Chapter 2
Entity-Relationship Model

2. Entity-Relationship Model

2.1 Basic concepts

2.2 Mapping constraints

2.3 Keys

2.4 Entity-relationship diagram

2.5 Weak entity sets

2.6 Extended E-R features

2.7 Reduction of an E-R schema to tables

Data Models

Data models are a collection of conceptual tools for describing (structure of database)

- data,
- data relationships,
- data semantics and
- data constraints

Different types of data models

- Entity-Relationship model
- Relational model
- Other models:
  - Object-oriented data model
  - Object-relational data model
  - Semi-structured data models
  - Older models: network data model and hierarchical data model

Entity-Relationship Model

Is a high-level data model

is based on a perception of the world as consisting of a collection of basic objects (entities) and relationships among these objects

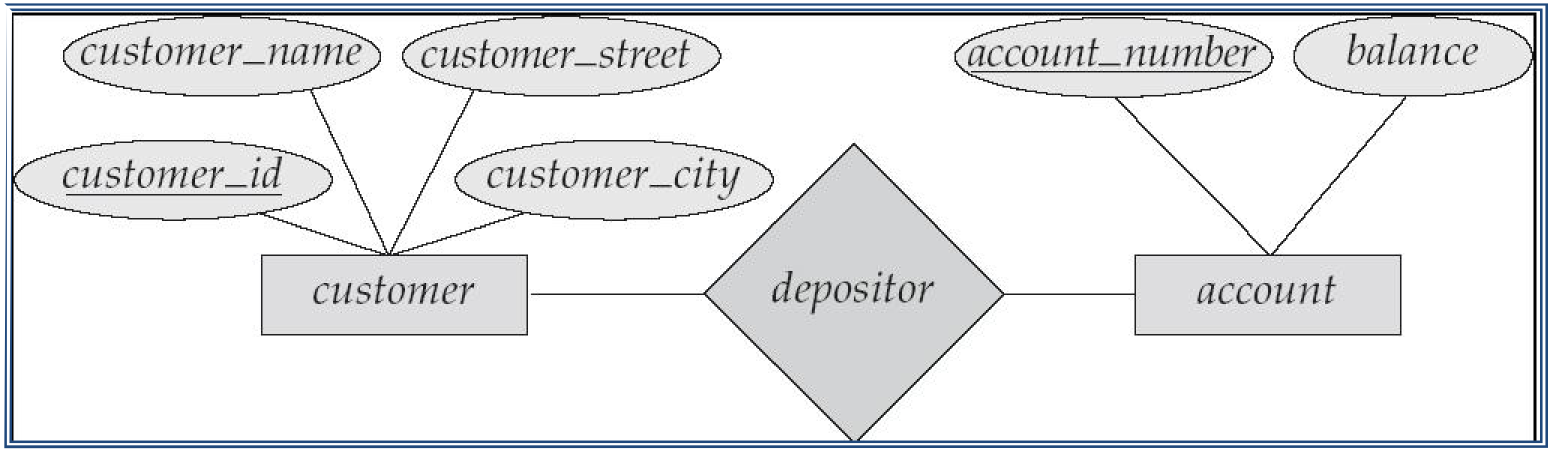


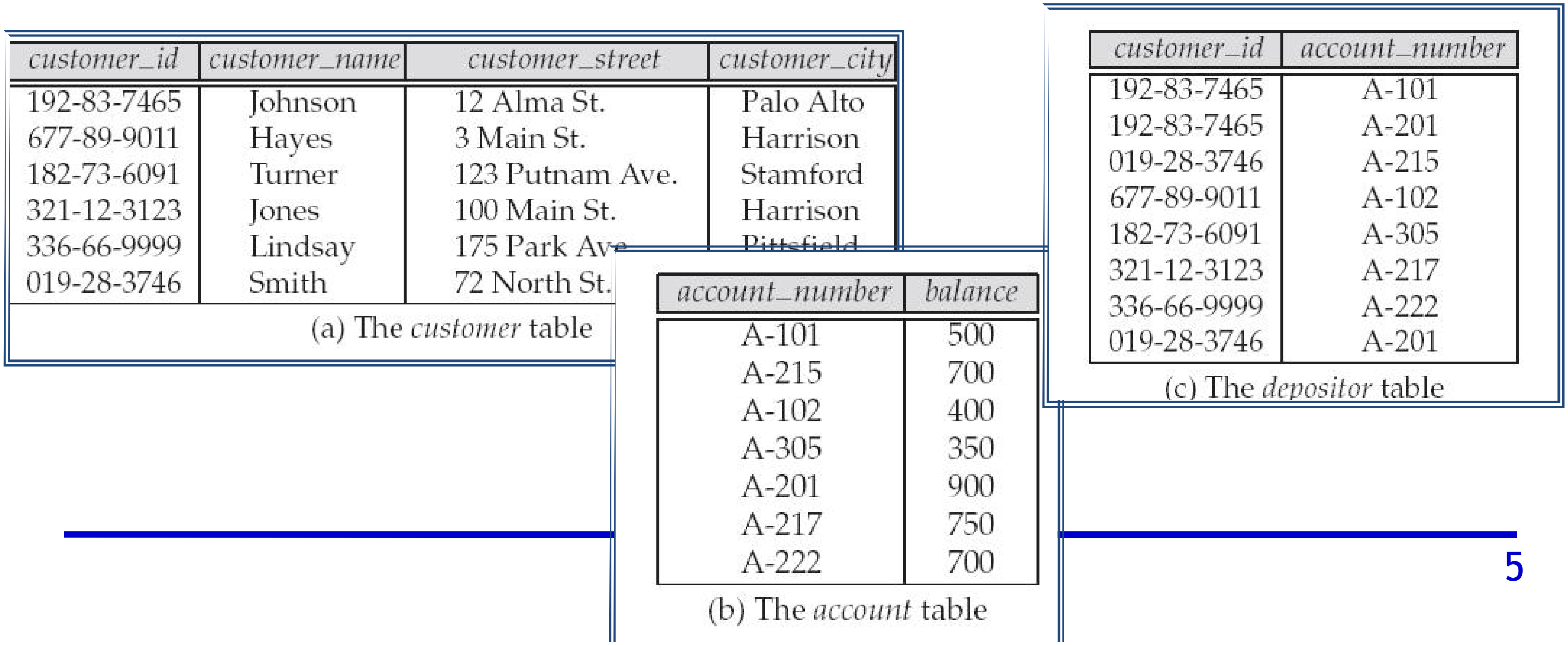
Fig. ER Diagram

Relational Model

Data and relationships are represented by a collection of tables.

Each table has multiple columns and and each column has a unique name.

A row in a table represents a relationship among set of values.



customer_id	*customer_name*	*customer_street*	*customer_city*
192-83-7465	Johnson	12 Alma St.	Palo Alto
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.	

(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

Object-oriented and Object-relational model

Object-oriented data Model

Entension of E-R Model

With notions of encapsulation, methods(functions), and object identity.

Object-relational data Model

Combines features of the object-oriented data model and relational data model.

Title	Author	Publisher
		(Pub-name, pub-branch)
Compilers	Smith	(McGraw-Hill, New York)
Compilers	Jones	(McGraw-Hill, New York)
Networks	Bob	(Oxford, London)

Semi-structured data model

Individual data items of the same type may have different sets of attributes.

Extensible markup language XML represents semi-structure data.

Example of semi-structured data model in XML

```
<customer>
  <firstname>Peter</firstname>
  <lastname>Wood</lastname>
</customer>
<customer>
        <firstname> Mark</firstname>
        <lastname> Levene</lastname>
        <email> mark@dcs.bbk.ac.uk</email>
  </customer>
```

Network Model

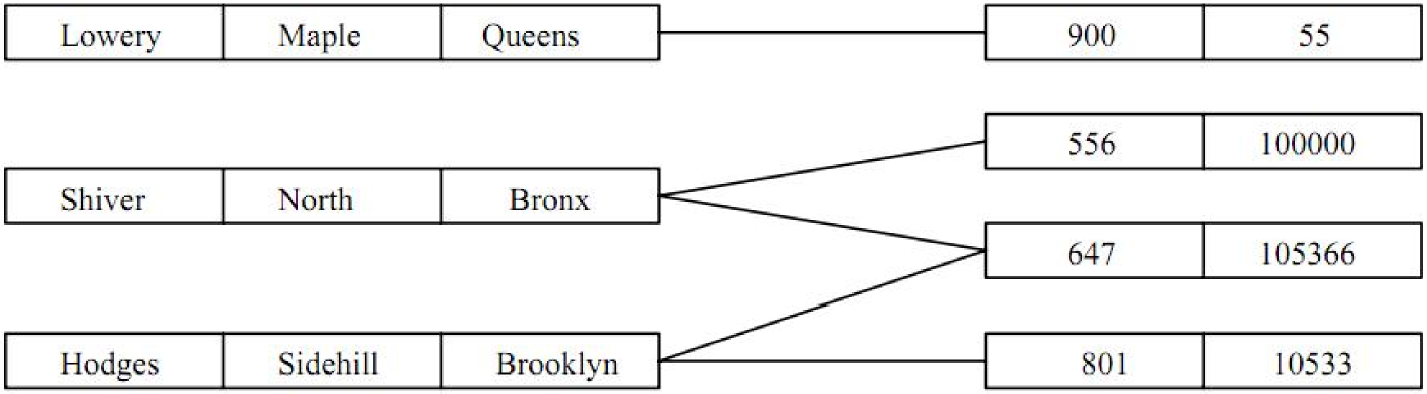
The Network Model

- Data are represented by collections of records.
- Relationships among data are represented by links.
- Organization is that of an arbitrary graph.

name	street	city	number
Lowery	Maple	Queens	900
Shiver	North	Bronx	556
Shiver	North	Bronx	647
Hodges	Sidehill	Brooklyn	801
Hodges	Sidehill	Brooklyn	647

number	balance
900	55
556	100000
647	105366
801	10533

Fig. Example of relational database



Lowery | Maple | Queens

Shiver | North | Bronx

Hodges | Sidehill | Brooklyn

900 | 55

556 | 100000

647 | 105366

801 | 10533

Fig. Example of network database

Hierarchical Model

The Hierarchical Model

- Similar to the network model.
- Organization of the records is as a collection of trees, rather than arbitrary graphs.

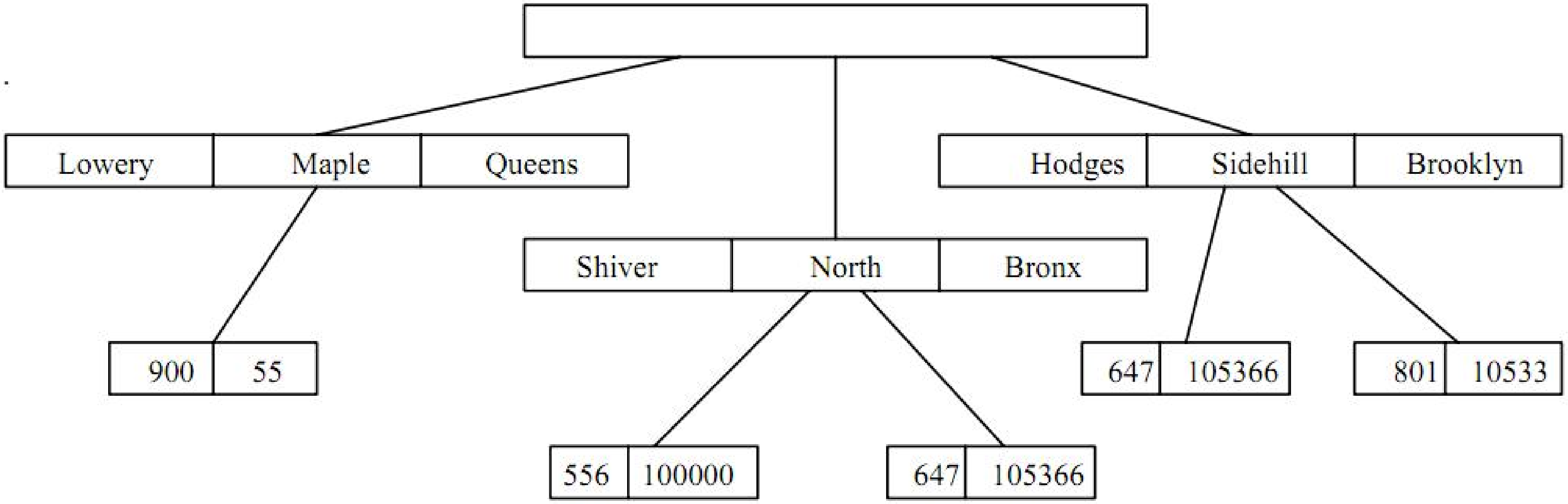
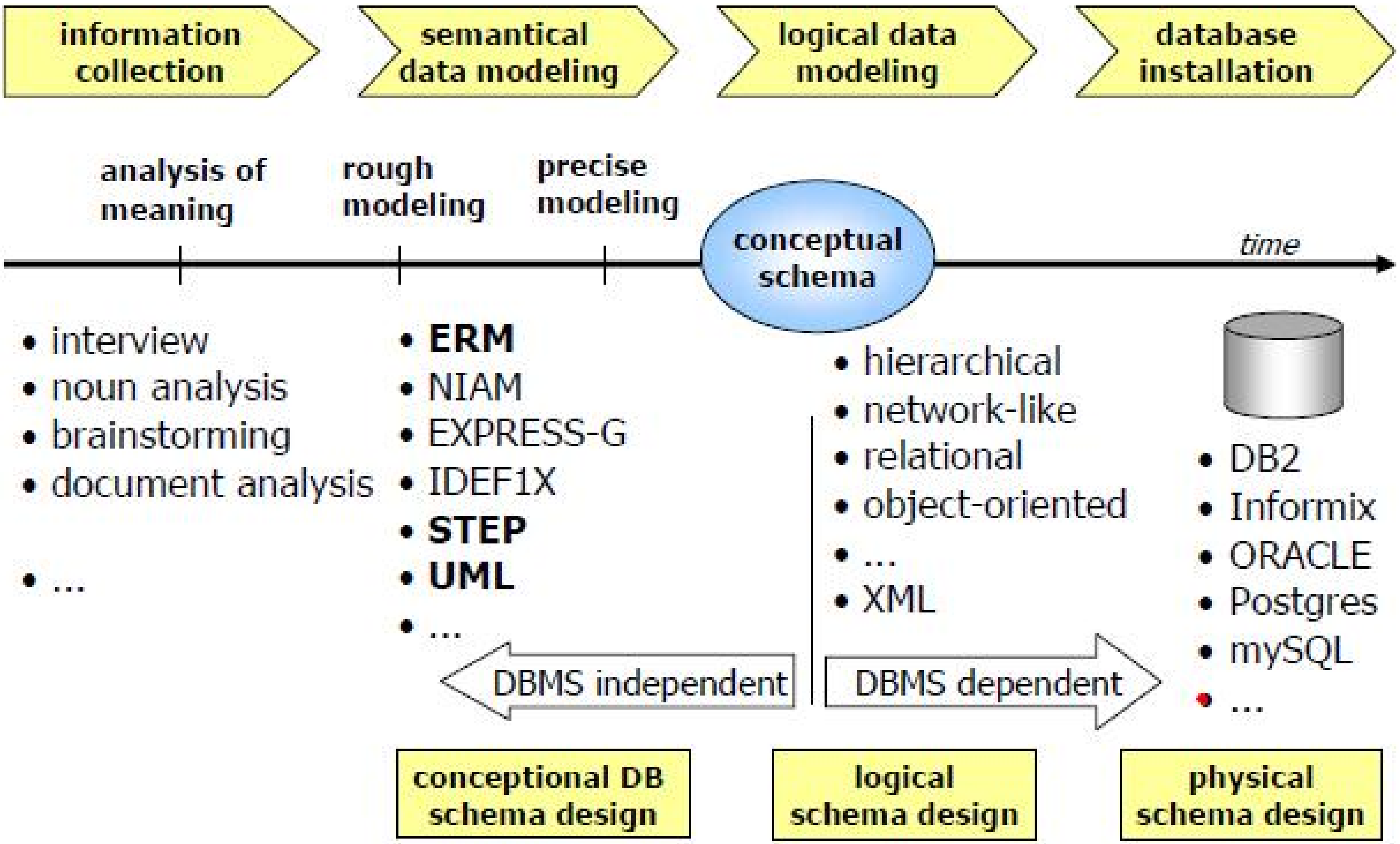


Fig. Example of network database

Database design process



information collection

semantical data modeling

logical data modeling

database installation

analysis of meaning

rough modeling

precise modeling

conceptual schema

time

- interview
- noun analysis
- brainstorming
- document analysis
- …

- **ERM**
- NIAM
- EXPRESS-G
- IDEF1X
- **STEP**
- **UML**
- …

- hierarchical
- network-like
- relational
- object-oriented
- …
- XML

- DB2
- Informix
- ORACLE
- Postgres
- mySQL
- …

DBMS independent

DBMS dependent

conceptional DB schema design

logical schema design

physical schema design

Entities and Entity Sets

A database can be modeled as:
– a collection of entities,
– relationships among entities.

An entity is an object that exists and is distinguishable from other objects.
Example: each person in an enterprise is an entity, a book, a loan etc

Each entity has a set of properties and values for some properties uniquely identify an entity.
Example: Property name "Roll-No" and value 1, uniquely identifies a particular student.

An entity set is a set of entities of the same type that share the same properties.
Example: set of all persons who are customers of bank is defined as the entity set "customer", set of all loans,

Entity sets need not be disjoint. For example, the entity set employee (all employees of a bank) and the entity set customer (all customers of the bank) may have a person entity as an employee entity, a customer entity, both or neither.

Attributes

An entity is represented by

- a set of attributes,
- Attributes are descriptive properties possessed by each members of an entity set.

customer = (customer-name, social-security, customer-street, customer-city)

account = (account-number, balance)

Each entity has a value for each of its attributes

Examples: for a particular customer

customer-id=321-12-3123

customer-name=Jones

Domain(value set) of the attribute: the set of permitted values for that attribute.

Examples: customer-name: text string with certain fixed length

Telephone number of seven positive digit (01442121)

Attribute types

Simple and composite attributes

- Simple attributes that are not divided into parts
- Composite attributes that can be divided into subparts

Single-valued and multivalued attributes

- The attributes that have a single value for a particular entity is Single-valued attributes.
- The attributes that can have a set of values for a particular entity is multivalued attributes.

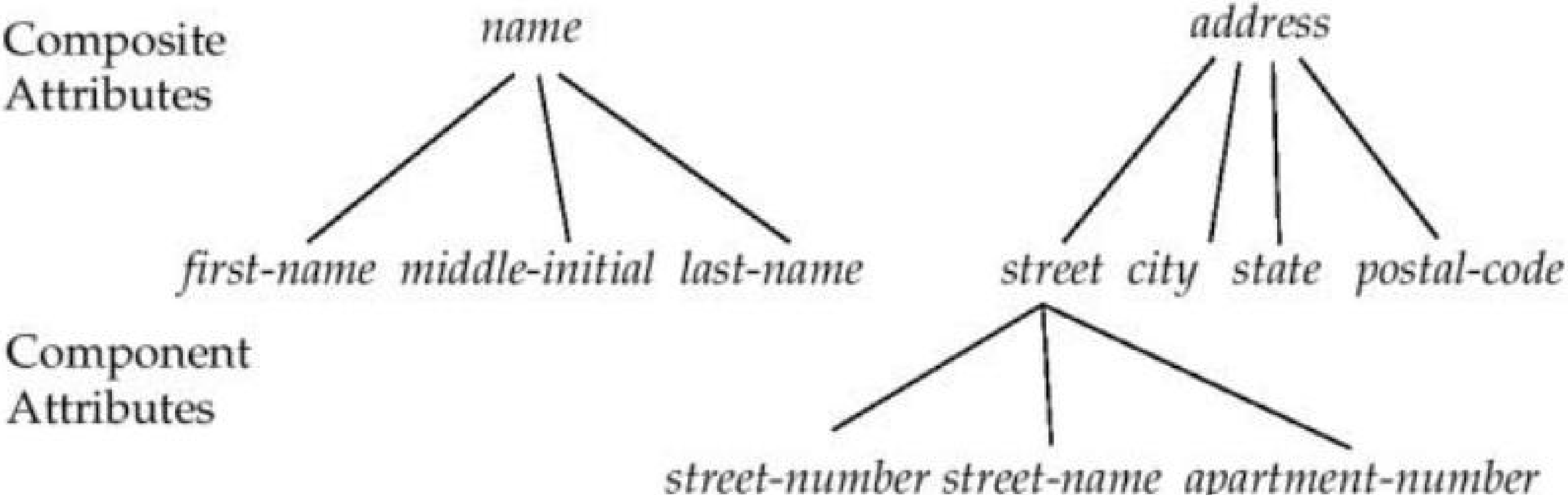
Derived attributes

- The value for this type of attribute can be derived from the value of other related attributes or entities.
  - From calculating age from date-of-birth and current date
  - Deriving number of loans for each customer by counting the loans associated with that customer.

