1. In an organization number of departments exist. Each department has a name & unique code. Number of employees work in each department. Each employee has unique employee code. Detailed information like name, address, city, basic, date of join are also stored. In a leave register for each employee leave records are kept showing leave type (CL/EL/ML etc.),from date , to-date. When an employee retires or resigns then all the leave information pertaining to him are also deleted. Basic salary must be within Rs.5000 to Rs.9000. A department can not be deleted if any employee record refers to it. Valid grades are A/B/C. Employee name must be in uppercase only. Default value for joining date is system date. Design & implement the tables with necessary constraints to support the scenario depicted above.

Answer)

EMP table is created like the following:

Create table DEPARTMENT(DEPT\_CODE CHAR(5) PRIMARY KEY, DEPT\_NAME CHAR(15));

Later, you will find, this wont work.

So, the ultimate solution was

**Create table DEPARTMENT(DEPT\_CODE CHAR(5) PRIMARY KEY, DEPT\_NAME CHAR(15)) ENGINE=InnoDB;**

**create table EMP(EMP\_CODE CHAR(5) NOT NULL PRIMARY KEY, EMP\_NAME CHAR(20), DEPT\_CODE CHAR(5), DESIG\_CODE CHAR(5), SEX CHAR(1), ADDRESS CHAR(25), CITY CHAR(20), STATE CHAR(20), PIN CHAR(6), BASIC DECIMAL(10,2), JN\_DATE DATE, FOREIGN KEY(DEPT\_CODE) REFERENCES DEPARTMENT(DEPT\_CODE) ON DELETE RESTRICT);**

Again, the ultimate solution was (why, you will find it later in this document)

**create table EMP(EMP\_CODE CHAR(5) NOT NULL PRIMARY KEY, EMP\_NAME CHAR(20), DEPT\_CODE CHAR(5), DESIG\_CODE CHAR(5), SEX CHAR(1), ADDRESS CHAR(25), CITY CHAR(20), STATE CHAR(20), PIN CHAR(6), BASIC DECIMAL(10,2), JN\_DATE DATE, FOREIGN KEY(DEPT\_CODE) REFERENCES DEPARTMENT(DEPT\_CODE) ON DELETE RESTRICT)ENGINE=InnoDB;**

Now, why these two tables are creates like that?  
  
I.e. why Department is created as a Parent Table, why EMP is created as a Child table (in the case of using foreign key DEPT\_CODE in EMP) and why is “DELETE RESTRICT” used?

**Because,**

A department can not be deleted if any employee record refers to it.

**Now, there is another table constraint: Default value for joining date is system date.**

Now, there’s a function named SYSDATE() in mysql. But, sysdate() returns the current date and time. Now, we could alternatively use CURDATE() as it returns the current date in 'YYYY-MM-DD' format.

But, there’s a big thing about it. Older versions of Mysql does not allow to set default value for mysql. WE can not use SYSDATE() or CURRDATE(). You can not even use NOW() or CURRENT\_TIMESTAMP. A good option is to write a trigger. Another approach is to change the datatype of JN\_DATE as TIMESTAMP instead of DATETIME.

**CREATE TRIGGER insertDefaultDateTime BEFORE INSERT ON `EMP`  
FOR EACH ROW SET NEW.JN\_DATE = IFNULL(NEW.JN\_DATE, CURRDATE());**

However, this trigger is giving error during insertion in EMP. So, the trigger is dropped using:

**DROP TRIGGER insertDefaultDateTime;**

And, another trigger is created:

**CREATE TRIGGER insertDefaultDateTime BEFORE INSERT ON `EMP`  
FOR EACH ROW SET NEW.JN\_DATE = IFNULL(NEW.JN\_DATE, NOW());**

Now, this trigger allows you to provide a value for your date column and it will take, and otherwise it'll default to the current time.

However, there’s a limitation on basic too.

Now, cannot directly write a validation checking, so another trigger is written:

**delimiter $$**

**CREATE TRIGGER BASIC\_LIMITATION BEFORE INSERT ON EMP FOR EACH ROW BEGIN IF (NEW.BASIC<600000) THEN SET NEW.BASIC=600000; ELSEIF (NEW.BASIC>3000000) THEN SET NEW.BASIC=3000000; END IF; END$$**

But this will give an error as until mysql 5.7, mysql does not support 'multiple triggers with the same action time and event for one table'.

