DATABASE MANAGEMENT SYSTEM
MCA
I Semester
AUG 2022

Database System Concepts

Abraham Silberschatz, Henry F. Korth, S. Sudarshan

Entity-Relationship Model

ER-Model

- Used for Conceptual Modeling of the Database schema.
- Meant for Human comprehension.
- To tell End User, what is understood about their Domain.
- High-level database design without implementation details.
- DBMS independent.
- Building Blocks: Entity, Attribute and Relationship

Entity

- An **entity** is an object that exists in the real world and is distinguishable from other objects.
- In real world, an entity has a set of properties and the values for those properties identify it.
  - Example: Faculty, Student, company, event, plant ,Course, Account
  - Properties of Student –RegNo, Name, Course, Phone.
  - Values of these property-(180370123,'Rajesh','MCA',9876999756) is an **entity**

Entity Sets

- An **entity set** is a set of entities of the same type that share the same properties.
  - Example: set of all students who share same properties.
  - {(200970123,'Rajesh','MCA',9876999756) ,

    (200968124,'Ramesh','DSE',8876999756) , ……..}

- Example:
  - Student entity set –Collection of all student entities.
  - Faculty entity set –Collection all faculties in a college

Entity Sets- instructor and student

ID	name
76766	Crick
45565	Katz
10101	Srinivasan
98345	Kim
76543	Singh
22222	Einstein

instructor

ID	name
98988	Tanaka
12345	Shankar
00128	Zhang
76543	Brown
76653	Aoi
23121	Chavez
44553	Peltier

student

Instructor={ (76766,Crick),45565,Katz),(10101,Srinivasan),…(22222, Einstein) }

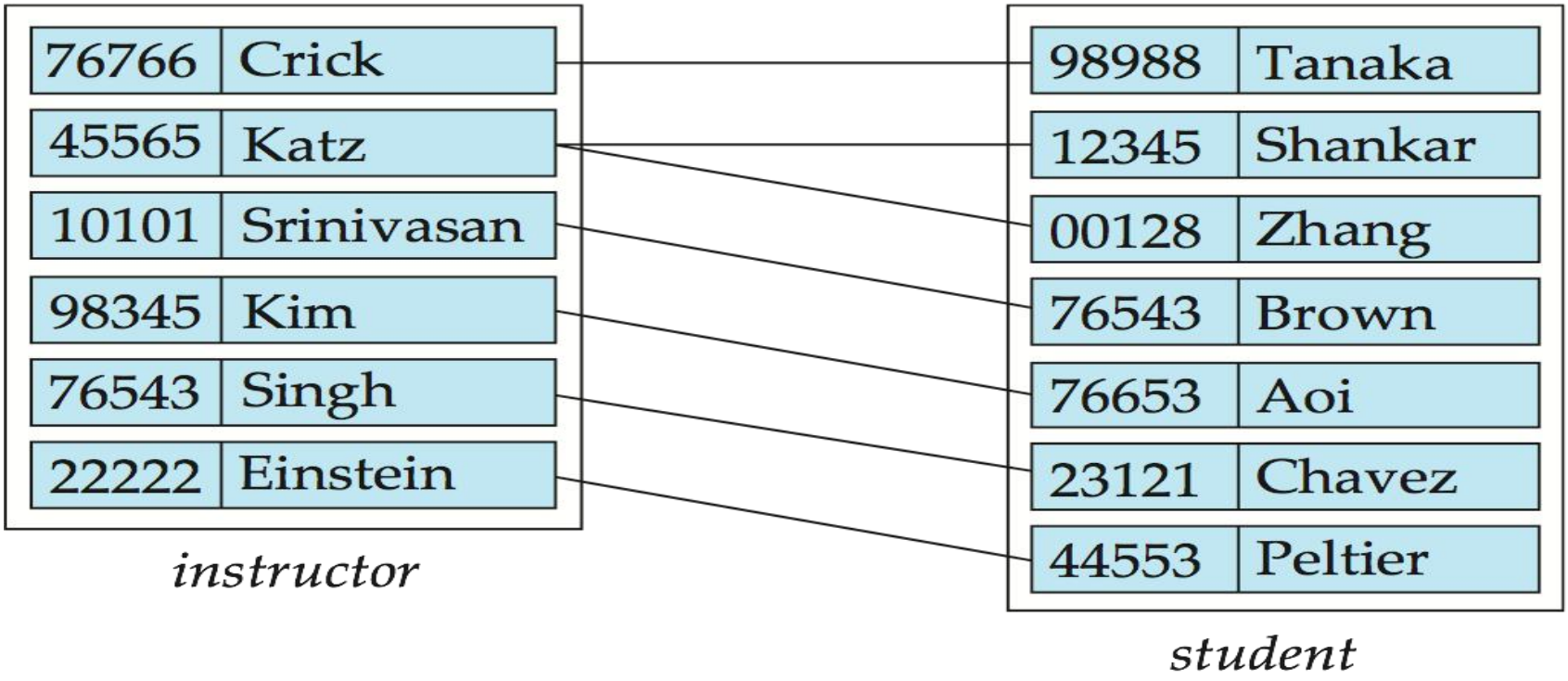
Student={ (98988,Tanaka),(12345,Shankar),(00128,Zhang),….(44553,Peltier) }

PRIMARY KEY: Each entity has one or more combination attributes can be used to identify each entity uniquely.
The **ID** attribute is used to identify instructors uniquely.

Relationship Sets

Assume that – we want to model the information such as – ***Crick is advisor for Tanaka ; Katz is advisor for Shankar & Zhang ; Srinivasan is advisor for Brown***; and so on.

In other words, we are interested in representing information that Some instructor will be advisor for Some Student/Students(Association between Instructor & Student).



Relationship set ={ **(76766,Crick , 98988,Tanaka) ,(45565,Katz,12345,Shankar), (45565,Katz,00128,Zhang),(10101,Srinivasan),… (22222, Einstein,44553,Peltier)** }

Relationship Sets

Consider formal definition of Relationship set.

- A **relationship** is an association among several entities

  Example:

  (22222 ,Einstein) *advisor* (44553 ,Peltier)

  instructor entity — relationship set — *student* entity

Collection of such relationship is Relationship set

- A **relationship set** is a mathematical relation among $n \geq 2$ entities, each taken from entity sets $E_1, E_2, .. E_n$.

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

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

i.e. (**44553 ,Peltier, 22222, Einstein)** a **relationship set element**(instance)

*Every instance in a entity set is distinguishable from other entities using **primary key** , in the same way every relationship in a relationship set is distinguishable.*

Primary Key of relationship set is – $PK(E_1) \cup PK(E_2) \ldots PK(E_n)$

Example:

(44553,22222) $\in$ *advisor.*

Relationship Sets (Cont.)

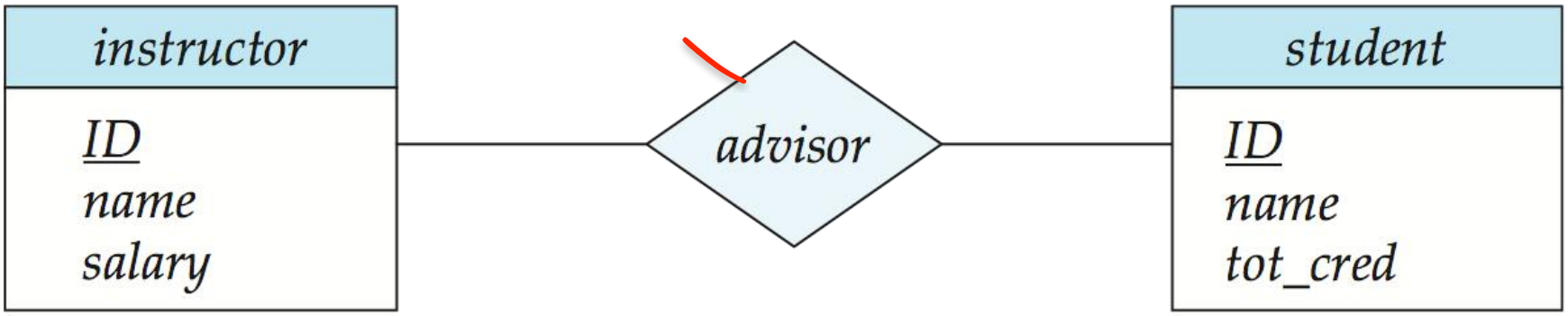
- The association between entity sets is referred to as **participation.**
- A **relationship instance** (**ex: (45565,Katz-00128,Zhang)**) in an E-R schema represents an association between the named entities (**ex**: **(45565,Katz) an Instructor entity & (00128,Zhang) a Student entity**) in the real-world enterprise that is being modeled.
- The function that an entity plays in a relationship is called entity's role.

Relationship set ={ **(76766,Crick - 98988,Tanaka) ,(45565,Katz- 12345,Shankar),**

(45565,Katz-00128,Zhang),…,… (22222, Einstein-44553,Peltier) }

E-R Diagrams

Following is the diagrammatic representation of Relationship(discussed in slide 11) in ER Model



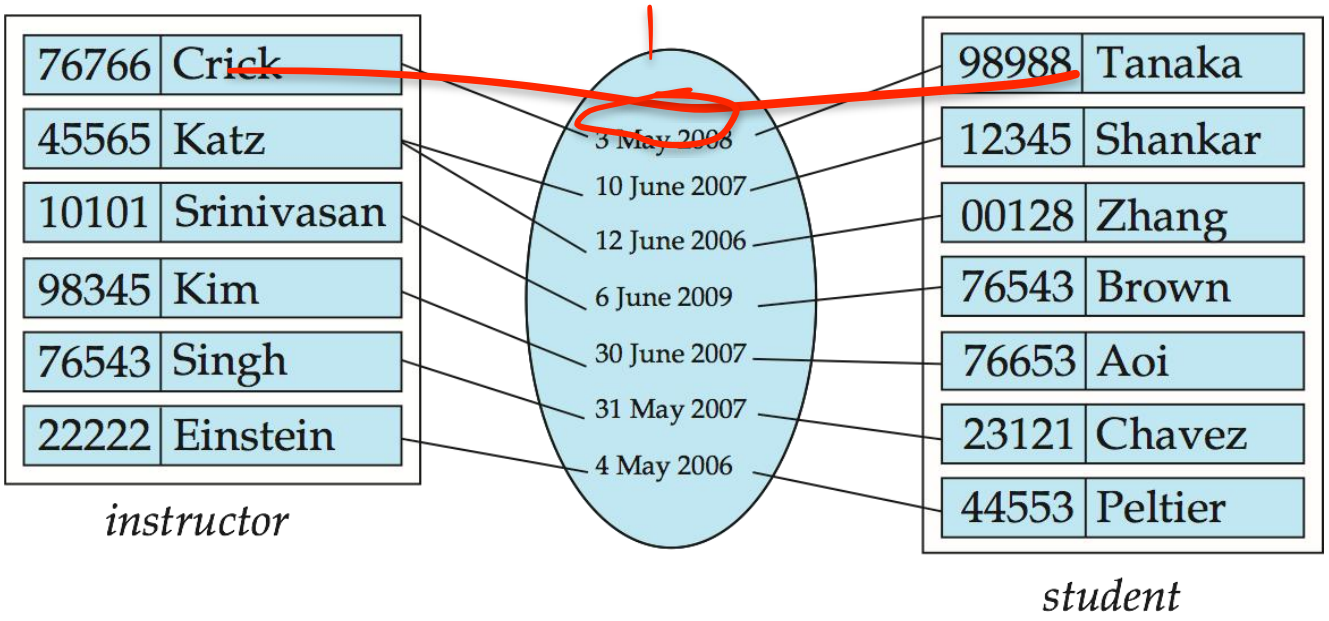
- Rectangles represent entity sets.
- Diamonds represent relationship set.
- Attributes listed inside entity rectangle.
- Underline indicates primary key attributes.

Relationship Sets (Cont.)

- A relationship may also have attributes called **descriptive attributes**.

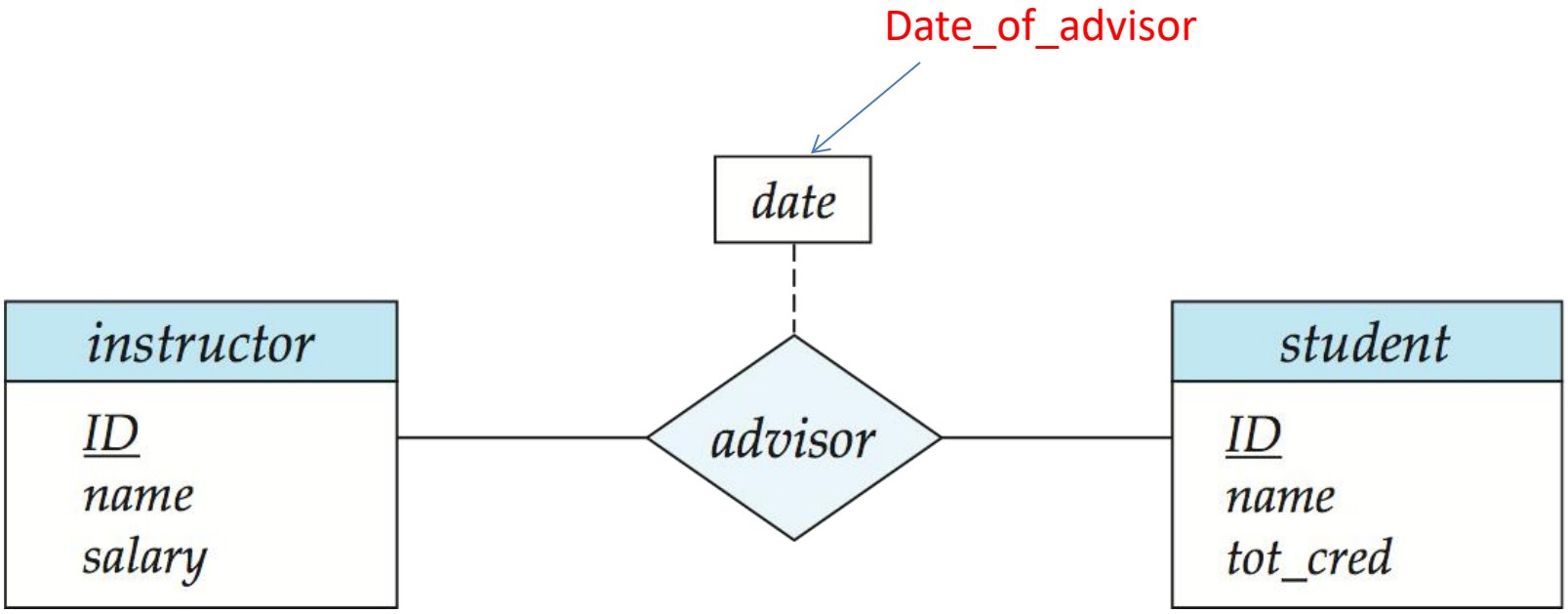
Assume that we want to store information that –An Instructor **I** became an Advisor to Student **S** on the date **D**.

Ex: **(76766,Crick)** became Advisor to a Student **(98988,Tanaka)** on 3rd May 2008.



3rd May 2008 date is property of neither Instructor nor Student, but it has meaning when referred w.r.t Advisor relation ship between Instructor & Student.

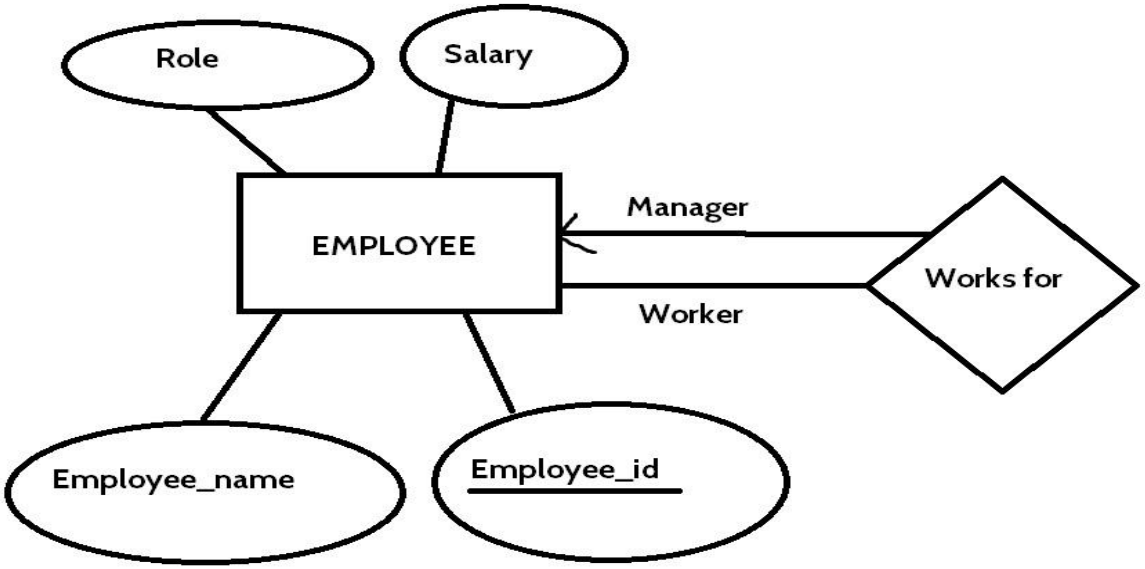
Relationship Sets with Attributes



The design decision of where to place descriptive attributes in such cases—as a relationship or entity attribute—should reflect the characteristics of the enterprise being modeled.

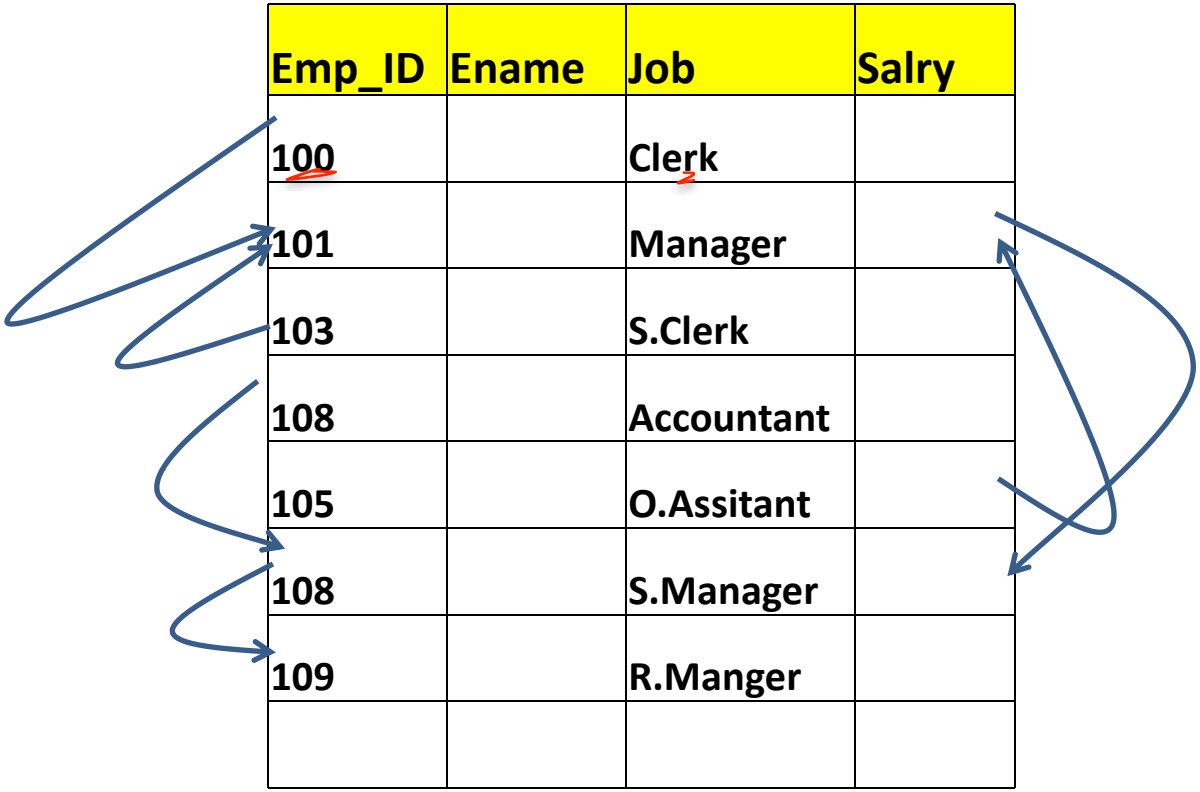
Recursive Relationship set

- If the same entity sets participates in a relationship set more than once in different roles-Recursive relationship



Recursive Relationship..

- Example: Sample relationship between EMP entities.



Emp_ID	Ename	Job	Salry
100		Clerk	
101		Manager	
103		S.Clerk	
108		Accountant	
105		O.Assitant	
108		S.Manager	
109		R.Manger	

(100,..,Clerk,..) is an entity in EMP entity set and is having relationship (association) with another entity (101,..,Manger,..) in the same Entity set(i.e. EMP)

Recursive Relationship when converted into schema, it looks as below-

ManagerNo is foreign key Referencing EMP which models the recursive relationship shown in ER diagram.

We will study, how ER diagram in converted into schema(set of Tables) later in this chapter.

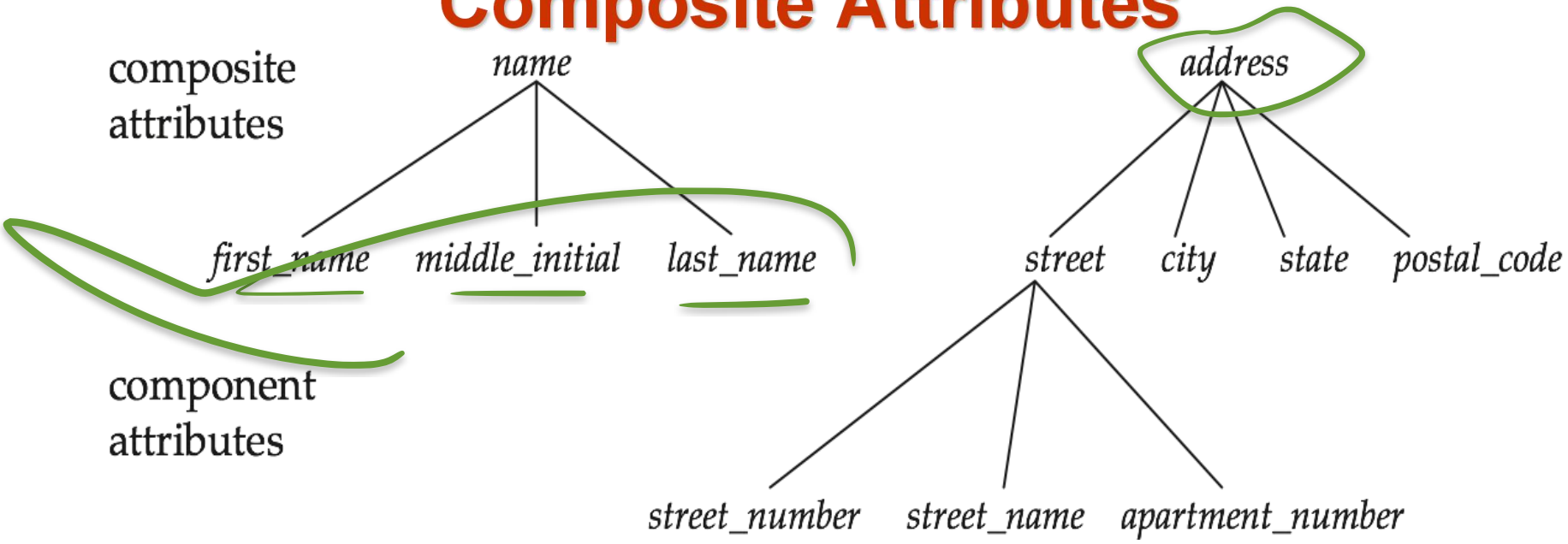
Emp_ID	Ename	Job	Salry	ManagerNo
100		Clerk		101
101		Manager		
103		S.Clerk		101
108		Accountant		109
105		O.Assitant		
108		S.Manager		
109		R.Manger		109

Attributes

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

    instructor = (ID, name, street, city, salary)
    course= (course_id, title, credits)
- **Domain** – the set of permitted values for each attribute
- **Attribute types:**
  - **Simple** and **composite** attributes.
    - **Simple-** having atomic or indivisible values.
      - Example: Department_Name , State, city
    - **Composite-** having several components in the value.
      - Example: Contact_number attribute further having components- STDCode and Phone Number

Composite Attributes



Single-valued and **multivalued** attributes

Single- having only one value rather than a set of values.

