Database Management System

Subject Code: 3130703

Prof. Amit Vyas
Department of Computer Engineering
V.V.P. Engineering College

राष्ट्राय स्वाहा इदं न मम ।

Teaching and Examination Scheme

Teaching Schemes

- ✓ Lecture (L) – 4 , Tutorial (T) – 0, Practical (P) - 2 = Credit – 5
- ✓ Total Marks of the Subject: 150

Examination Scheme

Theory Marks:

- ✓ ESE(E) – 70 (University Level Exam) (End Semester Examination)
- ✓ PA(M) – 30 (College Level Exam) (Progressive Assessment)

Practical Marks:

- ✓ ESE(V) – 30 (End Semester Examination (College level))
- ✓ PA(I) – 20 (Progressive Assessment (College Level))

Reference Books

1. "Database System Concepts", 6th Edition by Abraham Silberschatz, Henry F. Korth, S. Sudarshan, McGraw-Hill.
2. "Fundamentals of Database Systems", 7th Edition by R. Elmasri and S. Navathe, Pearson
3. "An introduction to Database Systems", C J Date, Pearson.

Cont….

4. “Modern Database Management”, Hoffer , Ramesh, Topi, Pearson.
5. “Principles of Database and Knowledge – Base Systems”, Vol 1 by J. D. Ullman, Computer Science Press.
6. “Understanding SQL”, Martin Gruber, BPB
7. SQL,PL/SQL – The Programing language of Oracle

List of Open Source learning website:

1. https://www.tutorialspoint.com/dbms/
2. https://www.w3schools.com/sql/
3. https://www.codecademy.com/learn/learn-sql

List of Experiments

1. To study DDL-create and DML-insert commands

2. Create table and insert sample data in tables.

3. Perform queries involving predicates LIKE, BETWEEN, IN etc.

4. To Perform various data manipulation commands, aggregate functions and sorting concept on all created tables.

Cont….

5.To study Single-row functions.

6.Displaying data from Multiple Tables (join)

7.To apply the concept of Aggregating Data using Group functions.

8. To solve queries using the concept of sub query.

Cont….

9. To apply the concept of security and privileges

10. To study Transaction control commands

11. Write Cursor

12. Write Trigger

Course Outcomes (Learning Outcomes)

CO_1.-Identify the basic concepts and various data model used in database design ER modeling concepts and architecture use and design queries using SQL

CO_2.-Apply relational database theory and be able to describe relational algebra expression, tuple and domain relation expression fro queries.

Cont….

CO_3.-Recognize and identify the use of normalization and functional dependency, indexing and hashing technique used in database design.

CO_4.-Recognize/ identify the purpose of query processing and optimization and also demonstrate the basic of query evaluation.

Cont….

CO_5.-Apply and relate the concept of transaction, concurrency control and recovery in database.

CO_6.-Discuss recovery system and be familiar with introduction to web database, distribute databases, data warehousing and mining.

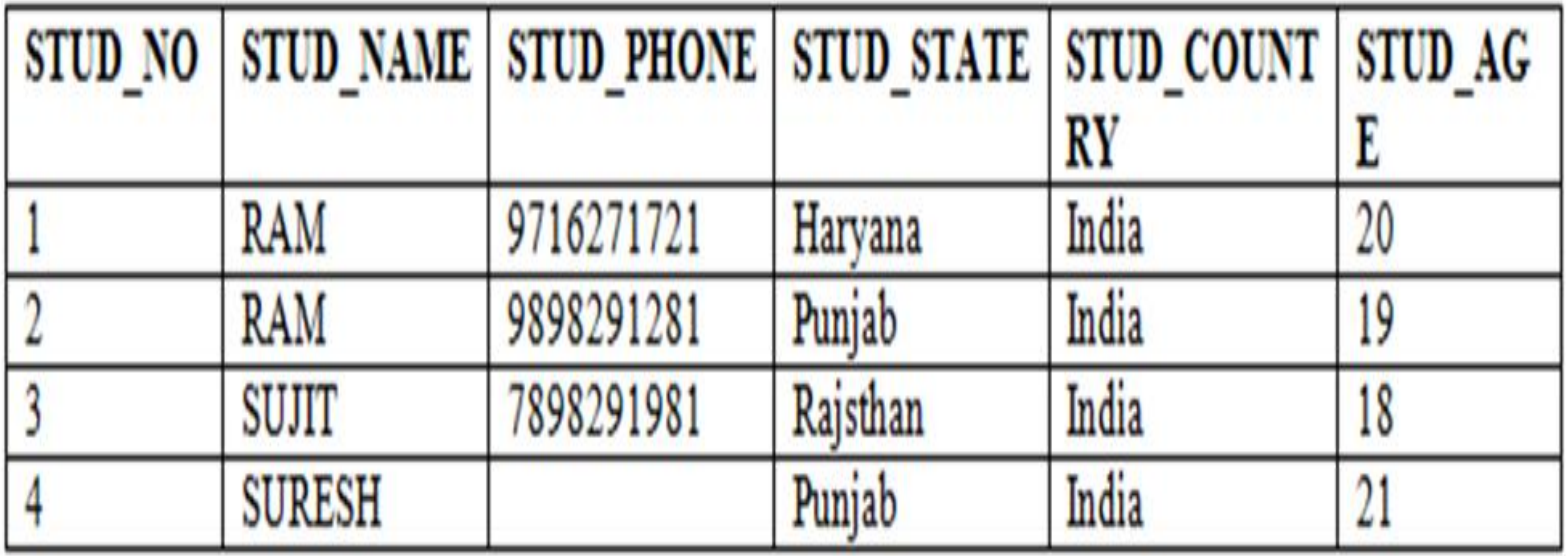
DBMS Keys

Primary Key:

A primary key is used to ensure data in the specific column is unique. It is a column cannot have NULL values. It is either an existing table column or a column that is specifically generated by the database according to a defined sequence.

- **Example:** STUD_NO, as well as STUD_PHONE both, are candidate keys for relation STUDENT but STUD_NO can be chosen as the primary key (only one out of many candidate keys).

STUDENT



STUD_NO	STUD_NAME	STUD_PHONE	STUD_STATE	STUD_COUNTRY	STUD_AGE
1	RAM	9716271721	Haryana	India	20
2	RAM	9898291281	Punjab	India	19
3	SUJIT	7898291981	Rajsthan	India	18
4	SURESH		Punjab	India	21

Table 1

Foreign Key:

A foreign key is a column or group of columns in a relational database table that provides a link between data in two tables. It is a column (or columns) that references a column (most often the primary key) of another table.

Example:

STUD_NO in STUDENT_COURSE is a foreign key to STUD_NO in STUDENT relation.

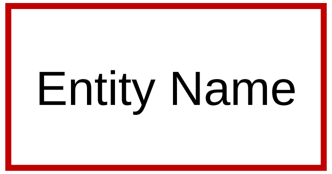
Stud no	COURSE_NO	COURSE_NAME
1	C1	DBMS
2	C2	DS
3	C3	PPS
4	C4	PPS

Sr.NO.	PRIMARY KEY	FOREIGN KEY
1	A primary key is used to ensure data in the specific column is unique.	A foreign key is a column or group of columns in a relational database table that provides a link between data in two tables.
2	It uniquely identifies a record in the relational database table.	It refers to the field in a table which is the primary key of another table.
3	**Only one primary key is allowed in a table.**	Whereas more than one foreign key are allowed in a table.
4	It is a combination of UNIQUE and Not Null constraints.	It can contain duplicate values and a table in a relational database.
5	It does not allow NULL values.	It can also contain NULL values.
6	Its value cannot be deleted from the parent table.	Its value can be deleted from the child table.
7	It constraint can be implicitly defined on the temporary tables.	It constraint cannot be defined on the local or global temporary tables.

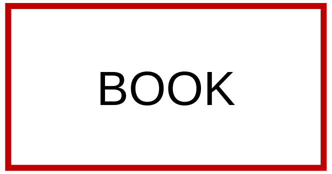
Entity

- An entity is thing in real world with its own independent existence.

