Chapter 3
Relational Data Model

Content

- Introduction
- Concepts
- Constraints
- From E/R diagram to relational design

Introduction

- Was first introduced by E. F. Codd
  - "A Relation Model for Large Shared Data Banks", Communications of ACM, 1970
- Commercial implementation
  - By IBM
    - System R (1974), SQL/DS (1981), DB2 (1983)
  - Oracle (1979)
  - By Sybase
    - SQL Server (1987), Adaptive Server Enterprise (1996)
  - By Microsoft
    - SQL Server (1989)
    - Access (1992)

Introduction

- Open source implementation
  - MySQL
    - By MySQL AB, 1995
  - PostgreSQL
    - Ingres project at the University of California, Berkeley, 1980s
    - By many developers, released in 1996
  - SQLite
    - By D. Richard Hipp working for General Dynamics, 2000

Introduction

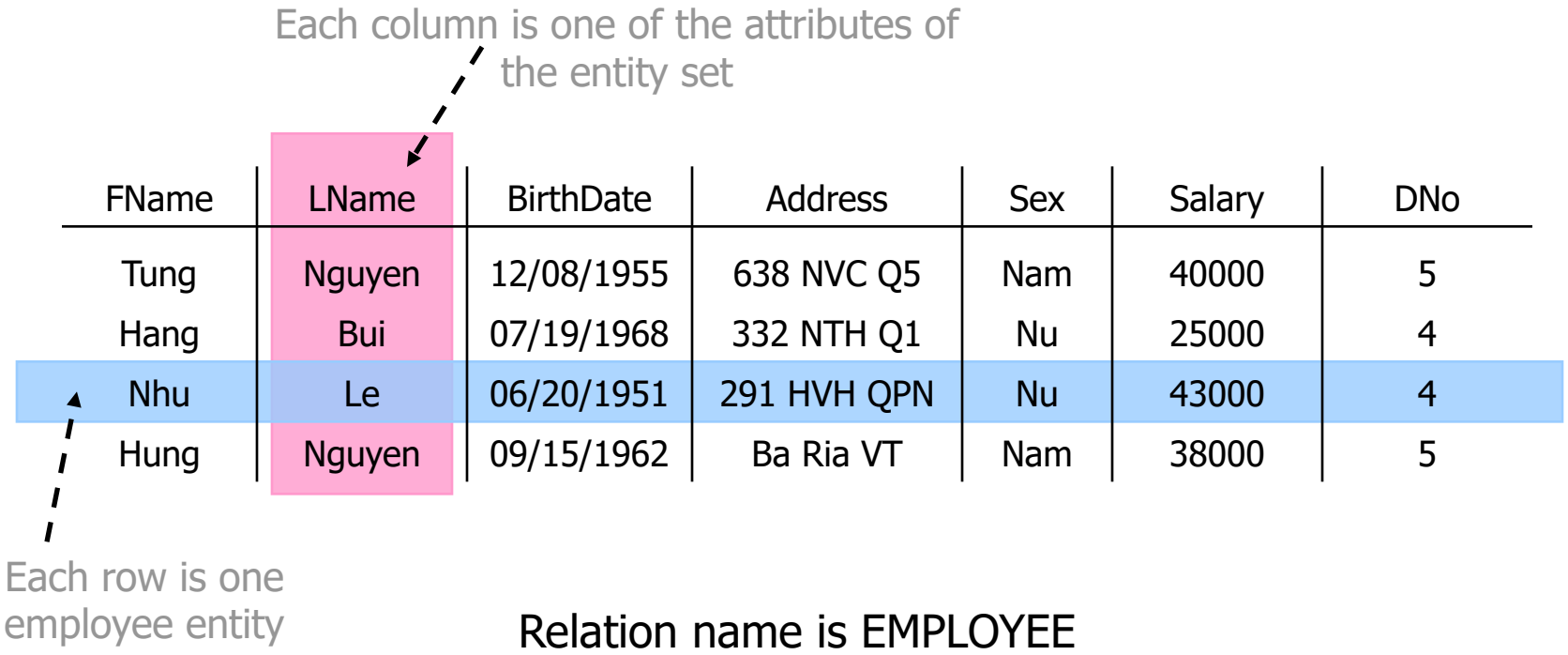
- Provide a simple way to represent data
  - The relation: a two-dimensional table
- The theoretical background
  - Set theory of mathematical logic

Content

- Introduction
- **Concepts**
  - Relation
  - Attribute
  - Schema
  - Tuple
  - Domain
  - Characteristics of relation
  - Notations
- Contraints
- From E/R diagram to relational design

Relation

- Relational model presents the DB as a collection of ***relations***
  - A relation = a two-dimensional table