Example: Birth_Place

Multivalued - having a set of values rather than a single value

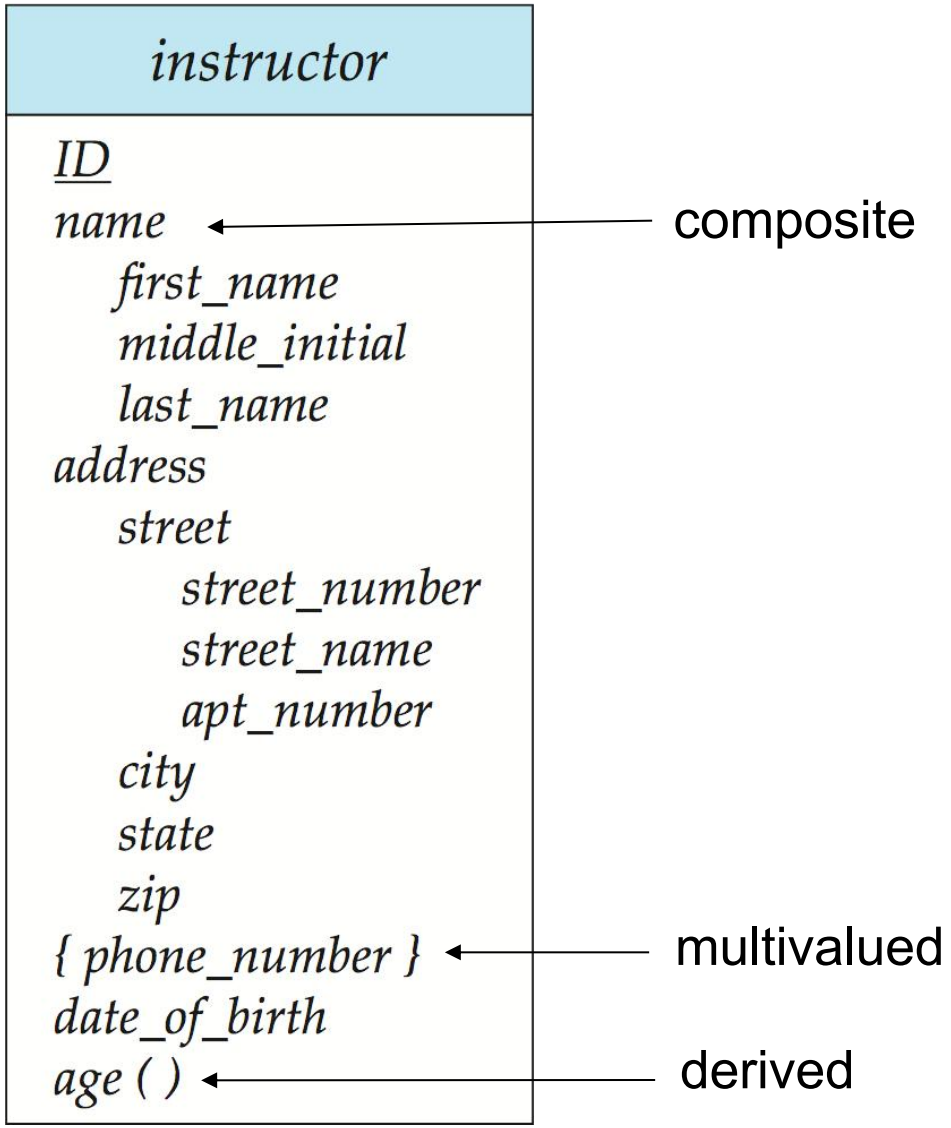
Example: *phone_numbers ,Emails of a person , Degrees acquired by a faculty*

Derived attributes

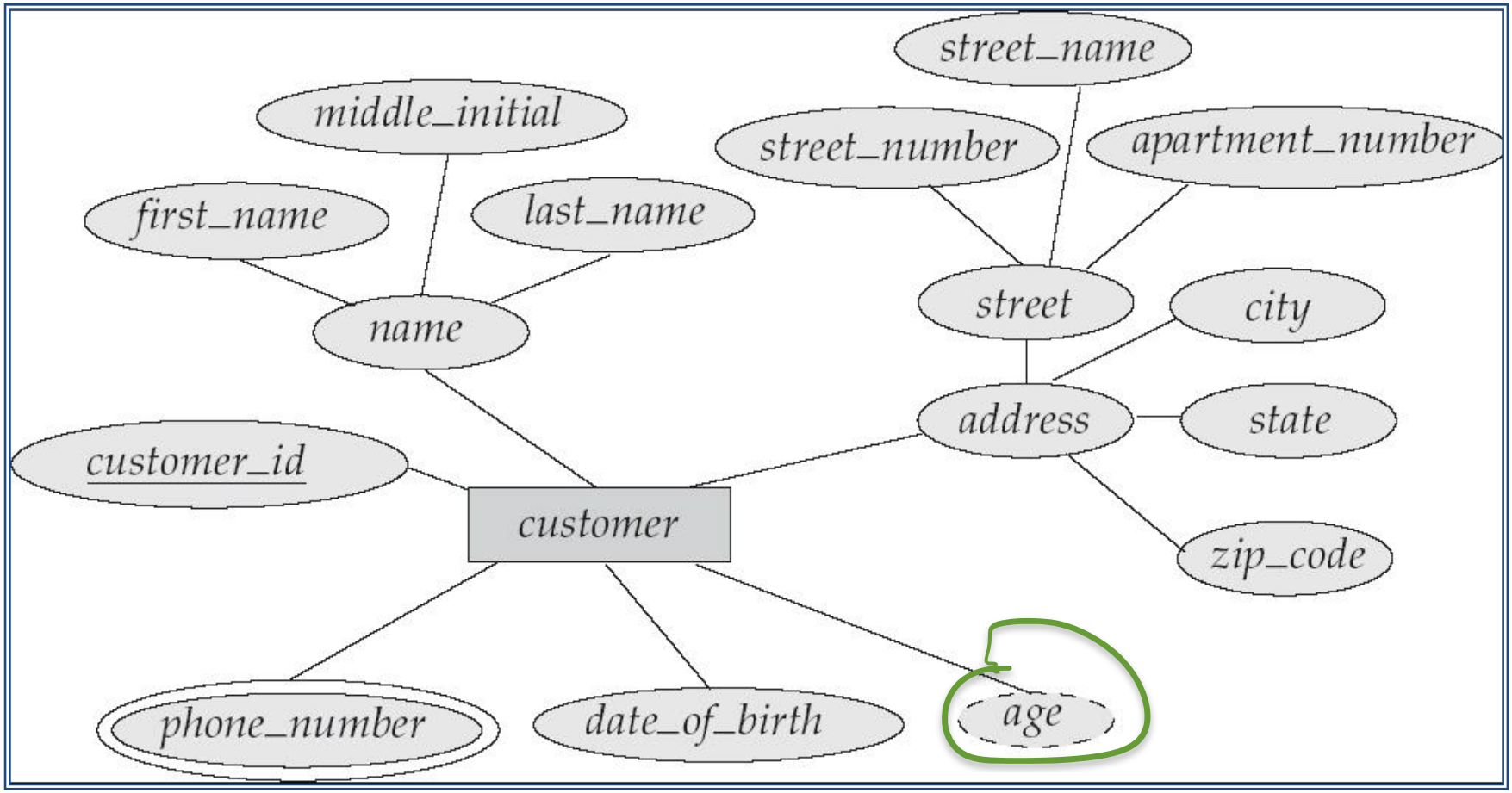
Can be computed from other attributes

Example: *Age, given date_of_birth*

Entity With Composite, Multivalued, and Derived Attributes

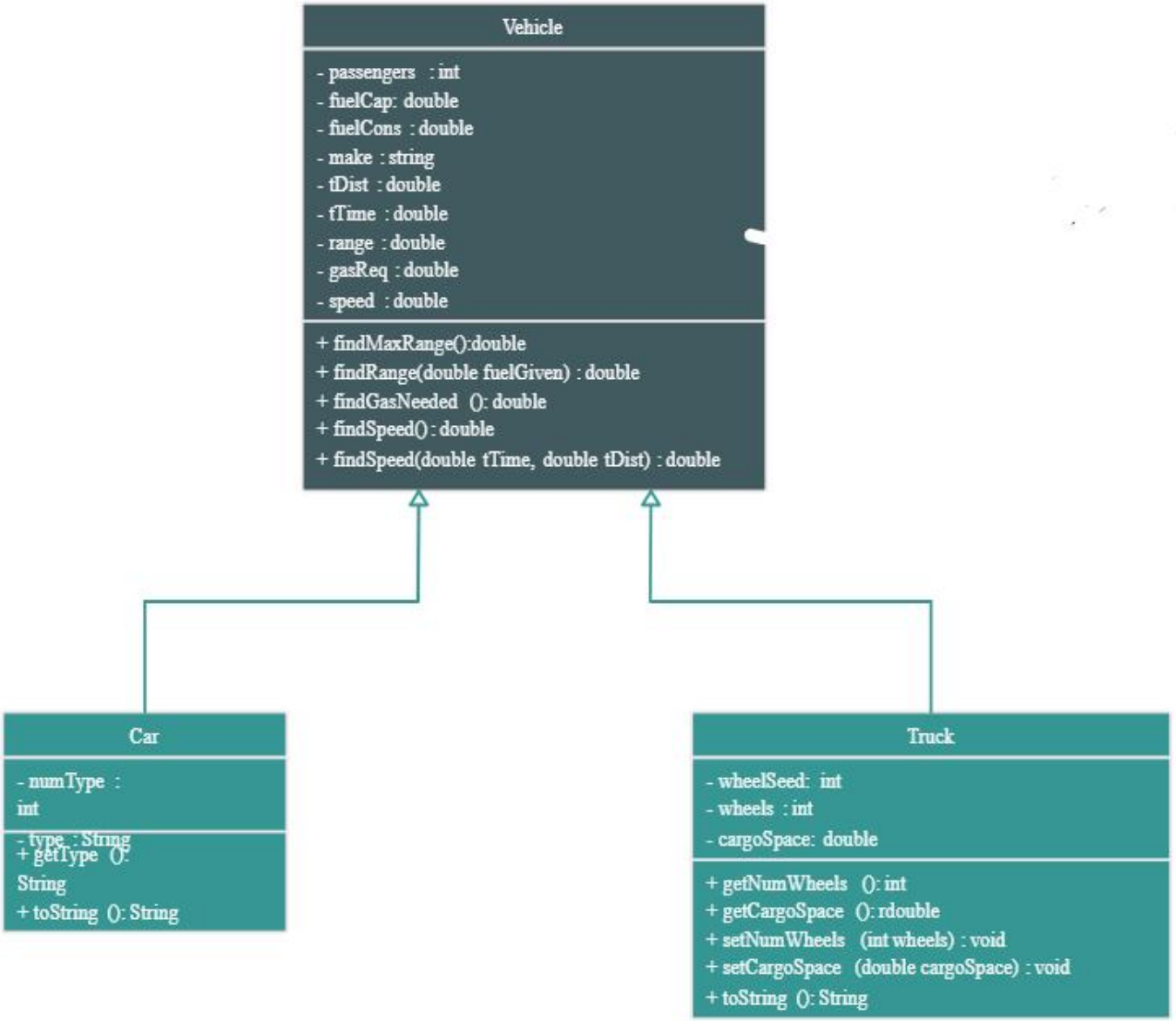


Entity With Composite, Multivalued, and Derived Attributes



In 4th Edition

Similar to UML- Class Diagram



Degree of a relationship

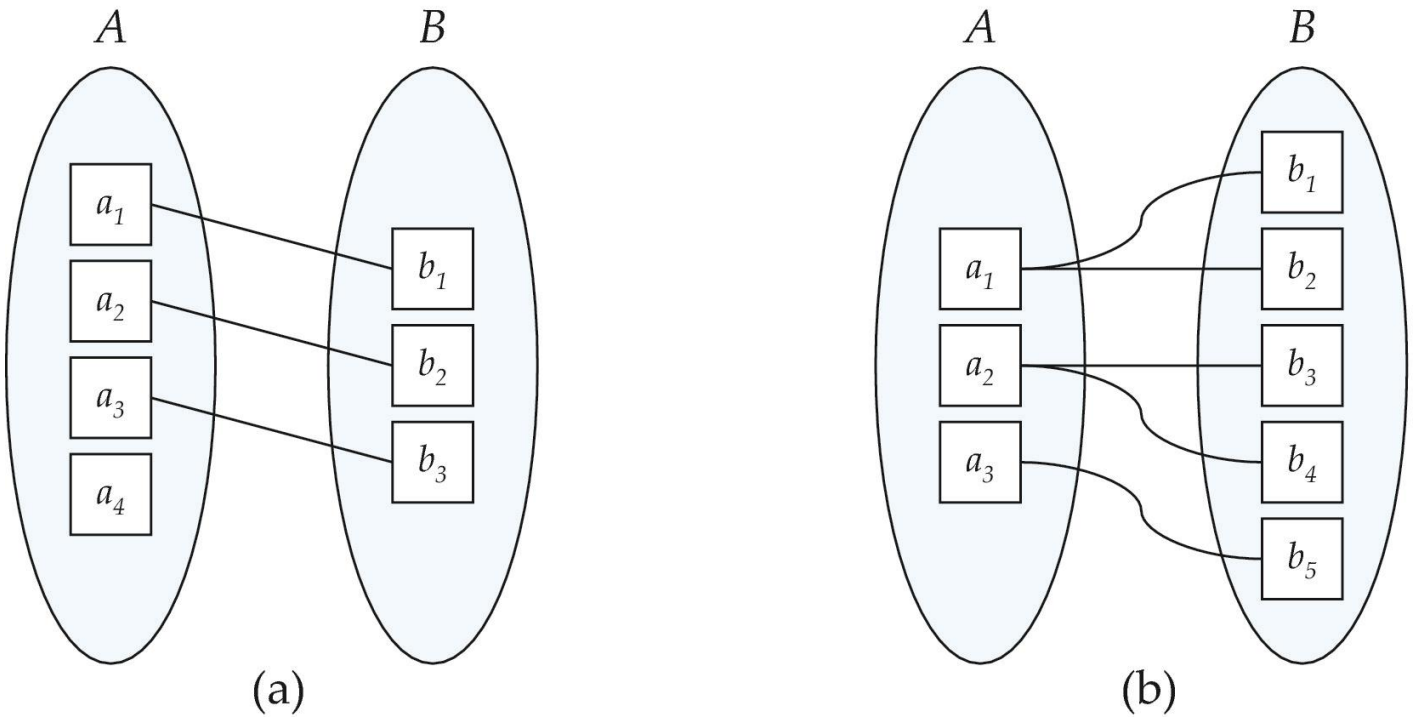
- ***Degree*: the number of participating entity sets in a relationship.**
  - Degree 2: *binary*

    involve two entity sets
  - Degree 3: *ternary*
  - Degree n: *n-ary*
- Binary relationships are very common and widely used.

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

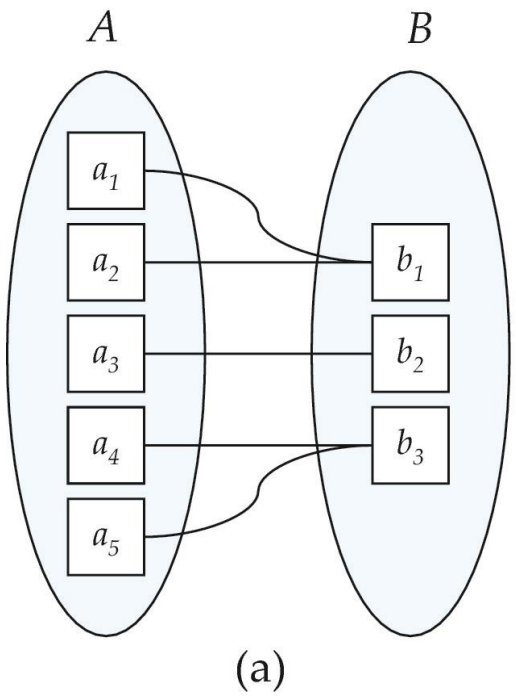


One to one

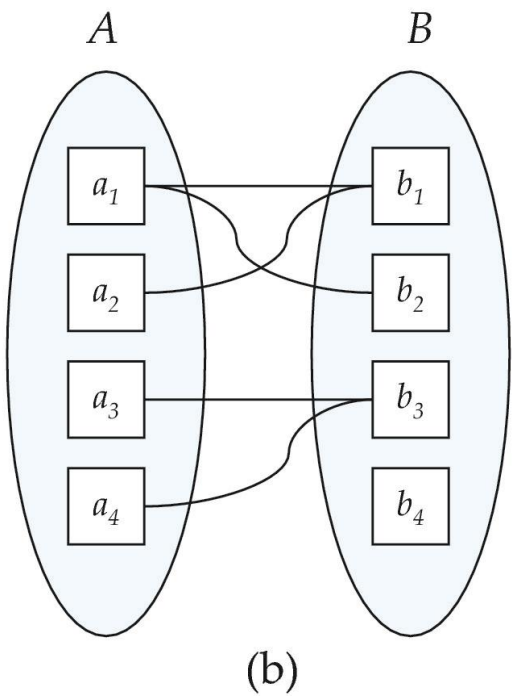
One to many

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

Mapping Cardinalities



Many to one



Many to many

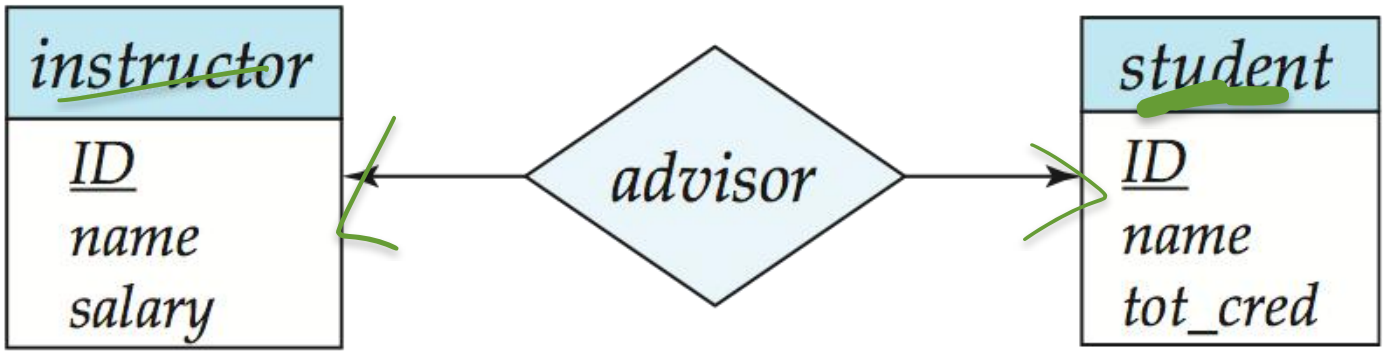
Note: Some elements in A and B may not be mapped to any elements in the other set

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.

One-to-One Relationship

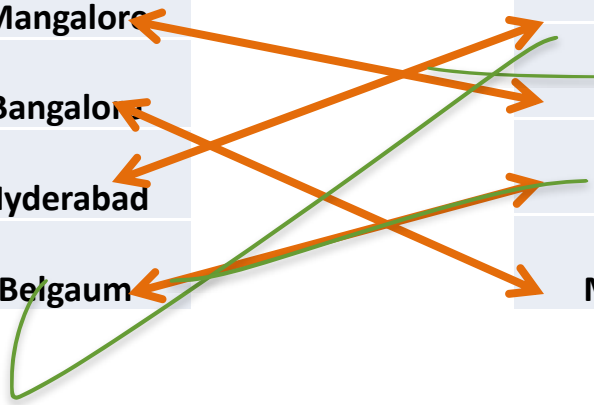
- one-to-one relationship between an *instructor* and a *student*
  - "an instructor is associated with at most one student via *advisor* "
  - and "a student is associated with at most one instructor via *advisor*"



Example one-one

Institute	
Institute Name	Location
ABC Tech	Mangalore
AAA Engg College	Bangalore
BBB Technology	Hyderabad
VTU	Belgaum

Head_Of_Institute		
Head_Name	Experience	Qualification
Rajesh	28	Mtech PhD
Ravi	30	MSc PhD
Blake	19	MS PhD
Mukhesk	25	Mtech PhD



A sample **one-to-one** relationship between an ***Institute*** and a ***Head_Of_Institute***

1 Institute will have only 1 Head of Institute – *Therefore exactly one Head_Of_Institute entity is associated with exactly one Institute Entity*

One-to-Many Relationship

- one-to-many relationship between an *instructor* and a *student*
  - **an instructor is associated with several (including 0) students via *advisor* relationship.**
  - a student is associated with **at most one** instructor via advisor,

Example one- Many

Customers

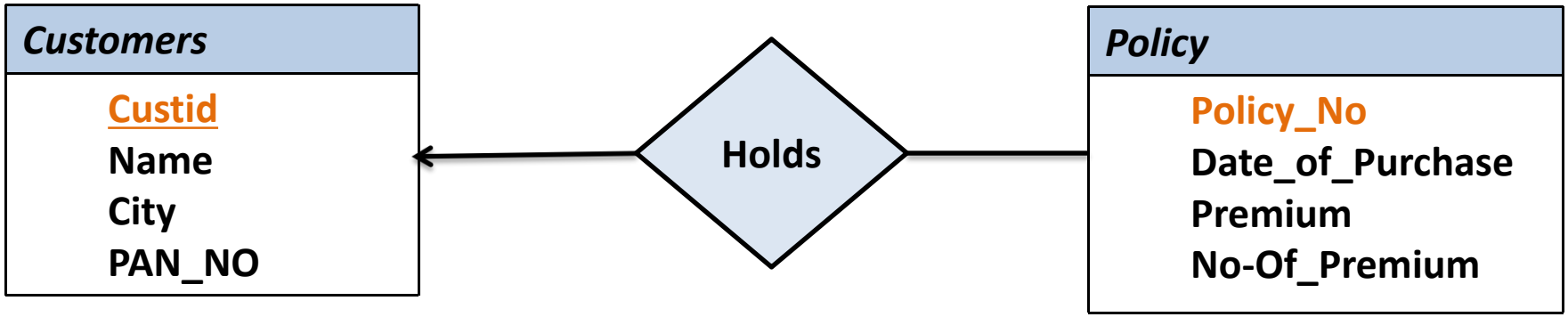
CustID	Name	City	PAN_NO
C101	Rajesh	Manipal	ABC1028
C102	Ravi	Mangalore	AFC1097
C103	Blake	Bangalore	BSD2013
C104	Mukhesk	Udupi	RES2001
C105	Ram	Udupi	FGU7629

