

Intro to Entity-Relationship Model

Chapter 6: Entity-Relationship Model

- Design Process
- Modeling
- Constraints
- E-R Diagram
- Design Issues
- Weak Entity Sets
- Extended E-R Features
- Design of the Bank Database
- Reduction to Relation Schemas
- Database Design
- UML

Modeling

- 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

Entity Sets *customer* and *loan*

customer_id customer_ customer_ customer_ loan_
amount

customer_id	customer_name	customer_street	customer_city	loan_id	loan_amount
321-12-3123	Jones	Main	Harrison	L-17	1000
019-28-3746	Smith	North	Rye	L-23	2000
677-89-9011	Hayes	Main	Harrison	L-15	1500
555-55-5555	Jackson	Dupont	Woodside	L-14	1500
244-66-8800	Curry	North	Rye	L-19	500
963-96-3963	Williams	Nassau	Princeton	L-11	900
335-57-7991	Adams	Spring	Pittsfield	L-16	1300
<i>customer</i>				<i>loan</i>	

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

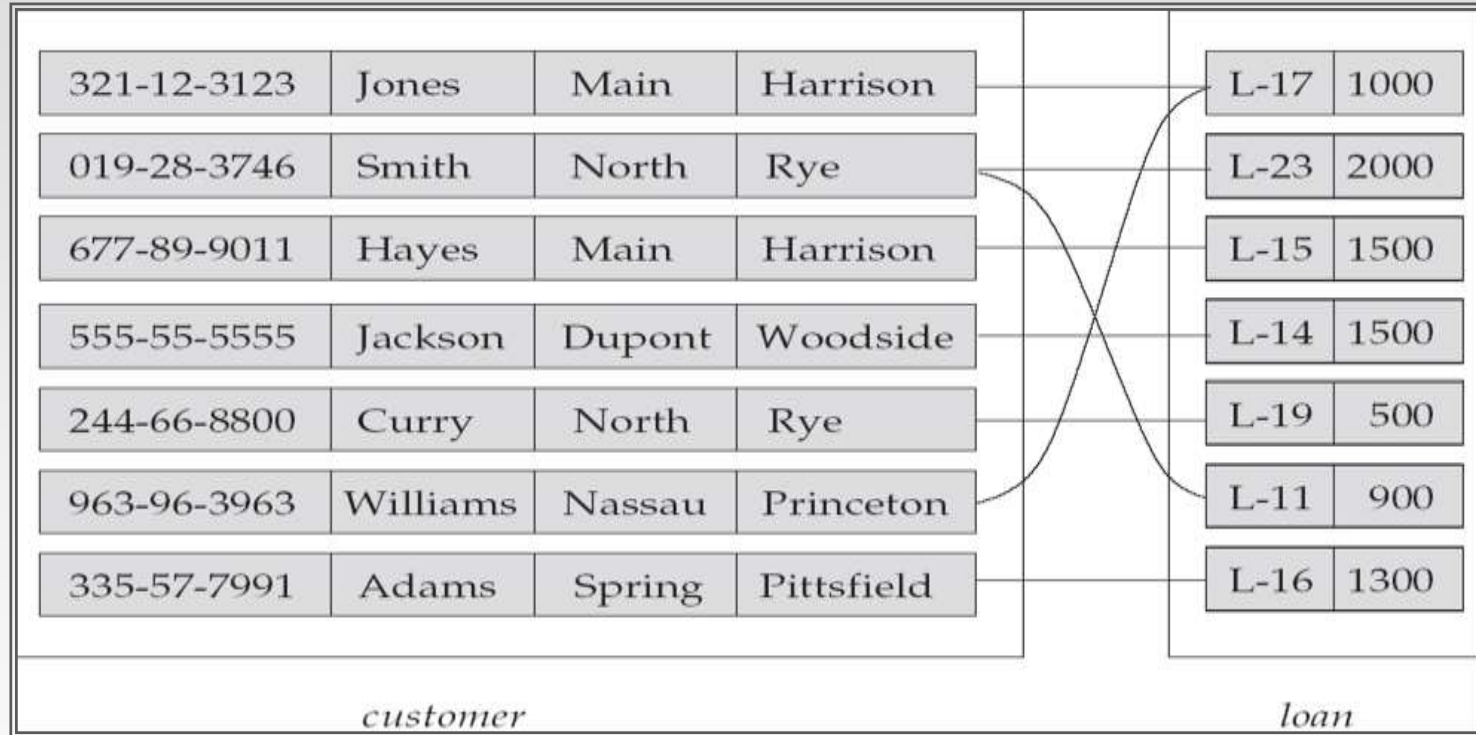
$$\{(e_1, e_2, \dots, e_n) \mid e_1 \in E_1, e_2 \in E_2, \dots, e_n \in E_n\}$$

