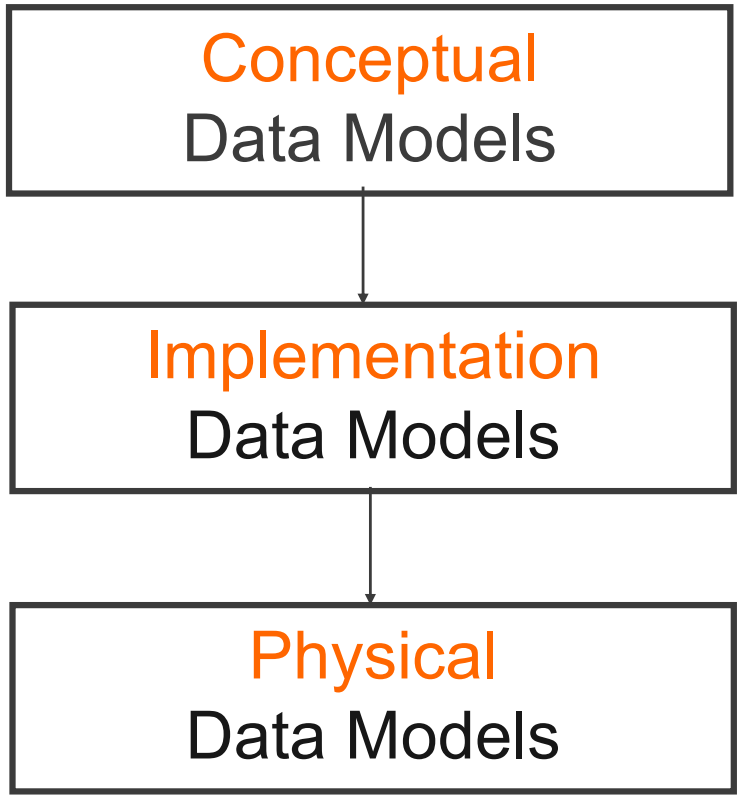
Data Modeling Using the Entity-Relationship Model

政治大學
資訊科學系
沈錳坤

Outline

- Data Models
- Database Design Process
- Example Database Application
- Entity-Relationship Model
- ER Diagrams, Naming Conventions & Design Issues

Data Models



Entity-Relationship Model

ER to Relational Mapping

Relational Data Model
SQL, Relational Algebra
Normalization

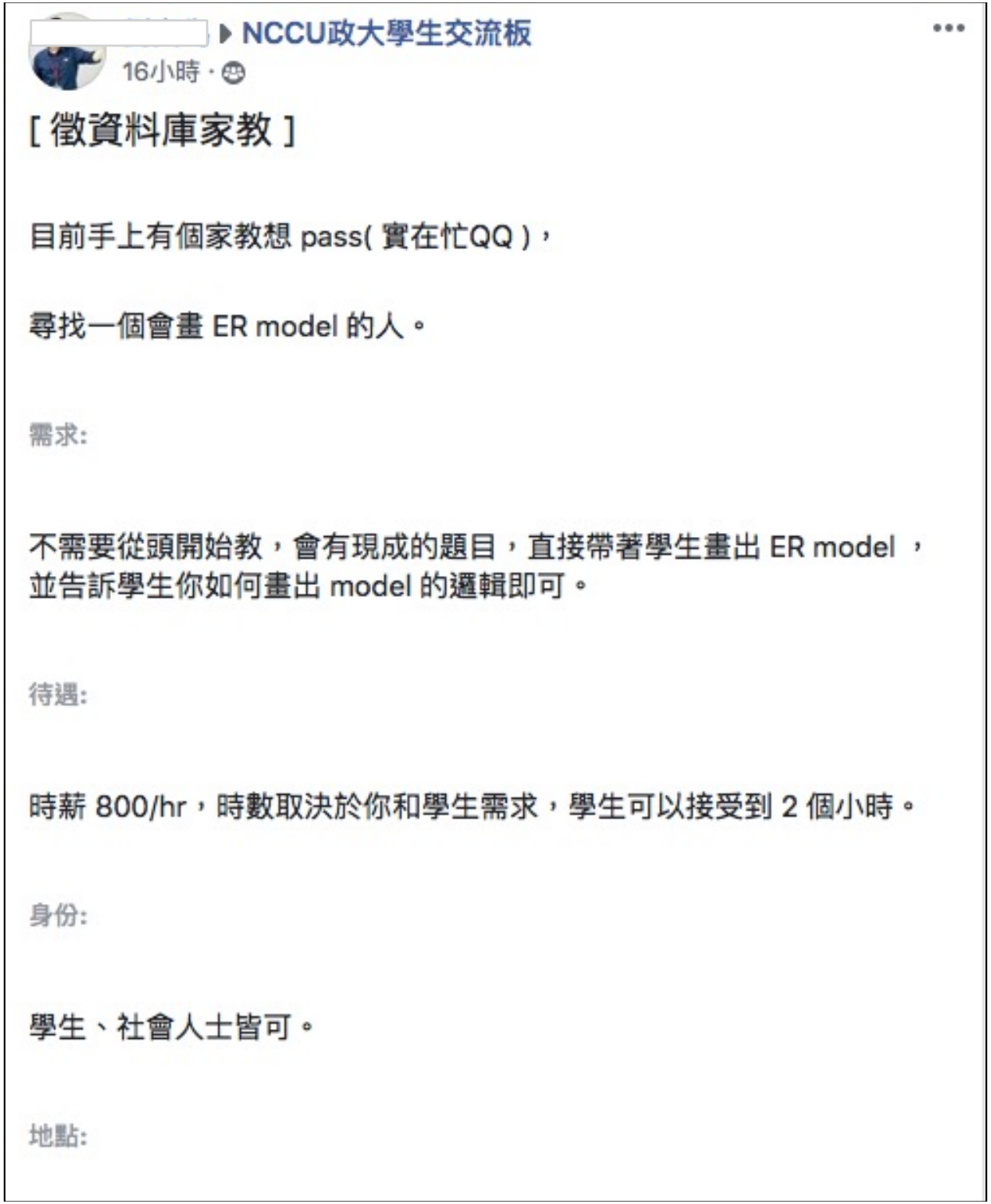
DBMS
Record Structure, Indexing
Query Processing
Query Optimization
Transaction Processing

Data Models (cont.)

- **Conceptual** data model
  - High level data model
  - Provide concepts that are close to the way **users** perceive data
- **Implementation** data model
  - Logical data model
  - Provide representations that may be understood by end users but not far removed from the way data is organized with the computer
  - Hide some details of data storage but can be **implemented** on a computer system in direct way
- **Physical** data model
  - Low level data model
  - Provides concept that describe the details of how data is **stored** in the computer
  - Meant for computer specialists, not for typical end users

描述資料需求

(ref: https://www.chinatimes.com/realtimenews/20220129002478-260410?chdtv)



NCCU政大學生交流板

16小時 ·

[徵資料庫家教]

目前手上有個家教想 pass(實在忙QQ)，

尋找一個會畫 ER model 的人。

需求:

不需要從頭開始教，會有現成的題目，直接帶著學生畫出 ER model ，並告訴學生你如何畫出 model 的邏輯即可。

待遇:

時薪 800/hr，時數取決於你和學生需求，學生可以接受到 2 個小時。

身份:

學生、社會人士皆可。

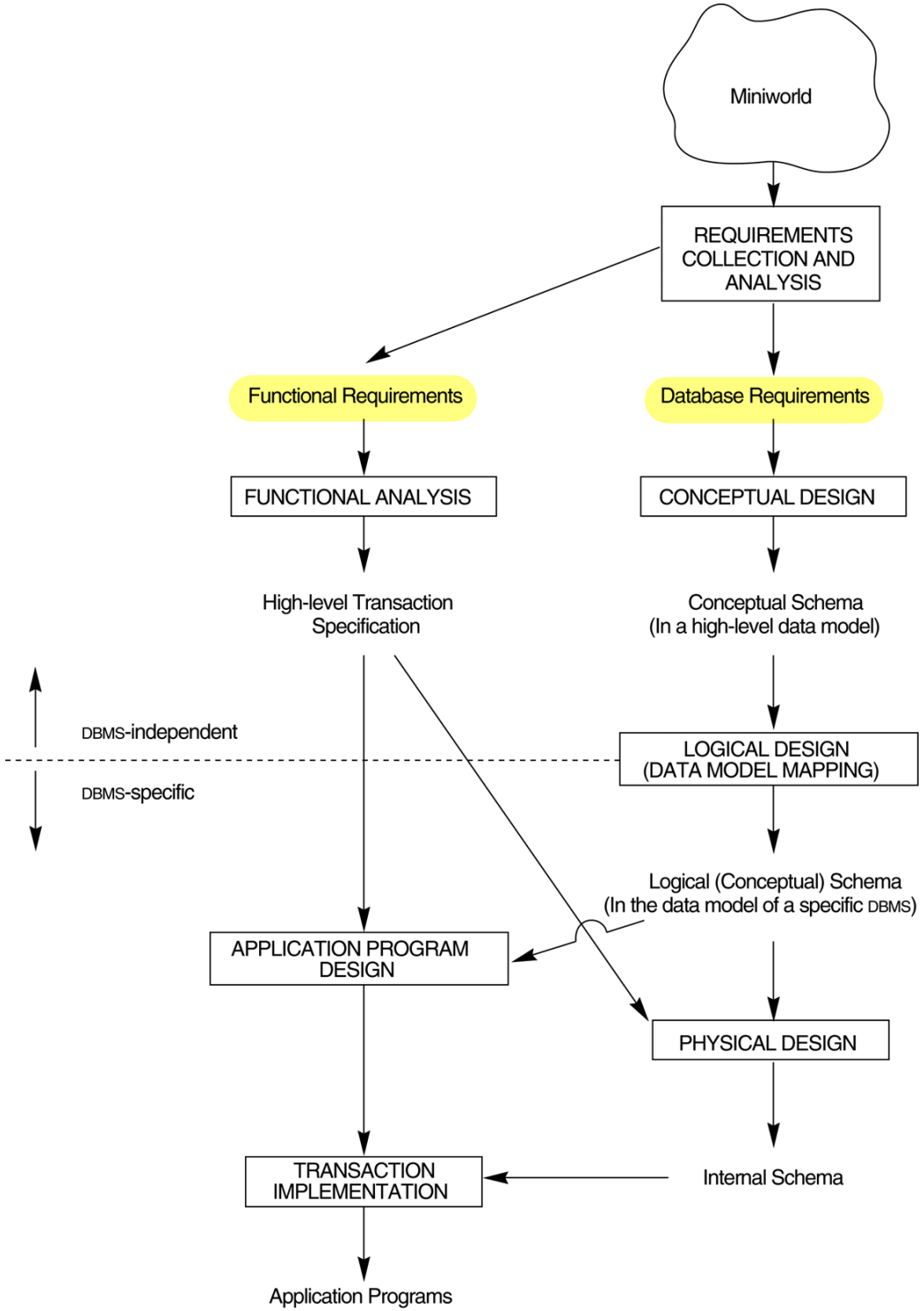
地點:

Outline

- Data Models
- **Database Design Process**
- Example Database Application
- Entity-Relationship Model
  - Entity Types, Entity Sets, Attributes & Keys
  - Relationships, Relationships, Roles & Structure Constraints
  - Weak Entity Types
- ER Diagrams, Naming Conventions & Design Issues
- Mapping ER into Relational Data Models

Database Design Process

- Step 1: Requirements collection & analysis
  - Database requirements
  - Functional requirements
- Step 2: Conceptual design
  - Create a conceptual schema for DB, using conceptual data model
- Step 3: Logical design (data model mapping)
  - Actual implementation of DB using a commercial DBMS
  - Transform from conceptual data model to implementation data model
- Step 4: Physical design & Application program design
  - Internal storage structure, indexing & file organization are specified
  - Application program are designed & implemented as DB transactions corresponding to high level transaction specifications


Miniworld
REQUIREMENTS COLLECTION AND ANALYSIS
Functional Requirements
Database Requirements
FUNCTIONAL ANALYSIS
CONCEPTUAL DESIGN
High-level Transaction Specification
Conceptual Schema (In a high-level data model)
DBMS-independent
DBMS-specific
LOGICAL DESIGN (DATA MODEL MAPPING)
Logical (Conceptual) Schema (In the data model of a specific DBMS)
APPLICATION PROGRAM DESIGN
PHYSICAL DESIGN
TRANSACTION IMPLEMENTATION
Internal Schema
Application Programs

Requirements Collection & Analysis

- Database requirements
  - DB designers interview database users to understand & document data requirements
- Functional requirements
  - User-defined operations (transactions) applied to the DB
  - Retrievals & updates operations
  - Data flow diagrams, sequence diagrams, scenarios to specify

    * Software engineering

Conceptual Database Design

- Create a conceptual schema using conceptual data model
- Conceptual schema
  - Concise description of data requirements of users & detailed descriptions of entity types, relationships & constraints
  - Be used
    - to communicate with non-technical users
    - as a reference to ensure that all users' data requirements are met and do not conflict
  - Not include implementation details
  - Enable DB designers to concentrate on

    specifying properties of data,

    without being concerned with storage details.

Outline

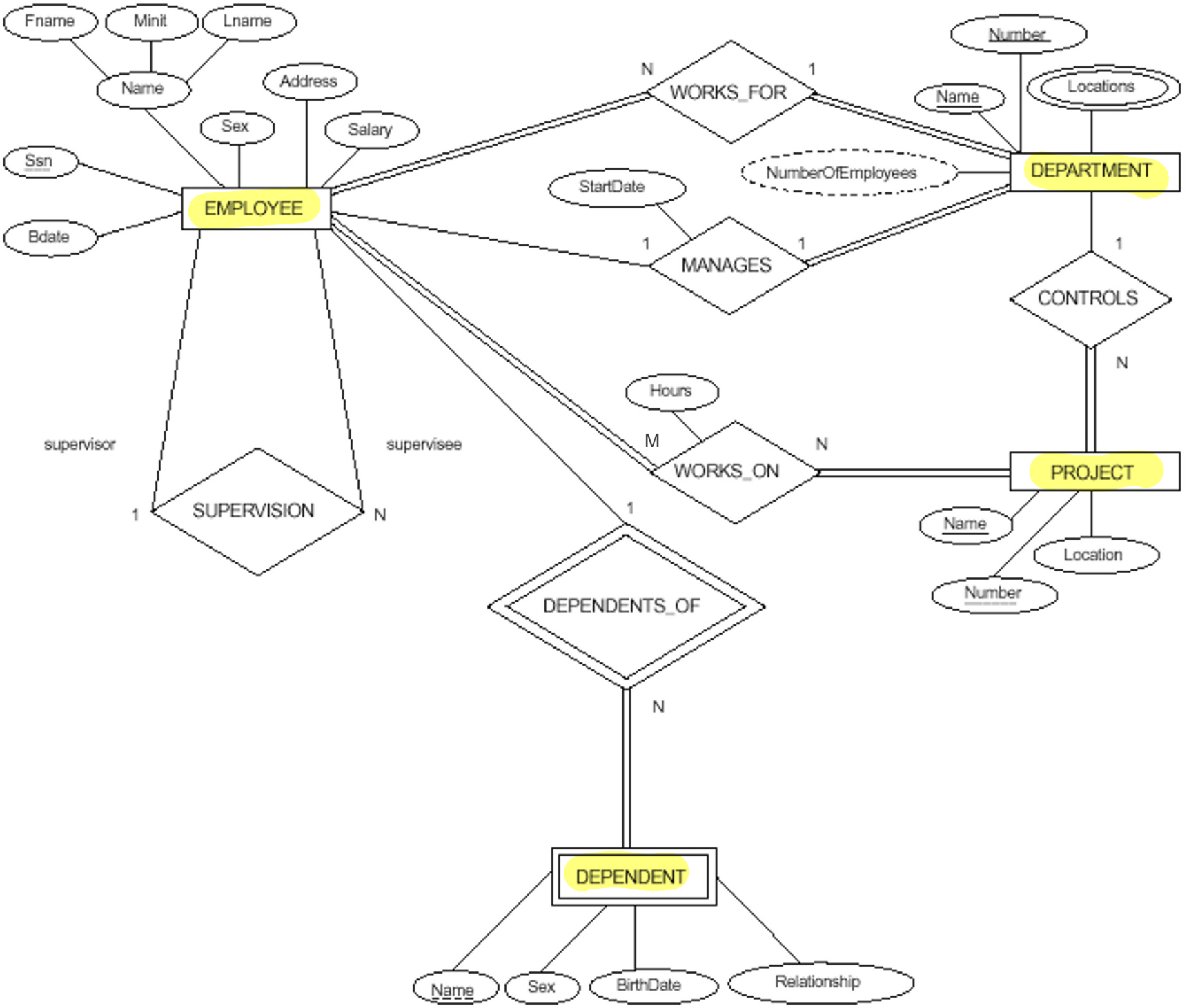
- Data Models
- Database Design Process
- **Example Database Application**
- Entity-Relationship Model
  - Entity Types, Entity Sets, Attributes & Keys
  - Relationships, Relationships, Roles & Structure Constraints
  - Weak Entity Types
- ER Diagrams, Naming Conventions & Design Issues
- Mapping ER into Relational Data Models

An Example Database Application

- keeps track of a company's employees, departments & projects
  - The company is organized into departments.
  - Each department has a unique name, a unique number & 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 & a single location
  - We store each employee's name, social security number, address, salary, sex & birth date

An Example Database Application (cont.)