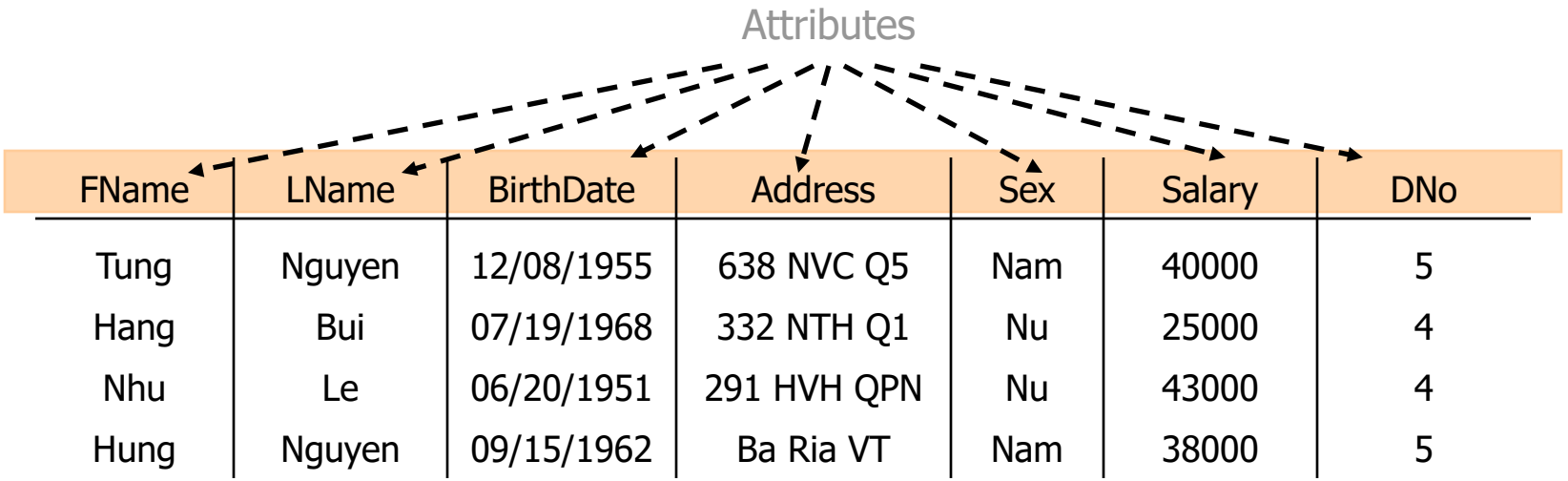
FName	LName	BirthDate	Address	Sex	Salary	DNo
Tung	Nguyen	12/08/1955	638 NVC Q5	Nam	40000	5
Hang	Bui	07/19/1968	332 NTH Q1	Nu	25000	4
Nhu	Le	06/20/1951	291 HVH QPN	Nu	43000	4
Hung	Nguyen	09/15/1962	Ba Ria VT	Nam	38000	5

Relation

- Includes
  - Name
  - Set of columns
    - Fixed
    - Named
    - Has data types
  - Set of rows
    - Changed by time
- A row ~ A real-world entity or relationship
- A relation ~ An entity set or relationship

Attribute

- The names for columns of the relation
- Describes the meaning of entries in the column below

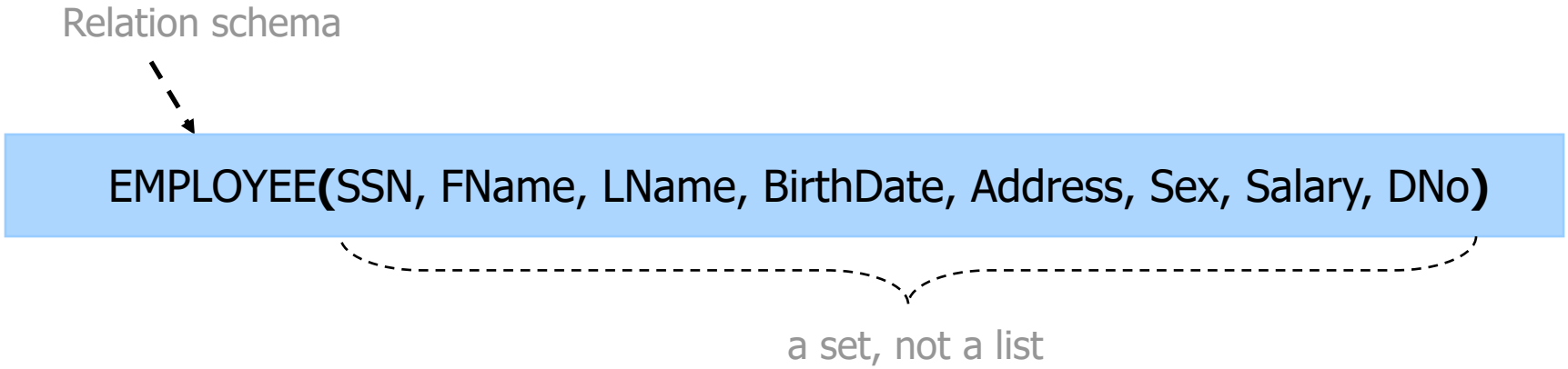


FName	LName	BirthDate	Address	Sex	Salary	DNo
Tung	Nguyen	12/08/1955	638 NVC Q5	Nam	40000	5
Hang	Bui	07/19/1968	332 NTH Q1	Nu	25000	4
Nhu	Le	06/20/1951	291 HVH QPN	Nu	43000	4
Hung	Nguyen	09/15/1962	Ba Ria VT	Nam	38000	5

