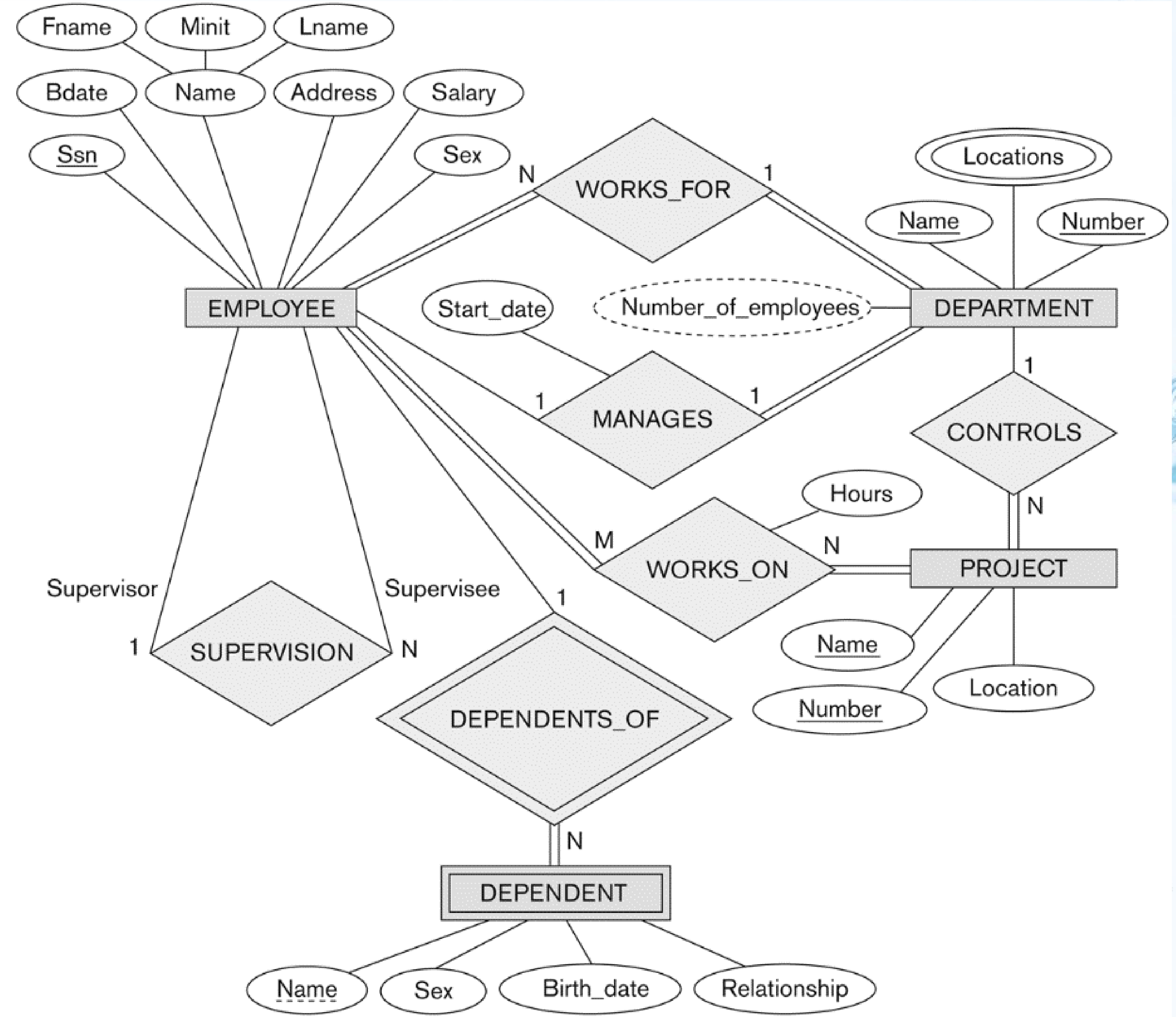
COSC 3380
Design of Database Systems

Data Modeling with Entity-Relationship (ER) Model

February 14, 2024

ER Diagram – Relationship Types



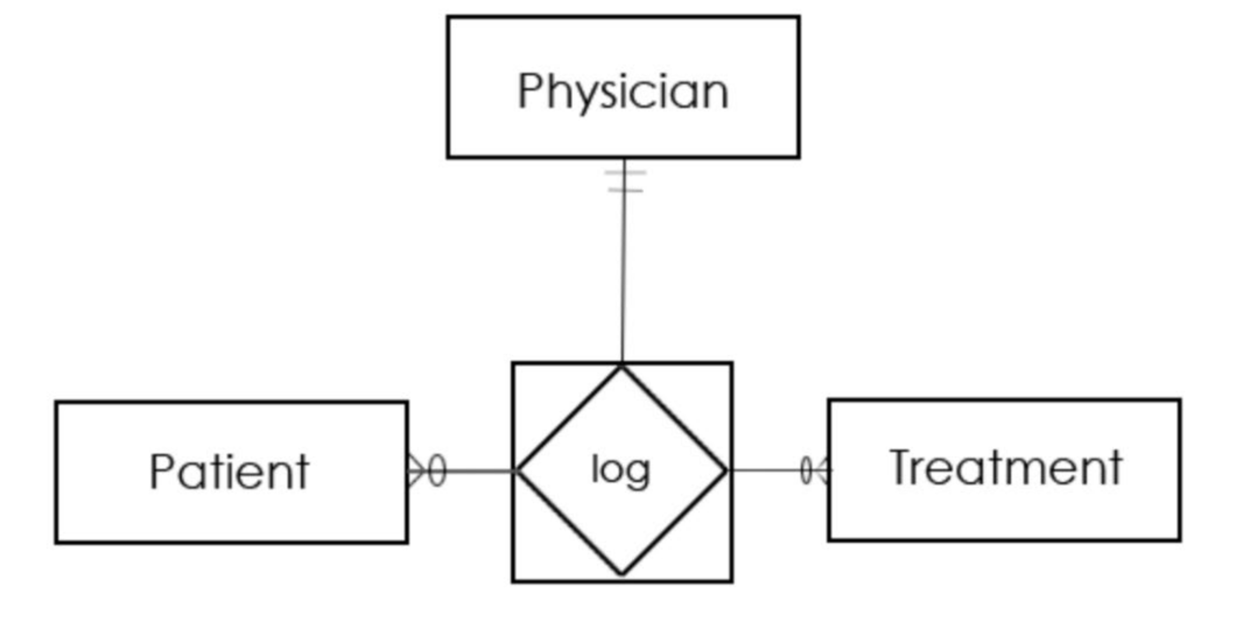
Relationship Types

- Some attributes from the initial entity types are refined into relationships:
  - Manager of DEPARTMENT -> MANAGES
  - Works_on of EMPLOYEE -> WORKS_ON
  - Department of EMPLOYEE -> WORKS_FOR
- More than one relationship type can exist between the same participating entity types
  - MANAGES and WORKS_FOR are distinct relationship types between EMPLOYEE and DEPARTMENT
  - Different meanings and different relationship instances

Constraints on Relationships

- Constraints on Relationship Types
  - Also known as ratio constraints
  - Cardinality Ratio (specifies *maximum* participation)
    - One-to-one (1:1)
    - One-to-many (1:N) or Many-to-one (N:1)
    - Many-to-many (M:N)
  - Existence Dependency Constraint (specifies *minimum* participation) - also called participation constraint
    - zero (optional participation, not existence-dependent)
    - one or more (mandatory participation, existence-dependent)