Policy

Policy_No	Date_of_Purchase	Premium	No_of_Premium
LIC101	10-10-2001	4000	40
LIC102	1-1-1990	5000	30
LIC107	2-5-2012	7000	20
LIC102	2-9-2012	7000	25
LIC105	10-4-2001	8000	40
LIC202	2-3-2003	9000	30
LIC321	2-5-1998	3000	25
LIC312	4-4-2001	4000	28
LIC383	9-1-2004	5000	15

a sample one-to-many relationship between an ***Customers*** and a ***Policy***

1 Customer may take 1 or More LIC policies



Many-to-One Relationships

- In a many-to-one relationship between an ***instructor*** **and a *student,***
  - an instructor is associated with at most one student via *advisor*,
  - and a student is associated with several (including 0) instructors via *advisor*

Example Many-one

Employee

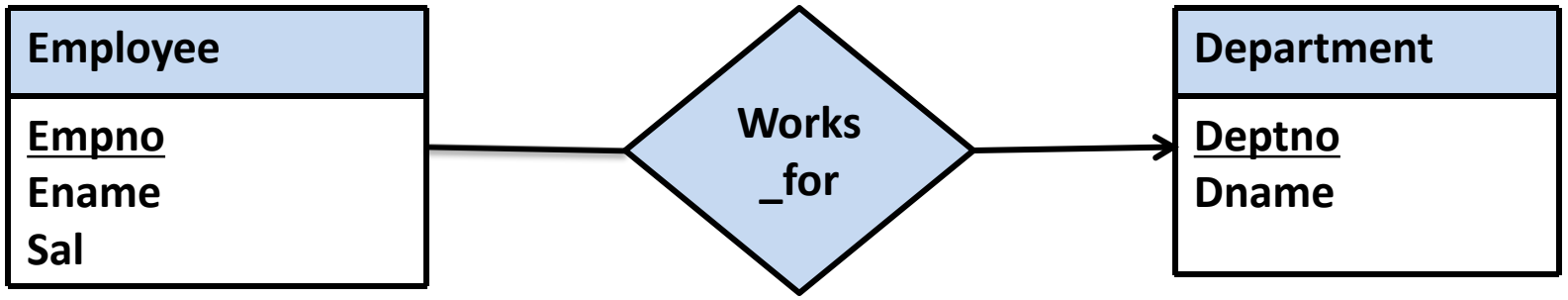
Empno	Ename	Sal
100	Ravi	2000
101	Raj	3000
102	Vikram	5000
103	Santhosh	4000
104	Ajay	7000
105	Anoop	8000
106	Sanoop	9000
107	Rakesh	7000

Department

Deptno	Dname
D1	MCA
D2	CS
D3	IT

A sample many-to-one relationship between an ***Employee*** and a ***Department***

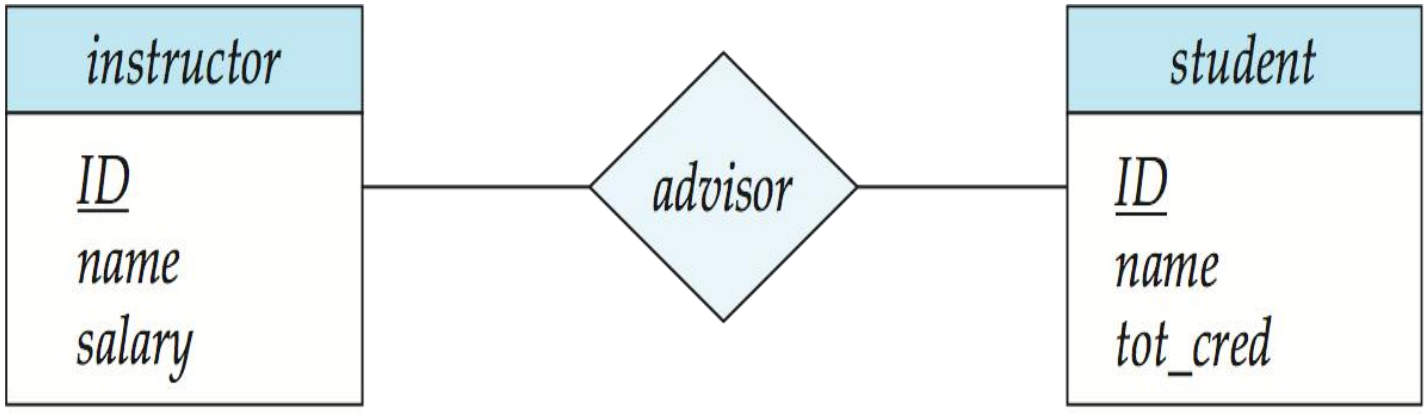
More than 1 employees work in 1 Department



One Department entity may be associated with one or more Employee entities

Many-to-Many Relationship

- An instructor is associated with several (possibly 0) students via *advisor*
- A student is associated with several (possibly 0) instructors via *advisor*



Example Many - Many

STUDENTS

RegNo	Name	Phone
MCA101	Rajesh	124478
MCA102	Ravi	344535
MCA103	Blake	473456
MCA104	Mukhesh	445987
MCA105	Ram	896949
MCA106	Vijay	800543

SUBJECTS

SubjectID	SubjectName	Credits
S101	OS	4
S102	OT	3
S103	INS	3
S104	ADBMS	4
S105	OOPD	4
S106	DS	4

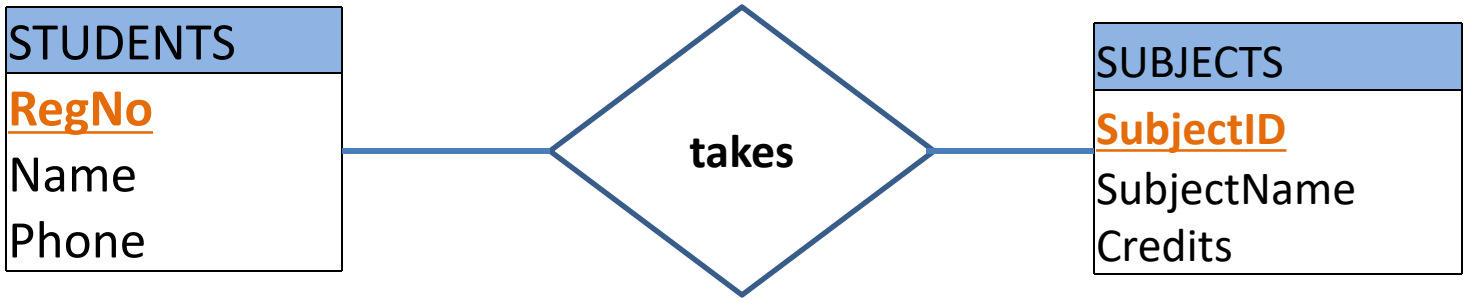
a many-to-many relationship between an ***STUDENTS*** and a ***SUBJECTS***

More than 1 Student can take 1 subject – therefore Many-1

More than 1 Subject cab be take by 1 students – therefore 1-Many

Equivalently

Multiple students can take Multiple Subjects. Therefore **Many-Many**



Example- Design ER Model

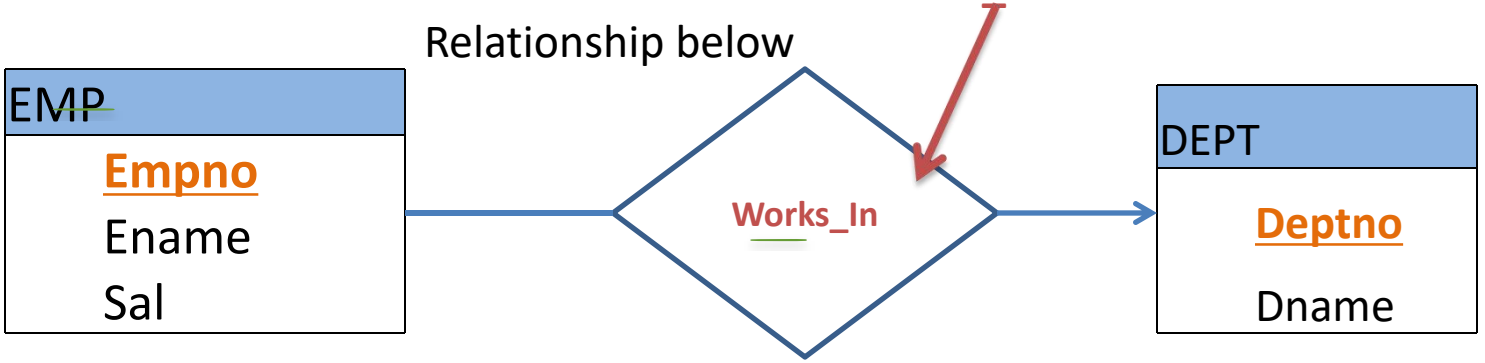
- Consider the following data requirement of an Institute. Assume that we want model the data requirements using ER modeling.
- The Institute is comprised of several departments such as –DSCA, CS, IT etc.. and each department has a Department Number such as D1,D2,.. used to identify each department. Each department has unique name. In every department many employees work. Each employee work in one particular department. Each employee is identified by a unique Employee Number. Other information about each employee we need is employee name and salary.

Entity Sets ,Sample Entities and relationship

Empno	Ename	Sal
100	Ravi	2000
101	Raj	3000
102	Vikram	5000
103	Santhosh	4000
104	Ajay	7000
105	Anoop	8000
106	Sanoop	9000
107	Rakesh	7000

Deptno	Dname
D1	MCA
D2	CS
D3	IT

A sample Employee and Department entities association is the Relationship set. and it is modelled as **Works_In** Relationship below



The relationship is having Many-1 Cardinality . An employee can work at most in only 1 department.

Note that If you are adding **Deptno** as another column in EMP to represent Employee working in a department association, it becomes redundant information as **Works_In** implicitly represents it. (Same is discussed under heading –Redundant Attributes)

Example- Design ER Model…

Assume that we want model following data requirements of Institute using ER modeling.

The college is comprised of several departments such as –DSCA,CS,IT etc.. and each department has a Department Number such as D1,D2,.. used to identify each department. In every department many employees work. Each employee is identified by a unique Employee Number. Information about each employee we need is employee name and salary.

Extend previous ER diagram with following Data Requirements. Assume appropriate attributes.

We also want to record information related to the courses offered by each department such as courses names, Number of semesters, Total Credits etc.

For example– IT department offers – Inform. & Tech., Computer & Comm. ; DSCA offers DSE,MCA ;Computer Science offers CS, Info.Security.

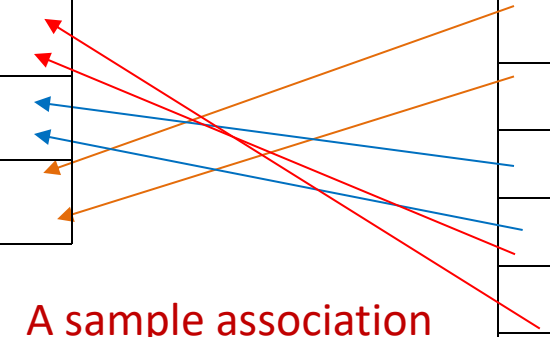
Entity Sets ,Sample Entities and relationship

Department Entity Set

Deptno	Dname
D1	DSCA
D2	CS
D3	IT

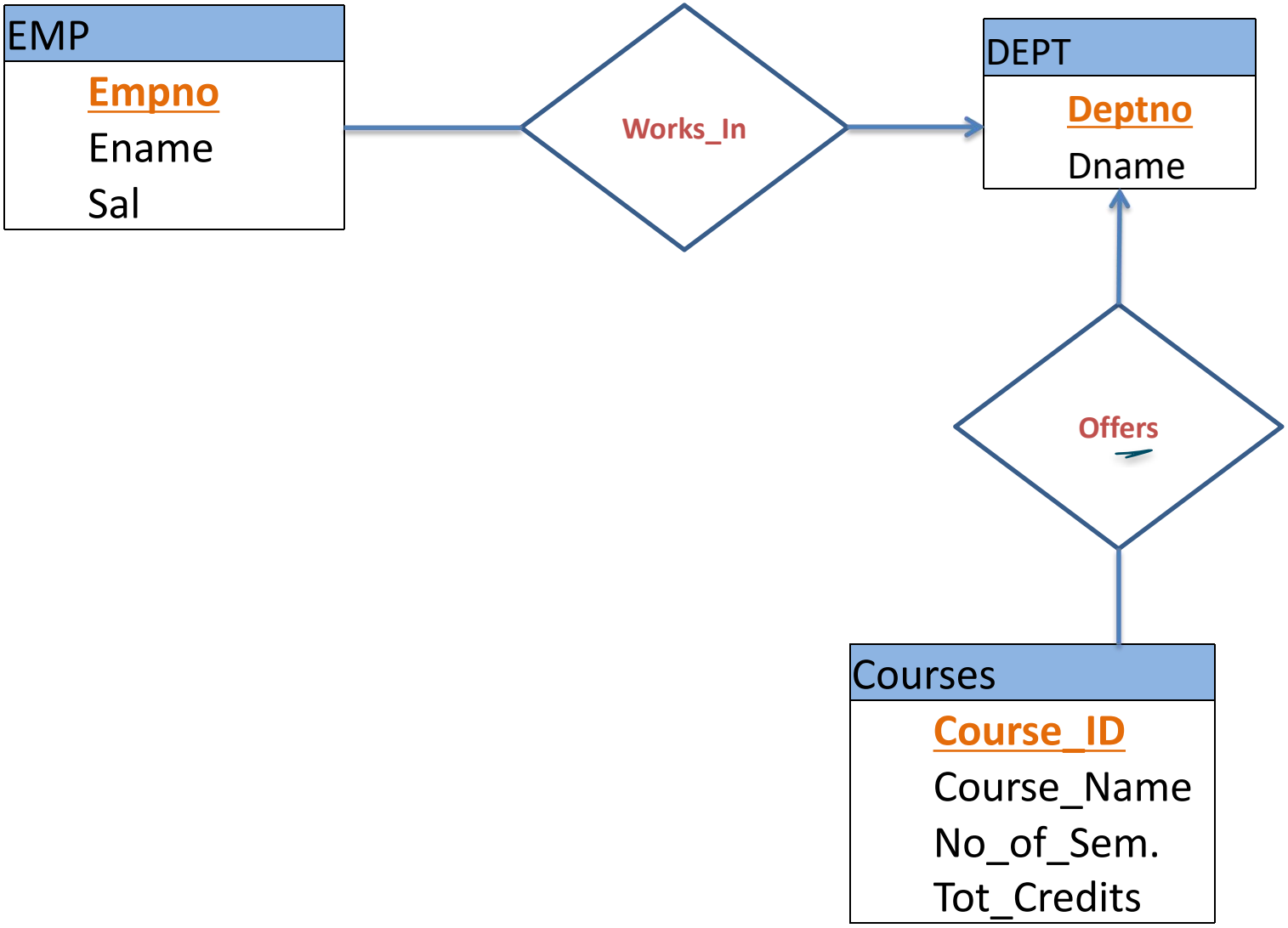
Course Entity Set

Course_ID	Course_name	No-Of Sem	Credits
100	Info & Tech		
101	Comp&Com		
102	Info.Security		
103	Comp Science		
104	DSE		
106	MCA		



A sample association between Department entities and Course entities

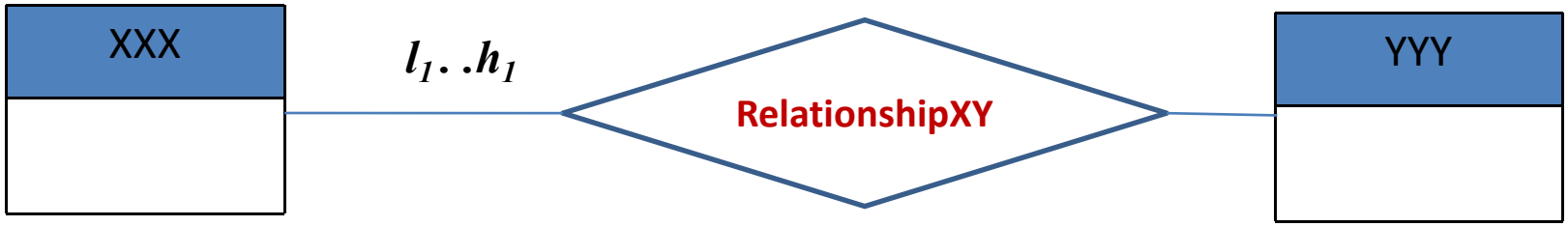
ER Diagram for the requirements given in slide 34 and 36



Notation for Expressing More Complex Constraints

- Along with Cardinality constraints such as **1-1**, **1-M** & **M-M**, Cardinality limits can also be used express participation constraints.
- Cardinality limits tells about-
  - The number of times (Minimum & Maximum) each entity participates in relationships in a relationship set.
  - A line may have an associated minimum and maximum cardinality, shown in the form ***l..h***
    - where ***l*** is the minimum and ***h*** the maximum cardinality

A **XXX** entity may be associated with a minimum of l_1 *number of YYY entities* or at the maximum h_1 number of **YYY entities through RelationshipXY relationship.**



Notation for Expressing More Complex Constraints

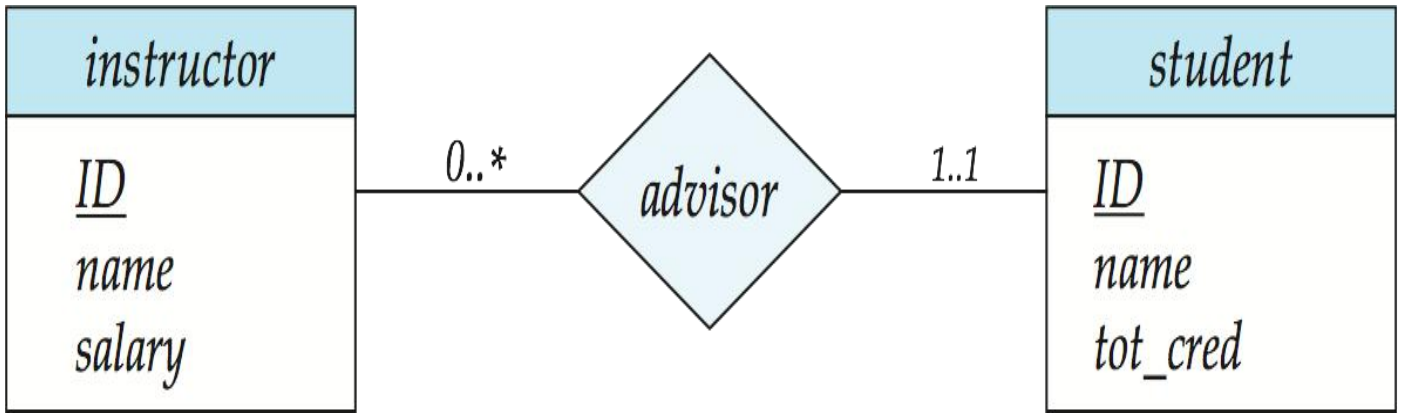
Example:

- The line between advisor and student has a cardinality constraint of **1..1**, meaning the **minimum** and the **maximum** cardinality are both **1**.

  A student entity is associated with max & min 1 instructor entity only.
  Means each student must have exactly one advisor.

- The limit **0..*** on the line between *Instructor* and *Advisor* indicates that an instructor may be associated with **zero or more students**.

  Means instructor can be advisor for **zero or more** students.

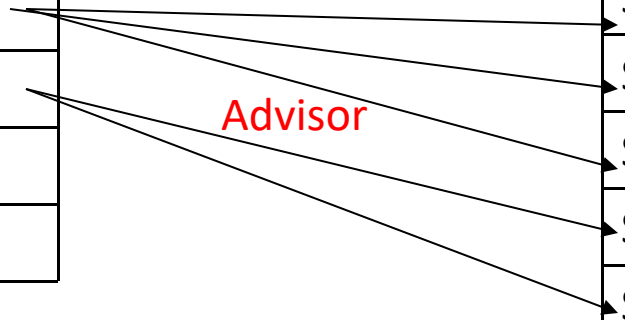


Instructor

ID	Name	Salary
I1		
I2		
I3		
I4		

Student

ID	Name	Total_Credit
S1		
S2		
S3		
S4		
S5		



An Instructor may or may not be an Advisor.

Hence **Minimum** number of Instructor entity that can be associated with student entities through advisor is **0.**

An Instructor may be Advisor to any number of Students.

Hence **Maximum** number of Instructor entity that can be associated with student entities through advisor is ***.**

0..*

Minimum-0 means every Instructor need not participate in Advisor.

i.e. Partial participation

Each student must have exactly one advisor.

Every student entity is Associated with an Instructor via Advisor Relationship.

Maximum=1 & Minimum=1 Student entity that can participate in Advisor is **1..1**

Minimum-1 means every Student must participate in Advisor.

