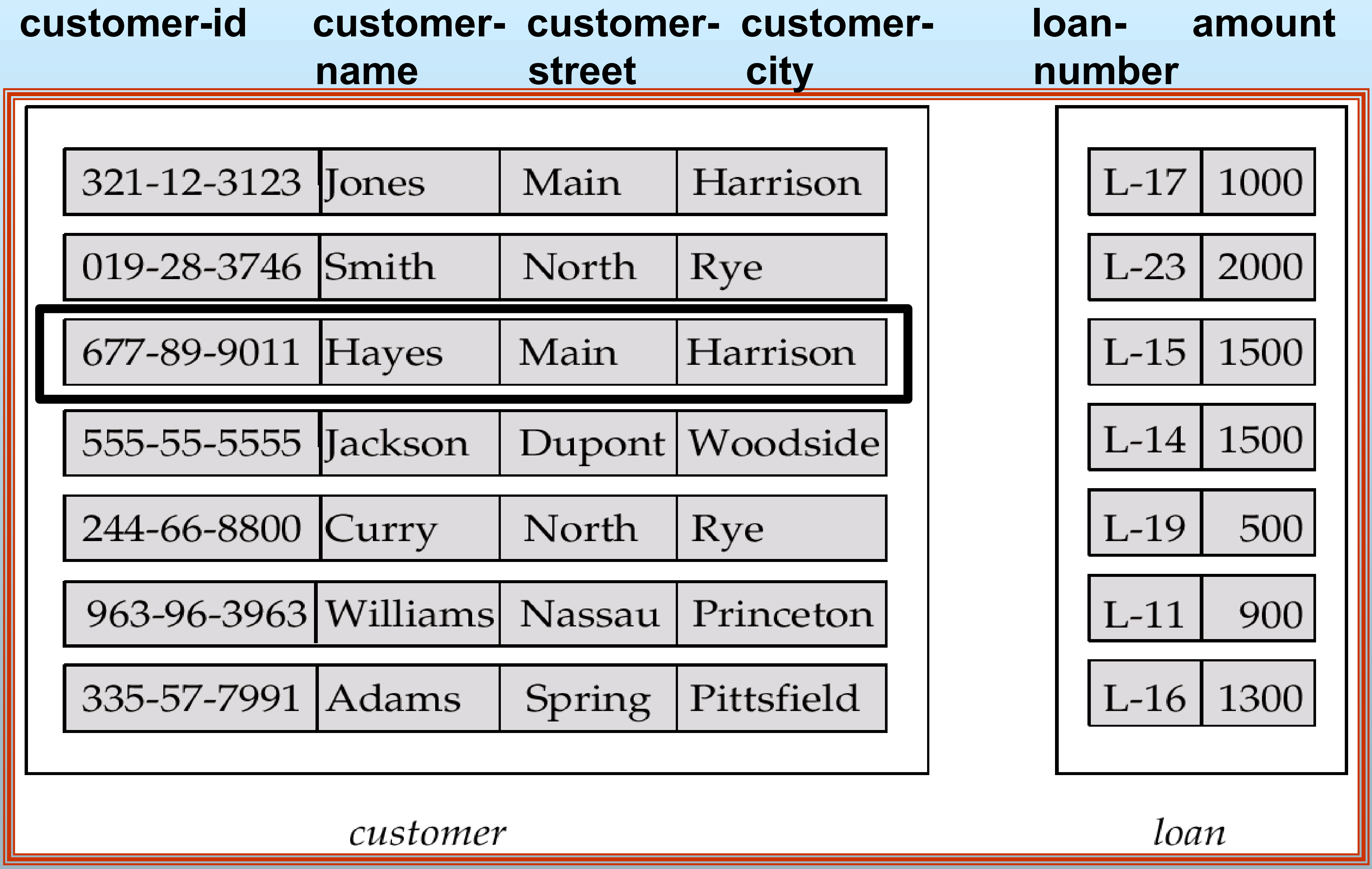
Database Management Systems

Shamim Ahmad

Entity and Attribute

Entity Sets *customer* and *loan*

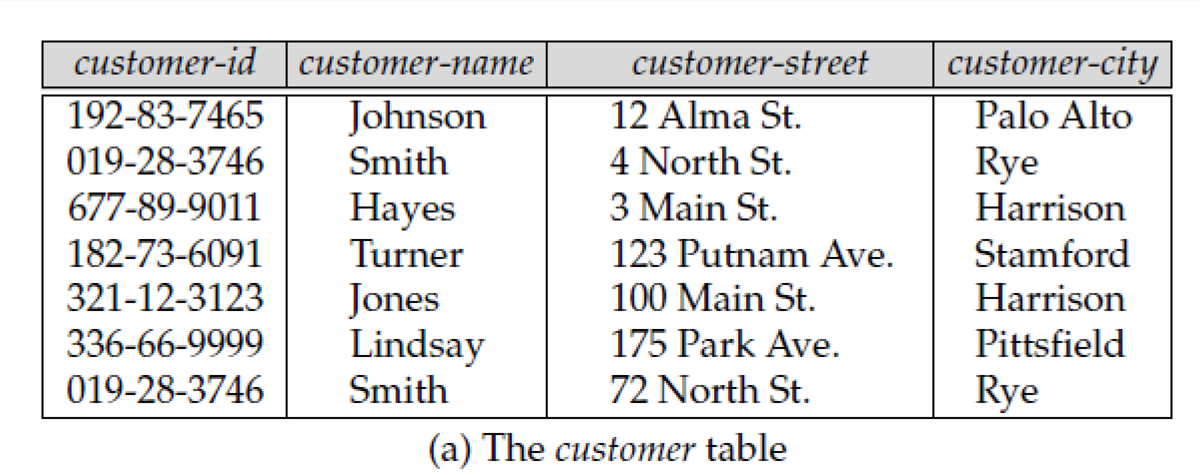


customer-id	customer-name	customer-street	customer-city
321-12-3123	Jones	Main	Harrison
019-28-3746	Smith	North	Rye
677-89-9011	Hayes	Main	Harrison
555-55-5555	Jackson	Dupont	Woodside
244-66-8800	Curry	North	Rye
963-96-3963	Williams	Nassau	Princeton
335-57-7991	Adams	Spring	Pittsfield

customer

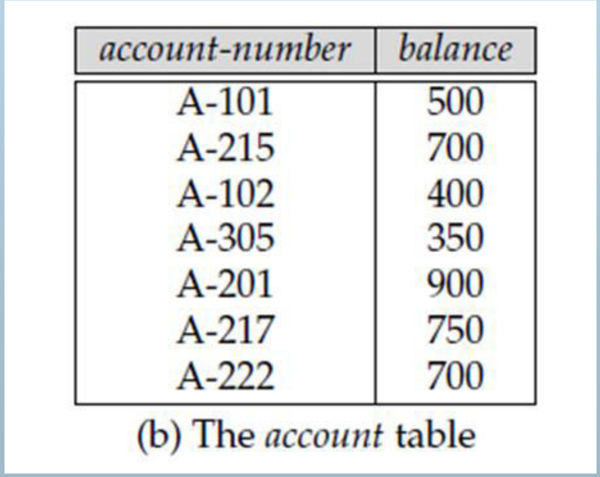
loan-number	amount
L-17	1000
L-23	2000
L-15	1500
L-14	1500
L-19	500
L-11	900
L-16	1300

loan



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

Basic Concepts (Cont.)

- Relationship Sets
  - A relationship is an association among several entities

Example:

