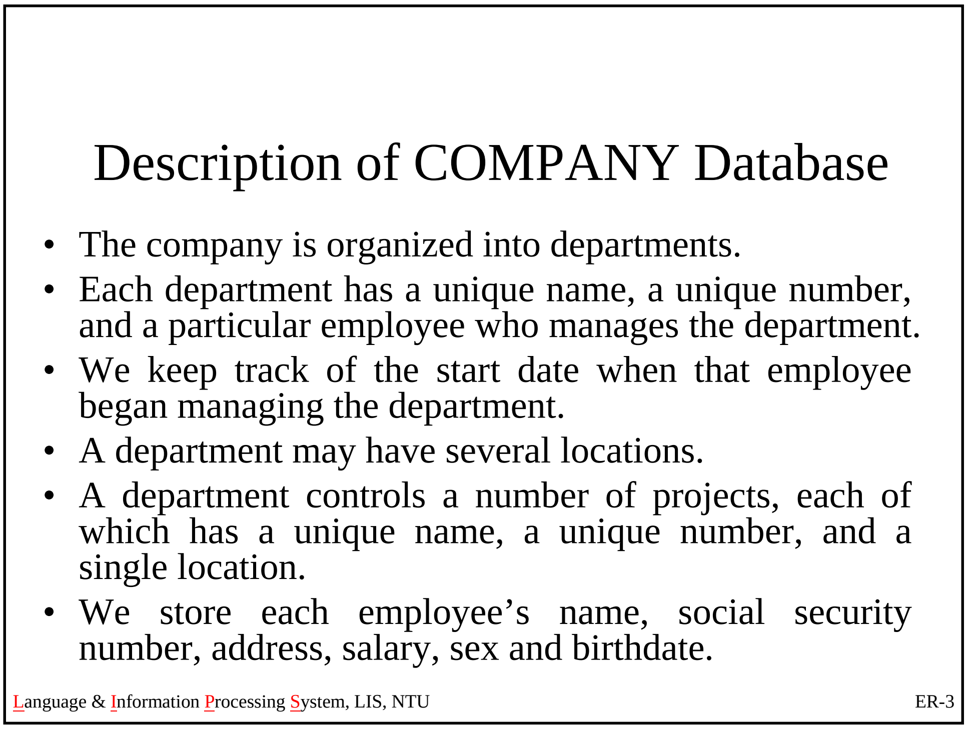
Entity-Relationship Model

Kuang-hua Chen
Department of Library and Information Science
National Taiwan University