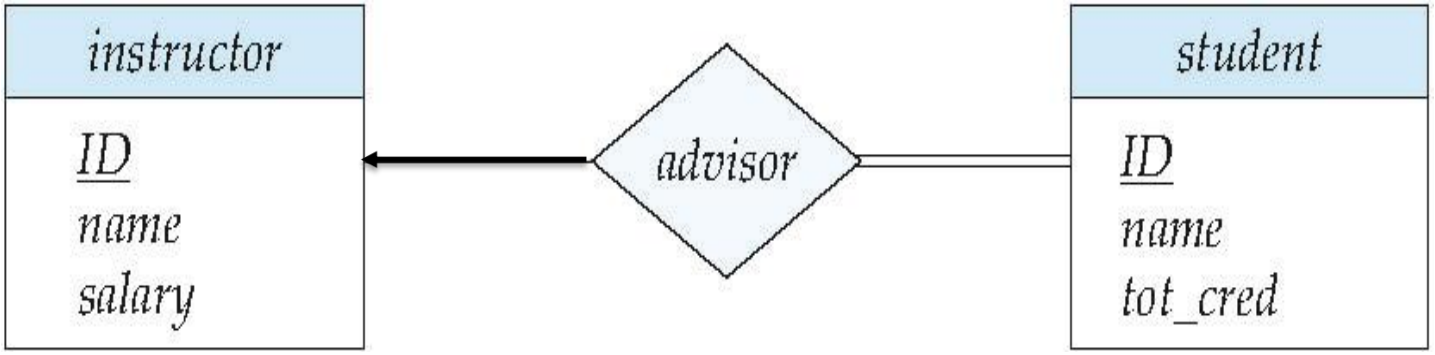
Total participation

Participation of an Entity Set in a Relationship Set

- **Total participation** (indicated by double line):

  every entity in the entity set participates in at least one relationship in the relationship set

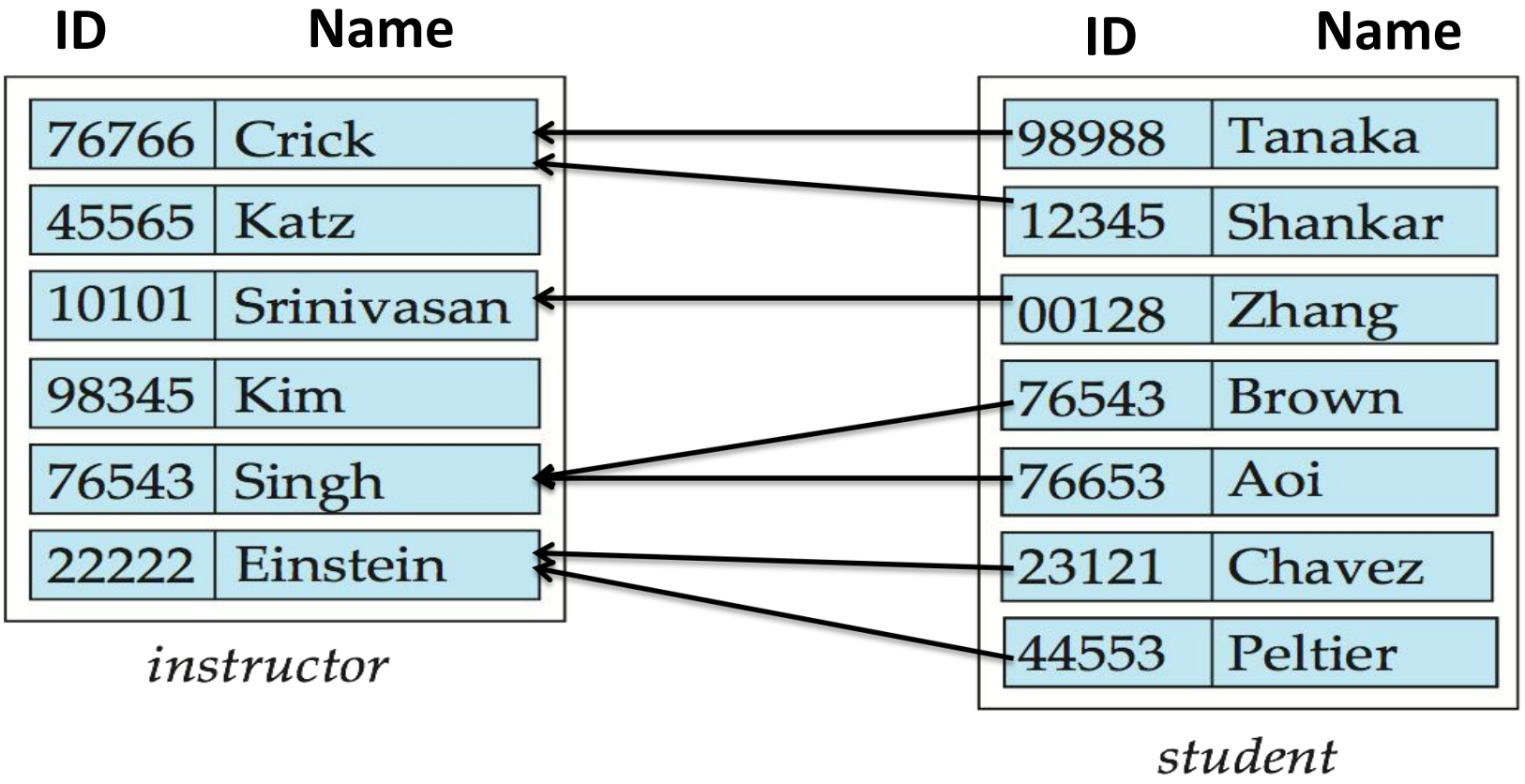
  - E.g., participation of *Student* in *advisor* is **total**
    - every *Student* must be associated with *instructor*



Partial participation:

some entities may not participate in any relationship in the relationship set

- Example: participation of ***instructor*** **in** ***advisor relationship*** **is partial** (see next slide)



For an Instructor, being an Advisor is optional –

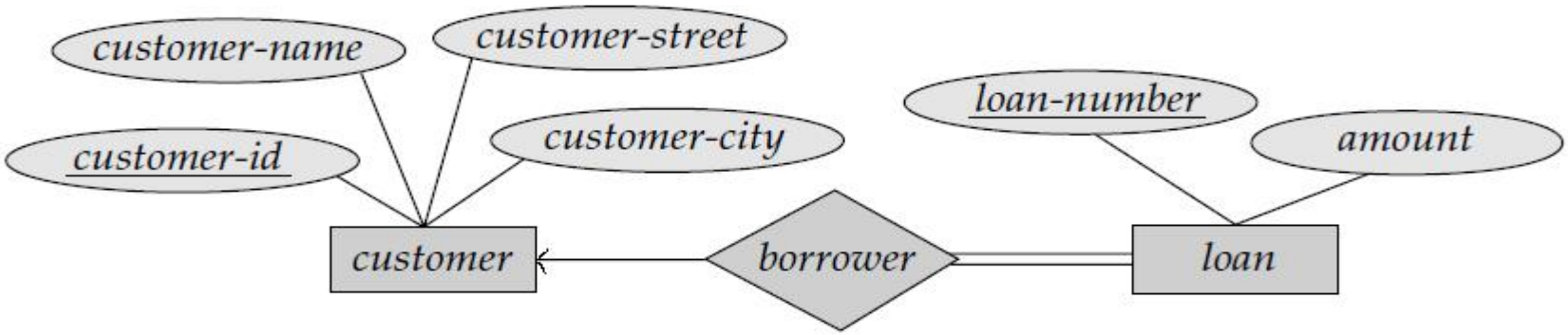
Therefore **Instructor** entity participation is **PARTIAL**)

Note: (45565,Katz) and (98345,Kim) are not participating in advisor relationship

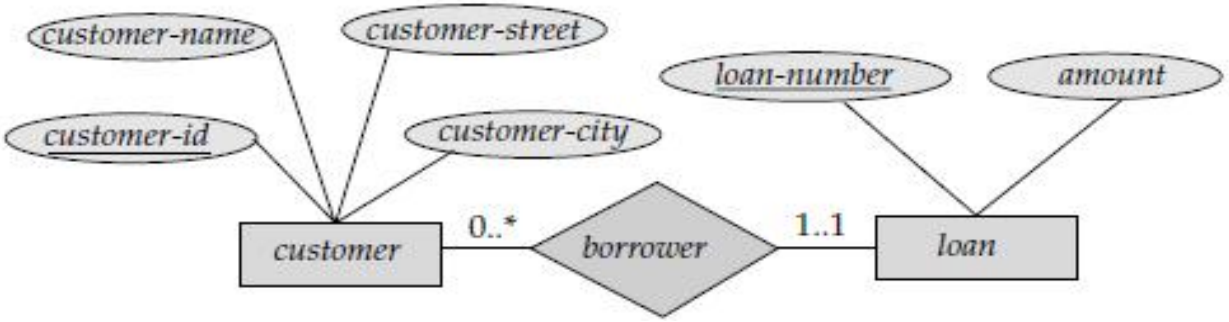
but Every student must have an Advisor –

Therefore **Student** entity participation is **TOTAL.**

Total and Partial participation …

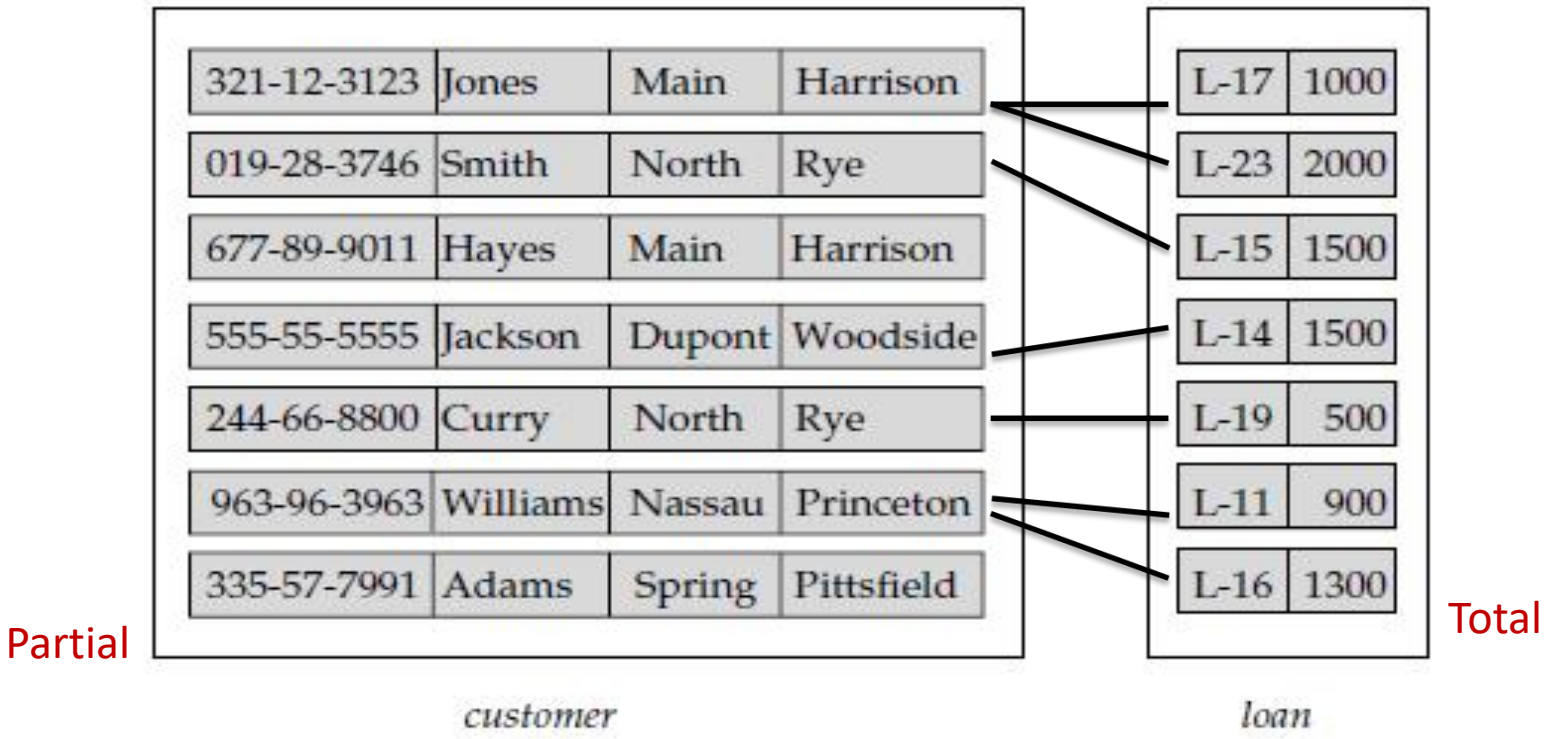


- participation of *loan* in *borrower* is total
  - In loan (entity type), every loan entity has to be associated with 1 or more customers, because a loan can't be given if there do not exist customer taking loan.
- participation of *customer* in *borrower* is Partial
  - In Customer, there may be some customers not taking loan and so do not participate in borrower relationship



Same ER diagram with Cardinality Limit for Participation

Total and Partial participation …



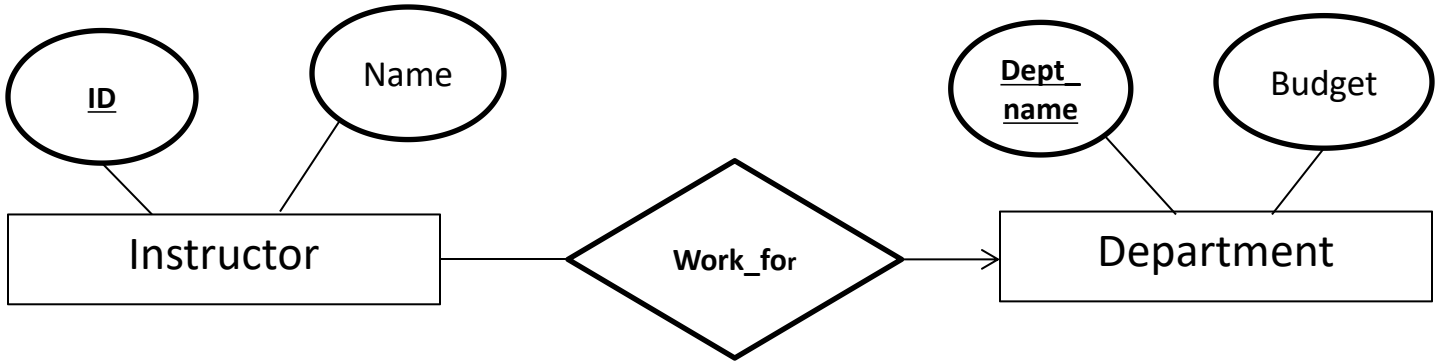
Note that Customers- Hayes & Curry haven't taken any loan , therefore Customer participation is Partial in Borrower relationship.

But, a Loan has to be given to a customer only , so a Loan entity can not exist without being associated with Customer . So Loan participation is Total in Borrower relationship

Redundant Attributes

- Suppose we have entity sets
  - *instructor*, with attributes including *dept_name*
  - *Department* and a relationship
  - *inst_dept* relating *instructor* and *department*

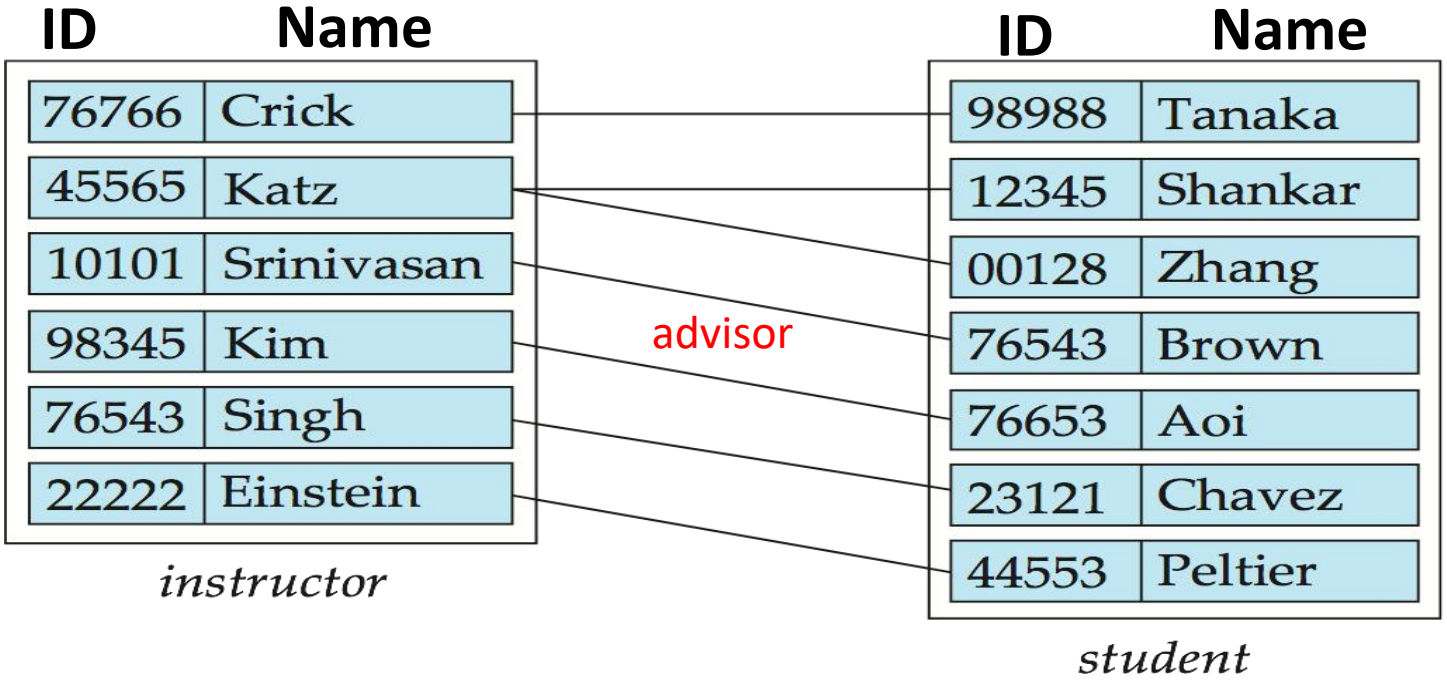
- If we put Attribute **dept_name** in entity instructor then information (employee works in a department) is redundant since there is an **explicit relationship inst_dept** which relates instructors to departments
  - The **dept_name** a*ttribute replicates information present in the relationship,* and **dept_name** should be removed from *instructor*
  - BUT: when converting ER back to tables, in some cases the attribute gets reintroduced.



Keys

- A **super key** of an entity set is a set of one or more attributes, allow us to identify uniquely every entity in the entity set.
- A **candidate key** of an entity set is a minimal super key.
  - *ID* is candidate key of *instructor*
  - *course_id* is candidate key of *course*
- Although several candidate keys may exist, one of the candidate keys is selected to be the **primary key**.
- Need to consider semantics of Entity set in selecting the ***primary key*** in case of more than one candidate key.
  - Concept is Discussed in Relational Model Chapter

Keys for Relationship Sets



Relationship set advisor ={ (76766,Crick - 98988,Tanaka) ,(45565,Katz- 12345,Shankar), (45565,Katz-00128,Zhang),(10101,Srinivasan),… (22222, Einstein-44553,Peltier) }

Every element(Relationship instance) in Relationship Set is to be identified uniquely, therefore we need a primary key for relationship set too.

Keys for Relationship Sets

- The combination of primary keys of the participating entity sets forms the Primary key of a relationship set.

- Let R be a relationship set involving entity sets $\mathbf{E_1, E_2, \ldots, E_n}$. Let **primary-key($\mathbf{E_i}$)** denote the set of attributes that forms the primary key for entity set E_i.

- If the relationship set R has no descriptive attributes associated with it, then the set of attributes.

  primary-key(E_1) U primary-key(E_2) U…primary-key(E_n)

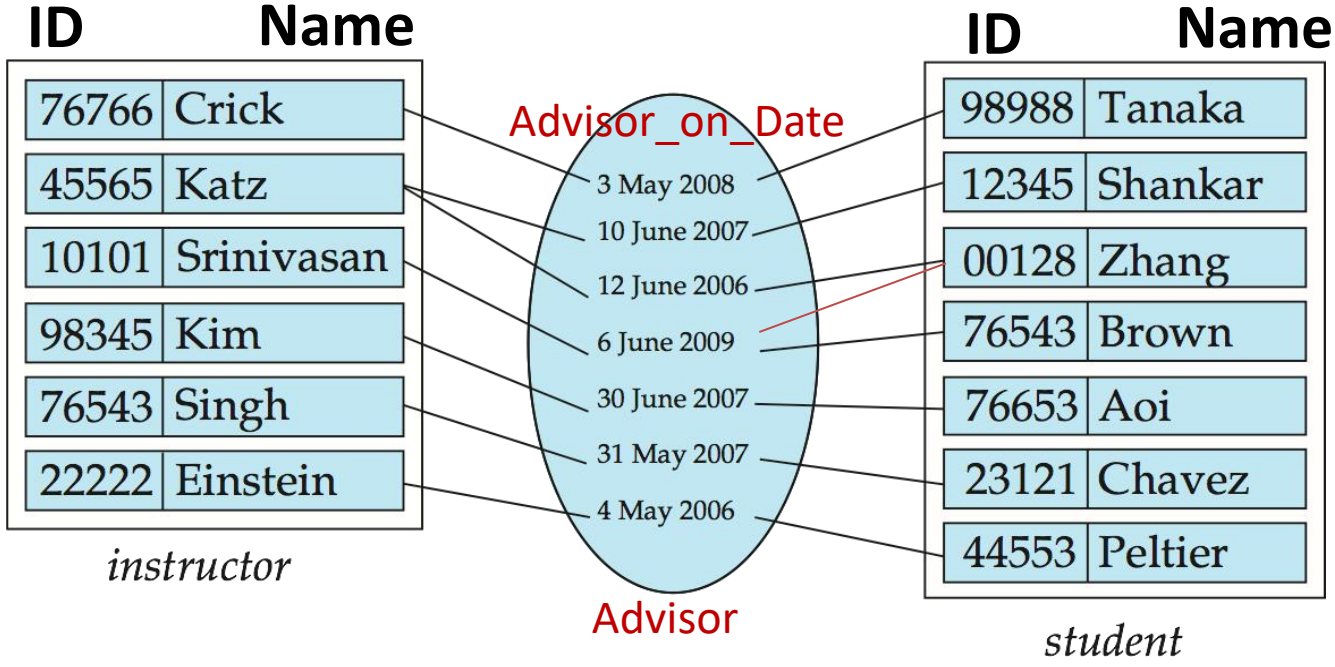
- Example:
  - Primary Key for Advisor relationship set is
    - – **Primary Key(Instructor) U Primary Key(Student)**
- i.e (ID,ID) , i.e. ID of Instructor & ID of Student is Pkey of Advisor

Keys for Relationship Sets having Attributes

- If the relationship set R has **descriptive attributes** $a_1, a_2, \ldots\ldots, a_m$ associated 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_m} forms the Primary Key for relationship set R.

Example



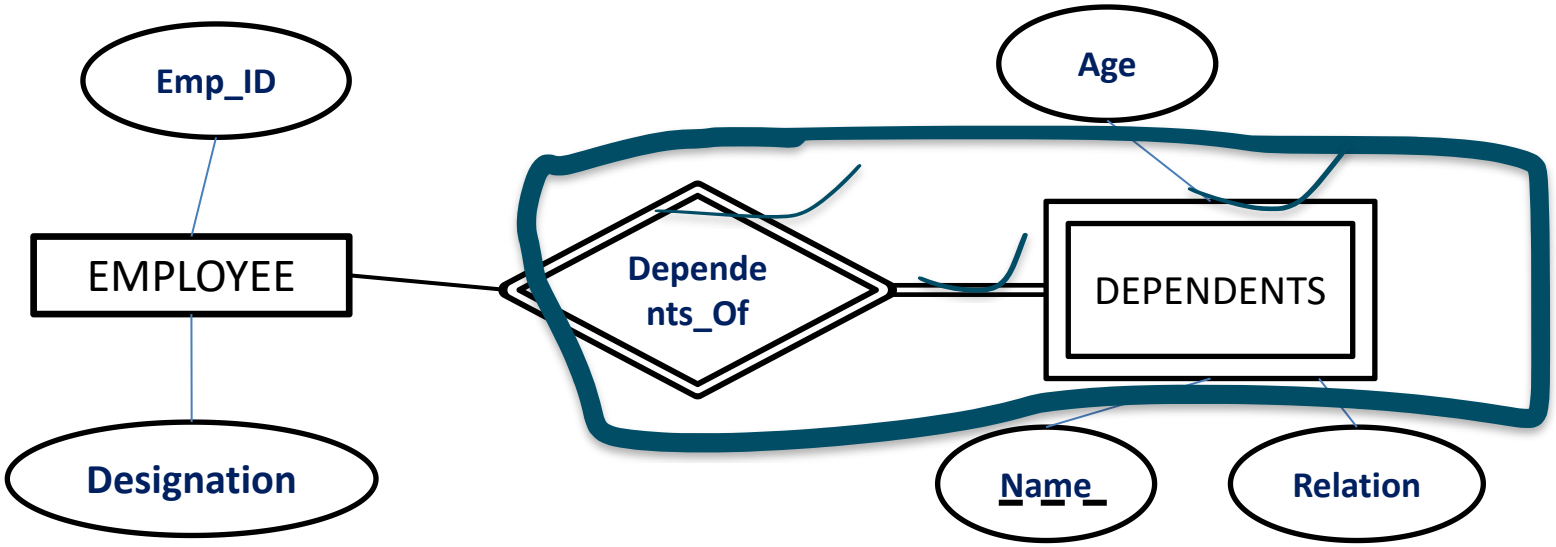
Pkey of Advisor relationship set is – **P.key (Instructor) U P.key (Student) U {Advisor_On_Date}**

(ID, ID, Advisor_On_Date)

Weak Entity Sets

- 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**
  - It must relate to the identifying entity set **via a total partition, one-to-many relationship set** from the identifying to the weak entity set
  - **Identifying relationship** depicted using a double diamond
- 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 Sets (Cont.)

