Data Modeling and Database Design

Chapter 3:
Entity-Relationship Modeling

Application of ER Modeling Grammar to the Conceptual Modeling Process

The ER model for conceptual modeling serves two major purposes

- As a communication/presentation device used by an analyst to interact with the end-user community (i.e., the Presentation Layer ER Model/Schema)
- As a design tool at the highest level of abstraction to convey a deeper level of understanding to the database designer (i.e., the Design-Specific ER Model/Schema)

The ER Model

The ER model includes:

- An ER diagram (ERD) portraying entity types, attributes for each entity type, and relationships among entity types
- Semantic integrity constraints that reflect the business rules about data not captured in the ER diagram.

The ER Model (continued)

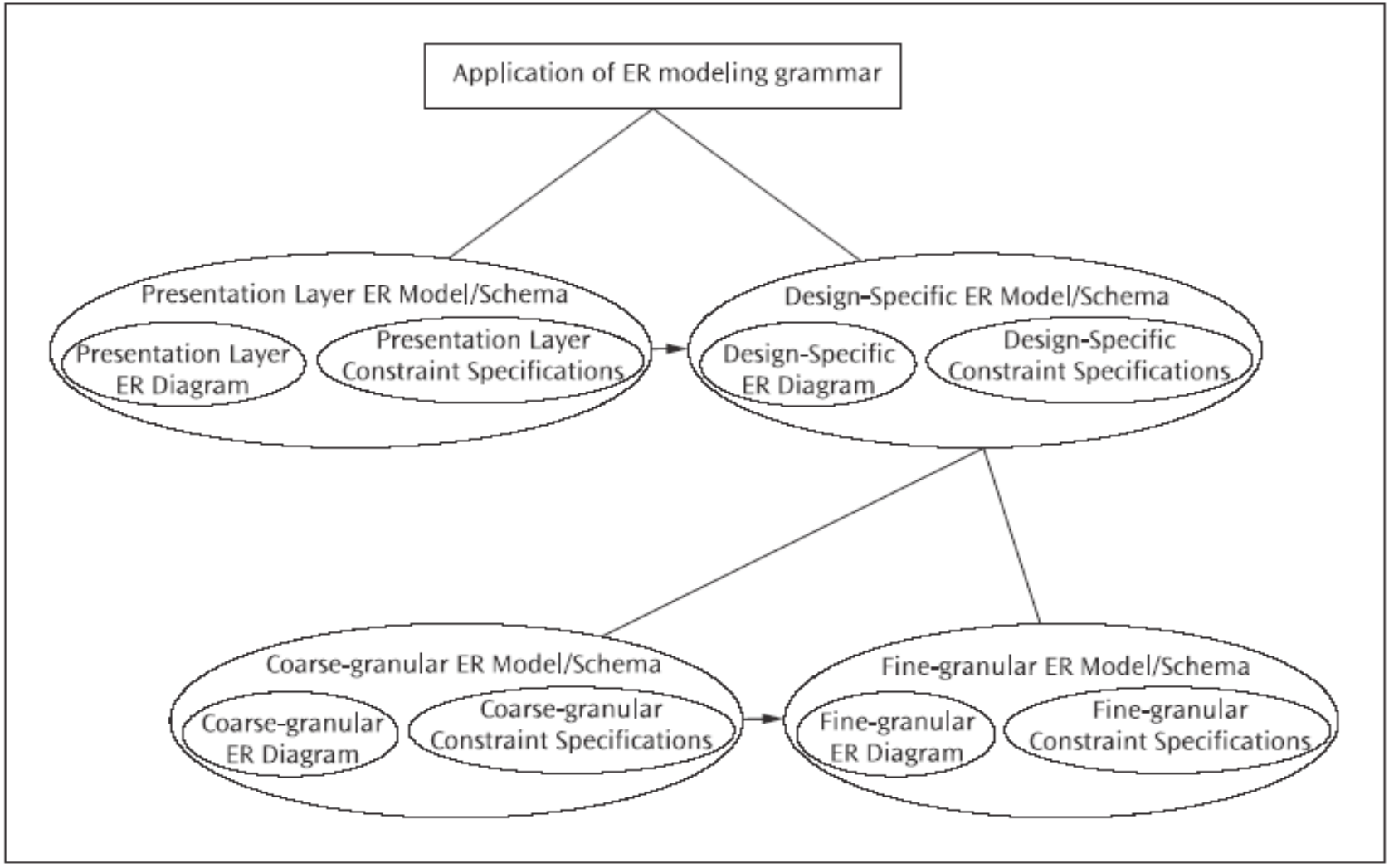


Figure 3.1 Conceptual modeling method using the ER modeling grammar

Notation

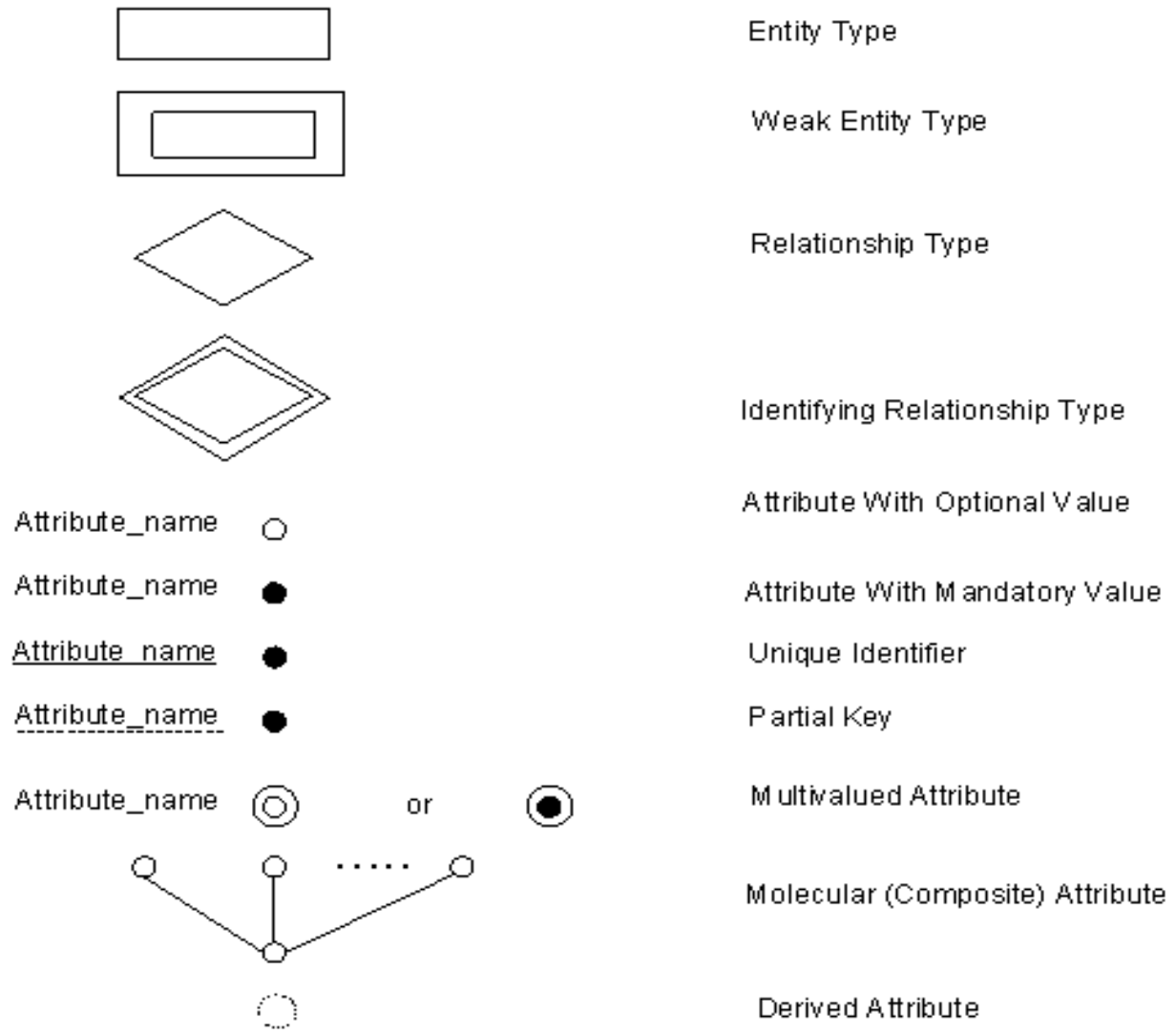
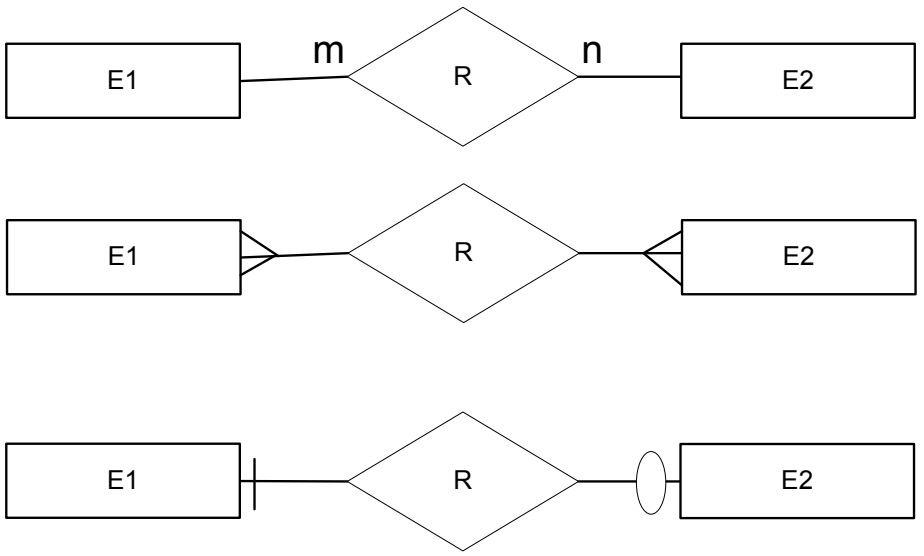


