پوهنتون کابل

پوهنحی کمپیوترساینس

Introduction to Database and Data Models

Lectures 14-16

تهیه کننده : پوهنیار محمد شعیب "زرین خیل"
سال : 1389

Introduction to Database and Data Models - Entity Relationship ER

14

By: M Shuaib Zarinkhail

2010

Database Design Process

- Why databases?
- Database Design
- Database Implementation

Why databases?

- Asking users and clients
  - Finding the scope
  - Preparing with the topic
- Drafting tables, forms, etc
- Creating a data model (The topic of this section)

Database Design

- Changing the data model to design the database
- Naming relations, attributes, relationships
- Declaring primary keys, foreign keys, constraints, …
- Implementing RIC, business rules, …

Database Implementation

- Creating tables, queries, forms, reports (practically)
- Writing application programs (if necessary)
- Entering user data (general database users)

Initial Steps (DB Design)

Step 1

- Collect, analyze and document requirements of a customer (UoD)
- Prepare a questionnaire to collect information:
  - What data would the customer like to store?
  - How would the customer like to access the data?

Initial Steps (DB Design)

Step 2

- Use the documentation (UoD) to develop the conceptual schema

Step 3

- Discuss the conceptual schema with the customer until the customer is satisfied

Data Modeling

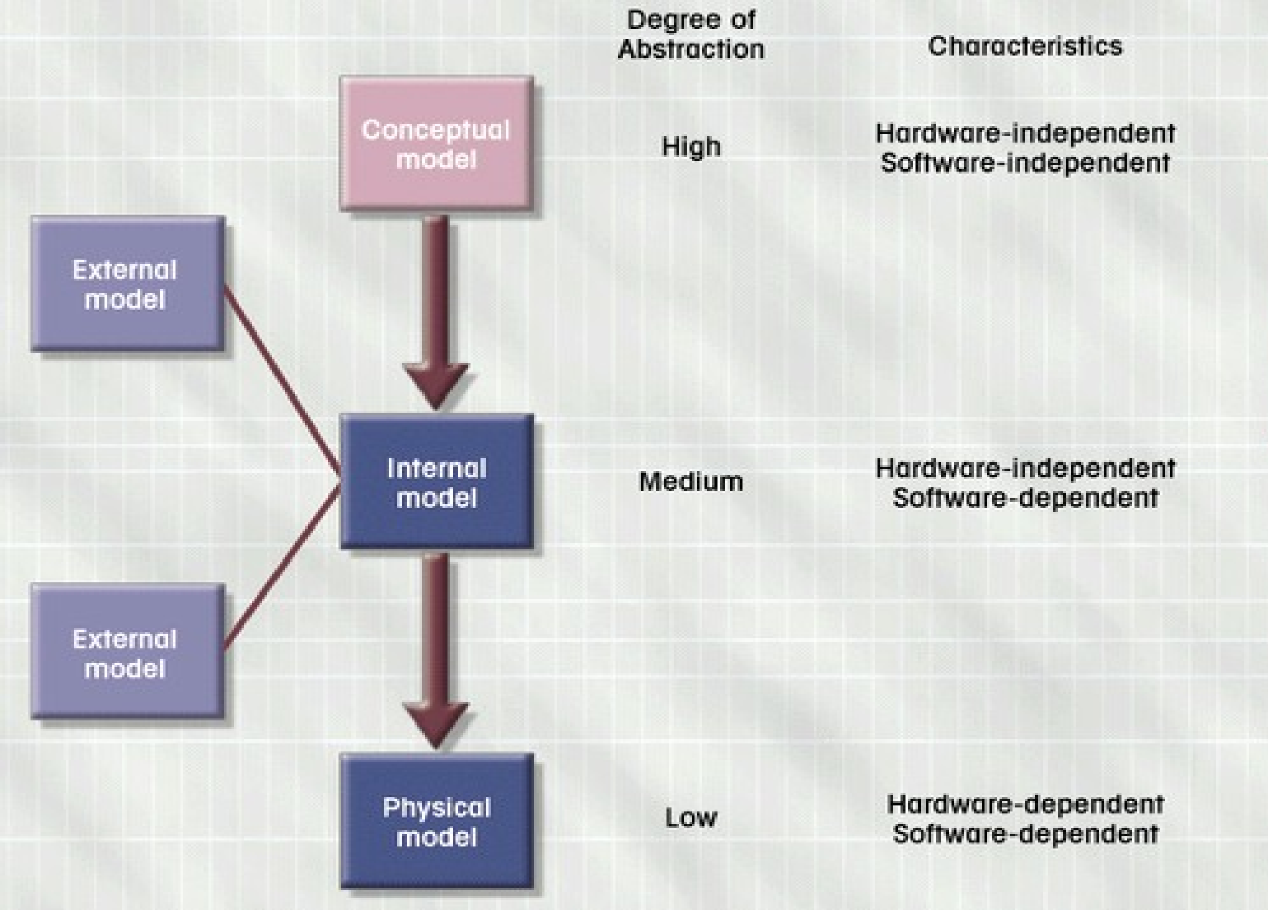
- Used as a mean of communication between database developer and client
  - Database developer -expert in database
    - Database developer needs to understand domains
  - Client -expert in their own domains
    - Independent of implementation
- Examples
  - Entity-Relationship Diagram (ERD)
  - Unified Modeling Language (UML)

Basic Modeling Concepts

- Art and science
- Good judgment coupled with powerful design tools
- Models
  - "Description or analogy used to visualize something that cannot be directly observed"

    Webster's Dictionary
- Data Model
  - Relatively simple representation of complex real-world data structures

Data Models: Degrees of Data Abstraction



Degree of Abstraction	Characteristics
High	Hardware-independent Software-independent
Medium	Hardware-independent Software-dependent
Low	Hardware-dependent Software-dependent

Degrees of Data Abstraction 1

- High-Level (Conceptual)
  - Global view of data
  - Basis for identification and description of <u>main data items</u>
  - ERD used to represent conceptual data model
  - Hardware and software independent

Degrees of Data Abstraction 2

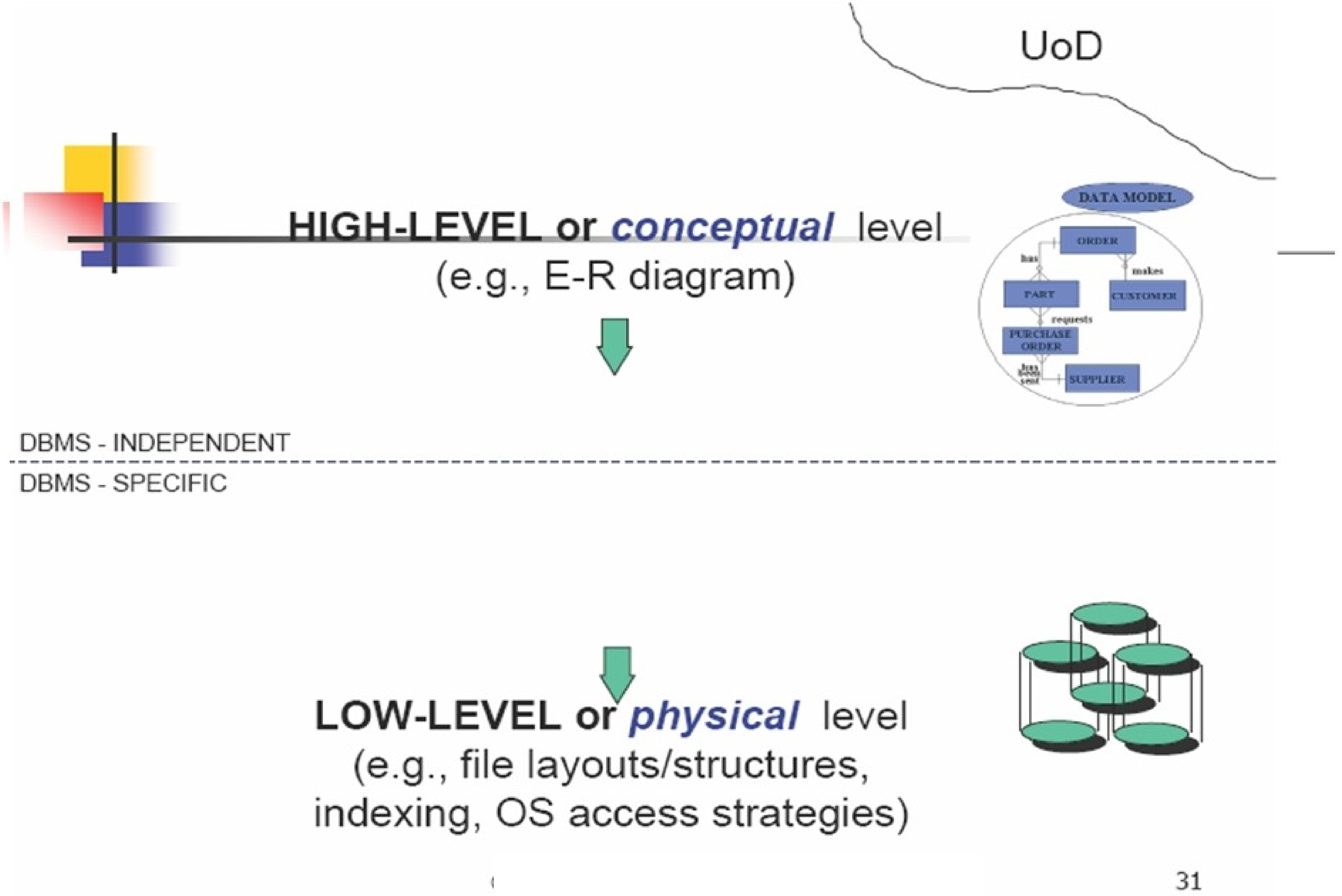
- Representational-Level (Internal)
  - Representation of database as seen by DBMS
  - Adapts conceptual model to specific DBMS
  - Software dependent or specific
  - Hardware independent