**Final Solution:**

**So, a combination is written:**

**delimiter $$  
CREATE TRIGGER TRIGGER\_ON\_EMP BEFORE INSERT ON EMP FOR EACH ROW BEGIN SET NEW.JN\_DATE = IFNULL(NEW.JN\_DATE, NOW()); IF(NEW.BASIC<600000) THEN SET NEW.BASIC=600000; ELSEIF(NEW.BASIC>3000000) THEN SET NEW.BASIC=3000000; END IF; END$$**

Also, a leave register is to be maintained.

**create table LEAVE\_REGISTER(EMP\_CODE CHAR(5) NOT NULL, LEAVE\_TYPE CHAR(2) NOT NULL, FROM\_DATE DATE NOT NULL, TO\_DATE DATE NOT NULL, FOREIGN KEY(EMP\_CODE) REFERENCES EMP(EMP\_CODE) ON DELETE CASCADE);**

Now, we need to check it.

Now, the following departments are inserted.

**insert into DEPARTMENT(DEPT\_CODE, DEPT\_NAME) VALUES("00001","Research");**

**insert into DEPARTMENT(DEPT\_CODE, DEPT\_NAME) VALUES("00002","Finance");**

**insert into DEPARTMENT(DEPT\_CODE, DEPT\_NAME) VALUES("00003","Purchase");**

**insert into DEPARTMENT(DEPT\_CODE, DEPT\_NAME) VALUES("00004","Production");**

**insert into DEPARTMENT(DEPT\_CODE, DEPT\_NAME) VALUES("00005","Personnel");**

**Or,**

**insert into DEPARTMENT VALUES("00001","Research"),("00002","Finance"),("00003","Purchase"),("00004","Production"),("00005","Personnel");**

And, following employees are inserted:

**insert into EMP(EMP\_CODE, EMP\_NAME, DEPT\_CODE, DESIG\_CODE, SEX, ADDRESS, CITY, STATE, PIN, BASIC, JN\_DATE) VALUES("00001", "Sayak Haldar", "00001", "00001", 'M', "C.R. Park, Delhi", "New Delhi", "New Delhi", "110019", 3000000.00, '2012-04-01');**

**insert into EMP(EMP\_CODE, EMP\_NAME, DEPT\_CODE, DESIG\_CODE, SEX, ADDRESS, CITY, STATE, PIN, BASIC, JN\_DATE) VALUES("00002", "Sayantan Acharya", "00001", "00002", 'M', "C.R. Park, Delhi", "New Delhi", "New Delhi", "110019", 2400000, '2013-04-01');**

**insert into EMP(EMP\_CODE, EMP\_NAME, DEPT\_CODE, DESIG\_CODE, SEX, ADDRESS, CITY, STATE, PIN, BASIC, JN\_DATE) VALUES("00003", "Suman Rudra", "00001", "00003", 'M', "C.R. Park, Delhi", "New Delhi", "New Delhi", "110019", 2100000, '2013-07-01');**

**insert into EMP(EMP\_CODE, EMP\_NAME, DEPT\_CODE, DESIG\_CODE, SEX, ADDRESS, CITY, STATE, PIN, BASIC, JN\_DATE) VALUES("00004", "Buddha Dutta", "00002", "00001", 'M', "Noida,Delhi", "New Delhi", "New Delhi", "201301", 2600000, '2013-07-01');**

**insert into EMP(EMP\_CODE, EMP\_NAME, DEPT\_CODE, DESIG\_CODE, SEX, ADDRESS, CITY, STATE, PIN, BASIC, JN\_DATE) VALUES("00005", "Neha Sen", "00002", "00002", 'F', "Noida,Delhi", "New Delhi", "New Delhi", "201301", 1900000, '2014-01-01');**

**insert into EMP(EMP\_CODE, EMP\_NAME, DEPT\_CODE, DESIG\_CODE, SEX, ADDRESS, CITY, STATE, PIN, BASIC, JN\_DATE) VALUES("00006", "Tilottoma Majumdar", "00002", "00003", 'F', "Nodia,Delhi", "New Delhi", "New Delhi", "201301", 1600000, '2014-01-01');**

**insert into EMP(EMP\_CODE, EMP\_NAME, DEPT\_CODE, DESIG\_CODE, SEX, ADDRESS, CITY, STATE, PIN, BASIC, JN\_DATE) VALUES("00008","Debarati Das", "00003", "00001", 'F', "Kalkaji,Delhi", "New Delhi", "New Delhi", "110019", 1600000, '2014-01-01');**

