Entity-Relationship Model

Dr. Odelu Vanga

Indian Institute of Information Technology Sri City

http://www.iiits.ac.in/people/regular-faculty/dr-odelu-vanga/

Entity-Relationship Model

- Entity Sets
- Relationship Sets
- Keys
- Mapping Constraints
- E-R Diagram

Entity Sets

- A ***database*** can be modeled as:
  - a collection of **entities**,
  - **relationship** among entities.
- ***Entity*** is an object that exists and is distinguishable from other objects.

  Example: specific person, student, instructor, 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

Entity Sets *customer* and *loan*

Customer ID	Customer Name	Customer Street	Customer City
321-12-3123	Jones	Main	Harrison
019-28-3746	Smith	North	Rye
677-89-9011	Hayes	Main	Harrison
555-55-5555	Jackson	Dupont	Woodside
244-66-8800	Curry	North	Rye
963-96-3963	Williams	Nassau	Princeton
335-57-7991	Adams	Spring	Pittsfield

customer

Loan Number	Amount
L-17	1000
L-23	2000
L-15	1500
L-14	1500
L-19	500
L-11	900
L-16	1300

loan

Attributes

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

  Example: *customer = (customer-id,*

  customer-name,

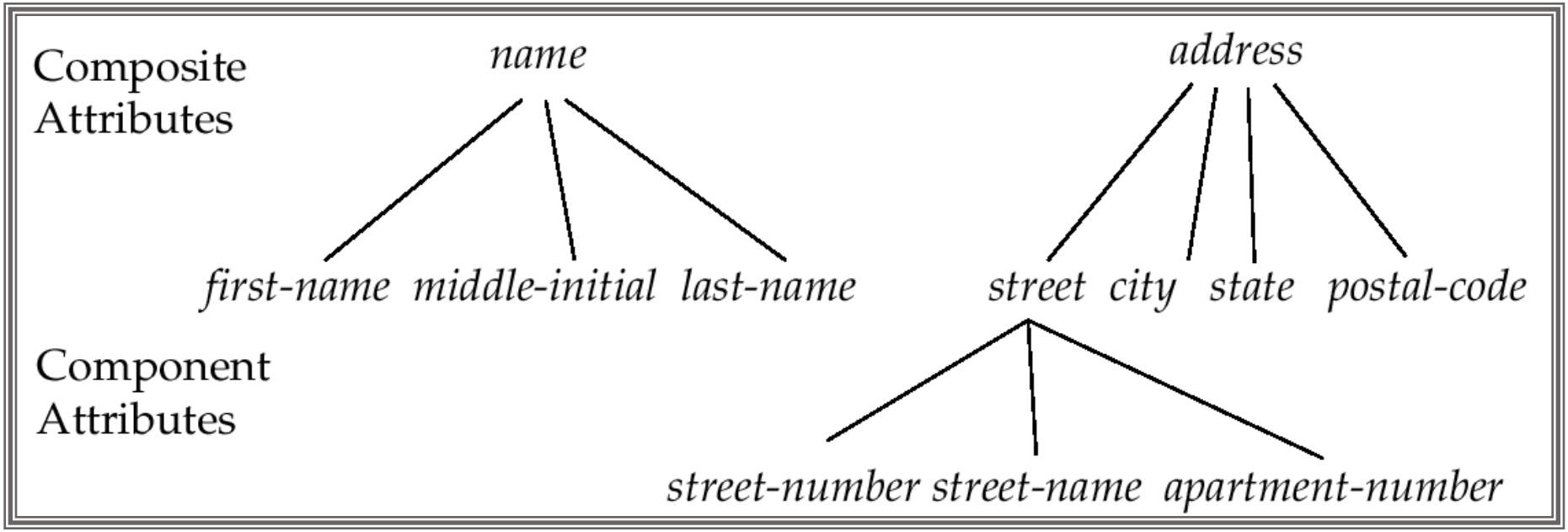
  customer-street,

  customer-city)

  loan = (loan-number, amount)

- ***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***, from given date of birth

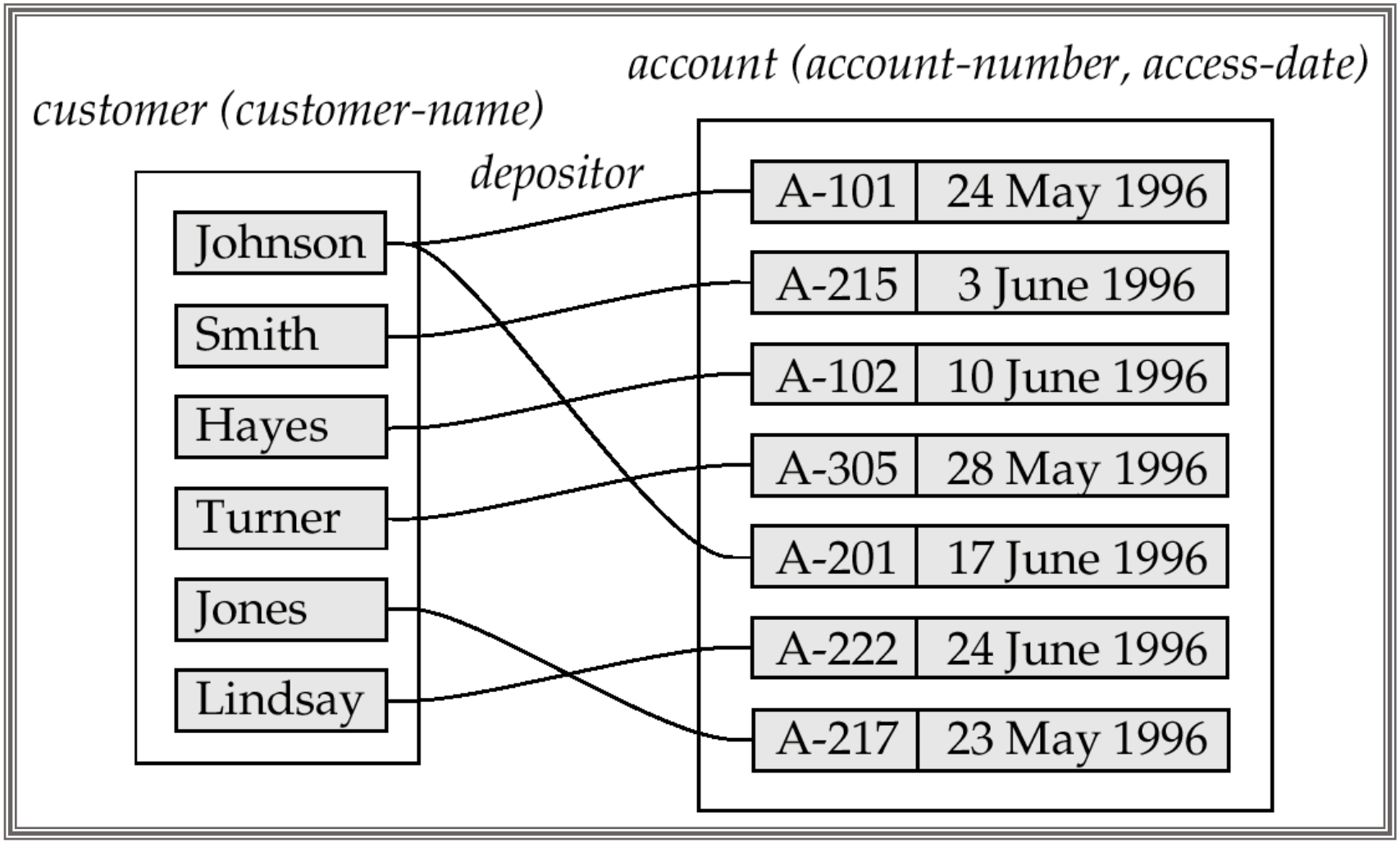
Composite Attributes



Relation Formal Terms

Informal Terms		Formal Terms
Table		Relation
Column Header		Attribute
All possible Column Values		Domain
Row		Tuple
Table Definition		Schema of a Relation
Populated Table		State of the Relation

Relationship Sets



Relationship Sets

- A relationship is an association among several entities
  Example:

Hayes	*depositor*	A-102
customer entity	relationship set	*account* entity

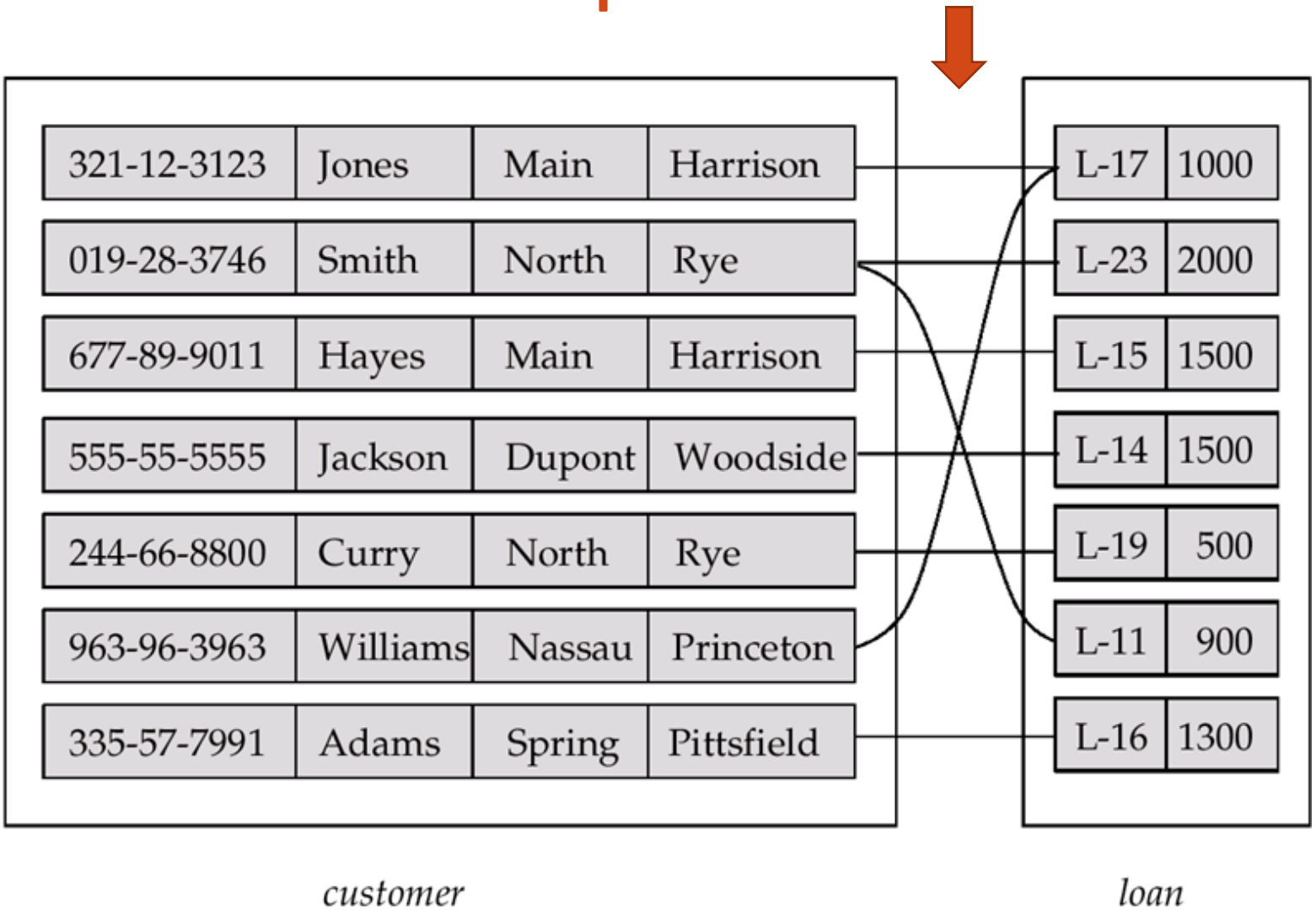
- A *relationship* set is a mathematical relation among $n \geq 2$ entities, each taken from entity sets

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

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

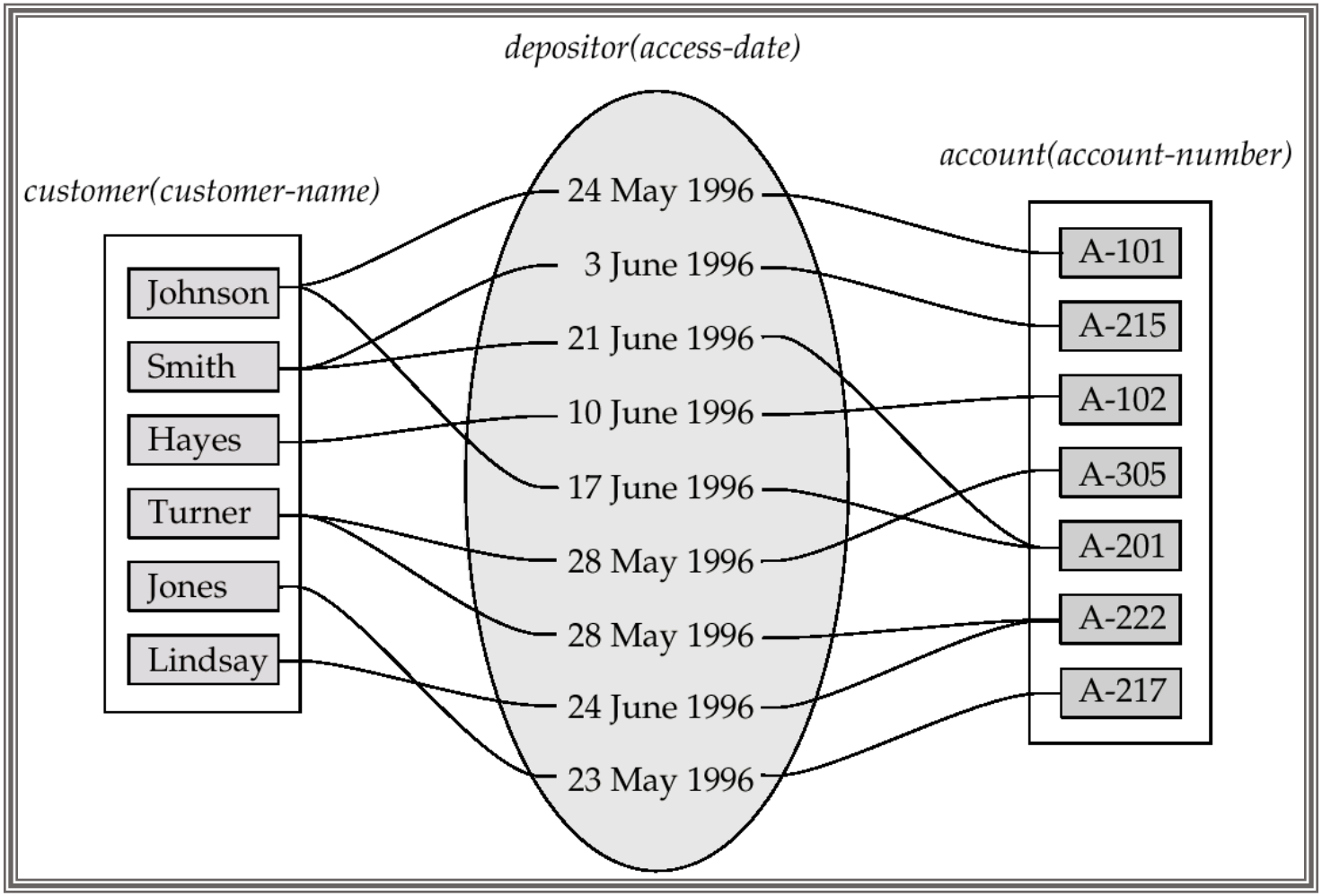
Example: (Hayes, A-102) $\in$ *depositor*

Relationship Set *borrower*



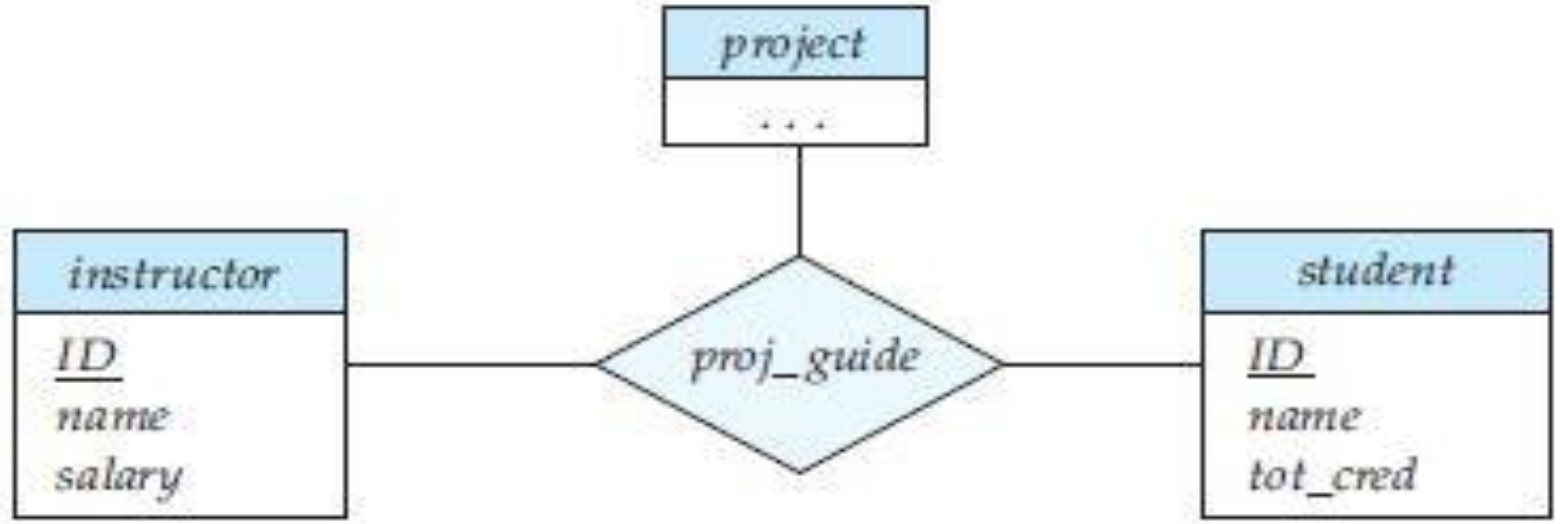
Relationship Sets (Cont.)