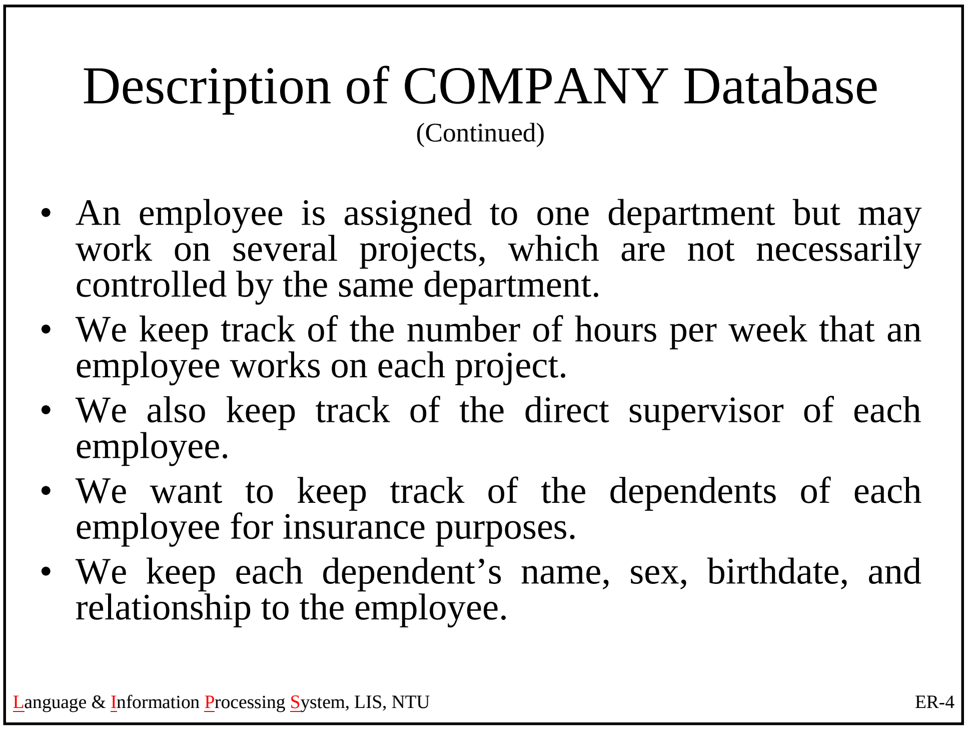
A Company Database

- Keeps track of a company’s employees, departments and projects
- After the requirements collection and analysis phase
  - designers stated some description of the “miniworld”
  - the part of the company to be represented



Description of COMPANY Database

- The company is organized into departments.
- Each department has a unique name, a unique number, and a particular employee who manages the department.
- We keep track of the start date when that employee began managing the department.
- A department may have several locations.
- A department controls a number of projects, each of which has a unique name, a unique number, and a single location.
- We store each employee's name, social security number, address, salary, sex and birthdate.

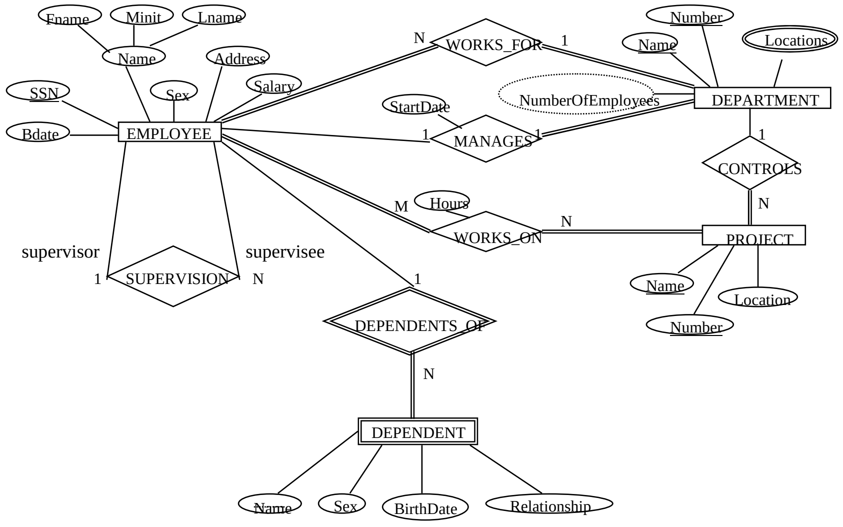


Description of COMPANY Database

(Continued)

- An employee is assigned to one department but may work on several projects, which are not necessarily controlled by the same department.
- We keep track of the number of hours per week that an employee works on each project.
- We also keep track of the direct supervisor of each employee.
- We want to keep track of the dependents of each employee for insurance purposes.
- We keep each dependent's name, sex, birthdate, and relationship to the employee.

ER schema diagram for the COMPANY database



Entities and Attributes

- **Entity**, which is a “thing” in the real world with an independent existence.
  - an object with a physical existence -- a particular person, car, house, or employee
  - an object with a conceptual existence -- a company, a job, or a university course.
- Each entity has particular properties, called **attributes**
  - employee entity may be described by the employee’s name, age, address...

Entities and Attributes (Continued)

- A particular entity will have a **value** for each of its attributes
- The attribute values that describe each entity become a major part of the data stored in the database.

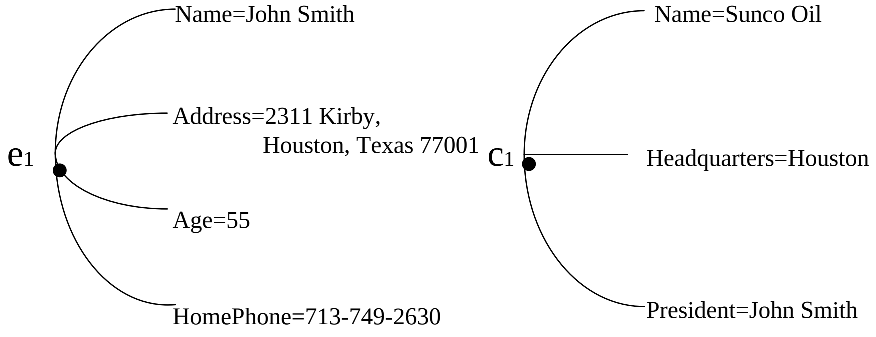
Types of Attribute

- **Composite attributes** can be divided into smaller subparts, which represent more basic attributes with independent meanings of their own.
- Attributes that are not divisible are called **simple** or **atomic** attributes.
- Most attributes have a single value for a particular entity; such attributes are called **single-valued**.
- In some cases an attribute can have a set of values of the same entity, such attributes are called **multivalued**.

