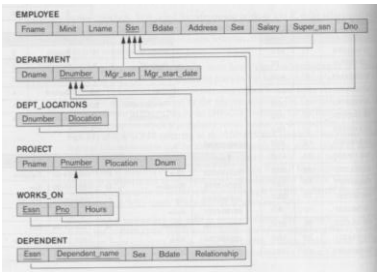
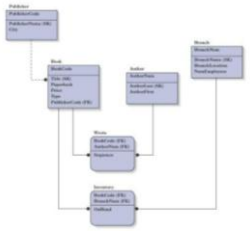
Relational Modeling I
Company database



Relational Modeling II
Henry Books



ER Modeling

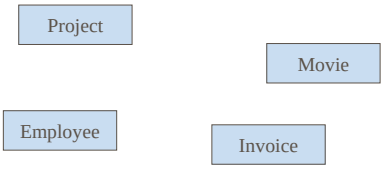
- Conceptual Model in Database Design Process
- Introduced by Peter Chen in 1976
- Centered around Entities, Attributes, and Relationships

Company Example Miniworld

Company consists of departments, which have unique name and number, and are managed by one employee. We need the start date of the manager. Department might be located in several locations. Departments control projects which have unique name number, and location.
For each employee we need name, SSN, address, salary, sex, and birth date. Employee belongs to one department, but can work on multiple projects in different departments. We need to store the weekly time spent by each employee on each project. Employees have supervisors.
We need the following information on dependents: first name, sex, birth date, and relationship to employee.

Entities

Entities are the objects (physical, or conceptual) of the model. Entity types are drawn as rectangular box.



Instances and Types

is an entity type, an “abstract” employee

John D Smith 555-55-5555

is an instance of the employee entity type

Entities, and Attributes

Attributes are properties of entities (or relationships).
They are included in the entity box

PROJECT
Name
Number
Location

Employee ?

EXCURSION: ATTRIBUTES

Attributes: Composite/Simple

Composite versus simple (atomic) attributes
(depends on miniworld)

EMPLOYEE
Name (Fname, Minit, Lname)

Attributes: Single/Multivalued

Multivalued attributes can hold multiple values simultaneously:

- Colors of a car
- Telephone number

Enclosed in curly braces: {…}.

DEPARTMENT
Number
Name
{Location}

Attributes: Stored/Derived

The value of derived attributes can be determined from stored attributes, e.g. Age from Birth Date, or other data in the schema, e.g. Number of Employees from employee.

Derived Attributes are enclosed in square brackets: […] .

DEPARTMENT
Number
Name
{Location}
[NumberofEmployees]

Attributes: Null Values

Possible meanings of null value:

- attribute does not apply (phone number to a person without telephone)
- attribute value is not known (missing)
- existence of attribute value is not known

Complex Attributes

Combination of multi-valued and composite attributes.

EMPLOYEE
Name (Fname, {MiddleName}, Lname)

Example: offices of an employee

END OF EXCURSION

Key (Identifier) Attributes

A key attribute (unique identifier) is an attribute that
- Uniquely identifies an entity
- None of its parts does
- Never contains a null value

Underline key (identifier) attributes in Entity

PROJECT
Name
Number
Location

Example: Employee

Key (Identifier) Attributes

If a key is made up of multiple attributes, introduce composite attribute and make it key.

FLIGHT
FlightNumber, Date
[NumberOfPassengers]
Destination

FLIGHT
Flight_ID (
(FlightNumber, Date)
[NumberOfPassengers]
Destination

Example: Lot (PropertyID, CountyName, Lot#)

Company Example

Design

- Employee
- Project
- Department
- Dependent

Company Example Miniworld

Company consists of departments, which have unique name and number, and are managed by one employee. We need the start date of the manager. Department might be located in several locations. Departments control projects which have unique name number, and location.

