Data Modeling Using the Entity Relationship (ER) Model

CH.3

Data Models

- A collection of tools for describing
  - data
  - data relationships
  - data semantics دلالات البيانات
  - data constraintsقيود البيانات
- Entity-Relationship model - Ex student and teacher
- Relational model R Database

Relational database consists of a collection of tables , each of which is assigned a unique name .

- Other models:
  - Older models: network model and hierarchical mode النموذج الشبكي و الهرمي

نموذج البيانات هو عبارة عن تمثيل بسيط لوصف تراكيب البيانات المعقدة في واقع الحياة العملية على شكل رسومي دون النظر الى مكان وكيفية تخزين أو الوصول الى هذه البيانات.

- يستخدم هذا النموذج كوسيلة اتصال ما بين المصمم من جهة وبين المبرمجين والمستخدمين من جهة اخرى.
- حتى لو كان لدينا العديد من المبرمجين المحترفين فلا نستطيع الحصول على نظام جيد دون ان يكون هذا النظام قد صمم بشكل صحيح.

ER Model Concepts

- Entities and attributes
- **Entities** :A “thing” in the real world with an independent existence
- are specific person , place , event ,objects in the user environment about which the user needs to keep data .

For example: the Employee , office , Task , Conference .

→ An object with physical existence (ex: house , person) or with conceptual existence(ex :course , job)

- **Attributes** : are properties used to describe an entity.
- For example : an EMPLOYEE entity may have the attributes Name ,SSN ,Address ,Sex ,Birthdate.

ال Attributes هي الخصائص التي يتم تخزينها عن كل Entity

Composite Attributes

- ★ Can be divided into further parts.
- ★ Ex: Name → First Name, Middle Name, Last Name

Simple Attributes

- ★ Cannot be divided further.
- ★ Ex: Weight → cannot be further divided.

Single-Valued Attributes

- ★ Have a single value for a particular entity.
- ★ Ex: Age → single-valued attribute of a person.

Multivalued Attributes

- ★ Can have set of values for a particular entity.
- ★ Ex: College degree, languages known → multivalued attributes of a person.


Derived Attributes
★ Can be derived from other attributes.
★ Ex: Age → can be derived from date of birth.
Stored Attributes
★ From which the value of other attributes are derived.
★ Ex: BirthDate of a person

ER Model Concepts

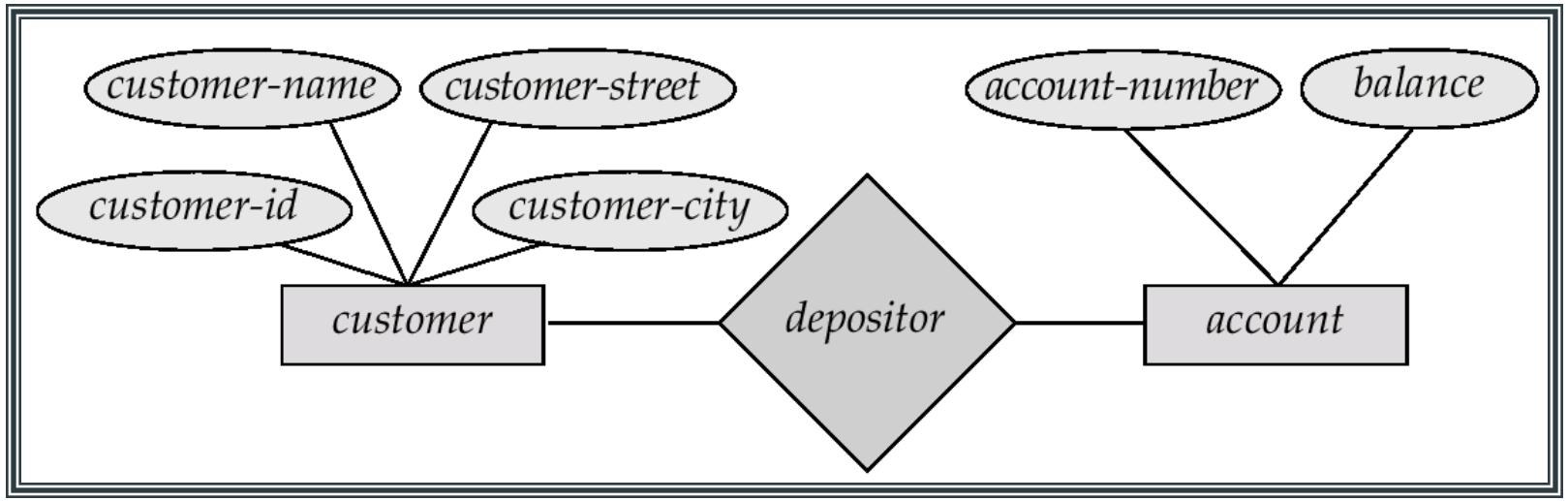
- Relationships :

A relationship relates two or more distinct entities with a specific meaning.

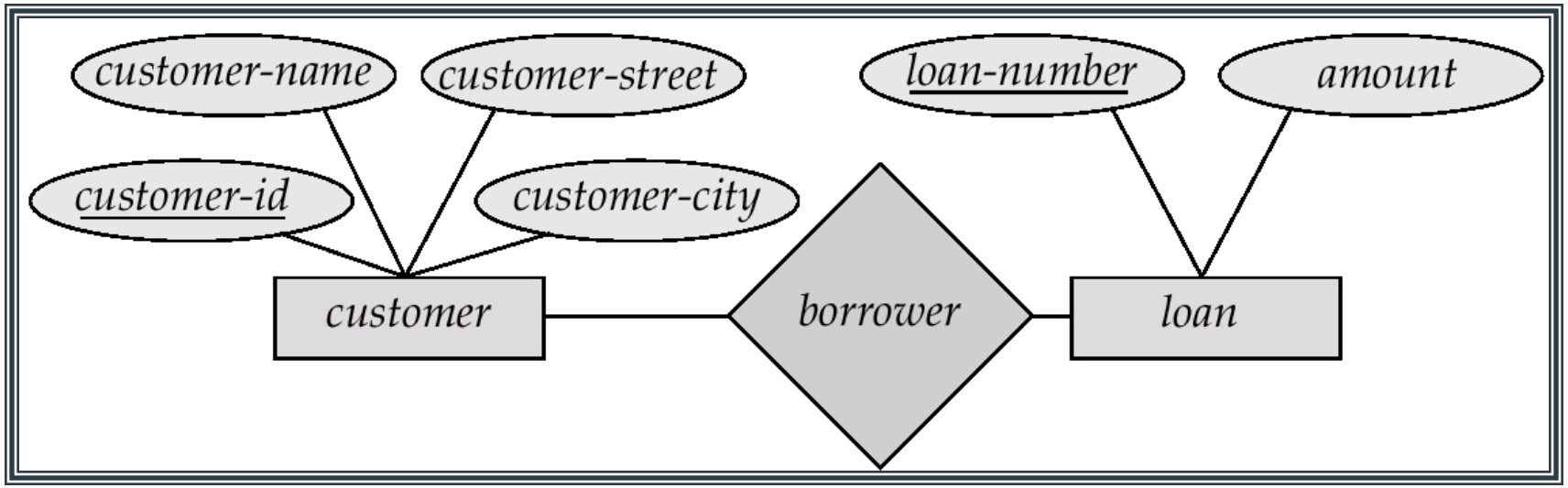
- For Example , Employee Ali works on the Project , or
- EMPLOYEE Hasan manages the Research DEPARTMENT .