Types of Attribute (Continued)

- In some cases two (or more) attribute values are related -- for example, the Age attribute is hence called a **derived attribute** and is said to be **derivable from** the BirthDate attribute, which is called a **stored attribute**.
- A **null** value for an attribute:
  - A particular entity may not have any applicable value for an attribute.
  - Null can also be used if we do not know the value of an attribute for a particular entity.

Sample Entities with Attribute Values



e_1

Name=John Smith

Address=2311 Kirby,
Houston, Texas 77001

Age=55

HomePhone=713-749-2630

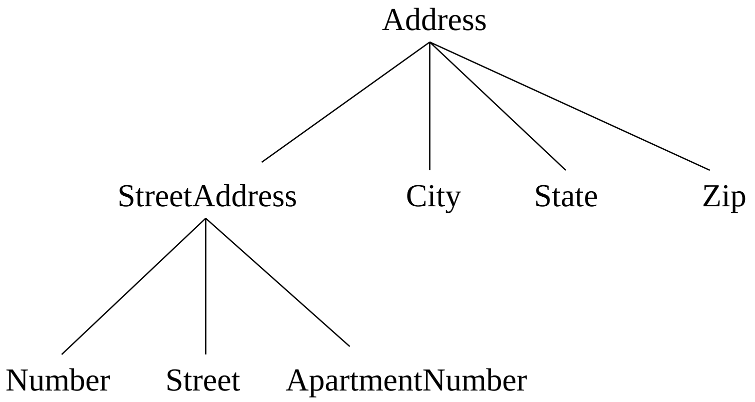
c_1

Name=Sunco Oil

Headquarters=Houston

President=John Smith

A Hierarchy of Composite Attributes

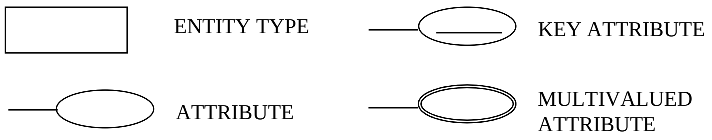

Address

StreetAddress City State Zip

Number Street ApartmentNumber

Entity Types

- An **entity type** defines a set of entities with the same attributes.
  - – EMPLOYEE and COMPANY


ENTITY TYPE

KEY ATTRIBUTE

ATTRIBUTE

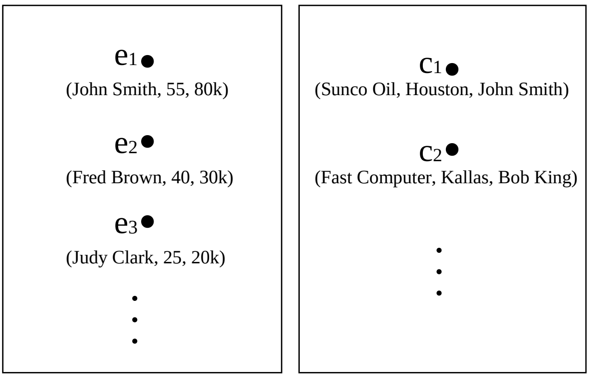
MULTIVALUED ATTRIBUTE

- An entity type describes the **schema** or **intension**
- The individual entities of a particular entity type are grouped into a **collection** or **entity set** (**extension**)

Two Entity Types and Member Entities

ENTITY TYPE NAME:	EMPLOYEE	COMPANY
ATTRIBUTES:	Name, Age, Salary	Name, Headquarters, President

ENTITY SET:
(EXTENSION)



Key Attributes

- An entity type has an attribute whose values are distinct for each individual entity
- Its value can be used to identify each entity uniquely
- e.g. Name attribute for COMPANY, SocialSecurityNumber for PERSON
- Sometimes, several attributes together form a key
- Key attribute must hold for all extensions
- Some entity types have more than one key attribute

