```
/*
https://docs.oracle.com/cd/A57673_01/DOC/server/doc/SCN73/ch15.htm
https://docs.oracle.com/cd/B28359_01/appdev.111/b28370/
triggers.htm#LNPLS2005
*/


***

CREATE or REPLACE TRIGGER Print_salary_changes
BEFORE UPDATE OF Salary ON employee
FOR EACH ROW

DECLARE

BEGIN
    dbms_output.put_line('salary change');
END;



create or replace TRIGGER Print_salary_changes
BEFORE UPDATE OF Salary ON employee
FOR EACH ROW

DECLARE
    sal_diff number;
BEGIN
    sal_diff  := :NEW.SALARY  - :OLD.SALARY;
    dbms_output.put('Old salary: ' || :OLD.salary);
    dbms_output.put('  New salary: ' || :NEW.salary);
    dbms_output.put_line('  Difference ' || sal_diff);
END;


update employee set salary = salary * 1.1 where dno = 6;


CREATE or REPLACE TABLE Emp_log (
  Emp_id     CHAR(9),
  New_salary NUMBER,
  Old_salary NUMBER,
  Log_date   DATE
);


create or replace TRIGGER Log_salary_changes
  AFTER UPDATE OF salary ON employee
  FOR EACH ROW
BEGIN
```

```
  INSERT INTO Emp_log (Emp_id, New_salary, Old_salary, Log_Date)
  VALUES (:new.ssn, :new.salary, :old.salary, SYSDATE);
END;



/* Update Total_salary on Department table after INSERT, DELETE or
UPDATE (of salary or deptno) on Employee table.
Attribute names are not exactly same with those we use in Company
database, but they are close enough that you should be able to follow.
*/


CREATE TRIGGER total_salary
AFTER DELETE OR INSERT OR UPDATE OF deptno, sal ON emp
FOR EACH ROW
BEGIN
  /* assume that DEPTNO and SAL are non-null fields */
  IF DELETING OR (UPDATING AND :old.deptno != :new.deptno) THEN
      UPDATE dept SET total_sal = total_sal - :old.sal WHERE deptno
= :old.deptno;
  END IF;

  IF INSERTING OR (UPDATING AND :old.deptno != :new.deptno) THEN
      UPDATE dept SET total_sal = total_sal + :new.sal WHERE deptno
= :new.deptno;
  END IF;
  IF (UPDATING AND :old.deptno = :new.deptno AND :old.sal !
= :new.sal ) THEN
      UPDATE dept SET total_sal = total_sal - :old.sal + :new.sal
WHERE deptno = :new.deptno;
  END IF;

END;


/*
:old is not available for INSERT
:new is not available for DELETE */


/* Trigger that implements following business rule:
An employee's salary increase must not exceed 10% of the average
salary for the employee's department.
*/


/* In this solution, we assume a new column (Avg_Salary) has been
added to Department table. */
```

```
create or replace TRIGGER Check_salary_changes
BEFORE UPDATE OF Salary ON employee
FOR EACH ROW

DECLARE
    sal_diff number;
    dept_avg number;
BEGIN
    sal_diff  := :NEW.SALARY  - :OLD.SALARY;
    select avg_salary into dept_avg from department where
dno=:NEW.dno;
    if sal_diff > dept_avg * 0.1 then
      Raise_Application_Error(-20000, 'Raise too big');
    end if;
END;




/* COMPOUND TRIGGERS */


/* Same business rule implemented by a compound trigger */

create or replace TRIGGER Check_Employee_Salary_Raise
FOR UPDATE OF Salary ON Employee
COMPOUND TRIGGER
  Ten_Percent                 CONSTANT NUMBER := 0.1;
  TYPE Salaries_t             IS TABLE OF Employee.Salary%TYPE;
  Avg_Salaries                Salaries_t;
  TYPE Department_IDs_t       IS TABLE OF Employee.DNo%TYPE;
  Department_IDs              Department_IDs_t;

  TYPE Department_Salaries_t  IS TABLE OF Employee.Salary%TYPE
                                INDEX BY VARCHAR2(80);
  Department_Avg_Salaries     Department_Salaries_t;

  BEFORE STATEMENT IS
  BEGIN
    SELECT               AVG(e.Salary), NVL(e.Dno, -1)
      BULK COLLECT INTO  Avg_Salaries, Department_IDs
      FROM               Employee e
      GROUP BY           e.Dno;
    FOR j IN 1..Department_IDs.COUNT() LOOP
      Department_Avg_Salaries(Department_IDs(j)) := Avg_Salaries(j);
    END LOOP;
  END BEFORE STATEMENT;

  AFTER EACH ROW IS
```

```
  BEGIN
    IF :NEW.Salary - :Old.Salary >
      Ten_Percent*Department_Avg_Salaries(:NEW.Dno)
    THEN
      Raise_Application_Error(-20000, 'Raise too big');
    END IF;
  END AFTER EACH ROW;
END Check_Employee_Salary_Raise;
```