**Oracle Lab 7: Cursor**

***Assoc. Prof. Dr. Nguyen Thai Nghe***

*Refer to this link* [*http://plsql-tutorial.com/plsql-cursors.htm*](http://plsql-tutorial.com/plsql-cursors.htm)

[*http://plsql-tutorial.com/plsql-explicit-cursors.htm*](http://plsql-tutorial.com/plsql-explicit-cursors.htm)

*Save the examples to* ***StudentID-lab7****.sql.*

Cursor is usually used to store the results of the SELECT command.

Using cursor: Declare → Open → Processing → Close

**Declare cursor**

CURSOR csName [(parameters>)] IS Select\_command;

**Example:**

Cursor without parameters

DECLARE

CURSOR c\_Dept1 IS

SELECT deptno, dname

FROM dept

WHERE deptno > 10;

BEGIN

--do something here;

END;

Cursor with parameters

CURSOR c\_Dept2(p\_Deptno NUMBER) IS

SELECT deptno, dname

FROM dept

WHERE deptno>p\_Deptno;

BEGIN

--do something here;

END;

**Open cursor**

OPEN [csName [(para\_values)] ];

**Example:**

OPEN c\_Dept1; -- without params

OPEN c\_Dept2(10); -- with params

**Getting data from cursor**

FETCH csName INTO csVariable;

**Close cursor**

CLOSE cursor\_name;

**example:**

DECLARE

CURSOR c\_Emp IS

SELECT \*

FROM EMP

WHERE deptno = 10;

v\_Emp c\_EMP%ROWTYPE;

BEGIN

OPEN c\_Emp;

dbms\_output.put\_line('EMPNO | ENAME | JOB');

LOOP

FETCH c\_Emp INTO v\_Emp;

EXIT WHEN c\_emp%NOTFOUND;

dbms\_output.put\_line(v\_Emp.empno || ' | ' || v\_Emp.ename || ' | ' || v\_emp.job);

END LOOP;

CLOSE c\_Emp;

END;

**Cursor Properties**

| **Properties** | **Desc** |
| --- | --- |
| %notfound | Before the first fetch from an open cursor, cursor\_name%NOTFOUND returns NULL. Thereafter, it returns FALSE if the last fetch returned a row, or TRUE if the last fetch failed to return a row. |
| %found | Before the first fetch from an open cursor, cursor\_name%FOUND returns NULL. Afterward, it returns TRUE if the last fetch returned a row, or FALSE if the last fetch failed to return a row. |
| %rowcount | When a cursor is opened, %ROWCOUNT is zeroed. Before the first fetch, cursor\_name%ROWCOUNT returns 0. Thereafter, it returns the number of rows fetched so far. The number is incremented if the latest fetch returned a row. |
| %isopen | If a cursor is open, cursor\_name%ISOPEN returns TRUE; otherwise, it returns FALSE. |

**Try the following blocks using both LOOP and FOR**

Open the following tables in Scott schema to observe data

**EMP**( EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO)

**BONUS**(ENAME, JOB, SAL, COMM)

Scott schema can be dowloaded here <https://github.com/oracle/dotnet-db-samples/blob/master/schemas/scott.sql>

* + - * 1. **Example 1:** The Employees (in scott) who have worked >=30 years but having salary <= 2000$ will be increased the commission (comm) to 500$, then saving to the Bonus table.

DECLARE

CURSOR c\_Emp IS

select \* from emp

where extract(year from sysdate) - extract(year from hiredate)>= 30

and sal <=2000;

v\_Emp c\_Emp%rowtype;

BEGIN

OPEN c\_Emp;

LOOP

FETCH c\_Emp INTO v\_Emp;

EXIT WHEN c\_Emp%notfound;

if v\_Emp.comm is null then

v\_Emp.comm:=500;

else

v\_Emp.comm:=v\_Emp.comm+500;

end if;

UPDATE Emp

SET comm = v\_Emp.comm

WHERE empno = v\_Emp.empno;

INSERT INTO BONUS(ename, job, sal, comm)

VALUES (v\_Emp.ename, v\_Emp.job, v\_Emp.sal, v\_Emp.comm);

END LOOP;

COMMIT;

CLOSE c\_Emp;

END;

* + - * 1. **Example 2:** The employees (scott) who have worked >=43 years will be retired. Save them to the EMP\_RETIRE table and remove them from the Emp table.