**insert into EMP(EMP\_CODE, EMP\_NAME, DEPT\_CODE, DESIG\_CODE, SEX, ADDRESS, CITY, STATE, PIN, BASIC, JN\_DATE) VALUES("00009","Debanjan Das", "00003", "00002", 'M', "Okhla", "New Delhi", "New Delhi", "110020", 1300000, '2014-04-01');**

**insert into EMP(EMP\_CODE, EMP\_NAME, DEPT\_CODE, DESIG\_CODE, SEX, ADDRESS, CITY, STATE, PIN, BASIC, JN\_DATE) VALUES("00010","Binayak Basu", "00003", "00003", 'M', "Okhla", "New Delhi", "New Delhi", "110020", 1200000, '2014-05-01');**

**insert into EMP(EMP\_CODE, EMP\_NAME, DEPT\_CODE, DESIG\_CODE, SEX, ADDRESS, CITY, STATE, PIN, BASIC, JN\_DATE) VALUES ("00011","Supratim Majumdar", "00003","00001",'M', "Malviya Nagar", "New Delhi", "New Delhi", "110017", 1300000, '2014-01-01');**

**insert into EMP(EMP\_CODE, EMP\_NAME, DEPT\_CODE, DESIG\_CODE, SEX, ADDRESS, CITY, STATE, PIN, BASIC, JN\_DATE) VALUES("00012","Subas Das", "00004", "00003", 'M', "Okhla", "New Delhi", "New Delhi", "110020", 1000000, '2014-05-01');**

**insert into EMP(EMP\_CODE, EMP\_NAME, DEPT\_CODE, DESIG\_CODE, SEX, ADDRESS, CITY, STATE, PIN, BASIC, JN\_DATE) VALUES("00013","Sananda Roy", "00005", "00003", 'F', "Okhla", "New Delhi", "New Delhi", "110020", 900000, '2014-08-01');**

**insert into EMP(EMP\_CODE, EMP\_NAME, DEPT\_CODE, DESIG\_CODE, SEX, ADDRESS, CITY, STATE, PIN, BASIC, JN\_DATE) VALUES("00014","Prerona Hazra", "00005", "00004", 'F', "Okhla", "New Delhi", "New Delhi", "110020", 700000, '2014-08-01');**

**insert into EMP(EMP\_CODE, EMP\_NAME, DEPT\_CODE, DESIG\_CODE, SEX, ADDRESS, CITY, STATE, PIN,BASIC,JN\_DATE) VALUES("00015","Ankan Debnath", "00005", "00004", 'M', "Okhla", "New Delhi", "New Delhi", "110020", 700000.00,'2014-08-01');**

**insert into EMP(EMP\_CODE, EMP\_NAME, DEPT\_CODE, DESIG\_CODE, SEX, ADDRESS, CITY, STATE, PIN,BASIC,JN\_DATE) VALUES("00016","Ankush Das", "00005", "00004", 'M', "Okhla", "New Delhi", "New Delhi", "110020",700000.00, '2014-08-01');**

**2.Try to violate the constraints that you have implemented in the table & note , what happens. [Try with suitable INSERT/UPDATE/DELETE instruction]**

First to check if salary is less than 600000 (i.e. we are checking the salary constraint)

insert into EMP(EMP\_CODE, EMP\_NAME, DEPT\_CODE, DESIG\_CODE, SEX, ADDRESS, CITY, STATE, PIN,BASIC,JN\_DATE) VALUES("00018","Mahabir Pathak", "00005", "00004", 'M', "Okhla", "New Delhi", "New Delhi", "110020",500000.00, '2014-08-01');

A sample data is inserted like this.

Now, Mahabir Pathak’s salary is automatically changed to 600000.00.

Now, next constraint is a department can not be deleted if any employee record refers to it.