Figure 3.2 Summary of Presentation Layer ER diagram notation

Notation (continued)

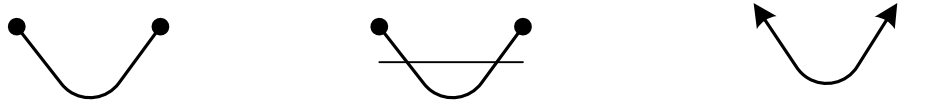


Cardinality Ratio of m:n
Chen's Notation

Cardinality Ratio of m:n
Crow's Foot Notation

E1 Optionally Related To E2
(Partial Participation of E1 in R)

E2 Mandatorily Related to E1
(Total Participation of E2 in R)
[Existence Dependency of E2 in R]



Inclusive Arc

Noninclusive Arc

Exclusive Arc

Figure 3.2 Summary of Presentation Layer ER diagram notation (continued)

Presentation Layer ER Model (An End-User Communication/Presentation Device)

- Not a Science !
- Art? Engineering? - Artistic engineering or Engineered art !
- Heuristic (Intuitive), iterative process
- Recursive incremental refinement
- Target Audience: End-user community
- Technology-independent

The Synthesis Approach to Generate an Initial Set of Entity Types and Attributes

- List all discernable data elements and treat them as attributes
- Group these attributes based on apparent commonalities
- Designate each cluster of attributes as an entity type
- Review leftover data elements and investigate the possibility that some of them serve as links among the entity types previously identified
- Designate these links as relationship types

The Analysis Approach to Generate an Initial Set of Entity Types and Attributes

- Begin by searching for things that can be labeled by singular nouns and call these things entity types
- Gather properties that appear to belong to individual entity types and label them as attributes of a particular entity type
- Be sensitive throughout the process to the identification of relationships among the various entity types

Bearcat, Incorporated
Initial Set of Entity Types

- Read through the vignette on pages 75-77
- Identify entity types
  - Plant - Company ?
  - Project - Manager ?
  - Employee - Supervisor ?
  - Dependent
  - Credit Union Account
  - Hobby

Bearcat, Incorporated
Revised Set of Entity Types

- Read through the vignette on pages 75-77
- Identify entity types
  - Plant
  - Project
  - Employee
  - Dependent
  - Credit Union Account
  - Hobby

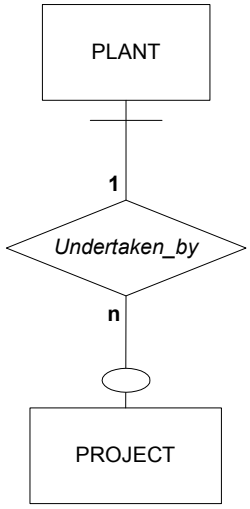
Development of Presentation Layer ER Model

- Divide story into meaningful "chunks" and identify relationship(s) among entity types
- Version 0 of ER diagram results: a punctuated equilibrium
- Go back and read the story (strike out story lines already captured if need be) and refine entity types and relationship types of the ER diagram based on additional facts discerned
- Gather business rules that can't be captured in the ERD into a list of Semantic Integrity Constraints, a supplement to the ERD
- Repeat last two steps (as many times as necessary) until all aspects of the story (i.e., the business rules) have been accounted for in the ER model

Development of Presentation Layer ER Model (continued)

Box 1

Bearcat Incorporated is a manufacturing company that has several plants in the northeastern part of the United States. *These plants are responsible for leading different projects that the company might undertake, depending on a plants' function. A certain plant might even be associated with several projects but a project is always under the control of just one plant. Some plants do not undertake any projects at all.* If a plant is closed down, the projects undertaken by that plant cannot be canceled. The project assignments from a closed plant must be temporarily removed in order to allow the project to be transferred to another plant.

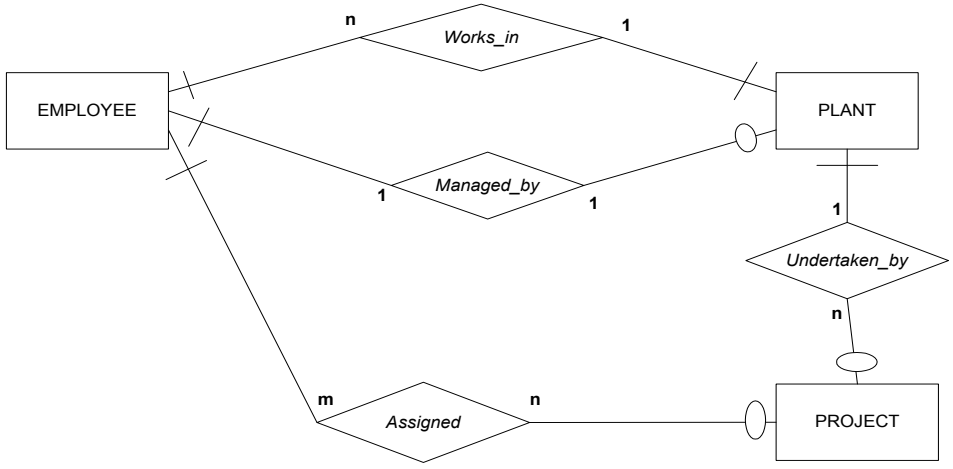


Note: The relationship shown models the italicized text

Development of Presentation Layer ER Model (continued)