Example of a ternary relationship



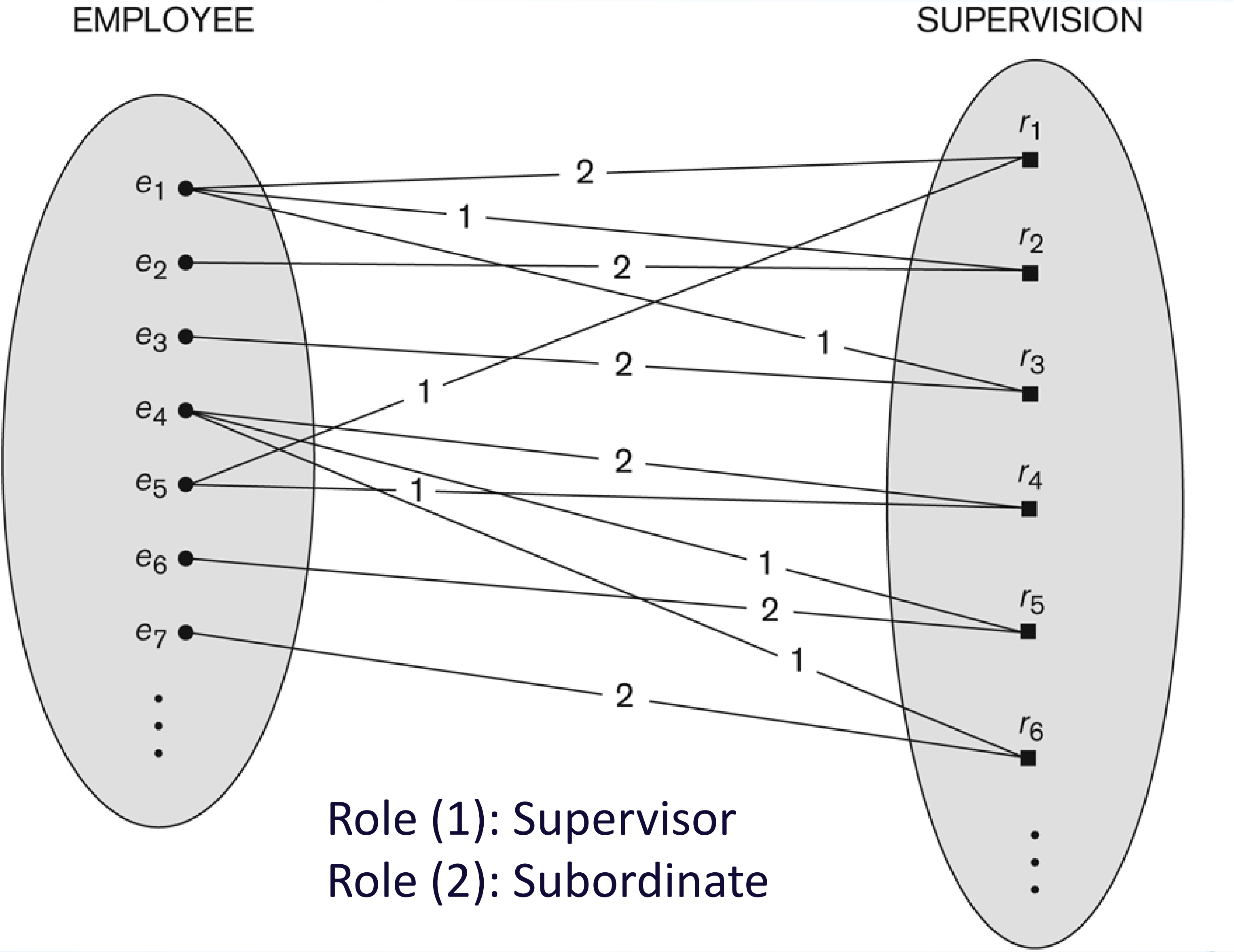
- 1 Physician with 1 Patient can log **M** Treatments
- 1 Physician logs 1 Treatment for **N** Patients
- 1 Patient logged 1 Treatment by **1** Physician

‘log’ is a M:N:1 relationship between participating entities, Treatment – Patient – Physician

Recursive Relationship Type

- A relationship type between the same participating entity type in **distinct roles**
- A **self-referencing** relationship type
- Example: the SUPERVISION relationship
- EMPLOYEE participates twice in two distinct roles:
  - supervisor (or boss) role
  - supervisee (or subordinate) role
- Each relationship instance relates two distinct EMPLOYEE entities:
  - One employee in *supervisor* role
  - One employee in *supervisee* role