- An *attribute* can also be property of a relationship set.
- For instance, the *depositor* relationship set between entity sets *customer* and *account* may have the attribute *access-date*



Degree of a Relationship Set

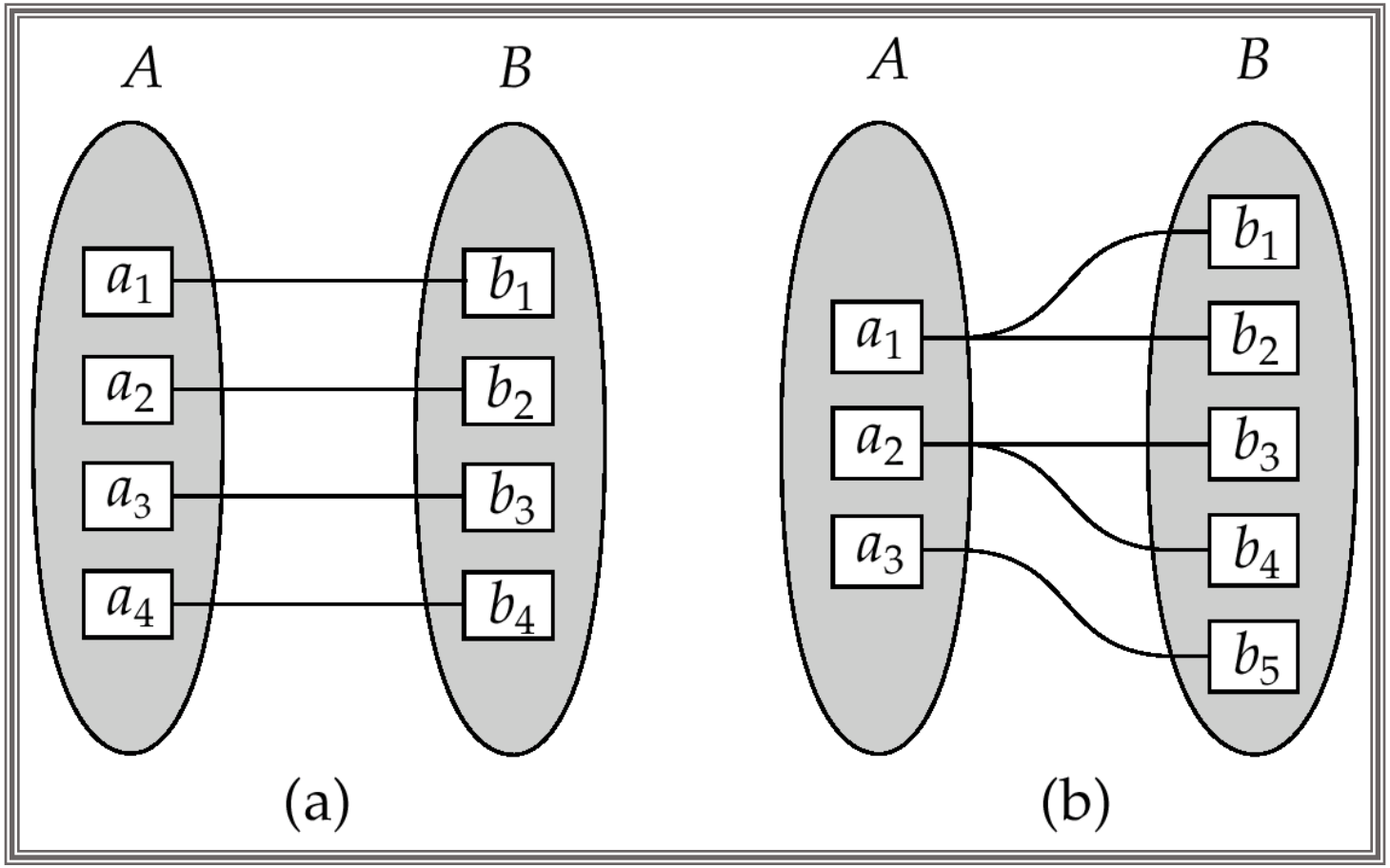
- Refers to number of entity sets that participate in a relationship set.
- Relationship sets that involve two entity sets are *binary* (or degree two).
  - Generally, most relationship sets in a database system are binary.
- Relationship sets may involve more than two entity sets.