Degrees of Data Abstraction 3

- Representational-Level (External)
  - Users’ views of data environment
  - Provides subsets of internal view
  - Makes application program development easier
  - Facilitates designers’ tasks
  - Ensures adequacy of conceptual model
  - Ensures security constraints in design

Degrees of Data Abstraction 4

- Low-Level (Physical)
  - The lowest level of abstraction
  - Software and hardware dependent
  - Requires definition of
    - physical storage devices
    - access methods
    - distribution methods



UoD

HIGH-LEVEL or *conceptual* level

(e.g., E-R diagram)

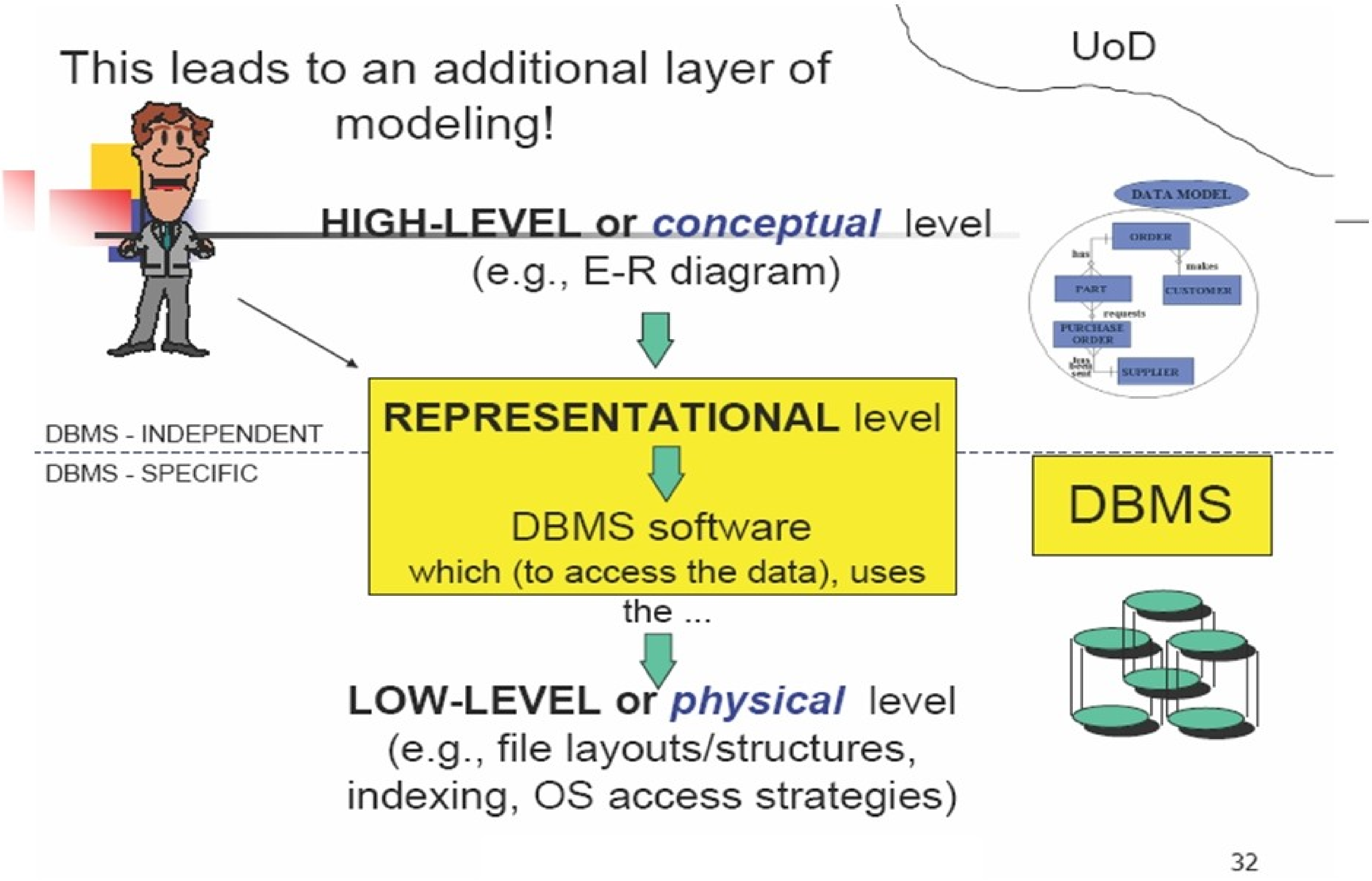
DBMS - INDEPENDENT

DBMS - SPECIFIC

LOW-LEVEL or *physical* level

(e.g., file layouts/structures, indexing, OS access strategies)

31



This leads to an additional layer of modeling!

UoD

HIGH-LEVEL or *conceptual* level
(e.g., E-R diagram)

REPRESENTATIONAL level

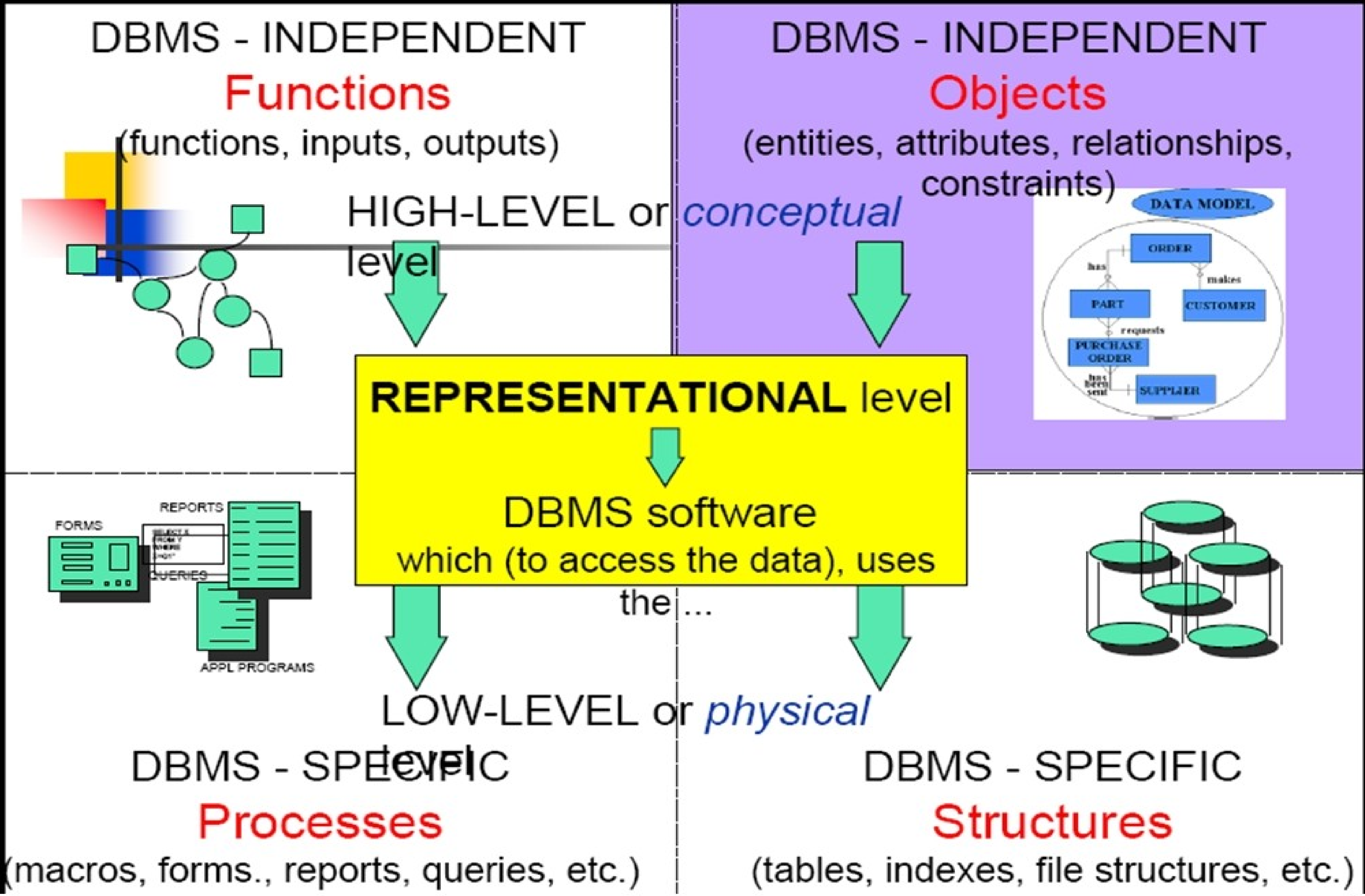
DBMS software which (to access the data), uses the ...

