Databases I

Etelka Szendrői Dr. (PhD)
associate professor
System and Software Technology Department

szendroi@mik.pte.hu

Data models

Data model: mathematical formalism, which is suitable for data-oriented description of reality

Components:

- *Structural part:* is used to describe the data types and their relationships found in reality
- *Operational part:* containing different query and modification activities
- *Integrity part:* constraints data types, data values, and connections in the database, and determines actions that can be performed

2

Types of Database Models

- By abstraction level:
  - *Semantic data models*
    - Human, highlights the point, inaccurate
  - *Database data models*
    - Machine close, give details, complete
- There are several models with different tools on each level

3

Database model levels

- A Conceptual model represents reality in an abstracted form that can be used in developing an information system in a wide variety of formats (e.g. relational, object-oriented, flat-file, etc.)
  - It is hardware and software independent
  - It is independent of any logical model type
- A Logical model represents reality in the format required by a particular database model (e.g. relational or object-oriented)
  - Is still hardware and software independent
  - Depends on the chosen logical model type
- A Physical model is created specifically for a particular database software package
  - Is dependent on hardware, software, and on the chosen logical model type

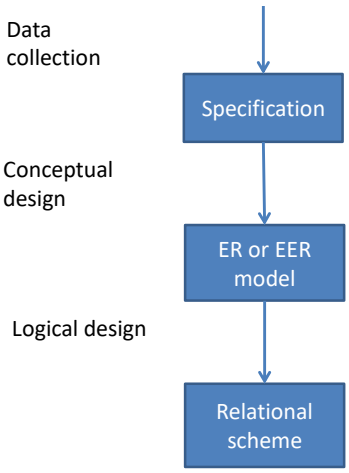
4

Main steps in Database design

- Needs assessments and analysis
- Preparing a conceptual database model
- Selecting a DBMS system
- Converting the conceptual model to a database data model
- Designing a physical data model
- Database implementation

5

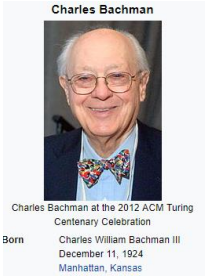
Modeling process



6

Entity-Relationship model

- Simplified semantic data model : ER
- Created by: Chen(1976)
- It contains only structural and elemental static integrity elements
- Not a complate model, but simplicity is widespread, and is still widely used today
- Design tool for relational database design
- Other researchers have also developed similar technics for example: Bachman, Martin
- IDEFX standards are also common



- There are 3 main components:
  - Entity types (all individual employees are instances of generic Employee entity type)
  - Relationships
  - Attributes (properties)

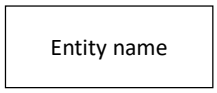
Entities

- is a real-world object distinguishable or unique from other objects.
- An entity can be a **concrete** or **physical** object like *employee*, *student*, *faculty*, *customer* etc. Or it could also be **conceptual** or **abstract** like *transaction*, *order*, *course*, *subjects* etc.
- Refers to entity set and not to single entity occurrence
- In Chen model, entity is represented by rectangle with entity's name
- The entity name, a noun, is written in capital letters

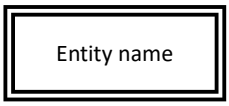
8

Entities

- Types of Entities:
  - Regular Entity :identifiable itself (Students, Cars)
  - Weak Entity: identified through contact with other individuals (employee's dependents, car engine)
    - A weak entity is an entity that is existence – dependent on some other entity in the sense that it cannot exist if that other entity does not exist. For example, if a given employee is deleted, all dependents of that employee must be deleted also.
- Symbols of entities:



Regular entity



Weak entity

9

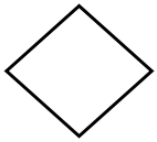
Attributes

- Characteristics of entities
- Chen notation: attributes represented by ovals connected to entity rectangle with a line
  - Each oval contains the name of attribute it represents

10

Chen Notation - Symbol

Rectangle represents an **Entity**



Diamond represents a **Relationship**

Peter Chen 1976

Lines with labels represents **Cardinality**

Oval represents properties

11

Symbols used in E-R diagrams

Meaning	Symbol
Entity	Entity
Attribute	Attribute
Key Attribute	Key Attribute
Relationship	Relationship

