Exp No. 10 Date : 19-08-2019

CURSOR

AIM:

To study the use and implementation of cursors in PL/SQL.

QUESTIONS:

1. Create table student (id, name, m1, m2, m3, grade). Insert 5 tuples into it. Find the total, calculate grade and update the grade in the table.

```
cursor=# CREATE TABLE student(id INT, name TEXT, m1 INT, m2 INT, m3 INT, grade TEXT);
CREATE TABLE
cursor=# INSERT INTO student(id, name, m1, m2, m3) VALUES(88, 'anu', 39, 67, 92),
                                                        (10, 'jan', 58, 61, 29),
                                                        (30, 'karuna', 87, 79, 77),
                                                        (29, 'jossy', 39, 80, 45),
                                                        (50, 'hisham', 60, 70, 80);
INSERT 0 5
cursor=# SELECT * FROM student ;
 id |  name   | m1 | m2 | m3 | grade
----+---------+----+----+----+-------
 88 | anu     | 39 | 67 | 92 |
 10 | jan     | 58 | 61 | 29 |
 30 | karuna | 87 | 79 | 77 |
 29 | jossy   | 39 | 80 | 45 |
 50 | hisham | 60 | 70 | 80 |
(5 rows)
```

```
cursor=# CREATE OR REPLACE FUNCTION find_grade() RETURNS INTEGER AS $$
cursor$# DECLARE
cursor$#     total FLOAT;
cursor$#     my_cursor CURSOR FOR SELECT * FROM student;
cursor$#     my_record RECORD;
cursor$#
cursor$# BEGIN
cursor$#     OPEN my_cursor;
cursor$#     LOOP
cursor$#         FETCH my_cursor INTO my_record;
cursor$#         EXIT WHEN NOT FOUND;
cursor$#         total = CEIL(my_record.m1 + my_record.m2 + my_record.m3) / 3;
cursor$#
cursor$#         IF  total > 80
cursor$#             THEN UPDATE student SET grade = 'a' WHERE CURRENT OF my_cursor;
cursor$#         ELSIF total > 70 and total <= 80
cursor$#             THEN UPDATE student SET grade = 'b' WHERE CURRENT OF my_cursor;
cursor$#         ELSIF total > 60 and total <= 70
cursor$#             THEN UPDATE student SET grade = 'c' WHERE CURRENT OF my_cursor;
cursor$#         ELSIF total > 40 and total <= 60
cursor$#             THEN UPDATE student SET grade = 'd' WHERE CURRENT OF my_cursor;
cursor$#         ELSE
cursor$#             UPDATE student SET grade = 'f' WHERE CURRENT OF my_cursor;
cursor$#         END IF;
cursor$#
cursor$#     END LOOP;
cursor$#     CLOSE my_cursor;
cursor$#     RETURN 0;
cursor$# END
cursor$# $$ LANGUAGE PLPGSQL;
CREATE FUNCTION
cursor=# SELECT find_grade();
 find_grade
------------
          0
(1 row)
cursor=# SELECT * FROM student ;
 id | sname   | m1 | m2 | m3 | grade
----+---------+----+----+----+-------
 88 | anu     | 39 | 67 | 92 | c
 10 | jan     | 58 | 61 | 29 | d
 30 | karuna | 87 | 79 | 77 | a
 29 | jossy   | 39 | 80 | 45 | d
 50 | hisham | 60 | 70 | 80 | c
(5 rows)
```

2. Create bank_details (accno, name, balance, adate). Calculate the interest of the amount and insert into a new table with fields (accno, interest). Interest= 0.08*balance.

```
cursor=# CREATE TABLE bank_details(accno INT, name VARCHAR(15), balance INT, adate DATE);
CREATE TABLE
cursor=# insert into bank_details values(1001,'aby',3005,'10-oct-15'),
                                        (1002,'alan',4000,'05-may-95'),
                                        (1003,'amal',5000,'16-mar-92'),
                                        (1004,'jeffin',3500,'01-apr-50'),
                                        (1005,'majo',6600,'01-jan-01');
INSERT 0 5
cursor=# SELECT * FROM bank_details ;
 accno |  name  | balance |   adate
-------+--------+---------+------------
  1001 | aby    |    3005 | 2015-10-10
  1002 | alan   |    4000 | 1995-05-05
  1003 | amal   |    5000 | 1992-03-16
  1004 | jeffin |    3500 | 2050-04-01
  1005 | majo   |    6600 | 2001-01-01
(5 rows)

cursor=#
```

```
cursor=# CREATE TABLE new_bank(accno INT, interest INT);
CREATE TABLE
cursor=# CREATE OR REPLACE FUNCTION calculate_interest() RETURNS VOID AS $$
cursor$# DECLARE
cursor$#     my_cursor CURSOR FOR SELECT * FROM bank_details;
cursor$#     my_record RECORD;
cursor$# BEGIN
cursor$#     OPEN my_cursor;
cursor$#     LOOP
cursor$#         FETCH my_cursor INTO my_record;
cursor$#         EXIT WHEN NOT FOUND;
cursor$#         INSERT INTO new_bank VALUES (my_record.accno, my_record.balance*0.08);
cursor$#     END LOOP;
cursor$#     CLOSE my_cursor;
cursor$# END
cursor$# $$ LANGUAGE PLPGSQL;
CREATE FUNCTION
cursor=# SELECT calculate_interest();
 calculate_interest
--------------------


(1 row)

cursor=# SELECT * FROM new_bank ;
 accno | interest
-------+----------
  1001 |      240
  1002 |      320
  1003 |      400
  1004 |      280
  1005 |      528
(5 rows)

cursor=#
```

