SQL> SET FEEDBACK ON;

**Query-01**

SQL> CREATE OR REPLACE PROCEDURE SHOW\_EMPLOYEE(ENO IN EMPP.EID%TYPE) IS

2 EMPREC EMPP%ROWTYPE;

3 BEGIN

4 SELECT \* INTO EMPREC FROM EMPP WHERE EID=ENO;

5 DBMS\_OUTPUT.PUT\_LINE('EID: '||EMPREC.EID||' ENAME:'||EMPREC.ENAME||' HIREDATE:'||EMPREC.HIREDATE||' SALARY:'||EMPREC.SALARY);

6

7 EXCEPTION

8 WHEN NO\_DATA\_FOUND THEN

9 DBMS\_OUTPUT.PUT\_LINE('NO RECORD EXISTS WITH EMP ID:='||ENO);

10 WHEN OTHERS THEN

11 DBMS\_OUTPUT.PUT\_LINE(TO\_CHAR(SQLCODE)||'::'||TO\_CHAR(SQLERRM));

12 END;

13 /

Procedure created.

SQL>

SQL> DECLARE

2 ENO EMPP.EID%TYPE;

3 BEGIN

4 DBMS\_OUTPUT.PUT\_LINE('ENTER EMPLOYEE ID:');

5 ENO:='&ENO';

6 SHOW\_EMPLOYEE(ENO);

7 END;

8 /

Enter value for eno: 7103

old 5: ENO:='&ENO';

new 5: ENO:='7103';

ENTER EMPLOYEE ID:

EID: 7103 ENAME:Julia Martin HIREDATE:01-DEC-99 SALARY:13320

PL/SQL procedure successfully completed.

**Query-02**

SQL> CREATE OR REPLACE PROCEDURE ADD\_EMPLOYEE(EMPREC EMPP%ROWTYPE) IS

2 BEGIN

3 INSERT INTO EMPP VALUES(EMPREC.EID,EMPREC.ENAME,EMPREC.HIREDATE,EMPREC.SALARY);

4 DBMS\_OUTPUT.PUT\_LINE('RECORD INSERTED');

5 EXCEPTION

6 WHEN OTHERS THEN

7 DBMS\_OUTPUT.PUT\_LINE(TO\_CHAR(SQLCODE)||'::'||TO\_CHAR(SQLERRM));

8 END;

9 /

Procedure created.

SQL>

SQL> SELECT RPAD(OBJECT\_NAME,15,' ') OBJECT\_NAME,OBJECT\_TYPE FROM USER\_OBJECTS WHERE OBJECT\_NAME='ADD\_EMPLOYEE';

OBJECT\_NAME OBJECT\_TYPE

--------------- -------------------

ADD\_EMPLOYEE PROCEDURE

1 row selected.

SQL>

SQL> DECLARE

2 EMPREC EMPP%ROWTYPE;

3 BEGIN

4 EMPREC.EID:=7111;

5 EMPREC.ENAME:='Justin Beiber';

6 EMPREC.HIREDATE:='01-Jan-2016';

7 EMPREC.SALARY:=12500;

8 ADD\_EMPLOYEE(EMPREC);

9 EMPREC.EID:=7112;

10 EMPREC.ENAME:='Wilfred Diesel';

11 EMPREC.HIREDATE:='02-Feb-2016';

12 EMPREC.SALARY:=13000;

13 ADD\_EMPLOYEE(EMPREC);

14 END;

15 /

RECORD INSERTED

RECORD INSERTED

PL/SQL procedure successfully completed.

SQL>

SQL> SELECT \* FROM EMPP WHERE EID IN(7111,7112);

EID ENAME HIREDATE SALARY

---------- --------------------- --------- ----------

7111 Justin Beiber 01-JAN-16 12500

7112 Wilfred Diesel 02-FEB-16 13000

2 rows selected.

SQL>

SQL> ROLLBACK;

Rollback complete.

**Query-03**

SQL> CREATE OR REPLACE PROCEDURE REMOVE\_EMPLOYEE(EMPNAME IN EMPP.ENAME%TYPE) IS

2 EMPID EMPP.EID%TYPE;

3 BEGIN

4 SELECT EID INTO EMPID FROM EMPP WHERE UPPER(ENAME)=UPPER(EMPNAME);

5 DELETE FROM EMPP WHERE EID=EMPID;

6 DBMS\_OUTPUT.PUT\_LINE('RECORD DELETED');

7 EXCEPTION

8 WHEN NO\_DATA\_FOUND THEN

9 DBMS\_OUTPUT.PUT\_LINE('NO RECORD FOR '||EMPNAME);

10 WHEN OTHERS THEN

11 DBMS\_OUTPUT.PUT\_LINE(TO\_CHAR(SQLCODE)||'::'||TO\_CHAR(SQLERRM));

12 END;

13 /

Procedure created.

