

# **Database Systems (M3020008)**

## **Lecture 3**

**School of Computer Science and Engineering  
(SoCSE)**

**Digital University Kerala ,Thiruvananthapuram**

# **The evolution of Data Models**

# Data Models

- To come across the limitations of file systems, there are lot of researchers and software developers designed and developed various data models.

**Data Models** : Underlying the structure of a database is the **data model**: a collection of conceptual tools for describing data, data relationships, data semantics, and consistency constraints. A data model provides a way to describe the design of a database at the physical, logical, and view levels.

The important and widely accepted models are:

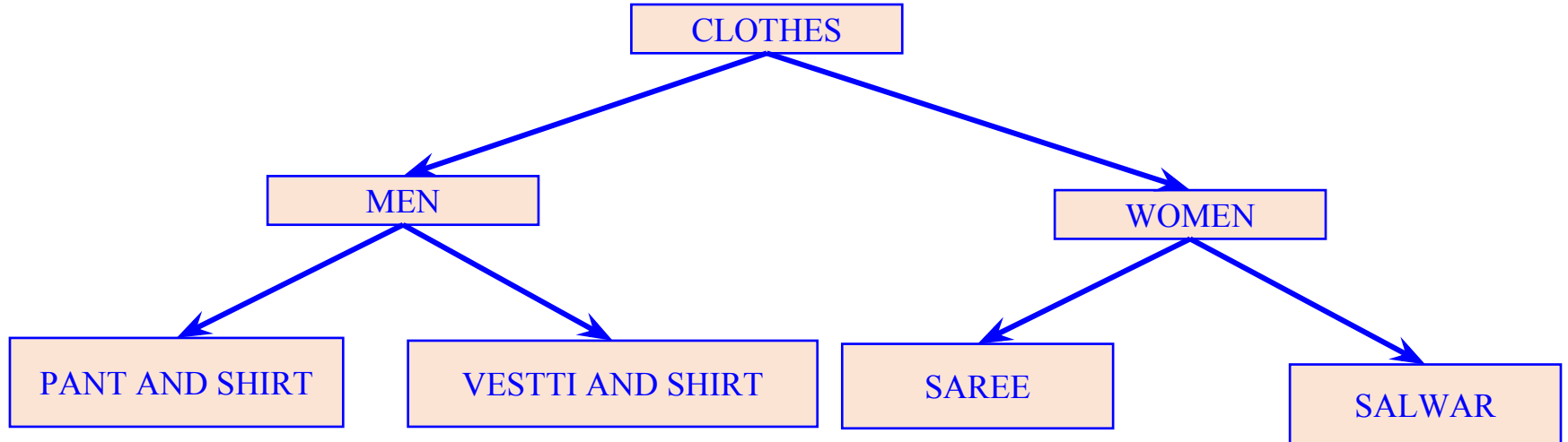
- ✓ Hierarchical
- ✓ Network
- ✓ Entity relationship
- ✓ Relational
- ✓ Object oriented

# Hierarchical Model

- The first and foremost model of the DBMS.
- This model organizes the data in the hierarchical tree structure.
- This model is easy to understand with real time examples site map of a website

**Example :** For example the following is the representation of relationships present online clothes shopping

# Example



# Features of a Hierarchical Model

- ✓ One-to-many relationship:
- ✓ Parent-Child Relationship
- ✓ Deletion Problem:
- ✓ Pointers

## Advantages of Hierarchical Model

- ✓ Simple and fast traversal because of using tree structure
- ✓ Changes in parent node automatically reflected in child node