Entity-Relationship Model

Example of schema in the entity-relationship model , BINARY relationship

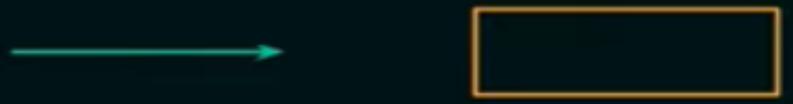


E-R Diagrams



- **Rectangles** represent entity sets.
- **Diamonds** represent relationship sets.
- **Lines** link attributes to entity sets and entity sets to relationship sets.
- **Ellipses** represent attributes
  - **Double ellipses** represent multivalued attributes.
  - **Dashed ellipses** denote derived attributes.
- **Underline** indicates primary key attributes (will study later)

★ Entity



★ Weak Entity

Entity type that do not have key attributes of there own , relating to another entity called identifying or the owner entity type

مثل كيان أبناء الموظفين

★ Attribute

★ Key Attribute

★ Multivalued Attribute

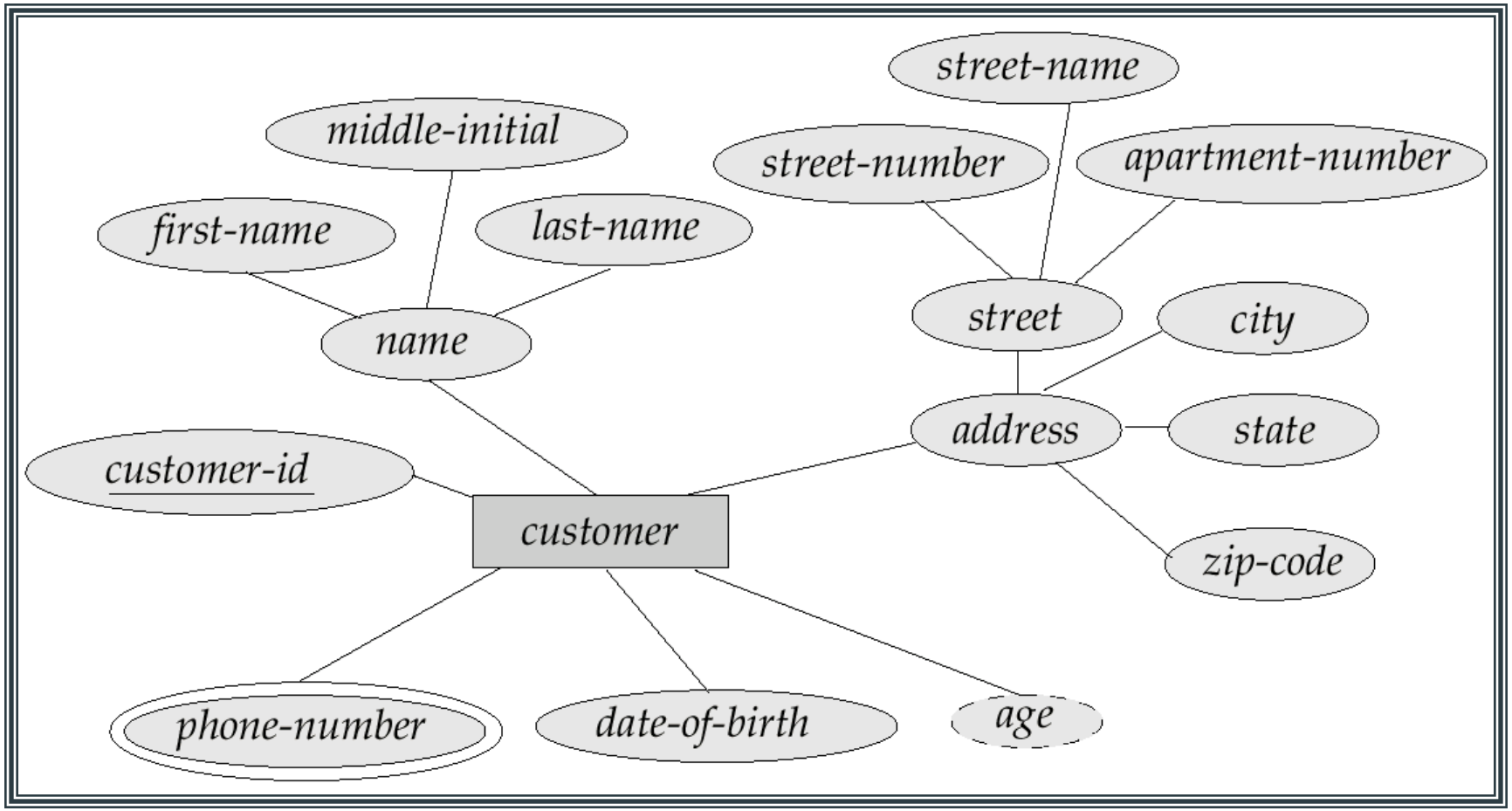
★ Composite Attribute

★ Derived Attribute

★ Identifying Relationship

a relation between a weak entity type and strong entity type use douple diamond , but if the relation between 2 strong entity types uses a single diamond

E-R Diagram With Composite, Multivalued, and Derived Attributes



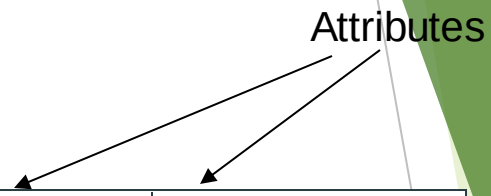
Entity Relationship Model (Cont.)

