Entity Relationship Diagram

Professor: Mohammad Soltanieh-ha TA: Ziyuan (Ryan) Li

1. Abstract

An Entity-Relationship Diagram (ERD) is an essential tool for visualizing the components and relationships within a database, defining entities, their attributes, and their interrelations in a clear graphical format. This is particularly beneficial in relational databases, which are pivotal to information management systems and are prevalent across various sectors. By illustrating the logical architecture of databases, ERDs facilitate both the documentation of existing systems and the strategic planning of new database designs. This document presents the rules and meanings associated with ERD in the context of data modeling and explores a systematic methodology for the creation of an ERD model.

2. Introduction to Entity-Relationship Diagram

Understanding and managing the meanings attributed to data are fundamental aspects of data management. Data represents real-world facts and has value only when it is contextual of corresponding meanings. Information is a collection of data organized within a specific context for a particular purpose, capable of forming different kinds of knowledge depending on how it is combined and used, while knowledge is a set of formalized information that help us learn our surroundings. The base of knowledge engineering and management is data and information management. Data Base Management System (DBMS) provides a technical way to manage data and information.

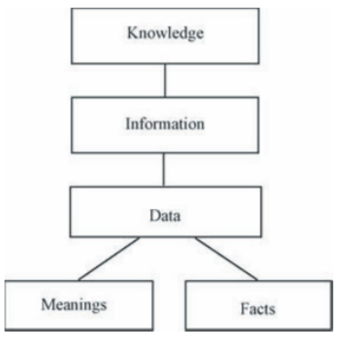
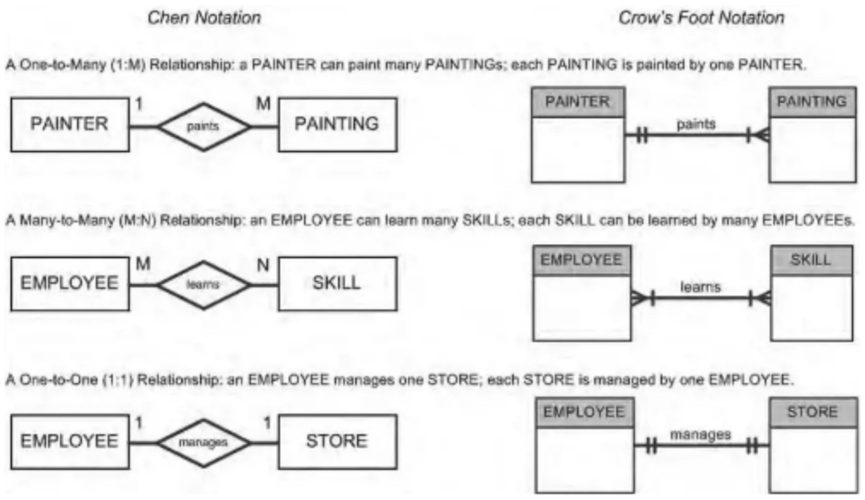


Fig. 6.1. Relationship among data, information and knowledge

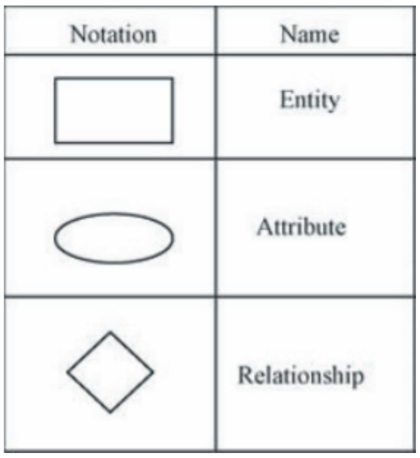
The need to define data from a conceptual standpoint has given rise to semantic data modeling techniques. These techniques generate an abstract model demonstrating how data, represented symbolically within a database, correspond to real-world entities. The actual model is commonly referred to as an Entity Relationship Diagram (ERD) because it visually represents the data by detailing the entities and their interrelationships. Most commercial Database Management Systems (DBMS) are founded on relational data models and typically begin with a preliminary ERD sketch.

3. History of Entity-Relationship Diagram

In the 1970s, computer scientist Peter Chen pioneered the first ERD concept, as outlined in his seminal 1976 paper. ERDs resolved the limitations of the existing network, relational, and entity set models by providing a clear and unified database structure. Building on the work of predecessors like Charles Bachman, Chen's model laid the groundwork for subsequent methodologies like the Unified Modeling Language (UML). Later, in the 1980s, James Martin refined Chen's ER model by introducing “Crow’s foot” symbols (also known as IE notation), to denote cardinality in database relationships.