For each employee we need name, SSN, address, salary, sex, and birth date. Employee belongs to one department, but can work on multiple projects in different departments. We need to store the weekly time spent by each employee on each project. Employees have supervisors.

We need the following information on dependents: first name, sex, birth date, and relationship to employee.

Relationships

Relationship relates n entities

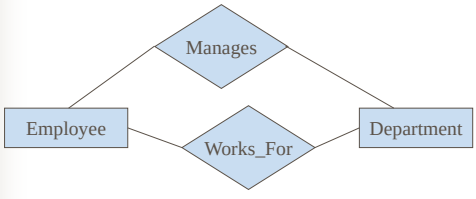
n=2: binary relation
n=3: ternary relation

Entities can appear in different **roles** in relationship

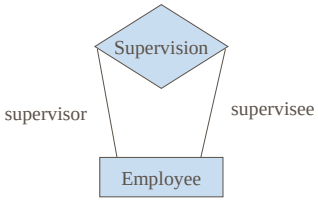
Examples:
- employee works for/manages department
- supplier supplies parts for project

Relations in Diagrams

- drawn as diamond shaped boxes (**different from book**)
- lines to participating entities
- roles on lines (if necessary)
- typically relationships are read from left to right, top to bottom



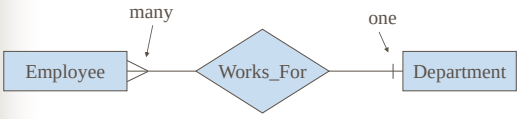
Multiple Roles Example



Example: items assembled from each other

Maximum Cardinality

many: crow's feet
one: horizontal bar



Cardinality Ratios Examples

(for binary relations)

one to one	MANAGES
one to many	CONTROLS
many to one	WORKS_FOR
many to mane	WORKS_ON

Cardinality Constraints

Examples:

- x teaches at university y
- x has SSN y
- x supplies y to z
- x speaks language y
- x is mother of y

Find examples for one to many to many
(one to one to many is usually bad design.)

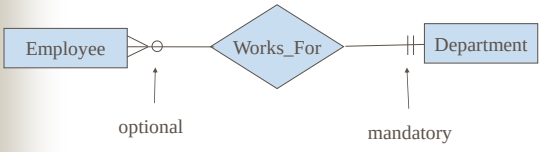
Minimum Cardinality

Mandatory (or total) participation of an entity type:
every entity must be in the relation
(depicted by a horizontal bar)

Optional (or optional) participation of an entity type:
not every entity has to be in a relation
(depicted by a 0)

Minimum Cardinality

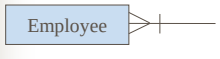
Example: Every employee works for a department



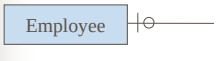
The symbols for minimum cardinality are closer to the relationship than the ones for maximum cardinality.

Cardinality Constraints

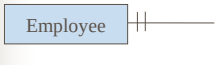
Optional many



Mandatory one



Optional one



Mandatory one

Alternative Notations

many: M
one: 1

Total participation: double line
Partial participation: single line

Example: Every employee works for a department

Alternative Notations

For more precise cardinality ratios

Example: Every employee works for a department

Example: Students enroll in at least 1 and at most 6 classes, and a class can have up to 40 students.

Henry Books Miniworld

Ray Henry, the owner of a bookstore chain named Henry Books, has decided to store his data in a database. In running his chain of bookstores, Ray gathers and organizes information about branches, publishers, authors, and books. Each branch has a number that uniquely identifies the branch. In addition, Ray tracks the branch's name, location, and number of employees. Each publisher has a code that uniquely identifies the publisher. In addition, Ray tracks the publisher's name and city. Each author has a number that uniquely identifies the author. In addition, Ray records each author's last and first names. Each book has a code that uniquely identifies the book. For each book, Ray also tracks the title, publisher, type of book, price, and whether the book is a paperback. We also need a list of authors for each book. Finally, Ray keeps track of how many copies of each book are available at each branch.