASIAN - Old Population Value: ' || :OLD.ASIAN);
    DBMS_OUTPUT.PUT_LINE('ASIAN - New Population Value: ' || :NEW.ASIAN);
  END IF;
  IF :NEW.PACIFIC_ISLANDER != :OLD.PACIFIC_ISLANDER THEN
    DBMS_OUTPUT.PUT_LINE('PACIFIC_ISLANDER - Old Population Value: ' || :OLD.PACIFIC_ISLANDER);
    DBMS_OUTPUT.PUT_LINE('PACIFIC_ISLANDER - New Population Value: ' || :NEW.PACIFIC_ISLANDER);
  END IF;
  IF :NEW.SOME_OTHER_RACE != :OLD.SOME_OTHER_RACE THEN
    DBMS_OUTPUT.PUT_LINE('SOME_OTHER_RACE - Old Population Value: ' || :OLD.SOME_OTHER_RACE);
    DBMS_OUTPUT.PUT_LINE('SOME_OTHER_RACE - New Population Value: ' || :NEW.SOME_OTHER_RACE);
  END IF;
END IF;
END;
/

```

► Pacific_Islander++ Output:

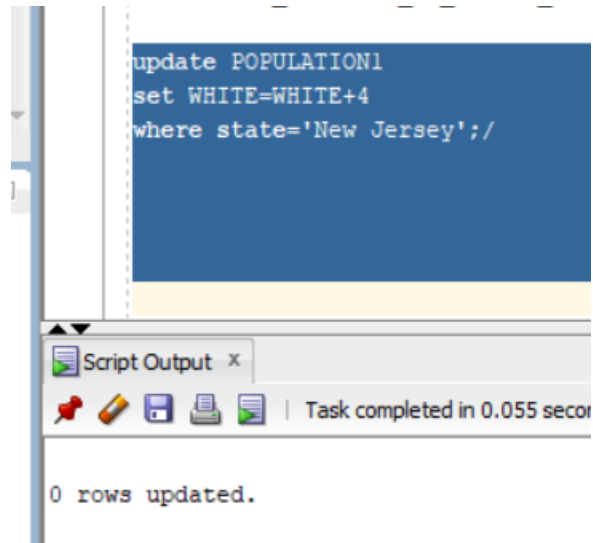
```

      characters other than the double quote
STATE: New Jersey
New Population Race:
PACIFIC_ISLANDER - Old Population Value: 1944
PACIFIC_ISLANDER - New Population Value: 1945

1 row updated.

```

White+4 Output:



Native_American Output:

```
STATE: Alabama  
New Population Race:  
NATIVE_AMERICAN_OR_ALASKA_NATIVE - Old Population Value: 23119  
NATIVE_AMERICAN_OR_ALASKA_NATIVE - New Population Value: 23120  
STATE: Alaska  
New Population Race:  
NATIVE_AMERICAN_OR_ALASKA_NATIVE - Old Population Value: 108838  
NATIVE_AMERICAN_OR_ALASKA_NATIVE - New Population Value: 108839  
STATE: Arizona  
New Population Race:  
NATIVE_AMERICAN_OR_ALASKA_NATIVE - Old Population Value: 263930  
NATIVE_AMERICAN_OR_ALASKA_NATIVE - New Population Value: 263931  
STATE: Arkansas  
New Population Race:  
NATIVE_AMERICAN_OR_ALASKA_NATIVE - Old Population Value: 20549  
NATIVE_AMERICAN_OR_ALASKA_NATIVE - New Population Value: 20550  
STATE: California  
New Population Race:  
NATIVE_AMERICAN_OR_ALASKA_NATIVE - Old Population Value: 156085  
NATIVE_AMERICAN_OR_ALASKA_NATIVE - New Population Value: 156086  
STATE: Colorado  
New Population Race:  
NATIVE_AMERICAN_OR_ALASKA_NATIVE - Old Population Value: 33768  
NATIVE_AMERICAN_OR_ALASKA_NATIVE - New Population Value: 33769
```

5- Create another trigger **NewPopulation2** that will guarantee any time the table POPULATION1's PACIFICISLANDER or OTHERRACE population is updated, the trigger updates the table PopulationTrack with old and new values of Pacific Islanders and Other Race values.

PopulationTrack

Test it with:

```
update POPULATION1  
set PACIFICISLANDER ++  
where STATE = 'New Jersey';/
```

```
update POPULATION1  
set OTHERRACE=OTHERRACE+5;  
/
```

PopulationTrack table should look like:

Show the PopulationTrack table creation SQL statement at the first red arrow.

Show the trigger code at the second red arrow.

Show the first 10 rows of the PopulationTrack table at the third red arrow by snipping it. [10]



```
CREATE TABLE PopulationTrack (  
  STATE VARCHAR2(50),  
  OLD_PACIFIC_ISLANDER NUMBER,
```

```

NEW_PACIFIC_ISLANDER NUMBER,
OLD_SOME_OTHER_RACE NUMBER,
NEW_SOME_OTHER_RACE NUMBER
);

```

```

CREATE OR REPLACE TRIGGER NewPopulation2
AFTER UPDATE OF PACIFIC_ISLANDER, SOME_OTHER_RACE ON POPULATION1
FOR EACH ROW
BEGIN
    -- Insert the old and new values into PopulationTrack
    INSERT INTO PopulationTrack (STATE, OLD_PACIFIC_ISLANDER, NEW_PACIFIC_ISLANDER, OLD_SOME_OTHER_RACE,
NEW_SOME_OTHER_RACE)
VALUES (:NEW.STATE, :OLD.PACIFIC_ISLANDER, :NEW.PACIFIC_ISLANDER, :OLD.SOME_OTHER_RACE,
:NEW.SOME_OTHER_RACE);
END;
/

```

► NJ has +1 pacific islander and +5 for some other race in output below

STATE	OLD_PACIFIC_ISLANDER	NEW_PACIFIC_ISLANDER	OLD_SOME_OTHER_RACE	NEW_SOME_OTHER_RACE
26 Missouri	9293	9293	22377	22382
27 Montana	839	839	4374	4379
28 Nebraska	1318	1318	6335	6340
29 Nevada	22970	22970	17171	17176
30 New Hampshire	388	388	5916	5921
31 New Jersey	1945	1946	70354	70359
32 New Mexico	1451	1451	10340	10345
33 New York	6097	6097	197107	197112
34 North Carolina	6980	6980	46340	46345
35 North Dakota	869	869	1853	1858
36 Ohio	4493	4493	45217	45222
37 Oklahoma	8168	8168	13602	13607
38 Oregon	18197	18197	22962	22967
39 Pennsylvania	3162	3162	54541	54546
40 Rhode Island	320	320	11392	11397
41 South Carolina	3085	3085	19354	19359
42 South Dakota	493	493	2050	2055
43 Tennessee	3594	3594	23977	23982
44 Texas	27857	27857	113584	113589
45 Utah	35831	35831	12566	12571
46 Vermont	170	170	2561	2566
47 Virginia	6195	6195	45394	45399
48 Washington	62490	62490	43221	43226
49 West Virginia	429	429	4652	4657
50 Wisconsin	1892	1892	17613	17618
51 Wyoming	489	489	2425	2430