- E-R model of real world
  - Entities (objects)
    - E.g. customers, accounts, bank branch
  - Relationships between entities
    - E.g. Account A-101 is held by customer Johnson
    - Relationship set *depositor* associates customers with accounts

    العلاقة تربط المودع بالحساب
- Widely used for database design
  - Database design in E-R model usually converted to design in the relational model (coming up next) which is used for storage and processing

Relational Model

- Example of data in the relational model



Customer-id	*customer-name*	*customer-street*	*customer-city*	*account-number*
192-83-7465	Johnson	Alma	Palo Alto	A-101
019-28-3746	Smith	North	Rye	A-215
192-83-7465	Johnson	Alma	Palo Alto	A-201
321-12-3123	Jones	Main	Harrison	A-217
019-28-3746	Smith	North	Rye	A-201

A Sample Relational Database

customer-id	*customer-name*	*customer-street*	*customer-city*
192-83-7465	Johnson	12 Alma St.	Palo Alto
019-28-3746	Smith	4 North St.	Rye
677-89-9011	Hayes	3 Main St.	Harrison
182-73-6091	Turner	123 Putnam Ave.	Stamford
321-12-3123	Jones	100 Main St.	Harrison
336-66-9999	Lindsay	175 Park Ave.	Pittsfield
019-28-3746	Smith	72 North St.	Rye

(a) The *customer* table

account-number	*balance*
A-101	500
A-215	700
A-102	400
A-305	350
A-201	900
A-217	750
A-222	700

(b) The *account* table

customer-id	*account-number*
192-83-7465	A-101
192-83-7465	A-201
019-28-3746	A-215
677-89-9011	A-102
182-73-6091	A-305
321-12-3123	A-217
336-66-9999	A-222
019-28-3746	A-201

(c) The *depositor* table

Data Definition Language (DDL)

- Specification notation for defining the database schema تعريف قاعدة البيانات
  - E.g.

```
create table account (
        account-number    char(10),
        balance           integer)
```

- DDL compiler generates a set of tables stored in a *data dictionary*
- Data dictionary contains metadata (i.e., data about data)
  - database schema
  - Data *storage and definition* language
    - language in which the storage structure and access methods used by the database system are specified
    - Usually an extension of the data definition language

Data Manipulation Language (DML)

- Language for accessing and manipulating the data organized by the appropriate data model
  - DML also known as query language
- Two classes of languages
  - Procedural – user specifies what data is required and how to get those data
  - Nonprocedural – user specifies what data is required without specifying how to get those data
- SQL is the most widely used query language

SQL

- SQL: widely used non-procedural language
  - E.g. find the name of the customer with customer-id 192-83-7465

    select *customer.customer-name*
    from *customer*
    where *customer.customer-id* = '192-83-7465'

  - E.g. find the balances of all accounts held by the customer with customer-id 192-83-7465

    select *account.balance*
    from *depositor, account*
    where *depositor.customer-id* = '192-83-7465' **and**
    depositor.account-number = *account.account-number*

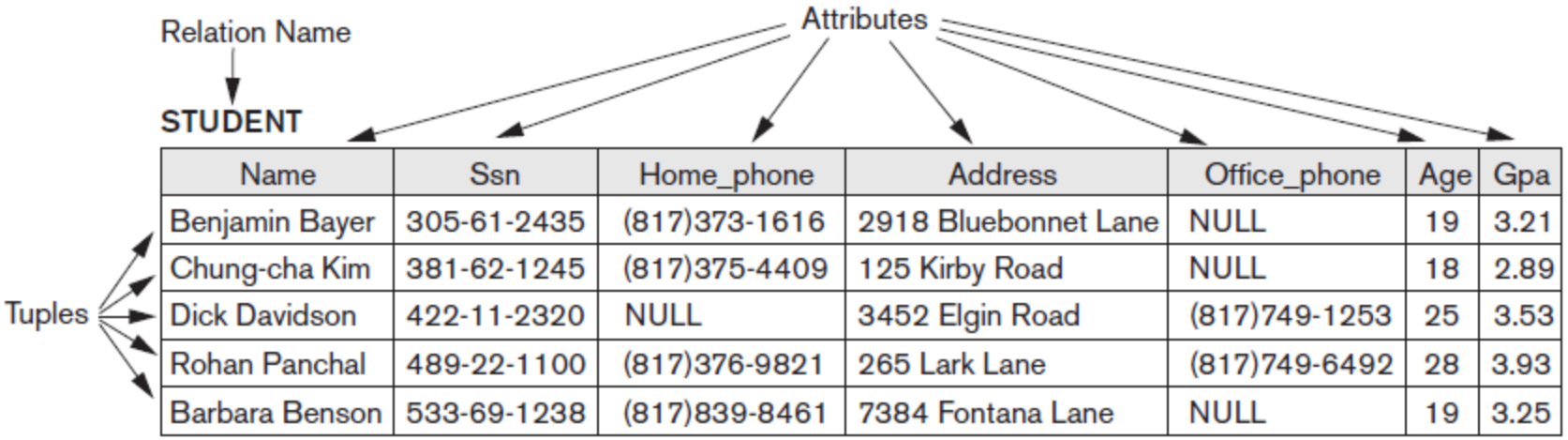
- Application programs generally access databases through one of
  - Language extensions to allow embedded SQL
  - Application program interface (e.g. ODBC/JDBC) which allow SQL queries to be sent to a database

Relational Model Concepts

- A relational database consists of a collection of tables , each of witch is assigned a unique name .
- Represents data as a collection of relations
- **Table** of values
  - Row
    - Represents a collection of related data values
    - Fact that typically corresponds to a real-world entity or relationship
    - *Tuple*
  - Table name and column names
    - Interpret the meaning of the values in each row *attribute*.

Relational instance *refers to a specific instance of a relation*

Relational Model Concepts (cont'd.)



Name	Ssn	Home_phone	Address	Office_phone	Age	Gpa
Benjamin Bayer	305-61-2435	(817)373-1616	2918 Bluebonnet Lane	NULL	19	3.21
Chung-cha Kim	381-62-1245	(817)375-4409	125 Kirby Road	NULL	18	2.89
Dick Davidson	422-11-2320	NULL	3452 Elgin Road	(817)749-1253	25	3.53
Rohan Panchal	489-22-1100	(817)376-9821	265 Lark Lane	(817)749-6492	28	3.93
Barbara Benson	533-69-1238	(817)839-8461	7384 Fontana Lane	NULL	19	3.25

Figure 3.1
The attributes and tuples of a relation STUDENT.