The CAR Entity Type

CAR

Registration(RegistrationNumber, State), VehicleID, Make, Model, Year, {Color}

car_1 ●

((ABC 123,TEXAS), TK629, Ford Mustang, convertible, 1989,{red, black})

car_2 ●

((ABC 123,NEW YORK), WP9872, Nissan Sentra, 2-door, 1992, {blue})

car_3 ●

((VSY 720, TEXAS), TD729, Chrysler LeBaron, 4-door, 1993, {white, blue})

Value Sets

- A **value set** (or **domain**) specifies the set of values that may be assigned to that attribute
  - – the value set for Age attribute of EMPLOYEE is 16 to 70
- Value sets are not displayed in ER diagram
- The value set V for an attribute A of entity type E can be defined as a function from E to the power set of V $\mathbf{A}:\mathbf{E}\rightarrow\mathbf{P}(\mathbf{V})$

Initial Conceptual Design for COMPANY Database

- An entity type DEPARTMENT with attributes Name, Number, Locations, Manager, and ManagerStartDate. Locations is the only multivalued attribute.
- We can specify that each of Name and Number is a key attribute, because each was specified to be unique.
- An entity type DEPENDENT with attributes Employee, DependentName, Sex, BirthDate, and Relationship (for the employee).

Initial Conceptual Design for COMPANY Database (Continued)

- An entity type PROJECT with attributes Name, Number, Location, and ControllingDepartment. Each of Name and Number is a key attribute.
- An entity type EMPLOYEE with attributes Name, SSN, Sex, Address, Salary, BirthDate, Department, and Supervisor. Both Name and Address may be composite attributes
- We must go back to the users to see if any of them will refer to the individual components of Name -- FirstName, MiddleInitial, LastName -- or of Address.

Preliminary Design of Entity Types

DEPARTMENT

Name, Number, {Locations}, Manager, ManagerStartDate

PROJECT

Name, Number, Location, ControllingDepartment

EMPLOYEE

Name(Fname, Minit, Lname), SSN, Sex, Address, Salary,
BirthDate, Department, Supervisor, {WorksOn(Project, Hours)}

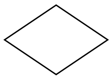
DEPENDENT

Employee, DependentName,Sex, BirthDate, Relationship

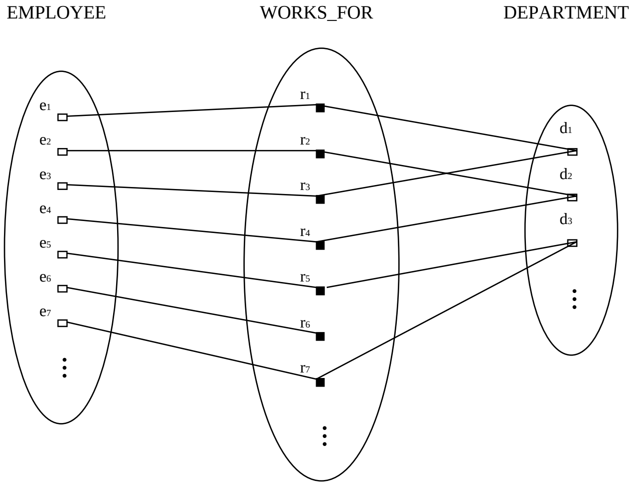
Relationships

Relationship Types and Relationship Instances:

- A **relationship type** R among n entity types E_1,E_2,...,E_n defines a set of associations among entities from these types.
- R is a set of **relationship instances** r_i, where each r_i associates n entities (e_1,e_2,...,e_n), and each entity e_j in r_i is a member of entity type E_j, $1 \leqq j \leqq n$.