*--Create EMP\_RETIRE table*

CREATE TABLE EMP\_RETIRE(

EMPNO NUMBER(4) NOT NULL,

ENAME VARCHAR2(10),

JOB VARCHAR2(9),

MGR NUMBER(4),

HIREDATE DATE,

SAL NUMBER(7,2),

COMM NUMBER(7,2),

DEPTNO NUMBER(2)

);

*--Using cursor*

DECLARE

CURSOR cEmpRetire IS

Select \*

From emp

where extract(year from sysdate) -

extract(year from hiredate) >= 43

FOR UPDATE;

BEGIN

**FOR emp\_rec IN cEmpRetire**

LOOP

INSERT INTO Emp\_Retire(Empno, EName, Job, Mgr,

Hiredate, Sal, Comm, Deptno)

VALUES (emp\_rec.Empno, emp\_rec.EName, emp\_rec.Job, emp\_rec.Mgr, emp\_rec.Hiredate, emp\_rec.Sal, emp\_rec.Comm, emp\_rec.Deptno);

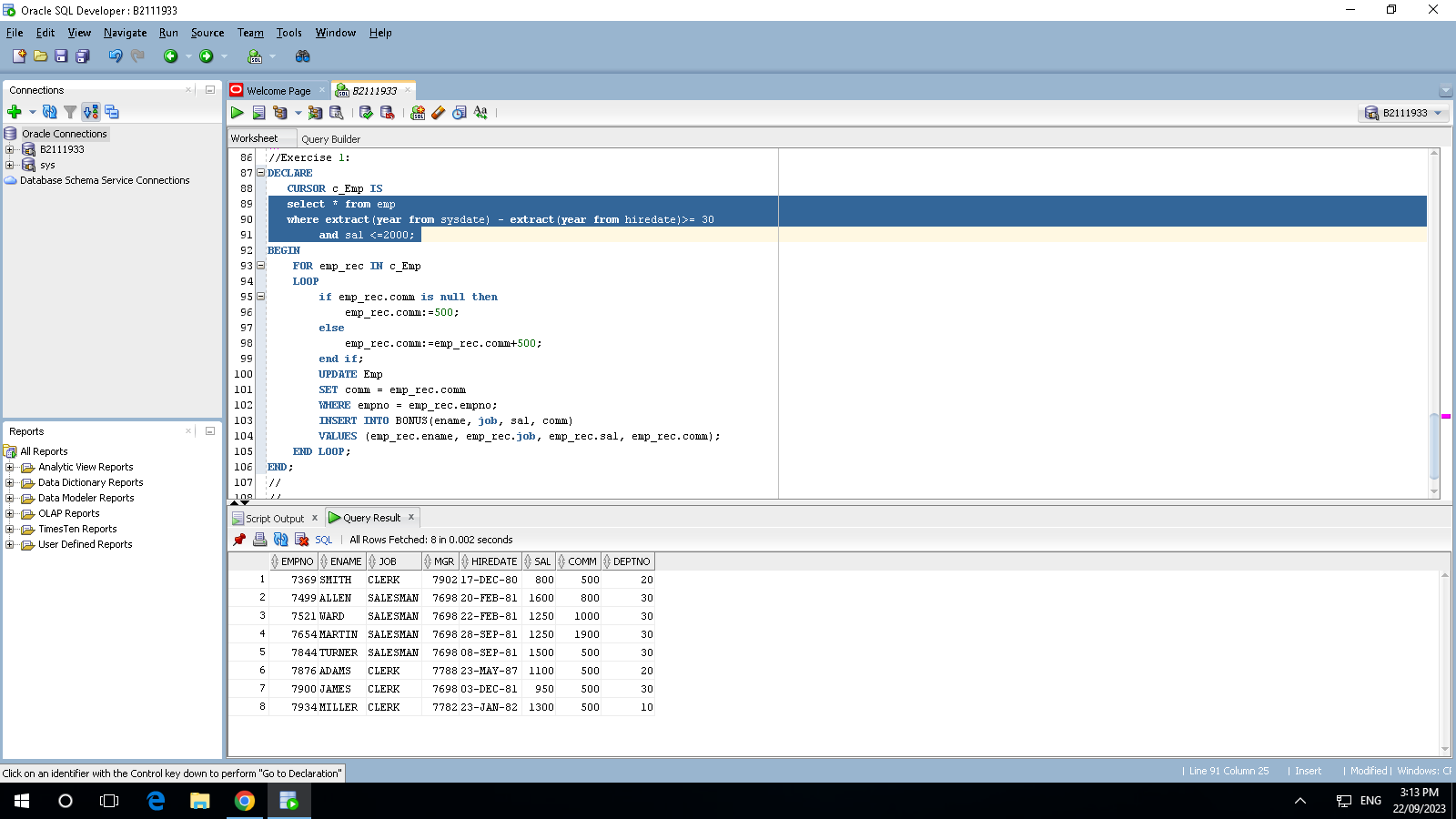
DELETE FROM Emp

WHERE CURRENT OF cEmpRetire;

