Project title

Training & Placement Management System

Index

1. Problem statement
2. Database
3. ER Diagram
4. Source Code
   * 1. Triggers
     2. Stored Procedure
5. Questions the database can answer

**Problem Statement**

The training and placement institute is a place where students enroll for the training of the technologies of their choice and try to get hired by companies invited by the institute. Institute has multiple branches

in different cities. The institute owners/ investors want to know how the institute is performing. Also it has no record of students and the companies and hence no way to analyze anything about the institute. Hence there is a need of creating a database which will record, the students’ database who enroll for a training in institute, the courses, trainers, individual batches for given courses, financial aspects of the institute.

**Database**

1. This is a relational database created using mysql5.6 which defines the relationship between the different entities like student ,trainer , course, etc. The database is queried depending upon the tasks. It is designed to maintain the training institute, hence is used by different people. The database can answer many questions , some of which are covered in this document.
2. Assumed Roles:

* Training institute administrator
* Branch Manager
* Administrative staff

The Branch manager can add :

a new course

A new batch for the created course

Can create student

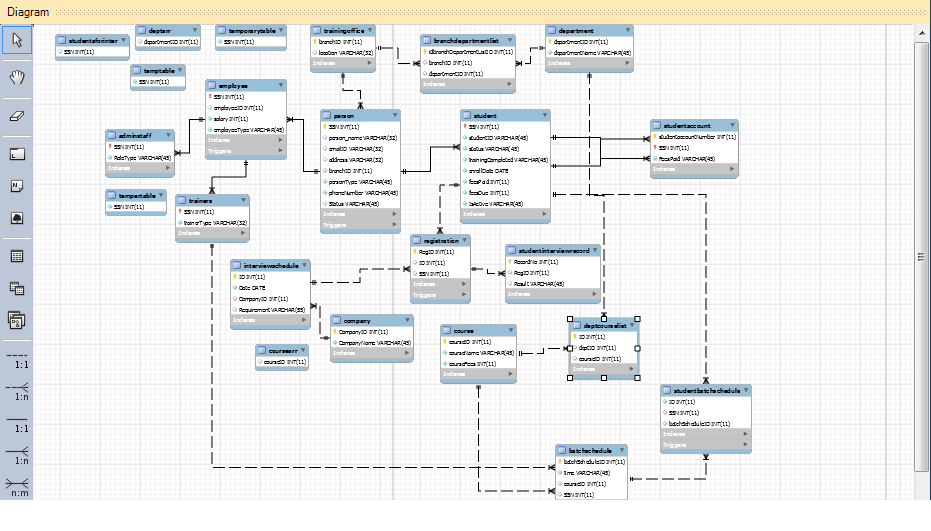
Can create department, etc.

3)Database stores the students’ record which includes

* Their fees due, fees paid
* Which companies have they enrolled.
* How was their performance/result of the given interview

4)Each branch has a certain number of departments and each department has its own list of courses and the employees as well.

**ER Diagram**

****

**Source Code**

Triggers

1.*On Person*

CREATE DEFINER=`root`@`localhost` TRIGGER `training\_and\_placement\_db`.`person\_AFTER\_INSERT` AFTER INSERT ON `person` FOR EACH ROW

BEGIN

set @ptype=new.persontype;

set @ssn=new.SSN;

if @ptype Like 'S' then

insert into student (SSN) values (@ssn);

elseif new.personType Like'E' then

insert into employee(SSN) values (@ssn);

end if ;

END

*2.On Registration table:*

CREATE DEFINER=`root`@`localhost` TRIGGER `training\_and\_placement\_db`.`registration\_AFTER\_INSERT` AFTER INSERT ON `registration` FOR EACH ROW

BEGIN

set @LastInsertedID =(select MAX(regID) from registration);

insert into studentinterviewrecord (RegID,Result)values(@LastInsertedID ,'painding');

END

*3.On student batchschedule:*

CREATE DEFINER=`root`@`localhost` TRIGGER `training\_and\_placement\_db`.`studentbatchschedule\_AFTER\_INSERT` AFTER INSERT ON `studentbatchschedule` FOR EACH ROW

BEGIN

set @newbatchSchID=new.batchScheduleID;

set @cID =(select courseID from batchSchedule where batchScheduleID=@newbatchSchID);

set @fees=(select courseFees from course where courseID=@cID);

set @currfees=(select feesDue from student where SSN=new.SSN);

update student set feesDue=(@currfees+@fees)

where SSN=new.SSN;

END

4.*On Employee table:*

*After insert:*

CREATE DEFINER=`root`@`localhost` TRIGGER `training\_and\_placement\_db`.`employee\_AFTER\_INSERT` AFTER INSERT ON `employee` FOR EACH ROW

BEGIN

set @etype=new.employeeType;

set @ssn=new.SSN;

if @etype Like 'T' then

insert into trainers (SSN) values (@ssn);

#elseif new.etype Like'A' then

#insert into employee(SSN) values (@ssn);

end if ;

