BITS, PILANI – K. K. BIRLA GOA CAMPUS

Database Systems

(CS F212)

by

Dr. Shubhangi

Dept. of CS and IS

Chapter 2: Entity-Relationship Model

- Entity Sets
- Relationship Sets
- Design Issues
- Mapping Constraints
- Participation Constraints
- Keys
- E-R Diagram
- Design of an E-R Database Schema

Revision

- Data abstraction
  - Physical level
  - Logical level
  - View level
- Data Independence
  - Physical data independence
  - Logical data independence
- Instances and Schema
- Database users
  - Naïve user
  - Sophisticated user
  - Specialized user
  - Application programmers
  - Database administrator
- Overall system structure

Data model

- A data model is the collection of tools for describing data, data relationship and consistency constraints.

- Data models:
  - ➢ER model- high level data model
  - ➢Relational model- low level data model
  - ➢Hierarchical model
  - ➢Network model
  - ➢Object oriented model
  - ➢Object relational model

ER diagram

- When to draw?
- Why to draw?
- What to draw?
- How to draw?

ER model

- **E**ntity **R**elationship model
- The ER model perceives the real world as consisting of basic objects, called **entities,** and **relationships** among these objects.
- **Entity:** An entity is an object that exists and is distinguishable from other objects.
  - Example: specific person, company, event, plant

ENTITY SET

- **Entity set:** 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

Person(uid, name, address, phone_no)

Classroom(lecture_hall_no, capacity, no_of_benches, no_of_boards)

ATTRIBUTES

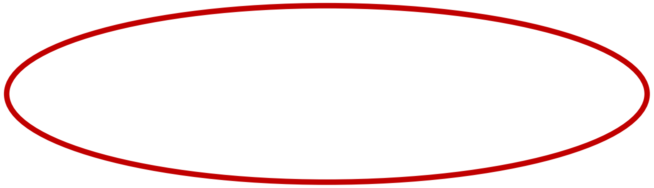
Mapping Entity set to Relation (table)

- Name of the entity set is name of the table

ATTRIBUTE

- 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)

account = (account _no, balance_amt)

loan = (loan-number, amount)

Notation: Ellipse

ATTRIBUTES (Contd…)

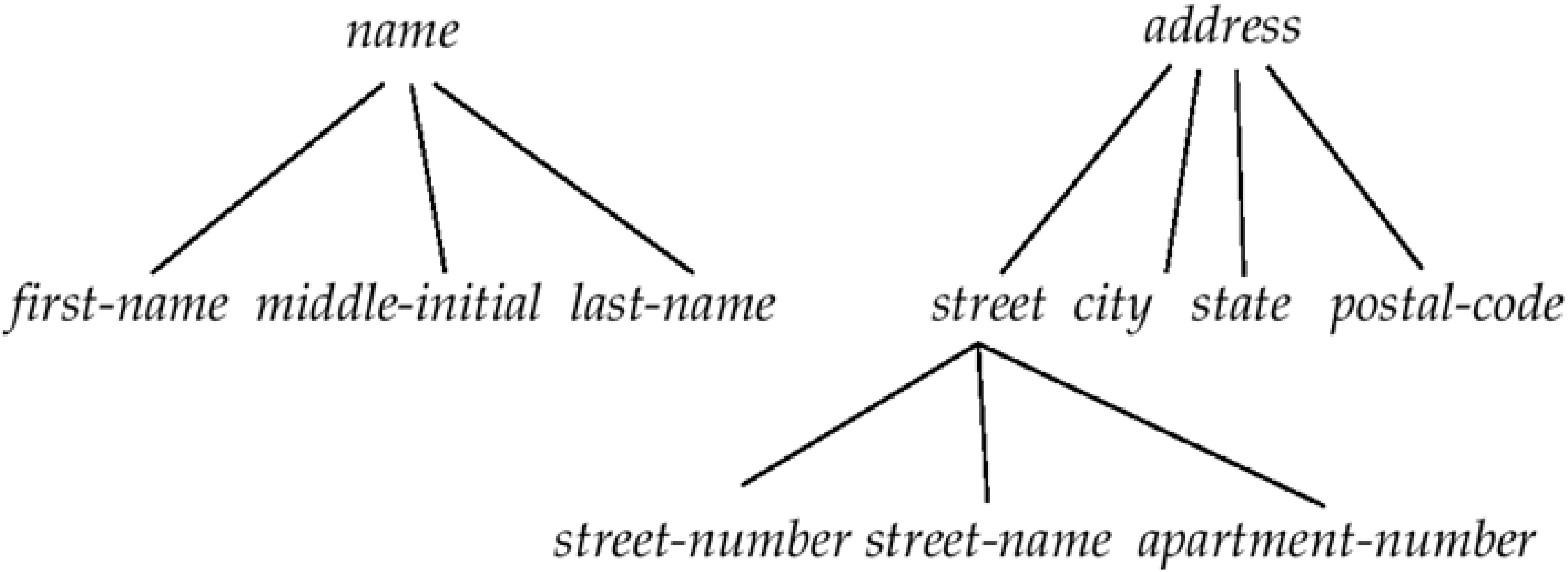
- **Domain/value set –** the set of permitted values for each attribute
- Eg:
  - size of shirt { L, XL, XXL }
  - id {alphanumeric values}
  - gender should be either **‘M’** or **‘F’**
  - course_no should start with **‘AAOC’**, **‘BITS’**, **‘CS’**, …..
  - room_no should be between **1** to **300**

Mapping simple attributes to relation

- Each attribute of an entity set maps to each column of the corresponding table

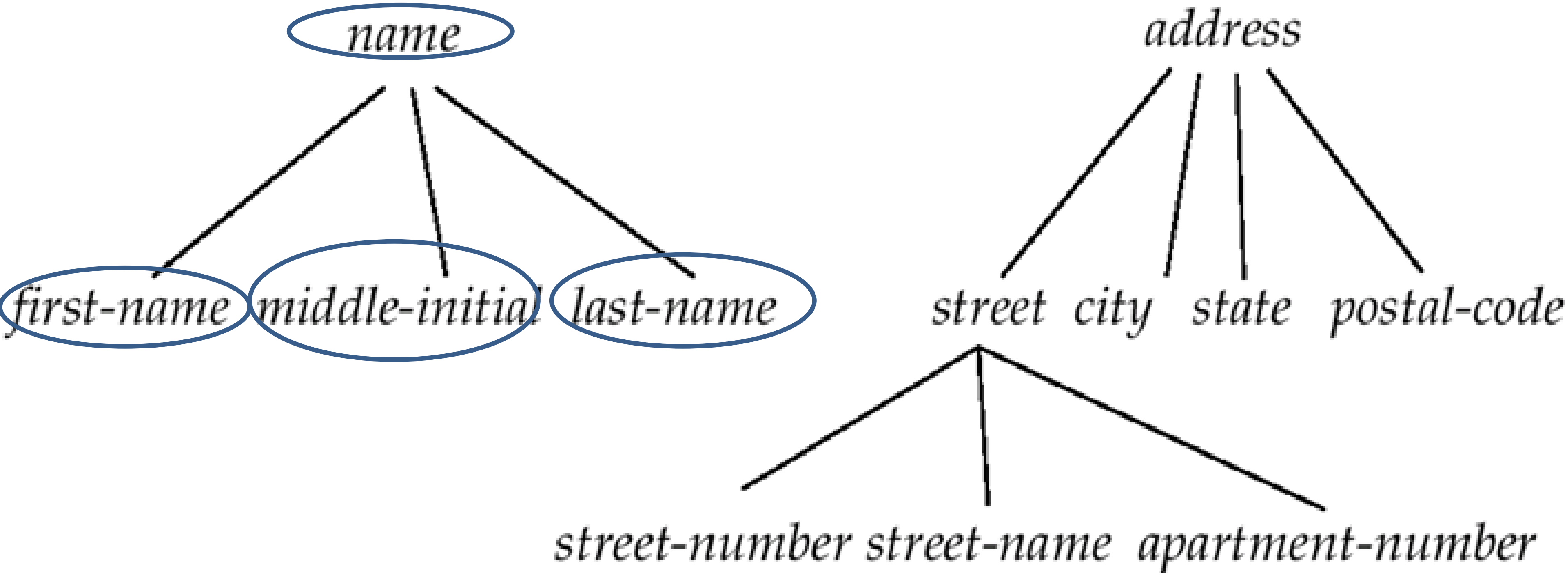
Composite attribute

- Example of attribute



Composite attribute (notation)

- Example of attribute



Composite attribute examples

- Aircraft location
  - Latitude
  - Longitude
  - Altitude

Mapping composite attribute to relation

- Each leaf sub attribute of the entity set is mapped to the column of the corresponding table.
- Egs.

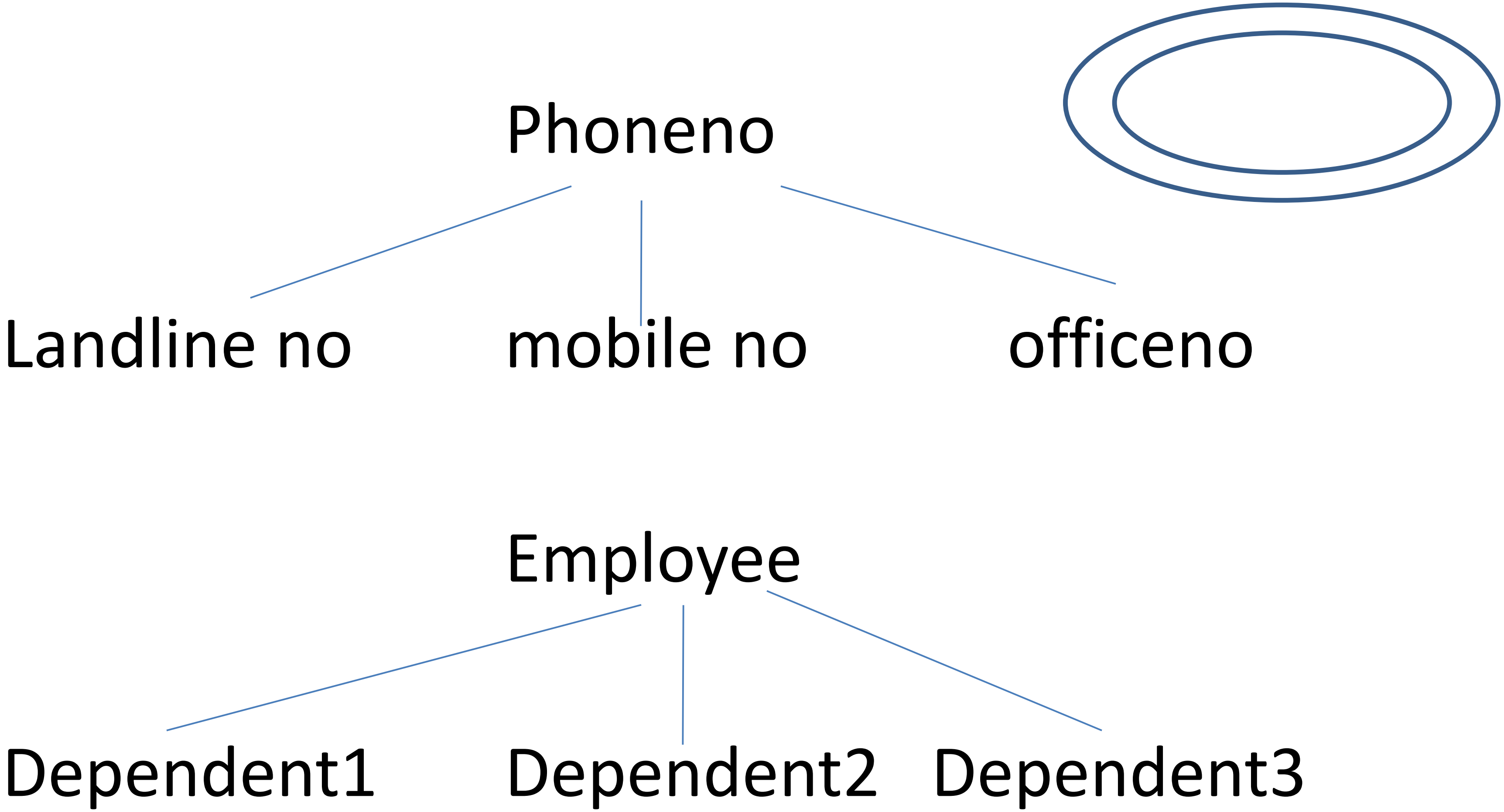
FIRST NAME	MIDDLE NAME	LAST NAME

Observe there is no column for attribute ‘name’

Street num	Street name	Apt num	City	State	Postal-code

Observe there is no column for attributes ‘street’ and ‘address’

Multivalued attributes



Phoneno

Landline no mobile no officeno

Employee

Dependent1 Dependent2 Dependent3

Mapping multivalued attribute to relation

Create a separate table for multivalued attribute with one column as primary key of corresponding entity set and its attributes as remaining columns.

Emp id	Emp name	Email-id	Expertise
E1	AAA	aaa@email.co	C
E2	BBB	bbb@email.co	SQL
E3	CCC	NULL	Java

Empid	Landline no	Mobile No	Office No
E1	111	222	333
E2	NULL	444	555
E3	NULL	NULL	NULL

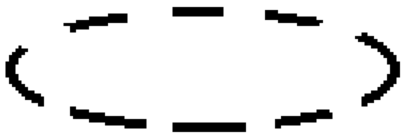
Observe: Though phoneno is shown as multivalued attribute in ER diagram, there is no separate column created for ‘phone no’ in table.

Another method

Empid	PhoneNo	Type
E1	111	Landline No
E1	222	Mobile No
E1	333	Office No
E2	444	Mobile No
E2	555	Office No

Derived attribute

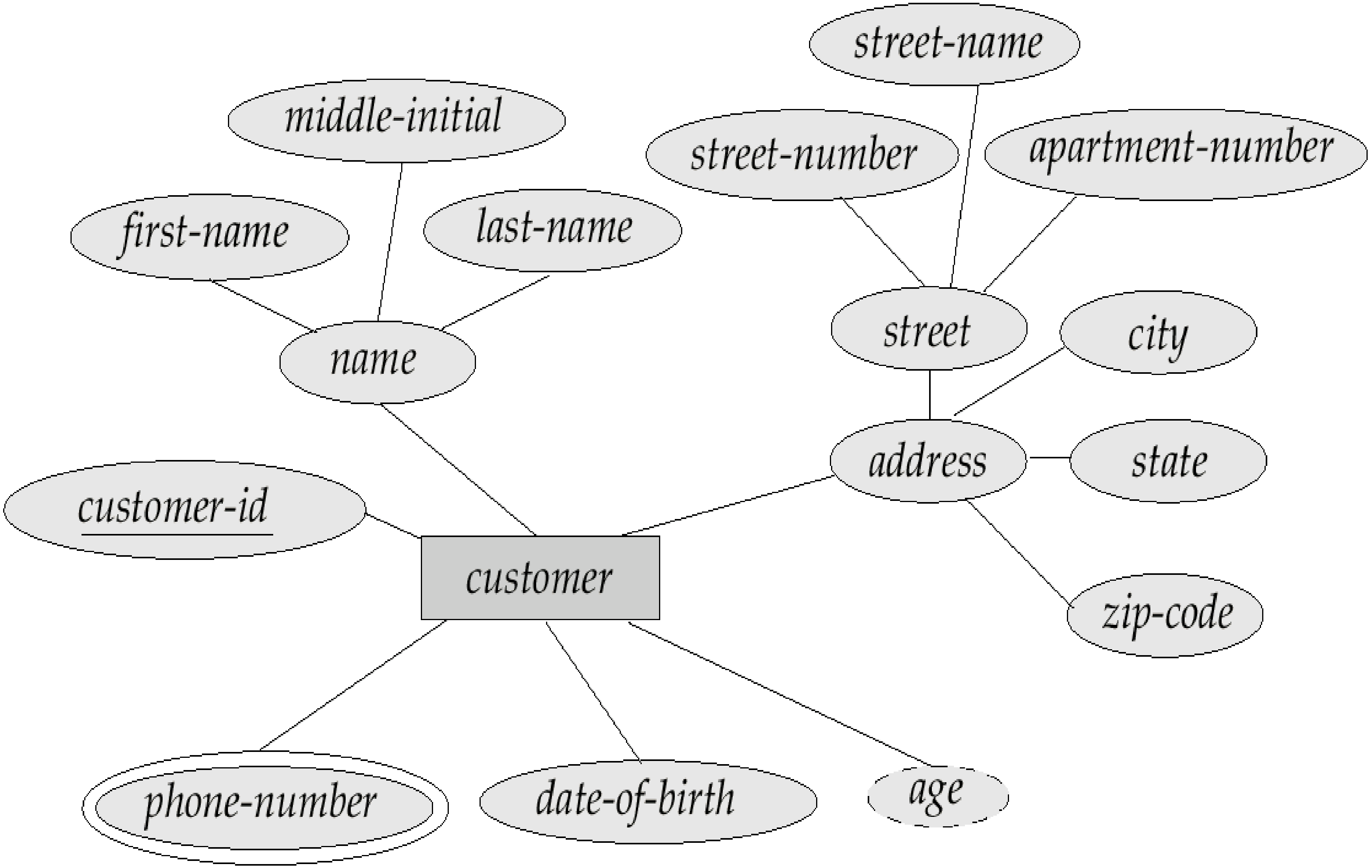
- Age = today's date – date of birth
- Total marks= Midsem_Labmarks + Compre_Labmarks + Midsem_Theory marks+ Compre_Theory marks
- Result = pass / fail depending on total marks
- Sal = no of days * hourly wages
- Annual salary= 12 * monthly salary
- Lib due= no of days exceeded * charge per day
- Notation : dashed ellipse



Types of attributes

- Simple and Composite
- Single valued and Multivalued
- Derived attribute

E-R Diagram with various types of Attributes



Example entity sets

BITS DB

- student
- course
- instructor
- dept

- **student**
  - st_id
  - st_name
  - birth_date
  - gender
  - address

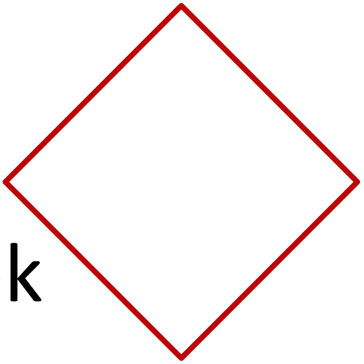
- **course**
  - course_no
  - course_title
  - units
  - l-t-p
  - Sections offered

- **instructor**
  - instructor_id
  - name
  - **dept_code**
  - specialization
  - tel_no

- **dept**
  - **dept_code**
  - dept_name
  - location

RELATIONSHIP AND RELATIONSHIP SET

- Relationship: an association among several entities.
- Relationship set: a set of relationships of same type.