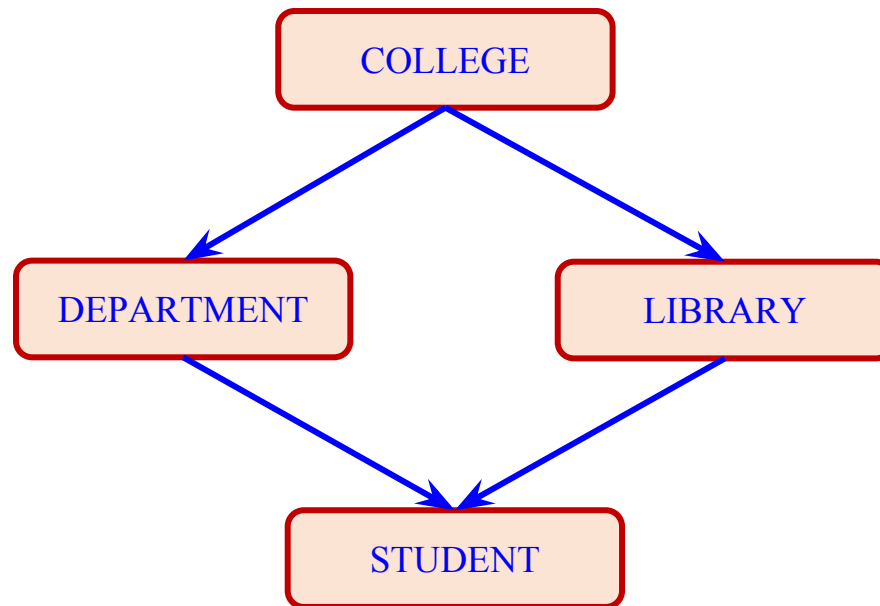
## Disadvantages of Hierarchical Model

- ✓ Complexity
- ✓ Parent node deleted automatically child node will be deleted

# Network Model

- ✓ Network model is an extension of hierarchical model.
- ✓ This model was recommended as the best before relationship model.
- ✓ Same like hierarchical model, the only difference between these two models are a record can have more than one parent
- ✓ For Example consider the following diagram a student entity has more than one parent

# Example





## Features of a Network Model

- ✓ Manage to Merge more Relationships
- ✓ More paths
- ✓ Circular Linked List

## Advantages of Network Model

- ✓ Data access is faster
- ✓ Because of parent child relationship , the changes in parent reflect in child

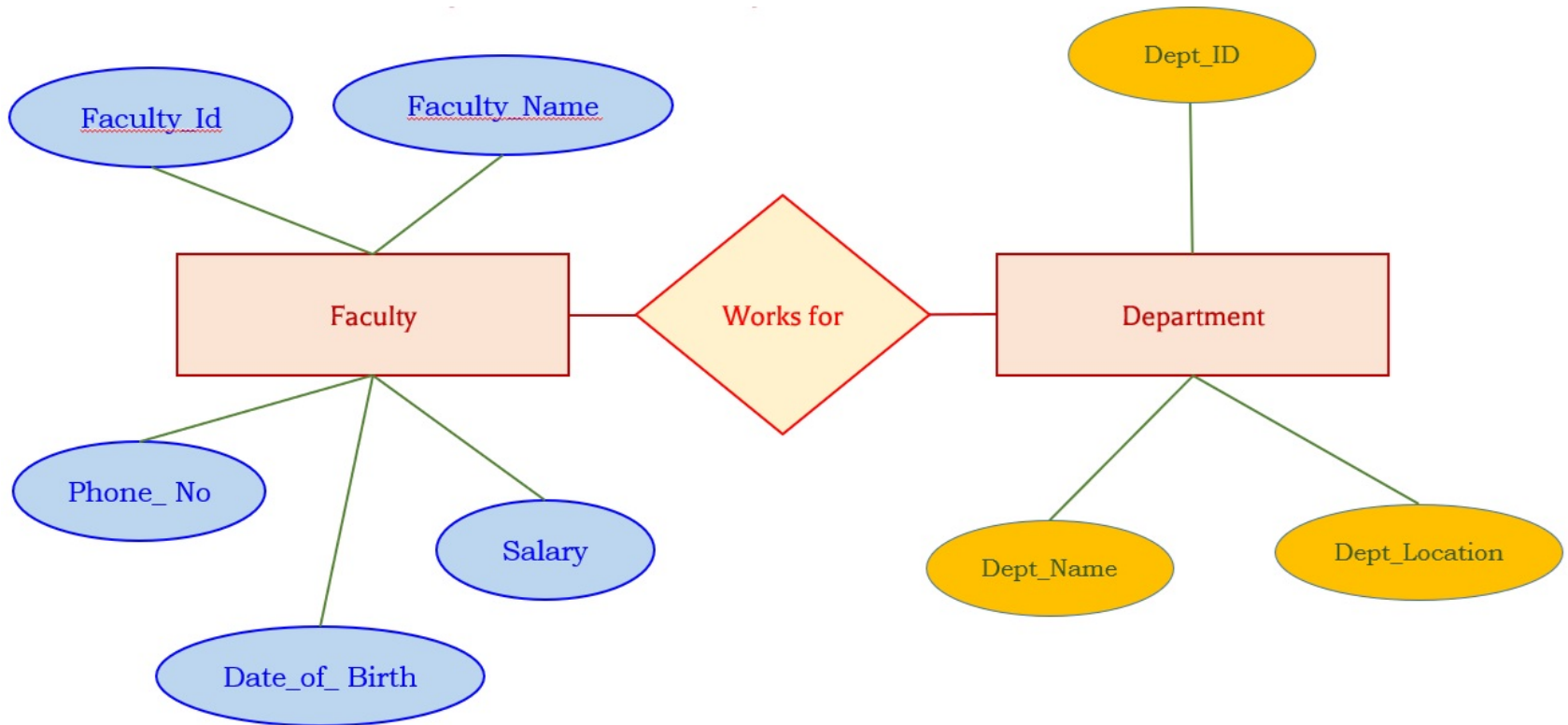
## Disadvantages of Network Model

- ✓ More complex because of more and more relations

# Entity-Relationship Model (ER Model)

- ✓ This model is a high level data model
- ✓ Represents the real – world problem as a pictorial representation
- ✓ Easy to understand by the developers about the specification
- ✓ It is like a visualization tool to represent a specific database
- ✓ It contains three components
  1. Entities
  2. Attributes
  3. Relationships