END

5. *On Employee table:*

*After update:*

CREATE DEFINER=`root`@`localhost` TRIGGER `training\_and\_placement\_db`.`employee\_AFTER\_UPDATE` AFTER UPDATE ON `employee` FOR EACH ROW

BEGIN

set @etype=new.employeeType;

set @ssn=new.SSN;

if @etype Like 'T' then

insert into trainers (SSN) values (@ssn);

#elseif new.etype Like'A' then

#insert into employee(SSN) values (@ssn);

end if ;

END

Stored Procedure

*1.Given the branch name , this stored procedure lists all the courses taught there.*

CREATE DEFINER=`root`@`localhost` PROCEDURE `deptCourseList`(IN Loc varchar(32))

BEGIN

Declare bID int;

set bID:=(select branchID from trainingoffice where location Like Loc) ;

select bID;

CREATE table if not exists DeptArr as

(SELECT departmentID

FROM branchdepartmentlist

WHERE branchID=bID);

select \* from DeptArr;

create table if not exists CourseArr as

(select courseID from deptcourselist

where

deptID in

(select departmentID from DeptArr )

);

select courseName as Course\_Names from course where courseID in

(select courseID from CourseArr);

END

*2. Given the branch name , what is the total number of students, that are enrolled ,are unplaced*

CREATE DEFINER=`root`@`localhost` PROCEDURE `Find\_unplacedStudents`(IN Loc varchar(32) )

BEGIN

Declare bID int;

Declare countOfUnplacedStudent int;

set bID:=(select branchID from trainingoffice where location Like Loc) ;

CREATE table if not exists temporaryTable as

(SELECT SSN

FROM Person

WHERE personType Like 'S' and branchID=bID);

set countOfUnplacedStudent=(select count(\*) from

temporaryTable tt inner join student s

on

tt.SSN=s.SSN

where status Like '%Unplaced%');

SELECT 'The total numberof Unplaced students are',countOfUnplacedStudent ;

END

=>SET @ Location=’<name of the location>’;

=>CALL Find\_unplacedStudents(@Location);

3. *Given the company’s name and the date on which they had conducted interviews in the institute, This stored procedure gives details of all students who registered for that interview.*

CREATE DEFINER=`root`@`localhost` PROCEDURE `studentsForInterview`(IN compnyName varchar(32), interDate date)

BEGIN

declare compnyID int ;

declare interviewschID int ;

set compnyID:=(select companyID from company where companyName Like compnyName );

set interviewschID:=(select ID from interviewschedule where companyID=compnyID

and Date=interDate);

select interviewschID;

select SSN from registration where ID=interviewschID;

Create table if not exists stdForInter as

(SELECT SSN

FROM

registration

where ID=interviewschID);

select \* from stdForInter;

select person\_name,emailID,phoneNumber,Status

from

person inner join stdForInter

on

person.SSN=stdForInter.SSN;

END

***Questions the database can answer***

**1] How many students are currently enrolled in the institute ?**

SQL: select count(\*) from person where (person Type like 'S') and (Status like 'Active');

**2]How many of those student belong to pune branch?**

SQL:

SET @location=<Location name comes from variable>

select count(\*) from person where (personType like 'S') and (Status like 'Active') and branchID in

(select branchID from trainingoffice where location like @location);

**3] How much is the total fee that a particular student has to pay?**

SQL:

select sum(courseFees)from course

where

courseID in(

select courseID from batchschedule

where batchScheduleID in(

select batchScheduleID

from

student inner join studentbatchschedule

ON

student.SSN=studentbatchschedule.SSN

where studentID='st1'

)

)

**4] Given the branch name , what is the total number of students, that are enrolled ,are unplaced?**

SQL:

***Stored procedure:***

**CREATE** DEFINER=`root`@`localhost` PROCEDURE `Find\_unplacedStudents`(IN Loc varchar(32) )

BEGIN

Declare bID int;

Declare countOfUnplacedStudent int;

set bID:=(select branchID from trainingoffice where location Like Loc) ;

CREATE table if not exists temporaryTable as

(SELECT SSN

FROM Person

WHERE personType Like 'S' and branchID=bID);

set countOfUnplacedStudent=(select count(\*) from

temporaryTable tt inner join student s

on

tt.SSN=s.SSN

where status Like '%Unplaced%');

SELECT 'The total numberof Unplaced students are',countOfUnplacedStudent ;

**END**

=>SET @ Location=’<name of the location>’;

=>CALL Find\_unplacedStudents(@Location);

5] **How will the student be charged if he enrolls for a particular batchschedule ?**

**SQL:**

**Trigger after insert on student batch schedule=>**

CREATEDEFINER=`root`@`localhost`TRIGGER `training\_and\_placement\_db`.`studentbatchschedule\_AFTER\_INSERT` AFTER INSERT ON `studentbatchschedule` FOR EACH ROW