- 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 keep track of the direct supervisor of each employee.
- We keep track of the dependents of each employee for insurance purpose.
- We keep each dependent's first name, sex, birth date & relationship to the employee


Fname
Minit
Lname
Name
Address
Sex
Salary
Ssn
Bdate
EMPLOYEE
N
WORKS_FOR
1
Number
Name
Locations
DEPARTMENT
NumberOfEmployees
StartDate
1
MANAGES
1
1
CONTROLS
N
Hours
M
N
WORKS_ON
PROJECT
Name
Location
Number
supervisor
supervisee
1
SUPERVISION
N
1
DEPENDENTS_OF
N
DEPENDENT
Name
Sex
BirthDate
Relationship

Outline

- Data Models
- Database Design Process
- Example Database Application
- **Entity-Relationship Model**
  - Entity Types, Entity Sets, Attributes & Keys
  - Relationships, Relationships, Roles & Structure Constraints
  - Weak Entity Types
- ER Diagrams, Naming Conventions & Design Issues
- Mapping ER into Relational Data Models

Entity-Relationship Model

Association of Computing Machinery

- A conceptual model proposed by Peter Chen (陳品山)
- ACM Transactions on Database Systems, 1976/3

  The Entity-Relationship Model: Toward a Unified View of Data
- ER-Model
  - Entity-Type
  - Relationship-Type

The Entity-Relationship Model—Toward a Unified View of Data

PETER PIN-SHAN CHEN

Massachusetts Institute of Technology

A data model, called the entity-relationship model, is proposed. This model incorporates some of the important semantic information about the real world. A special diagrammatic technique is introduced as a tool for database design. An example of database design and description using the model and the diagrammatic technique is given. Some implications for data integrity, information retrieval, and data manipulation are discussed.

The entity-relationship model can be used as a basis for unification of different views of data: the network model, the relational model, and the entity set model. Semantic ambiguities in these models are analyzed. Possible ways to derive their views of data from the entity-relationship model are presented.

Key Words and Phrases: database design, logical view of data, semantics of data, data models, entity-relationship model, relational model, Data Base Task Group, network model, entity set model, data definition and manipulation, data integrity and consistency
CR Categories: 3.50, 3.70, 4.33, 4.34

1. INTRODUCTION

The logical view of data has been an important issue in recent years. Three major data models have been proposed: the network model [2, 3, 7], the relational model [8], and the entity set model [25]. These models have their own strengths and weaknesses. The network model provides a more natural view of data by separating entities and relationships (to a certain extent), but its capability to achieve data independence has been challenged [8]. The relational model is based on relational theory and can achieve a high degree of data independence, but it may lose some important semantic information about the real world [12, 15, 23]. The entity set model, which is based on set theory, also achieves a high degree of data independence, but its viewing of values such as "3" or "red" may not be natural to some people [25].

This paper presents the entity-relationship model, which has most of the advantages of the above three models. The entity-relationship model adopts the more natural view that the real world consists of entities and relationships. It

A version of this paper was presented at the International Conference on Very Large Data Bases, Framingham, Mass., Sept. 22–24, 1975.
Author's address: Center for Information System Research, Alfred P. Sloan School of Management, Massachusetts Institute of Technology, Cambridge, MA 02139.

Entity Types, Entity Sets, Attributes, and Keys

Entity

- Entity
  - Basic object that ER model represents
  - A thing in the real world with an independent existence
  - An object with
    - Physical existence: a person, a car, a house, a employee
    - Conceptual existence: a company, job, a course

Attribute

- Attribute
  - Particular properties that describe the entity
  - Attribute value
- Types of attributes
  - Simple vs. composite
  - Single-valued vs. multivalued
  - Stored vs. derived

Simple vs. Composite Attributes

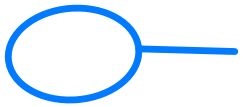
- Simple attributes
  - Atomic attributes
  - Attributes that are not divisible
- Composite attribute
  - be divided into smaller subparts, which represent more basic attributes with independent meaning
  - value is the concatenation of the values of its constituent simple attributes
  - are useful when a user sometimes refers to the composite attribute as a unit but at other times refers to its component
  - If the composite attribute is referenced only as a whole, there is no need to subdivide it into component attributes

Address 是 Simple Attribute

還是 Composite Attribute ?

Composite attribute

Single-valued vs. Multi-valued Attributes

- Single-valued attribute 
  - Attribute with a single value
- Multi-valued attribute 
  - Attribute with different number of values

Stored vs. Derived Attributes

- Derived attributes 
  - Attribute whose value is derived from a stored attribute

    E.g. Age is derived from the BirthDate
  - Some attribute values can be derived from related entities

    E.g. NumberOfEmployee can be derived by counting the number of employees related to the department

為什麼要特別區別

Stored vs. Derived Attributes ?

constraints

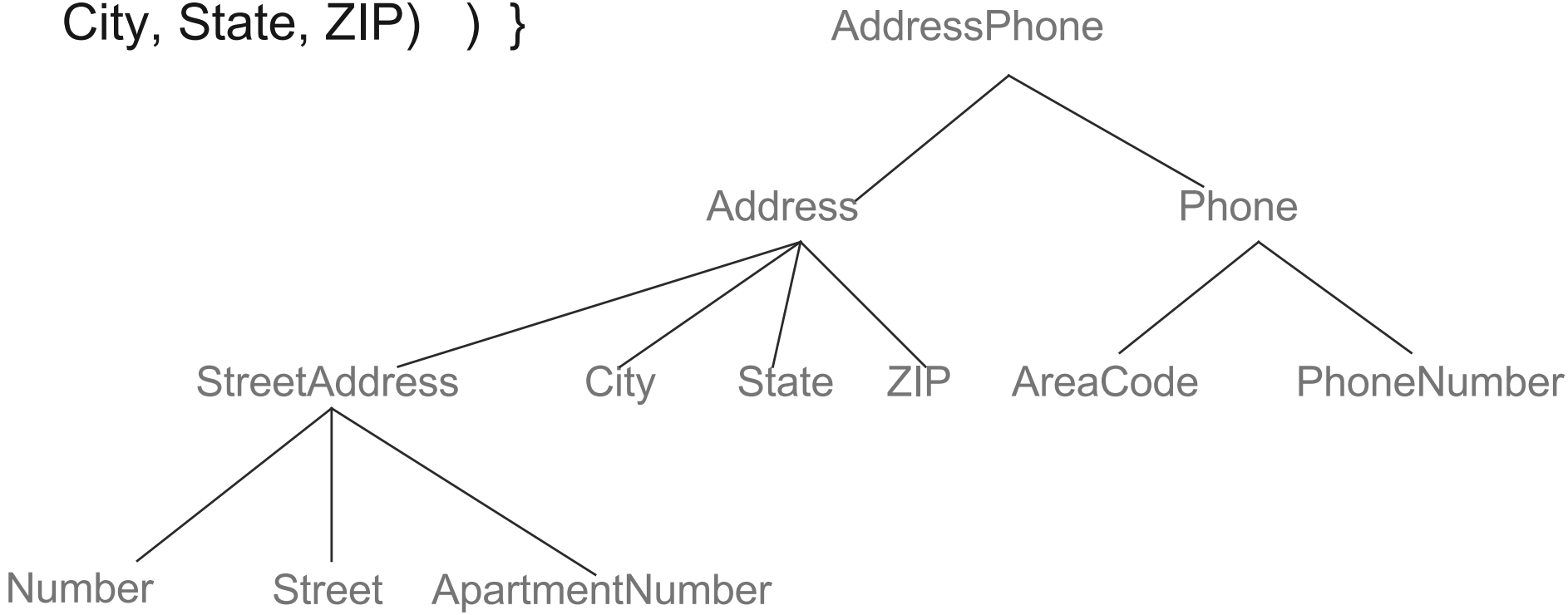
Value Sets of Attributes

- Value sets (domains) of attributes
  - Specifies the set of values that may be assigned to that attribute for each individual entity
  - A: E ➔ P(V)

Complex Attributes

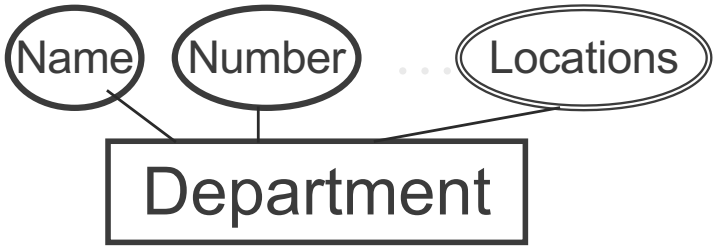
- Complex attributes
  - Attributes with composite components & multi-valued
  - E.g.

{ AddressPhone ({Phone(AreaCode, PhoneNumber)}
Address(StreetAddress(Number, Street, ApartmentNumber),
City, State, ZIP)) }



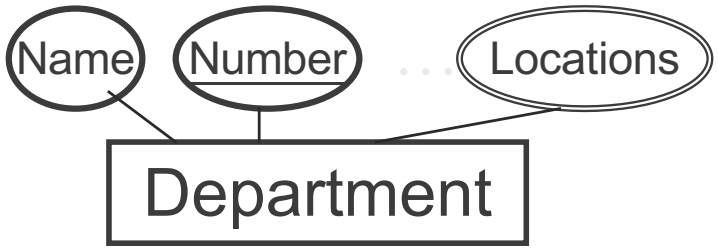
Entity Types

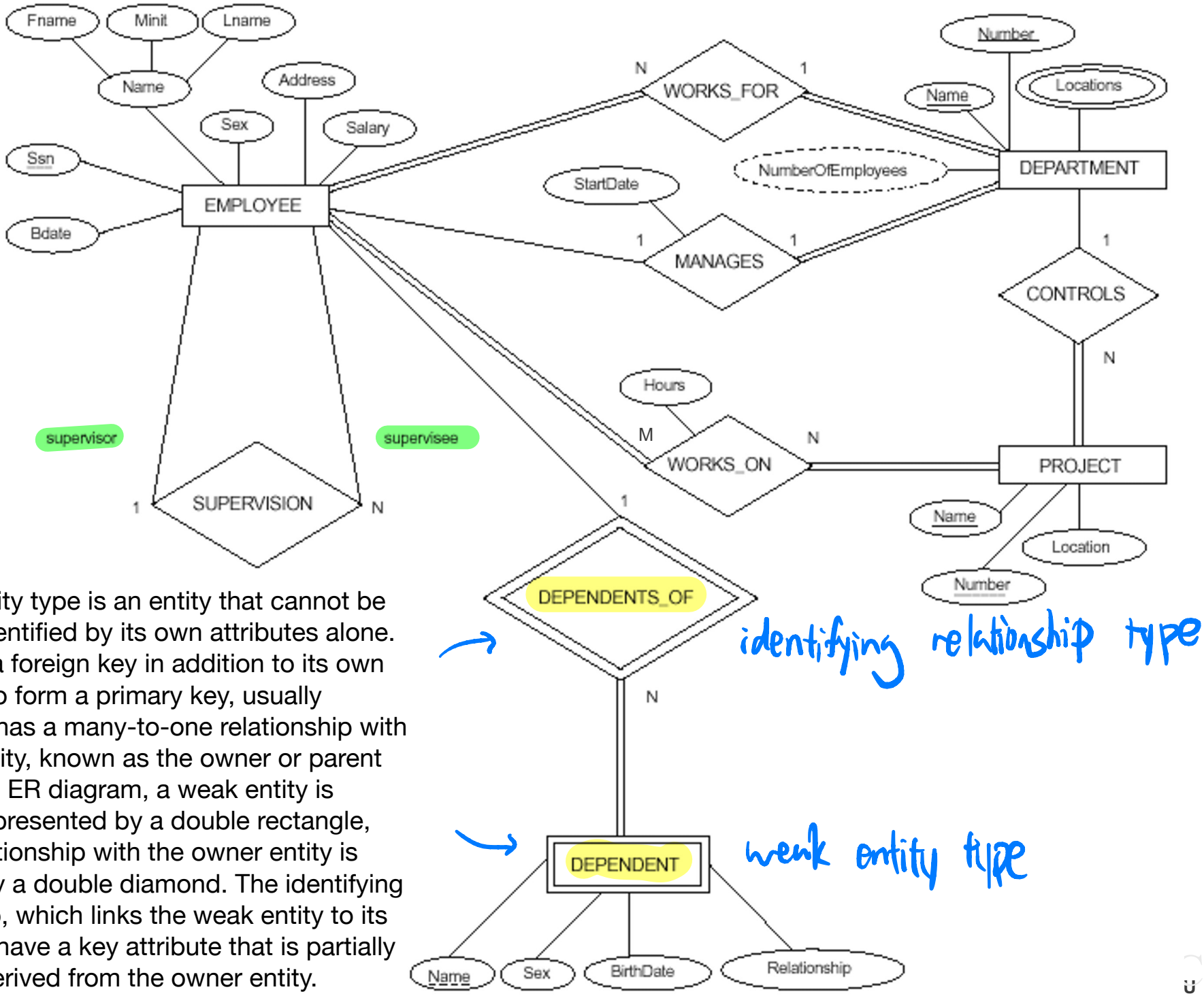
- Entity type
  - A collection of entities that have the same attributes
  - Describes the schema for a set of entities that share the same structure
  - In ER diagram
    - Entity type is represented as a rectangular box
    - Attribute is represented as a ovals and are attached to their entity type by straight lines
    - Multi-valued attributes are represented as double ovals



Keys

- Key attribute
  - Attribute whose values are distinct for each individual entity in the collection of entities
  - Uniqueness constraint on entities of an entity type
  - Weak entity type: entity type have no key attributes
  - In ER diagram
    - Key attribute has its name underlined inside the oval





A weak entity type is an entity that cannot be uniquely identified by its own attributes alone. It requires a foreign key in addition to its own attributes to form a primary key, usually because it has a many-to-one relationship with another entity, known as the owner or parent entity. In an ER diagram, a weak entity is typically represented by a double rectangle, and its relationship with the owner entity is depicted by a double diamond. The identifying relationship, which links the weak entity to its owner, will have a key attribute that is partially or totally derived from the owner entity.

Initial Conceptual Design of the Company DB

- Department

  Name, Number, {Locations}, Manager, MamagerStartDate

- 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

Outline

- Data Models
- Database Design Process
- Example Database Application
- Entity-Relationship Model
  - Entity Types, Entity Sets, Attributes & Keys
  - **Relationships, Relationships, Roles & Structure Constraints**
  - Weak Entity Types
- ER Diagrams, Naming Conventions & Design Issues
- Mapping ER into Relational Data Models

Relationships, Relationship Types, Roles, & Structural Constraints

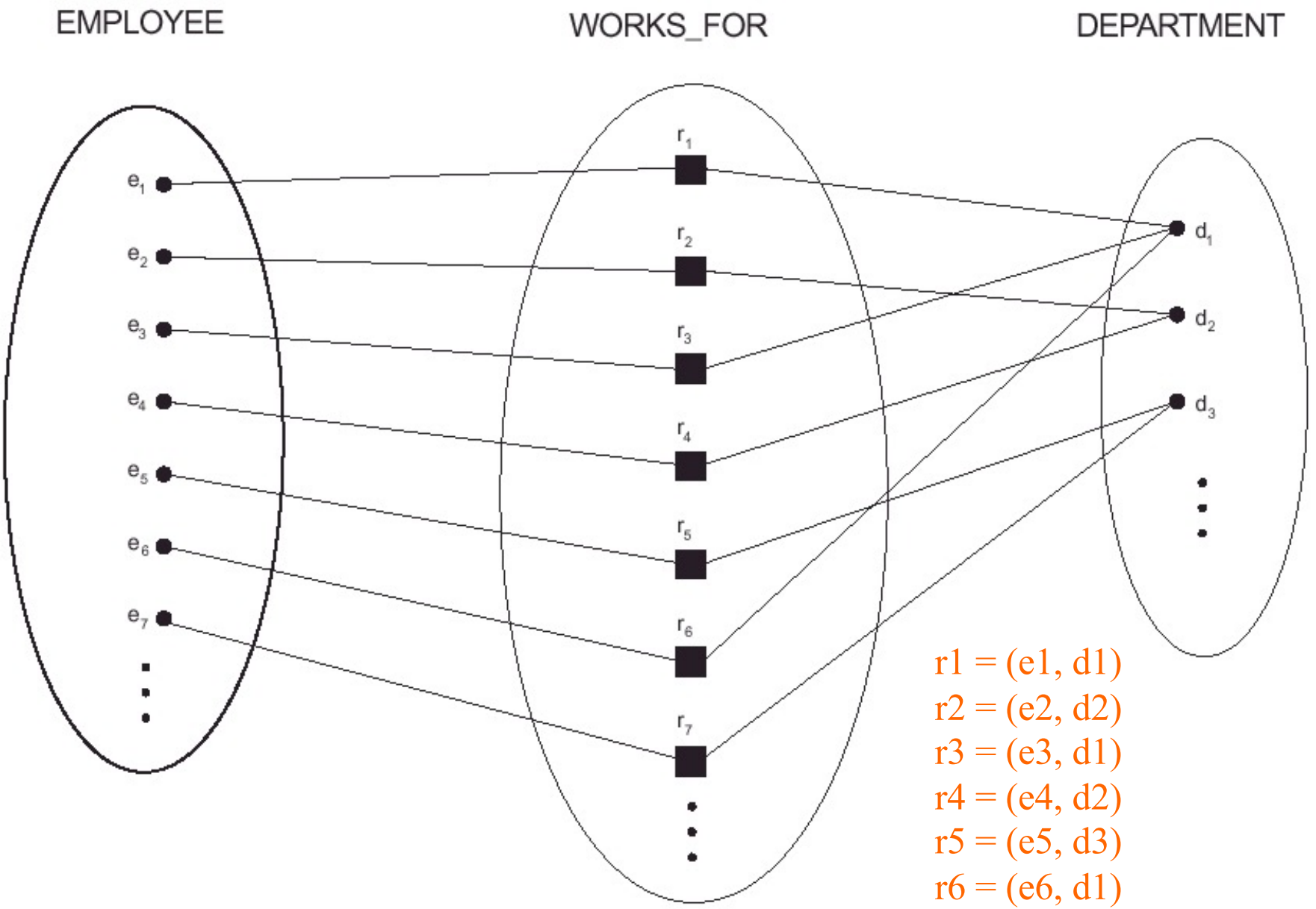
Relationship Types

- Relationship type *R* among *n* entity type E_1, E_2, …E_n
  - defines a set of associations (relationship set)

