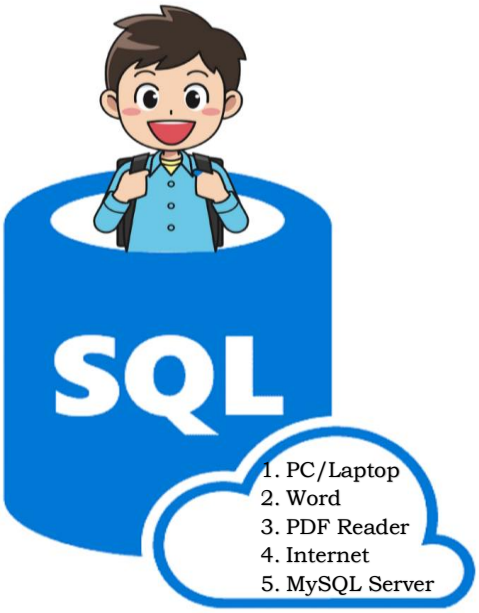
[Laboratory No. 3: ERD & Business Rules]

Objectives

1. To identify the relationship between ERD and business rules
2. To know how to create business rules and design an ERD

Materials



Background

Although all of the topics covered in this chapter are important, our students have given us consistent feedback: If you can write precise business rules from a description of operations, database design is not that difficult. Therefore, once data modeling (Sections 2-1, "Data Modeling and Data Models," Section 2-2, "The Importance of Data Models," and 2-3, "Data Model Basic Building Blocks") has been examined in detail, Section 2-4, "Business Rules," should receive a lot of class time and attention. Perhaps it is useful to argue that the answers to questions 2 and 3 in the Review Questions section are the key to successful design. That's why we have found it particularly important to focus on business rules and their impact on the database design process.

What are business rules, their source, and why are they crucial?

Business rules are written and unambiguous statements from a detailed description of an organization's operations. When written correctly, business rules define one or more of the following modeling components:

- Entities
- Relationships
- Attributes
- Connectivities
- cardinalities – these will be examined in detail in ***Chapter 3, "The Relational Database Model."*** The cardinalities yield the minimum and the maximum number of entity occurrences in an entity.

  For example, the relationship described by ***"a professor teaches one or more classes"*** means that the **PROFESSOR** entity is referenced at least once and no more than four times in the **CLASS** entity constraints

Because the business rules form the basis of the data modeling process, their precise statement is crucial to the success of the database design. And, because the business rules are derived from a detailed description of operations, much of the design's success depends on the accuracy of the definition of operations.

Examples of business rules are:

- An invoice contains one or more invoice lines.
- Each invoice line is associated with a single invoice.
- A store employs many employees.
- Each employee is employed by only one store.
- A college has many departments.
- Each department belongs to a single college. (This business rule reflects a university with multiple colleges such as Business, Liberal Arts, Education, Engineering, etc.)
- A driver may be assigned to drive many different vehicles.
- Each vehicle can be driven by many drivers. (Note: Keep in mind that this business rule reflects the assignment of drivers during some period of time.)
- A client may sign many contracts.
- Each contract is signed by only one client.
- A sales representative may write many contracts.
- Each contract is written by one sales representative.

Note that each relationship definition requires the definition of two business rules. For example, the relationship between the INVOICE and (invoice) LINE entities is defined by the first two business rules in the bulleted list. This two-way requirement exists because there is always a two-way relationship between related entities. (This two-way relationship description also reflects the implementation by many of the available database design tools.)

Keep in mind that the ER diagrams cannot always reflect all of the business rules. For example, examine the following business rule:

> *A customer cannot be given a credit line over $10,000 unless that customer has maintained a satisfactory credit history (as determined by the credit manager) during the past two years.*

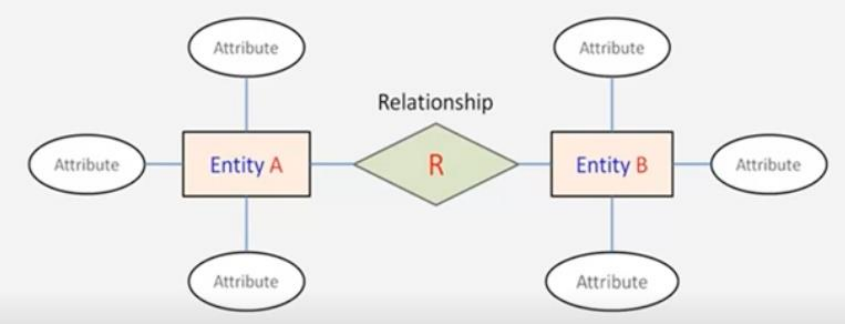
This business rule describes a constraint that cannot be shown in the ER diagram. The business rule reflected in this constraint would be handled at the application software level using a trigger or a stored procedure. (Your students will learn about triggers and stored procedures in Chapter 8, "Advanced SQL.")