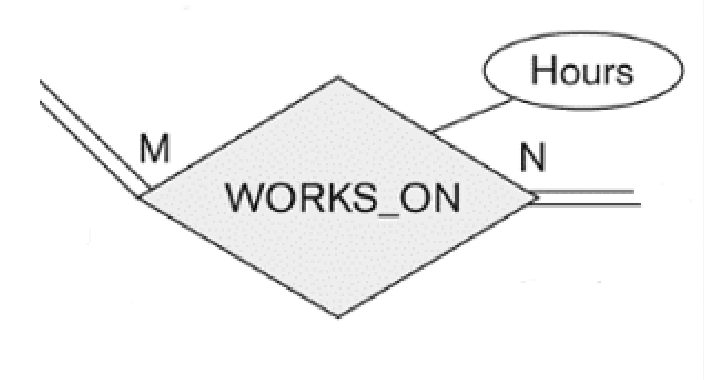
Recursive Relationship - Supervision



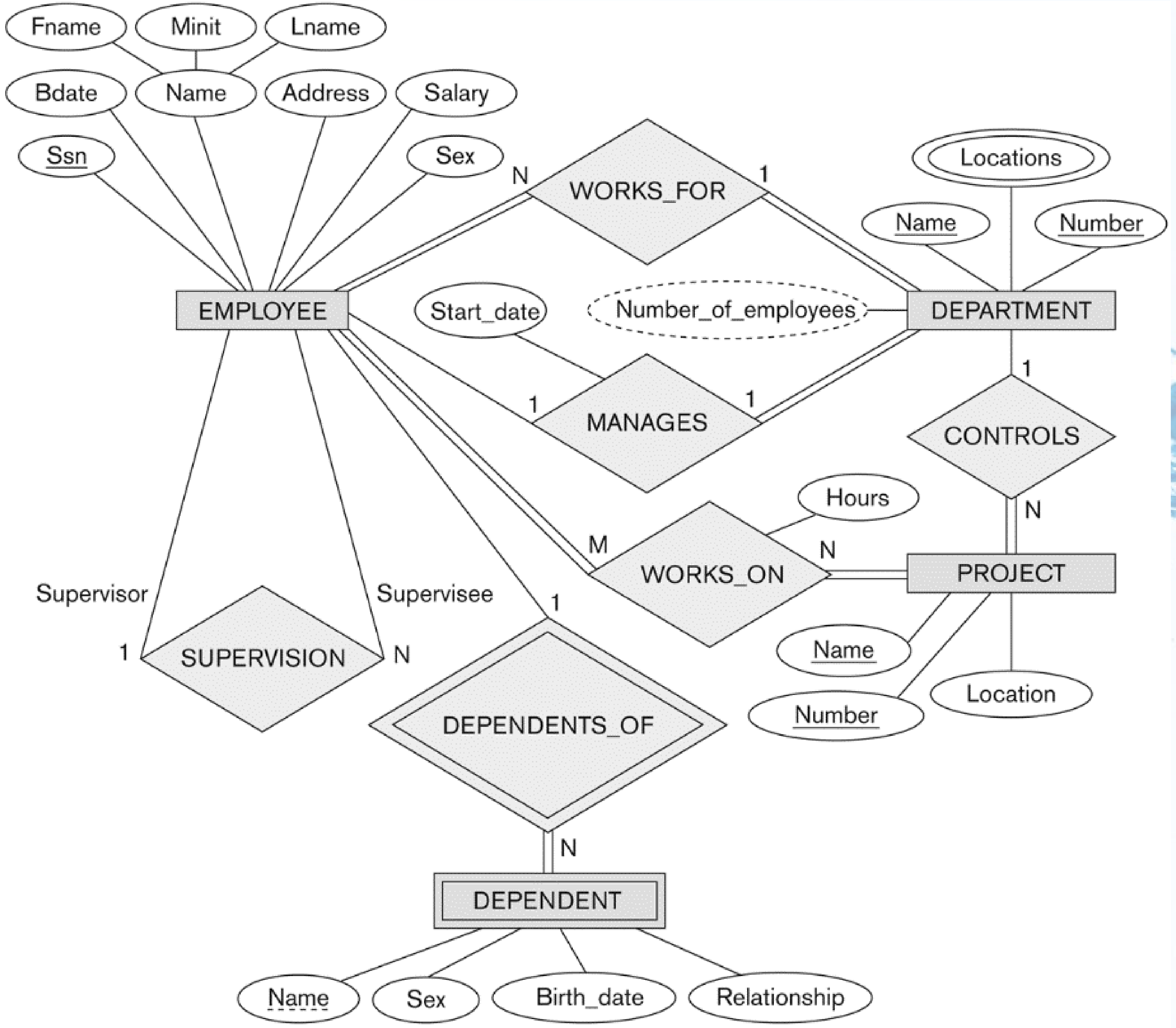
Role (1): Supervisor

Role (2): Subordinate

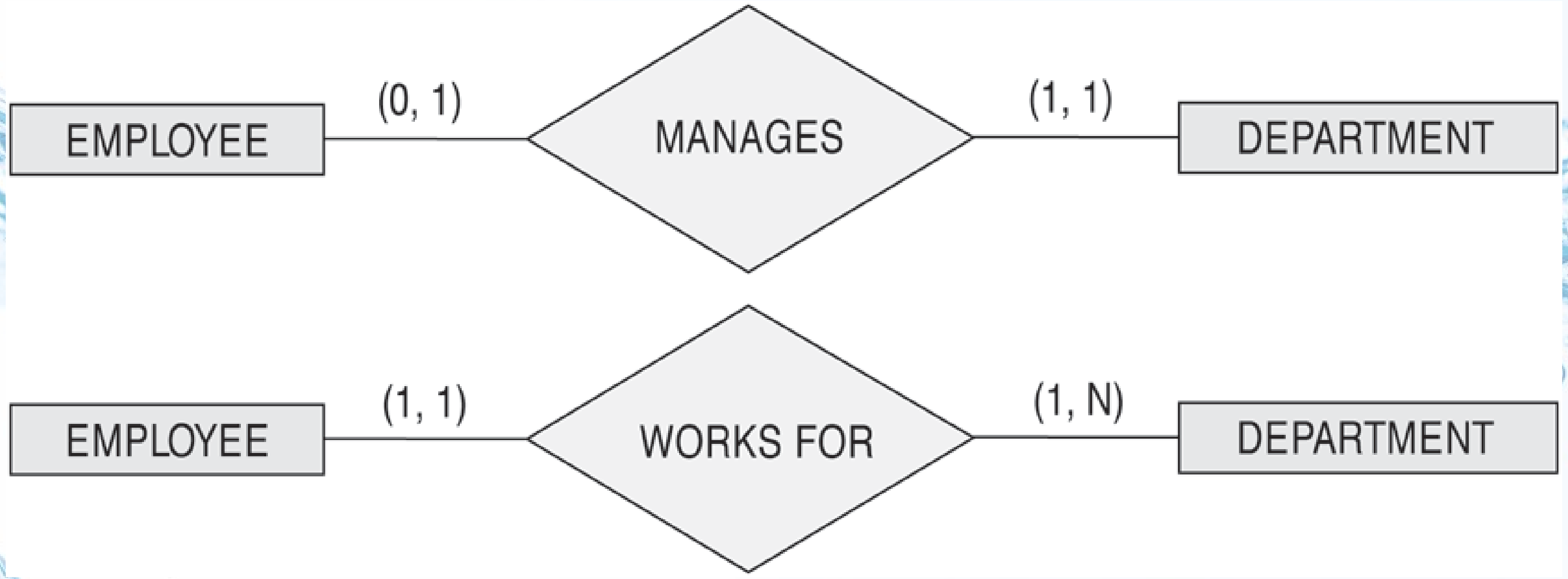
Attributes of Relationship types



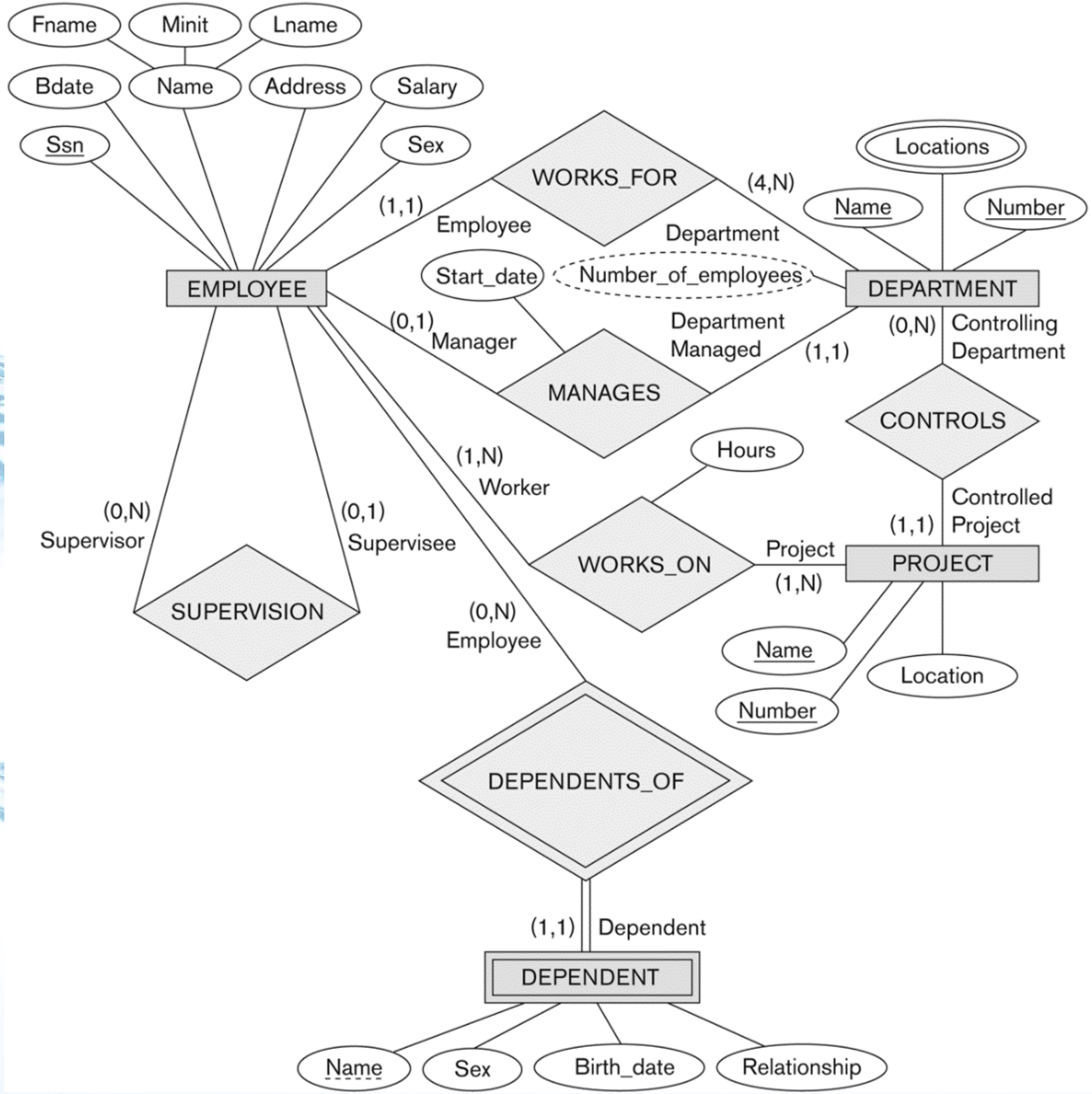
- A relationship type can have attributes
  - For example, HoursPerWeek of WORKS_ON
  - Value for each relationship instance describes the number of hours per week that an EMPLOYEE works on a PROJECT.
    - A value of HoursPerWeek depends on a particular (employee, project) combination
  - Most relationship attributes are used with M:N relationships
    - In 1:N relationships, they can be transferred to the entity type on the N-side of the relationship


