CHAPTER 3: DATA MODELING USING THE ENTITY-RELATIONSHIP (ER) MODEL

Answers to Selected Exercises

3.16 - Consider the following set of requirements for a UNIVERSITY database that is used to keep track of students' transcripts. This is similar but not identical to the database shown in Figure 1.2:

(a) The university keeps track of each student's name, student number, social security number, current address and phone, permanent address and phone, birthdate, sex, class (freshman, sophomore, ..., graduate), major department, minor department (if any), and degree program (B.A., B.S., ..., Ph.D.). Some user applications need to refer to the city, state, and zip of the student's permanent address, and to the student's last name. Both social security number and student number have unique values for each student.

(b) Each department is described by a name, department code, office number, office phone, and college. Both name and code have unique values for each department.

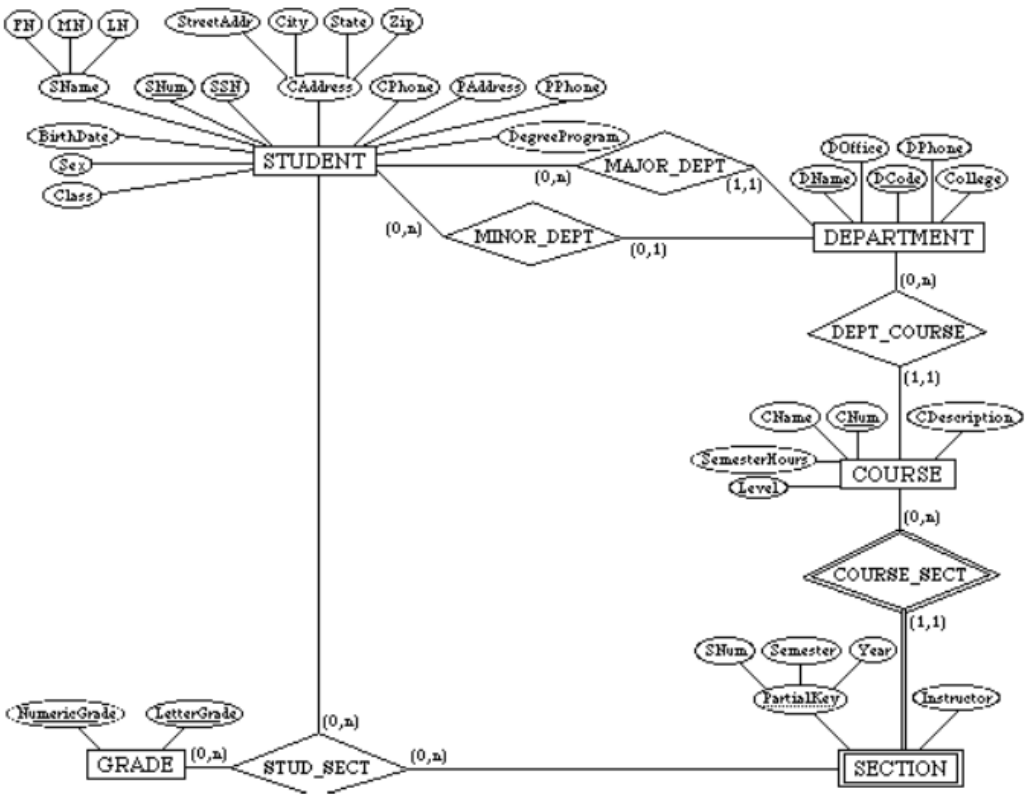
(c) Each course has a course name, description, course number, number of semester hours, level, and offering department. The value of course number is unique for each course.

(d) Each section has an instructor, semester, year, course, and section number. The section number distinguishes different sections of the same course that are taught during the same semester/year; its values are 1, 2, 3, ..., up to the number of sections taught during each semester.

(e) A grade report has a student, section, letter grade, and numeric grade (0, 1, 2, 3, 4 for F, D, C, B, A, respectively).

Design an ER schema for this application, and draw an ER diagram for that schema. Specify key attributes of each entity type and structural constraints on each relationship type. Note any unspecified requirements, and make appropriate assumptions to make the specification complete.