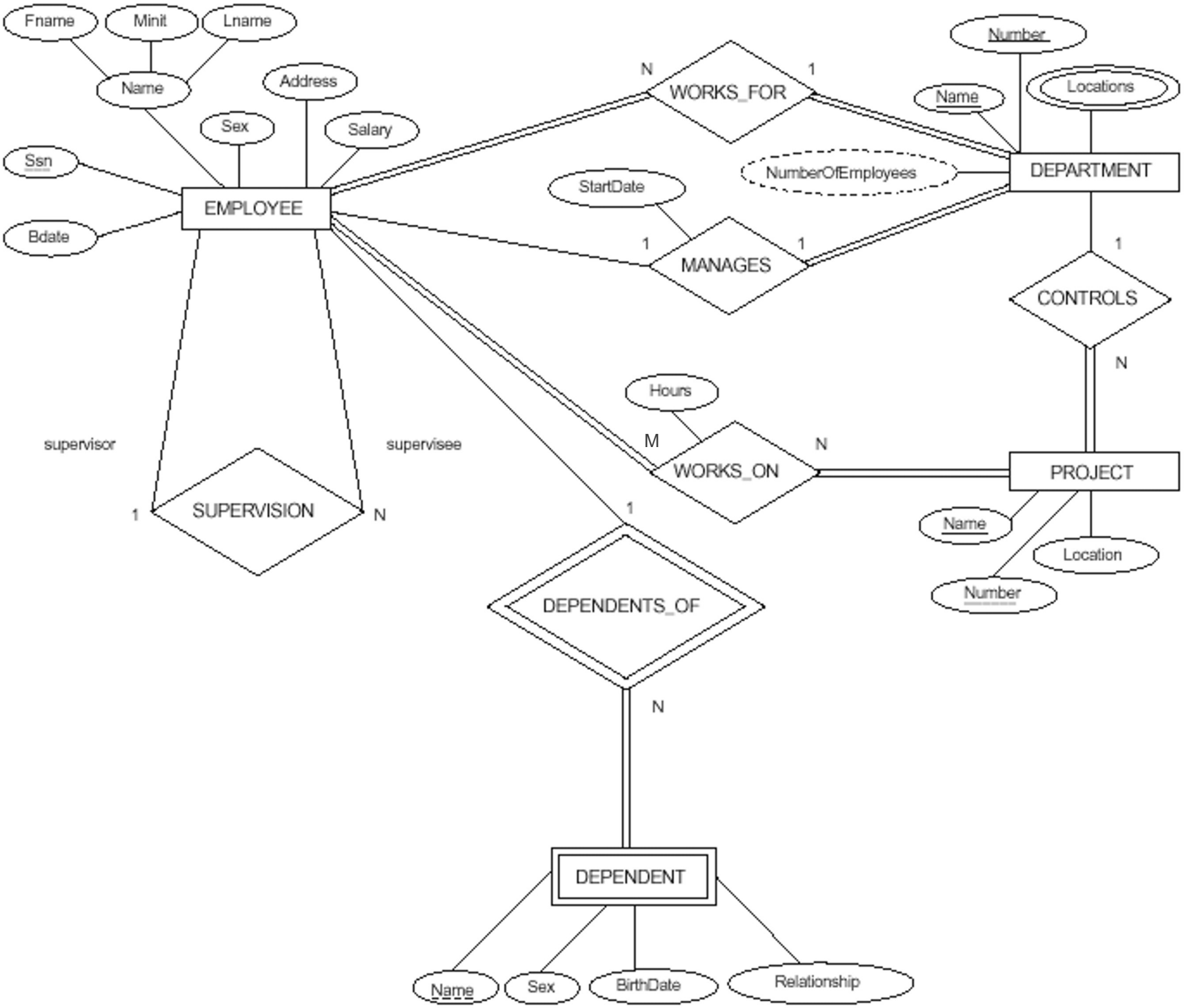
    among entities
  - each relationship instance in *R* is an association of entities
  - the association includes exactly one entity from each participating entity type
  - relationship types are represented as diamond-shaped boxes, which are connected by straight lines to

    the rectangular boxes

    representing the participating entity types

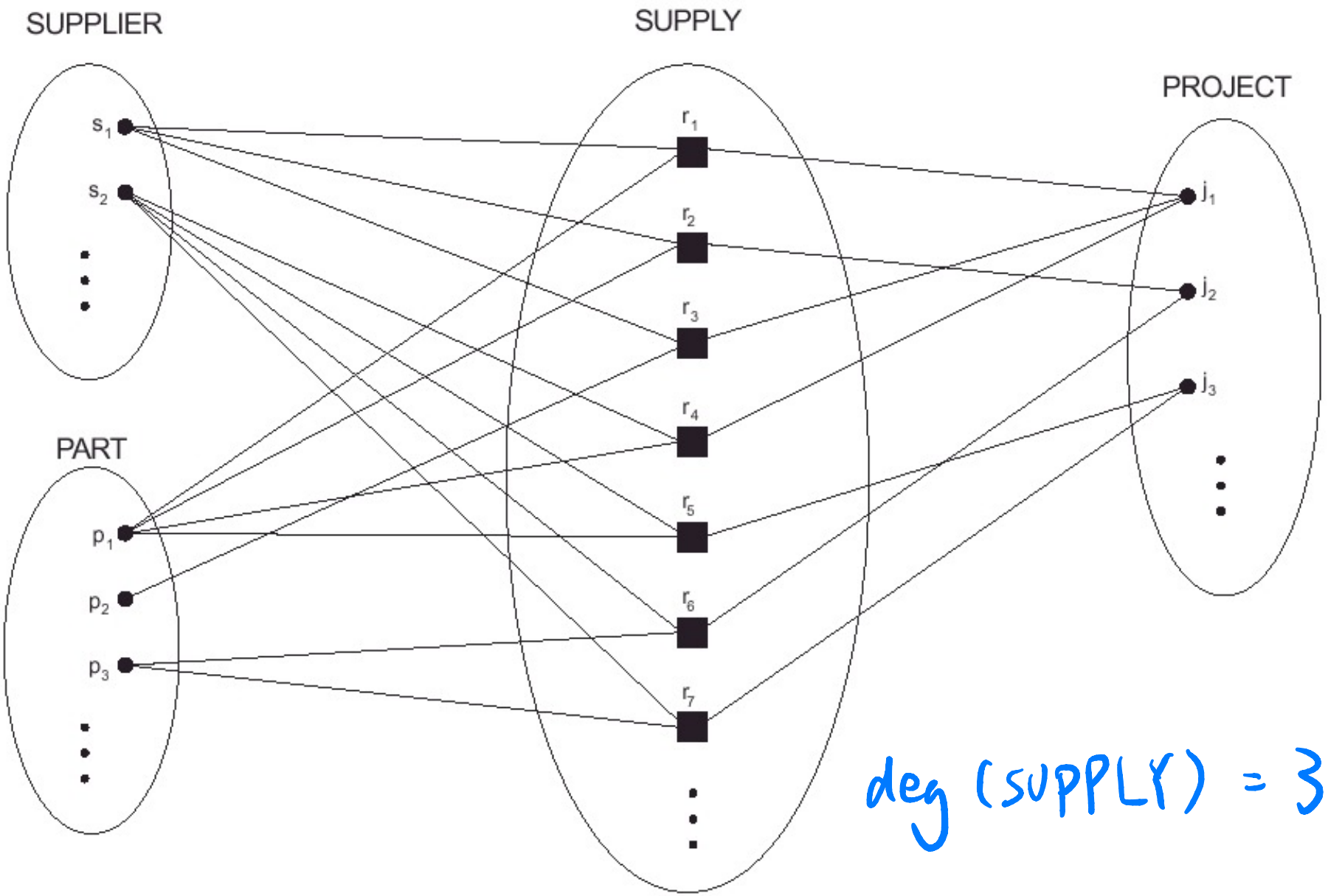


r1 = (e1, d1)
r2 = (e2, d2)
r3 = (e3, d1)
r4 = (e4, d2)
r5 = (e5, d3)
r6 = (e6, d1)
r7 = (e7, d3)


Fname
Minit
Lname
Name
Address
Sex
Salary
Ssn
Bdate
EMPLOYEE
N
WORKS_FOR
1
Number
Name
Locations
DEPARTMENT
NumberOfEmployees
StartDate
1
MANAGES
1
1
CONTROLS
N
Hours
M
N
WORKS_ON
PROJECT
Name
Location
Number
supervisor
supervisee
1
SUPERVISION
N
1
DEPENDENTS_OF
N
DEPENDENT
Name
Sex
BirthDate
Relationship

Degree of Relationship Type

- Degree of relationship type
  - Number of participating entity type
  - Binary: of degree two
  - Ternary: of degree three

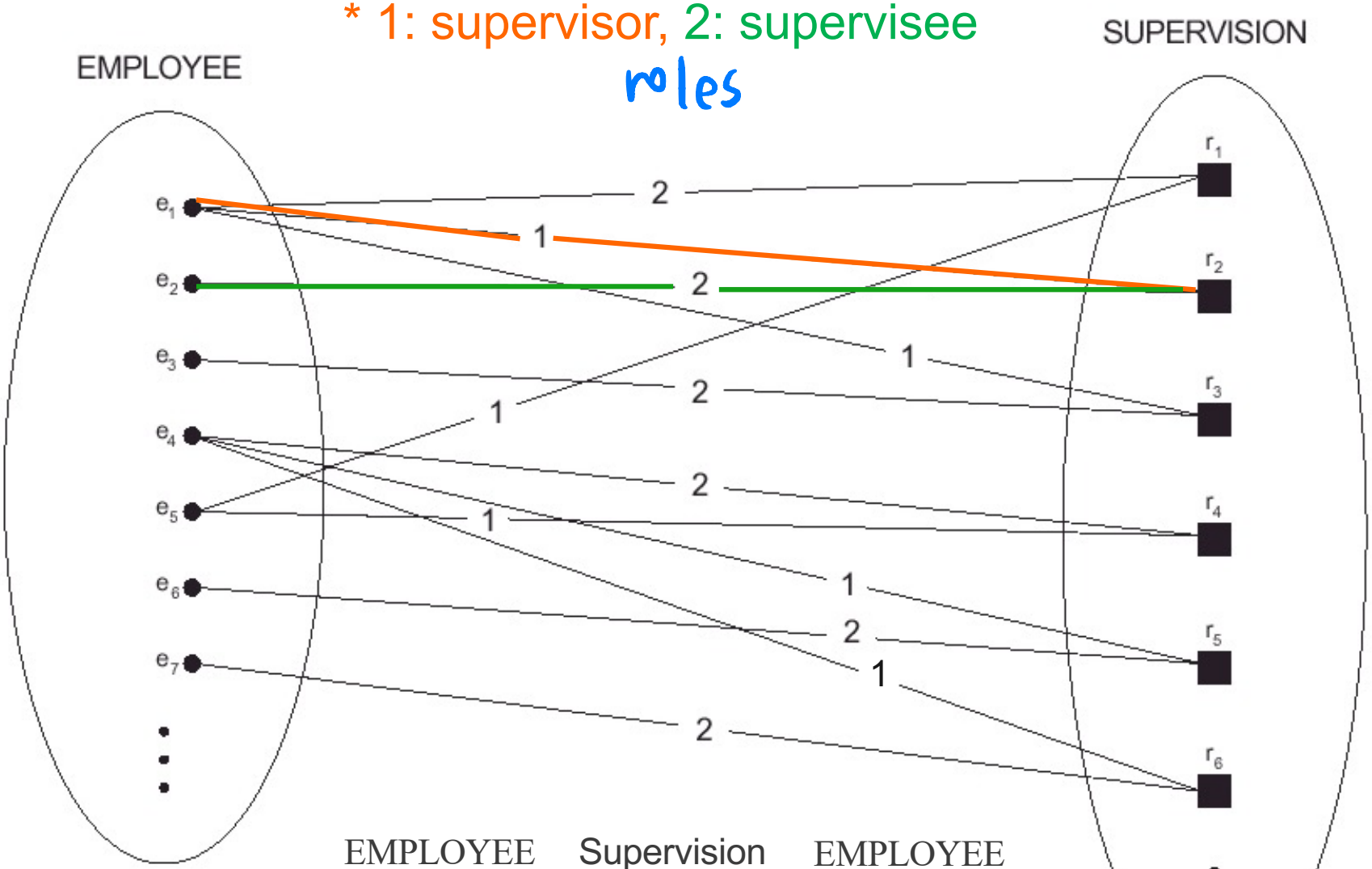

SUPPLIER
SUPPLY
PROJECT
PART
s_1
s_2
p_1
p_2
p_3
r_1
r_2
r_3
r_4
r_5
r_6
r_7
j_1
j_2
j_3
deg (SUPPLY) = 3

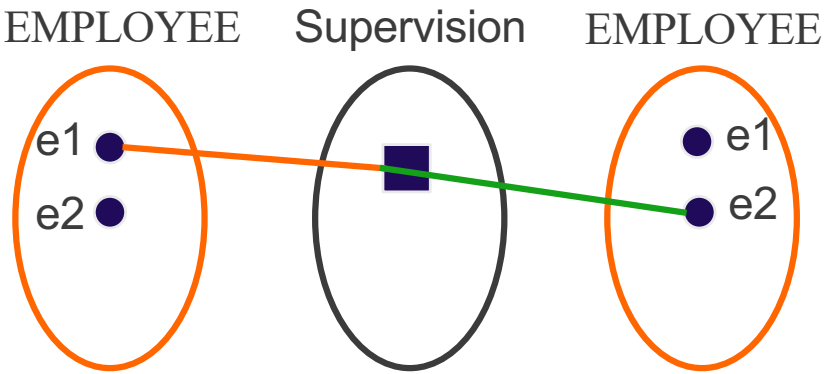
Role Name & Recursive Relationships

- Role
  - Each entity type that participates in a relationship type plays a particular role in the relationship
  - It's useful when the same entity type participates more than once in a relationship type in different roles
- Recursive relationships
  - The same entity participates more than once in a relationship type in different roles

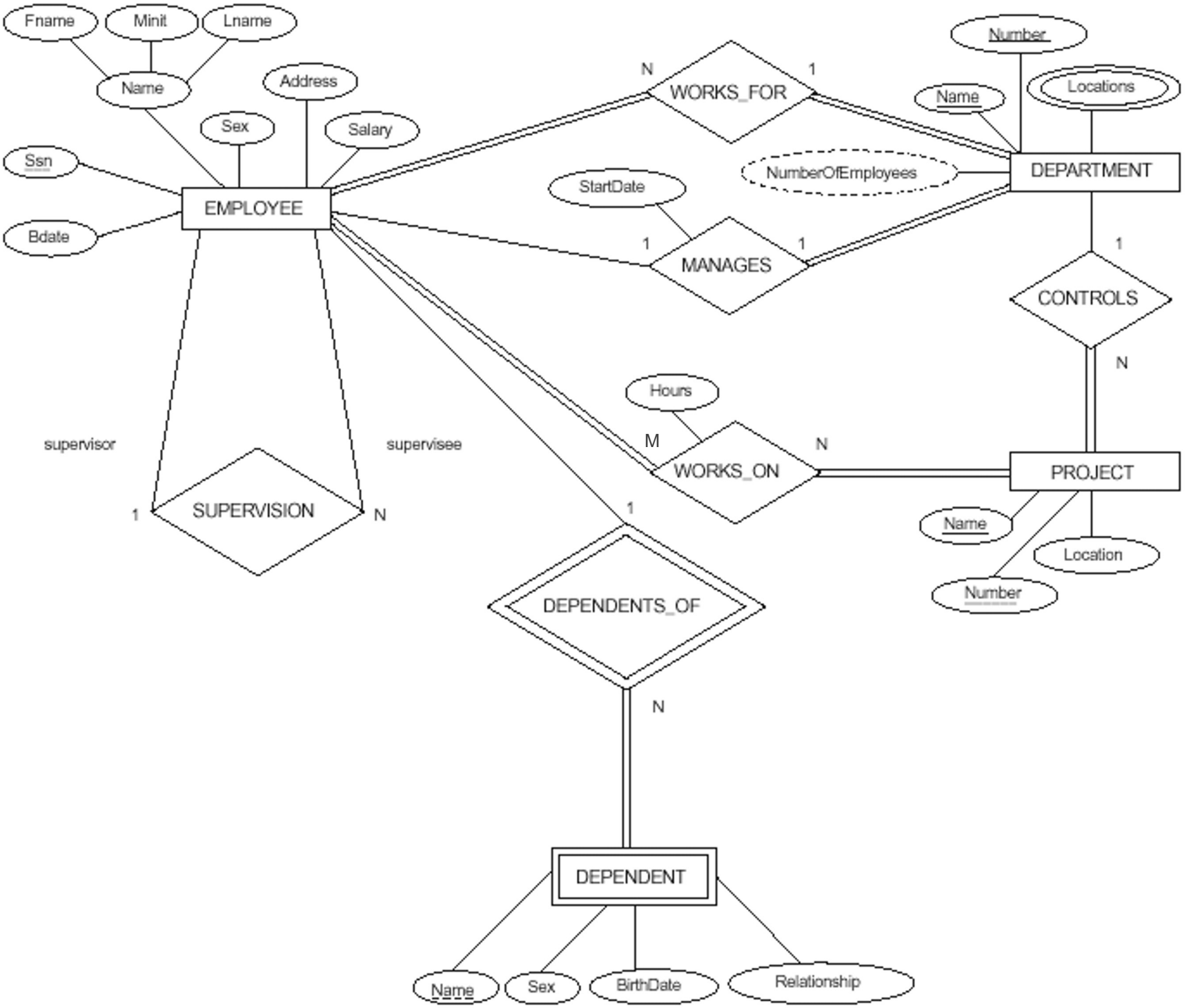
* 1: supervisor, 2: supervisee

roles





recursive relationship


Fname
Minit
Lname
Name
Address
Sex
Salary
Ssn
Bdate
EMPLOYEE
N
WORKS_FOR
1
Number
Name
Locations
DEPARTMENT
NumberOfEmployees
StartDate
1
MANAGES
1
1
CONTROLS
N
Hours
M
N
WORKS_ON
PROJECT
Name
Location
Number
supervisor
supervisee
1
SUPERVISION
N
1
DEPENDENTS_OF
N
DEPENDENT
Name
Sex
BirthDate
Relationship