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

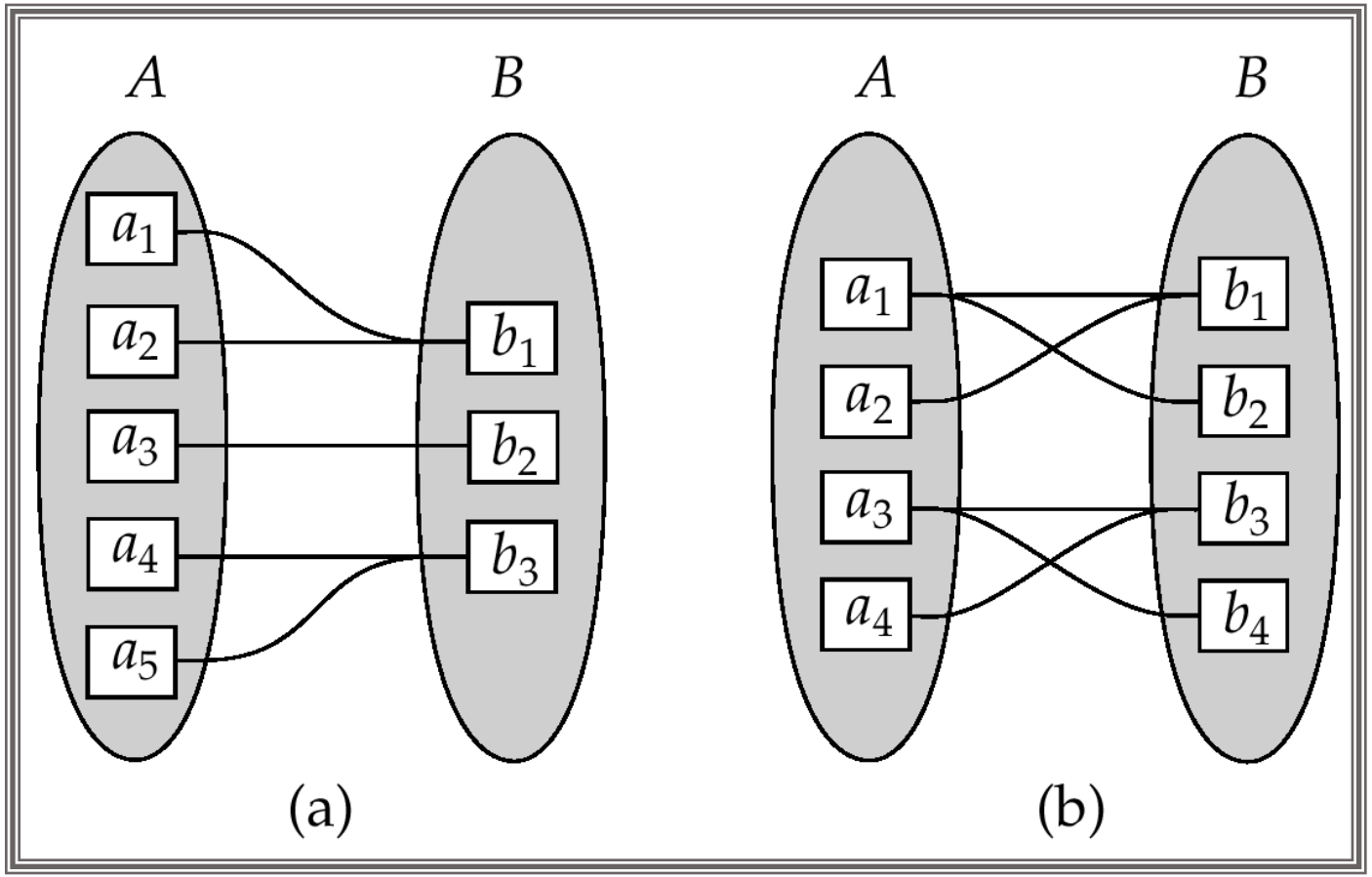


One to one

One to many

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

Mapping Cardinalities



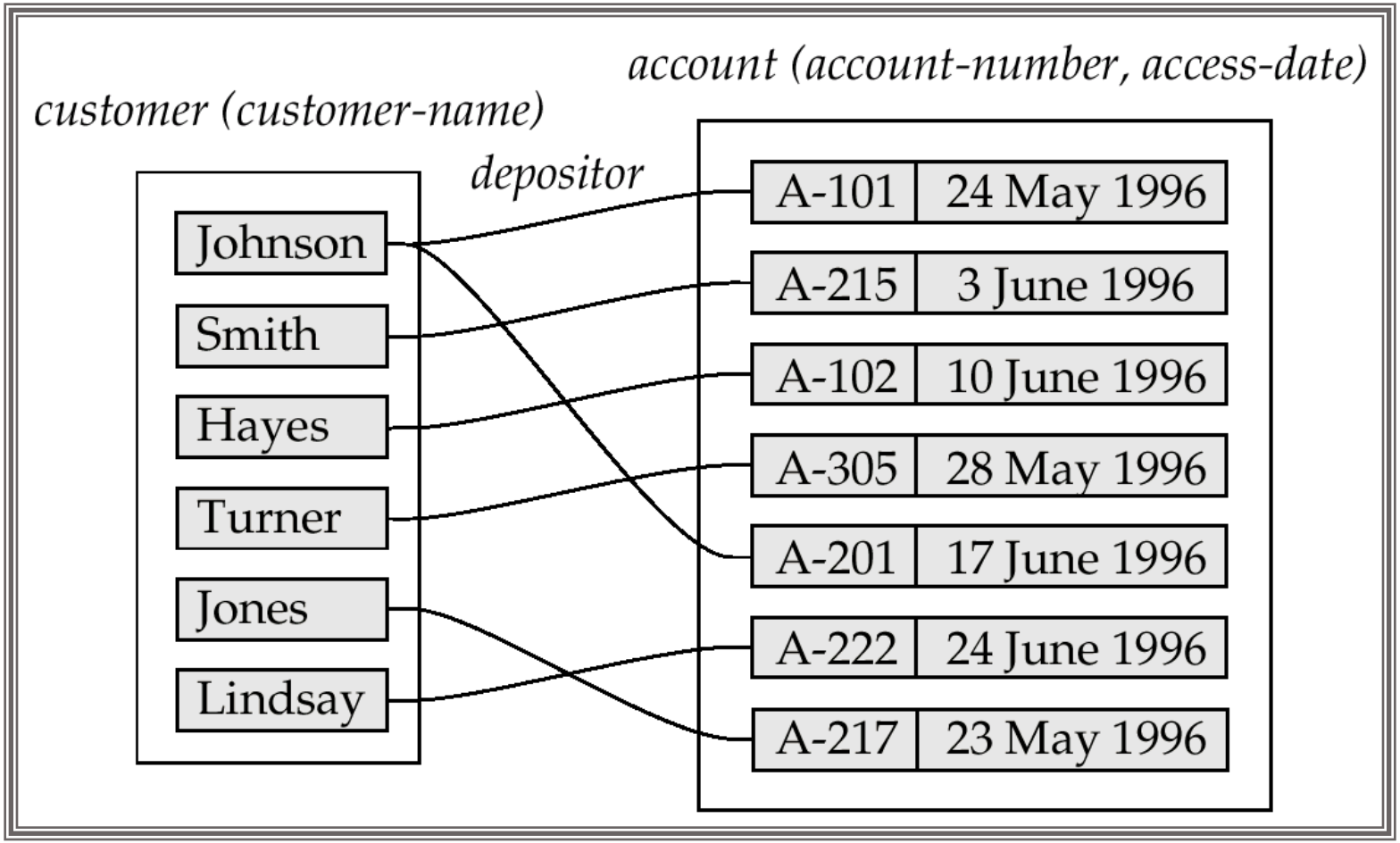
Many to one

Many to many

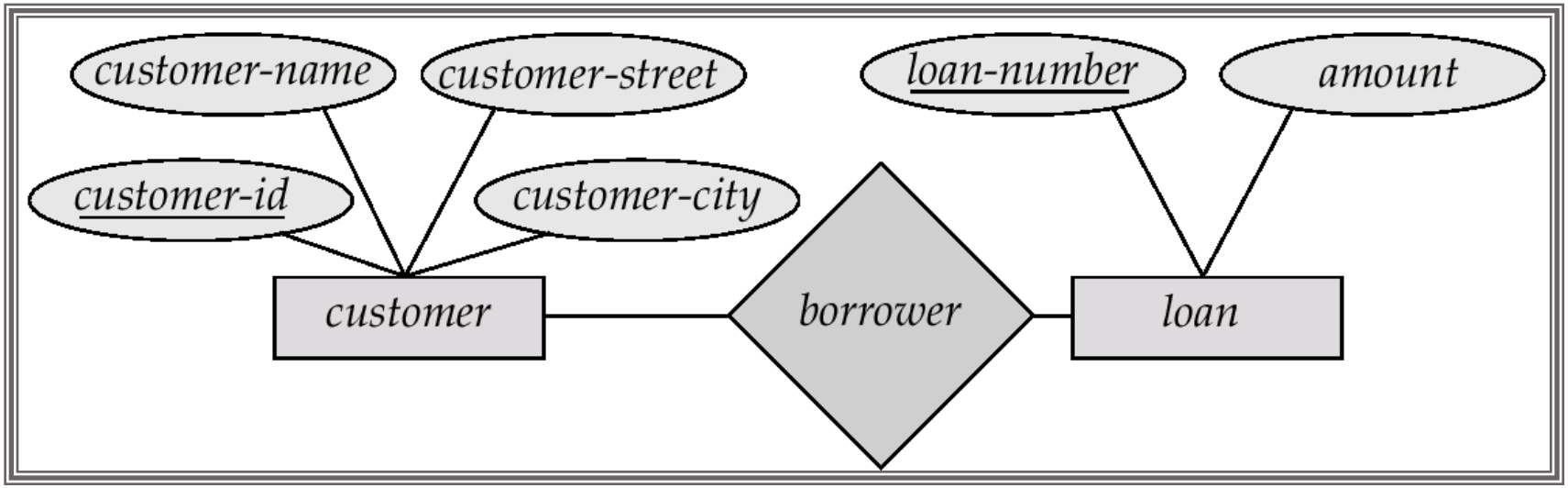
Note: Some elements in A and B may not be mapped to any elements in the other set

Mapping Cardinalities affect ER Design

- Can make *access-date* an attribute of account, instead of a relationship attribute, if each account can have only one customer
- That is, the relationship from account to customer is many to one, or equivalently, customer to account is one to many