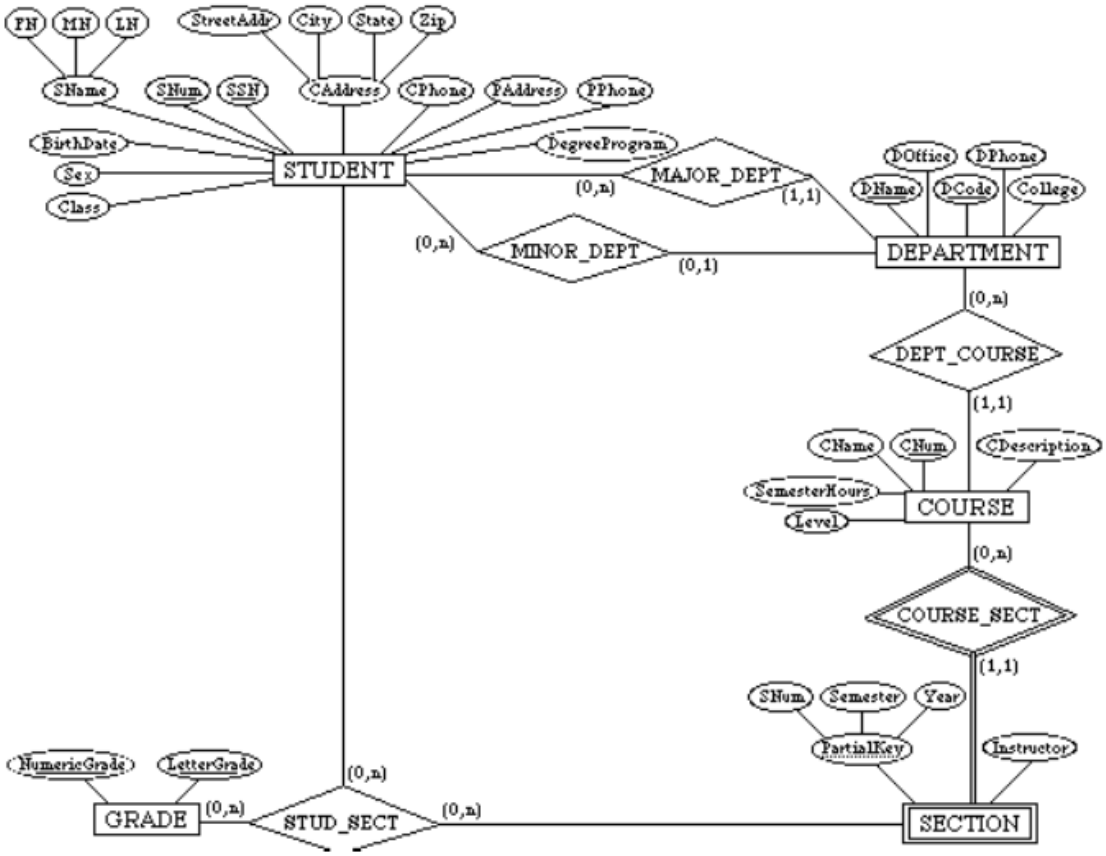
Answer:



ER Schema diagram for exercise 3.16:

3.17 - Composite and multi-valued attributes can be nested to any number of levels. Suppose we want to design an attribute for a STUDENT entity type to keep track of previous college education. Such an attribute will have one entry for each college previously attended, and this entry is composed of: college name, start and end dates, degree entries (degrees awarded at that college, if any), and transcript entries (courses completed at that college, if any). Each degree entry is formed of degree name and the month and year it was awarded, and each transcript entry is formed of a course name, semester, year, and grade. Design an attribute to hold this information. Use the conventions of Figure 7.5.

Answer:
{ PreviousEducation (CollegeName, StartDate, EndDate,
{ Degree (DegreeName, Month, Year) },
{ Transcript (CourseName, Semester, Year, Grade) }) }

