DBMS - 1

Theory Assignment 3

**SUBMITTED BY:**

**SHAHZANEER AHMED**

**REGISTRATION**

**NUMBER**

**:**

**SP21-BCS-087**

**SUBMITTED TO**

**:**

**Dr. Basit Raza**

**DATE**

**OF**

**SUBMISSION:**

**December 29th, 2022**

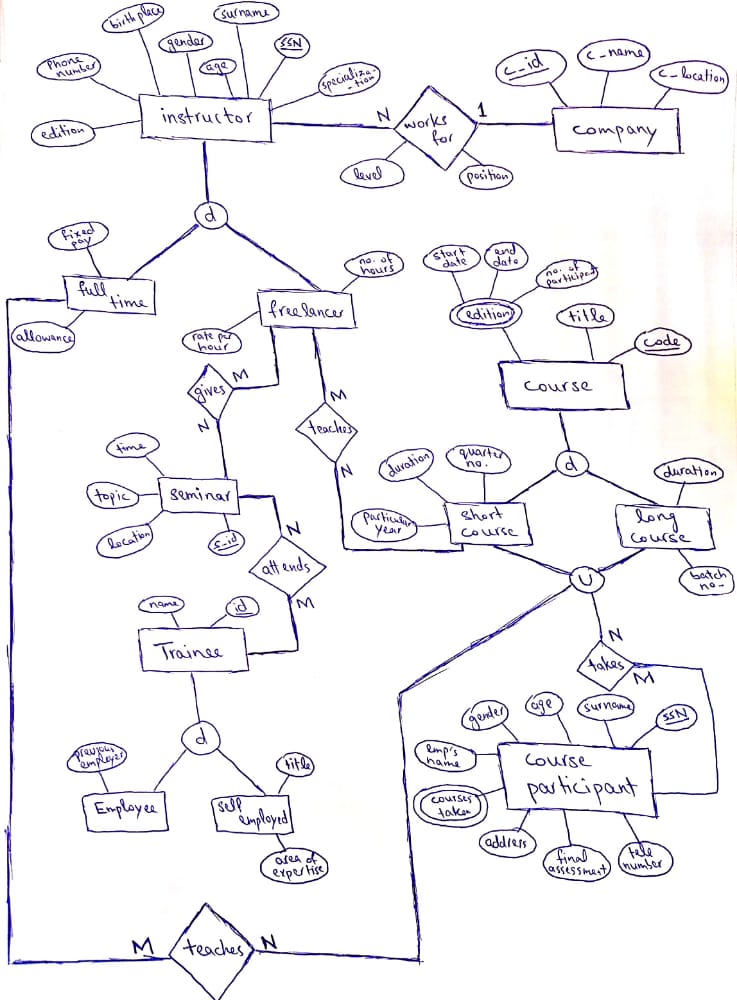


**For each of the following problems, first create an Enhance Entity Relationship (EER) diagram and then convert it into a relational database schema. Assume attributes as per your real-world knowledge for the given scenarios, if not mentioned.**

# Problem-1: Course training [1]

Create a database for a company that provides training. This necessitates the storage of information about the trainees and instructors. For each course participant, you must store their social security number, surname, age, gender, place of birth, employer’s name, address, and telephone number, courses taken (approximately 200 courses), and final assessment for each course. Trainee can be an employee or can be self-employed. If the trainee is an employee, you need to store previous employers (and periods employed). If a trainee is self-employed, you must recognize their area of expertise and, if applicable, their title. You must save the level and position held by someone who works for a company. You must also reflect on the seminars that each participant is currently attending, as well as the locations and times of the classes for each day. Each course has a code and a title, and any course can be given multiple times. Every time a specific course is taught, it is referred to as an ‘edition’ of the course. You represent the start date, end date, and the number of participants for each edition. The course can be for three months (short course) and can be of six months (long course). short courses have quarter number with particular year. Long courses have batch numbers (like spring22 or fall22). You will list the surname, age, place of birth, edition of the course taught, previous courses taught, and courses that the tutor is qualified to teach for each instructor. The phone numbers of all the instructors are also saved. An instructor can work for a training company full-time or as a freelancer. Full-time instructors get fixed pay and allowances. Freelancer gets pay according to the no. of hours and rate per hour (which is different for different instructor).