E.g., Abhishek **registers** for DB course

DB course is allocated to Abhishek

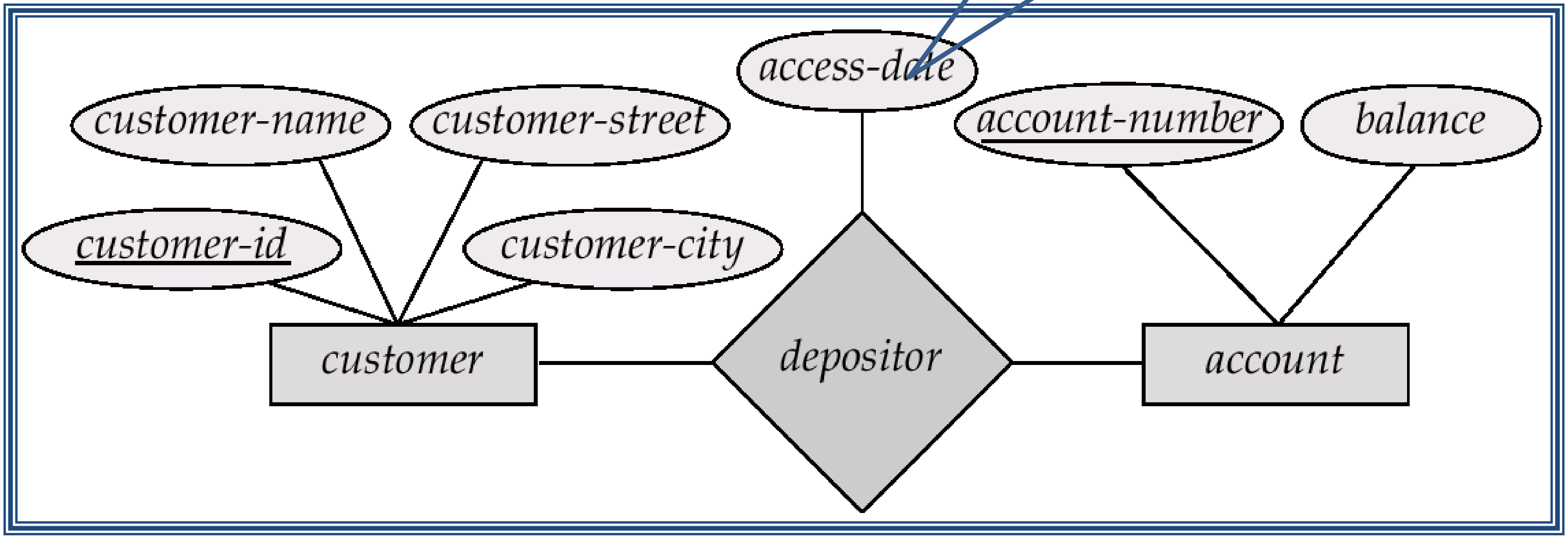
E.g., Customer C1 **holds** an Account A1

Account A1 belongs to Customer C1

E.g., Customer C2 **borrows** a Loan L1

Loan L1 is taken by Customer C2

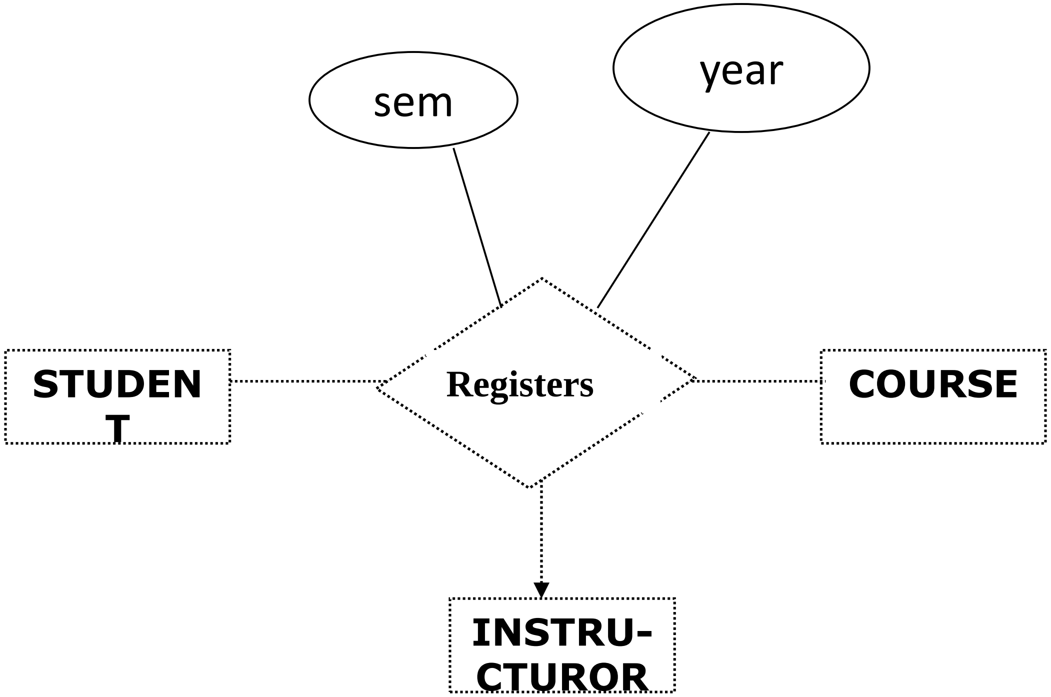
Example



Descriptive attribute

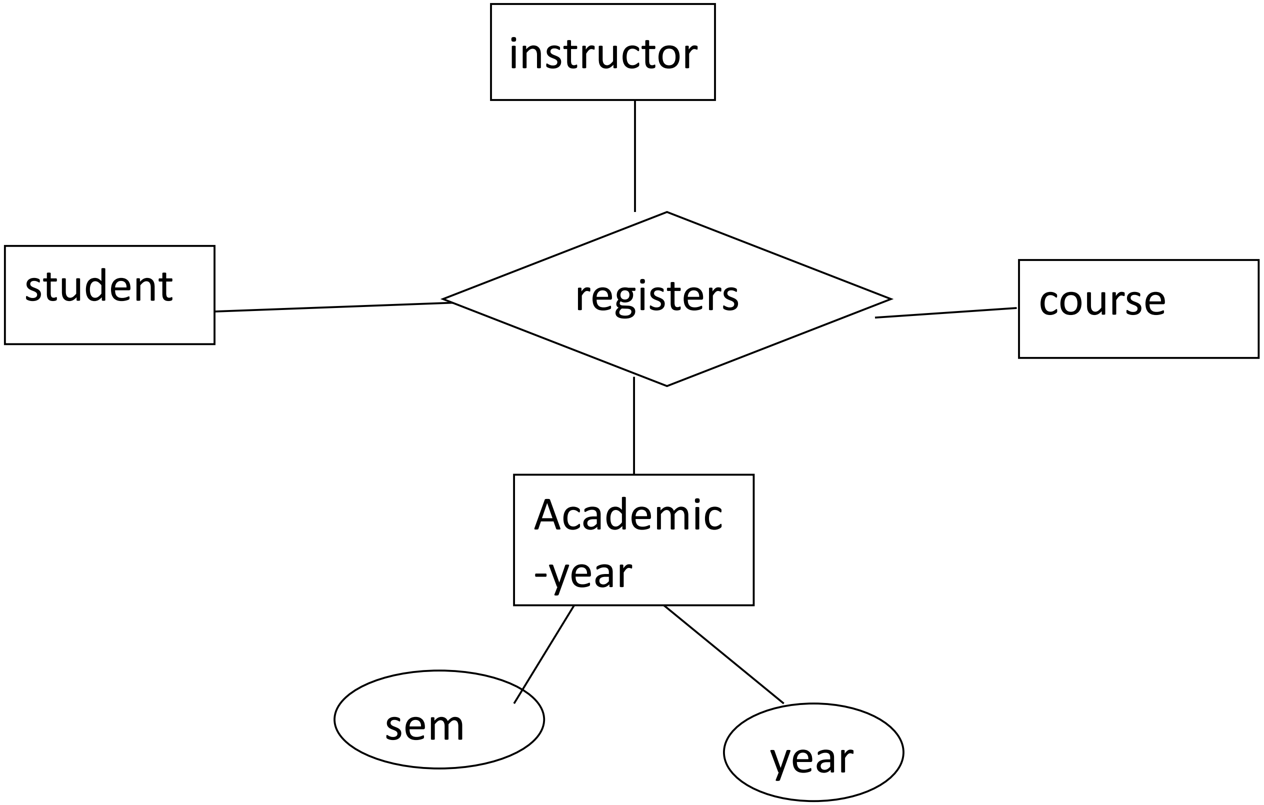
Attributes on Relationships

- Sometimes it is useful to attach an attribute to a relationship.
- Think of this attribute as a property of tuples in the relationship set.



Equivalent Diagrams Without Attributes on Relationships

- Create an entity set representing values of the attribute.
- Make that entity set participate in the relationship.



Degree of Relationships

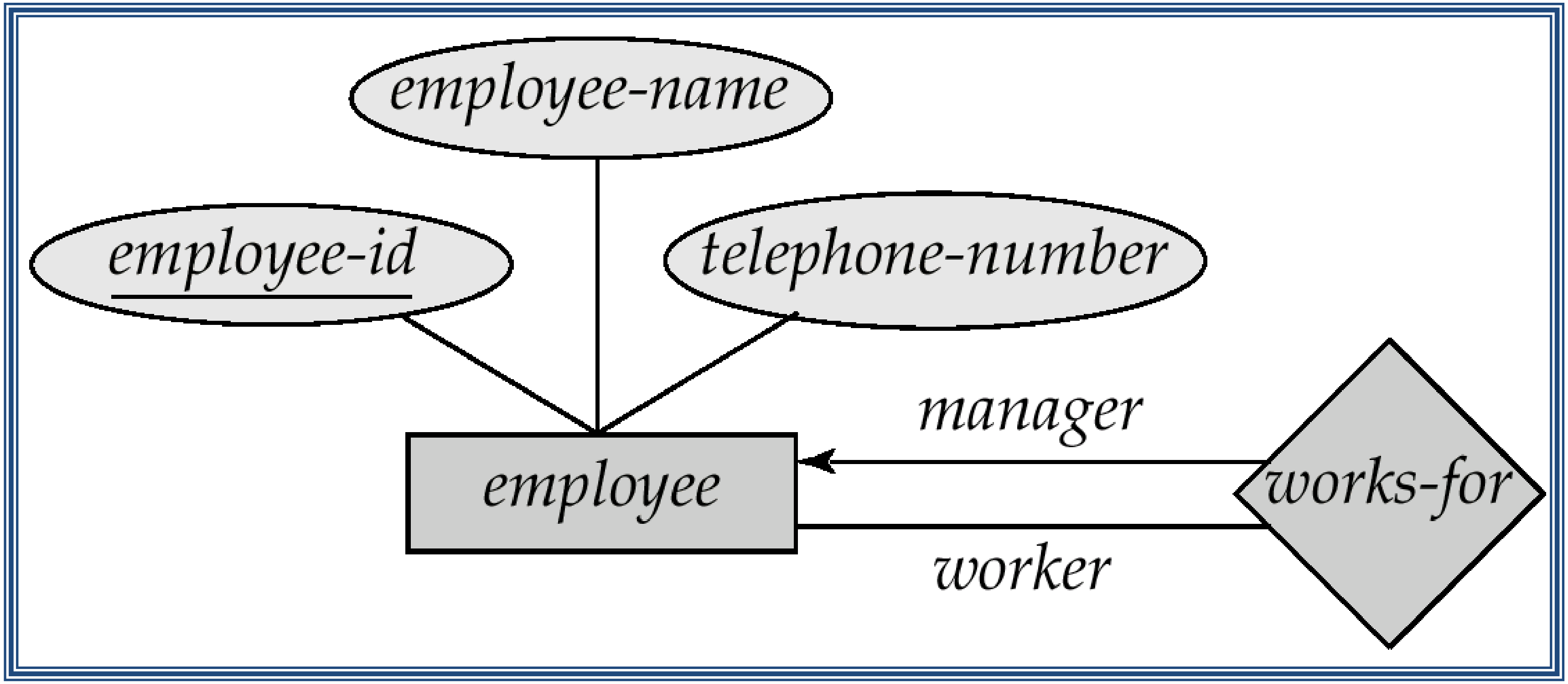
Unary, Binary, Ternary

- **Unary**: A relationship between the instances of a single entity set.

- **Binary**: A relationship between the instances of two entity sets.

- **Ternary**: A simultaneous relationship between the instances of three entity sets.

Example of Unary Relationship



Example of Unary Relationship

- 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

- Also called as **recursive relationship**

Mapping Unary relationship into table

Empid	Empname	Telephone number	Works for(foreign key refers to Empid from same table
E1	AAA	111	NULL
E2	BBB	222	E1
E3	CCC	333	E1
E4	DDD	NULL	E2
E5	EEE	44	E5 (acc to foreign key this is a valid entry though logically this do not make sense)
E6	FFF	55	E7 X

SQL query

```
create table Employee ( empid varchar(3),
empname varchar(10) NOT NULL,    worksfor
varchar(3),    primary key(empid),     foreign key
(worksfor) references Employee(empid));
```

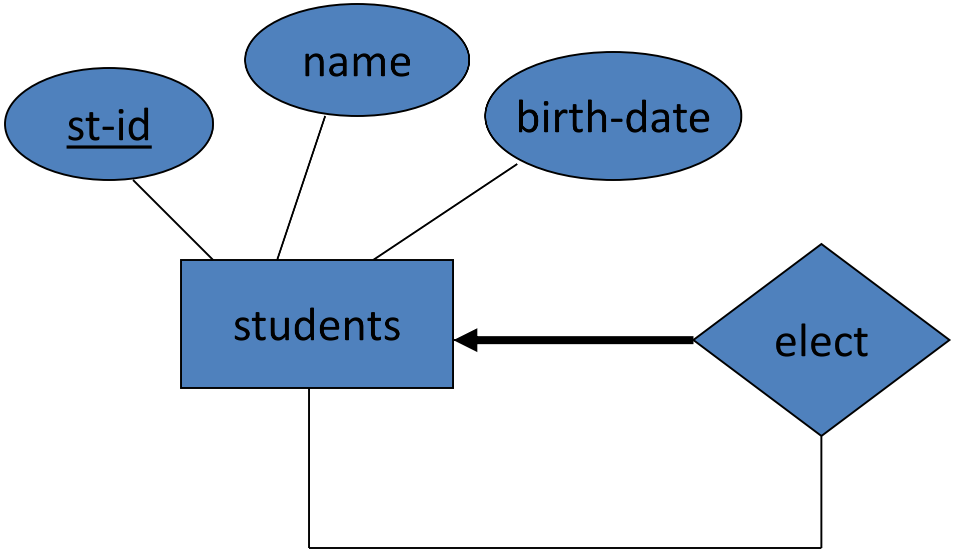
Cardinality constraint

Having works for column in same table and Empid as primary key, has automatically taken care of 1:M cardinality constraint

For 1:1 , same as above, ie works for column in same table , with unique constraint on Works for column

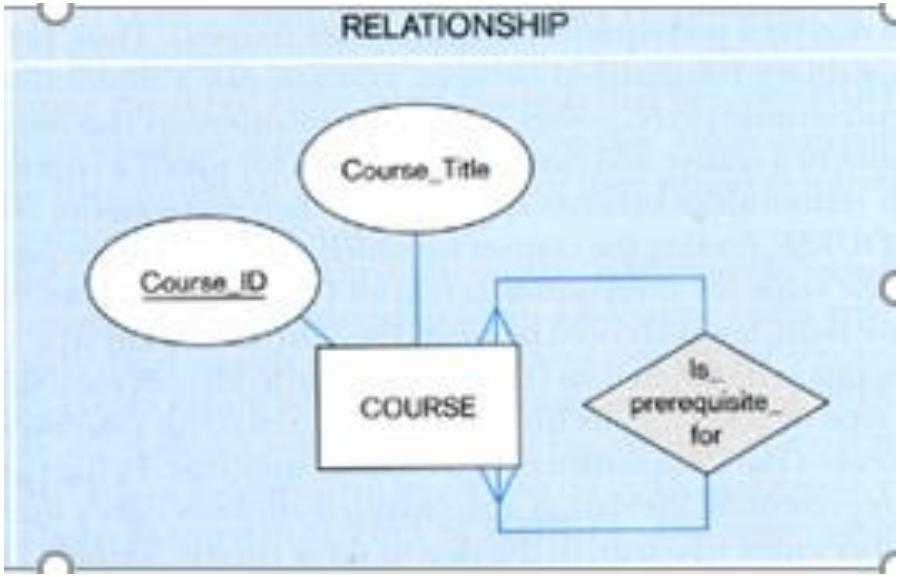
For M:1, i.e 1 employee can report to many managers, make a new table.
create table reports(empid varchar(3),managerid varchar(3) references Employee(empid));
This takes care of 1:M i.e 1 manager can manage many employees
This is same for M:M

Example of Unary Relationship and 1:M cardinality constraint

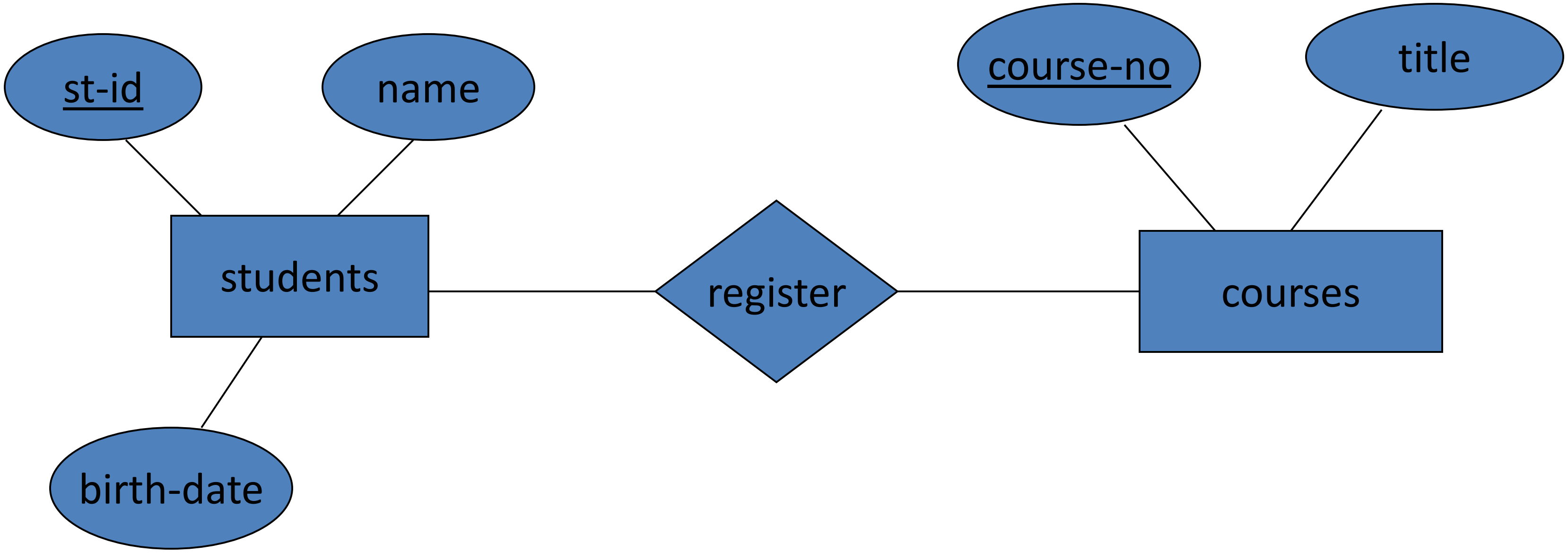


Unary relationship example

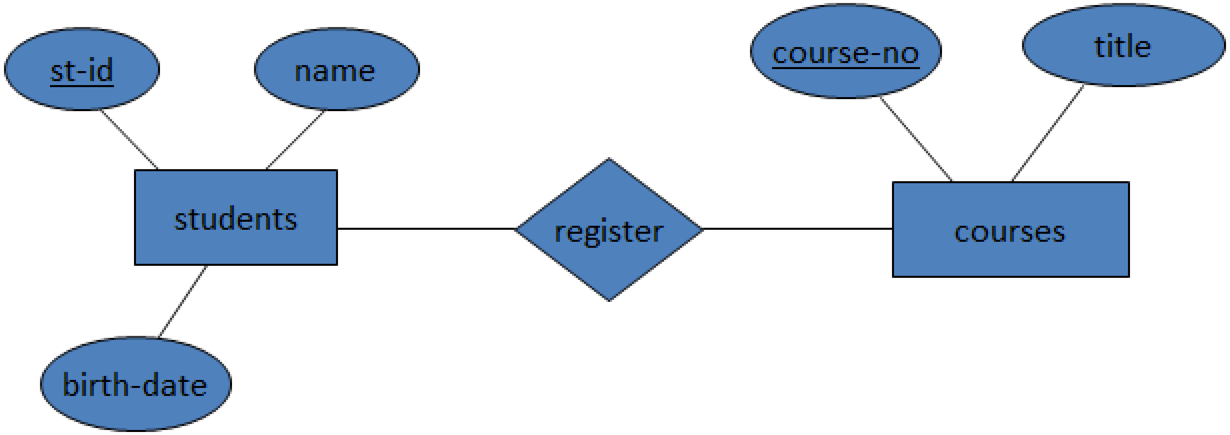
Course	Description	Prerequisite
CIS81	Programming	---
CIS209	VB	CIS81
CIS281	Systems Analysis	CIS 209, CIS330
CIS330	Networks	CIS209, CIS330