Hayes *depositor* A-102

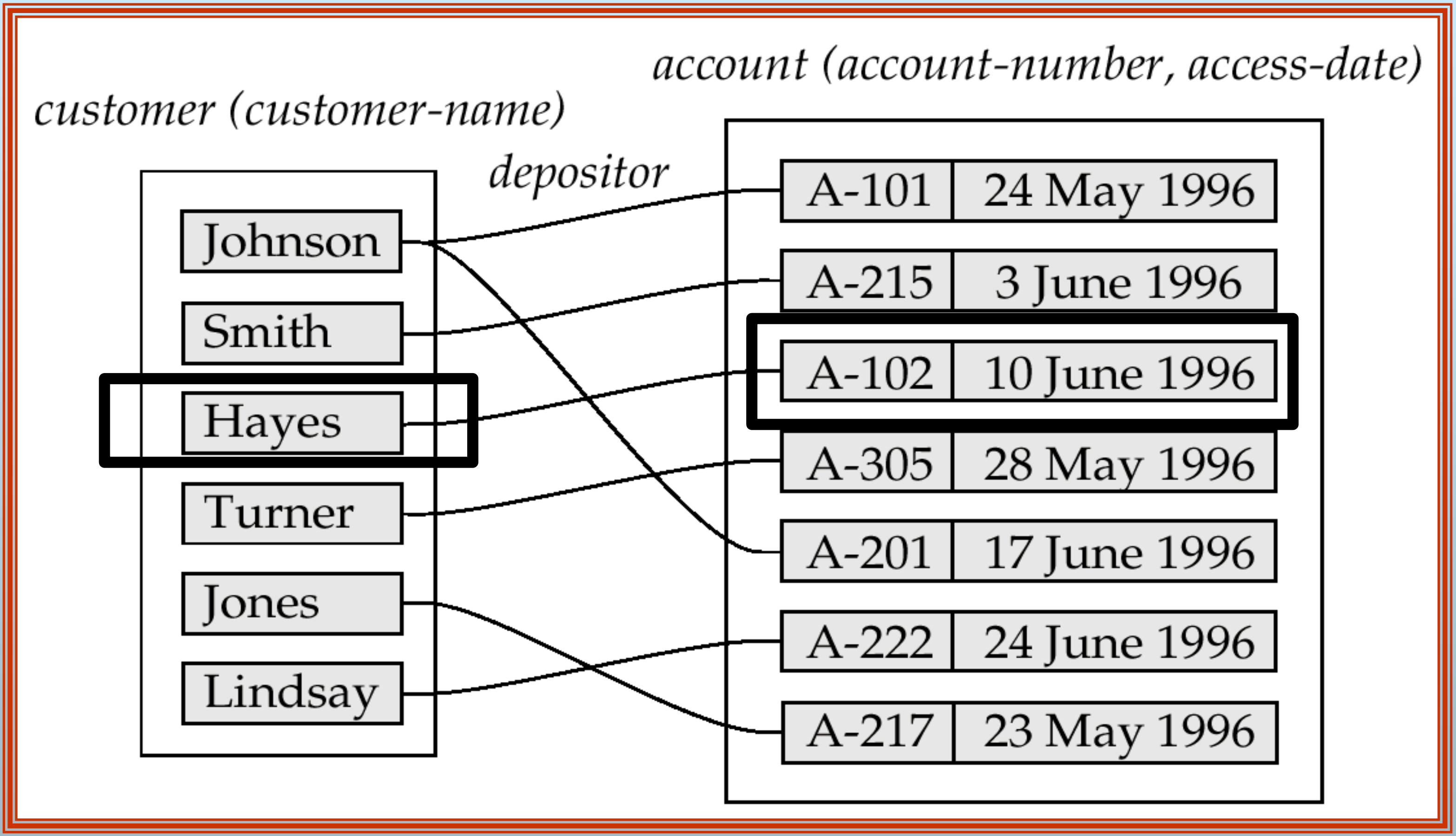
customer entity relationship set *account* entity

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

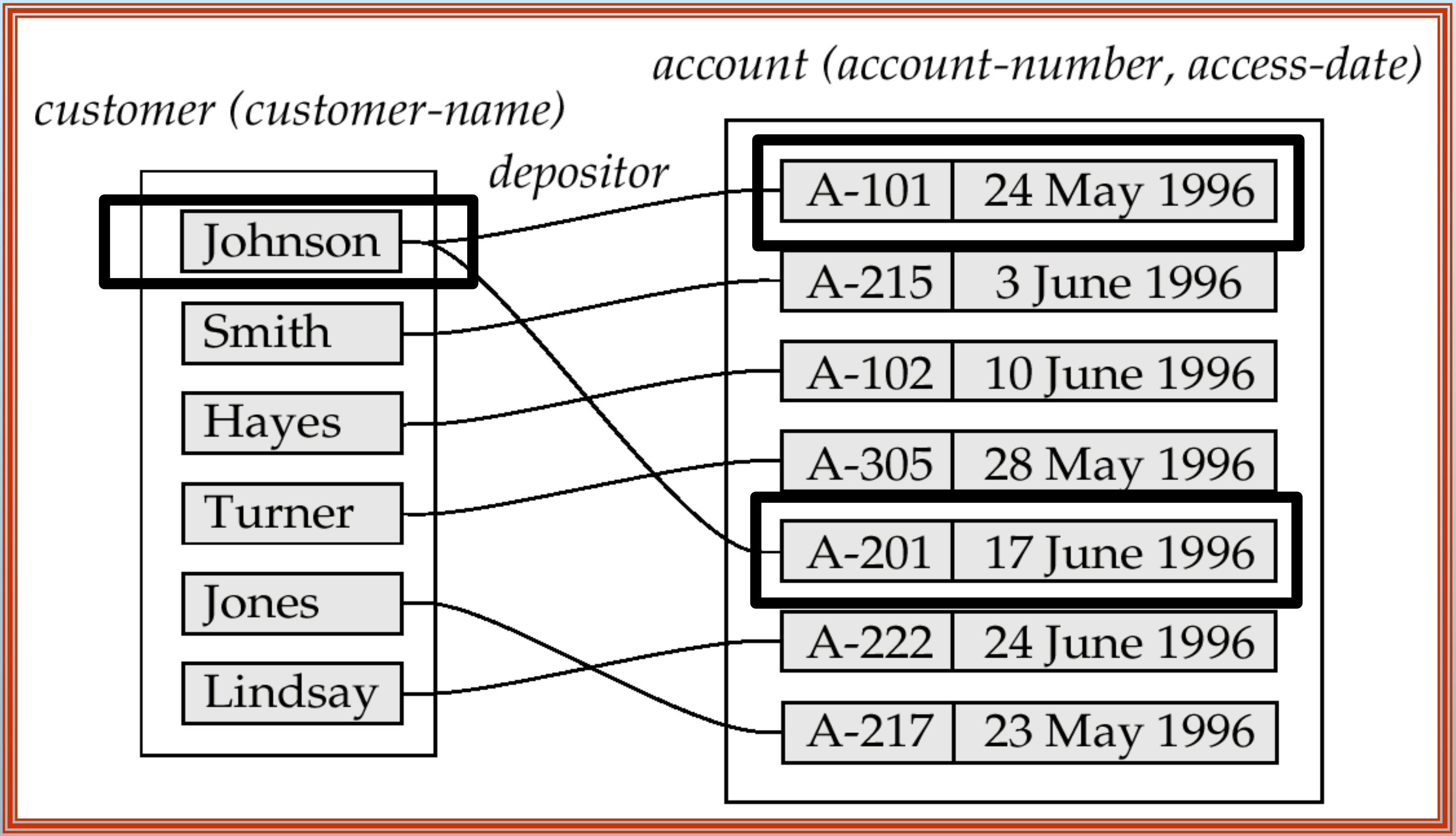
$$\{(e_1, e_2, \ldots e_n) \mid e_1 \in E_1, e_2 \in E_2, \ldots, e_n \in E_n\}$$

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

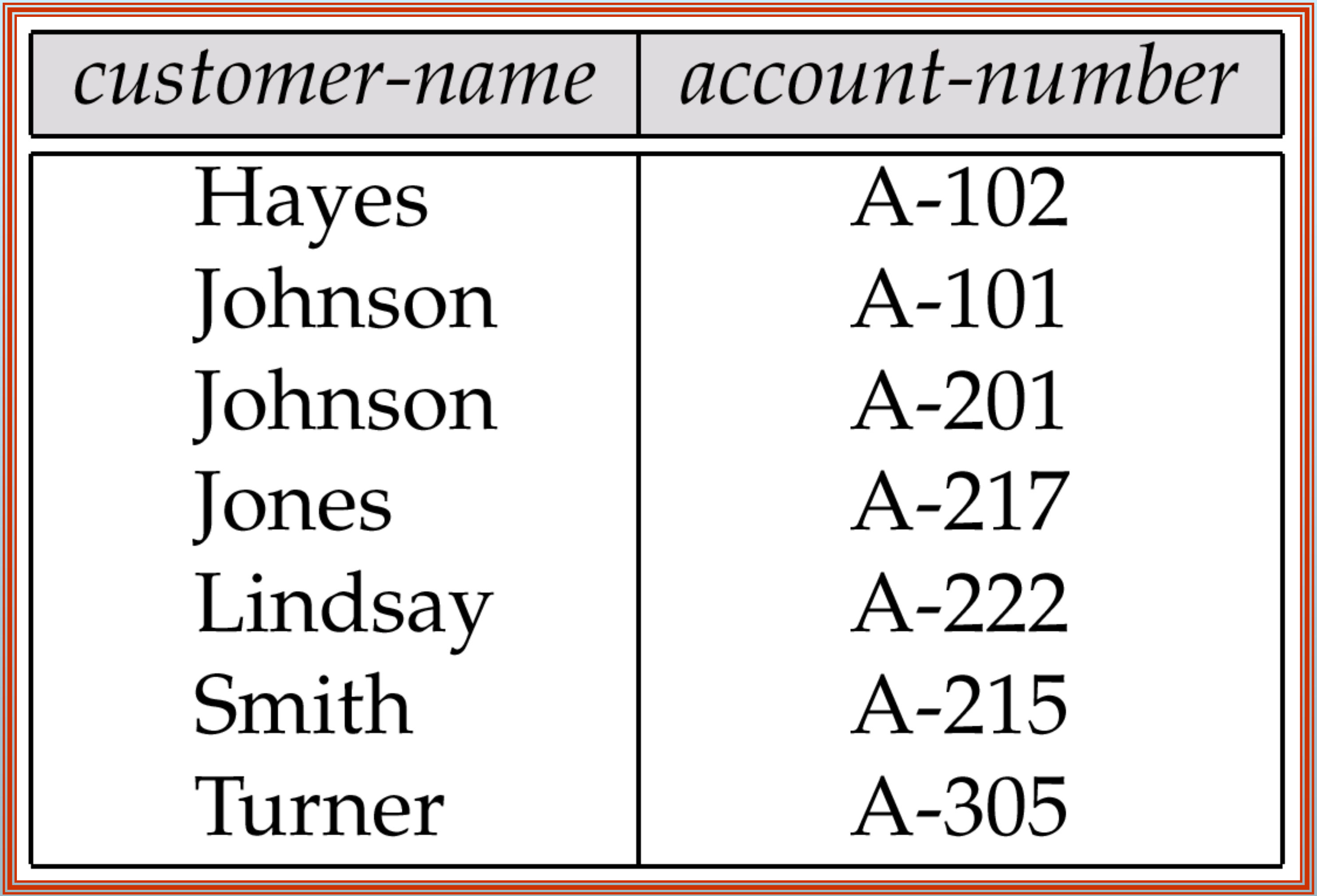
- **Example:**
  (Hayes, A-102) ∈ *depositor*



