Home work #1

Problem :

list of all employees info such as first\_name, last\_name, salary, department\_name, city for those who work in 'Seattle'

once you get this list, then load all these info into a cursor (x\_cursor)

fetch them one by one to a record (x\_record)

if employee earns less than 7000 add their salary 300

if employee earns more than 7000 add only 100 to their salary

====================================

DECLARE

CURSOR emp\_cursor IS

-- in order to actually select employee in the later record, employee\_id is best to use.

SELECT e.employee\_id, e.first\_name, e.last\_name, e.salary, d.department\_name, l.city

FROM HR\_EMPLOYEES e

JOIN HR\_DEPARTMENTS d ON e.department\_id = d.department\_id

JOIN HR\_LOCATIONS l ON d.location\_id = l.location\_id

WHERE l.city = 'Seattle';

emp\_record emp\_cursor%ROWTYPE;

-- instead of re-selecting the same SELECT statement as emp\_cursor,

-- updating salary and using this variable to store the increased salary for log.

v\_increased\_salary NUMBER;

BEGIN

-- don't need OPEN emp\_cursor because for loop automatically opens it.

-- also don't need FETCH or CLOSE.

FOR emp\_record IN emp\_cursor LOOP

DBMS\_OUTPUT.PUT\_LINE('Employee: ' || emp\_record.first\_name || ' ' || emp\_record.last\_name || ' (' || emp\_record.employee\_id || ') Salary: ' || emp\_record.salary);

IF emp\_record.salary < 7000 THEN

UPDATE HR\_EMPLOYEES

SET salary = salary + 300

WHERE employee\_id = emp\_record.employee\_id;

v\_increased\_salary := emp\_record.salary + 300;

DBMS\_OUTPUT.PUT\_LINE('Salary less than 7000, adding 300');

DBMS\_OUTPUT.PUT\_LINE('Employee: ' || emp\_record.first\_name || ' ' || emp\_record.last\_name || ' (' || emp\_record.employee\_id || ') Salary: ' || v\_increased\_salary);

DBMS\_OUTPUT.PUT\_LINE('------------------------------------------');

ELSIF emp\_record.salary >= 7000 THEN

UPDATE HR\_EMPLOYEES

SET salary = salary + 100

WHERE employee\_id = emp\_record.employee\_id;

v\_increased\_salary := emp\_record.salary + 100;

DBMS\_OUTPUT.PUT\_LINE('Salary is 7000 or more, adding 100');

DBMS\_OUTPUT.PUT\_LINE('Employee: ' || emp\_record.first\_name || ' ' || emp\_record.last\_name || ' (' || emp\_record.employee\_id || ') Salary: ' || v\_increased\_salary);

DBMS\_OUTPUT.PUT\_LINE('------------------------------------------');

END IF;

END LOOP;

-- Am not commiting because I do not want this to actually change the database.

--COMMIT;

END;

The code screenshotted:

A white screen with black text

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

The DBMS output:

A screenshot of a computer

AI-generated content may be incorrect.

We can see the updates were successful:

A screenshot of a computer

AI-generated content may be incorrect.

Home work #2

Go thru all the customers from JL\_XXX system.. load all customers into a cursor

fetch each record from CURSOR into a record by using FOR LOOP tech (or Manual OPEN and CLOSE cursor technique) and then

for each fetched record check and process only the customer who provided an email address.. If customer has NULL email address not an action is taken

IF email is NOT NULL (email is provided) then

if customer is from NW region then assign 5$ credit via a local variable and print eligibly and the credit amount

if customer is from SE region, then assign 7$ credit via a local variable and print eligibly and the credit amount

if customer is from N region, then assign 10$ credit via a local variable and print eligibly and the credit amount

Every for other region then assign 2$ credit via a local variable

ELSE if email is NULL then print ' this customer is not eligible for online campaign'

DECLARE

CURSOR cust\_cursor IS SELECT \* FROM JL\_CUSTOMERS;

cust\_record cust\_cursor%ROWTYPE;

v\_credit NUMBER;

BEGIN

FOR cust\_record IN cust\_cursor LOOP

IF cust\_record.email IS NOT NULL THEN

IF cust\_record.region = 'NW' THEN v\_credit := 5;

ELSIF cust\_record.region = 'SE' THEN v\_credit := 7;

ELSIF cust\_record.region = 'N' THEN v\_credit := 10;

ELSE v\_credit := 2;

END IF;

DBMS\_OUTPUT.PUT\_LINE(cust\_record.firstname || ' ' || cust\_record.lastname || ' is eligible for the credit amount of ' || v\_credit);

ELSE DBMS\_OUTPUT.PUT\_LINE(cust\_record.firstname || ' ' || cust\_record.lastname || ' is NOT eligible for online campaign');

END IF;

END LOOP;

END;

A screenshot of a computer

AI-generated content may be incorrect.The DBMS output and code: