**Data Management and Database Design**

**INFO 6210**

**Fall 2016**

**Assignment \_4**

**Submission Date – Oct/12/2016 Wednesday**

**Student Name: Malick Fairoz Sayeed Abuthahir**

**NUID: 001235450**

**Program: MS in Information Systems**

**Professor Name: Yusuf Ozbek**

**College: College Of Engineering**

**University: Northeastern University**

**PART – 1**

4.

a) Can’t tell

b) City A

c) Can’t tell

5.

Activity

Student

**Student**

**Student ID**

Name

Address

Age

Activity

No of Years

The student ID will be the identifier which is the primary key as it is unique for every student.

6.

The entity which depends on another entity is known week entity. Associate entities require attributes that are special to the relationship between two entities.

7.

The associative entities in the figure are:

a) Does business in

b) Order line

c) Produced in

d) Uses

e) Works in

f) Has skill

g) Supplies

8.

Student Id, Course Id and Instructor Name are identifiers for entities STUDENT, COURSE and INSTRUCTOR respectively

Student

**Student ID**

Student Name

Address

Major

Instructor

**Name**

Location

Course

**Course ID**

Title

Grade

9.

a)

Dependent

DependentName

Date of Birth

Employee

Employee **ID**

Employee Name

b)

Course

Course ID

Course Title

Employee

Employee **ID**

Employee Name

Date Of Birth

c)

Certificate

Certificate Number

Date Completed

Employee

Employee **ID**

Employee Name

Date Of Birth

Course

Course ID

Course Title

d) Is married to Manages

Employee

Person

One to one One to many

Stands after

Team

One to one

Is Assigned Contains

Product

Parking Space

Product Line

Employee

One to One One to Many

Registers For

Course

Student

Many to Many

Parts

Warehouse

Vendor

e)

Has Components

Bottom Structure

Effective Date

Quantity

Item

Used In Assemblies

f)

Parts

Warehouse

Supply Schedule

Shipping mode

Unit Cost

Vendor

10)

a)

Is Married to

Person

Person ID

Person Name

b)

is Married to

Person

Person ID

Person Name

c)

is Married to

Person

Person ID

Person Name

Date

Marriage

Dissolution

d)

Re marriage date

Date

Marriage

Dissolution

Person

Person ID

Person Name

Re marriage

e) Since in section c we have already mentioned in the ER diagram that many person and marry many so in the section e also same is applicable

is Married to

Person

Person ID

Person Name

Date

Marriage

Dissolution

**PART - 2**

**CHAPTER - 4**

1)

|  |  |
| --- | --- |
| Convert | A user who originally was against your project that you include in the development process to bring them onto your side. |
| Customer Champion | Thoroughly understands the customers’ needs. Has the authority to make decisions that stick. |
| Customer Representative | Answers your questions about the project. |
| Devil’s Advocate | Provides a reality check and prevents groupthink. |
| Executive Champion | The highest ranking customer driving the project. Willing to fight super villains. |
| Generic Bad Guy | Ranges from annoying naysayer to malicious saboteur/super villain. |
| Short-Timer | Someone who won't be around for long. May be helpful or may not care all that much. |
| Sidekick/Gopher | Makes things generally run smoothly. Not glamorous but very useful. |
| Stakeholder | Anyone who has an interest in the project. |

2) Ans – C

3) Ans – D

4) Ans – B

5) Ans – A

6)

| Field | Required? | Domain |
| --- | --- | --- |
| Address one | Yes | Valid street addresses or street names without numbers. |
| Address two | No | Apartment, suite, floor, etc. |
| City | Yes | Valid cities. |
| State | Yes | Valid states. |
| ZIP Code | No | Five digit or ZIP+4 codes as in 12345 and 12345-6789. |

7) Ans – C

8) Ans – D

9)

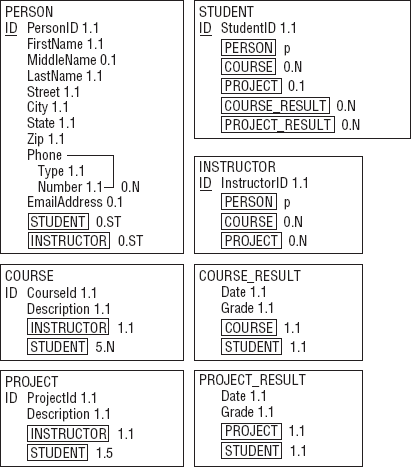
* + The user—tries to log in.
  + The operating system—validates the user Id and password and grants or revoke access.