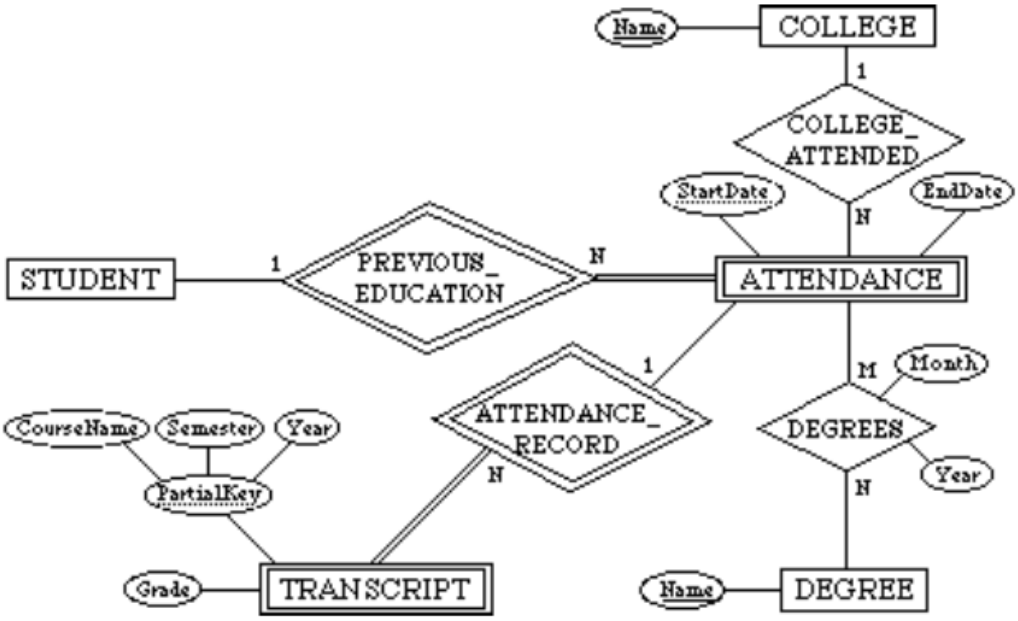


ER Schema Diagram for Exercise 3.16

3.18 - Show an alternative design for the attribute described in Exercise 7.17 that uses only entity types (including weak entity types if needed) and relationship types.

Answer:

This example illustrates a perceived weakness of the ER model, which is: how does the database designer decide what to model as an entity type and what to model as a relationship type. In our solution, we created a weak entity type ATTENDANCE; each (weak) entity in ATTENDANCE represents a period in which a STUDENT attended a particular COLLEGE, and is identified by the STUDENT and the StartDate of the period. Hence, the StartDate attribute is the partial key of ATTENDANCE. Each ATTENDANCE entity is related to one COLLEGE and zero or more DEGREEs (the degrees awarded during that attendance period). The TRANSCRIPT of the STUDENT during each attendance period is modeled as a weak entity type, which gives the records of the student during the attendance period. Each (weak) entity in TRANSCRIPT gives the record of the sudent in one course during the attendance period, as shown in the ER diagram below. Other ER schema designs are also possible for this problem.



3.19 - Consider the ER diagram of Figure 7.20, which shows a simplified schema for an airline reservations system. Extract from the ER diagram the requirements and constraints that resulted in this schema. Try to be as precise as possible in your requirements and constraints specification.

Answer:
(1) The database represents each AIRPORT, keeping its unique AirportCode, the AIRPORT Name, and the City and State in which the AIRPORT is located.

(2) Each airline FLIGHT has a unique number, the Airline for the FLIGHT, and the Weekdays on which the FLIGHT is scheduled (for example, every day of the week except Sunday can be coded as X7).

(3) A FLIGHT is composed of one or more FLIGHT LEGs (for example, flight number CO1223 from New York to Los Angeles may have two FLIGHT LEGs: leg 1 from New York to Houston and leg 2 from Houston to Los Angeles). Each FLIGHT LEG has a DEPARTURE AIRPORT and Scheduled Departure Time, and an ARRIVAL AIRPORT and Scheduled Arrival Time.

(4) A LEG INSTANCE is an instance of a FLIGHT LEG on a specific Date (for example, CO1223 leg 1 on July 30, 1989). The actual Departure and Arrival AIRPORTs and Times are recorded for each flight leg after the flight leg has been concluded. The Number of available seats and the AIRPLANE used in the LEG INSTANCE are also kept.

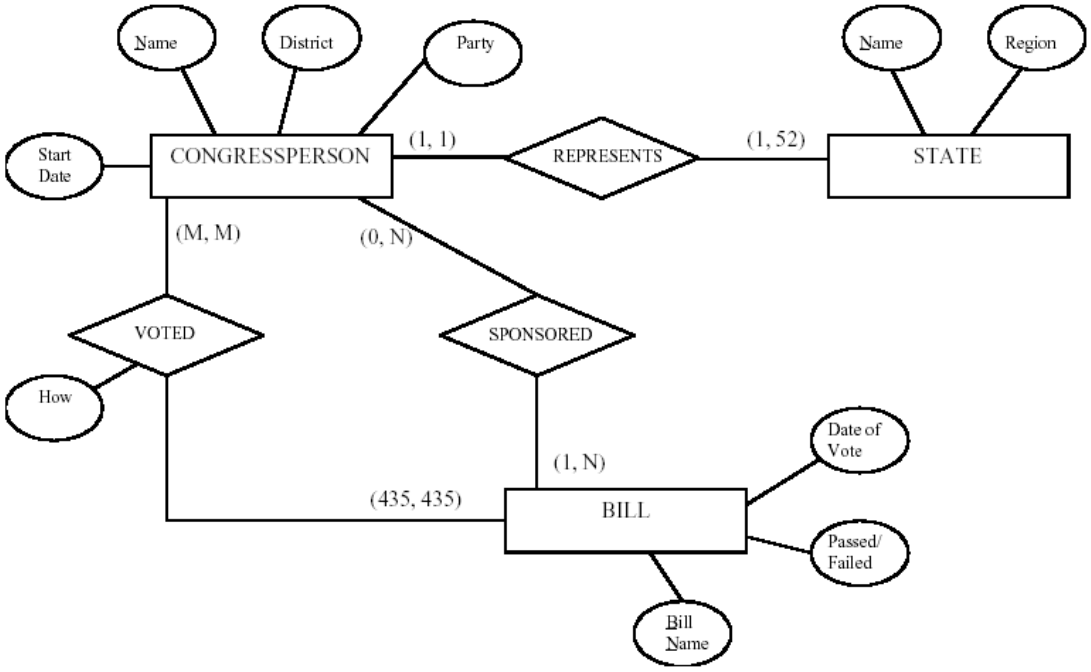
(5) The customer RESERVATIONs on each LEG INSTANCE include the Customer Name, Phone, and Seat Number(s) for each reservation.