Box 2

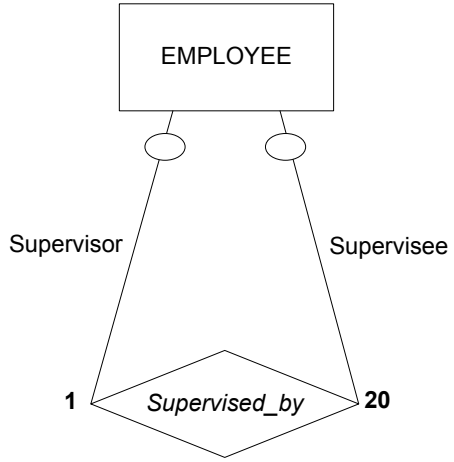
Employees work in these plants and each employee works in only one plant. A plant could employ many employees but must have at least 100 employees. A plant with employees cannot be closed down. *Every plant is managed by an employee who works in the same plant; but every employee is not a plant manager nor can an employee manage more than one plant. Company policy dictates that every plant must have a manager.* Therefore, an employee currently managing a plant cannot be deleted from the database. If a plant is closed down, the employee no longer manages the plant but becomes an employee of another plant. Some employees *are assigned to work on projects and in some cases might even be assigned to work on several projects simultaneously. For a project to exist, it must have at least one employee assigned to it. Projects might need several employees depending on their size and scope.* As long as an employee is assigned to a project, his or her record cannot be removed from the database. However, once a project ends it is removed from the system and all assignments of employees to that project must be removed.



Development of Presentation Layer ER Model (continued)

Box 3

Some of the employees also supervise other employees but all employees need not be supervised – the employees that are supervised, are supervised by just one employee. An employee can be a supervisor of several employees, but no more than 20. The Human Resources Department uses a default employee number to replace a supervisor who leaves the company. It is not possible for an employee to be his or her own supervisor.



Note: The relationship shown models the italicized text

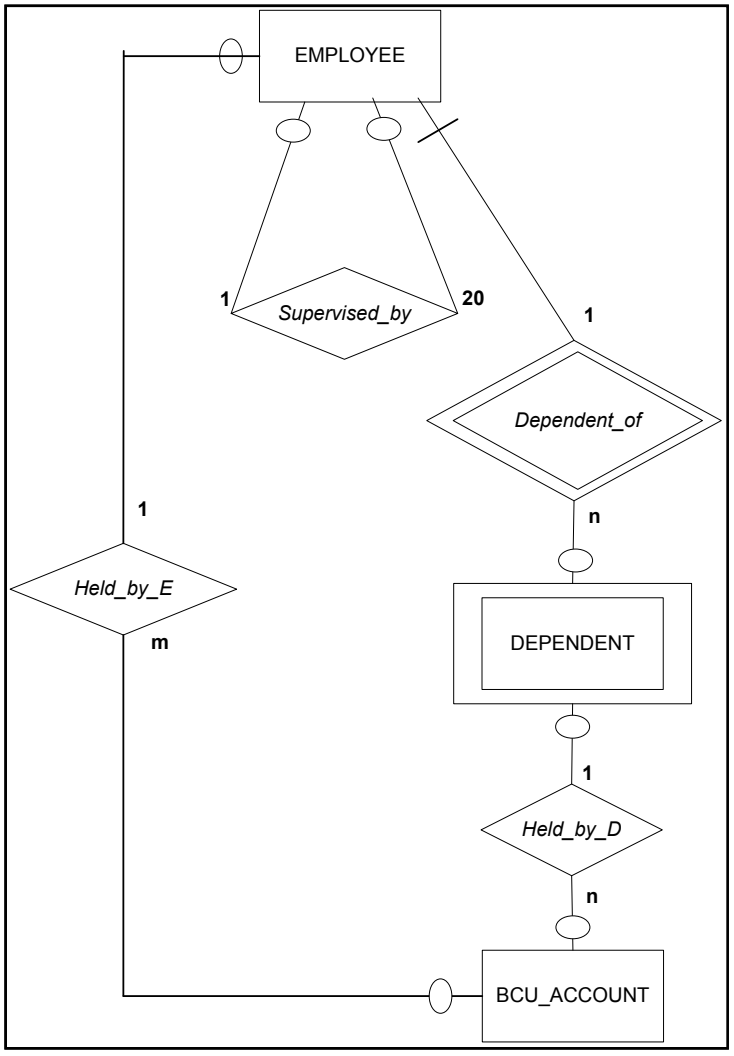
Development of Presentation Layer ER Model (continued)

Box 4 Storyline

Some employees may have several dependents. Bearcat Incorporated does not allow both husband and wife to be an employee of the company. Also, a dependent can only be a dependent of one employee at any time. Bearcat Incorporated offers credit union facilities as a service to its employees and to their dependents. An employee is not required to become a member of Bearcat Credit Union (BCU). However, most employees and some of their dependents have accounts in BCU. Some BCU accounts are individual accounts and others are joint accounts between an employee and his or her dependent(s). Every BCU account must belong to at least an employee or a dependent. Each joint account must involve no more than one employee and no more than one of his or her dependents. If an employee leaves the company, all dependents and BCU accounts of the employee must be removed. In addition, as long as a dependent has a BCU account, deletion of the dependent is not permitted.

Development of Presentation Layer ER Model (continued)

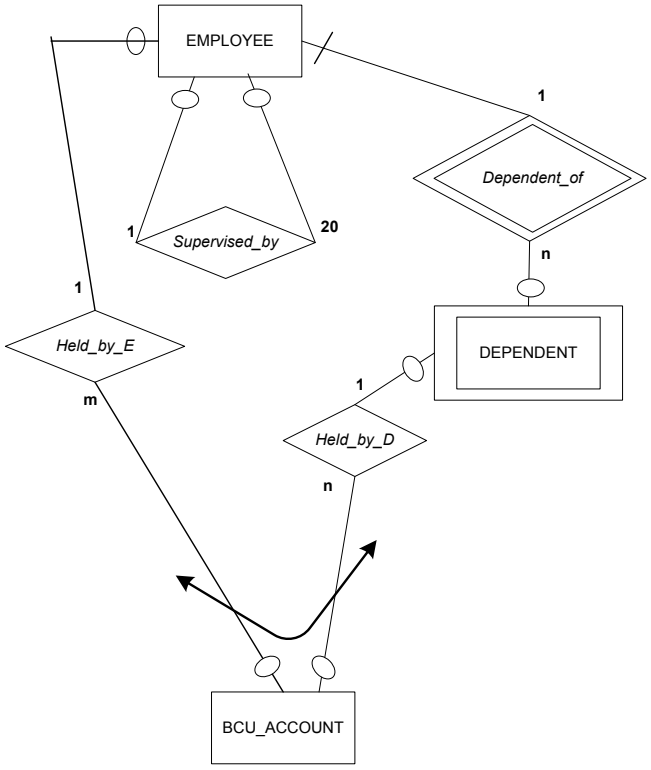
Box 4



Development of Presentation Layer ER Model (continued)

Business rule: Joint-account between an employee and a dependent is not permitted

Box 5

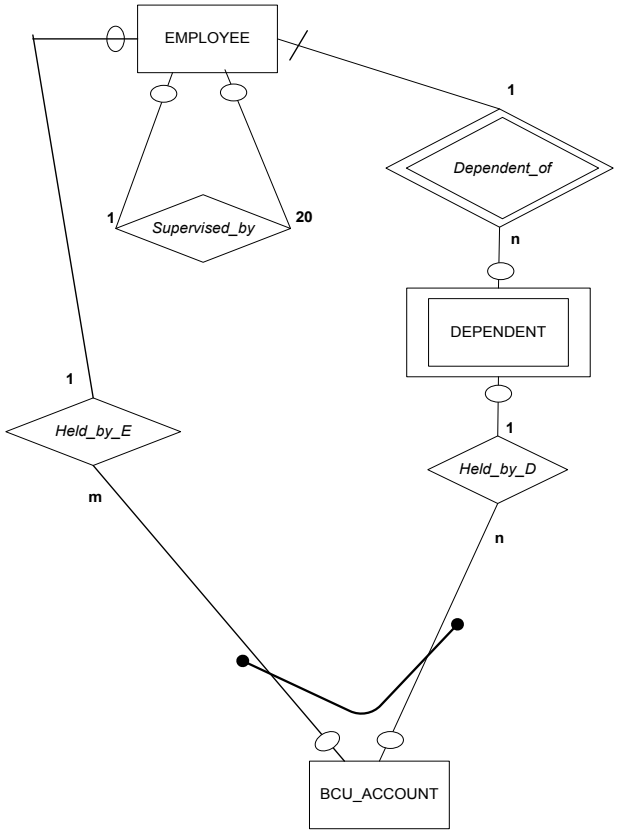


Note: The exclusive arc indicates that the relationship types *Held_by_E* and *Held_by_D* are mutually exclusive; i.e., *Held_by_E* and *Held_by_D* are manditorily exclusive.