Attribute Types

- The set of allowed values for each attribute is called the domain of the attribute.
- Attribute values are required to be atomic قيمة واحدة; that is indivisible(بقيمة وحيدة)
- The null value means value is unknown or does not exit .
- The special value null is a member of every domain.
- The null value cause complications in in the definition of many operations.

We require that, for all relations r , the domains of all attributes of r be atomic . A domain is atomic if elements of the domain are considered to be indivisible Units.

Relation schema and Instance

- **Relation schema** R
  - Denoted by $R = (A_1, A_2, \ldots, A_n)$
  - Made up of a relation name R and a list of attributes, $A_1, A_2, \ldots, A_n$
- Example : Student = (ID , name , dept_name , Avg)
- Attribute A_i
  - Name of a role played by some domain D in the relation schema R
  - *The current values (relation instance) of a relation are specified by a table*
  - *An element t of R is a tuple , represented by a row in a table*

Database

- A database consists of multiple relations .
- Information about an organization is broken into parts

  instructor

  Student

  Advisor
- Bad design univ (instructor-Id , name , dept_name, student_id,….)

Result in

- repetition of information (e.g. , two students have the same instructor)
- the need for null values (e.g. , represent a student with no advisor)

- Normalization theory deals with how to design “good” relation schemas .

Relations are Unordered

- Order of tuples is irrelevant (tuples may be stored in an arbitrary order)
- Example: *instructor* relation with unordered tuples

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

Entity Sets

- A *database* can be modeled as:
  - a collection of entities,
  - relationship among entities.
- An ***entity*** is an object that exists and is distinguishable from other objects.
  - Example: specific person, company, event, plant
- Entities have *attributes*
  - Example: people have *names* and *addresses*
- An ***entity set*** is a set of entities of the same type that share the same properties.
  - Example: set of all persons, companies, trees, holidays , set of students have the ages between 18-21 ,…

Entity Sets instructor and student

instructor_ID	instructor_name
76766	Crick
45565	Katz
10101	Srinivasan
98345	Kim
76543	Singh
22222	Einstein

instructor

student-ID	student_name
98988	Tanaka
12345	Shankar
00128	Zhang
76543	Brown
76653	Aoi
23121	Chavez
44553	Peltier

student

Relationship Sets

- A relationship is an association among several entities

  Example:

  44553 (Peltier) advisor 22222 (Einstein) student entity

  relationship set instructor entity

  A *relationship* set is a mathematical relation among $n \geq 2$ entities, each taken from entity sets

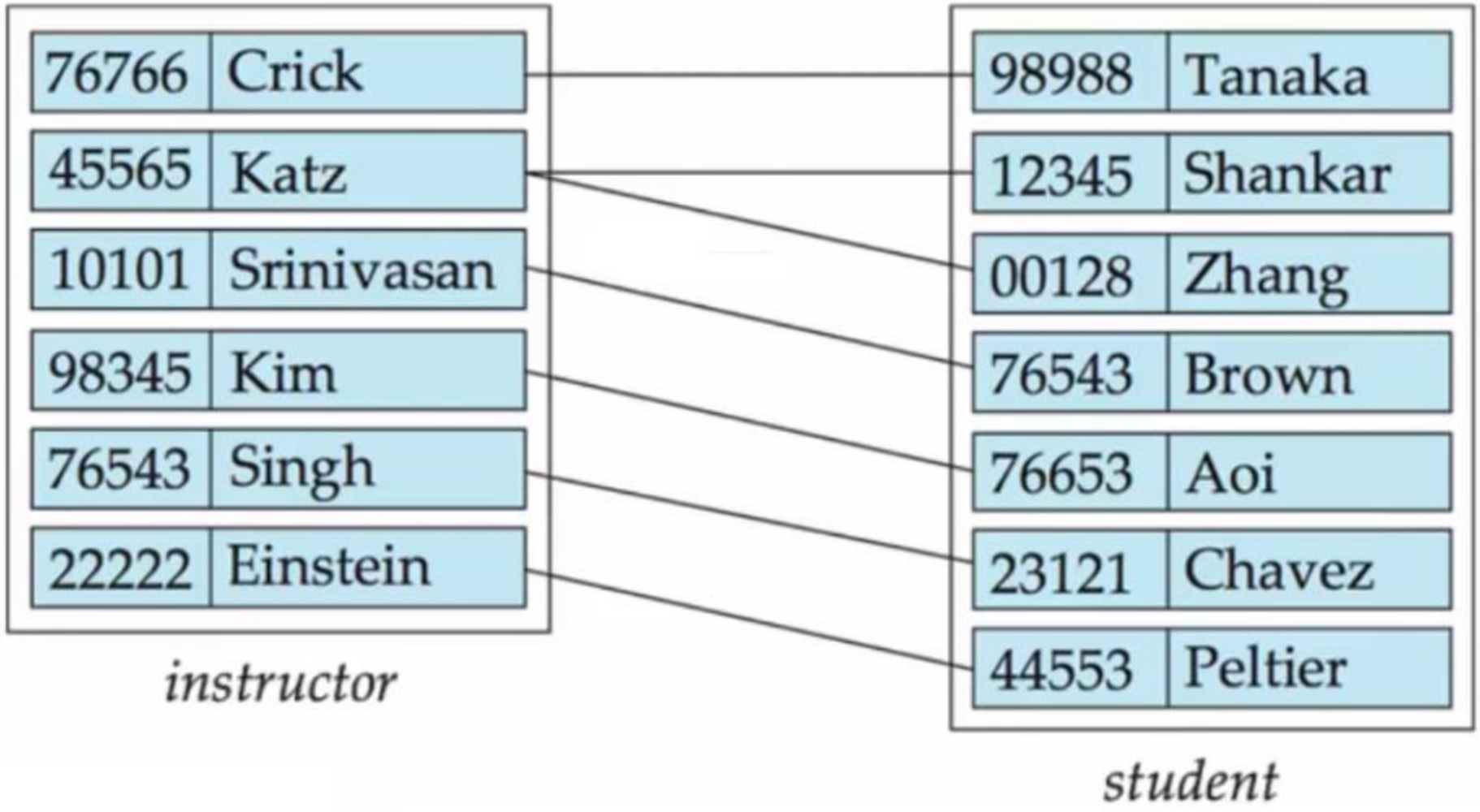
  $$\{(e_1, e_2, \ldots e_n) \mid e_1 \in E_1, e_2 \in E_2, \ldots, e_n \in E_n\}$$