- All values in a column are of the same data type

Schema

- Schema of a relation
  - Name
  - Set of attributes



Schema

- Database schema
  - A design consist of one or more relational schemas

Database schema

EMPLOYEE(SSN, FName, LName, BirthDate, Address, Sex, Salary, DNo)

DEPARTMENT(DNumber, DName, MgrSSN, MgrStartDate)

DEPT_LOCATION(DNumber, DLocation)

DEPENDENT(SSN, Dependent_Name, Sex, BDate, Relationship)

PROJECT(PName, PNumber, PLocation, DNum)

Tuple

- Row of a relation
  - Except the header row containing the attribute names
- Contains many components
  - One component for each attributes of the relation

<Tung, Nguyen, 12/08/1955, 638 NVC Q5, Nam, 40000, 5>

a component

Domain

- Each attribute of a relation associates with a ***domain***
  - A particular elementary type
  - Domain ~ Data type
- A component of each tuple
  - Is ***atomic:***
    - String, Number, Date/time, …
  - Has a ***value*** that belongs to the domain of the corresponding attribute
- Example
  - FName: string, DOM(FName): the set of strings
  - Salary: integer, DOM(Salary): the set of integers

Characteristics of relation

- The order of tuples in a relation is not important

LName	FName	BirthDate	Address	Sex	Salary	DNo
Nguyen	Tung	12/08/1955	638 NVC Q5	Nam	40000	5
Bui	Hang	07/19/1968	332 NTH Q1	Nu	25000	4
Le	Nhu	06/20/1951	291 HVH QPN	Nu	43000	4
Nguyen	Hung	09/15/1962	null	Nam	38000	5

- The order of values in a tuple is important

<Nguyen, Tung, 12/08/1955, 638 NVC Q5, ***Nam, 40000***, 5>

Differs from

<Nguyen, Tung, 12/08/1955, 638 NVC Q5, ***40000, Nam***, 5>

Characteristics of relation

- Each value of components in a tuple
  - Atomic or
  - NULL

- Relations are sets of tuples, not lists of tuples
  - There are no identical tuples

Relational model notation

- Relation schema
  - Given A_1, A_2, …, A_n are attributes
  - Has domains D_1, D_2, …, D_n respectively
  - Is denoted by $R(A_1:D_1, A_2:D_2, \ldots, A_n:D_n)$

  - Example
    - EMPLOYEE(SSN:DOM(integer), FName:DOM(string), LName:DOM(string), Birthday:DOM(date), Address:DOM(string), Sex:DOM(string), Salary:DOM(integer), DNo:DOM(integer))

- The degree of a relation is the number of attributes of its relation schema
  - EMPLOYEE is a relation schema of degree 8

Relational model notation

- Relation instances
  - A relation instance r of relation schema $R(A_1, A_2, \ldots, A_n)$, denoted by r(R), is a set of tuples $r = \{t_1, t_2, \ldots, t_k\}$
  - Where each t_i is an ordered list of n values $t_i = <v_1, v_2, \ldots, v_n>$
    - Each v_j is a member of $DOM(A_j)$ or NULL value

	FName	LName	BirthDate	Address	Sex	Salary	DNo
t_1	Tung	Nguyen	12/08/1955	638 NVC Q5	Nam	40000	5
t_2	Hang	Bui	07/19/1968	332 NTH Q1	Nu	25000	4
t_3	Nhu	Le	06/20/1951	291 HVH QPN	Nu	43000	4
t_4	Hung	Nguyen	09/15/1962	null	Nam	38000	5

v_i

Summary of denotations

- The relation schema R of the degree n
  - $R(A_1, A_2, \ldots, A_n)$
- The attribute set of R
  - R^+
- Relations
  - R, S, P, Q
- Tuples
  - t, u, v
- The domain of the attribute A
  - DOM(A)
- The value at the attribute A of the t^{th} tuple
  - t.A or t[A]

Content

- Introduction
- Concepts
- **Constraints**
  - Superkey
  - Key
  - Primary key
  - Reference
  - Foreign key
- From E/R diagram to relational design

Constraint

- Integrity constraint
  - Rules, conditions need to satisfy for all of instances of relational database

- Constraints
  - Defined when the relation schema is modeled
  - Checked when the data in relations are modified

Superkey

- Definition
  - Assume SK is a subset of attributes of R, SK ≠ ∅
  - SK is the super key if

$$\forall r, \forall t1,t2 \in r,\ t1 \neq t2 \Rightarrow t1[SK] \neq t2[SK]$$

Any two distinct tuples have the different values at the superkey

- Remark
  - No two tuples in any state r of R can have the same value for superkey
  - Every relation has at least one default superkey

Example

- Find all superkeys of R

R

A	B	C	D
x	1	10	a
x	2	20	a
y	1	40	b
y	1	40	c
z	1	50	d

Key

- Definition
  - Assume K is a subset of attributes of R, $K \neq \emptyset$
  - K is a key if
    - **K is a superkey of R and**
    - **$\forall K' \subset K$, $K' \neq K$, K' is not the superkey of R**