Null atttributes

- An attribute that takes a null value when an entity does not have a value for it.
- null indicates "not applicable", E.g. some people may not have middle name

Composite Attributes



Relationship and Relationship Sets (1)

A relationship is an association among several entities.

Example		
Hayes	Depositor	A-102
Customer entity	*Relationship set*	*Account Entity*

A relationship set is a set of relationships of the same type.

- students *registered* in a course
- passengers *booked* on a flight
- Customers *deposits* into Account

Relationship and Relationship Sets (2)

Formally, relationship set is a mathematical relation on n>=2 entity sets.

If E_1, E_2, ..E_n are entity sets, then relationship set R is a subset of

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

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

Example:

o *depositor* = $\{(c, a) \mid c \in \text{customer}, s \in \textit{Account})\}$

Relationship and Relationship Sets (2)

- *(Hayes, A-102) ⊆ depositor*

Relationship and Relationship Sets

The association between entity sets is referred to as participation, that is the entity set E1, E2, ..En participates in relationship set R.

A Relationship instance in a E-R schema represents association between the named entities in the real-world enterprise that is being modeled.

Eg. The relationship instance represents that, in the real-world enterprise, the person called Hayes who holds customer-id 677-89-9011 has taken the loan that is numbered L-15

The role of an entity

The role of an entity is the function it plays in a relationship.

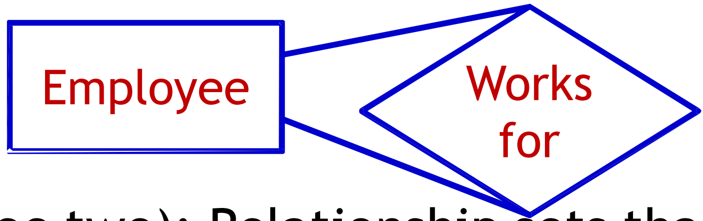
For example, the relationship *works-for* could be ordered pairs of employee entities.

The first employee takes the role of manager, and the second one will take the role of worker.

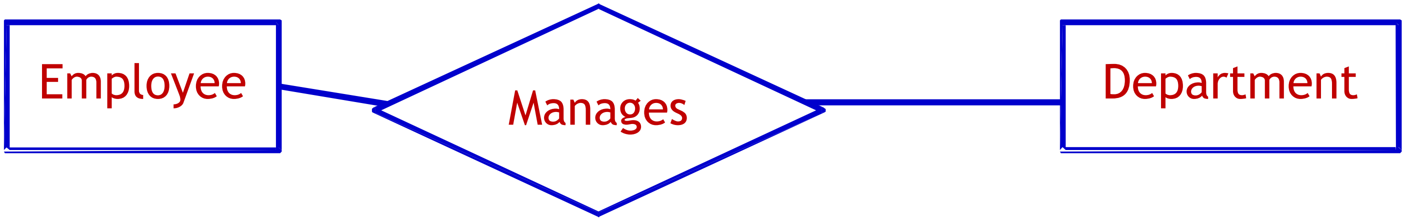
Degree of a Relationship Set

Refers to number of entity sets that participate in a relationship set.

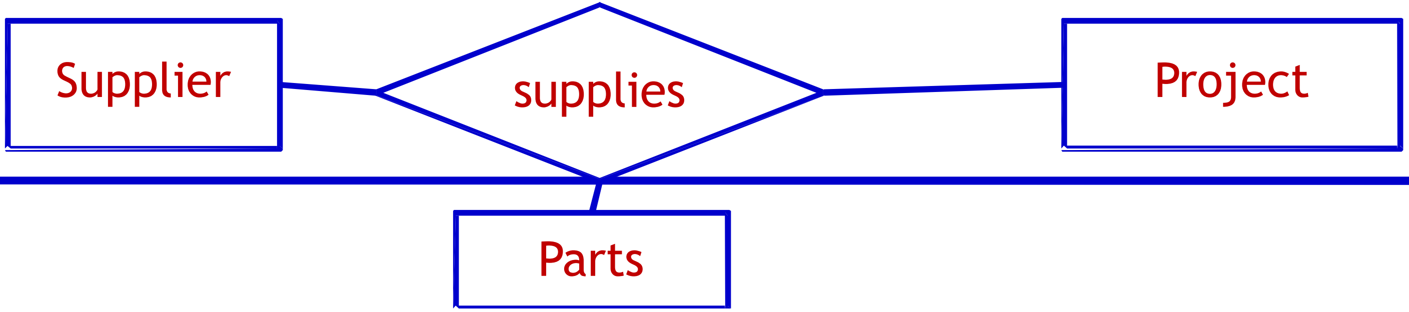
Unary(degree one): Relationship sets that involve one entity set only.



Binary(degree two): Relationship sets that involve two entity sets.



N-ary(degree n): Relationship sets that involve n enitity sets.

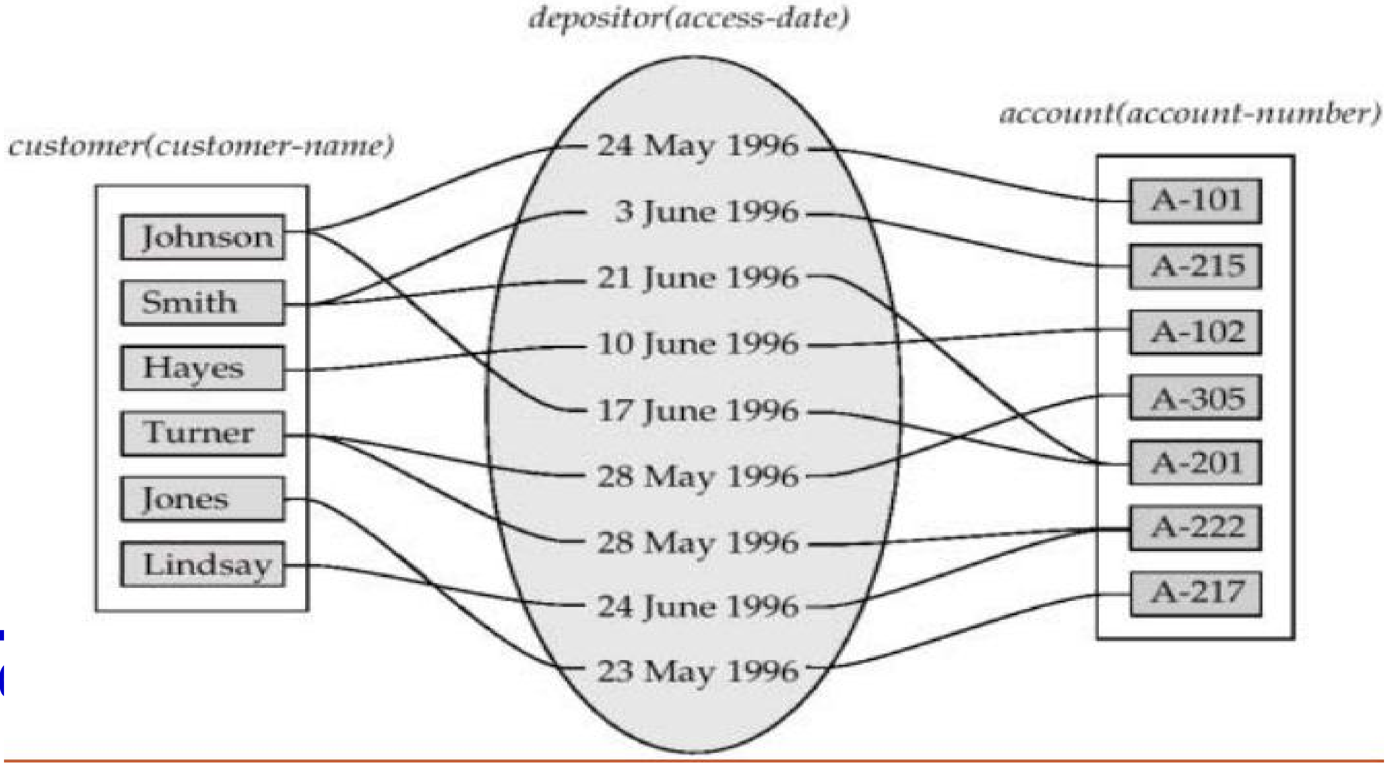


Constraints

A relationship may also have descriptive attributes.

For example, date (account access) could be an attribute of the depositor relationship set.

A relationship instance in a given relationship set must be uniquely identifiable from participating entities, without using descriptive attributes.



Constraints

Constrains that may be defined on ER Schema diagram, which the contents of database must confirm are

- Mapping cardinalities
- participation constraints

Mapping cadrinalities (Cardinality ratio) express the number of entities to which another entity can be associated via a relationship set.

Mapping Cardinalities (1)

For a binary relationship set R between entity set A and B, the mapping cardinality must be

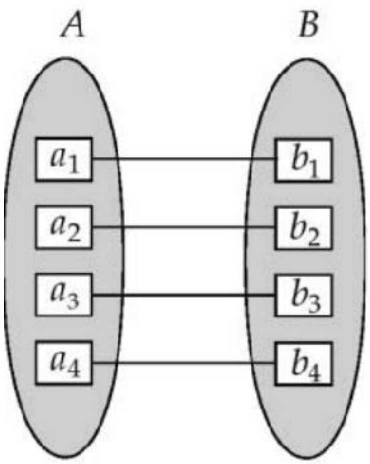
Onetoone: An entity in A is associated with at most one entity in B, and an entity in B is associated with at most one entity in A.

Onetomany: An entity in A is associated with any number(zero or more) of entities in B. But, An entity in B can be associated with at most one entity in A.

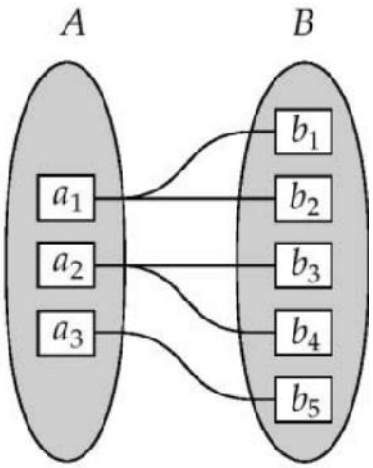
Manytoone: An entity in A is associated with at most one entity in B. But, an entity in B can be associated with any number (zero or more) of entities in A.

Manytomany: An entity in A is associated with any number (zero or more) of entities in B, and an entity in B is associated with any number (zero or more) of entities in A.

Mapping cardinalities (2)

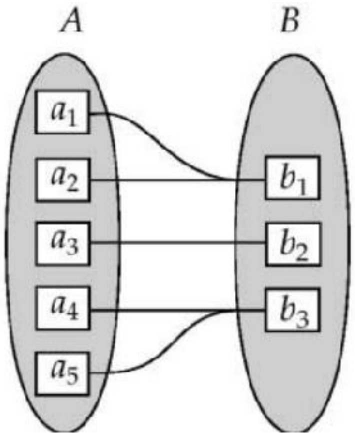


One to One

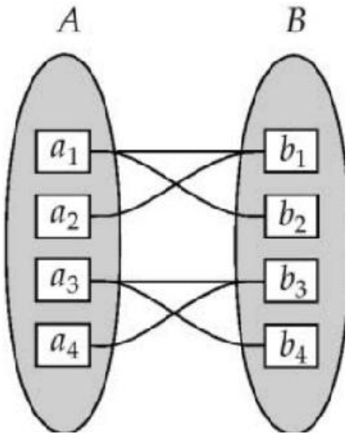


One to Many

Onetoone: Person marriages person
Onetomany: A customer has a multiple loans
Manytomany: Students are registerd for courses

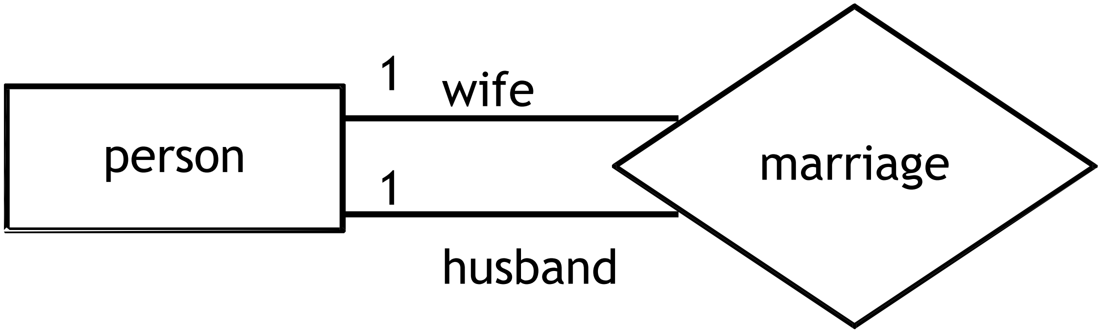


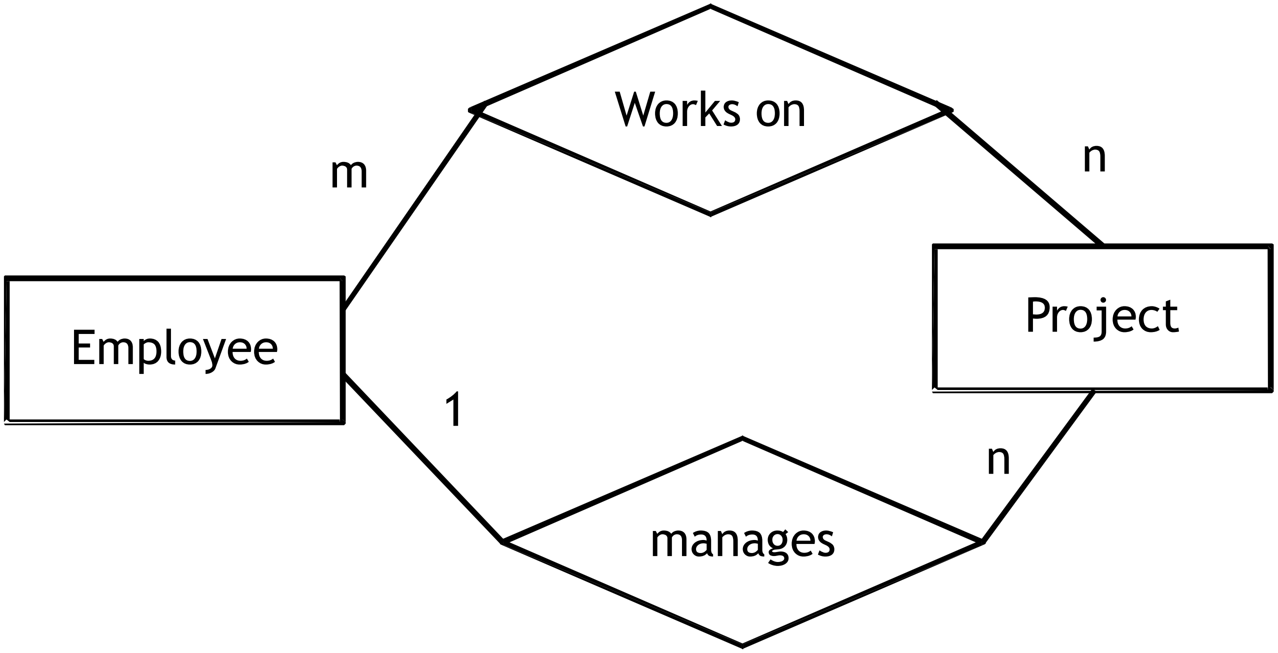
Many to One



Many to Many

Mapping cardinalities (3)





Participation Constraints

Totalparticipation:

the participation of an entity set E in a relationship set R is said to total if every entity in E participates in at least one relationship in R.

Partialparticipation:

the participation of an entity set E in a relationship set R is said to partial if only some entity in E participates in relationships in R.

Examples:

Total: Every loan entity is related with at least one customer

Partial: Some of the customer of bank are related to the loan entity.

Keys

A superkey of an entity set is a set of one or more attributes whose values uniquely determine each entity.

A candidatekey of an entity set is a minimal super key.

- *Customer-id* is a candidate key of customer
- *Account-number* is a candidate key of account

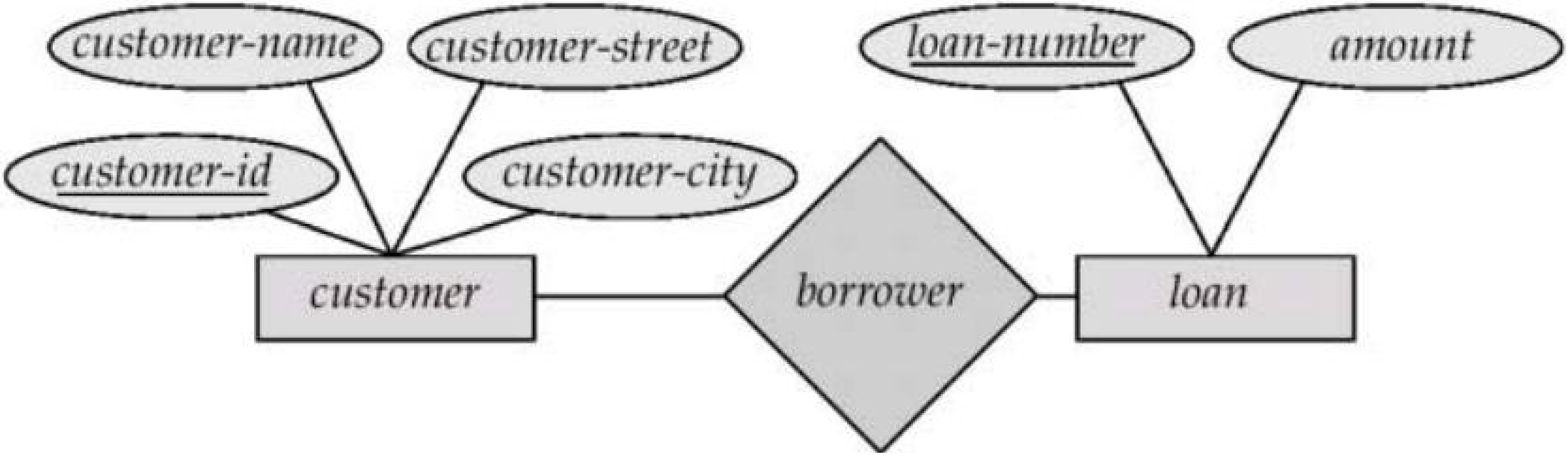
Although several candidate keys may exist, one of the candidate keys is selected to be the primarykeys.

The values of the primary key attribute(s) can be used to identify each entity uniquely.

- *Sometimes it has to be created artificially (serial number)*

CompositeKey is a candidate key that consists of two or more attributes .