Development of Presentation Layer ER Model (continued)

Business rule: All BCU accounts are joint-accounts between an employee and a dependent

Box 6

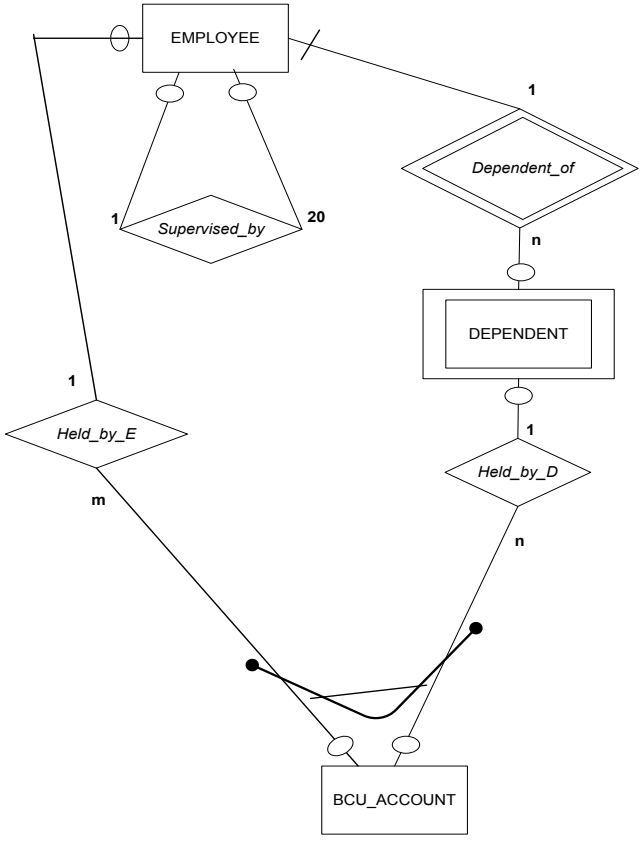


Note: The inclusive arc indicates that the relationship types *Held_by_E* and *Held_by_D* are mutually inclusive; i.e., *Held_by_E* and *Held_by_D* are mandatorily inclusive.

Development of Presentation Layer ER Model (continued)

Business rule: Joint-account between an employee and a dependent is permitted

Box 7

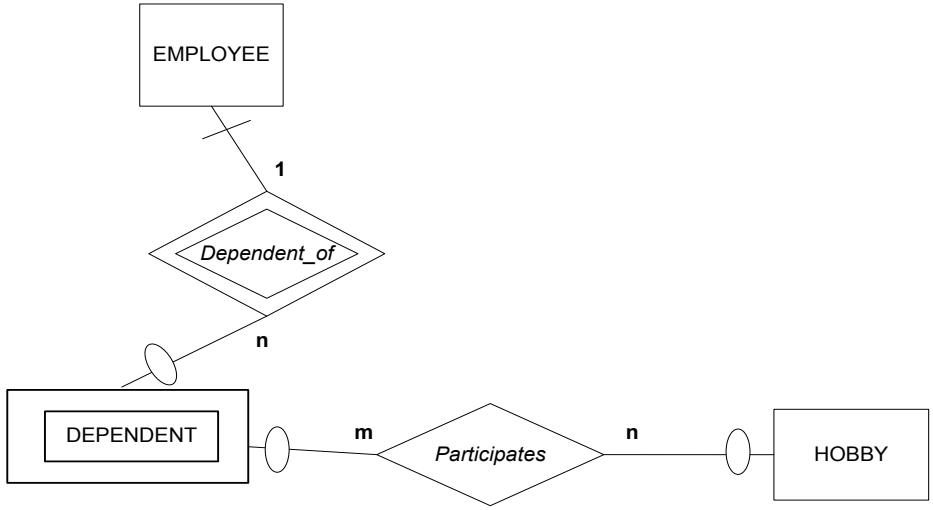


Note: The noninclusive arc indicates that the relationship types *Held_by_E* and *Held_by_D* are mutually noninclusive; i.e., *Held_by_E* and *Held_by_D* are optionally inclusive.

Development of Presentation Layer ER Model (continued)

Box 8

Bearcat Incorporated sponsors recreational opportunities for the dependents of employees in order to nurture the hobbies of the dependents. Some dependents need not have a hobby, however, it is possible that some dependents may have several hobbies. Because some hobbies are not as popular as others, every hobby need not have participants. If a dependent is no longer in the database, all records of the participation of that dependent in hobbies should not exist in the database either. Finally, as long as at least one dependent participates in a hobby, that hobby should continue to exist.