(6) Information on AIRPLANEs and AIRPLANE TYPEs are also kept. For each AIRPLANE TYPE (for example, DC-10), the TypeName, manufacturing Company, and Maximum Number of Seats are kept. The AIRPORTs in which planes of this type CAN LAND are kept in the database. For each AIRPLANE, the AirplaneId, Total number of seats, and TYPE are kept.

3.20 - No solution provided.

3.21 -
Additional information:
- There are 435 congresspersons in the U.S. House of Representatives.
- States have between one (AK, DE, MT, ND, SD, VT, and WY) and 52 (CA) representatives.
- M represents number of bills during the 2-year session.

The resulting ER Diagram is shown in Figure A.

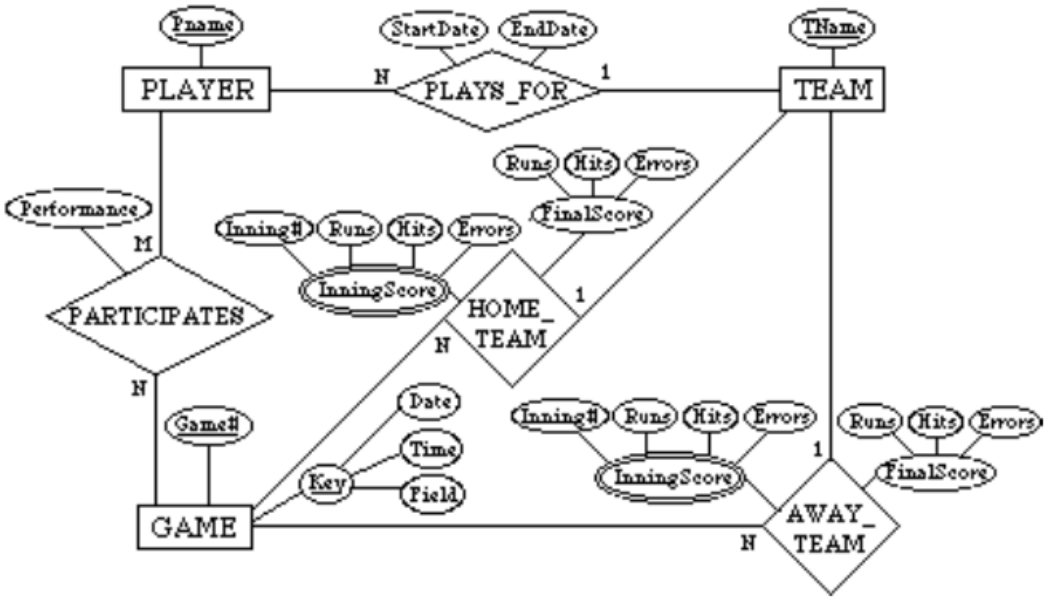


3.22 - A database is being constructed to keep track of the teams and games of a sports league. A team has a number of players, not all of whom participate in each game.
It is desired to keep track of the players participating in each game for each team, the positions they played in that game, and the result of the game. Try to design an ER schema diagram for this application, stating any assumptions you make. Choose your favorite sport (soccer, football, baseball ...).