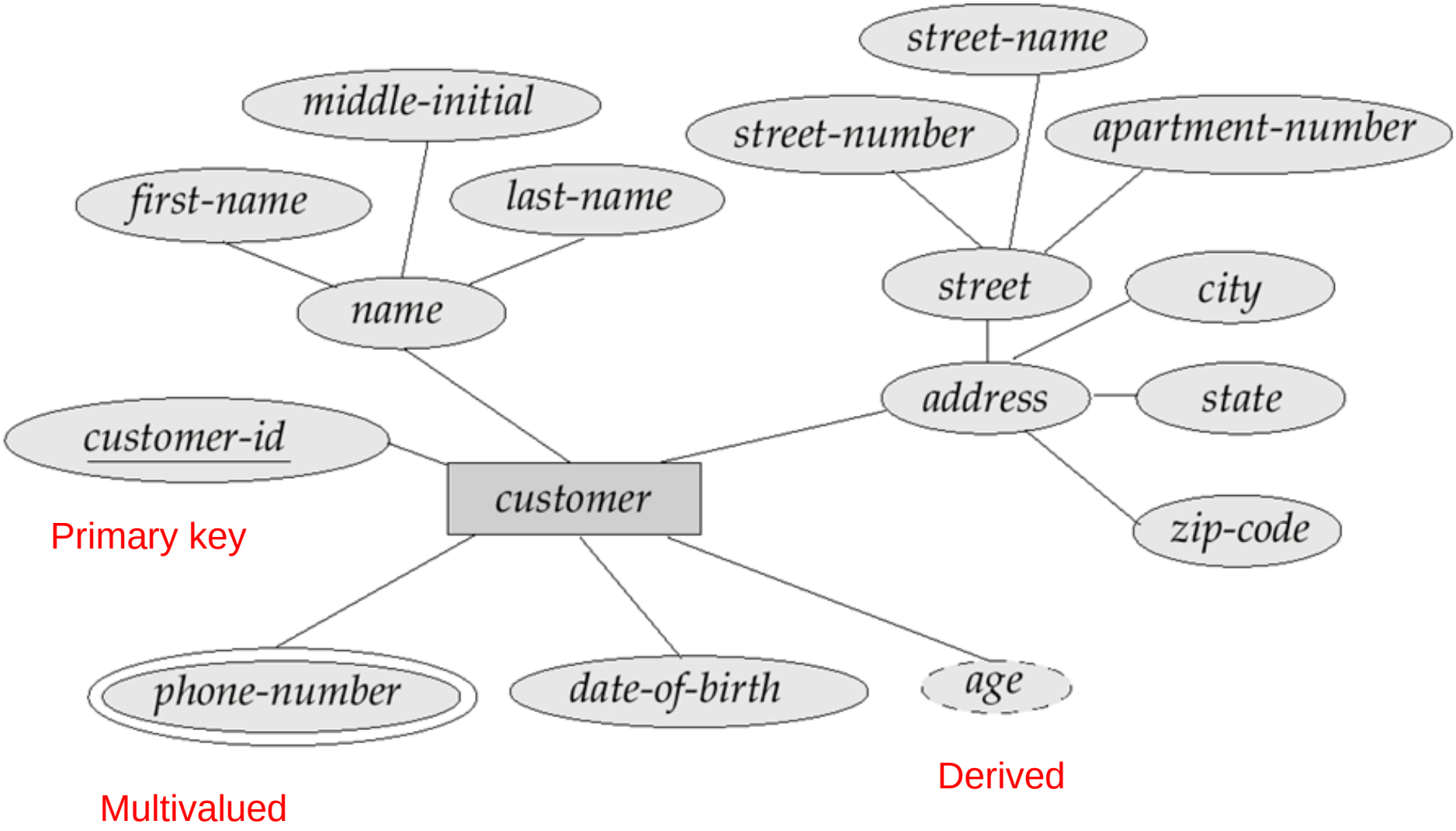


E-R Diagrams

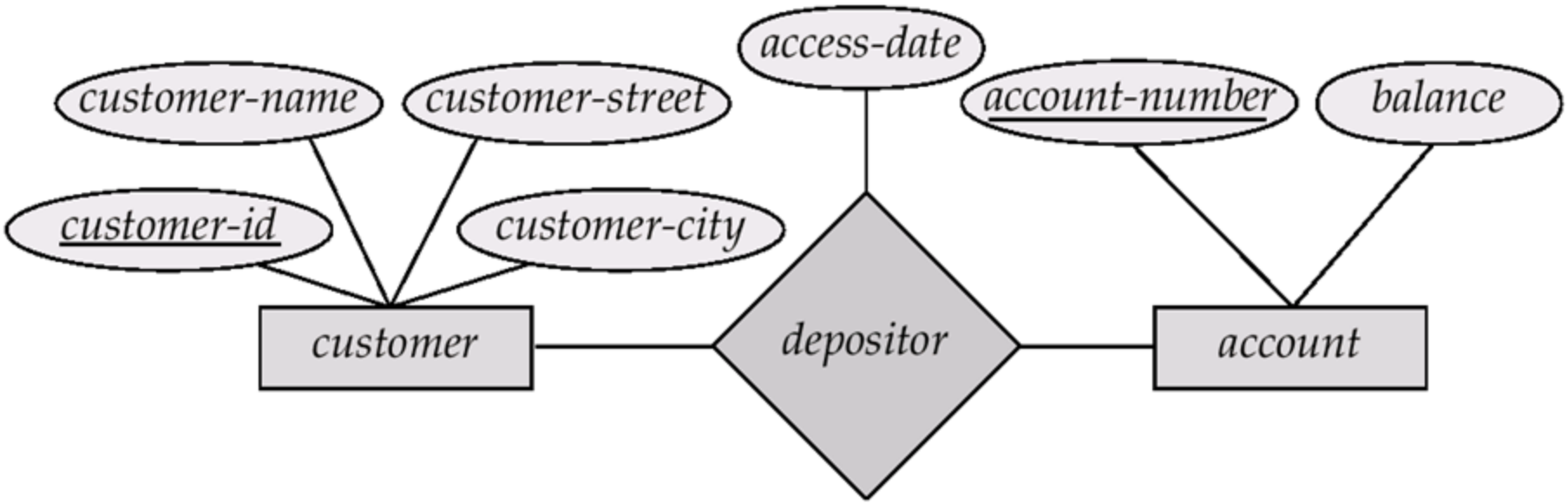


- **Rectangles** represent entity sets.
- **Diamonds** represent relationship sets.
- **Lines** link attributes to entity sets and entity sets to relationship sets.
- **Ellipses** represent attributes
  - **Double ellipses** represent multivalued attributes.
  - **Dashed ellipses** denote derived attributes.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

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

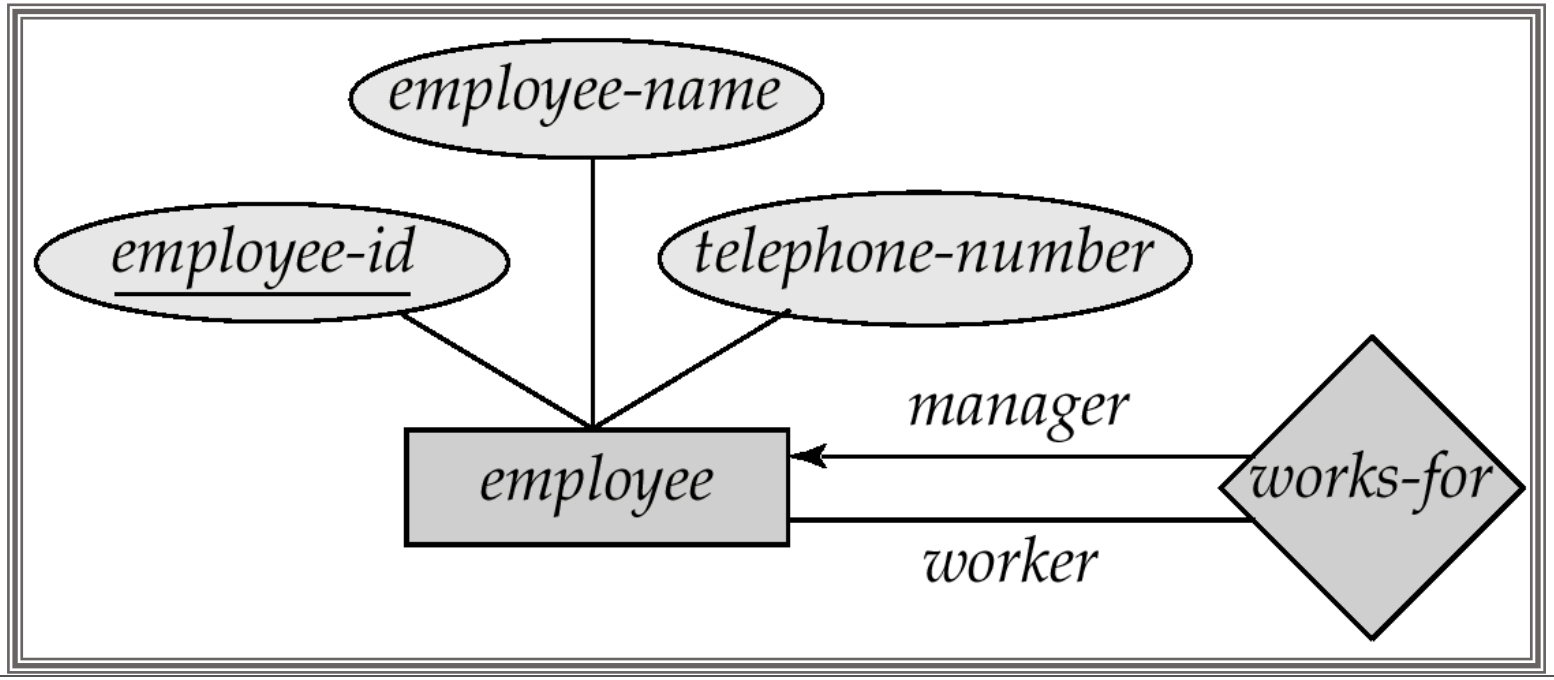


Relationship Sets with Attributes



Roles

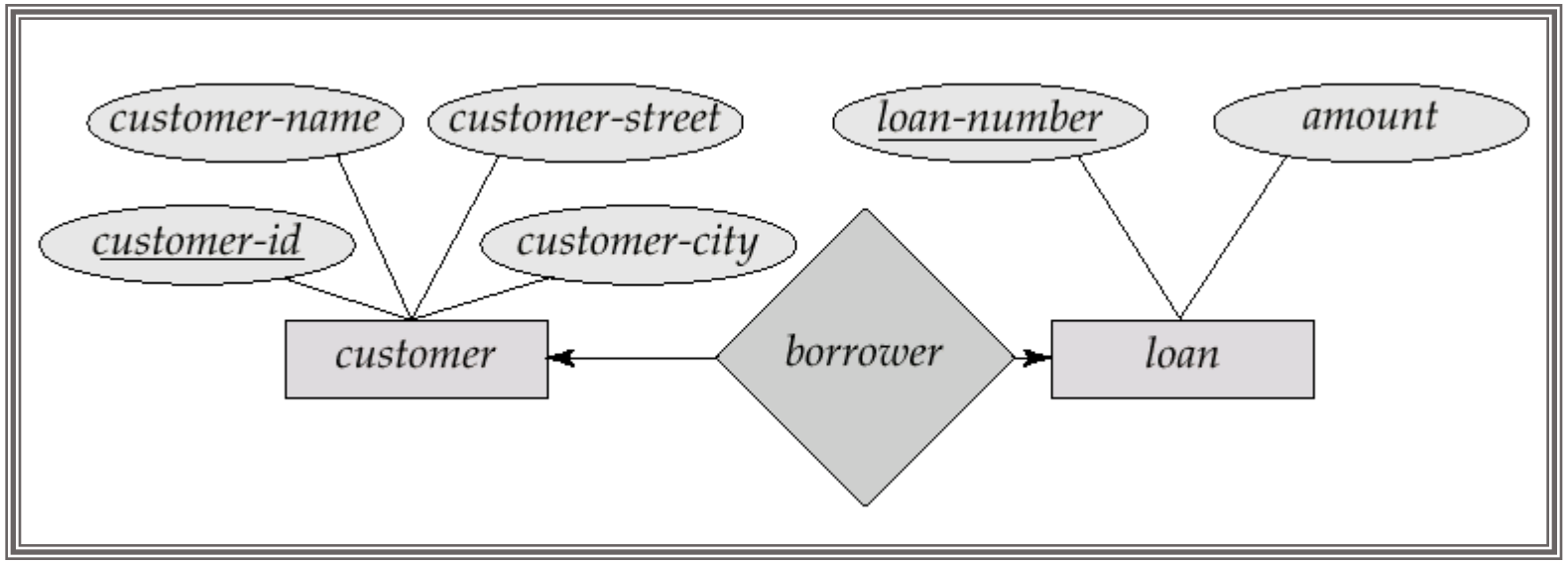
- Entity sets of a relationship need not be distinct
  - The labels "manager" and "worker" are called roles; they specify how employee entities interact via the works-for relationship set.
  - Roles are indicated in E-R diagrams by labeling the lines that connect diamonds to rectangles.
  - Role labels are optional, and are used to clarify semantics of the relationship