- **Example:**
  (Hayes, A-102) ∈ *depositor*

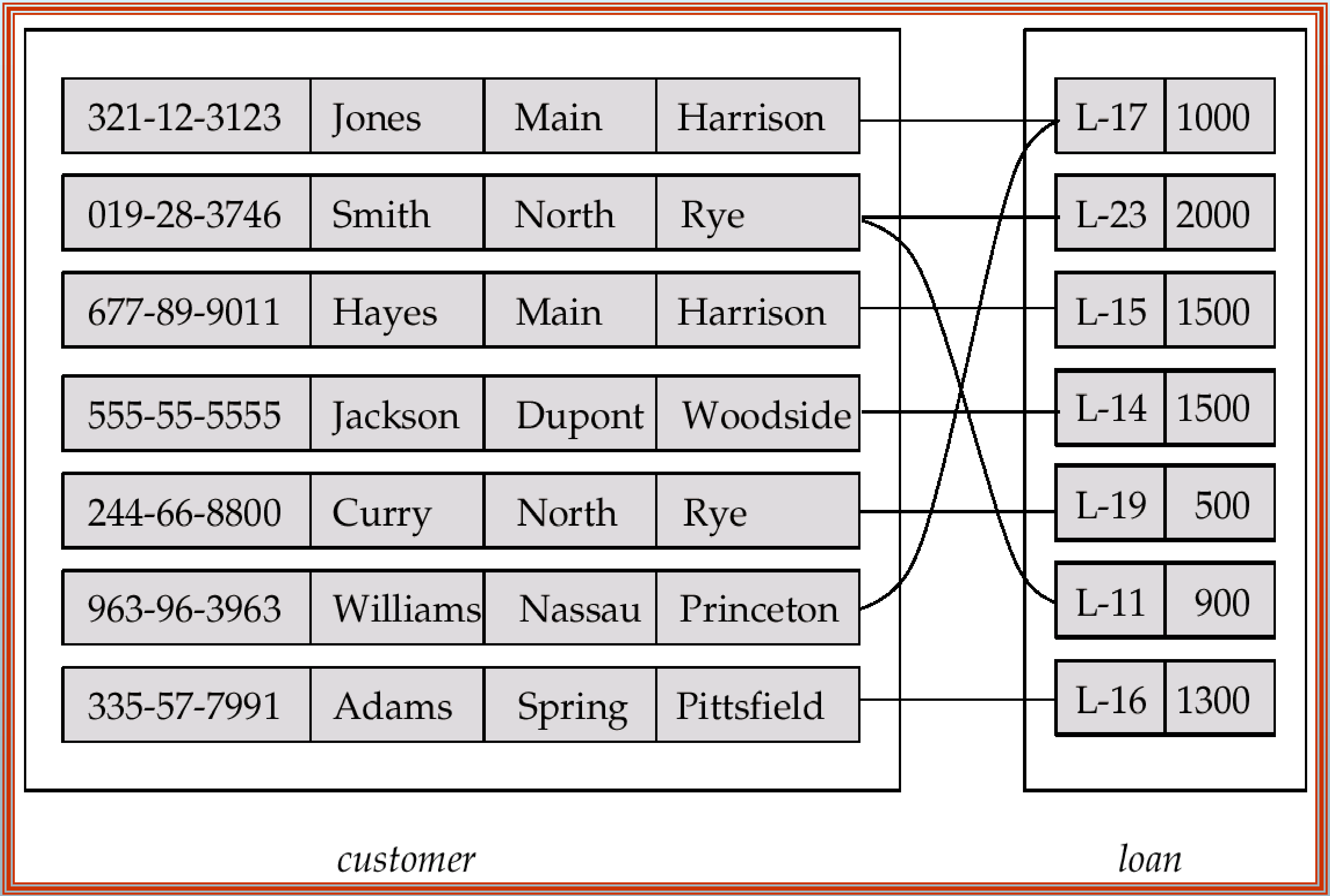


The *depositor* Relation



customer-name	*account-number*
Hayes	A-102
Johnson	A-101
Johnson	A-201
Jones	A-217
Lindsay	A-222
Smith	A-215
Turner	A-305

Relationship Set *borrower*

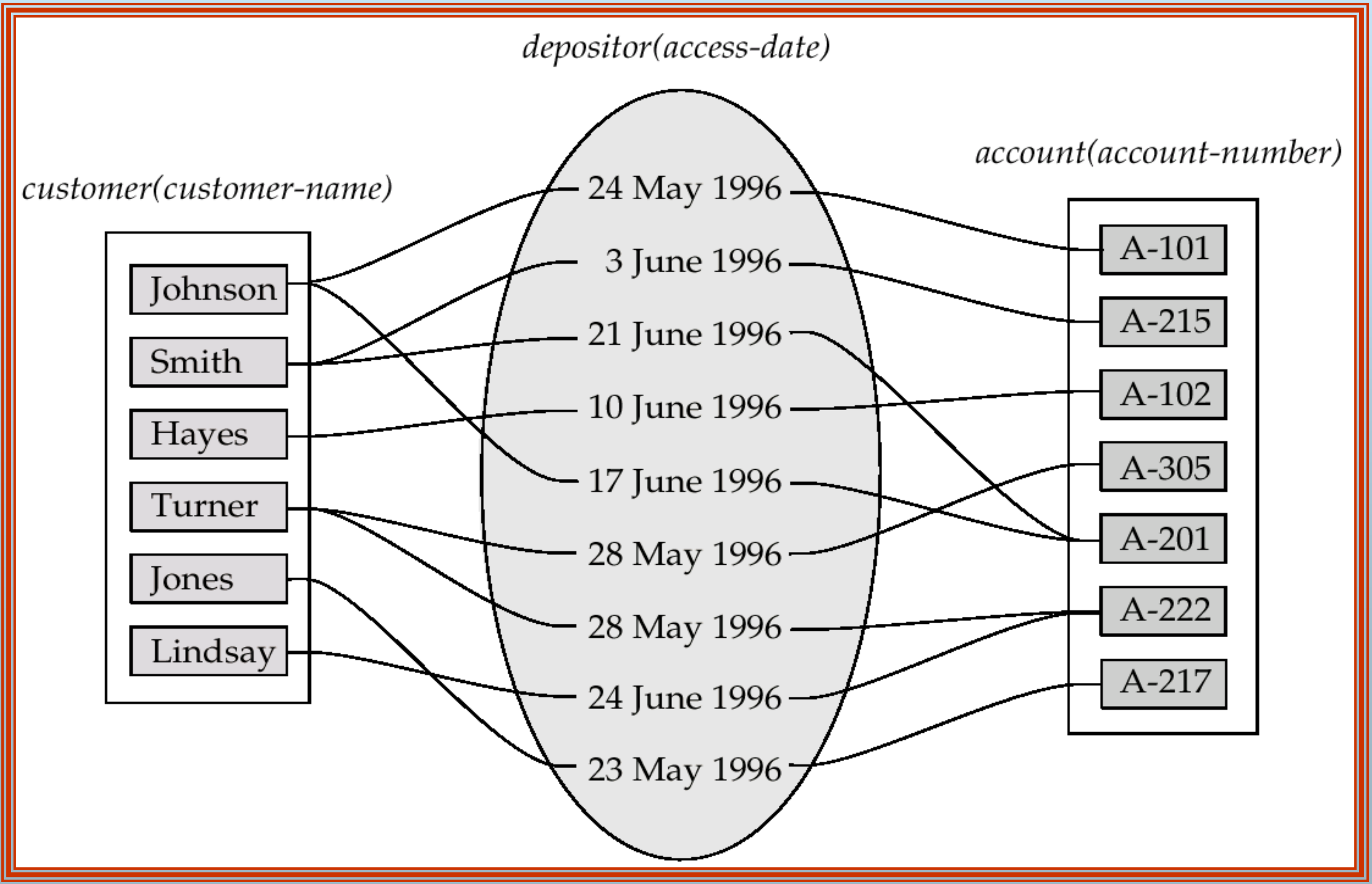


321-12-3123	Jones	Main	Harrison
019-28-3746	Smith	North	Rye
677-89-9011	Hayes	Main	Harrison
555-55-5555	Jackson	Dupont	Woodside
244-66-8800	Curry	North	Rye
963-96-3963	Williams	Nassau	Princeton
335-57-7991	Adams	Spring	Pittsfield