# Example for ER Diagram ( Faculty and Department entity set)



In the above example:

- ✓ There are two entities , Faculty and Department
- ✓ The attributes of Faculty entities are
  - Faculty\_Id
  - Faculty\_Name
  - Phone\_No
  - Date\_of\_birth
  - Salary
- ✓ The attributes of Department entities are
  - Dept\_ID
  - Dept\_Name
  - Dept\_Location
- ✓ Relationship : Faculty works for a department

## Features of ER Model

- ✓ Graphical representation
- ✓ Visualization
- ✓ Good Database design (Widely used)

## Advantages of ER Model

- ✓ Very Simple
- ✓ Better communication
- ✓ Easy to convert to any model

## Disadvantage of ER Model

- ✓ No industry standard
- ✓ Hidden information

# Relational Model

- ✓ Widely used model
- ✓ Data are represented as row-wise and column-wise  
( 2 Dimensional Array)

**Example :** EMP (Employee) Table

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800	-	20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7566	JONES	MANAGER	7839	02-APR-81	2975	-	20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7698	BLAKE	MANAGER	7839	01-MAY-81	2850	-	30
7782	CLARK	MANAGER	7839	09-JUN-81	2450	-	10
7788	SCOTT	ANALYST	7566	09-DEC-82	3000	-	20
7839	KING	PRESIDENT	-	17-NOV-81	5000	-	10
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7876	ADAMS	CLERK	7788	12-JAN-83	1100	-	20
7900	JAMES	CLERK	7698	03-DEC-81	950	-	30
7902	FORD	ANALYST	7566	03-DEC-81	3000	-	20
7934	MILLER	CLERK	7782	23-JAN-82	1300	-	10

# Relational Model

- ✓ Each row is known as RECORD or TUPLE
- ✓ Each Column is known as ATTRIBUTE or FIELD
- ✓ The collection of attributes are called as record – An Entity
- ✓ The collection of records are called as Table – Entity Set
- ✓ In the above example:

Table – EMP

Attributes – Empno, Ename, Sal,....