Example of Binary Relationship



Mapping binary relation into table



students

St-id	Name	Birth-date

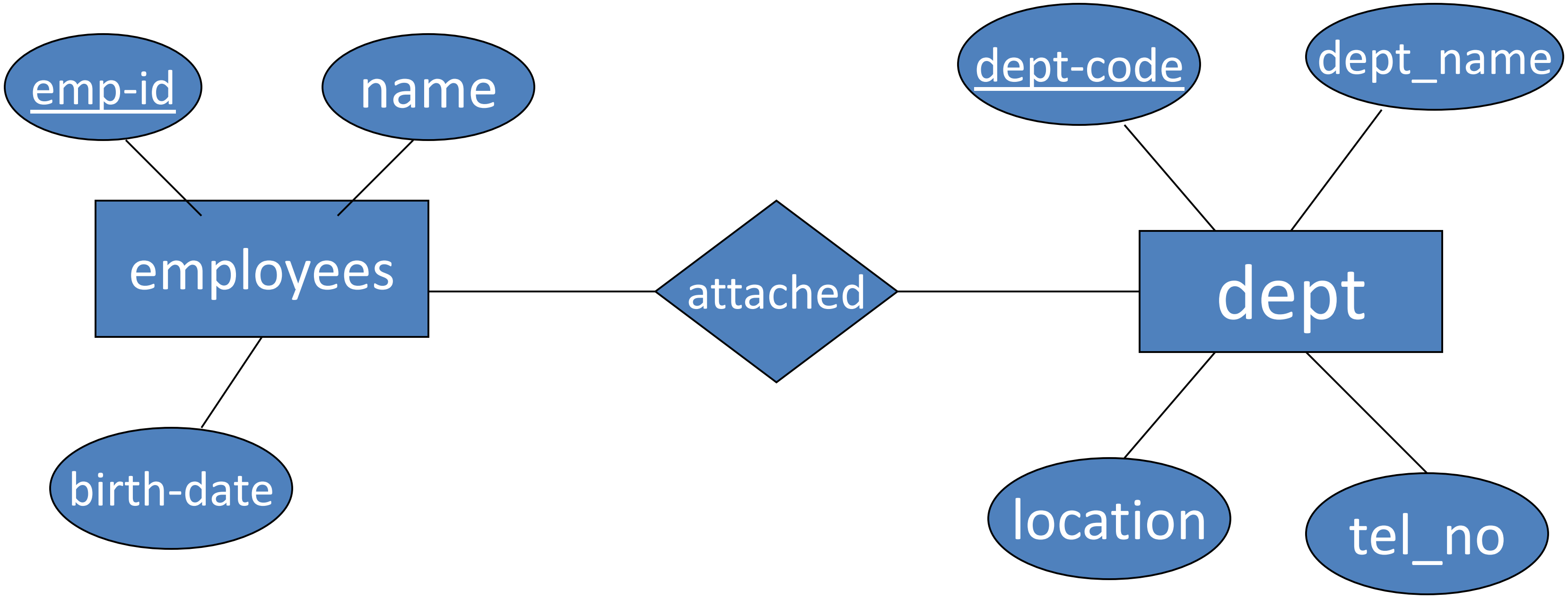
courses

Course-no	title

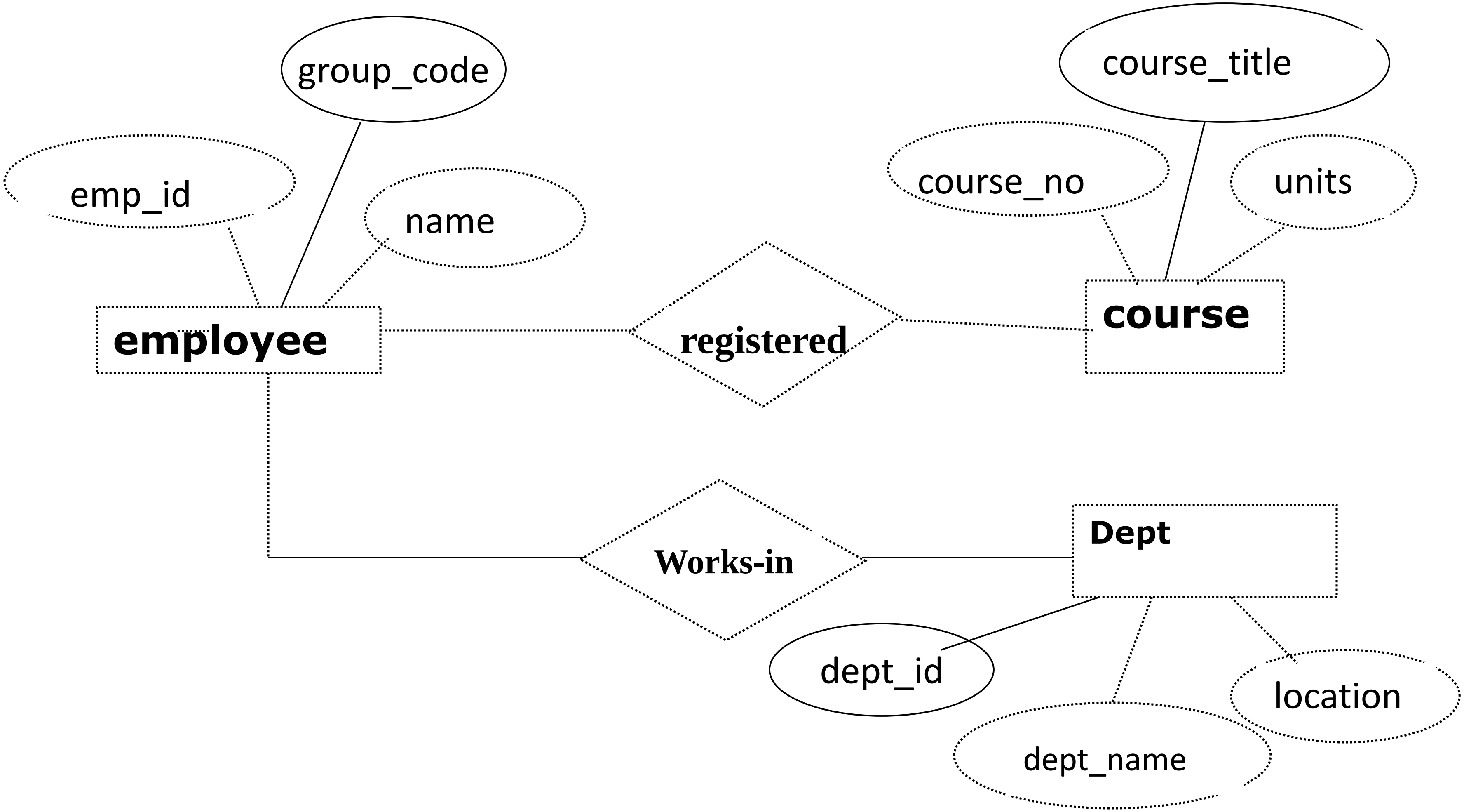
register

St-id	Course-no

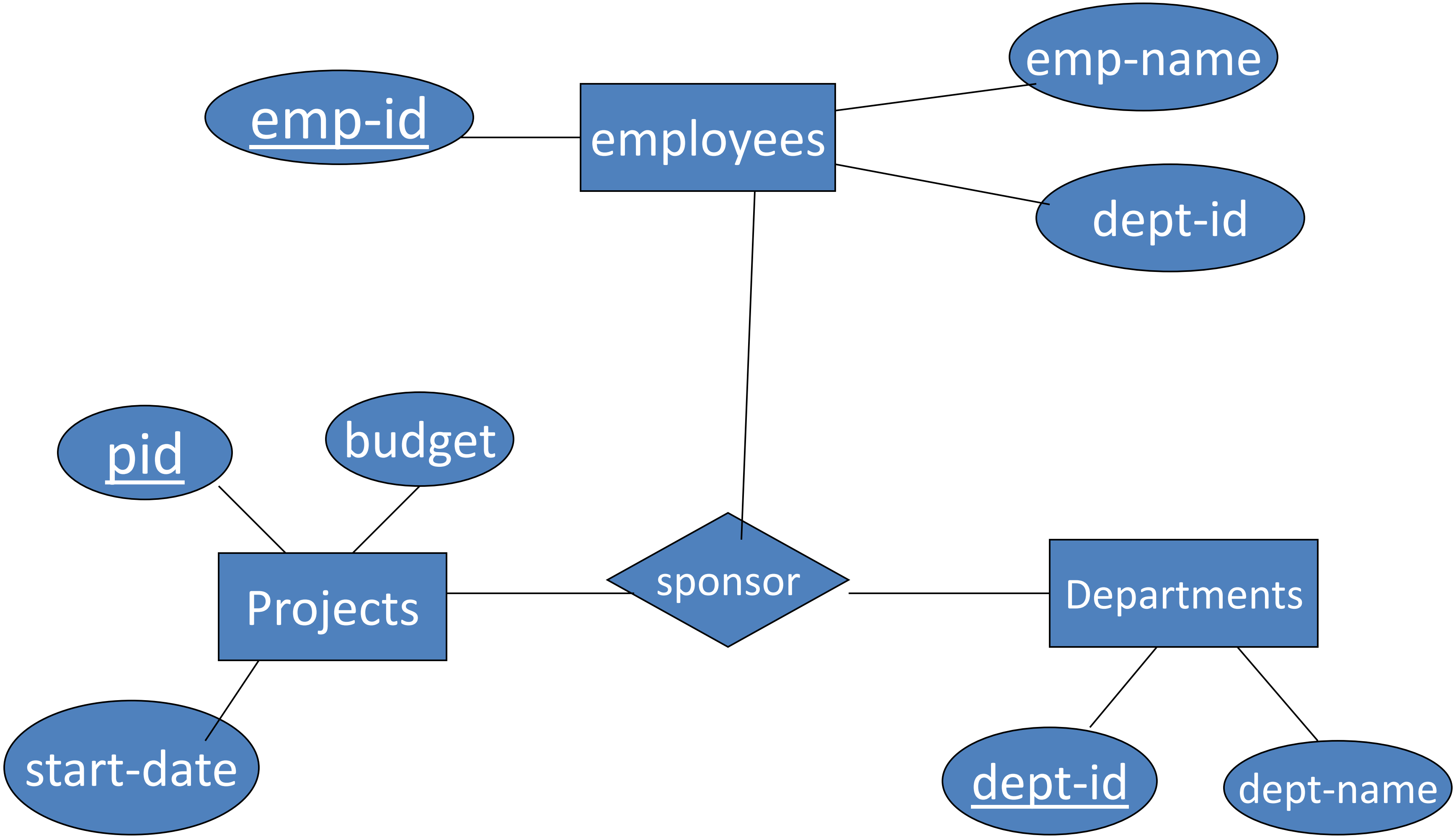
Example of Binary Relationship



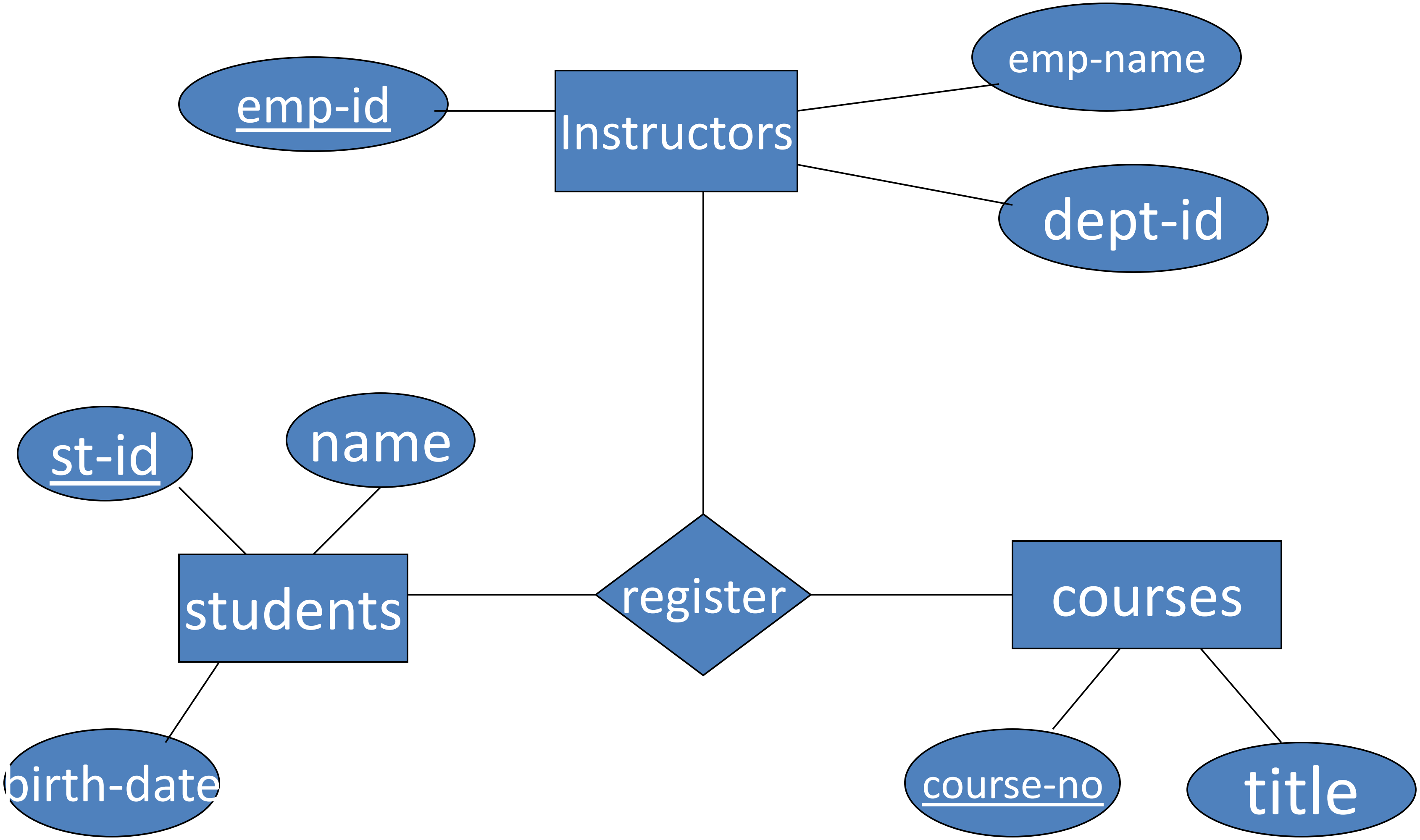
Example of Binary Relationship



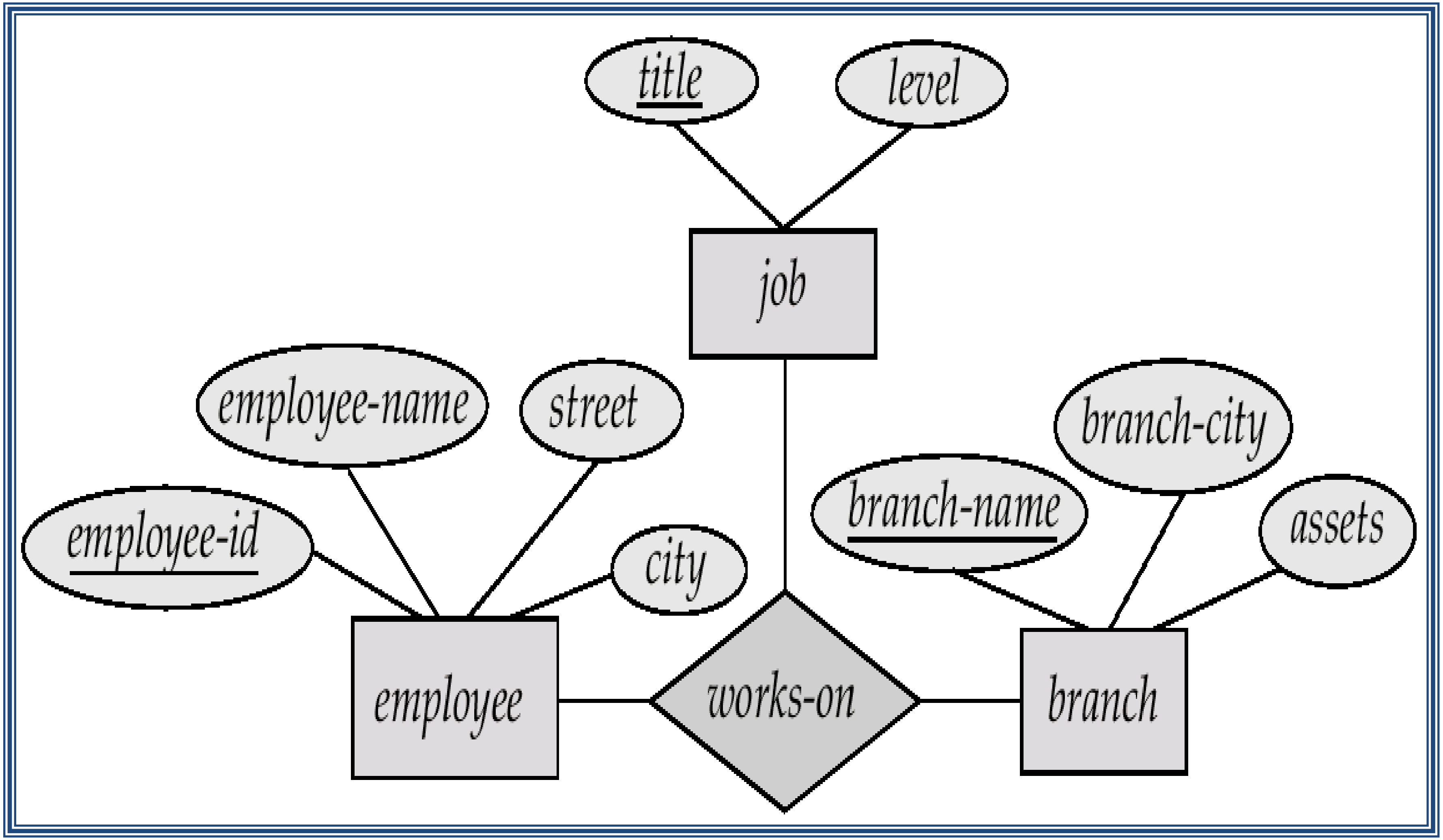
Example of Ternary Relationship



Example of Ternary Relationship



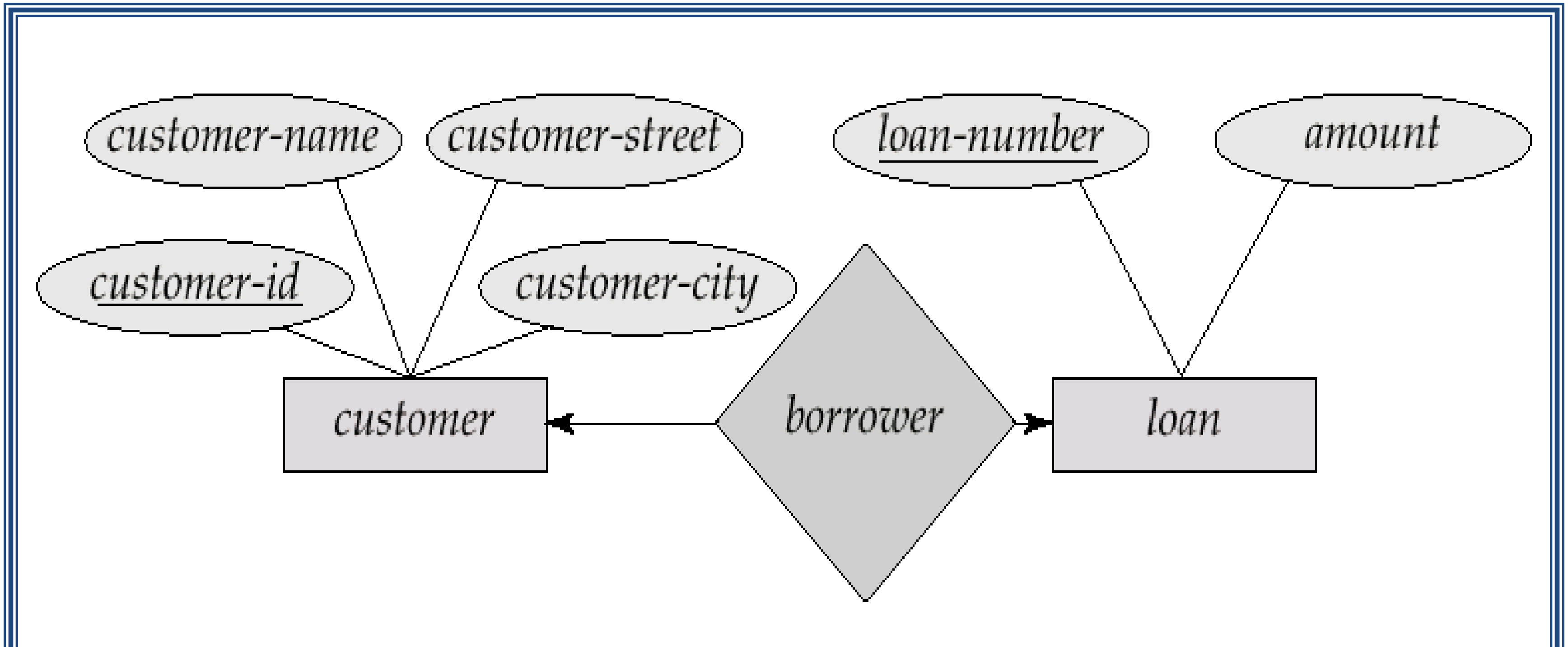
Example of Ternary Relationship



Example of Binary Relationship

One to one relationship

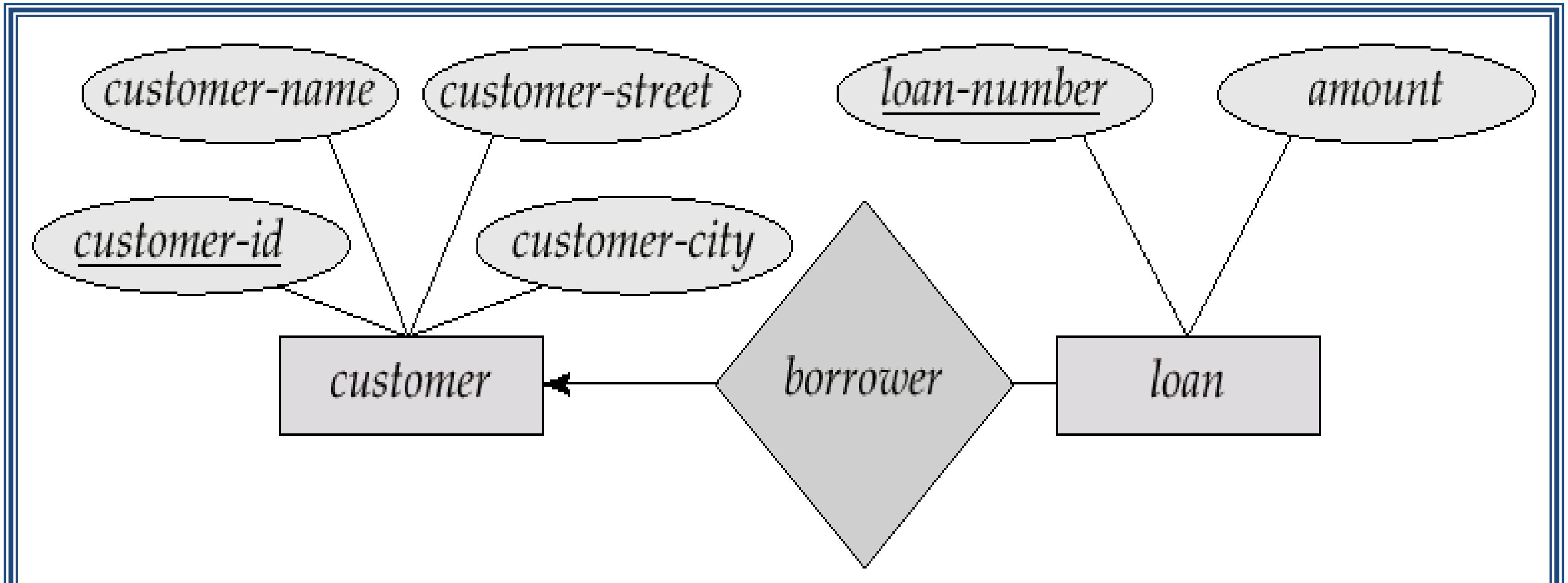
- A customer is associated with at most one loan via the relationship *borrower*
- A loan is associated with at most one customer via *borrower*



Example of Binary Relationship

One to many relationship

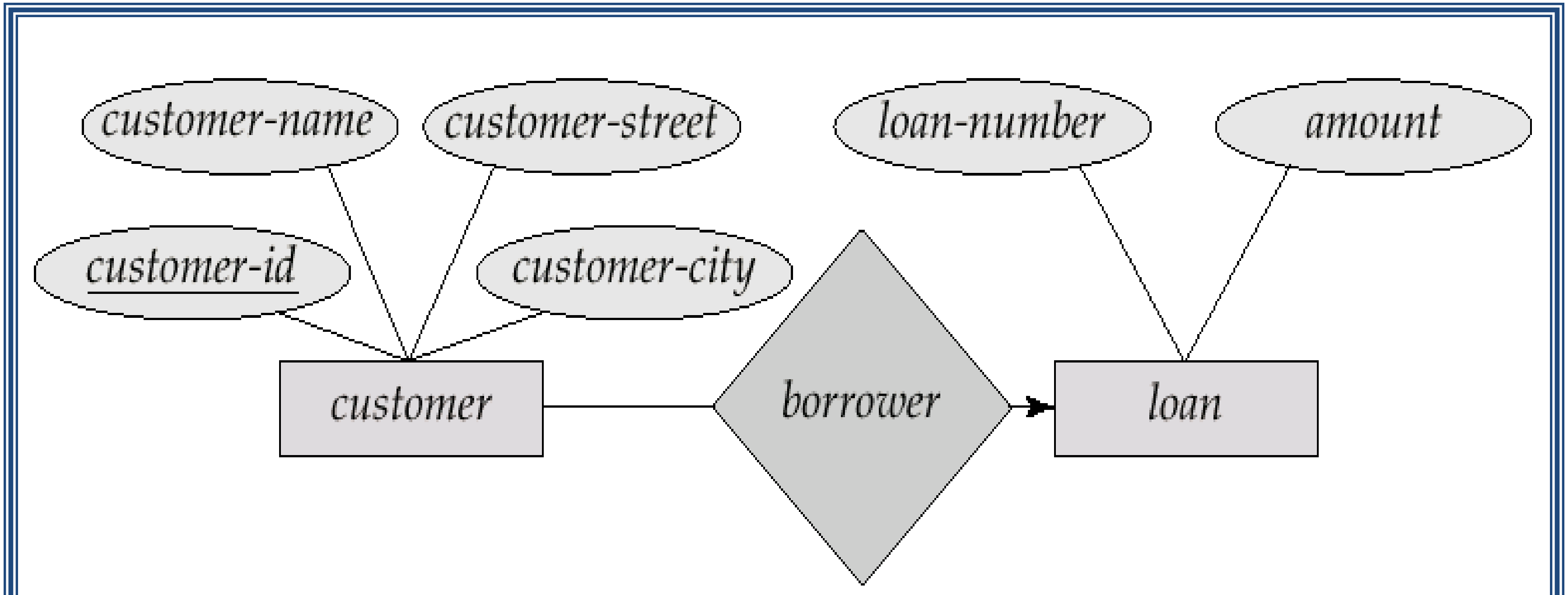
- A loan is associated with at most one customer via borrower,
- A customer is associated with several (including 0) loans via *borrower*



Example of Binary Relationship

Many to one relationship

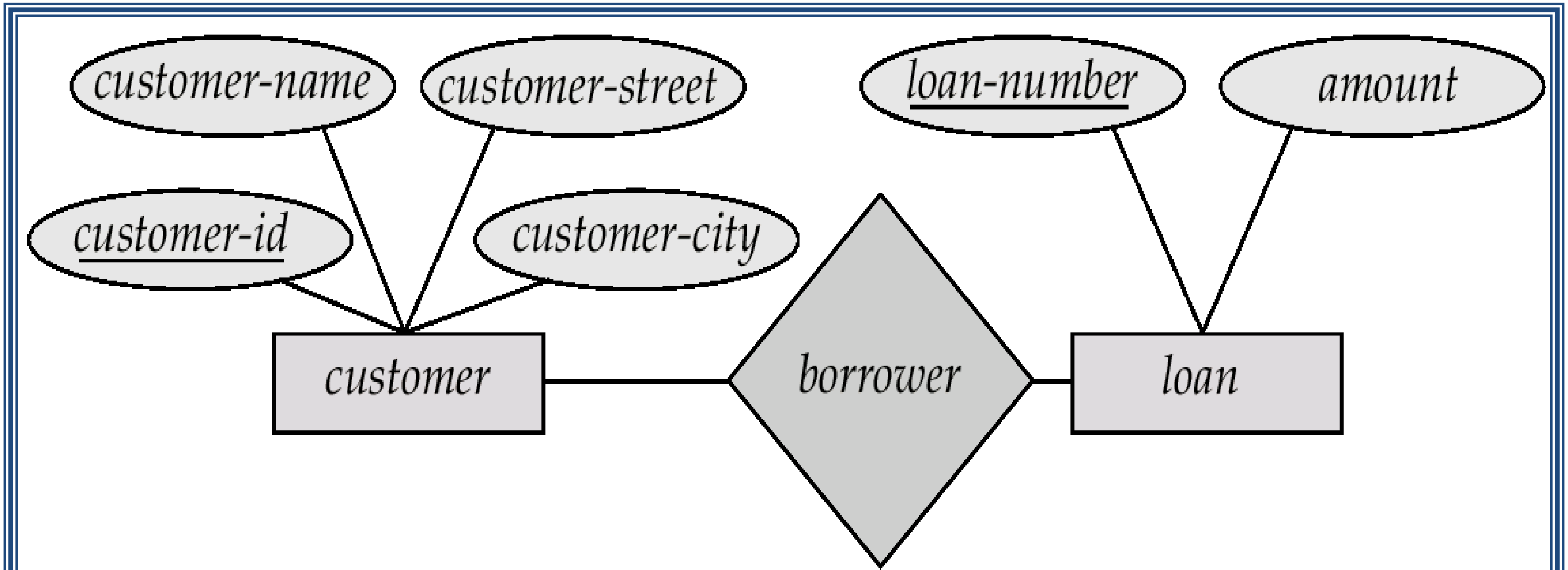
- A loan is associated with several (including 0) customers via *borrower*,
- A customer is associated with at most one loan via *borrower*



Example of Binary 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



Types of Relationships/ Mapping Cardinality Constraints

- Types of binary relationship from entity set ‘A’ to entity set ‘B’

  one – one: an entity in A is related to

  at most one entity in B and vice versa

  many – one: an entity in A is related to

  at most one entity in B

  many – many: an entity in A is related to

  0 or more entities in B and vice versa

Cardinality constraint on a relationship

A cardinality constraint between two entities A and B, specifies the number of instances of entity B that can (or must) be associated with each instance of entity A.

- **Minimum cardinality**: the minimum number of instances of one entity that may be associated with each instance of another entity.

- **Maximum cardinality**: the maximum number of instances of one entity that may be associated with each instance of another entity.

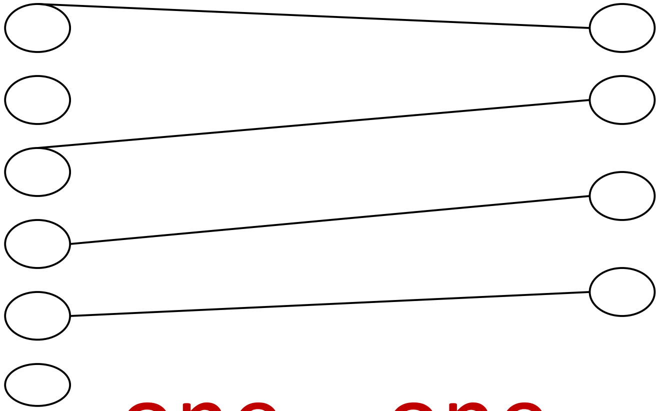
min-max constraints

- Suppose that each customer must have at least one account, but is restricted to at most two loans at a time, and that a bank branch cannot have more than 1000 loans. How does this show up as min-max constraints.
- Customer-Loan (0, 2)
- Customer-Account (1, N)
- Bank-Branch – Loan (0, 1000)

Example : One-One Relationships

Relationship Allocated between entity sets hostel-rooms and Students.