where (e_1, e_2, \dots, e_n) is a relationship

- Example:

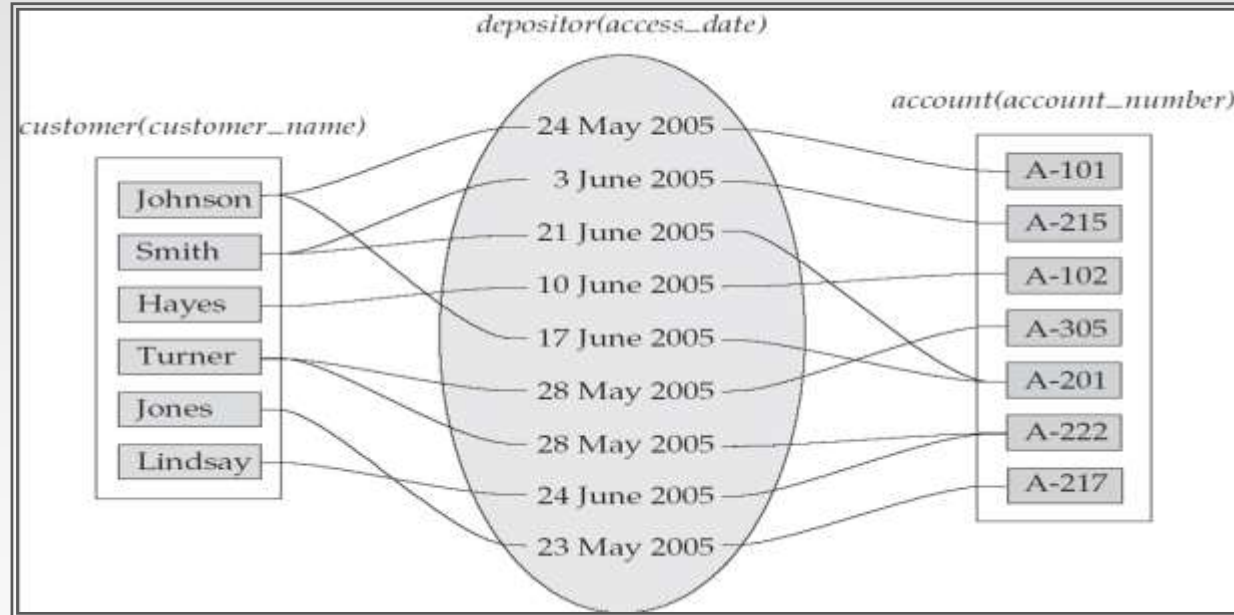
$(\text{Hayes}, \text{A-102}) \in \text{depositor}$

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.

Example: 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.)