ER-diagram

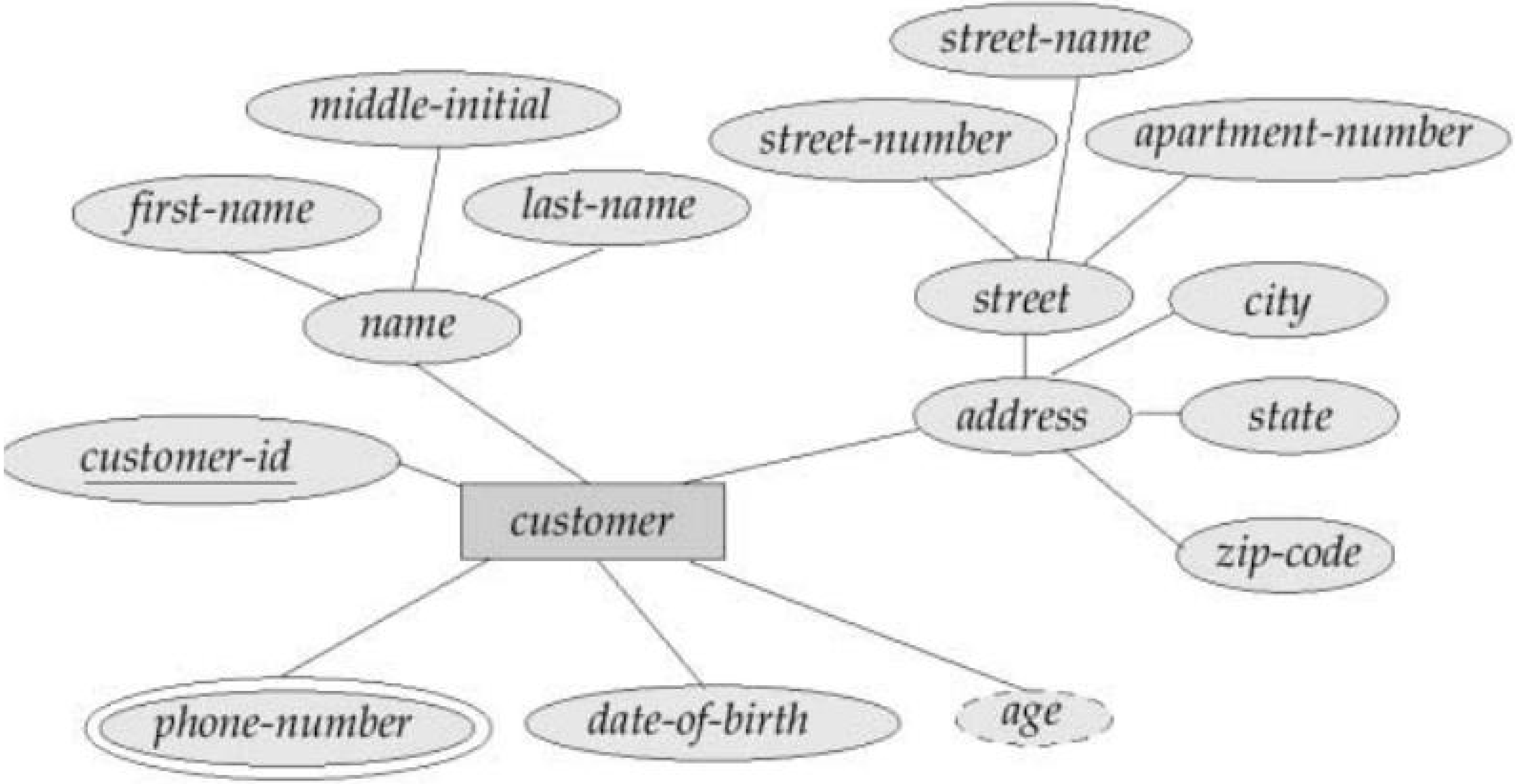


Rectangles represent entity sets.
Diamonds represent relationship sets.
Lines link attributes to entity sets and entity sets to relationship sets.
Ellipses represent attributes

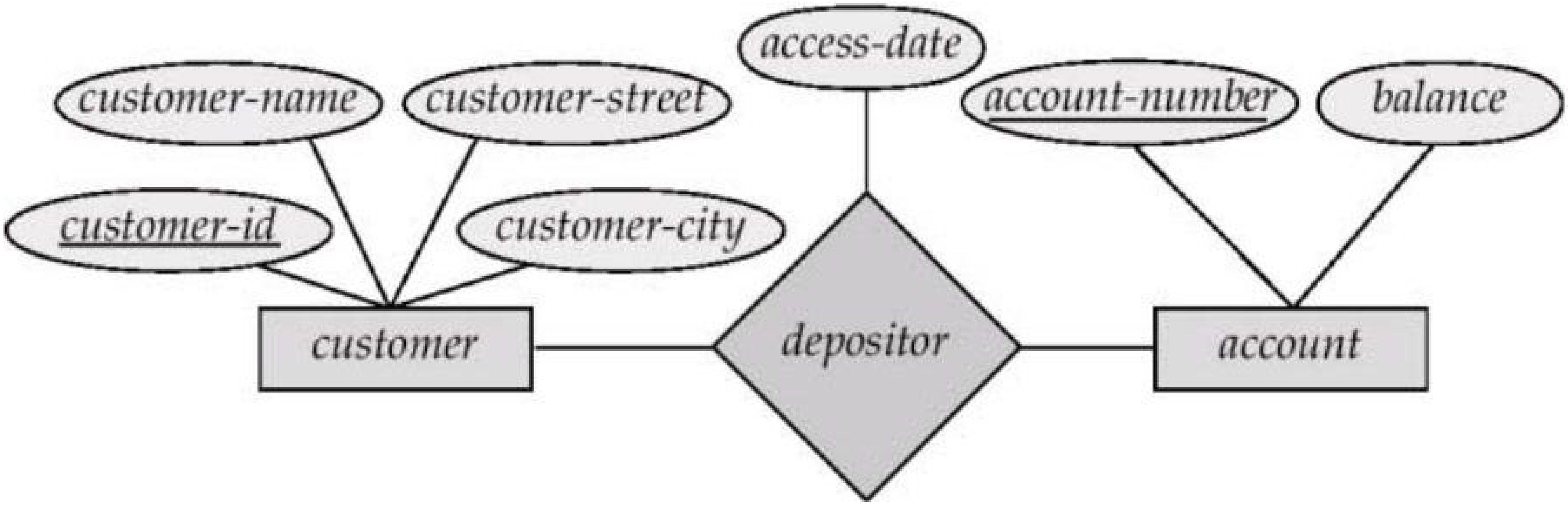
- Double ellipse represent multivalued attributes.
- Dashed ellipse denote derived attributes.

Underline indicates primary key attributes
Double line indicate total participation of an entity in relationship set
Double rectangles represents weak entity sets

ER Diagram with composite, multivalued and derived attributes



Relationship set with Attributes



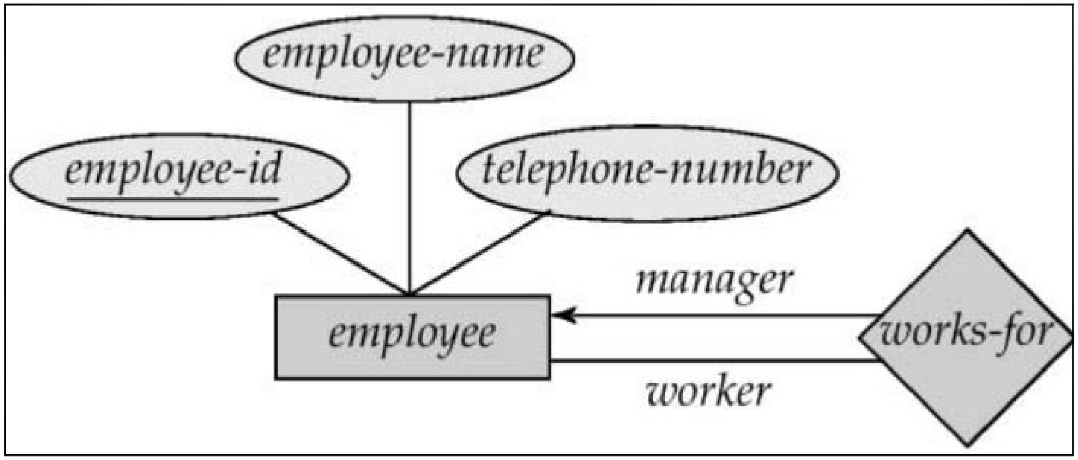
Roles

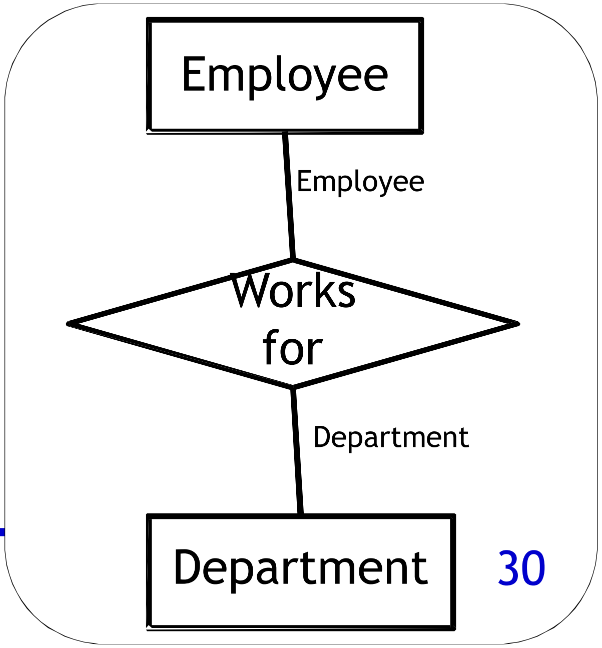
Entity sets of a relationship need not be distinct

they specify how roles The labels “manager” and “worker” are called 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



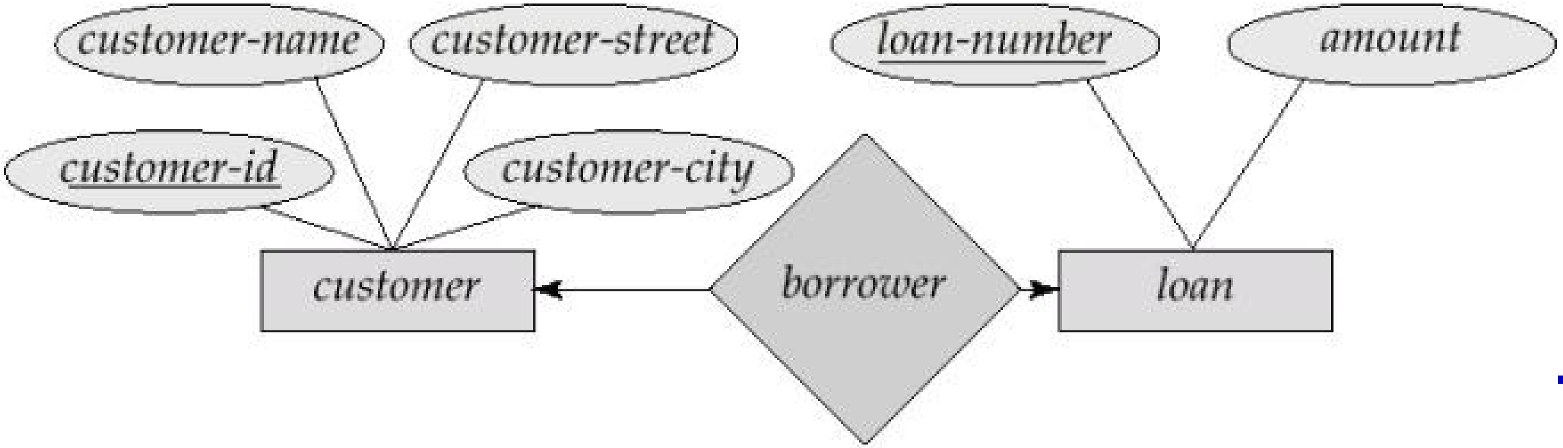


Mapping cardinalities (1)

Cardinality constraints are expressed by drawing either a directed line ( ), signifying “one,” or an undirected line (—), signifying “many,” between the relationship set and the entity set.

Example One to one:

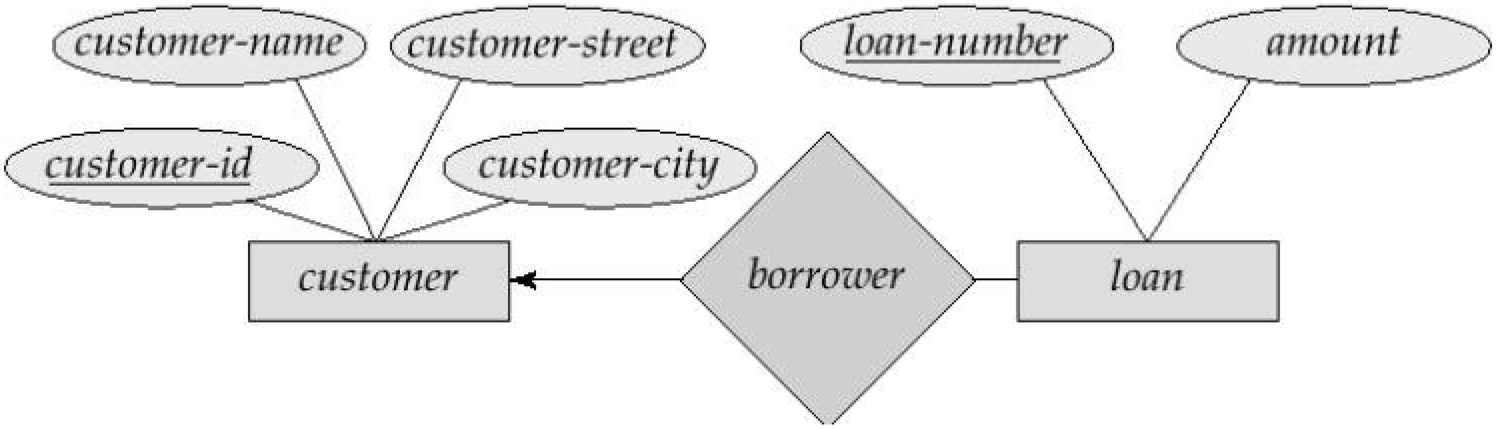
- A customer is associated with at most one loan via the relationship “borrower”
- A borrower loan is associated with at most one customer via the relationship “borrower"



One-to-Many (2)

A loan is associated with at most one customer via borrower

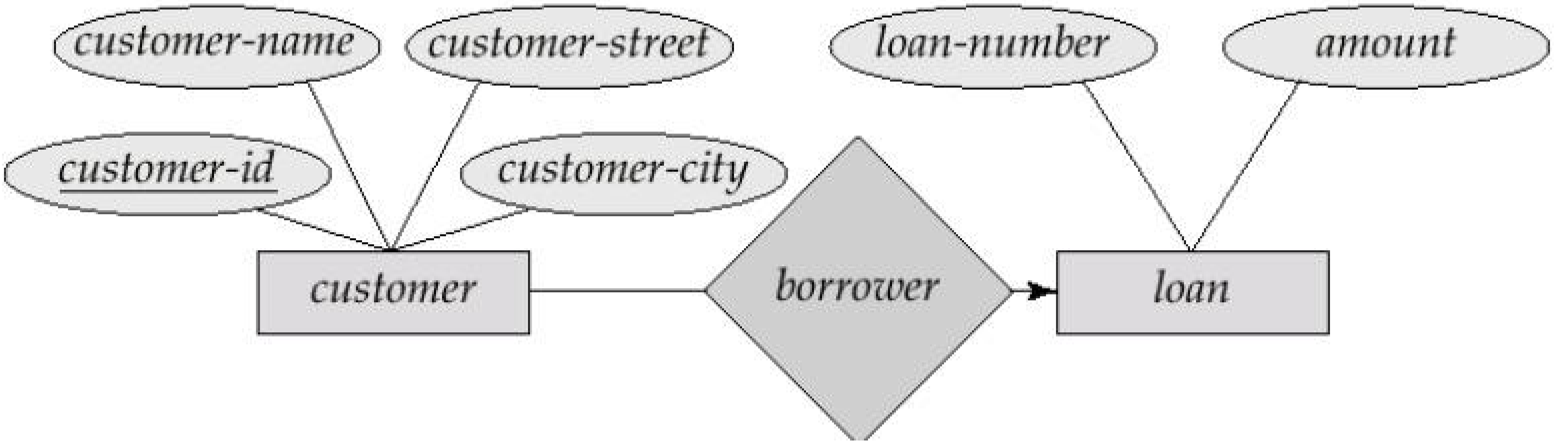
A loan is associated with several several(including 0) loans via borrower



Many-to-one (3)

In a many-to-one a loan is associated with several (including 0) customers via borrower.

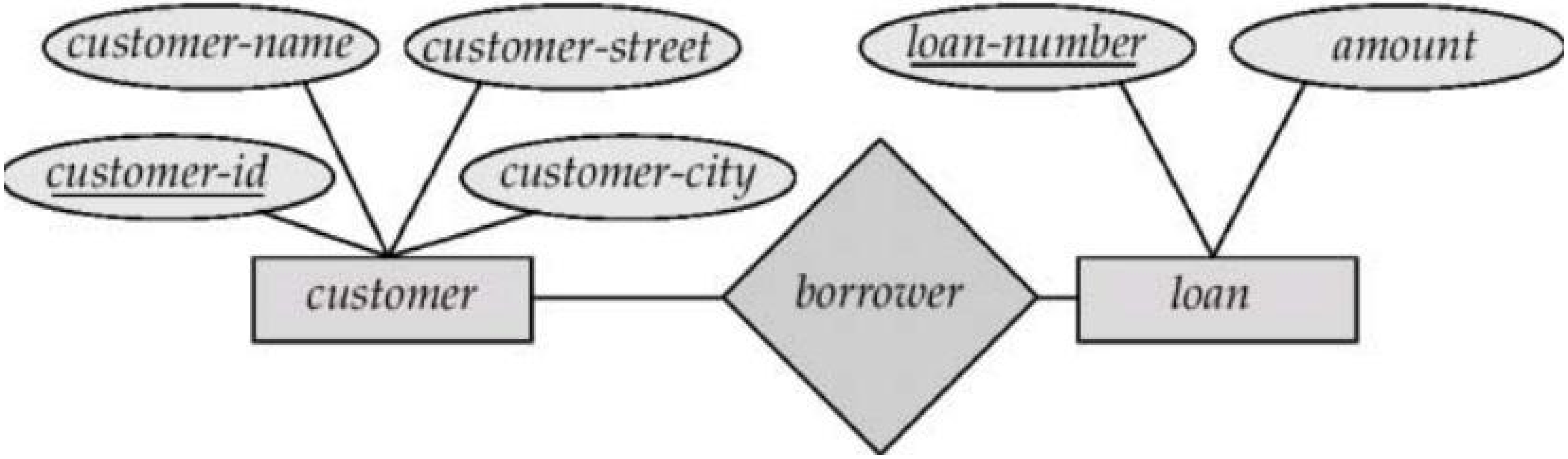
a customer is associated with at most one loan via borrower.



Many-to-many (4)

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

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



Participation of Entity Sets in a Relationship Sets

TotalParticipation (indicated by double line): every entity in the entity set participates in at least one relationship in the relationship set

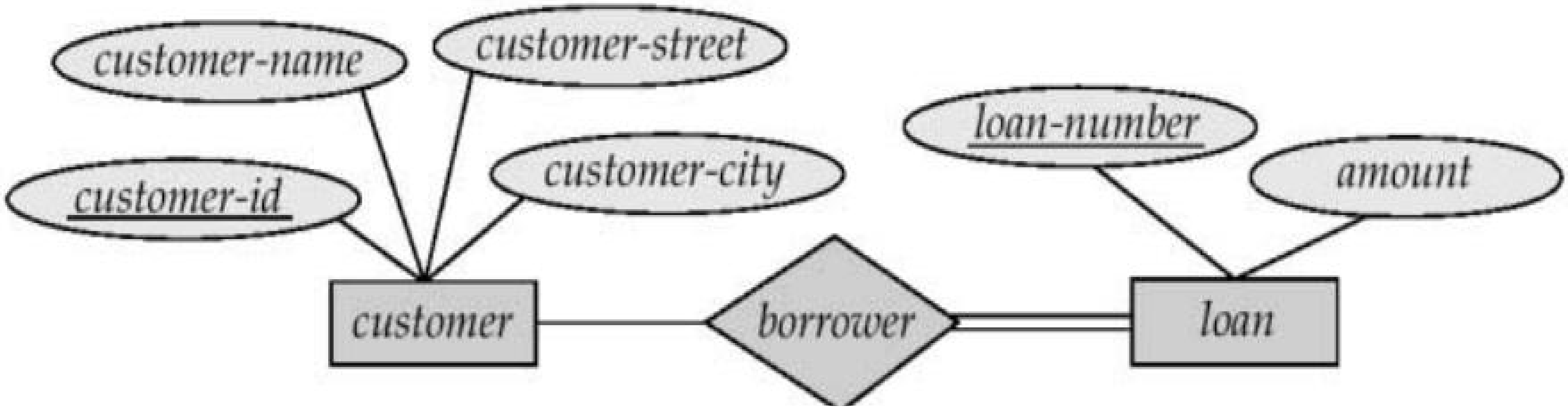
E.g. participation of loan in borrower is total.

- every loan must have a customer associated to it via borrower.

Partialparticipation (single line): some entities may not participate in any relationship in the relationship set

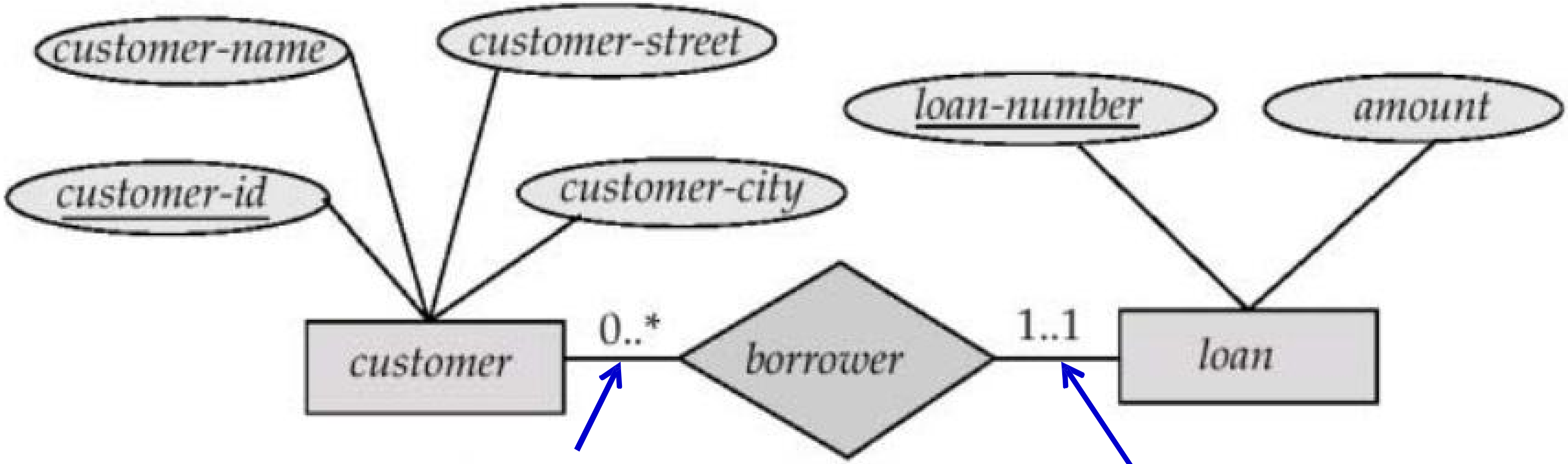
E.g. participation of is partial borrower in customer

- Some customer of bank may not borrow loan. Hence, some of the customers may not have any relationship in the relationship set.