A room cannot be allocated to more than one student, and no student can have more than one room.

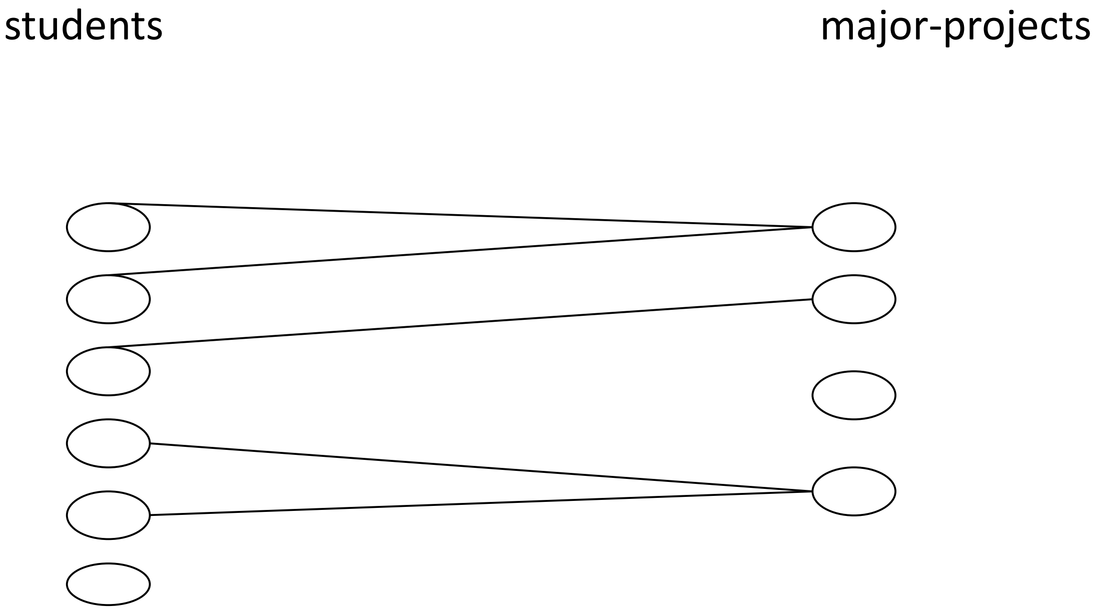


hostel-rooms students

one – one

Example : Many-One Relationships

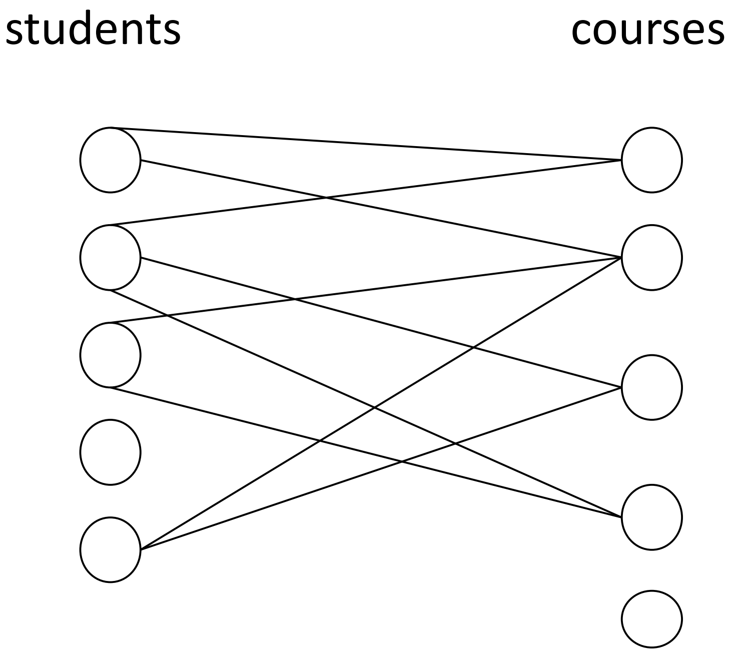
- Each entity of the first set is connected to at most one entity of the second set.
- But an entity of the second set can be connected to zero, one, or many entities of the first set.
- E.g. A student can do at most one major project and a major project can be done by more than one student



many – one

Example : Many-Many Relationships

- Focus: binary relationships, such as **Registers** between **Students** and **Courses**.
- In a *many-many* relationship, an entity of either set can be connected to many entities of the other set.
- E.g., a student registers many courses; an instructor is allotted many courses.



many - many

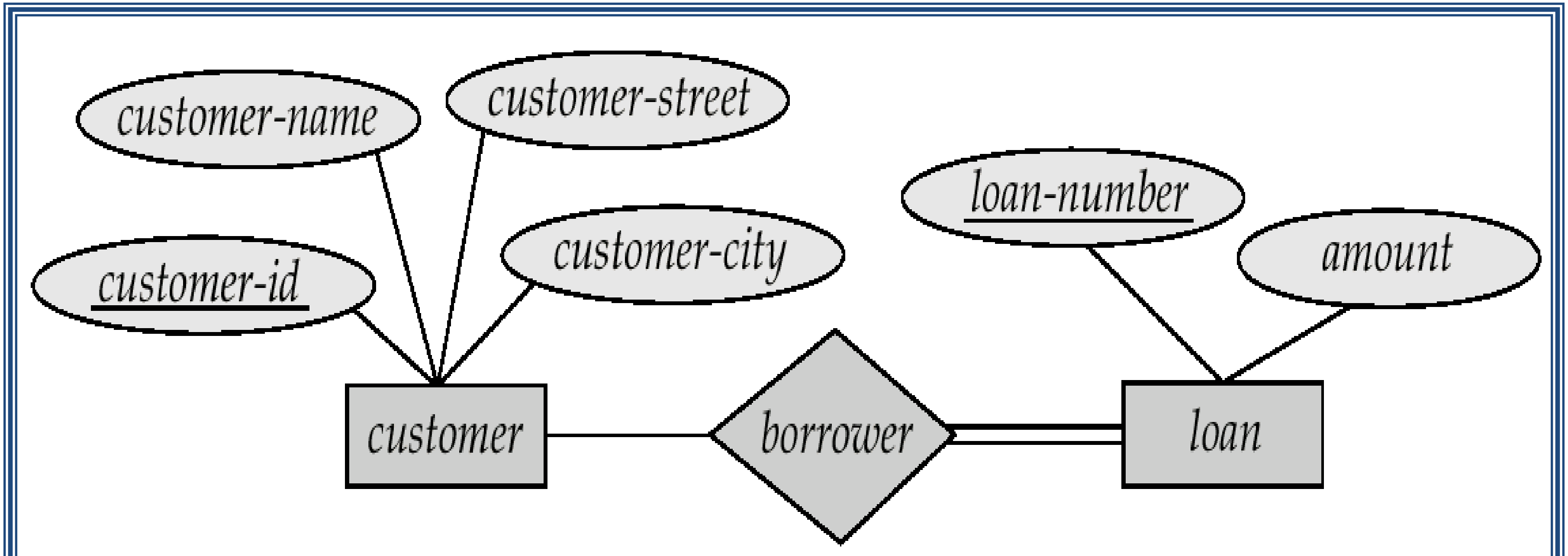
Examples

- Phone number (landline no, mobile no, office no)
- Hobbies of an employee (1 employee can have many hobbies)
- Are these multivalued attributes or 1:m mapping cardinality?
- Addresses (shipping address, billing address)
- Email_ids of an employee (personal email id, official email id)
- Loan taken by customer
- Dependents of an employee

Participation of an Entity Set in a Relationship Set

Total participation

- every loan must have a customer associated to it via borrower
- participation of loan in borrower is **total**
- **Indicated by double line**



Participation constraint on a Relationship Set

Total participation : every entity in the entity set participates in at least one relationship in the relationship set

Partial participation: some entities may not participate in any relationship in the relationship set

E.g. participation of customer in borrower is partial

Indicated by single line

Keys

- Student (id, name, dob, city, street, state, hostel_no, mobile_no, email_id)
- **Superkey:** {id, name, dob, city, street, state, hostel_no, mobile_no, email_id}
- **Candidate keys:**
  - {id}
  - {id,name}
  - {name,mobile_no}
  - {id, email_id}
  - {email_id}
- **Primary key** {id}

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.
- Although several candidate keys may exist, one of the candidate keys is selected to be the **primary key**.

Primary Key

- Primary key is an unique identifier and cannot have NULL value.
- It is always good to define a primary key to a table.
- A table(relation) may or may not have a primary key.
- Primary key can be a combination of more than one attribute.
- A table has only one primary key.

Keys

- **Primary key:**

____

An attribute (or combination of attributes) that uniquely identifies each row in relation.