LOW-LEVEL or *physical* level
(e.g., file layouts/structures, indexing, OS access strategies)

DBMS - INDEPENDENT

DBMS - SPECIFIC

DBMS



DBMS - INDEPENDENT

Functions

(functions, inputs, outputs)

DBMS - INDEPENDENT

Objects

(entities, attributes, relationships, constraints)

HIGH-LEVEL or *conceptual* level

REPRESENTATIONAL level

DBMS software which (to access the data), uses the ...

LOW-LEVEL or *physical* level

DBMS - SPECIFIC

Processes

(macros, forms., reports, queries, etc.)

DBMS - SPECIFIC

Structures

(tables, indexes, file structures, etc.)

Conceptual Level Design

- Data modeling creates abstract data structure to
  - represent UoD
  - help data modeler to confirm final decisions with clients
- High level of data abstraction
- Include four steps

Conceptual Level Design

- Four Steps
  - **Data analysis and requirements**
  - Entity relationship modeling and normalization
  - Data model verification
  - Distributed database design

Data Analysis & Requirements 1

- Focus on:
  - Information - <u>needs</u>
    - What are needs of company/person
  - Information - <u>users</u>
    - Who are users of the system
  - Information - <u>sources</u>
    - What are information sources of company/person
  - Information - <u>constitution</u>
    - What general constitution and structure the information may have

Data Analysis & Requirements 2

- Developing and gathering end-user data views
  - What do end users of system want
- Direct observation of current system
  - What and how does current system used by company/person
  - Interfacing with systems' design group
    - Talking and gathering information from previous system's design group

Data Analysis & Requirements 3

- Business rules
  - Identifying business rules
  - What business rules need to be implemented
  - What business rules may need to be implemented in the future

Introduction to Database and Data Models - Entity Relationship ER

15

By: M Shuaib Zarinkhail

2010

Conceptual Level Design

- Four steps
  - Data analysis and requirements
  - **Entity relationship modeling and normalization**
  - Data model verification
  - Distributed database design

E-R Modeling 1

Purpose

- The E/R model allows us to sketch database schema design
  - Includes some constraints, but not operations
- Designs are pictures called E-R Diagrams (ERDs)

E-R Modeling 2

Framework

- Design is a serious of business
- The user knows they want a DB, but they don’t know what they want in it
- Sketching the key components is an efficient way to develop a working database

E-R Modeling 3

Advantages

- Data Analysis vs. Process Analysis
  - Data is more stable than processes
- Graphical Models vs. Prose
  - Graphical model is more decisional than text

E-R Modeling 4

- High-level / conceptual data model
- Is used to develop
  - the initial conceptual schema
  - the logical structure of the DB
- Easy to understand -- Peter Chen, 1976
  - E-R Models
  - E-R Diagrams

No Single Standard!

E-R Modeling Process 1

1. Identify, analyze, and refine the business rules
    - All business rules between UoD components for a specific field should be analyzed and refined
    - All transactions between UoD components for a specific field should be analyzed and refined

E-R Modeling Process 2

1. Identify the main entities, using the results of step 1
   - All types of entities should be identified and added to the model as:
     - Main entities – Sub entities
     - Strong entities – Weak entities
     - Centralized entities – Decentralized entities
     - Super-class (parent) entities – Sub-class (child) entities

E-R Modeling Process 3

1. Define the attribute names, identifiers, and foreign keys for
   - Each entity class
   - Each Relationship
2. Define the relationships
   - Between the entity classes
   - Between the entity instances
   - Using the results of steps 1 and 2

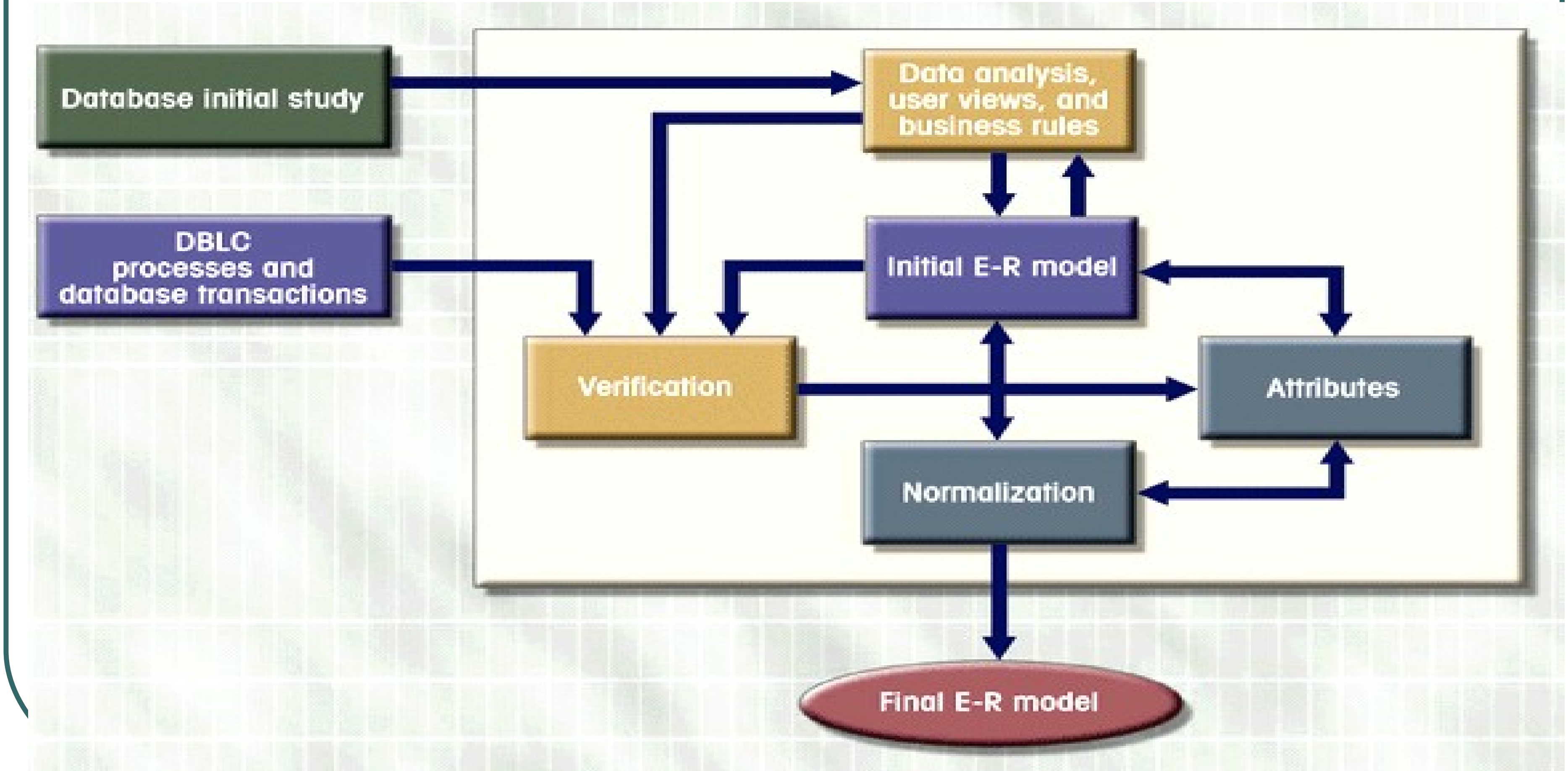
E-R Modeling Process 4

1. Normalize the entities to reduce Data redundancy in
   - Data insertion process
   - Data update process
   - Data deletion process
2. Complete the initial E-R diagram
   - Update, update, & update the diagram

E-R Modeling Process 5

1. Have the main end-users verify the model in step 6 against
   - The data information
   - Processing requirements
2. Modify the E-R diagram, using the results of step 7 to be finalized

E-R Modeling is Iterative



E-R Model Components

- Represents conceptual view
- Many Components
- Main Components include
  - Entities
    - Corresponds to entire table, not row
    - Represented by rectangle
  - Attributes
  - Relationships

E-R Model Components

Entity

- Entity Class - Entity Instance
- Strong Entity - Weak Entity
- Composite Entity Instances