Fname
Minit
Lname
Bdate
Name
Address
Salary
Ssn
Sex
N
WORKS_FOR
1
Locations
Name
Number
EMPLOYEE
Start_date
Number_of_employees
DEPARTMENT
1
1
MANAGES
1
CONTROLS
N
Hours
M
WORKS_ON
N
PROJECT
Supervisor
Supervisee
1
1
SUPERVISION
N
DEPENDENTS_OF
Name
Number
Location
N
DEPENDENT
Name
Sex
Birth_date
Relationship

(min,max) notation for relationship constraints

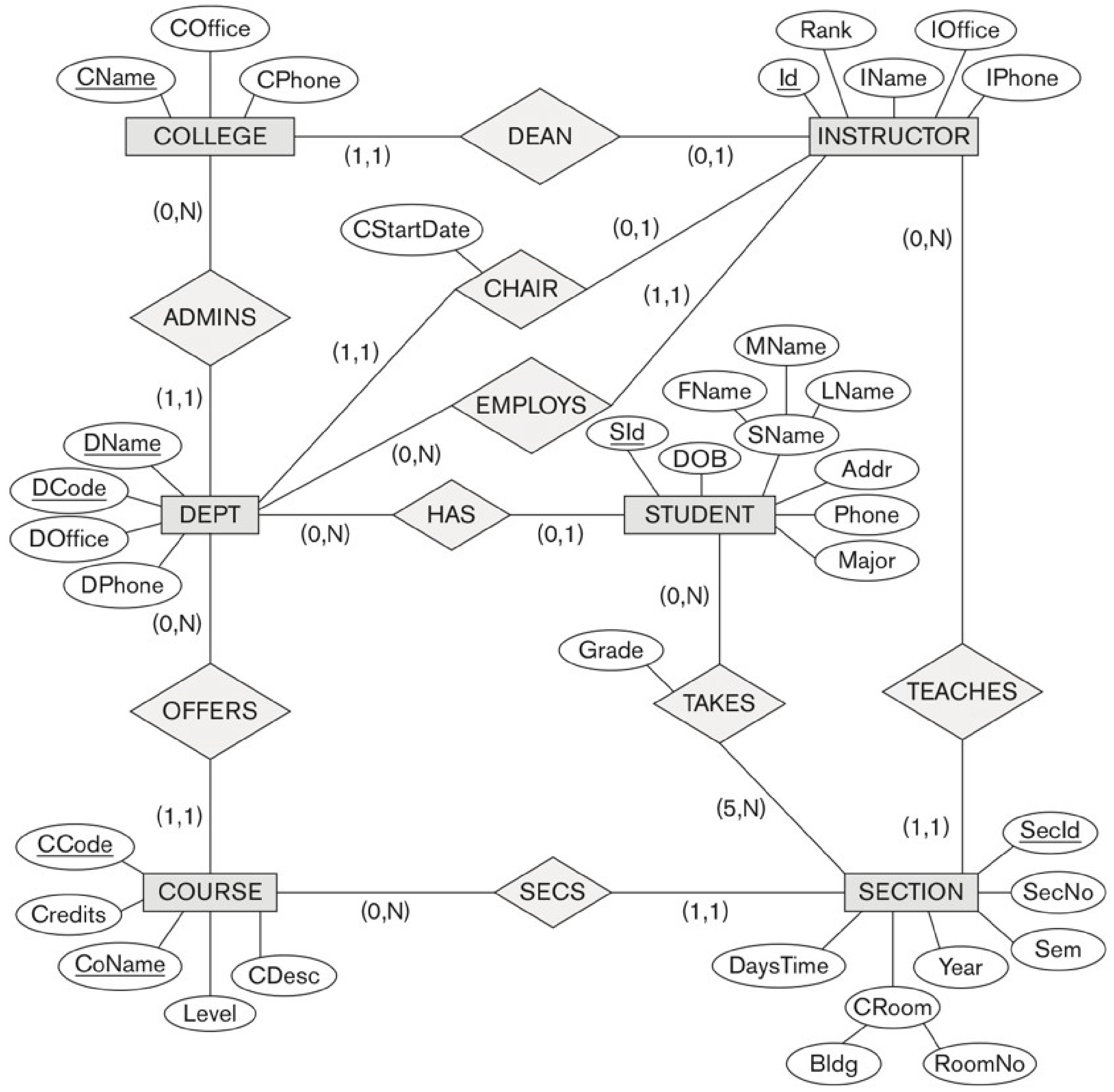


Read the (min,max) numbers next to the entity type and looking **away from** the entity type



Fname
Minit
Lname
Bdate
Name
Address
Salary
Ssn
Sex
EMPLOYEE
WORKS_FOR
(1,1)
Employee
(4,N)
Department
Locations
Name
Number
DEPARTMENT
Start_date
Number_of_employees
(0,1)
Manager
MANAGES
Department Managed
(1,1)
(0,N)
Controlling Department
CONTROLS
Hours
(1,N)
Worker
WORKS_ON
Project
(1,N)
Controlled Project
(1,1)
PROJECT
Name
Number
Location
(0,N)
Supervisor
(0,1)
Supervisee
SUPERVISION
(0,N)
Employee
DEPENDENTS_OF
(1,1)
Dependent
DEPENDENT
Name
Sex
Birth_date
Relationship