SQL>

SQL> EXEC REMOVE\_EMPLOYEE('YOGESH JOG');

NO RECORD FOR YOGESH JOG

PL/SQL procedure successfully completed.

SQL> EXEC REMOVE\_EMPLOYEE('Julia Martin');

RECORD DELETED

PL/SQL procedure successfully completed.

SQL>

SQL> ROLLBACK;

Rollback complete.

**Query-04**

SQL> CREATE TABLE ITEMS AS

2 SELECT P\_CODE,DESCRIPT AS DESCR,P\_DATE AS IN\_DATE,

3 P\_MIN AS MIN\_QTY,QTY,P\_PRICE AS PRICE ,V\_CODE FROM PRODUCT;

Table created.

SQL>

SQL>

SQL> ALTER TABLE ITEMS ADD PRIMARY KEY (P\_CODE);

Table altered.

SQL>

SQL> ALTER TABLE ITEMS MODIFY IN\_DATE DEFAULT SYSDATE;

Table altered.

SQL> ALTER TABLE ITEMS MODIFY MIN\_QTY DEFAULT 2;

Table altered.

SQL>

SQL> CREATE OR REPLACE FUNCTION CHECK\_ITEM(PCODE ITEMS.P\_CODE%TYPE)

2 RETURN NUMBER IS

3 CNT NUMBER;

4 N INT(1) :=1 ;

5 BEGIN

6 SELECT COUNT(\*) INTO CNT FROM ITEMS WHERE P\_CODE=PCODE;

7 IF CNT>=1 THEN

8 RETURN N;

9 ELSE

10 N:=0;

11 RETURN N;

12 END IF;

13 END;

14 /

Function created.

SQL>

SQL> BEGIN

2 DBMS\_OUTPUT.PUT\_LINE('STATUS: '||CHECK\_ITEM('JB008'));

3 DBMS\_OUTPUT.PUT\_LINE('STATUS: '||CHECK\_ITEM('JB555'));

4 END;

5 /

STATUS: 1

STATUS: 0

PL/SQL procedure successfully completed.

**Query-05**

SQL> CREATE OR REPLACE FUNCTION IREC\_CREATOR(ICODE ITEMS.P\_CODE%TYPE,IDESC ITEMS.DESCR%TYPE,IQTY ITEMS.QTY%TYPE,IPRICE ITEMS.PRICE%TYPE,VCODE ITEMS.V\_CODE%TYPE)

2 RETURN ITEMS%ROWTYPE IS

3 IREC ITEMS%ROWTYPE;

4 BEGIN

5 IREC.P\_CODE:=ICODE;

6 IREC.DESCR:=IDESC;

7 IREC.IN\_DATE:=SYSDATE;

8 IREC.MIN\_QTY:=NULL;

9 IREC.QTY:=IQTY;

10 IREC.PRICE:=IPRICE;

11 IREC.V\_CODE:=NULL;

12 RETURN IREC;

13 END;

14 /

Function created.

SQL>

SQL> CREATE OR REPLACE PROCEDURE ADD\_ITEM(IREC IN ITEMS%ROWTYPE)IS

2 BEGIN

3 INSERT INTO ITEMS(P\_CODE,DESCR,QTY,PRICE,V\_CODE) VALUES(IREC.P\_CODE,IREC.DESCR,IREC.QTY,IREC.PRICE,IREC.V\_CODE);

4 DBMS\_OUTPUT.PUT\_LINE('RECORD INSERTED');

5 END;

6 /

Procedure created.

SQL>

SQL> EXEC ADD\_ITEM(IREC\_CREATOR('YJ304','DIGGER',6,999.99,25595));

RECORD INSERTED

PL/SQL procedure successfully completed.

**Query-06**

SQL> CREATE OR REPLACE PROCEDURE UPDATE\_ITEM (ICODE IN ITEMS.P\_CODE%TYPE,IQTY IN ITEMS.QTY%TYPE,IPRICE IN ITEMS.PRICE%TYPE) IS

2 BEGIN

3 IF CHECK\_ITEM(ICODE)=1 THEN

4 UPDATE ITEMS SET QTY= IQTY, PRICE=IPRICE WHERE P\_CODE=ICODE;

5 DBMS\_OUTPUT.PUT\_LINE('RECORD UPDATED');

6 ELSE

7 RAISE NO\_DATA\_FOUND;

8 END IF;

9 EXCEPTION

10 WHEN NO\_DATA\_FOUND THEN

11 DBMS\_OUTPUT.PUT\_LINE('NO DATA FOUND FOR ITEM CODE: '||ICODE);

12 WHEN OTHERS THEN

13 DBMS\_OUTPUT.PUT\_LINE(TO\_CHAR(SQLCODE)||'::'||TO\_CHAR(SQLERRM));

14 END;

15 /

Procedure created.