10)

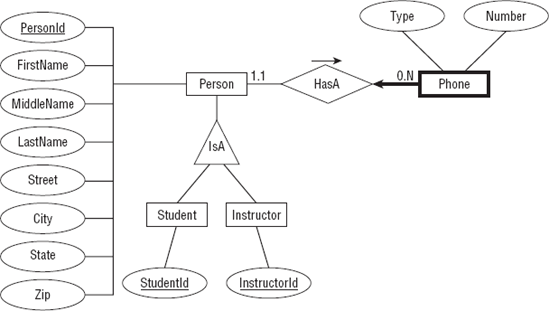
In this kind of situation, you need to call your executive Champion. Ideally he can point to your requirements document and show that you did, in fact, consider farbulistic granilation and that everyone agreed the allowance was sufficient. If you are not able to convince then you need to study some extra to provide your executive Champion ammunition. If your executive champion doesn't have enough clout to fight off the Super Villain, you could be in trouble. Executive Champion and Customer Champion spent a huge amount of time fending off attacks for about two years before the project finished.

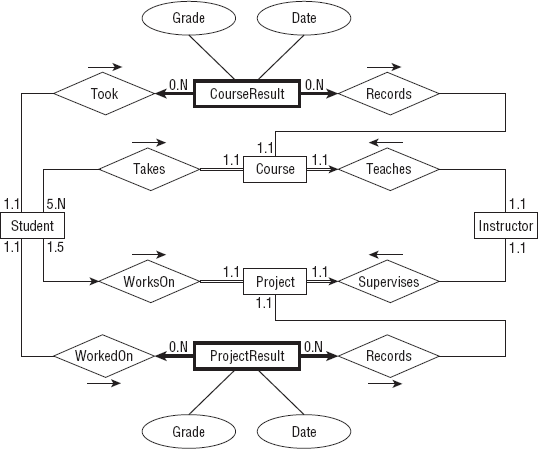
**CHAPTER - 5**

1)



2)





The Student entity's relationships with Course and Project do not indicate that a Student must be involved with at least one Course or a Project.

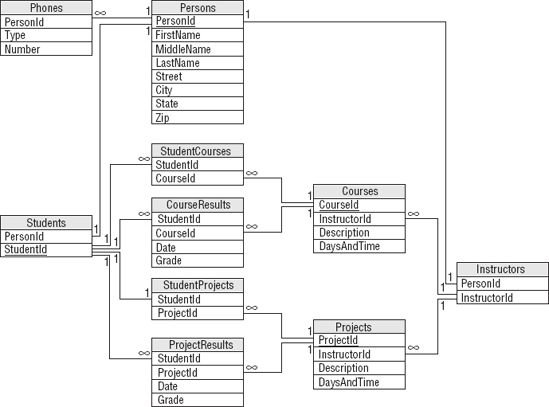
A Course must be involved in a relationship with a Student (or else the Course is canceled) so its line leading toward Student is double (a participation constraint).

A Student can work on at most one Project at a time so its line leading to Project is an arrow (key constraint).

3)

Notice the way this model handles the fact that Student and Instructor inherit from Person. The Persons table holds the basic Person information and a PersonId. The Students and Instructors tables include PersonId foreign keys to link to the corresponding basic Person data.

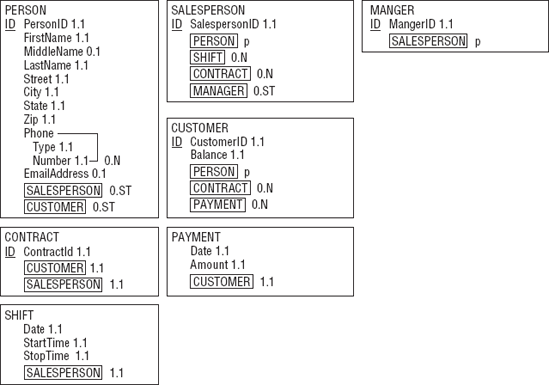
Note also the different approach used for the Student/Course and Instructor/Course relationships. Because a course has exactly one instructor, the Instructors and Courses tables are connected with a simple one-to-many relationship. In contrast, a course has many students so the relationship uses an intermediate StudentCourses table to connect the two to build a many-to-many relationship. (The same reasoning applies to the Student/Project and Instructor/Project relationships.)

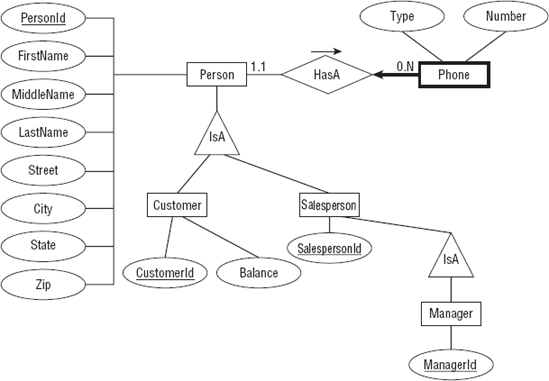


4)

The semantic object model actually does a pretty good job of capturing the Mike's Trikes data. About the only item that isn't described explicitly is the manager's role. In this model, you can deduce the manager at any given time by examining the manager's shift data. If Mike needed a more explicit record of who is managing during a salesperson's shift or when a contract was sold, the model would need to be modified.