UNIVERSITY database conceptual schema



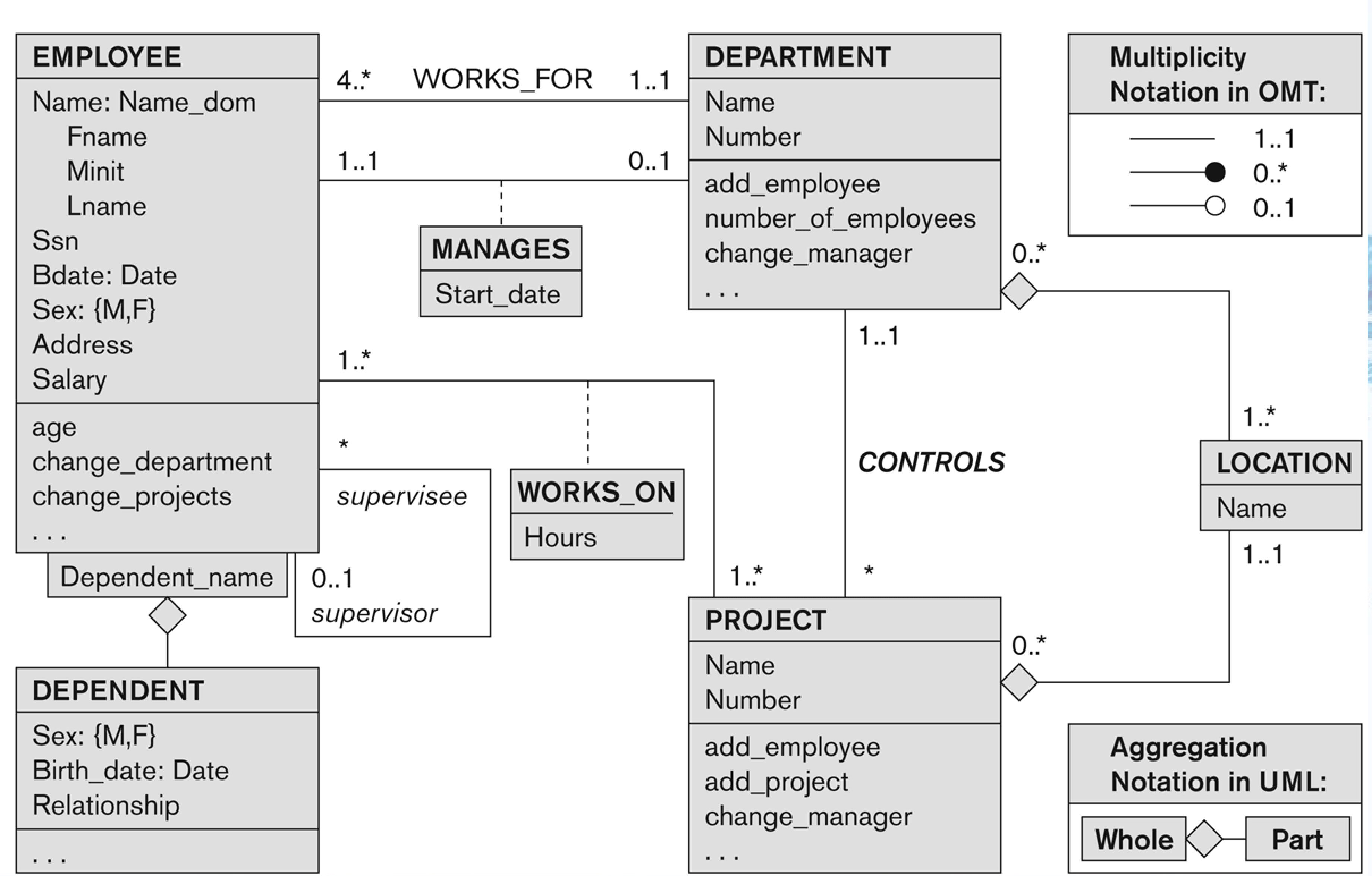
Alternative diagrammatic notation

- ER diagrams is one popular example for displaying database schemas
- Many other notations exist
- UML class diagrams are used in commercial/software design tools