Attribute

- Simple - Composite
- Single Value - Multi Value

Identifier

- Unique - Not Unique (Common)
- Composite

E-R Model Components

Relationships

- Relationship Class - Relationship Instance
- Relationship Degree

Cardinality Ratios

- Maximum Cardinality
- Minimum Cardinality
- Existence Dependency

E-R Model Symbols

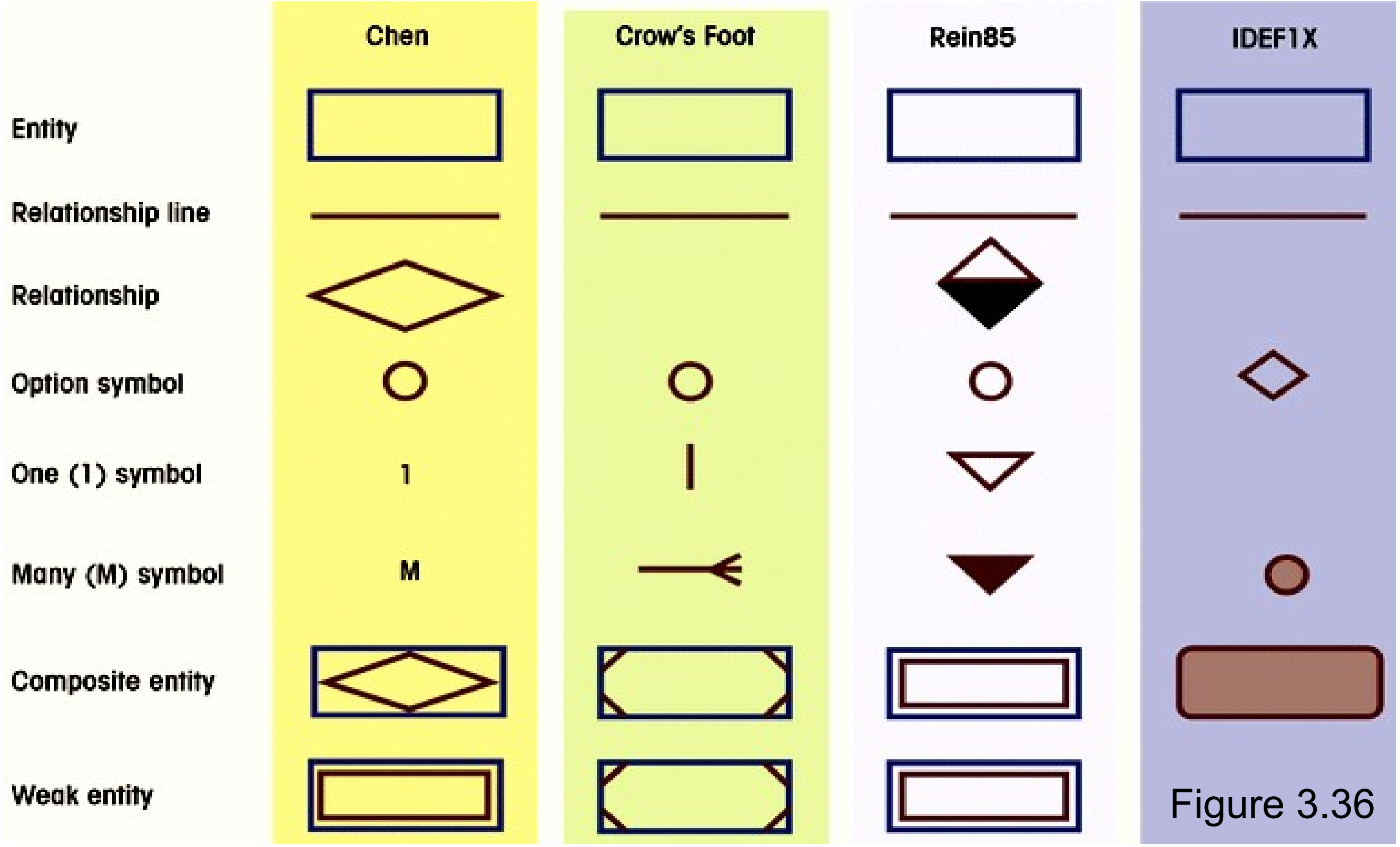


Figure 3.36

Entity (Entity Class)

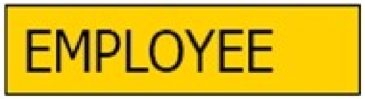
- Something the user wants to track
- A group of entity instances
- Nouns
- Represented by rectangle
  - Examples

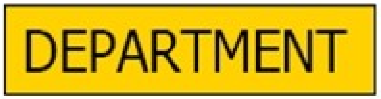
Entity (Entity Instance)

- An actual occurrence of data for an entity class
- Usually not shown

Entity (Entity Class)

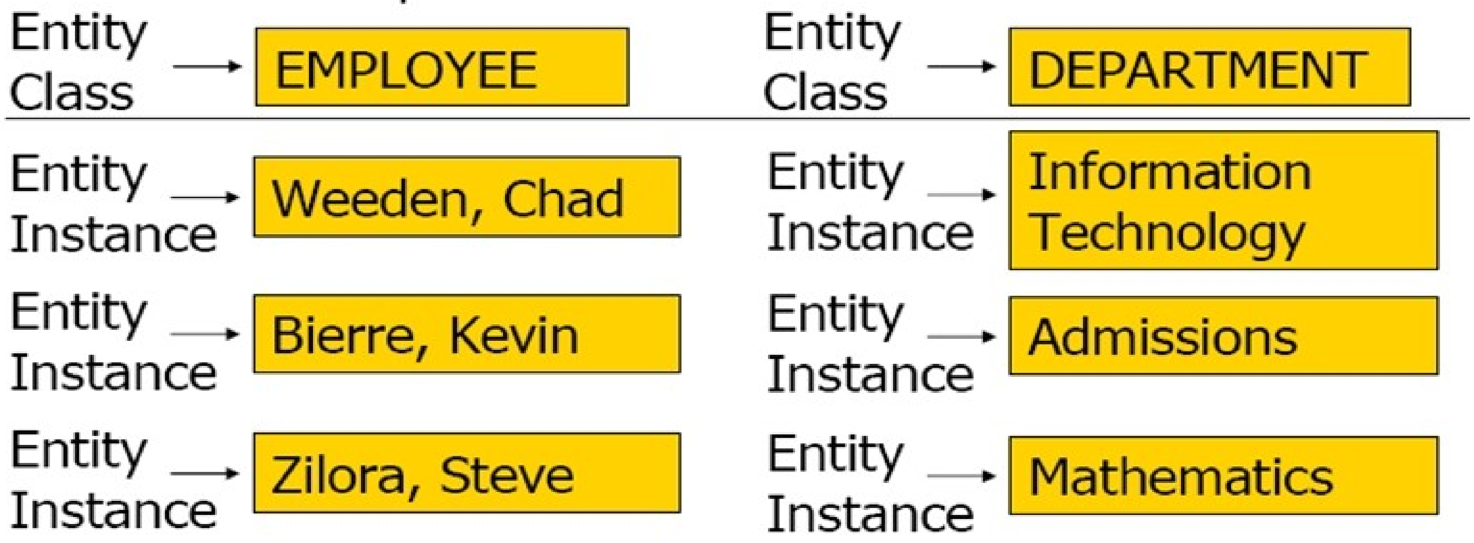
- Something the user wants to track
- A group of entity instances
- Nouns
- Represented by rectangle
  - Examples





Entity (Entity Instance)

- An actual occurrence of data for an entity class
- Usually not shown



Entity Class →	EMPLOYEE	Entity Class →	DEPARTMENT
Entity Instance →	Weeden, Chad	Entity Instance →	Information Technology
Entity Instance →	Bierre, Kevin	Entity Instance →	Admissions
Entity Instance →	Zilora, Steve	Entity Instance →	Mathematics

Strong and Weak Entities

Strong Entity

- Can exist independently

Weak Entity

- Can not exist on it's own
- Must have another entity to support it

ID-Dependent Weak Entity

- Uses the identifier of its 'Supporter Entity'

Composite Entity Instances

While N:M relationships

- Each relationship splits to two 1:N relationships
- The entity instances of the new created relations are called ‘Composite Entity Instances’