12

Type of Attributes

- Simple attribute cannot be subdivided (unit price, length)

- Composite attribute can be subdivided (address)

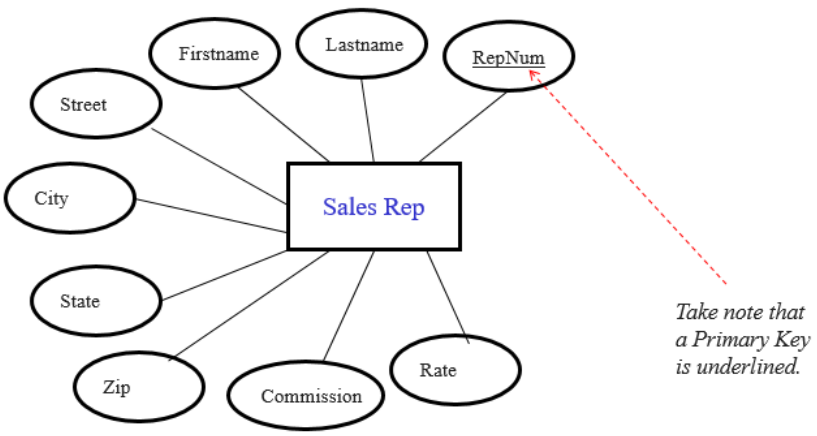
- Key property unique and identifies the entity instance

- Multivalued attributes can have many value

- Derived property (calculated value)

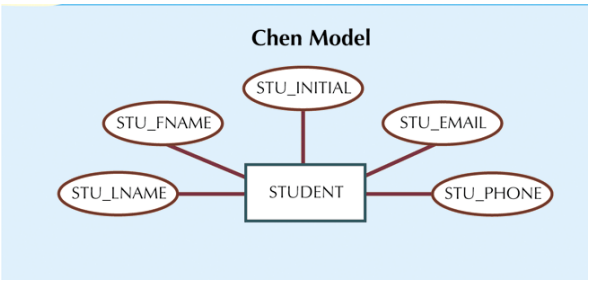
13

Attributes - example



14

The attributes of STUDENT entity



15

Keys

- A **super key** of an entity set is a set of one or more attributes whose values uniquely determine each entity.
- A **candidate key** of an entity set is a minimal super key
  - *ID* is candidate key of *instructor*
  - *course_id* is candidate key of *course*
- Although several candidate keys may exist, one of the candidate keys is selected to be the **primary key**.

16

Attributes (cont'd.)

- **Required** attribute: must have a value
- **Optional** attribute: may be left empty
- **Domain**: set of possible values for an attribute
  - Attributes may share a domain
- **Identifiers**: one or more attributes that uniquely identify each entity instance (Primary key)
- **Composite identifier**: primary key composed of more than one attribute

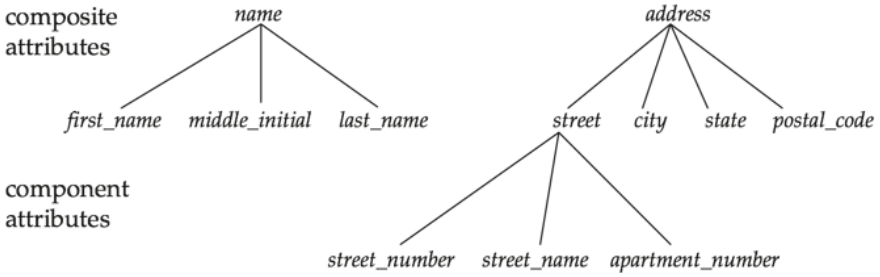
17

Attributes (cont'd.)