- **Composite key:**

A primary key that consists of more than one attribute.

- **Foreign key:**

An attribute in a relation that serves as the primary key of another relation in the same database.

Q.

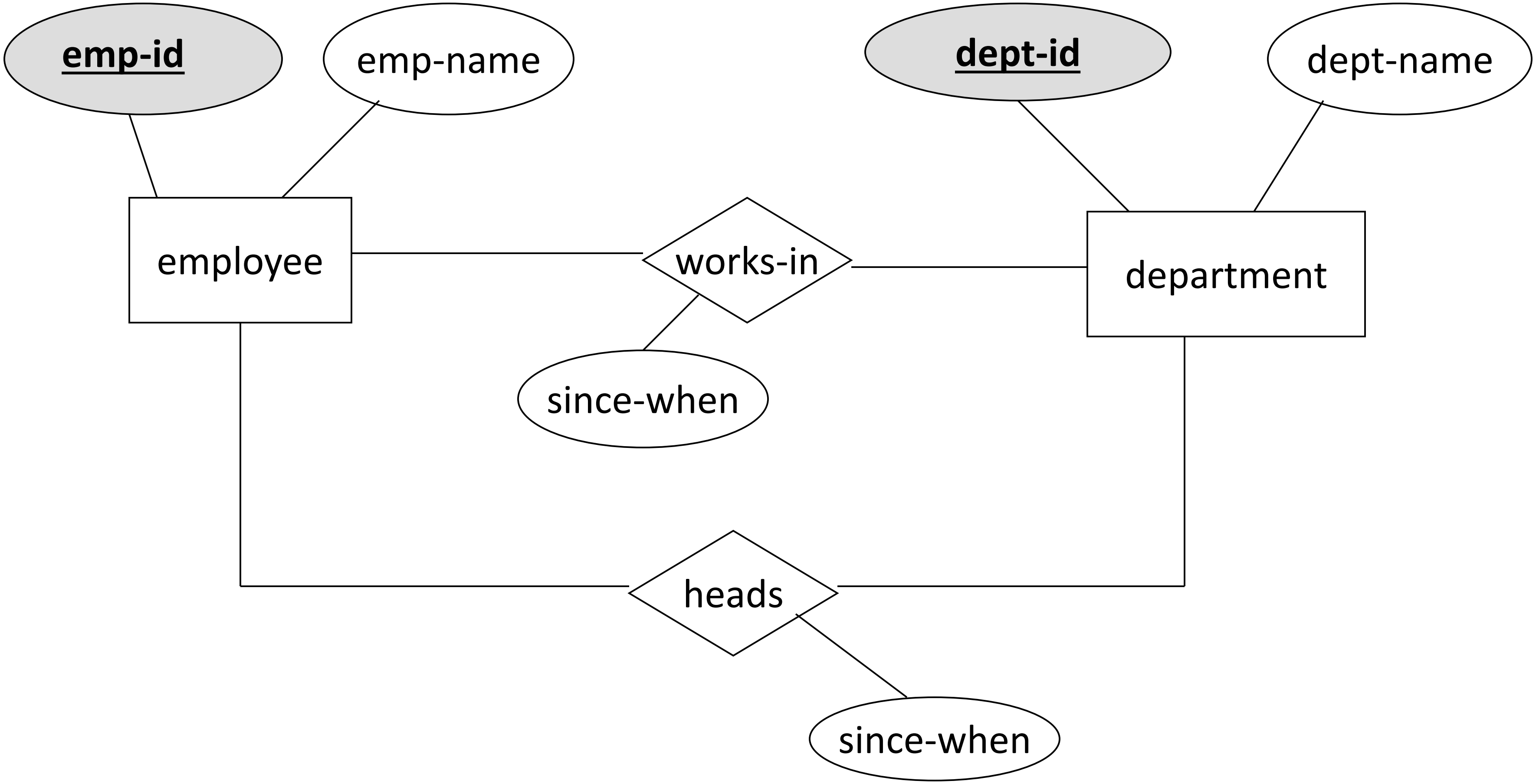
Difference between candidate key and composite key?

Keys for Relationship Sets

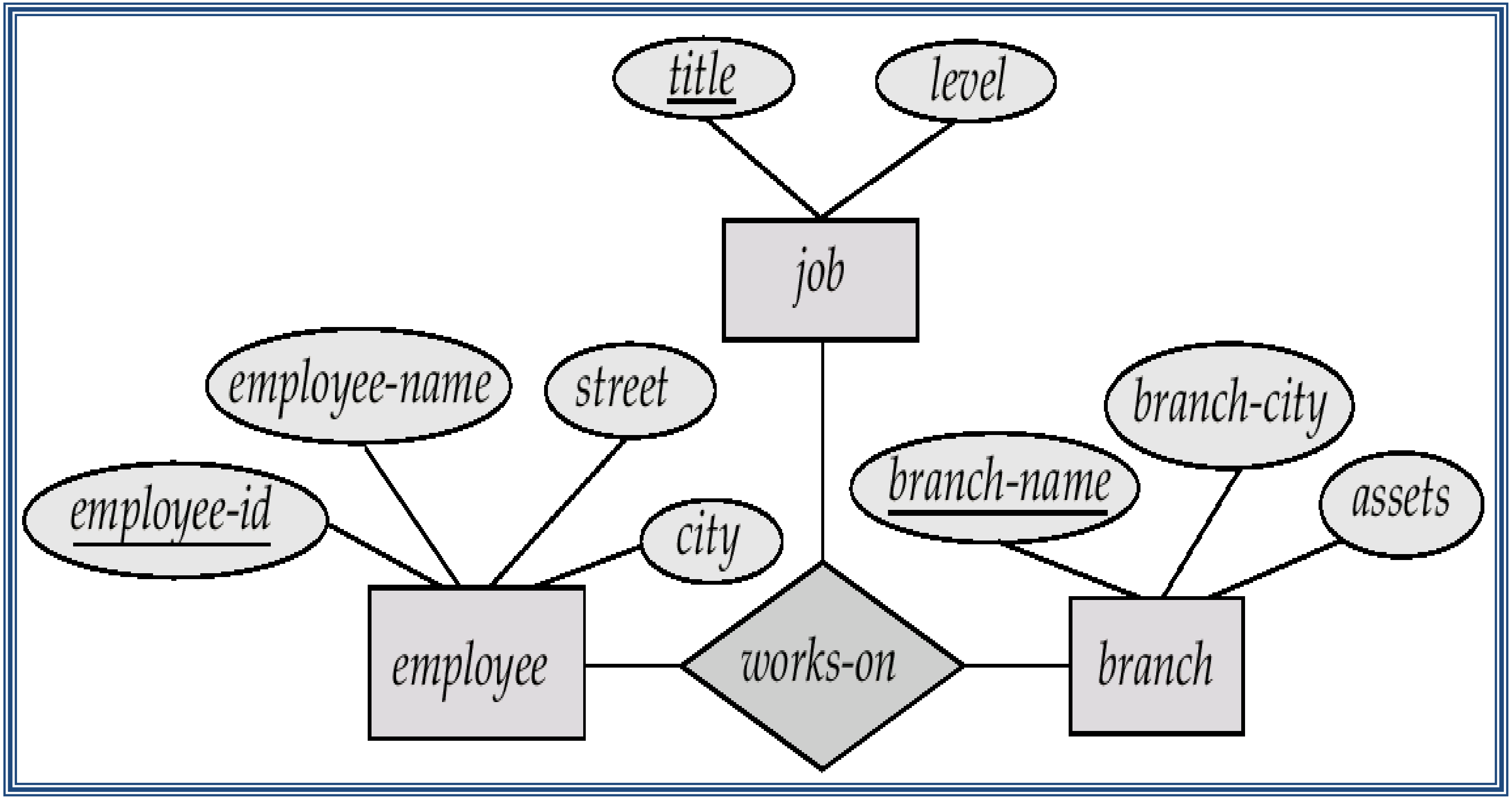
- The combination of primary keys of the participating entity sets forms a super key of a relationship set.
  - (*customer-id, account-number*) is the super key of *depositor*
  - *NOTE: this means a pair of entity sets can have at most one relationship in a particular relationship set.*

- Must consider the mapping cardinality of the relationship set when deciding the what are the candidate keys

Key Constraints



Key constraint in ternary relationship

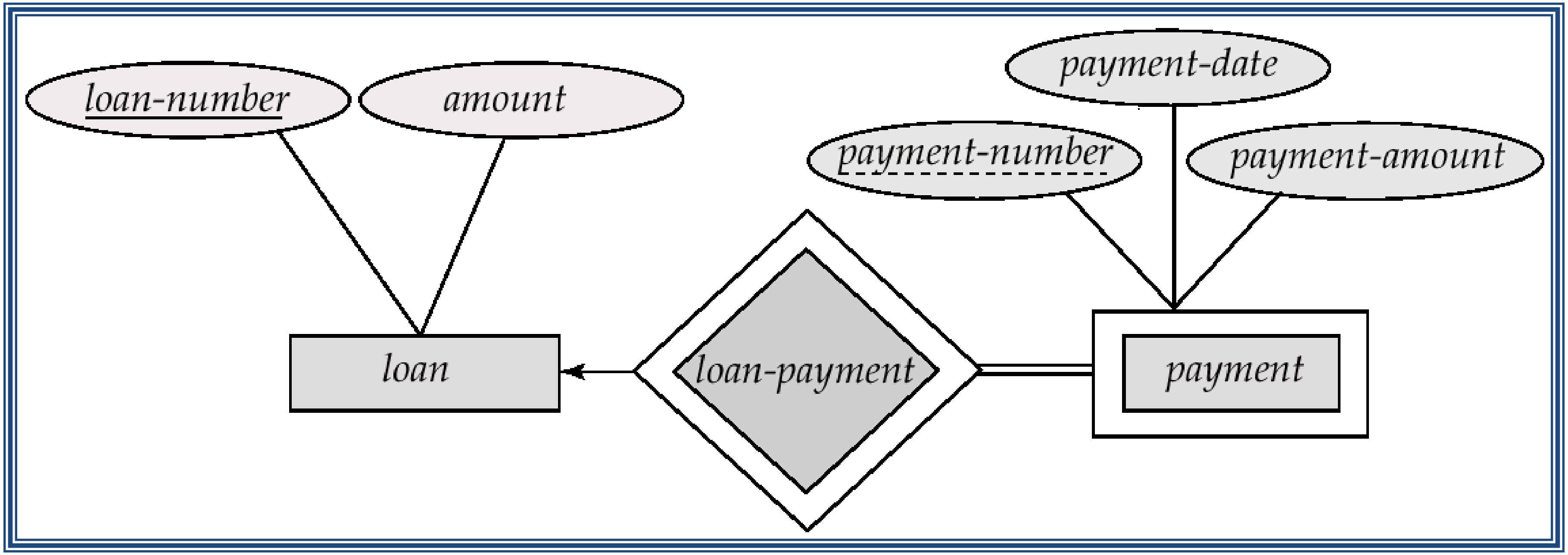


Consider Relation having loan details

Loan no	**Payment no / EMI**	**Payment-Amount**	**Payment-Date**
L1	P1	1 lakh	1st jan
L1	P2	1 lakh	1st feb
L2	P1	2 lakh	1st jan
L2	P2	5000	15th feb

Weak Entity Set

- Loan is a strong entity set and payment is a weak entity set
- Weak entity set is denoted by double rectangles.
- *payment-number* – discriminator of the *payment* entity set
- underline the discriminator of a weak entity set with a dashed line.
- Primary key for *payment* – (*loan-number, payment-number*)



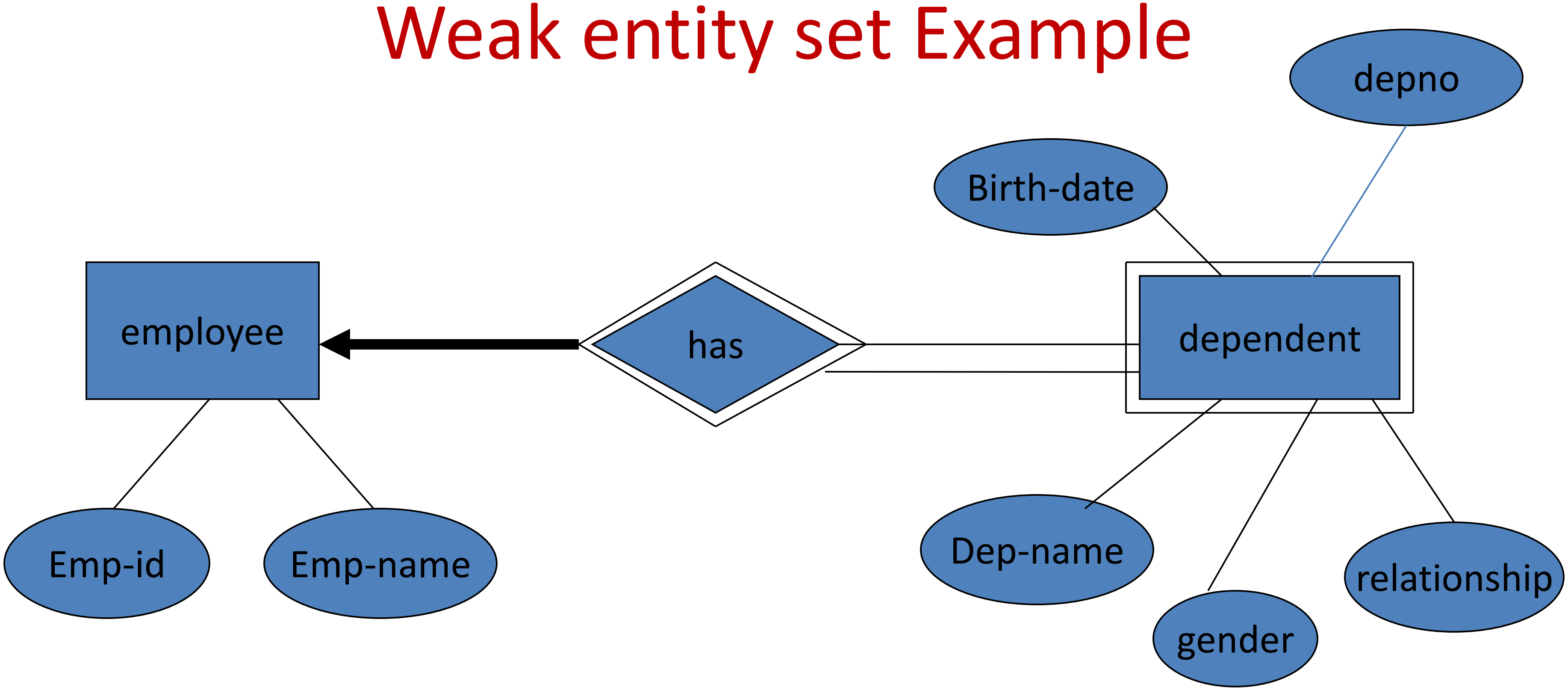
Weak Entity Sets (Cont.)

- An entity set that does not have a primary key is referred to as a weak entity set.
- The existence of a weak entity set depends on the existence of a identifying entity set (loan in above eg)
  - it must relate to the identifying entity set via a **total, one-to-many relationship set** from the identifying to the weak entity set
  - Identifying relationship depicted using a double diamond

Weak Entity Sets (Cont.)

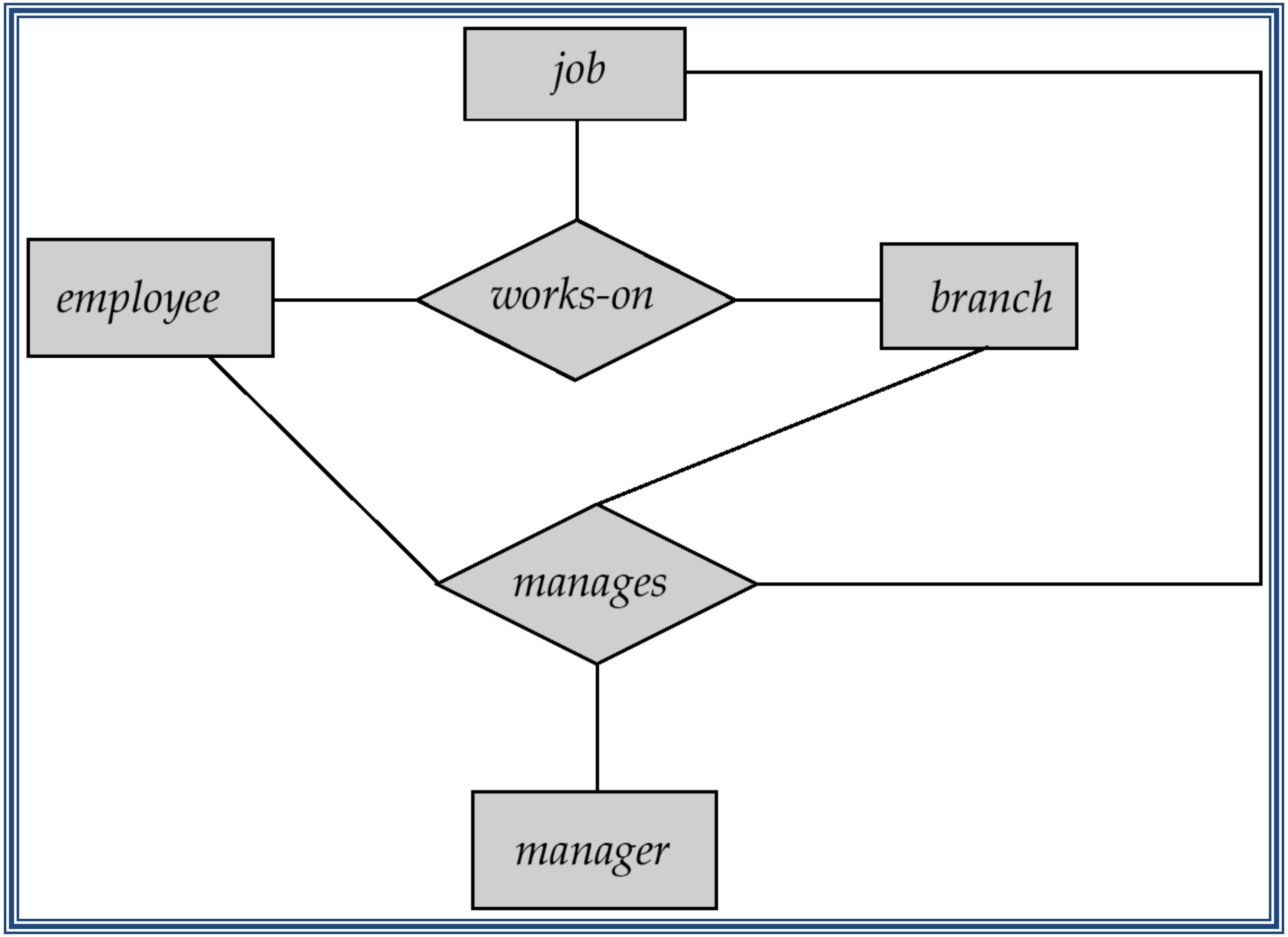
- The discriminator (or partial key) of a weak entity set is the set of attributes that distinguishes among all the entities of a weak entity set.
- The primary key of a weak entity set is formed by the primary key of the strong entity set on which the weak entity set is existence dependent, plus the weak entity set’s discriminator.