- E.g. book, Faculty



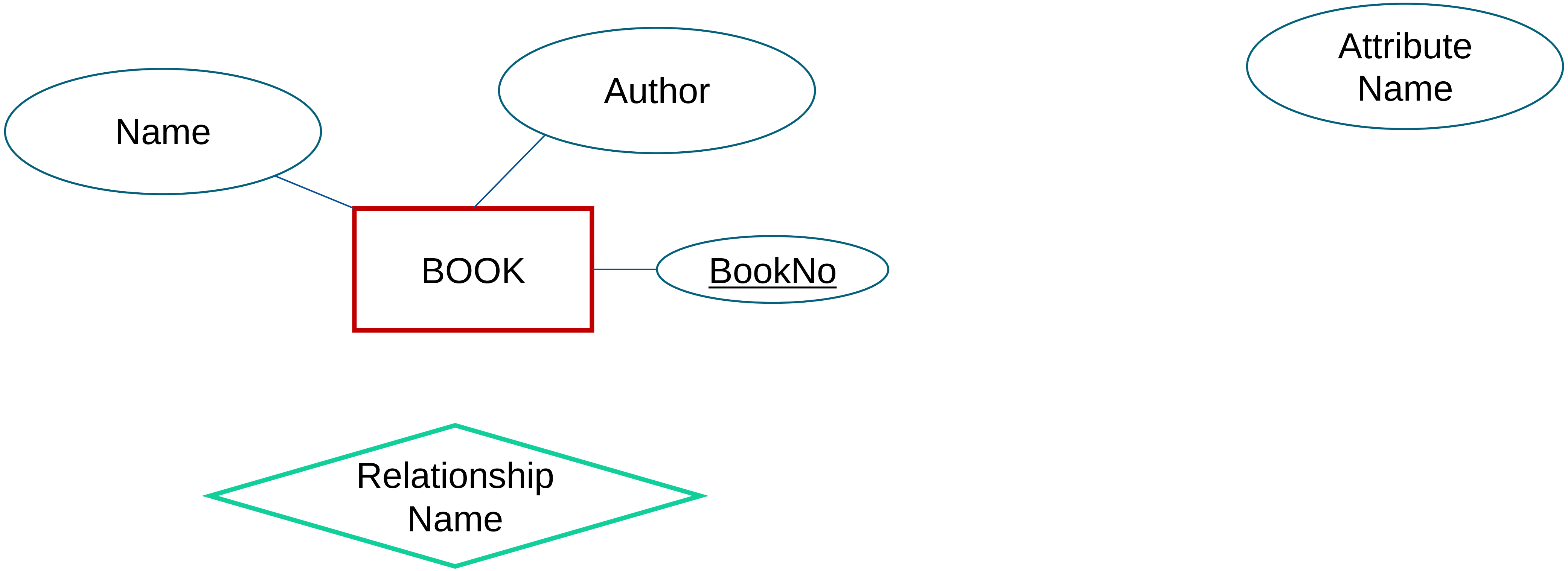
Entity Name



BOOK

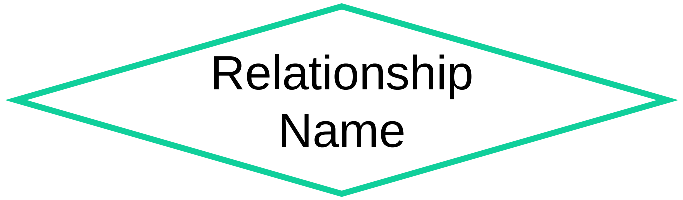
Attributes

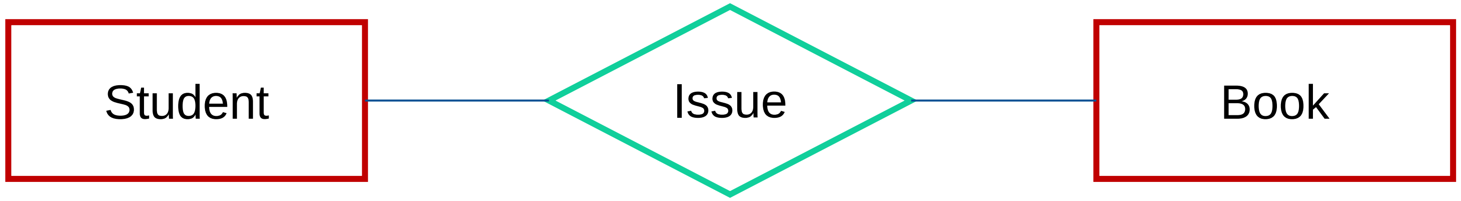
- Attribute is **properties** or details about an entity.



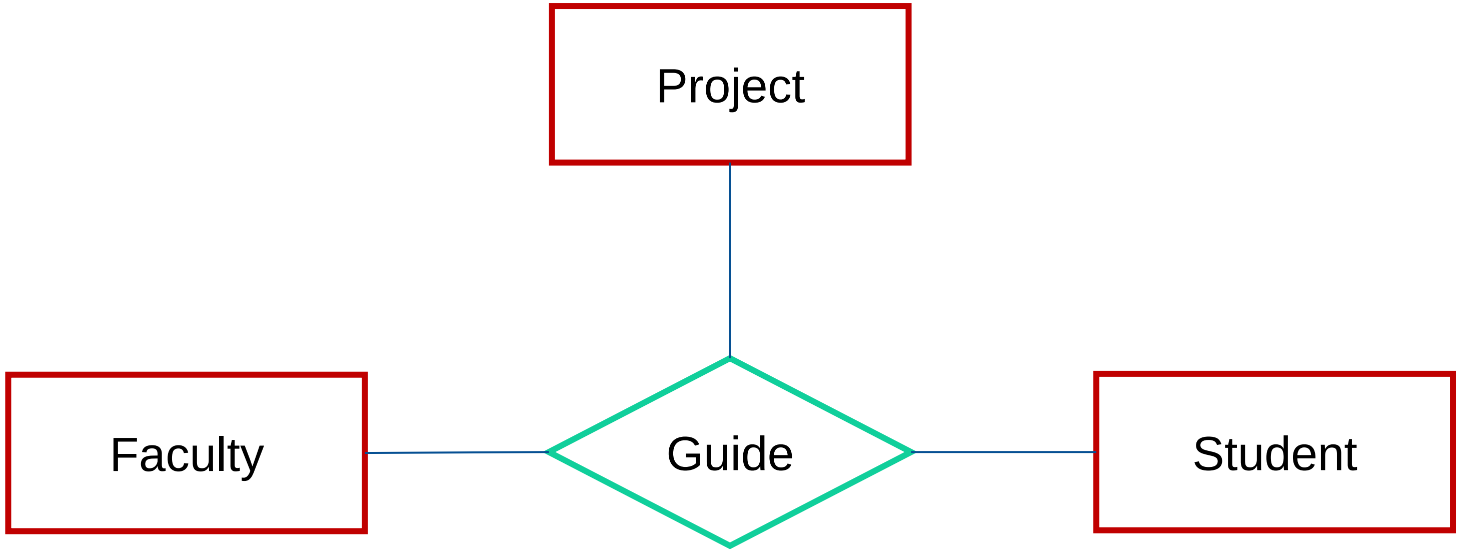
Relationship

- It should be placed between two entities and a line connecting it to an entity.





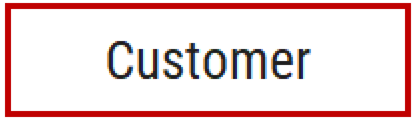
Ternary Relationship



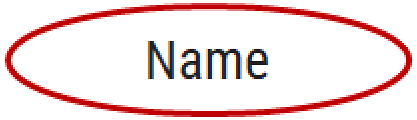
Relationship between 3 entities is called ternary relationship.

Exercise

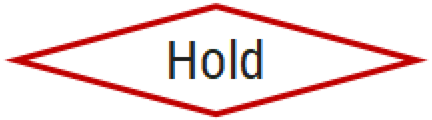
- Draw an E-R diagram of Library Management System.
- Draw an E-R diagram of Banking Management System.



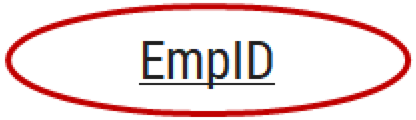
Entity



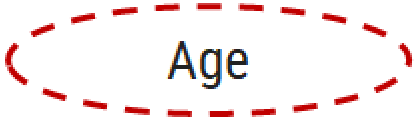
Attribute



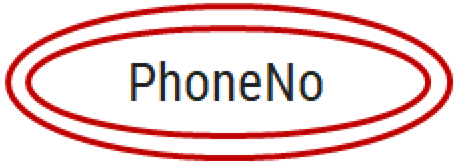
Relationship



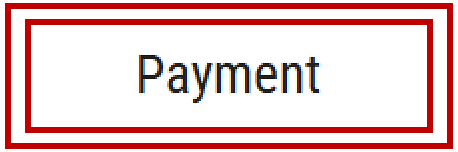
Primary Key
Attribute



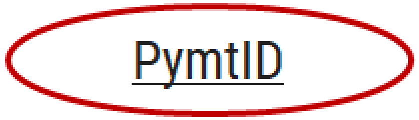
Derived
Attribute



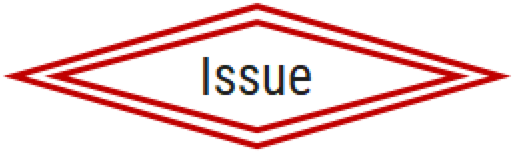
Multi Valued
Attribute



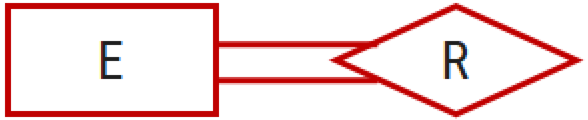
Weak Entity



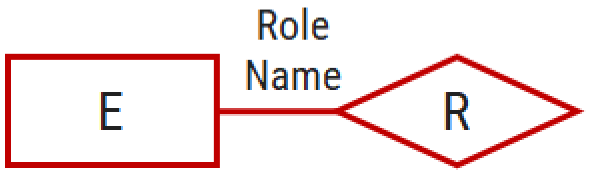
Discriminating
Attribute



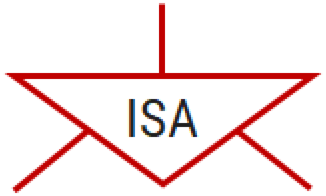
Weak Entity
Relationship



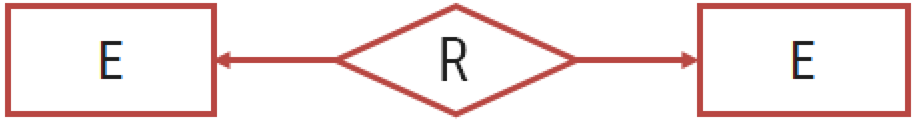
Total
Participation



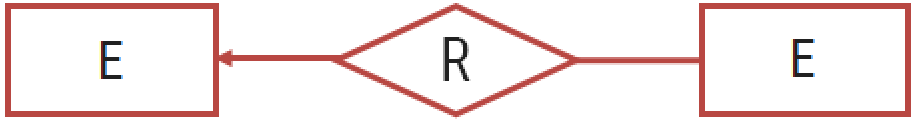
Role
Indicator



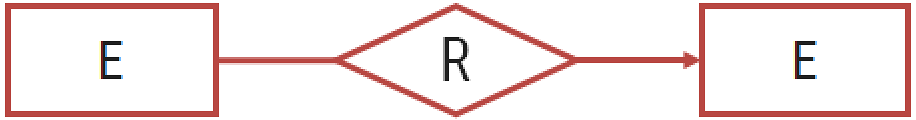
Specialization/
Generalization



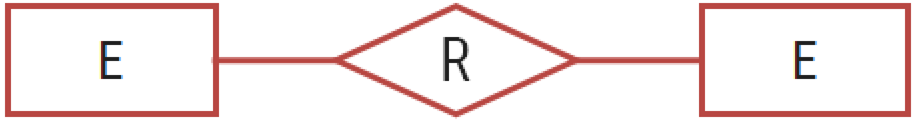
One to One



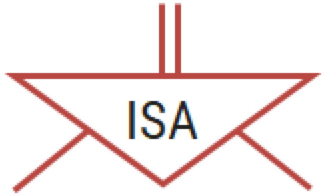
One to Many



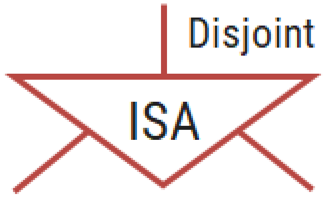
Many to One



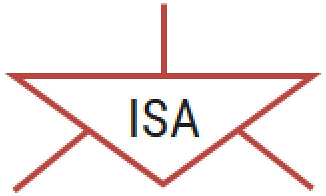
Many to Many



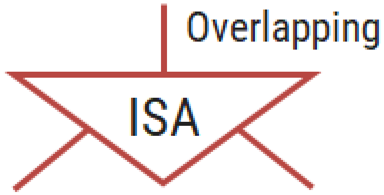
Total Specialization/ Generalization



Disjoint Specialization/ Generalization

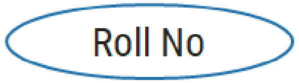
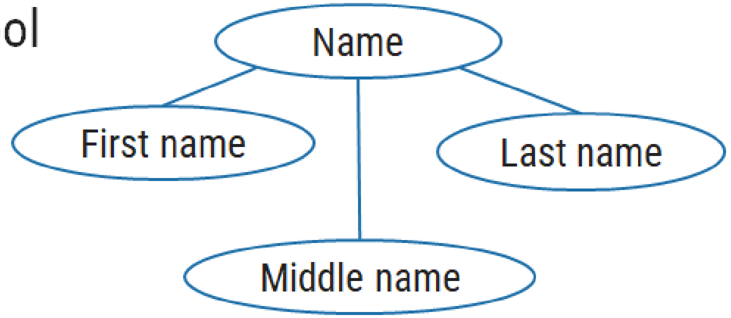