Alternative Notation for Cardinality Limits (Participation constraints) (1)

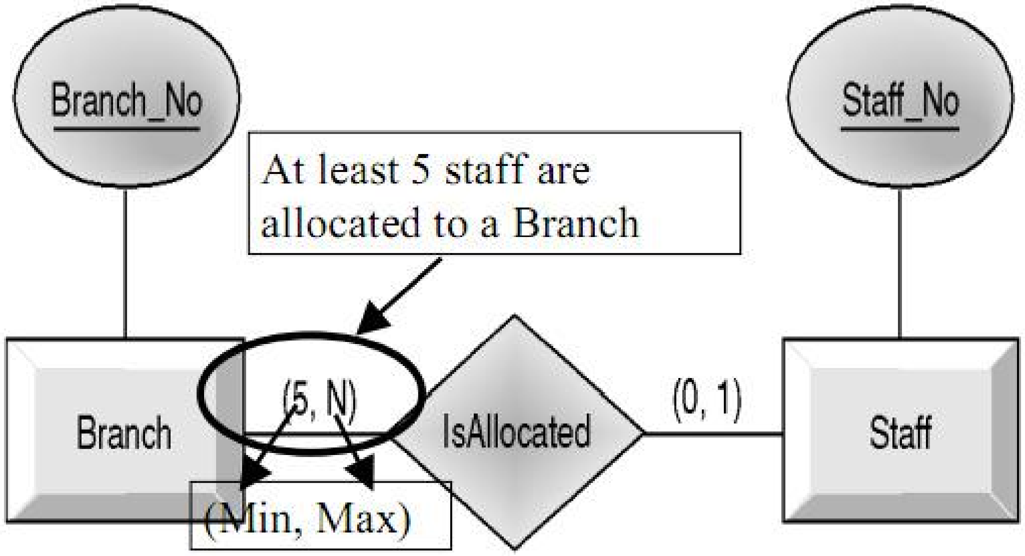
Cardinality limits can also express participation contraints.



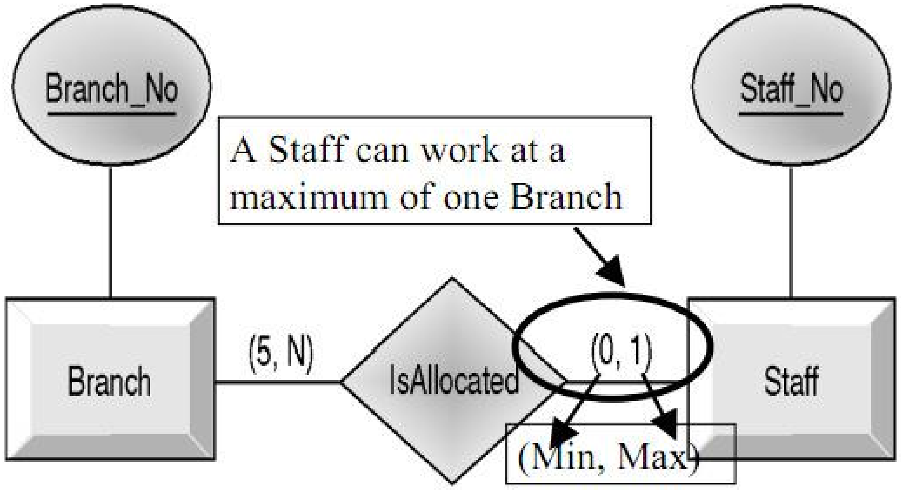
0 to many loans are allocated for a customer

No more double lines
A loan must have one customer
If(Min>0), then total participation

Alternative Notation for Cardinality Limits (Participation constraints) (2)

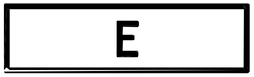
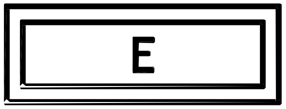
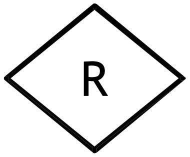
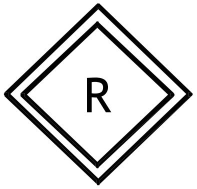


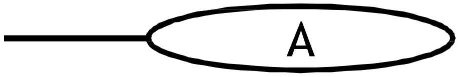
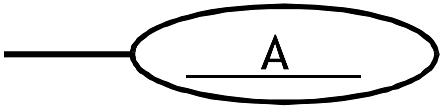
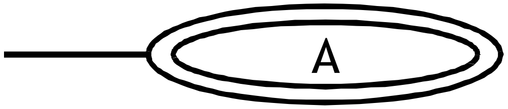
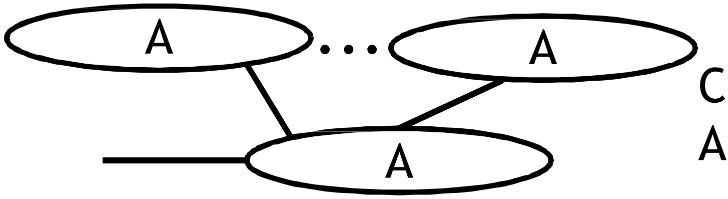
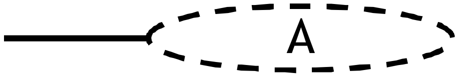
At least 5 staff are allocated to a Branch



A Staff can work at a maximum of one Branch

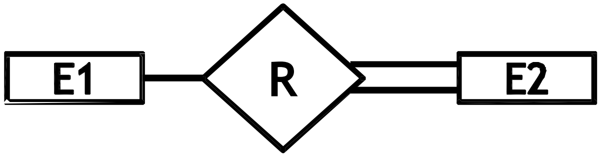
ER Diagram Symbols and Meaning (1)

Symbol	Meaning
	Entity Set
	Weak Entity Set
	Relationship Set
	Indentifying Relationship Set

Symbol	Meaning
	Attribute
	Key Attribute
	Multivalued Attribute
	Composite Attribute
	Derived Attribute

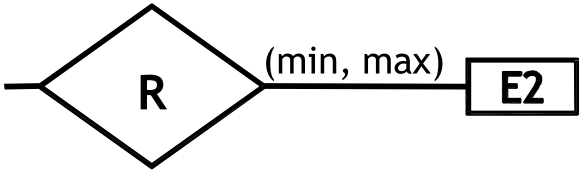
ER Diagram Symbols and Meaning

Symbol Meaning

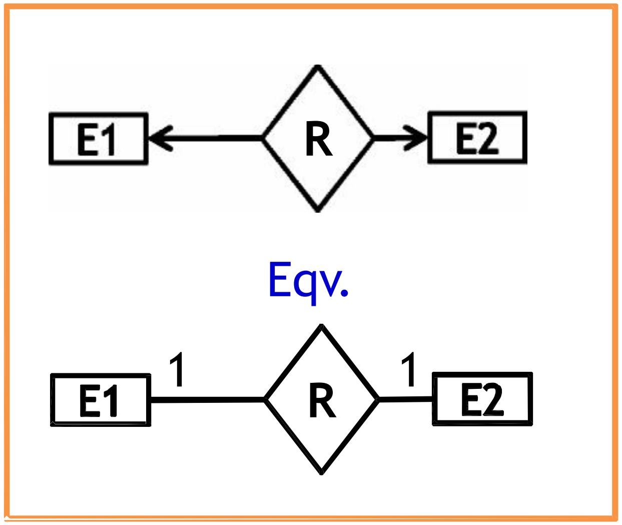


Total Participation of E2 in R

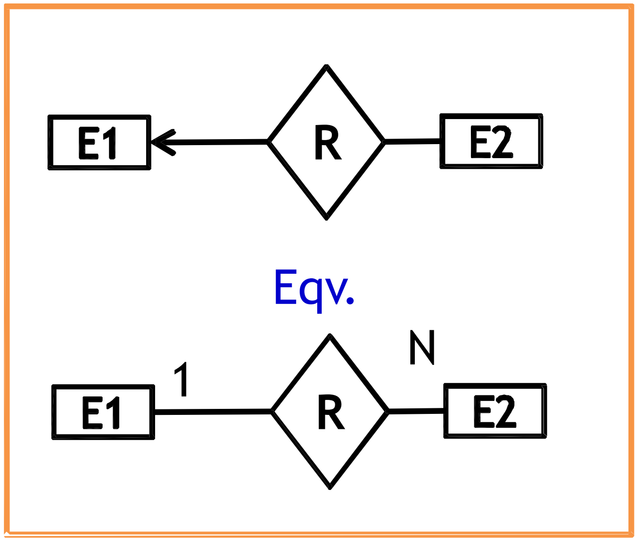
Symbol Meaning



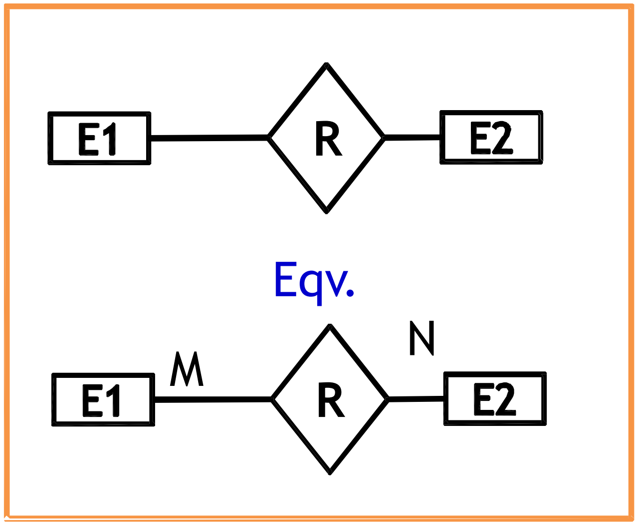
Cardinality limits



One to one



One to many



Many to many

Draw a ER-Diagram of a Company (1)

Consider a company database which keeps track of a company‘s employees, departments and projects

The company is organized into departments. Each department has a name, a unique number, and a particular employee who manages the department. It also keeps track of the start date when the employee began managing the department. A department may have several locations.

A department controls a number of projects, each of which has unique name, and a single location.

Draw a ER-Diagram of a Company (2)

Employee name, social security number, address, salary, sex, and birthdate are required. Many employees works in a same department and an employee is assigned to one department

An employee may work on several projects, which are not necessarily controlled by the same department. We keep track of the number of hours per week that an employee works on each project.

We also keep track of the direct supervisor of each employee.

We want to keep track of the dependents of each employee for insurance purposes. We keep each dependent‘s first name, sex, birth date and relationship to the employee.

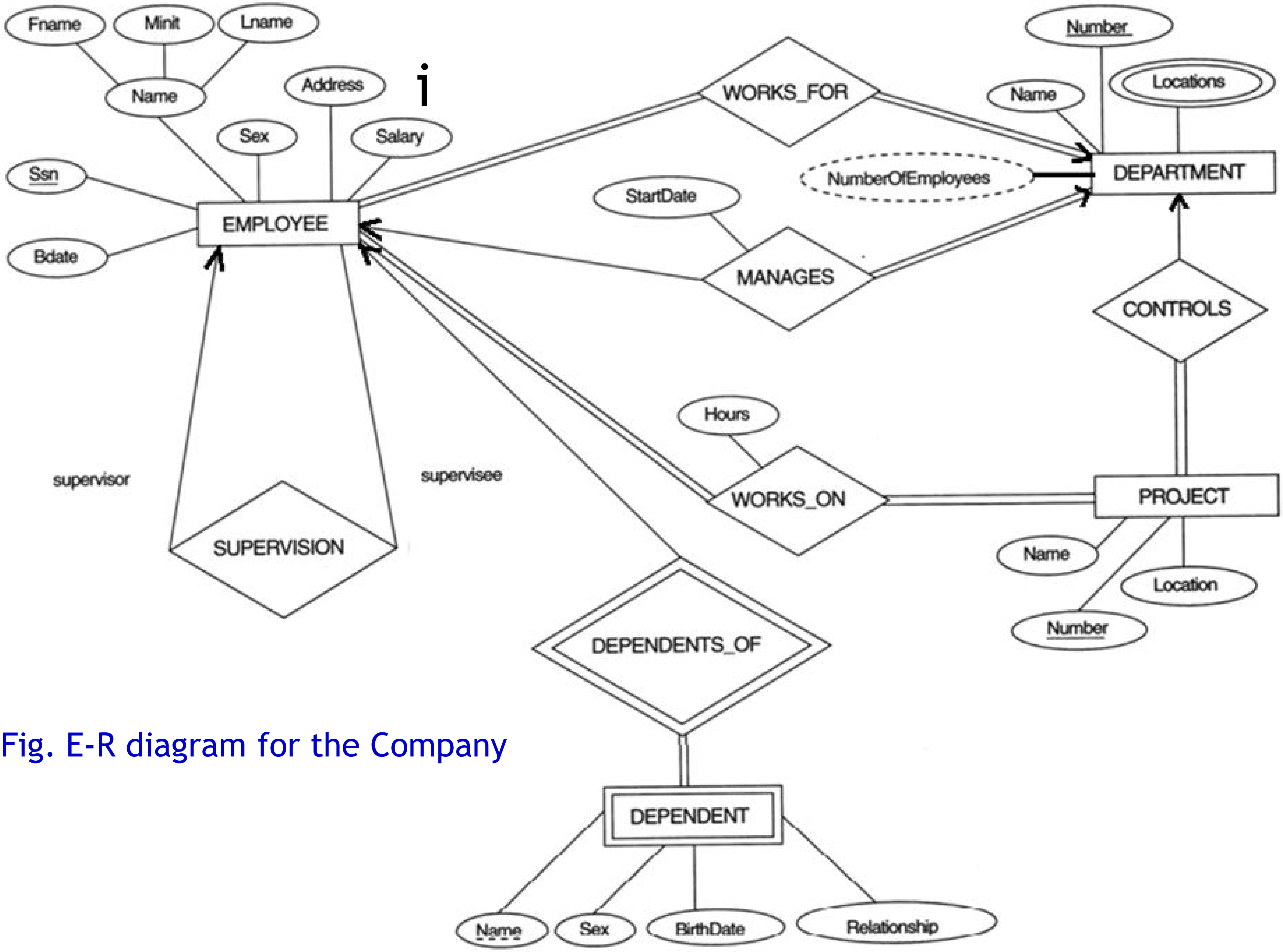


Fig. E-R diagram for the Company

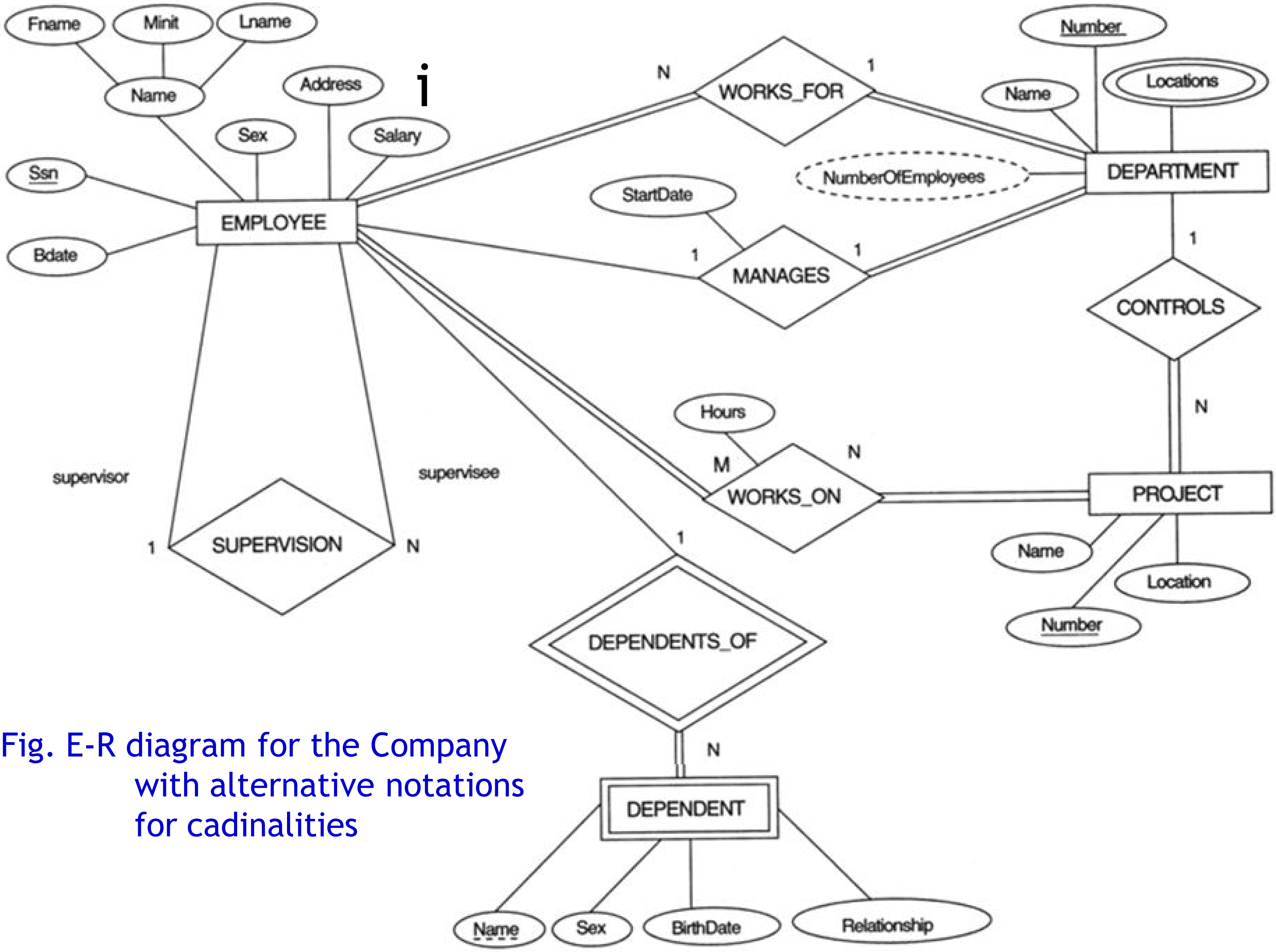


Fig. E-R diagram for the Company
with alternative notations
for cadinalities

Weak Entity Set (1)

An entity set that does not have a primary key is referred to as a weak entity set.

An entity set that has a primary key is termed as a strong entity set.

Every weak entity set must be associated with another entity set, called identifying or owner entity set. Thus, the weak entity set is said to be existence dependent on the identifying entity set. The identifying entity set is said to own the weak entity set that it identifies.

The relationship assocating the weak entity set with the identifying entity set is called the indentifying relationship.

The discriminator (or Partial Key) is the set of attributes that uniquely identify weak entities that are related to the same identifying entity.

The primary key of a weak entity set is formed by the primary key of the strong entity set on which the weak entity set is existence dependent, plus the weak entity set's discriminator.

Weak Entity Set (2)

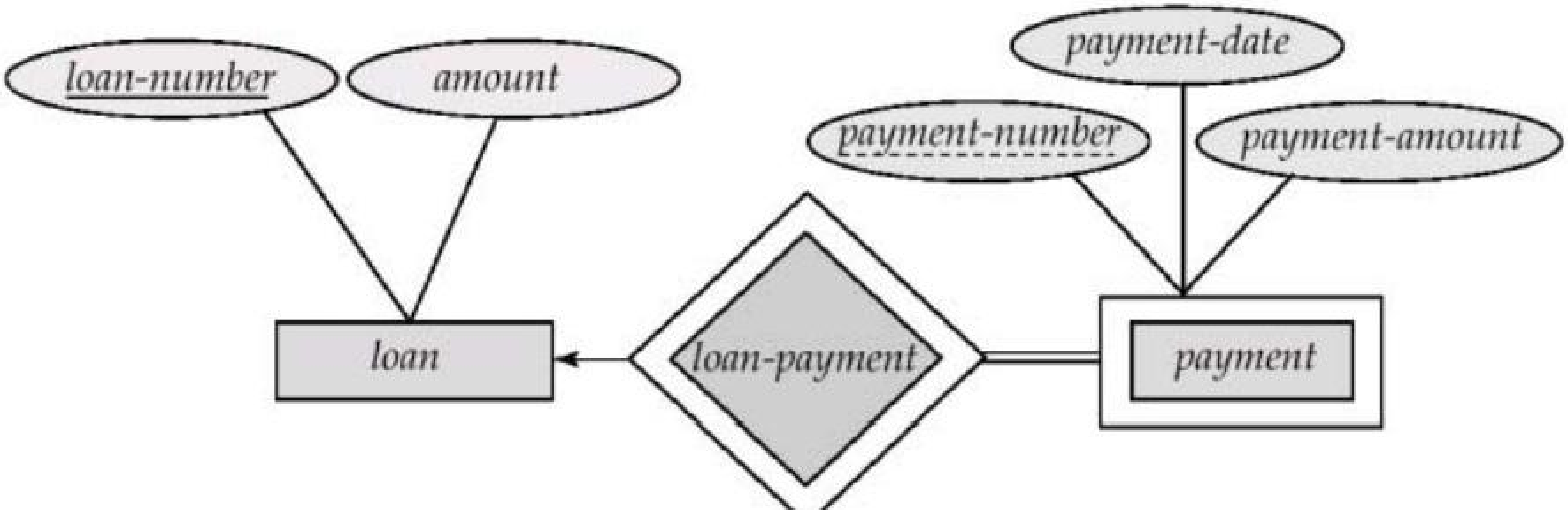
We depict a weak entity set by double rectangles and a identifying relationship set by double diamond.