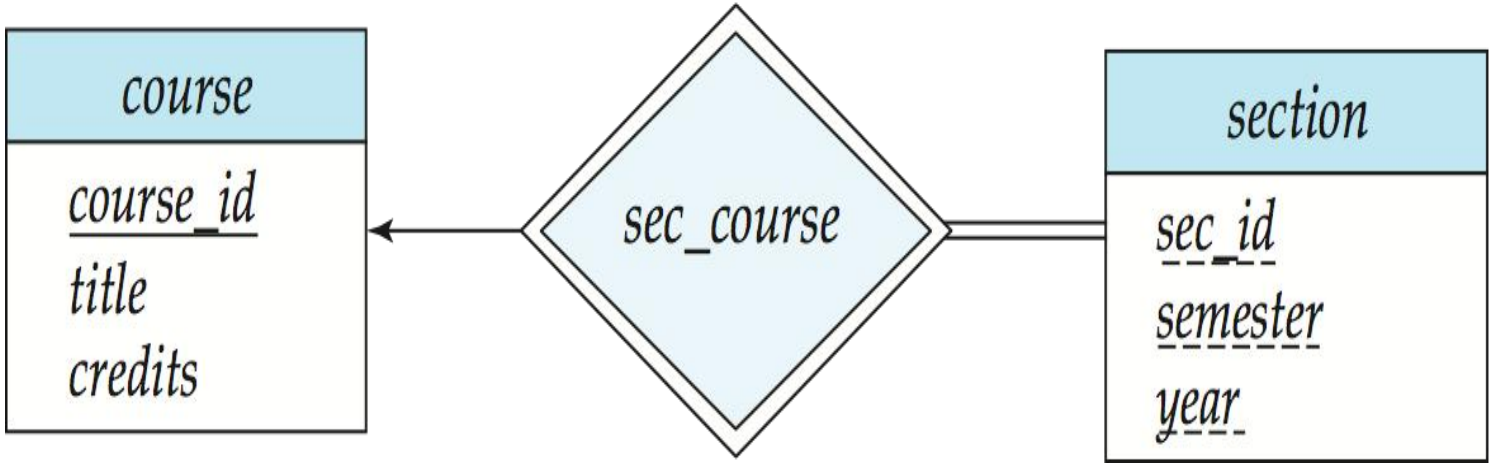


Empno	Designation
100	
101	
102	

Name	Relation	Age
X	Son	12
Y	Daughter	5
Y	Wife	25
P	Son	7

Weak Entity Sets (Cont.)

- We underline the discriminator of a weak entity set with a dashed line.
- We put the identifying relationship of a weak entity in a double diamond.
- Primary key for *section* – (***course_id, sec_id, semester, year***)



Partial key/ Discriminator: Uniquely identifies a section among the set of sections of a particular course

Weak Entity Sets (Cont.)

Course

Course_Id	Title	Credits
MCA	Master of..	36
MTech	Info. Security	38
MTech	S/W Engg	36

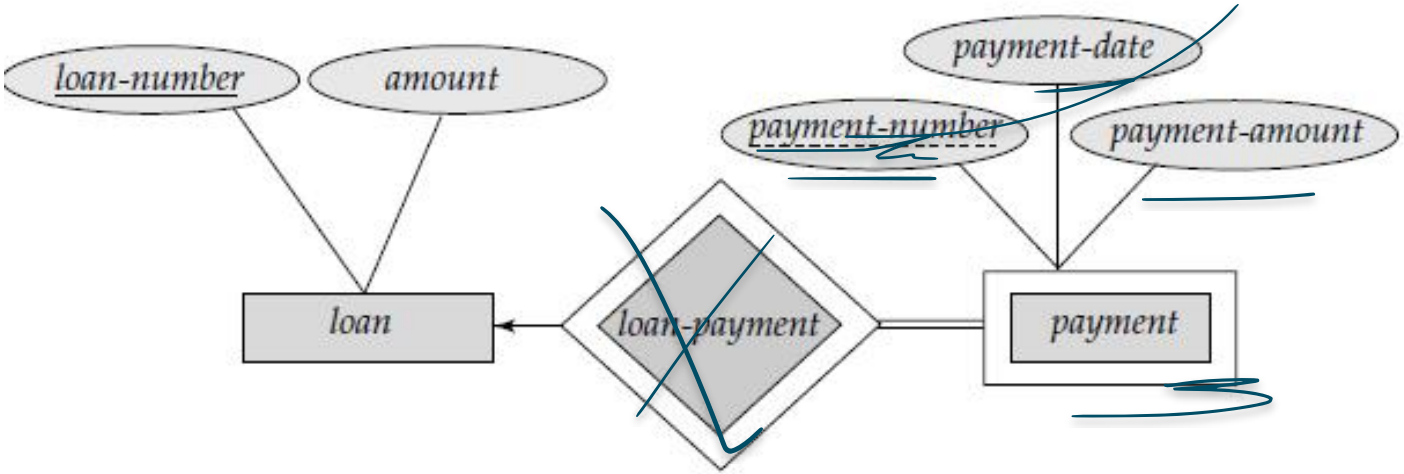
Sec_Course

Section

Sec_Id	Semester	Year
A	I	2021
B	I	2021
A	III	2020
B	III	2020
A	II	2020
B	II	2020
A	IV	2019
B	IV	2019
A	I	2019
A	III	2018
…	..	..

Weak Entity Sets (Cont.)

- **Note**: the primary key of the strong entity set is not explicitly stored with the weak entity set, since it is **implicit** in the **identifying relationship**.
- If *course_id* were explicitly stored, *section* could be made a strong entity, but then the relationship between *section* and *course* would be duplicated by an implicit relationship defined by the attribute *course_id* common to *course* and *section.*
- Similar idea applies to loan, payment relations given below(how it is represented in scheme ?see next slide)-



loan-number	amount
L-11	900
L-14	1500
L-15	1500
L-16	1300
L-17	1000
L-23	2000
L-93	500

Loan Entity Type

Loan-Payment

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

Payment Entity Type – Weak Entity

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

Payment Entity Type can be Identified with respect Loan using PK of Loan

Information to be fetched from Requirements

- Entities
  - Strong
    - Attributes-Primary key, Composite/ Simple, Multivalued/Single valued, Derived.
  - Weak
    - Partial key
- Relationship Between Entities
  - Relationship Attributes
  - Name of relationship
  - Cardinality Constraint
    - **1-1**, **1-M**, **M-M**
    - Cardinality Limits
  - Participation
    - Total/Partial

Example

An Institute want to keep track of information about Funded Projects, Agencies which are Funding them and Faculties who work on those Projects.

The college is comprised of several departments such as – MCA,CS,IT ,MECH, EEE etc.. and each department has a Department Number such as D1,D2,.. used to identify each department. Many faculties work in every department . Each faculty is identified by a unique Employee Number. Information about each faculty we need is employee name, Qualification, Research-domain. Each Departments may have many funded research projects. Information about these projects such as an unique Project-ID, Title, Fund-Received, Duration. Each of these projects are funded by one or more funding agencies such as – MHRD,DSR, DST,BARC etc.. We also need to record information about Funded Funding agencies funding the projects such as- Grant_ord_No, Agency-Name, Contact-Person, Email, Phone, Total_Grant,Year-of-Grant. An agency may fund multiple project and a project may also receive grant from multiple agencies.

Model above data requirements using ER modeling.

What are Entities here ?

What are attributes & sample entities here.

Department

Deptno, Dname

D1	CS
D2	IT

Faculty

Empno, Ename, Qualification, Research-Domain

101	Raj	MTech	Data-Mining
102	Vinay	Mtech,PhD	Network Eng
106	Manu	MCA, PhD	AI

Projects

Project-ID, Title, Fund-Received, Duration

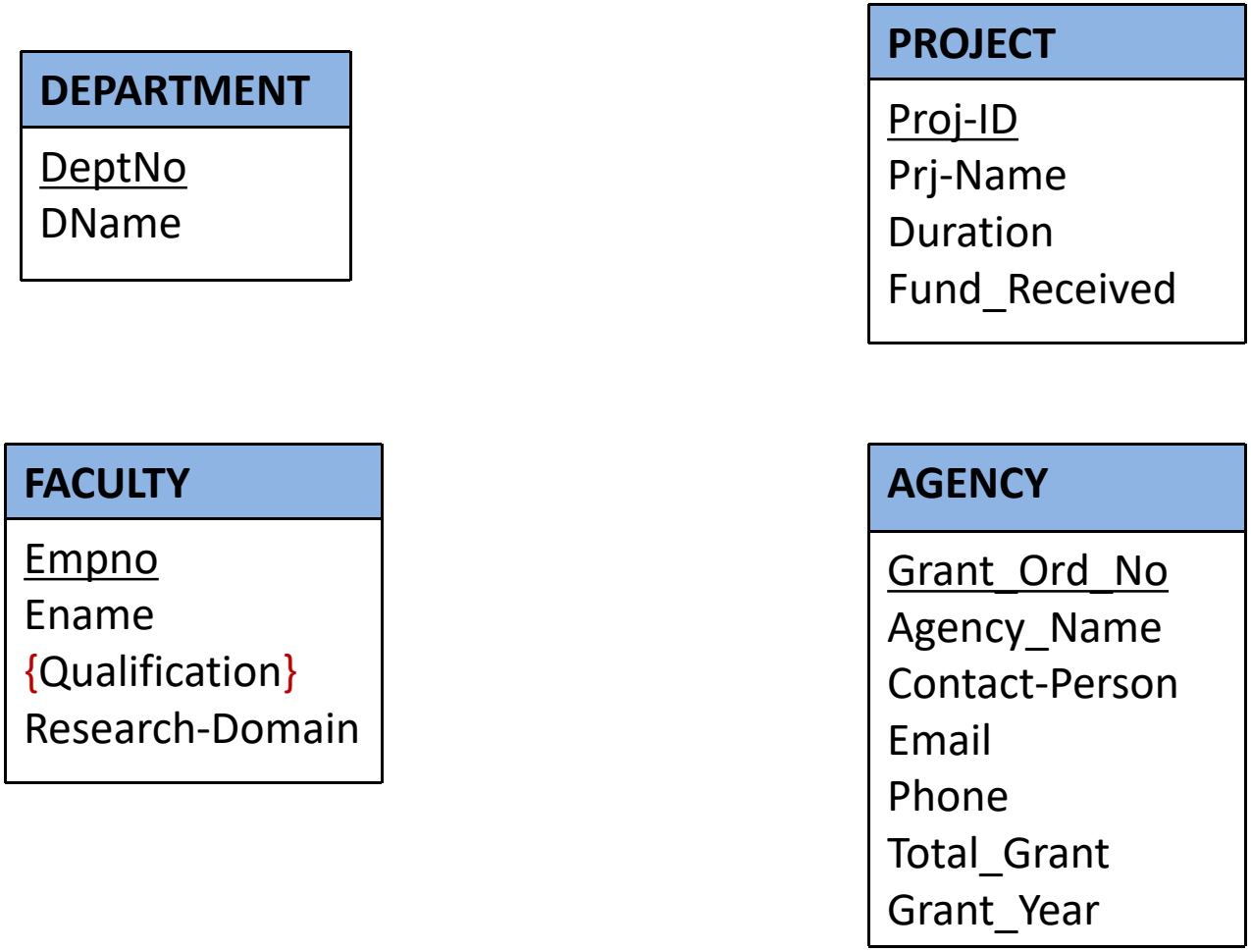
P1	XYXX	200K	2
P2	GHKL	500K	3
P3	FGHK		

Agency

Grant_order_No,Agency-Name, Contact-Person, Email,Phone,Total_Grant,Year-of-Grant

MH17-1	MHRD	RamRao	ram@gmail.com	78998667	2000K	2017
MH18-5	MHRD	Vijay	Vij@ymail.com	89532689	5000K	2018
DST17-3	DST	Ravi	Ravi@gmail.com	99644775	3500K	2017

Entities


DEPARTMENT
DeptNo
DName
PROJECT
Proj-ID
Prj-Name
Duration
Fund_Received
FACULTY
Empno
Ename
{Qualification}
Research-Domain
AGENCY
Grant_Ord_No
Agency_Name
Contact-Person
Email
Phone
Total_Grant
Grant_Year

Entities and Relationship

Deptno,	Dname
D1	CS
D2	IT
D3	EEE

1-Many

Empno	Ename	Qualification	Research-Domain
101	Raj	MTech	Data-Mining
102	Vinay	MTech,PhD	Network Eng
106	Manu	MCA, PhD	AI
108	Rakesh	MCA,MTech	SDN

1-Many

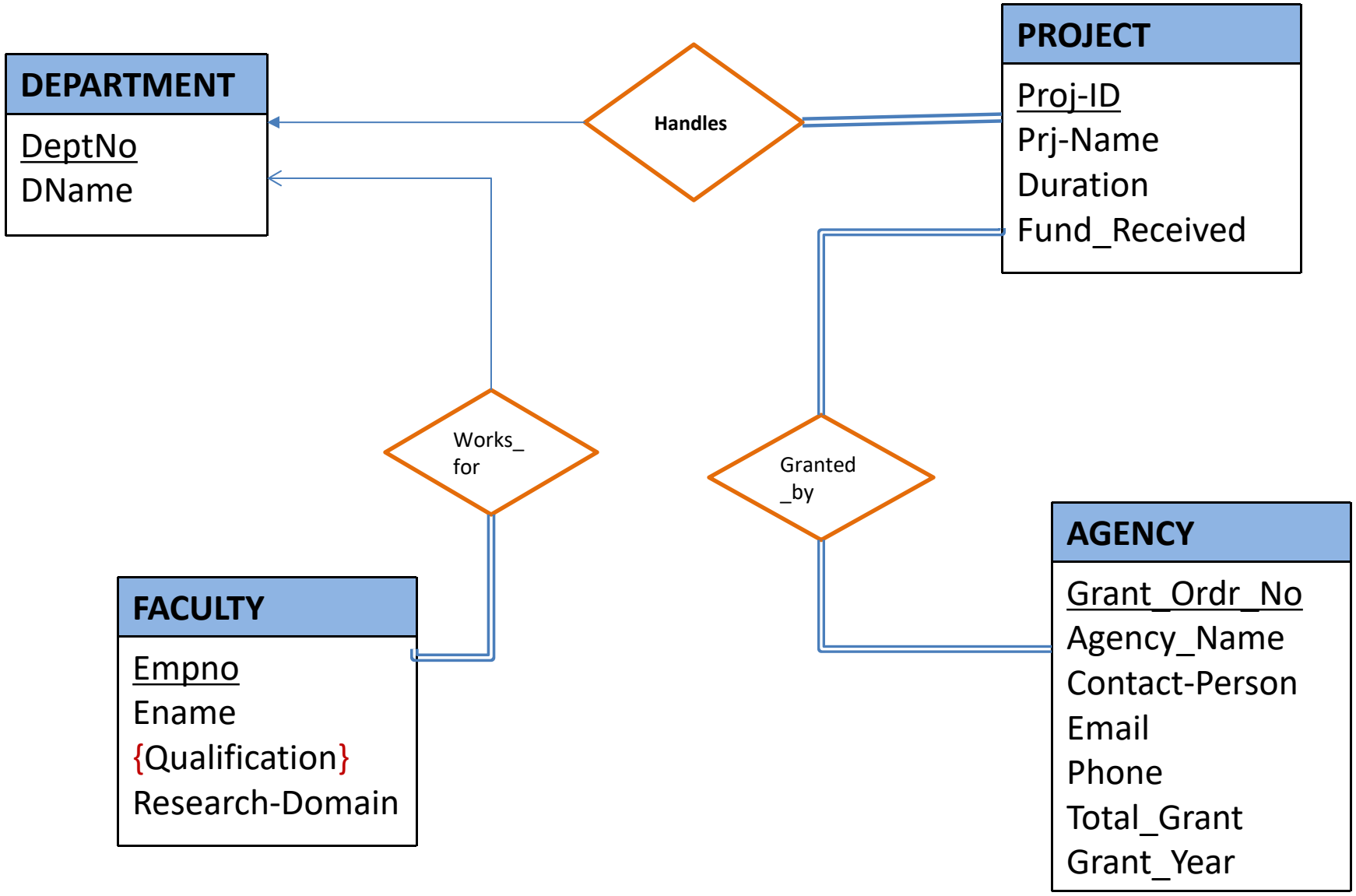
Project-ID	Title	Fund-Received	Duration
P1	XYXX	200K	2
P2	GHKL	500K	3
P3	FGHK	180K	1.5

Many-Many

Grant_order_No,Agency-Name, Contact-Person, Email,Phone,Total_Grant,Year-of-Grant

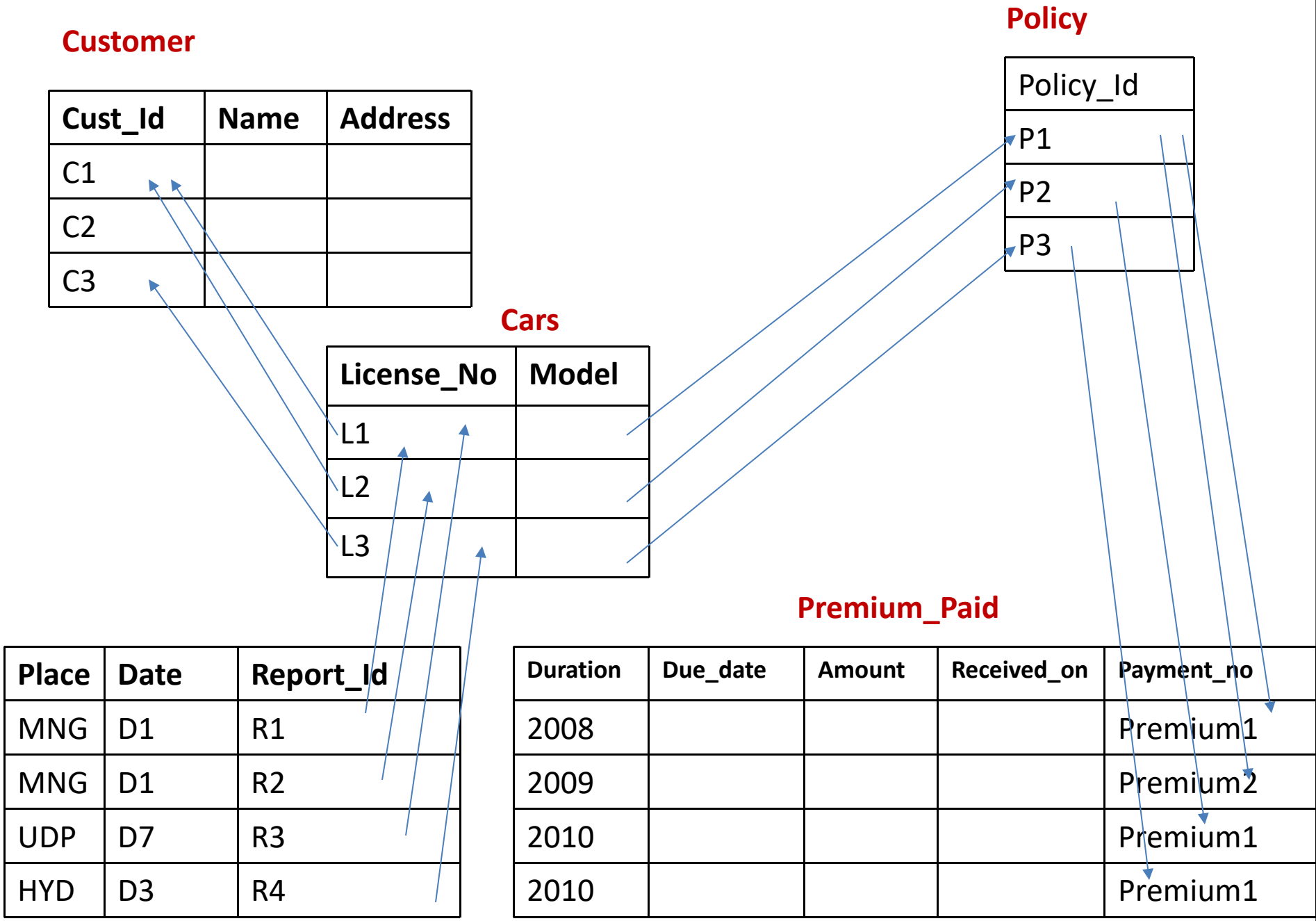
MH17-1	MHRD	RamRao	ram@gmail.com	78998667	2000K	2017
MH18-5	MHRD	Vijay	Vij@ymail.com	89532689	5000K	2018
DST17-3	DST	Ravi	Ravi@gmail.com	99644775	3500K	2017

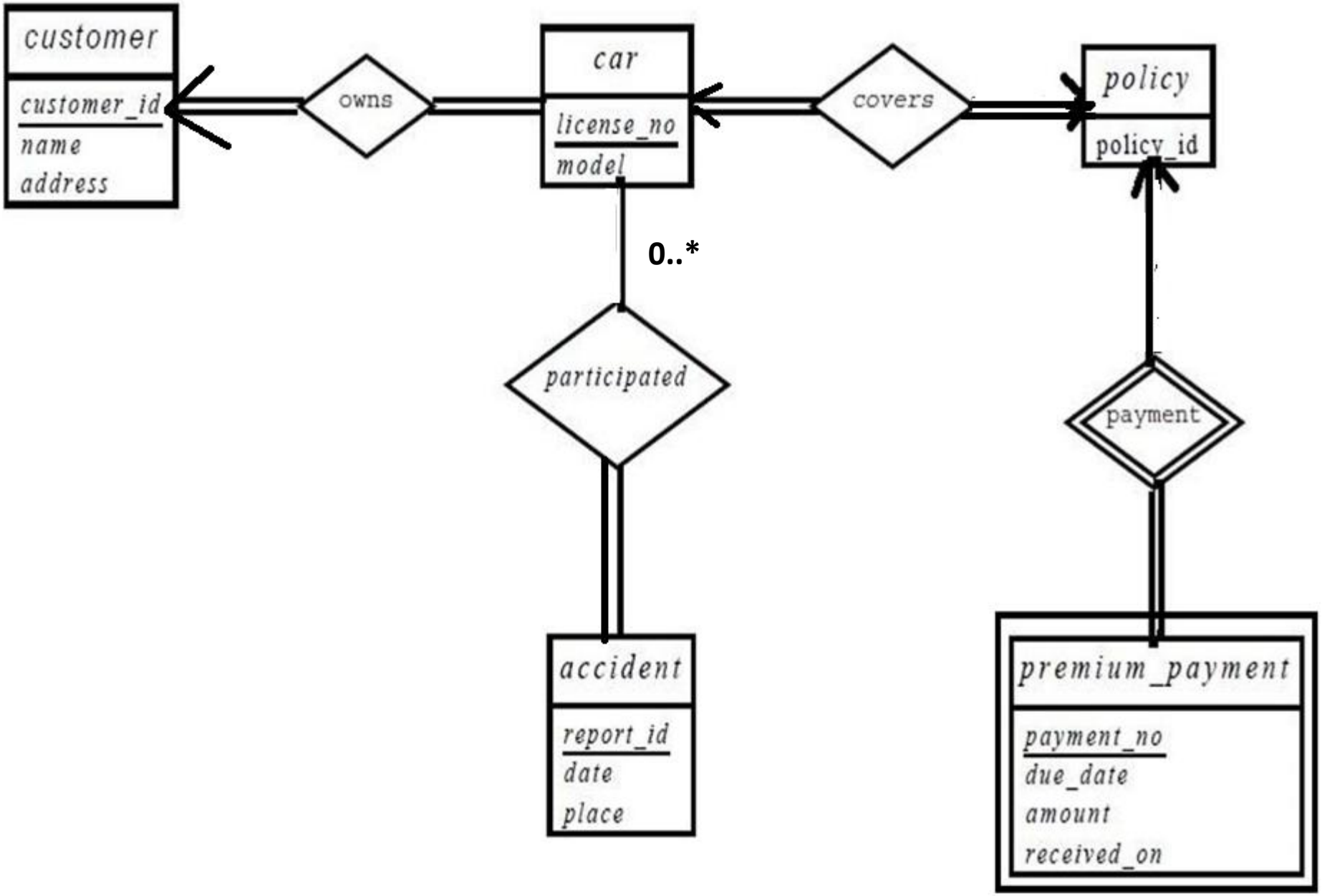
Entities & Relationship



Example

Construct an E-R diagram for a car insurance company whose customers own one or more cars each. Each car has associated with it zero to any number of recorded accidents. Each insurance policy covers one car. (Identify the change needed in ER diagram if requirement is - Each insurance policy covers one or more cars), and has one or more premium payments associated with it. Each payment is for a particular period of time, and has an associated due date, and the date when the payment was received.