  where $(e_1, e_2, \ldots, e_n)$ is a relationship

  - Example:

    $$(44553, 22222) \in advisor$$

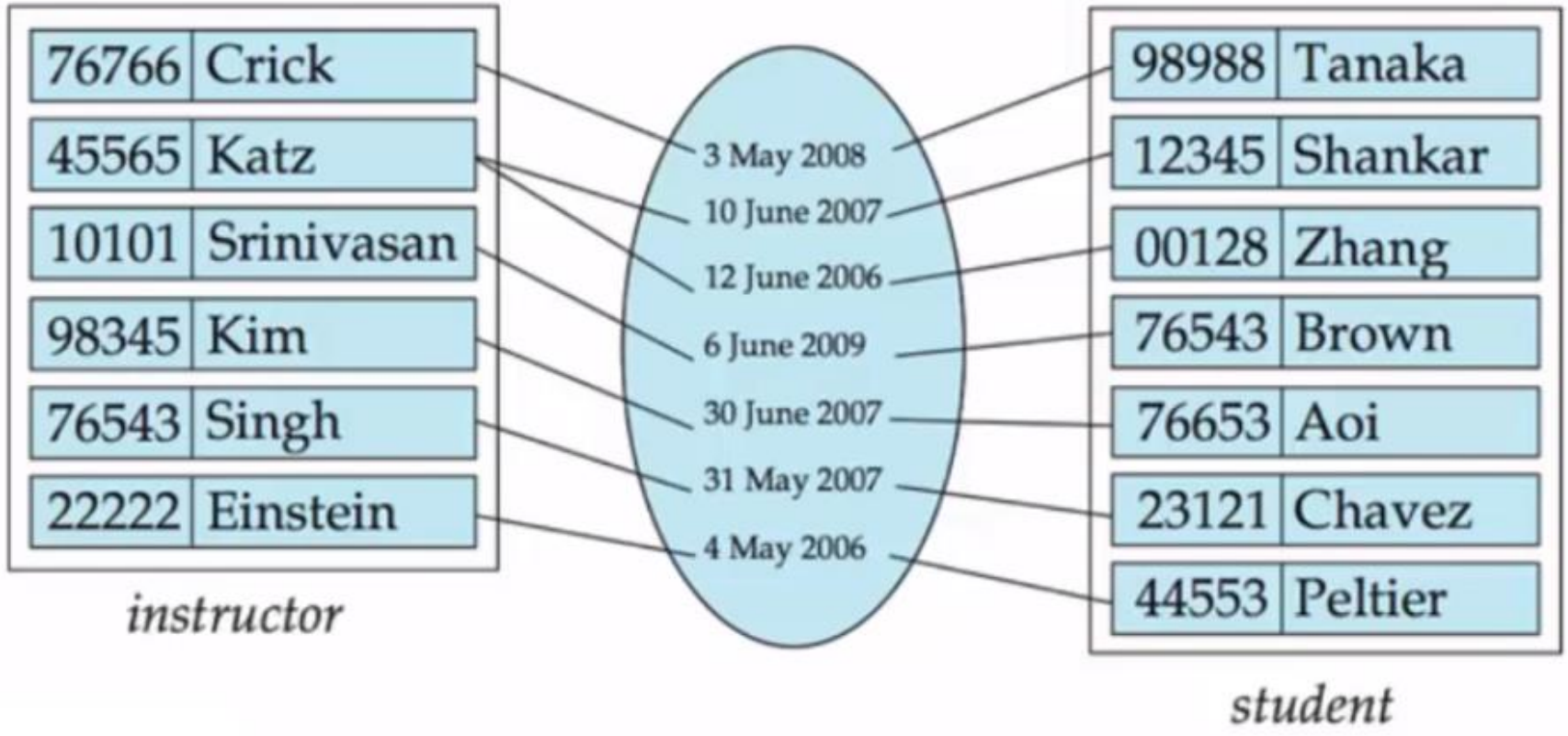
Relationship Set advisor



Relationship Sets (Cont.)

An **attribute** can also be property of a relationship set.

For instance, the *advisor* relationship set between entity sets *instructor* and *student* may have the attribute *date* which tracks when the student started being associated with the advisor



Keys

- A *super key* of an entity set is a set of one or more attributes whose values uniquely determine each entity.
- A *candidate key* of an entity set is a minimal super key
  - *Customer-id* is candidate key of *customer*
  - *account-number* is candidate key of *account*
- Although several candidate keys may exist, one of the candidate keys is selected to be the *primary key*.
- The primary key should be chosen such that its attribute values are never , or very rarely, changed.