Cardinality Constraints

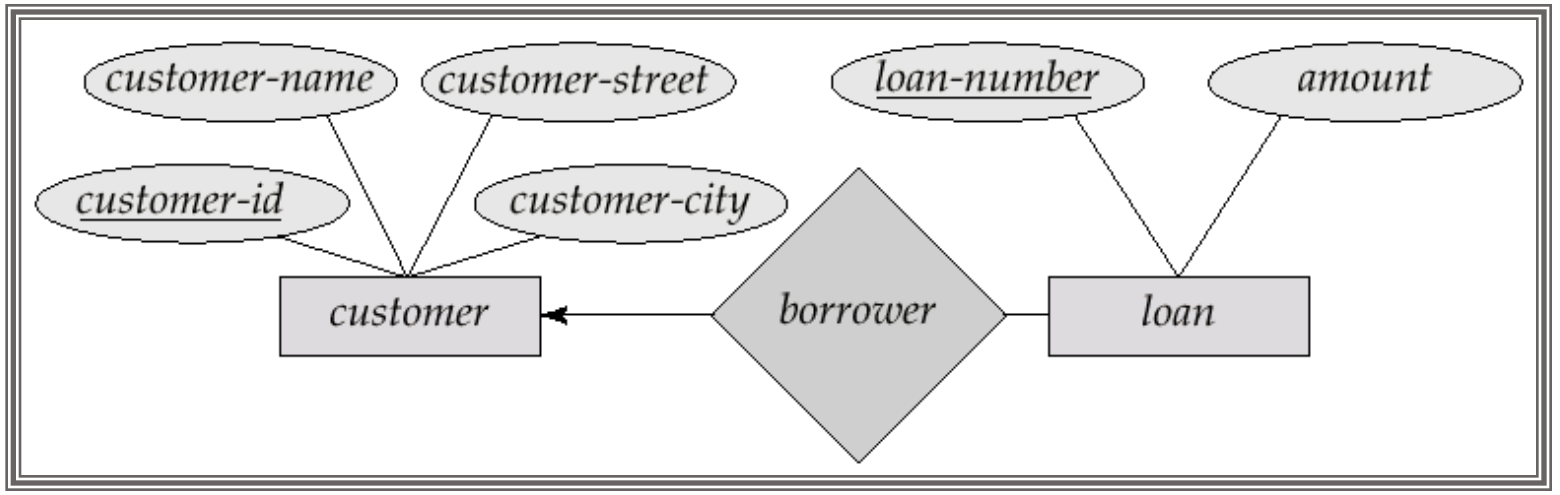
- We express cardinality constraints by drawing either a
  - directed line (→), signifying "one," or
  - an undirected line (—), signifying "many,"

  between the relationship set and the entity set.
- E.g.: One-to-one relationship:
  - A customer is associated with at most one loan via the relationship *borrower*
  - A loan is associated with at most one customer via *borrower*



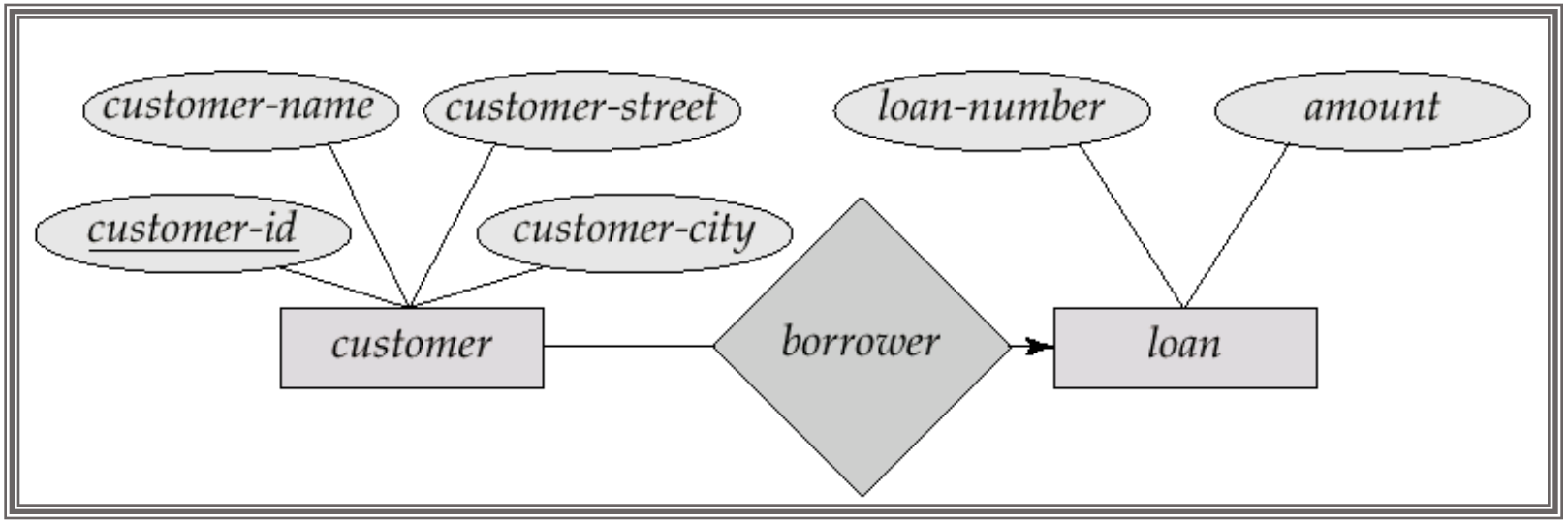
One-To-Many Relationship

- In the one-to-many relationship a loan is associated with at most one customer via *borrower*, a customer is associated with several (including 0) loans via *borrower*

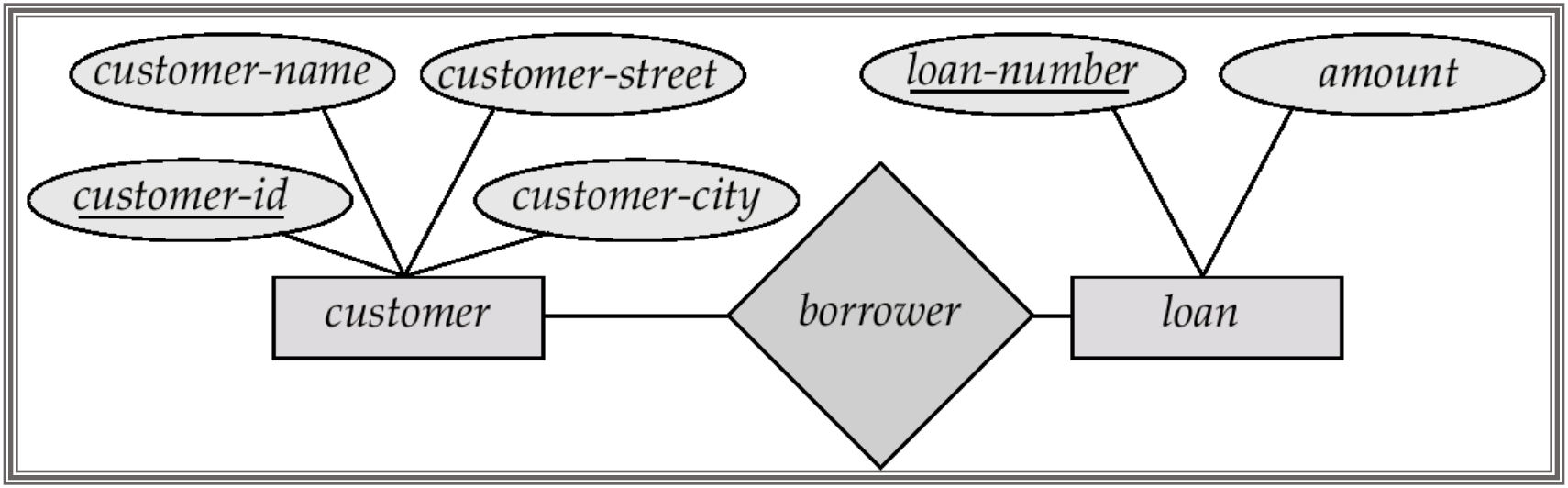


Many-To-One Relationships

- In a many-to-one relationship a loan is associated with several (including 0) customers via *borrower*, a customer is associated with at most one loan via *borrower*