BEGIN

set @newbatchSchID=new.batchScheduleID;

set @cID =(select courseID from batchSchedule where batchScheduleID=@newbatchSchID);

set @fees=(select courseFees from course where courseID=@cID);

set @currfees=(select feesDue from student where SSN=new.SSN);

update student set feesDue=(@currfees+@fees)

where SSN=new.SSN;

END

**6] Given the company’s name and the date on which they had conducted interviews in the institute, Give details of all students who registered for that interview.**

**SQL=>**

**Procedure :**

**CREATE DEFINER**=`root`@`localhost` PROCEDURE `studentsForInterview`(IN compnyName varchar(32), interDate date)

BEGIN

declare compnyID int ;

declare interviewschID int ;

set compnyID:=(select companyID from company where companyName Like compnyName );

set interviewschID:=(select ID from interviewschedule where companyID=compnyID

and Date=interDate);

select interviewschID;

select SSN from registration where ID=interviewschID;

Create table if not exists stdForInter as

(SELECT SSN

FROM

registration

where ID=interviewschID);

select \* from stdForInter;

select person\_name,emailID,phoneNumber,Status

from

person inner join stdForInter

on

person.SSN=stdForInter.SSN;

**END**

set @compnyName='Microsoft';

CALL studentsForInterview(@compnyName,'2016-02-02');

**7]Given the Teacher’s name , List the courses that he/she teaches.**

**SQL=>**

1)set @Professor='E\_kal'

2)select courseName from course

where courseID in

(select courseID from

person inner join batchschedule

on

person.SSN=batchschedule.SSN

where person\_name=@Professor

);

**8]The students recruited by a given the company**

**SQL=>**

Select Count(\*) from studentinterviewrecord

Where

Result Like 'placed' and RegID in (

Select RegID from registration

Where

ID in(

Select ID from interviewschedule

Where

companyID in(

select companyID from company

where companyName Like @compNAme

)

)

)

set @compNAme='Apple'

**9]Given the branch name , list the courses that are taught there.**

**SQl=>**

CREATE DEFINER=`root`@`localhost` PROCEDURE `deptCourseList`(IN Loc varchar(32))

BEGIN

Declare bID int;

set bID:=(select branchID from trainingoffice where location Like Loc) ;

select bID;

CREATE table if not exists DeptArr as

(SELECT departmentID

FROM branchdepartmentlist

WHERE branchID=bID);

create table if not exists CourseArr as

(select courseID from deptcourselist

where

deptID in

(select departmentID from DeptArr )

);

select courseName as Course\_Names from course where courseID in

(select courseID from CourseArr);

END

set @loc='pune';

CALL deptCourseList(@loc);

**Q10]Given the location name , list the students who are placed in through the same branch**

**SQL=>**

select person\_name,phoneNumber

from person inner join student

on

person.SSN=student.SSN

where trainingCompleted Like 'yes'

and branchID in (select branchID from trainingoffice where location Like @locations)

**APPENDIX**

(MYSQL DUMP)

-- MySQL dump 10.13 Distrib 5.7.12, for Win64 (x86\_64)

--

-- Host: 127.0.0.1 Database: training\_and\_placement\_db

-- ------------------------------------------------------

-- Server version 5.6.33

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!40101 SET NAMES utf8 \*/;

/\*!40103 SET @OLD\_TIME\_ZONE=@@TIME\_ZONE \*/;

/\*!40103 SET TIME\_ZONE='+00:00' \*/;

/\*!40014 SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0 \*/;

/\*!40014 SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0 \*/;

/\*!40101 SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='NO\_AUTO\_VALUE\_ON\_ZERO' \*/;

/\*!40111 SET @OLD\_SQL\_NOTES=@@SQL\_NOTES, SQL\_NOTES=0 \*/;

--

-- Table structure for table `adminstaff`

--

DROP TABLE IF EXISTS `adminstaff`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `adminstaff` (

`SSN` int(11) NOT NULL,

`RoleType` varchar(45) NOT NULL,

PRIMARY KEY (`SSN`),

KEY `adminstaff\_ibfk\_1\_idx` (`SSN`),

CONSTRAINT `adminstaff\_ibfk\_1` FOREIGN KEY (`SSN`) REFERENCES `employee` (`SSN`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `adminstaff`

--

LOCK TABLES `adminstaff` WRITE;

/\*!40000 ALTER TABLE `adminstaff` DISABLE KEYS \*/;

/\*!40000 ALTER TABLE `adminstaff` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `batchschedule`

--

DROP TABLE IF EXISTS `batchschedule`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `batchschedule` (

`batchScheduleID` int(11) NOT NULL,

`time` varchar(45) NOT NULL,

`courseID` int(11) DEFAULT NULL,

`SSN` int(11) DEFAULT NULL,

PRIMARY KEY (`batchScheduleID`),

KEY `courseFK\_idx` (`courseID`),

KEY `ssnFK\_idx` (`SSN`),

CONSTRAINT `batchschedule\_ibfk\_1` FOREIGN KEY (`SSN`) REFERENCES `trainers` (`SSN`),