A key is the minimal superkey

- Remark
  - The value of a key identifies uniquely each tuple in the relation
  - A key is a *property* of the relation schema
    - Time-invariant: a constraint should hold on every valid state
  - A key is determined from the meaning of attributes
  - A relation has more than one key

Primary key

- Designate one of the key as the primary key (PK)
  - The value for PK is constrained to be not null
  - Underline the attributes of PK when displaying its relation schema

- The choice of PK
  - Influence some implementation issues
  - Usually with a single attribute or a small number of attributes

Reference

- R refers to S when
  - An attribute A of a tuple in relation R receives a value from an attribute B of relation S
    - Must refer to an existing tuple

S

DName	DNumber
Nghien cuu	**5**
Dieu hanh	4
Quan ly	1

R

FName	LName	BirthDate	Address	Sex	Salary	DNo
Tung	Nguyen	12/08/1955	638 NVC Q5	Nam	40000	**5**
Hang	Bui	07/19/1968	332 NTH Q1	Nu	25000	4
Nhu	Le	06/20/1951	291 HVH QPN	Nu	43000	4
Hung	Nguyen	09/15/1962	Ba Ria VT	Nam	38000	5

Foreign key

- Examine two relation schemas R and S
  - Assume FK is a set of attributes of R, FK ≠ ∅
  - FK is a foreign key of R if
    - Attributes in FK have the same domains as the primary key attributes PK of S
    - A value of FK in a tuple $t_1 \in R$
      - Either is a value of PK for some tuple $t_2 \in S$
      - Or is null

- Example

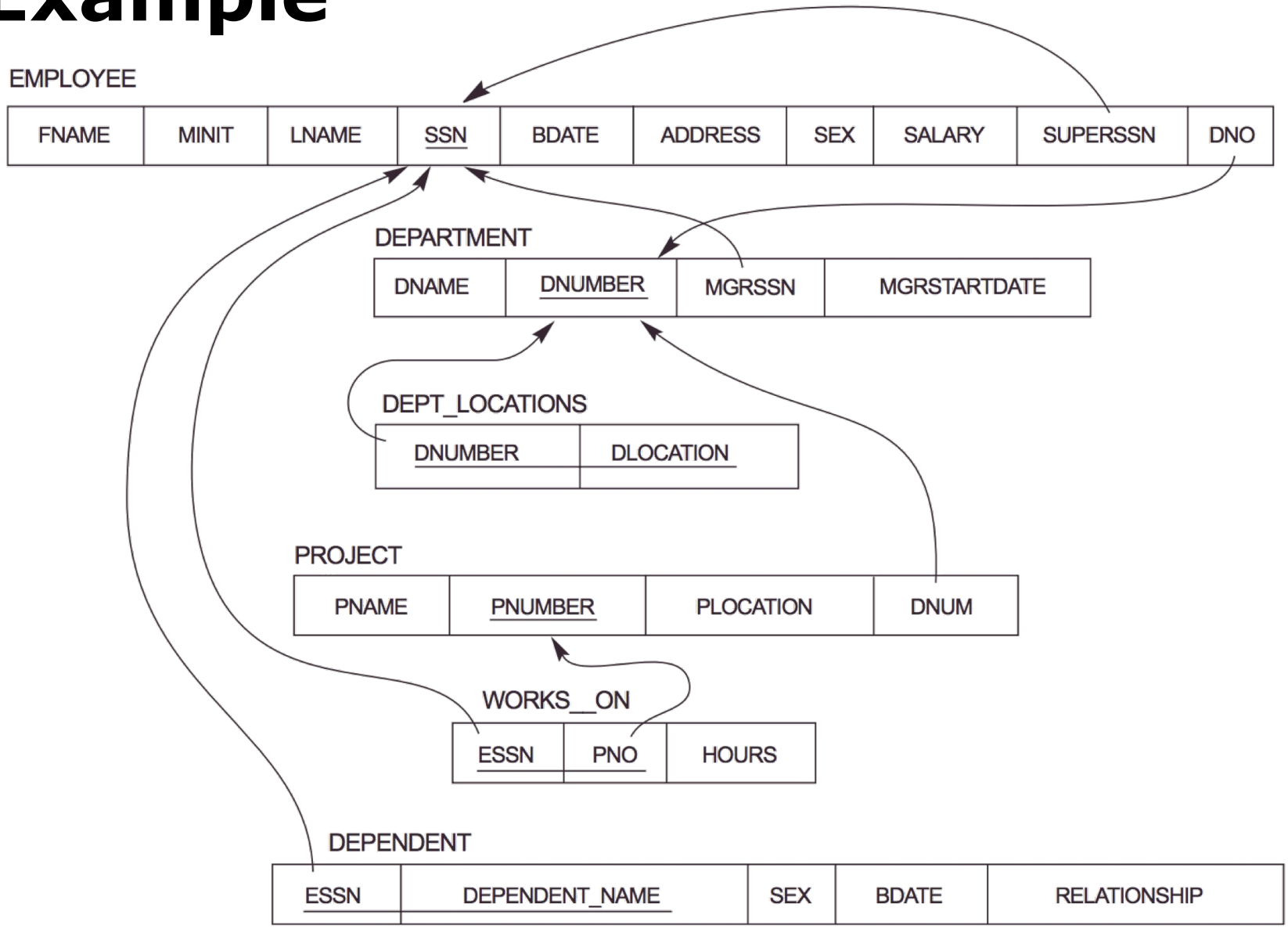
EMPLOYEE(SSN, FName, LName, BirthDate, Address, Sex, Salary, DNo)

