**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.salary \* 0.05 );

end loop;

update instructor set salary = salary \* 1.05;

end;

/

**OUTPUT:Text

Description automatically generated with low confidence**

**CursorName%ROWCOUNT:**

1. **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 || ' Dept:' || stu.dept\_name || ' Credits:' || stu.tot\_cred );

        cnt := cnt + 1;

        exit when cnt >= 10;

    end loop;

    close c;

end;

/

**OUTPUT:**

**Text

Description automatically generated**

**Cursor For Loops:**

1. **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.title || ' Dept: ' || co.dept\_name || ' Credits: ' || co.credits || ' Total ' || co.tot );

    end loop;

end;

/

**OUTPUT:**

**Graphical user interface, text

Description automatically generated**

1. **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:**

**Text

Description automatically generated**

**Where Current of:**

1. **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 LetterGrade = 'e' where current of c;

        elsif stu.gpa > 5 and stu.gpa <= 6 then update studenttable set LetterGrade = 'd' where current of c;

        elsif stu.gpa > 6 and stu.gpa <= 7 then update studenttable set LetterGrade = 'c' where current of c;

        elsif stu.gpa > 7 and stu.gpa <= 8 then update studenttable set LetterGrade = 'b' where current of c;

        elsif stu.gpa > 8 and stu.gpa <= 9 then update studenttable set LetterGrade = 'a' where current of c;

        elsif stu.gpa > 9 and stu.gpa <= 10 then update studenttable set LetterGrade = 'a+' where current of c;

        end if;

    end loop;

end;

/

select \* from studenttable;

**OUTPUT:**

**Shape

Description automatically generated with medium confidence**

**Parameterized Cursors:**

1. **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 join 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:**

**Shape

Description automatically generated with medium confidence**

1. **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\_name);

            cnt := cnt + 1;

        end loop;

    end loop;

    dbms\_output.put\_line(cnt || ' rows selected');

end;

/

**OUTPUT:**

**Shape

Description automatically generated with medium confidence**

**Transactions (COMMIT / ROLLBACK / SAVEPOINT):**

1. **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:**

**Text

Description automatically generated**

**THE END**