**EER DIAGRAM of PROBLEM 1:**

****

**Relational Schema of Problem 1**

Company (companyID, companyName, companyLocation)

Instructor (SSN, surname, specialization, age, gender, birthplace, phoneNo, edition)

WorksFor (companyID [FK], SSN [FK], level, position)

Full-time\_Instructor (SSN [FK], fixedPay, allowance)

Freelancer (SSN [FK], ratePerHour, noOfHours)

Seminar (seminarID, time, topic, location)

freelancerGivesSeminar (SSN [FK], seminarID [FK])

Trainee (traineeID, name)

traineeAttendsSeminar (traineeID [FK], seminarID [FK])

Employee (traineeID [FK], previousEmployer)

Self\_Employeed (traineeID [FK], title, areaOfExpertise)

Course (course\_code, title)

Course\_Edition (course\_code [FK], start\_date, end\_date, noOfParticipants)

Short\_Course (course\_code [FK], year, duration = 3 months, quarterNo)

Long\_Course (course\_code [FK], duration = 6 months, batchNo)

Course\_Participant (SSN, surname, age, gender, employerName, address, finalAssessment, phoneNo)

Courses\_Taken (SSN [FK], course\_code [FK])

# Problem-2: Canteen/cafeteria [2]

Design an EER-schema for a database of a canteen/cafeteria. The canteen has a certain feature of menus it can produce. Each menu has an identifying number, a name, and a price. The name is used for advertising the menu; menus which differ only slightly can have the same name. Each day, the canteen offers several menus. It wants to store which menu was offered on which day and how often it was sold. Internally, the menus are constructed from a main course (usually meat) and several side dishes (such as soup, salad, vegetables, dessert). In this canteen the customer cannot choose the side dishes. The composition of menu is used only for the preparation because every component (main dish or side dish) can be prepared independently. Also, if some component is used in different menus, the information about it does not have to be stored redundantly. For every menu component, the recipe has to be stored (how to cook this part of the meal). It is important that the type of distinction (main course or side dish) is represented and that every menu consists of exactly one main course. Finally, the ingredients of the menu components have to be stored (e.g. potatoes, carrots, cheese,). For each ingredient, the name and number of calories per 100g are stored. An ingredient can be used for several menu components. You also have to store how many grams of each ingredient are used for a menu component. Please do not forget to show the keys and cardinalities. If other constraints should be needed, it suffices to sketch them in natural language.

**EER DIAGRAM of PROBLEM 2:**

**Diagram

Description automatically generated**

**Relational Schema of Problem 2**

Canteen (canteenID, menuName, menusSold, day)

Menu (menuID, name, price, mainCourse, canteenID [FK])

Menu\_Components (componentID, recipe, menuID [FK])

Main\_Dish (componentID [FK], name, quantity)

Side\_Dish (componentID [FK], name, quantity)

Ingredients (ingredientID, name, noOfCalories, gramsUsed)

Makes (ingredientID [FK], componentID [FK])

# Problem-3: Cricket Website Database [3]

The database for a cricket website needs to be built. The following set of requirements have been identified:

Players group together to form teams. Players can be local and international. Local players will be part of any club. International players will have no. of international matches played.

Teams have coaches. Coaches have certain information which is needed to store in database.

Teams can participate in tournaments.

Tournaments comprise of matches. These matches contain match id, date, and other description. Match can be one day or test match. For one day match we need to store the number of overs and for test matches we need to store no. of days and no. of innings.

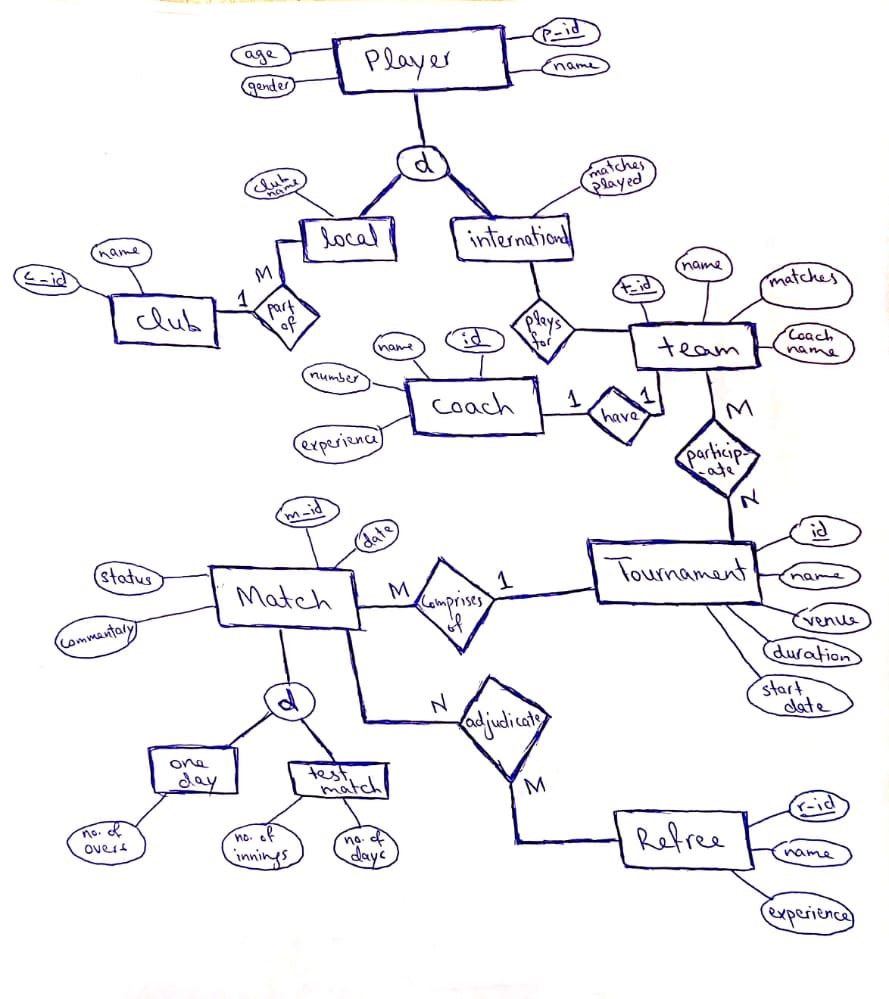
Referees adjudicate matches.