3. Create table people_list (id, name, dt_joining, place). If person's experience is above 10 years, put the tuple in table exp_list (id, name, experience).

```
cursor=# CREATE TABLE people_list(id INT, name VARCHAR(30), dt_joining DATE, place VARCHAR(30));
CREATE TABLE
cursor=# INSERT INTO people_list VALUES (101,'Robert','2005-04-03','CHY'),
cursor-#                                (102, 'Mathew', '2008-06-07', 'CHY'),
cursor-#                                (103, 'Luffy', '2005-04-15', 'FSN'),
cursor-#                                (104, 'Lucci', '2009-08-13', 'KTM'),
cursor-#                                (105, 'Law', '2005-04-12', 'WTC'),
cursor-#                                (106, 'Vivi', '2010-09-21', 'ABA');
INSERT 0 6
cursor=# SELECT * FROM people_list ;
 id  |  name  | dt_joining | place
-----+--------+------------+-------
 101 | Robert | 2005-04-03 | CHY
 102 | Mathew | 2008-06-07 | CHY
 103 | Luffy  | 2005-04-15 | FSN
 104 | Lucci  | 2009-08-13 | KTM
 105 | Law    | 2005-04-12 | WTC
 106 | Vivi   | 2010-09-21 | ABA
(6 rows)

cursor=#
```

```
cursor=# CREATE TABLE experiance_list(id INT, name TEXT, exp INT);
CREATE TABLE
cursor=# CREATE OR REPLACE FUNCTION calculate_experiance() RETURNS INTEGER AS $$
cursor$# DECLARE
cursor$#     my_cursor CURSOR FOR SELECT * FROM people_list;
cursor$#     my_record RECORD;
cursor$#     yd INT;
cursor$# BEGIN
cursor$#     OPEN my_cursor;
cursor$#     LOOP
cursor$#         FETCH FROM my_cursor INTO my_record;
cursor$#         EXIT WHEN NOT FOUND;
cursor$#         yd := date_part('year',age(my_record.dt_joining));
cursor$#         IF yd > 10 THEN
cursor$#             INSERT INTO experiance_list VALUES(my_record.id, my_record.name, yd);
cursor$#         END IF;
cursor$#     END LOOP;
cursor$#     CLOSE my_cursor;
cursor$#     RETURN 0;
cursor$# END
cursor$# $$ LANGUAGE PLPGSQL;

CREATE FUNCTION
cursor=#
cursor=# SELECT calculate_experiance();
 calculate_experiance
----------------------
                    0
(1 row)

cursor=# SELECT * FROM experiance_list ;
 id  |  name  | exp
-----+--------+-----
 101 | Robert |  14
 102 | Mathew |  11
 103 | Luffy  |  14
 105 | Law    |  14
(4 rows)

cursor=#
```

4. Create table employee_list(id, name, monthly salary).

If: annual salary < 60000, increment monthly salary by 25%
between 60000 and 200000, increment by 20%
between 200000 and 500000, increment by 15%
annual salary>500000, increment monthly salary by 10%

```
cursor=# create table emp_list(id INT, name varchar(20), m_sal INT);
CREATE TABLE
cursor=# insert into emp_list values(101,'Mathew',55000),
cursor-#                            (102,'Jose',80000),
cursor-#                            (103,'John',250000),
cursor-#                            (104,'Ann',600000);
INSERT 0 4
cursor=# SELECT * FROM emp_list;
 id  |  name  | m_sal
-----+--------+--------
 101 | Mathew |  55000
 102 | Jose   |  80000
 103 | John   | 250000
 104 | Ann    | 600000
(4 rows)

cursor=#
```

```
cursor=# CREATE OR REPLACE FUNCTION update_salary() RETURNS INTEGER AS $$
cursor$# DECLARE
cursor$#     my_cursor CURSOR FOR SELECT * FROM emp_list;
cursor$#     my_record RECORD;
cursor$# BEGIN
cursor$#     OPEN my_cursor;
cursor$#     LOOP
cursor$#         FETCH FROM my_cursor INTO my_record;
cursor$#         EXIT WHEN NOT FOUND;
cursor$#         IF my_record.m_sal*12 < 60000 THEN
cursor$#             UPDATE emp_list SET m_sal = m_sal*1.25  WHERE CURRENT OF my_cursor;
cursor$#         ELSIF my_record.m_sal*12 >= 60000 AND my_record.m_sal*12 < 200000 THEN
cursor$#             UPDATE emp_list SET m_sal = m_sal*1.20  WHERE CURRENT OF my_cursor;
cursor$#         ELSIF my_record.m_sal*12 >= 200000 AND my_record.m_sal*12 < 500000 THEN
cursor$#             UPDATE emp_list SET m_sal = m_sal*1.15  WHERE CURRENT OF my_cursor;
cursor$#         ELSIF my_record.m_sal*12 >= 500000 THEN
cursor$#             UPDATE emp_list SET m_sal = m_sal*1.10  WHERE CURRENT OF my_cursor;
cursor$#         END IF;
cursor$#     END LOOP;
cursor$#     RETURN 0;
cursor$# END;
cursor$# $$ LANGUAGE PLPGSQL;
CREATE FUNCTION
cursor=#
cursor=# SELECT update_salary();
 update_salary
---------------
             0
(1 row)

cursor=# SELECT * FROM emp_list;
 id  |  name  | m_sal
-----+--------+--------
 101 | Mathew |  60500
 102 | Jose   |  88000
 103 | John   | 275000
 104 | Ann    | 660000
(4 rows)

cursor=#
```

RESULT:

The cursor programs was executed successfully and the output was obtained.