Entity-Relationship Diagram (ERD)

ER Model/Diagram

- ✓ a data modeling technique that graphically illustrates an information system’s entities and the relationships between those entities.
- ✓ a conceptual and representational model of data used to represent the entity framework infrastructure.
- ✓ used as a high-level logical data model, which is useful in developing a conceptual design for databases.
- ✓ An ER Model maps well to the relational model. i.e. the constructs used in ER Model can be easily transferred into relational table.
- ✓ An ER Model can be used by the database designer to communicate the database design to the user.
- ✓ An ER Model can be used as a design plan by the database developer to implement a data model in specific DBMS software like Oracle, Ms SQL .

Entity-Relationship (E-R) Model

The basics of ER Model

- **Entities**
- **Attributes**
- **Relationship**

Entities

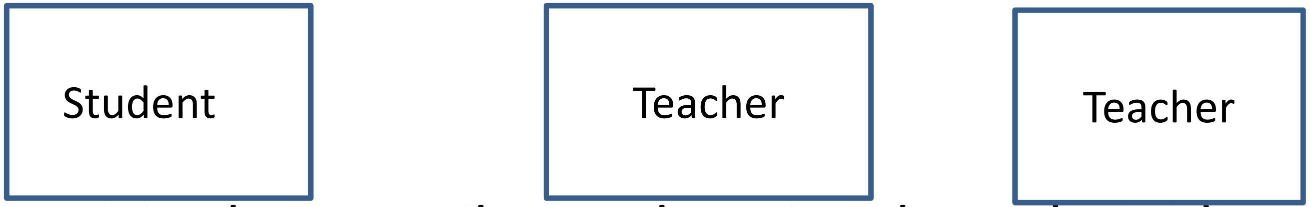
- An entity is a “thing” or “object” in the real world that is distinguishable from other objects.
- Entity can have anything that has an independent existence and about which we collect data.

Entity-Relationship (E-R) Model

E.g.

In a school database, the students, teachers, classes, courses, or projects can be taken as an entity.

Entities are represented by means of rectangle



Student

Teacher

Teacher

- ➢ Entities have attributes that give them their identity.

E.g.: students have roll_no, names, and address.

- ➢ Entities becomes table in relational model.

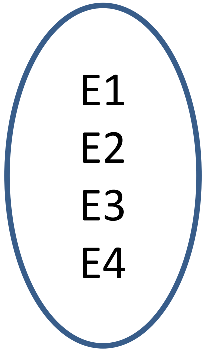
Entity-Relationship (E-R) Model

Entity set

- An entity set is a collection of similar type of entities that share same attributes.

E.g.: a student set contain all the students of a school. A teacher set may contain all the teachers of a school from all faculties.

- Entity set need not be disjoint.



Entity set

Entity-Relationship (E-R) Model

E.g.: the entity set Employee (all employees of a bank) and the entity set Customer(all customer of a bank) may have members in common.

Attributes

- An entity is represented by a set of attributes.
- Attributes are used to describe the property of an entity.

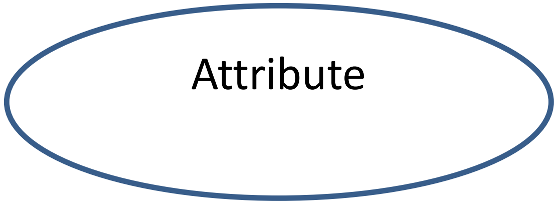
E.g.: a student entity may have Roll_no, Name, DOB, Age, Address , Mobile_no as attribute.

- For each attribute there is a set of permitted values called domain(or range) of that attribute.

Entity-Relationship (E-R) Model

E.g.: a student’s name cannot be numeric value, it has to be alphabetic. A student’s age cannot be negative value etc. A student’s roll no. can be numeric between some range(0-10000).

- **Attributes** are represented by ellipse.



Schema: entity type(name of attributes)

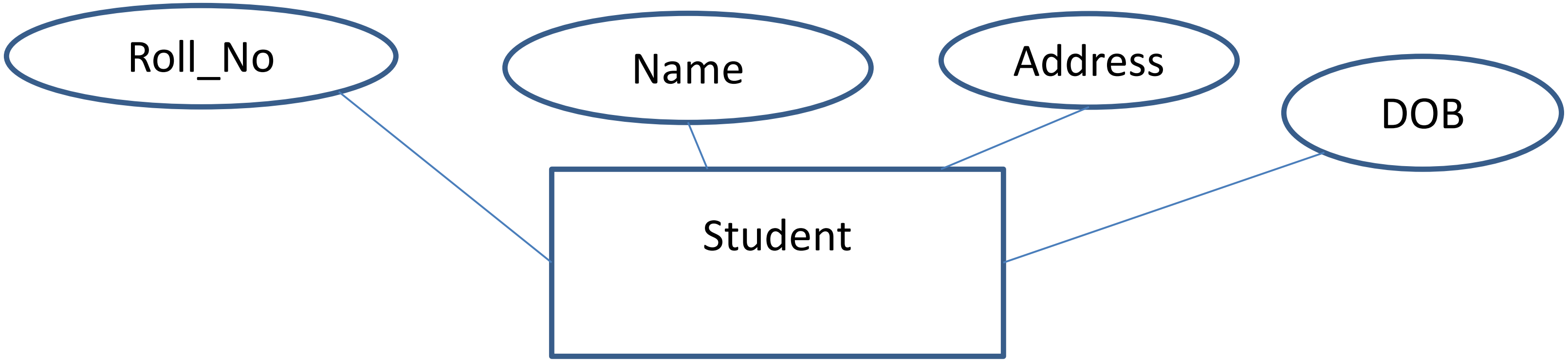
Schema: student(Roll_No, Name, DOB, Address)

Schema: employee(Emp_ID, Name, DOB, age, salary, Address, Designation, contact_no, Deptt_Name)

Entity-Relationship (E-R) Model

E.g.

Schema: Student(Roll_no, Name, DOB, Address)



Entity1: 101, Rakesh, 10-02-2005, kathmandu

Entity 2: 102, Prakash, 11-12-2006, Pokhara

Entity3: 103, Shyam, 12-09-2005, Biratngar

Entity-Relationship (E-R) Model

Relationships

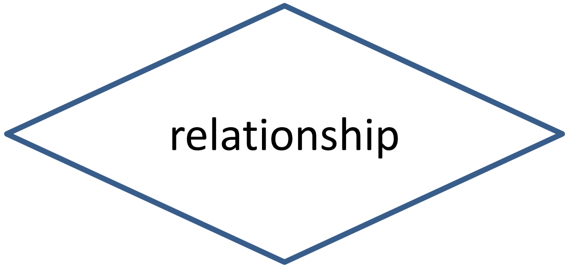
- A relationship is an association among entities.

E.g.: an employee **works_at** a department.

a student **Enrolls** in a course.

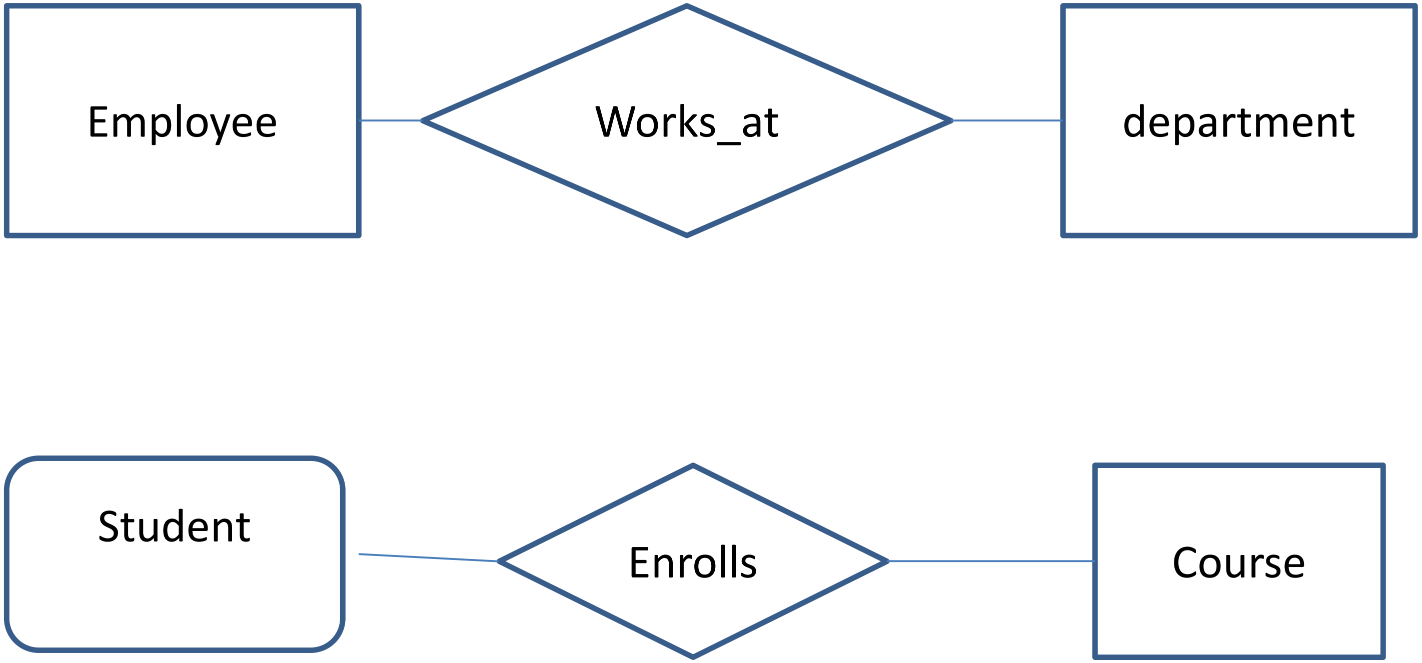
Here, **works_at** and **Enrolls** are called relationships.

- Relationships are represented by **diamond_shaped** box.



Entity-Relationship (E-R) Model

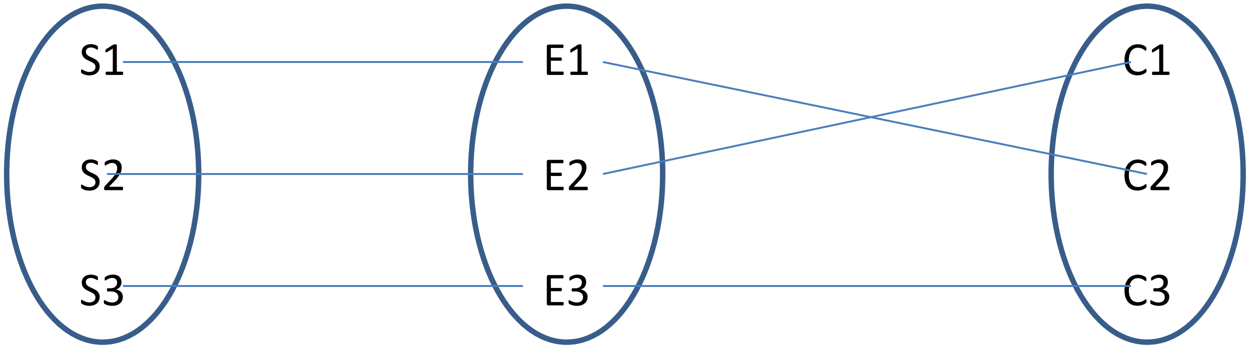
Relationships



Entity-Relationship (E-R) Model

Relationship set

- A set of relationships of similar types is called relationship set.
- The following relationship set Enrolls (E1, E2, E3) depicts;



Student
Entity set

Enrolled_in
relationship set

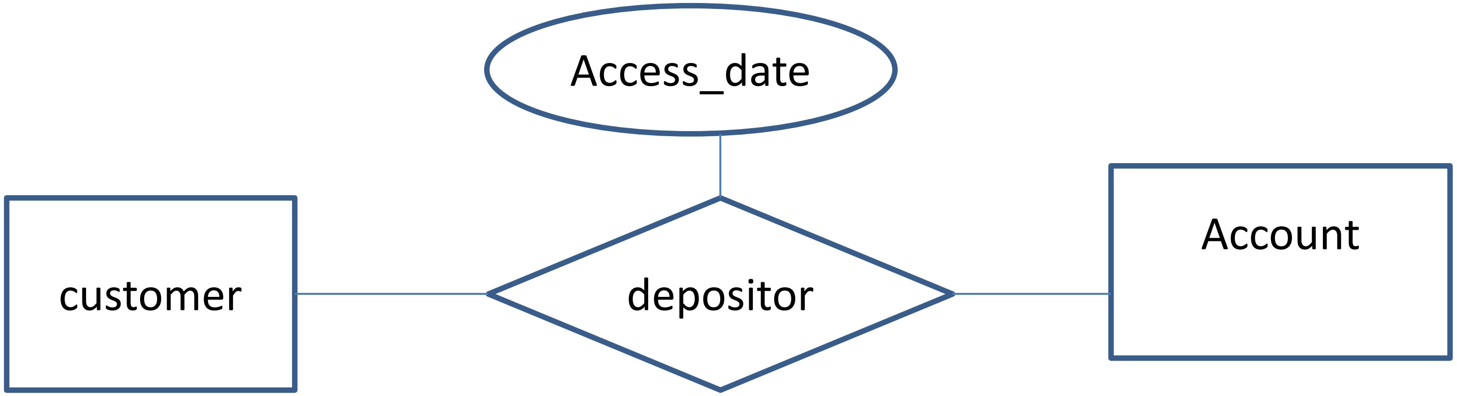
Course
Entity Set

S1 is Enrolled in C2, S2 is enrolled in C1, and S3 is enrolled in C3

Entity-Relationship (E-R) Model

Descriptive Attribute

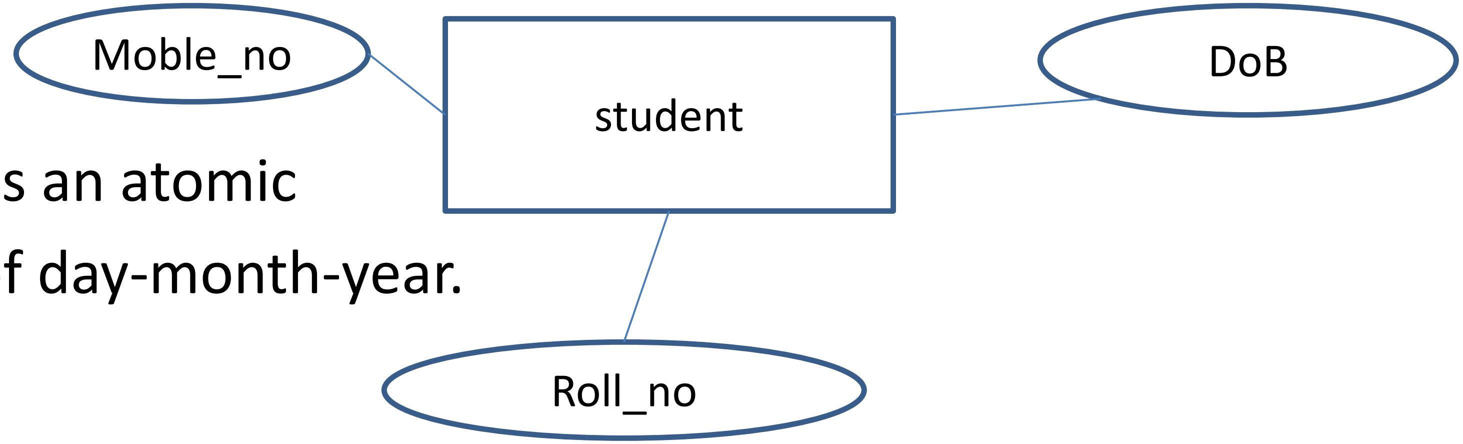
Like entities, relationship too can have attributes. These attributes are called descriptive attributes.