We underline the discriminator of a weak entity set with a dashed line.

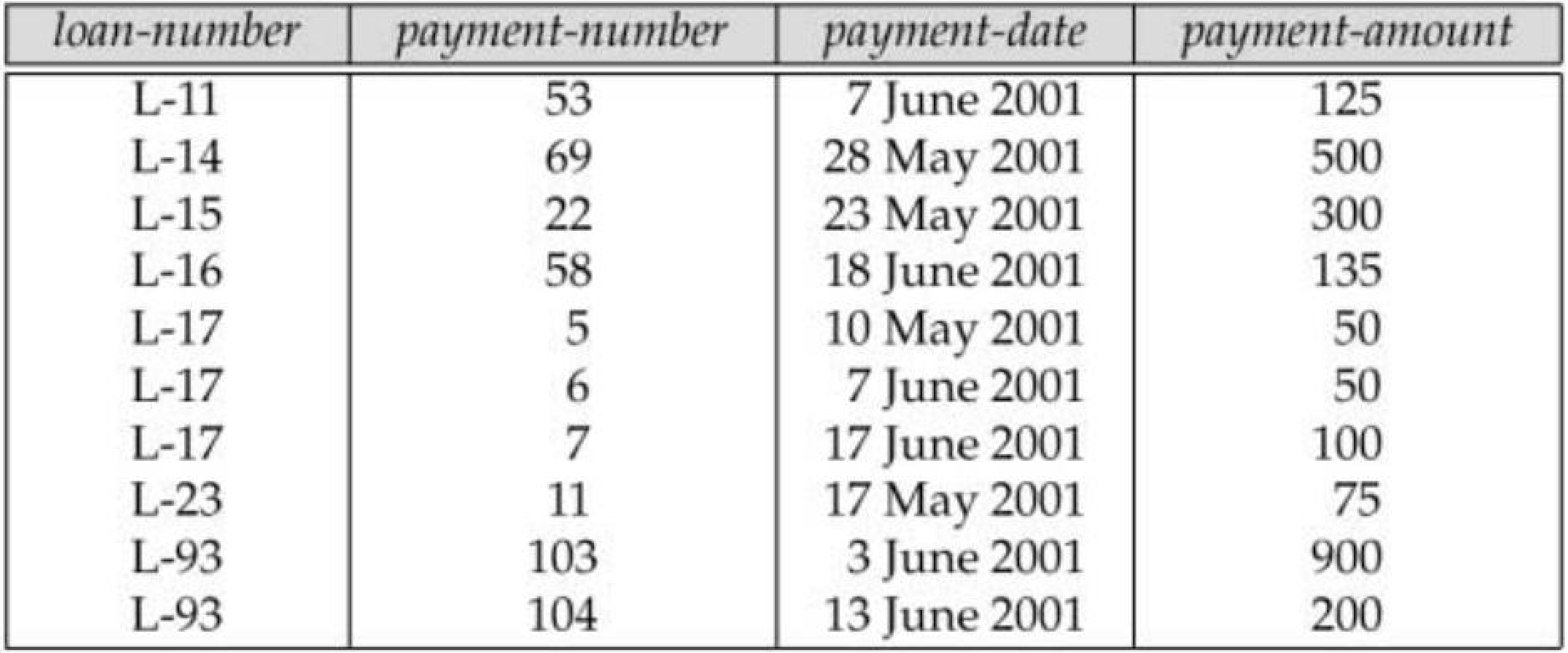
payment-number – discriminator of the *payment entity set*.

Primary key for *payment* – *(loan-number, payment-number)*

It must relate to the identifying entity set via a total, one-to-many relationship set from the identifying to the weak entity set



Weak Entity Set



loan-number	*payment-number*	*payment-date*	*payment-amount*
L-11	53	7 June 2001	125
L-14	69	28 May 2001	500
L-15	22	23 May 2001	300
L-16	58	18 June 2001	135
L-17	5	10 May 2001	50
L-17	6	7 June 2001	50
L-17	7	17 June 2001	100
L-23	11	17 May 2001	75
L-93	103	3 June 2001	900
L-93	104	13 June 2001	200

Fig. Payment Table

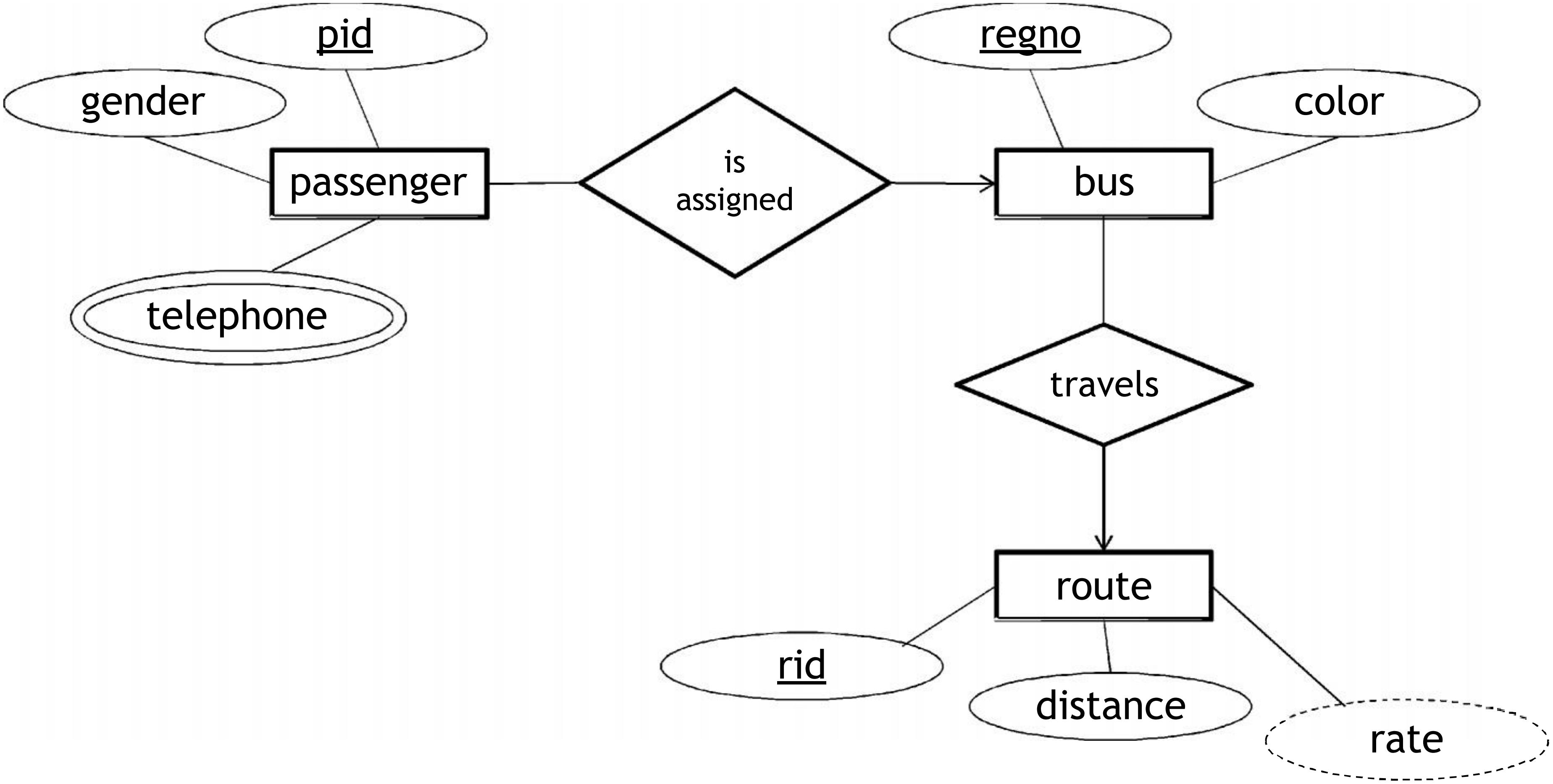
Draw E-R diagram of a bus ticketing system
(Paper 2010)

Consider a bus ticketing system that records information about the passenger, bus and route.

Passenger is assigned to a bus and bus travels to route. A bus contains many passenger and a passenger can be assigned into only one bus. Many bus travels in same route but a bus can travel in only one route.

The attributes of passenger are pid(unique), gender, and telephone(multivalued). Similarly bus contains regno(unique) and color and route contains rid(unique), distance and rate(based on distance).

E-R Diagram for a bus ticketing system



Design Issues

Use of entity sets vs. attributes

- Choice mainly depends on the structure of the enterprise being modeled, and on the semantics associated with the attribute in question

Use of entity sets vs. relationship sets

- Each relation could also be an entity – there's no general criteria.
- Possible guideline is to designate a relationship set to describe an action that occurs between entities

Binary versus n-ary relationship sets

- Although it is possible to replace a non-binary (n-ary, for n>2) relationship set by a number of distinct binary relationship sets, a n-ary relationship set shows more clearly that several entities participate in a single relationship.

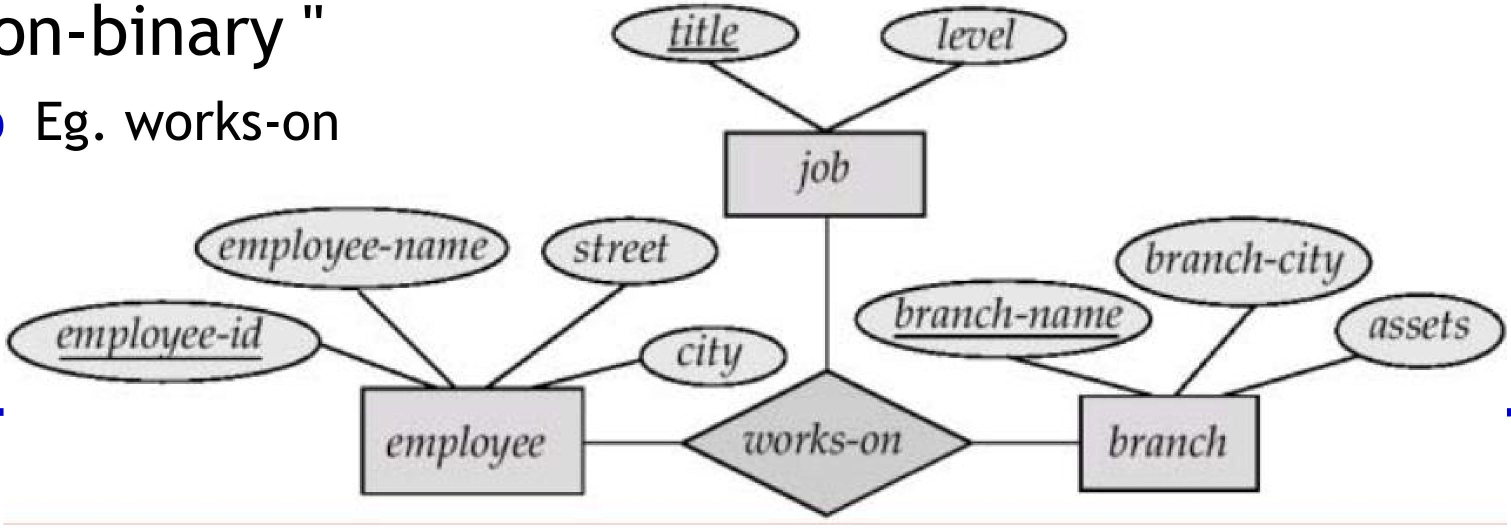
Binary Vs. Non-Binary Relationships (1)

Some relationships that appear to be non-binary may be better represented as binary relationships

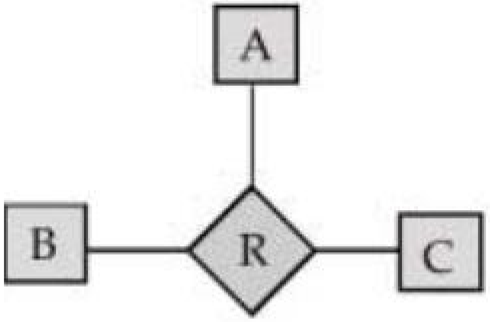
- E.g. A ternary relationship parents, relating child to his/her father and mother, is best replaced by two relationship binary relationship father and mother.
- Using two binary relationships allows partial information (e.g. only mother being known)

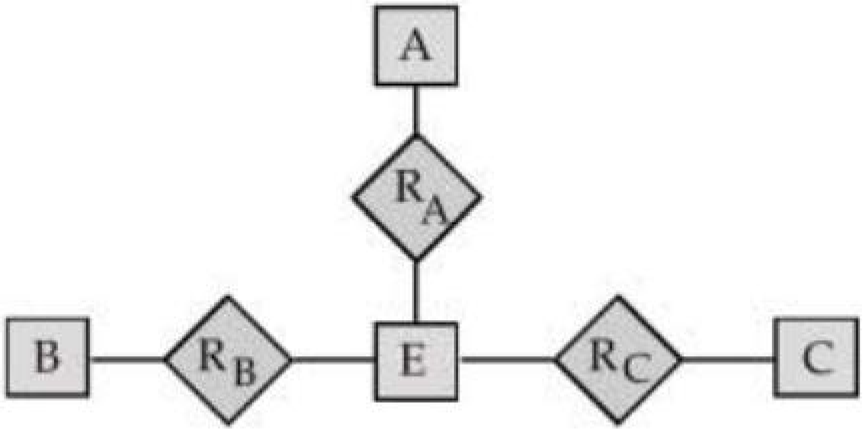
But there are some relationships that are naturally non-binary "

- Eg. works-on

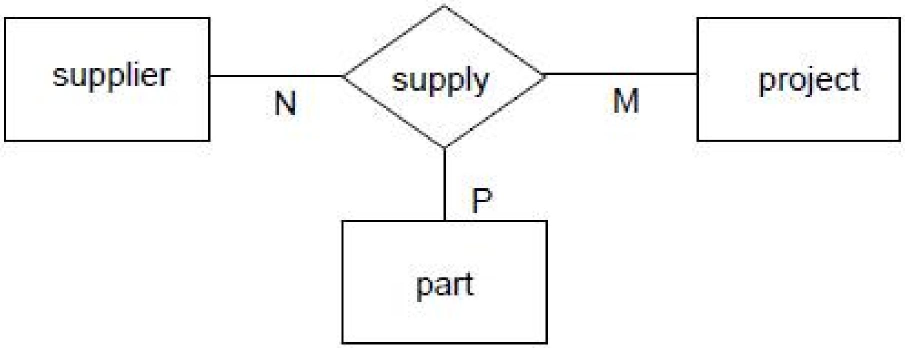


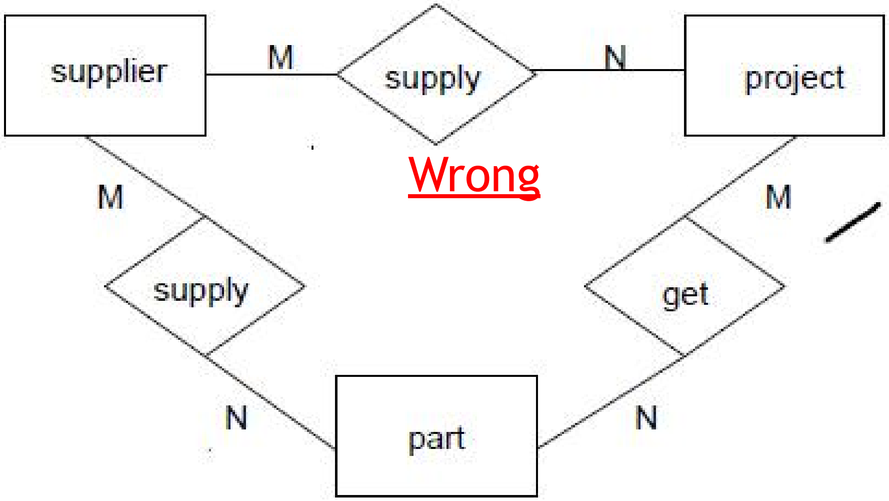
Binary Vs. Non-Binary Relationships (2)





- Consider The ternary relationship "supply"
- Attention: this is not the same as three binary relationship sets!





Wrong

Keys for Relationship Sets (1)

Let *R* be relationship set involving entity sets E_1, E_2..E_n.

Primary Key (E_i) denotes primary key for entity set E_i.

If the relationship set *R* has no attributes associted with it, then the set of attributes

Primary key(E_1) U *Primary key*(E_2) U .. *Primary key*(E_n)

describes an individual relationship in set R.

If the relationship set *R* has attributes a_1, a_2, ...a_n associted with it, then the set of attributes

Primary key(E_1) U *Primary key*(E_2) U.. *Primary key*(E_n) U {a_1, a_2, ..a_n }

describes an individual relationship in set R.

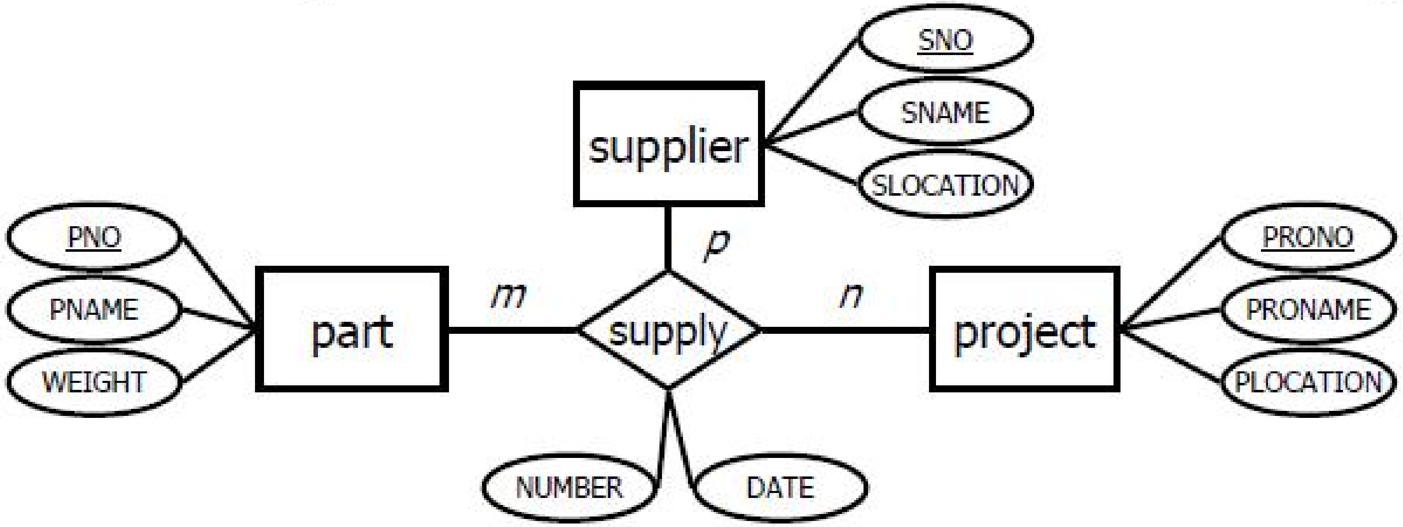
In both cases, the set of attributes

Primary key(E_1) U *Primary key*(E_2) U.. *Primary key*(E_n)

forms a superkey for the relationship set.

The structure of primary key depends on mapping cardinalities.