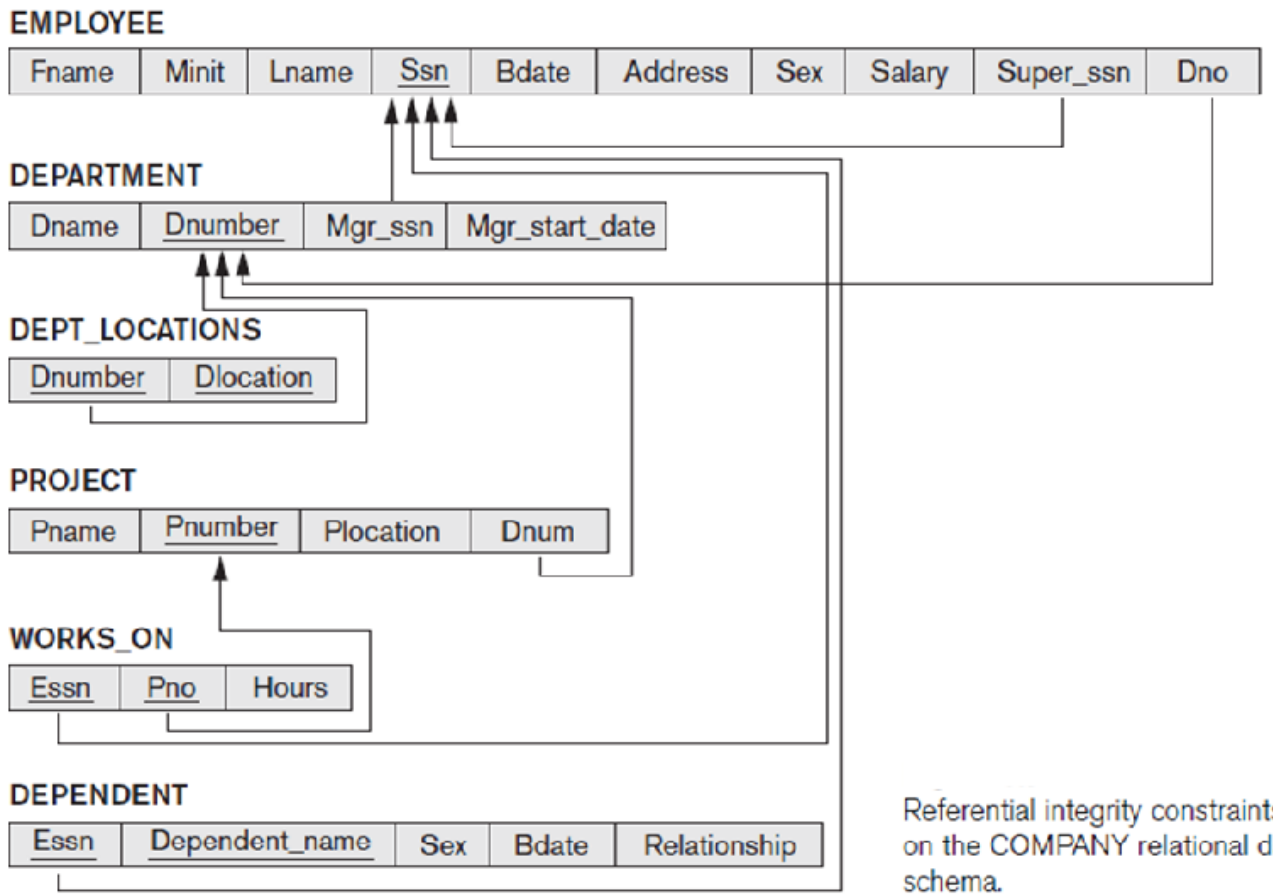
Keys

- A relation , say r1 , may include the primary key of another key of another relation ,say r2, This attribute is called a foreign key from r1,referening r2. the relation r1 is also called the referencing relation of the foreign key dependency , and r2 is called the referenced relation of the foreign key .

Keys for Relationship Sets

- The combination of primary keys of the participating entity sets forms a super key of a relationship set.
  - (*s-id*, *i-id*) is the super key of *advisor*
  - *NOTE: this means a pair of entity sets can have at most one relationship in a particular relationship set.*
    - E.g. if we wish to track multiple meeting dates between a student and his advisor, we cannot assume a relationship for each meeting. We can use a multivalued attribute though
- Must consider the mapping cardinality of the relationship set when deciding the what are the candidate keys
- Need to consider semantics of relationship set in selecting the *primary key* in case of more than one candidate key

COMPANY relation database schema



Referential integrity constraints displayed on the COMPANY relational database schema.

Attributes

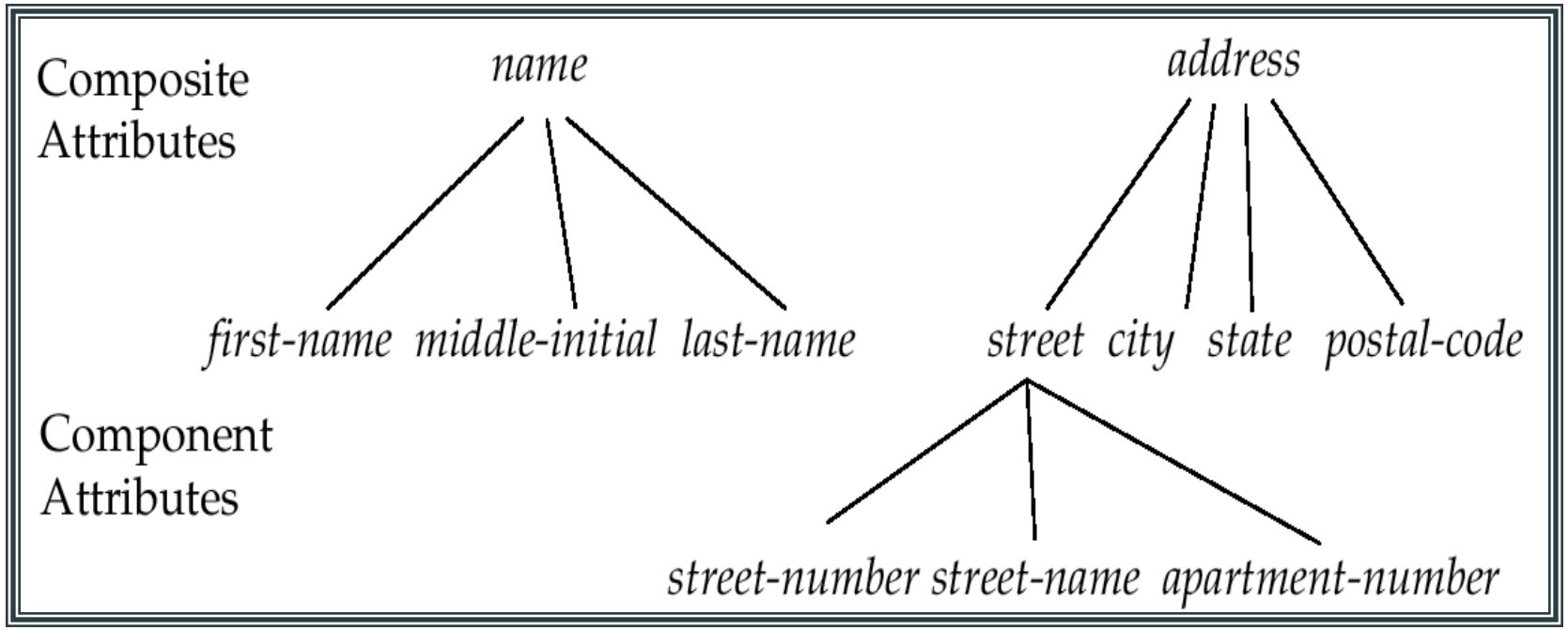
- An entity is represented by a set of attributes, that is descriptive properties possessed by all members of an entity set.