> Given their importance to successful design, we cannot overstate the importance of business rules and their derivation from properly written descriptions of operations. It is not too early to ask students to write business rules for simple descriptions of operations. Begin by using familiar operational scenarios, such as buying a book at the bookstore, registering for a class, paying a parking ticket, or renting a DVD.

What are Entity Relationship Diagrams?

ER diagram is widely used in database design

- Represent the conceptual level of a database system
- Describe things and their relationships at high level
- illustrate the logical structure of databases.



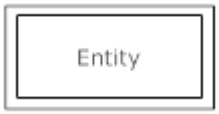
An ER Diagram

Entity Relationship Diagram Notations

Peter Chen developed ERDs in 1976. Since then, Charles Bachman and James Martin have added some slight refinements to the basic ERD principles.

Entity - An entity is an object in the world that can be distinguished from other objects. It is an object or concept about which you want to store information.

Weak Entity - A weak entity is an entity that must be defined by a foreign key relationship with another entity, as it cannot be uniquely identified by its own attributes alone.



Attributes: Used to describe entities

- All entities in the set have the same attributes
- A minimal set of attributes that uniquely identify an entity is called a key
- An attribute contains a single piece of information (and not a list of data)

Key attribute - A key attribute is the unique, distinguishing characteristic of the entity. For example, an employee's social security number might be the employee's key attribute.

Multivalued attribute - A multivalued attribute can have more than one value. For example, an employee entity can have multiple skill values.

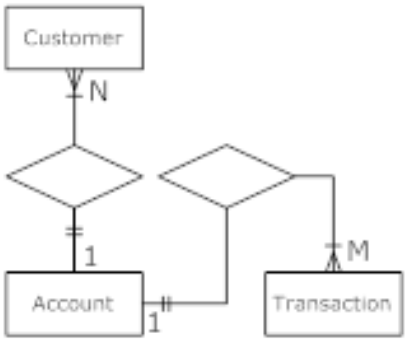
Prepared by: Martzel P. Baste

Derived attribute - A derived attribute is based on another attribute. For example, an employee's monthly salary is based on the employee's annual salary.

Relationships- Relationships illustrate how two entities share information in the database structure.

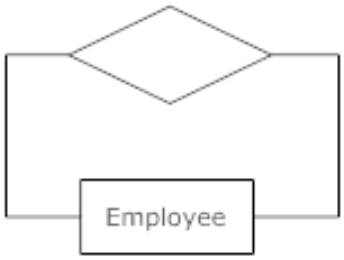
Cardinality - Cardinality specifies how many instances of an entity relate to one instance of another entity.

Ordinality is also closely linked to cardinality. While cardinality specifies the occurrences of a relationship, ordinality describes the relationship as either mandatory or optional. In other words, cardinality specifies the maximum number of relationships, and ordinality specifies the absolute minimum number of relationships.



Recursive relationship

In some cases, entities can be self-linked. For example, employees can supervise other employees.



Cardinality Notations

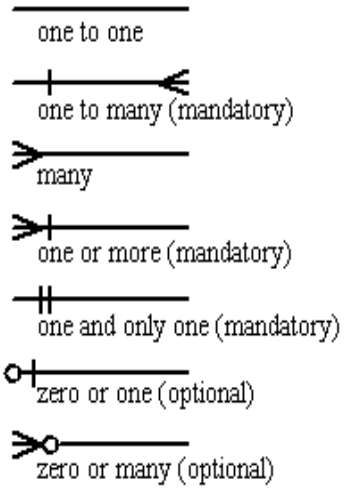
Cardinality specifies how many instances of an entity relate to one instance of another entity.

Ordinality is also closely linked to cardinality. While cardinality specifies the occurrences of a relationship, ordinality describes the relationship as either mandatory or optional. In other words, cardinality specifies the maximum number of relationships, and ordinality specifies the absolute minimum number of relationships. When the minimum number is zero, the association is usually called optional; when the minimum number is one or more, the relationship is usually called mandatory.

