**Assignment-12**

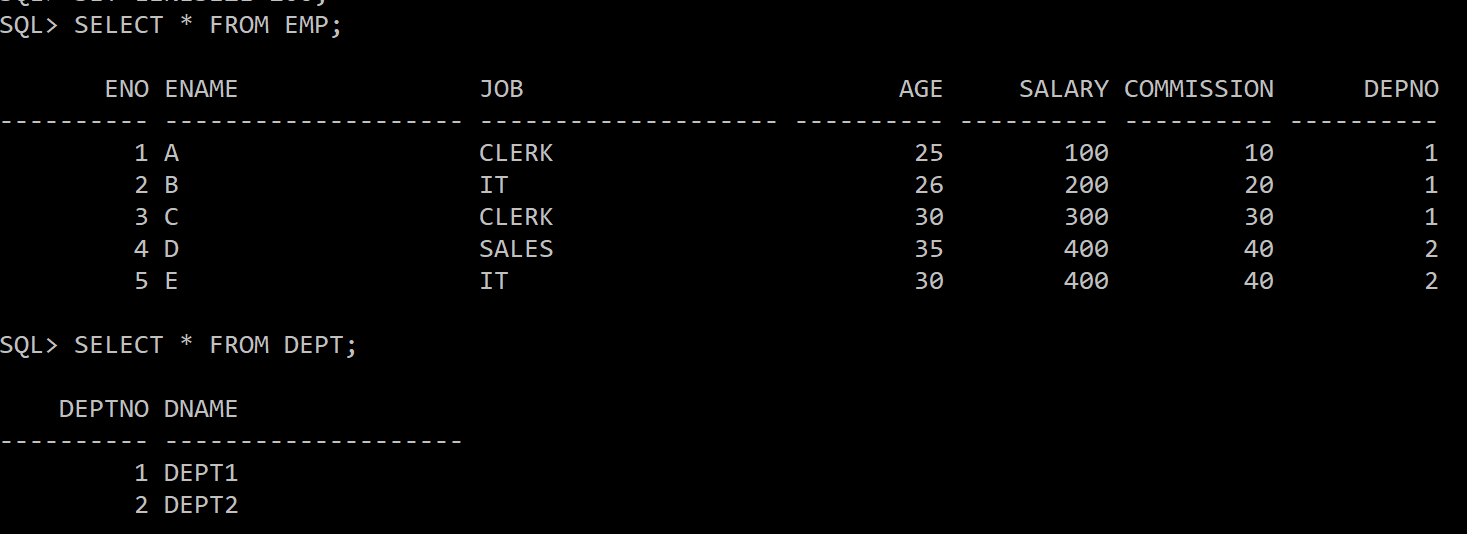
**(Database Triggers in PLSQL)**

**EMP (ENO, ENAME, JOB, AGE, SALARY, COMMISSION, DEPTNO).**

**DEPT (DEPTNO, DNAME)**

**(Deptno of EMP is a foreign key which refers to the Deptno of Dept Table)**

**Assume and create required tables with appropriate attributes, if any**

****

**1. Write a database trigger on the EMP table, which will store each deleted**

**record from the EMP table in another table called EMP\_OLD.**

CREATE OR REPLACE TRIGGER EONE

BEFORE DELETE ON EMP

FOR EACH ROW

BEGIN

INSERT INTO EMP\_OLD VALUES(:OLD.ENO, :OLD.ENAME, :OLD.JOB, :OLD.AGE, :OLD.SALARY, :OLD.COMMISSION, :OLD.DEPTNO);

END;

/

A black screen with white text

Description automatically generated

**2. Create a database trigger on the EMP table, verifying that no employee is**

**under 25 while entering the details (ENO, AGE) into the EMP table.**

CREATE TRIGGER ETWO

AFTER INSERT ON EMP FOR EACH ROW

BEGIN

IF :NEW.AGE<25 THEN

RAISE\_APPLICATION\_ERROR(-20000, 'AGE IS LESS THAN 25');

END IF;

END;

/

**A black screen with white text

Description automatically generated**

**3. Write a database trigger on the EMP table, which verifies that each employee's**

**updated salary should not less than their old salary.**

CREATE TRIGGER ETHREE

BEFORE UPDATE ON EMP FOR EACH ROW

BEGIN

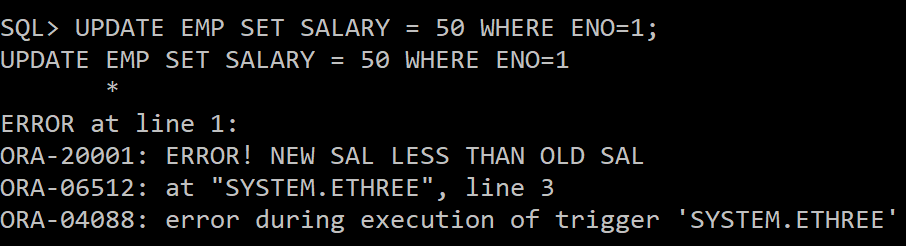
IF :NEW.SALARY<:OLD.SALARY THEN

RAISE\_APPLICATION\_ERROR(-20001,'ERROR! NEW SAL LESS THAN OLD SAL');

END IF;

END;

/

****

**4. Write a database trigger which is an example of a statement-level trigger.**

CREATE TRIGGER EFOUR

BEFORE UPDATE ON EMP

BEGIN

DBMS\_OUTPUT.PUT\_LINE('TABLE EMP IS UPDATED');

END;