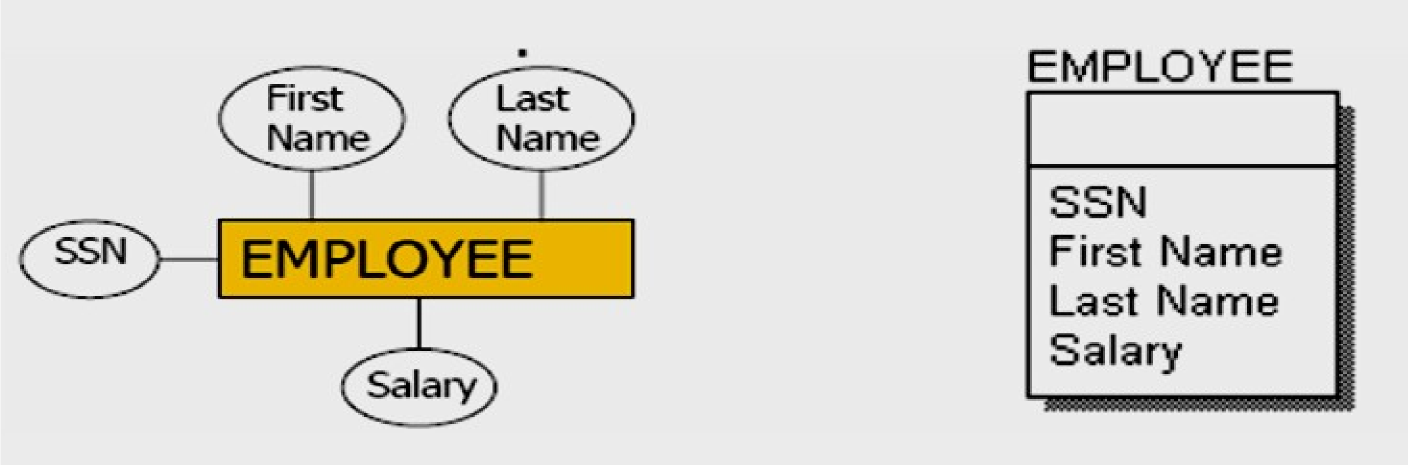
In-Class Exercise (ICE) #1

- Datahouse Reality
- Using the description of DataHouse Reality draw Entities for the system

HINT: Centrality count -if you count the number of times the different "entities" are mentioned, you can tell which one is central to the system

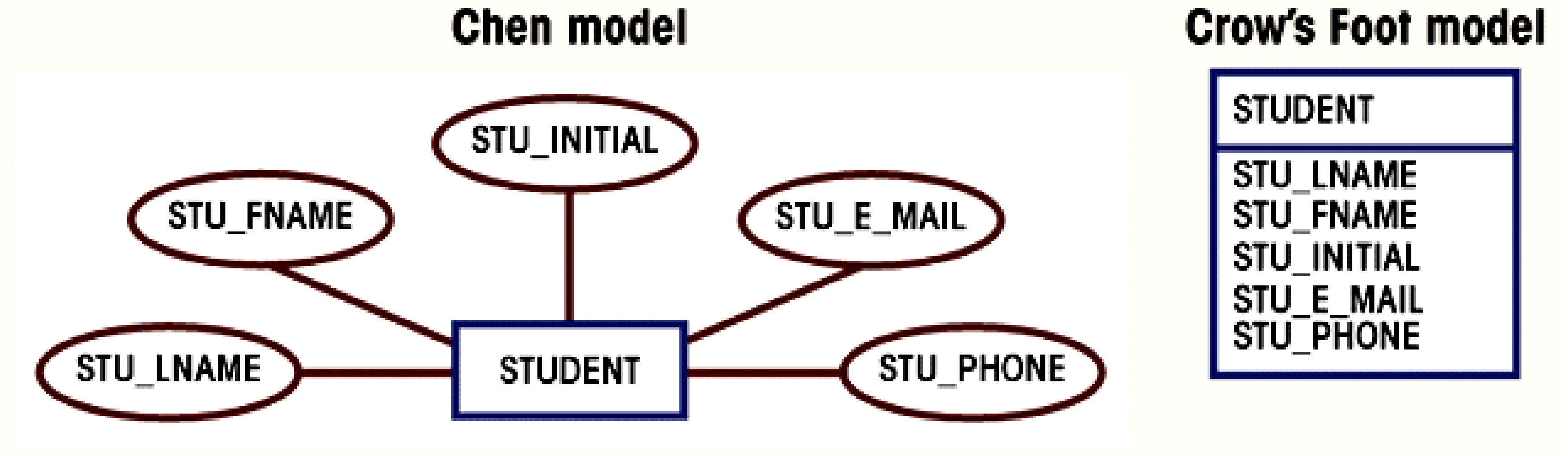
Attributes

- A data item that is used to describe an entity
  - Can be shown in an ERD
  - Each has its own domain



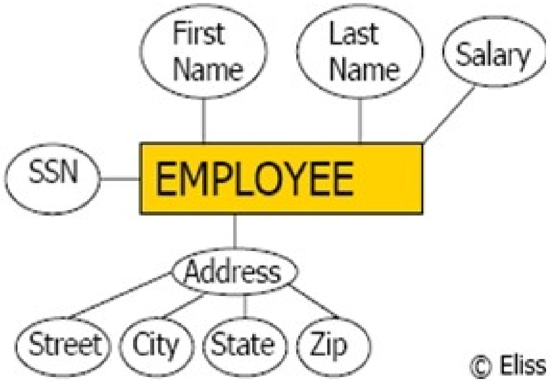
Attributes

- Characteristics of entities
- Each attribute has a domain
  - Domain is set of possible values

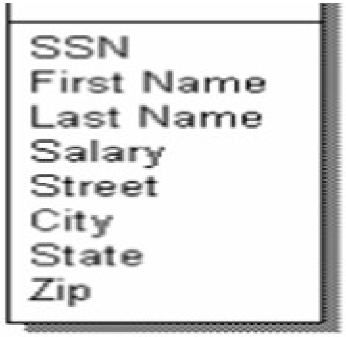


Attributes (Simple vs Composite)

- Simple Attribute: an attribute composed of one piece of data (can not be subdivided)
- Composite attribute: an attribute composed of other attributes (can be subdivided into additional attributes)



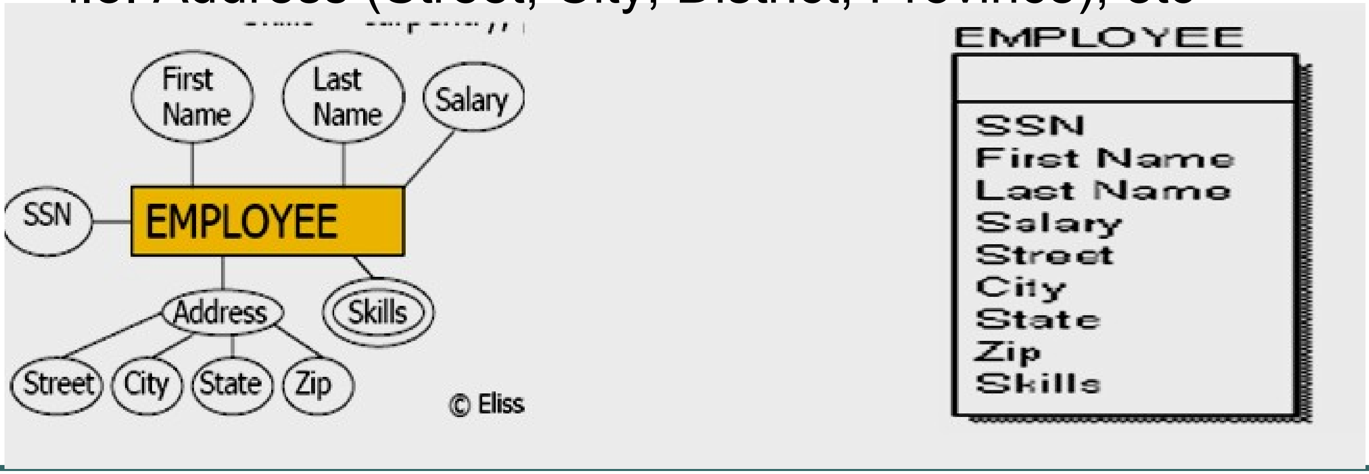
© Eliss



SSN
First Name
Last Name
Salary
Street
City
State
Zip

Attributes (Single-Values Vs Multi-Values)

- Single-Value: An attribtues that stores one single data value
  - i.e. Name, Salary, etc
- Multi-Value: An attribute that stores multiple data values
  - i.e. Address (Street, City, District, Province), etc



Attributes (Derived)

- Derived – An attribute that derived from one or more other attributes
  - Can derive with algorithm

  For Example:

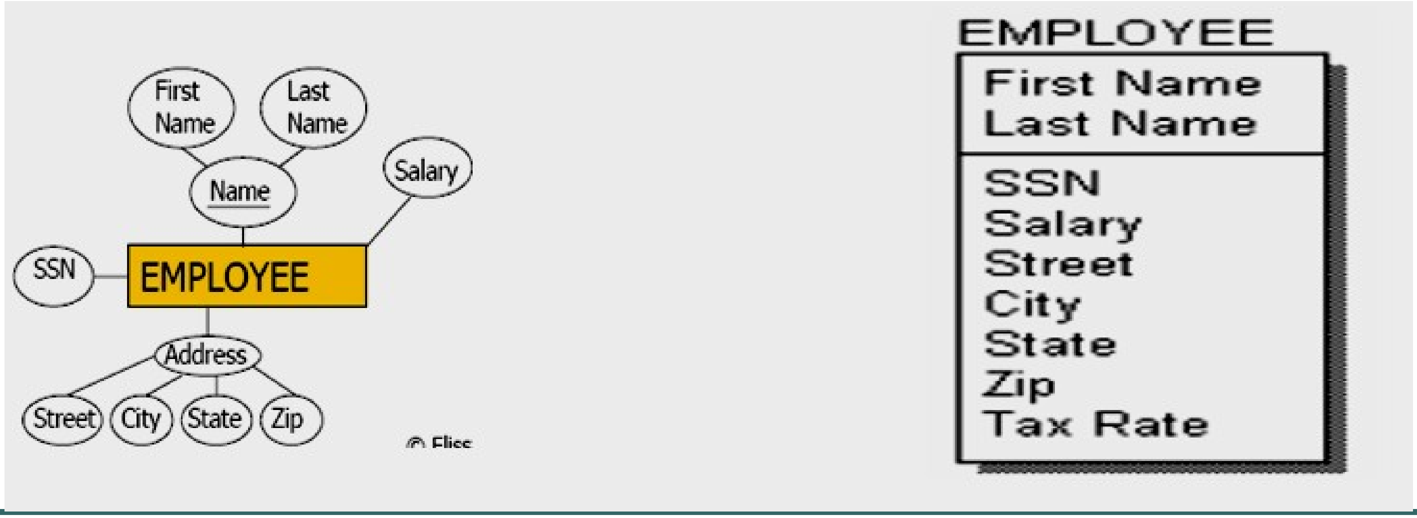
  - Age can derive from date of birth
  - Tax can derive from Salary and TaxRate