Weak entity set Example

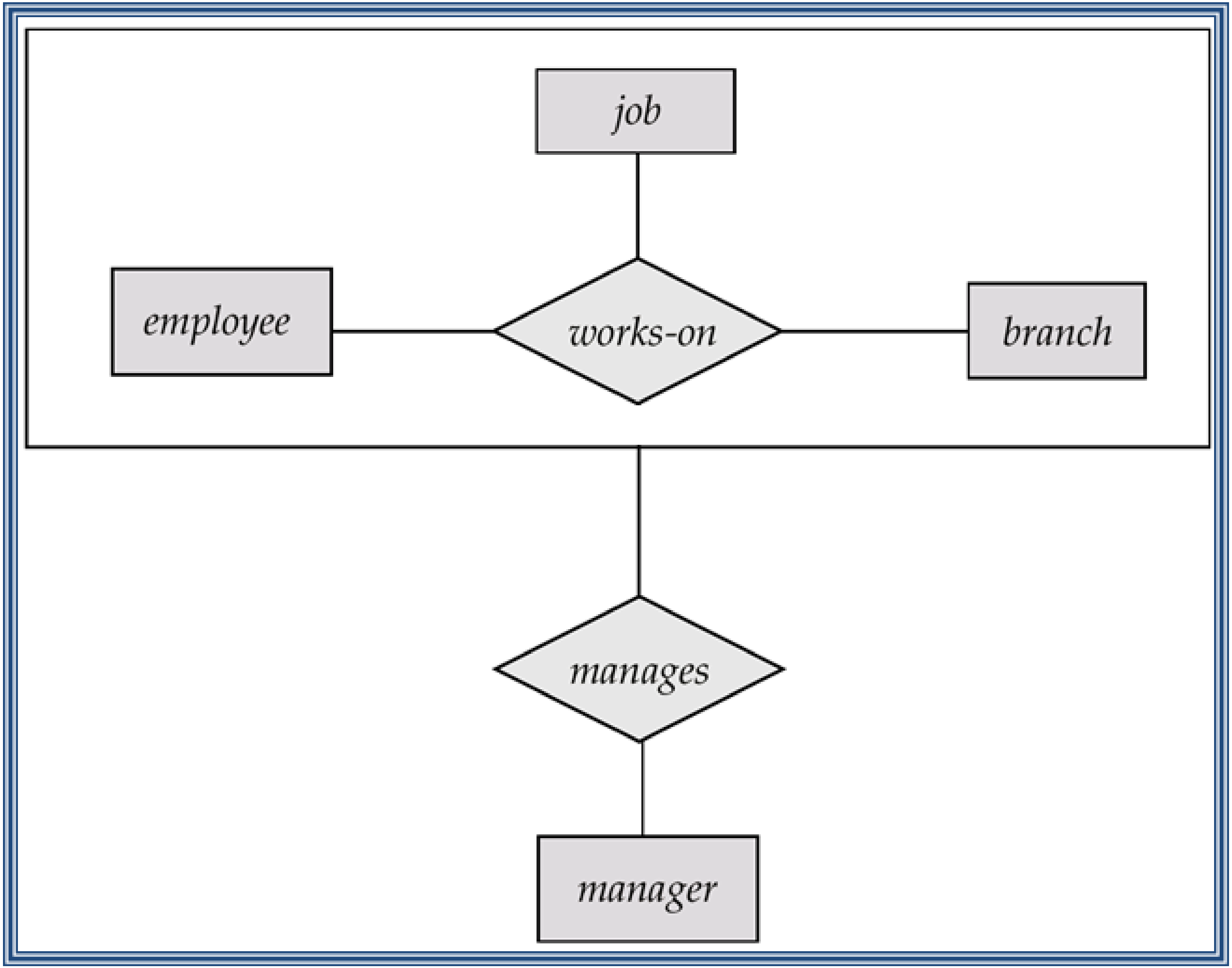


Aggregation

- Consider the ternary relationship works-on, which we saw earlier
- Suppose we want to record managers for tasks performed by an employee at a branch

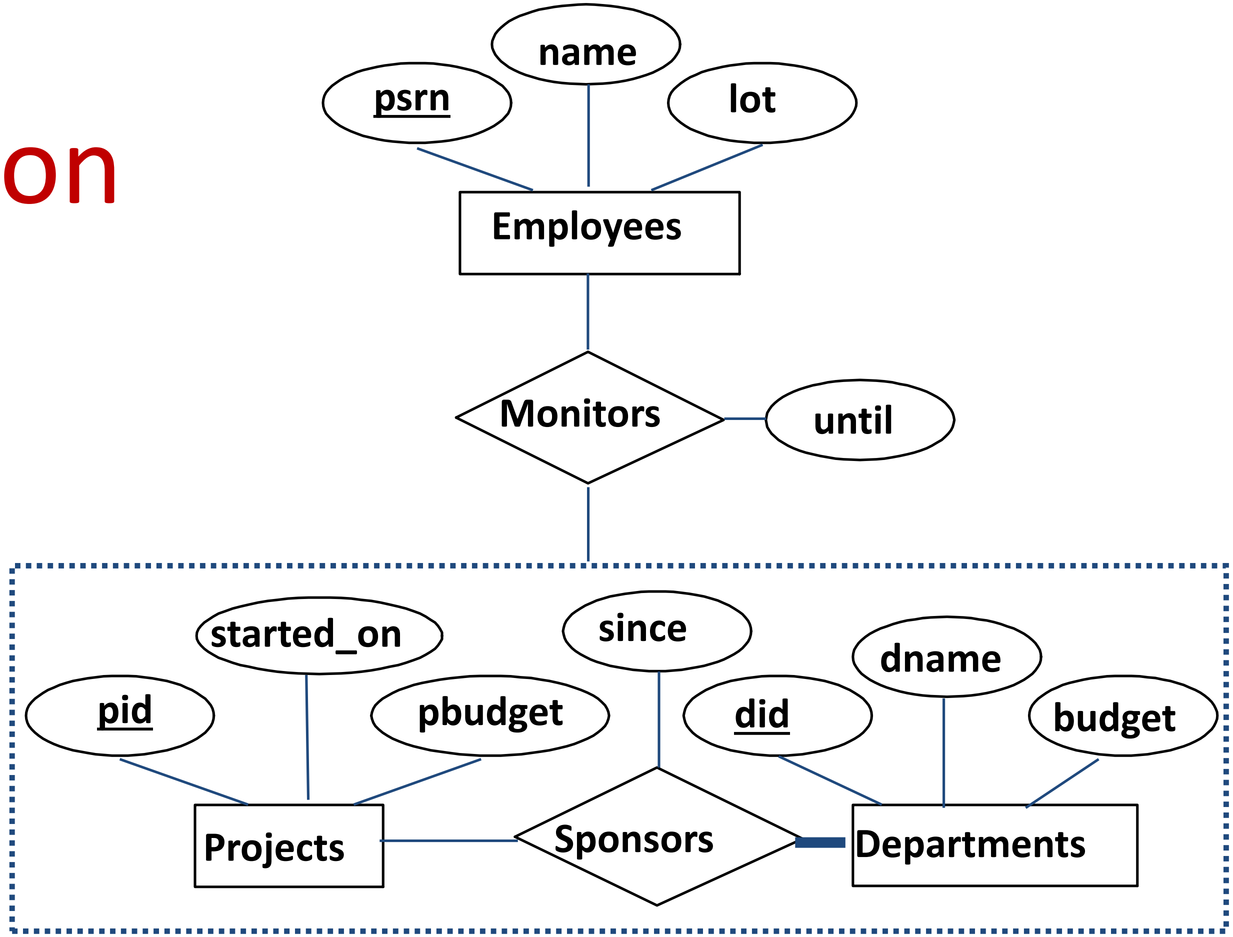


E-R Diagram With Aggregation



Aggregation

- Used when we have to model a relationship involving (entity sets and) a *relationship set*.
  - *Aggregation* allows us to treat a relationship set as an entity set for purposes of participation in (other) relationships.

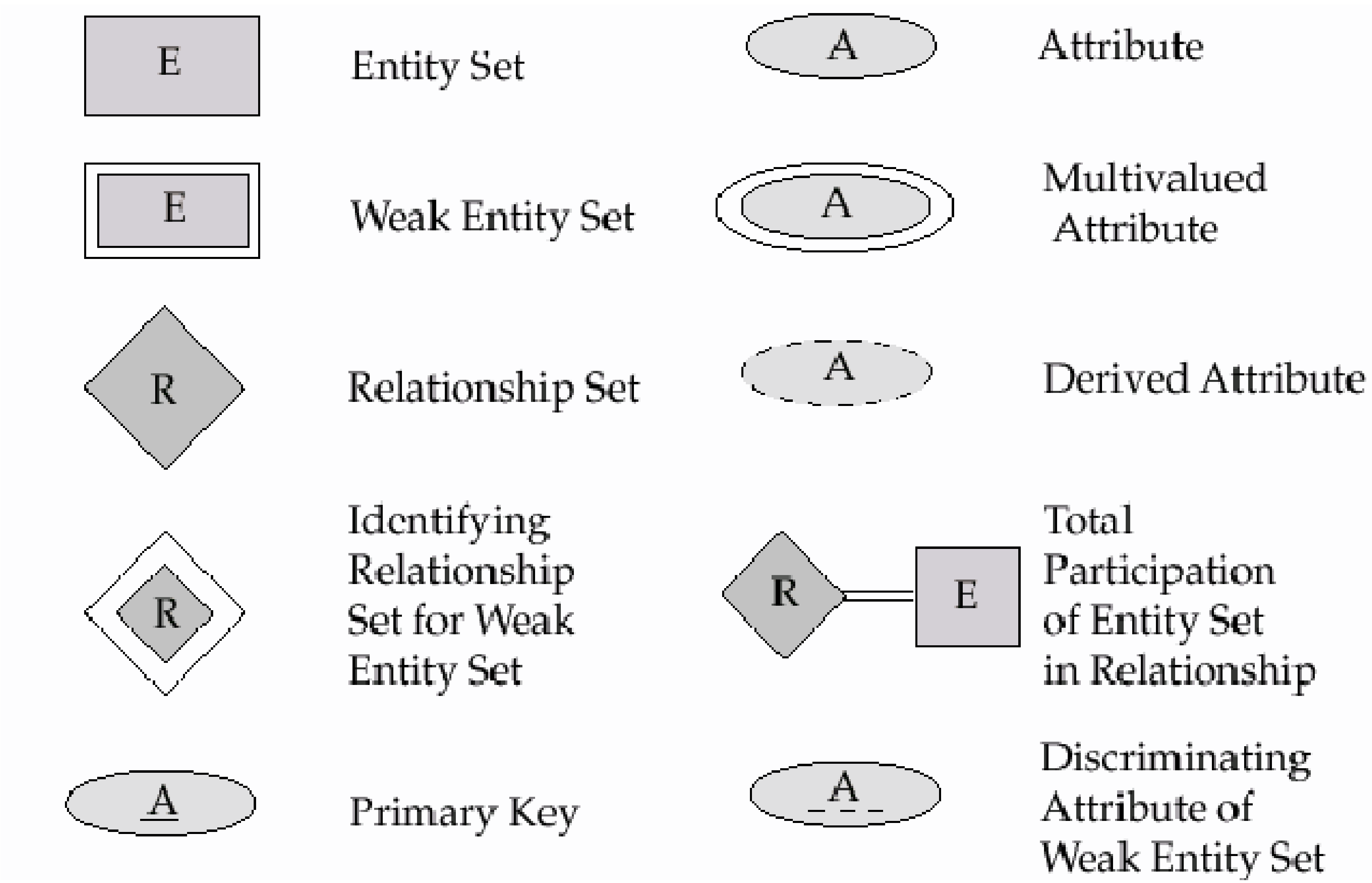


Aggregation vs. ternary relationship:

❖ Monitors is a distinct relationship, with a descriptive attribute. (i.e., until)

❖ Also, can say that each sponsorship is monitored by at most one employee.

Symbols used in ER diagram



Entity Set

Attribute

Weak Entity Set

Multivalued Attribute

Relationship Set

Derived Attribute

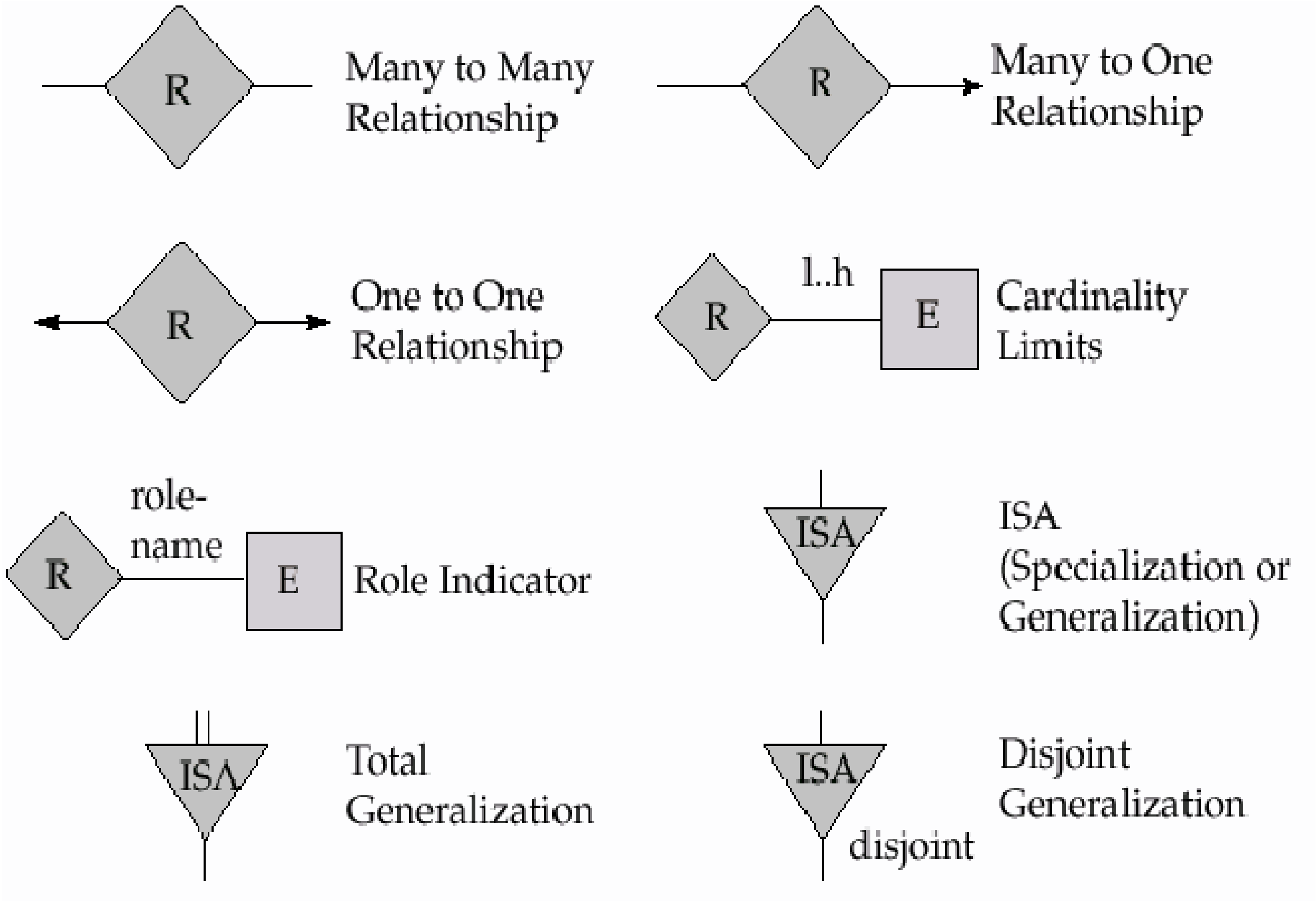
Identifying Relationship Set for Weak Entity Set

Total Participation of Entity Set in Relationship

Primary Key

Discriminating Attribute of Weak Entity Set

Symbols used in ER diagram



Many to Many Relationship

Many to One Relationship

One to One Relationship

Cardinality Limits

Role Indicator

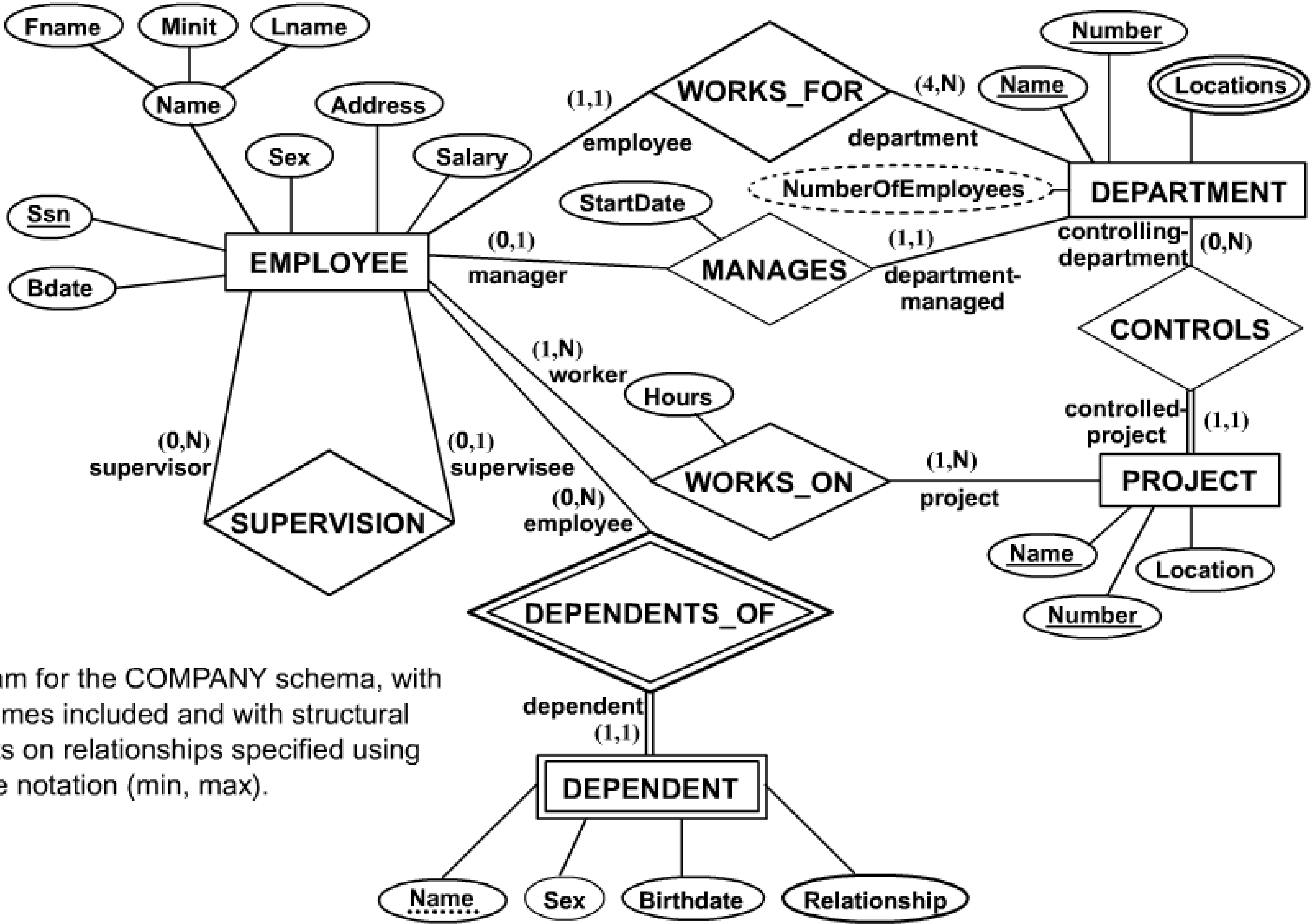
ISA (Specialization or Generalization)

Total Generalization

Disjoint Generalization

COMPANY ER Schema Diagram using (min, max)

Alternative ER Notations



ER diagram for the COMPANY schema, with all role names included and with structural constraints on relationships specified using alternative notation (min, max).

Entity Relationship (ER) modelling

Entity Relationship (ER) modelling

- a way of representing data and relationship among data
- is a tool for communication between requirement analysts and DB designer
- is a graphical representation of the database system
- provides a high-level conceptual data model
- supports the user's perception of the data
- is DBMS and hardware independent
- had many variants
- is composed of entities, attributes, and relationships and constraints(mapping and participation)

BANKING ENTERPRISE DATA REQUIREMENTS:

- DB user says: The name of our bank is Dena bank. We have few branches. Each located in a particular city and is identified by unique name. Basically the bank monitors the assets of each branch.
- DB user says: We have many customers. So its difficult for us to manage details of all in file system. It will be better if you design something good for us. The customers are identified by their account number. Also we need to know their name, street and city where they live. Customers may have accounts and may take loans. A customer may be associated with a particular banker, who may act a loan officer or a personal banker for that customer.

- RA asks: Is the customer_id same as account number?
- DB user says: Yes. One customer can have many accounts or some can have joint account.
- RA asks: If a customer wants a loan, is it necessary to have an account with the bank?
- DB user: No. not so necessary.
- DB user says: We also find difficult to manage employee details because many time new people joins and old people retires or some leaves job. So it will be good if you give us some structure where employee data can be stored in an organized way.