/

**A black screen with white text

Description automatically generated**

--here four rows are updated but the trigger is only fired once

**5. Create a trigger on the EMP table to convert supplied values of the ENAME**

**column to uppercase for INSERT and UPDATE statements.**

**CREATE TRIGGER EFIVE**

**BEFORE INSERT OR UPDATE ON EMP**

**FOR EACH ROW**

**BEGIN**

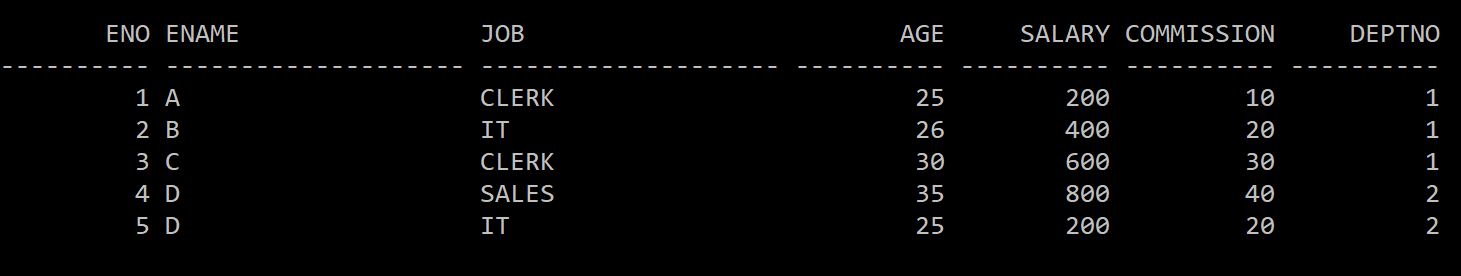
**:NEW.ENAME:=UPPER(:NEW.ENAME);**

**END;**

**/**

**INSERT INTO EMP VALUES(5,’d’,’IT’, 25, 200, 20, 2);**

**SELECT \* FROM EMP;**

****

**6. Create a trigger on the emp table, which shows the old and new values of**

**ENAME after every updation on the ENAME of the EMP table.**

CREATE TRIGGER ESIX

AFTER UPDATE ON EMP

FOR EACH ROW

BEGIN

IF :OLD.ENAME <> :NEW.ENAME THEN

DBMS\_OUTPUT.PUT\_LINE('OLD NAME IS: '||:OLD.ENAME);

DBMS\_OUTPUT.PUT\_LINE('NEW NAME IS: '||:NEW.ENAME);

END IF;

END;

/

UPDATE EMP SET ENAME='E' WHERE ENO=5;

A black screen with white text

Description automatically generated

**7. Write a trigger to ensure that an employee's commission cannot be greater**

**than his salary.**

CREATE TRIGGER ESEVEN

BEFORE INSERT OR UPDATE ON EMP

FOR EACH ROW

BEGIN

IF :NEW.COMMISSION > :NEW.SALARY THEN

RAISE\_APPLICATION\_ERROR(-20003, 'COMMISSION CANNOT BE GREATER THAN SALARY');

END IF;

END;

/

INSERT INTO EMP VALUES(7,'F','IT',28,10,30,2);

A black screen with white text

Description automatically generated

**8. Create a trigger so that no operation can be performed on the EMP table on**

**Sunday.**

CREATE TRIGGER EEIGHT

BEFORE INSERT OR DELETE OR UPDATE ON EMP

FOR EACH ROW

DECLARE

N VARCHAR2(20);

BEGIN

SELECT TO\_CHAR(SYSDATE,'DAY') INTO N FROM DUAL;

IF N='SUNDAY' THEN

RAISE\_APPLICATION\_ERROR(-20004, 'CANNOT MAKE CHANGES ON SUNDAY');

END IF;

END;

/

**9. Create a trigger to implement the primary key constraint on column ENO of**

**table EMP.**

CREATE TRIGGER ENINE

BEFORE INSERT OR UPDATE ON EMP

FOR EACH ROW

BEGIN

IF :NEW.ENO IS NULL THEN

RAISE\_APPLICATION\_ERROR(-20000,'PRIMARY KEY NOT NULL');

END IF;

FOR R IN (SELECT ENO FROM EMP WHERE ENO=:NEW.ENO) LOOP

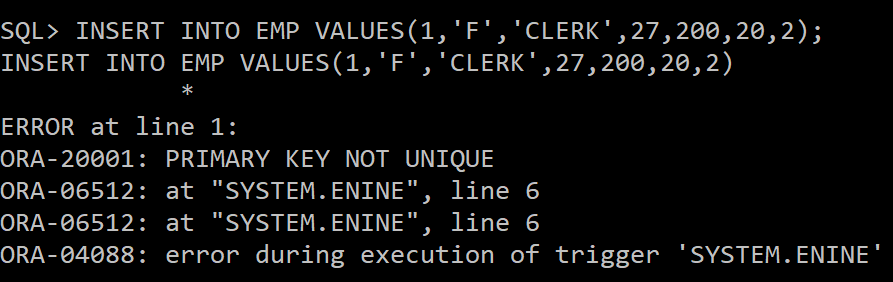
RAISE\_APPLICATION\_ERROR(-20001,'PRIMARY KEY NOT UNIQUE');

END LOOP;

END;

/

INSERT INTO EMP VALUES(1,'F','CLERK',27,200,20,2);



**10. Write a trigger to implement a foreign key constraint on the DEPTNO column**

**of the EMP table which refers to the DEPTNO of DEPT Table.**

CREATE OR REPLACE TRIGGER ETEN

BEFORE UPDATE OR INSERT ON EMP

FOR EACH ROW

DECLARE

D NUMBER;

BEGIN

SELECT DEPT.DEPTNO INTO D FROM DEPT WHERE DEPTNO=:NEW.DEPTNO;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RAISE\_APPLICATION\_ERROR(-20005,'FOREIGN KEY CONSTRAINT VIOLATED');

END;

/

INSERT INTO EMP VALUES (6,'F','SALES',28,200,20,3);