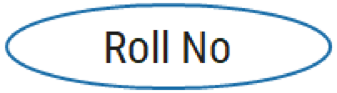
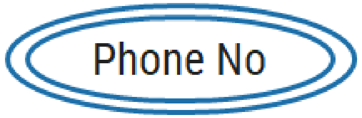
Partial Specialization/ Generalization

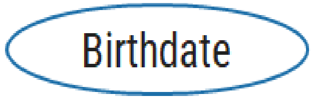
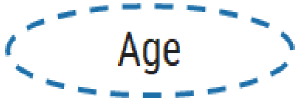


Overlapping Specialization/ Generalization

Types of Attributes

Simple Attribute	**Composite Attribute**
Cannot be divided into subparts	Can be divided into subparts
E.g. RollNo, CPI	E.g. Name (first name, middle name, last name) Address (street, road, city)
Symbol 	Symbol 

Single-valued Attribute	Multi-valued Attribute
Has single value	Has multiple (more than one) value
E.g. RollNo, CPI	E.g. PhoneNo (person may have multiple phone nos) EmailID (person may have multiple emails)
Symbol  	Symbol 

Stored Attribute	**Derived Attribute**
It’s value is stored manually in database	It’s value is derived or calculated from other attributes
E.g. Birthdate	E.g. Age (can be calculated using current date and birthdate)
Symbol  	Symbol 

Participation Constraints

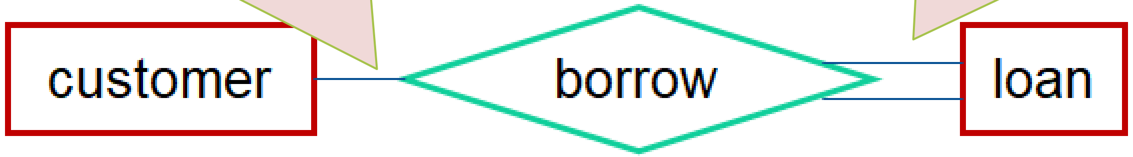
- It specifies the **participation of an entity set** in a relationship set.
- There are two types participation constraints
  - Total participation
  - Partial participation

Partial participation

- some entities in the entity set may **not participate** in any relationship in the relationship set.
- indicated by single line

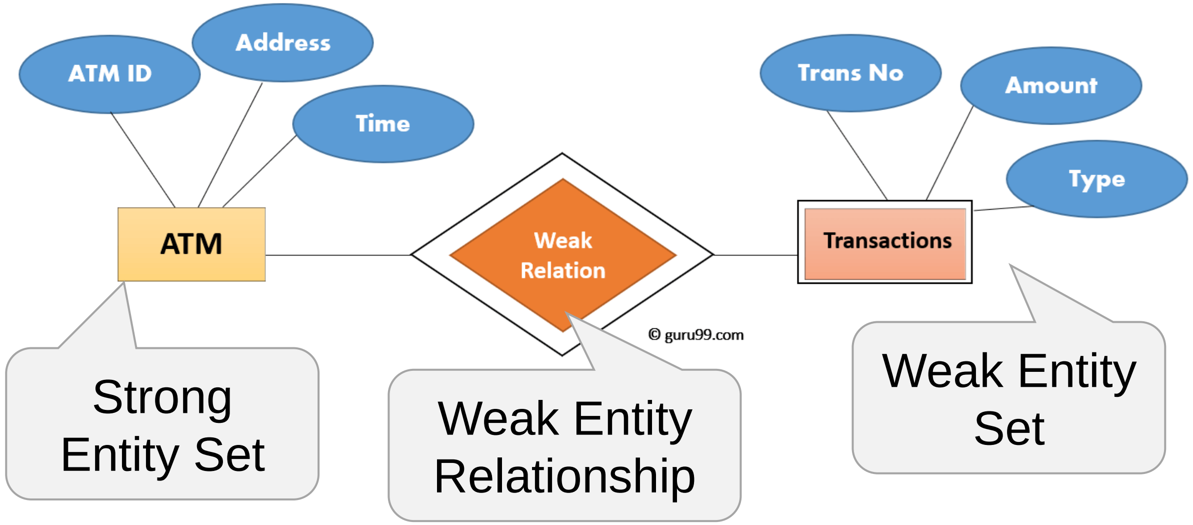
Total participation

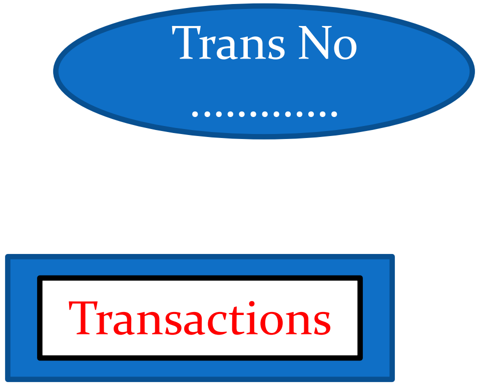
- every entity in the entity set participates in at least one relationship in the relationship set.
- indicated by double line



Weak Entity Set

- An entity set that does not have a primary key is called weak entity set.
- A weak entity is a type of entity which doesn't have its key attribute.
- ATM ID AND TRANS NO IS primary key payment.

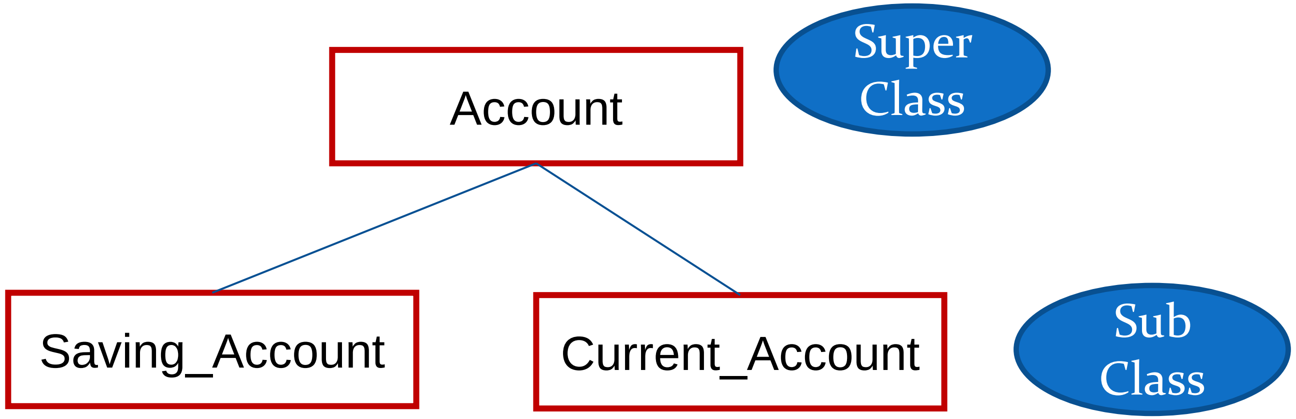




Strong Entity Set	Weak Entity Set
Strong entity set always has a primary key.	It does not have enough attributes to build a primary key.
It is represented by a rectangle symbol.	It is represented by a double rectangle symbol.
It contains a Primary key represented by the underline symbol.	It contains a Partial Key which is represented by a dashed underline symbol.
The member of a strong entity set is called as dominant entity set.	The member of a weak entity set called as a subordinate entity set.
Primary Key is one of its attributes which helps to identify its member.	In a weak entity set, it is a combination of primary key and partial key of the strong entity set.
In the ER diagram the relationship between two strong entity set shown by using a diamond symbol.	The relationship between one strong and a weak entity set shown by using the double diamond symbol.
The connecting line of the strong entity set with the relationship is single.	The line connecting the weak entity set for identifying relationship is double.

Superclass v/s Subclass

Super Class	Sub Class
A superclass is an entity from which **another entities can be derived**.	A subclass is an entity that is **derived from another entity**.
E.g, an entity account has two subsets saving_account and current_account So an **account is superclass**.	E.g, saving_account and current_account entities are derived from entity account. So **saving_account and current_account are subclass**.



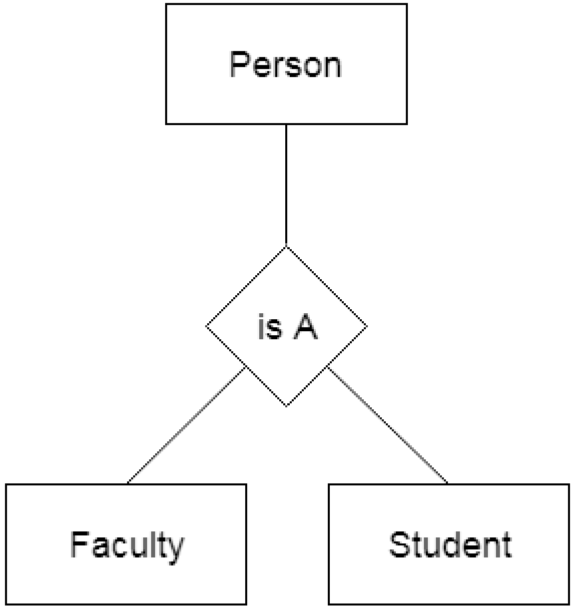
Generalization v/s Specialization

DBMS Generalization

- ✓ Generalization is like a **bottom-up approach** in which **two or more entities of lower level combine** to form a higher level entity if they have some attributes in common.
- ✓ Generalization is more like subclass and superclass system, but the only difference is the approach. Generalization uses the bottom-up approach.
- ✓ In generalization, entities are combined to form a more generalized entity, i.e., subclasses are combined to make a superclass

DBMS Generalization

- **For example**, Faculty and Student entities can be generalized and create a higher level entity Person.