Types of Attributes

1. **Simple attribute:** simple attributes are atomic values, which cannot be divided further

E.g.: a student’s mobile number is an atomic value of 10 digits.

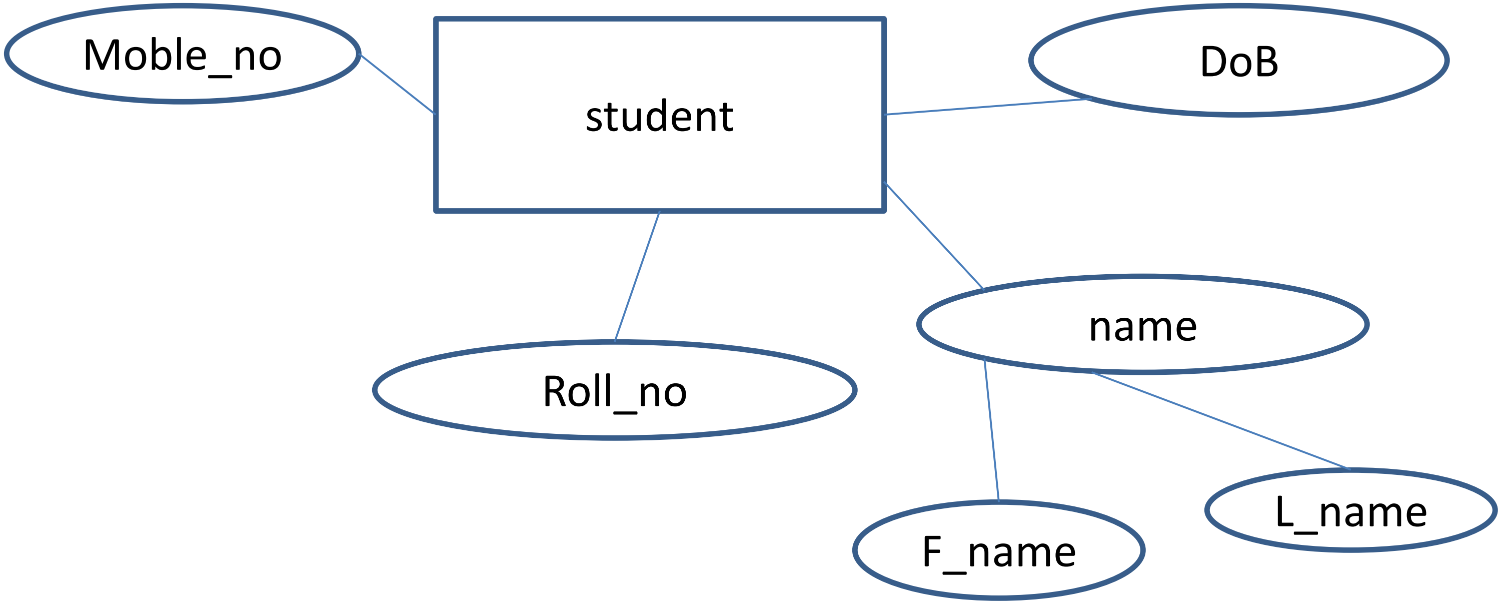


a DOB is an atomic

Value of day-month-year.

Types of Attributes

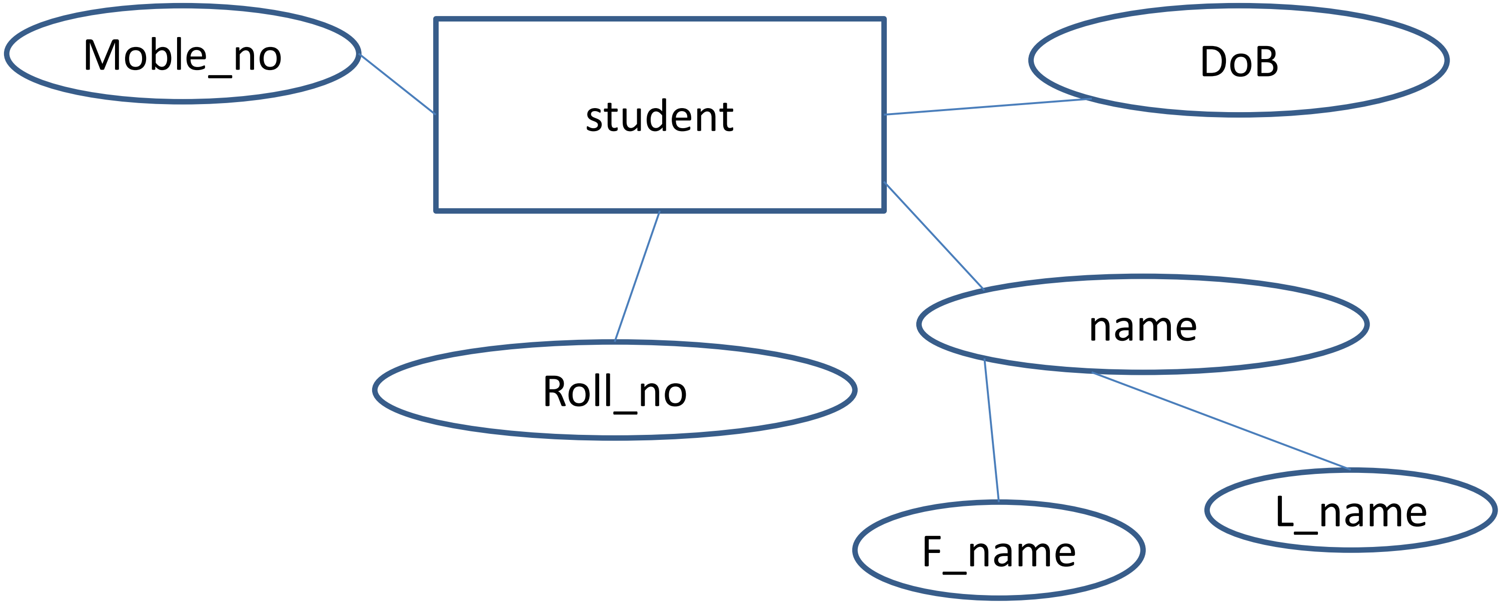
2. Composite Attribute: composite attributes are made of more than one simple attributes.



Types of Attributes

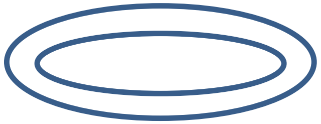
3. Single valued attribute and multi valued attribute

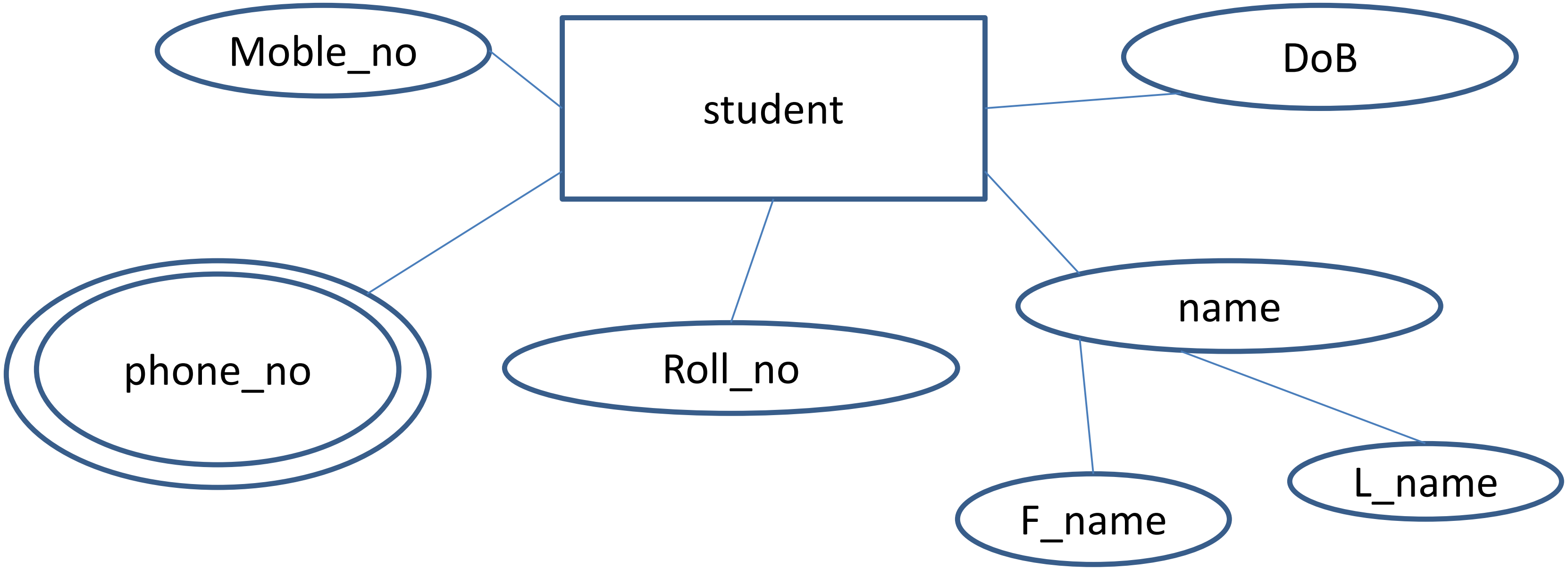
Single valued attribute contain single value. E.g. citizenship number, Roll no, social_security_number .



Types of Attributes

Multivalued attribute may contain more than one values.

Represented by double ellipse. 



Types of Attributes

4. Stored and derived attributes: stored attributes are physically stored in the database.

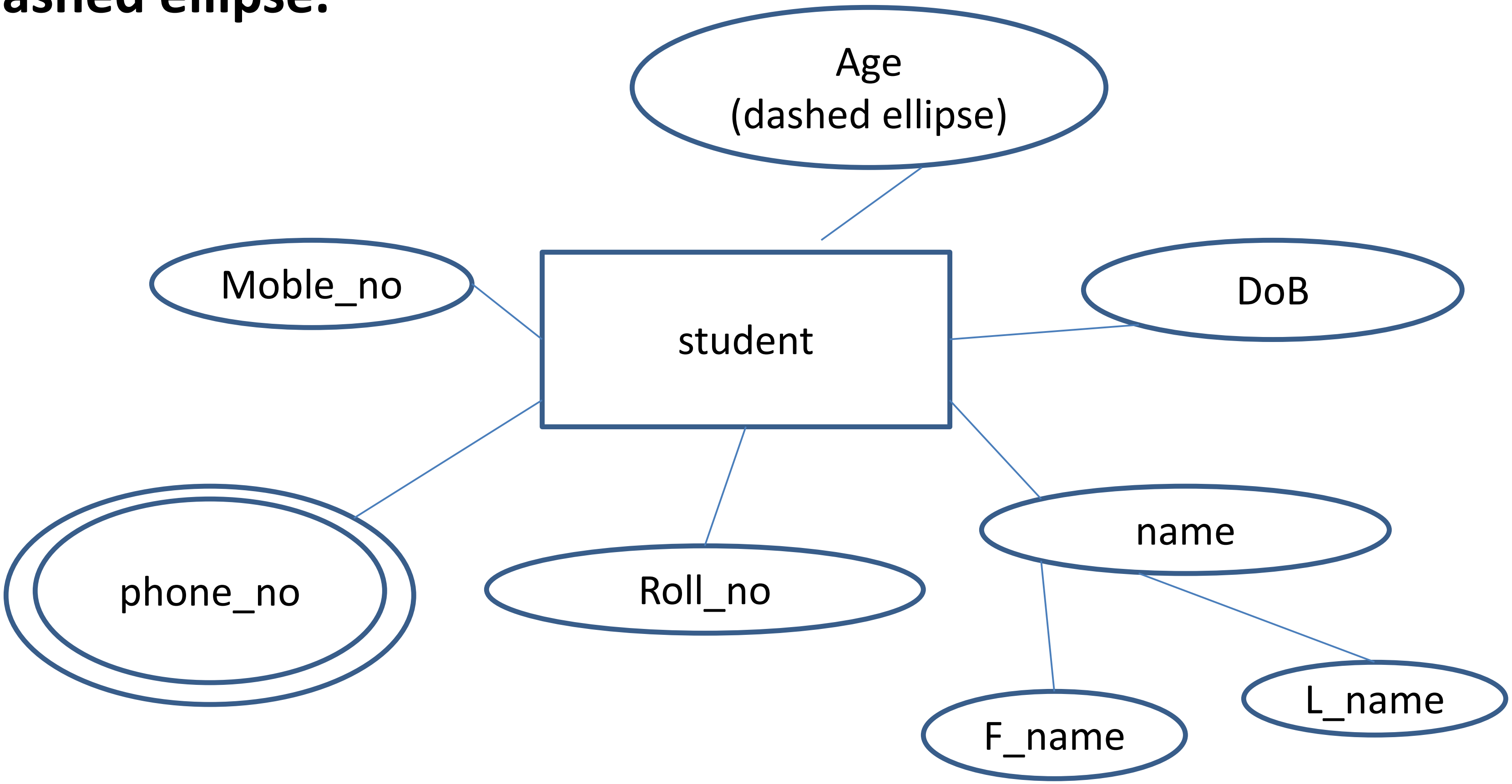
Derived attributes are the attributes that do not exist in the physical database , but their values are derived from other attributes available in the database. Derived attributes are represented by **dashed ellipse**.

dashed ellipse

- Age can be derived from DOB.
- Average salary in the department should not be saved directly in the database, instead it can be derived.