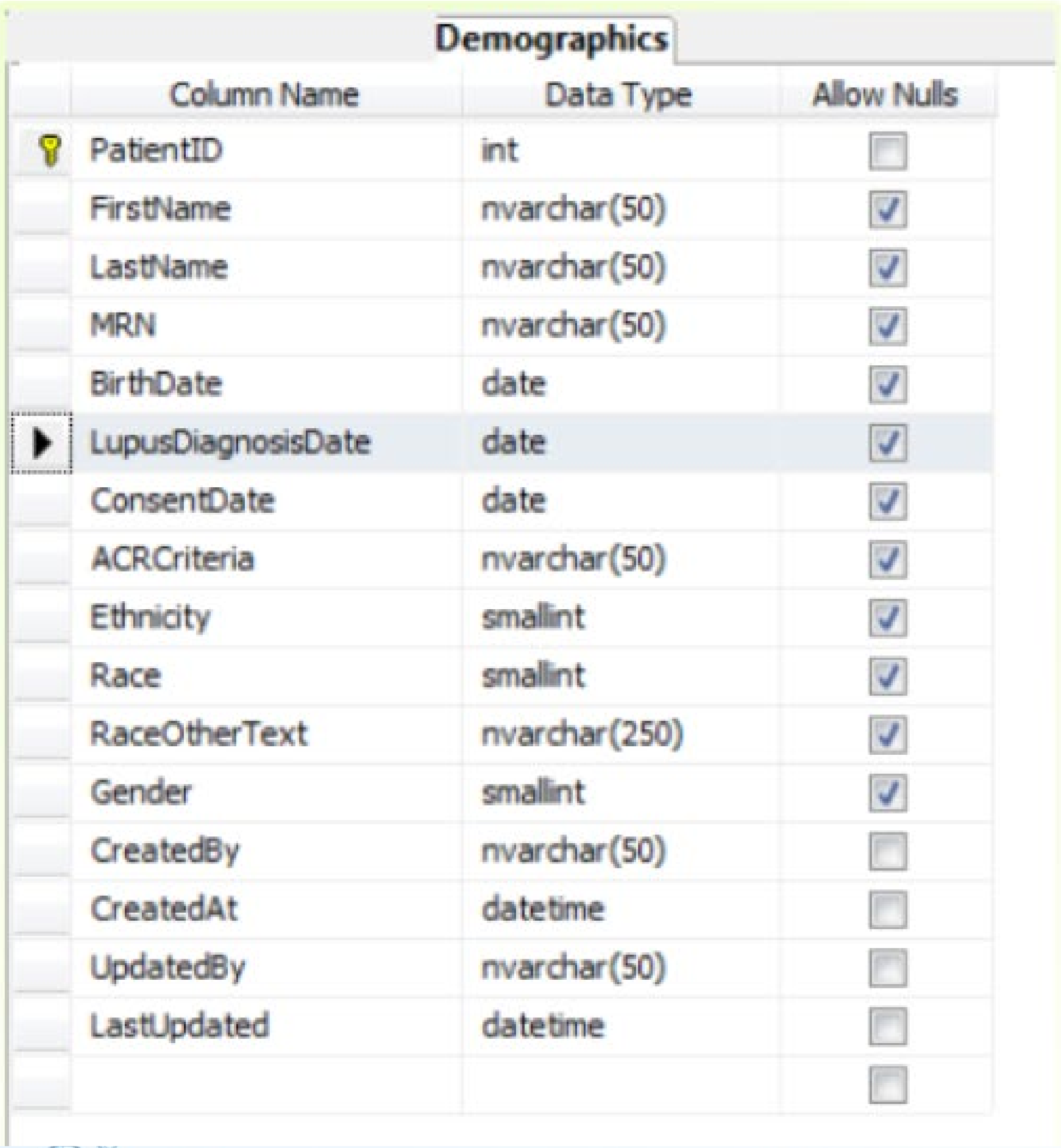
UML class diagrams

- Represent classes (similar to entity types) as large rounded boxes with three sections:
  - Top section includes entity type (class) name
  - Second section includes attributes
  - Third section includes class operations (operations are not in basic ER model)
- Relationships (called associations) represented as lines connecting the classes
  - Other UML terminology also differs from ER terminology
- Used in database design and object-oriented software design
- UML has many other types of diagrams for software design

UML class diagram for COMPANY database

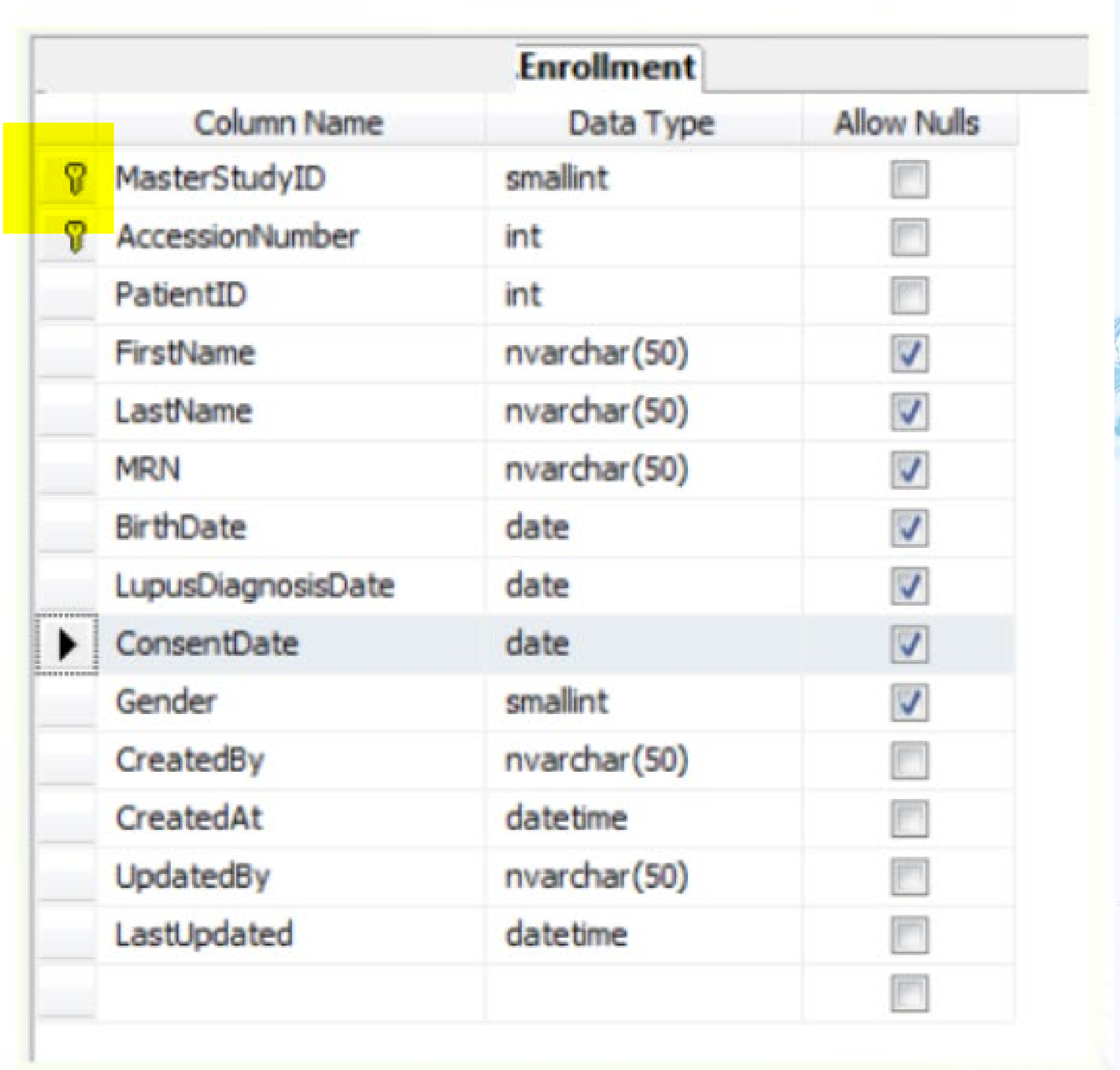


Database Catalog



Demographics

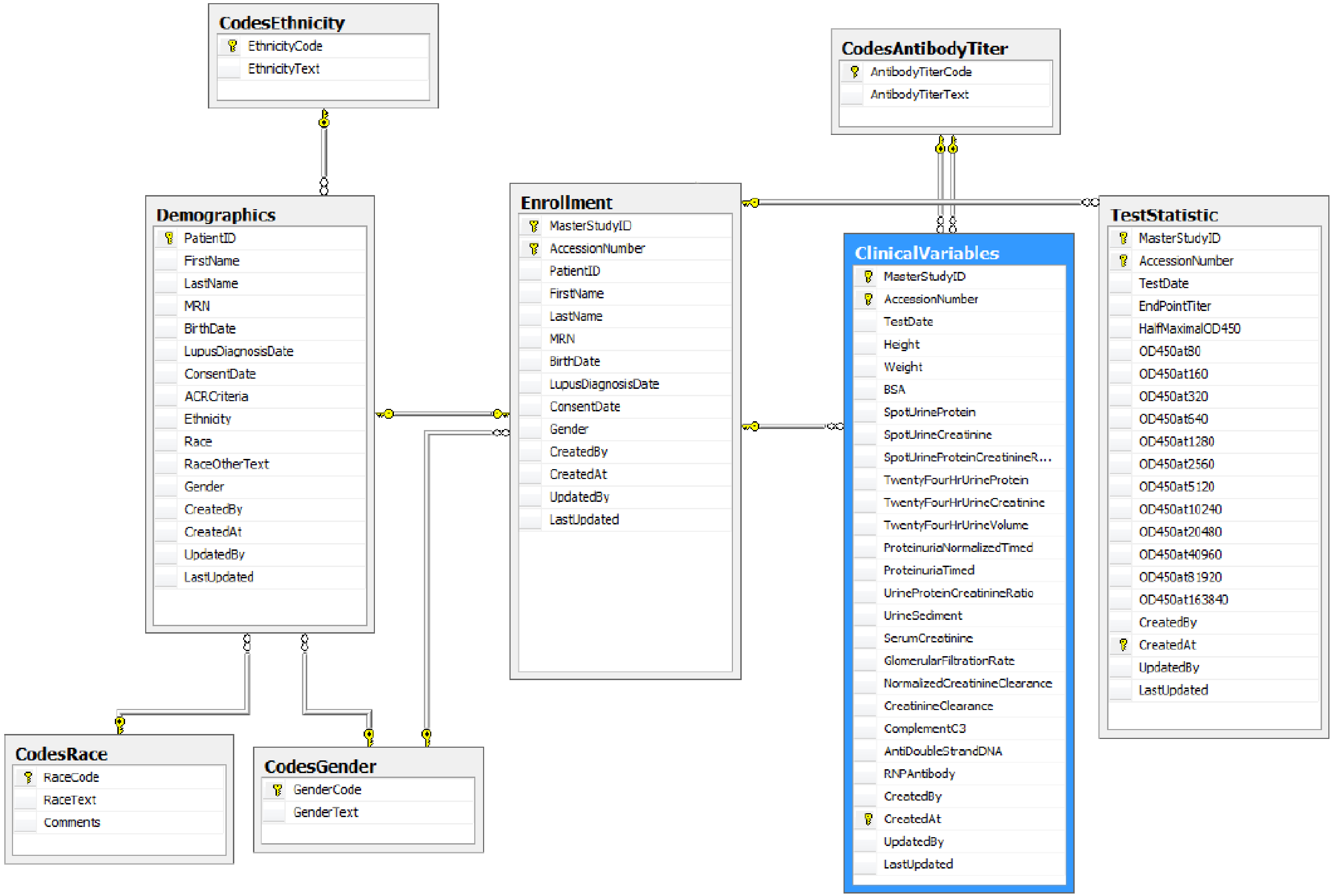
	Column Name	Data Type	Allow Nulls
🔑	PatientID	int	☐
	FirstName	nvarchar(50)	☑
	LastName	nvarchar(50)	☑
	MRN	nvarchar(50)	☑
	BirthDate	date	☑
▶	LupusDiagnosisDate	date	☑
	ConsentDate	date	☑
	ACRCriteria	nvarchar(50)	☑
	Ethnicity	smallint	☑
	Race	smallint	☑
	RaceOtherText	nvarchar(250)	☑
	Gender	smallint	☑
	CreatedBy	nvarchar(50)	☐
	CreatedAt	datetime	☐
	UpdatedBy	nvarchar(50)	☐
	LastUpdated	datetime	☐
			☐