Attributes

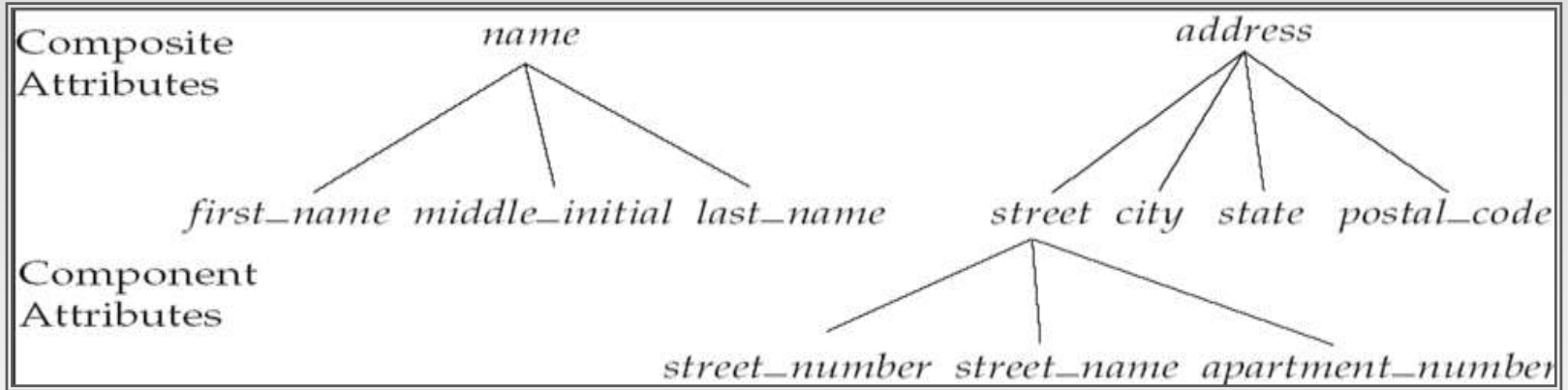
- An entity is represented by a set of attributes, that is descriptive properties possessed by all members of an entity set.

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
 - 4 Example: multivalued attribute: *phone_numbers*
 - *Derived* attributes
 - 4 Can be computed from other attributes
 - 4 Example: age, given date_of_birth