Answer:
The following design may be used for a baseball league. Here, we assumed that each game in the schedule is identified by a unique Game#, and a game is also identified uniquely by the combination of Date, starting Time, and Field where it is played. The Performance attribute of PARTICIPATE is used to store information on the individual performance of each

player in a game. This attribute can be designed to keep the information needed for statistics, and may be quite complex. One possible design for the Performance attribute may be the following (using the notation of Figure 7.8):

Performance({Hitting(AtBat#, Inning#, HitType, Runs, RunsBattedIn, StolenBases)},
{Pitching(Inning#, Hits, Runs, EarnedRuns, StrikeOuts, Walks, Outs,
Balks, WildPitches)},
{Defense(Inning#, {FieldingRecord(Position, PutOuts, Assists, Errors)})})
Here, performance is a composite attribute made up of three multivalued components: Hitting, Pitching, and Defense. Hitting has a value for each AtBat of a player, and records the HitType (suitable coded; for example, 1 for single, 2 for double, 3 for triple, 4 for home run, 0 for walk, -1 for strikeout, -2 for fly out, ...) and other information concerning the AtBat. Pitching has a value for each inning during which the player pitched. Defense has a value for each inning a player played a fielding position. We can have a less detailed or a more detailed design for the performance of a player in each game, depending on how much information we need to keep in the database. Suitable variations of the ER diagram shown below can be used for other sports.



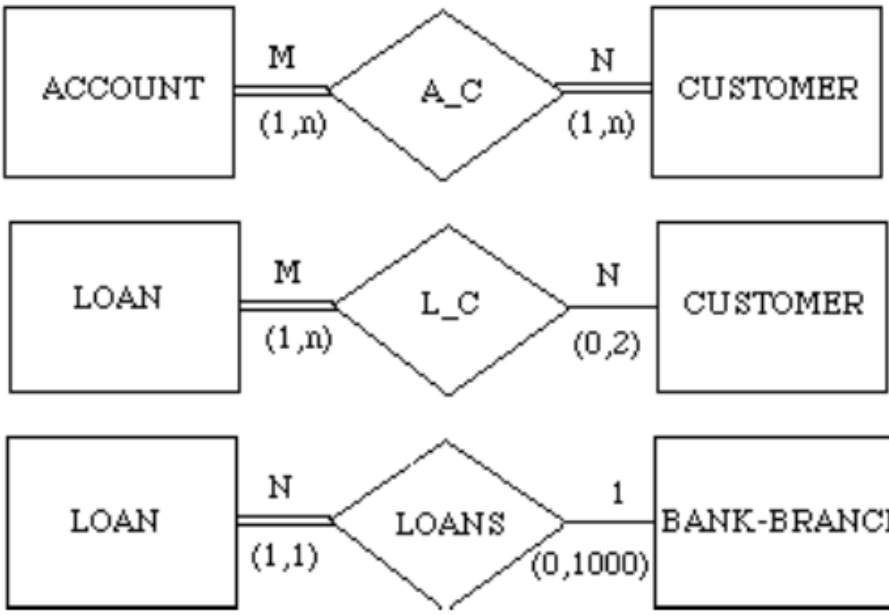
3.23 - Consider the ER diagram shown in Figure 7.21 for part of a BANK database. Each bank can have multiple branches, and each branch can have multiple accounts and loans.

(a) List the strong (nonweak) entity types in the ER diagram.

(b) Is there a weak entity type? If so, give its name, its partial key, and its identifying relationship.

(c) What constraints do the partial key and the identifying relationship of the weak entity type specify in this diagram?

(d) List the names of all relationship types, and specify the (min,max) constraint on each participation of an entity type in a relationship type. Justify your choices.



(e) List concisely the user requirements that led to this ER schema design.

(f) Suppose that every customer must have at least one account but is restricted to at most two loans at a time, and that a bank branch cannot have more than 1000 loans. How does this show up on the (min,max) constraints?

Answer:
(a) Entity types: BANK, ACCOUNT, CUSTOMER, LOAN

(b) Weak entity type: BANK-BRANCH. Partial key: BranchNo.
Identifying relationship: BRANCHES.

(c) The partial key BranchNo in BANK-BRANCH specifies that the same BranchNo value ay occur under different BANKs. The identifying relationship BRANCHES specifies that

BranchNo values are uniquely assigned for those BANK-BRANCH entities that are related to the same BANK entity. Hence, the combination of BANK Code and BranchNo together constitute a full identifier for a BANK-BRANCH.

(d) Relationship Types: BRANCHES, ACCTS, LOANS, A-C, L-C. The (min, max) constraints are shown below.

(e) The requirements may be stated as follows: Each BANK has a unique Code, as well as a Name and Address. Each BANK is related to one or more BANK-BRANCHes, and the BranhNo is unique among each set of BANK-BRANCHes that are related to the same BANK. Each BANK-BRANCH has an Address. Each BANK-BRANCH has zero or more LOANS and zero or more ACCTS. Each ACCOUNT has an AcctNo (unique), Balance, and Type and is related to exactly one BANK-BRANCH and to at least one CUSTOMER. Each LOAN has a LoanNo (unique), Amount, and Type and is related to exactly one BANK-BRANCH and to at least one CUSTOMER. Each CUSTOMER has an SSN (unique), Name, Phone, and Address, and is related to zero or more ACCOUNTs and to zero or more LOANs.