Identifiers (Unique / NonUnique)

- Distinguish between different entity instances
- Unique Identifier: determines a specific entity instance
  - i.e. NationID, VIN, StudentID, etc
- Non Unique Identifer: may determinne several entity instances
  - Generally not shown in diagrams *NEXT SLIDE*

Composite Identifier

- When more than one attribute is needed to identify an entity instance
- A ‘Compound Key’ is type of composite identifier



In-Class Exercise (ICE) #2

- DataHouse Reality
  - Add attributes and identifiers to your result from In-Class Exercise #1

Introduction to Database and Data Models - Entity Relationship ER

16

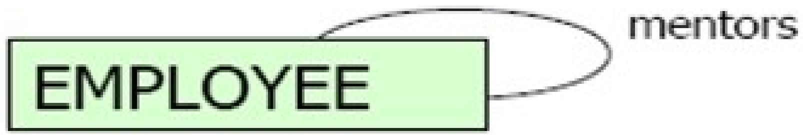
By: M Shuaib Zarinkhail

2010

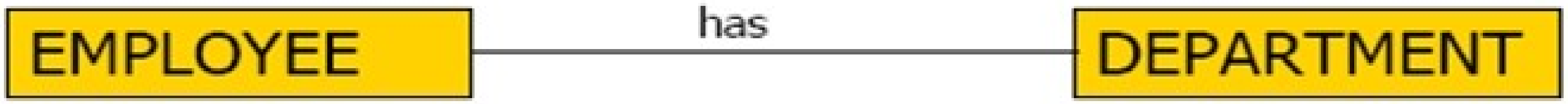
Relationship Class

- Denotes a connection between entity classes
- Can be multiple relationships between entity classes
- Can be named (optional)
- There several variations to show relationships
  - i.e Crow’s feet, Diamonds, etc

Relationship Class - Example



- An employee mentors another employee



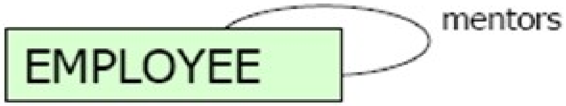
- An employee relates to a department

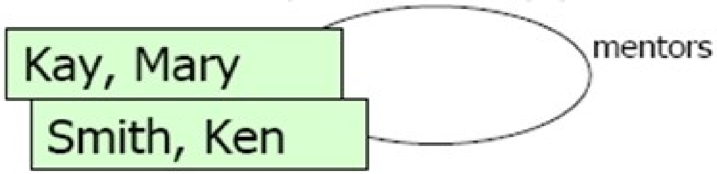
Relationship Instance

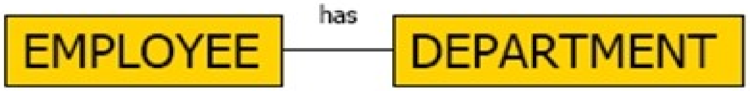
- Denotes a connection between entity instances

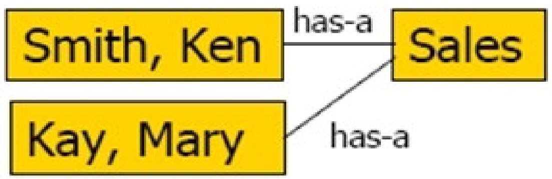
Entity Class | Entity Instances

Relation Class | Relationship Instances



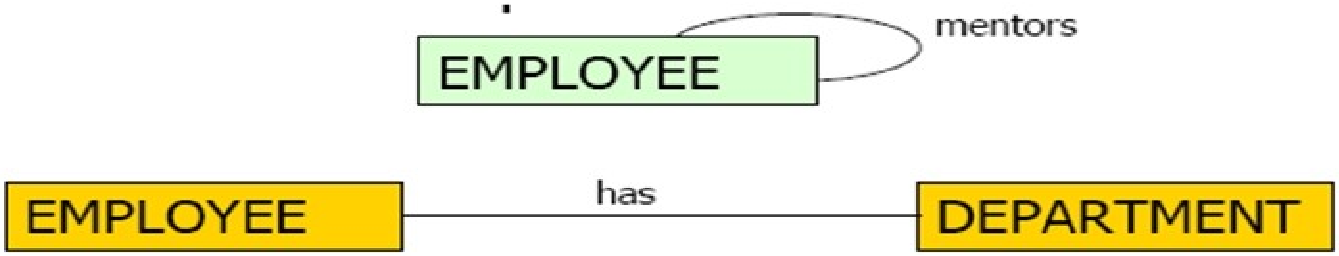






Relationship Degree

- The degree of a relationship is the number of entity classes that participate in the relationship



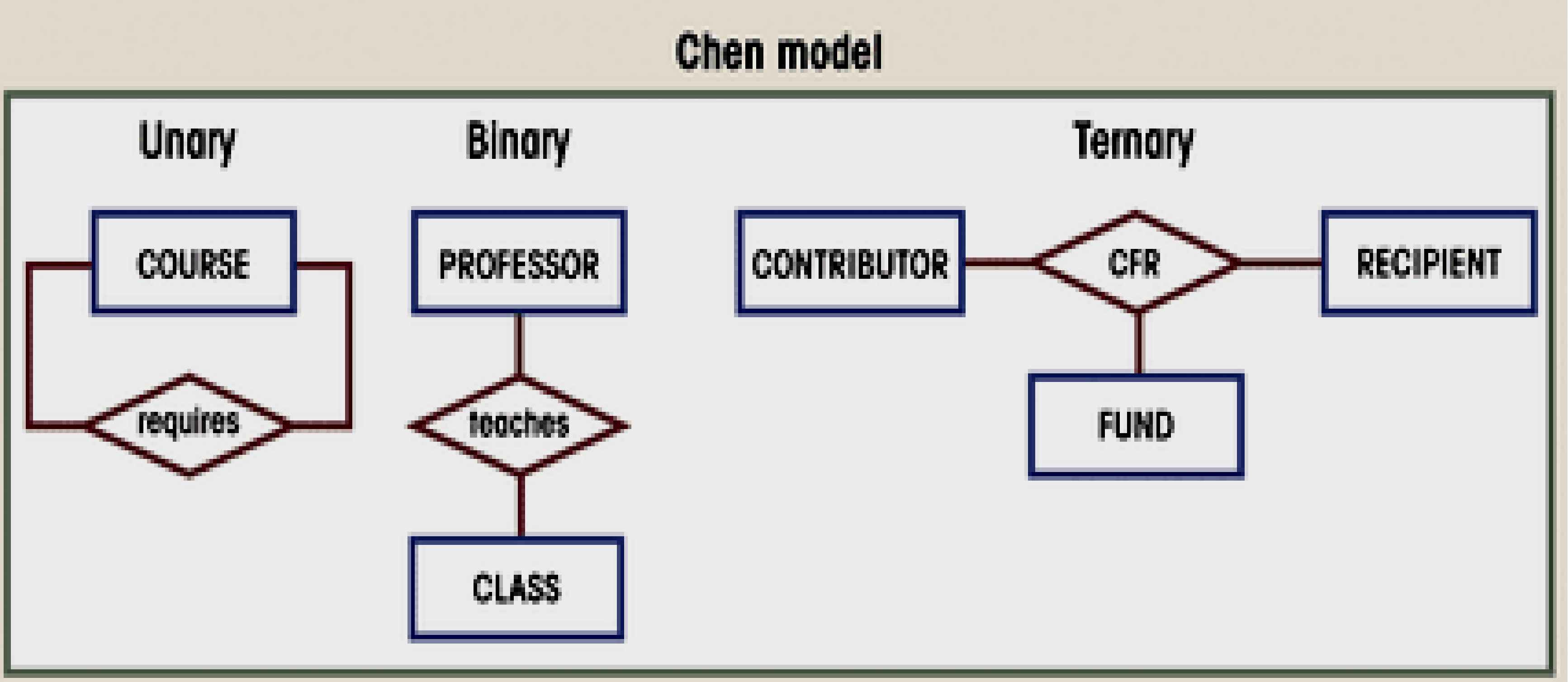
Relationship Degree

- Indicates number of associated entities
- Unary
  - Single entity
  - Recursive
  - Exists between occurrences of the same entity set