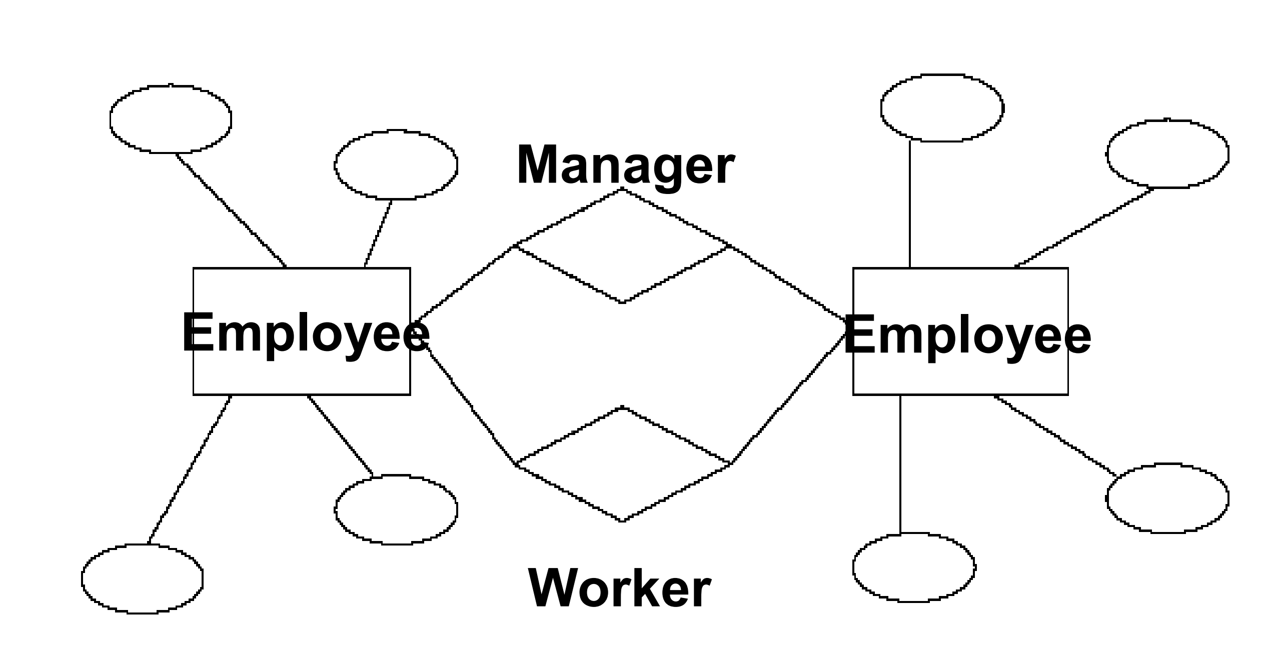
L-17	1000
L-23	2000
L-15	1500
L-14	1500
L-19	500
L-11	900
L-16	1300

Basic Concepts(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***



Role: Recursive Relationship



A relationship Set

works-for

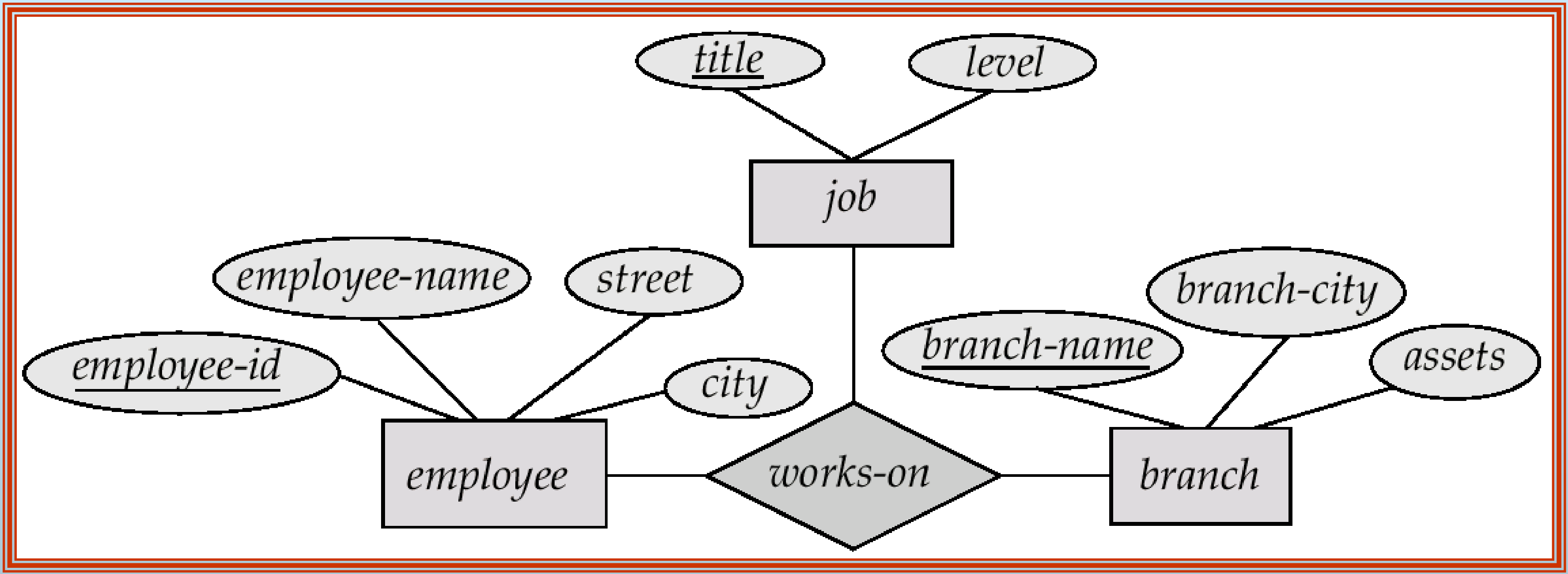
that is modeled by

ordered pairs of **employee entities**

Basic Concepts(Cont.)

- **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.
    - E.g. Suppose **employees** of a bank may have **jobs** (responsibilities) at multiple branches, **with different jobs at different branches.** Then there is **a ternary** relationship set between entity sets *employee, job and branch*
  - Relationships between more than two entity sets are rare. Most relationships are binary. (More on this later.)