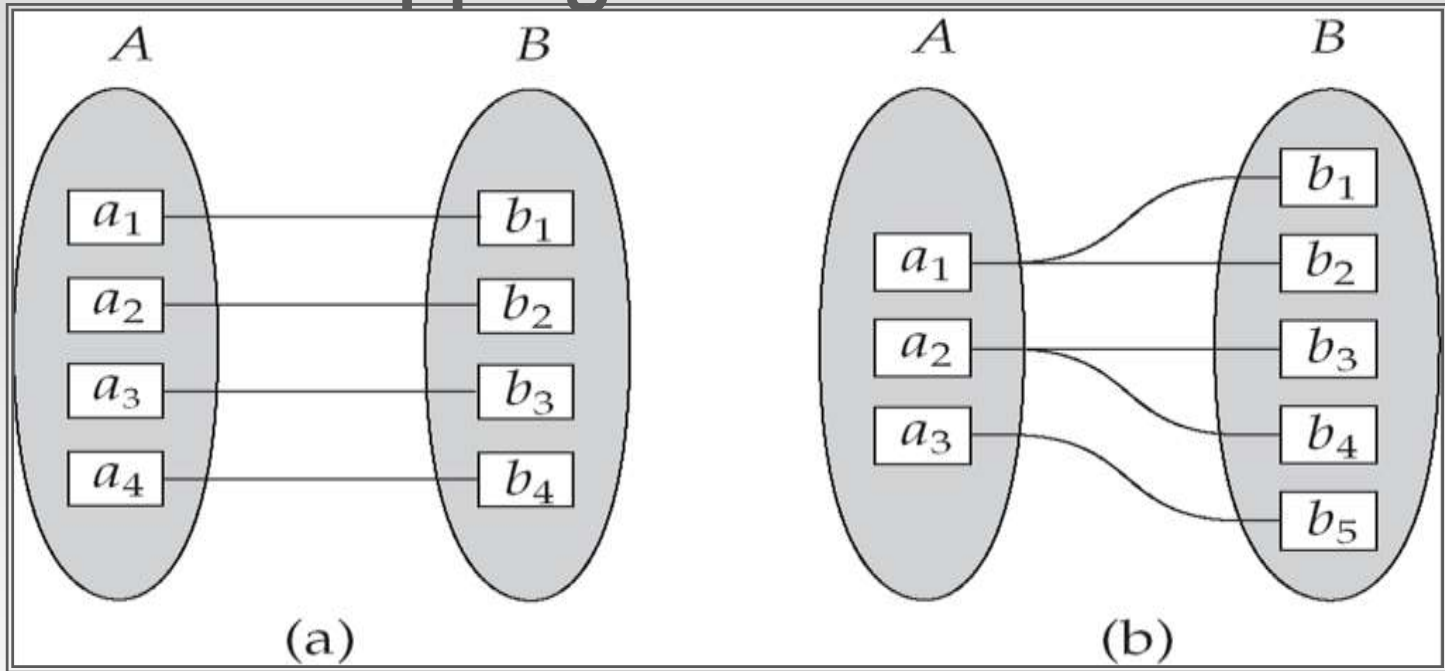
Composite Attributes



Mapping Cardinality Constraints

- 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

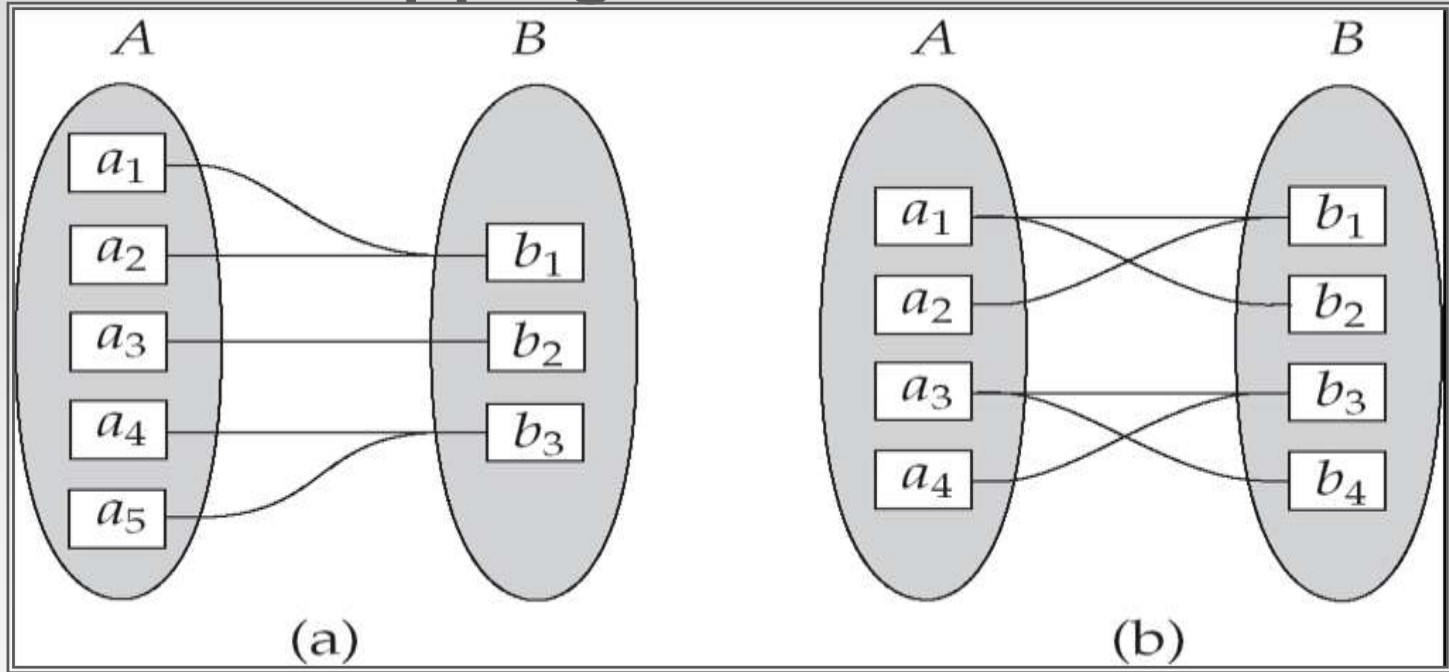


One to one

One to

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

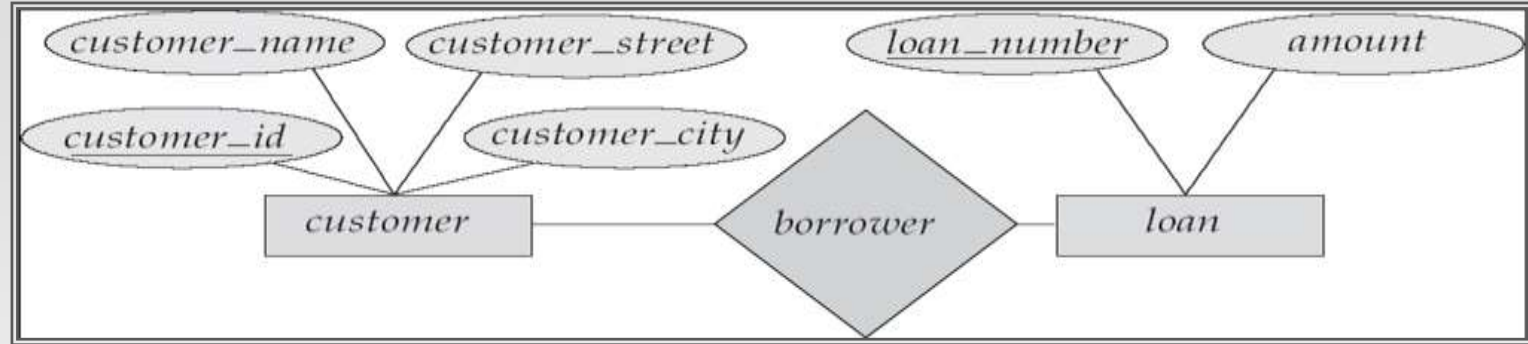
Keys

- 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**.

Keys for Relationship Sets

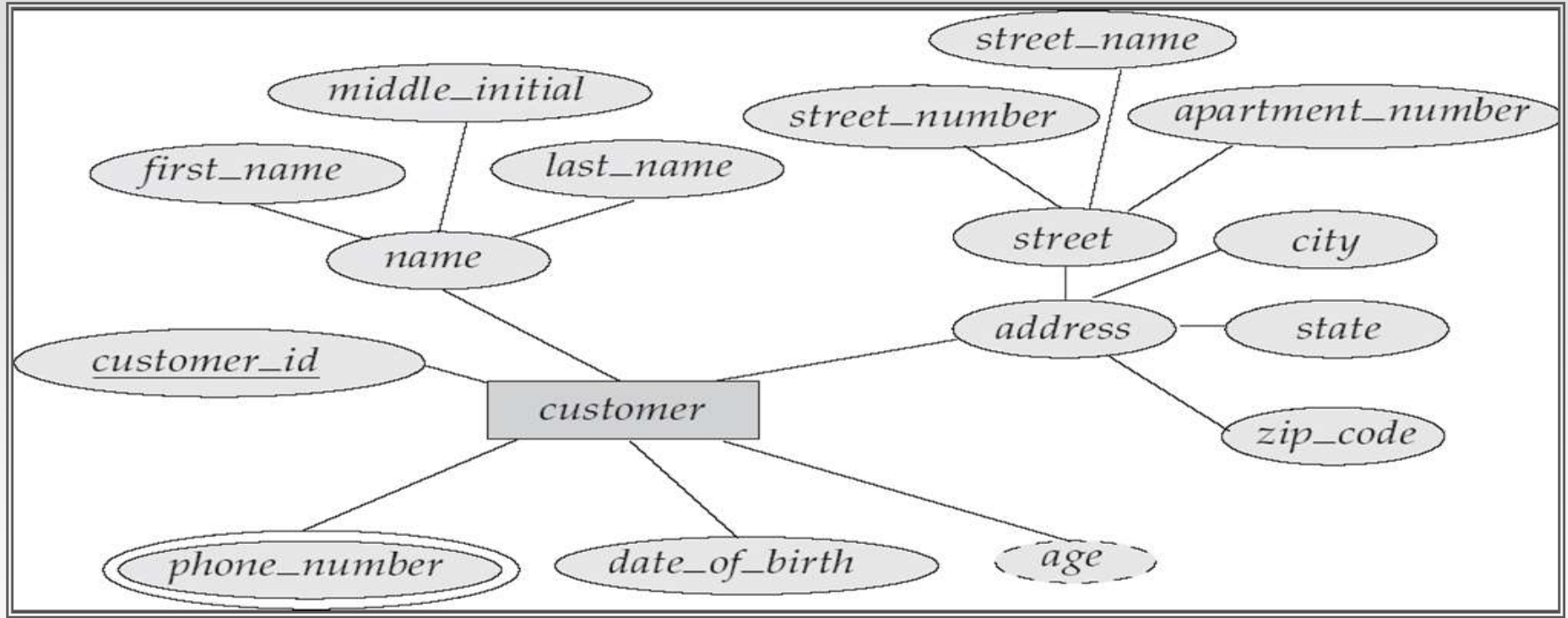
- The combination of primary keys of the participating entity sets forms a super key of a relationship set.
 - *(customer_id, account_number)* is the super key of *depositor*
 - *NOTE: this means a pair of entity sets can have at most one relationship in a particular relationship set.*
- 4 Example: if we wish to track all *access_dates* to each account by each customer, we cannot assume a relationship for each access. We can use a multivalued attribute though
- Must consider the mapping cardinality of the relationship set when deciding 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