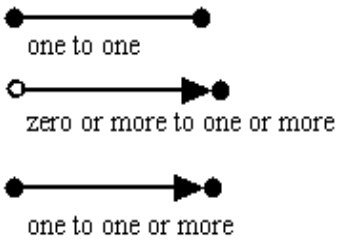
Prepared by: Martzel P. Baste

Information Engineering

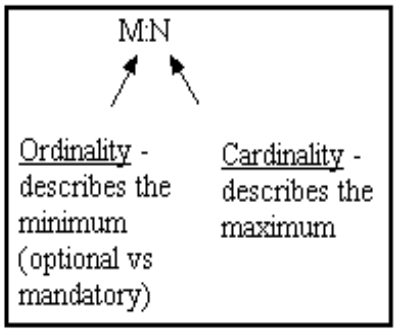
Information Engineering style



Bachman style



Chen style



1:N (n=0,1,2,3...)
one to zero or more

M:N (m and n=0,1,2,3...)
zero or more to zero or more (many to many)

1:1
one to one

Martin style

1 - one, and only one (mandatory)

* - many (zero or more - optional)

1...* - one or more (mandatory)

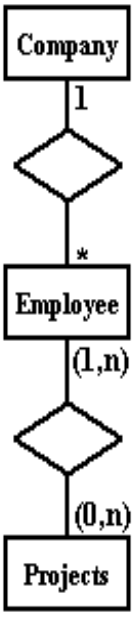
0...1 - zero or one (optional)

(0,1) - zero or one (optional)

(1,n) -one or more (mandatory)

(0,n) - zero or more (optional)

(1,1) - one and only one (mandatory)



Prepared by: Martzel P. Baste

Tips for Effective ER Diagrams

1) Make sure that each entity only appears once per diagram.
2) Name every entity, relationship, and attribute on your diagram.
3) Examine relationships between entities closely. Are they necessary? Are there any relationships missing? Eliminate any redundant relationships. Don't connect relationships to each other.

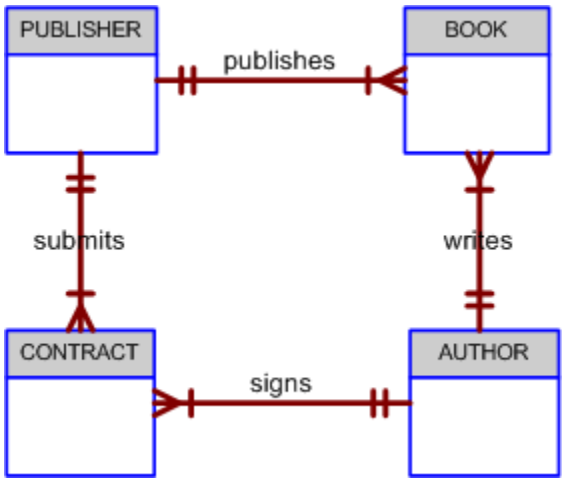
Procedure

Answer the following questions:

1. How do you translate business rules into data model components?

2. Design a Data Model based on the business rules written as follows:
   a. A professor can teach many classes.
   b. One professor teaches each class.
   c. A professor can advise many students.
   d. One professor advises each student.

3. Write the business rules reflected in the Data Model (ERD) below.



4. The local city youth league needs a database system to help track children that sign up to play soccer. Data needs to be kept on each Team and the children playing on each Team, and their parents. Also, data needs to be kept on the coaches for each Team. Draw the data model described below.

   Entities required: Team, Player, Coach, and Parent.

   Attributes required:
   - **Team** : Team ID number, Team name, and Team colors.
   - **Player** : Player ID number, Player first name, Player last name, and Player age.
   - **Coach** : Coach ID number, Coach first name, Coach last name, and Coach home phone number.
   - **Parent** : Parent ID number, Parent last name, Parent first name, Home phone number, and Home Address (Street, City, State, and ZIP Code).

Prepare by: ***Martzel P. Baste***

The following relationships must be defined:

- Team is related to Player.
- Team is related to Coach.
- Player is related to Parent.

Connectivities and participation are defined as follows:

- A Team may or may not have a Player.
- A Player must have a Team.
- A Team may have many Players.
- A Player has only one Team.
- A Team may or may not have a Coach.
- A Coach must have a Team.
- A Team may have many Coaches.
- A Coach has only one Team.
- A Player must have a Parent.
- A Parent must have a Player.
- A Player may have many Parents.
- A Parent may have many Players.

5. Make an initial business rule of your chosen company. Afterward, design an ERD.

Insights

Write your takeaways in the blank provided.

Scoring System

Task	Total Points	Score
Task 1	**10**	
Task 2	**20**	
Task 3	**10**	
Task 4	**30**	
Task 5	**20**	
Insights	**10**	