CONSTRAINT `courseFK` FOREIGN KEY (`courseID`) REFERENCES `course` (`courseID`) ON DELETE NO ACTION ON UPDATE NO ACTION

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `batchschedule`

--

LOCK TABLES `batchschedule` WRITE;

/\*!40000 ALTER TABLE `batchschedule` DISABLE KEYS \*/;

INSERT INTO `batchschedule` VALUES (1,'morning',1,114),(2,'evening',2,116),(3,'morning',1,104);

/\*!40000 ALTER TABLE `batchschedule` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `branchdepartmentlist`

--

DROP TABLE IF EXISTS `branchdepartmentlist`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `branchdepartmentlist` (

`idbranchDepartmentListID` int(11) NOT NULL AUTO\_INCREMENT,

`branchID` int(11) DEFAULT NULL,

`departmentID` int(11) DEFAULT NULL,

PRIMARY KEY (`idbranchDepartmentListID`),

KEY `branchFK\_idx` (`branchID`),

KEY `departmentFK\_idx` (`departmentID`),

CONSTRAINT `branchFK` FOREIGN KEY (`branchID`) REFERENCES `trainingoffice` (`branchID`) ON DELETE NO ACTION ON UPDATE NO ACTION,

CONSTRAINT `departmentFK` FOREIGN KEY (`departmentID`) REFERENCES `department` (`departmentID`) ON DELETE NO ACTION ON UPDATE NO ACTION

) ENGINE=InnoDB AUTO\_INCREMENT=7 DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `branchdepartmentlist`

--

LOCK TABLES `branchdepartmentlist` WRITE;

/\*!40000 ALTER TABLE `branchdepartmentlist` DISABLE KEYS \*/;

INSERT INTO `branchdepartmentlist` VALUES (1,101,1),(2,102,1),(3,103,2),(4,101,2),(5,102,2),(6,103,1);

/\*!40000 ALTER TABLE `branchdepartmentlist` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `company`

--

DROP TABLE IF EXISTS `company`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `company` (

`CompanyID` int(11) NOT NULL AUTO\_INCREMENT,

`CompanyName` varchar(45) NOT NULL,

PRIMARY KEY (`CompanyID`)

) ENGINE=InnoDB AUTO\_INCREMENT=5 DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `company`

--

LOCK TABLES `company` WRITE;

/\*!40000 ALTER TABLE `company` DISABLE KEYS \*/;

INSERT INTO `company` VALUES (1,'Microsoft'),(2,'Apple'),(3,'Google'),(4,'Amazon');

/\*!40000 ALTER TABLE `company` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `course`

--

DROP TABLE IF EXISTS `course`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `course` (

`courseID` int(11) NOT NULL,

`courseName` varchar(45) NOT NULL,

`courseFees` int(11) NOT NULL,

PRIMARY KEY (`courseID`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `course`

--

LOCK TABLES `course` WRITE;

/\*!40000 ALTER TABLE `course` DISABLE KEYS \*/;

INSERT INTO `course` VALUES (1,'Automation Testing',1000),(2,'Core Java ',1000);

/\*!40000 ALTER TABLE `course` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `coursearr`

--

DROP TABLE IF EXISTS `coursearr`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `coursearr` (

`courseID` int(11) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `coursearr`

--

LOCK TABLES `coursearr` WRITE;

/\*!40000 ALTER TABLE `coursearr` DISABLE KEYS \*/;

INSERT INTO `coursearr` VALUES (1),(2),(2),(1);

/\*!40000 ALTER TABLE `coursearr` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `department`

--

DROP TABLE IF EXISTS `department`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `department` (

`departmentID` int(11) NOT NULL,

`departmentName` varchar(45) NOT NULL,

PRIMARY KEY (`departmentID`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `department`

--

LOCK TABLES `department` WRITE;

/\*!40000 ALTER TABLE `department` DISABLE KEYS \*/;

INSERT INTO `department` VALUES (1,'Testing'),(2,'Development');

/\*!40000 ALTER TABLE `department` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `deptarr`

--

DROP TABLE IF EXISTS `deptarr`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `deptarr` (

`departmentID` int(11) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `deptarr`

--

LOCK TABLES `deptarr` WRITE;

/\*!40000 ALTER TABLE `deptarr` DISABLE KEYS \*/;

INSERT INTO `deptarr` VALUES (1),(2);

/\*!40000 ALTER TABLE `deptarr` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `deptcourselist`

--

DROP TABLE IF EXISTS `deptcourselist`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `deptcourselist` (

`ID` int(11) NOT NULL,

`deptID` int(11) DEFAULT NULL,

`courseID` int(11) DEFAULT NULL,

PRIMARY KEY (`ID`),

KEY `courseID` (`courseID`),

KEY `deptcourselist\_ibfk\_2\_idx` (`deptID`),

CONSTRAINT `deptcourselist\_ibfk\_1` FOREIGN KEY (`courseID`) REFERENCES `course` (`courseID`),