Types of Attributes

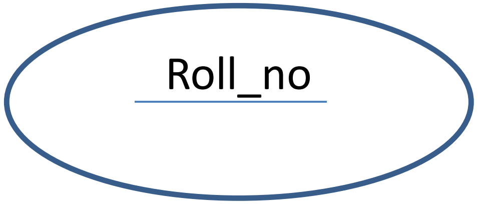
Dashed ellipse.



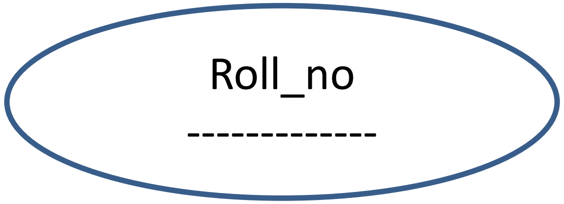
Types of Attributes

5. Key attribute: the attribute which uniquely identifies each entity in the entity set is called key attribute.

- It represents a primary key and foreign key .



Primary Key



Foreign Key

Steps involved in creating an ERD include:

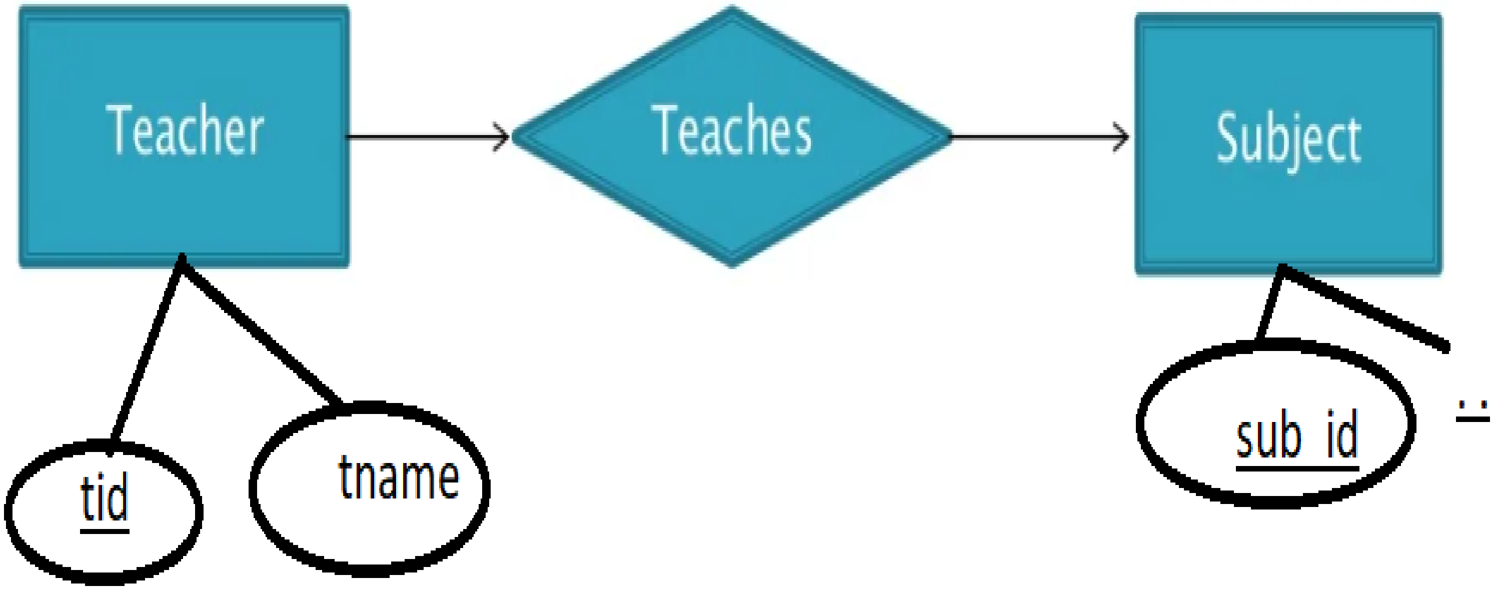
1. Identifying and defining the entities.

2. Determining all interactions between the entities.

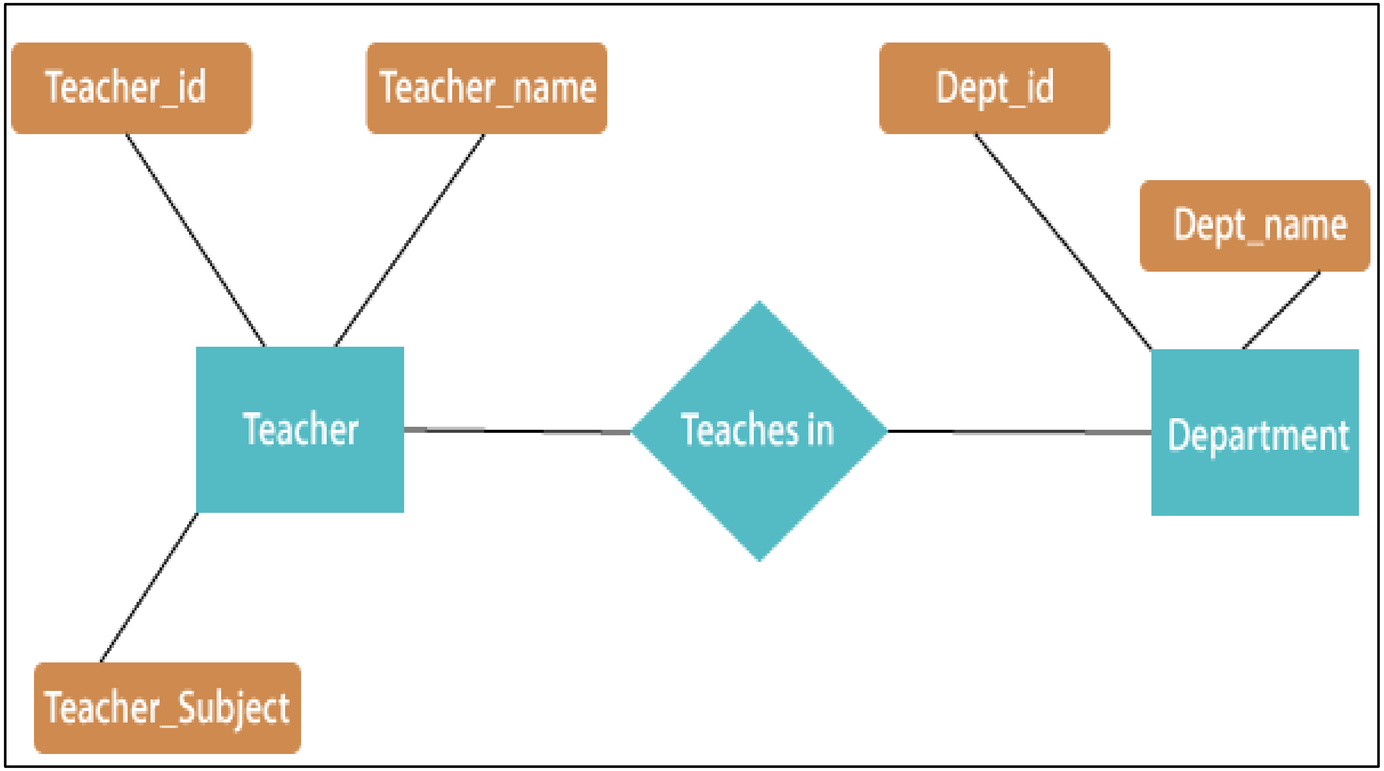
3. Analyzing the nature of interactions/determining the cardinality of the relationships.

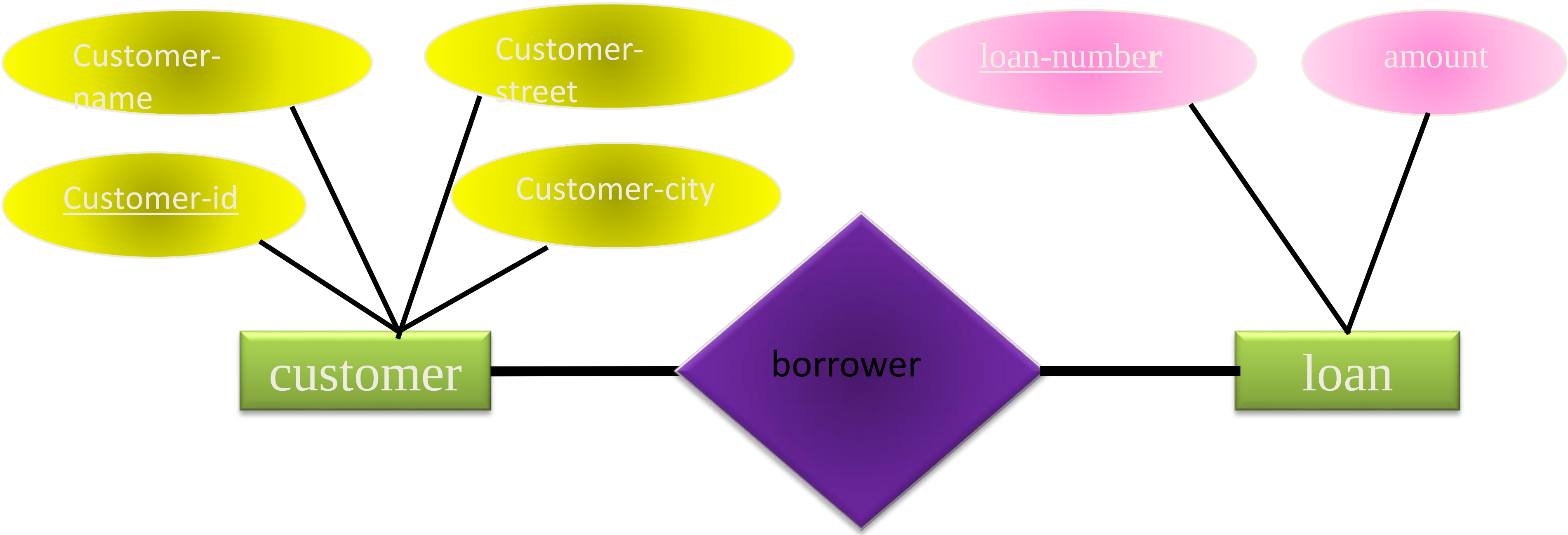
4. Creating the ERD.

Example

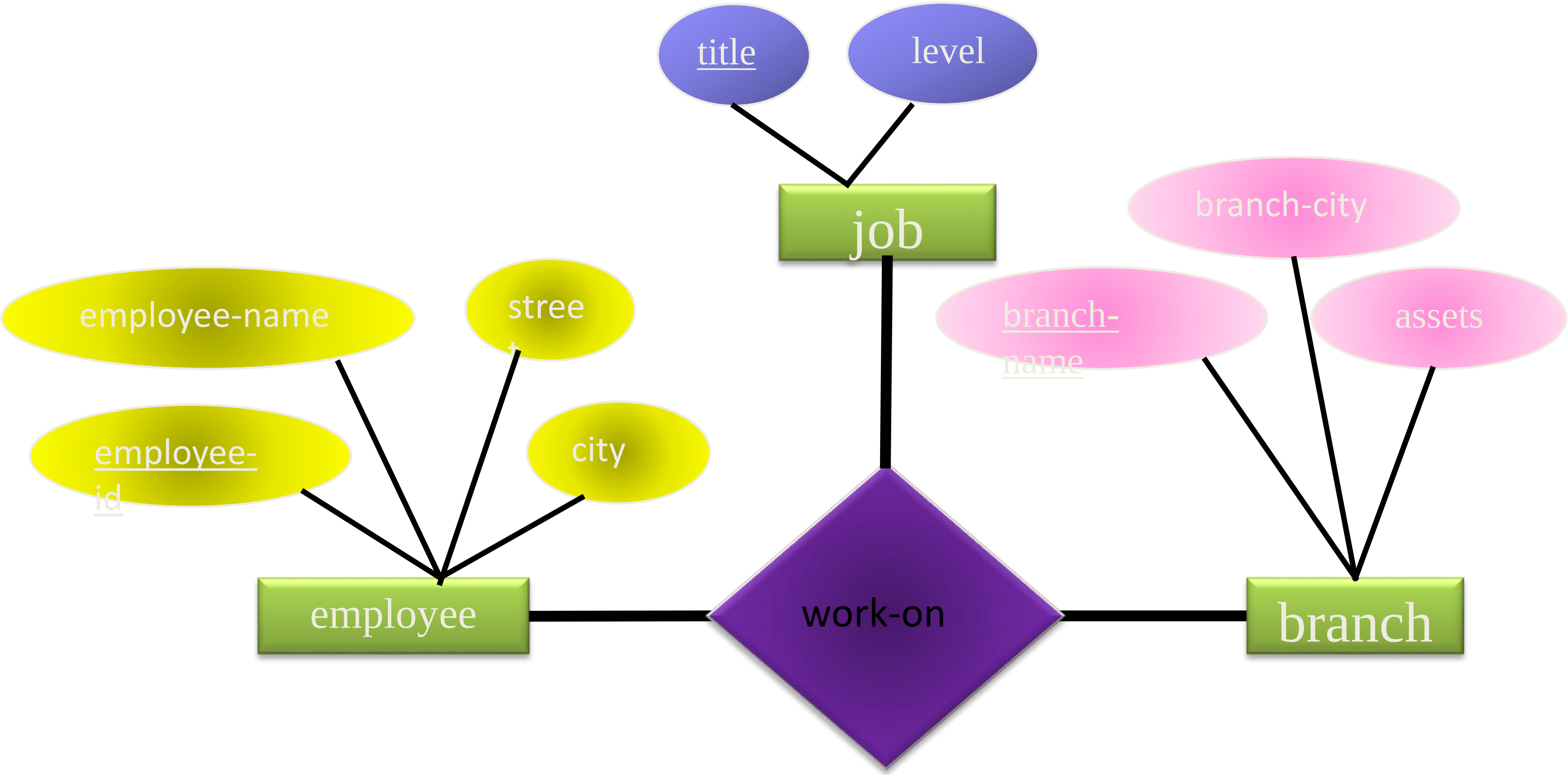


Example


Teacher_id
Teacher_name
Dept_id
Dept_name
Teacher
Teaches in
Department
Teacher_Subject



E-R diagram corresponding to customers and loans



title

level

job

branch-city

employee-name

street

branch-name

assets

employee-id

city

employee

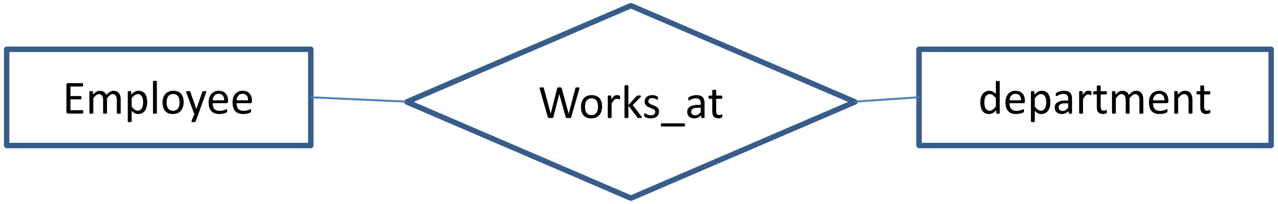
work-on

branch

Mapping Cardinality or Cardinality Ratio

- Relationship is an association among entities.