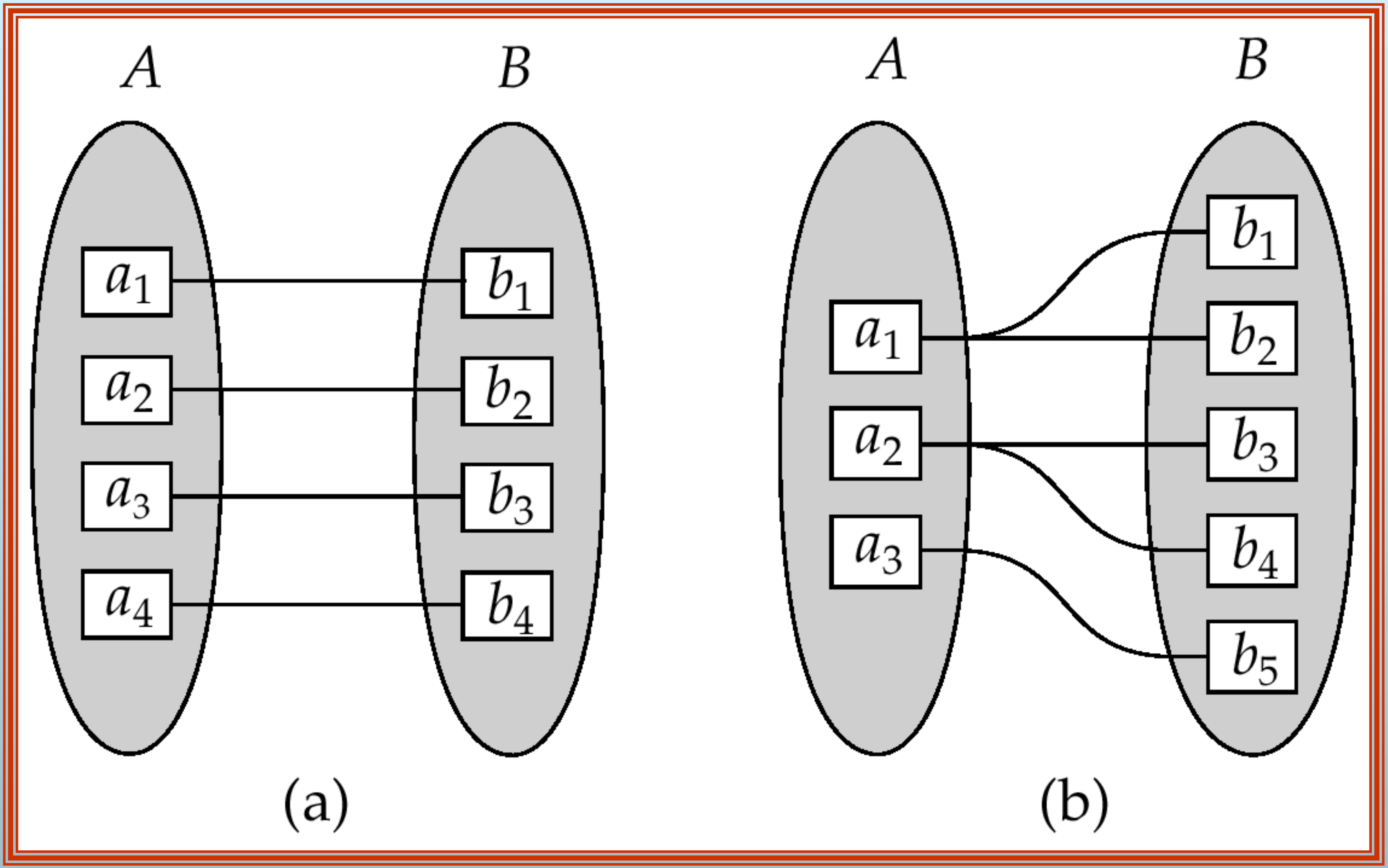
E-R Diagram with a Ternary Relationship



Constraints

- **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

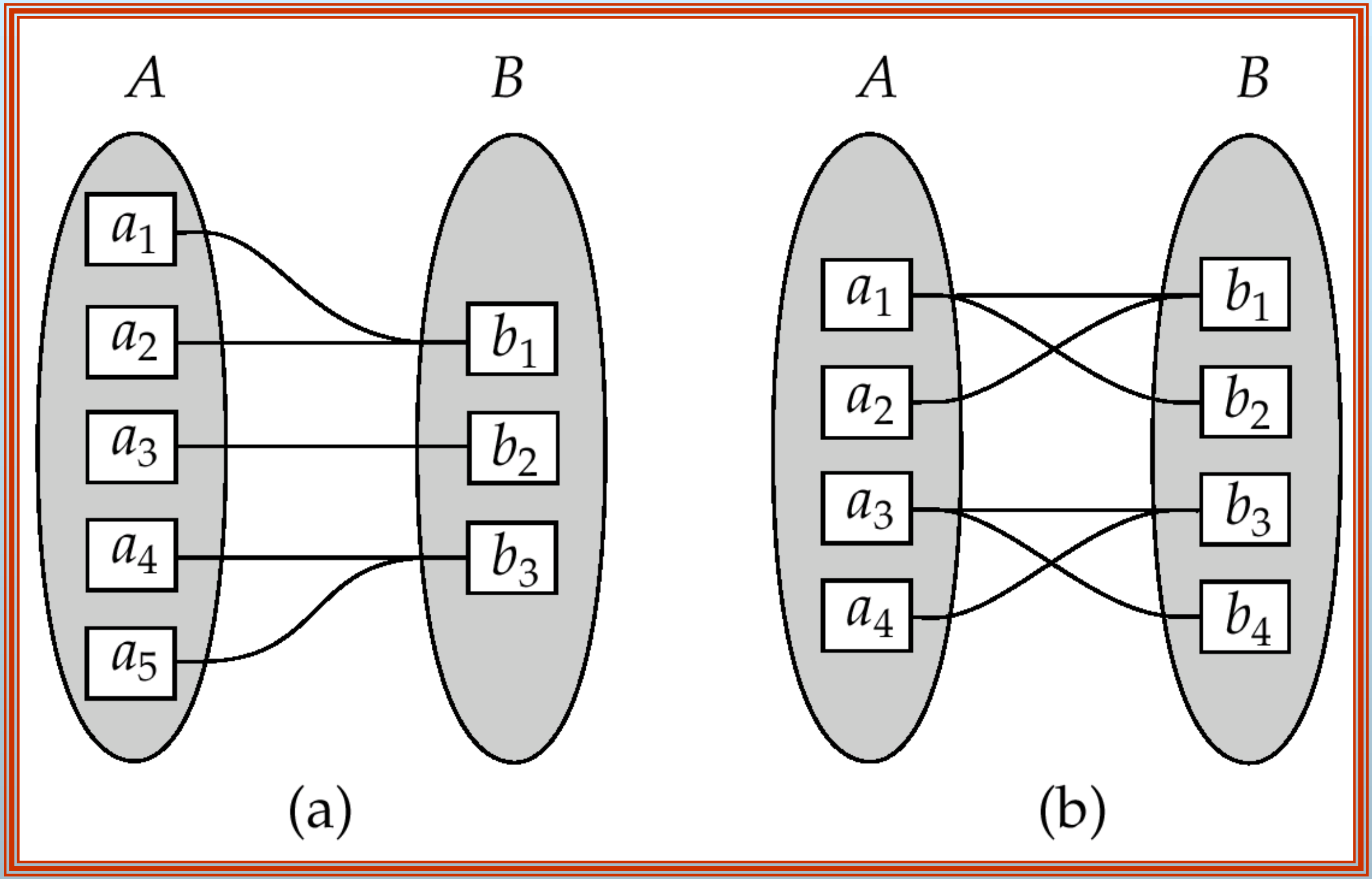
Constraints (Cont.)



One to one One to many

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

Constraints (Cont.)



Many to one Many to many

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

Constraints (Cont.)

- **Participation Constraints**
  - The participation of an **entity set *E*** in **a relationship set *R*** is said to be total if every **entity in *E*** participates in **at least one relationship in *R*.**
    - For example, we expect every loan entity to be related to at least one customer through the borrower relationship.

চিহ্নতকরণকে

  - If only **some entities in E** participate in relationship in R, the participation of **entity set E** in **relationship R** is said to be **partial.**
    - For example, the participation of customer in the borrower relationship set is therefore set is therefore partial.

Keys

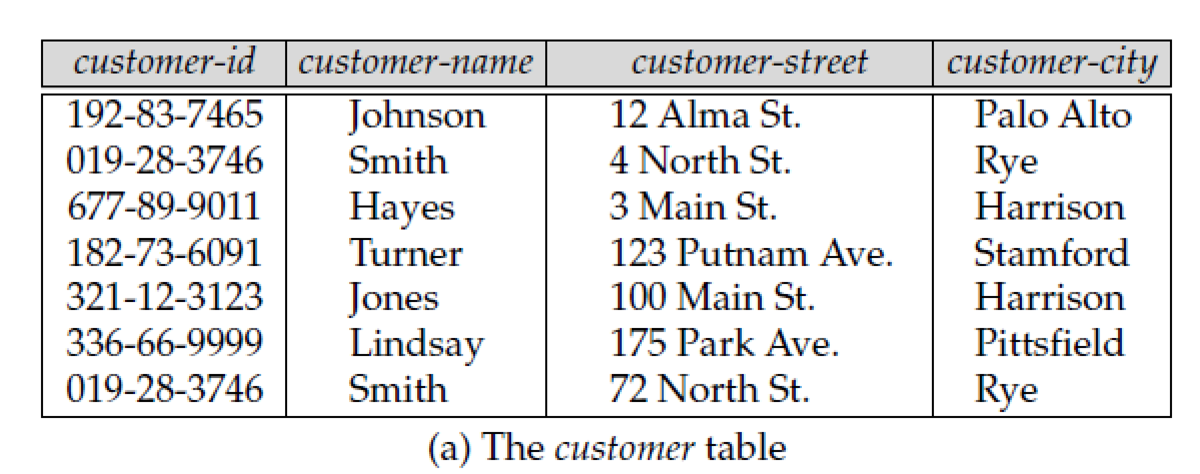
- **Entity Sets**
  - A ***super key*** of an entity set is a set of one or more attributes whose values uniquely determine each entity.
  - A ***candidate key*** of an entity set is a minimal super key
    - *Customer-id* is candidate key of *customer*
    - *account-number* is candidate key of *account*
  - Although several candidate keys may exist, one of the candidate keys is selected to be the *primary key*.

- **Relationship Sets**
  - **The combination of primary keys of the participating entity sets forms a super key of a relationship set.**

$$\textit{primary-key}(E_1) \cup \textit{primary-key}(E_2) \cup \cdots \cup \textit{primary-key}(E_n)$$

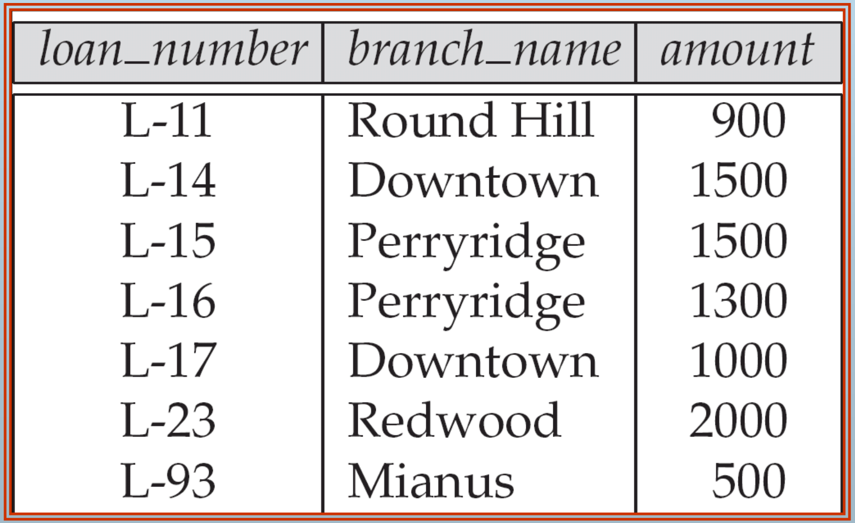
describes an individual relationship in set *R*.