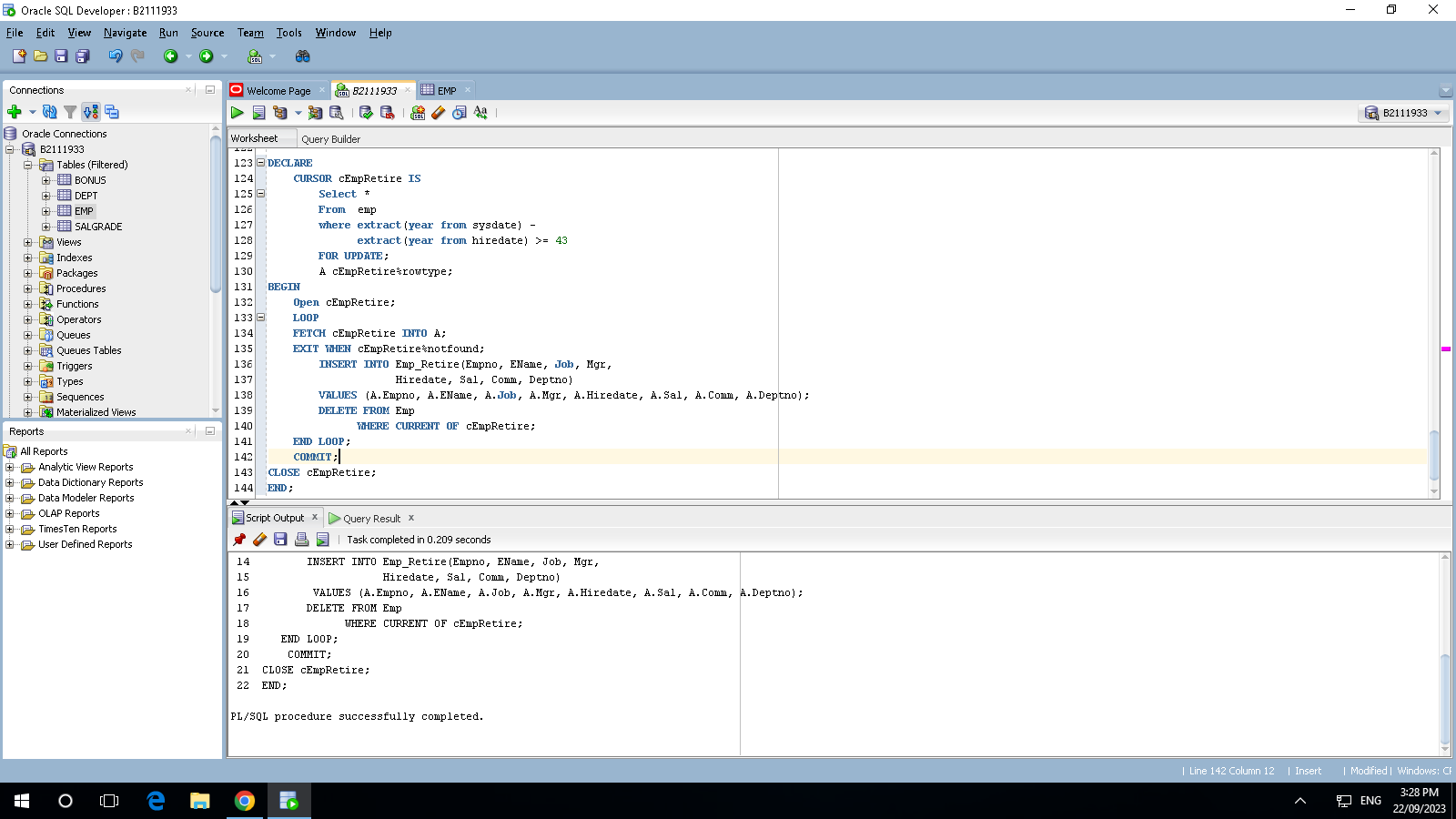
END LOOP;

END;

**Redo the example1 using FOR (similar as example 2)**

****

**Redo the example2 using FETCH (similar as example 1)**

****

**Try all examples from the following links:**

* [*http://plsql-tutorial.com/plsql-cursors.htm*](http://plsql-tutorial.com/plsql-cursors.htm)
* [*http://plsql-tutorial.com/plsql-explicit-cursors.htm*](http://plsql-tutorial.com/plsql-explicit-cursors.htm)