Keys for Relationship sets (2)



Representation Possibilities in RM:

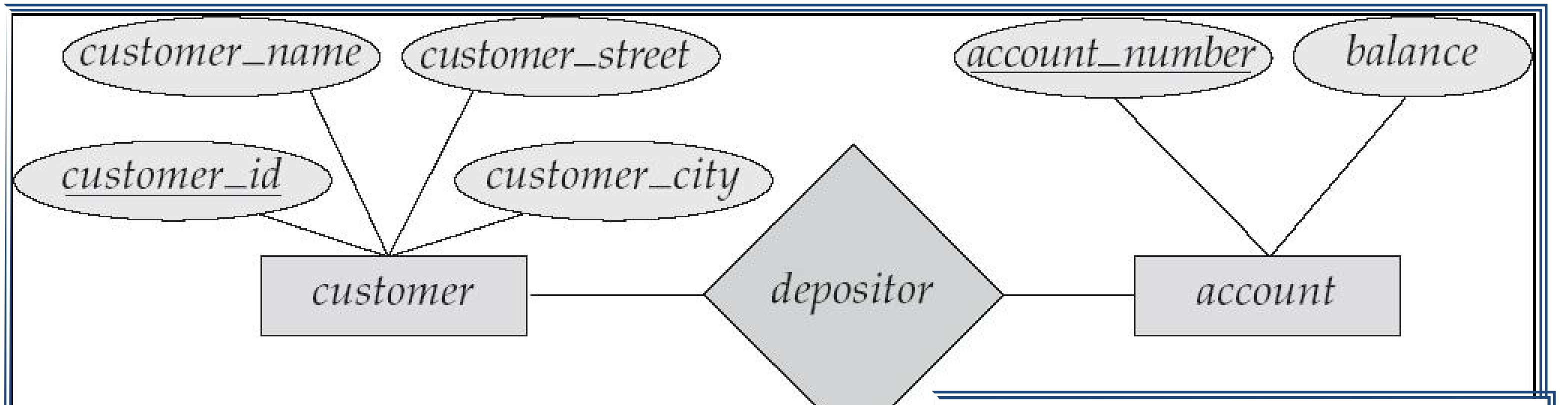
SUPPLIER (<u>SNO</u>, SNAME, SLOCATION)

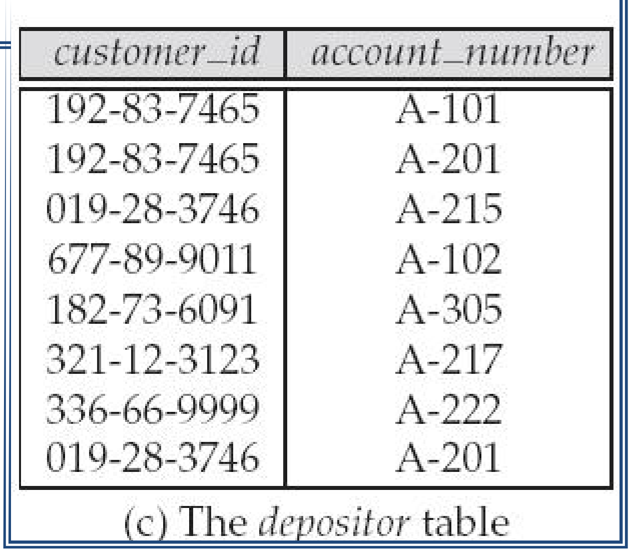
PROJECT (<u>PRONO</u>, PRONAME, PLOCATION)

PART (<u>PNO</u>, PNAME, WEIGHT)

SUPPLY (<u>SNO</u>, <u>PRONO</u>, <u>PNO</u>, NUMBER, DATE)

Keys for Relationship Sets (3)





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

The primary key of depostior in

1. Many to many (customer-account)
   Is union of primary
   keys of customer and account.
2. One to many (customer-account)
   Is primary key of account.
3. Many to One (customer-account)
   Is primary key of customer.
4. One to one (customer-account)
   is anyone of primary keys

Fig. For depositor with many to many, the relational schema for depositor is depositor(customer_id, account_number)

Reduction of an E-R schema to Tables

Primary keys allow entity sets and relationship sets to be expressed uniformly as tables which represent the contents of the database.

A database which conforms to an E-R diagram can be represented by a collection of tables.

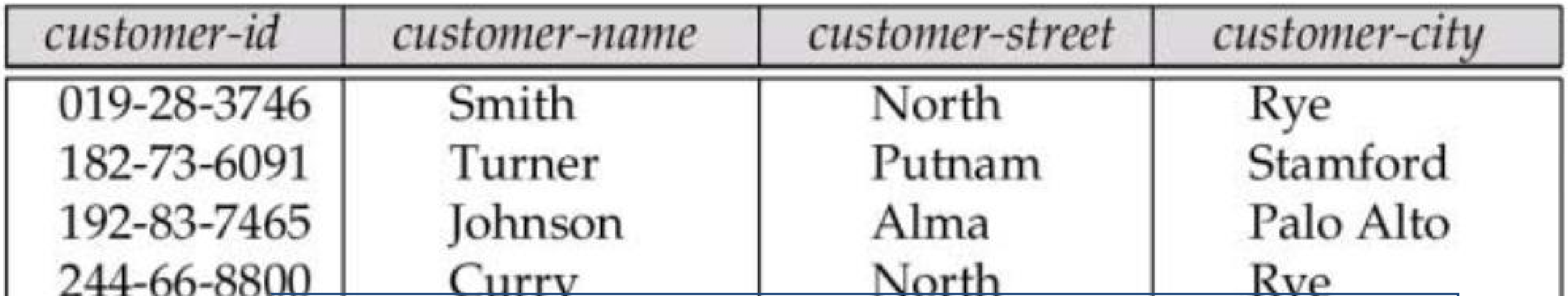
For each entity set and relationship set there is a unique table which is assigned the name of the corresponding entity set or relationship set.

Each table has a number of columns (generally corresponding to attributes), which have unique names.

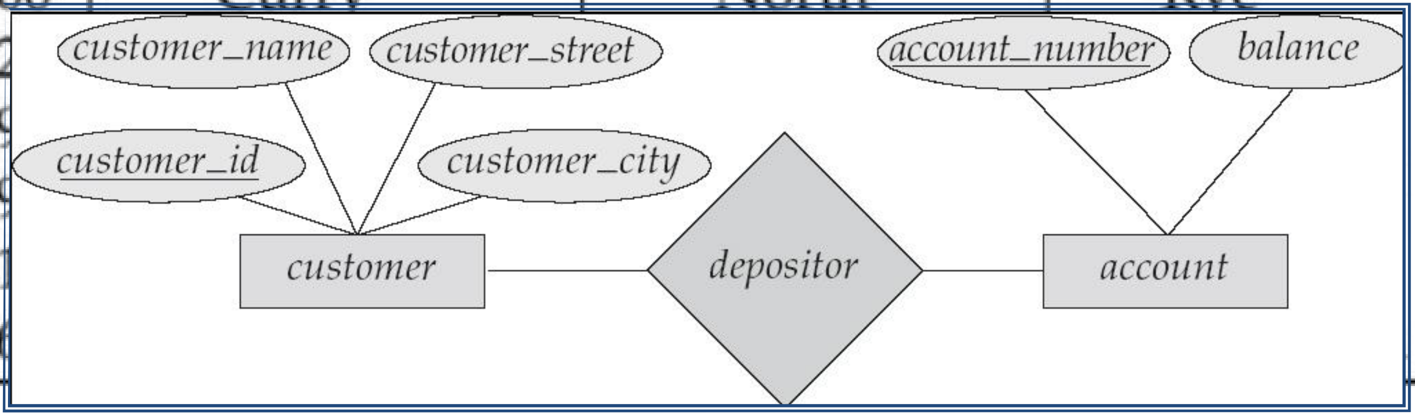
Converting an E-R diagram to a table format is the basis for deriving a relational database design from an E-R diagram.

Representing Strong Entity Sets

A strong entity set reduces to a table with the same attributes.



customer-id	*customer-name*	*customer-street*	*customer-city*
019-28-3746	Smith	North	Rye
182-73-6091	Turner	Putnam	Stamford
192-83-7465	Johnson	Alma	Palo Alto
244-66-8800	Curry	North	Rye
321-12-312			
335-57-79			
336-66-99			
677-89-90			
963-96-39			

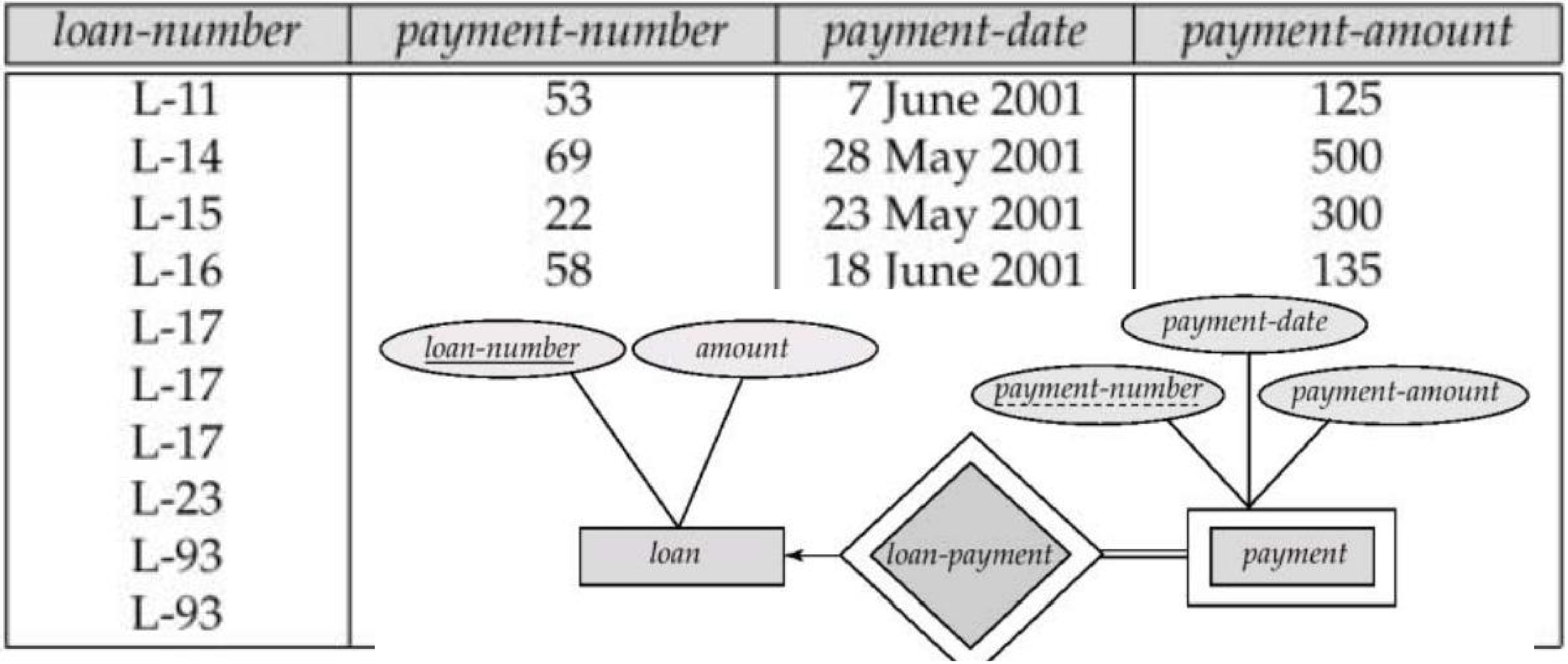


Relational Schemas of strong entity sets are

customer(<u>customer-id</u>, customer_name, customer_stree,customer-city)
account(<u>account_number</u>, balance)

Representing Weak Entity Sets

A weak entity set becomes a table that includes a column for the primary key of the identifying strong entity set



loan-number	payment-number	payment-date	payment-amount
L-11	53	7 June 2001	125
L-14	69	28 May 2001	500
L-15	22	23 May 2001	300
L-16	58	18 June 2001	135
L-17			
L-17			
L-17			
L-23			
L-93			
L-93			

Relational Schema for weak entity set

payment(payment-number,loan-number, payment-date,payment-amount)

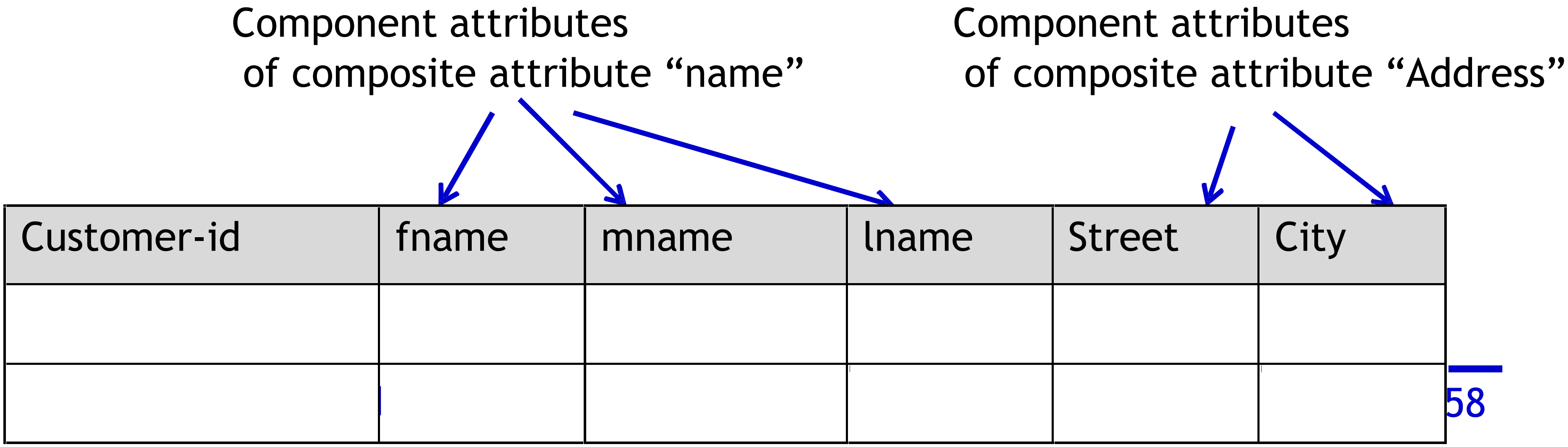
Representing Composite Attributes

Composite attributes are handled by creating a separate attribute for each of the component attributes.

A separate column for the composite attribute itself is not created.

E.g. *address* is a composite attribute of customer and component attributes of address are *street* and *city*.

Hence, *address-street* and *address-city* are also columns for customer table but no column named *address*.



Component attributes of composite attribute “name”

Component attributes of composite attribute “Address”

Customer-id	fname	mname	lname	Street	City

Representing Multivalued Attributes

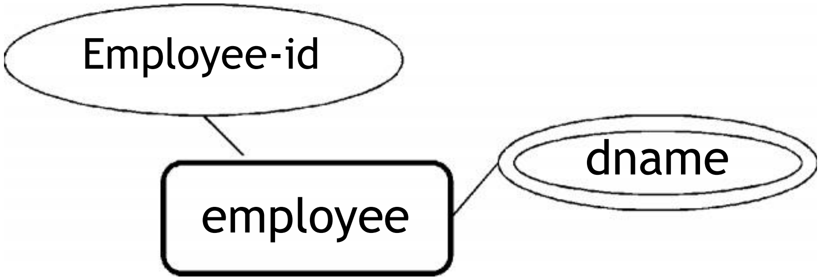
A multivalued attribute M of an entity E is represented by a separate table EM

Table EM has attributes corresponding to the primary key of E and an attribute corresponding to multivalued attribute M.

E.g. Multivalued attribute dependent-names of employee is represented by a table

- employee-dependent-names (employee-id, dname)
- Each value of the multivalued attribute maps to a separate row of the table EM

employee-id	dname
101	Thomas
101	Micheal
102	Bob
103	Alice
104	Jenny



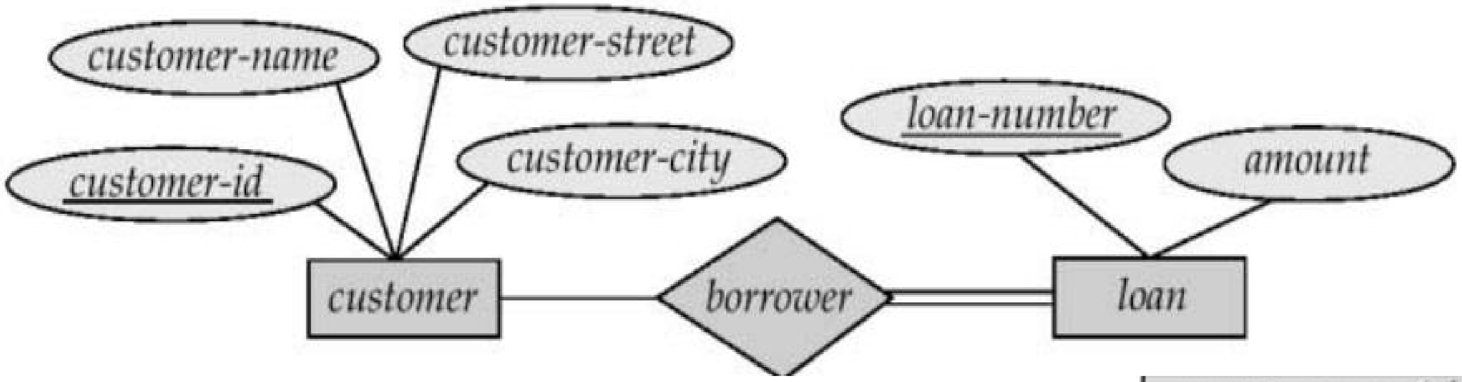
Relational Schemas of multivalued attribute

employee-dependent(<u>employee-id</u>, <u>dname</u>)

Representing Relationship Sets as Tables (1)

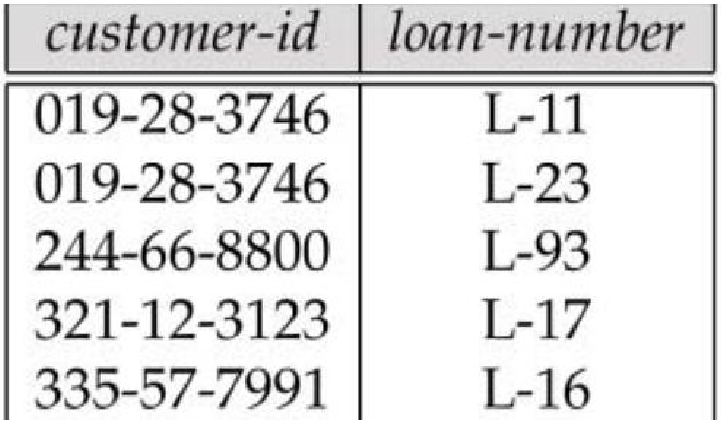
Relationship set is represented as a table with columns for the primary keys of the two participating entity sets, and any descriptive attributes of the relationship set.