Enrollment

	Column Name	Data Type	Allow Nulls
🔑	MasterStudyID	smallint	☐
🔑	AccessionNumber	int	☐
	PatientID	int	☐
	FirstName	nvarchar(50)	☑
	LastName	nvarchar(50)	☑
	MRN	nvarchar(50)	☑
	BirthDate	date	☑
	LupusDiagnosisDate	date	☑
▶	ConsentDate	date	☑
	Gender	smallint	☑
	CreatedBy	nvarchar(50)	☐
	CreatedAt	datetime	☐
	UpdatedBy	nvarchar(50)	☐
	LastUpdated	datetime	☐
			☐

Database Diagram



CodesEthnicity
- 🔑 EthnicityCode
- EthnicityText

CodesAntibodyTiter
- 🔑 AntibodyTiterCode
- AntibodyTiterText

Demographics
- 🔑 PatientID
- FirstName
- LastName
- MRN
- BirthDate
- LupusDiagnosisDate
- ConsentDate
- ACRCriteria
- Ethnicity
- Race
- RaceOtherText
- Gender
- CreatedBy
- CreatedAt
- UpdatedBy
- LastUpdated

Enrollment
- 🔑 MasterStudyID
- 🔑 AccessionNumber
- PatientID
- FirstName
- LastName
- MRN
- BirthDate
- LupusDiagnosisDate
- ConsentDate
- Gender
- CreatedBy
- CreatedAt
- UpdatedBy
- LastUpdated

ClinicalVariables
- 🔑 MasterStudyID
- 🔑 AccessionNumber
- TestDate
- Height
- Weight
- BSA
- SpotUrineProtein
- SpotUrineCreatinine
- SpotUrineProteinCreatinineR...
- TwentyFourHrUrineProtein
- TwentyFourHrUrineCreatinine
- TwentyFourHrUrineVolume
- ProteinuriaNormalizedTimed
- ProteinuriaTimed
- UrineProteinCreatinineRatio
- UrineSediment
- SerumCreatinine
- GlomerularFiltrationRate
- NormalizedCreatinineClearance
- CreatinineClearance
- ComplementC3
- AntiDoubleStrandDNA
- RNPAntibody
- CreatedBy
- 🔑 CreatedAt
- UpdatedBy
- LastUpdated

TestStatistic
- 🔑 MasterStudyID
- 🔑 AccessionNumber
- TestDate
- EndPointTiter
- HalfMaximalOD450
- OD450at80
- OD450at160
- OD450at320
- OD450at640
- OD450at1280
- OD450at2560
- OD450at5120
- OD450at10240
- OD450at20480
- OD450at40960
- OD450at81920
- OD450at163840
- CreatedBy
- 🔑 CreatedAt
- UpdatedBy
- LastUpdated

CodesRace
- 🔑 RaceCode
- RaceText
- Comments

CodesGender
- 🔑 GenderCode
- GenderText

Relational Data Model
and
Relational Database Constraints

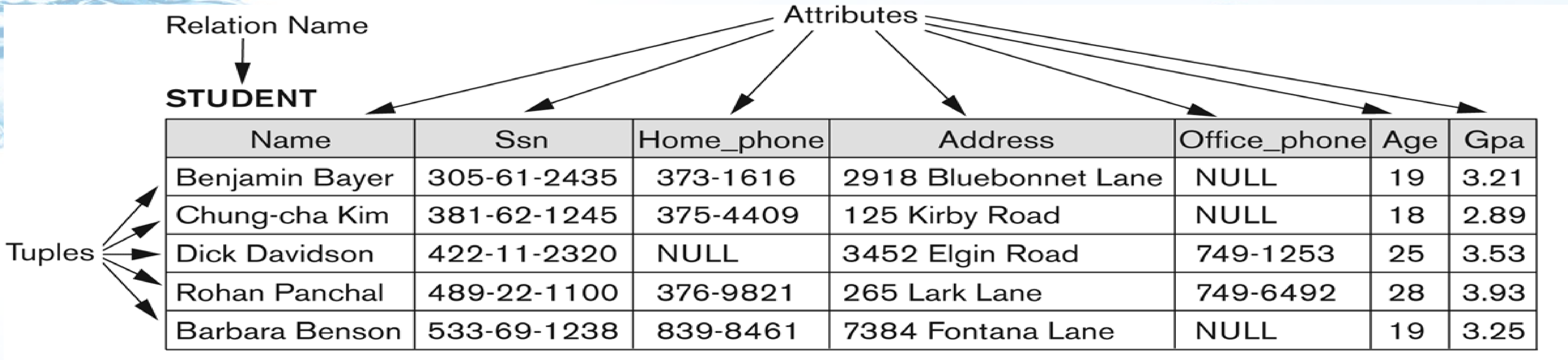
Relational Model Concepts

- The relational Model of Data is based on the concept of a ***Relation***
  - The strength of the relational approach to data management comes from the formal foundation provided by the theory of relations
- A Relation is a mathematical concept based on the ideas of sets
- The model was first proposed by Dr. E.F. Codd of IBM Research in an ACM paper:
  - *"A Relational Model for Large Shared Data Banks," Communications of the ACM, June 1970*