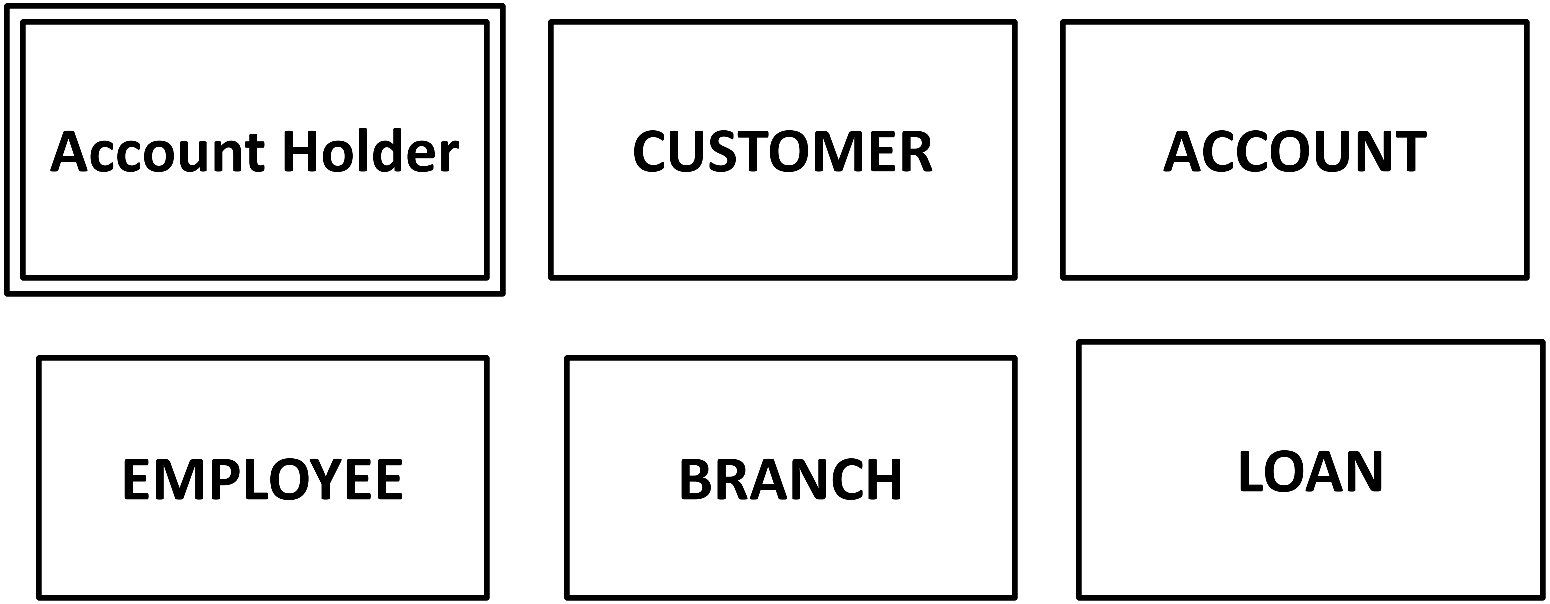
- DB user says: Bank employees are identified by their employee_id values. The bank administration need name, telephone number and the employee’s dependents, and the employee_id of employee’s manager. The bank also keeps track of the employee’s start date and, thus, length of employment.

- DB user says: The bank offers two types of accounts-Savings and Checking account. Each account has unique account number. The bank maintains a record of each account’s balance, and the most recent date on which the account was accessed by each customer holding the account. In addition, each saving account has an interest rate, and overdrafts are recorded for each checking account.

- DB user says: A loan originates at a particular branch and can be held by one or more customers. A loan is identified by a unique loan number. For each loan, the bank keeps track of the loan amount and loan payments. Although a loan payment number does not uniquely identify a particular payment among those for all bank’s loans, a payment number does identify a particular payment for a specific loan. The date and amount are recorded for each payment.

