-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- Name: Prince Jodhani

-- Student ID: 149455206

-- Date: 19-03-2022

-- Purpose: Lab 6 DBS311

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**-- Question 1 –** Write a stored procedure named calculate\_salary which gets an employee ID from the user and for that employee, calculates the salary based on the number of years the employee has been working in the company. (Use a loop construct the calculation of the salary).

**-- Q1 SOLUTION –**

create or replace PROCEDURE calculate\_salary (empid NUMBER)

IS

firstname employees.first\_name%type;

lastname employees.last\_name%type;

finalsalary varchar(50);

base\_salary NUMBER := 10000;

emp\_cn NUMBER := 0;

year NUMBER := 0;

BEGIN

SELECT COUNT(\*) INTO emp\_cn FROM EMPLOYEES WHERE EMPLOYEE\_ID = empid;

IF emp\_cn > 0 THEN

SELECT EXTRACT(YEAR FROM SYSDATE)- EXTRACT(YEAR FROM HIRE\_DATE), first\_name, last\_name INTO year, firstname, lastname

FROM EMPLOYEES WHERE EMPLOYEE\_ID = empid;

LOOP

year := year - 1;

EXIT WHEN year = 0;

base\_salary := base\_salary + (base\_salary\*.05);

END LOOP;

SELECT TO\_CHAR(base\_salary , 'fmU99G999D00') INTO finalsalary FROM DUAL;

dbms\_output.put\_line( 'FirstName: ' || firstname );

dbms\_output.put\_line( 'LastName: ' || lastname );

dbms\_output.put\_line( 'Salary: ' || finalsalary );

ELSE

dbms\_output.put\_line( 'Employee not found!' );

END IF;

EXCEPTION

WHEN INVALID\_NUMBER THEN

DBMS\_OUTPUT.PUT\_LINE('Error!');

ROLLBACK;

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Error!');

ROLLBACK;

END calculate\_salary;

Graphical user interface, text, application, email

Description automatically generated

**-- Question 2 –** Write a stored procedure named employee\_works\_here to print the employee\_id, employee Last name and department name.(USING FOR LOOP)

**-- Q2 SOLUTION –**

create or replace PROCEDURE employee\_works\_here

IS

empno employees.employee\_id%type;

firstname employees.first\_name%type;

lastname employees.last\_name%type;

emp\_cn NUMBER := 0;

year NUMBER := 0;

BEGIN

SELECT COUNT(\*) INTO emp\_cn FROM EMPLOYEES;

IF emp\_cn > 0 THEN

dbms\_output.put\_line( 'Employee # ' || chr(9) || chr(9)||'Last Name: ' ||chr(9)|| 'Department Name ');

LOOP

SELECT EXTRACT(YEAR FROM SYSDATE)- EXTRACT(YEAR FROM HIRE\_DATE), first\_name, last\_name, EMPLOYEE\_ID INTO year, firstname, lastname ,empno

FROM EMPLOYEES WHERE employee\_id = emp\_cn

ORDER BY EMPLOYEE\_ID ASC;

EXIT WHEN emp\_cn = 0;

dbms\_output.put\_line( empno ||chr(9)||chr(9)||chr(9)||chr(9)|| lastname ||chr(9)||chr(9)|| 'no department name ');

emp\_cn := emp\_cn - 1;

END LOOP;

ELSE

dbms\_output.put\_line( 'Employee not found!' );

END IF;

EXCEPTION

WHEN INVALID\_NUMBER THEN

DBMS\_OUTPUT.PUT\_LINE('Error!');

ROLLBACK;

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Error!');

ROLLBACK;

END employee\_works\_here;



**-- Question 3 –** Write a PL/SQL anonymous block to provide a list of customers name and credit limit using CURSORS.

**-- Q3 SOLUTION –**

DECLARE

cust\_name customers.name%type;

credit customers.credit\_limit%type;

cat varchar(50);

CURSOR cus\_rows IS

SELECT name, credit\_limit FROM customers;

BEGIN

OPEN cus\_rows;

LOOP

FETCH cus\_rows INTO cust\_name, credit;

EXIT WHEN cus\_rows%NOTFOUND;

if credit < 1000 then

cat := 'new customer';

elsif credit >= 1000 AND credit <= 4000 then

cat := 'existing customers';

else

cat := 'credit approved for new increase';

end if;

DBMS\_OUTPUT.PUT\_LINE(cust\_name || chr(9)|| credit || chr(9)|| cat) ;

END LOOP;

CLOSE cus\_rows;

END;

Text

Description automatically generated

**-- Question 4 –** Write a PL/SQL anonymous block to provide a list of customers name and credit limit using CURSORS.

**-- Q4 SOLUTION –**

DECLARE

cnt NUMBER := 0;

oid NUMBER := 0;

cust NUMBER := 0;

status varchar(50) := 0;

salsman varchar(50) := 0;

orderdate date;

BEGIN

SELECT COUNT(\*) cnt INTO cnt FROM ORDERS ;

WHILE cnt > 0

LOOP

SELECT ORDER\_ID, CUSTOMER\_ID , STATUS , SALESMAN\_ID , ORDER\_DATE INTO oid, cust, status, salsman, orderdate

FROM ORDERS WHERE ORDER\_ID = cnt;

DBMS\_OUTPUT.PUT\_LINE (oid || ' ' || cust || ' '|| status || ' ' || salsman || ' ' || orderdate);

cnt := cnt - 1;

END LOOP;

END;

A picture containing text, receipt

Description automatically generated

**-- Question 5 –** Write a PL/SQL anonymous block to provide a list of customers name and credit limit using CURSORS.

**-- Q5 SOLUTION –**

Add new\_price column

ALTER TABLE PRODUCTS

ADD new\_price number;

DECLARE

prod\_id products.product\_id%type;

list\_price products.list\_price%type;

price products.list\_price%type;

cat varchar(50);

CURSOR cur\_rows IS

SELECT product\_id, list\_price FROM products

FOR UPDATE OF new\_price;

BEGIN

OPEN cur\_rows;

LOOP

FETCH cur\_rows INTO prod\_id, list\_price;

EXIT WHEN cur\_rows%NOTFOUND;

Case

when list\_price < 50 then

price := list\_price + 30;

when list\_price >= 50 AND list\_price <= 100 then

price := list\_price + 50;

when list\_price >= 200 AND list\_price <= 500 then

price := list\_price + 150;

else price := list\_price + 200;

end case;

DBMS\_OUTPUT.PUT\_LINE('product\_id: ' || prod\_id|| ' list\_price: ' || list\_price || ' new\_price: ' || price) ;

UPDATE PRODUCTS SET new\_price = price WHERE product\_id = prod\_id;

END LOOP;

CLOSE cur\_rows;

END;

Text

Description automatically generated