- **Composite attribute** can be subdivided
- Simple attribute cannot be subdivided
- Single-value attribute can have only a single value
- Multivalued attributes can have many value

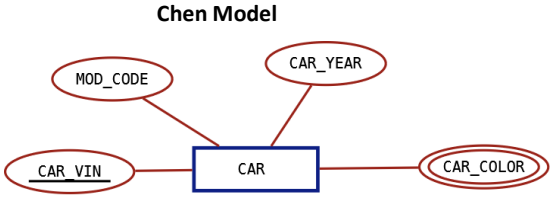
18

Composite attribute

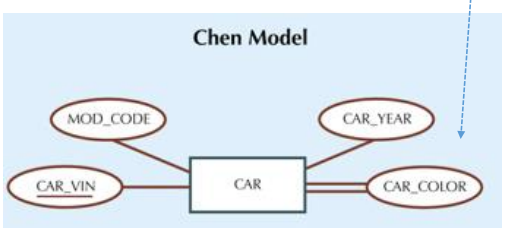


19

A multivalued attribute in an entity



OR



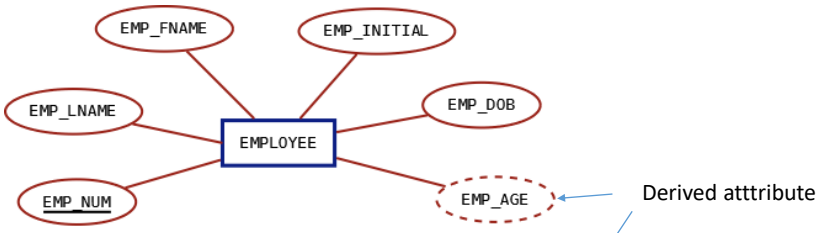
20

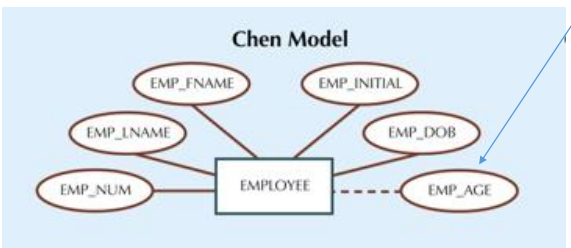
Attributes (cont'd.)

- M:N relationships and multivalued attributes should not be implemented
  - Create several new attributes for each of the original multivalued attributes' components
  - Create new entity composed of original multivalued attributes' components
- Derived attribute: value may be calculated from other attributes
  - Need not be physically stored within database

21

Depiction of a derived attribute





22

Derived attributes

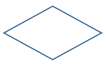
Advantages and Disadvantages of Storing Derived Attributes

	DERIVED ATTRIBUTE	
	STORED	NOT STORED
Advantage	Saves CPU processing cycles Saves data access time Data value is readily available Can be used to keep track of historical data	Saves storage space Computation always yields current value
Disadvantage	Requires constant maintenance to ensure derived value is current, especially if any values used in the calculation change	Uses CPU processing cycles Increases data access time Adds coding complexity to queries

23

Relationships

- Association between entities
- Participants are entities that participate in a relationship
- Relationships between entities always operate in both directions
- Relationship can be classified as 1:M
- Relationship classification is difficult to establish if only one side of the relationship is known
- It is normally represented by a diamond shape.



24

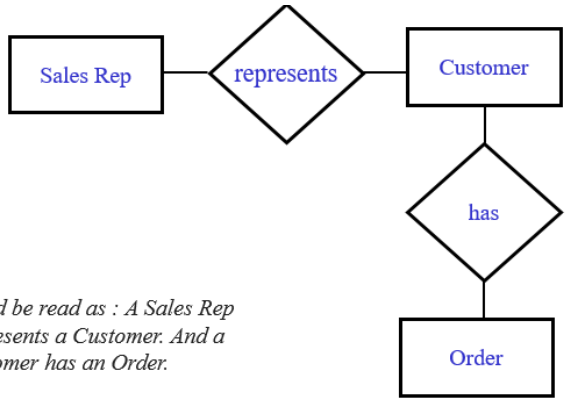
Connectivity and Cardinality

- Connectivity
  - Describes the relationship classification
- Cardinality
  - Expresses minimum and maximum number of entity occurrences associated with one occurrence of related entity
  - An entity in a relationship with minimum cardinality of zero plays an **optional role** in the relationship
  - An entity with a minimum cardinality of one plays a **mandatory role** in the relationship
- Established by very concise statements known as business rules

25

Relationship - Example

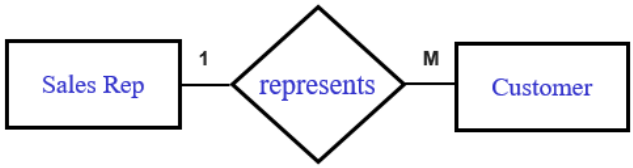
- For example in a Sales system we have this relationships among entities:



Could be read as : A Sales Rep Represents a Customer. And a Customer has an Order.