E.g. an employee **works_at** a department



Cardinality defines the number of entity of entity set participates in a relationship sets.

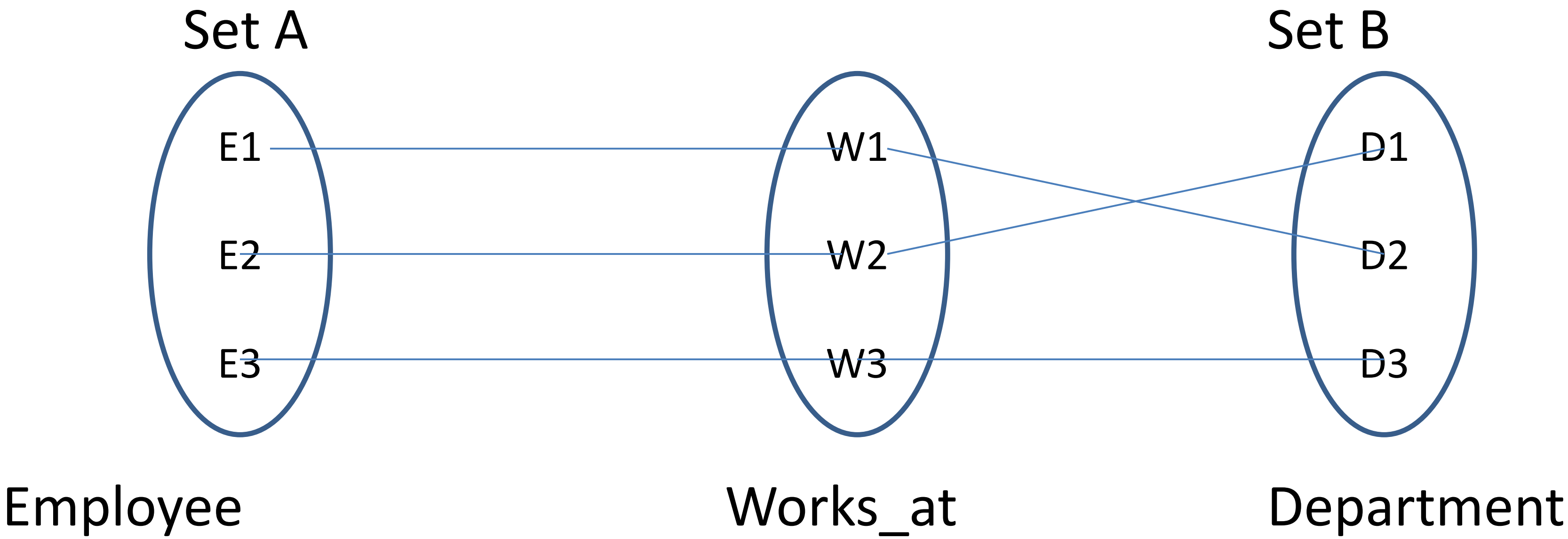
Types of Cardinality or relationships

- There are four types of relationships
- **One-to-One (1-1)**
- **One-to Many(1-M)**
- **Many-to-One(M-1)**
- **May-to-Many(M-N)**

One-to-One(1-1) relationships

- One entity from entity set A can be associated with at most one entity of entity set B and vice versa.

E.g.: Employee works at one department and a department has only one employee.



Set A

Set B

E1

E2

E3

W1

W2

W3

D1

D2

D3

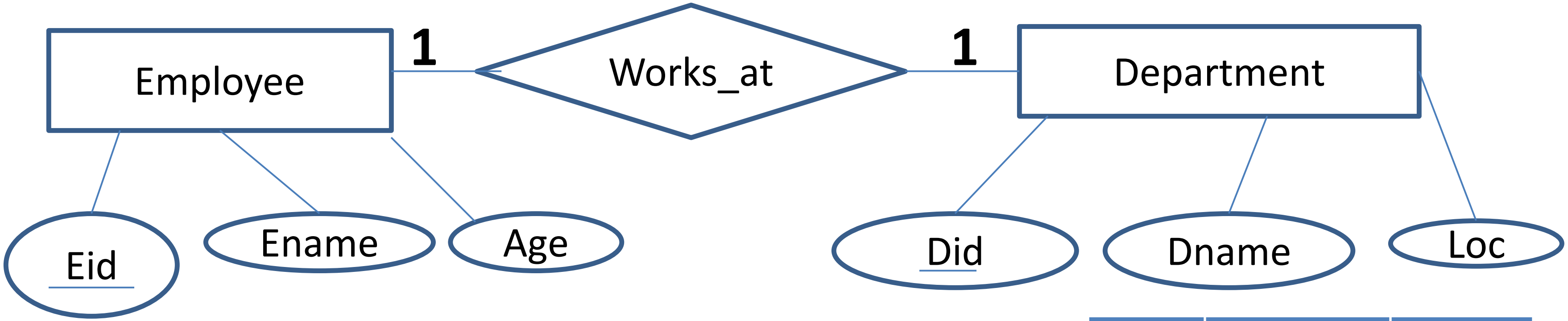
Employee

Works_at

Department

One-to-One(1-1) relationships

E-R Diagram



Eid(P.K)	Ename	Age
E1	A	25
E2	B	38
E3	C	22
E4	B	38

Eid(F.K)	Did(F.k)
E1	D2
E2	D1
E3	D3
E4	D4

Did(P.K)	Dname	Loc
D1	IT	ktm
D2	Account	pok
D3	HR	Bir
D4	Admin	Bhaira

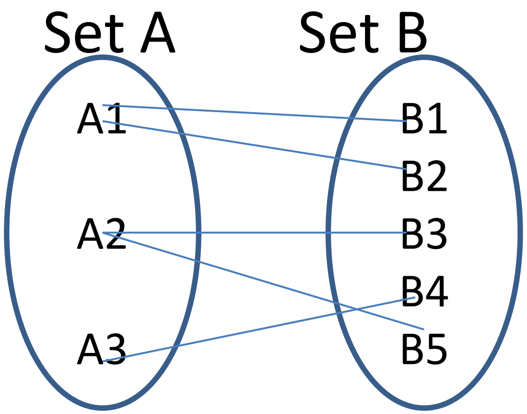
One-to-One(1-1) relationships

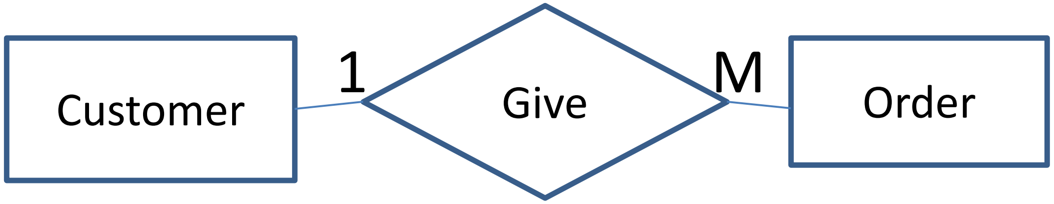
Eid(P.K)	Ename	Age	Did(F.K)
E1	A	25	D2
E2	B	38	D1
E3	C	22	D3
E4	B	38	D4

Did(P.K.)	Dname	Loc
D1	IT	ktm
D2	Account	pok
D3	HR	Bir
D4	Admin	Bhaira

One-to-Many(1-M) Relationship

One entity from entity set A can be associated with more than one entities of entity set B however an entity from entity set B, can be associated with at most one entity of entity set A.

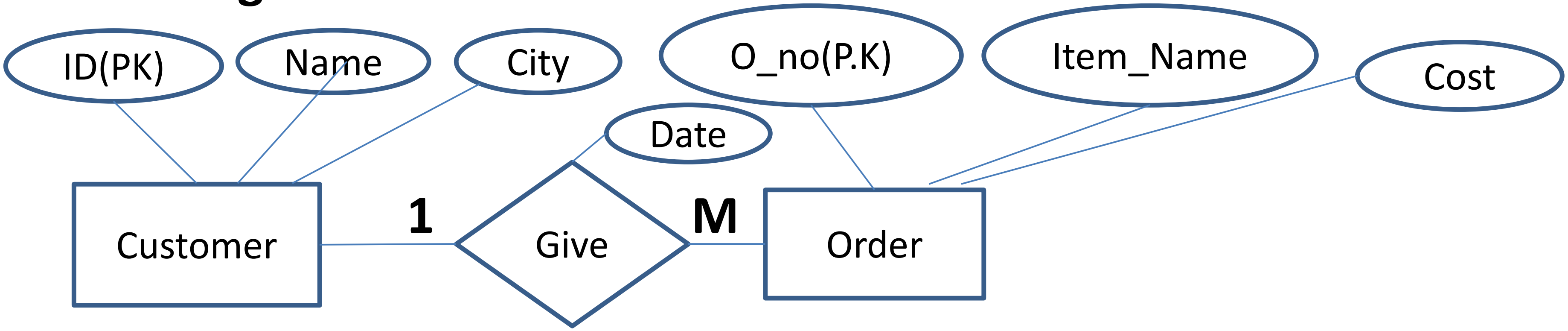




E.g.: A customer can place many orders but a order cannot be placed by many customers.

One-to-Many(1-M) Relationship

E-R Diagram

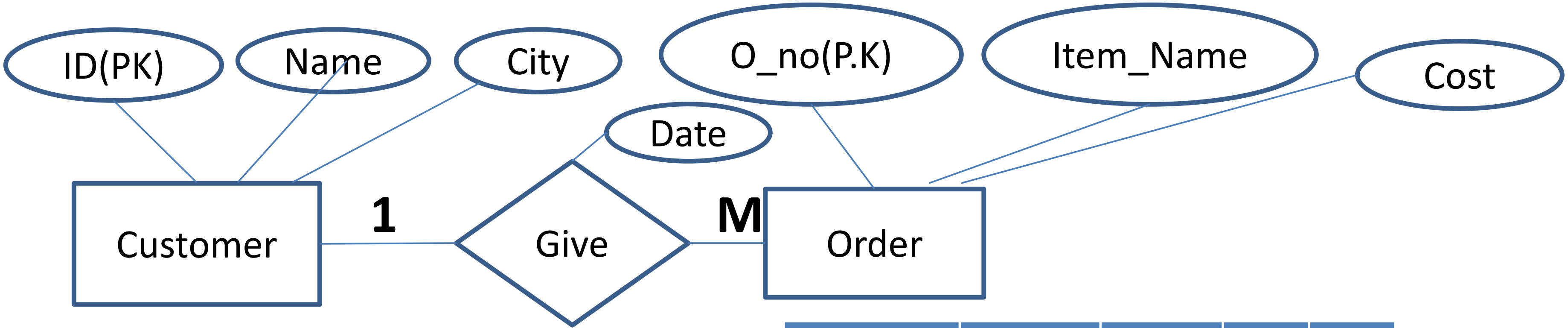


ID(PK)	Name	City
C1	A	ktm
C2	B	Brt
C3	C	Pok

ID(FK)	O_no(FK)	Date
C1	O1	
C1	O2	
C2	O3	
C2	O5	
C3	O4	

O_no(pk)	Item_name	Cost
O1	shirt	1400
O2	shoes	2300
O3	Jeans	1900
O4	Bucket	700
O5	Stove	1300

One-to-Many(1-M) Relationship



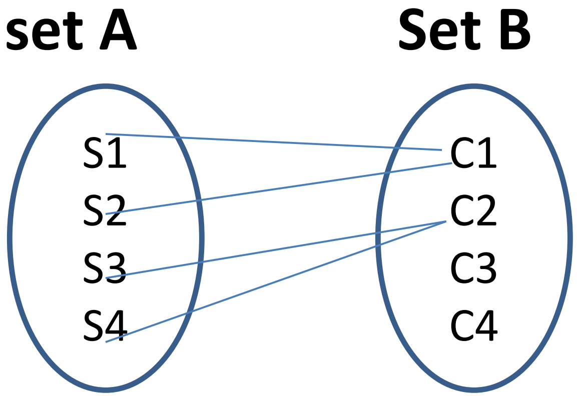
ID(PK)	Name	City
C1	A	ktm
C2	B	Brt
C3	C	Pok

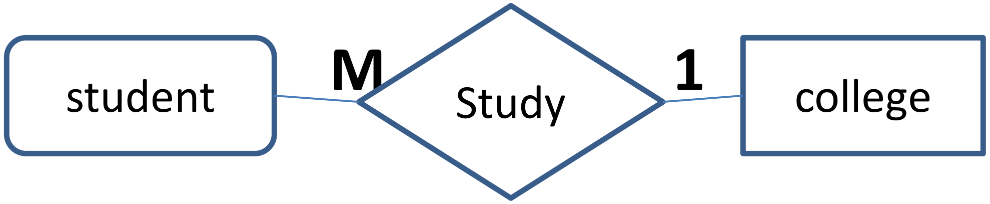
O_no(pk)	Item_name	Cost	ID(FK)	Date
O1	shirt	1400	C1	
O2	shoes	2300	C1	
O3	Jeans	1900	C2	
O4	Bucket	700	C3	
O5	Stove	1300	C2	

Many-to-One(M-1) Relationship

More than one entities from entity set A can be associated with at most one entity of entity set B, however an entity from entity set B can be associated with more than one entity from entity set A.

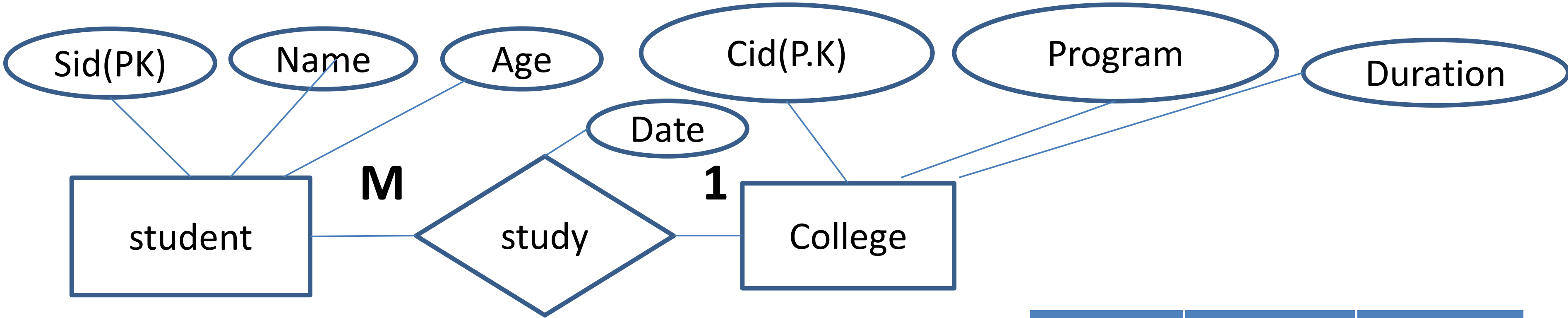
E.g.: many students can study in a single college but a student cannot study in many colleges at the same time.