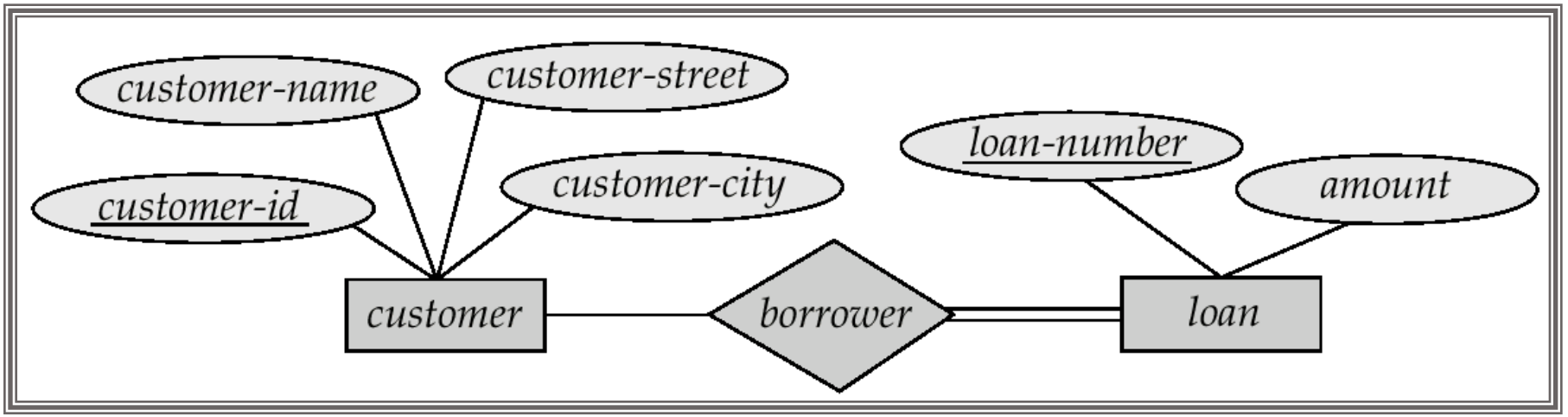
Many-To-Many Relationship



- A customer is associated with several (possibly 0) loans via borrower
- A loan is associated with several (possibly 0) customers via borrower

Participation of an Entity Set in a Relationship Set

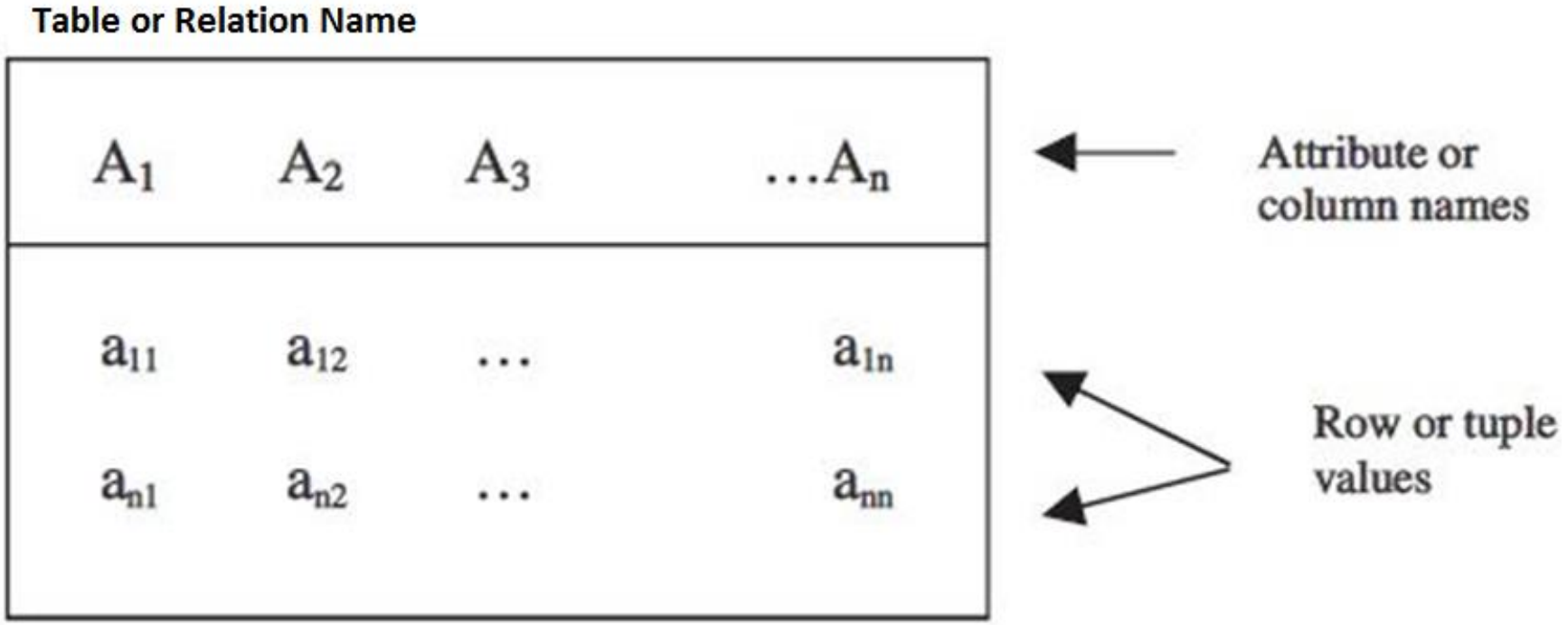
- Total participation (indicated by double line): every entity in the entity set participates in at least one relationship in the relationship set
  - E.g. participation of *loan* in *borrower* is total
    - every loan must have a customer associated to it via borrower
- Partial participation: some entities may not participate in any relationship in the relationship set
  - E.g. participation of *customer* in *borrower* is partial



Keys

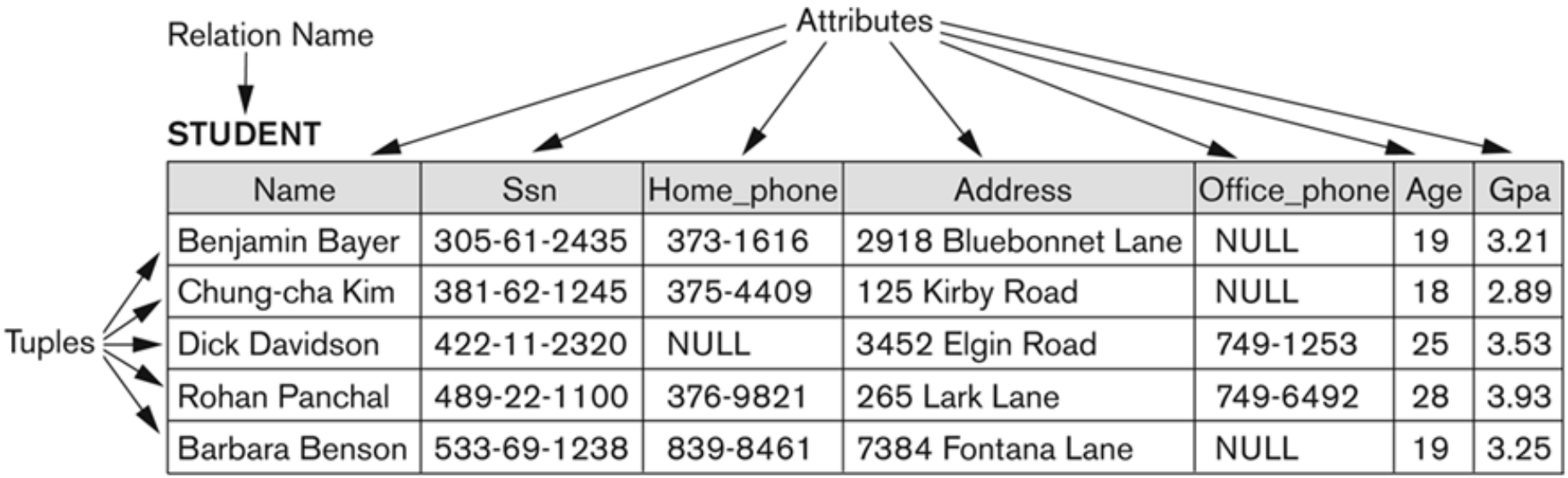
- A *super key* of an entity set is a set of one or more attributes whose values uniquely determine each entity.
- A *candidate key* of an entity set is a minimal super key
  - *Customer-id* is candidate key of *customer*
  - *account-number* is candidate key of *account*
- Although several candidate keys may exist, one of the candidate keys is selected to be the *primary key*.
- Primary must not NULL, called entity integrity constraint

General Format of a Relation when represent as Table



- Degree of Table – total number of columns
- Cardinality of Table – total number of rows at that time
- Instance of Table – the content of table at any particular point of time

Example of a Relation



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

- **IDENTIFY:**
  - **Degree**
  - **Cardinality**
  - **For each attribute, identify possible domain**

The domain of the attribute Age, denoted by Dom(Age), is a set of positive number (within the limit).