Chen Notation
Crow's Foot Notation
A One-to-Many (1:M) Relationship: a PAINTER can paint many PAINTINGs; each PAINTING is painted by one PAINTER.
PAINTER
1
paints
M
PAINTING
PAINTER
paints
PAINTING
A Many-to-Many (M:N) Relationship: an EMPLOYEE can learn many SKILLs; each SKILL can be learned by many EMPLOYEEs.
EMPLOYEE
M
learns
N
SKILL
EMPLOYEE
learns
SKILL
A One-to-One (1:1) Relationship: an EMPLOYEE manages one STORE; each STORE is managed by one EMPLOYEE.
EMPLOYEE
1
manages
1
STORE
EMPLOYEE
manages
STORE

4. Syntax and Semantics of ERD



Notation	Name
▭	Entity
⬭	Attribute
◇	Relationship

Entity

An entity is a class of persons, places, objects, events, or concepts we need to capture and store data. An entity instance is a single occurrence of an entity. For example, each instance of the entity STUDENT represents ONE individual student in the classroom.

Attribute

An attribute is a descriptive property or characteristic of an entity. Its synonyms include element, property, and field. A key/primary attribute is the unique, distinguishing characteristic of the entity and is usually represented by an underlined attribute.

Besides the regular attributes, there are various types of attributes: multivalued attributes, derived attributes, complex attributes, etc. Since this class focuses not on DBMS, we will not dive deeper into those.

Here is a demonstration of entity and attribute:

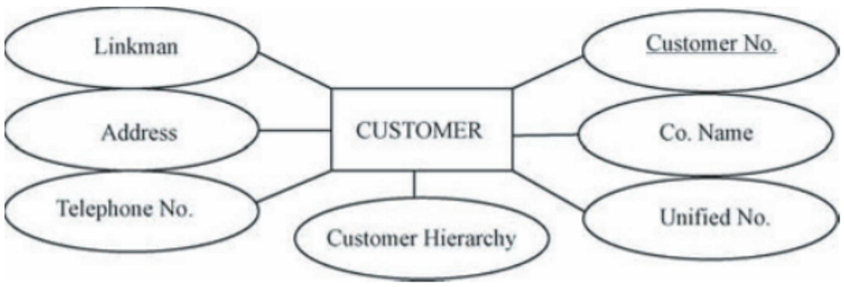


Fig. 6.5. An example of a person entity

Each customer is one entity. Each entity has attributes: Linkman, Address, Telephone No., Customer Hierarchy, Customer No., Co. Name, and Unified No. Out of all the attributes, Customer No. is the key attribute and is unique for each entity.

Relationship

A relationship is a natural business association that exists between one or more entities. The relationship may represent an event that links the entities or merely a logical affinity that exists between the entities.

A **cardinality** defines the minimum and maximum number of occurrences of one entity that may be related to a single occurrence of the other entity. Because all relationships are bidirectional, a cardinality must be defined in both directions for every relationship. There are three kinds of cardinality relationships: one-to-one, one-to-many and many-to-many. Here are some examples:

One to one:

Each wife can have one husband, and each husband can have one wife.

Many to one / many to one:

Each mother can have many kids, but each kid can have only one mother.

Many to many:

Each student can take many classes, and each class can be taken by many students.

Participation characterizes the extent to which entities are involved in a relationship, categorized as total or partial. Total participation requires every instance of an entity to be associated with another entity in the relationship, reflecting a *mandatory* link. Conversely, partial participation allows some instances of an entity to exist without being linked, indicating an *optional* relationship. Here is an example:

Full participation / Mandatory:

Each department must be managed by an employee.

Partial participation / Optional:

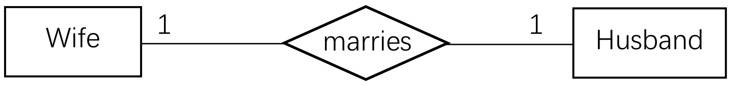
Each employee may or may not manage a department.

5. Notations

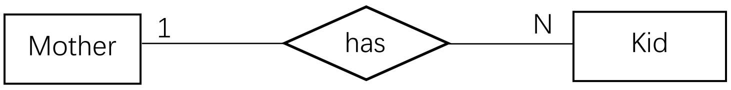
Chen's Notation

Cardinality:

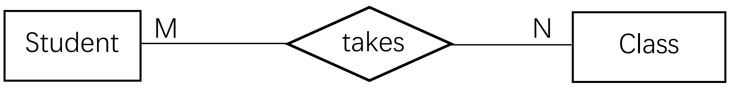
One to One:



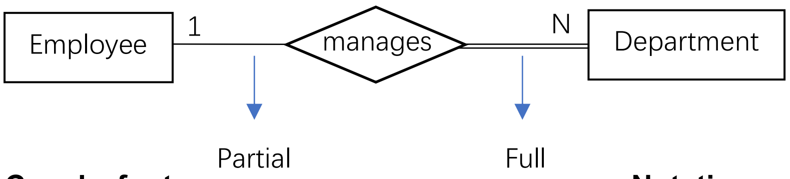
One to Many:



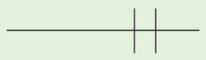
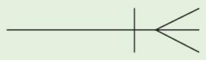
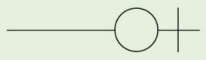
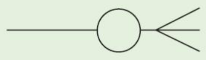
Many to Many:



Participation:

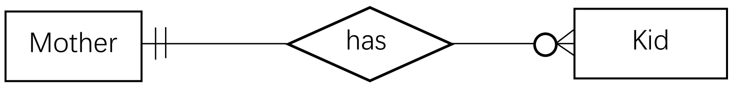


Crow's foot Notation

Symbol	Meaning	Count
	One—Mandatory	Exactly One
	Many—Mandatory	One or More
	One—Optional	Zero or One
	Many—Optional	Zero, One, or More

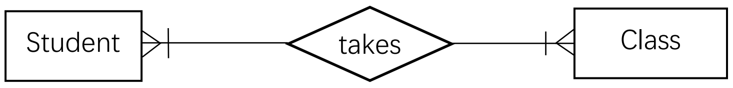
Example:

One to Many & Mandatory to Optional:



Each mother can have zero, one of more kids.
Each kid can have one and only one mother.

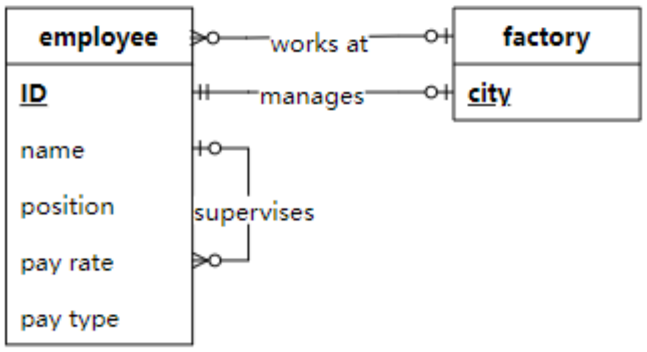
Many to Many & Mandatory to Mandatoy:

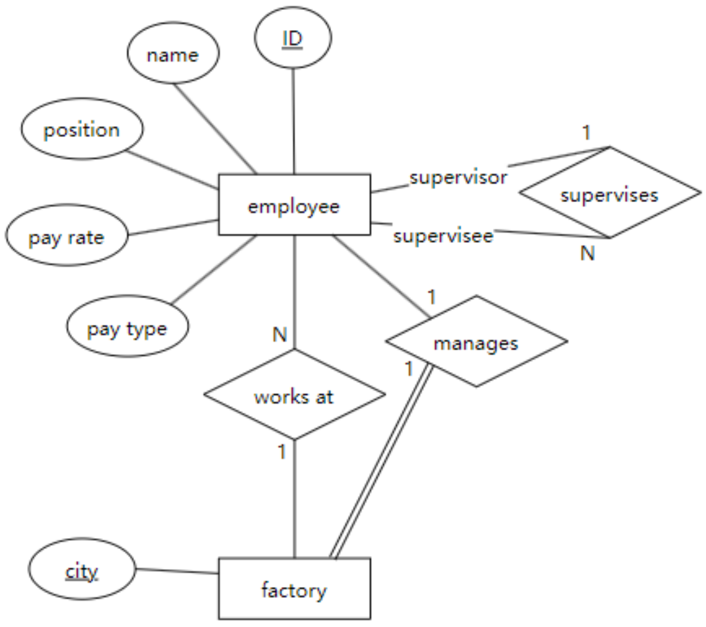


Each student can take one or more classes. (Student CANNOT take 0 class.)
Each class can be taken by one or more students. (Class CANNOT have 0 student.)

Chen's notation is valued for its conceptual clarity and academic standardization but can become cumbersome with complex databases and features a less intuitive cardinality representation than alternative notations. Crow's Foot notation, on the other hand, is favored in industry for its visual immediacy in expressing cardinality and efficiency in detailing intricate database relationships, though it can present an initial learning hurdle and risks ambiguity if not employed meticulously. The choice between these notations typically hinges on the database's complexity, the project team's familiarity, and whether the focus is on the conceptual design or the practical aspects of database implementation.

6. Let's put everything together!





(ref:https://runestone.academy/ns/books/published/practical_db/PART2_DATA_MODELING/04-other-notations/other-notations.html)

The two diagrams above essentially depict the same database. If you can clearly comprehend the relationships between entities, then you have effectively mastered this subject.

References

- Textbook Reference: Modeling and Analysis of Enterprise and Information Systems By Qing Li, Yu-Liu Chen (Link)
- SmartDraw. (n.d.). Entity-Relationship Diagram. Retrieved November 3, 2023
- RelationalDBDesign. (n.d.). IDEF1X, Crow's Foot & Chen's Model