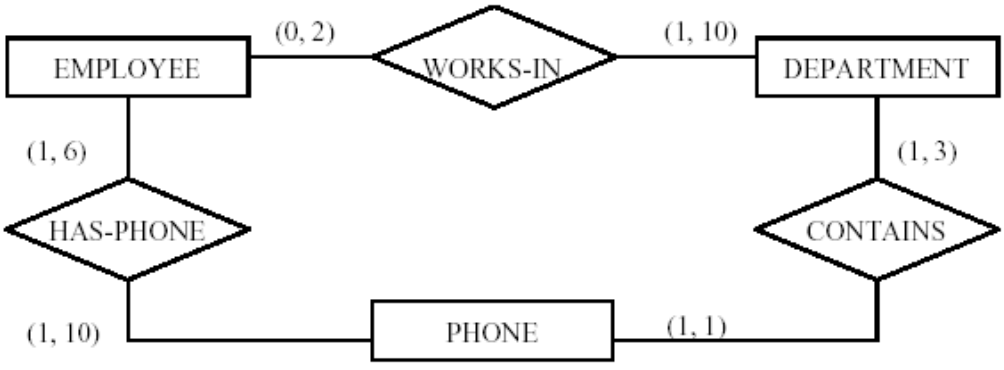
(f) The (min, max) constraints would be changed as follows:

3.24 - Consider the ER diagram in Figure 7.22. Assume that an employee may work in up to two departments or may not be assigned to any department. Assume that each department must have one and may have up to three phone numbers. Supply (min, max) constraints on this diagram. State clearly any additional assumptions you make. Under what conditions would the relationship HAS_PHONE be redundant in this example?

Answer:
Assuming the following additional assumptions:
- Each department can have anywhere between 1 and 10 employees.
- Each phone is used by one, and only one, department.
- Each phone is assigned to at least one, and may be assigned to up to 10 employees.
- Each employee is assigned at least one, but no more than 6 phones.

The resulting ER Diagram will have the (min, max) constraints shown in Figure A.

Figure A



Relationship HAS-PHONE would be redundant under the following conditions:
- Each employee is assigned all of the phones of each department that he/she works in.

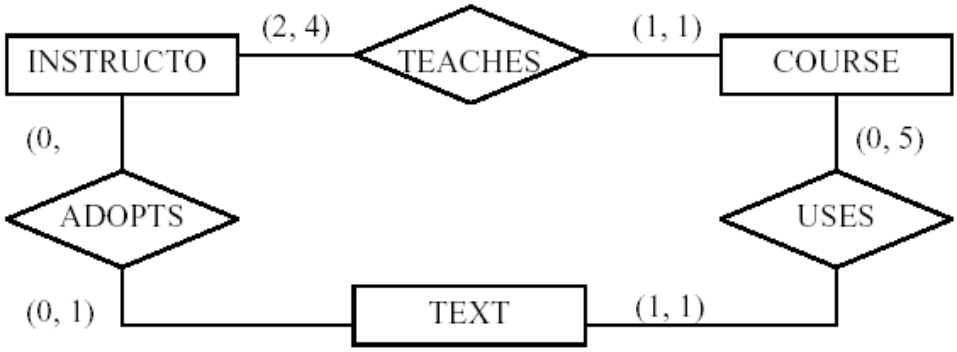
- An employee cannot have any other phones outside the departments he/she works is.
EMPLOYEE
PHONE
DEPARTMENT
HAS-PHONE CONTAINS
WORKS-IN
(0, 2)
(1, 3)
(1, 10)
(1, 10) (1, 1)
(1, 6)

3.25 - Consider the ER diagram in Figure 7.23. Assume that a course may or may not use a textbook, but that a text by definition is a book that is used in some course. A course may not use more than five books. Instructors teach from two to four courses. Supply (min, max) constraints on this diagram. State clearly any additional assumptions you make. If we add the relationship ADOPTS between INSTRUCTOR and TEXT, what (min, max) constraints would you put on it? Why?

Answer:
Assuming the following additional assumptions:
- Each course is taught by exactly one instructor.
- Each textbook is used by one and only one course.
- An instructor does not have to adopt a textbook for all courses.
- If a text exists:
- ___it is used in some course,
- ___hence it is adopted by some instructor who teaches that course.
- An instructor is considered to adopt a text if it is used in some course taught
- by that instructor.

The resulting ER Diagram will have the (min, max) constraints shown in Figure B.