$$\textit{primary-key}(E1) \cup \textit{primary-key}(E2) \cup \cdots \cup \textit{primary-key}(En) \cup \{a1, a2, \ldots, am\}$$



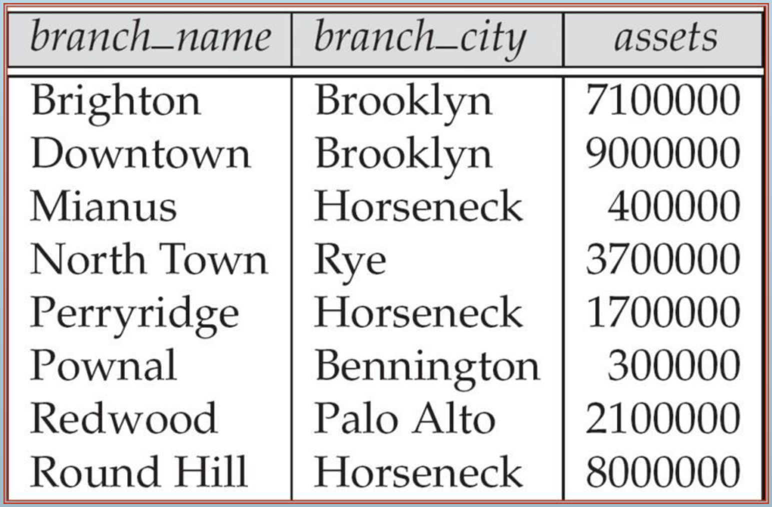
customer-id	*customer-name*	*customer-street*	*customer-city*
192-83-7465	Johnson	12 Alma St.	Palo Alto
019-28-3746	Smith	4 North St.	Rye
677-89-9011	Hayes	3 Main St.	Harrison
182-73-6091	Turner	123 Putnam Ave.	Stamford
321-12-3123	Jones	100 Main St.	Harrison
336-66-9999	Lindsay	175 Park Ave.	Pittsfield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

(a) The *customer* table



loan_number	*branch_name*	*amount*
L-11	Round Hill	900
L-14	Downtown	1500
L-15	Perryridge	1500
L-16	Perryridge	1300
L-17	Downtown	1000
L-23	Redwood	2000
L-93	Mianus	500

The loan relation

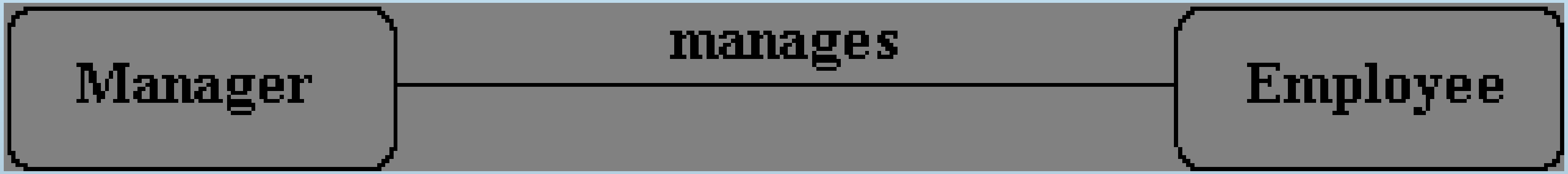


branch_name	*branch_city*	*assets*
Brighton	Brooklyn	7100000
Downtown	Brooklyn	9000000
Mianus	Horseneck	400000
North Town	Rye	3700000
Perryridge	Horseneck	1700000
Pownal	Bennington	300000
Redwood	Palo Alto	2100000
Round Hill	Horseneck	8000000

The *branch* relation

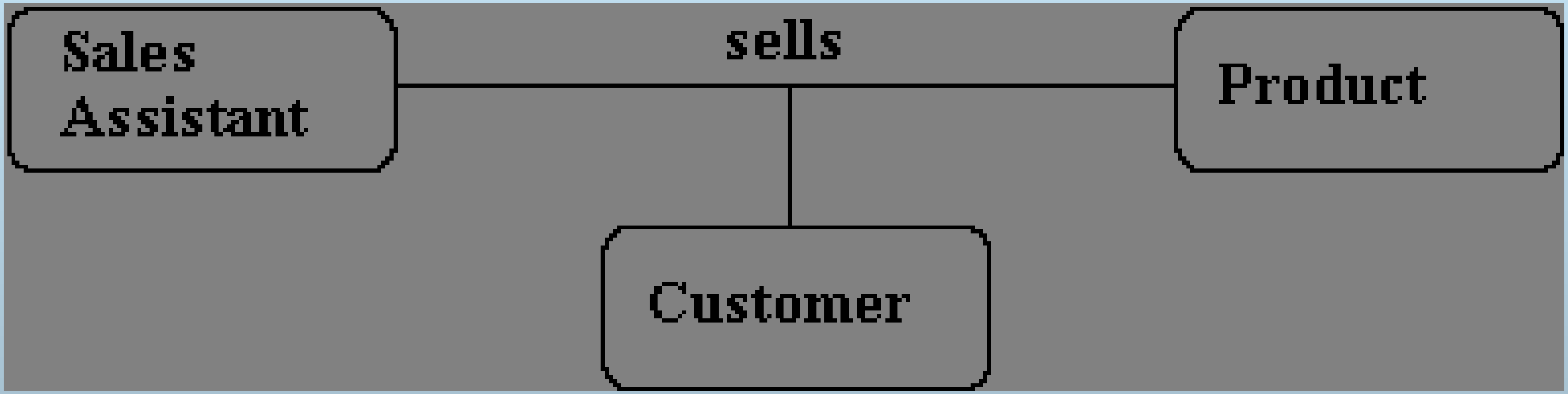
Binary Relationships

- If there are two entity types involved it is a *binary* relationship type



Ternary relationship

- If there are three entity types involved it is a *ternary* relationship type

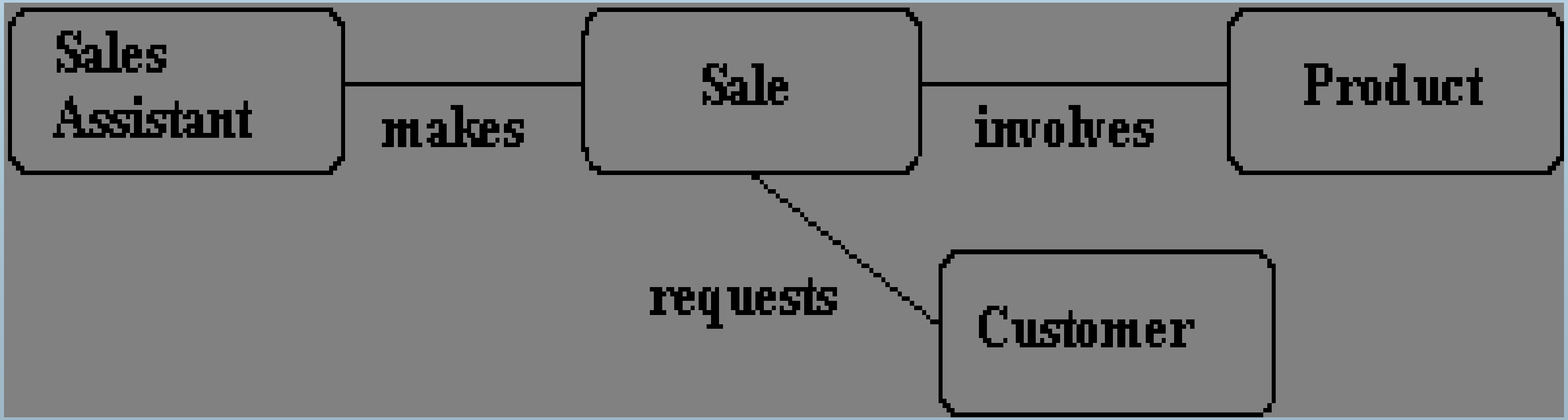


Replacing ternary relationships

- When ternary relationships occurs in an ER model they should always be removed before finishing the model.
- Sometimes the relationships can be replaced by a series of **binary relationships that link pairs** of the original ternary relationship.

