AIM:

To execute PL/SQL programs based on the given constraints

9. Create a Trigger which comes into action when updating is performed onto the Employee table. The new action triggered is: Insertion of a row into the Employee History table which has all information in Employee table in addition to the date and time when the salary was updated in the original Employee table.

10. Create a Trigger which comes into action when deletion is performed onto the Employee table. The new action triggered is: Insertion of a row into the Employee Resigned table which has all information in Employee table in addition to the date and time when the Employee was relieved from duty officially effecting the removal of his/her information from the Employee table.

11. Create a Trigger which comes into action when insertion is performed onto the Employee table. The new action triggered is: Computation and Display of age of the new employee after the addition of new employee information into the Employee table.

9)

```
create table Employee_details(e_id integer primary key,
e_name varchar2(20),salary Integer
);
insert into Employee_details values (1,'Bheem', 50000);
insert into Employee_details values(2,'Raju',55000);
Insert Into Employee_details values (3,'Chutki',60000);
insert into Employee_details values(4, 'Kichak',70000);

create table Employee_history(emp_his_id integer primary key,
e_id integer references Employee_details(e_id),
e_name varchar2(20),salary integer,
update_date timestamp);
```

```

create sequence employee_history_seq start with 100;
create or replace trigger salary_update
after update of salary on Employee_details
for each row
begin
insert into Employee_history values(employee_history_seq.nextval,:old.e_id,:old.e_name,:new.salary,systimestamp);
END;

update Employee_details set salary=95000 where e_id=1;
update Employee_details set salary=77000 where e_id=2;
select * from Employee_history;
```

Results | Explain | Describe | Saved SQL | History

EMP_HIS_ID	E_ID	E_NAME	SALARY	UPDATE_DATE
120	2	Raju	77000	05-MAY-24 01.05.46.587979 PM
100	1	Bheem	95000	05-MAY-24 01.05.34.714621 PM

2 rows returned in 0.01 seconds Download

10)

```
create table Employee_join(
    emp_id integer primary key,
    emp_name varchar2(20),
    dept varchar2(20),
    joindate timestamp
);
 insert into Employee_join values(1,'John','Hr',timestamp '2022-01-19 05:26:14');
 insert into Employee_join values(2,'Joe','it',timestamp '2020-05-29 10:36:24');

```

```
create table Employee_resignation(
    emp_id integer primary key,
    emp_name varchar2(20),
    dept varchar2(20),
    joindate timestamp,
    resignationdate timestamp
);

create or replace trigger After_Delete_Employee
after delete on  Employee_join
FOR EACH ROW
BEGIN
    INSERT INTO Employee_resignation (emp_id,emp_name,dept,joindate, resignationdate)
    VALUES (:OLD.emp_id ,:OLD.emp_name,:OLD.dept, :OLD.joindate,systimestamp);
END;
```

```
select * from Employee_join;
delete from Employee_join where emp_id=1;
select * from Employee_resignation;
```

Results | Explain | Describe | Saved SQL | History

EMP_ID	EMP_NAME	DEPT	JOINDATE	RESIGNATIONDATE
1	John	Hr	19-JAN-22 05.26.14.000000 AM	05-MAY-24 08.27.11.533055 AM

1 rows returned in 0.01 seconds Download

11)

```
create table Empdetails(
    emp_id integer primary key,
    emp_name varchar2(20),
    dob date,
    salary integer
);
insert into Empdetails values(1,'Ram',date '2023-09-29',40000);
insert into Empdetails values(2,'Vasanth',date '2022-05-31',55000);
insert into Empdetails values(3,'Hari',date'2021-04-27',95000);
```

```
CREATE OR REPLACE TRIGGER emp_age_trigger
AFTER INSERT ON Empdetails
FOR EACH ROW
DECLARE
    age NUMBER;
BEGIN
    age := TRUNC(MONTHS_BETWEEN(sysdate, :NEW.dob) / 12);
    DBMS_OUTPUT.PUT_LINE('Employee Age: ' || age);
END;

insert into Empdetails(emp_id,emp_name,dob,salary) values(4,'Vicky',date'2020-05-17',97000);
```