Informal Definitions

- A **relation** looks like a **table** of values.
- A relation typically contains a **set of rows**.
- The data elements in each **row** represent certain facts that correspond to a real-world **entity** or **relationship**
  - In the formal model, rows are called **tuples**
- Each **column** has a column header that gives an indication of the meaning of the data items in that column
  - In the formal model, the column header is called an **attribute name** (or just **attribute**)

Example of a Relation



Relation Name

Attributes

Tuples

STUDENT

Name	Ssn	Home_phone	Address	Office_phone	Age	Gpa
Benjamin Bayer	305-61-2435	373-1616	2918 Bluebonnet Lane	NULL	19	3.21
Chung-cha Kim	381-62-1245	375-4409	125 Kirby Road	NULL	18	2.89
Dick Davidson	422-11-2320	NULL	3452 Elgin Road	749-1253	25	3.53
Rohan Panchal	489-22-1100	376-9821	265 Lark Lane	749-6492	28	3.93
Barbara Benson	533-69-1238	839-8461	7384 Fontana Lane	NULL	19	3.25

Informal Definitions

- ***Key*** of a relation:
  - Each row has a value of a data item (or set of items) that uniquely identifies that row in the table
  - In the STUDENT table, SSN is the key
  - Sometimes row-ids or sequential/random numbers are assigned as keys to identify the rows in a table
    - Called ***artificial key*** or ***surrogate key***
    - ***Why?***

Formal Definitions - Schema

- The **Schema** (or description) of a Relation:
  - Denoted by R(A_1, A_2, ...A_n)
  - R is the **name** of the relation
  - The **attributes** of the relation are A_1, A_2, ..., A_n
- **Degree** (or **arity**) of a relation
  - Number of attributes *n* of its relation schema
- CUSTOMER (Cust-id, Cust-name, Address, Phone#)
  - CUSTOMER is the relation name (Degree of 4)
  - Defined over the four attributes: Cust-id, Cust-name, Address, Phone#
- Each attribute has a **domain** or a set of valid values.
  - For example, the domain of Cust-id is 6 digit numbers.

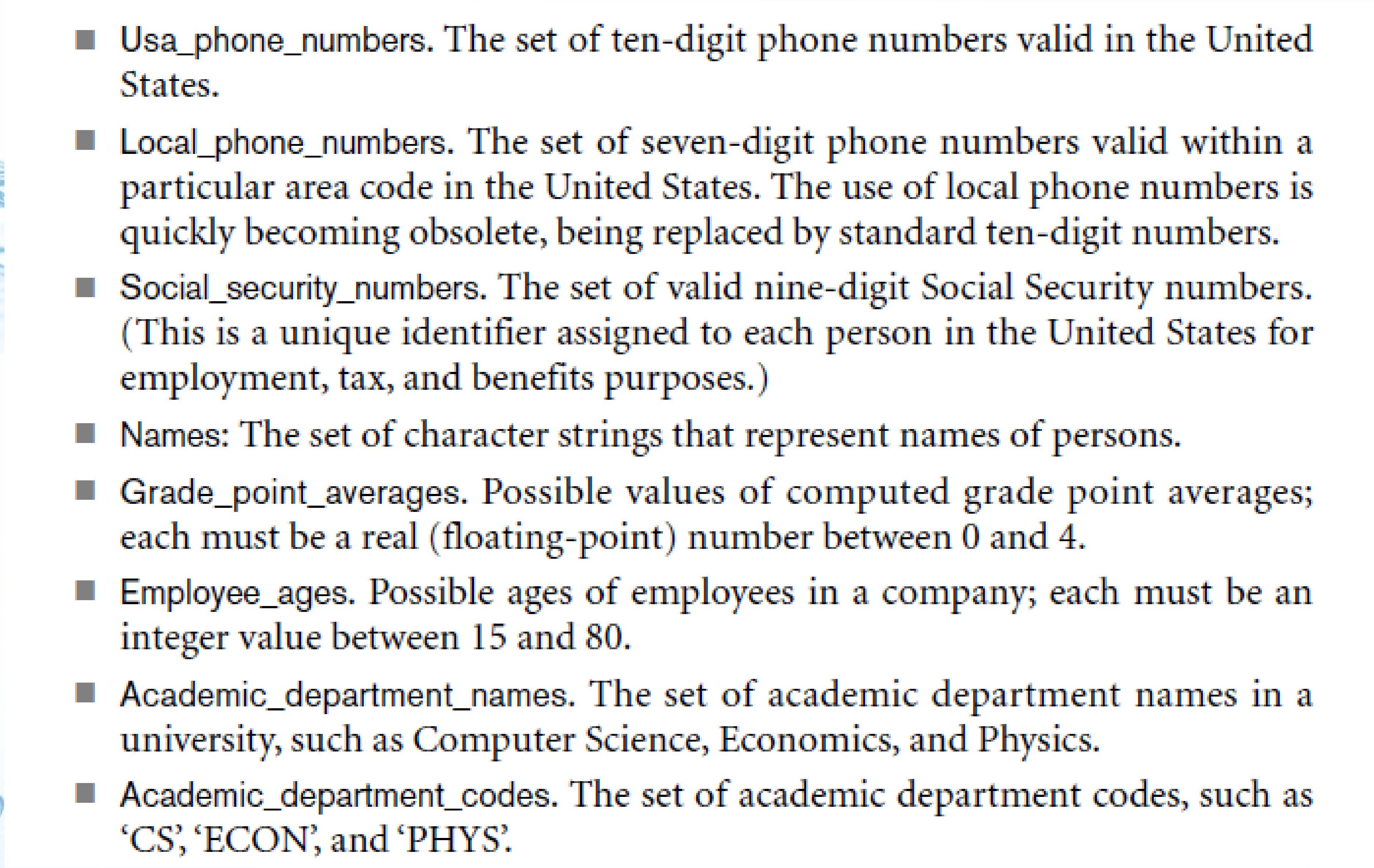
Attributes and Relations Examples

- A Relation of degree seven for university students
  - STUDENT (Name, SSN, Address, Home_phone, Office_Phone, Age, GPA)
- Using the data type of each attribute
  - STUDENT (Name: string, SSN:string, Address: string, Home_phone: string, Office_Phone: string, Age: integer, GPA: real)

Domains

- **Domain** D
  - Set of atomic values
- **Atomic**
  - Each value indivisible
- Specifying a domain
  - **Data type** specified for each domain

Domain Examples



- Usa_phone_numbers. The set of ten-digit phone numbers valid in the United States.
- Local_phone_numbers. The set of seven-digit phone numbers valid within a particular area code in the United States. The use of local phone numbers is quickly becoming obsolete, being replaced by standard ten-digit numbers.
- Social_security_numbers. The set of valid nine-digit Social Security numbers. (This is a unique identifier assigned to each person in the United States for employment, tax, and benefits purposes.)
- Names: The set of character strings that represent names of persons.
- Grade_point_averages. Possible values of computed grade point averages; each must be a real (floating-point) number between 0 and 4.
- Employee_ages. Possible ages of employees in a company; each must be an integer value between 15 and 80.
- Academic_department_names. The set of academic department names in a university, such as Computer Science, Economics, and Physics.
- Academic_department_codes. The set of academic department codes, such as 'CS', 'ECON', and 'PHYS'.