QUESTATION NO 1  
  
CREATE TABLE NEWEMP(

EMPNO NUMBER,

ENAME VARCHAR2(50),

JOB VARCHAR2(50),

MGR NUMBER,

HIREDATE DATE,

SAL NUMBER,

COMM NUMBER,

DEPTNO NUMBER);

CREATE OR REPLACE TRIGGER display\_salary\_changes

BEFORE DELETE OR INSERT OR UPDATE ON EMP

FOR EACH ROW

WHEN (NEW.EMPNO > 0)

DECLARE

  sal\_diff number;

BEGIN

  dbms\_output.put\_line ('ALERT CHANGE IN EMP\_TABLE');

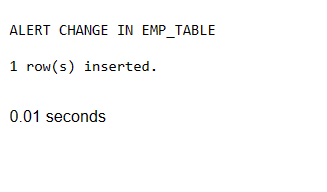
  INSERT INTO NEWEMP (EMPNO,ENAME,JOB,MGR,HIREDATE,SAL,COMM,DEPTNO)

  VALUES (:OLD.EMPNO,:OLD.ENAME,:OLD.JOB,:OLD.MGR,:OLD.HIREDATE,:OLD.SAL,:OLD.COMM,:OLD.DEPTNO);

END;

INSERT INTO emp (empno, ename, job, mgr, hiredate, sal, comm, deptno)

VALUES (1, 'John Smith', 'Manager', null, TO\_DATE('2023-05-08', 'YYYY-MM-DD'), 5000, null, 10);

OUTPUT  
  


Q2

CREATE OR REPLACE TRIGGER display\_salary\_changes BEFORE DELETE OR INSERT OR UPDATE ON EMP

FOR EACH ROW

WHEN (NEW.SAL > OLD.SAL\*0.1)

DECLARE

sal\_diff number;

BEGIN

dbms\_output.put\_line (:OLD.SAL);

dbms\_output.put\_line (:NEW.SAL);

dbms\_output.put\_line (:NEW.EMPNO);

dbms\_output.put\_line (:NEW.ENAME);

dbms\_output.put\_line (:NEW.JOB);

dbms\_output.put\_line (:NEW.MGR);

dbms\_output.put\_line (:NEW.HIREDATE);

dbms\_output.put\_line (:NEW.SAL);

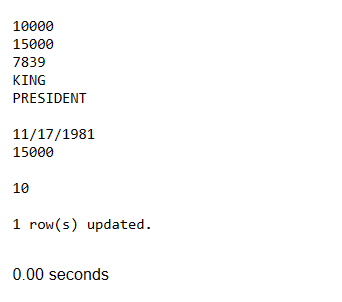
dbms\_output.put\_line (:NEW.COMM);

dbms\_output.put\_line (:NEW.DEPTNO);

END;

UPDATE emp SET sal = 15000

WHERE ename = 'KING';

OUTPUT  
  
Q3

Create or replace Trigger DATA\_DEL

before delete on dept

For EACH ROW

Begin

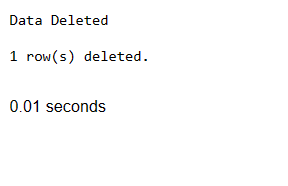
  delete from emp where deptno=20;

  dbms\_output.put\_line('Data Deleted');

END;

delete from dept

where deptno=20;

OUTPUT  
  
Q4  
Create view NEW\_VIEW as

Select e.empno,e.ename,e.sal,d.deptno,d.dname

from emp e join dept d on e.deptno=d.deptno;