SQL>

SQL> EXEC UPDATE\_ITEM('YJ404',10,599.99);

NO DATA FOUND FOR ITEM CODE: YJ404

PL/SQL procedure successfully completed.

SQL> EXEC UPDATE\_ITEM('YJ304',10,599.99);

RECORD UPDATED

PL/SQL procedure successfully completed.

**Query-07**

SQL> CREATE OR REPLACE PROCEDURE UPDATE\_ITEM\_ADD\_WHEN\_NOT\_FOUND (ICODE IN ITEMS.P\_CODE%TYPE,IQTY IN ITEMS.QTY%TYPE,IPRICE IN ITEMS.PRICE%TYPE) IS

2 IREC ITEMS%ROWTYPE;

3 BEGIN

4 IF CHECK\_ITEM(ICODE)=1 THEN

5 UPDATE ITEMS SET QTY= IQTY, PRICE=IPRICE WHERE P\_CODE=ICODE;

6 DBMS\_OUTPUT.PUT\_LINE('RECORD UPDATED');

7 ELSE

8 IREC.P\_CODE:=ICODE;

9 IREC.DESCR:='NEW ITEM';

10 IREC.IN\_DATE:=SYSDATE;

11 IREC.MIN\_QTY:=IQTY/8;

12 IREC.QTY:=IQTY;

13 IREC.PRICE:=IPRICE;

14 IREC.V\_CODE:=NULL;

15 ADD\_ITEM(IREC);

16 END IF;

17 END;

18 /

Procedure created.

SQL>

SQL> EXEC UPDATE\_ITEM\_ADD\_WHEN\_NOT\_FOUND('YJ404',10,599.99);

RECORD INSERTED

PL/SQL procedure successfully completed.

**Query-08**

SQL> CREATE OR REPLACE PROCEDURE SHOW\_ITEM(ICODE ITEMS.P\_CODE%TYPE) IS

2 IREC ITEMS%ROWTYPE;

3 BEGIN

4 IF CHECK\_ITEM(ICODE)=1 THEN

5 SELECT \* INTO IREC FROM ITEMS WHERE P\_CODE=ICODE;

6 DBMS\_OUTPUT.PUT\_LINE(IREC.P\_CODE||' '||RPAD(IREC.DESCR,15,' ')||' '||IREC.IN\_DATE||' '||IREC.MIN\_QTY||' '||IREC.QTY||' '||IREC.PRICE||' '||IREC.V\_CODE);

7 ELSE

8 RAISE NO\_DATA\_FOUND;

9 END IF;

10 EXCEPTION

11 WHEN NO\_DATA\_FOUND THEN

12 DBMS\_OUTPUT.PUT\_LINE(ICODE||' NO DATA FOUND');

13 END;

14 /

Procedure created.

SQL>

SQL>

SQL> BEGIN

2 SHOW\_ITEM('HH15P');

3 SHOW\_ITEM('HH15X');

4 SHOW\_ITEM('YJ304');

5 SHOW\_ITEM('SHU00');

6 SHOW\_ITEM('SH100');

7 SHOW\_ITEM('MP100');

8 END;

9 /

HH15P NO DATA FOUND

HH15X HANGING HOOK 15 10-JAN-13 25 200 5.75 24992

YJ304 DIGGER 02-APR-17 2 10 599.99

SHU00 NO DATA FOUND

SH100 Sledge Hammer 02-JAN-12 5 8 14.4

MP100 NO DATA FOUND

PL/SQL procedure successfully completed.

**Query-09**

SQL> ALTER TABLE ITEMS DROP PRIMARY KEY;

Table altered.

SQL> EXEC ADD\_ITEM(IREC\_CREATOR('HH15P','NEW ITEM ..',150,00,25));

RECORD INSERTED

PL/SQL procedure successfully completed.

SQL> EXEC ADD\_ITEM(IREC\_CREATOR('HH15P','NEW ITEM-2',150,00,25));

RECORD INSERTED

PL/SQL procedure successfully completed.