CONSTRAINT `deptcourselist\_ibfk\_2` FOREIGN KEY (`deptID`) REFERENCES `department` (`departmentID`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `deptcourselist`

--

LOCK TABLES `deptcourselist` WRITE;

/\*!40000 ALTER TABLE `deptcourselist` DISABLE KEYS \*/;

INSERT INTO `deptcourselist` VALUES (1,1,1),(2,2,2),(3,1,2),(4,2,1);

/\*!40000 ALTER TABLE `deptcourselist` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `employee`

--

DROP TABLE IF EXISTS `employee`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `employee` (

`SSN` int(11) NOT NULL DEFAULT '0',

`employeeID` int(11) DEFAULT '101',

`salary` int(11) NOT NULL DEFAULT '800',

`employeeType` varchar(45) NOT NULL,

PRIMARY KEY (`SSN`),

CONSTRAINT `SSN` FOREIGN KEY (`SSN`) REFERENCES `person` (`SSN`) ON DELETE NO ACTION ON UPDATE NO ACTION

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `employee`

--

LOCK TABLES `employee` WRITE;

/\*!40000 ALTER TABLE `employee` DISABLE KEYS \*/;

INSERT INTO `employee` VALUES (104,1,1000,'T'),(105,2,1000,'A'),(106,3,1000,'T'),(109,5,800,'A'),(111,6,800,'T'),(112,101,800,'T'),(113,101,800,'T'),(114,101,800,'T'),(116,101,800,'T');

/\*!40000 ALTER TABLE `employee` ENABLE KEYS \*/;

UNLOCK TABLES;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

/\*!50003 CREATE\*/ /\*!50017 DEFINER=`root`@`localhost`\*/ /\*!50003 TRIGGER `training\_and\_placement\_db`.`employee\_AFTER\_INSERT` AFTER INSERT ON `employee` FOR EACH ROW

BEGIN

set @etype=new.employeeType;

set @ssn=new.SSN;

if @etype Like 'T' then

insert into trainers (SSN) values (@ssn);

#elseif new.etype Like'A' then

#insert into employee(SSN) values (@ssn);

end if ;

END \*/;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

/\*!50003 CREATE\*/ /\*!50017 DEFINER=`root`@`localhost`\*/ /\*!50003 TRIGGER `training\_and\_placement\_db`.`employee\_AFTER\_UPDATE` AFTER UPDATE ON `employee` FOR EACH ROW

BEGIN

set @etype=new.employeeType;

set @ssn=new.SSN;

if @etype Like 'T' then

insert into trainers (SSN) values (@ssn);

#elseif new.etype Like'A' then

#insert into employee(SSN) values (@ssn);

end if ;

END \*/;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

--

-- Table structure for table `interviewschedule`

--

DROP TABLE IF EXISTS `interviewschedule`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `interviewschedule` (

`ID` int(11) NOT NULL,

`Date` date NOT NULL,

`CompanyID` int(11) DEFAULT NULL,

`Requirement` varchar(55) DEFAULT NULL,

PRIMARY KEY (`ID`),

KEY `companyFK\_idx` (`CompanyID`),

CONSTRAINT `companyFK` FOREIGN KEY (`CompanyID`) REFERENCES `company` (`CompanyID`) ON DELETE NO ACTION ON UPDATE NO ACTION

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `interviewschedule`

--

LOCK TABLES `interviewschedule` WRITE;

/\*!40000 ALTER TABLE `interviewschedule` DISABLE KEYS \*/;

INSERT INTO `interviewschedule` VALUES (1,'2016-02-02',1,'C#'),(2,'2016-02-02',2,'Obj C'),(3,'2016-02-02',3,'web'),(4,'2016-02-02',4,'System Algorithm');

/\*!40000 ALTER TABLE `interviewschedule` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `person`

--

DROP TABLE IF EXISTS `person`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `person` (

`SSN` int(11) NOT NULL,

`person\_name` varchar(32) DEFAULT NULL,

`emailID` varchar(32) DEFAULT NULL,

`address` varchar(32) DEFAULT NULL,

`branchID` int(11) DEFAULT NULL,

`personType` varchar(45) NOT NULL,

`phoneNumber` varchar(45) NOT NULL,

`Status` varchar(45) DEFAULT NULL,

PRIMARY KEY (`SSN`),

UNIQUE KEY `phoneNumber\_UNIQUE` (`phoneNumber`),

UNIQUE KEY `emailID` (`emailID`),

KEY `branchFK\_idx` (`branchID`),

CONSTRAINT `person\_ibfk\_1` FOREIGN KEY (`branchID`) REFERENCES `trainingoffice` (`branchID`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `person`

--

LOCK TABLES `person` WRITE;

/\*!40000 ALTER TABLE `person` DISABLE KEYS \*/;