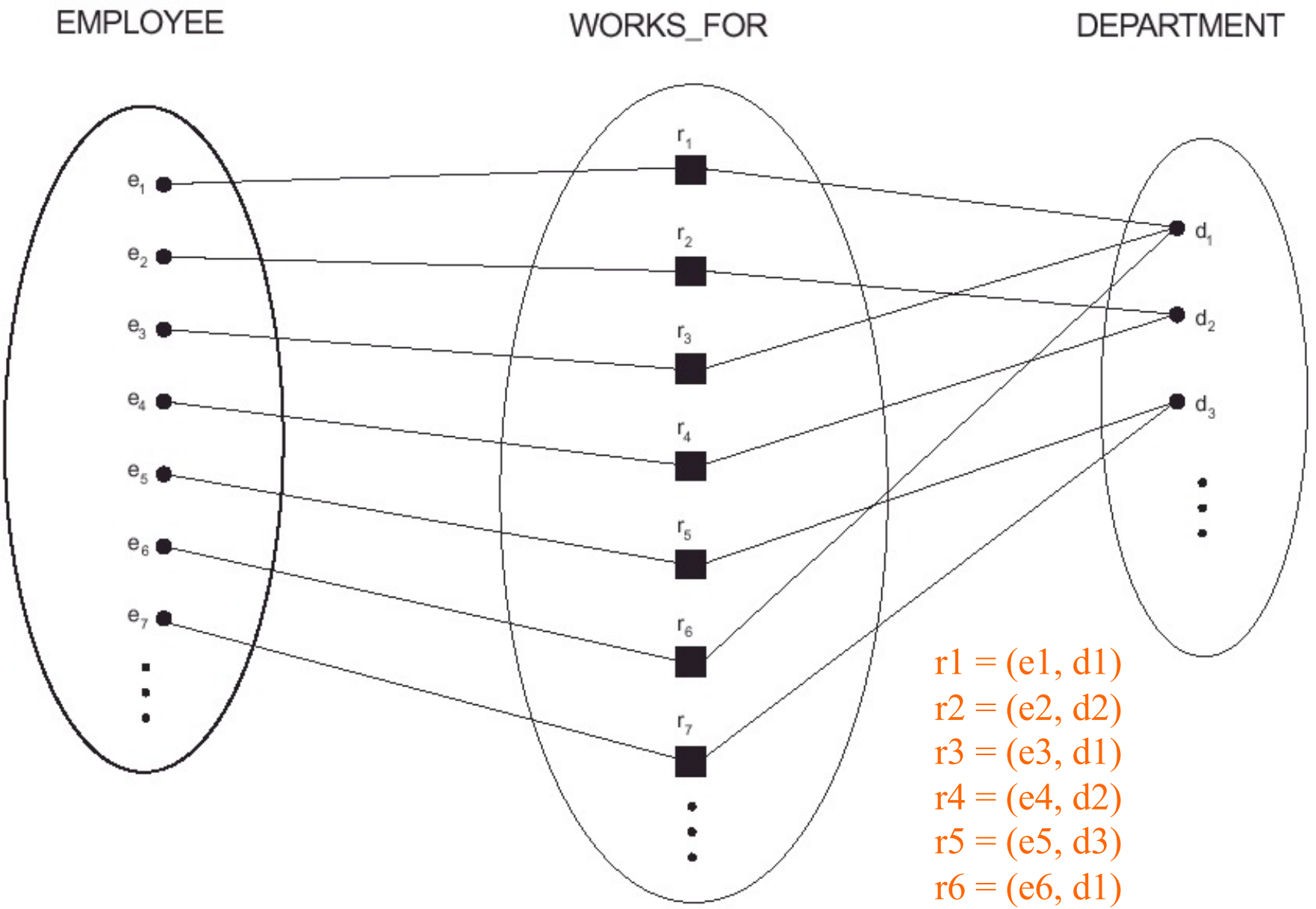
Facebook 的 Entity Types中
有 Recursive Relationship 嗎？ Yes (Friends)

Constraints on Relationship Types

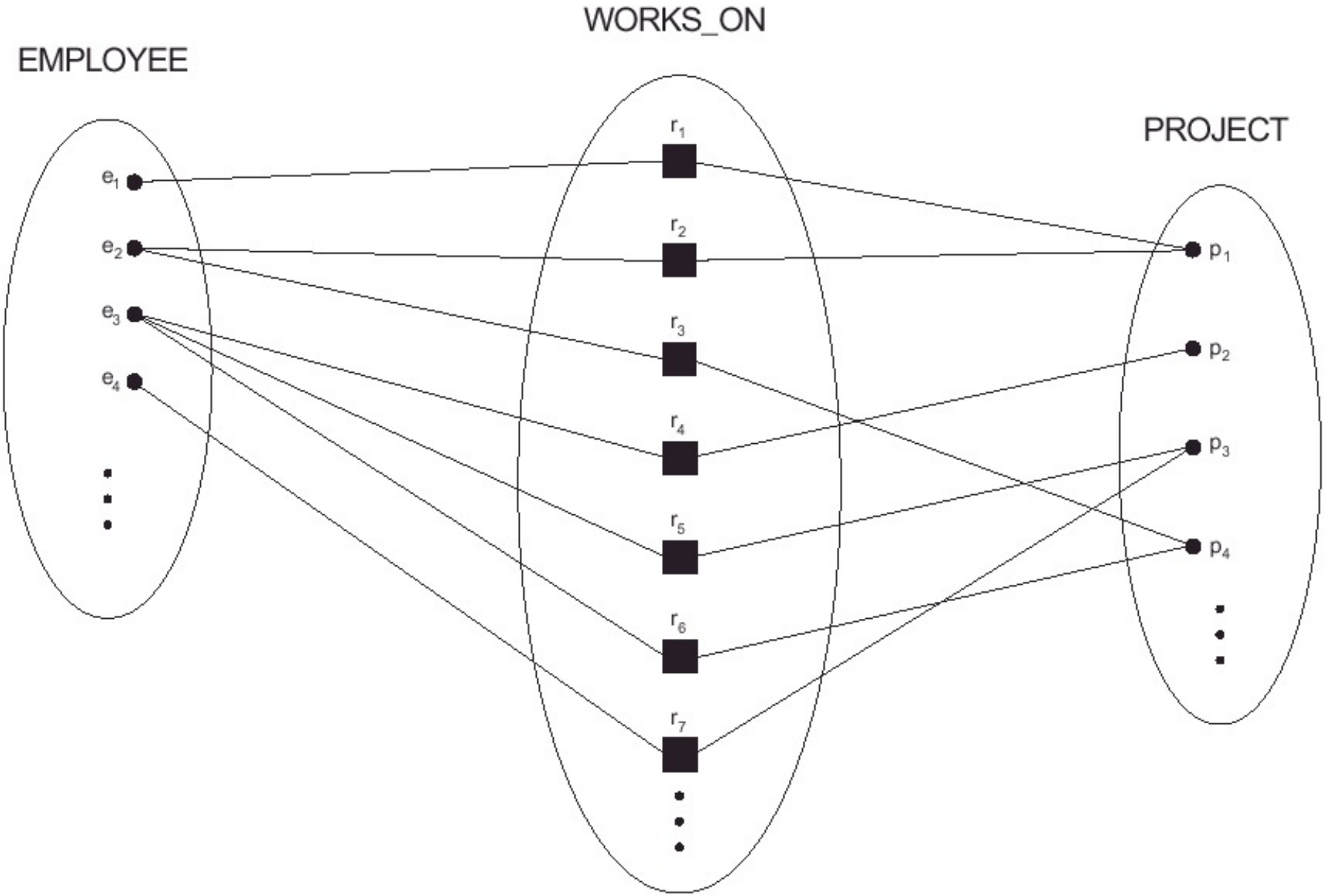
- 2 types of relationship constraints (structural constraints)
  - Cardinality ratio constraints (for binary relationships)
  - Participation constraints

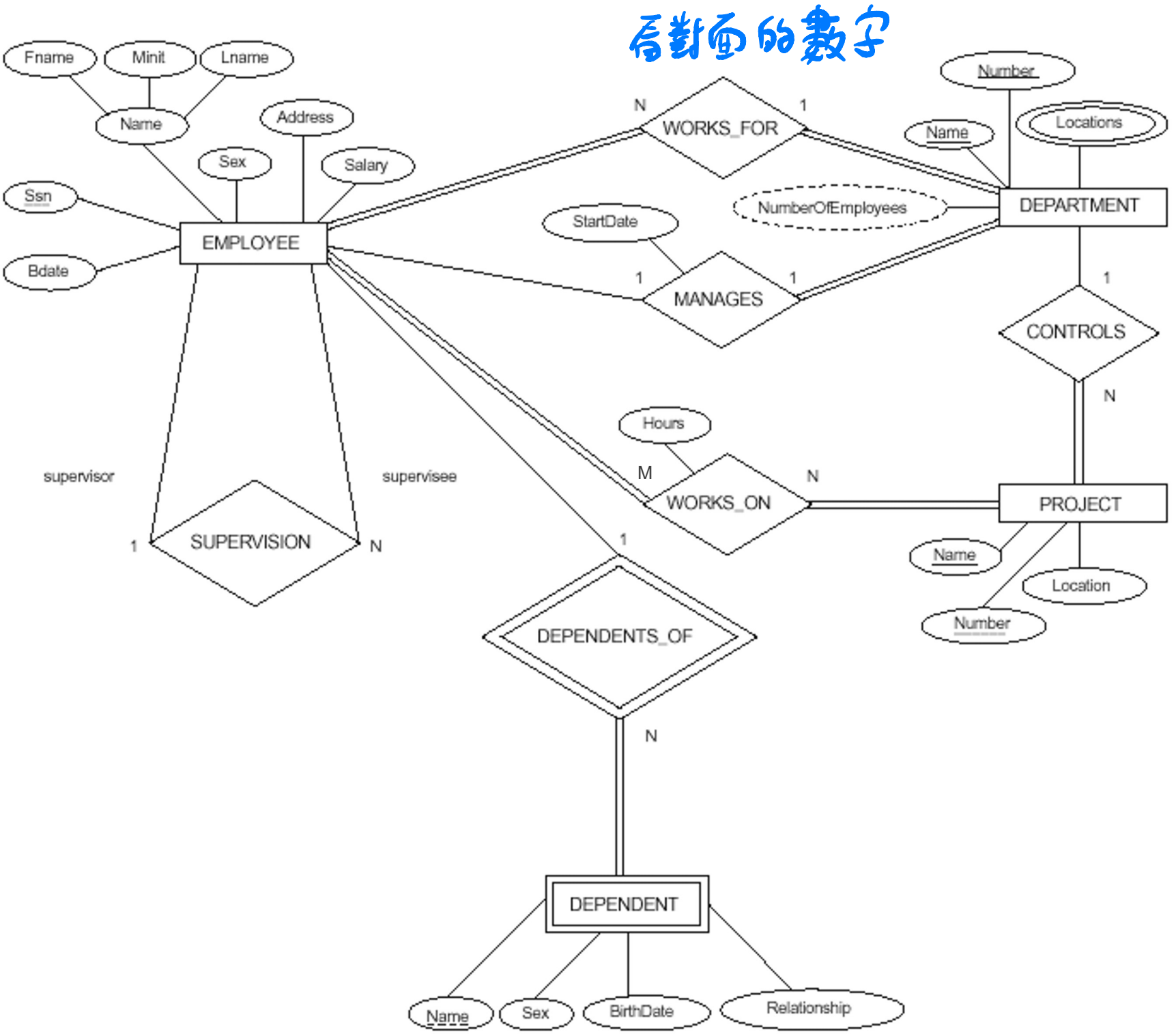
Cardinality Ratios Constraints

- Cardinality Ratios constraints
  - For binary relationship
  - Specifies the number of relationship instances that an entity can participate in
  - Possible cardinality ratios
    - 1:1 each employee can manage only one department
    - 1:N each department can be related to numerous employees, but an employee can be related to only one department
    - N:1
    - M:N an employee can work on several projects & a project can have several employees



r1 = (e1, d1)
r2 = (e2, d2)
r3 = (e3, d1)
r4 = (e4, d2)
r5 = (e5, d3)
r6 = (e6, d1)
r7 = (e7, d3)


EMPLOYEE
WORKS_ON
PROJECT
e1
e2
e3
e4
r1
r2
r3
r4
r5
r6
r7
p1
p2
p3
p4


看對面的數字
Fname
Minit
Lname
Name
Address
Sex
Salary
Ssn
Bdate
EMPLOYEE
N
WORKS_FOR
1
Number
Name
Locations
DEPARTMENT
NumberOfEmployees
StartDate
1
MANAGES
1
1
CONTROLS
N
Hours
M
N
WORKS_ON
PROJECT
supervisor
supervisee
1
SUPERVISION
N
1
Name
Location
Number
DEPENDENTS_OF
N
DEPENDENT
Name
Sex
BirthDate
Relationship

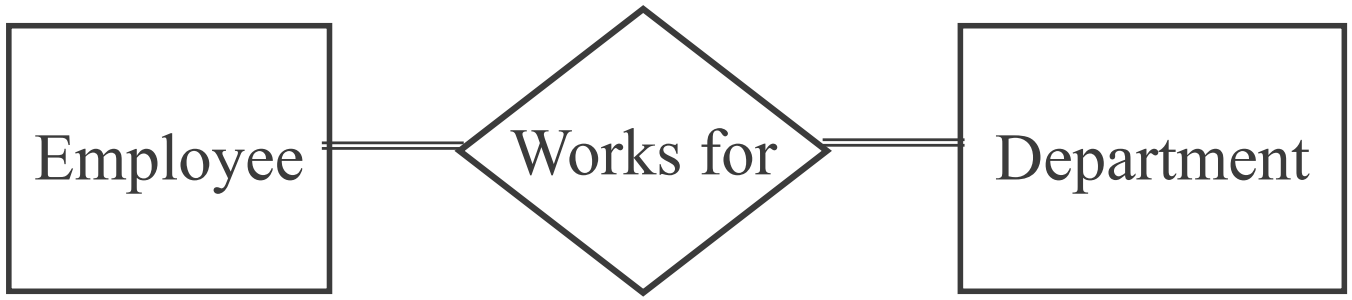
Participation Constraints

- Participation constraints
  - Specifies whether the existence of an entity depends on its being related to another entity via the relationship type
  - Two types of participation
    - Total participation (Existence Dependency)
    - Partial participation

Participation Constraints (cont.)

- Total participation (Existence Dependency)
  - If a company policy states that every employee must work for a department

    ➔ An employee entity can exist only if it participate in a Works_For relationship instance
  - In ER diagram, total participation is represented as a double line connecting the participating entity type to the relationship



Participation Constraints (cont.)