DEPARTMENT(DName, DNumber)

Foreign key

- Remark
  - An attribute can both participate in PK and participate in FK
  - A FK can refer to its own relation
  - Many FKs might refer to the same primary key
  - Referential constraint = Foreign key constraint

Example

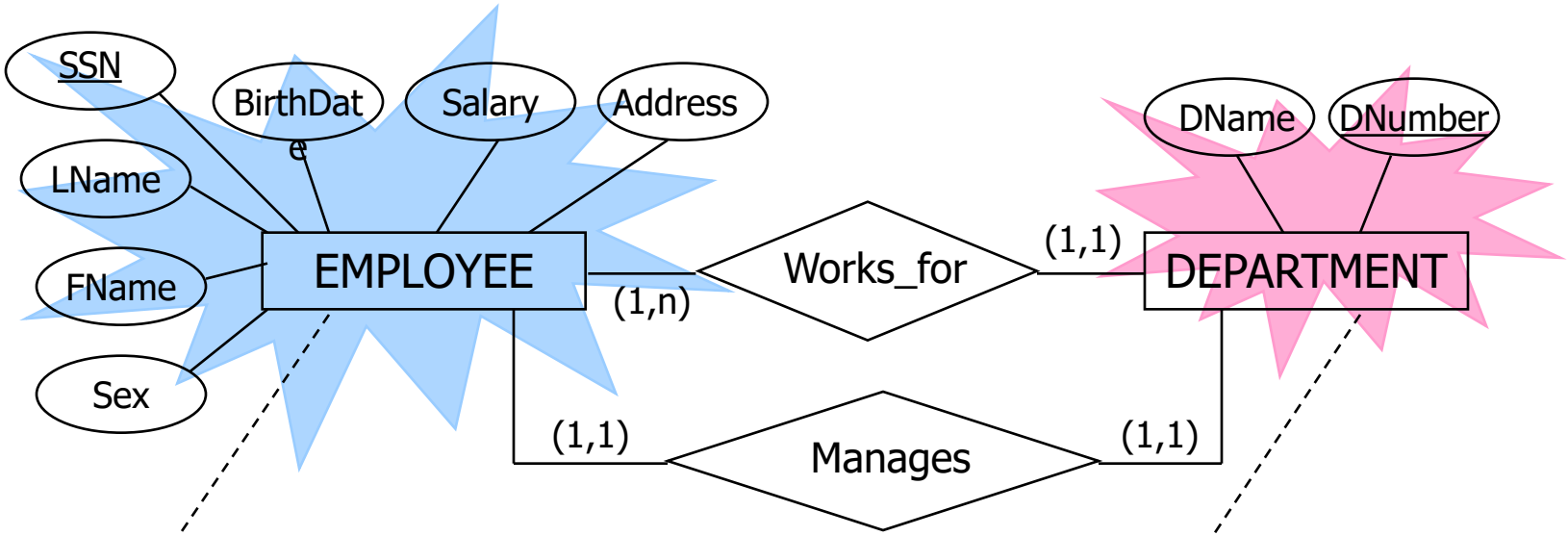

EMPLOYEE
FNAME
MINIT
LNAME
SSN
BDATE
ADDRESS
SEX
SALARY
SUPERSSN
DNO
DEPARTMENT
DNAME
DNUMBER
MGRSSN
MGRSTARTDATE
DEPT_LOCATIONS
DNUMBER
DLOCATION
PROJECT
PNAME
PNUMBER
PLOCATION
DNUM
WORKS_ON
ESSN
PNO
HOURS
DEPENDENT
ESSN
DEPENDENT_NAME
SEX
BDATE
RELATIONSHIP

Content

- Introduction
- Concepts
- Constraints
- **From E/R diagrams to relational design**
  - Rules