ER Model:  RELATIONSHIP TYPE

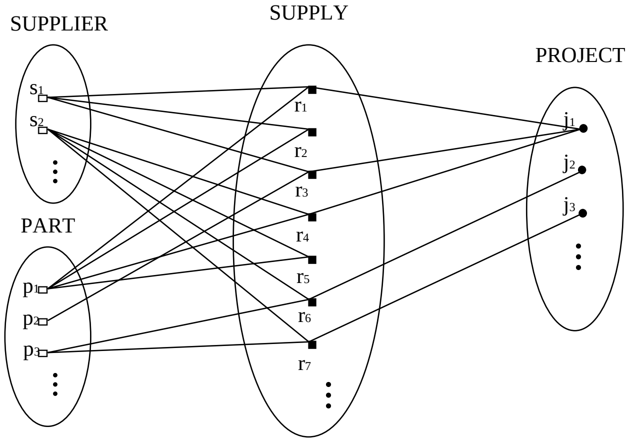
Instances of the WORKS_FOR Relationship



Degree of a Relationship Type

- Each of the entity types E_1,E_2,...E_n is said to **participate** in the relationship type R, and similarly each of the individual entities e_1, e_2,...,e_n is said to participate in the relationship instance r_i=(e_1, e_2,..., e_n)
- The **degree** of a relationship type is the number of participating entity types.
- A relationship type of degree two is called **binary**, and one of degree three is called **ternary**.

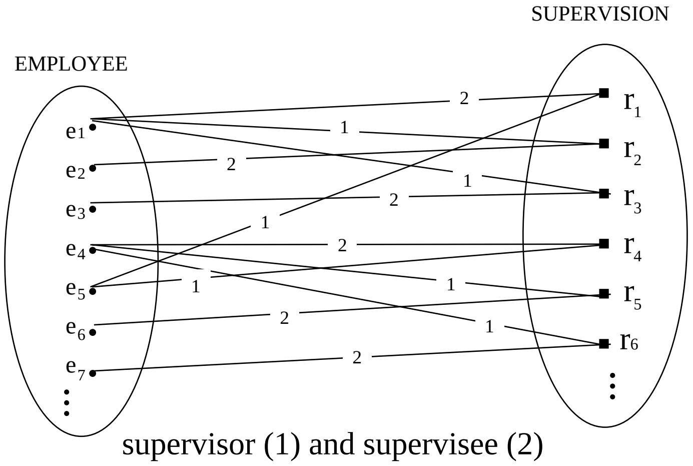
The Ternary Relationship SUPPLY



Role Names and Recursive Relationships

- The role name signifies the role that a participating entity plays in each relationship instance
- When all entity types are distinct, role name is not needed
- Sometimes, the same entity type participates more than once in a relationship type is different roles

Recursive Relationship SUPERVISION



supervisor (1) and supervisee (2)

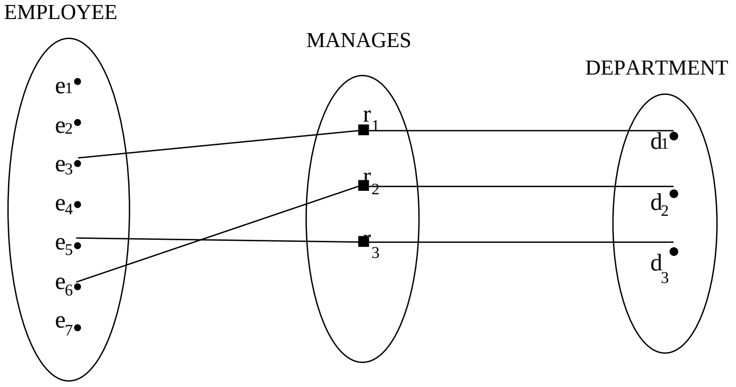
Constraints on Relationship Types

- Cardinality ratio
  - specifies the number of relationship instances that an entity can participate in
- Participation constraint
  - specifies whether the existence of an entity depends on its being related to another entity via the relationship type
- Structural constraints
  - cardinality ratio + participation constraint

Cardinality Ratio

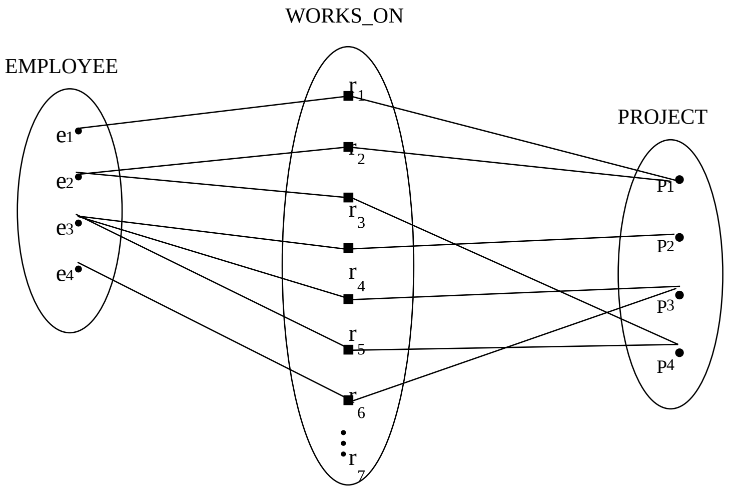
- DEPARTMENT:EMPLOYEE   1:N
- EMPLOYEE:PROJECT M:N (WORK_ON)
- EMPLOYEE:DEPARTMENT 1:1 (MANAGE)