- Partial participation
  - Not every employee manages a department
  - Participation of Employee in the Manages relationship type is partial
  - In ER diagram, partial participation is represented as a single line connecting the participating entity type to the relationship

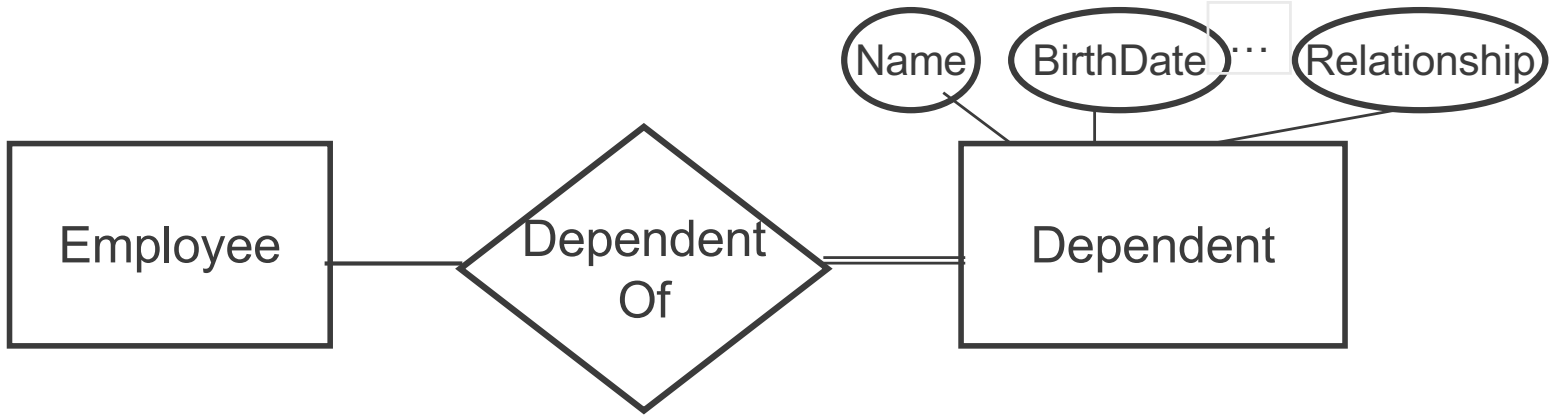
Attributes of Relationship Types

- Relationship types can also have attributes.
  - E.g. Employee–Works_On(Hour)–Project
  - E.g. Employee–Manages(StartDate)–Department

Weak Entity Types

Weak Entity Types

- Weak entity types
  - Entity types that do not have key attributes of their own
  - Entities belonging to a weak entity type are identified by being related to specific entities from another entity type in combination with some of their attribute values
  - Owner(parent) entity type: Employee



Weak Entity Types (cont.)

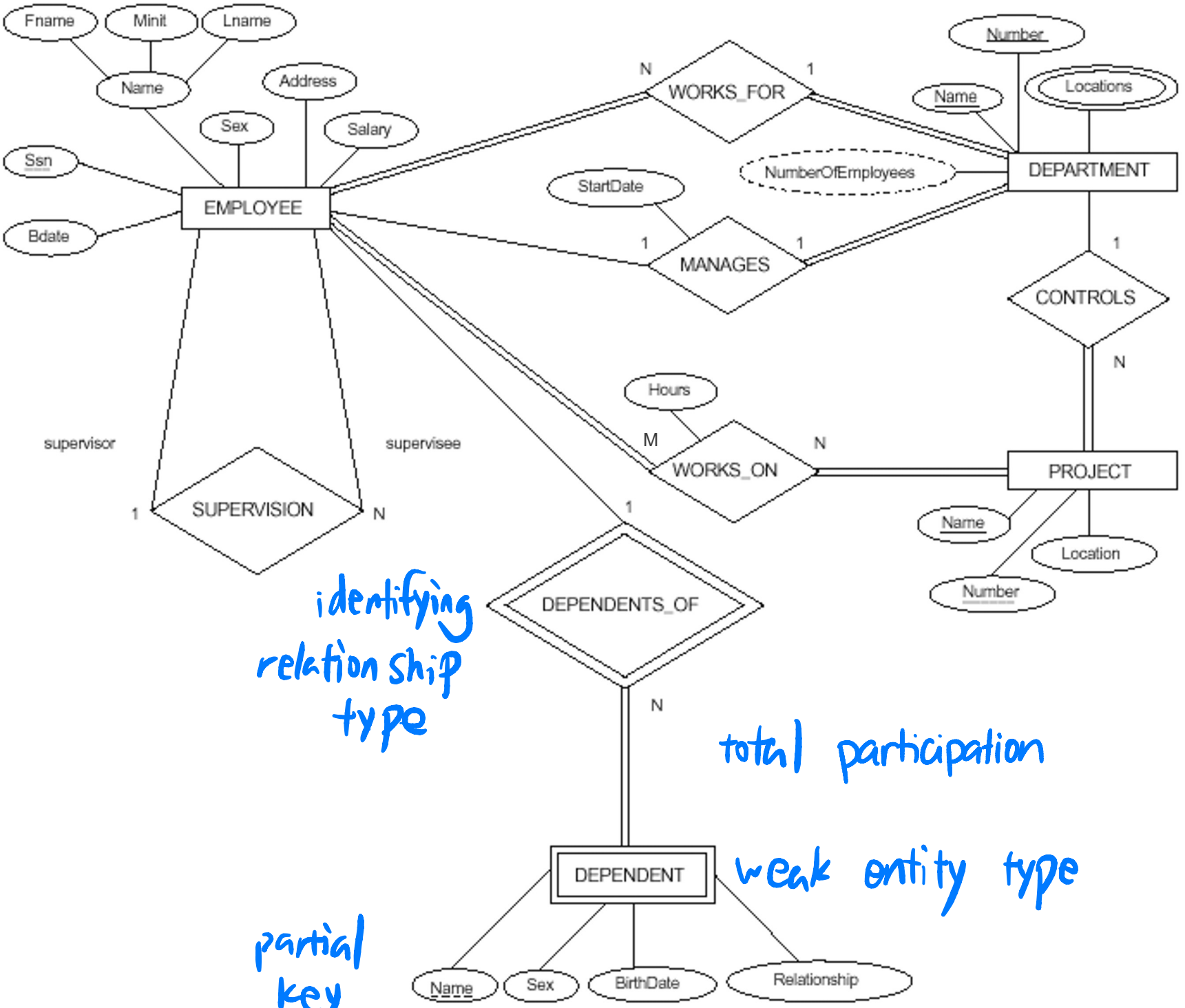
- Weak entity types
  - A weak entity type always has a total participation constraint with respect to its identifying relationship
  - a weak entity cannot be identified without an owner entity
  - Not every total participation result in a weak entity type
    - E.g. Driver_License cannot exist without Person entity

      Driver_License has its own key(LicenseNumber)

      Driver_License is not a weak entity

Weak Entity Types (cont.)

- Weak entity types
  - A weak entity type has a partial key
  - Partial key: the set of attributes that can uniquely identify weak entities that are related to the same owner entity
  - e.g. No two dependents of the same employee have the same first name ➔ Name is the partial key
  - In ER diagrams
    - Weak entity type & identifying relationship are distinguished by surrounding boxes and diamonds with double lines
    - Partial key attribute is underlined with a dashed or dotted line


Fname
Minit
Lname
Name
Address
Sex
Salary
Ssn
Bdate
EMPLOYEE
N
WORKS_FOR
1
Number
Name
Locations
DEPARTMENT
NumberOfEmployees
StartDate
1
MANAGES
1
1
CONTROLS
N
Hours
M
WORKS_ON
N
PROJECT
Name
Location
Number
supervisor
supervisee
1
SUPERVISION
N
1
DEPENDENTS_OF
N
DEPENDENT
Name
Sex
BirthDate
Relationship
identifying relationship type
total participation
weak entity type
partial key

Refining the ER Design

Refining the ER diagram for the Company DB

- Specify relationship types
  - Manages
    - 1:1 relationship type between Employee & Department
    - Employee participation is partial
    - Department is full participation (a department must have a manager)
    - StartDate is assigned to Manages
  - Works_For
    - 1:N relationship type between Department & Employee
    - Both are total participation
  - Controls
    - 1:N relationship type between Department & Project
    - Participation of Project is total
    - Participation of Department is partial

Refining the ER diagram for the Company DB (cont.)

- Supervision
  - 1:N relationship between Employee & Employee
  - Both are partial (not every employee is a supervisor & not every employee has a supervisor)
- Works_On
  - M:N relationship type with attribute Hours
  - Both are total participation (a project can have several employee working on)
- Dependents_Of
  - 1:N relationship type between Employee & Dependent
  - is also the identifying relationship for the weak entity type Dependent
  - Participation of Employee is partial
  - Participation of Dependent is total

- Remove from the entity types all attributes that have been refined into relationships

Relationships as Attributes

- It is sometimes convenient to think of a relationship type in terms of attributes
  - E.g.

  (1) Employee

  Name, SSN, Sex, Address, Salary, BirthDate,

  Department, Supervisor, {WorksON (project, Hours)}

  (2) Department

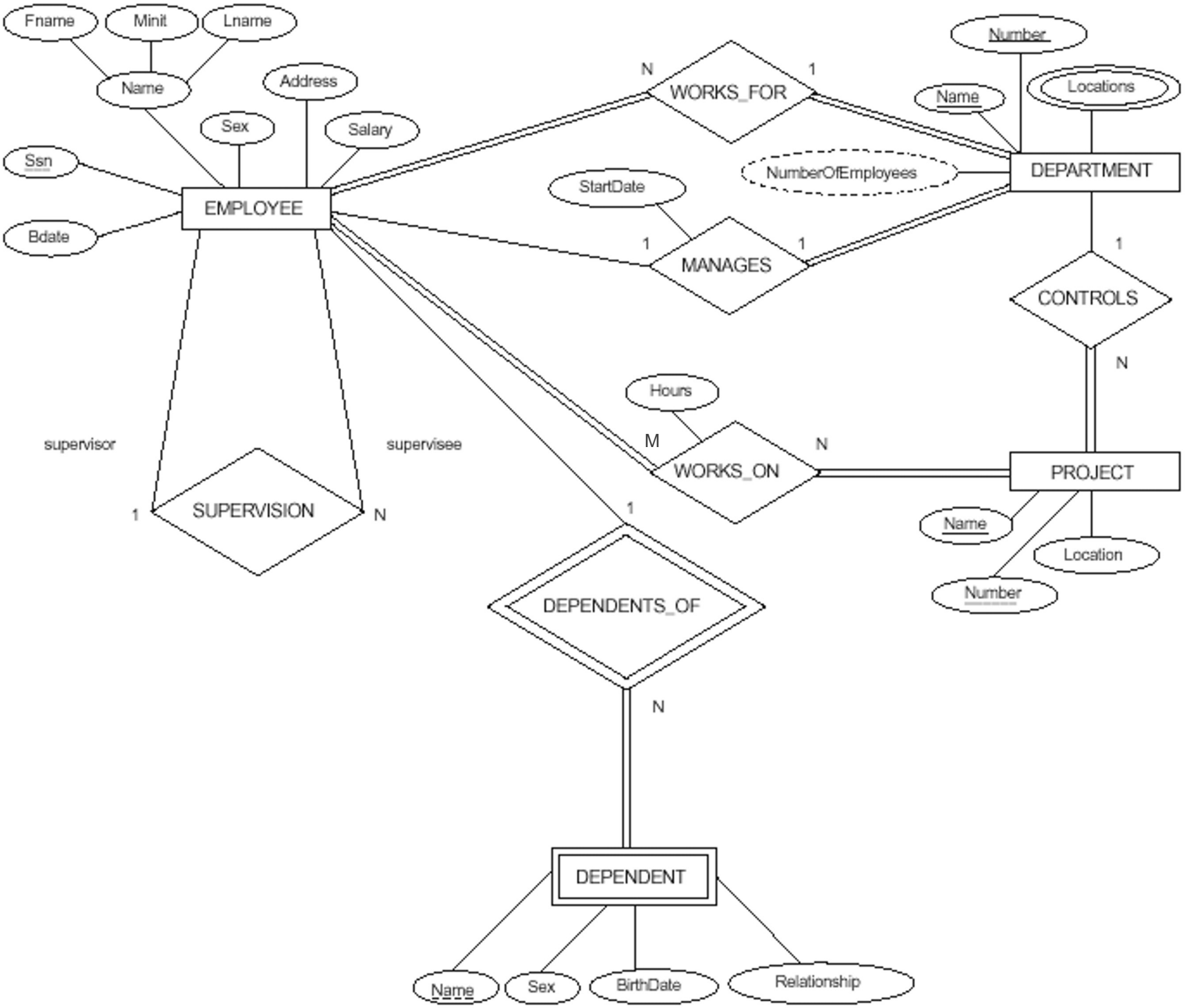
  Name, Number, {Locations}, Manager, ManagerStartDate, {Employee}

Design Choices for ER Conceptual Design

- To decide whether to model as an entity type, attribute or a relationship
  - Attribute ➔ Relationship (attribute is a reference to another entity type, WORKS_On)
  - Attribute ➔ Entity type (attribute exists in several entity types, e.g. PROJECT)
  - Entity type ➔ Attribute (attribute is related to only one other entity type)

Design Choices for ER Conceptual Design (cont.)

- Attributes of relationship types can be migrated to participating entity types
  - 1:1 relationship type
    - Be migrated to one of the participating entity types (total participation is preferable)
    - e.g. Employee–Manages(StartDate)–Department
      - ➔ Employee(StartDate)–Manages–Department
      - ➔ Employee–Manages–Department(StartDate)
  - 1:N relationship types
    - Be migrated to the N-side of the relationship
    - e.g. Employee–Works_For(StartDate)–Department
      - ➔ Employee(StartDate)–Works_For–Department

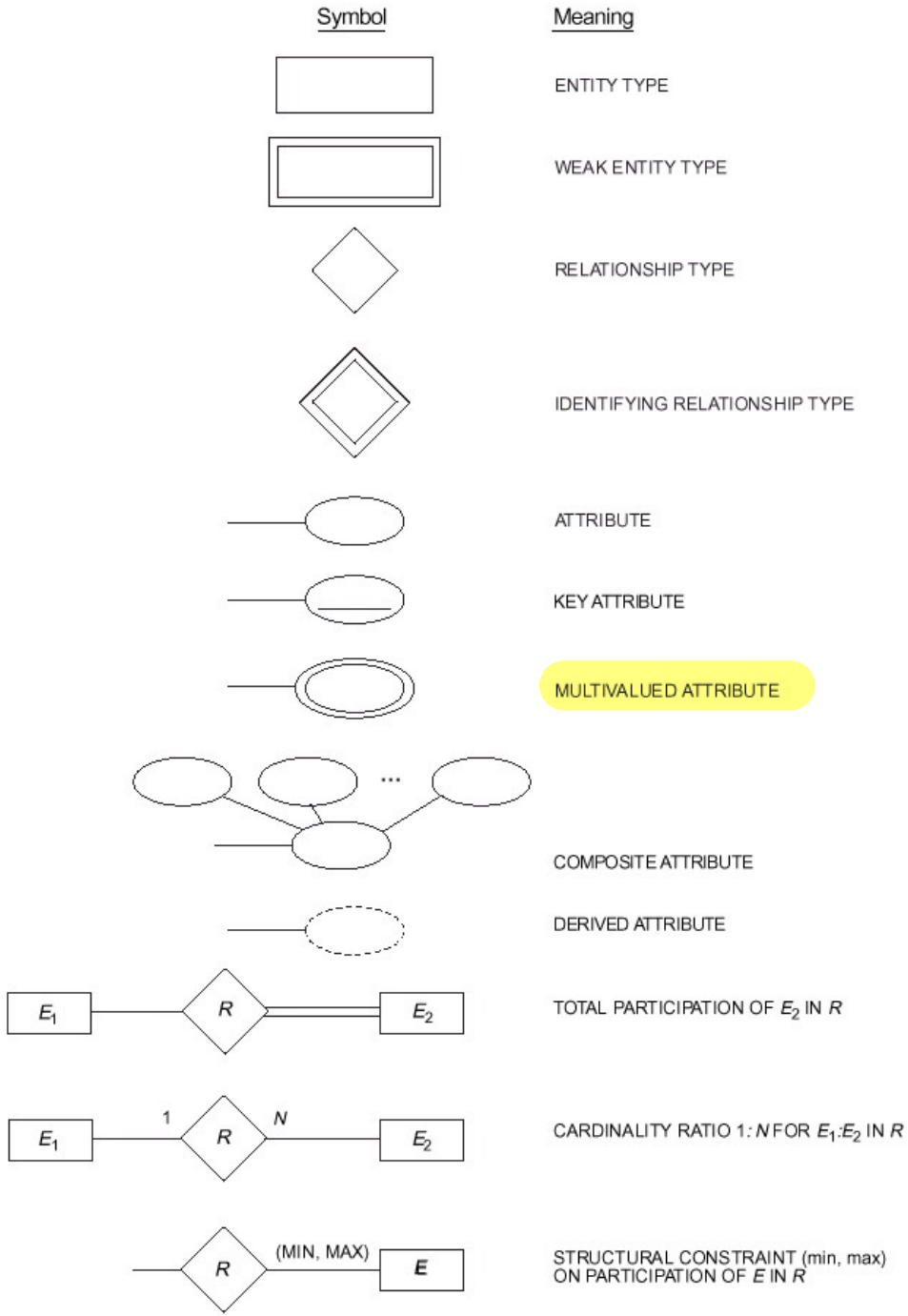

Fname
Minit
Lname
Name
Address
Sex
Salary
Ssn
Bdate
EMPLOYEE
N
WORKS_FOR
1
Number
Name
Locations
DEPARTMENT
NumberOfEmployees
StartDate
1
MANAGES
1
1
CONTROLS
N
Hours
M
N
WORKS_ON
PROJECT
Name
Location
Number
supervisor
supervisee
1
SUPERVISION
N
1
DEPENDENTS_OF
N
DEPENDENT
Name
Sex
BirthDate
Relationship