INSERT INTO `person` VALUES (100,'Prashant','psswl21@gmail.com','boston',101,'S','857-544-1009','Active'),(101,'Prathamesh','Landekar.p@gmail.com','boston',102,'S','857-544-1040','Active'),(103,'Rohan','Magare.R@gmail.com','boston',103,'S','857-544-1001','Inactive'),(104,'E\_kal','Kal@gmail.com','pune',101,'E','857-544-1002','Active'),(105,'E\_Sachin','Sachin@gmail.com','mumbai',102,'E','857-544-1003','Active'),(106,'E\_jyoti','jyoti@gmail.com','bangalore',103,'E','857-544-1004','Active'),(107,'Vedant','vedant@gmail.com','pune',101,'S','857-544-1005','Active'),(108,'E\_pranav','pranav@gmail.com','mumbai',102,'E','857-544-1006','Inactive'),(109,'E\_prof1','profEID','pune',101,'E','857','Active'),(110,'Saurabh','saurabh@gmail.com','banagalore',103,'S','857-544-1011','Active'),(111,'SavhinGavande','SavhinGavande@gmail.com','mumbai',102,'E','857-544-1089','Inactive'),(112,'E\_prof2','prof2@gmail.com','pune',101,'E','857-544-1090','Active'),(113,'E\_prof3','E\_prof3@gmail.com','mumbai',102,'E','857-544-1091','Inactive'),(114,'E\_prof4','E\_prof4@gmail.com','pune',101,'E','857-544-1092','Active'),(115,'Shruthey','Shruthey@gmail.com','mumbai',102,'S','857-544-1093','Inactive'),(116,'E\_prof5','E\_prof5@gmail.com','bangalore',103,'E','857-544-1094','Active'),(117,'S\_vedant','S\_vedant@gmail.com','pune',101,'S','857-544-1095','Active');

/\*!40000 ALTER TABLE `person` ENABLE KEYS \*/;

UNLOCK TABLES;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

/\*!50003 CREATE\*/ /\*!50017 DEFINER=`root`@`localhost`\*/ /\*!50003 TRIGGER `training\_and\_placement\_db`.`person\_AFTER\_INSERT` AFTER INSERT ON `person` FOR EACH ROW

BEGIN

set @ptype=new.persontype;

set @ssn=new.SSN;

if @ptype Like 'S' then

insert into student (SSN) values (@ssn);

elseif new.personType Like'E' then

insert into employee(SSN) values (@ssn);

end if ;

END \*/;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

--

-- Table structure for table `registration`

--

DROP TABLE IF EXISTS `registration`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `registration` (

`RegID` int(11) NOT NULL,

`ID` int(11) DEFAULT NULL,

`SSN` int(11) DEFAULT NULL,

PRIMARY KEY (`RegID`),

KEY `ID` (`ID`),

KEY `SSN` (`SSN`),

CONSTRAINT `registration\_ibfk\_1` FOREIGN KEY (`ID`) REFERENCES `interviewschedule` (`ID`),

CONSTRAINT `registration\_ibfk\_2` FOREIGN KEY (`SSN`) REFERENCES `student` (`SSN`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `registration`

--

LOCK TABLES `registration` WRITE;

/\*!40000 ALTER TABLE `registration` DISABLE KEYS \*/;

INSERT INTO `registration` VALUES (1,1,100),(2,2,101),(3,3,103),(4,4,110),(5,4,115),(6,3,110),(7,2,115),(8,1,103),(9,3,115);

/\*!40000 ALTER TABLE `registration` ENABLE KEYS \*/;

UNLOCK TABLES;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

/\*!50003 CREATE\*/ /\*!50017 DEFINER=`root`@`localhost`\*/ /\*!50003 TRIGGER `training\_and\_placement\_db`.`registration\_AFTER\_INSERT` AFTER INSERT ON `registration` FOR EACH ROW

BEGIN

set @LastInsertedID =(select MAX(regID) from registration);

insert into studentinterviewrecord (RegID,Result)values(@LastInsertedID ,'painding');

END \*/;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

--

-- Table structure for table `stdforinter`

--

DROP TABLE IF EXISTS `stdforinter`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `stdforinter` (

`SSN` int(11) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `stdforinter`

--

LOCK TABLES `stdforinter` WRITE;

/\*!40000 ALTER TABLE `stdforinter` DISABLE KEYS \*/;

INSERT INTO `stdforinter` VALUES (100),(103);

/\*!40000 ALTER TABLE `stdforinter` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `student`

--

DROP TABLE IF EXISTS `student`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `student` (

`SSN` int(11) NOT NULL,

`studentID` varchar(45) NOT NULL,

`status` varchar(45) NOT NULL,

`trainingCompleted` varchar(45) NOT NULL,

`enrollDate` date NOT NULL,

`feesPaid` int(11) NOT NULL,

`feesDue` int(11) NOT NULL,

`IsActive` varchar(45) NOT NULL,

PRIMARY KEY (`SSN`),

CONSTRAINT `ssnfk` FOREIGN KEY (`SSN`) REFERENCES `person` (`SSN`) ON DELETE NO ACTION ON UPDATE NO ACTION

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `student`

--

LOCK TABLES `student` WRITE;