26

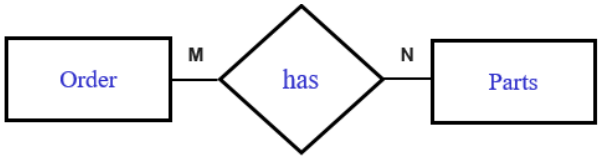
Cardinality Symbols - Example



Could be read as : A Sales Rep could represent 1 or Many Customers.

27

Cardinality Symbols - Example



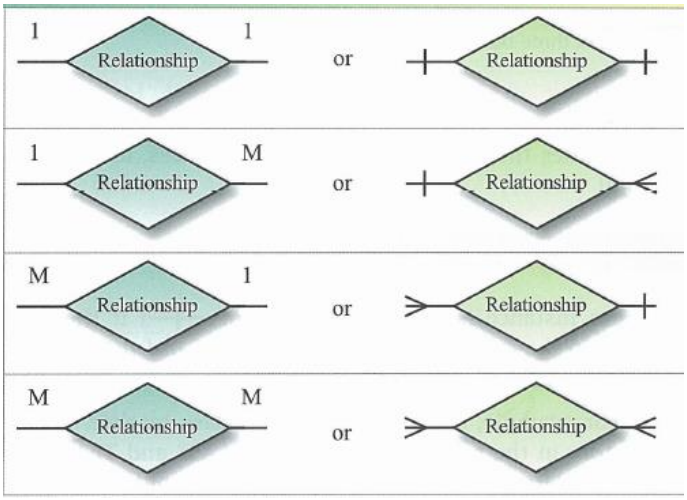
Could be read as : An Order could have many Parts (e.g. Products Ordered) and a Part could have many Orders.

28

Symbols of relationships

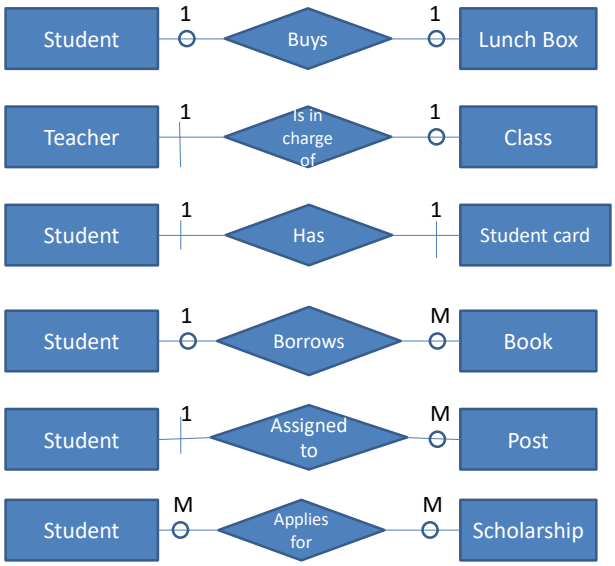
CHEN modell

Crow's foot jelölés (Martin)



29

Sample relationships



One-to-one relationship; Optional instances on both sides

One-to-one relationship; Optional instance on one side, mandatory instance on the other side.

One-to-one relationship; Mandatory instances on both sides.

One-to-Many relationship; optional instances on both sides

One-to-Many relationship; optional instance on one side

Many-to-many relationship; Optional instances on both sides

30

Relationship Participation

- Optional participation
  - One entity occurrence does not require corresponding entity occurrence in particular relationship
- Mandatory participation
  - One entity occurrence requires corresponding entity occurrence in particular relationship

31

Crow's foot Symbols of cardinality

CROW'S FOOT SYMBOLS	CARDINALITY	COMMENT
	(0,N)	Zero or many; the "many" side is optional.
	(1,N)	One or many; the "many" side is mandatory.
	(1,1)	One and only one; the "1" side is mandatory.
	(0,1)	Zero or one; the "1" side is optional.