**DELETE FROM DEPARTMENT WHERE DEPT\_CODE="00005";**

**Now, 4/5 records in EMP refers to DEPT\_CODE yet it is deleted.**

**So, clearly ON DELETE RESTRICT does not work.**

**So, we need a different solution.**

**Now, the problem is I arranged it in a wrong manner.**

**create table EMP(EMP\_CODE CHAR(5) NOT NULL PRIMARY KEY, EMP\_NAME CHAR(20), DEPT\_CODE CHAR(5), DESIG\_CODE CHAR(5), SEX CHAR(1), ADDRESS CHAR(25), CITY CHAR(20), STATE CHAR(20), PIN CHAR(6), BASIC DECIMAL(10,2), JN\_DATE DATE);**

**And,**

**create table DEPARTMENT(DEPT\_CODE CHAR(5), DEPT\_NAME CHAR(15),FOREIGN KEY(DEPT\_CODE) REFERENCES EMP(DEPT\_CODE) ON DELETE RESTRICT);**

And, leave\_register is not changed:

**create table LEAVE\_REGISTER(EMP\_CODE CHAR(5) NOT NULL, LEAVE\_TYPE CHAR(2) NOT NULL, FROM\_DATE DATE NOT NULL, TO\_DATE DATE NOT NULL, FOREIGN KEY(EMP\_CODE) REFERENCES EMP(EMP\_CODE) ON DELETE CASCADE);**

Now, actually my previous logic was right. So, this does not obviously work.

**This is problem of Engine.**

**Create table DEPARTMENT(DEPT\_CODE CHAR(5) PRIMARY KEY, DEPT\_NAME CHAR(15))ENGINE=InnoDB;**

**create table EMP(EMP\_CODE CHAR(5) NOT NULL PRIMARY KEY, EMP\_NAME CHAR(20), DEPT\_CODE CHAR(5), DESIG\_CODE CHAR(5), SEX CHAR(1), ADDRESS CHAR(25), CITY CHAR(20), STATE CHAR(20), PIN CHAR(6), BASIC DECIMAL(10,2), JN\_DATE DATE, FOREIGN KEY(DEPT\_CODE) REFERENCES DEPARTMENT(DEPT\_CODE) ON DELETE RESTRICT)ENGINE=InnoDB;**

**create table LEAVE\_REGISTER(EMP\_CODE CHAR(5) NOT NULL, LEAVE\_TYPE CHAR(2) NOT NULL, FROM\_DATE DATE NOT NULL, TO\_DATE DATE NOT NULL, FOREIGN KEY(EMP\_CODE) REFERENCES EMP(EMP\_CODE) ON DELETE CASCADE)Engine=InnoDB;**

Now, try to delete a department from DEPARTMENT like the following:

**delete from DEPARTMENT WHERE DEPT\_CODE="00005";**

It will give you the error:

**Cannot delete or update a parent row: a foreign key constraint fails (`ProblemSet3`.`EMP`, CONSTRAINT `EMP\_ibfk\_1` FOREIGN KEY (`DEPT\_CODE`) REFERENCES `DEPARTMENT` (`DEPT\_CODE`))**

Another testing is to be done on the constraints applied in leave\_register.

When an employee retires or resigns then all the leave information pertaining to him are also deleted.

Now, that’s why ON DELETE CASCADE is applied on LEAVE\_REGISTER whose foreign key is EMP\_CODE;

Now, I insert a leave information:

**INSERT INTO LEAVE\_REGISTER(EMP\_CODE,LEAVE\_TYPE,FROM\_DATE, TO\_DATE)VALUES("00011","CL",'2017-07-03','2017-07-09');**

And,

After that, I delete the employee record corresponding to EMP\_CODE=00011.

**delete from EMP WHERE EMP\_CODE="00011";**

Now, I check the LEAVE\_REGISTER again.

The records are deleted.

**Now, another thing is to be checked.**

**That if no join date is specified, join date is included as current\_date or not.**

So, a row is inserted:

**insert into EMP(EMP\_CODE,EMP\_NAME,DEPT\_CODE, DESIG\_CODE, SEX, ADDRESS, CITY, STATE, PIN, BASIC) VALUES ("00017", "Sayan Hazra", "00003", "00004", 'M', "Okhla", "New Delhi", "New Delhi", 110020, 1000000.00);**

I.e. without mentioning JN\_DATE.

2017-06-04 is inserted as JN\_DATE.

So, the constraints are ok.

3. a) create a view showing employee code, Name,dcode & Basic for a particular department.

b)Try to insert a row into the view with valid department & also with invalid ones.

c)Find the newly inserted row in the table From which view was created.

d) Try to increment basic by Rs.100/-) Check it in the original table.

f) Delete the view.