Customer
Cust_Id
Name
Address
C1
C2
C3
Policy
Policy_Id
P1
P2
P3
Cars
License_No
Model
L1
L2
L3
Premium_Paid
Duration
Due_date
Amount
Received_on
Payment_no
2008
2009
2010
2010
Premium1
Premium2
Premium1
Premium1
Place
Date
Report_Id
MNG
D1
R1
MNG
D1
R2
UDP
D7
R3
HYD
D3
R4


customer
customer_id
name
address
owns
car
license_no
model
covers
policy
policy_id
0..*
participated
payment
accident
report_id
date
place
premium_payment
payment_no
due_date
amount
received_on

Reduction to Relational Schemas

Reduction to Relation Schemas

- **Entity sets** and **relationship sets** can be expressed uniformly as ***relation schemas*** that represent the contents of the database.

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

- For each entity set and relationship set there is a **unique schema that is assigned** the name of the corresponding entity set or relationship set.

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

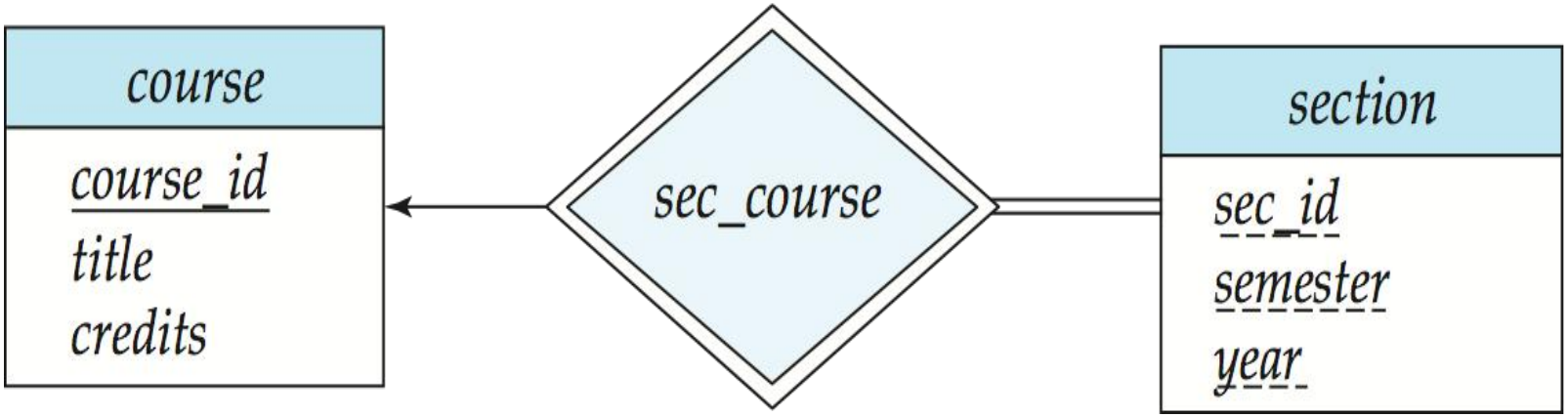
Representing Entity Sets With Simple Attributes

– A strong entity set reduces to a schema with the same attributes

Course(Course_ID, title, tot_cred)

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

section (course_id, sec_id, sem, year)



Note: schemas **sec_course(*course_id, sec_id, sem, year)*** is not represented because ***sec_course*** **schema is redundant in Section *(course_id, sec_id, sem, year)* schema**

Redundancy in Schema Representation

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

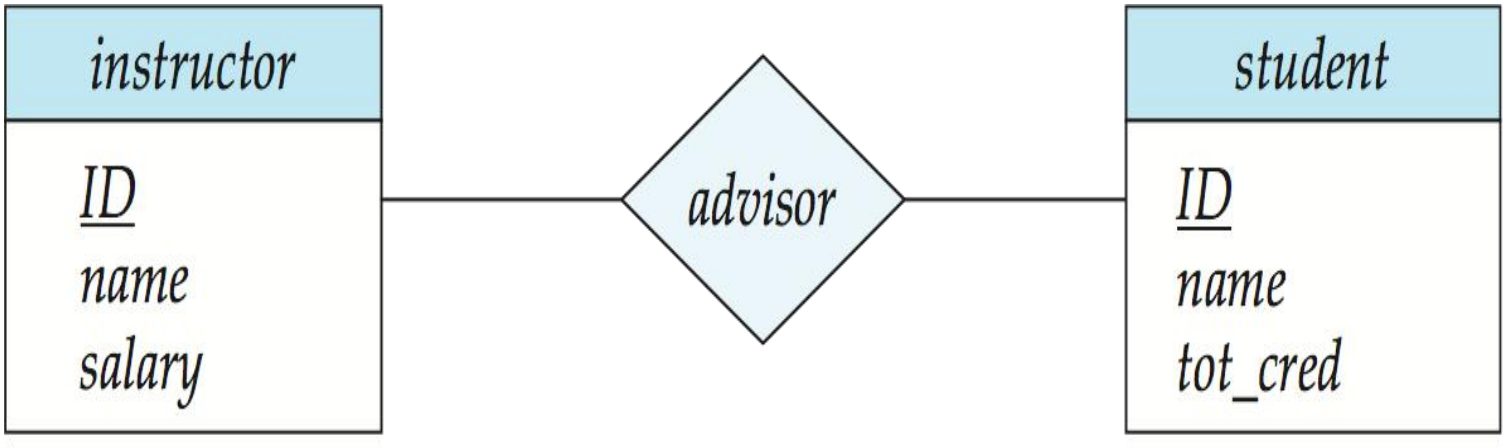
Example: The *section* schema already contains the attributes that would appear in the *sec_course* schema

Representing Relationship Sets Many-Many

- A **many-to-many** relationship set is represented as a schema with attributes for the **primary keys of the two participating entity sets**, and **any descriptive attributes** of the relationship set.
- Example: schema for relationship set *advisor*

advisor (S_ID, I_ID)

Instructor(ID, Name, Salary) & Student(ID, Name, tot_cred)



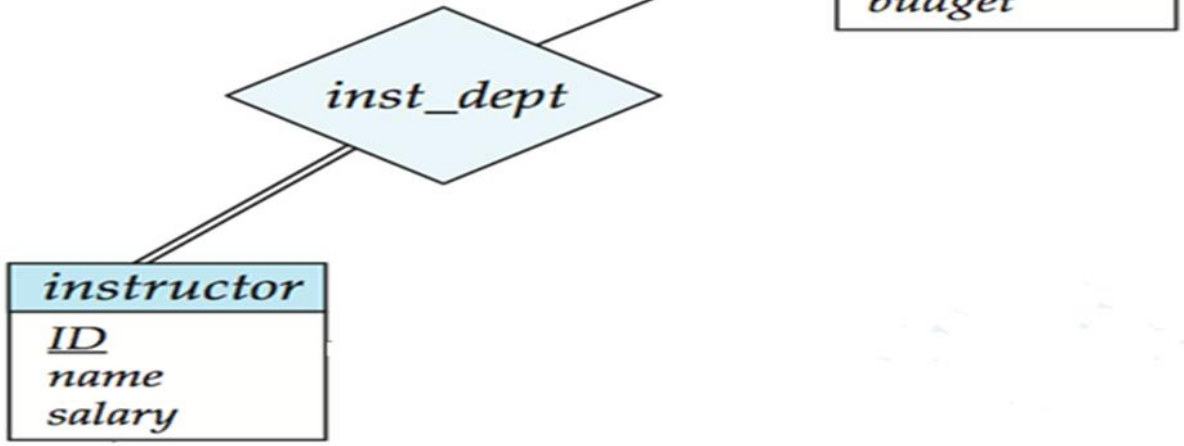
Representing Relationship sets- Many-1/1-Many

- 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
- Example: Instead of creating a schema for relationship set *inst_dept*, add an attribute *dept_name* to the schema arising from entity set *instructor*

Instructor(ID, Name, Salary, Dept_Name)

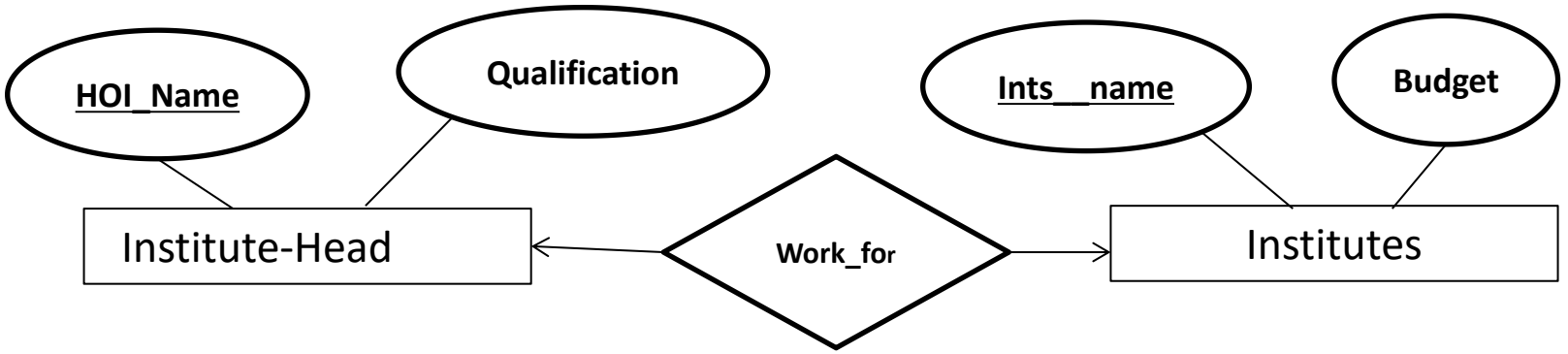
Department(Dept_Name, Building, Budget)

Dept_Name in Instructor is taken as **Foreign key** referencing Department.



Representing Relationship sets- 1 to 1

- For one-to-one relationship sets, either side P.key can be chosen to act as F.key at other side

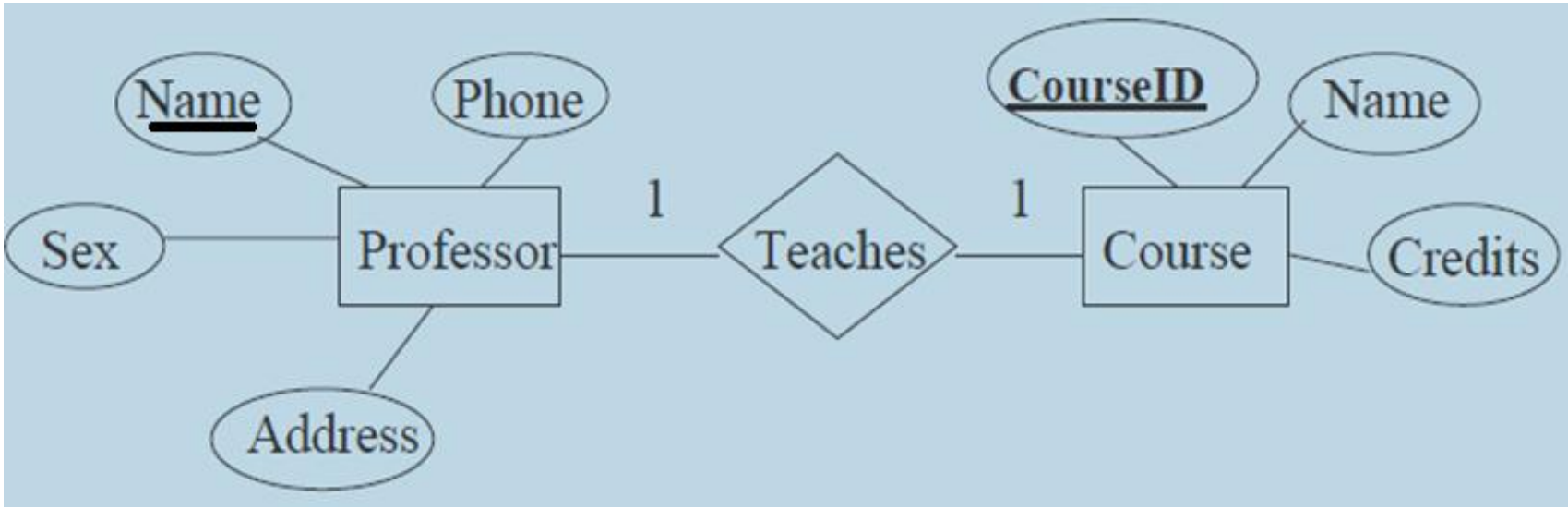


Institute_Head(HOI_Name, Qualification)

Institute(Inst_Name,Budget, HOI_Name)

HOI_Name in Institute is taken as Foreign key referencing Institute_Head.

Reduce the Following ER Diagrams into Relational Schema.

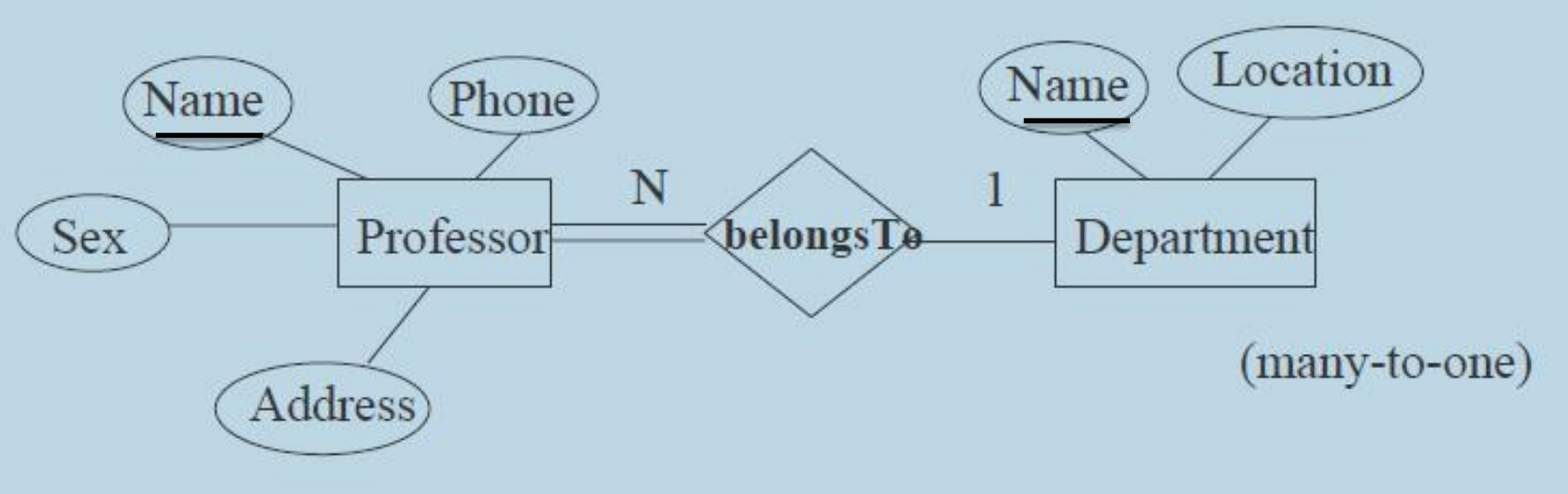


Professor(Name, Phone, Address, Sex, CourseID)

Course(CourseId, Name, Credits)

Professor.CourseID is foreign key referencing Course.CourseID

Reduce the Following ER Diagrams into Relational Schema.



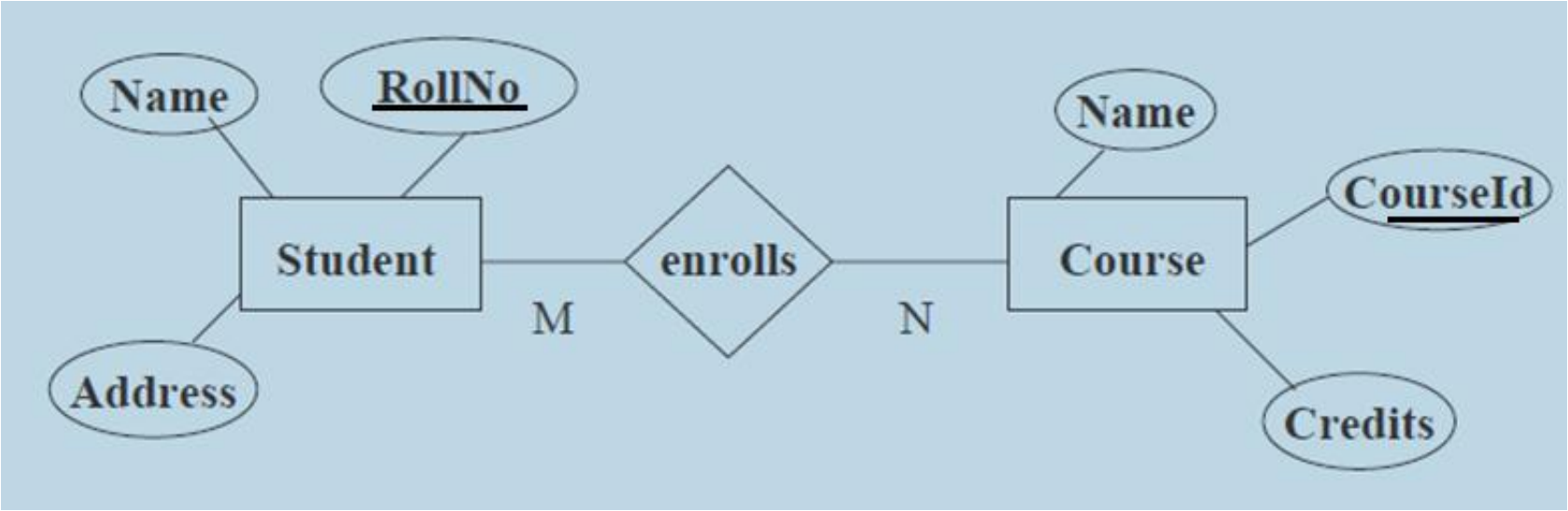
Requirement collected says-**A department contains many Professor**

Professor(Name, Phone, Address, Sex, Dep_Name)

Department(Name, Location)

Dep_Name is foreign key referencing Department. Name

Reduce the Following ER Diagrams into Relational Schema.



Assume, Requirement says- **A course is enrolled by multiple students and a student may also enroll to multiple course**

Assume **Student.Rollno** and **Course. CourseId** is Primary Key in Student and Course entity types respectively.

Student(Name, **RollNo**, Address)

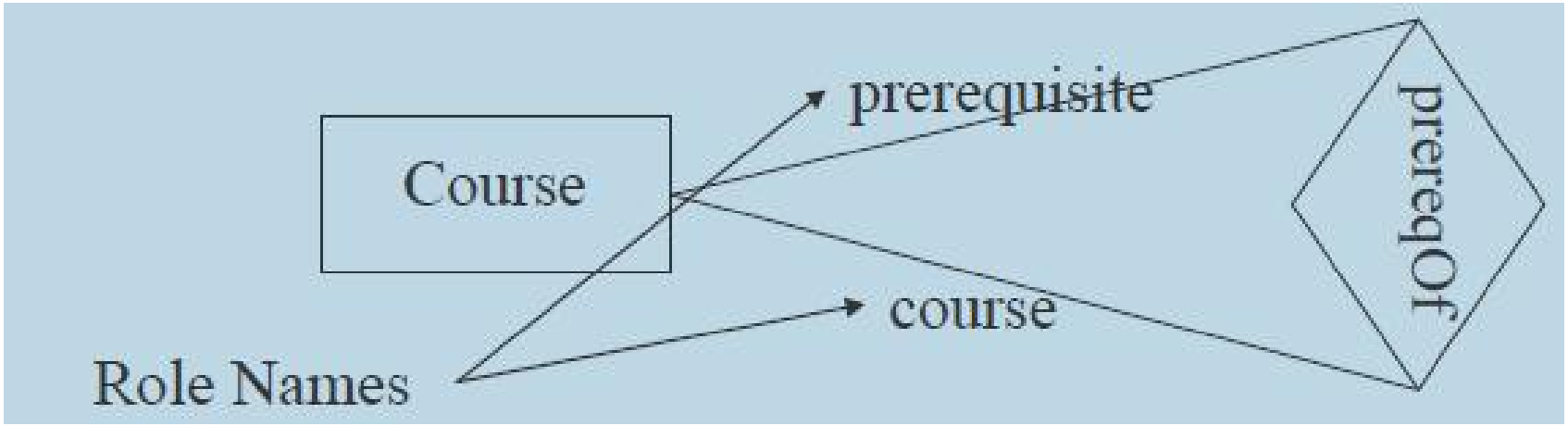
Course(Name, **CourseId**, Credits)

Enrolls(**RollN,CourseId**)

Reduce the Following ER Diagrams into Relational Schema.- Recursive Relationship

Assume that the Prerequisite for a Couse is another Course.