Placement of Attributes on Relationship Types

- M:N relationship
  - Cannot be migrated
  - Some attributes determined by the combination of participating entities, not by any single entity, must be specified as relationship attributes
    - E.g. Employee–Works_On(Hours)–Project

Outline

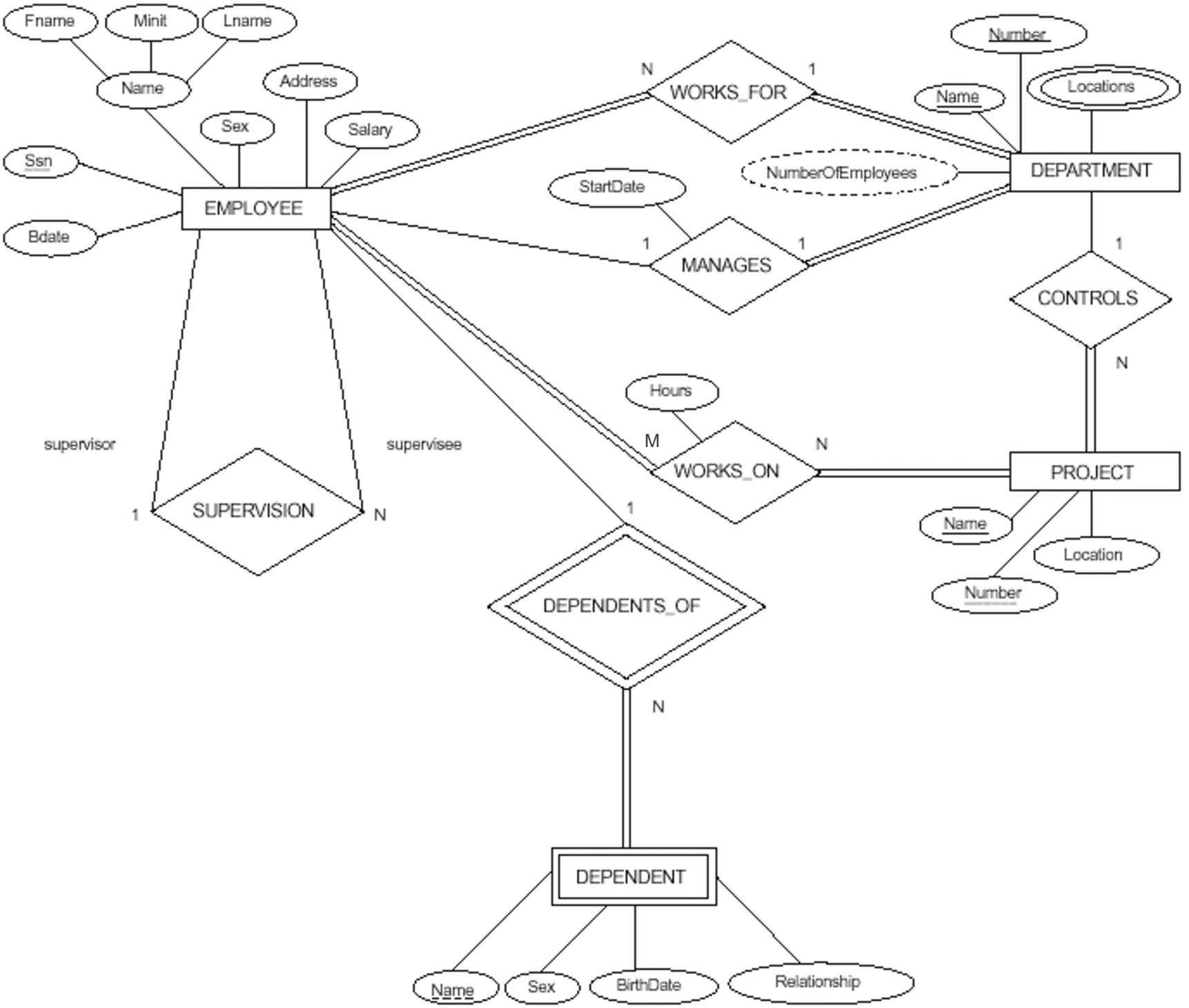
- Data Models
- Database Design Process
- Example Database Application
- Entity-Relationship Model
  - Entity Types, Entity Sets, Attributes & Keys
  - Relationships, Relationships, Roles & Structure Constraints
  - Weak Entity Types
- **ER Diagrams, Naming Conventions & Design Issues**
- Mapping ER into Relational Data Models

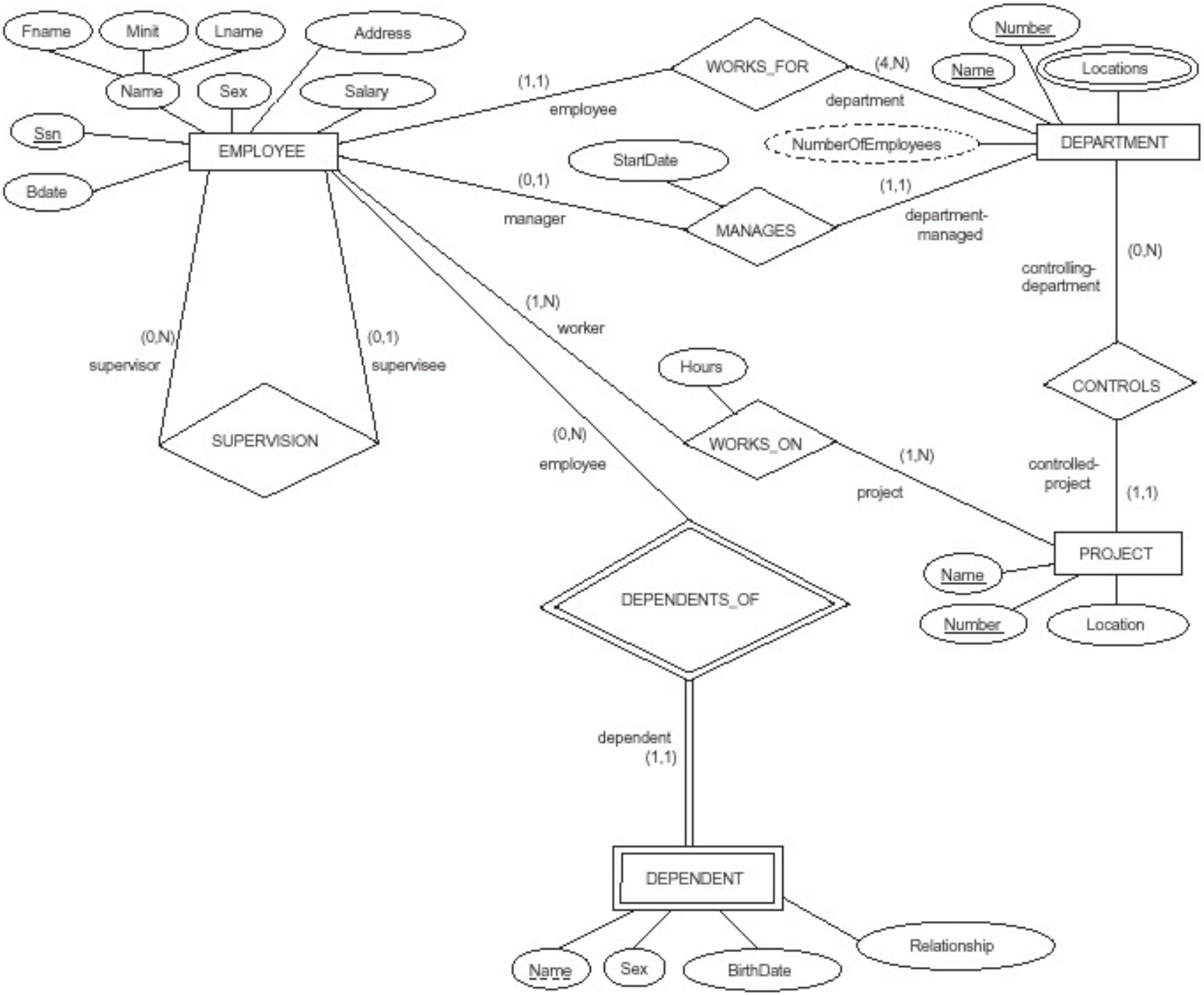

Symbol
Meaning
ENTITY TYPE
WEAK ENTITY TYPE
RELATIONSHIP TYPE
IDENTIFYING RELATIONSHIP TYPE
ATTRIBUTE
KEY ATTRIBUTE
MULTIVALUED ATTRIBUTE
COMPOSITE ATTRIBUTE
DERIVED ATTRIBUTE
E1
R
E2
TOTAL PARTICIPATION OF E2 IN R
1
N
CARDINALITY RATIO 1:N FOR E1:E2 IN R
(MIN, MAX)
E
STRUCTURAL CONSTRAINT (min, max)
ON PARTICIPATION OF E IN R

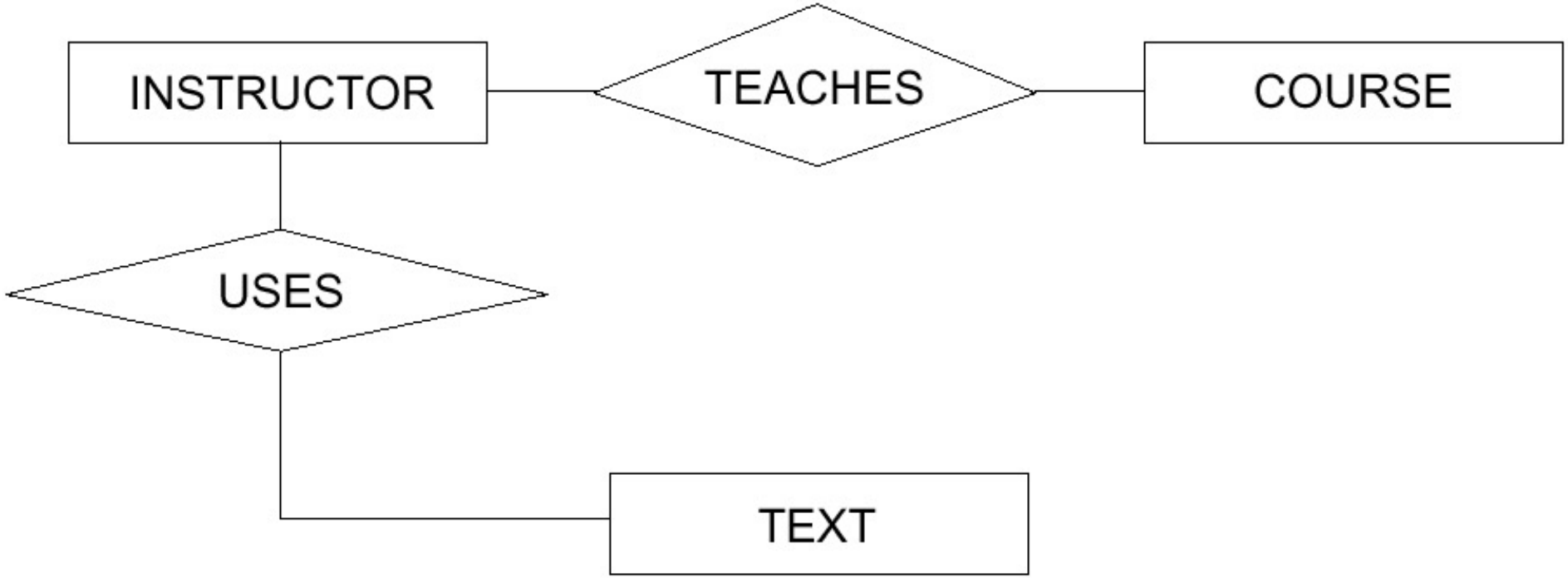
Alternative Notations for the ER Diagrams

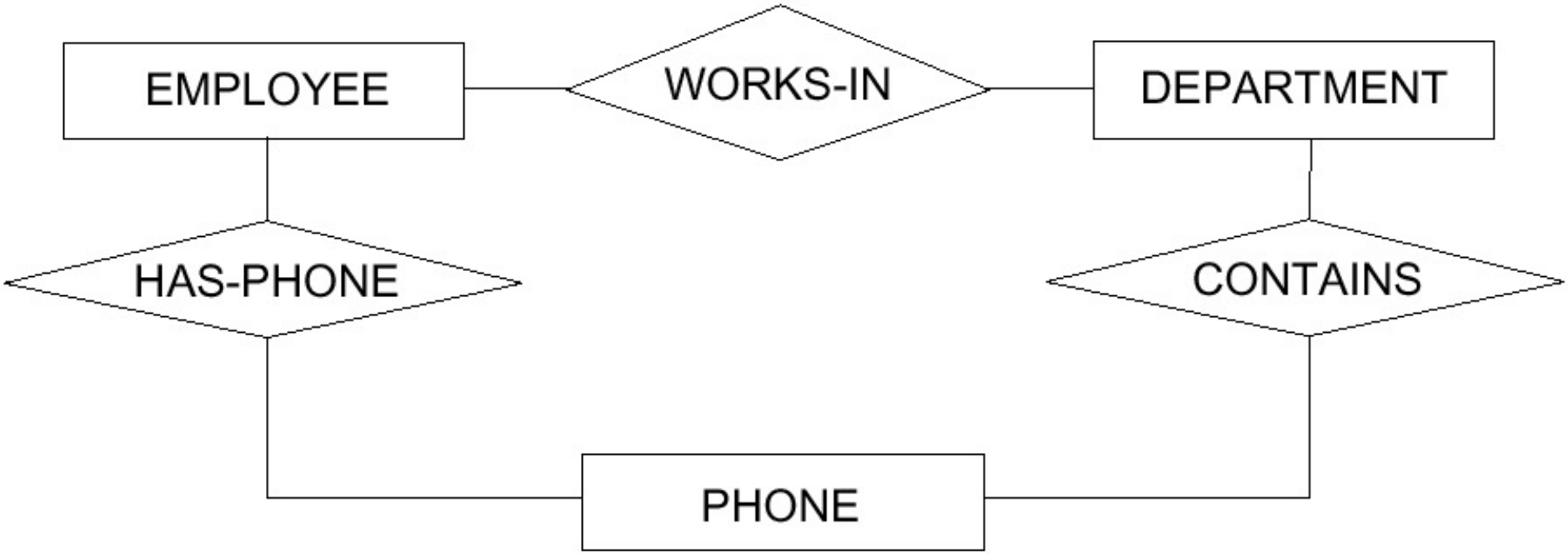
- (*min*, *max*) with each participation of an entity type E in a relationship type R
  - For each entity *e* in E, *e* must participate in at least *min* & at most *max* relationship instances in R at any point in time
  - *min* = 0 implies partial participation
  - *min* > 0 implies total participation
  - *max* specifies cardinality ~~ration~~ constraint

