Triggers examples

```
CREATE OR REPLACE TRIGGER Log_emp_update
AFTER UPDATE ON Emp_tab
BEGIN
        INSERT INTO Emp_log (Log_date, Action)
        VALUES (SYSDATE, 'Emp_tab COMMISSIONS CHANGED');
END;

UPDATE Emp_tab SET Sal = Sal + 1000.0 WHERE Deptno = 20; - a command
causing the trigger to fire
```

Trigger run conditionally

```
CREATE OR REPLACE TRIGGER Log_salary_increase
AFTER UPDATE ON Emp_tab
FOR EACH ROW
WHEN (new.Sal > 1000)
BEGIN
        INSERT INTO Emp_log (Emp_id, Log_date, New_salary, Action)   VALUES
(:new.Empno, SYSDATE, :new.SAL, 'NEW SAL');
END;
```

Syntax of a stored procedure:

```
CREATE [OR REPLACE] PROCEDURE [schema.]procedure
    [ (argument [IN | OUT | IN OUT] datatype
    [, argument [IN | OUT | IN OUT] datatype] ...)]
    {IS | AS} pl/sql_subprogram_body ]
```

Example 1) Procedure that inserts data into the table.

```
CREATE TABLE T2 (
a INTEGER,     b
CHAR(10)
);


CREATE PROCEDURE addtuple1(i IN NUMBER) AS
BEGIN
    INSERT INTO T2 VALUES(i, 'xxx');
END addtuple1;
```

Example 2)

```
CREATE PROCEDURE remove_emp (employee_id NUMBER) AS
tot_emps NUMBER;
BEGIN
        DELETE FROM employees
      WHERE employees.employee_id = remove_emp.employee_id;
tot_emps := tot_emps - 1;
END;
```

Example 3) Procedure containing section that deals with an exception.

```
CREATE OR REPLACE PROCEDURE MYPROC1
(REQISBN IN NUMBER,
MYVAR1 IN OUT CHAR,
TCOST OUT NUMBER)
TEMP_COST NUMBER(10,2))
IS BEGIN
   SELECT COST INTO TEMP_COST FROM JD11.BOOK WHERE ISBN = REQISBN;
   IF TEMP_COST > 0 THEN
      UPDATE JD11.BOOK SET COST = (TEMP_COST*1.175) WHERE ISBN = REQISBN;
   ELSE
      UPDATE JD11.BOOK SET COST = 21.32 WHERE ISBN = REQISBN;
   END IF;
   TCOST := TEMP_COST;
   COMMIT;
EXCEPTION
   WHEN NO_DATA_FOUND THEN
      INSERT INTO JD11.ERRORS (CODE, MESSAGE) VALUES(99, 'ISBN NOT FOUND');
END MYPROC1;
```

EXECUTE JD11.MYPROC1 (21, :PCOST) – runs the procedure in SQL *Plus (PCOST is a bind variable name of the SQL *PLUS environment). 21 will be passed to the procedure, the TCOST value will be passed outside of the procedure and remembered in :PCOST.

Example 4) Another procedure containing an extended *exception* section

```
PROCEDURE raise_salary (emp_id INTEGER, amount REAL) IS
current_salary REAL;    salary_missing EXCEPTION;
BEGIN
   SELECT sal INTO current_salary FROM emp
   WHERE empno = emp_id;
   IF current_salary IS NULL THEN
      RAISE salary_missing;
   ELSE
      UPDATE emp SET sal = sal + amount
      WHERE empno = emp_id;
   END IF;
EXCEPTION
   WHEN NO_DATA_FOUND THEN
      INSERT INTO emp_audit VALUES (emp_id, 'No such number');
   WHEN salary_missing THEN
      INSERT INTO emp_audit VALUES (emp_id, 'Salary is null'); END;
```

Example 5) Example of a function – as opposed to the procedure it returns a value.

```
CREATE OR REPLACE FUNCTION MYFUNC1
RETURN NUMBER
IS RCOST NUMBER(10,2);
BEGIN
   SELECT COST INTO RCOST FROM JD11.BOOK WHERE ISBN = 21;
   RETURN (RCOST); END
MYFUNC1;
```