Example:

instructor = (ID, name,Street ,city ,salary)
course = (course_id , title , credits)

- *Domain* – the set of permitted values for each attribute
- Attribute types:
  - *Simple* and *composite* attributes.
  - *Single-valued* and *multi-valued* attributes
    - E.g. multivalued attribute: *phone-numbers*
  - *Derived* attributes
    - Can be computed from other attributes
    - E.g. *age*, given date of birth

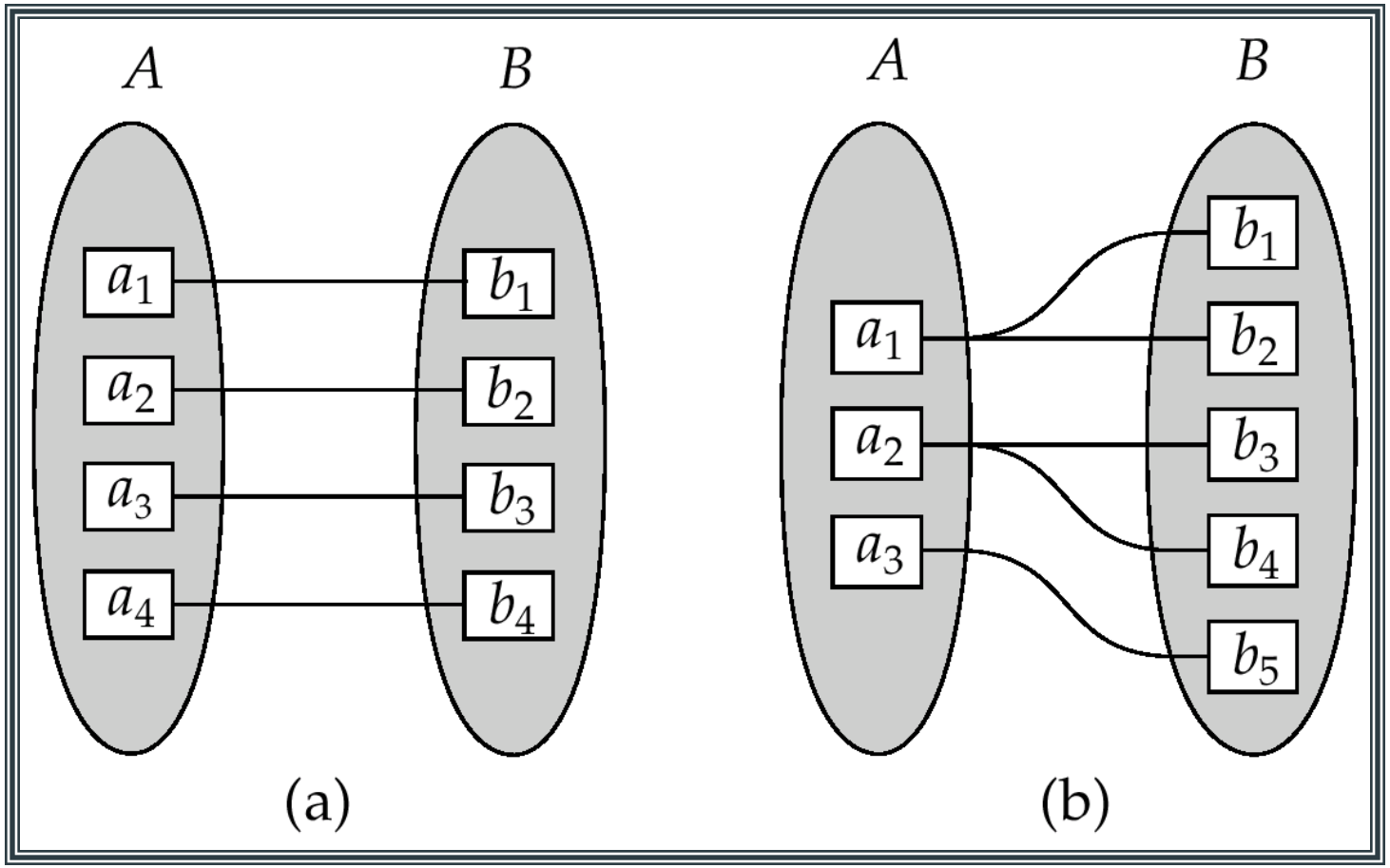
Composite Attributes



Mapping Cardinalities

- Express the number of entities to which another entity can be associated via a relationship set.
- Most useful in describing binary relationship sets.
- For a binary relationship set the mapping cardinality must be one of the following types:
  - One to one
  - One to many
  - Many to one
  - Many to many

