**PLSQL/Checkpoint 2**

---1

CREATE TABLE emp\_static AS

SELECT employee\_id, first\_name, last\_name

FROM employees;

DECLARE

emp\_id emp\_static.employee\_id%TYPE := 285;

emp\_f\_name emp\_static.first\_name%TYPE := 'Meera';

emp\_l\_name emp\_static.last\_name%TYPE := 'Kumar';

BEGIN

INSERT INTO emp\_static (employee\_id, first\_name, last\_name)

VALUES (emp\_id, emp\_f\_name, emp\_l\_name);

UPDATE emp\_static

SET first\_name = 'Meera'

WHERE employee\_id = emp\_id;

DELETE FROM emp\_static

WHERE employee\_id = emp\_id

RETURNING first\_name, last\_name

INTO emp\_f\_name, emp\_l\_name;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE (emp\_f\_name || ' ' || emp\_l\_name);

END;

--2

CREATE TABLE emp\_static1 AS

SELECT employee\_id, first\_name, last\_name

FROM employees;

DECLARE

seq\_value NUMBER;

BEGIN

seq\_value := emp\_seq.NEXTVAL;

DBMS\_OUTPUT.PUT\_LINE ('Initial sequence : ' || TO\_CHAR(seq\_value) );

INSERT INTO emp\_static (employee\_id, first\_name, last\_name) VALUES (emp\_seq.CURRVAL,'Meera', 'Kumar');

INSERT INTO emp\_static1 VALUES (emp\_seq.CURRVAL,'Yedhu', 'Raj');

seq\_value := emp\_seq.CURRVAL;

DELETE FROM emp\_static

WHERE employee\_id = seq\_value;

UPDATE emp\_static

SET employee\_id = emp\_seq.NEXTVAL

WHERE first\_name = 'Meera'

AND last\_name = 'Kumar';

seq\_value := emp\_seq.CURRVAL;

DBMS\_OUTPUT.PUT\_LINE ('Ending sequence : ' || TO\_CHAR(seq\_value));

END;

---3

CREATE TABLE employees\_temp AS SELECT \* FROM employees;

DECLARE

mgr\_no NUMBER(6) := 100;

BEGIN

DELETE FROM employees\_temp WHERE manager\_id = mgr\_no;

DBMS\_OUTPUT.PUT\_LINE

('Number of employees deleted: ' || TO\_CHAR(SQL%ROWCOUNT));

END;

---4

DECLARE

emp\_first\_name VARCHAR2(35);

emp\_last\_name VARCHAR2(35);

temp\_id NUMBER:=&employee\_id;

BEGIN

SELECT first\_name,

last\_name

INTO emp\_first\_name, emp\_last\_name

FROM employees

WHERE employee\_id = temp\_id;

dbms\_output.Put\_line ('Employee name: ' || emp\_first\_name ||' ' ||emp\_last\_name);

EXCEPTION

WHEN no\_data\_found THEN

dbms\_output.Put\_line ('There is no employee with the ID '||to\_char(zemp\_id));

END;

---5

CREATE TABLE emp\_temp AS

SELECT employee\_id, first\_name, last\_name

FROM employees;

CREATE OR REPLACE PROCEDURE test\_proc (temp\_emp\_id NUMBER) AUTHID DEFINER AS

BEGIN

DELETE FROM emp\_temp

WHERE employee\_id = temp\_emp\_id;

IF SQL%FOUND THEN

DBMS\_OUTPUT.PUT\_LINE ('Delete succeeded for employee\_id: ' ||temp\_emp\_id);

ELSE

DBMS\_OUTPUT.PUT\_LINE ('No employee of ID '|| temp\_emp\_id||'found');

END IF;

END;

/

BEGIN

test\_proc(7);

test\_proc(6);

END;

--6

DECLARE

t\_emp\_id NUMBER:=&employee\_id;

BEGIN

UPDATE emp\_temp

SET first\_name = 'not available'

WHERE employee\_id = t\_emp\_id;

IF SQL%NOTFOUND THEN

DBMS\_OUTPUT.PUT\_LINE ('No employee of ID '|| t\_emp\_id||' found.');

ELSE

DBMS\_OUTPUT.PUT\_LINE ('Update succeeded for employee\_id: ' || t\_emp\_id);

END IF;

END;

---7

DECLARE

v\_employee employees%ROWTYPE;

v\_emp\_id NUMBER:=&employee\_id;

BEGIN

SELECT \* INTO v\_employee FROM employees WHERE employee\_id = v\_emp\_id;

dbms\_output.Put\_line (v\_employee.first\_name||' '||v\_employee.last\_name||' has an ID of '||v\_emp\_id);

dbms\_output.Put\_line (v\_employee.first\_name||' '||v\_employee.last\_name||'is not found');

EXCEPTION

WHEN no\_data\_found THEN

Raise\_application\_error(-20001, 'The Employee '|| 'is not in the database');

END;

--8

DECLARE

v\_employee employees%ROWTYPE;

BEGIN

SELECT \* INTO v\_employee FROM employees WHERE employee\_id = 149;

dbms\_output.Put\_line(' Details : ID:---' ||v\_employee.employee\_id ||' Name: '||v\_employee.first\_name||' '||v\_employee.last\_name||' Salary: '||v\_employee.salary);

END;

---9

DECLARE

CURSOR v\_emp\_info IS SELECT employee\_id, first\_name, last\_name, salary FROM employees;

var\_emp\_info v\_emp\_info%ROWTYPE;

BEGIN

OPEN v\_emp\_info;

LOOP

FETCH v\_emp\_info INTO var\_emp\_info;

EXIT

WHEN v\_emp\_info%NOTFOUND;

dbms\_output.Put\_line('Details:---- ' ||' ID: ' ||var\_emp\_info.employee\_id ||' Name: '||var\_emp\_info.first\_name ||' '||var\_emp\_info.last\_name);

END LOOP;

dbms\_output.Put\_line('Total row number = ' ||v\_emp\_info%rowcount);

CLOSE v\_emp\_info;

END;

--10

DECLARE

v\_empid employees.employee\_id%TYPE;

v\_empname employees.first\_name%TYPE;

v\_salary employees.salary%TYPE;

CURSOR emp\_cursor IS

SELECT employee\_id,

first\_name,

salary

FROM employees;

BEGIN

OPEN emp\_cursor;

LOOP

FETCH emp\_cursor

INTO v\_empid,

v\_empname,

v\_salary;

EXIT

WHEN emp\_cursor%NOTFOUND;

IF (v\_salary > 5000) THEN

dbms\_output.Put\_line(v\_empid|| ' ' || v\_empname|| ' '|| v\_salary);

ELSE

dbms\_output.Put\_line(v\_empname|| ' salary less than 5000');

END IF;

END LOOP;

CLOSE emp\_cursor;

END;