Many-to-One(M-1) Relationship

E-R diagram



Sid(PK)	Name	Age
S1	A	20
S2	B	19
S3	C	18
S4	D	17
S5	E	18

Sid(FK)	Cid(FK)	Date
S1	C1	
S2	C1	
S3	C5	
S4	C5	
S5	C1	

Cid(PK)	Program	Duration(Years)
C1	IT	4
C2	BCA	3
C3	BIT	4
C4	Civil	4
C5	E&C	4

Many-to-One(M-1) Relationship

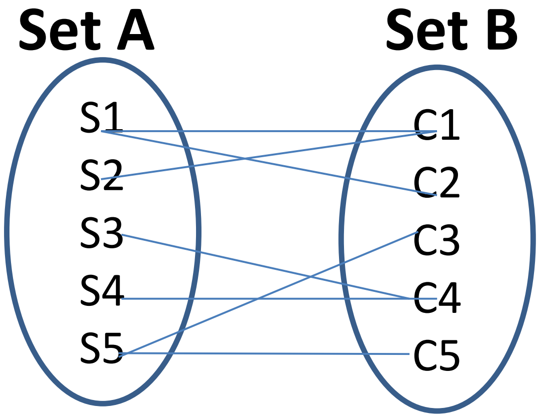
Study Relation Combined with student Relation.

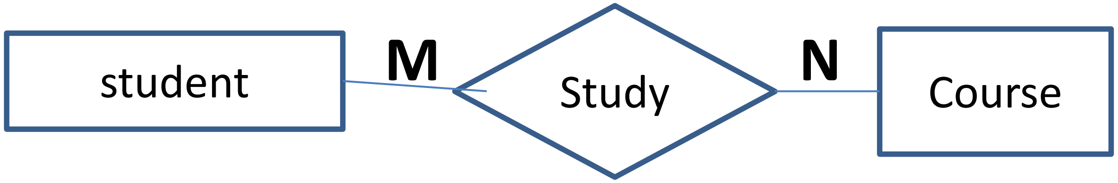
Sid(PK)	Name	Age	Cid(FK)	Date
S1	A	20	C1	
S2	B	19	C1	
S3	C	18	C5	
S4	D	17	C5	
S5	E	18	C1	

Cid(PK)	Program	Duration(Years)
C1	IT	4
C2	BCA	3
C3	BIT	4
C4	Civil	4
C5	E&C	4

Many-to-Many(M:N) Relationship

One entity from A can be associated with more than one entity from B and vice versa.

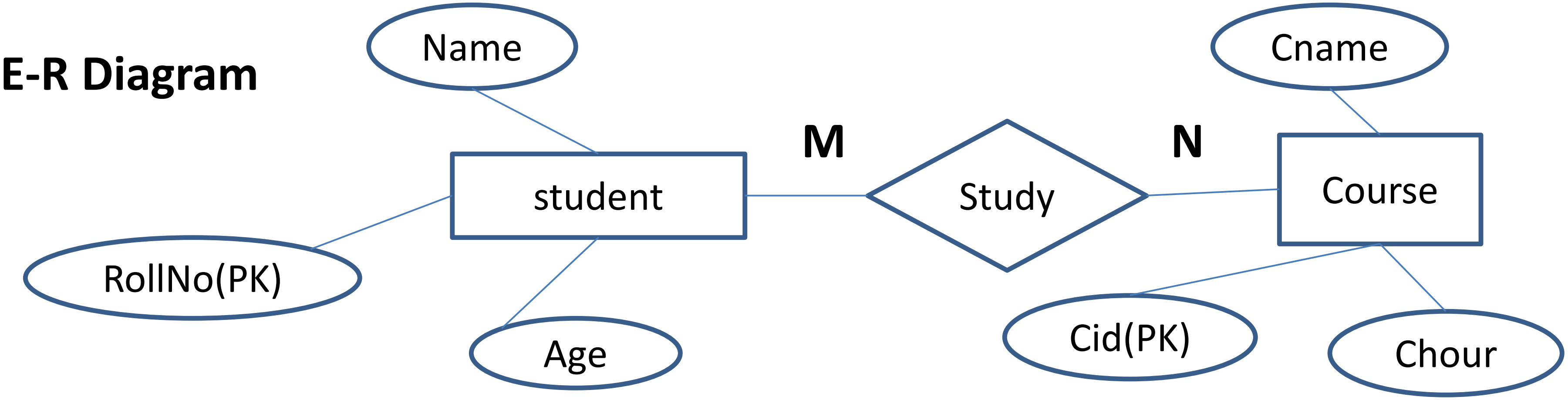




A student can study many courses and a course can be studied by many students.

Many-to-Many(M:N) Relationship

E-R Diagram



RollNo(PK)	Nam e	Age
1	A	16
2	B	17
3	C	15
4	D	19
5	A	20

RollNo(FK)	Cid(FK)
1	C1
1	C2
2	C3
3	C3
4	C1

Cid(PK)	Cname	Chour
C1	Maths	4
C2	Physics	4
C3	Chemistry	4
C4	Computer	3
C5	English	3

Many-to-Many(M:N) Relationship

Student Table

RollNo(PK)	Name	Age
1	A	16
2	B	17
3	C	15
4	D	19
5	A	20

Study Table

RollNo(FK)	Cid(FK)
1	C1
1	C2
2	C3
3	C3
4	C1

Course Table

Cid(PK)	Cname	Chour
C1	Maths	4
C2	Physics	4
C3	Chemistry	4
C4	Computer	3
C5	English	3

(RollNo+Cid) composite key will become the primary key of the Study Relation.

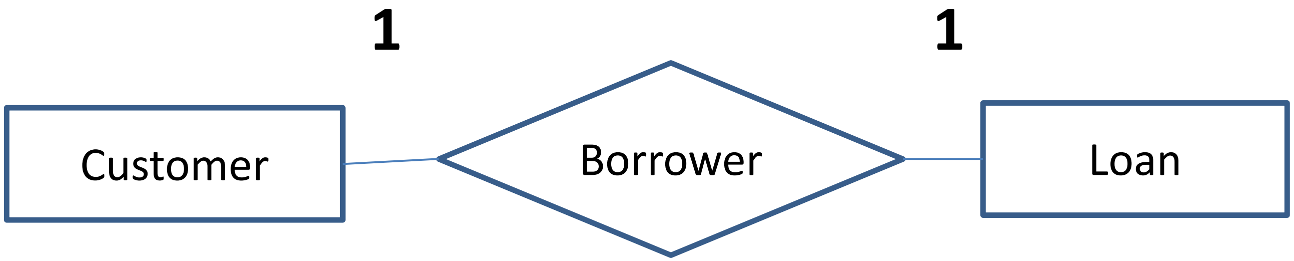
Cannot reduce the table in M-N relationship.

How to choose Relationship?

The appropriate mapping cardinality for a particular relationship set depends on the real world situation being modeled.

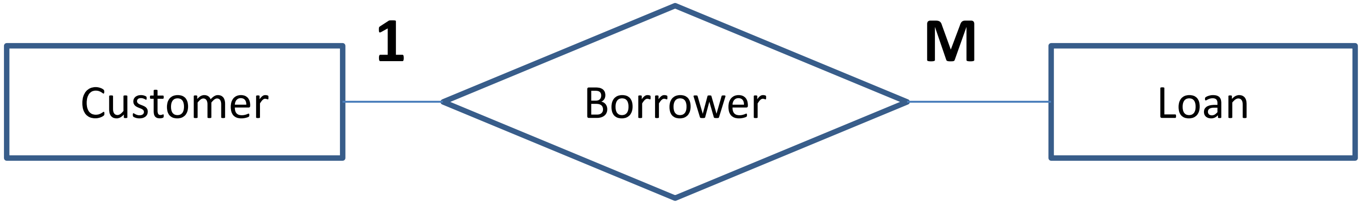
Consider **Borrower** relationship set in a bank

- If in a particular bank, a loan can belong to only one customer and a customer can have only one loan then the relationship set from customer to loan is **1-1**



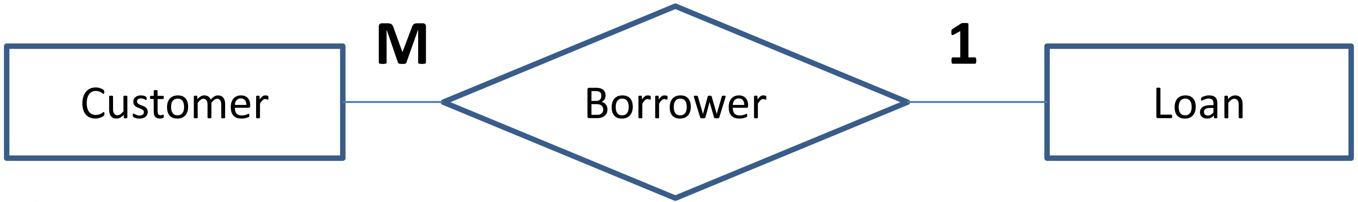
How to choose Relationship?

- If a loan can belong to only one customer and a customer can have several loans then the relationship set from customer to loan is **1-M**

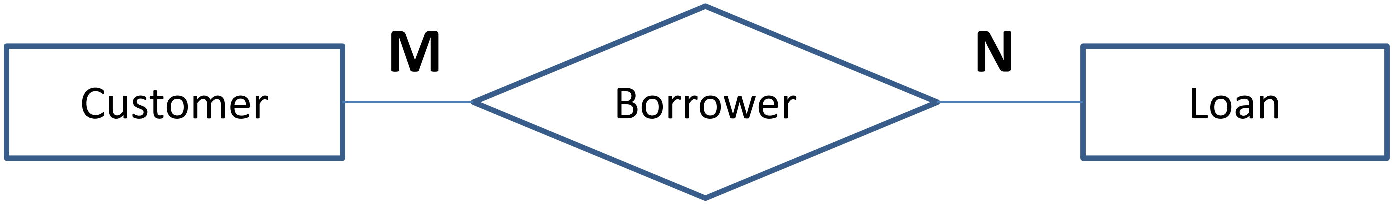


- If a loan can belong to several customers and a customer can have only one loan then the relationship set from customer to loan is **M-1**

How to choose Relationship?



- if a loan can belong to several customers and a customer can have several loans then the relationship set is **M-N**



Participation Constraints

Participation constraint is applied on the entity participating in the relationship set.

- **Total Participation**
- **Partial Participation**
- **Total Participation:** each entity is involved in the relationship. Total Participation is represented by **double lines.**

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

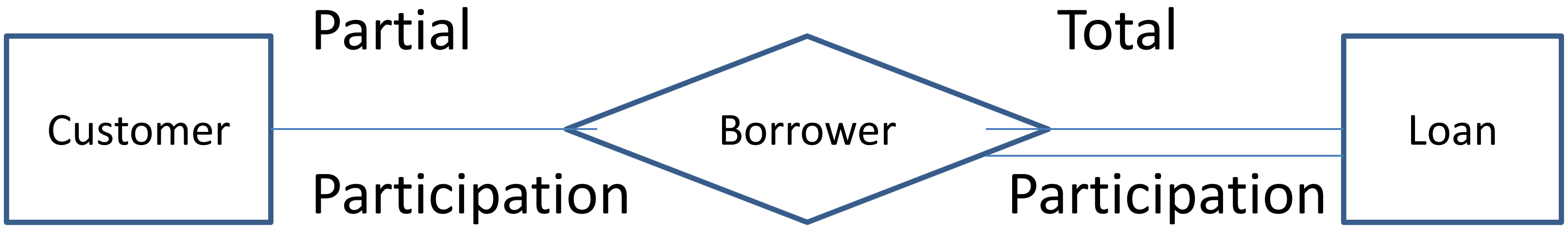
Every loan must have a customer associated to it via borrower.

- **Partial Participation:** Not all entities are involved in the relationship. Partial Participation is represented by **single line.**

Participation of customer in borrower is partial.

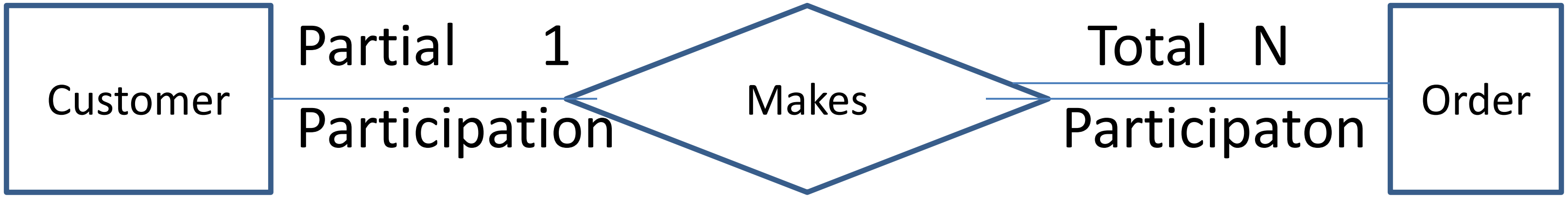
A customer may have no loans.

Participation Constraints



Partial Participation exists when all the entity of an entity type is not associated with one or the other entity of another entity type

E.g. we have two entity type ‘Customer’ and ‘Order’. Then there can be ‘Customer’ who have not done any order. So, here there is partial participation of the entity in the relationship.