E-R Diagrams

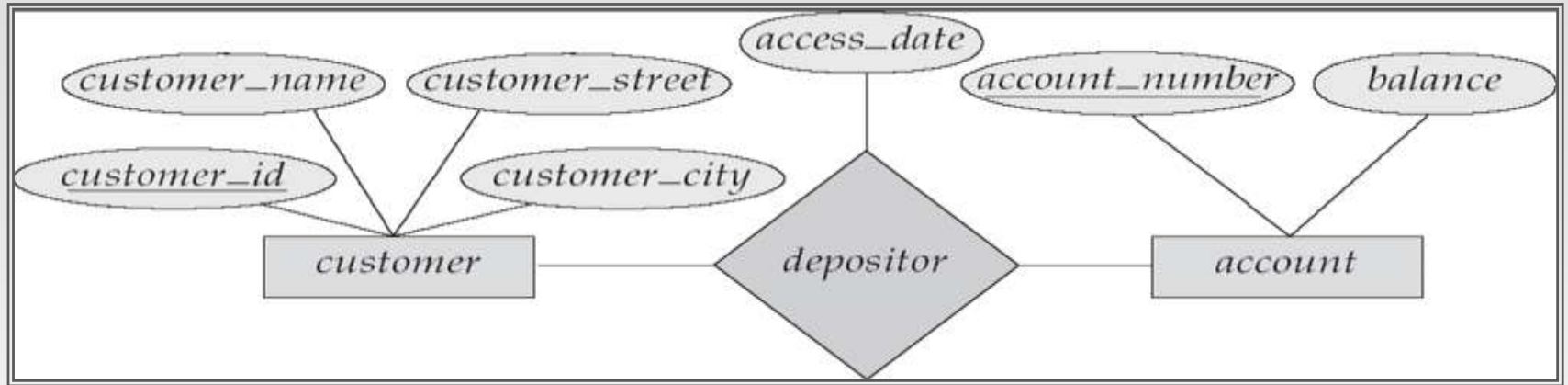


- Rectangles represent entity sets.
- Diamonds represent relationship sets.
- Lines link attributes to entity sets and entity sets to relationship sets.
- Ellipses represent attributes
 - Double ellipses represent multivalued attributes.
 - Dashed ellipses denote derived attributes.
- Underline indicates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