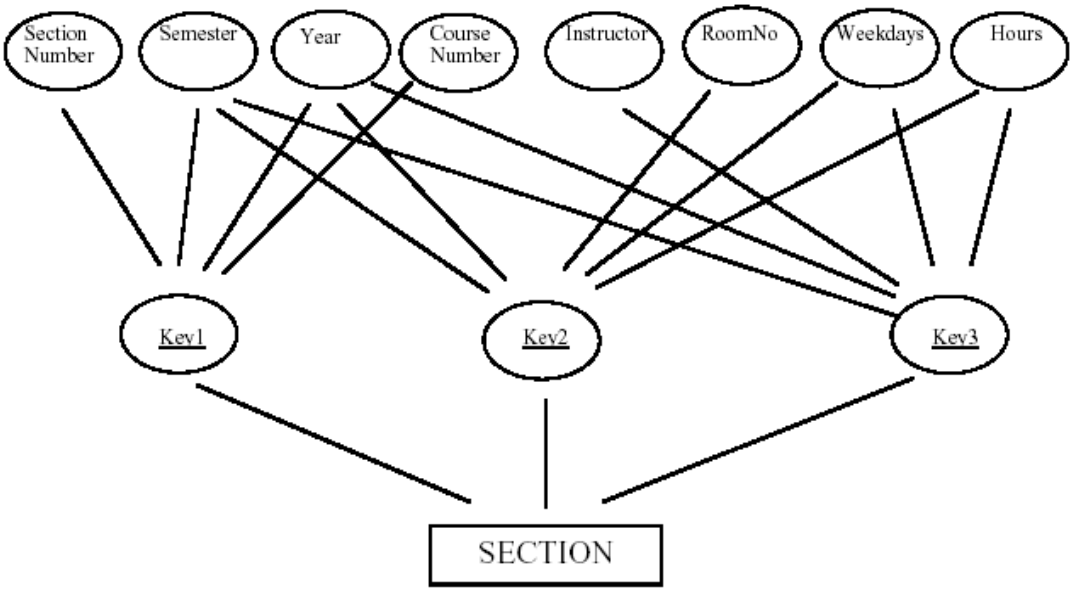
Figure B



3.26 - Consider an entity type `SECTION` in a `UNIVERSITY` database, which describes the section offerings of courses. The attributes of `SECTION` are SectionNumber, Semester, Year, CourseNumber, Instructor, RoomNo (where section is taught), Building (where section is taught), Weekdays (domain is the possible combinations of weekdays in which a section can be offered {MWF, MW, TT, etc.}). Assume tat SectionNumber is unique for each course

within a particular semester/year combination (that is, if a course if offered multiple times during a particular semester, its section offerings are numbered 1, 2, 3, etc.). There are several composite keys for SECTION, and some attribute sare components of more than one key. Identify three composite keys, and show how they can be represented in an ER schema diagram.

Answer:



3.27 - Cardinality ratios often dictate the detailed design of a database. The cardinality ratio depends on the real-world meaning of the entity types involved and is defined by the specific application. For the binary relationships below, suggest cardinality ratios based on common-sense meaning of the entity types. Clearly state any assumptions you make.

	Entity 1	Cardinality Ratio	Entity 2
1.	Student		SocialSecurityCard
2.	Student		Teacher
3.	ClassRoom		Wall
4.	Country		CurrentPresident
5.	Course		TextBook
6.	Item (that can be found in an order)		Order

7.	Student		Class
8.	Class		Instructor
9.	Instructor		Office
10.	E-bay Auction item		E-bay bid

Answer:

	Entity 1	Cardinality Ratio	Entity 2
1.	Student	1-many A student may have more than one social security card (legally with the same unique social security number), and every social security number belongs to a unique student.	SocialSecurityCard
2.	Student	Many-many Generally students are taught by many teachers and a teacher teaches many students.	Teacher
3.	ClassRoom	Many-many Don't forget that the wall is usually shared by adjacent rooms.	Wall
4.	Country	1-1 Assuming a normal country under normal circumstances having one president at a time.	CurrentPresident
5.	Course	Many-many A course may have many textbooks and a text book may be prescribed for different courses.	TextBook

6.	Item (that can be found in an order)	Many-many Assuming the same item can appear in different orders.	Order
7.	Student	Many-many One student may take several classes. Every class usually has several students.	Class
8.	Class	Many-to-1 Assuming that every class has a unique instructor. In case instructors were allowed to team teach, this will be many-many.	Instructor
9.	Instructor	1-1 Assuming every instructor has only one office and it is not shared. In case of offices shared by 2 instructors, the relationship will be 2-1. Conversely, if an instructor has a joint appointment (in two departments) and offices in both departments, then the relationship will be 1-2. In a very general case, it may be many-many.	Office
10.	E-bay Auction item	1-many 1 item has many bids and a bid is unique to an item (assuming a regular auction format).	E-bay bid

3.28-3.29 – No solutions provided.