E.g.: table for relationship set borrower



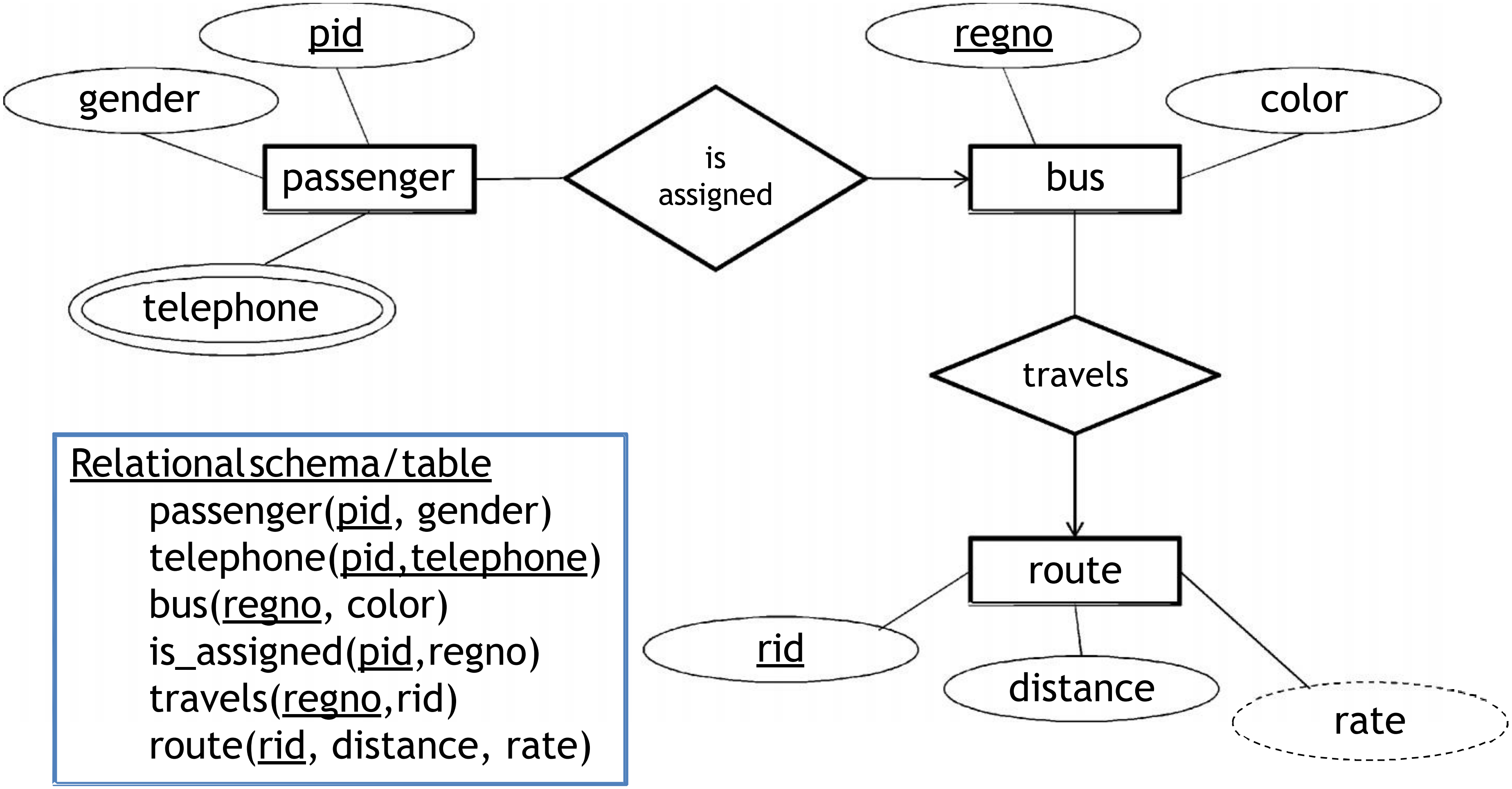
Relational Schemas of relationship borrower

borrower(customer-id,loan-number)



customer-id	loan-number
019-28-3746	L-11
019-28-3746	L-23
244-66-8800	L-93
321-12-3123	L-17
335-57-7991	L-16

Representing Relationship Sets as Tables (2)



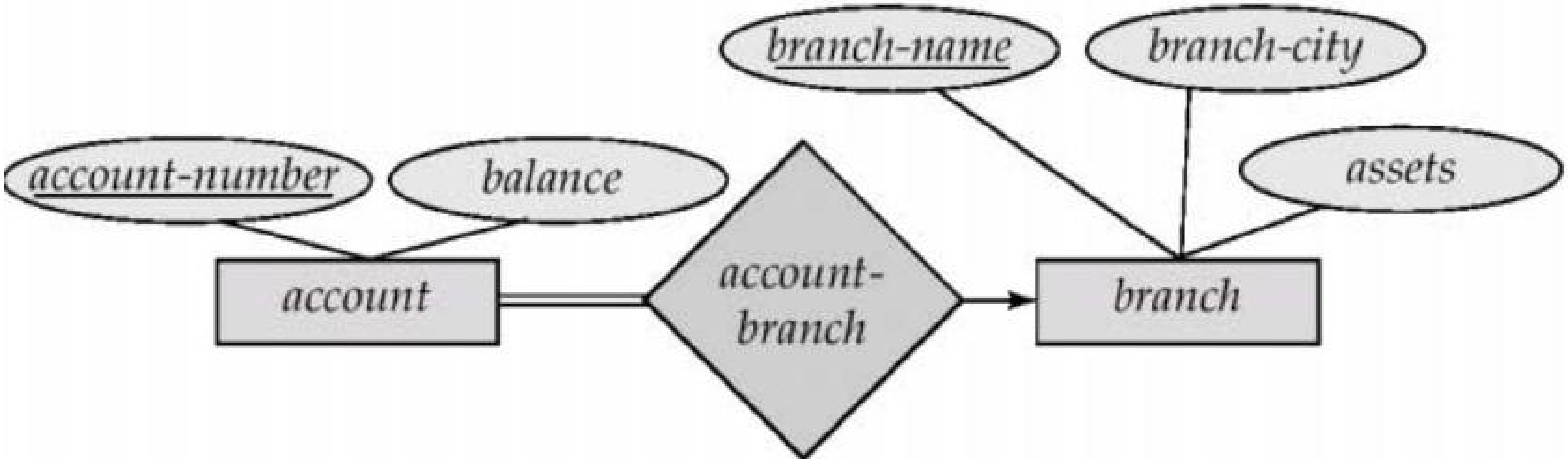
Relationalschema/table

passenger(pid, gender)
telephone(pid,telephone)
bus(regno, color)
is_assigned(pid,regno)
travels(regno,rid)
route(rid, distance, rate)

Combination of Tables (1)

Many-to-one and one-to-many relationship sets that are total on the many-side

- can be represented by adding an extra attribute to *the many side*, containing the primary key of the *one side*.
- E.g.: Instead of creating a table for relationship account-branch, add an attribute branch-name into account entity set.



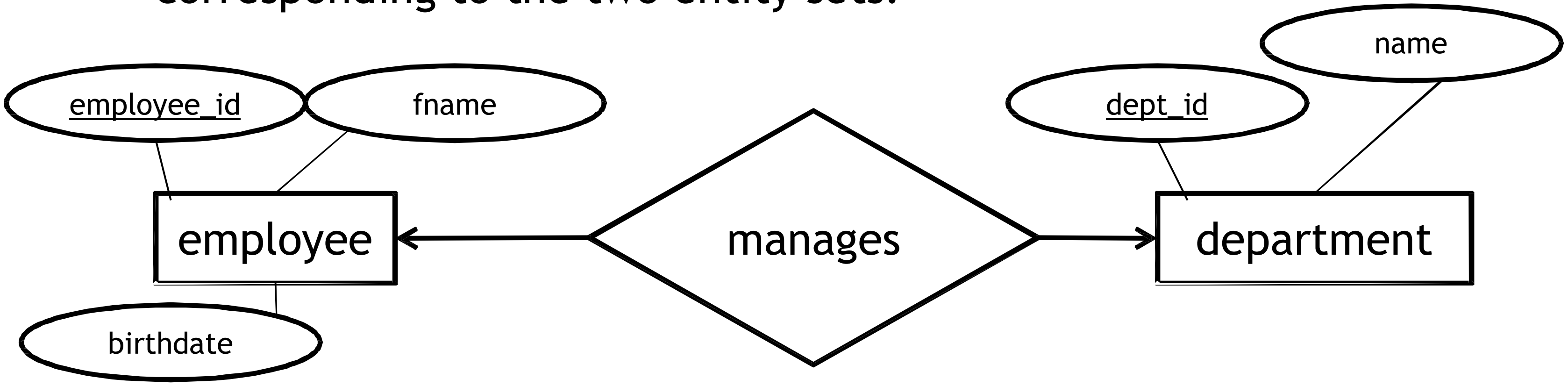
Relational Schemas above E-R diagram are

account(<u>account-number</u>,balance,branch-name)
branch(<u>branch-name</u>,branch-city, assets)

Combination of Tables (2)

For one-to-one relationship sets, either side can be chosen to act as the “many” side

- That is, extra attribute can be added to either of the tables corresponding to the two entity sets.



1st possible relational schemas of above E-R diagram

employee(employee_id, fname, birthdate, dept_id)
department(dept_id,name)

2nd possible relational schema of above E-R diagram

employee(employee_id, fname, birthdate)
department(dept_id,name, employee_id)

Combination of Tables (3)

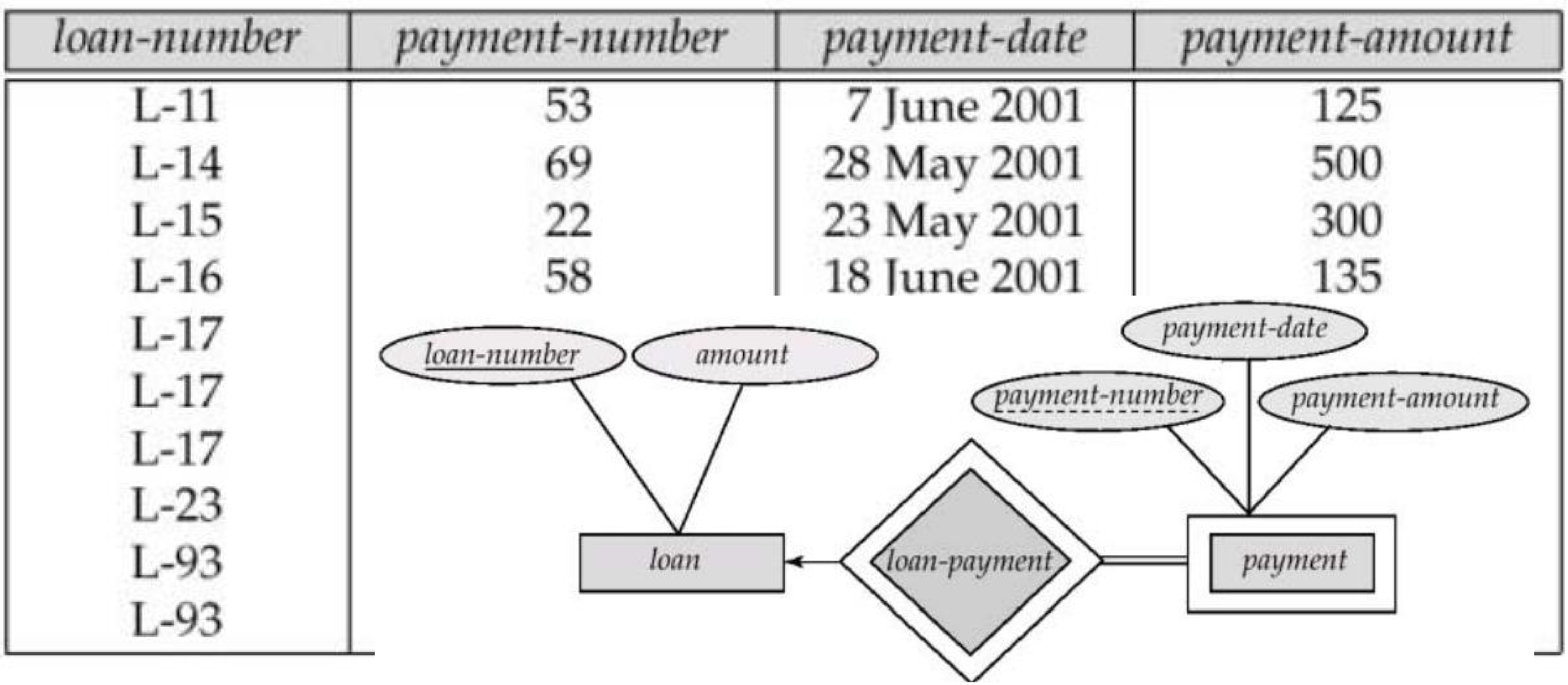
If participation is partial on the many side, replacing a table by an extra attribute in the relation corresponding to the “many” side could result in null values

For many to many relationship, the relationship set must be replaced a relationship table.

Redundancy of Tables

The table corresponding to a relationship set linking a weak entity set to its identifying strong entity set is redundant.

E.g.: The table payment already contains the information that would appear in the loan-payment table (i.e., the columns loan-number and payment-number).



loan-number	payment-number	payment-date	payment-amount
L-11	53	7 June 2001	125
L-14	69	28 May 2001	500
L-15	22	23 May 2001	300
L-16	58	18 June 2001	135
L-17			
L-17			
L-17			
L-23			
L-93			
L-93			

Draw E-R Diagram from following relational schema

Example 1.

person (driver-id, name, address)
car (car-id, year , model)
accident (report-number, date, location)
participated(driver-id, car-id, report-number, damage-amount)
owns(driver-id, car-id)

Example 2.

Computer(**C_no**, price)
Student(**Roll**, name, grade)
Faculty(**Fname**, credit_hours)
Uses(**C_no, Roll**)
Belongs(**C_no**, **Fname**)

Extended E-R Features

Specialization

It is the result of taking a subset of a higher-level entity set to form a lower level entity set.

Top-down design process; we designate subgroupings within an entity set that are distinctive from other entities in the set. The process of designating subgrouping of an entity set is called specialization.

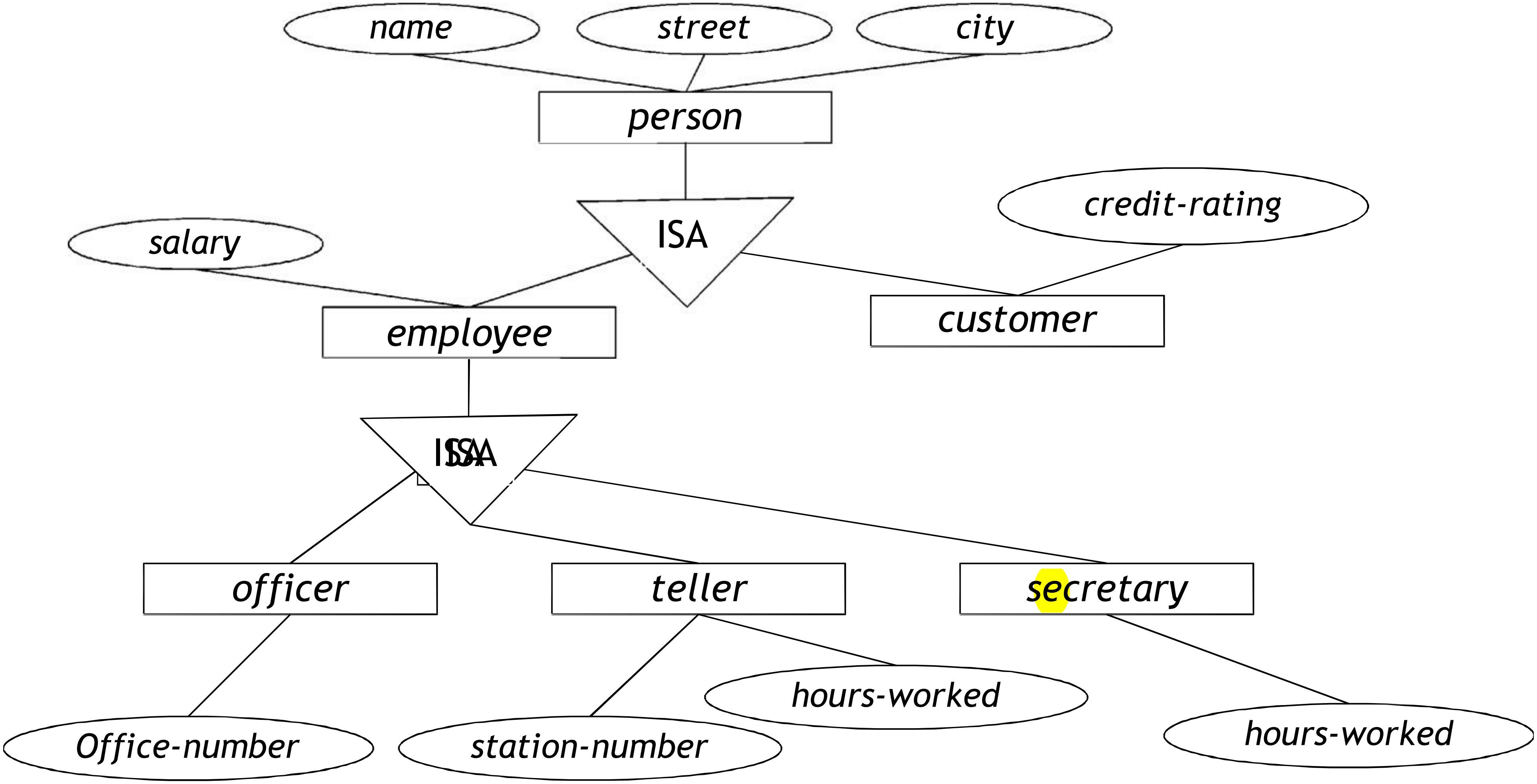
These subgroupings become lower-level entity sets that have attributes or participate in relationships that do not apply to the higher-level entity set

Depicted by a triangle component labeled ISA

- (E.g. Customer “is a” person).

Attribute inheritance – a lower-level entity set inherits all the attributes and relationship participation of the higher-level entity set to which it is linked.

Example: Specialization and Generalization



Generalization

It is a result of taking the union of two or more lower-level entity sets to form a higher-level entity set.

A bottom-up design process – combine a number of entity sets that share the same features into a higher-level entity set.

Specialization and generalization are simple inversions of each other; they are represented in an E-R diagram in the same way.

The terms specialization and generalization are used interchangeably.

Specialization and Generalization (Contd..)

Can have multiple specializations of an entity set based on different features.

E.g. Permant-employee vs. Temporary employee, in addition to officer vs teller vs secretary.

Each particular employee would be

- a member of one of temporary-employee or permanent-employee,
- And also a member of one of officer, secretary, or teller.

ISA relationship also referred to as superclass-subclass relationship

Design Constraints on a Specialization/Generalization (1)

Constraintonwhichentitiescanbemembersofagivenlower-level entityset

- condition-defined: database designer can define condition to lower level entity set that must follow entities in higher level entity set of lower entity set.
  - E.g. all customers over 65 years are members of senior-citizen entity set; senior-citizen ISA person
- User defined: It does not enforce any membership condition in lower level entity set. Database user assigns entities into other entity sets.

Design Constraints on a Specialization/Generalization (2)

Constraint on whether or not entities may belong to more than one lower-level entity set within a single generalization.

- Disjoint
  - an entity can belong to only one lower-level entity set
  - Noted in E-R diagram by writing disjoint next to the to the ISA triangle
- Overlapping
  - an entity can belong to more than one lower-level entity set

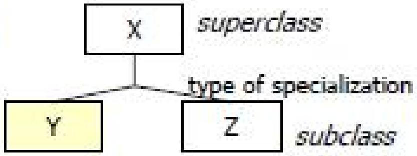
Design Constraints on a Specialization/Generalization (2)

Completeness constraint: specifies whether or not an entity in the higher-level entity set must belong to at least one of the lower-level entity sets within a generalization.

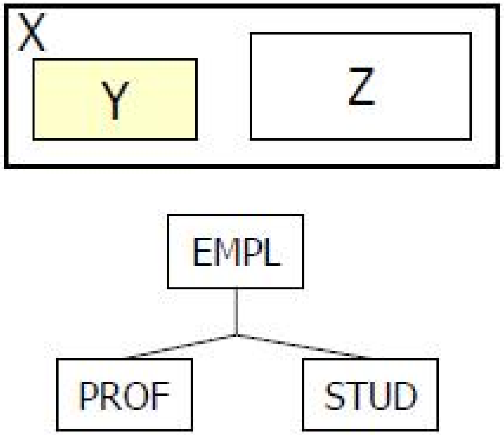
Total: an entity must belong to one of the lower-level entity sets

Partial: an entity need not belong to one of the lower-level entity

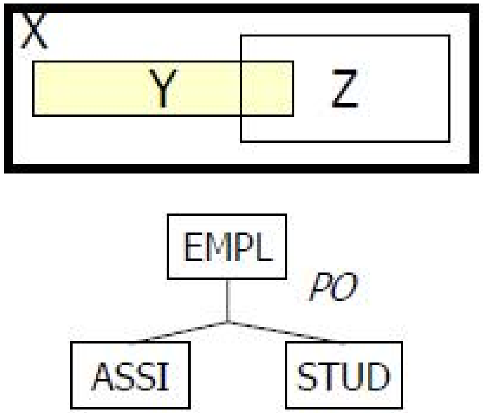
Types of Specialization



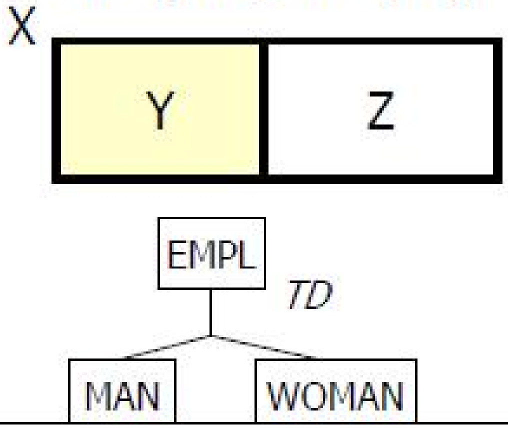
1. partial, disjoint (PD)



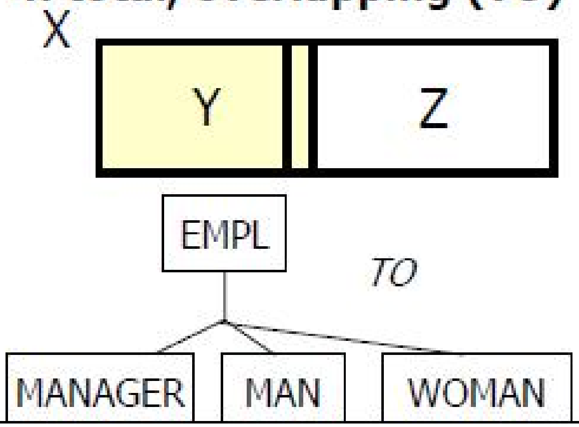
2. partial, overlapping (PO)



3. total, disjoint (TD)



4. total, overlapping (TO)



Aggregation (1)

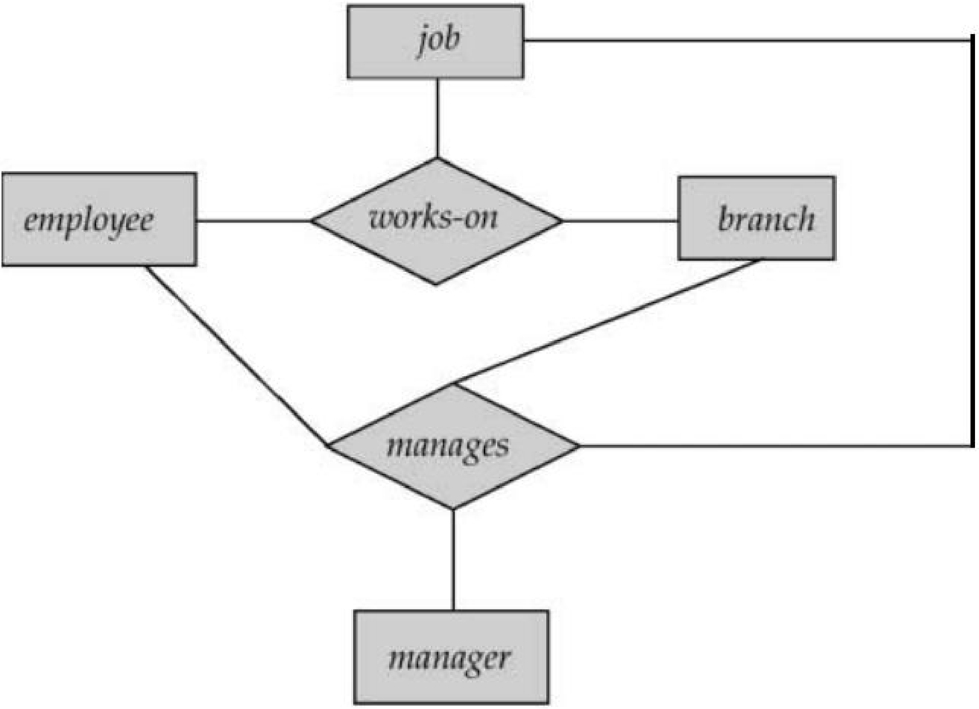
E-R model does not allow relationships among relationships.