Assume **COURSE** has attributes CourseId, Course_name, Credits



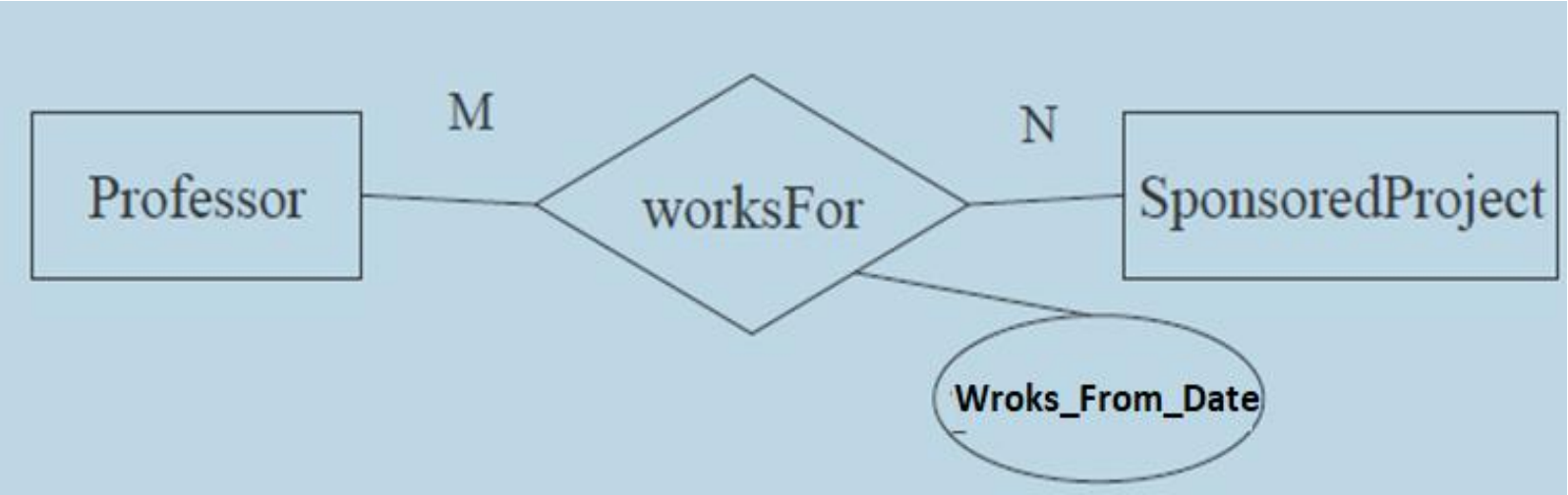
Course(<u>CourseId</u>, Course_Name, Credits, Pre_CourseId)

Note: It is recursive relationship

Pre_CourseId (F.key) Referencing **CourseId** (P.Key)

Reduce the Following ER Diagrams into Relational Schema –with Descriptive Attribute

Assume Professor has attributes – Empno, Name ,Address
and SponseredProject has attributes – Prj_Id, Pname, Duration)
Professors start working on different project from different dates



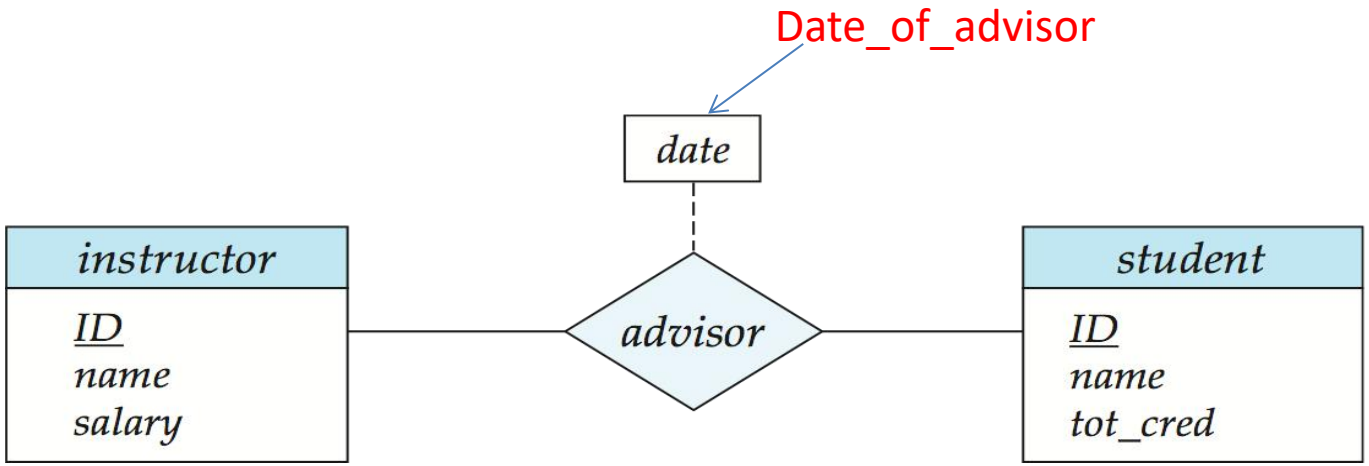
Professor(Empno, Name ,Address)

SponseredProject(Prj_Id, Name, Duration)

Works_for(Empno, Prj_Id, Works_From_Date)

Reduce the Following ER Diagrams into Relational Schema –with Descriptive Attribute

Assume the requirement is- many Instructors can be advisor to multiple students at different dates.

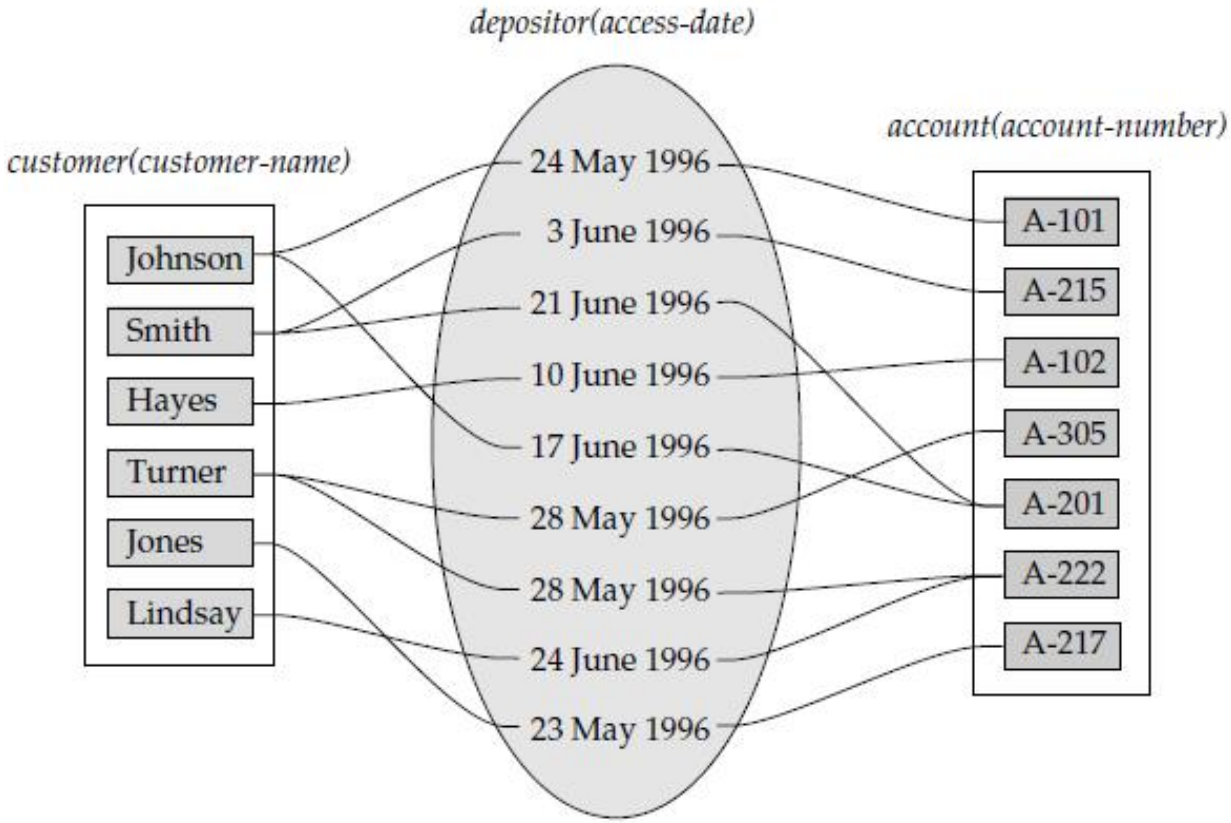


Instructor(ID, name, Salary)

Student(ID, name, tot_cred)

Advisor(I.Id, S.Id, Date_of_advisor)

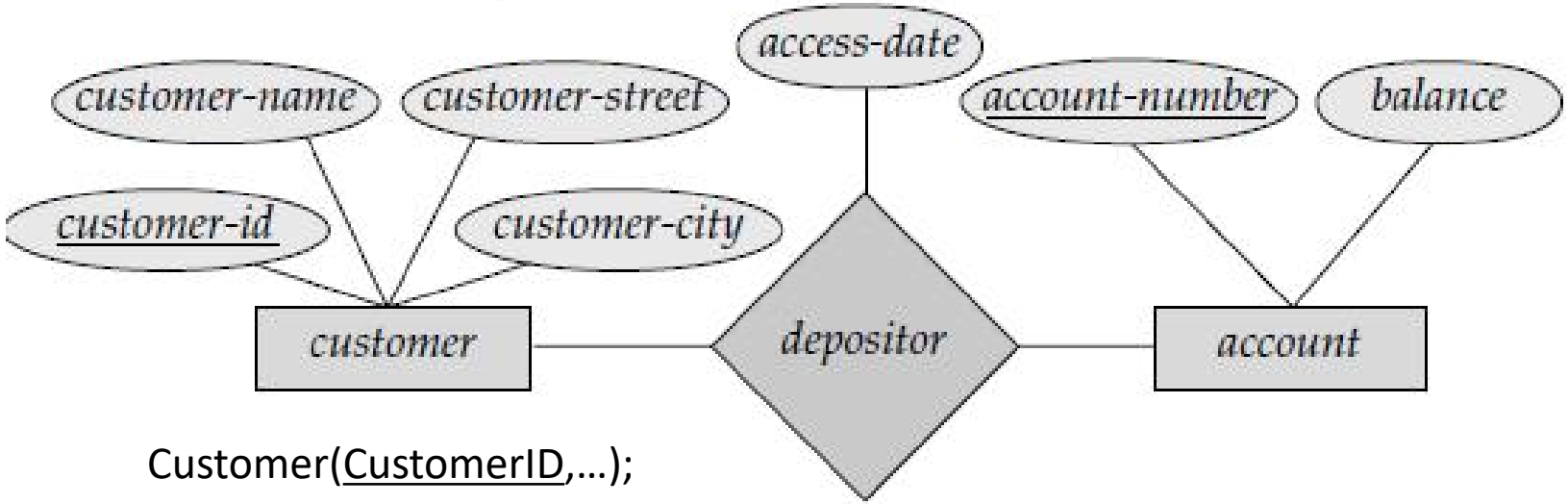
Relation –with Descriptive Attribute



Customer and Account are having many-to-many relationship
and the *access _date* descriptive attribute attached to the relationship set *depositor* to specify the most recent date on which a customer accessed that account.

However if we want to record all dates when customers have accessed their accounts then scenario will be different while converting into schema.
In that case relationships like –
(Smith,21-June-1996,A-201) and (Smith,21-June-1996,A-215) may exist

ER Diagram & SChema –with Descriptive Attribute



Customer(CustomerID,…);

Account(Account_Number, Balance)

Depositor(Account_Number,CustomerID,Access_Date)

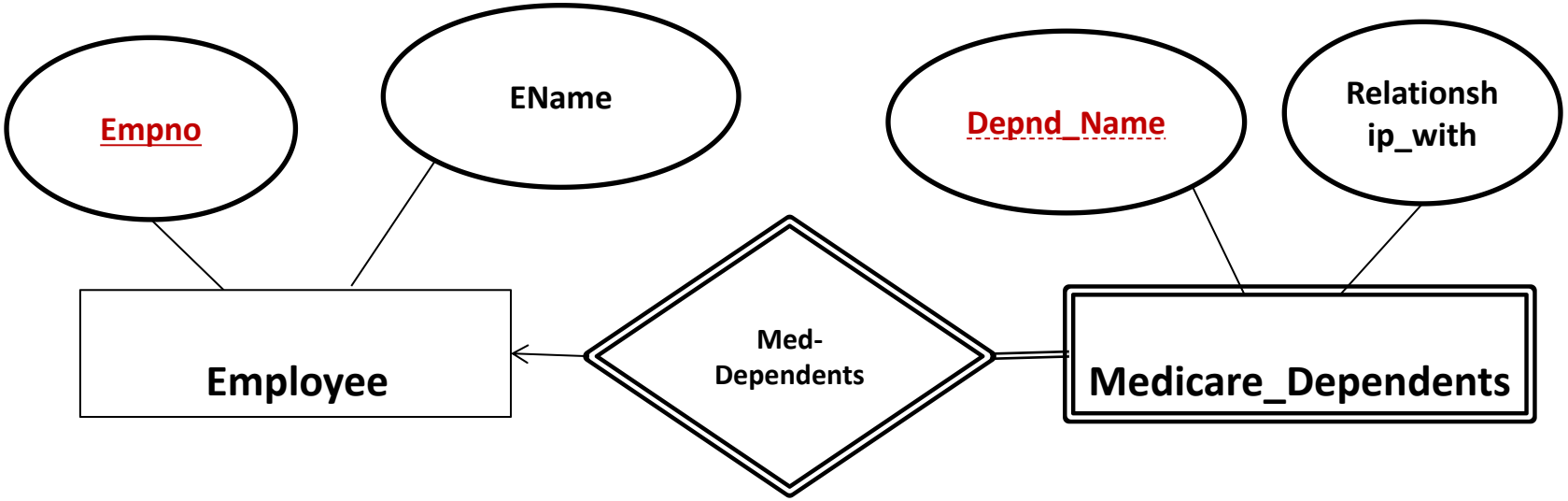
However if we want to record all dates when customers have accessed their accounts then scenario will be different while converting into schema.
In that case relationships like –
(Smith,21-June-1996,A-201) and **(Smith,21-June-1996,A-215)** may exist.
Now Access_Date also will be part of Primary key

Customer(CustomerID,…);

Account(Account_Number, Balance)

Depositor(Account_Number, CustomerID, Access_Date)

Reduce the Following ER Diagrams into Relational Schema- Weak Entity.

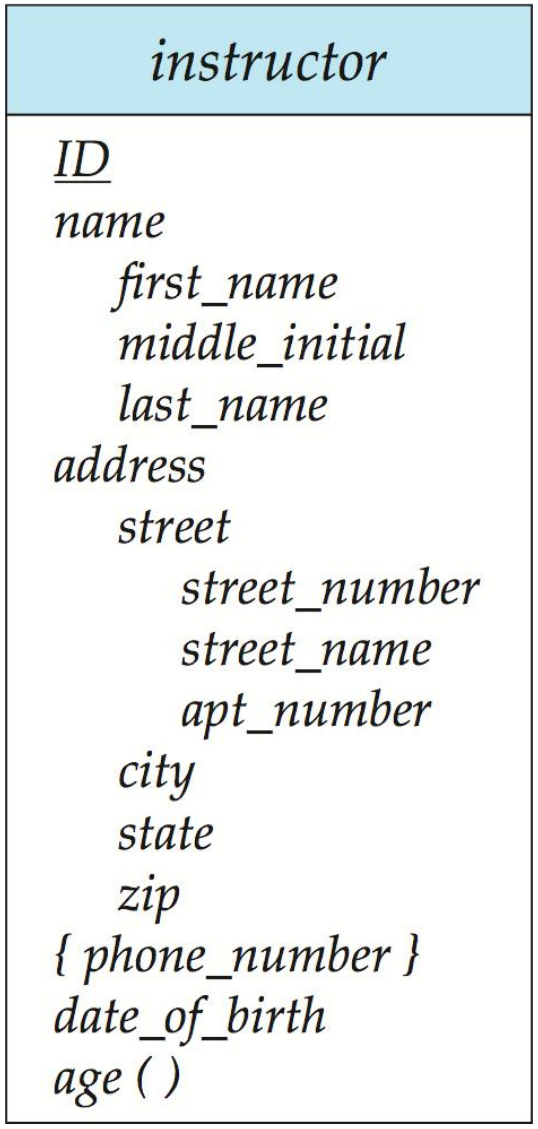


Employee(Empno,Ename)

Medicare_Dependents(Empno,Depnd_Name, Relationship_with)

Empno in Medicare_Dependets is Foreign key Referencing Empno in Employee.

Representing Composite Attributes in Schema



- Composite attributes are **flattened out by creating a separate attribute** for each component attribute
  - Example: given entity set *instructor* with composite attribute *name* with component attributes *first_name* and *last_name* the schema corresponding to the entity set has two attributes *name_first_name* and *name_last_name*
    - *Prefix omitted if there is no ambiguity.*
- instructor schema is
  - ***Instructor (ID, first_name, middle_initial, last_name, street_number, street_name, apt_number, city, state, zip_code, date_of_birth)***

Representing Multivalued Attributes in Schema

- A multivalued attribute ***M*** of an entity ***E*** is represented by a **separate schema *EM***
  - Schema ***EM* has attributes** corresponding to the **primary key of *E*** and an **attribute corresponding to multivalued attribute *M***
  - **Example**: Multivalued attribute *phone_number* of *instructor* is represented by a schema:

    ***inst_phone* (*ID*, *phone_number*)**
  - Each value of the multivalued attribute maps to a separate tuple of the relation on schema *EM*
    - For example, an *instructor* entity with primary key 22222 and phone numbers 456-7890 and 123-4567 maps to two tuples(multi valued):

      (22222, 456-7890) and (22222, 123-4567)

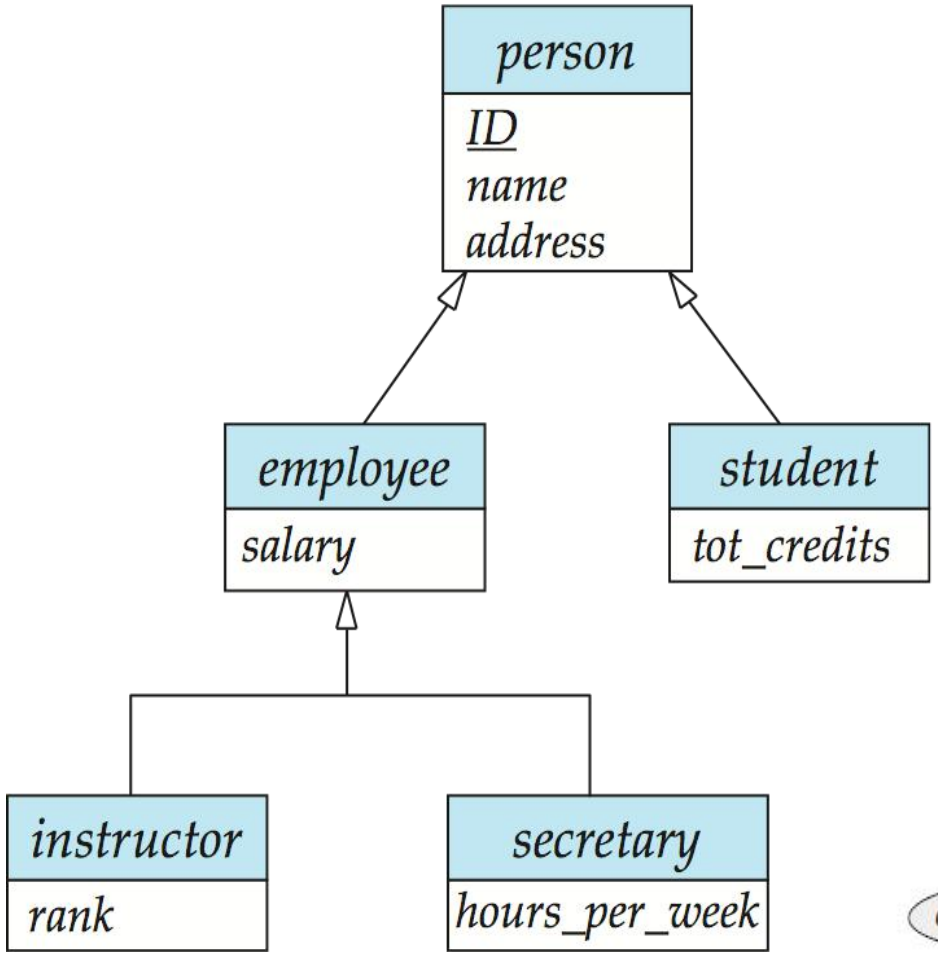
Extended E-R Features

Extended E-R Features: Specialization

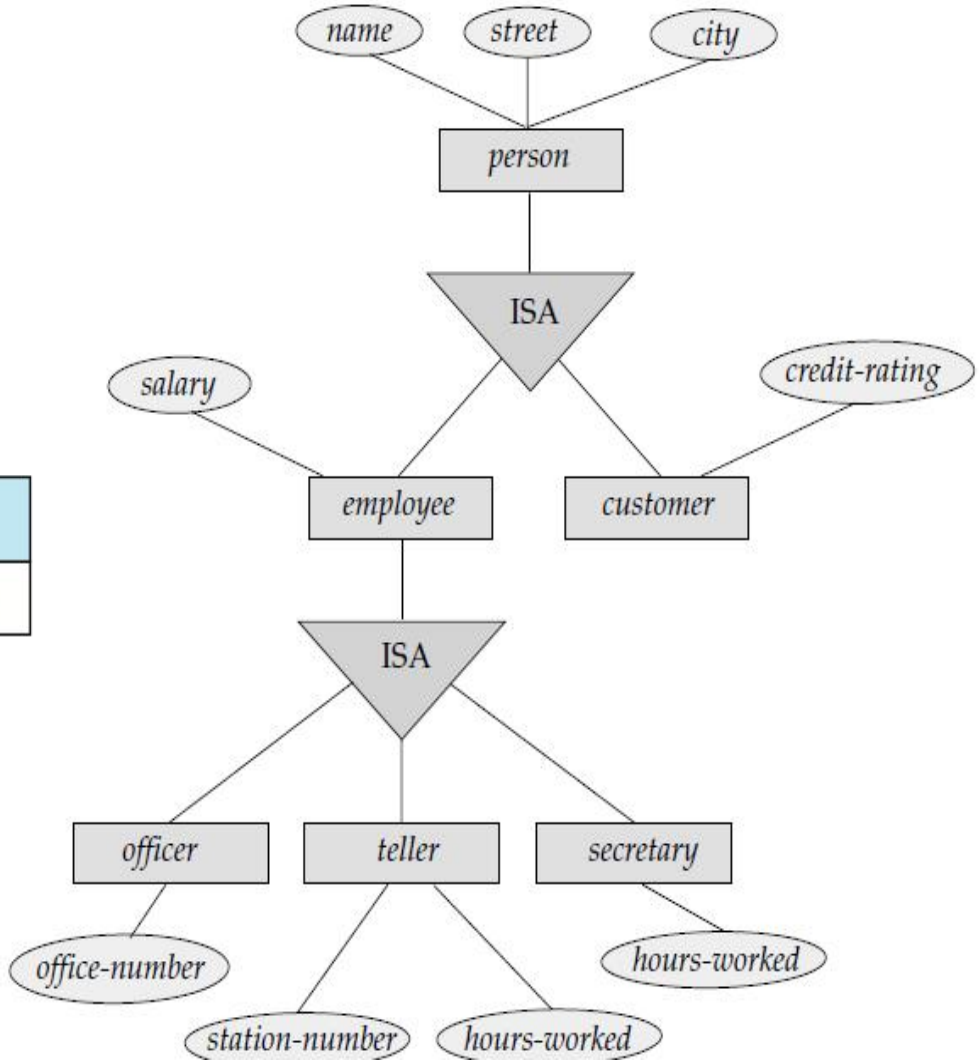
- Specialization is a Top-down design process in which we designate subgroupings within an entity set.
- These subgroupings become lower-level entity sets that **have attributes that do not** apply to the higher-level entity set.
- **Attribute inheritance** – A lower-level entity set inherits all the attributes and relationship participation of the higher-level entity set to which it is linked.
- Depicted by a *triangle* component labeled **ISA**

  (E.g. *instructor* "is a" *person*).
- It can also be referred as a ***superclass-subclass*** relationship

Specialization / Generalization Example



(6th Edition)



(4th Edition)

Design Constraints on a Specialization/ Generalization

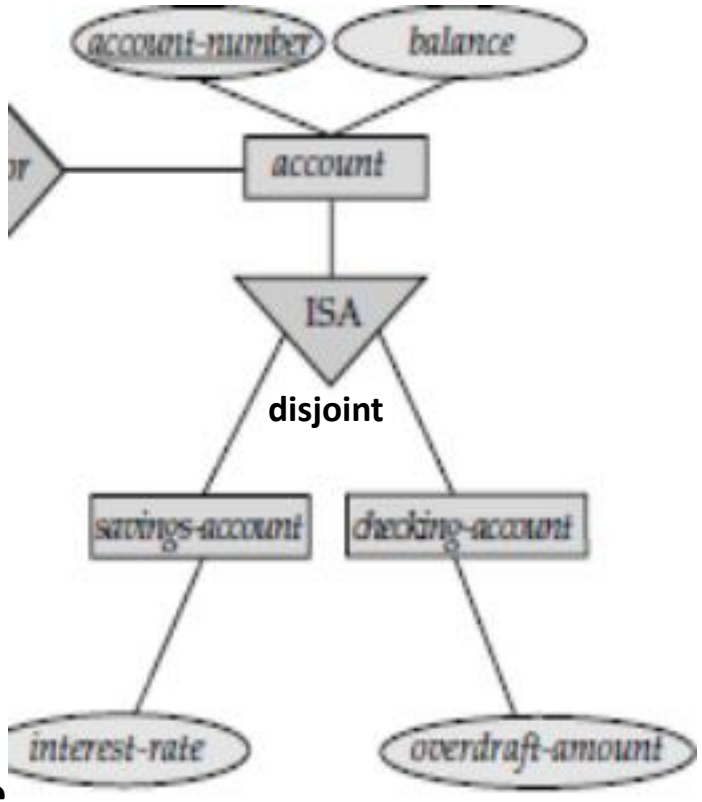
- Constraint on whether or not entities may belong to more than one lower-level entity set within a single generalization.
  - **Disjoint**
    - An higher level entity can belong to **only one lower-level** entity set
    - Noted in E-R diagram by having multiple lower-level entity sets link to the same triangle see Accounts , Savings and Checking entities in next slide.
  - **Overlapping**
    - an entity can belong to **more than one lower-level** entity set.
    - In a Person Entity set is Specialized into Sub classes- Student entity set & Teaching_Assistants.
    - –Some persons are student only and some persons may be students as well as Teaching Assistants

Design Constraints on a Specialization/Generalization (Cont.)

