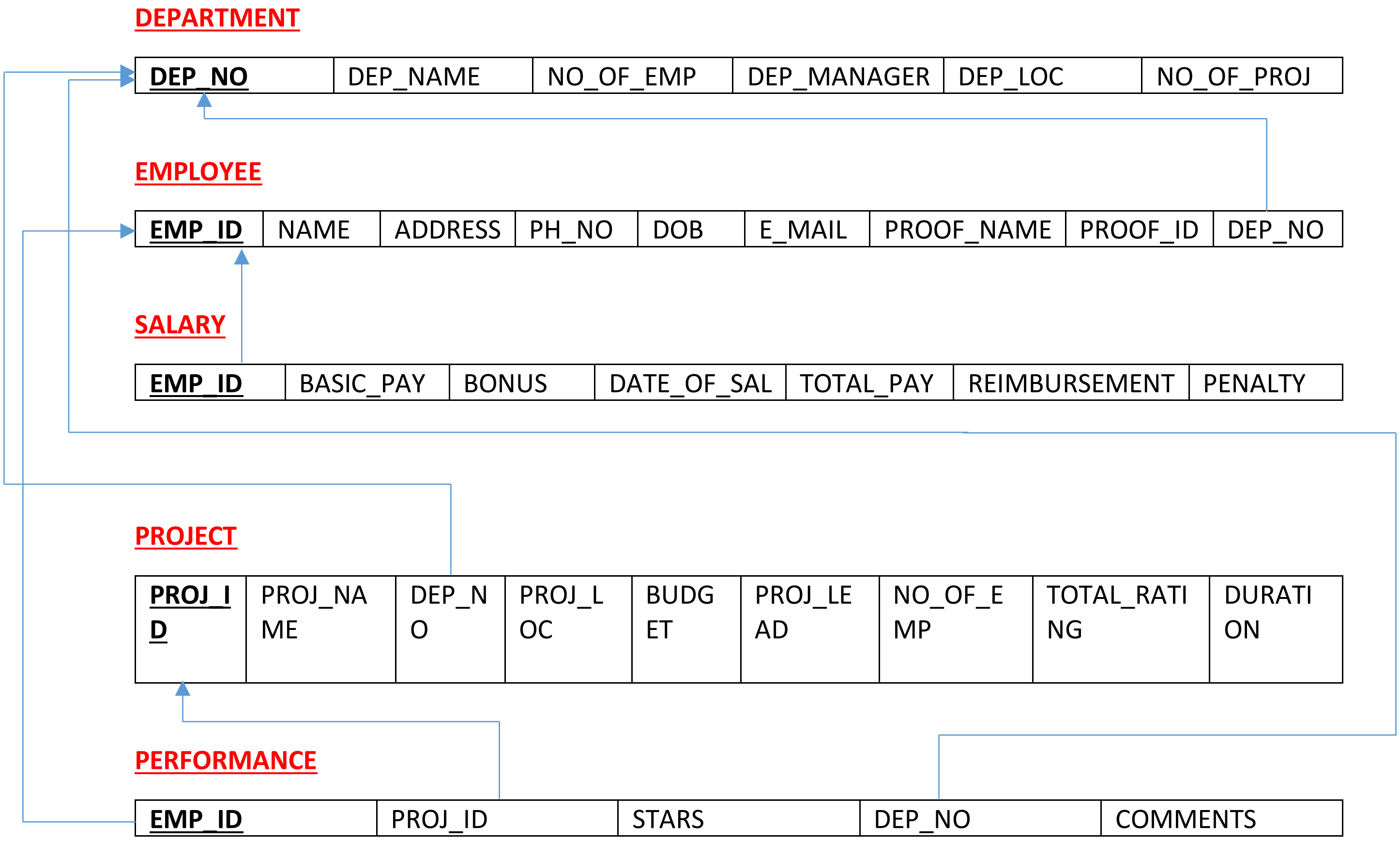

DEPARTMENT

DEP_NO	DEP_NAME	NO_OF_EMP	DEP_MANAGER	DEP_LOC	NO_OF_PROJ

EMPLOYEE

EMP_ID	NAME	ADDRESS	PH_NO	DOB	E_MAIL	PROOF_NAME	PROOF_ID	DEP_NO

SALARY

EMP_ID	BASIC_PAY	BONUS	DATE_OF_SAL	TOTAL_PAY	REIMBURSEMENT	PENALTY

PROJECT

PROJ_ID	PROJ_NAME	DEP_NO	PROJ_LOC	BUDGET	PROJ_LEAD	NO_OF_EMP	TOTAL_RATING	DURATION

PERFORMANCE

EMP_ID	PROJ_ID	STARS	DEP_NO	COMMENTS

```
CREATE TABLE DEPARTMENT
(
        DEP_NO INTEGER,
        DEP_NAME VARCHAR(20),
        NO_OF_EMP INTEGER,
        DEP_MANAGER VARCHAR(20),
        DEP_LOC VARCHAR(20),
        NO_OF_PROJECT INTEGER,
        PRIMARY KEY(DEPT_NO)
);

CREATE TABLE EMPLOYEE
 (
        EMP_ID INTEGER,
        NAME VARCHAR(20),
        ADDRESS VARCHAR(40),
        PH_NO INTEGER,
        DOB VARCHAR(11),
        E_MAIL VARCHAR(25),
        PROOF_NAME VARCHAR(10),
        PROOF_ID VARCHAR(20),
        DEP_NO INTEGER,
        PRIMARY KEY(EMP_ID),
        FOREIGN KEY(DEP_NO) REFERENCES DEPARTMENT(DEP_NO) ON
        DELETE CASCADE
);

CREATE TABLE SALARY
(
        EMP_ID INTEGER,
        BASIC_PAY INTEGER,
        BONUS INTEGER,
        DATE_OF_SAL VARCHAR(11),
        TOTAL_PAY INTEGER,
        REIMBURSEMENT INTEGER,
        PENALTY INTEGER,
        PRIMARY KEY(EMP_ID),
        FOREIGN KEY(EMP_ID) REFERENCES EMPLOYEE(EMP_ID) ON
        DELETE CASCADE
);
```

```
CREATE TABLE PROJECT
(
        PROJ_ID INTEGER,
        PROJ_NAME VARCHAR(20),
        DEP_NO INTEGER,
        PROJ_LOC VARCHAR(50),
        BUDGET INTEGER,
        PROJECT_LEAD VARCHAR(20),
        NO_OF_EMP INTEGER,
        TOTAL_RATING INTEGER(1),
        DURATION VARCHAR(10),
        PRIMARY KEY(PROJ_ID),
        FOREIGN KEY(DEP_NO) REFERENCES DEPARTMENT(DEP_NO) ON
        DELETE CASCADE
);

CREATE TABLE PERFORMANCE
(
        EMP_ID INTEGER,
        PROJ_ID INTEGER,
        STARS INTEGER,
        DEP_NO INTEGER,
        COMMENTS VARCHAR(50),
        PRIMARY KEY(EMP_ID),
        FOREIGN KEY(PROJ_ID) REFERENCES PROJECT(PROJ_ID) ON