For such situation, use aggregation.

Aggregation is an abstraction in which relationship sets are treated as higher entity sets and can participate in relationships.

Consider the ternary relationship *works-on*

Suppose we want to record managers for tasks performed by an employee at a branch.



Aggregation (2)

Relationship sets works-on and manages represent overlapping information

- Every manages relationship corresponds to works-on relationship
- However, some works-on relationships many not corresponds to any manages relationships
- So we can‘t discard the work-on relationship

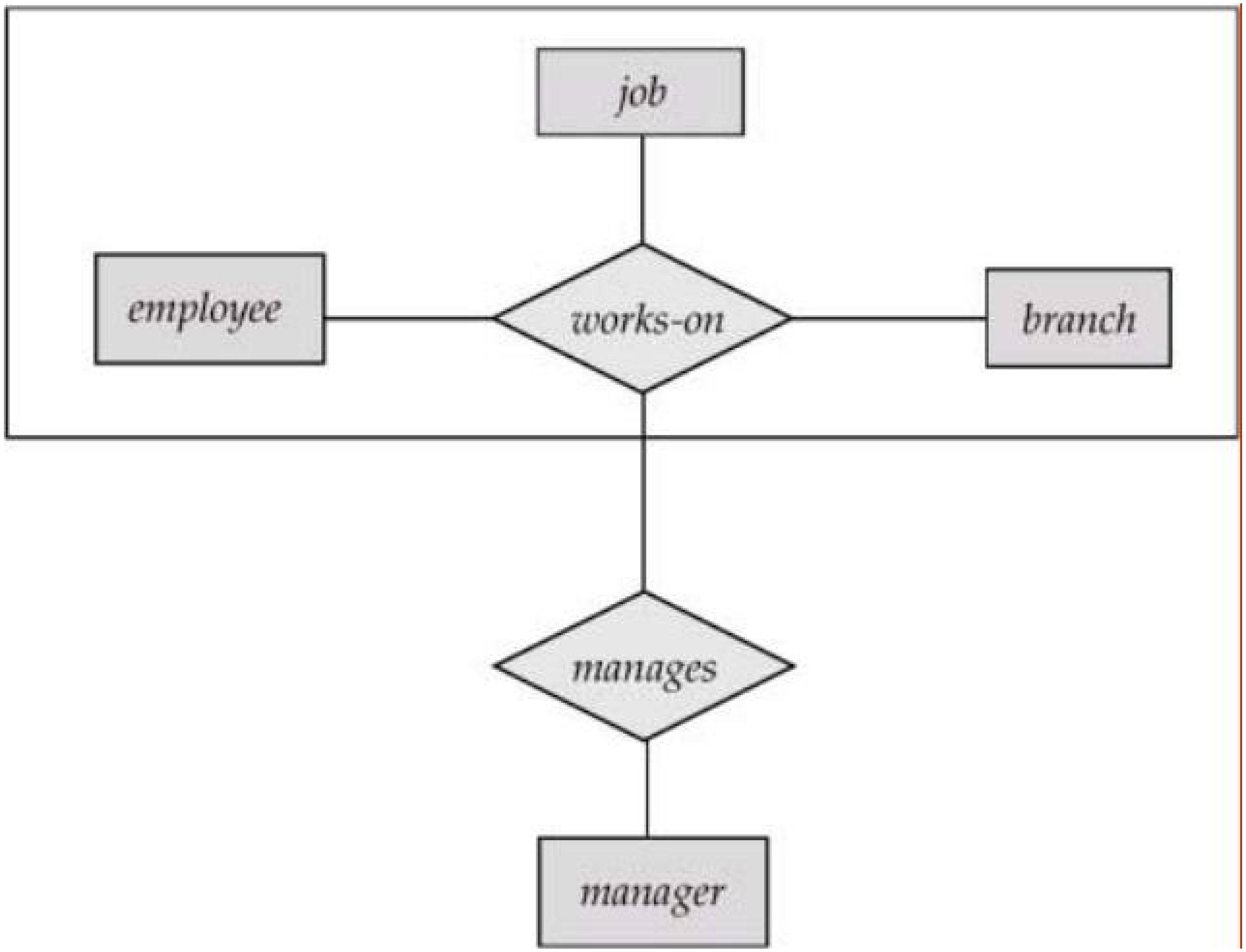
Eliminate this redundancy via aggregation

- Treat relationship as an abstract entity
- Allows relationships between relationships
- Abstraction of relationship into new entity

Without introducing redundancy, the following diagram represents

- An employee works on a particular job at a particular branch
- An employee, branch, job combination may have an associated manager.

E-R diagram with Aggregation



Tabular representation of Generalization

Two different methods of transforming to a tabular form

First Method

- Create a table for the higher-level entity set.
- Create tables for lower entity sets along with primary key of higher entity set.

Example: Account(account-number, balance)
Saving-Account(account-number, interest-rate)
Current-Account(account-number, overdraft-amount)

Second Method: if the generalization is disjoint and complete

- Do not create table for higher-level entity set
- Create tables for lower entity set along with primary key of higher entity set and other attributes of higher of entity set.

Example: Saving-Account(account-number, interest-rate, balance)
Current-Account(account-number, overdraft-amount, balance)

E-R Design Decisions

The use of an attribute or entity set to represent an object.

Whether a real-world concept is best expressed by an entity set or a relationship set.

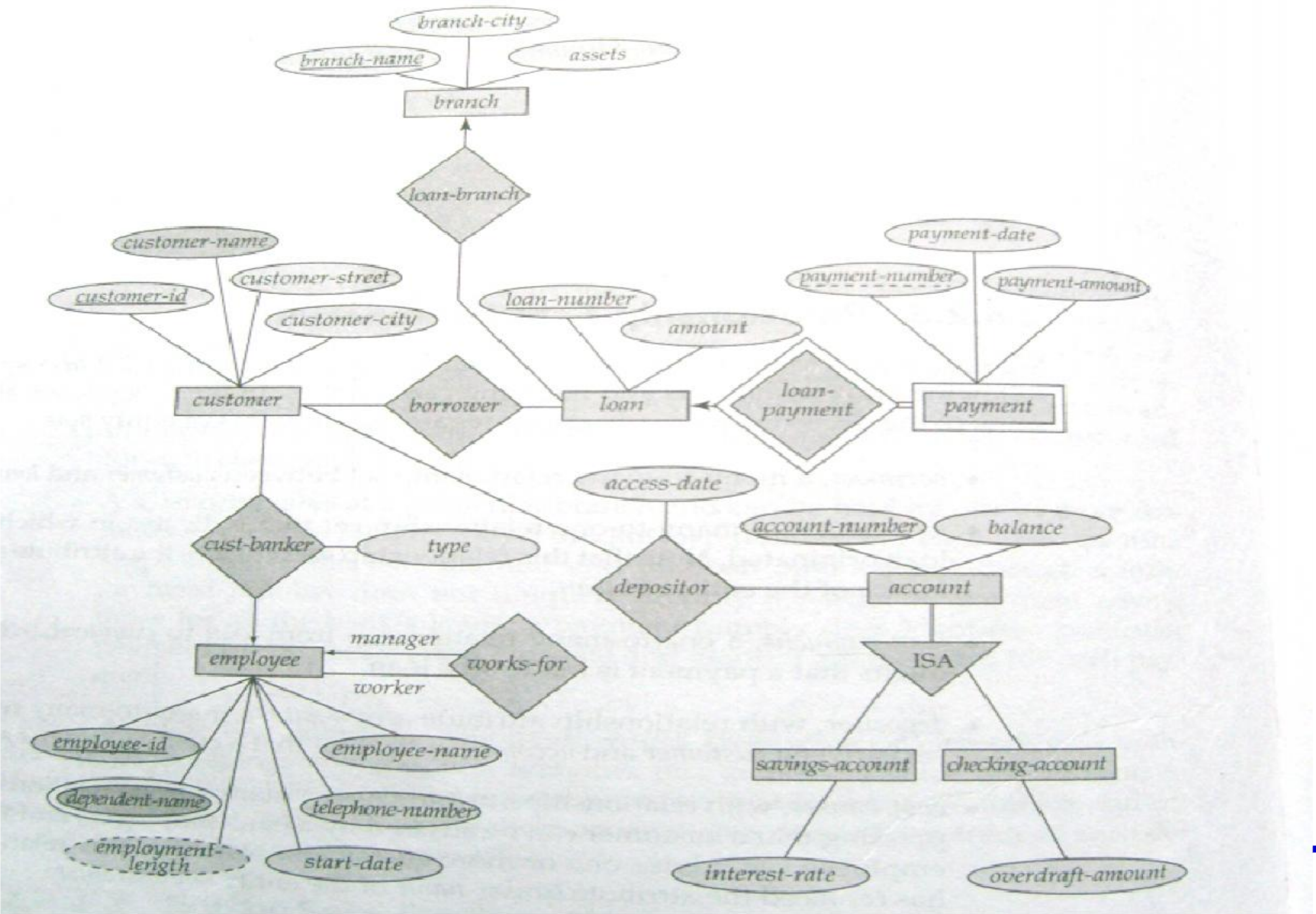
The use of a ternary relationship versus a pair of binary relationships.

The use of a strong or weak entity set.

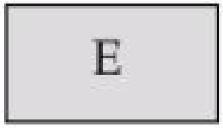
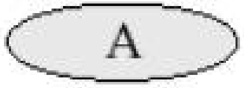
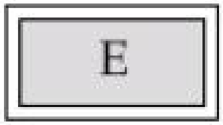
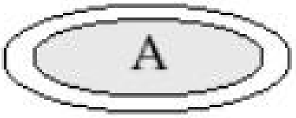
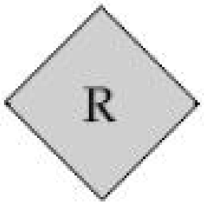
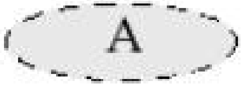
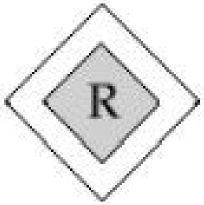
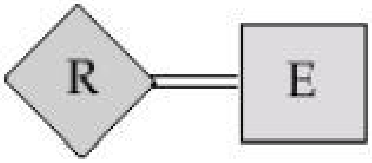
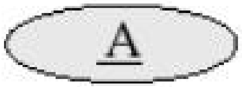
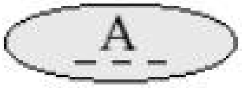
The use of specialization/generalization – contributes to modularity in the design.

The use of aggregation – can treat the aggregate entity set as a single unit without concern for the details of its internal structure.

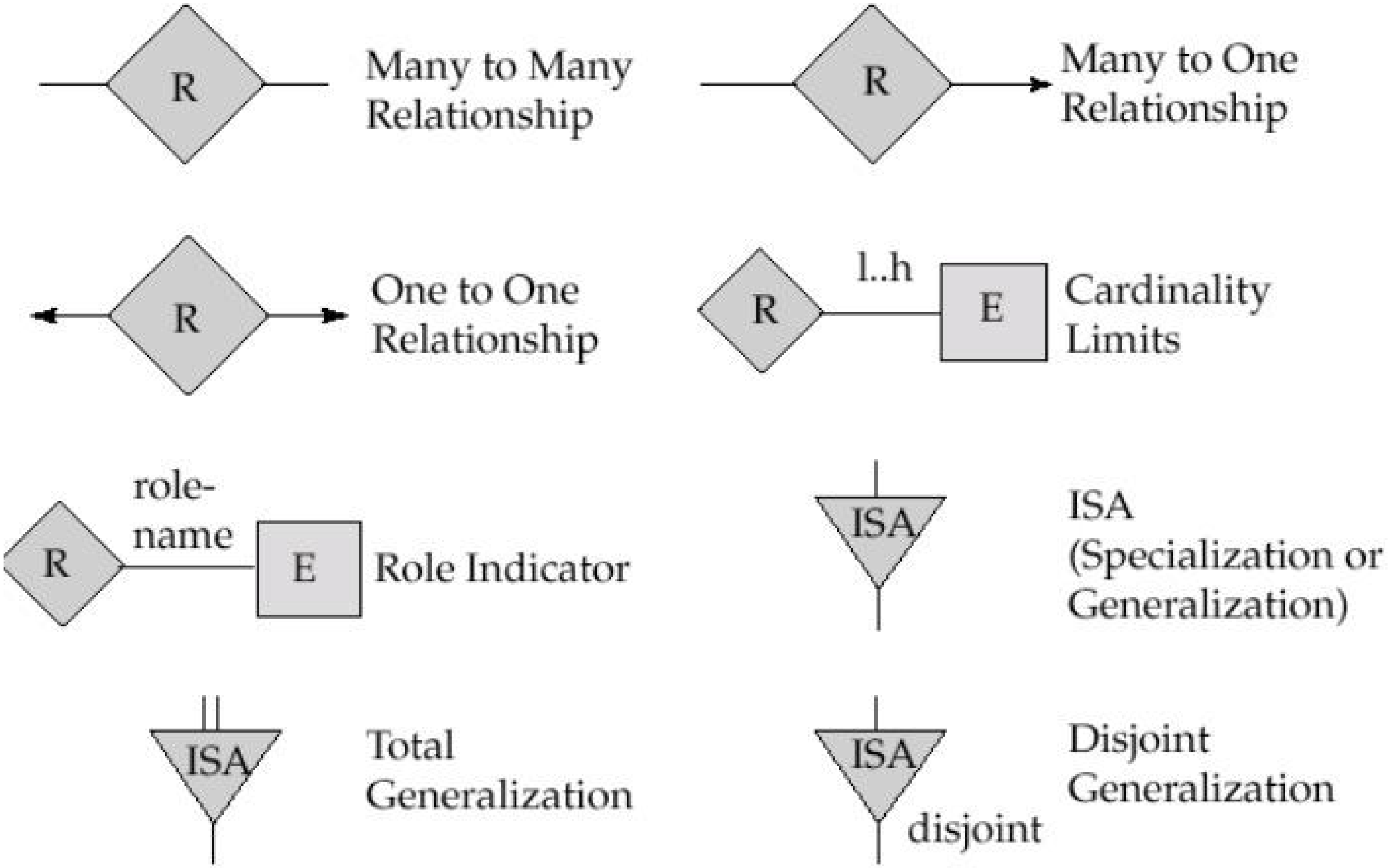
E-R diagram for Bank Enterprise (See book)



Summary of Symbols Used in E-R Notation (1)

	Entity Set		Attribute
	Weak Entity Set		Multivalued Attribute
	Relationship Set		Derived Attribute
	Identifying Relationship Set for Weak Entity Set		Total Participation of Entity Set in Relationship
	Primary Key		Discriminating Attribute of Weak Entity Set

Summary of Symbols Used in E-R Notation (2)



Many to Many Relationship

Many to One Relationship

One to One Relationship

Cardinality Limits

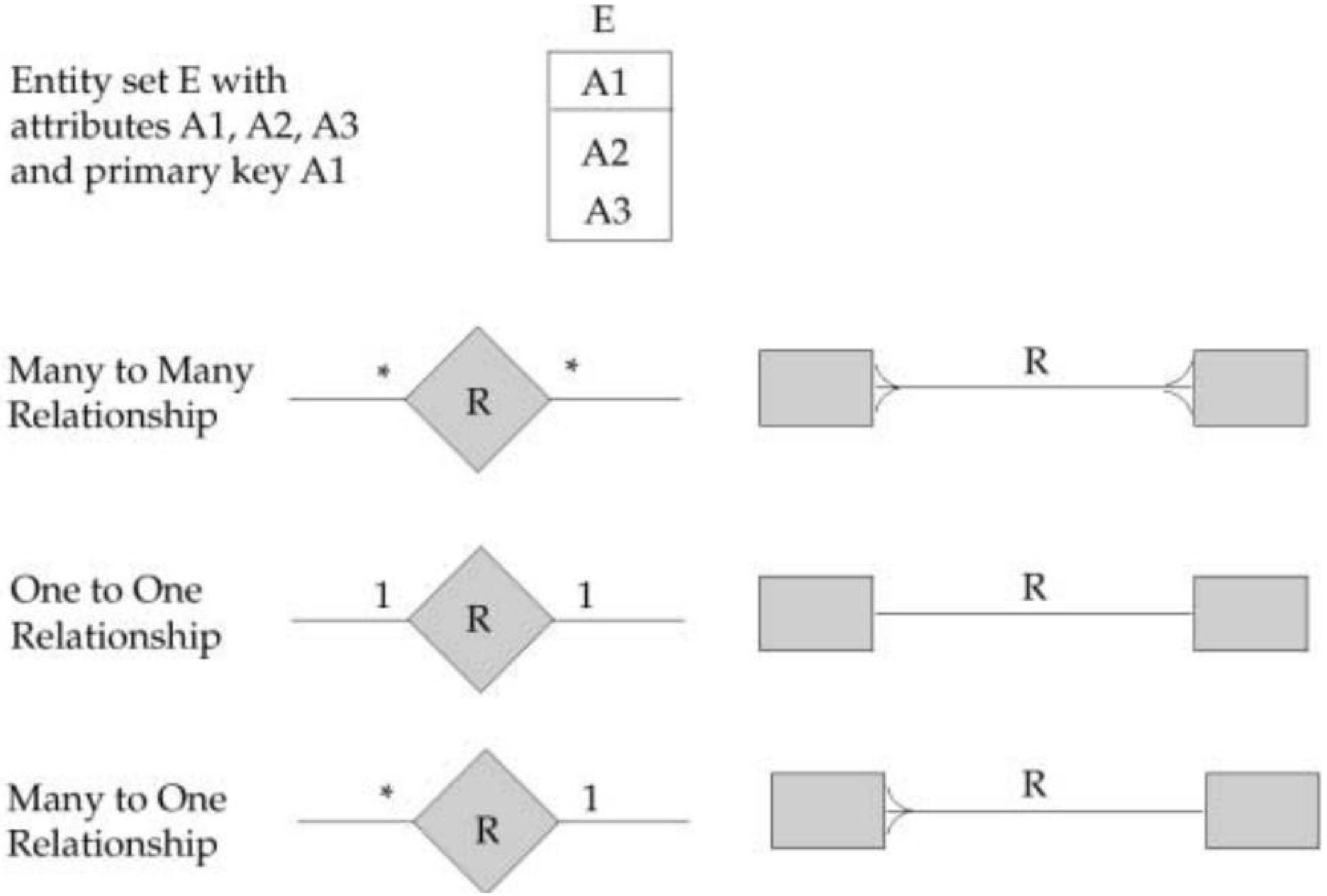
Role Indicator

ISA (Specialization or Generalization)

Total Generalization

Disjoint Generalization

Alternative E-R Notations



Entity set E with attributes A1, A2, A3 and primary key A1

Many to Many Relationship

One to One Relationship

Many to One Relationship