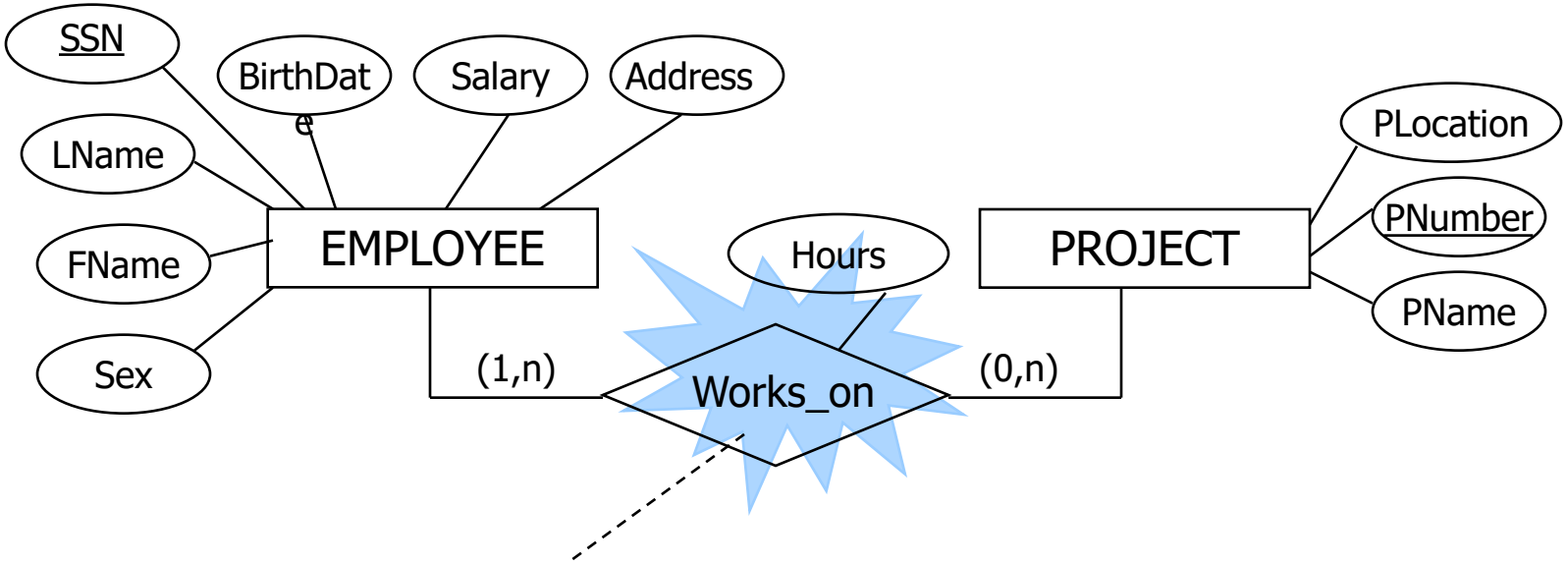
Rules

- (1) Entity set
  - Turn each entity set (except weak entity set) into a relation with the same set of attributes



Rules

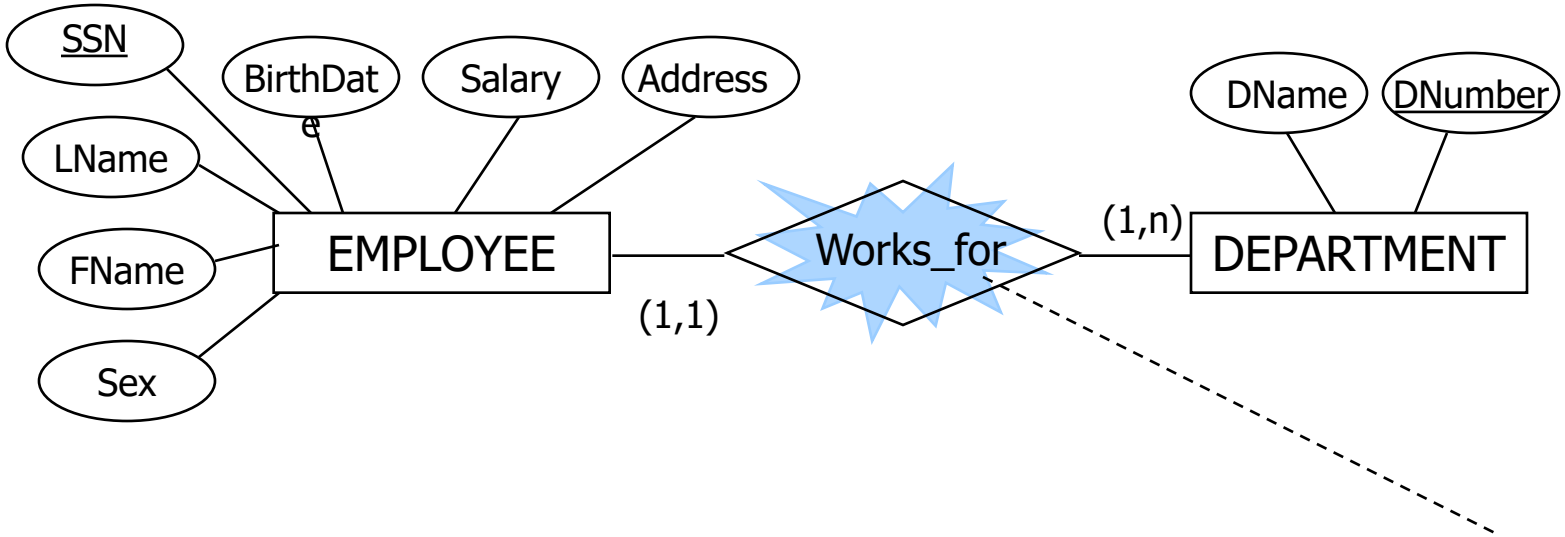
- (2) Relationship
  - (2a) Many-Many
    - Create a new relation
      - Relation name is the name of the relationship
      - Attributes are the key attributes of connected entity sets



WORKS_ON(**SSN, PNumber,** Hours)