Note: The relationship shown models the italicized text

Presentation Layer ER Diagram

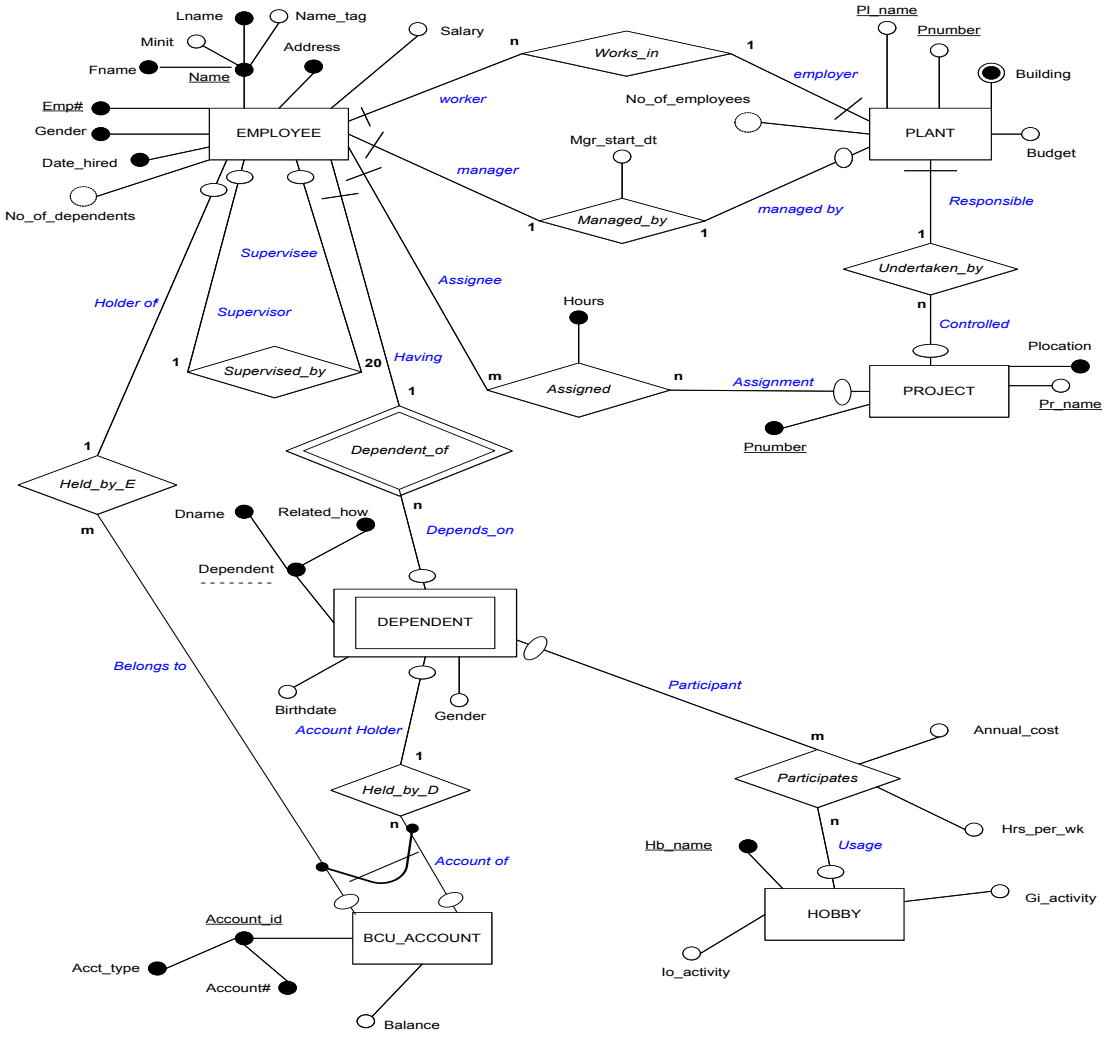


Figure 3.3 Presentation Layer ER diagram for Bearcat, Incorporated

Presentation Layer
Semantic Integrity Constraints

Semantic integrity constraints are grouped into the following categories:

- Attribute level business rules
- Entity level business rules
- Business rules governing entity deletion
- Miscellaneous business rules

Presentation Layer
Semantic Integrity Constraints (continued)

Table 3.1 Semantic integrity constraints for the Presentation Layer ER Model

Attribute-Level Business Rules

1. Each plant has a plant number that ranges from 10 to 20.
2. The gender of each employee or dependent is either male or female.
3. Project numbers range from 1 to 40.
4. Project locations are confined to the cities of Bellaire, Blue Ash, Mason, Stafford, and Sugarland.
5. Account types are coded as C – checking account, S – savings account, I – investment account.
6. A hobby can be either I – indoor activity or O – outdoor activity.
7. A hobby can be a G – group activity or I – individual activity.

Entity-Level Business Rules

1. A mother or daughter dependent must be a female, a father or son dependent must be a male, and a spouse dependent can be either male or female.
2. An employee cannot be his or her own supervisor.
3. A dependent may have a joint account only with an employee of Bearcat Incorporated to whom he or she is related.
4. Either plant number or plant name must be present.
5. Every plant is managed by an employee who works in the same plant.

Business Rules Governing Entity Deletion

1. A plant with employees cannot be closed down.
2. If an employee is deleted, all BCU accounts of that particular employee must be deleted.
3. *If a plant is closed down, the projects undertaken by that plant cannot be canceled. The project assignments from a closed plant must be temporarily removed in order to allow the project to be transferred to another plant.
4. The Human Resources Department uses a designated default employee number to replace a supervisor who leaves the company.
5. An employee currently managing a plant cannot be deleted from the database.
6. If a plant is closed down, the employee no longer manages the plant but becomes an employee of another plant.
7. **If an employee leaves the company, all dependents and BCU accounts of the employee must be removed.
8. **As long as a dependent has a BCU account, the deletion of that dependent is prohibited.
9. As long as an employee is assigned to a project his or her record cannot be removed from the database.
10. If a project is deleted, all assignments of employees to that project must be deleted.
11. If a dependent is deleted, all records of the participation of that dependent in hobbies must be deleted.
12. A hobby with at least one dependent participating in it cannot be deleted.

**Honoring this rule entails relaxation of the requirement at the beginning of Section 3.1 that a project is always under control of a plant.*

***Rule 7 cannot be honored for dependents who have bank accounts because Rule 8 prohibits deletion of such dependents. This is resolved in favor of Rule 7 by letting the bank account be deleted when a dependent is deleted. Additional discussion of this conflict resolution appears in Section 3.2.3.*

Miscellaneous Business Rules

1. Each plant has at least three buildings.
2. Each plant must have at least 100 employees.
3. The salary of an employee cannot exceed the salary of the employee's supervisor.

Design-Specific ER Model (Technology-Independent Design Tool)

- Not a Science !
- Art? Engineering?

  Not an Art anymore; more of an Engineering
- Systematic process
- Target Audience: Database Designer/Administrator
  - Coarse granularity
  - Fine granularity

Coarse-granular Design-Specific ER Model

- Tasks
  - Collect additional attribute characteristics relevant to design/implementation (e.g., data type, size) and prepare an updated list of Semantic Integrity Constraints
  - Introduce a technically more precise notation, viz., (min, max) notation, for expressing the structural constraints of a relationship type
  - Map deletion rules to the ER diagram

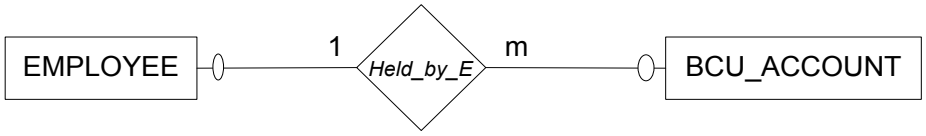
Design-Specific ER Model (Coarse Granularity)

Components of a Design-specific ER Model (Coarse Granularity) comprise:

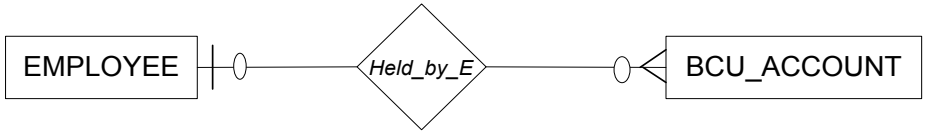
- Collection of a few characteristics for attributes (i.e., data type and size)
- Using the technically more precise (min, max) notation for the specification of relationships
- Mapping deletion rules to the ER diagram

See Figures 3.4 – 3.8 in the textbook

Various Structural Constraint Notations

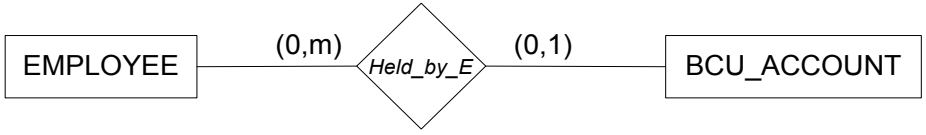


(a)
'Look Across' (Chen's) Notation for a Binary Relationship



Note: the bar adjacent to EMPLOYEE and the crow's foot adjacent to BCU_ACCOUNT convey the cardinality ratio of 1:m.