/\*!40000 ALTER TABLE `student` DISABLE KEYS \*/;

INSERT INTO `student` VALUES (100,'st1','unplaced','yes','2016-10-10',1000,0,'Active'),(101,'st2','placed','no','2016-10-11',500,0,'Active'),(103,'st3','placed','yes','2016-06-12',1000,2000,'Active'),(110,'st5','unplaced','yes','2016-02-12',900,0,'Active'),(115,'st6','unplaced','yes','2016-04-01',600,1000,'Active'),(117,'st7','unplaced','yes','2016-05-01',0,0,'Active');

/\*!40000 ALTER TABLE `student` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `studentaccount`

--

DROP TABLE IF EXISTS `studentaccount`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `studentaccount` (

`studentaccountNumber` int(11) NOT NULL,

`SSN` int(11) NOT NULL,

`FeesPaid` varchar(45) NOT NULL,

PRIMARY KEY (`studentaccountNumber`,`SSN`),

KEY `ssnFK\_idx` (`SSN`),

CONSTRAINT `studentaccount\_ibfk\_1` FOREIGN KEY (`SSN`) REFERENCES `student` (`SSN`),

CONSTRAINT `studentaccount\_ibfk\_2` FOREIGN KEY (`SSN`) REFERENCES `student` (`SSN`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `studentaccount`

--

LOCK TABLES `studentaccount` WRITE;

/\*!40000 ALTER TABLE `studentaccount` DISABLE KEYS \*/;

/\*!40000 ALTER TABLE `studentaccount` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `studentbatchschedule`

--

DROP TABLE IF EXISTS `studentbatchschedule`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `studentbatchschedule` (

`ID` int(11) NOT NULL,

`SSN` int(11) DEFAULT NULL,

`batchScheduleID` int(11) DEFAULT NULL,

KEY `batchScheduleID` (`batchScheduleID`),

KEY `SSN` (`SSN`),

CONSTRAINT `studentbatchschedule\_ibfk\_1` FOREIGN KEY (`batchScheduleID`) REFERENCES `batchschedule` (`batchScheduleID`),

CONSTRAINT `studentbatchschedule\_ibfk\_2` FOREIGN KEY (`SSN`) REFERENCES `student` (`SSN`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `studentbatchschedule`

--

LOCK TABLES `studentbatchschedule` WRITE;

/\*!40000 ALTER TABLE `studentbatchschedule` DISABLE KEYS \*/;

INSERT INTO `studentbatchschedule` VALUES (1,100,1),(2,101,2),(3,100,2),(4,115,1),(5,103,1),(6,103,2);

/\*!40000 ALTER TABLE `studentbatchschedule` ENABLE KEYS \*/;

UNLOCK TABLES;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

/\*!50003 CREATE\*/ /\*!50017 DEFINER=`root`@`localhost`\*/ /\*!50003 TRIGGER `training\_and\_placement\_db`.`studentbatchschedule\_AFTER\_INSERT` AFTER INSERT ON `studentbatchschedule` FOR EACH ROW

BEGIN

set @newbatchSchID=new.batchScheduleID;

set @cID =(select courseID from batchSchedule where batchScheduleID=@newbatchSchID);

set @fees=(select courseFees from course where courseID=@cID);

set @currfees=(select feesDue from student where SSN=new.SSN);

update student set feesDue=(@currfees+@fees)

where SSN=new.SSN;

END \*/;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

--

-- Table structure for table `studentinterviewrecord`

--

DROP TABLE IF EXISTS `studentinterviewrecord`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `studentinterviewrecord` (

`RecordNo` int(11) NOT NULL AUTO\_INCREMENT,

`RegID` int(11) DEFAULT NULL,

`Result` varchar(45) DEFAULT NULL,

PRIMARY KEY (`RecordNo`),

KEY `RegID` (`RegID`),

CONSTRAINT `studentinterviewrecord\_ibfk\_1` FOREIGN KEY (`RegID`) REFERENCES `registration` (`RegID`)

) ENGINE=InnoDB AUTO\_INCREMENT=12 DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `studentinterviewrecord`

--

LOCK TABLES `studentinterviewrecord` WRITE;

/\*!40000 ALTER TABLE `studentinterviewrecord` DISABLE KEYS \*/;

INSERT INTO `studentinterviewrecord` VALUES (1,1,'places'),(2,2,'placed'),(5,3,'painding'),(6,4,'painding'),(7,5,'painding');

/\*!40000 ALTER TABLE `studentinterviewrecord` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `studentsforinter`

--

DROP TABLE IF EXISTS `studentsforinter`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `studentsforinter` (

`SSN` int(11) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `studentsforinter`

--

LOCK TABLES `studentsforinter` WRITE;

/\*!40000 ALTER TABLE `studentsforinter` DISABLE KEYS \*/;

INSERT INTO `studentsforinter` VALUES (100),(103);

/\*!40000 ALTER TABLE `studentsforinter` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `tempertable`