Answer)

a) **CREATE VIEW ProblemSet3.RESEARCH\_DEPT\_DETAILS AS SELECT EMP\_CODE, EMP\_NAME, DEPT\_CODE, BASIC FROM EMP WHERE DEPT\_CODE="00001";**

This will create the view.

b)

**insert into RESEARCH\_DEPT\_DETAILS VALUES("00004","Sukanta Jathi", 00001, 1600000);**

Initially tried this.

This generates the error Duplicate entry '00004' for key 'PRIMARY'.

Now, this is generated as there’s an entry with ‘00004’ EMP\_CODE in EMP.

So, tried this.

**insert into RESEARCH\_DEPT\_DETAILS VALUES("00018","Sukanta Jathi", “00001”, 1600000);**

This is inserted.

Also, a corresponding record is created in EMP.

Now, this is a valid department, but invalid in case of view.

**insert into RESEARCH\_DEPT\_DETAILS VALUES("00019","Sayani Paul", "00002", 1300000);**

However, it is inserted.

And, again, a corresponding record is created in EMP.

However, if a record is inserted with wrong/ invalid Department code, (a department code which does not exist in DEPARTMENT) , it will give error.

**Cannot add or update a child row: a foreign key constraint fails (`ProblemSet3`.`EMP`, CONSTRAINT `EMP\_ibfk\_1` FOREIGN KEY (`DEPT\_CODE`) REFERENCES `DEPARTMENT` (`DEPT\_CODE`)).**

Now,I will do a personal testing.

Suppose, an employee is inserted in EMP table with DEPT\_CODE=”00001” will it be reflected in view?

**insert into EMP VALUES("00020","Sabarnya Chowdhury", "00001", "00005", 'M', "Kalkaji, Delhi", "New Delhi", "New Delhi" , "110019", 1100000, NOW());**

This is inserted in **EMP table**.

The view table RESEARCH\_DEPT\_DETAILS is already updated.

c)Find the newly inserted row in the table From which view was created.

Don’t get It.

d) Try to increment basic by Rs.100/-

e) Check it in the original table.

Now, an user is there named sabarnya Chowdhury in the view. Let’s try to increment his basic by 100 and see the original table.

So, performed the following:

**UPDATE RESEARCH\_DEPT\_DETAILS SET BASIC="1100100.00" WHERE EMP\_CODE="00020";**

And It is reflected too in main table EMP as well as RESEARCH\_DEPT\_DETAILS.

f) Delete the view.

**drop view RESEARCH\_DEPT\_DETAILS;**

5. a)Create a table having empcode , Name, Deptname, & basic From the existing tables along with the records of the employee who are in a particular department(say, d1) .

b) From the existing table add the employees with the basic salary greater than or equal to 2400000/-

c) Alter the table to add a net pay column.

d) Replace net pay with 1.5\* Basic.

e) Try to remove the net net pay column.

[ It may require no. of steps]

Answer) a)

create table EMP\_RESEARCH(EMP\_CODE CHAR(5) NOT NULL PRIMARY KEY, EMP\_NAME CHAR(20), DEPT\_NAME CHAR(15),BASIC DECIMAL(10,2));

First let’s create the table.

**Insert data needed:**

insert into EMP\_RESEARCH(EMP\_CODE,EMP\_NAME, DEPT\_NAME, BASIC) SELECT EMP.EMP\_CODE, EMP.EMP\_NAME, DEPARTMENT.DEPT\_NAME, EMP.BASIC FROM EMP, DEPARTMENT WHERE EMP.DEPT\_CODE=DEPARTMENT.DEPT\_CODE AND EMP.DEPT\_CODE="00001";

b) From the existing table add the employees with the basic salary greater than or equal to 2400000/ -

Now, I won’t perform it. It’s too easy.

c) Alter the table to add a net pay column.

**ALTER TABLE EMP\_RESEARCH ADD NET\_PAY DECIMAL(10,2);**

d) Replace net pay with 1.5\* Basic.

**UPDATE EMP\_RESEARCH SET NET\_PAY=1.5\*BASIC;**

e) Try to remove the net net pay column.

[ It may require no. of steps]

**ALTER TABLE EMP\_RESEARCH DROP NET\_PAY;**

1. **Drop all tables you have created.**

Drop table department;

Drop table EMP;

DROP table EMP\_RESEARCH;

DROP table LEAVE\_REGISTER;