Replacing ternary relationships (Cont)

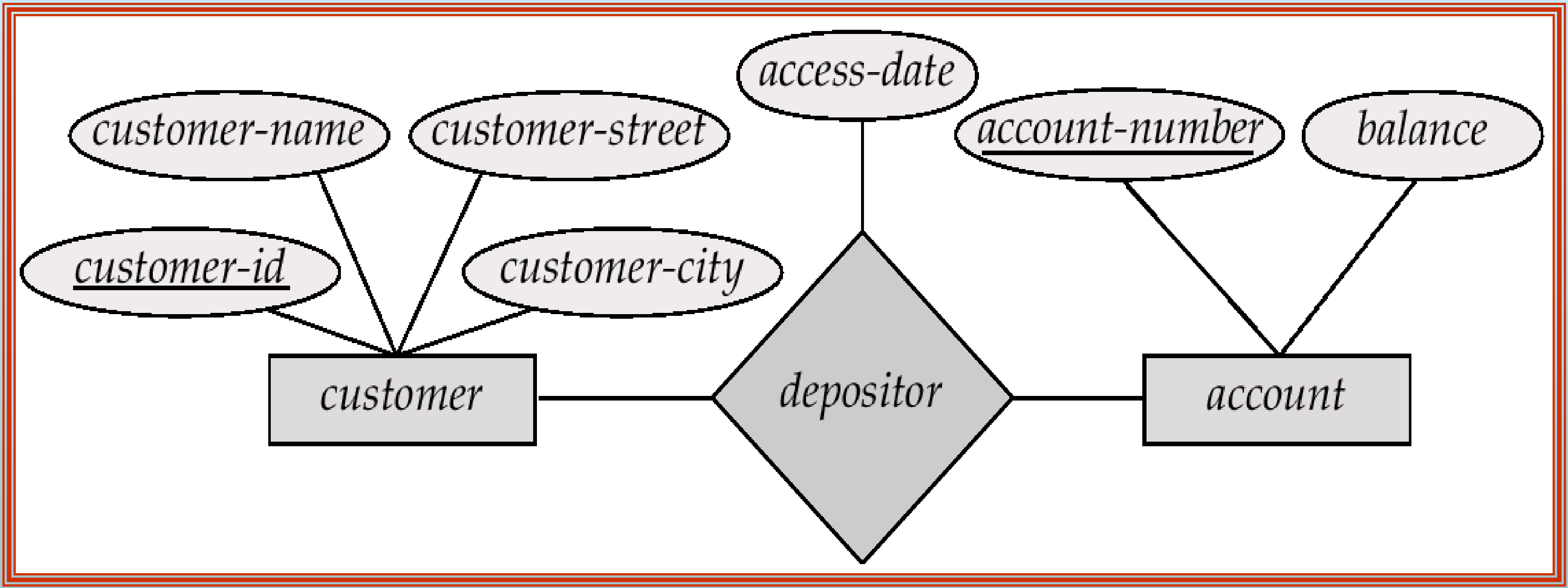
- Relationships are usually verbs, so name the new entity type by the relationship verb rewritten as a noun.
- The relationship *sells* can become the entity type *sale*.
- So a sales assistant can be linked to a specific customer and both of them to the sale of a particular product.
- This process also works for higher order relationships.



E-R Diagrams

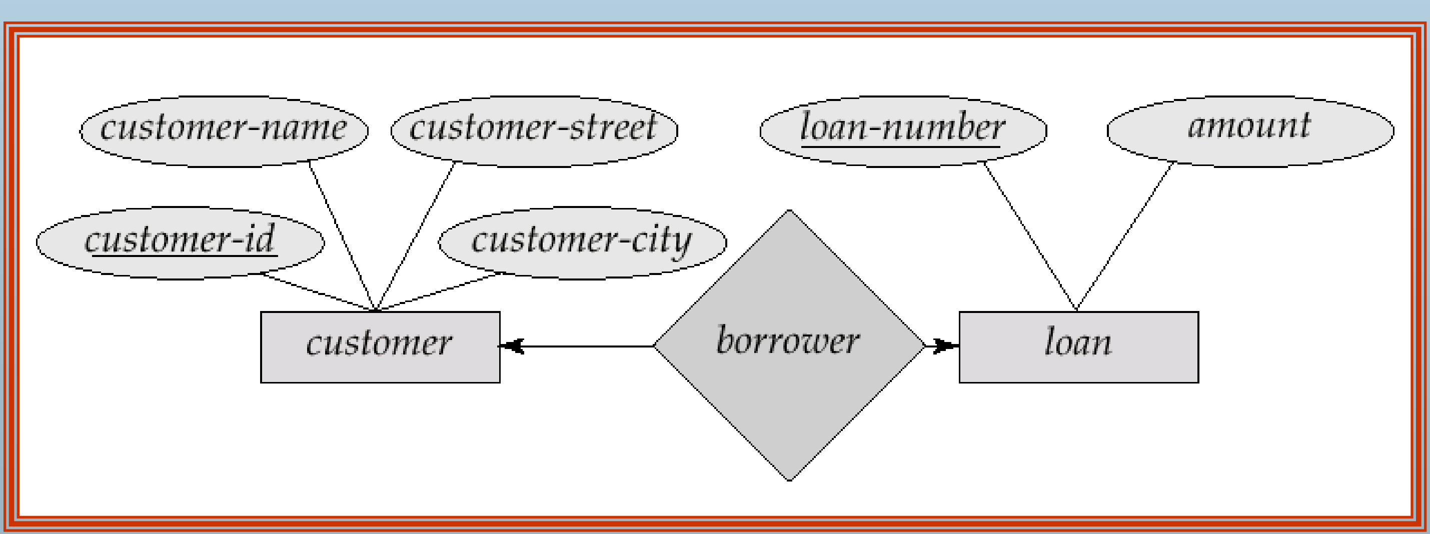
- **Rectangles** represent entity sets.
- **Diamonds** represent relationship sets.
- **Lines** link attributes to entity sets and entity sets to relationship sets.
- **Ellipses** represent attributes
  - **Double ellipses** represent multivalued attributes.
  - **Dashed ellipses** denote derived attributes.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

Relationship Sets with Attributes



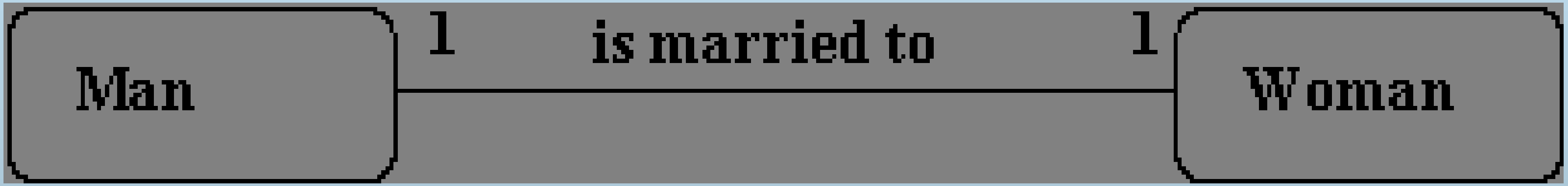
Cardinality Constraints

- 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*



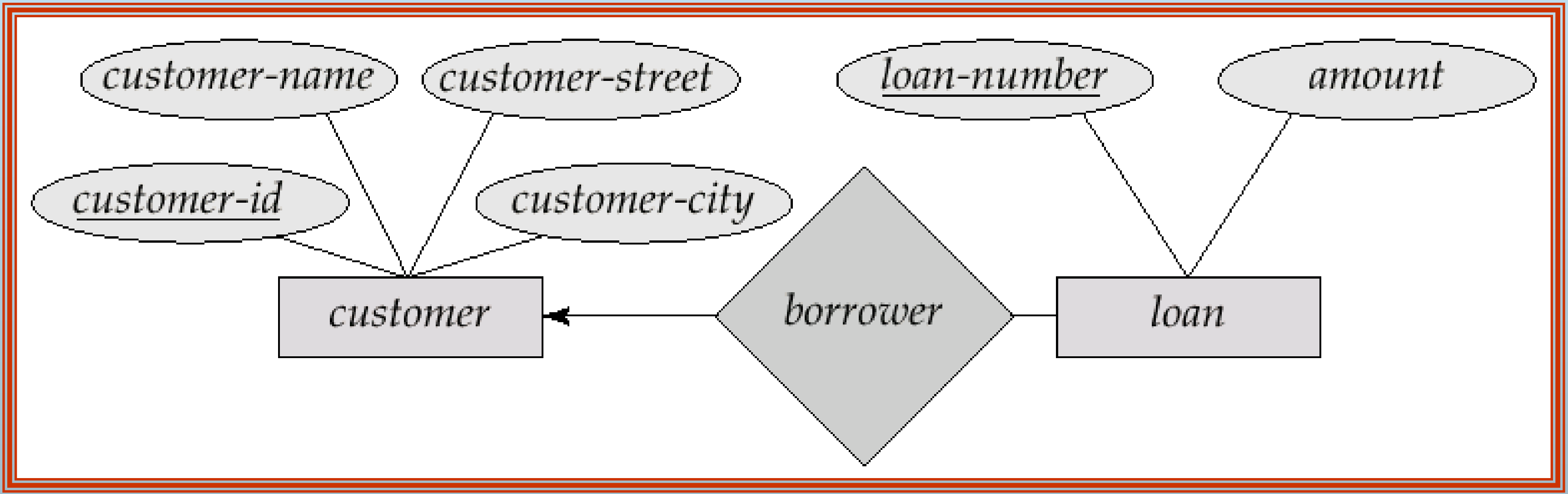
One-To-One Relationship

- A man can only marry one woman, and a woman can only marry one man, so it is a one to one (1:1) relationship