- **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

  Account either have to be savings or checking account

- **partial**: Some higher-level entities may not belong to any lower-level entity set..



Assume that, **after 3 months(or may be after passing some qualifying exam)** of employment, bank employees are assigned to one of four work teams. Therefore before 3 months there will be some employee entities in the top not belonging to any work team lower entities.

Representing Specialization via Schemas

- Method 1:
  - Form a schema for the higher-level entity
  - Form a schema for each lower-level entity sets, include primary key of higher-level entity set and local attributes

schema	attributes
person	***ID, name, street, city***
student	***ID, tot_cred***
employee	***ID, salary***

  - Drawback: getting information about, an *employee* requires accessing two relations, the one corresponding to the low-level schema(**employee**) and the one corresponding to the high-level schema(**person**)

Representing Specialization as Schemas (Cont.)

- Method 2:
  - Form a schema for each entity set with all local and inherited attributes

schema	attributes
student	***ID, name, street, city, tot_cred***
employee	***ID, name, street, city, salary***

  - If specialization is total, the schema for the generalized entity set (*person*) not required to store information
    - Can be defined as a "view" relation containing union of specialization relations
  - Drawback: *name, street* and *city* may be stored redundantly for people who are both students and employees

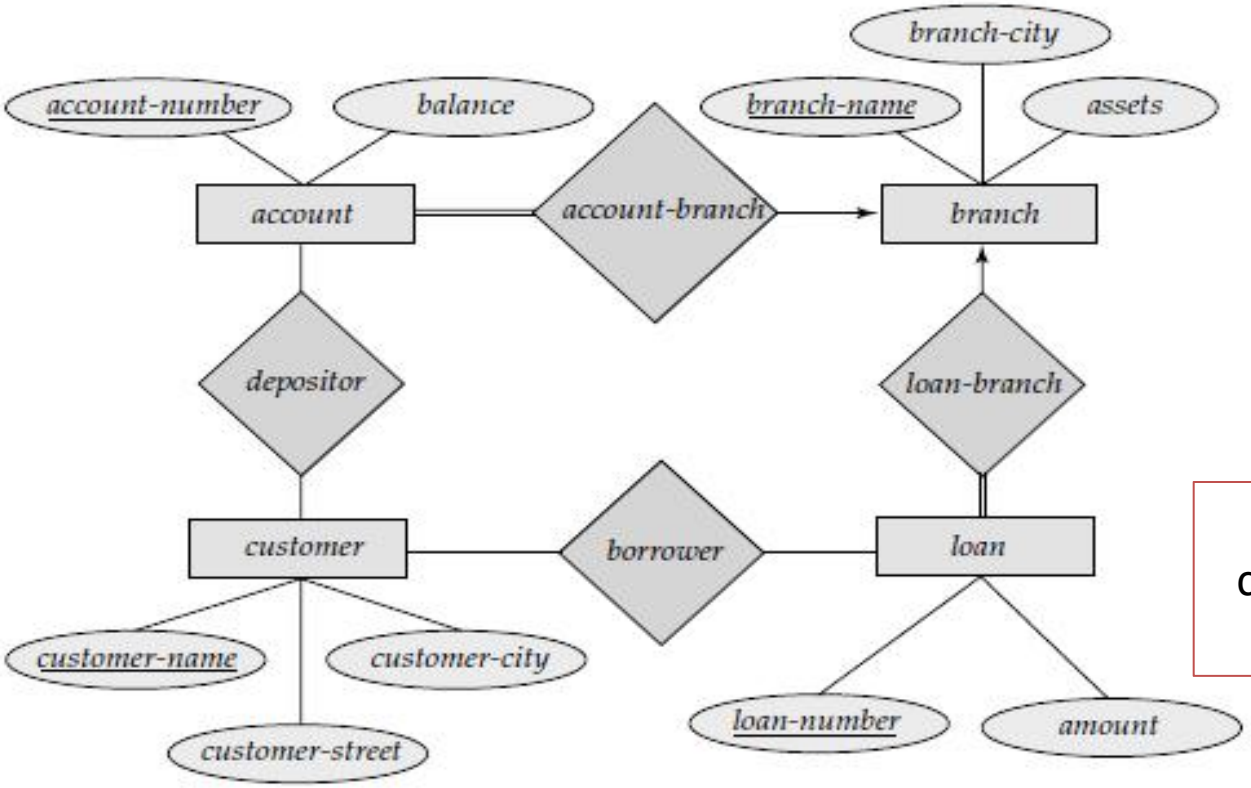
Draw an ER diagram for the following Requirements

Consider a Banking system, which stores information about it's different branches at different locations. Each branch has a unique branch name, location and assets (in Rupees).

Each branch maintains many Savings accounts of different customers. System is interested to record SB account number and balance amount of each savings account.

Each customer may have many accounts and an account may be owned jointly by multiple Customers. Customer information such as-unique Customer_Name, city and street need to be stored in the system.

Each branch also gives loans to customers. About Loans we are interested to store information such as unique loan_number, Loan_amount. Each branch maintains many loan accounts. Each customer is allowed to take many loans and at the same time a loan may be jointly taken by many Customer.



The relational schema corresponding to this ER diagram is below-

Account(account-number, balance, branch-name) branch-name References Branch

Branch(branch-name, branch-city, assets)

Loan(loan-number, amount, branch-name) branch-name References Branch

Customer(customer-name, customer-city, customer-street)

Borrower(customer-name,loan-number) customer-name References Customer & loan-number References Loan.

Depositor(account-number,customer-name) account-number References Account, customer-name References Customer

Schema Diagrams

A database schema, along with primary key and foreign key dependencies, can be depicted by **schema diagrams**.

Each relation appears as a box, with the relation name at the top in box, and the attributes listed inside the box. **Primary key** attributes are shown **underlined**. **Foreign key** dependencies appear as **arrows from the foreign key attributes of the referencing relation to the primary key** of the referenced relation.

Schema diagram corresponding to Relational schema is given in the next slide.

Relational Schema & Schema Diagram

Account(account-number, balance, branch-name) branch-name References Branch

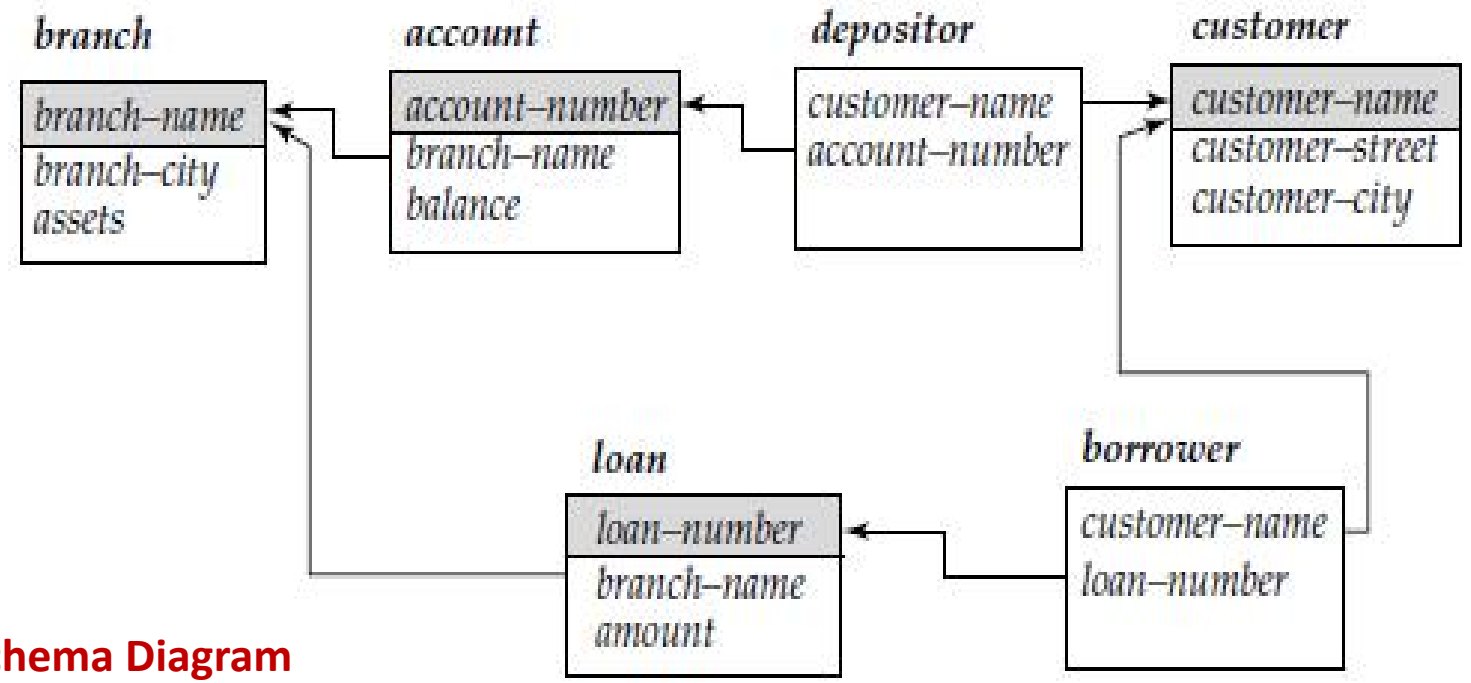
Branch(branch-name, branch-city, assets)

Loan(loan-number, amount, branch-name) branch-name References Branch

Customer(customer-name, customer-city, customer-street)

Borrower(customer-name,loan-number) customer-name References Customer & loan-number References Loan.

Depositor(account-number,customer-name) account-number References Account, customer-name References Customer



Schema Diagram

Oracle SQL Commands Corresponding to Relational schema given in Previous Slide(For Lab implementation)

CREATE TABLE BRANCH (BRANCH-NAME VARCHAR2(15) PRIMARY KEY, BRANCH-CITY VARCHAR2(20), ASSETS NUMBER(10,2));

CREATE ACCOUNT (ACCOUNT-NUMBER VARCHAR2 (3) PRIMARY KEY, BALANCE NUMBER(8,2) CHECK(BALANCE >1000), BRANCH-NAME VARCHAR2(15) NOT NULL REFERENCES BRANCH)

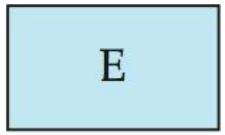
CREATE TABLE LOAN(LOAN-NUMBER VARCHAR2(3) PRIMARY KEY, AMOUNT NUMBER(5,1), BRANCH-NAME VARCHAR2(15) REFERENCES BRANCH)

CREATE TABLE CUSTOMER (CUSTOMER-NAME VARCHAR2(10) PRIMARY KEY, CUSTOMER-CITY VARCHAR2(110),CUSTOMER-STREET VARCHAR2(120))

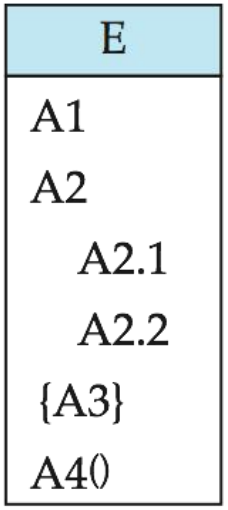
CREATE TABLE BORROWER (CUSTOMER-NAME VARCHAR2(10) REFERENCES CUSTOMER,LOAN-NUMBER VARCHAR2(3) REFERENCES LOAN, PRIMARY KEY(CUSTOMER-NAME, LOAN-NUMBER));

CREATE TABLE DEPOSITOR (ACCOUNT-NUMBER VARCHAR2 (3) REFERENCES ACCOUNT,CUSTOMER-NAME VARCHAR2(10) REFERENCES CUSTOMER), PRIMARY KEY(CUSTOMER-NAME, ACCOUNT-NUMBER));

Summary of Symbols Used in E-R Notation



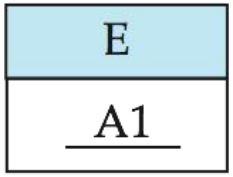
entity set



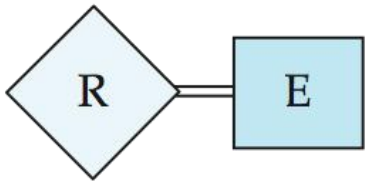
attributes:
simple (A1),
composite (A2) and
multivalued (A3)
derived (A4)

relationship set

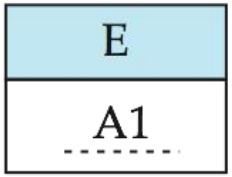
identifying
relationship set
for weak entity set



primary key

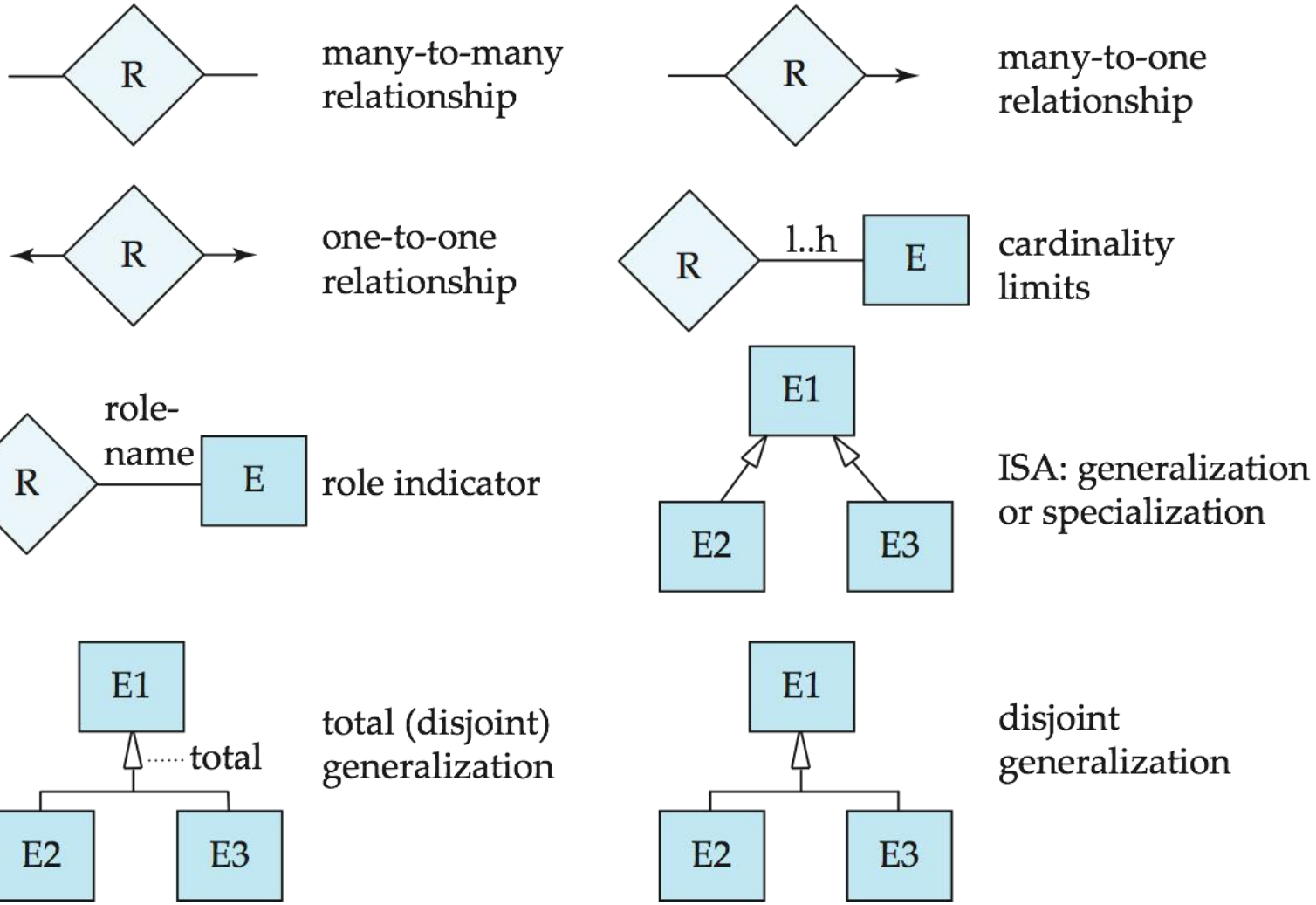


total participation
of entity set in
relationship



discriminating
attribute of
weak entity set

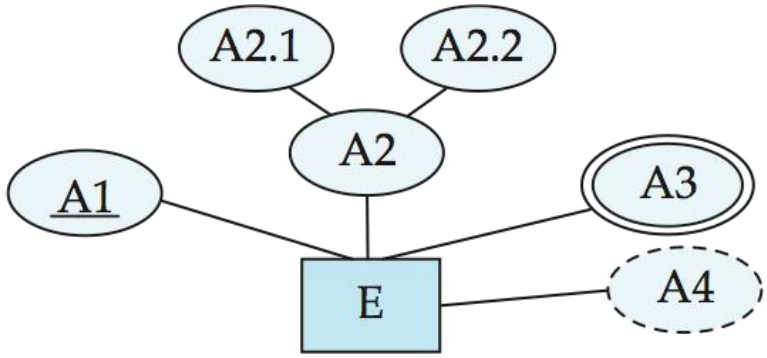
Symbols Used in E-R Notation (Cont.)



Alternative ER Notations

- Chen, IDE1FX, …

entity set E with
simple attribute A1,
composite attribute A2,
multivalued attribute A3,
derived attribute A4,
and primary key A1

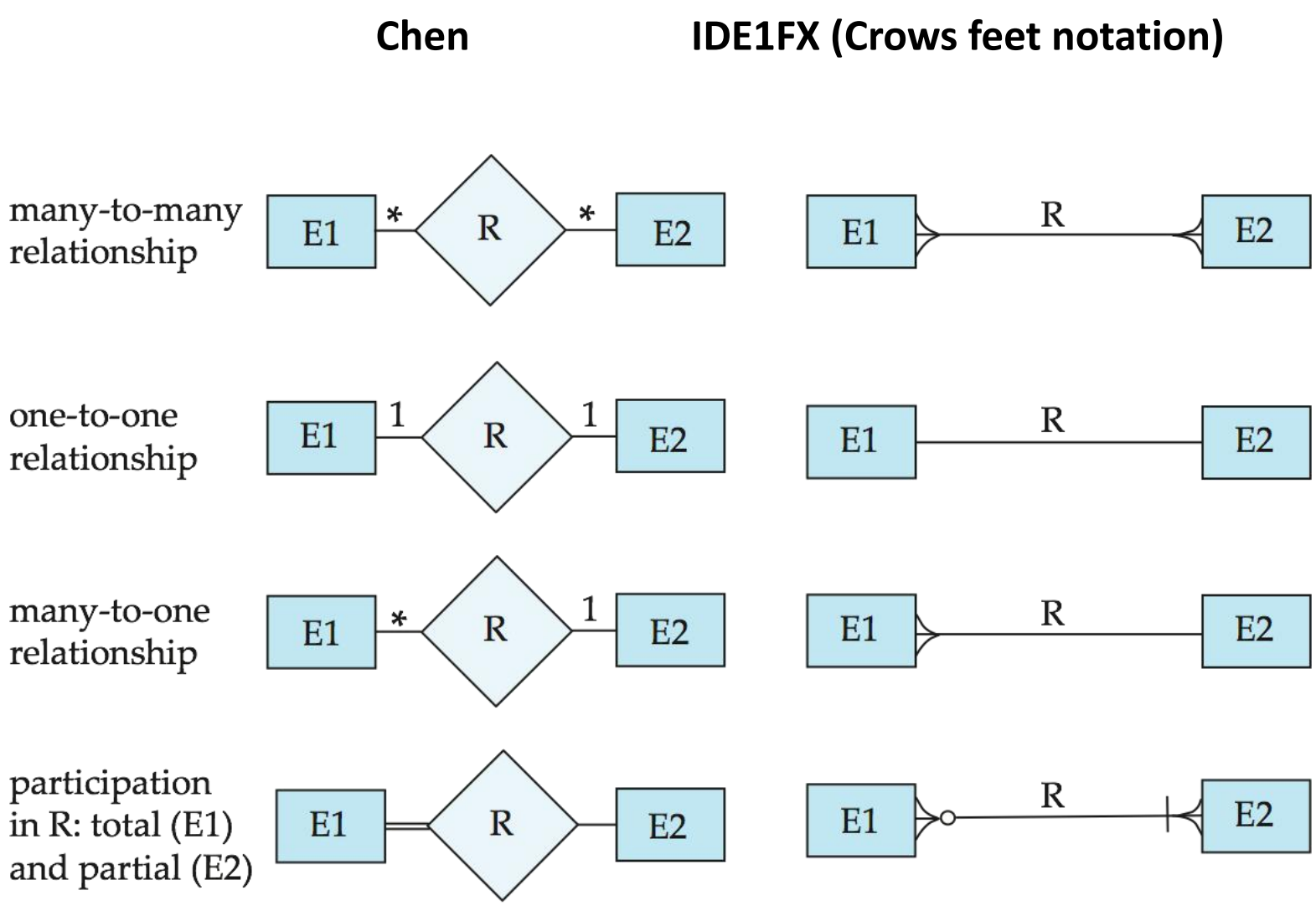


weak entity set 

generalization 

total
generalization 

Alternative ER Notations



Summary

- A **database-management system** (DBMS) consists of a collection of interrelated data.
- Database design mainly involves the design of the database schema. The **entity-relationship (E-R)** data model is a widely used data model for database design.
- An **entity** is an object that exists in the real world and is distinguishable from other objects by the properties(attributes) it is having.
- A **relationship** is an association among several entities. A **relationship set** is a collection of relationships of the same type, and an **entity set** is a collection of entities of the same type.
- The terms **super key**, **candidate key**, and **primary key** apply to entity and relationship sets as they do for relation schemas.
- **Mapping cardinalities** express the number of entities to which another entity can be associated via a relationship set.
- An entity set that does not have sufficient attributes to form a primary key is termed a **weak entity set**. An entity set that has a primary key is termed a **strong entity set**.
- **Specialization** and **generalization** define a containment relationship between a higher-level entity set and one or more lower-level entity sets.

Some Questions

1. Explain the difference between a weak and a strong entity set.
2. Explain the distinctions among the terms primary key, candidate key, and super key with an example.
3. A weak entity set can always be made into a strong entity set by adding to its attributes the primary-key attributes of its identifying entity set. Outline what sort of redundancy will result if we also take into account of cardinality also while converting to schema.
4. Is an weak entity has to have Total Participation ?
5. Construct an E-R diagram for a car-insurance company whose customers own one or more cars each. Each car has associated with it zero to any number of recorded accidents.

ER MODEL

END OF CHAPTER