Let us draw an ER diagram for a Banking system

ENTITY AND ENTITY SETS



Account Holder

CUSTOMER

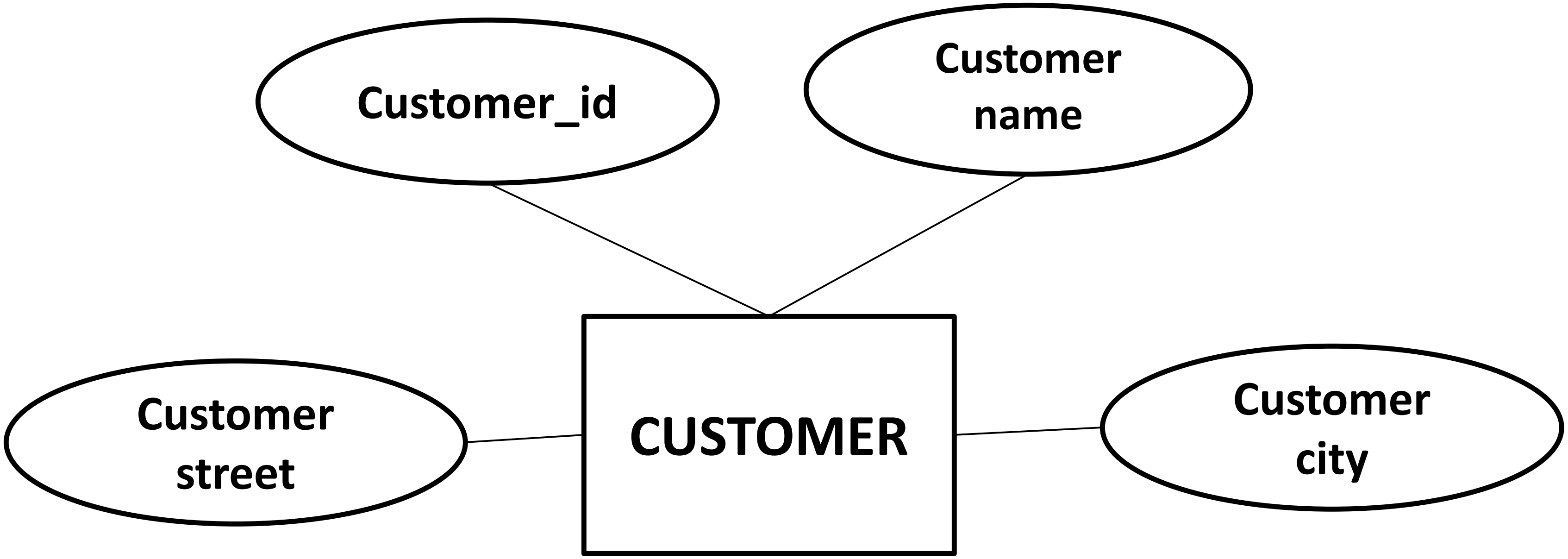
ACCOUNT

EMPLOYEE

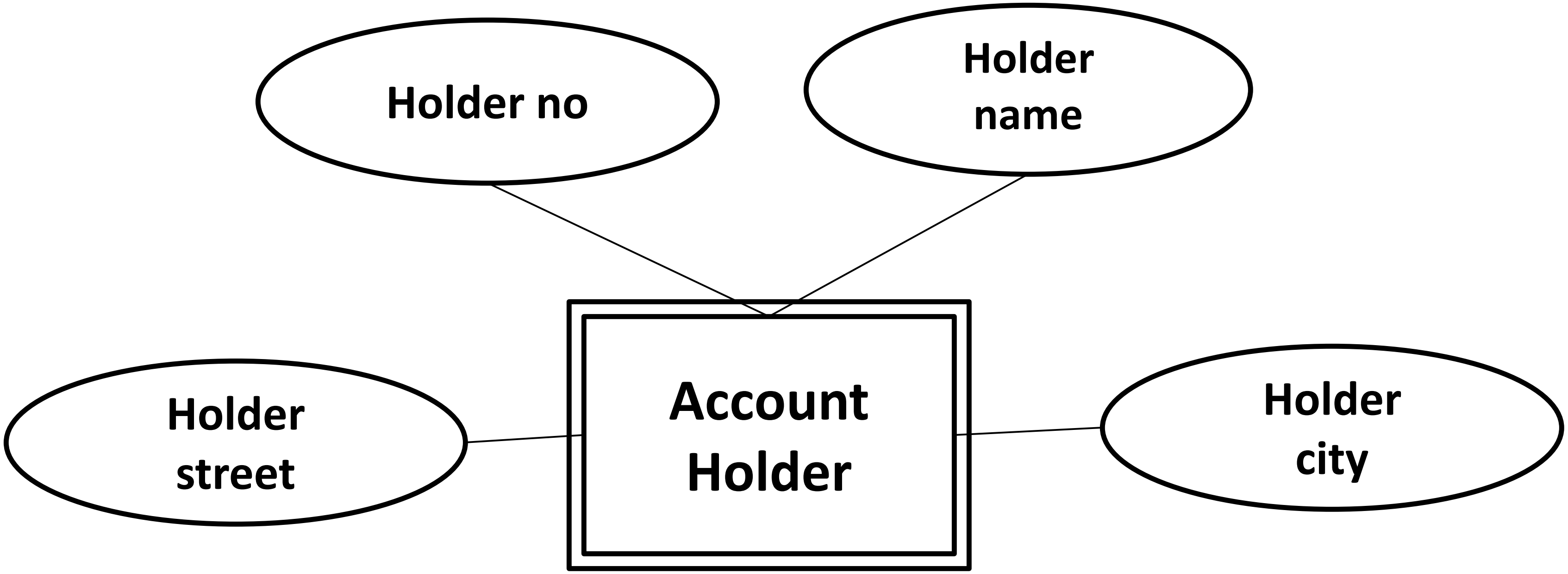
BRANCH

LOAN

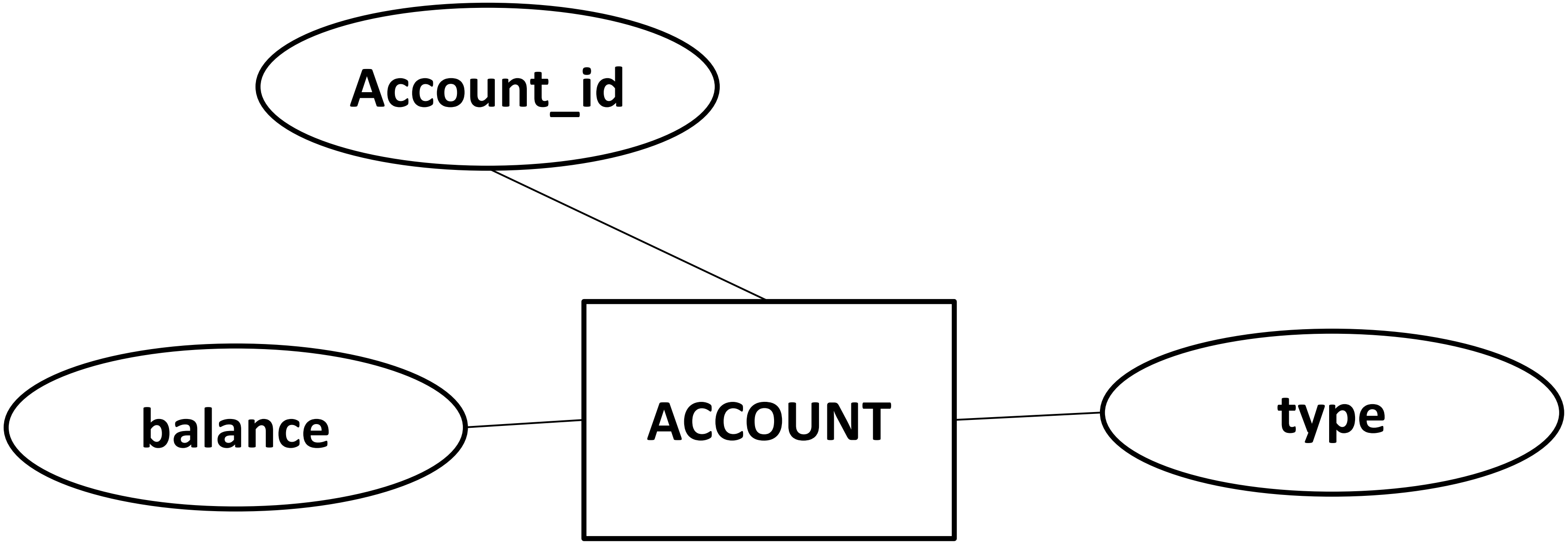
ATTRIBUTE


Customer_id
Customer name
Customer street
CUSTOMER
Customer city

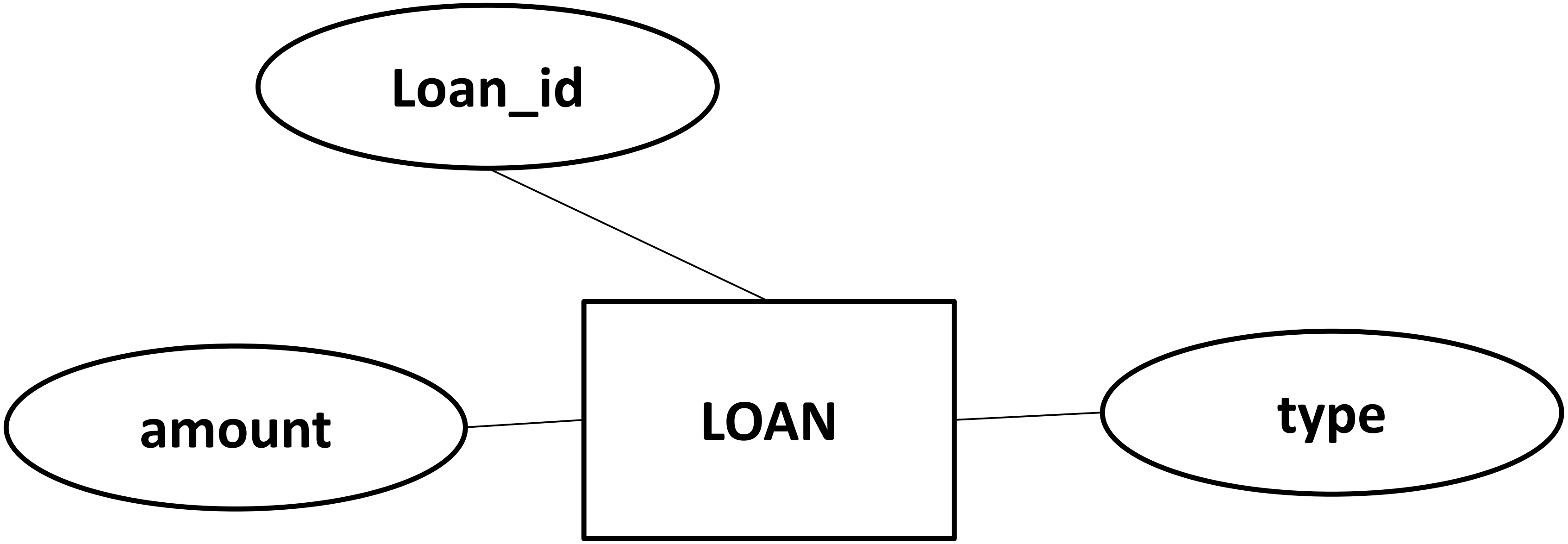
ATTRIBUTE


Holder no
Holder name
Account Holder
Holder street
Holder city

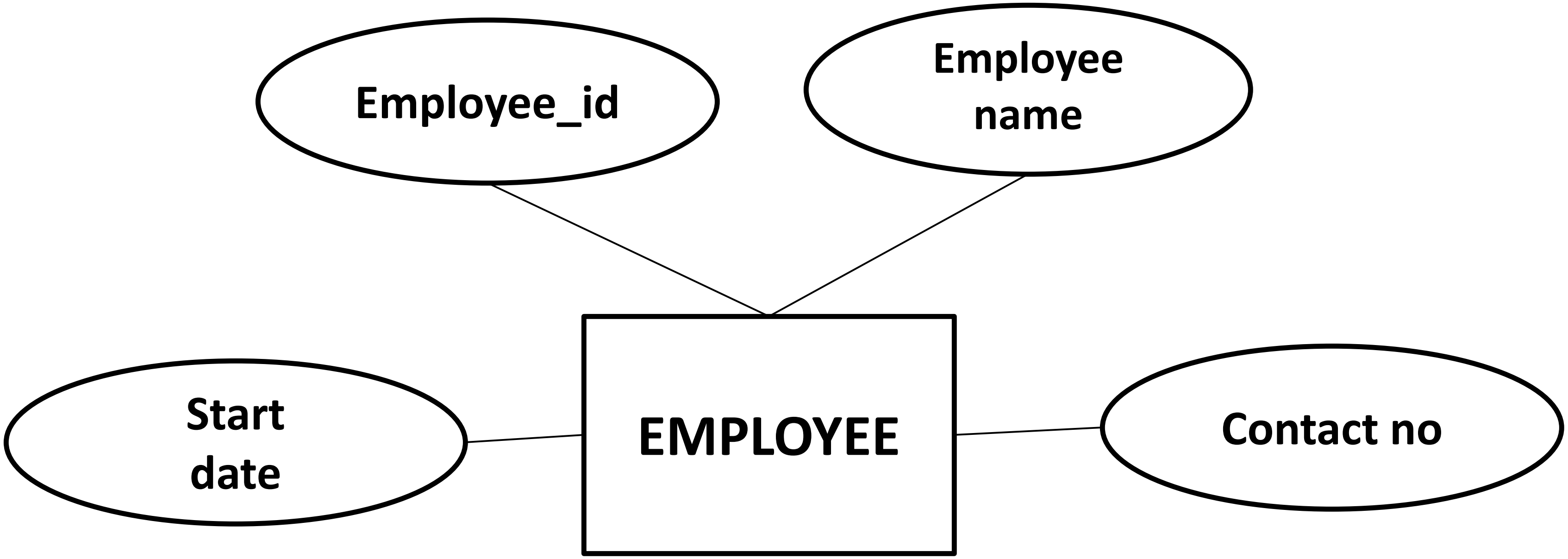
ATTRIBUTE



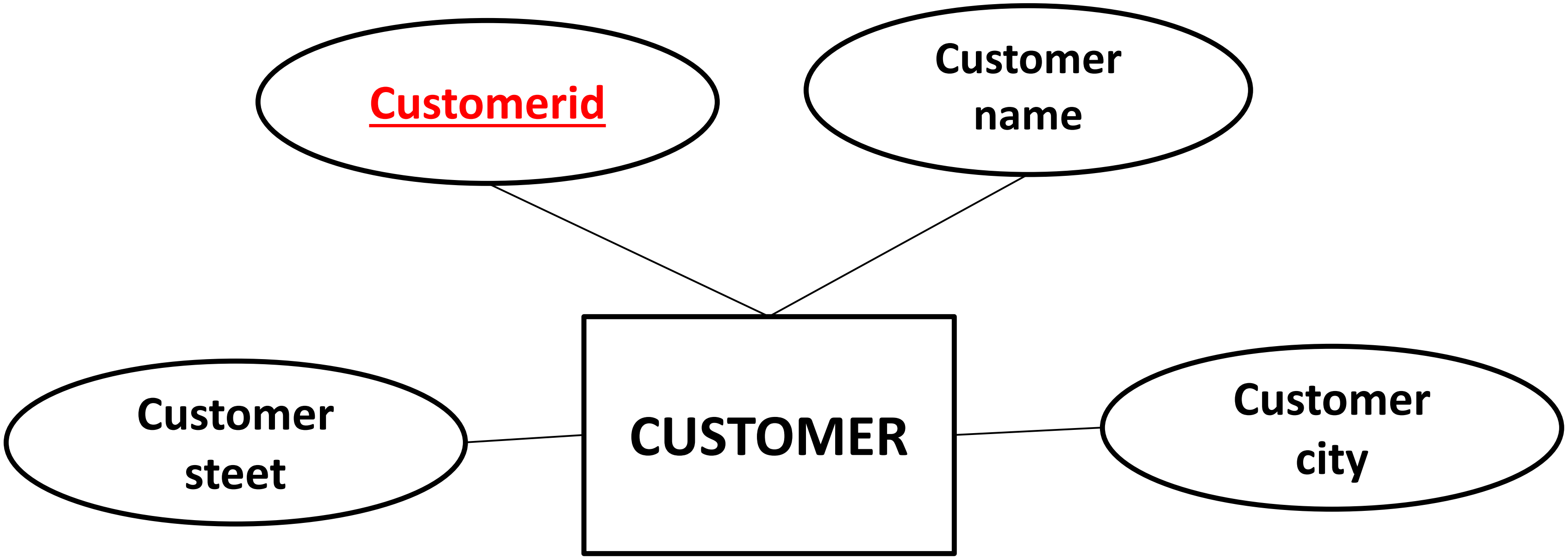
ATTRIBUTE



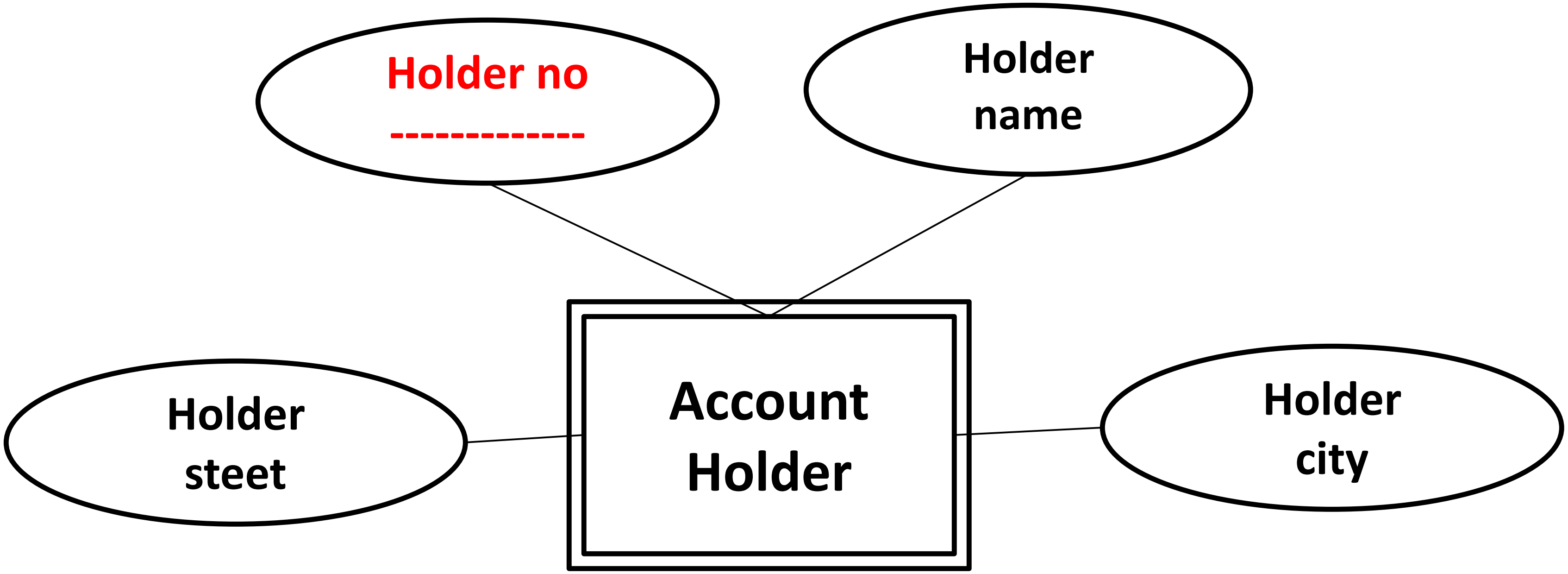
ATTRIBUTE



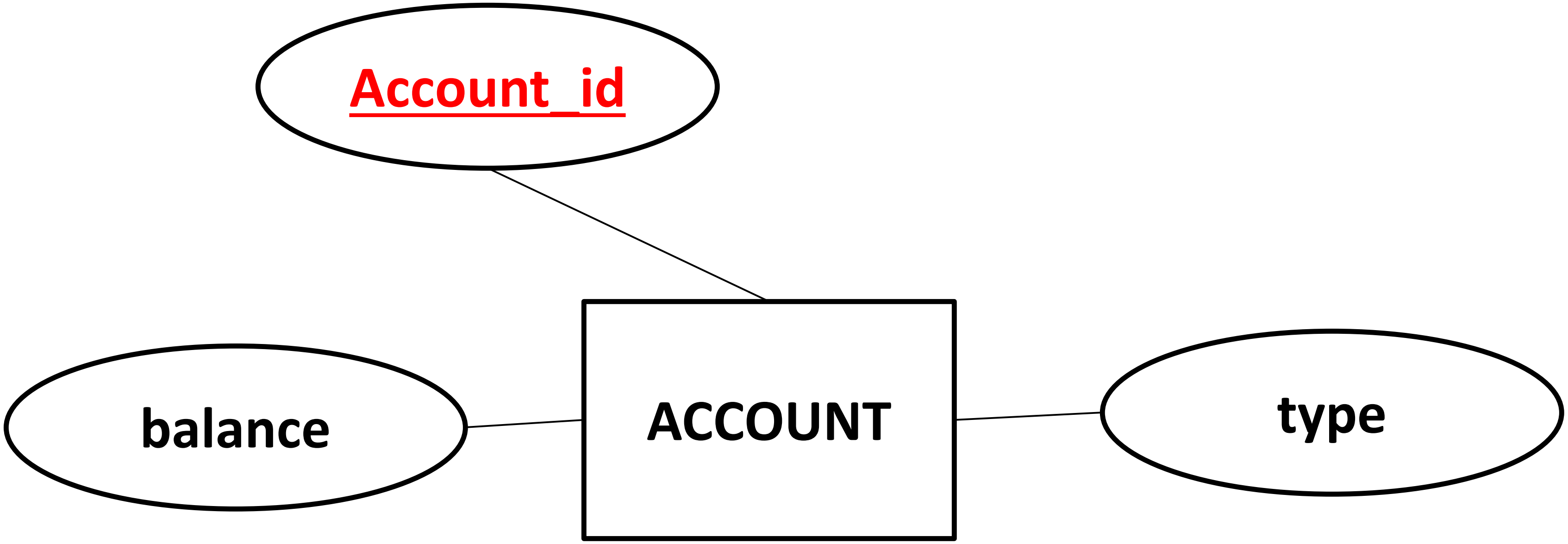
KEY


Customerid
Customer name
Customer steet
CUSTOMER
Customer city

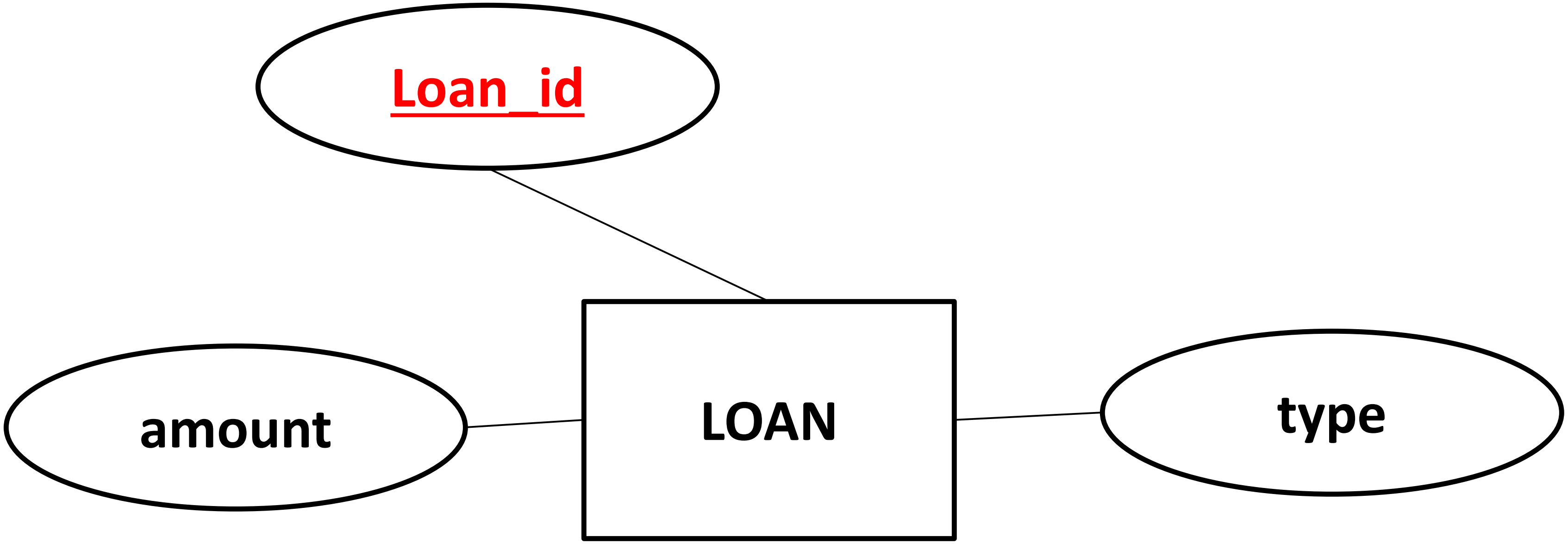
ATTRIBUTE


Holder no
Holder name
Account Holder
Holder steet
Holder city

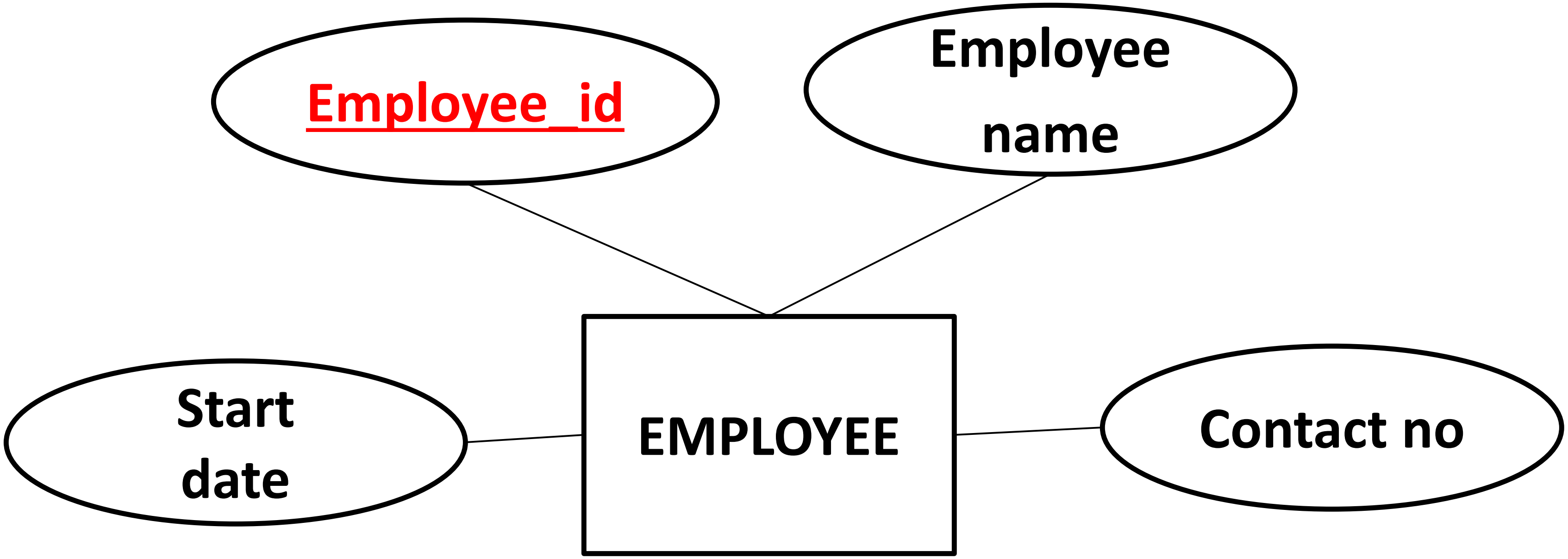
KEY



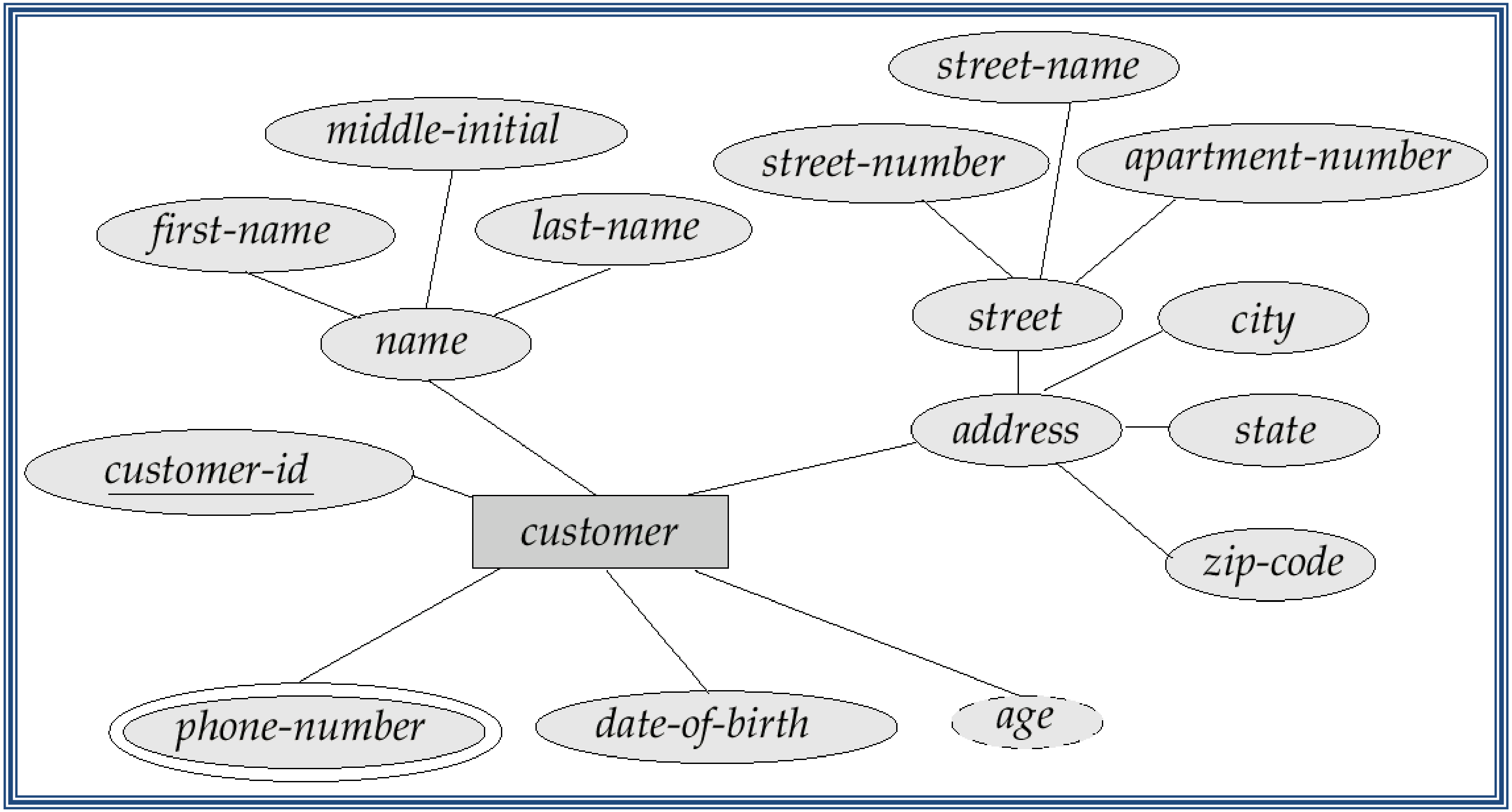
KEY



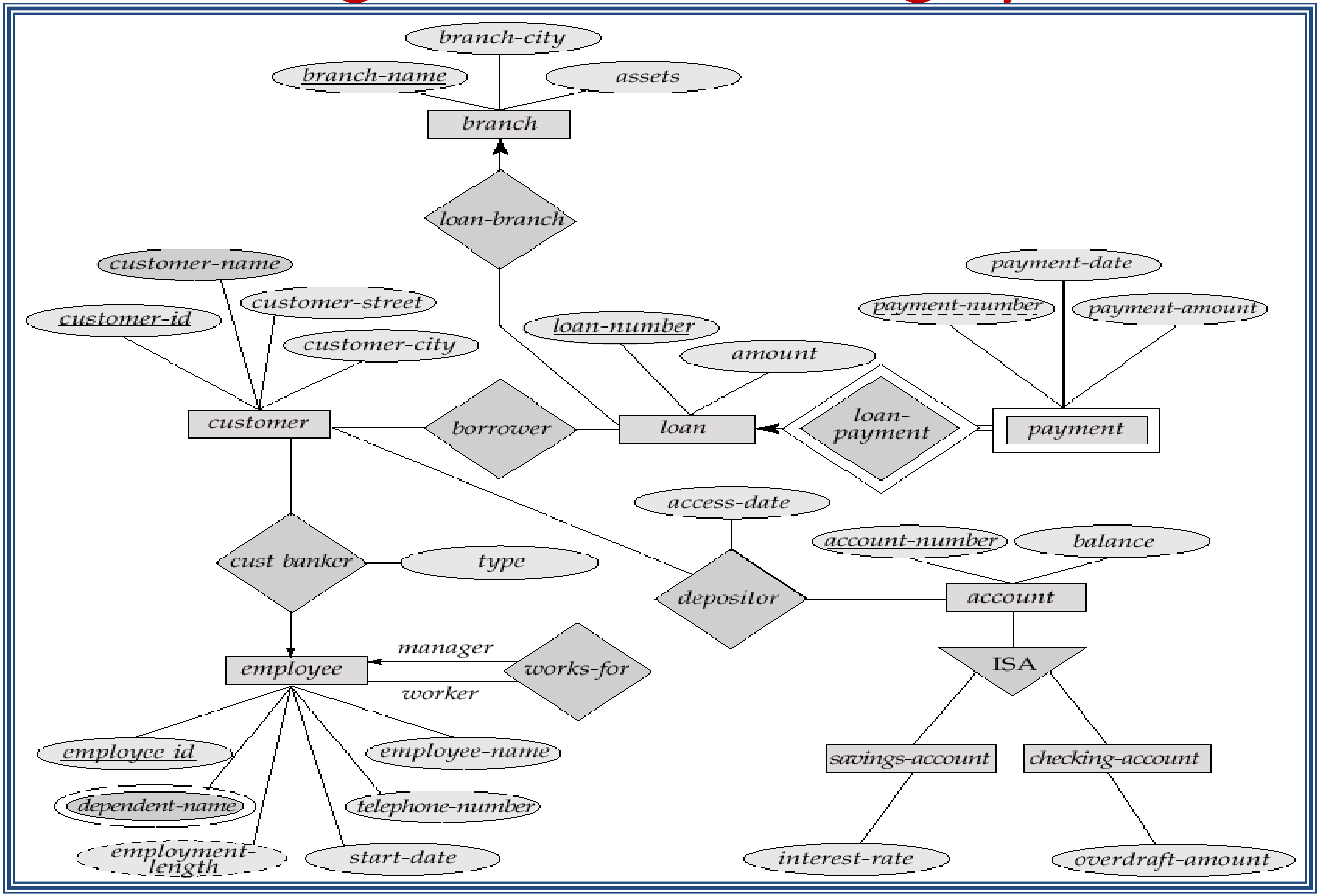
KEY



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



ER Diagram for banking system



Constructing an ER model

Before beginning to draw the ER model, read the requirements specification carefully. Document any assumptions you need to make.

- Identify entities - list all potential entity types. These are the object of interest in the system. It is better to put too many entities in at this stage and them discard them later if necessary.
- Remove duplicate entities - Ensure that they really separate entity types or just two names for the same thing.
- Also do not include the system as an entity type

Constructing an ER model…

- List the attributes of each entity (all properties to describe the entity which are relevant to the application).
  - Ensure that the entity types are really needed.
  - are any of them just attributes of another entity type?
  - if so keep them as attributes and cross them off the entity list.
  - Do not have attributes of one entity as attributes of another entity!
- Mark the primary keys.
  - Which attributes uniquely identify instances of that entity type?
  - This may not be possible for some weak entities.
- Define the relationships
  - Examine each entity type to see its relationship to the others.

Constructing an ER model…

- Describe the cardinality of the relationships
  - Examine the constraints between participating entities.

- ER modelling is an iterative process, so draw several versions, refining each one until you are happy with it. Note that there is no one right answer to the problem, but some solutions are better than others!

How about doing an ER design interactively on the paper?

Problem statement

Design a database that manages information about publishers, authors, and books. A publisher has a name and address for the headquarters. Each publisher also has a set of branches, each branch having an address and two phone numbers. An author has a name and an address. A book is published by a publisher and has a list of authors associated with it. An author can write several books & a book can be published by only one publisher. Draw an E-R diagram for the given system.