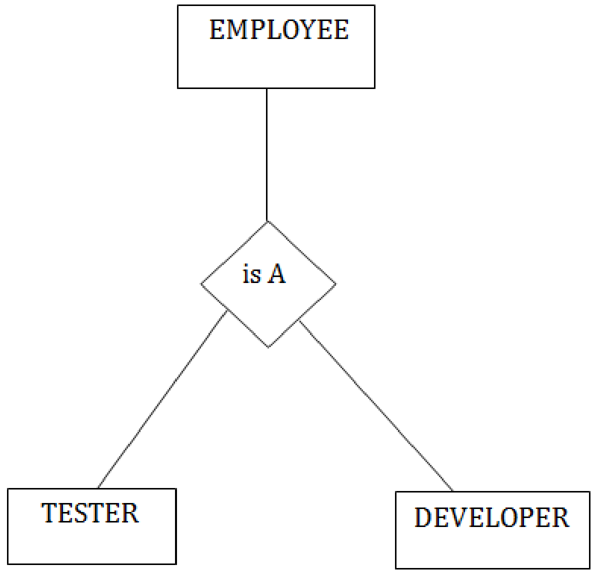


DBMS Specialization

- ✓ Specialization is a **top-down** approach, and it is **opposite to Generalization**. In specialization, one higher level entity can be broken down into **two lower level entities**.
- ✓ Specialization is used to identify the subset of an entity set that shares some distinguishing characteristics.
- ✓ Normally, the superclass is defined first, the subclass and its related attributes are defined next, and relationship set are then added

DBMS Specialization

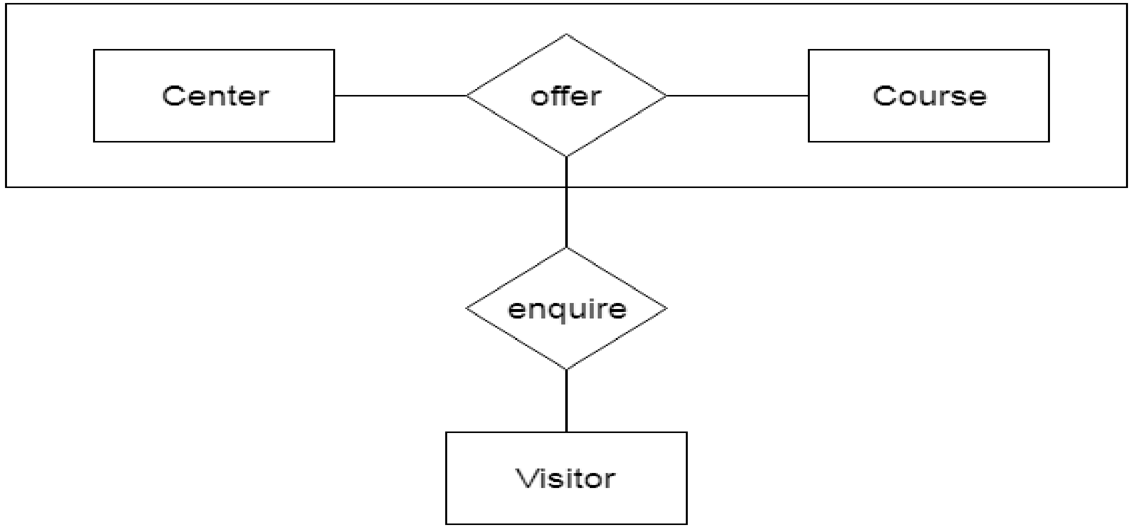
For example: In an Employee management system, EMPLOYEE entity can be specialized as TESTER or DEVELOPER based on what role they play in the company.



DBMS Aggregation

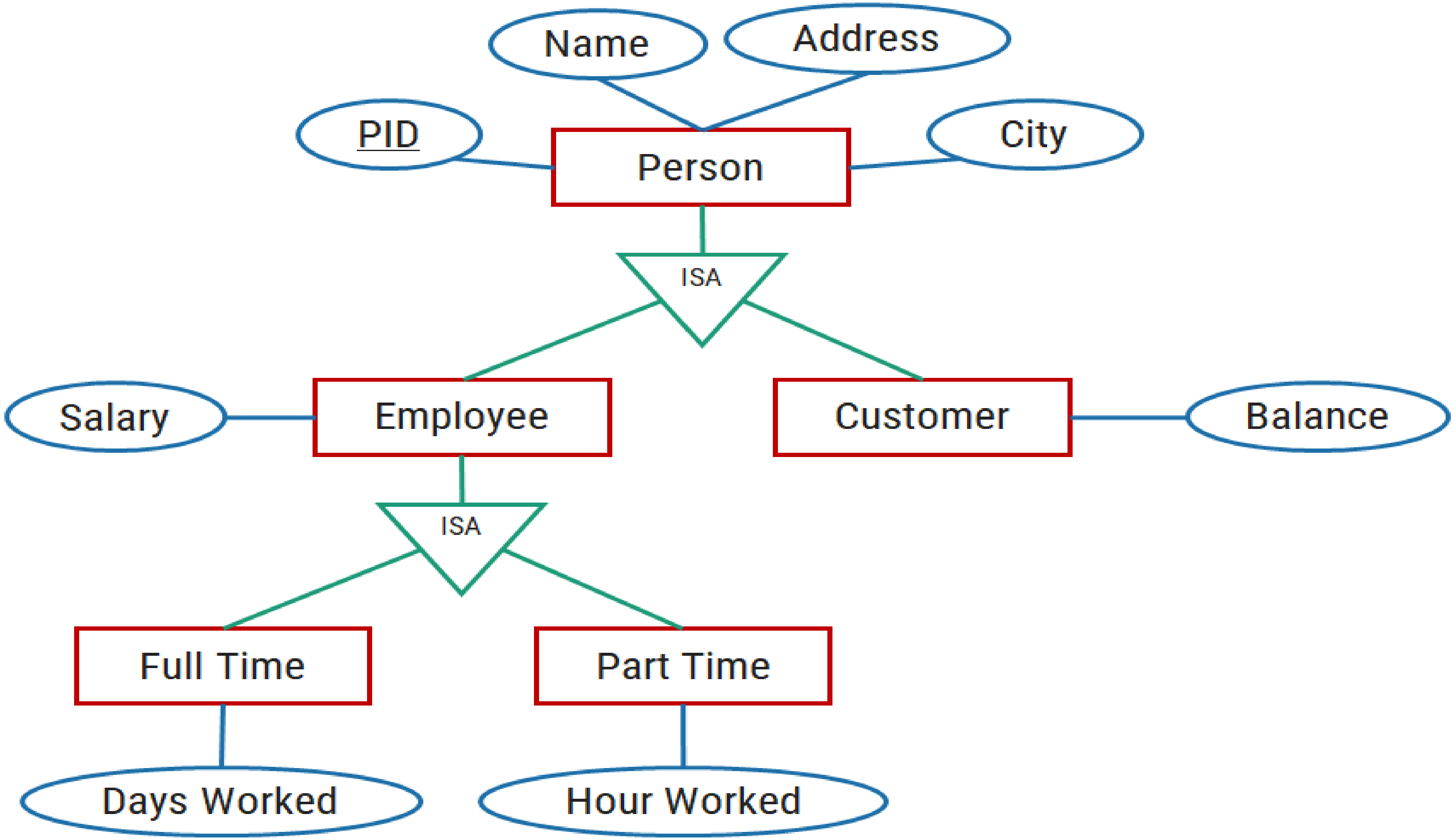
- ✓ In aggregation, the relation between two entities is treated as a single entity. In aggregation, relationship with its corresponding entities is aggregated into a higher level entity.

- **For example:** Center entity offers the Course entity act as a single entity in the relationship which is in a relationship with another entity visitor. In the real world, if a visitor visits a coaching center then he will never enquiry about the Course only or just about the Center instead he will ask the enquiry about both.



Generalization	**Specialization**
The process of **creation of group from various entities** is called generalization.	The process of **creation of sub-groups within an entity** is called specialization.
It is **Bottom-up** approach.	It is **Top-down** approach.
The process of taking the **union of two or more lower level entity** sets to produce a higher level entity set.	The process of taking a **sub set of higher level entity set** to form a lower level entity set.

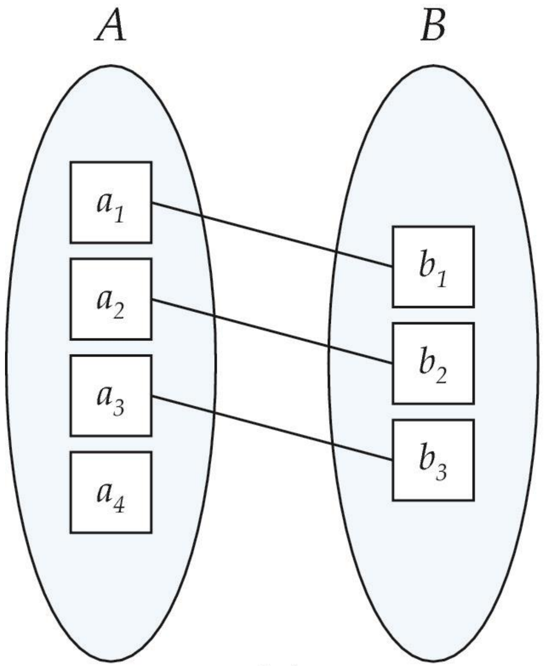
Generalization & Specialization example



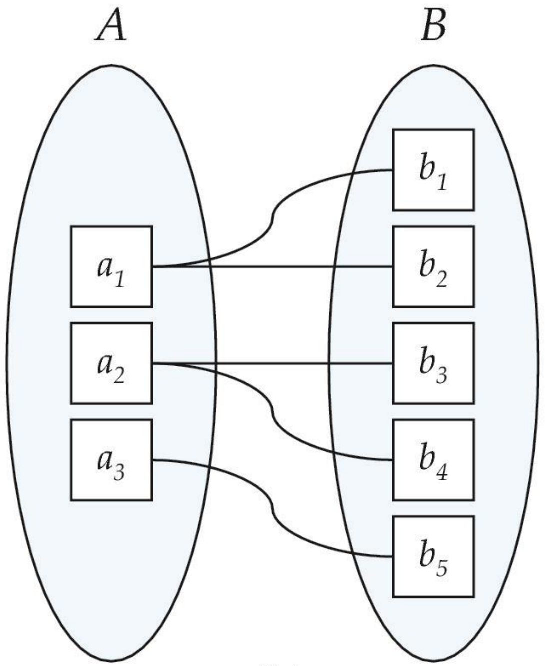
Mapping Cardinality Constraints

- One to one
- One to many
- Many to one
- Many to many

Mapping Cardinalities



(a)