(b)
'Look across' (Crow's foot variant) notation for a binary relationship



(c)
'Look here' (min, max) notation for a binary relationship

Figure 3.4 Introduction of (min, max) notation for a binary relationship

Differentiating Chen's Notation From the (min, max) Notation

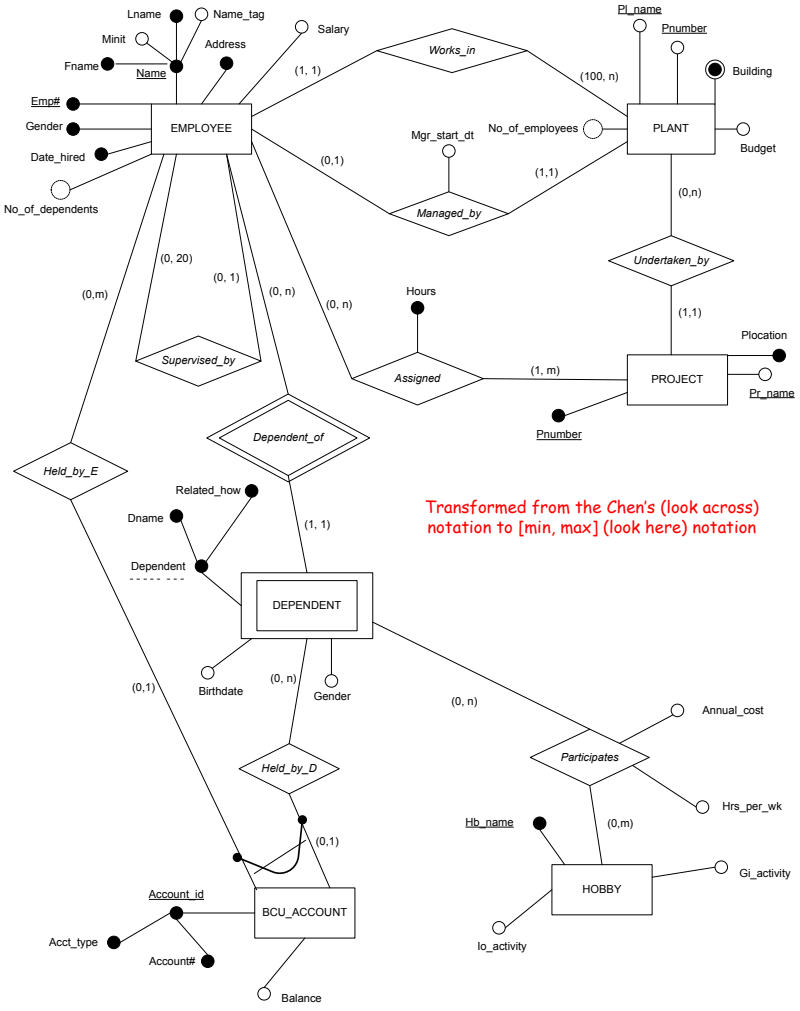


Figure 3.5 Course-granular Design-Specific ER diagram for Bearcat, Incorporated - Stage 1

Deletion Constraints

Restrict Rule (R): If a parent entity in a relationship is deleted and if all child entities related to this parent in this relationship should not be deleted, then the deletion of the parent should be disallowed

Cascade Rule (C): If a parent entity in a relationship is deleted and if all child entities related to this parent in this relationship should also be deleted, the cascade rule applies

Note: Conventionally, when a deletion constraint is not specified, the restrict rule is implied by default

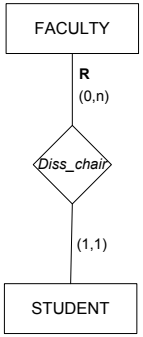
Deletion Constraints (continued)

Set Null Rule (N): If a parent entity in a relationship is deleted and if all child entities related to this parent in this relationship should be retained but no longer referenced to this parent, the 'set null' rule applies

Set Default Rule (D): If a parent entity in a relationship is deleted and if all child entities related to this parent in this relationship should be retained but no longer referenced to this parent but should be referenced to a predefined default parent, the 'set default' rule applies

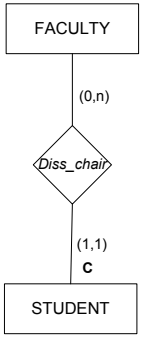
The Four Deletion Rules

Deletion Rule Examples



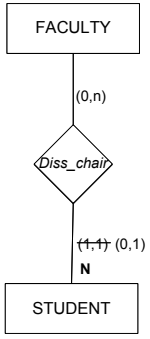
The Restrict Rule

When a parent entity in a relationship is deleted, if all child entities related to this parent in this relationship should not be deleted, then the deletion of the parent should be **restricted**.



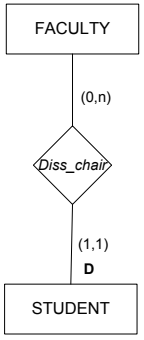
The Cascade Rule

When a parent entity in a relationship is deleted, if all child entities related to this parent should be deleted, then the **cascade rule** applies.



The ‘Set Null’ Rule

When a parent entity in a relationship is deleted, if all child entities related to this parent in this relationship should be retained but no longer referenced to this parent, the **“set null”** rule applies.



The ‘Set Default’ Rule

When a parent entity in a relationship is deleted, if all child entities related to this parent in this relationship should be retained but no longer referenced to this parent, but should be referenced to a predefined default parent, the **“set default”** rule applies.

Representing Deletion Constraints Using Deletion Rules

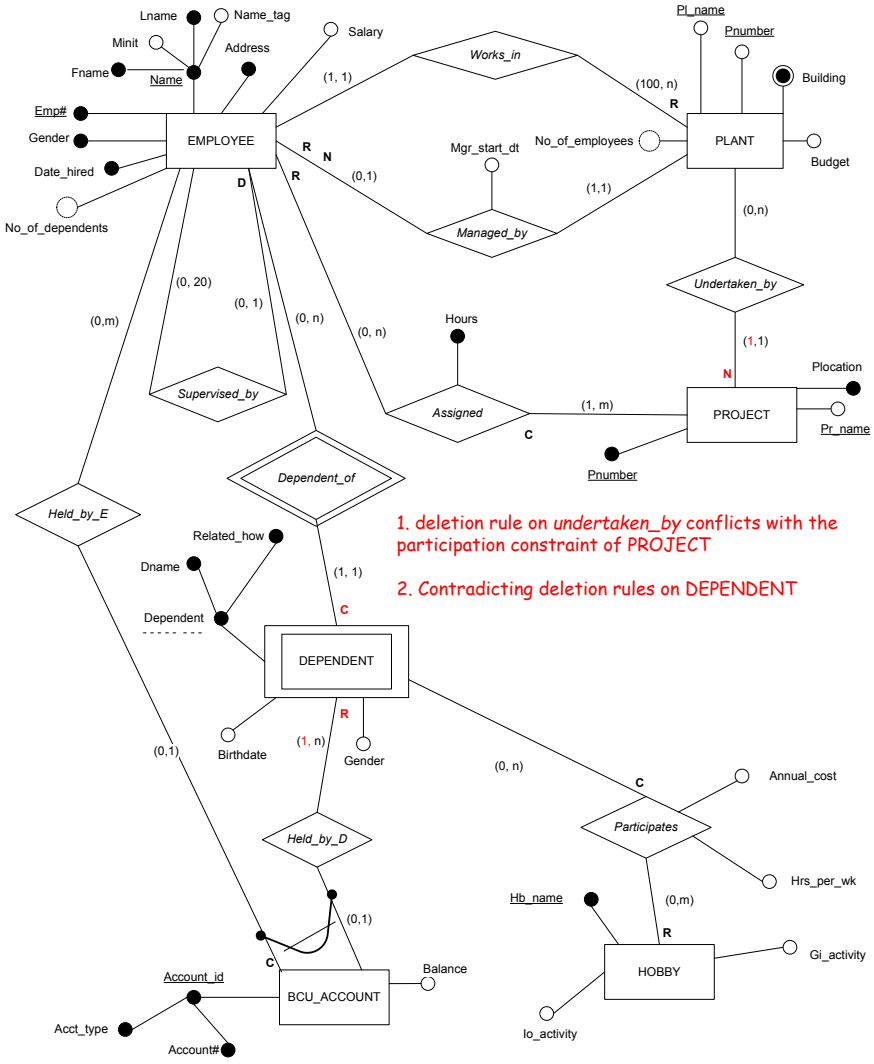


Figure 3.7 Course-granular Design-Specific ER diagram for Bearcat, Incorporated - Stage 2
{deletion constraints added}

Representing Deletion Constraints Using Deletion Rules (continued)