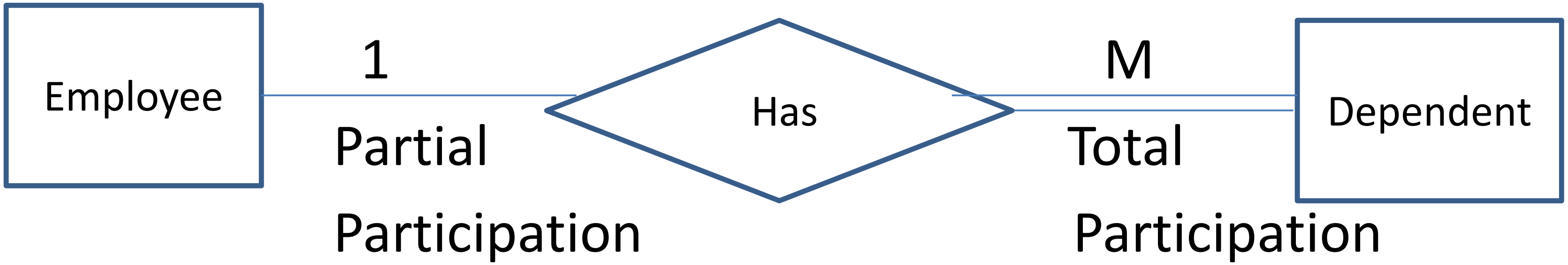


Participation Constraints

Total Participation exists when all the entity of an entity type is associated with one or the other entity of another entity type. Such a relationship usually exists between **strong entity** and **weak entity**.

E.g. We have two entities ‘**Employee’** and **‘ Dependent’.** Then all the **‘Dependent’** entity are related to one or the other **‘Employee’** entity. This is called **total participation** of the entity in the relation. But, it may be possible that some **‘Employee’** is not related to any of the **‘Dependent’** entity. So, ‘**Employee’** is showing **partial participation** whereas the **‘Dependent’** is showing **total participation** in the relation.

Participation Constraints



Employee

1

Partial Participation

Has

M

Total Participation

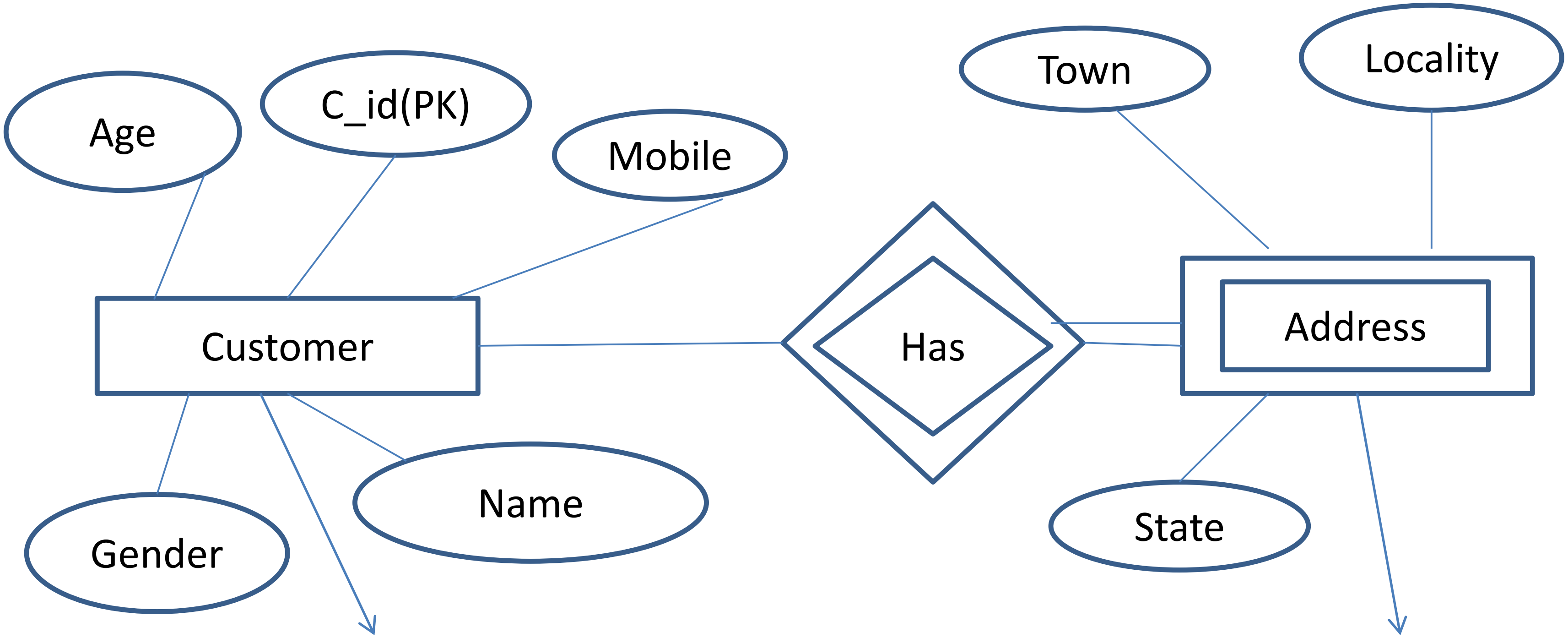
Dependent

Weak Entity Type

Weak entity type doesn't have a key attribute. Weak entity type can't be identified on its own. It depends upon some other strong entity for its distinct identity. This can be understood with a real-life example. There can be children only if the parent exits. There can be no independent existence of children. There can be a room only if building exits. There can be no independent existence of a room. A weak entity is represented by a double outlined rectangle. The relationship between a weak entity type and strong entity type is called an **identifying relationship** and shown with a double outlined diamond instead of a single outlined diamond. This representation can be seen in the diagram below.

Example: If we have two tables of Customer(C_id, Name, Mobile, Age, Gender) and Address(Locality, Town, State). Here we cannot identify the address uniquely as there can be many customers from the same locality. So, for this, we need an attribute of Strong Entity Type i.e ‘Customer’ here to uniquely identify entities of 'Address' Entity Type.

Entity, Entity Type, and Entity Set



Strong Entity Type

Weak Entity Type

Difference between Strong Entity and Weak Entigy

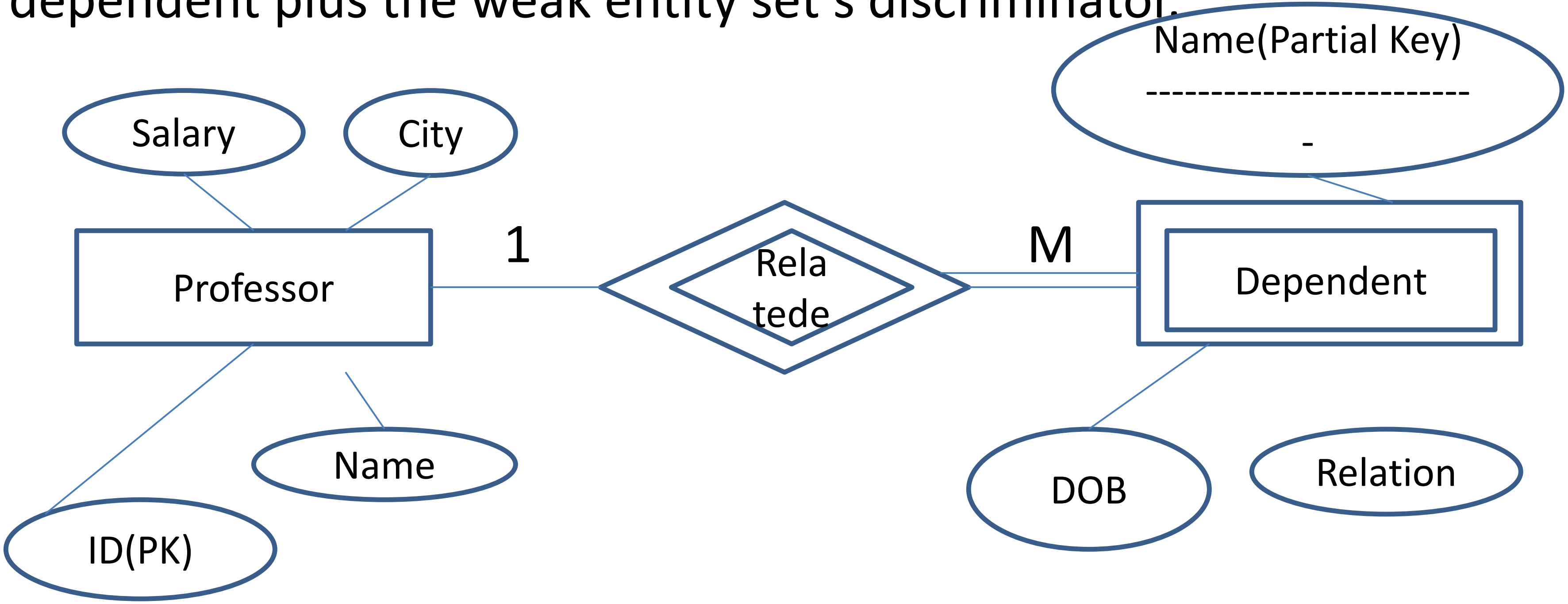
Strong Entity

- Strong Entity always has a **primary key** .
- Strong Entity is not dependent on any other entity.
- Strong Entity is represented by single rectangle.
- Two strong entity’s relationship is represented by single diamond.
- Strong Entity have either total participation or not.

Weak Entity

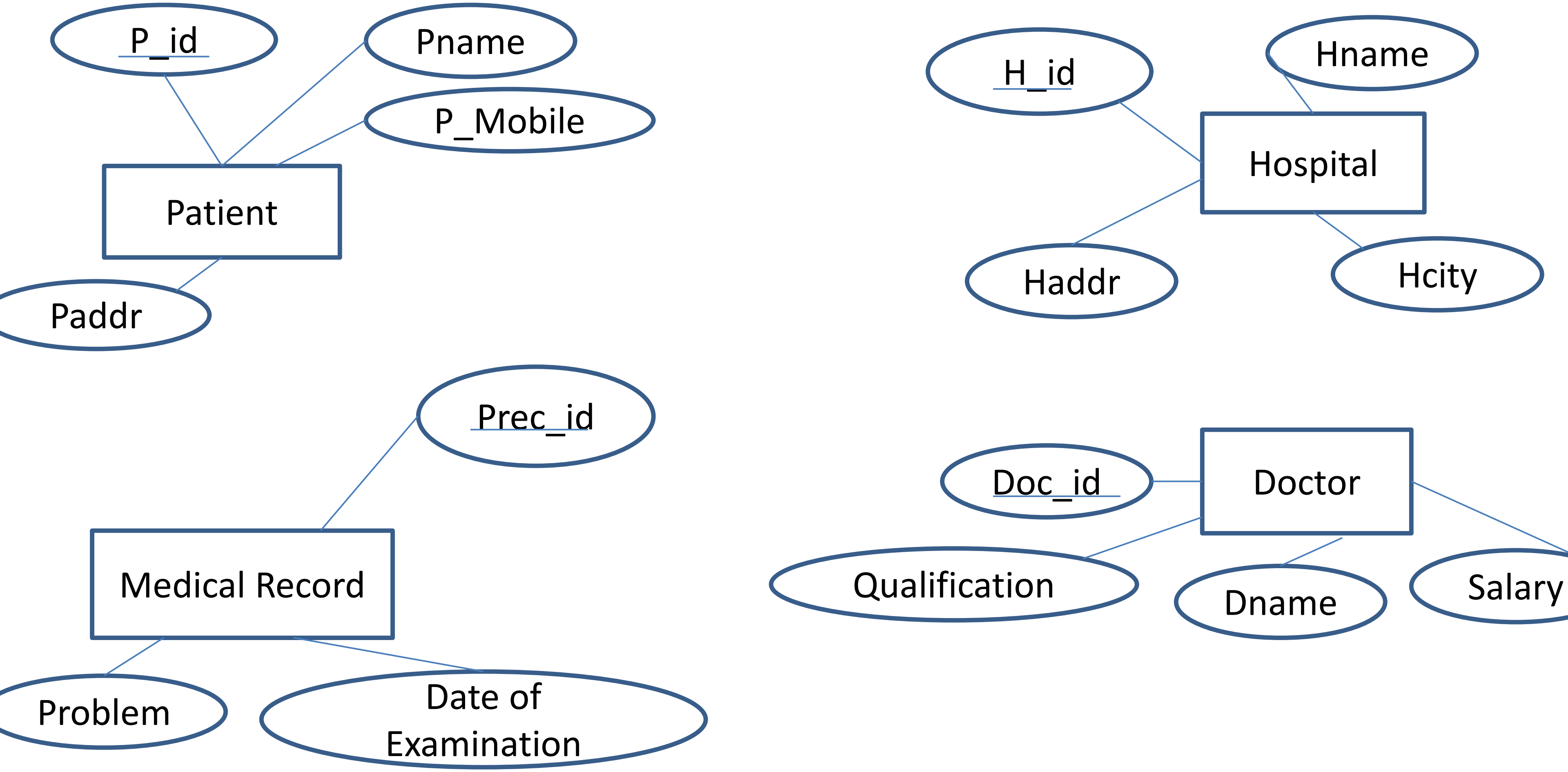
- While Weak Entity has **Partial key** or **Discriminator key**
- Weak Entity depends on strong entity.
- Weak Entity is represented by double rectangle.
- While the relation between one strong and one weak entity is represented by double diamond.
- While Weak Entity always have total participation.