Rules

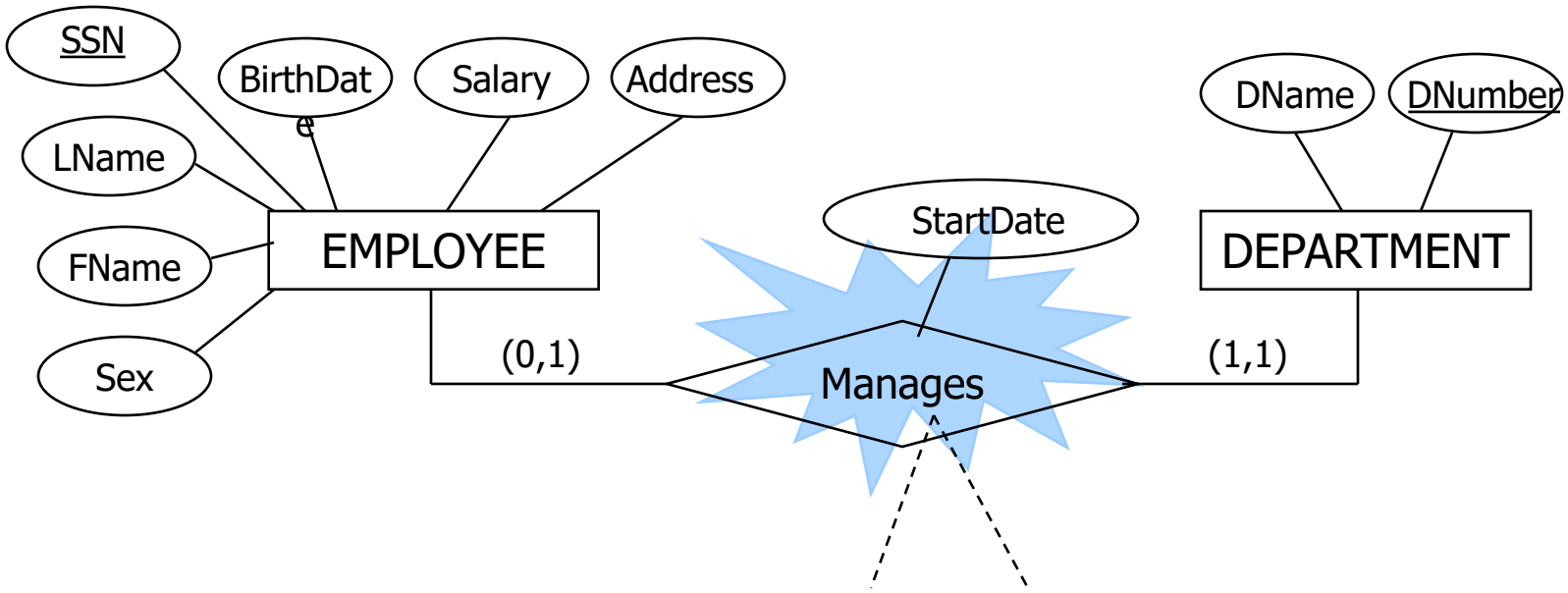
- (2) Relation
  - (2b) One-Many
    - Adding the key of the many-relation to the one-relation



EMPLOYEE(SSN, FName, LName, BirthDate, Address, Sex, Salary, **DNumber**)

Rules

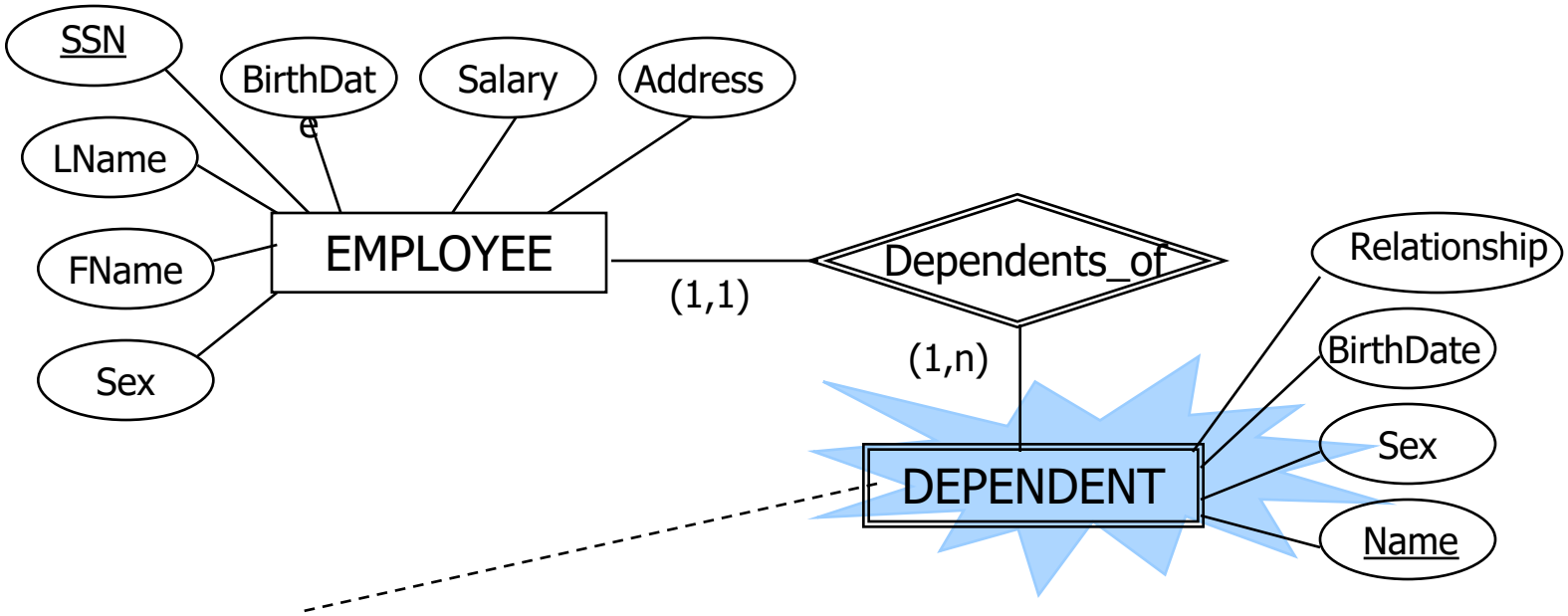
- (2) Relationship
  - (2c) One-One
    - Either adding the key of a relation to another relation
    - Or adding the key to both relations