        DELETE CASCADE,
        FOREIGN KEY(DEP_NO) REFERENCES DEPARTMENT(DEP_NO) ON
        DELETE CASCADE
);
```

```
CREATE TABLE CALC
(
        EMP_ID INTEGER,
        OLD_SAL INTEGER DEFAULT NULL,
        NEW_SAL INTEGER,
        HIKE INTEGER DEFAULT NULL,
        PRIMARY KEY(EMP_ID),
        FOREIGN KEY (EMP_ID) REFERENCES EMPLOYEE (EMP_ID);
);
```

TRIGGERS:-

1)

```
DELIMITER $$

CREATE TRIGGER `before_salary_insert`
BEFORE INSERT ON `salary`
FOR EACH ROW

BEGIN
   INSERT INTO calc
   SET
        emp_id  = new.emp_id,
        new_sal = new.total_pay;
END

$$

DELIMITER ;
```

2)

```
DELIMITER $$

CREATE TRIGGER `after_salary_edit`
AFTER UPDATE ON `salary`
FOR EACH ROW

BEGIN
        UPDATE calc
        SET
        old_sal = old.total_pay,
        new_sal = new.total_pay
        WHERE
        emp_id = old.emp_id;
END

$$

DELIMITER ;
```

```
3)  DELIMITER $$

    CREATE TRIGGER `after_salary_delete`
    AFTER DELETE ON `salary`
    FOR EACH ROW

    BEGIN
            DELETE from calc
            WHERE emp_id = old.emp_id;
    END

    $$

    DELIMITER ;
```

PROCEDURE:-

```
DELIMITER $$

CREATE PROCEDURE CalcHike()

BEGIN

    DECLARE c_emp_id INTEGER;
    DECLARE c_stars INTEGER;
    DECLARE c_hike_per VARCHAR(4);
    DECLARE finished INTEGER DEFAULT 0;

    DECLARE
    curEmp CURSOR FOR
    SELECT emp_id , stars FROM performance;

    DECLARE CONTINUE HANDLER
   FOR NOT FOUND SET finished = 1;


    OPEN curEmp;

    hikeLoop:LOOP

    FETCH curEmp INTO c_emp_id, c_stars;

    IF finished = 1 THEN
    LEAVE hikeLoop;
    END IF;

    IF( c_stars <= 4 )
        THEN SET c_hike_per = "4%";

    ELSEIF ( c_stars >= 5 AND  c_stars <= 7 )
        THEN SET c_hike_per = "10%";
    ELSEIF ( c_stars >= 8 AND  c_stars <= 9 )
        THEN SET c_hike_per = "14%";

    ELSE
        SET c_hike_per = "20%";

    END IF;

    UPDATE calc SET hike = c_hike_per WHERE emp_id = c_emp_id;

    END LOOP;

    CLOSE curEmp;

END$$

DELIMITER ;
```