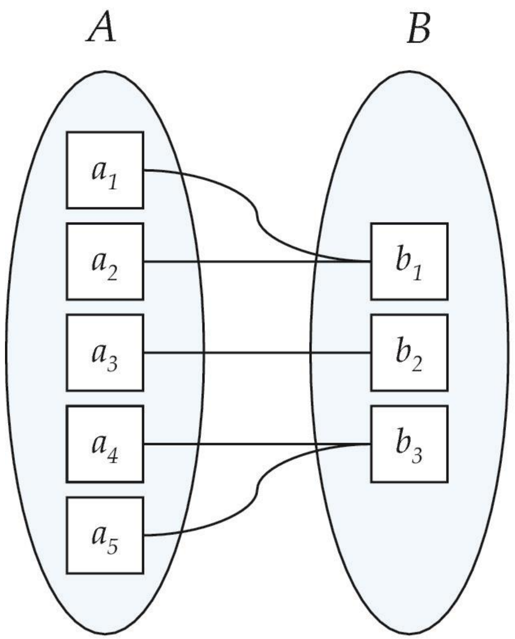


(b)

One to one

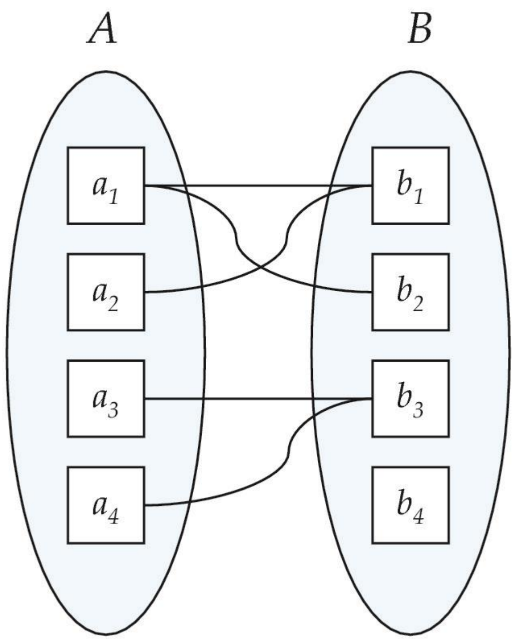
One to many

Mapping Cardinalities



(a)

Many to one

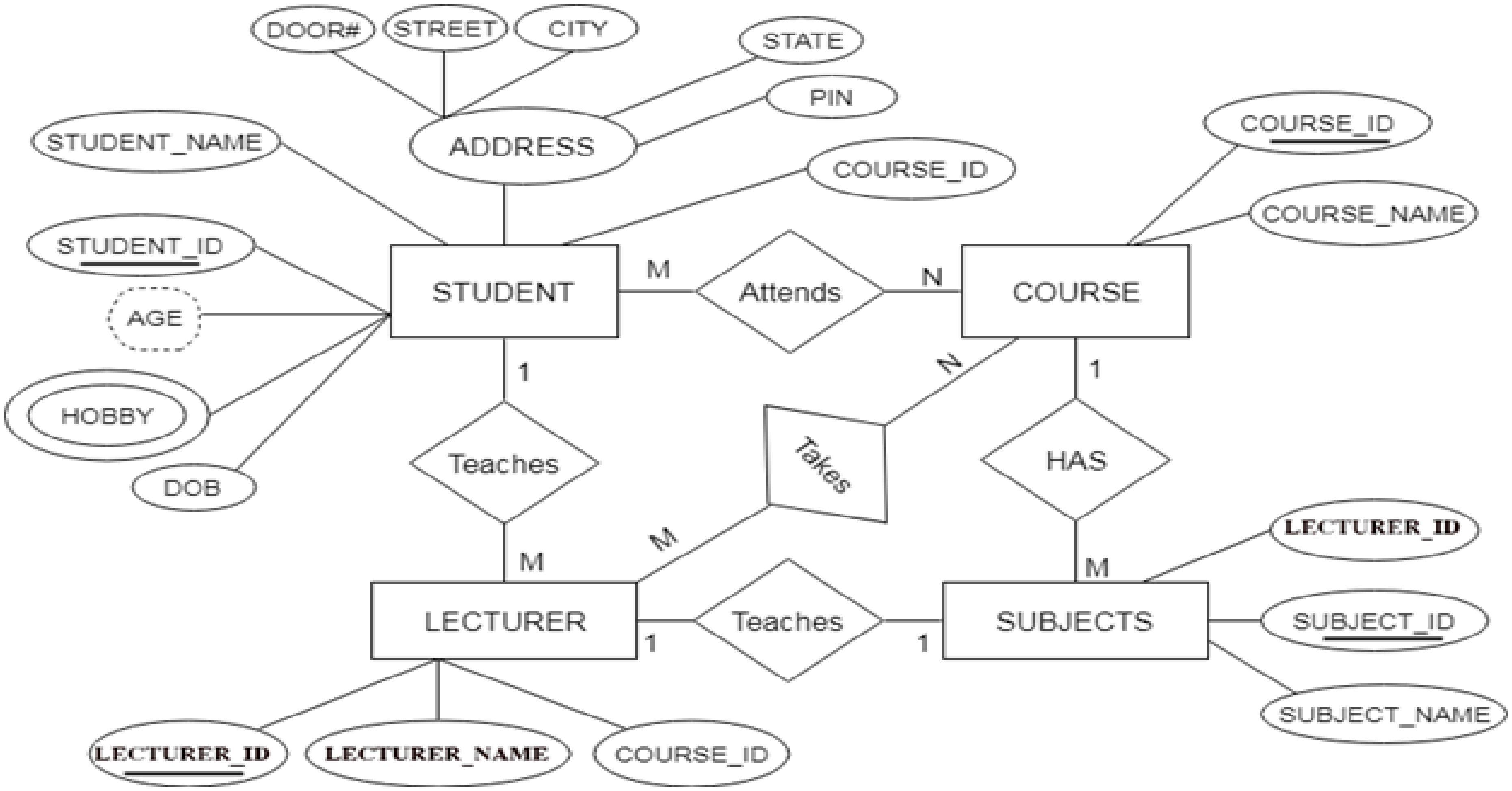


(b)

Many to Many

Reduction of ER diagram to Table

- ✓ There are some points for converting the ER diagram to the table:
  - ✓ Entity type becomes a table.
  - ✓ All single-valued attribute becomes a column for the table.
  - ✓ A key attribute of the entity type represented by the primary key.
  - ✓ The multivalued attribute is represented by a separate table.
  - ✓ Composite attribute represented by components.
  - ✓ Derived attributes are not considered in the table.


DOOR#
STREET
CITY
STATE
PIN
ADDRESS
STUDENT_NAME
COURSE_ID
COURSE_ID
COURSE_NAME
STUDENT_ID
AGE
HOBBY
DOB
STUDENT
M
Attends
N
COURSE
1
N
1
Teaches
Takes
HAS
M
M
M
LECTURER_ID
SUBJECT_ID
SUBJECT_NAME
LECTURER
1
Teaches
1
SUBJECTS
LECTURER_ID
LECTURER_NAME
COURSE_ID

Reduction of ER diagram to Table

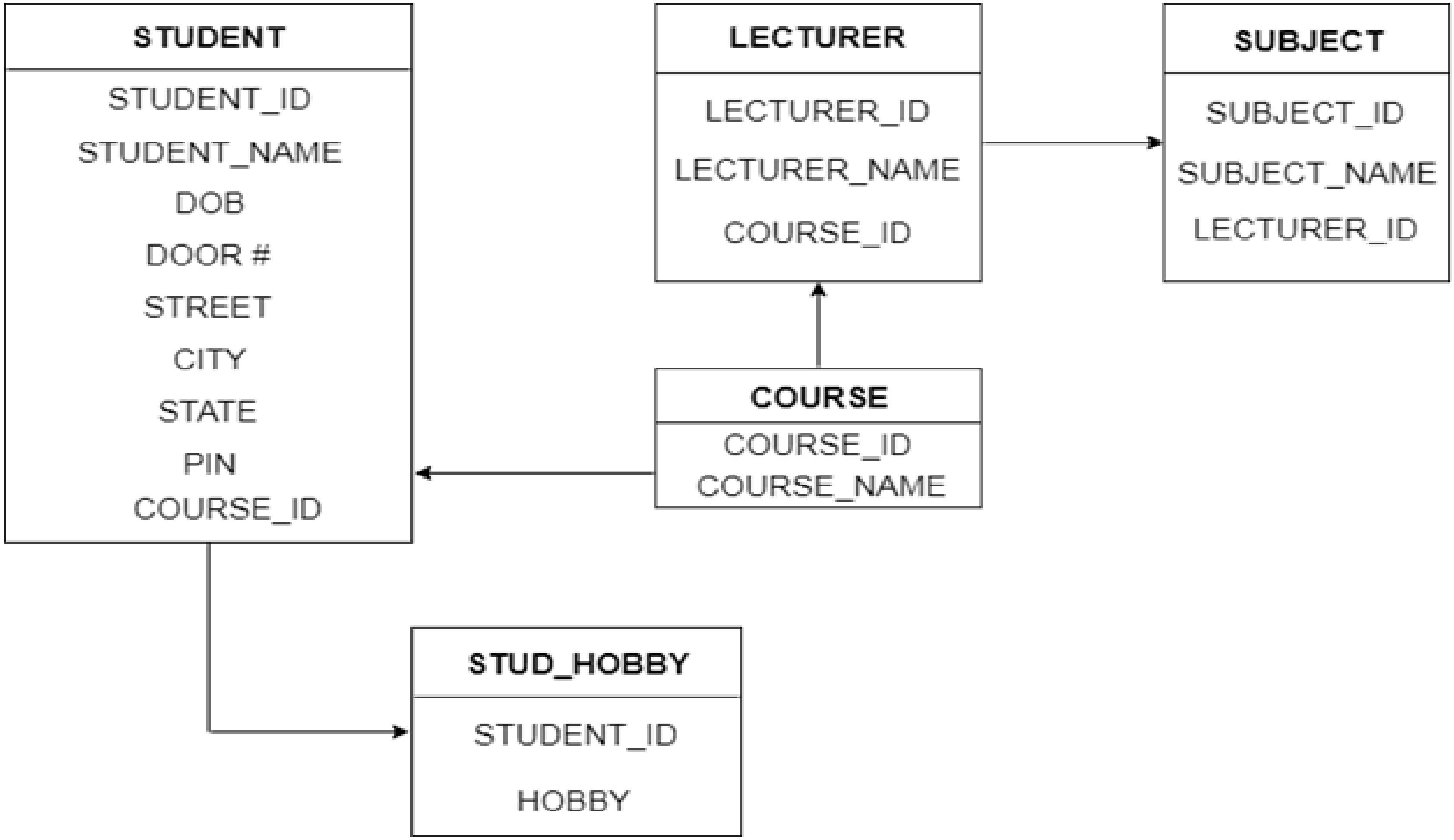
Entity type becomes a table.