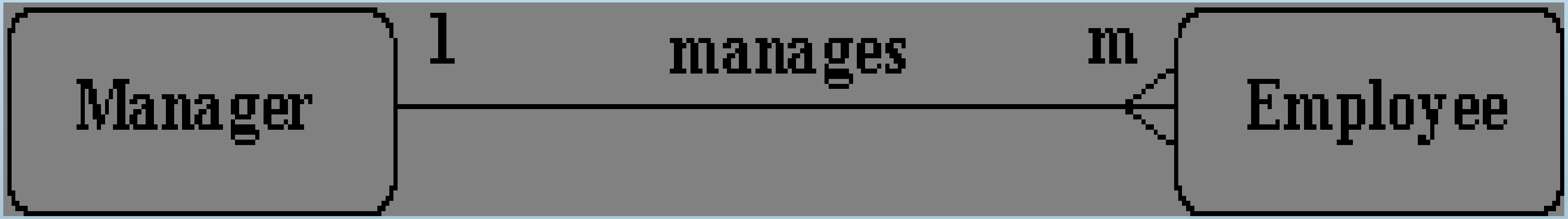
One-To-Many Relationship

- 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*



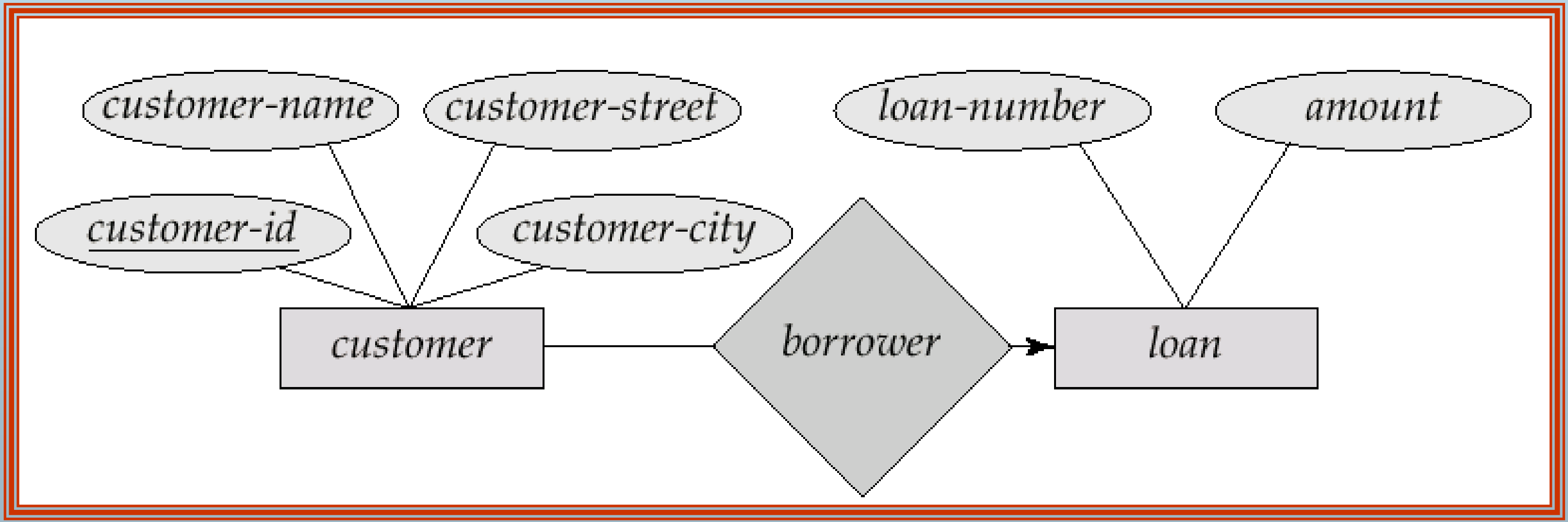
One-To-Many Relationship

- One manager manages many employees, but each employee only has one manager, so it is a one to many (1:n) relationship



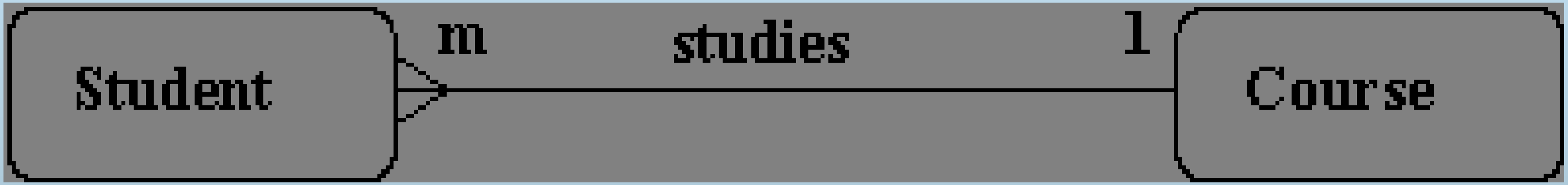
Many-To-One Relationships

- 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 lo**an via *borrower*

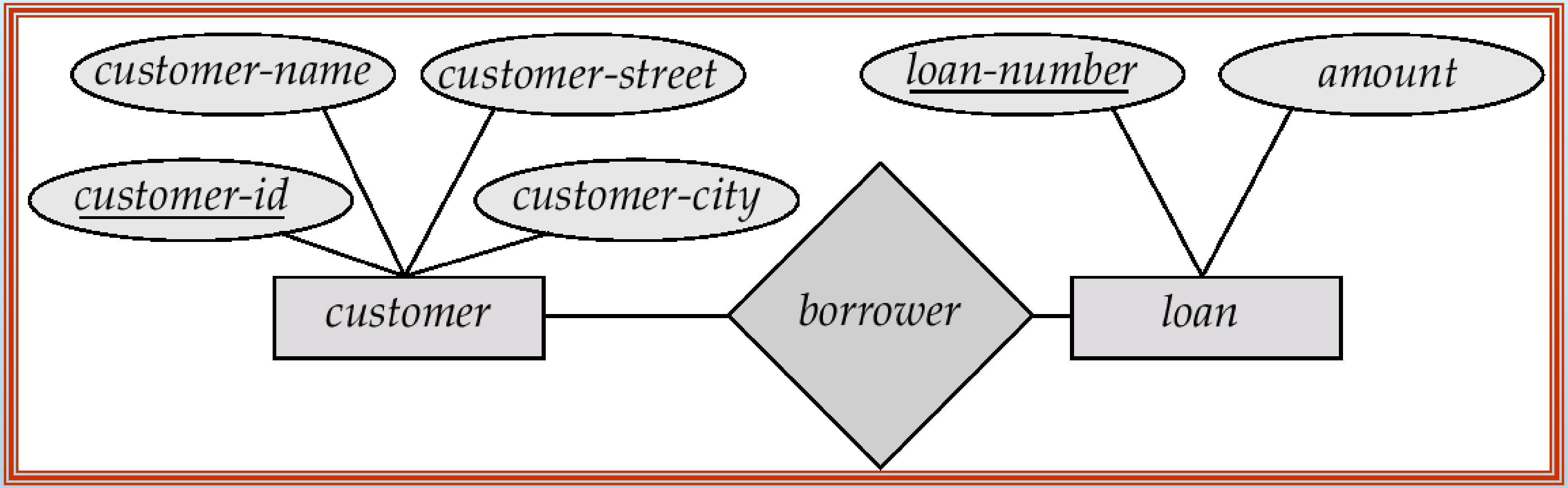


Many-To-One Relationships

- many students study one course. They do not study more than one course, so it is a many to one (m:1) relationship



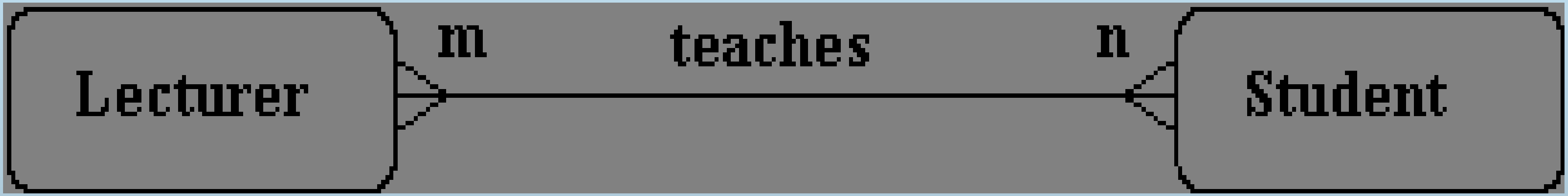
Many-To-Many Relationship



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

Many-To-Many Relationship

- One lecturer teaches many students and a student is taught by many lecturers, so it is a many to many (m:n) relationship



Deriving the relationship parameters

- To check we have the correct parameters (sometimes also known as the **degree**) of a **relationship**, ask two questions:
- **One course** is studied by **how many** students? **Answer = `zero or more'**
  - This gives us the **degree at the `student' end**
  - **The `more'** part means that the cardinality **is `many**
  - The `**zero'** part means that the relationship is `**optional'**. (denoted y ‘O”
  - If the answer was `**one or more'**, then the relationship would be `**mandatory'.**
- **One student** studies **how many** courses? **Answer = `One'**
  - This gives us **the degree at the `course' end** of the relationship**.**
  - The answer `**one'** means that the cardinality of this relationship is 1, and is `**mandatory'**