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



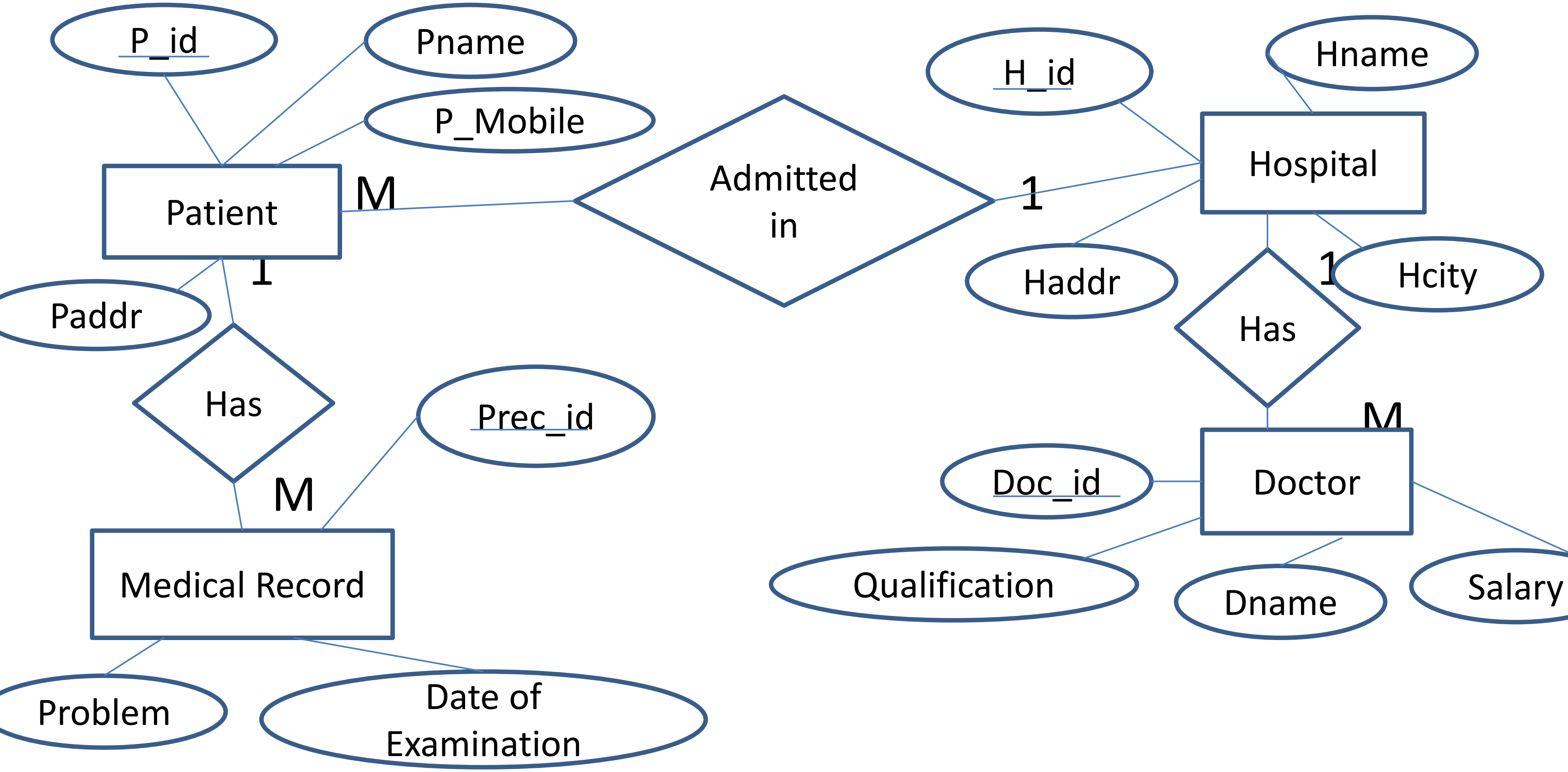
Q. Construct an E-R Diagram for Hospital Management System. Convert this diagram into Relation Schema

Step 2: Identify the attributes and primary key



Q. Construct an E-R Diagram for Hospital Management System. Convert this diagram into Relation Schema

Step 3: Identify the Relationship and Participation Constraints.



Relation Schema of Hospital Management System

Schemas

Patient(P_id(PK), Pname, P_mobile, P_addr, H-id(FK))

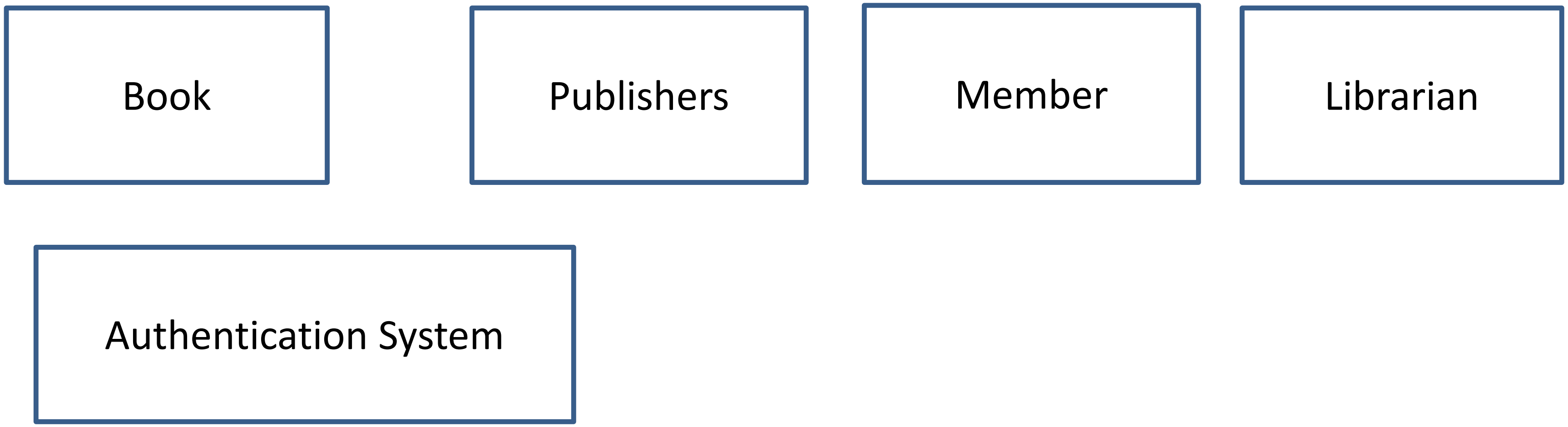
MedicalRecord(Prec_id(PK), DateofExamination, Problem, P_id(FK))

Doctor(D_id(PK), Dname, Salary, Qualification, H_id(FK))

Hospital(H_id(PK), Hname, Haddr, Hcity)

Q. Construct an E-R Diagram for Library Management System. Showing the proper relationship Book, Publishers, Member and Librarian. Convert this diagram into Relation Schema

Step 1: Identify the Entities.



Book

Publishers

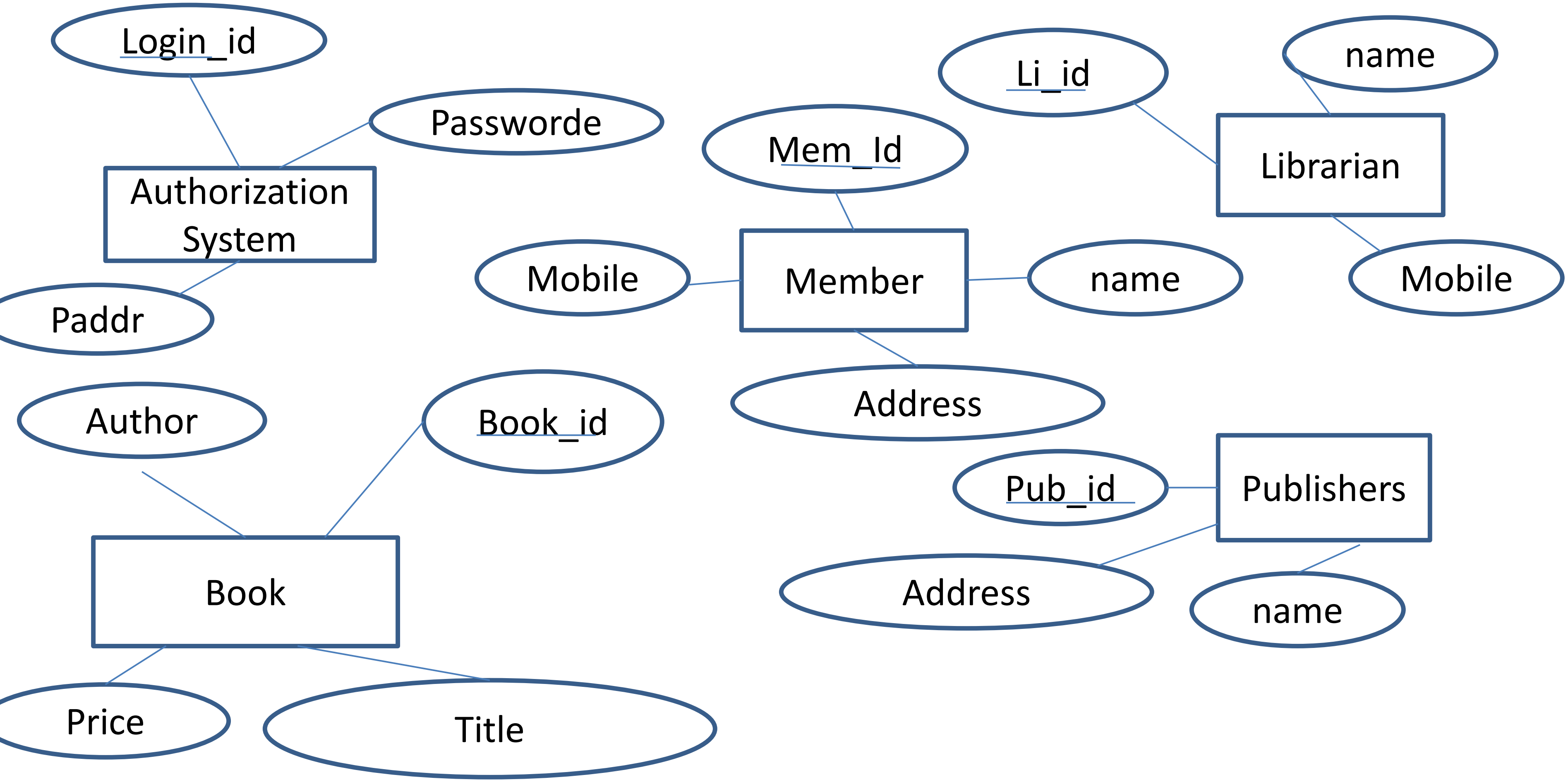
Member

Librarian

Authentication System

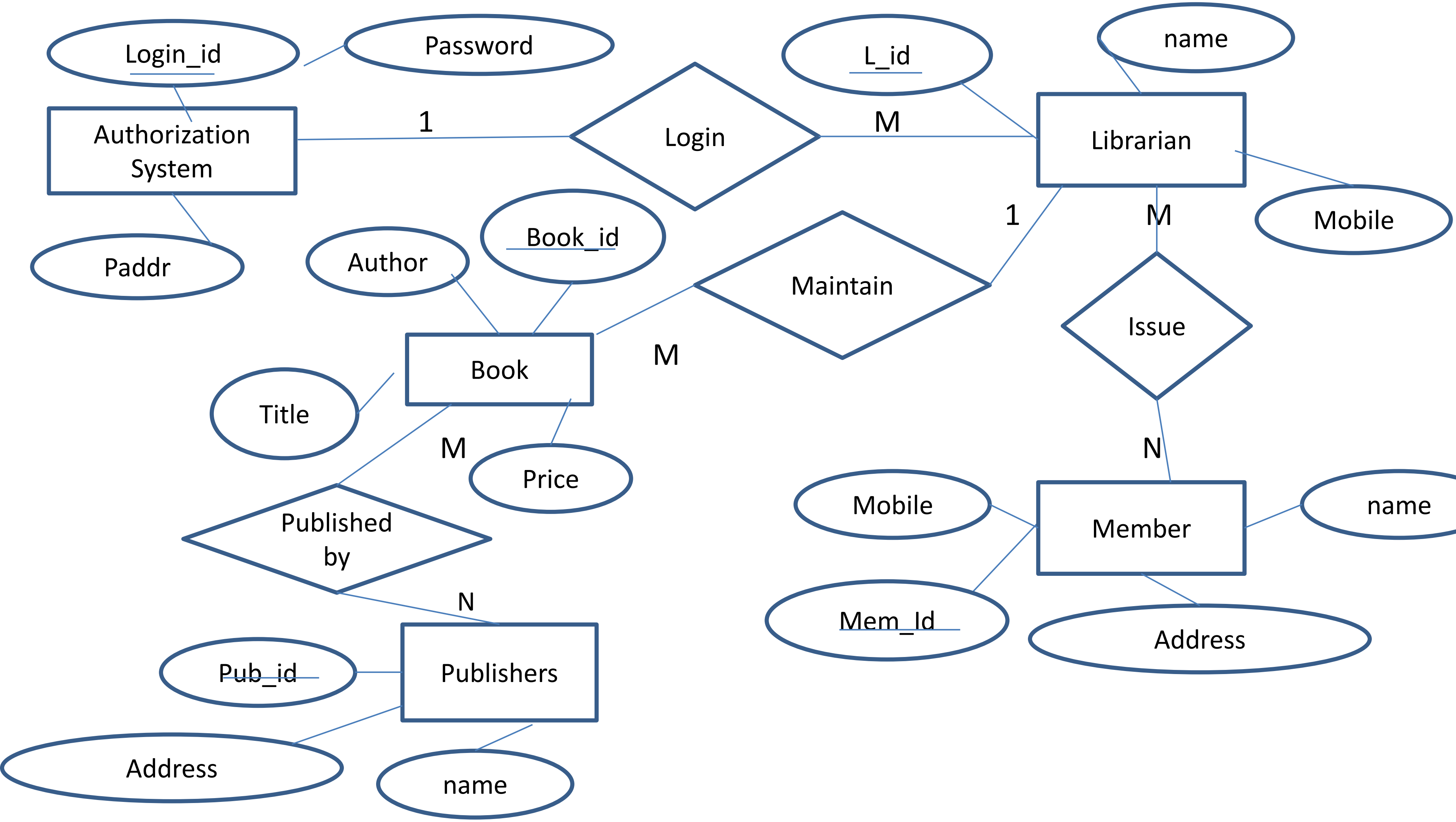
Q. Construct an E-R Diagram for Library Management System. Showing the proper relationship Book, Publishers, Member and Librarian. Convert this diagram into Relation Schema

Step 2: Identify the attributes and primary key



Q. Construct an E-R Diagram for Library Management System. Showing the proper relationship Book, Publishers, Member and Librarian. Convert this diagram into Relation Schema

Step 3: Identify the Relationship an participation Constraints



Relation Schema of Library Management System

Schemas

Authorization System(Login_id(PK), Password)

Librarian(L-id(PK), name, mobile, Login_id(FK))

Book(book_id(PK), Title, Author, Price, L_id(FK), Mem_id(FK))

Members(Mem_id(PK), name, address, mobile)

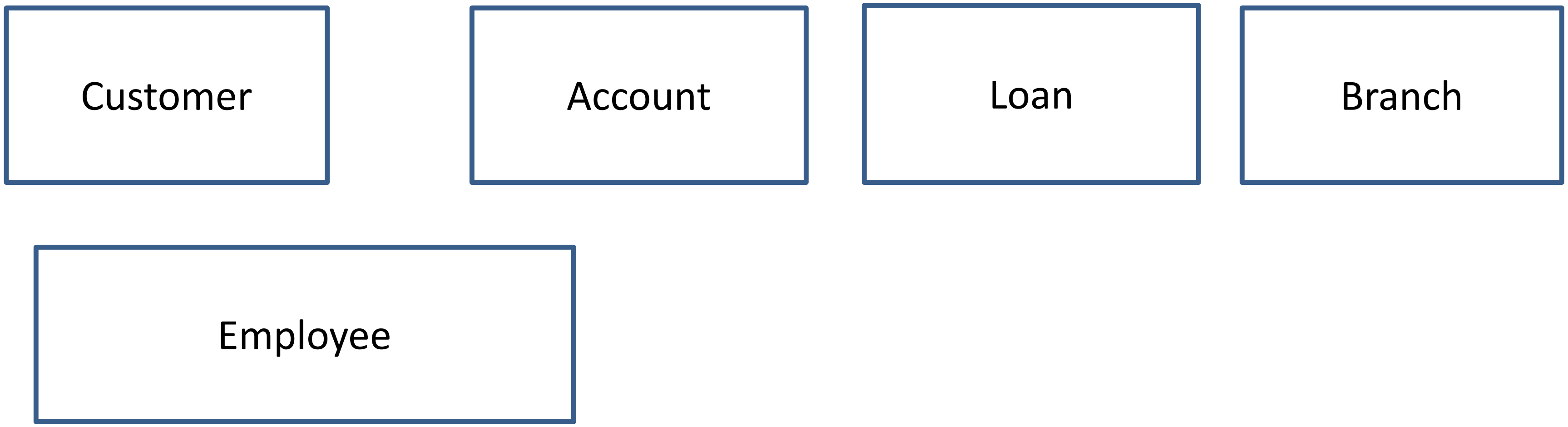
Published by(book_id(FK), Pub_id(FK)

Publishers(Pub_id (PK) , Address, Name)

Issue(L_id(FK), Mem_id(FK))

Q. Construct an E-R Diagram for Banking System. Convert this diagram into Relation Schema

Step 1: Identify the Entities.