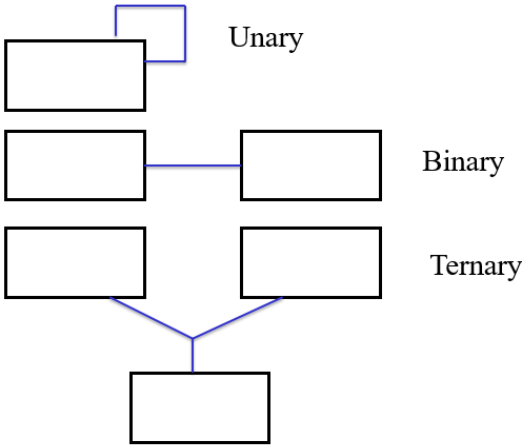
32

Relationship Degree

- Indicates number of entities or participants associated with a relationship
- Unary relationship
  - Association is maintained within single entity
- Binary relationship
  - Two entities are associated
- Ternary relationship
  - Three entities are associated

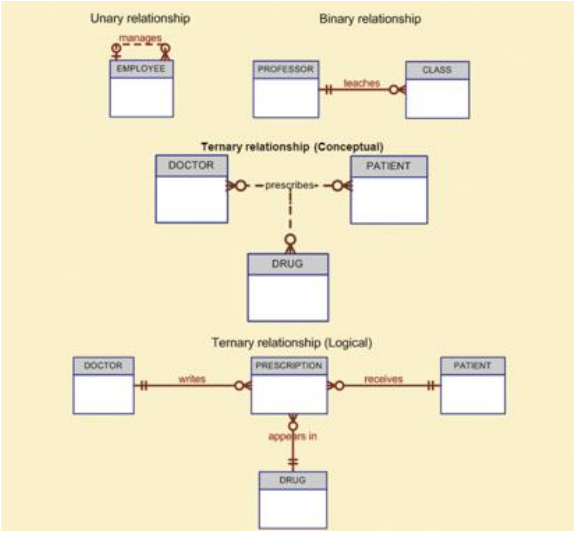
34

Degree of Relationship



35

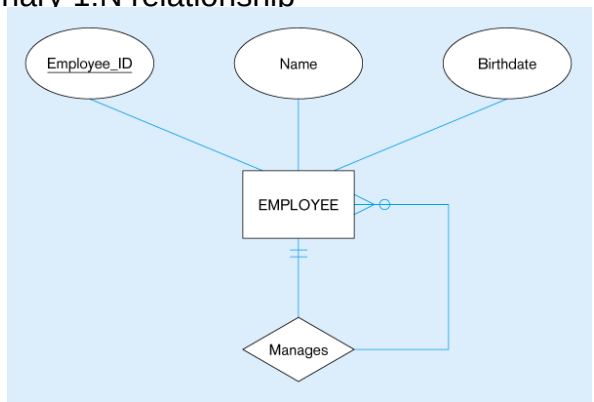
Degree of Relationship with Crow's Foot Symbols



36

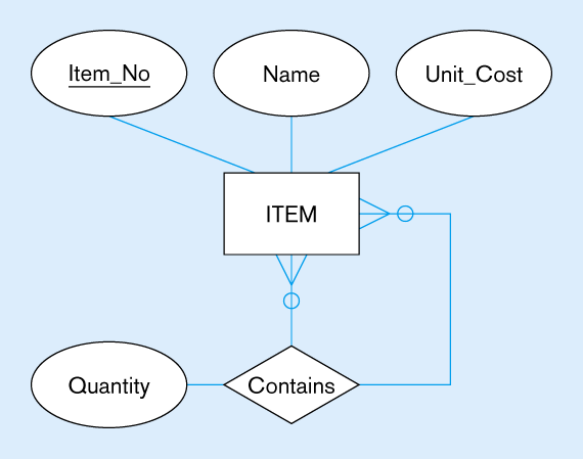
Recursive Relationships

- Relationship can exist between occurrences of the same entity set
  - Naturally found within unary relationship. A sample unary 1:N relationship