EMPLOYEE TABLE

EMPLOYEE

Id	Last_Name	First_Name	Department	Salary
555294562	Martin	Nicholas	Accounting	55000
397182093	Benakritis	Ben	Marketing	33500
907803123	Adams	Larry	Human Resources	40000

- **IDENTIFY:**
  - **Degree**
  - **Cardinality**
  - **For each attribute, identify possible domain**

  The domain of the attribute Id, denoted by Dom(Id), is a set of nine-digit positive number

Relation

- The table has a unique name.
- Assume, every entry of a table has at most a single value.
- That is, at the intersection of every column and row, there is at most a single value.
- For any given relation *r*, for any attribute *A* of *r*, and an arbitrary tuple *t* of *r*,

  t(A): value of the entry of tuple *t* under the column *A*

  t calls the first tuple: *t(Id)* – 555294562, *t(Last_Name)* - ?

EMPLOYEE

Id	Last_Name	First_Name	Department	Salary
555294562	Martin	Nicholas	Accounting	55000
397182093	Benakritis	Ben	Marketing	33500
907803123	Adams	Larry	Human Resources	40000

Mathematical Definitions of a Relation

- Finite set of attributes, called relational schema

$$R = \{A_1, A_2, \ldots, A_n\}$$

- Non-empty set D_i, $(1 \leq i \leq n)$, domain of attribute A_i
- Domain $D = D_1 \cup D_2 \ldots\ldots \cup D_n$
- We define a relation r on relational schema R as a finite set of mappings $r = \{t_1, t_2, \ldots., t_k\}$ from R to D.

Table or Relation Name

A_1	A_2	A_3	$\ldots A_n$
a_{11}	a_{12}	$\ldots$	a_{1n}
a_{n1}	a_{n2}	$\ldots$	a_{nn}

If t is tuple of the relation r then $t(A_i) \in \text{Dom}(A_i)$

Candidate Key and Primary Key

- **Candidate Key:** Any subset $K = \{A_1, A_2, \ldots, A_k\}$ with $1 \leq k \leq n$ for given a relation r and its attributes $\{A_1, A_2, \ldots, A_n\}$, which satisfies:
  - For any $t_1 \neq t_2$ of the relation *r*, there exists an attribute A_j of *K* such that $\boldsymbol{t_1(A_j) \neq t_2(A_j)}$ – **Uniqueness property of the key**
  - No proper subset *K'* of *K* satisfy the uniqueness property – **Minimality Property of the key**
- Although several candidate keys may exist, one of the candidate keys is selected to be *primary key* (one per table)
  - Single-primary key and Composite primary key
  - Primary keys defined using DDL, at the time of table created
- The remaining candidate keys are called *alternate keys*
- *Prime attributes* –part of primary or alternate keys

CAR table with two candidate keys

CAR

License_number	Engine_serial_number	Make	Model	Year
Texas ABC-739	A69352	Ford	Mustang	02
Florida TVP-347	B43696	Oldsmobile	Cutlass	05
New York MPO-22	X83554	Oldsmobile	Delta	01
California 432-TFY	C43742	Mercedes	190-D	99
California RSK-629	Y82935	Toyota	Camry	04
Texas RSK-629	U028365	Jaguar	XJS	04

- **Candidate Keys: ??**
  License_number and Engine_serial_number

- LicenseNumber chosen as **Primary Key**

Example Department

DEPT

Assume DEPARTMENT is a key

DEPARTMENT	NAME	LOCATION	BUDGET
20	Sales	Miami	1700000
10	Marketing	New York	2000000

Can we insert the rows ?

DEPARTMENT	NAME	LOCATION	BUDGET
10	Research	New York	1500000
	Accounting	Atlanta	1200000
15	Computing	Miami	1500000

10	Research	New York	1500000
	Accounting	Atlanta	1200000
15	Computing	Miami	1500000

No, this row cannot be inserted. It violates the uniqueness property of the key since there is a department 10 already in the table.

No, this row cannot be inserted. It violates the integrity constraint of the key since the department key cannot be NULL.

Yes, this row can be inserted with no problems since no constraint is violated.

Super Key

- The set of attributes which can uniquely identify a tuple is known as Super Key. (No minimality property)

STUDENT

STUD_NO	STUD_NAME	STUD_PHONE	STUD_STATE	STUD_COUNTRY	STUD_AGE
1	RAM	9716271721	Haryana	India	20
2	RAM	9898291281	Punjab	India	19
3	SUJIT	7898291981	Rajsthan	India	18
4	SURESH		Punjab	India	21

Table 1

STUDENT_COURSE

STUD_NO	COURSE_NO	COURSE_NAME
1	C1	DBMS
2	C2	Computer Networks
1	C2	Computer Networks

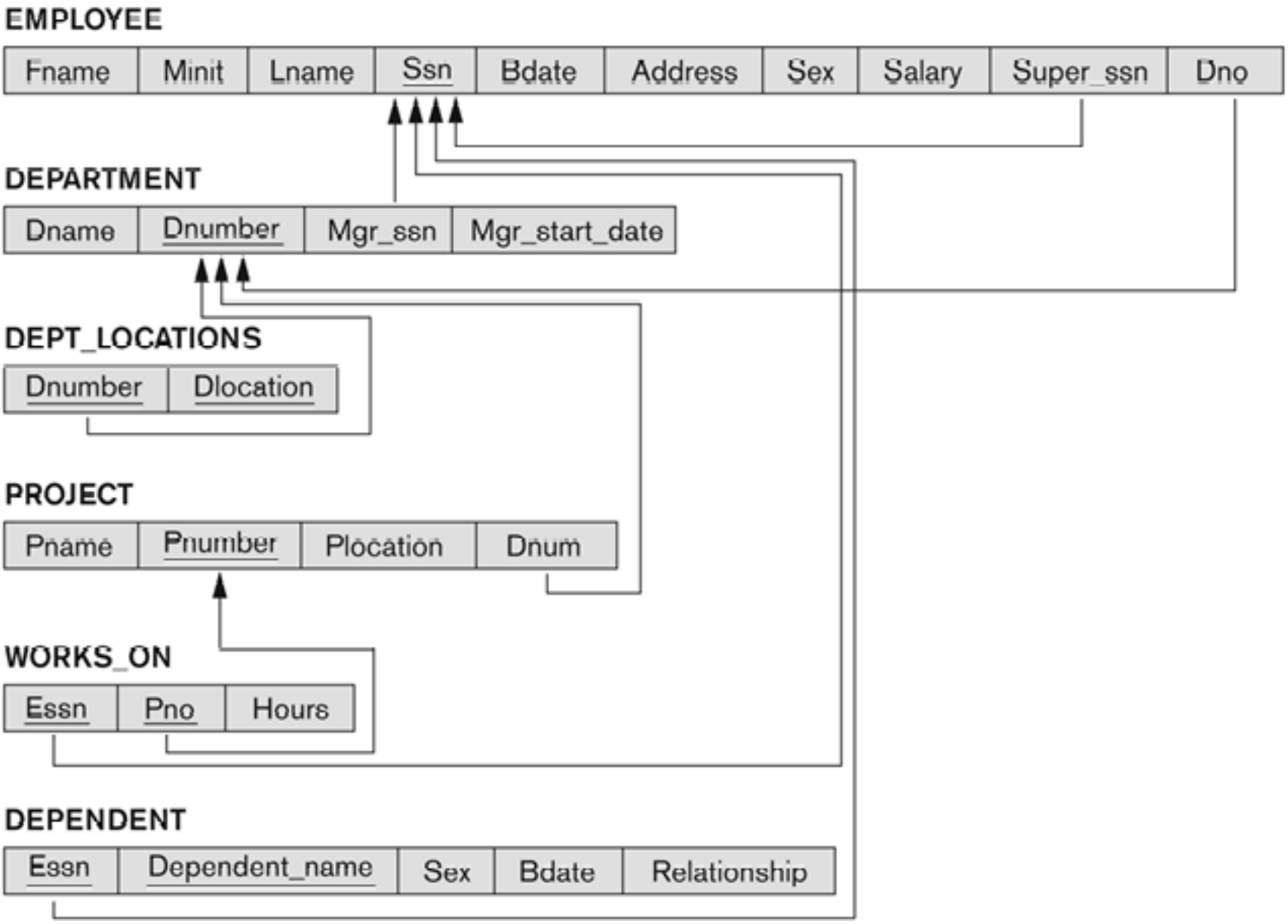
Table 2

Foreign Keys

- Given two relations R_1 and R_2 of the same database, a set of attributes *FK* of relation R_1 is said to be a *foreign key* of R_1 (with respect to R_2) if
  - The attributes of the FK have the same underlying domain as a set of attributes of a relation R_2 that have been defined as PK of R_2. The FK is said to reference the PK attribute(s) of the relation R_2.
  - The FK-values in any tuple of relation R_1 are either NULL or must appear as the PK-values of a tuple relation R_2
- The foreign key concept ensures that the tuples of relation R_1 that refers to tuples of relation R_2 must refer to tuples of R_2 that already exist.

  A tuple t_1 in R_1 is said to **reference** a tuple t_2 in R_2 if $t_1[\text{FK}] = t_1[\text{PK}]$.
- This condition imposed on foreign key is called referential integrity

Referential Integrity



THANKS