Customer

Account

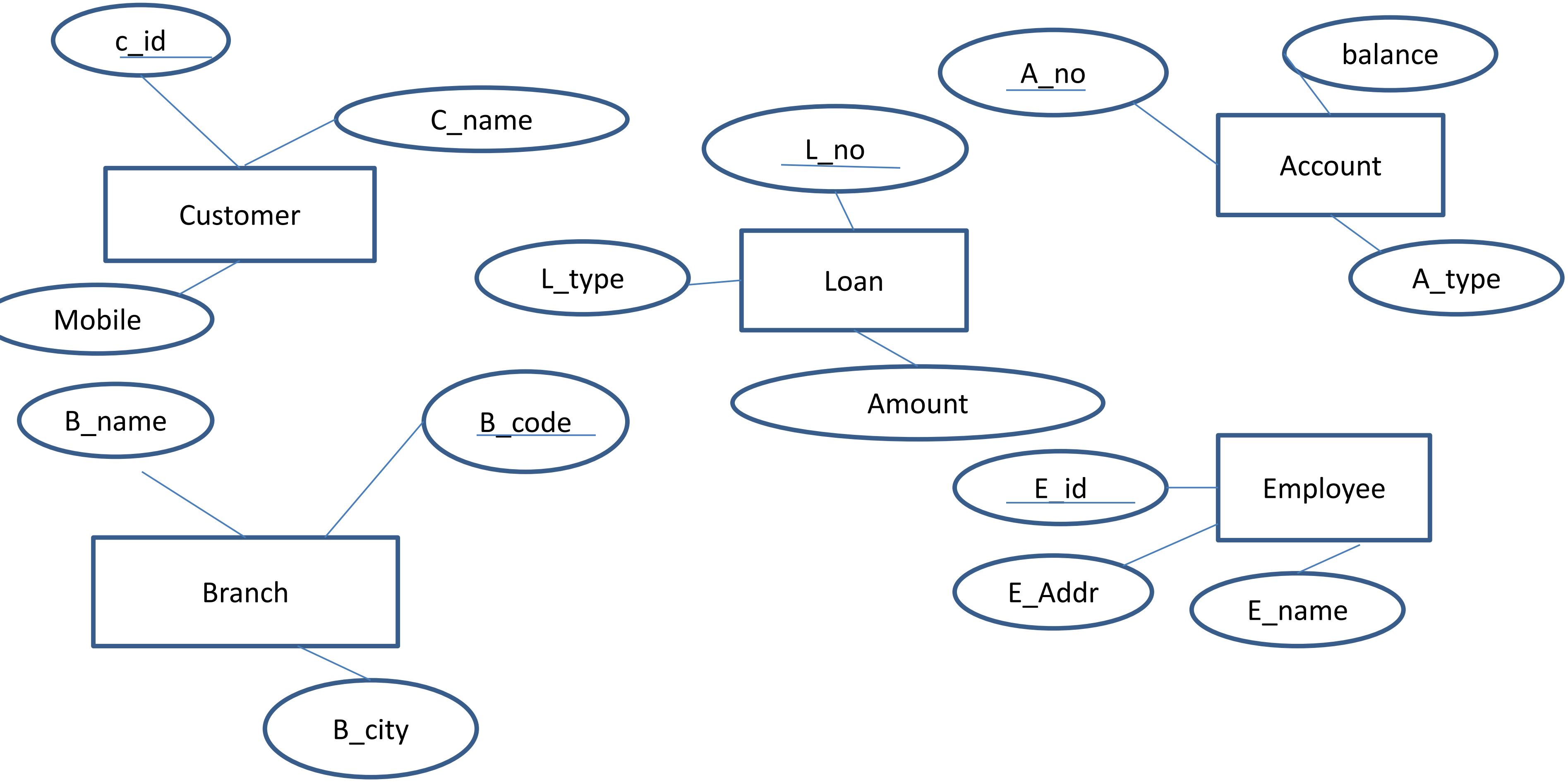
Loan

Branch

Employee

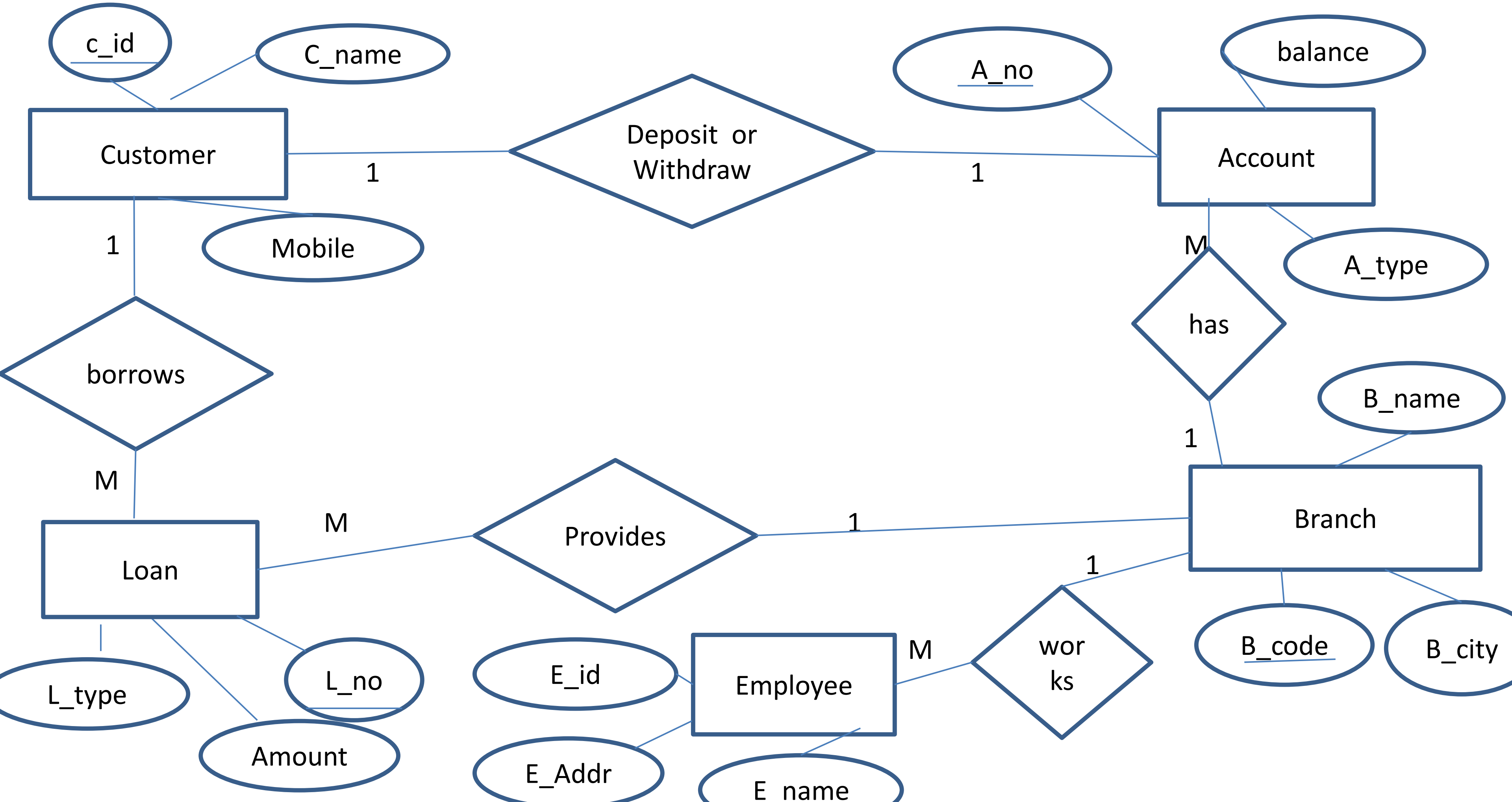
Q. Construct an E-R Diagram for Banking System. Convert this diagram into Relation Schema

Step 2: Identify the attributes and primary key



Q. Construct an E-R Diagram for Banking System. Convert this diagram into Relation Schema

Step 2: Identify the Relationship and Participation Constraints.



Relation Schema of Banking System

Schemas

Customer(C_id(PK) , C_name, mobile, A_no(FK))

Loan (L_no(PK), Amount, L_type, C_id(FK), B_code(FK))

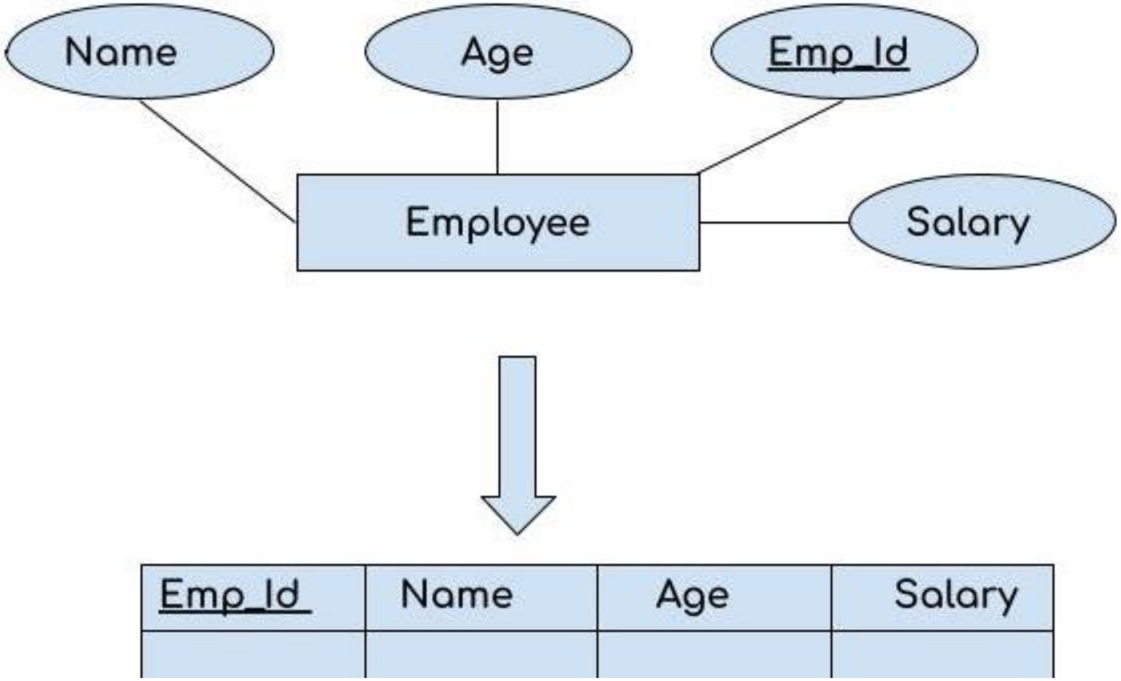
Account (A_no(PK), balance, A_type, B_code(FK))

Works(E_id(PK), E_name, E_addr, B_code(FK))

Branch(B_code(PK), B_city, B_name)

Conversion from ER diagram to relational Model

- ✓ Rule 1: Entity type becomes a table
- ✓ Rule 2: All single-valued attribute becomes a column for the table
- ✓ Rule 3: A key attribute of the entity type represented by the primary key.
- ✓ Rule 4: The multivalued attribute is represented by a separate table
- ✓ Rule 5: Composite attribute represented by components
- ✓ Rule 6: Derived attributes are not considered in the table



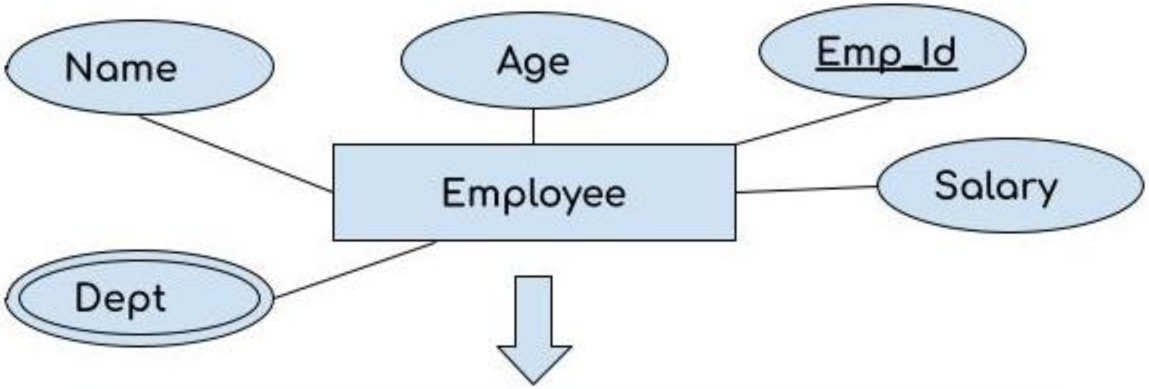
Emp_Id	Name	Age	Salary

Example 2: With composite attribute

The entity set will be the table and the simple attributes of the composite attributes will become the attributes of the table while the composite attribute itself will be ignored during conversion.

Example 3: With Multi Valued Attributes

Entity set with multi-valued attributes will require two tables in the relational model. Whenever we have a multi-valued attribute, there needs to be **more than one table** to represent the ER diagram.



Emp_Id	First_N	Last_N	Age	Salary

+

Emp_Id	Dept

Rule 7: For binary relationship with 1:1 cardinality

Add primary key attribute of the entity on any side of the relationship as a foreign key in the relation on other side.

Rule 8: For binary relationship with N: 1 cardinality

Add the Primary key attribute of the entity on the one side of the relationship as a foreign key in the relation on the other (N) side

Rule 9: For binary relationship with M: N cardinality

Create another relation (table) and include primary keys of all relations as primary key of new relation

Thank you.