Relationship Degree

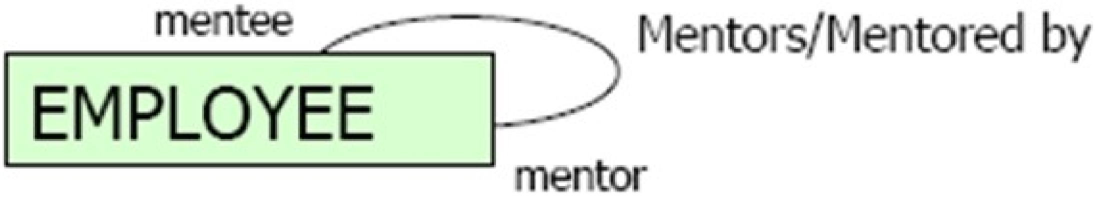
- Binary
  - Two entities associated
  - Relationship between two different entities
- Ternary
  - Three entities associated
  - Relationship between three different entities

Three Types of Relationships



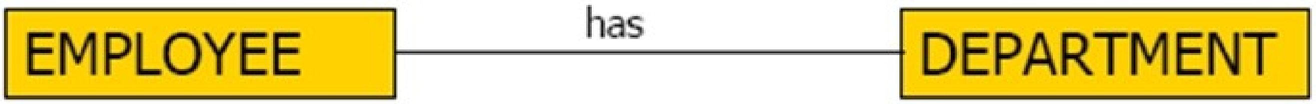
Relationships: Unary (Recursive)

- A relationship among entity instances of the same type
  - The same entity participates more thatn once in differnet roles
- A recursive relationship will always have a degree of ________.



Relationships: Binary

- A relationship between exactly two entities or tables
- A binary relationship will always have a degree of ___________.



In-Class Exercise (ICE) #3

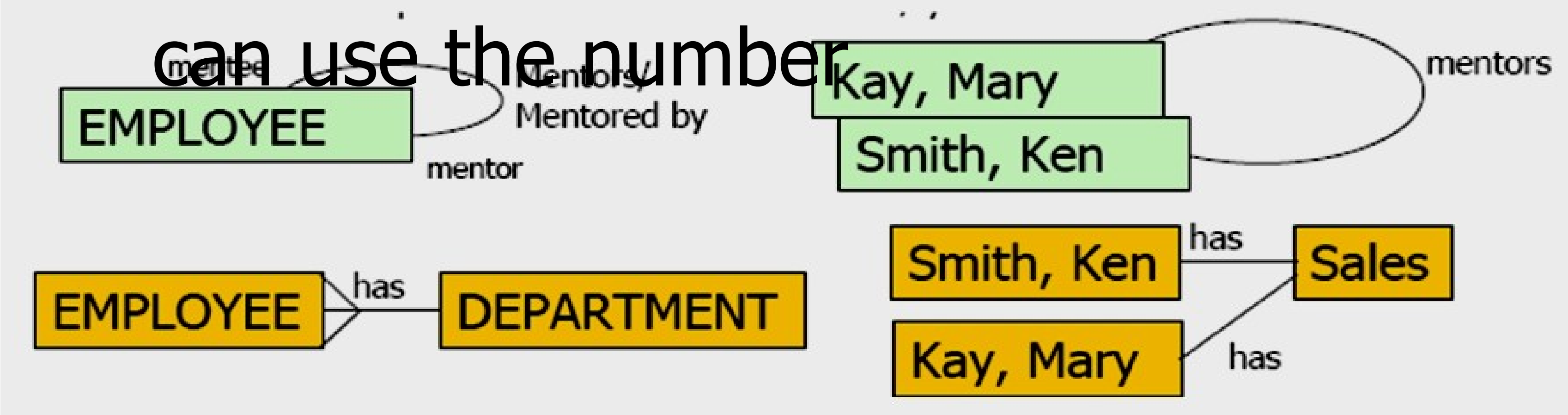
- DataHouse Reality
- Add relationship lines to your E-R Diagram

Cardinality Ratios

- Def. Number of relationship instances that an entity instance can at most participate in
- Expresses number of entity occurrences associated with one occurrence of related entity
- Based on the relationship types, four cardinality ratios are possible
  - 1:1, 1:N, N:1, N:M

Maximum Cardinality

- The maximum number of entity instances that can occur on one side of a relationship
- If a specific maximum is known, you can use the number

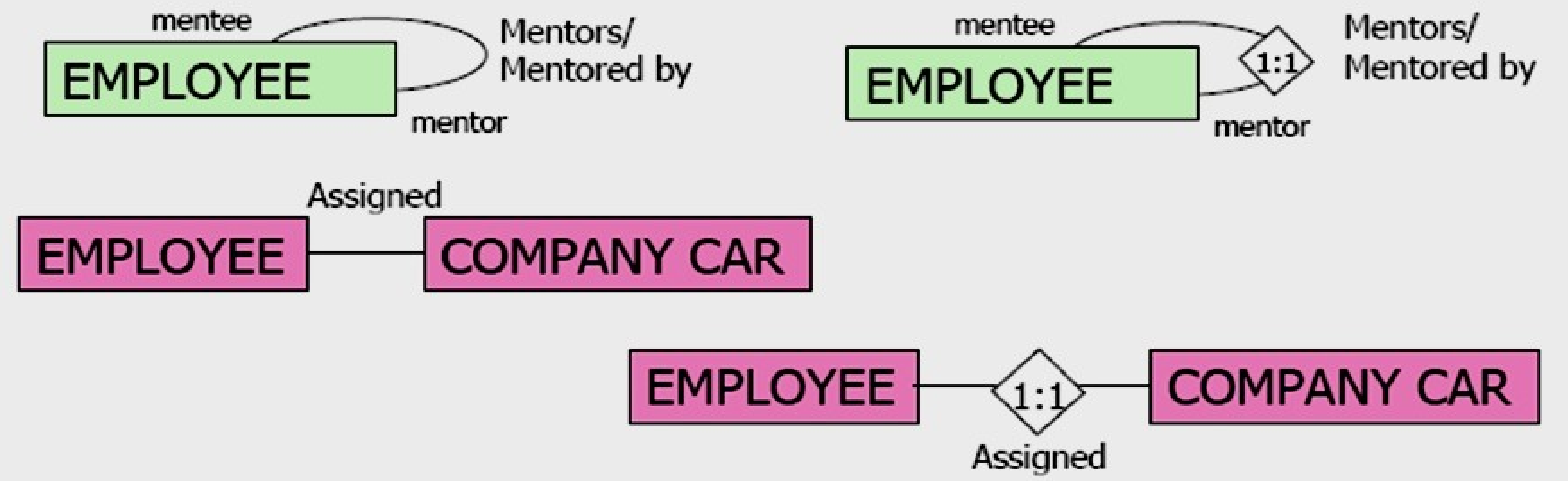


Cardinality Ratios 1:1

- One entity instance to One entity instance

Crow’s Feet

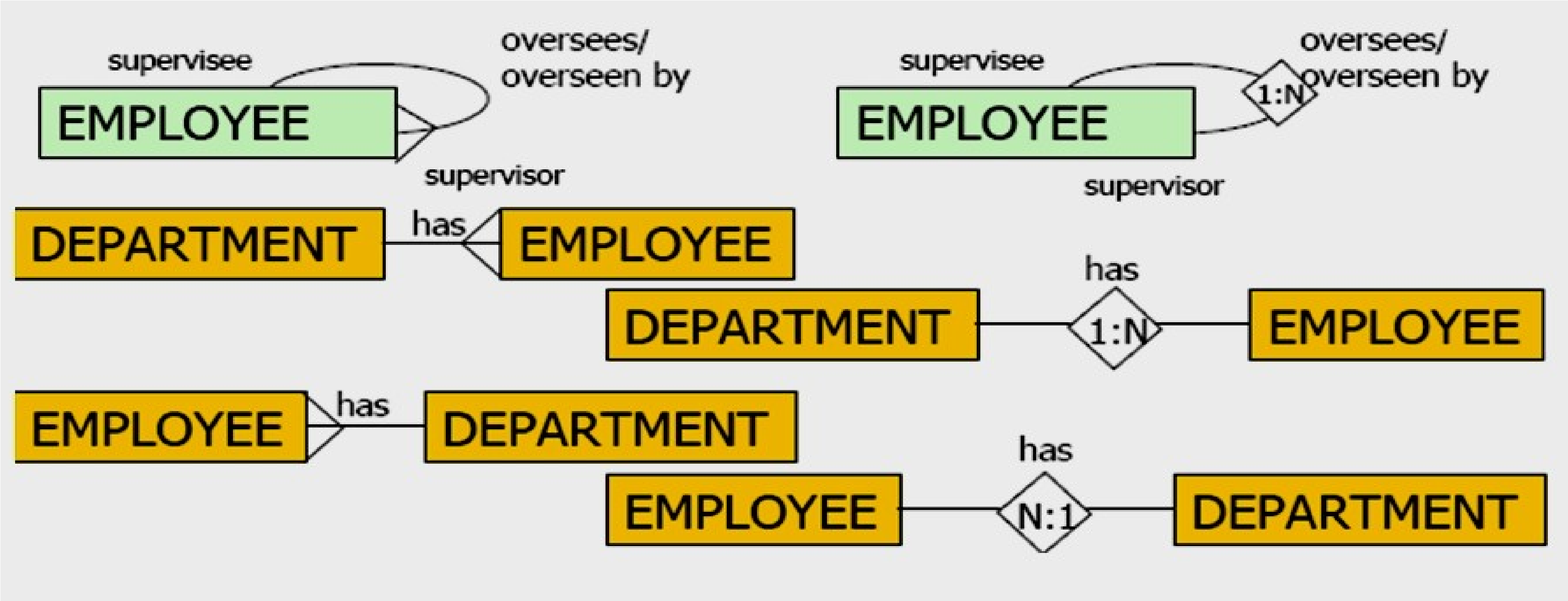
Diamonds



Cardinality Ratios 1:N (N:1)