37

Unary M:N relationship



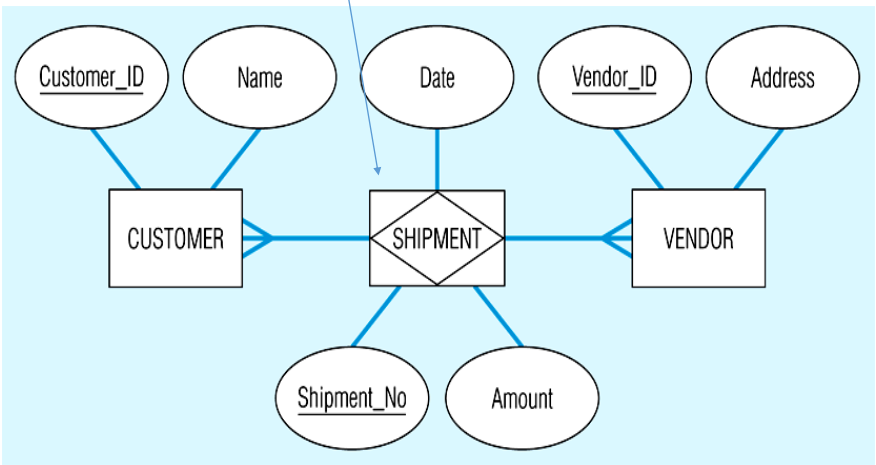
38

Associative (Composite) Entities

- Also known as bridge entities
- Used to implement M:N relationships
- Composed of primary keys of each of the entities to be connected
- May also contain additional attributes that play no role in connective process
- Converting the M:N relationship into two 1:N relationships with it

39

Associative entity



40

Developing an ER Diagram

- Database design is an iterative process
  - Create detailed narrative of organization's description of operations
  - Identify business rules based on description of operations
  - Identify main entities and relationships from business rules
  - Develop initial ERD
  - Identify attributes and primary keys that adequately describe entities
  - Revise and review ERD

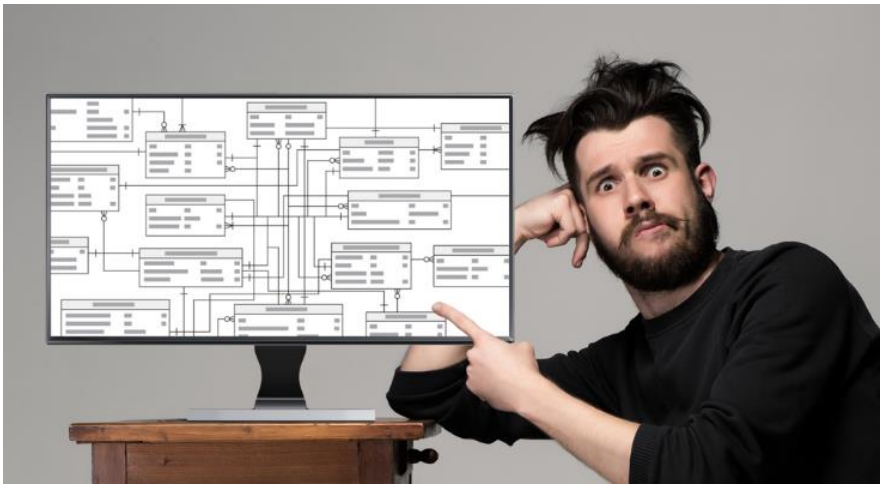
41

Summary

- Entity relationship (ER) model
  - Uses ERD to represent conceptual database as viewed by end user
  - ERM's main components:
    - Entities
    - Relationships
    - Attributes
  - Includes connectivity and cardinality notations

Summary (cont'd.)

- Connectivities and cardinalities are based on business rules
- M:N relationship is valid at conceptual level
  - Must be mapped to a set of 1:M relationships
- ERDs may be based on many different ERMs
- UML class diagrams are used to represent the static data structures in a data model
- Database designers are often forced to make design compromises



Thank you for your attention!