Ayush Goyal

190905522 CSE D 62

DBS Lab 7 (Week 7) – Cursors

Note: Use University DB schema for the following, unless a different DB schema is explicitly specified

Cursors: CursorName %ISOPEN / FOUND / NOT FOUND:

1. **The HRD manager has decided to raise the salary of all the Instructors in a given department number by 5%. Whenever, any such raise is given to the instructor, a record for the same is maintained in the salary_raise table. It includes the Instuctor Id, the date when the raise was given and the actual raise amount. Write a PL/SQL block to update the salary of each Instructor and insert a record in the salary_raise table.salary_raise(Instructor_Id, Raise_date, Raise_amt)**

CODE:

```
create table salaryraise(
    id number(8),
    raise_date date,
    raise_amt number(8)
    );

declare dt constant varchar(20) := '09/06/2021';
cursor c is select * from instructor;
begin
    for ins in c loop
    insert into salaryraise values( ins.id, to_date(dt, 'dd/mm/yyyy'), ins.sal
ary * 0.05 );
end loop;
update instructor set salary = salary * 1.05;
end;
/
```

OUTPUT:

```
PL/SQL procedure successfully completed.

SQL> select * from salaryraise;

        ID RAISE_DAT  RAISE_AMT
---------- --------- ----------
     10101 09-JUN-21       3250
     12121 09-JUN-21       4500
     15151 09-JUN-21       2000
     22222 09-JUN-21       4988
     32343 09-JUN-21       3000
     33456 09-JUN-21       4568
     45565 09-JUN-21       3750
     58583 09-JUN-21       3100
     76543 09-JUN-21       4000
     76766 09-JUN-21       3600
     83821 09-JUN-21       4600

        ID RAISE_DAT  RAISE_AMT
---------- --------- ----------
     98345 09-JUN-21       4000

12 rows selected.

SQL>
```

CursorName%ROWCOUNT:

2. **Write a PL/SQL block that will display the ID, name, dept_name and tot_cred of the first 10 students with lowest total credit.**

CODE:

```
declare
cursor c is select * from student order by tot_cred asc;
stu student %rowtype;
cnt number(5);
begin
    cnt := 0;
    open c;
    loop fetch c into stu;
        dbms_output.put_line( 'ID:' || stu.id || ' Name:' || stu.name || ' Dep
t:' || stu.dept_name || ' Credits:' || stu.tot_cred );
        cnt := cnt + 1;
        exit when cnt >= 10;
    end loop;
    close c;
end;
/
```

OUTPUT:

```
ID:70557 Name:Snow Dept:Physics Credits:0
ID:12345 Name:Shankar Dept:Comp. Sci. Credits:32
ID:55739 Name:Sanchez Dept:Music Credits:38
ID:45678 Name:Levy Dept:Physics Credits:46
ID:54321 Name:Williams Dept:Comp. Sci. Credits:54
ID:44553 Name:Peltier Dept:Physics Credits:56
ID:76543 Name:Brown Dept:Comp. Sci. Credits:58
ID:76653 Name:Aoi Dept:Elec. Eng. Credits:60
ID:19991 Name:Brandt Dept:History Credits:80
ID:98765 Name:Bourikas Dept:Elec. Eng. Credits:98

PL/SQL procedure successfully completed.

SQL>
```

Cursor For Loops:

3. **Print the Course details and the total number of students registered for each course along with the course details -(Course-id, title, dept-name, credits, tot_student_no)**

CODE:

```
declare
cursor c is select course_id, title, dept_name, credits, tot
            from course natural join ( select course_id, count(*) as tot from
takes group by course_id );
begin
```

```
    for co in c
    loop dbms_output.put_line( 'ID: ' || co.course_id || ' Title: ' || co.titl
e || ' Dept: ' || co.dept_name || ' Credits: ' || co.credits || ' Total ' || c
o.tot );
    end loop;
end;
/
```

OUTPUT:

```
ID: BIO-101 Title: Intro. to Biology Dept: Biology Credits: 4 Total 1
ID: BIO-301 Title: Genetics Dept: Biology Credits: 4 Total 1
ID: CS-101 Title: Intro. to Computer Science Dept: Comp. Sci. Credits: 4 Total 7
ID: CS-190 Title: Game Design Dept: Comp. Sci. Credits: 4 Total 2
ID: CS-315 Title: Robotics Dept: Comp. Sci. Credits: 3 Total 2
ID: CS-319 Title: Image Processing Dept: Comp. Sci. Credits: 3 Total 2
ID: CS-347 Title: Database System Concepts Dept: Comp. Sci. Credits: 3 Total 2
ID: EE-181 Title: Intro. to Digital Systems Dept: Elec. Eng. Credits: 3 Total 1
ID: FIN-201 Title: Investment Banking Dept: Finance Credits: 3 Total 1
ID: HIS-351 Title: World History Dept: History Credits: 3 Total 1
ID: MU-199 Title: Music Video Production Dept: Music Credits: 3 Total 1
ID: PHY-101 Title: Physical Principles Dept: Physics Credits: 4 Total 1

PL/SQL procedure successfully completed.

SQL>
```

4. **Find all students who take the course with Course-id: CS101 and if he/ she has less than 30 total credit (tot-cred), deregister the student from that course. (Delete the entry in Takes table)**