3.30 - Illustrate the UML Diagram for exercise 7.16. Your UML design should observe the following requirements:

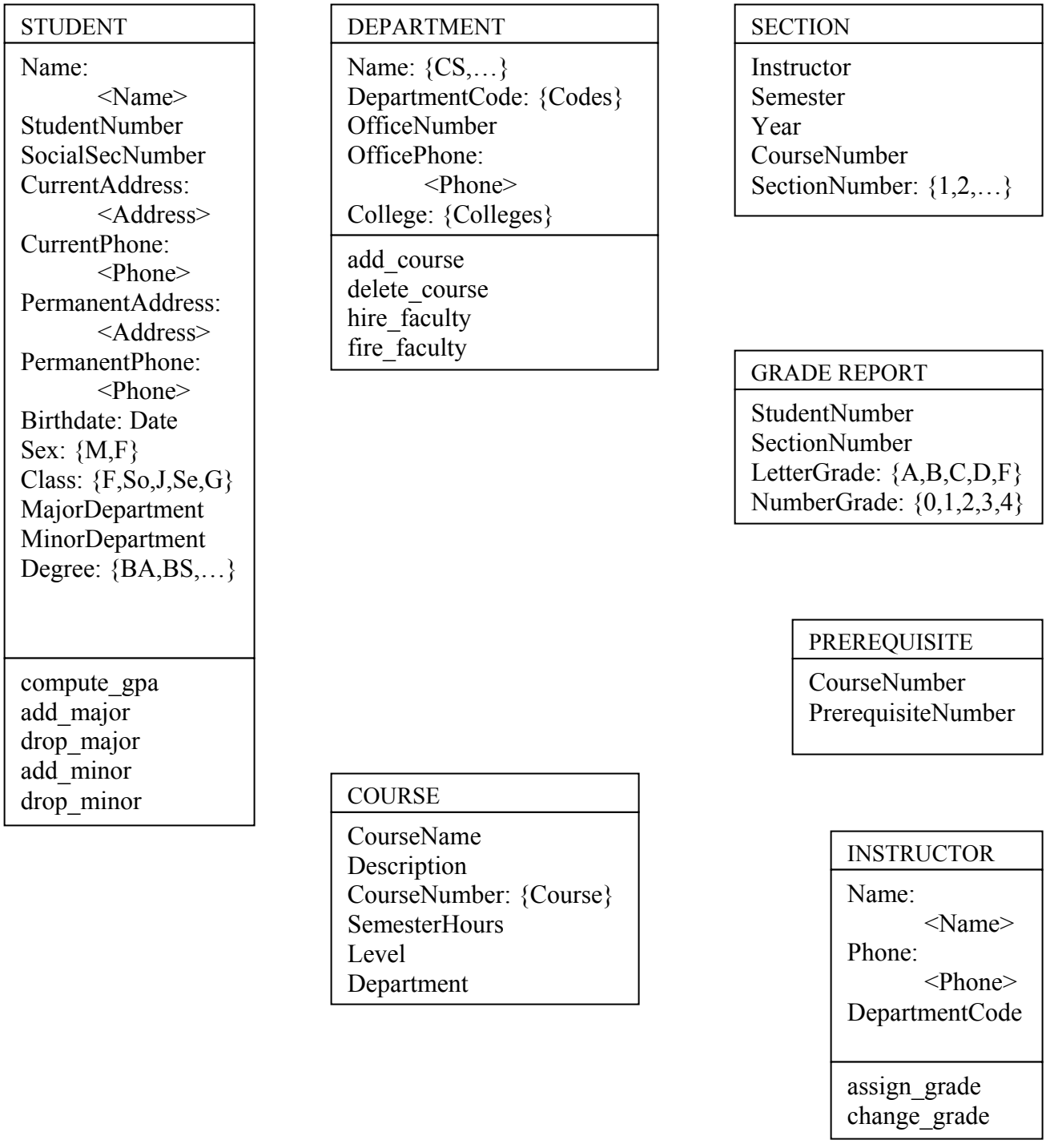
a. The student should have the ability to compute his/her GPA and add or drop majors and minors.

b. Each department should be to able add or delete courses and hire or terminate faculty.

c. Each instructor should be able to assign or change a grade to a student for a course.

Note: Some of these functions may be spread over multiple classes.

Answer:



This one isn't finished yet. The relationships need to be added and some of the functions should be moved to span classes.