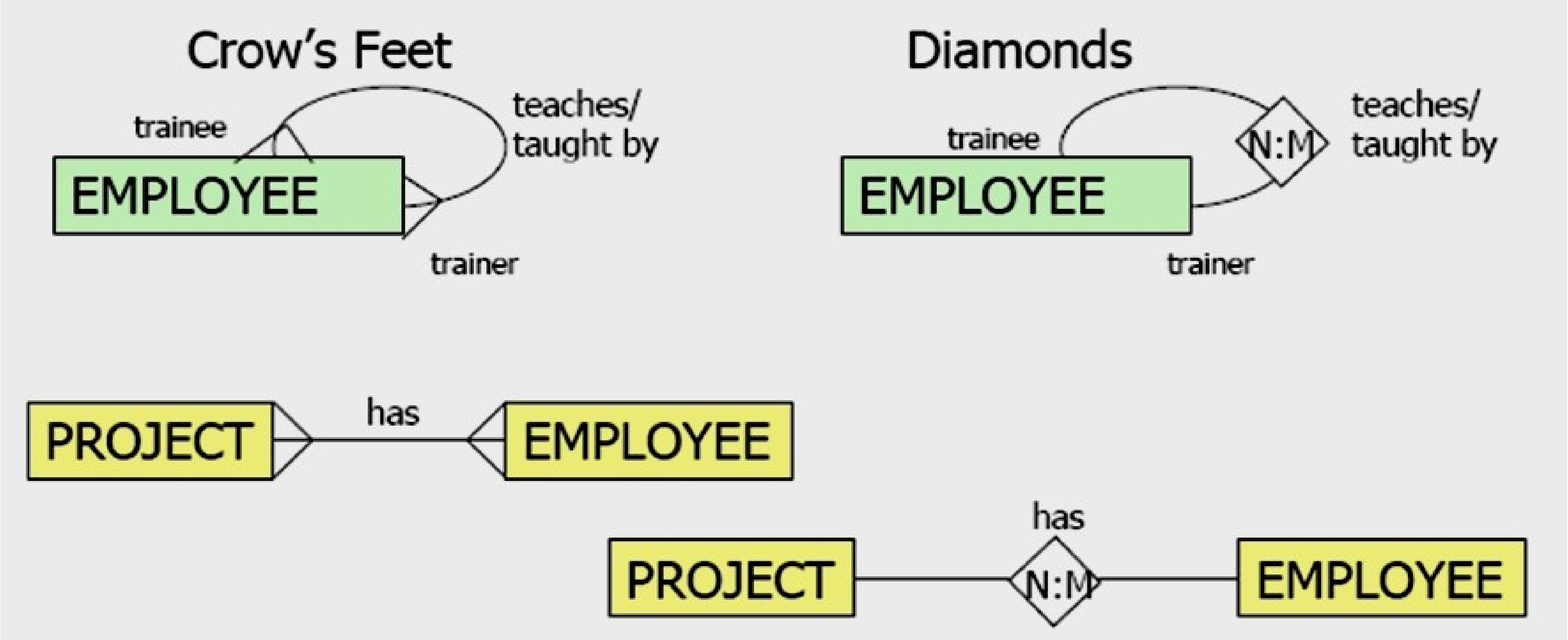
- One entity instance to Many entity instances

Crow's Feet Diamonds



Cardinality Ratios N:M

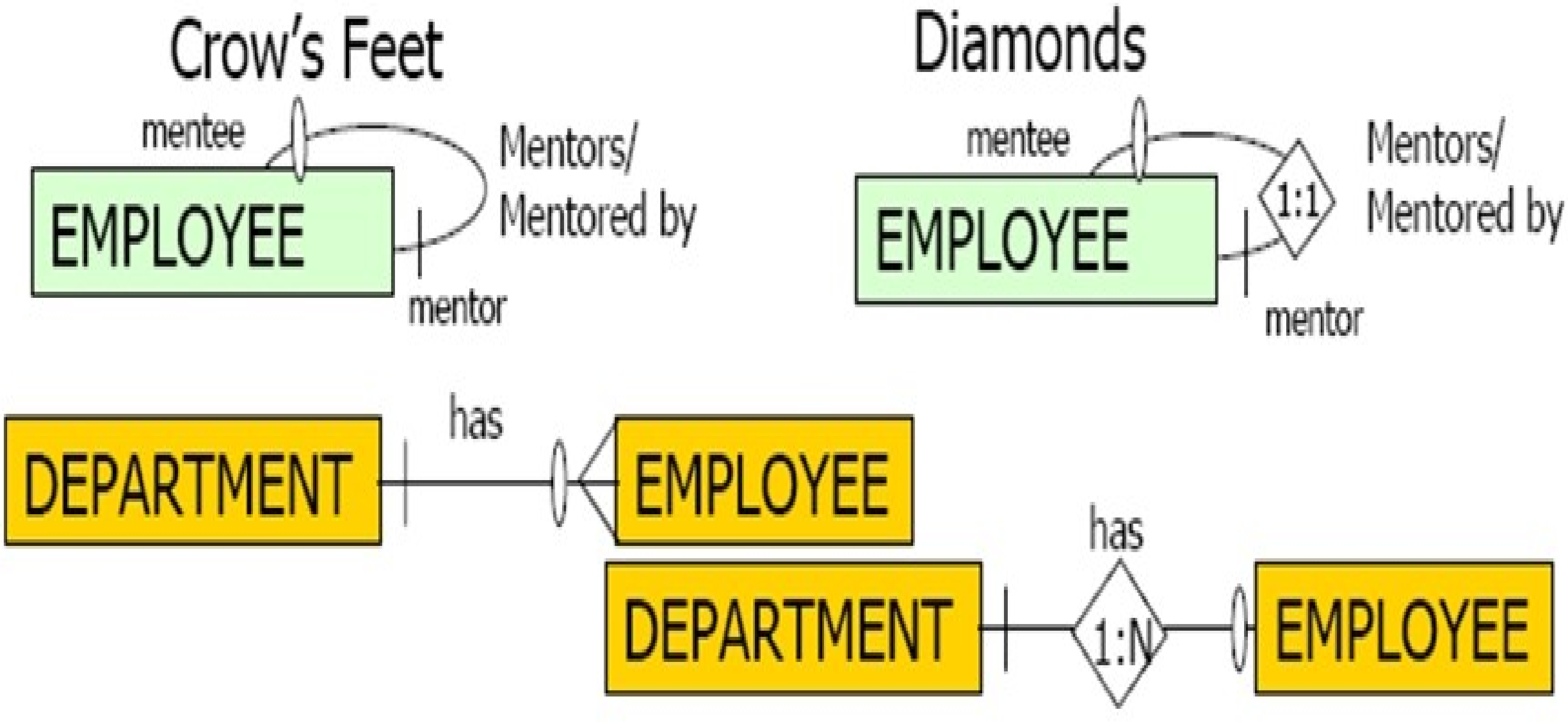
- Many entity instances to Many entity instances



Minimal Cardinality

- The minimum number of entity instances that have to occur on one side of a relationship
- ‘ – ‘Hash Mark: denotes that one entity instance must participate in the relationship (MANDATORY)
- ‘ 0 ‘Oval Mark: denotes that an entity instance doesn’t have to participate in the relationship (OPTIONAL)

Minimal Cardinality Example



Minimal Cardinality

- Optional
  - Entity occurrence does not require a corresponding occurrence in related entity
  - Shown by drawing a small circle on side of optional entity on ERD

Minimal Cardinality

- Mandatory
  - Entity occurrence requires corresponding occurrence in related entity
  - If no optionality symbol is shown on ERD, it is mandatory

In-Class Exercise (ICE) #4

- DataHouse Reality
  - Go through and add the minimum and maximum cardinalities to your relationships