The 1:1 Relationship Manages



EMPLOYEE: partial participation; DEPARTMENT: total participation

The M:N relationship WORKS_ON



Participation Constraint

- Total
  - if every employee must work for a department, the entity EMPLOYEE in WORKS_FOR is called total
  - sometimes called existence dependency
- Partial
  - some or “part” of the set of employee entities are related ot a department entity via MANAGES, but not necessarily all

Attributes of Relationship

- The number of hours per week that an employee works on a project
- Attributes of 1:1 or 1:N relationship types can be migrated ot one of the participating entity types
  - StartDate of MANAGES may be migrated to EMPLOYEE or DEPARTMENT
- For M:N relationship, the attribute should be determined by the combination of participating entities

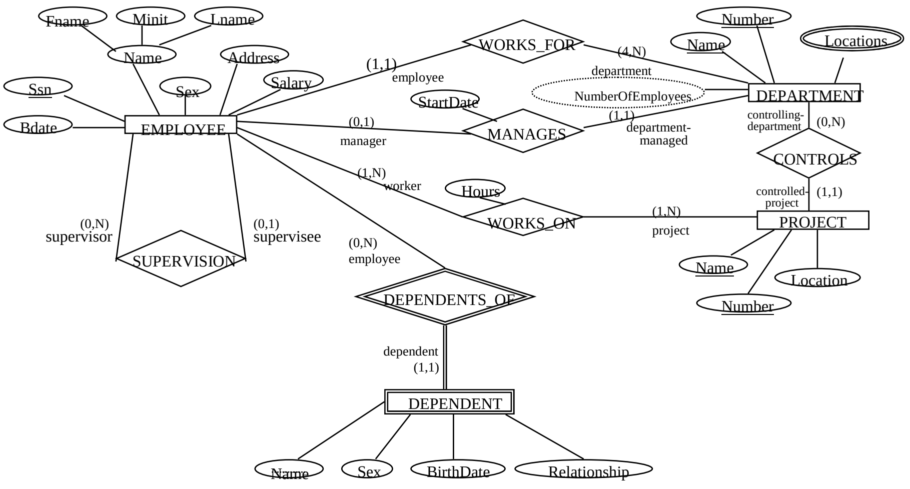
Weak Entity Types

- Entity type without any key attribute is called weak entity type
- Entities of a week entity type is identified by specific entities from another entity type via *identifying relationship*
- A weak entity type always has a total participation constraint

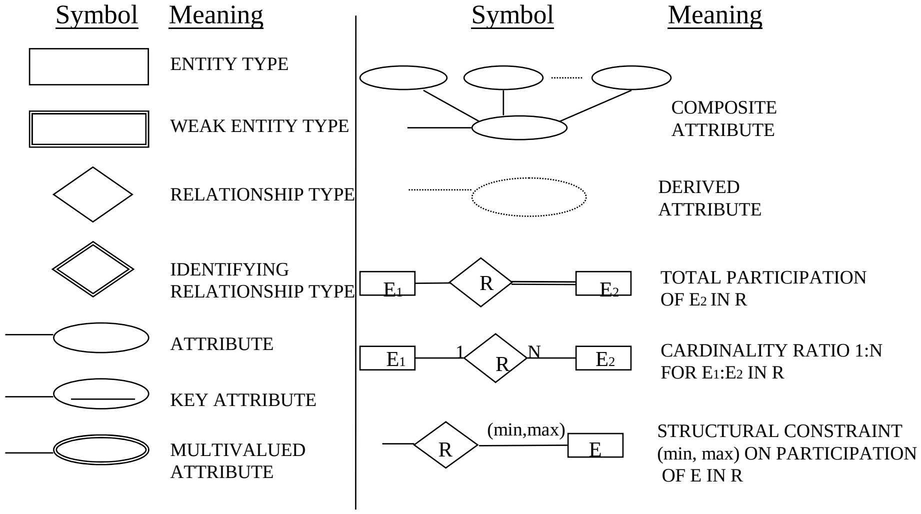
Structure Constraints

- Specifying structural constraints involves associating a pair of integer numbers (min, max) with each participation of an entity type E in a relationship R
- 0<=min<=max, max>=1
- Each entity e in E must participation in at least min and at most max relationship instances in R
- min=0 means partial participation
- min>0 means total participation