Mapping Cardinalities

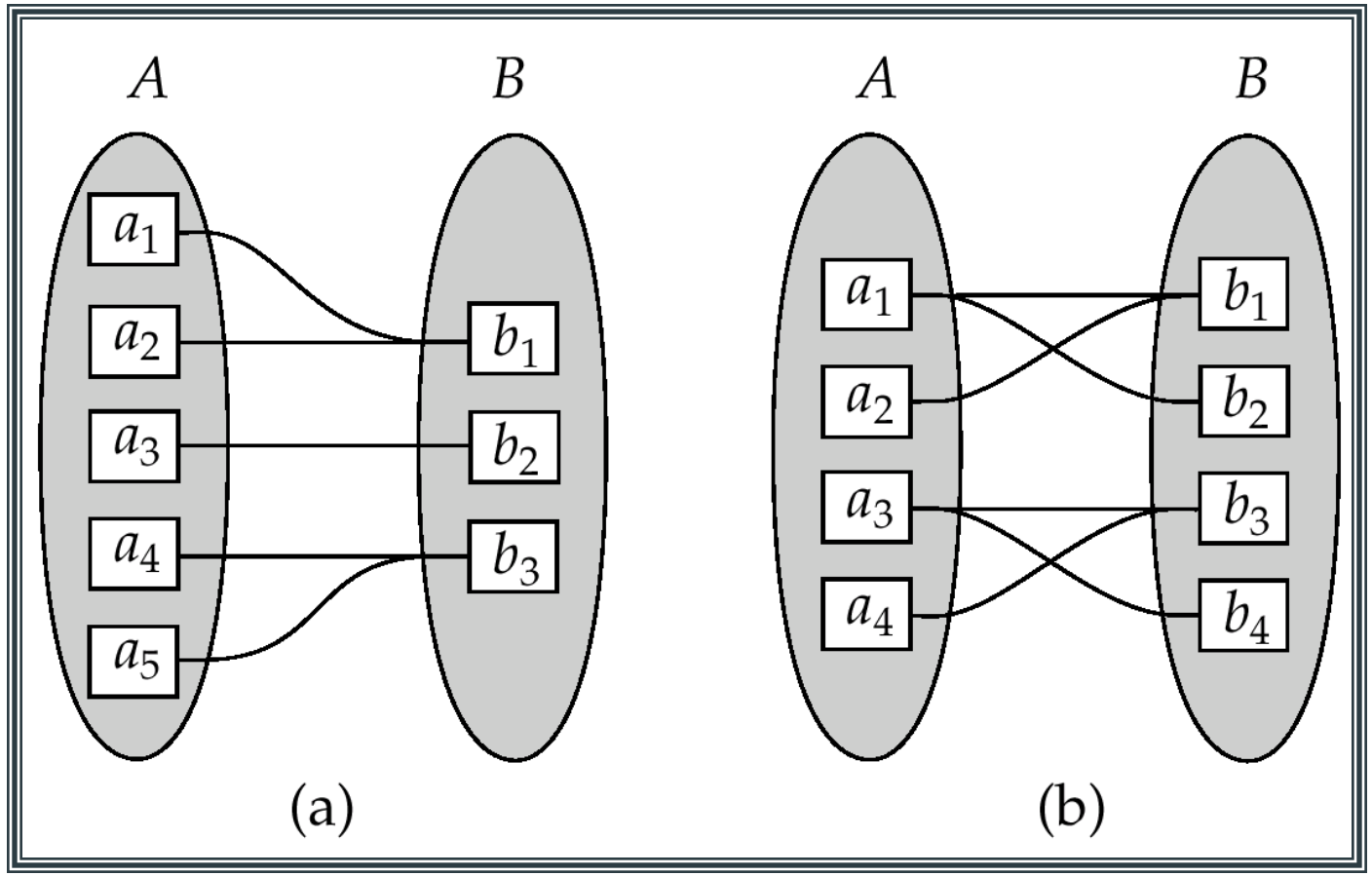


One to one

One to many

Note: Some elements in A and B may not be mapped to any elements in the other set

Mapping Cardinalities



Many to one

Many to many

Note: Some elements in A and B may not be mapped to any elements in the other set

E-R Diagram

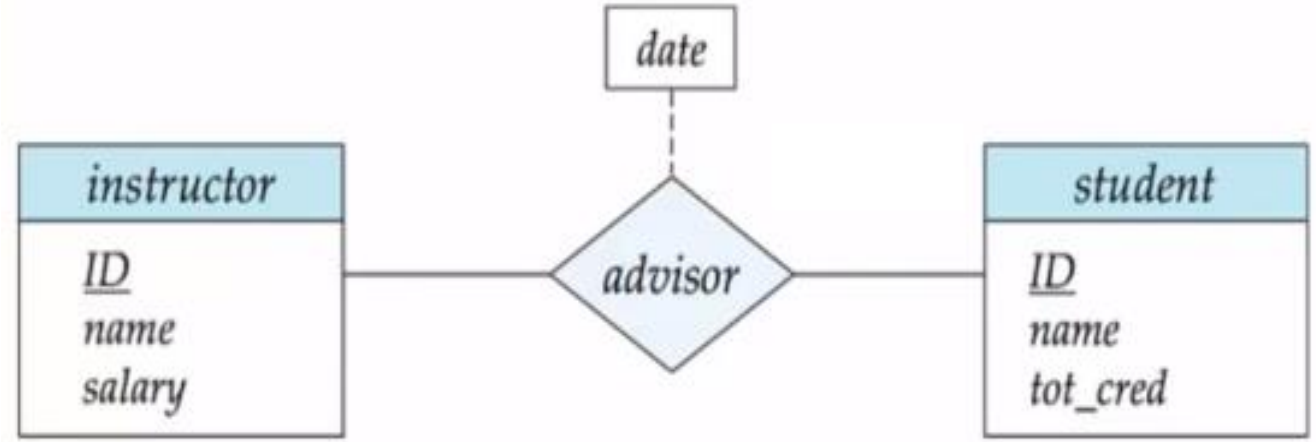
Rectangles represent entity sets.

Diamonds represent relationship sets.

Attributes listed inside entity rectangle

Underline indicates primary key attributes

Relationship Sets with Attributes

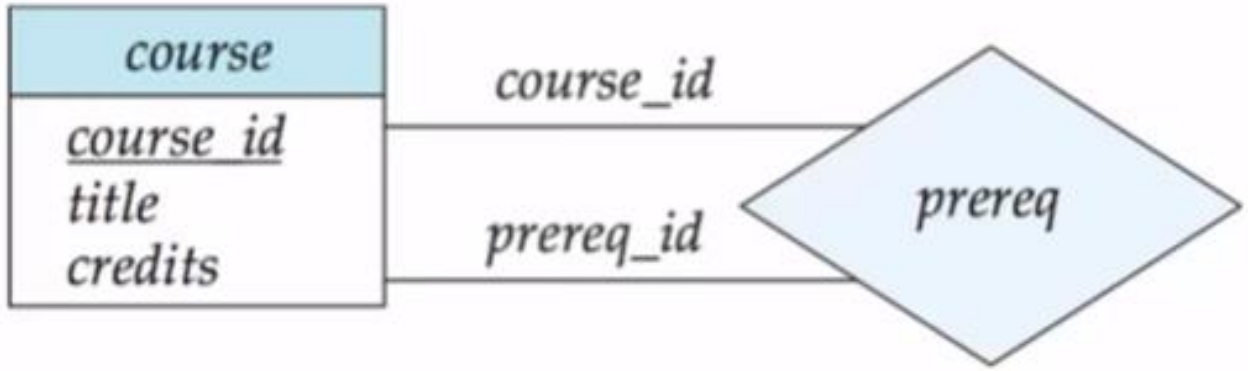


Roles

Entity sets of a relationship need not be distinct

Each occurrence of an entity set plays a "role" in the relationship

The labels "*course_id*" and "*prereq_id*" are called **roles**.



Cardinality Constraints

- We express cardinality constraints by drawing either a directed line ($\rightarrow$), signifying “one,” or an undirected line ($-$), signifying “many,” between the relationship set and the entity set.

One – to – one Relationship

one-to-one relationship between an *instructor* and a *student*

- an instructor is associated with at most one student via *advisor*
- and a student is associated with at most one instructor via *advisor*

One – to – Many Relationship

one-to-many relationship between an *instructor* and a *student*

- an instructor is associated with several (including 0) students via *advisor*
- a student is associated with at most one instructor via advisor,

Many – to – one Relationship

In a many-to-one relationship between an *instructor* and a *student*,

- an instructor is associated with at most one student via *advisor*,
- and a student is associated with several (including 0) instructors via *advisor*

Many – to – Many Relationship

An instructor is associated with several (possibly 0) students via *advisor*

A student is associated with several (possibly 0) instructors via *advisor*

Thanks for your kind attention

Mohammed khillah