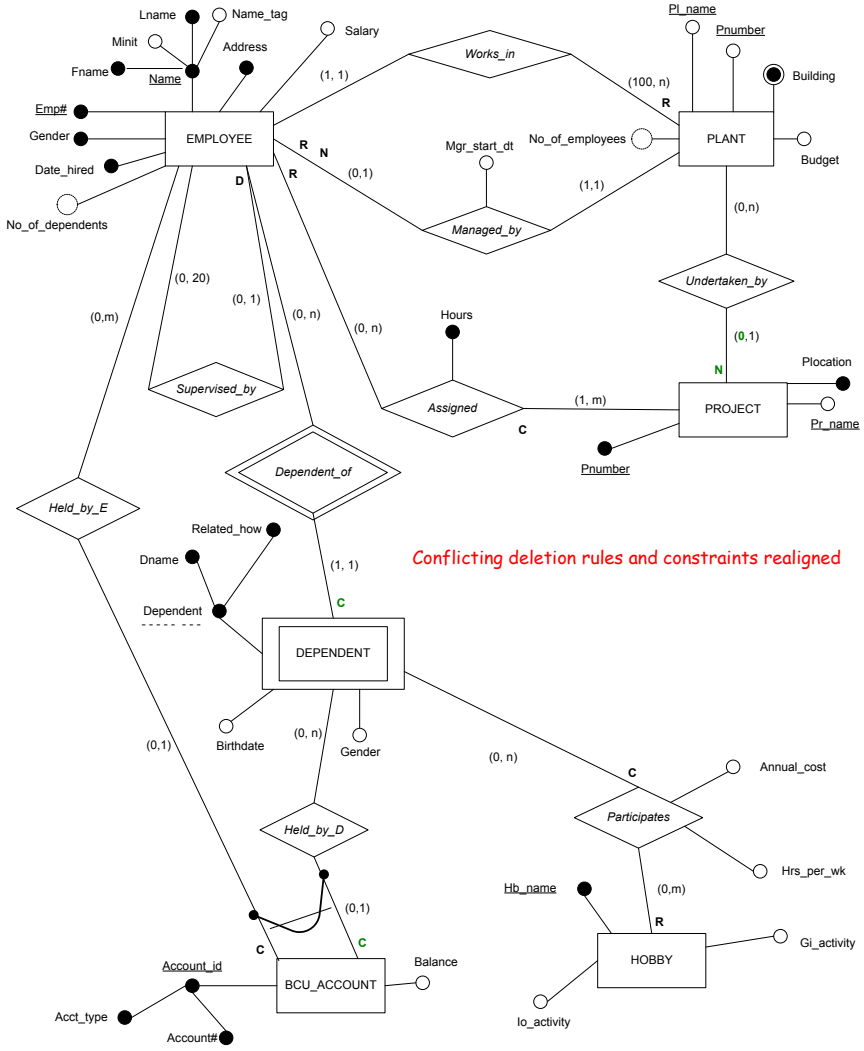


Figure 3.8 Course-granular Design-Specific ER diagram for Bearcat, Incorporated - Final

Fine-granular Design-Specific ER Model

- Tasks
  - Map the additional attribute characteristics to the ER diagram
  - Decompose ER constructs preparatory to logical model mapping
    - Decomposition of multi-valued attribute
    - Decomposition of **m:n** cardinality constraint (ratio)
  - Generate an updated list of Semantic Integrity Constraints

Design-Specific ER Model (Fine Granularity)

Components of the Fine-granular Design-Specific ER Model comprise:

- Mapping the attribute characteristics into the ER diagram (i.e., A-Alphabetic, N-Numeric, X-Alphanumeric, Dt-Data type)
- Decomposing constructs that cannot directly be mapped to a logical schema (i.e., multi-valued attributes and n:m relationship types)

See Figures 3.9 – 3.11 and Table 3.3 in Textbook

Replacing a Multi-valued Attribute with a Single-valued Attribute

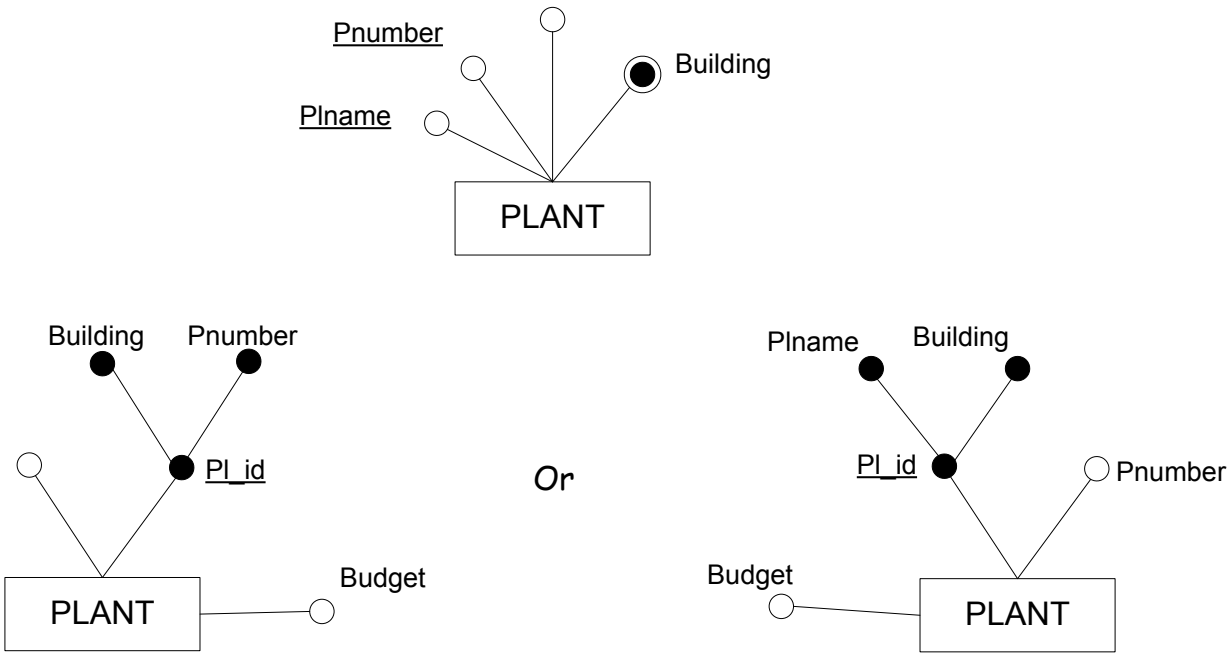


Figure 3.9 (a) Transformation of a multi-valued attribute to a single-valued attribute
Two design variations

An Illustration Using Sample Data

Original PLANT Data Set

Pnumber	Plname	Budget	Building
10	Underwood	3000000	1
			2
			3
11	Garnett	3000000	1
			2
12	Belmont	3500000	1
13	Vanderbilt	3500000	1
			2

Revised PLANT Data Set- Variation 1

Pnumber	Building	Plname	Budget
10	1	Underwood	3000000
10	2	Underwood	3000000
10	3	Underwood	3000000
11	1	Garnett	3000000
11	2	Garnett	3000000
12	1	Belmont	3500000
13	1	Vanderbilt	3500000
13	2	Vanderbilt	3500000

Revised PLANT Data Set- Variation 2

Pnumber	Building	Plname	Budget
10	1	Underwood	3000000
10	2	Underwood	3000000
10	3	Underwood	3000000
11	1	Garnett	3000000
11	2	Garnett	3000000
12	1	Belmont	3500000
13	1	Vanderbilt	3500000
13	2	Vanderbilt	3500000

(a)

Sample data illustrating the transformation of a multi-valued attribute to a single-valued attribute (see Figure 3.9a)