ER Diagram for the COMPANY Schema



Summary of ER Diagram Notation



Symbol Meaning

ENTITY TYPE

WEAK ENTITY TYPE

RELATIONSHIP TYPE

IDENTIFYING RELATIONSHIP TYPE

ATTRIBUTE

KEY ATTRIBUTE

MULTIVALUED ATTRIBUTE

Symbol Meaning

COMPOSITE ATTRIBUTE

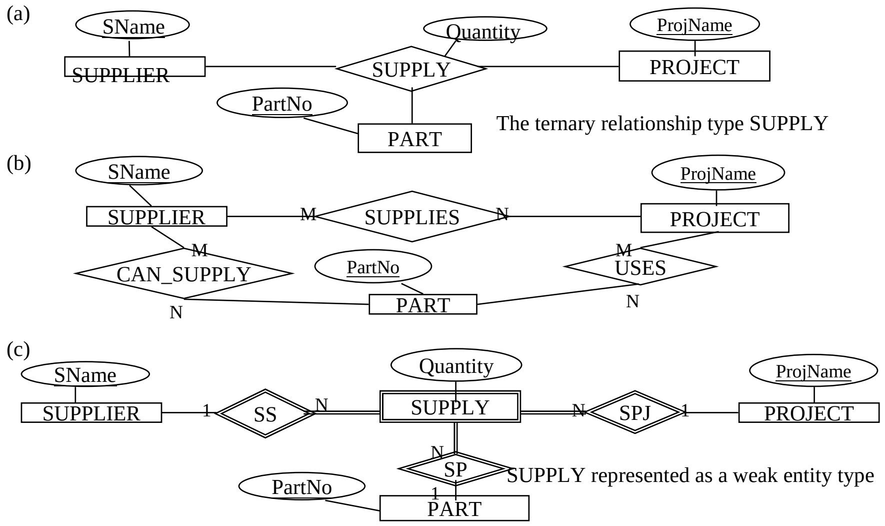
DERIVED ATTRIBUTE

TOTAL PARTICIPATION OF E2 IN R

CARDINALITY RATIO 1:N FOR E1:E2 IN R

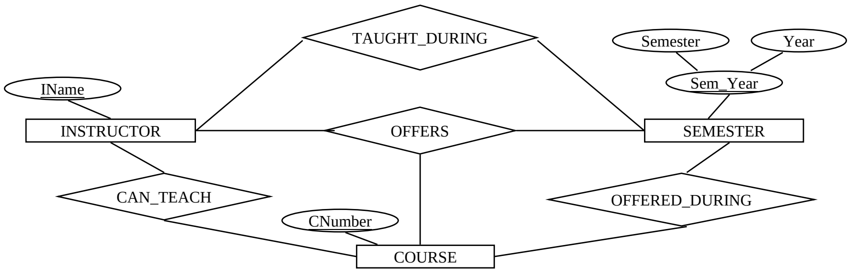
STRUCTURAL CONSTRAINT (min, max) ON PARTICIPATION OF E IN R

Ternary Relationship Types

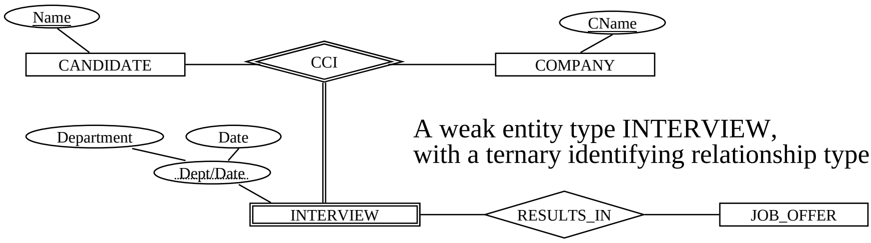


The ternary relationship type SUPPLY

SUPPLY represented as a weak entity type



Another example of ternary versus binary relationship types.



A weak entity type INTERVIEW, with a ternary identifying relationship type.