SQL>

SQL> CREATE OR REPLACE PROCEDURE SHOW\_ITEM\_TMR\_HANDLED(ICODE ITEMS.P\_CODE%TYPE) IS

2 IREC ITEMS%ROWTYPE;

3 BEGIN

4 IF CHECK\_ITEM(ICODE)=1 THEN

5 SELECT \* INTO IREC FROM ITEMS WHERE P\_CODE=ICODE;

6 DBMS\_OUTPUT.PUT\_LINE(IREC.P\_CODE||' '||RPAD(IREC.DESCR,15,' ')||' '||IREC.IN\_DATE||' '||IREC.MIN\_QTY||' '||IREC.QTY||' '||IREC.PRICE||' '||IREC.V\_CODE);

7 ELSE

8 RAISE NO\_DATA\_FOUND;

9 END IF;

10 EXCEPTION

11 WHEN TOO\_MANY\_ROWS THEN

12 DBMS\_OUTPUT.PUT\_LINE('MULTIPLE ROWS DETECTED FOR ITEM CODE : '||ICODE);

13 WHEN NO\_DATA\_FOUND THEN

14 DBMS\_OUTPUT.PUT\_LINE(ICODE||' NO DATA FOUND');

15 WHEN OTHERS THEN

16 DBMS\_OUTPUT.PUT\_LINE(TO\_CHAR(SQLCODE)||'::'||TO\_CHAR(SQLERRM));

17 END;

18 /

Procedure created.

SQL>

SQL> BEGIN

2 SHOW\_ITEM\_TMR\_HANDLED('HH15P');

3 SHOW\_ITEM\_TMR\_HANDLED('HH15X');

4 END;

5 /

MULTIPLE ROWS DETECTED FOR ITEM CODE : HH15P

HH15X HANGING HOOK 15 10-JAN-13 25 200 5.75 24992

PL/SQL procedure successfully completed.

**Query-10**

SQL> CREATE OR REPLACE PROCEDURE SHOW\_ITEM\_TMR\_HANDLED(ICODE ITEMS.P\_CODE%TYPE) IS

2 IREC ITEMS%ROWTYPE;

3 CURSOR IRECS IS SELECT \* FROM ITEMS WHERE P\_CODE=ICODE;

4 BEGIN

5 IF CHECK\_ITEM(ICODE)=1 THEN

6 SELECT \* INTO IREC FROM ITEMS WHERE P\_CODE=ICODE;

7 DBMS\_OUTPUT.PUT\_LINE(IREC.P\_CODE||' '||RPAD(IREC.DESCR,15,' ')||' '||IREC.IN\_DATE||' '||IREC.MIN\_QTY||' '||IREC.QTY||' '||IREC.PRICE||' '||IREC.V\_CODE);

8 ELSE

9 RAISE NO\_DATA\_FOUND;

10 END IF;

11 EXCEPTION

12 WHEN TOO\_MANY\_ROWS THEN

13 DBMS\_OUTPUT.PUT\_LINE('ERROR MULTIPLE ROWS DETECTED FOR ITEM CODE : '||ICODE);

14 OPEN IRECS;

15 LOOP

16 FETCH IRECS INTO IREC;

17 EXIT WHEN IRECS % NOTFOUND;

18 DBMS\_OUTPUT.PUT\_LINE(IREC.P\_CODE||' '||RPAD(IREC.DESCR,15,' ')||' '||IREC.IN\_DATE||' '||IREC.MIN\_QTY||' '||IREC.QTY||' '||IREC.PRICE||' '||IREC.V\_CODE);

19 END LOOP;

20 CLOSE IRECS;

21 WHEN NO\_DATA\_FOUND THEN

22 DBMS\_OUTPUT.PUT\_LINE(ICODE||' NO DATA FOUND');

23 WHEN OTHERS THEN

24 DBMS\_OUTPUT.PUT\_LINE(TO\_CHAR(SQLCODE)||'::'||TO\_CHAR(SQLERRM));

25 END;

26 /

Procedure created.

SQL>

SQL> BEGIN

2 SHOW\_ITEM\_TMR\_HANDLED('HH15P');

3 SHOW\_ITEM\_TMR\_HANDLED('HH15X');

4 END;

5 /

ERROR MULTIPLE ROWS DETECTED FOR ITEM CODE : HH15P

HH15P NEW ITEM .. 02-APR-17 2 150 0

HH15P NEW ITEM-2 02-APR-17 2 150 0

HH15X HANGING HOOK 15 10-JAN-13 25 200 5.75 24992

PL/SQL procedure successfully completed.

SQL> -------------------------------------------------------------------------------------------------

SQL> COMMIT;

Commit complete.

SQL> SET FEEDBACK OFF;

SQL> SET ECHO OFF;