5)



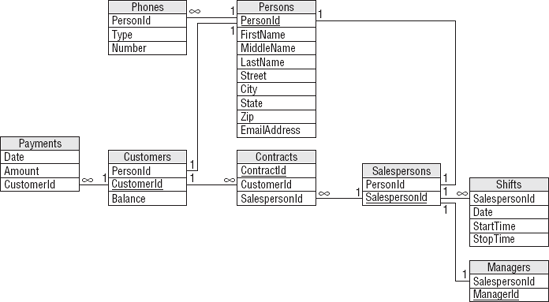


Pay is a weak entity because you look up payments via the Customer who made them. Payment is drawn with a bold rectangle and a thick arrow pointing toward its identifying relationship.

Shifting is also is a weak entity because you look up shift data via the Salesperson who works them. Shift is taken with a thick rectangle and a bold arrow pointing toward its identifying relationship.

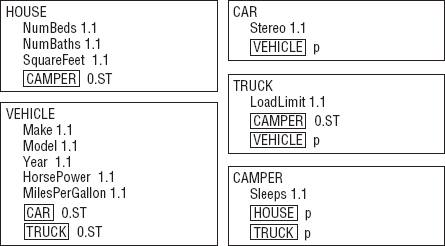
A Contractor must have exactly one cust and exactly one sale person so the lines leading out of Contract toward those other entities are double and key constrains.

6)

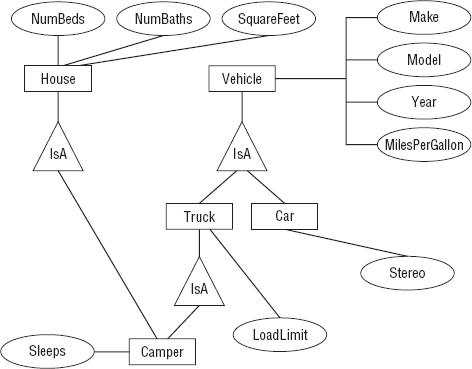


Notice how this model builds the inheritance hierarchy. The Customers and Salespersons tables use PersonId foreign key fields to link to their corresponding records. As usual, the design doesn't capture all of the content available about the situation. In particular, it doesn't indicate that a Customers record must be associated with at least one Contracts record. You should write down this and other facts such as data types and domain info in separate file.

7)



8)



**CHAPTER – 6**

1)

| Field | Required | Data Type | Domain | Sanity Checks |
| --- | --- | --- | --- | --- |
| PersonId | Yes | ID | Persons.PersonId |  |
| Type | Yes | String | List: Cell, Home, Fax |  |
| Number | Yes | String | Phone numbers |  |
| Field | Required | Data Type | Domain | Sanity Checks |
| PersonId | Yes | ID | Any ID |  |
| FirstName | Yes | String | Any string |  |
| MiddleName | No | String | Any string |  |
| LastName | Yes | String | Any string |  |
| Street | Yes | String | Any string |  |
| City | Yes | String | Any string |  |
| State | Yes | String | List: (states) |  |
| Zip | Yes | String | ZIP or ZIP+4 format | Verify ZIP or ZIP+4 format |
| EmailAddress | No | String | Valid email address | Contains one @ symbol |
| MedicalNotes | ? | String | Any string |  |
| IceQualified? | ? | Yes/No | Yes or No |  |
| RockQualified? | ? | Yes/No | Yes or No |  |
| JumpQualified? | ? | Yes/No | Yes or No |  |

| Field | Required | Data Type | Domain | Sanity Checks |
| --- | --- | --- | --- | --- |
| PersonId | Yes | ID | Persons.PersonId |  |
| GuideId | Yes | ID | Any ID |  |
| IceInstructor? | Yes | Yes/No | Yes or No |  |
| RockInstructor? | Yes | Yes/No | Yes or No |  |
| JumpInstructor? | Yes | Yes/No | Yes or No |  |
| Field | Required | Data Type | Domain | Sanity Checks |
| PersonId | Yes | ID | Persons.PersonId |  |
| ExplorerId | Yes | ID | Any ID |  |
| Field | Required | Data Type | Domain | Sanity Checks |
| PersonId | Yes | ID | Persons.PersonId |  |
| OrganizerId | Yes | ID | Any ID |  |
| Field | Required | Data Type | Domain | Sanity Checks |
| AdventureId | Yes | ID | Any ID |  |
| ExplorerId | Yes | ID | Explorers.ExplorerId |  |
| EmergencyContact | Yes | ID | Persons.PersonId |  |
| OrganizerId | Yes | ID | Organizers.OrganizerId |  |
| TrekId | Yes | ID | Treks.TrekId |  |
| DateSold | Yes | Date | Any date | Before the trek's start date. Between January 1, 2000 and December 31, 2050 (or some other very early and late dates). |
| IncludeAir? | Yes | Yes/No | Yes or No |  |
| IncludeEquipment? | Yes | Yes/No | Yes or No |  |
| TotalPrice | Yes | Currency | Monetary amount > $0 | Price > $250 (or some minimum sane value). |
| Notes | ? | Yes/No | Yes or No |  |