CODE:

```
declare
cursor c is select * from takes where course_id = '747';
cre student.tot_cred %type;
cnt number(8);
begin cnt := 0;
    for s in c
    loop select tot_cred into cre from student where id = s.id;
        if cre < 30 then delete from takes where course_id = '747' and id = s.
id;
            dbms_output.put_line('deleted : ' || s.id || ' credits : ' || cre)
;
            cnt := cnt + 1;
        end if;
    end loop;
    dbms_output.put_line( cnt || ' students de-
enrolled from the course 747' );
end;
/
```

OUTPUT:

```
SQL> declare
  2  cursor c is select * from takes where course_id = '747';
  3  cre student.tot_cred %type;
  4  cnt number(8);
  5  begin cnt := 0;
  6      for s in c
  7      loop select tot_cred into cre from student where id = s.id;
  8          if cre < 30 then delete from takes where course_id = '747' and id = s.id;
  9              dbms_output.put_line('deleted : ' || s.id || ' credits : ' || cre);
 10              cnt := cnt + 1;
 11          end if;
 12      end loop;
 13      dbms_output.put_line( cnt || ' students de-enrolled from the course 747' );
 14  end;
 15  /
0 students de-enrolled from the course 747

PL/SQL procedure successfully completed.

SQL>
```

Where Current of:

5. **Alter StudentTable(refer Lab No. 8 Exercise)by resetting column LetterGrade to F. Then write a PL/SQL block to update the table by mapping GPA to the corresponding letter grade foreach student.**

CODE:

```
update studenttable set LetterGrade = 'F';

declare
cursor c is select * from studenttable for update;
begin
    for stu in c
    loop if stu.gpa > 4 and stu.gpa <= 5 then update studenttable set LetterGr
ade = 'e' where current of c;
        elsif stu.gpa > 5 and stu.gpa <= 6 then update studenttable set Letter
Grade = 'd' where current of c;
        elsif stu.gpa > 6 and stu.gpa <= 7 then update studenttable set Letter
Grade = 'c' where current of c;
        elsif stu.gpa > 7 and stu.gpa <= 8 then update studenttable set Letter
Grade = 'b' where current of c;
        elsif stu.gpa > 8 and stu.gpa <= 9 then update studenttable set Letter
Grade = 'a' where current of c;
        elsif stu.gpa > 9 and stu.gpa <= 10 then update studenttable set Lette
rGrade = 'a+' where current of c;
        end if;
    end loop;
end;
/

select * from studenttable;
```

OUTPUT:

```
PL/SQL procedure successfully completed.

SQL> select * from StudentTable;

    ROLLNO        GPA LE
---------- ---------- --
         1        5.8 d
         2        6.5 c
         3        3.4 F
         4        7.8 b
         5        9.5 a+

SQL>
```

Parameterized Cursors:

6. **Write a PL/SQL block to print the list of Instructors teaching a specified course.**

CODE:

```
declare
cursor c1(c_id teaches.course_id%type) is select * from (instructor natural jo
in teaches) where course_id = c_id;
temp teaches.course_id%type;
begin
    temp := '&Course_ID';
    for info in c1(temp)
    loop dbms_output.put_line(info.name);
    end loop;
end;
/
```

OUTPUT:

```
Enter value for course_id: CS-101
old   5:     temp := '&Course_ID';
new   5:     temp := 'CS-101';
Srinivasan
Katz

PL/SQL procedure successfully completed.

SQL>
```

7. **Write a PL/SQL block to list the students who have registered for a course taught by his/her advisor.**

CODE:

```
declare
cursor a is select unique t.id as s, s.id as i from takes t, teaches s where t
.course_id = s.course_id;
```

```
cursor b(s student.id %type, i instructor.id %type) is select unique s_id from
 advisor where s_id = s and i_id = i;
st student %rowtype;
cnt number(8);
begin
    cnt := 0;
    for tuple in a
    loop for stu in b(tuple.s, tuple.i)
        loop select * into st from student where id = stu.s_id;
            dbms_output.put_line(st.name || ' ' || st.id || ' ' || st.dept_nam
e);
            cnt := cnt + 1;
        end loop;
    end loop;
    dbms_output.put_line(cnt || ' rows selected');
end;
/
```

OUTPUT:

```
Peltier 44553 Physics
Shankar 12345 Comp. Sci.
Aoi 76653 Elec. Eng.
Brown 76543 Comp. Sci.
Zhang 00128 Comp. Sci.
Tanaka 98988 Biology
6 rows selected

PL/SQL procedure successfully completed.

SQL>
```

Transactions (COMMIT / ROLLBACK / SAVEPOINT):

8. **Write a PL/SQL block that updates the salary of 'Biology' department instructors by 20%. Subsequently, check the whether the department budget can support the raise. If not, undo the raise given to the instructors.**

CODE:

```
declare
cursor c is select * from instructor where dept_name = 'Biology' for update;
cnt number(20);
temp number(20);
begin
    savepoint a;
    cnt := 0;
    for ins in c
    loop cnt := cnt + ins.salary * 1.2;
        update instructor set salary = salary * 1.2 where current of c;
```

```
    end loop;
    select budget into temp from department where dept_name = 'Biology';
    if temp < cnt then rollback to savepoint a;
    else commit;
    end if;
end;
/

select * from instructor where dept_name = 'Biology';
select * from department where dept_name = 'Biology';
```

OUTPUT:

```
PL/SQL procedure successfully completed.

SQL> select * from instructor where dept_name = 'Biology';

ID    NAME                 DEPT_NAME                SALARY
----- -------------------- -------------------- ----------
76766 Crick                Biology                   75600

SQL> select * from department where dept_name = 'Biology';

DEPT_NAME            BUILDING            BUDGET
-------------------- --------------- ----------
Biology              Watson               90000

SQL>
```

THE END