- In the given ER diagram, LECTURE, STUDENT, SUBJECT and COURSE forms individual tables.

All single-valued attribute becomes a column for the table.

- In the STUDENT entity, STUDENT_NAME and STUDENT_ID form the column of STUDENT table. Similarly, COURSE_NAME and COURSE_ID form the column of COURSE table and so on.

Reduction of ER diagram to Table



STUDENT

STUDENT_ID

STUDENT_NAME

DOB

DOOR #

STREET

CITY

STATE

PIN

COURSE_ID

LECTURER

LECTURER_ID

LECTURER_NAME

COURSE_ID

SUBJECT

SUBJECT_ID

SUBJECT_NAME

LECTURER_ID

COURSE

COURSE_ID

COURSE_NAME

STUD_HOBBY

STUDENT_ID

HOBBY

➢Reduction of ER diagram to Table

A key attribute of the entity type represented by the primary key.

- In the given ER diagram, COURSE_ID, STUDENT_ID, SUBJECT_ID, and LECTURE_ID are the key attribute of the entity.

The multivalued attribute is represented by a separate table.

- In the student table, a hobby is a multivalued attribute. So it is not possible to represent multiple values in a single column of STUDENT table. Hence we create a table STUD_HOBBY with column name STUDENT_ID and HOBBY. Using both the column, we create a composite key.

Composite attribute represented by components.

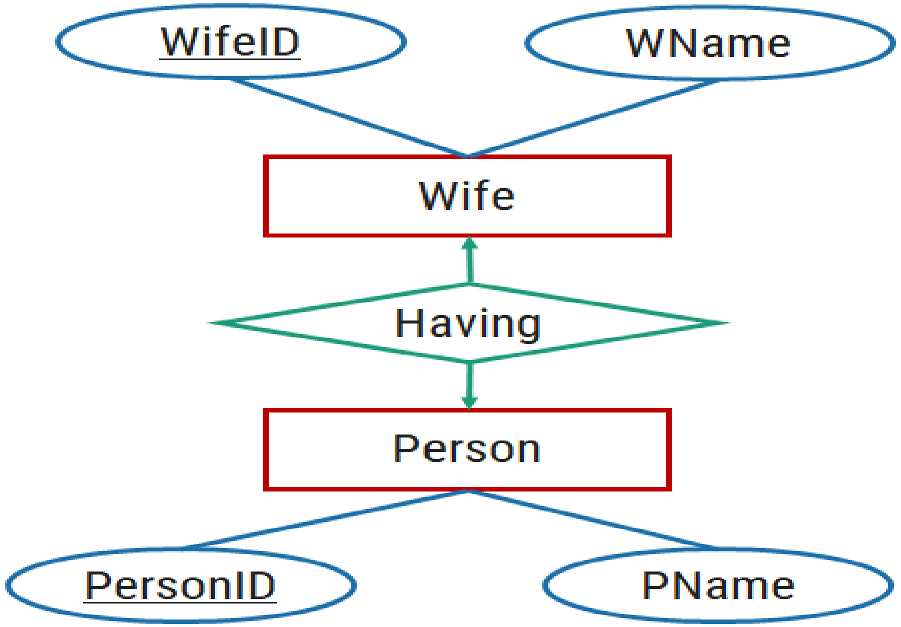
- In the given ER diagram, student address is a composite attribute. It contains CITY, PIN, DOOR#, STREET, and STATE. In the STUDENT table, these attributes can merge as an individual column.

Derived attributes are not considered in the table.

- In the STUDENT table, Age is the derived attribute. It can be calculated at any point of time by calculating the difference between current date and Date of Birth.

1:1 Mapping Cardinality:

The **primary key** of any **one table** in to the **another table** as a **foreign key**.

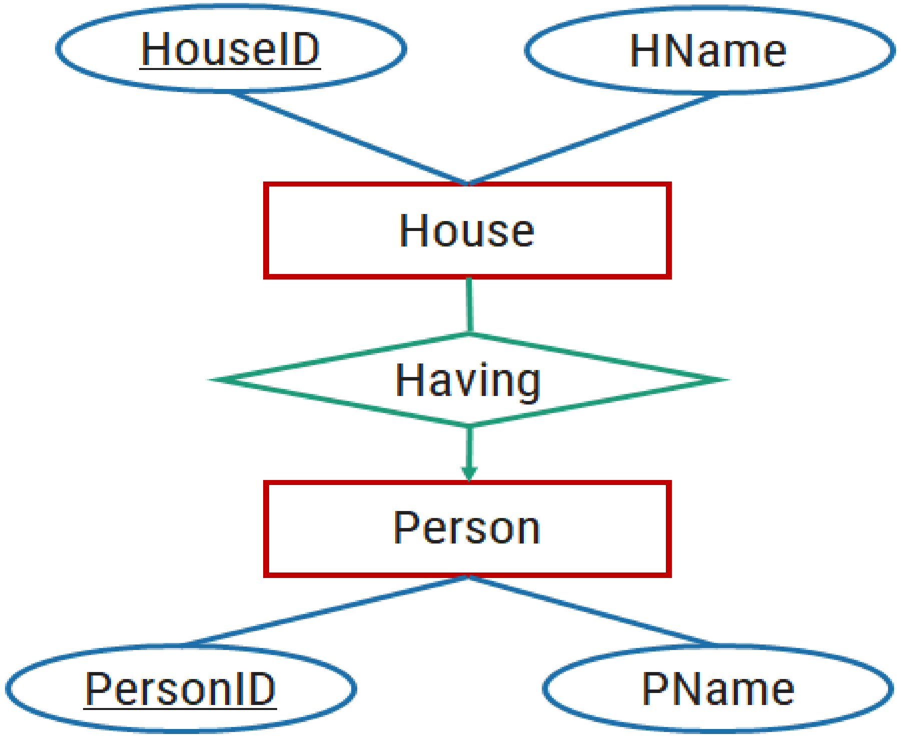


Person (PersonID, PName)
Wife (WifeID, Wname, PersonID)

Wife (WifeID, Wname)
Person (PersonID, Pname, WifeID)

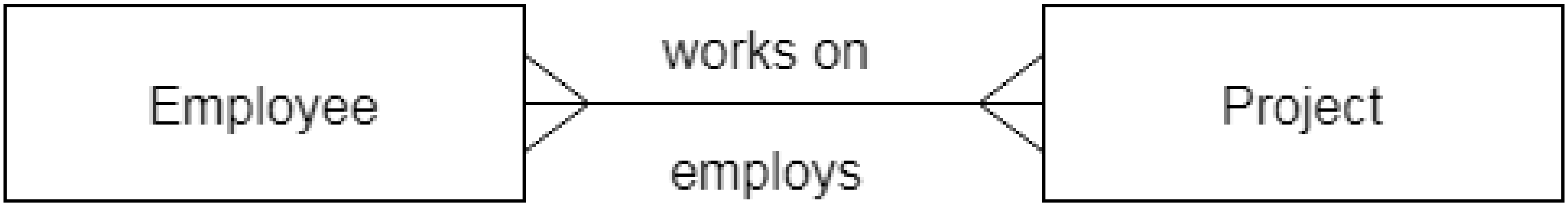
1:N Mapping Cardinality:

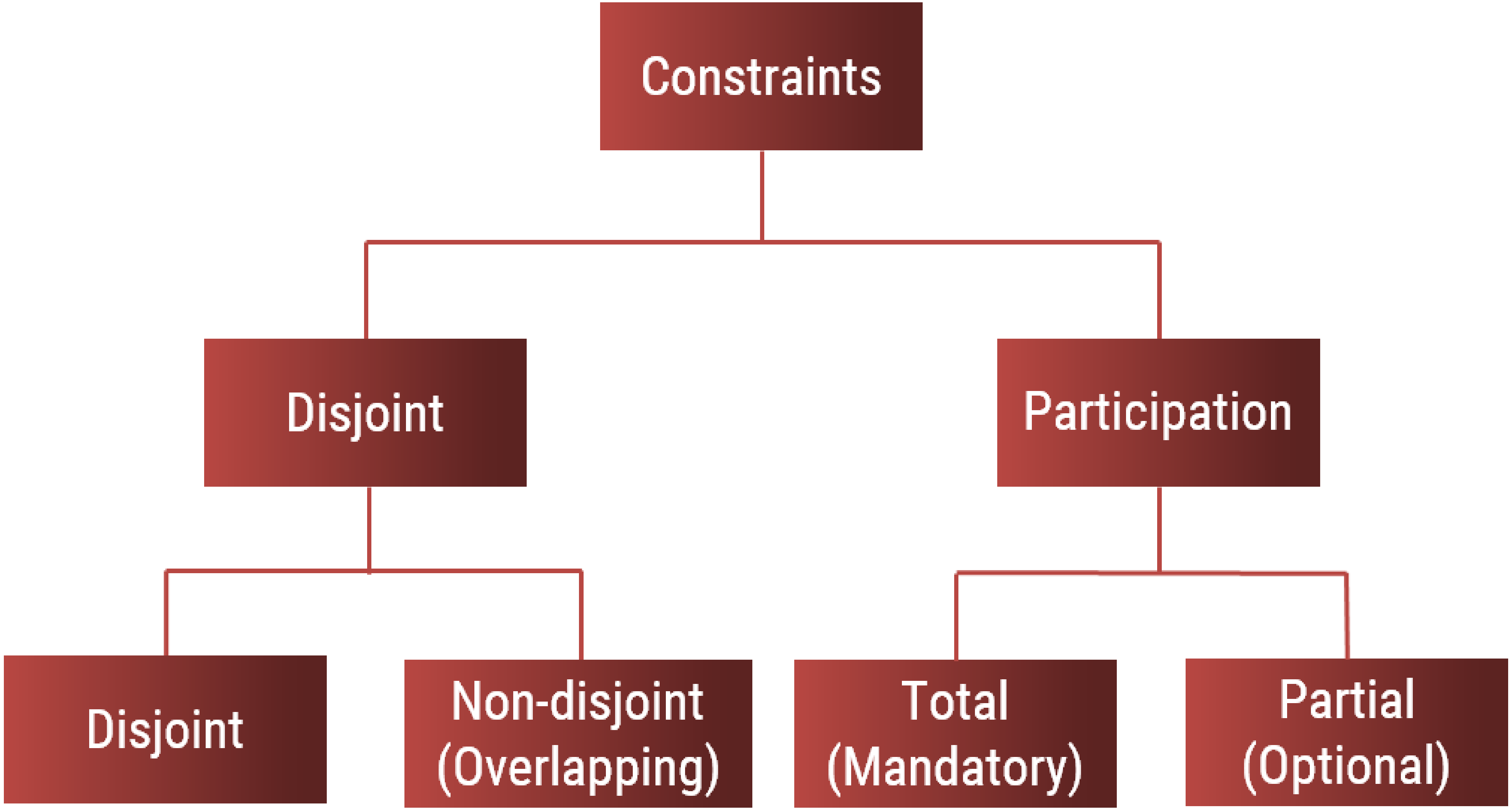
primary key of **table having 1 mapping** in to the another **table having many cardinality as a Foreign key**.