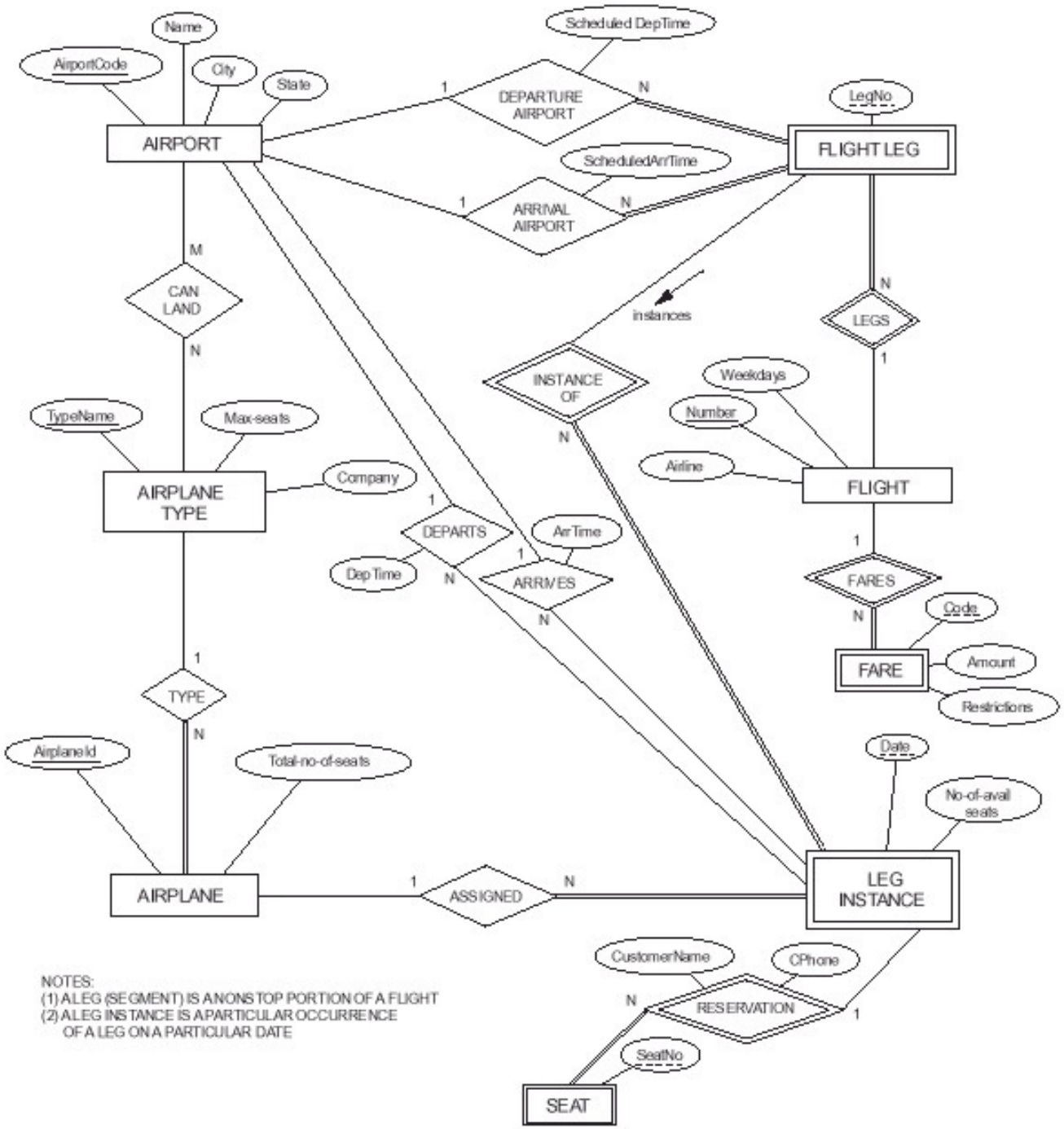
ratio

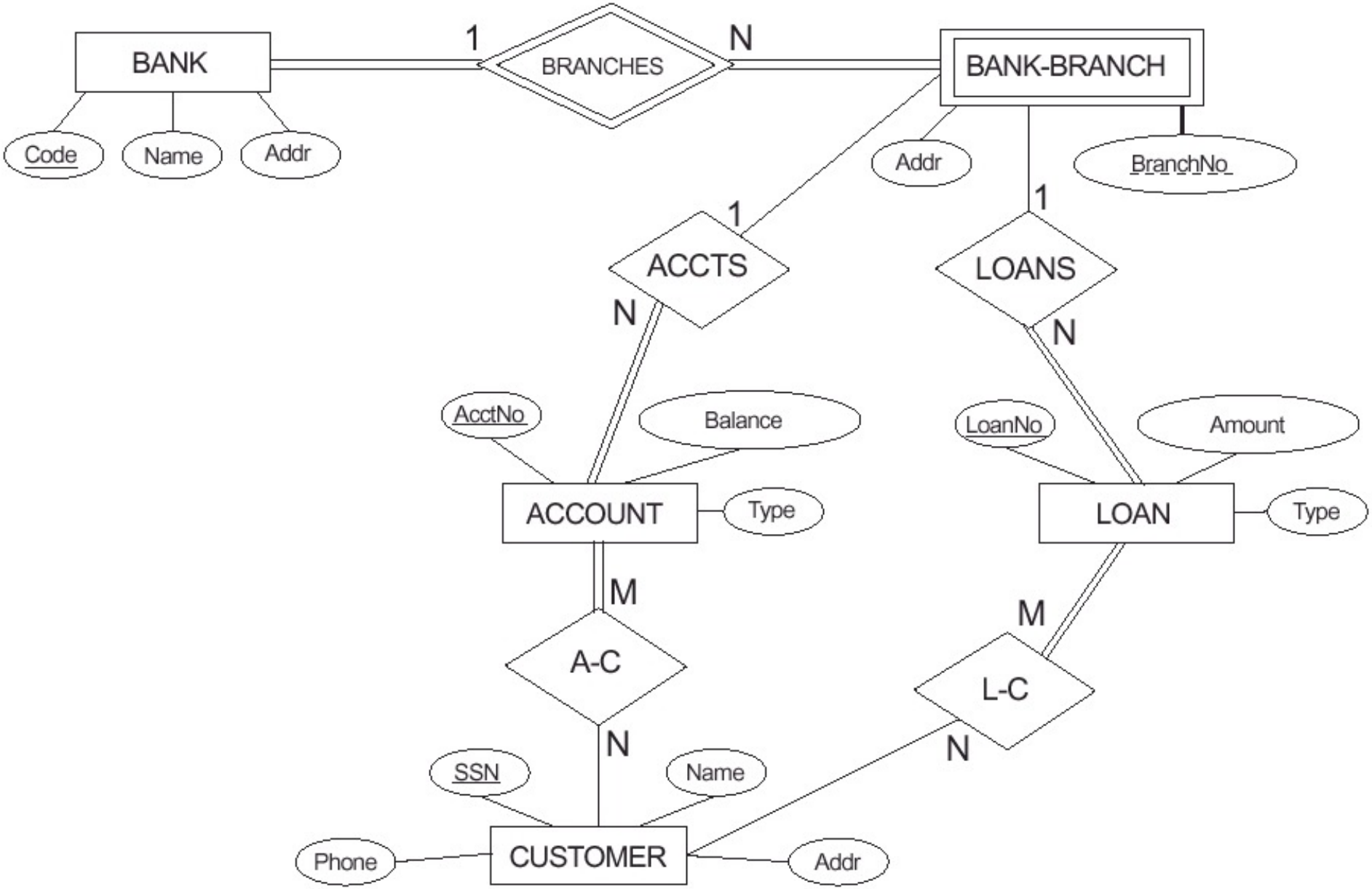

Fname
Minit
Lname
Name
Address
Sex
Salary
Ssn
Bdate
EMPLOYEE
N
WORKS_FOR
1
Number
Name
Locations
DEPARTMENT
NumberOfEmployees
StartDate
1
MANAGES
1
1
CONTROLS
N
Hours
M
N
WORKS_ON
PROJECT
Name
Location
Number
supervisor
supervisee
1
SUPERVISION
N
1
DEPENDENTS_OF
N
DEPENDENT
Name
Sex
BirthDate
Relationship


Fname
Minit
Lname
Address
Name
Sex
Salary
Ssn
Bdate
EMPLOYEE
WORKS_FOR
(1,1)
employee
(4,N)
department
Number
Name
Locations
DEPARTMENT
NumberOfEmployees
StartDate
(0,1)
manager
MANAGES
(1,1)
department-
managed
(0,N)
controlling-
department
(1,N)
worker
(0,N)
supervisor
(0,1)
supervisee
SUPERVISION
Hours
CONTROLS
WORKS_ON
(0,N)
employee
(1,N)
project
controlled-
project
(1,1)
PROJECT
Name
Number
Location
DEPENDENTS_OF
dependent
(1,1)
DEPENDENT
Name
Sex
BirthDate
Relationship


INSTRUCTOR
TEACHES
COURSE
USES
TEXT


EMPLOYEE
WORKS-IN
DEPARTMENT
HAS-PHONE
CONTAINS
PHONE


Name
AirportCode
City
State
AIRPORT
Scheduled DepTime
DEPARTURE AIRPORT
1
N
ScheduledArrTime
ARRIVAL AIRPORT
1
N
LegNo
FLIGHT LEG
M
CAN LAND
N
instances
N
LEGS
1
INSTANCE OF
N
Weekdays
Number
Airline
FLIGHT
TypeName
Max-seats
AIRPLANE TYPE
Company
1
DEPARTS
DepTime
N
ArrTime
1
ARRIVES
N
1
FARES
N
Code
FARE
Amount
Restrictions
1
TYPE
N
AirplaneId
Total-no-of-seats
AIRPLANE
1
ASSIGNED
N
Date
No-of-avail seats
LEG INSTANCE
CustomerName
CPhone
N
RESERVATION
1
SeatNo
SEAT
NOTES:
(1) A LEG (SEGMENT) IS A NONSTOP PORTION OF A FLIGHT
(2) A LEG INSTANCE IS A PARTICULAR OCCURRENCE
OF A LEG ON A PARTICULAR DATE


BANK
1
BRANCHES
N
BANK-BRANCH
Code
Name
Addr
Addr
BranchNo
1
1
ACCTS
LOANS
N
N
AcctNo
Balance
LoanNo
Amount
ACCOUNT
Type
LOAN
Type
M
M
A-C
L-C
N
N
SSN
Name
Phone
CUSTOMER
Addr

Summary

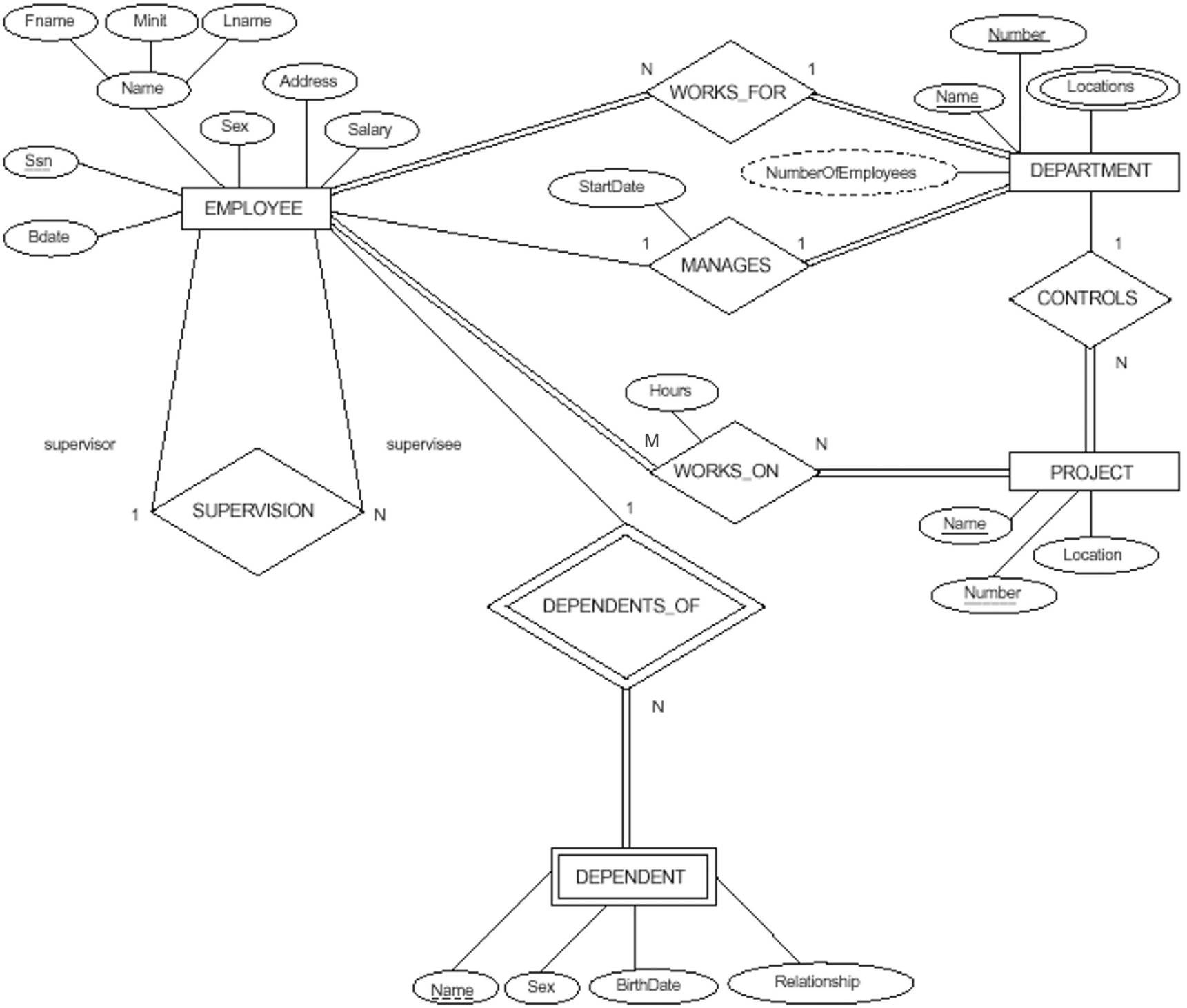
- Entity-Relationship Model
  - Entity Types
    - Attributes
      - Simple vs. composite
      - Single-valued vs. multivalued
      - Stored vs. derived
    - Domain, Null
    - Key attributes
  - Relationship Types
    - Cardinality ratio constraints
    - Participation constraints
      - Total participation
      - Partial participation

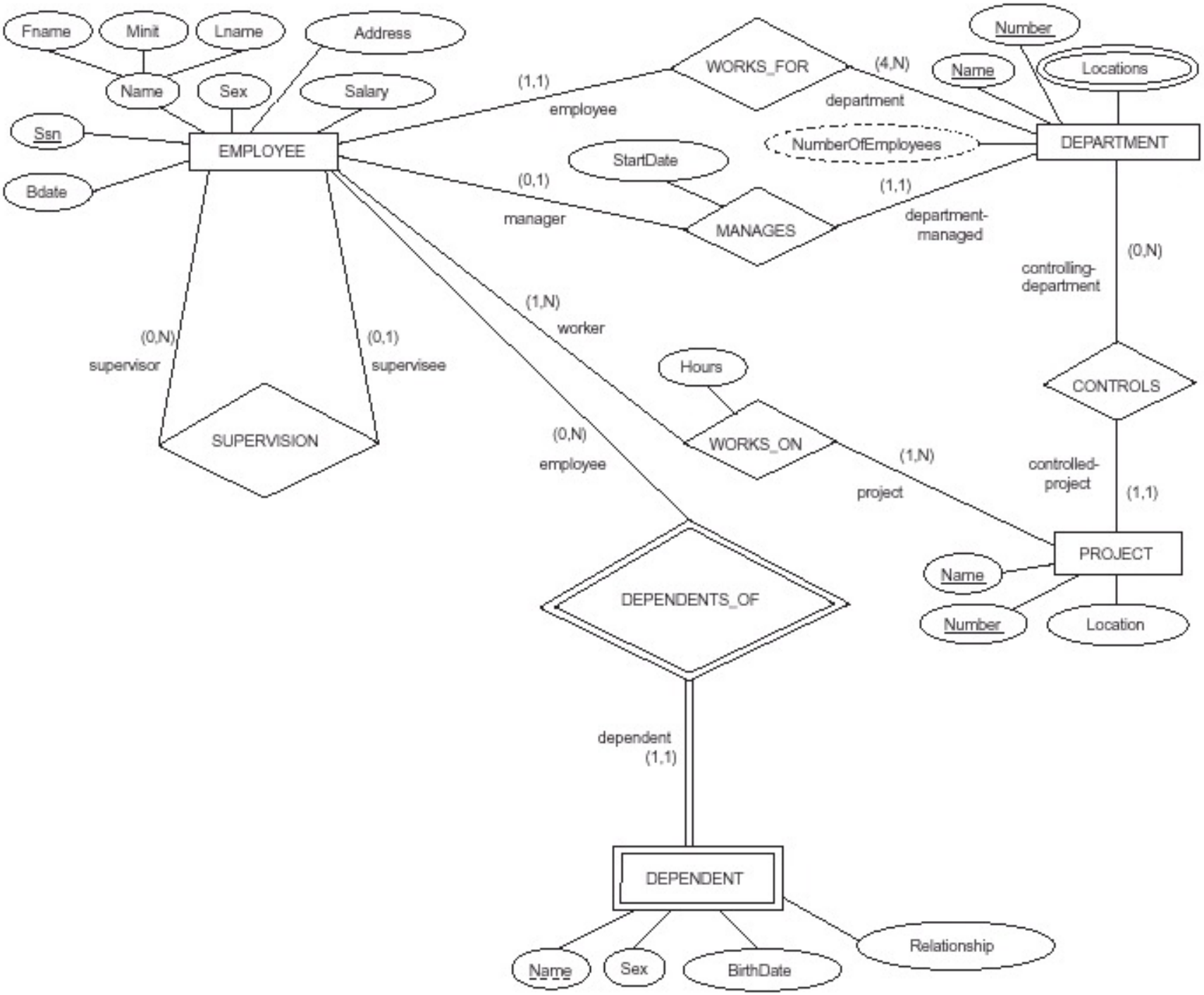
An Example Database Application

- keeps track of a company's employees, departments & projects
  - The company is organized into departments.
  - Each department has a unique name, a unique number & 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 & a single location
  - We store each employee's name, social security number, address, salary, sex & birth date

An Example Database Application (cont.)