DEPARTMENT(<u>DNumber</u>, DName, **SSN, StartDate**)

Rules

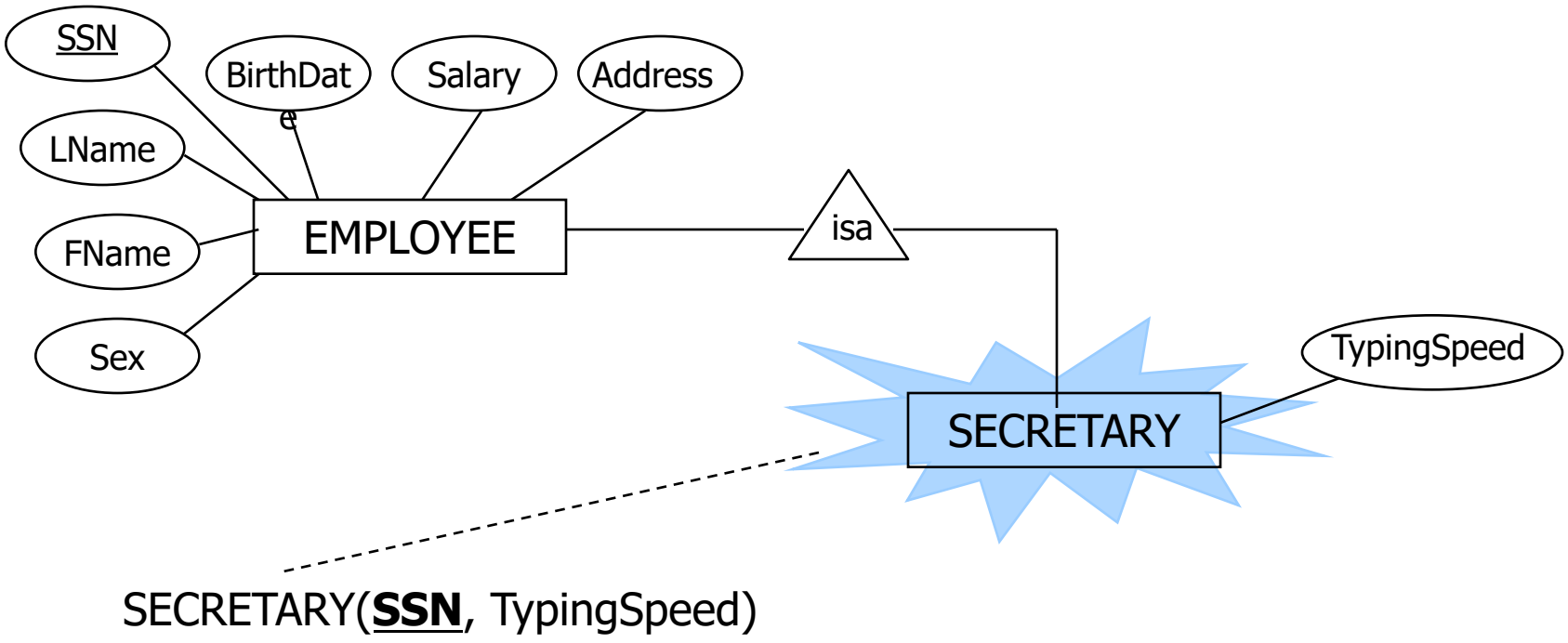
- (3) Weak entity set
  - Turn into a relation
    - Has the same name
    - Add the key of related entity sets



DEPENDENT(**SSN, Name**, Sex, BirthDate, Relationship)

Rules

- (4) Subclass
  - Turn into a relation
    - Has the same name
    - Add the key of the superclass



SECRETARY(**SSN**, TypingSpeed)