Person (PersonID, PName)
House (HouseID, Hname, PersonID)

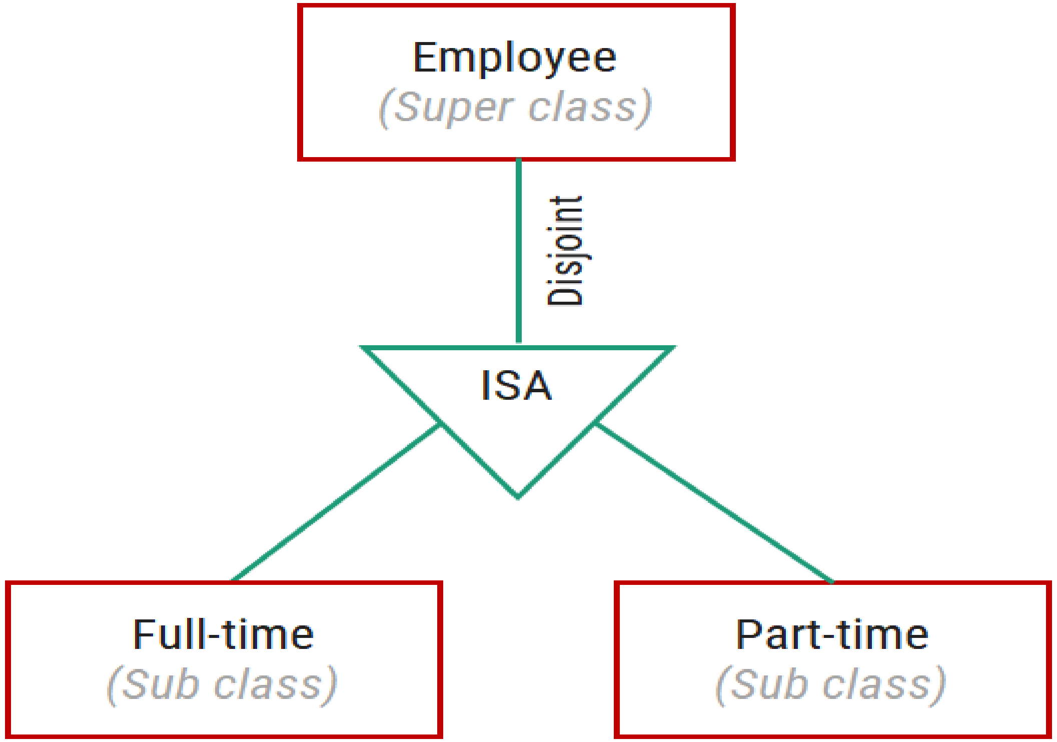
N:N Mapping Cardinality:




Constraints
Disjoint
Participation
Disjoint
Non-disjoint (Overlapping)
Total (Mandatory)
Partial (Optional)

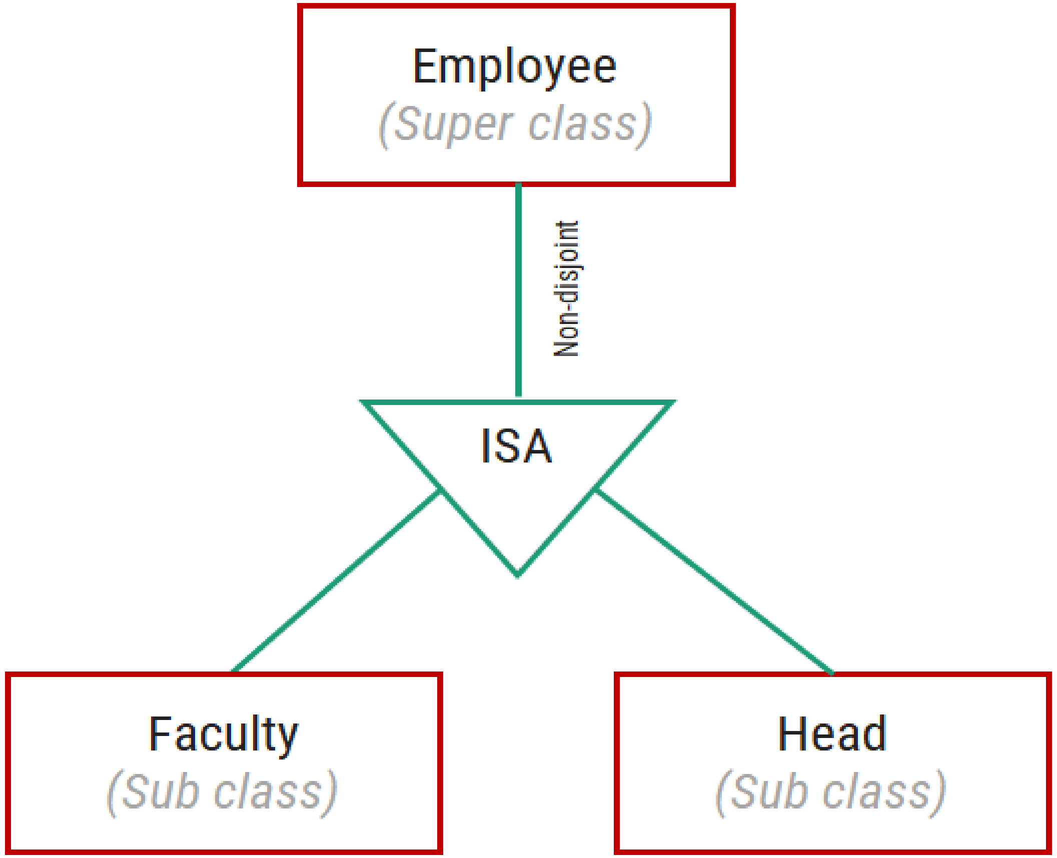
Disjoint Constraint

It specifies that the **entity of a super class can belong to only one lower-level entity set** (sub class).



Non-disjoint (Overlapping) Constraint

It specifies that an **entity of a super class can belong to more than one lower-level entity** set (sub class).



Participation Constraints

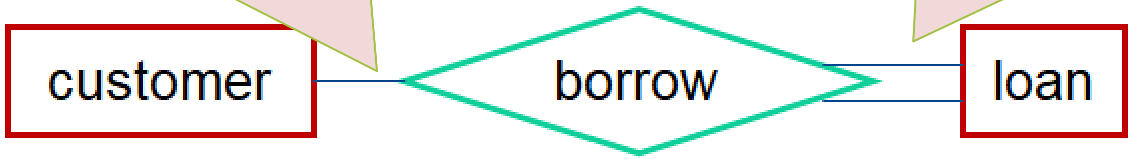
- It specifies the **participation of an entity set** in a relationship set.
- There are two types participation constraints
  - Total participation
  - Partial (optional)participation

Partial participation

- some entities in the entity set may **not participate** in any relationship in the relationship set.
- indicated by single line

Total participation

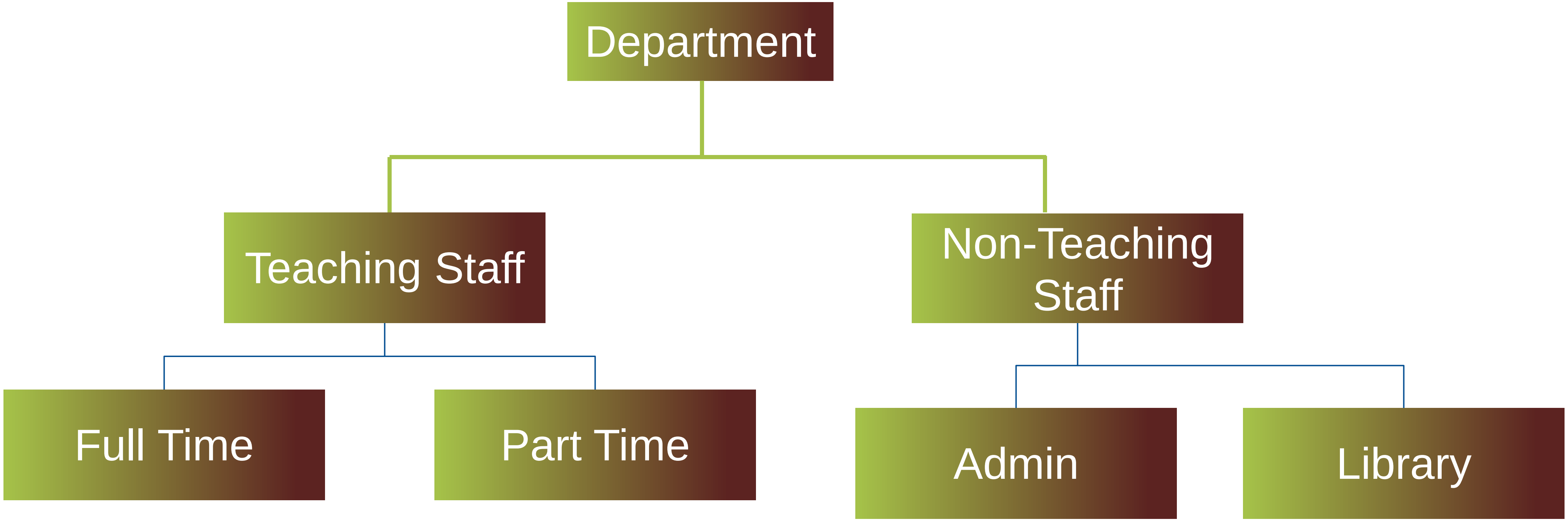
- every entity in the entity set participates in at least one relationship in the relationship set.
- indicated by double line