--

DROP TABLE IF EXISTS `tempertable`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `tempertable` (

`SSN` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `tempertable`

--

LOCK TABLES `tempertable` WRITE;

/\*!40000 ALTER TABLE `tempertable` DISABLE KEYS \*/;

INSERT INTO `tempertable` VALUES (100),(107);

/\*!40000 ALTER TABLE `tempertable` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `temporarytable`

--

DROP TABLE IF EXISTS `temporarytable`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `temporarytable` (

`SSN` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `temporarytable`

--

LOCK TABLES `temporarytable` WRITE;

/\*!40000 ALTER TABLE `temporarytable` DISABLE KEYS \*/;

INSERT INTO `temporarytable` VALUES (100),(107);

/\*!40000 ALTER TABLE `temporarytable` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `temptable`

--

DROP TABLE IF EXISTS `temptable`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `temptable` (

`SSN` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `temptable`

--

LOCK TABLES `temptable` WRITE;

/\*!40000 ALTER TABLE `temptable` DISABLE KEYS \*/;

INSERT INTO `temptable` VALUES (100),(107);

/\*!40000 ALTER TABLE `temptable` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `trainers`

--

DROP TABLE IF EXISTS `trainers`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `trainers` (

`SSN` int(11) NOT NULL,

`trainerType` varchar(32) NOT NULL,

PRIMARY KEY (`SSN`),

KEY `trainers\_ibfk\_1\_idx` (`SSN`),

CONSTRAINT `trainers\_ibfk\_1` FOREIGN KEY (`SSN`) REFERENCES `employee` (`SSN`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `trainers`

--

LOCK TABLES `trainers` WRITE;

/\*!40000 ALTER TABLE `trainers` DISABLE KEYS \*/;

INSERT INTO `trainers` VALUES (104,'Permanent'),(113,'Guest'),(114,'Permanent'),(116,'Permanent');

/\*!40000 ALTER TABLE `trainers` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `trainingoffice`

--

DROP TABLE IF EXISTS `trainingoffice`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `trainingoffice` (

`branchID` int(11) NOT NULL,

`location` varchar(32) DEFAULT NULL,

PRIMARY KEY (`branchID`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `trainingoffice`

--

LOCK TABLES `trainingoffice` WRITE;

/\*!40000 ALTER TABLE `trainingoffice` DISABLE KEYS \*/;

INSERT INTO `trainingoffice` VALUES (101,'Pune'),(102,'Mumbai'),(103,'Bangalore');

/\*!40000 ALTER TABLE `trainingoffice` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Dumping events for database 'training\_and\_placement\_db'

--

--

-- Dumping routines for database 'training\_and\_placement\_db'

--

/\*!50003 DROP PROCEDURE IF EXISTS `deptCourseList` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `deptCourseList`(IN Loc varchar(32))

BEGIN

Declare bID int;

set bID:=(select branchID from trainingoffice where location Like Loc) ;

select bID;

CREATE table if not exists DeptArr as

(SELECT departmentID

FROM branchdepartmentlist

WHERE branchID=bID);

select \* from DeptArr;

create table if not exists CourseArr as

(select courseID from deptcourselist

where

deptID in

(select departmentID from DeptArr )

);

select courseName as Course\_Names from course where courseID in

(select courseID from CourseArr);

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `Find\_unplacedStudents` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `Find\_unplacedStudents`(IN Loc varchar(32) )

BEGIN

Declare bID int;

Declare countOfUnplacedStudent int;

set bID:=(select branchID from trainingoffice where location Like Loc) ;

CREATE table if not exists temporaryTable as

(SELECT SSN

FROM Person

WHERE personType Like 'S' and branchID=bID);

set countOfUnplacedStudent=(select count(\*) from

temporaryTable tt inner join student s

on

tt.SSN=s.SSN

where status Like '%Unplaced%');

SELECT 'The total numberof Unplaced students are',countOfUnplacedStudent ;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `studentsForInterview` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `studentsForInterview`(IN compnyName varchar(32), interDate date)

BEGIN

declare compnyID int ;

declare interviewschID int ;

set compnyID:=(select companyID from company where companyName Like compnyName );

set interviewschID:=(select ID from interviewschedule where companyID=compnyID

and Date=interDate);

select interviewschID;

select SSN from registration where ID=interviewschID;

Create table if not exists stdForInter as

(SELECT SSN

FROM

registration

where ID=interviewschID);

select \* from stdForInter;

select person\_name,emailID,phoneNumber,Status

from

person inner join stdForInter

on

person.SSN=stdForInter.SSN;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!40103 SET TIME\_ZONE=@OLD\_TIME\_ZONE \*/;

/\*!40101 SET SQL\_MODE=@OLD\_SQL\_MODE \*/;

/\*!40014 SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS \*/;

/\*!40014 SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS \*/;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

/\*!40111 SET SQL\_NOTES=@OLD\_SQL\_NOTES \*/;

-- Dump completed on 2016-12-12 10:40:41