2)

Type: Verify that the type is one of Home, Work, Cell, or Fax. Alternatively if you think this list might change in the future, you could put these values in a lookup table.

Number: Verify that the value has a valid phone number format. In the United States, you would probably want to verify that it is a 10-digit number of the format ??? ???? ???? and you should allow for an extension.

Table Verify that the trek has room for this explorer.

Verify that the explorer's IceQualified?, RockQualified?, and JumpQualified? values include those required for this trek.

ExplorerId/TrekId: Verify that this combination is unique. An explorer should not buy the same trek twice. (We're assuming that the same trip on different dates gets a different record in the Treks table. Some people may very well want to go to the same places again.)

EmergencyContact: Verify that the EmergencyContact is not going on the same trek listed for this Adventures record.

IncludeAir?/Notes: If IncludeAir? is Yes, the Notes field should include flight information such as the explorer's starting airport and meal preferences. The database can probably not verify that the notes make sense (who knows if the low sodium meal is available on that flight?) but it can verify that the Notes entry has some minimum length if IncludeAir? is Yes.

3)

The fact that one of the company's owners asked which calculation would give the customer the biggest discount if they both purchase airline tickets and rent equipment (adding the two discounts and take 15% off gives the biggest discount) further implies that they might someday change the way they perform this calculation. That gives you more reason to extract this rule from the database so it's easier to change later.

4)

| Name | Value | Purpose |
| --- | --- | --- |
| MinimumDate | January 1, 2000 | Sanity check date for DateSold, StartDate, and EndDate. |
| Maximum Date | December 31, 2000 | Sanity check date for DateSold, StartDate, and EndDate. |
| Minimum TotalPrice | $250 | Sanity check price for an Adventure's TotalPrice. |
| Minimum TrekPrice | $250 | Sanity check price for a Trek's Price. |
| Minimum PricePerDay | $100 | Sanity check minimum price per day for a Trek's Price. |
| Maximum Explorers | 20 | Sanity check maximum number of explorers on a trek. |

**PART – 3**

**CHAPTER - 4**

When seeing from left to right

Author has the optionality 1, since an author can have one book or more than one book; moreover author cannot be assign to zero books.

When seeing from right to left

Book has optionality 1, since a book can have one author or more than one author, moreover book should have atleast one author.

It is a many to many concept.

Author Book

Book1

Author 1

Book2

Auhor 2

Author 3

Book3

**Book**

Book Name

No of Authors written

Author names

**Author**

Author Name

No of Books written

Book names

1 1

In this data model for cocktail recipes, the cocktail will be using one or more ingredient and similarly ingredient can be used in one or more cocktails so it is an many to many relationship and there might be some missing item also.

uses 1…n 1…n

**Ingredient**

Ingredient name

Amount used

Used in Cocktail names

**Cocktail**

Cocktail Name

Brand

Price

3.

In this data model, the guest has to have one room in the hostel but the room in the hostel may or may not guest, so there may be empty rooms without guest.

To maintain the historical information about the room occupancy, the data model should hold an attribute to maintain the previous data on the room like who stayed in that room before the present guest.

Occupies 0..1 1…1

**Room**

Room number

Guest assigned

Date of occupancy

Date of leaving

Previous guest name

**Guest**

Name

Room assigned

Guest ID

**CHAPTER – 5**

1..1 n..1

**Location**

Location Name

**Manager**

Manager Name

Manager ID

1..1 n..1

This data model refers the teaching course were the one lecture can be taken by many staff but in that one staff will be assigned as supervisor. The three classes are staff, supervisor and lecture

**Staff**

Staff name

Staff ID

Position

**Lecture**

Course name

Course ID

Time schedule

Teaches

Supervisor

**Supervisor**

Supervisor name

Supervisor ID

Course Name

Teachers Provides Assign

**Marriage**

Date

Location

Getting Married Getting Married

Marrying

**Male**

Name

Location

**Female**

Name

Location

Same Gender Marriage Same Gender Marriage

**Player**

Name

Team

Contact

**Work**

Name

Composer

Date

0..1

Involved

1..n

Performed Paid

0..n

**Concert**

Location

Time

Date

**Payment**

Payment ID

Amount

Receipt Number

Price