Create or replace Trigger V\_TR

Instead of Update on NEW\_VIEW

For EACH ROW

BEGIN

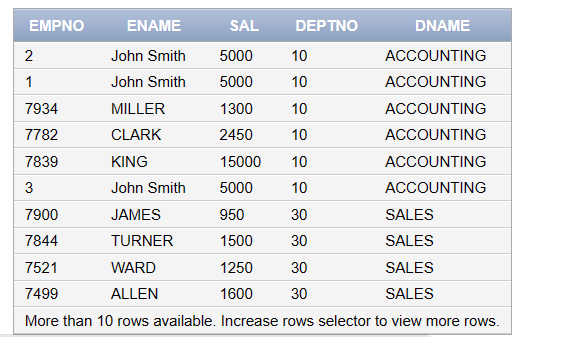
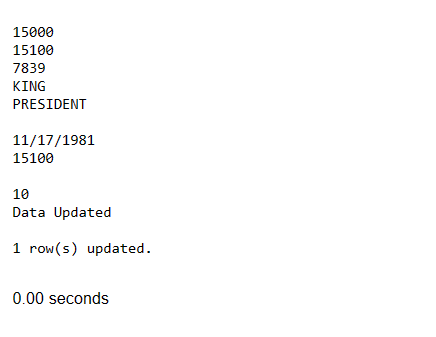
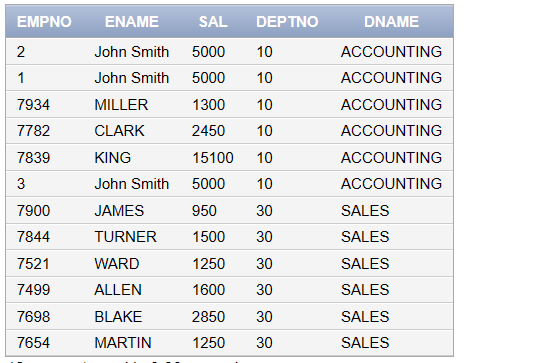
    UPDATE emp

    SET sal= :new.sal

    where empno=:old.empno;

      dbms\_output.put\_line('Data Updated');

END;

BEFORE  
  
  
AFTER  
  
  
Q5

DECLARE

CURSOR c\_emp IS

SELECT \* FROM emp ;

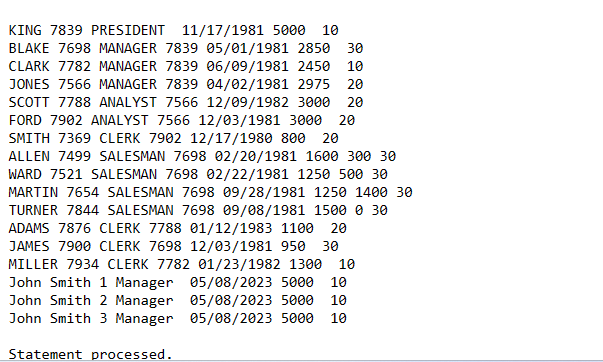
BEGIN

FOR v\_c IN c\_emp LOOP

DBMS\_OUTPUT.PUT\_LINE(v\_c.ename||' '||v\_c.EMPNO||' '||v\_c.JOB||' '||v\_c.MGR||' '||v\_c.HIREDATE||' '||v\_c.SAL||' '||v\_c.COMM||' '||v\_c.DEPTNO);

END LOOP;

end;

OUTPUT  
  
  
Q6

DECLARE

CURSOR c\_emp IS

SELECT \* FROM emp where sal >(select AVG(SAL) FROM EMP) ;

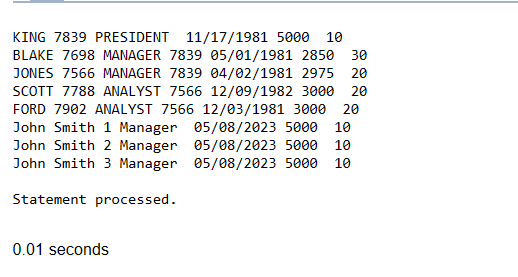
BEGIN

FOR v\_c IN c\_emp LOOP

DBMS\_OUTPUT.PUT\_LINE(v\_c.ename||' '||v\_c.EMPNO||' '||v\_c.JOB||' '||v\_c.MGR||' '||v\_c.HIREDATE||' '||v\_c.SAL||' '||v\_c.COMM||' '||v\_c.DEPTNO);

END LOOP;

end;

OUTPUT  


Q7

DECLARE

CURSOR c\_emp IS

SELECT \* FROM emp WHERE SAL > 3500 AND DEPTNO = 30 ;

BEGIN

FOR v\_c IN c\_emp LOOP

DBMS\_OUTPUT.PUT\_LINE(v\_c.ename);

DBMS\_OUTPUT.PUT\_LINE(v\_c.EMPNO);

DBMS\_OUTPUT.PUT\_LINE(v\_c.JOB);

DBMS\_OUTPUT.PUT\_LINE(v\_c.MGR);

DBMS\_OUTPUT.PUT\_LINE(v\_c.HIREDATE);

DBMS\_OUTPUT.PUT\_LINE(v\_c.SAL);

DBMS\_OUTPUT.PUT\_LINE(v\_c.COMM);

DBMS\_OUTPUT.PUT\_LINE(v\_c.DEPTNO);

END LOOP;

end;

OUTPUT  