Type of Database Models

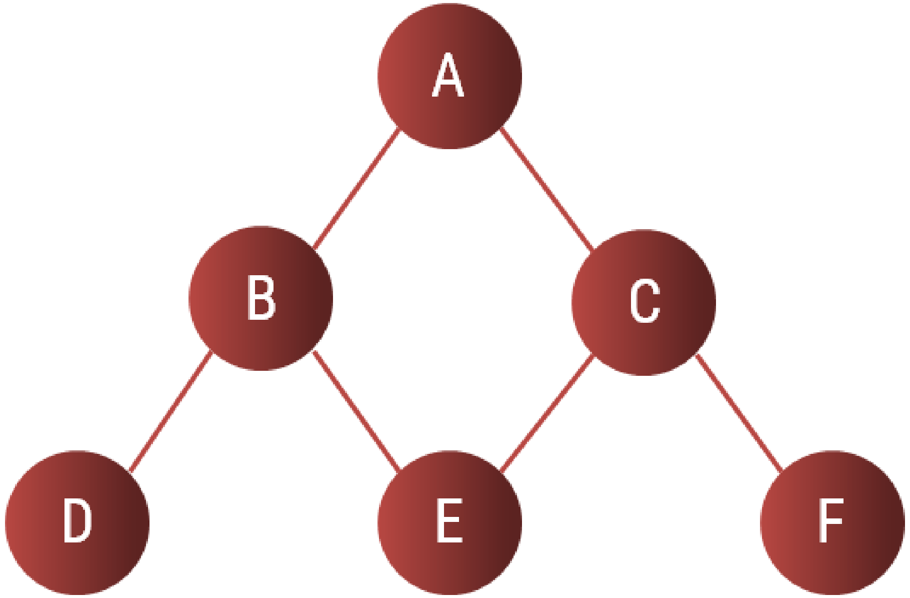
1. Hierarchical Model
2. Network Model
3. Entity-relationship Model
4. Relational Model
5. Object-oriented database Model

Hierarchical Model

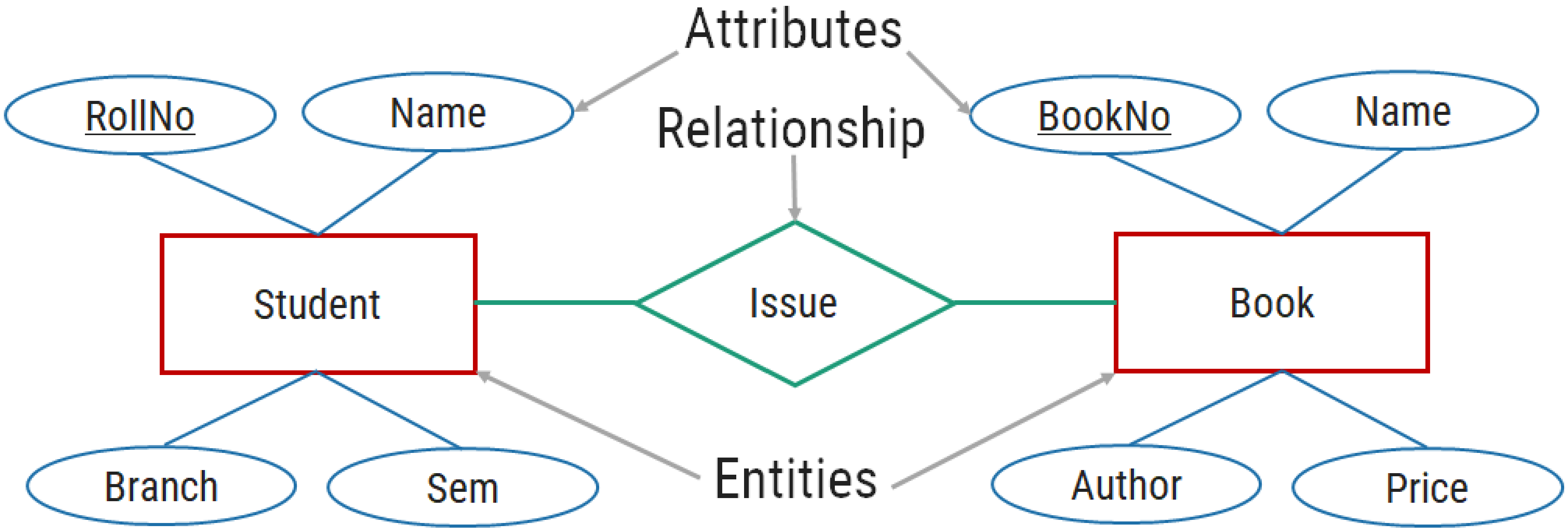


Network Model

- This is an **extension of the hierarchical model**, allowing **many-to-many relationships** in a tree-like structure that **allows multiple parents**.

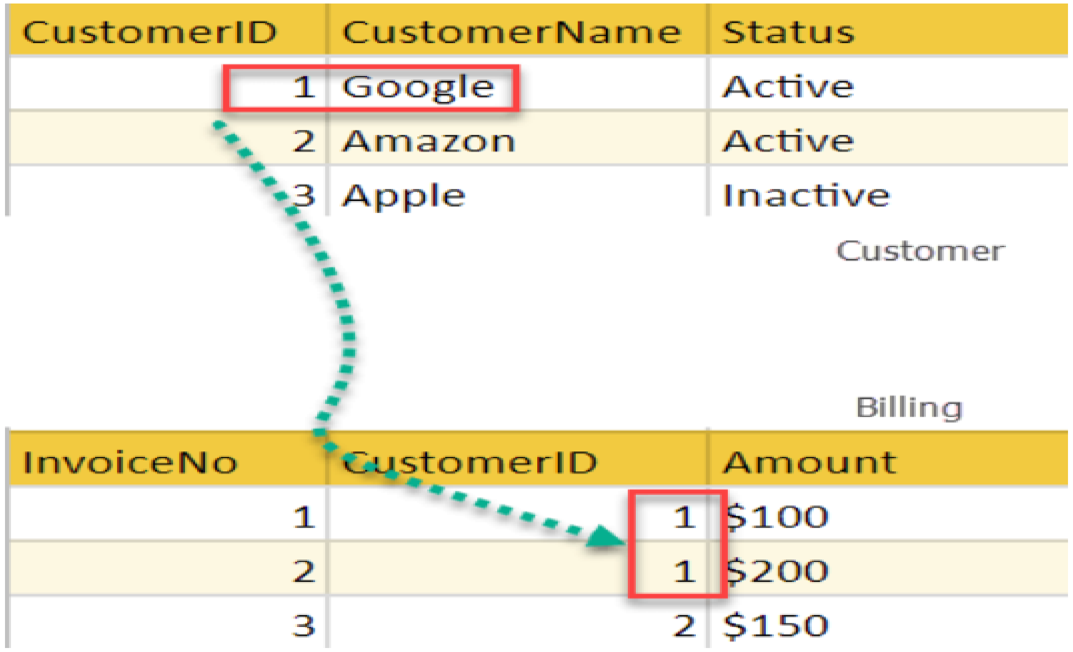


Entity-relationship Model



Relational Model

Data is organized in two-dimensional tables and the relationship is maintained by **storing a common attribute**. Also define some key relation in table (**EX. Customer and Bill**)



CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive

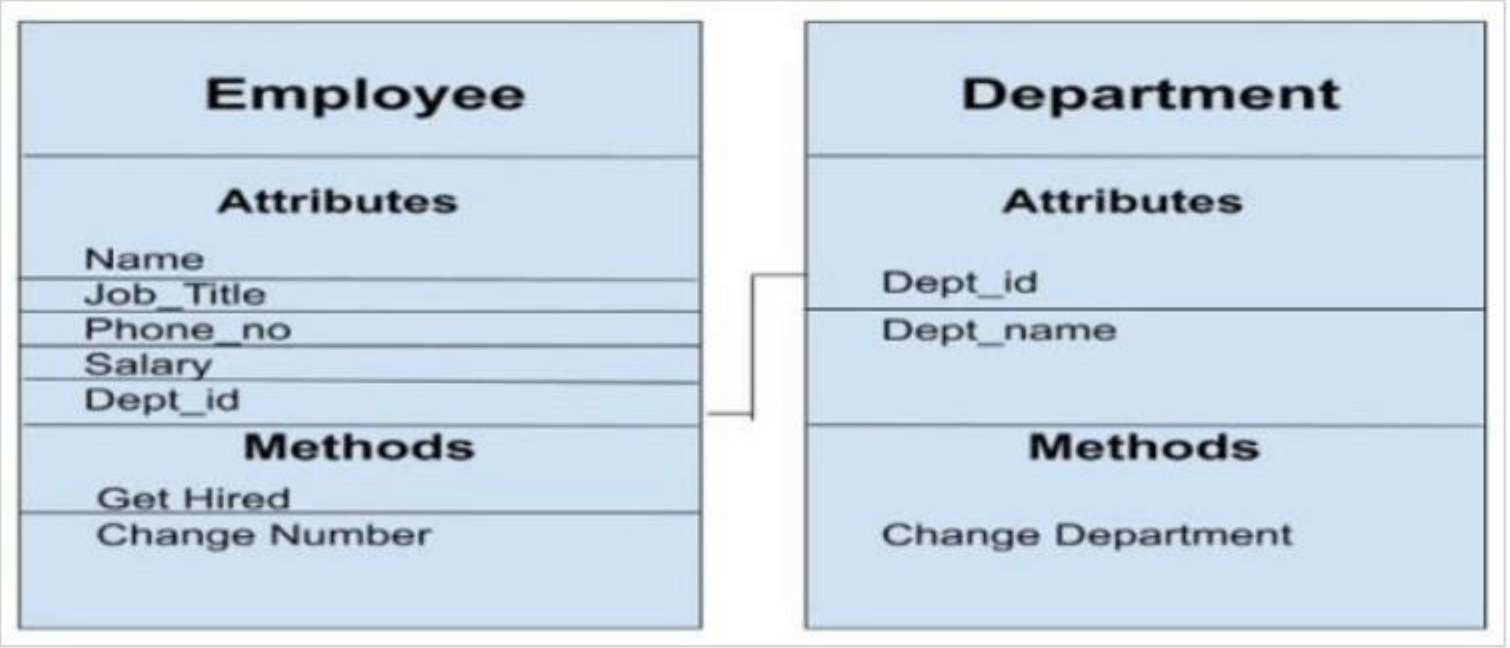
Customer

Billing

InvoiceNo	CustomerID	Amount
1	1	$100
2	1	$200
3	2	$150

Object-oriented database Model

Object oriented data model is based upon real world situations. These situations are represented as objects, with different attributes. All these object have multiple relationships between them.



Employee

Attributes

Name
Job_Title
Phone_no
Salary
Dept_id

Methods

Get Hired
Change Number

Department

Attributes

Dept_id
Dept_name

Methods

Change Department

Integrity Constraints

- **Check**

Limits the data values of variables to a **specific set, range, or list of values**.

The constraint can be applied for a single column or a group of columns.

E.g. value of SPI should be between 0 to 10.

- Not null
- Unique
- Primary key
- Foreign key

Thank You