Task 1:

CREATE TABLE STUDENTS(

STUDENT\_NAME VARCHAR2(250),

CGPA NUMBER

) ;

CREATE OR REPLACE TRIGGER task\_1

AFTER INSERT

ON STUDENTS

FOR EACH ROW

BEGIN

DBMS\_OUTPUT.PUT\_LINE('The CGPA of Student name ' || :NEW.STUDENT\_NAME ||

' has been inserted');

DBMS\_OUTPUT.PUT\_LINE('The CGPA is ' || :NEW.CGPA);

END;

/

INSERT INTO STUDENTS VALUES ('Bijoy Ahmed', 3.87);

/



TASK-2:

CREATE OR REPLACE TRIGGER task\_2

BEFORE INSERT

ON STUDENTS

FOR EACH ROW

BEGIN

DBMS\_OUTPUT.PUT\_LINE('The CGPA of Student name ' || :NEW.STUDENT\_NAME ||

' will be inserted');

DBMS\_OUTPUT.PUT\_LINE('The CGPA will be ' || :NEW.CGPA);

END;

/

INSERT INTO STUDENTS VALUES ('Saeem Ahmed', 3.87);

/



**Task 3-1:**

CREATE OR REPLACE TRIGGER task\_3\_1

BEFORE INSERT OR DELETE

ON STUDENTS

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Record has been inserted or deleted');

END;

/

INSERT INTO STUDENTS VALUES ('SOUMIK SARKAR', 3.85);



**Task 3-2**:

CREATE OR REPLACE TRIGGER task\_3\_2

AFTER INSERT OR DELETE

ON STUDENTS

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Record has been inserted or deleted');

END;

/

INSERT INTO STUDENTS VALUES ('SOUMIK AHMED', 3.85);



**Task 4**:

CREATE OR REPLACE TRIGGER task\_4

AFTER UPDATE

OF CGPA

ON STUDENTS

BEGIN

DBMS\_OUTPUT.PUT\_LINE('CGPA has been updated');

END;

/

UPDATE STUDENTS SET CGPA = CGPA \* 1.10 WHERE CGPA <= 3.70;



**Task 5**:

CREATE OR REPLACE TRIGGER task\_5

BEFORE UPDATE

OF CGPA

ON STUDENTS

FOR EACH ROW

BEGIN

DBMS\_OUTPUT.PUT\_LINE('CGPA of student ' || :OLD.STUDENT\_NAME

|| ' has been updated with new CGPA ' || :NEW.CGPA);

DBMS\_OUTPUT.PUT\_LINE('The previous CGPA of the student was ' || :OLD.CGPA);

END;

/

UPDATE STUDENTS SET CGPA = CGPA \* 0.9 WHERE CGPA >= 3.50;



**Task 6:**

CREATE TABLE LOG\_TABLE\_CGPA\_UPDATE

(

USERNAME VARCHAR2(25),

DATETIME DATE,

OLD\_CGPA NUMBER,

NEW\_CGPA NUMBER

) ;

CREATE OR REPLACE TRIGGER LOG\_CGPA\_UPDATE

AFTER UPDATE

OF CGPA

ON STUDENTS

FOR EACH ROW

DECLARE

USERNAME VARCHAR2(25);

BEGIN

USERNAME := USER; --USER is a function that returns current username

INSERT INTO LOG\_TABLE\_CGPA\_UPDATE VALUES (USERNAME, SYSDATE,:OLD.CGPA, :NEW.CGPA);

END ;

/

UPDATE STUDENTS SET CGPA = CGPA + 0.01 ;

**Task 7**:

CREATE TABLE STUDENTS\_DELETED(

STUDENT\_NAME VARCHAR2(25),

CGPA NUMBER,

USERNAME VARCHAR2(25),

DATETIME DATE

) ;

CREATE OR REPLACE TRIGGER BACKUP\_DELETED\_STUDENTS

BEFORE DELETE

ON STUDENTS

FOR EACH ROW

DECLARE

V\_NAME VARCHAR2(25);

V\_USERNAME VARCHAR2(25);

V\_CGPA NUMBER;

V\_DATETIME DATE;

BEGIN

V\_NAME := :OLD.STUDENT\_NAME ;

V\_CGPA := :OLD.CGPA ;

V\_USERNAME := USER ;

V\_DATETIME := SYSDATE ;

INSERT INTO STUDENTS\_DELETED VALUES (V\_NAME,V\_CGPA,V\_USERNAME,V\_DATETIME);

END ;

/

DELETE FROM STUDENTS WHERE CGPA < 3.85 ;

**Task 8**:

CREATE TABLE LOW\_CGPA\_STUDENTS (

STUDENT\_NAME VARCHAR2(25)

) ;

CREATE OR REPLACE TRIGGER task\_8

BEFORE INSERT OR UPDATE

OF CGPA

ON STUDENTS

FOR EACH ROW

BEGIN

IF (:NEW.CGPA < 2.O) THEN

INSERT INTO LOW\_CGPA\_STUDENTS VALUES(:OLD.STUDENT\_NAME);

END IF;

END;

/