Replacing a Multi-valued Attribute With a Weak Entity Type

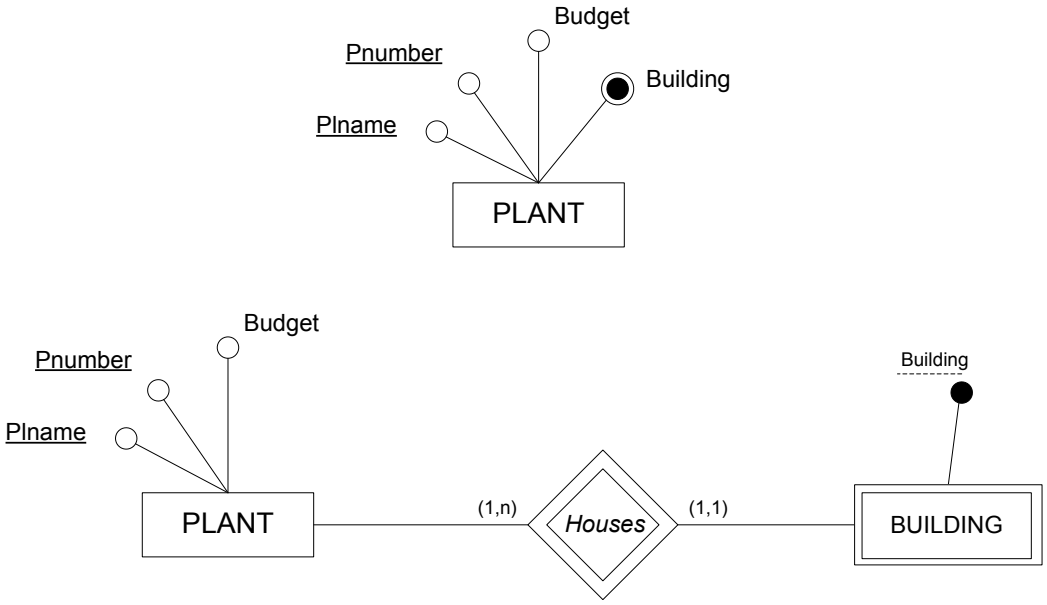


Figure 3.9 (b) Mapping a multi-valued attribute to a weak entity type
A structural alternative to Figure 3.9(a)

An Illustration Using Sample Data

Original PLANT Data Set

Pnumber	Plname	Budget	Building
10	Underwood	3000000	1 2 3
11	Garnett	3000000	1 2
12	Belmont	3500000	1
13	Vanderbilt	3500000	1 2

Revised PLANT Data Set

Pnumber	Plname	Budget
10	Underwood	3000000
11	Garnett	3000000
12	Belmont	3500000
13	Vanderbilt	3500000

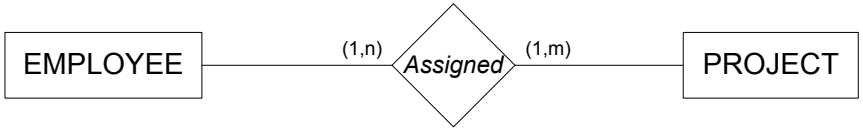
Revised BUILDING Data Set

Building
1
2
3
1
2
1
1
2

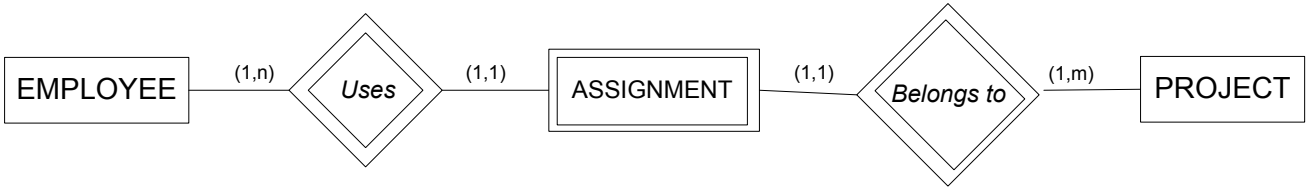
(b)

Sample data illustrating the transformation of a multi-valued attribute to a weak entity type (see Figure 3.9b)

Decomposing an m:n Relationship Type (by Creating a Gerund Entity Type)



(a)
An m:n relationship between EMPLOYEE and PROJECT



(b)
Decomposition of *Assigned* relationship type to eliminate m:n cardinality ratio

Figure 3.10 Resolution of m:n cardinality constraint (ratio)

The Fine-granular Design-Specific ER Diagram For Bearcat, Inc.

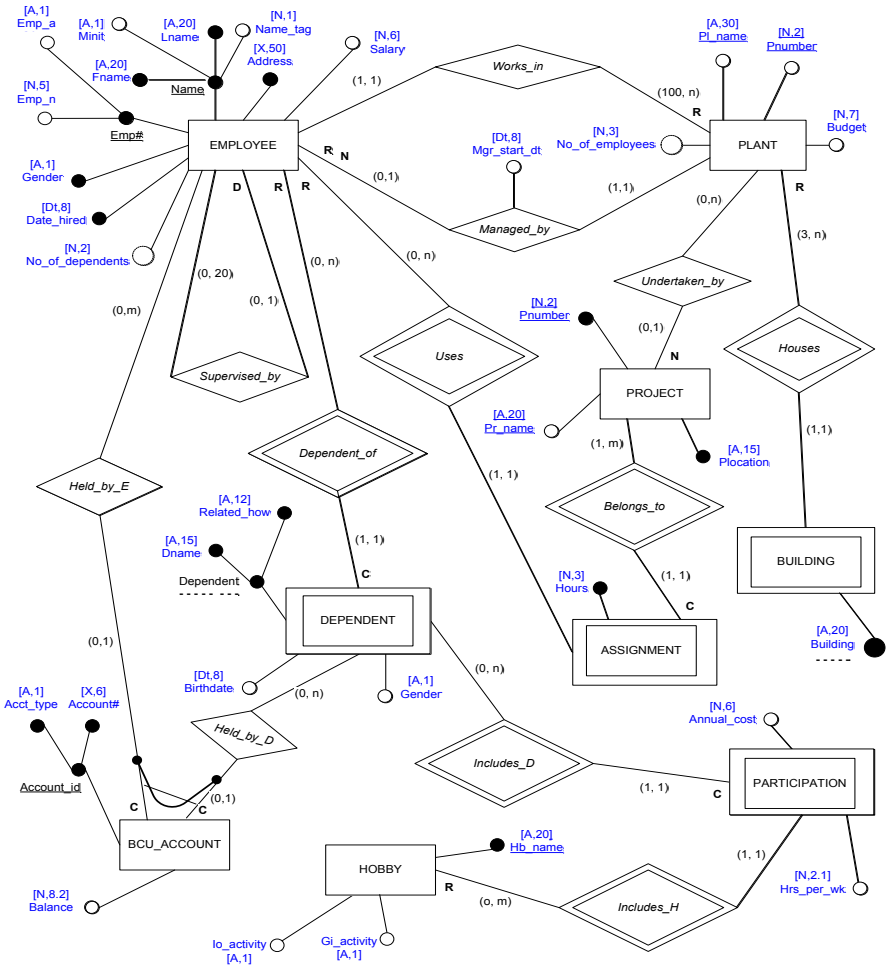


Figure 3.11 Fine-granular Design-Specific ER diagram for Bearcat, Incorporated

Summary

Presentation Layer ER Model

- Diagram + Integrity Constraints

Coarse-Granular ER Model

- Collection of a few characteristics for attributes (i.e., data type and size)
- Using the technically more precise (min, max) notation for the specification of relationships
- Mapping deletion rules to the ER diagram

Summary (continued)

Fine-Granular ER Model

- Mapping the attribute characteristics into the ER diagram (i.e., A-Alphabetic, N-Numeric, X-Alphanumeric, Dt-Data type)
- Decomposing constructs that cannot directly be mapped to a logical schema (i.e., multi-valued attributes and n:m relationship types)