Matches are played at a certain venue.

A subset of players from a team participates in a certain match.

For matches, apart from the basic information e.g., won/lost, its event-wise chronological coverage throughout the course of the match, that include commentary and other associated information, e.g., goal scored, red card given etc., also needs to be stored.

**EER DIAGRAM of PROBLEM 3:**



**Relational Schema of Problem 3**

Player (playerID, name, age, gender)

Local\_Player (playerID [FK], clubID [FK])

Intrnational\_Player (playerID [FK], matchesPlayed, teamID [FK])

Club (clubID, name)

Team (teamID, name, matches, coachID [FK])

Coach (coachID, name, number, experience)

Tournament (tournamentID, name, venue, duration, start\_date)

Participates (teamID [FK], tournamentID [FK])

Match (matchID, date, status, commentary, tournamentID [FK])

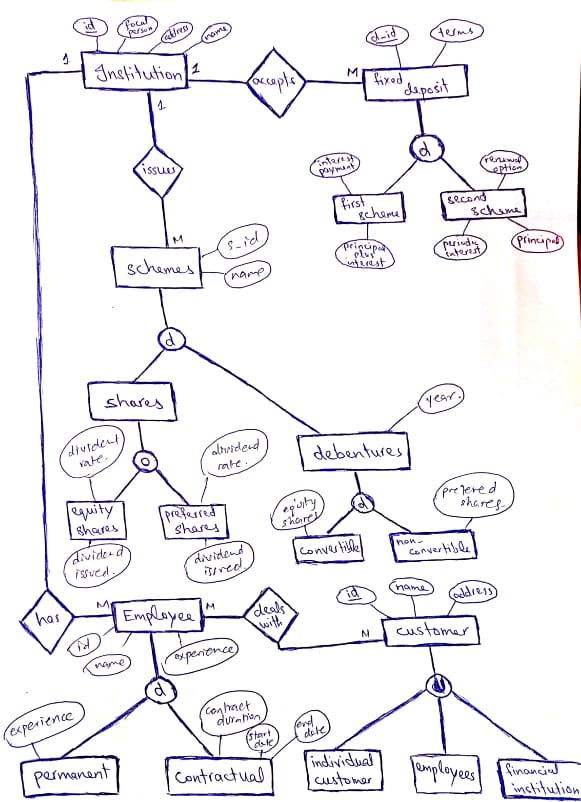
One\_Day\_Match (matchID [FK], noOfOvers)

Test\_Match (matchID [FK], noOfInnings, noOfDays)

Referee (refereeID, name, experience)

Adjudicate (matchID [FK], refereeID [FK])

**EER DIAGRAM of PROBLEM 4:**



**Relational Schema of Problem 4**

Institution (institution\_id, focalPerson, address, name)

Fixed\_Deposits (deposit\_id, terms, institution\_id [FK])

First\_Scheme (deposit\_id [FK], interest, principal\_payment)

Second\_Scheme (deposit\_id [FK], renewalOption, periodicInterest, principalPayment)

Scheme (scheme\_id, name, institution\_id [FK])

Shares (scheme\_id [FK], share\_id)

Equity\_Share (share\_id [FK], dividendRate, dividendIssued)

Preferred\_Share (share\_id [FK], dividendRate, dividendIssued)

Debentures (scheme\_id [FK], year, debenture\_id)

Convertible\_Debenture (debenture\_id [FK], equity\_share)

Non-Convertible\_Debenture (debenture\_id [FK], prefered\_share)

Employee (employeeID, name, experience)

Permanent\_Employee (employeeID [FK], experience)

Contractual\_Employee (employeeID [FK], contractDuration, start\_date, end\_date)

Customer (customerID, name, address)

Individual\_Customer (customerID [FK])

Employee (customerID [FK], employeeID [FK])

Financial\_Institution (customerID [FK], institution\_id [FK])

Employee\_Manages\_Customer (employeeID [FK], customerID [FK])