- 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 keep track of the direct supervisor of each employee.
- We keep track of the dependents of each employee for insurance purpose.
- We keep each dependent's first name, sex, birth date & relationship to the employee


Fname
Minit
Lname
Name
Address
Sex
Salary
Ssn
Bdate
EMPLOYEE
N
WORKS_FOR
1
Number
Name
Locations
DEPARTMENT
NumberOfEmployees
StartDate
1
MANAGES
1
1
CONTROLS
N
Hours
M
N
WORKS_ON
PROJECT
Name
Location
Number
supervisor
supervisee
1
SUPERVISION
N
1
DEPENDENTS_OF
N
DEPENDENT
Name
Sex
BirthDate
Relationship


Fname
Minit
Lname
Address
Name
Sex
Salary
Ssn
Bdate
EMPLOYEE
WORKS_FOR
(1,1)
employee
(4,N)
department
Number
Name
Locations
DEPARTMENT
NumberOfEmployees
StartDate
(0,1)
manager
MANAGES
(1,1)
department-
managed
(0,N)
controlling-
department
(1,N)
worker
(0,N)
supervisor
(0,1)
supervisee
SUPERVISION
Hours
CONTROLS
WORKS_ON
(0,N)
employee
(1,N)
project
controlled-
project
(1,1)
PROJECT
Name
Number
Location
DEPENDENTS_OF
dependent
(1,1)
DEPENDENT
Name
Sex
BirthDate
Relationship

ER Model 有

Entity Types, Relationship Types,

Relational Data Model 呢？

ER Model 有

Stored vs. Derived Attributes之區別，

Relational Data Model 有嗎？ 沒有

ER Model 有 Cardinality Ratio Constraint,

Participation Constraint，

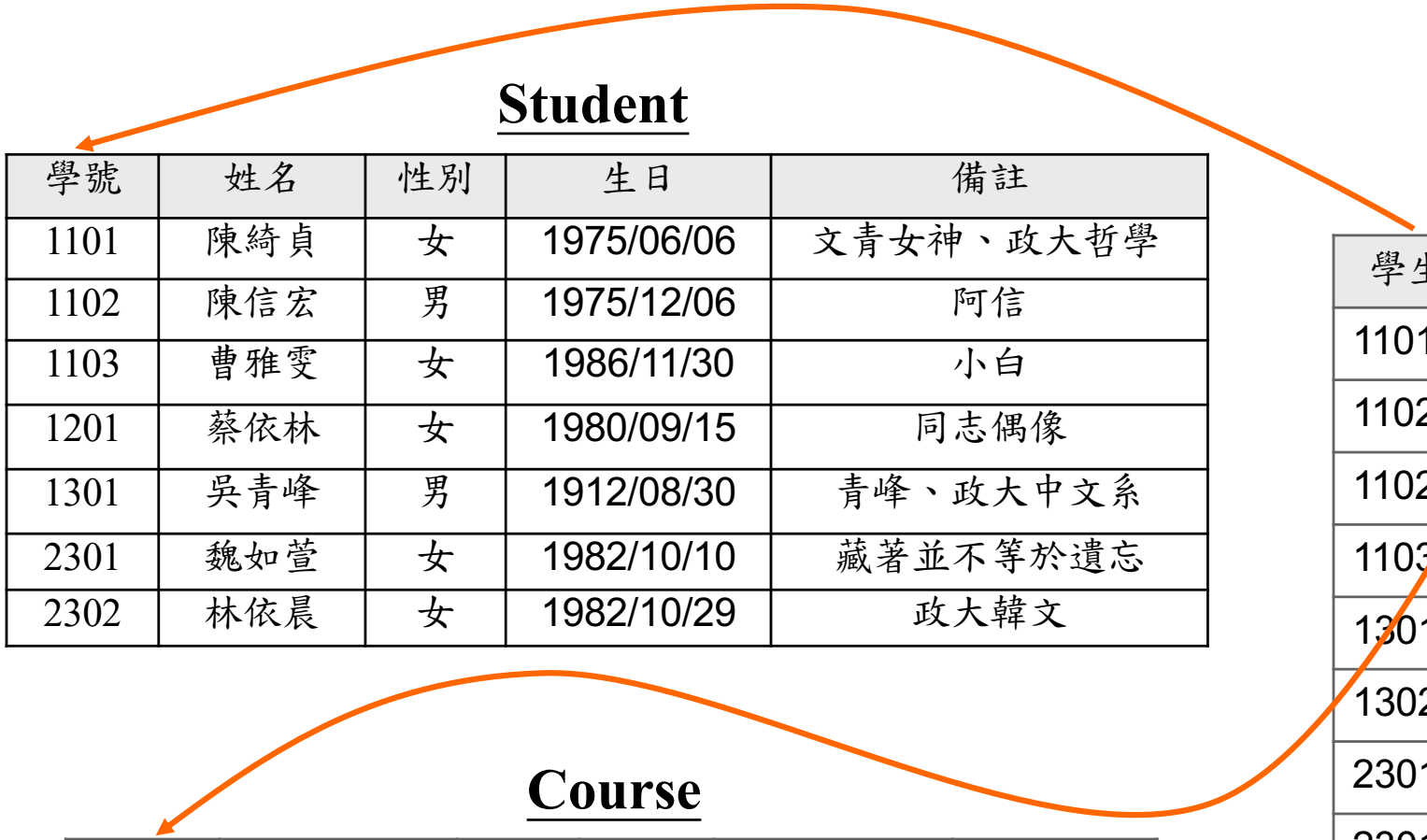
Relational Data Model沒有嗎？ foreign key ?

ER Model 的 Cardinality Ratio Constraint

如何對應到

Relational Data Model 嗎？

Referential Integrity Constraints



Student

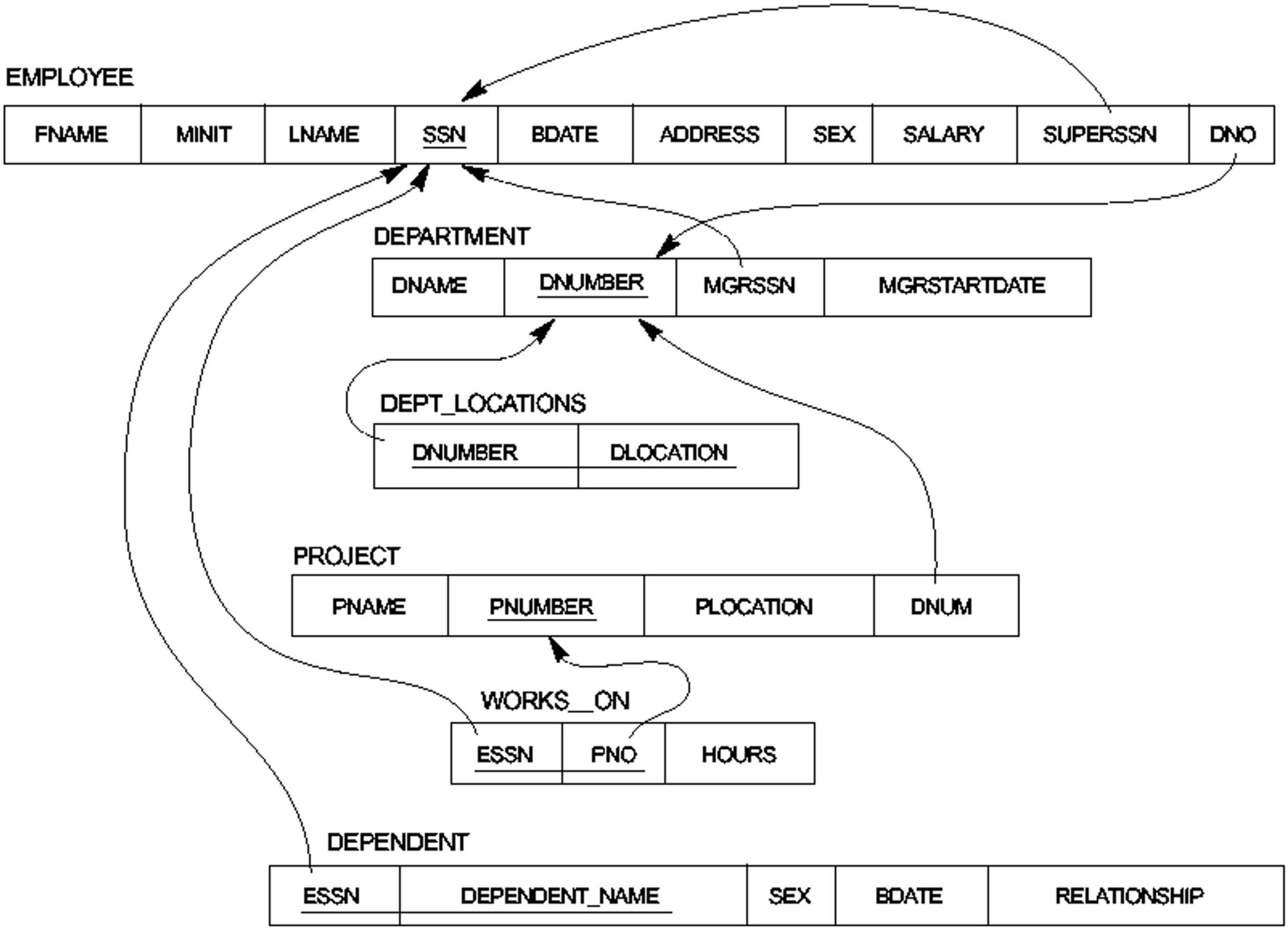
學號	姓名	性別	生日	備註
1101	陳綺貞	女	1975/06/06	文青女神、政大哲學
1102	陳信宏	男	1975/12/06	阿信
1103	曹雅雯	女	1986/11/30	小白
1201	蔡依林	女	1980/09/15	同志偶像
1301	吳青峰	男	1912/08/30	青峰、政大中文系
2301	魏如萱	女	1982/10/10	藏著並不等於遺忘
2302	林依晨	女	1982/10/29	政大韓文

SC

學生	課程	分數
1101	C3001	90
1102	C3001	70
1102	J2010	80
1103	C3001	100
1301	C3001	80
1302	J2010	85
2301	C3001	90
2301	J3020	85
2302	J2010	70
2302	J3020	80

Course

代碼	名稱	學分	時數	人數限制	選課人數
C3001	資料庫系統	3	3	80	83
J2010	資料探勘	3	3	50	30
J3020	人工智慧	3	3	70	70
X5010	制服誘惑	4	1	100	300


EMPLOYEE
FNAME
MINIT
LNAME
SSN
BDATE
ADDRESS
SEX
SALARY
SUPERSSN
DNO
DEPARTMENT
DNAME
DNUMBER
MGRSSN
MGRSTARTDATE
DEPT_LOCATIONS
DNUMBER
DLOCATION
PROJECT
PNAME
PNUMBER
PLOCATION
DNUM
WORKS_ON
ESSN
PNO
HOURS
DEPENDENT
ESSN
DEPENDENT_NAME
SEX
BDATE
RELATIONSHIP

ER Model vs.
Relational Data Model
哪個 Model 比較簡單？

Model 越簡單越好，

還是越複雜越好？簡單

為什麼 ER Model

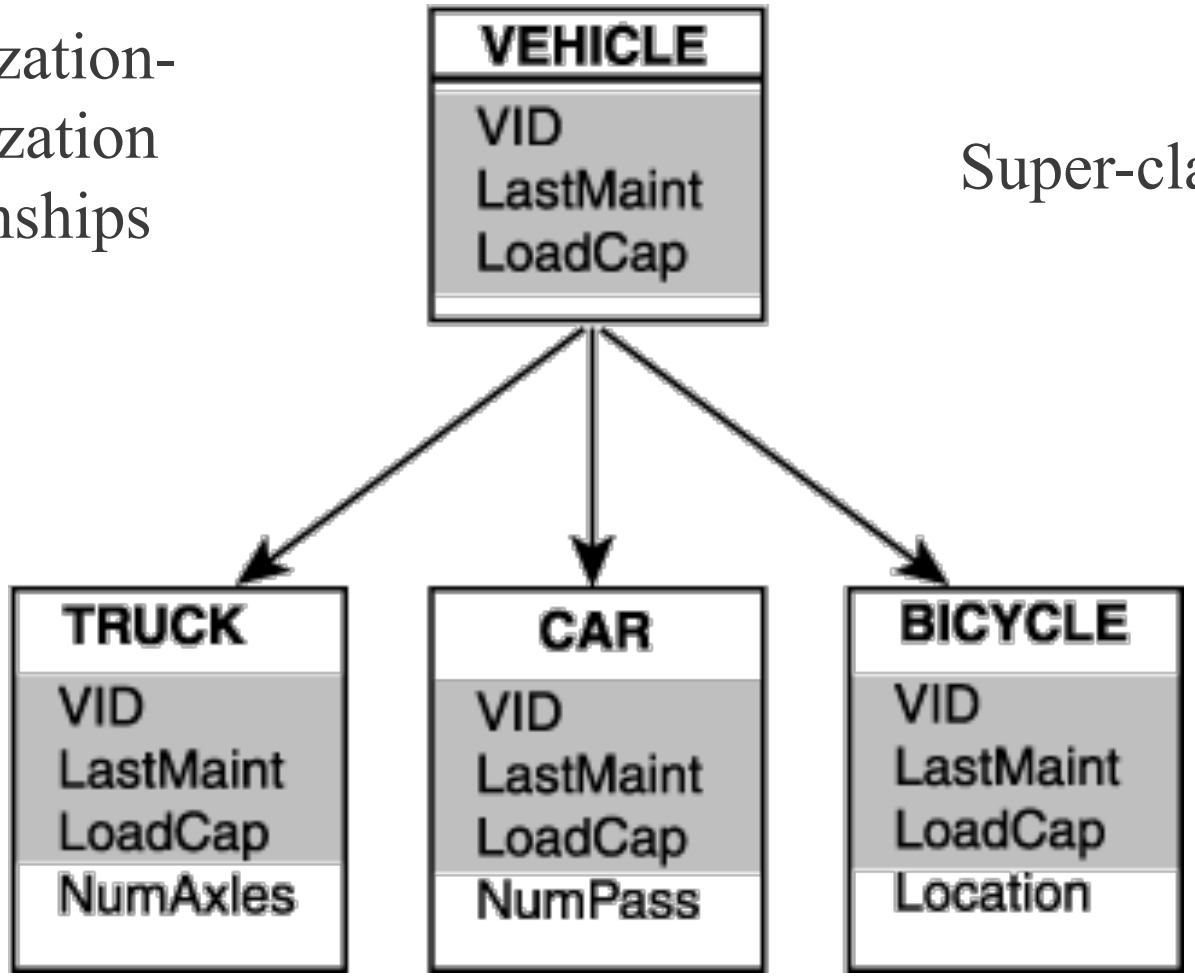
是 Conceptual Model,

而 Relational Data Model 是

Implementation Model ？

ER Model 有
Entity Types, Relationship Types,
有什麼 Relationship 的資訊是
ER Model 無法描述的呢？

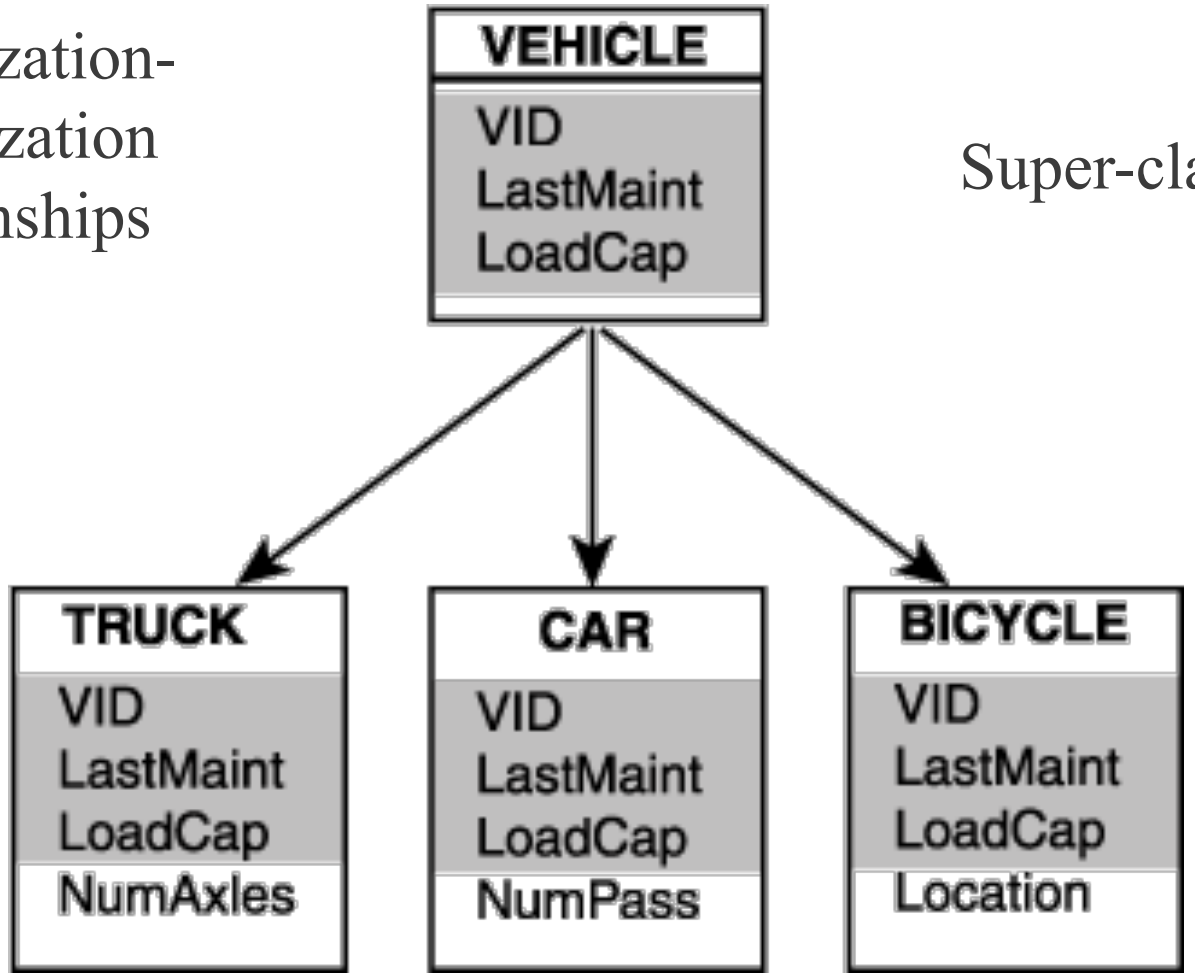
Generalization-
Specialization
Relationships



Super-class

Sub-class

Generalization-
Specialization
Relationships



Super-class

Sub-class