## Features of Relational Model

- ✓ Records
- ✓ Attributes

## Advantages of Relational Model

- ✓ Simple
- ✓ Scalable
- ✓ Structured format
- ✓ Isolation

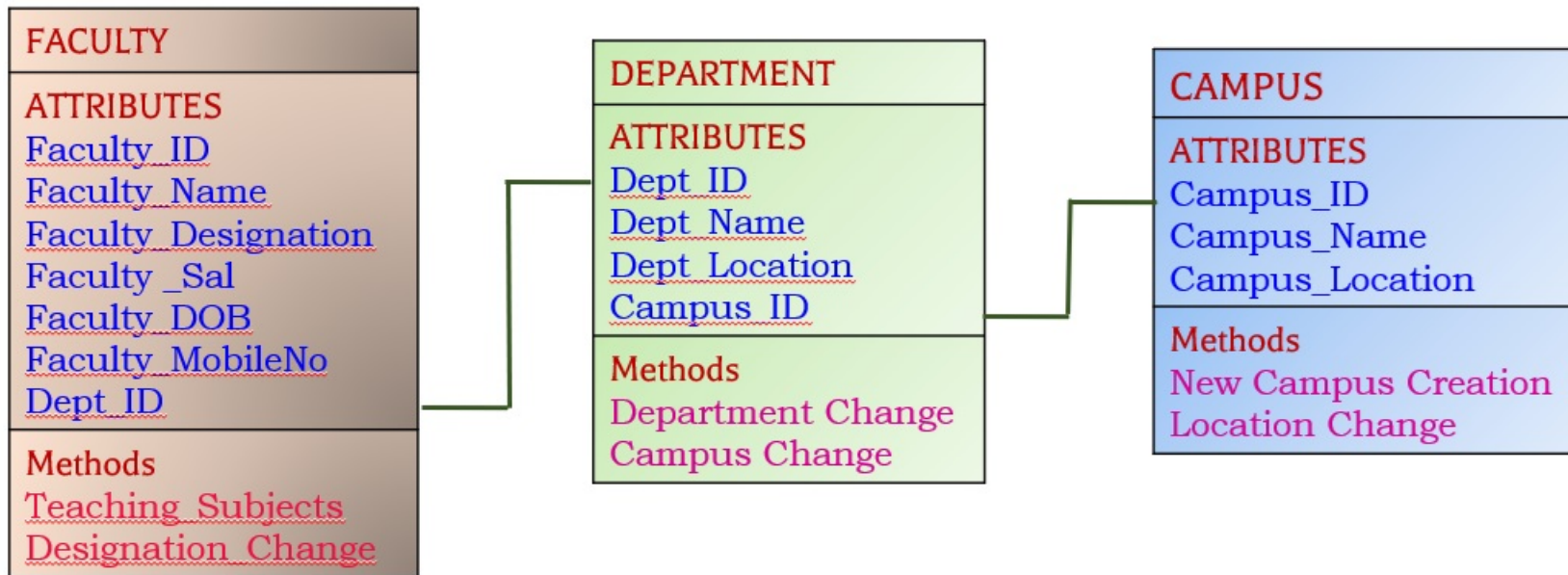
## Disadvantages of Relational Model

- ✓ Hardware overheads



## Object Oriented Model

- ✓ The real- time problems are easily represented through object-oriented data model which is an OBJECT.
- ✓ In this Model, the data and its relationship present in the single structure
- ✓ Complex data like images, audio, videos can be stored easily
- ✓ Objects connected through links using common attribute(s)
- ✓ **Example** : Three Objects Faculty, Department and Campus linked using common attribute



# Design process

# The Entity-Relationship (E-R) Model

Entity : Any object in the real world is an entity

Example : Person, Furniture, University / Department

The ER data model uses a collection of entities (objects) and relationships among these entities

Entities in database are described using their attributes / properties

Example 1 : The attributes like dept\_id, dept\_name, dept\_location, etc., describes about a particular department in an university.

Example 2 : The attributes Faculty\_id, Faculty\_name, Faculty\_salary, etc., describes about a faculty works for the particular department.

Note : The attributes dept\_id, faculty\_id used to identify an entity in an entity set. Like AADHAR CARD number for a person . ( Will be discussed later in detail )

# The Entity-Relationship (E-R) Model

Relationship :

- ✓ It is an association among several entities
- ✓ For example , a member is associates as faculty in her/his department.
- ✓ Faculty works for the department.

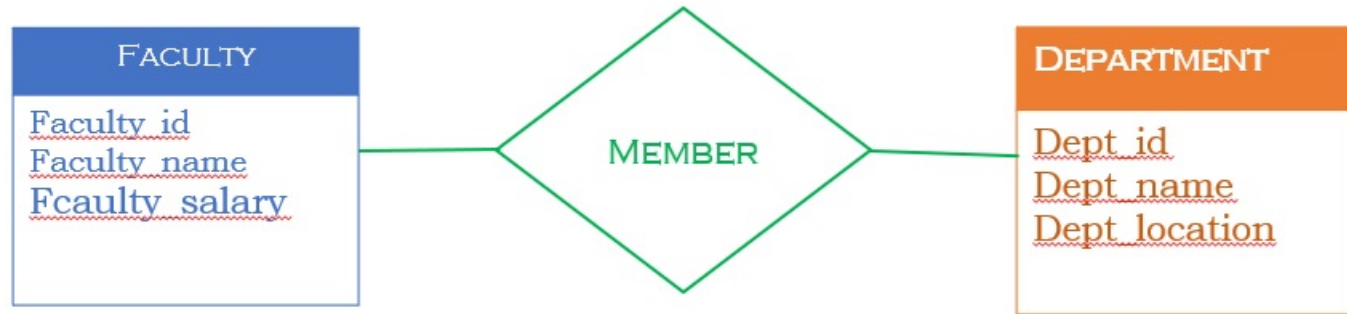
Entity set : Set of all entities of the same type

Relationship set : Set of all relationships of the same type

- ✓ The overall logical structure of a database can be represented using graphical notations by an E-R diagram.
- ✓ One of the most popular model is to use UML ( Unified Modeling Language)

# The Entity-Relationship (E-R) Model

## A Sample E-R Diagram



- ✓ Entity sets are represented by a **Rectangle** : Faculty and Department
  - Header as Name of the Entity set
  - Attributes are listed below the header
- ✓ Relationship sets are represented as **Diamond** : Member
- ✓ The above E-R diagram represents the relationship member between faculty and department

- ✓ Entity – Relationship (E-R) Model is the overall logical structure of database design about a particular enterprise or domain.
- ✓ E-R model is very useful in mapping the meaning and interactions of real world enterprises to conceptual schema.
- ✓ E-R Model is widely used model in database design.
- ✓ E-R Model employs three basic concepts
  - Entity sets
  - Relationship sets
  - Attributes

# Entity Sets

- ✓ Any object in the real world is an entity
- ✓ For example , each faculty in an university is an entity
- ✓ An entity has a set of properties called attributes
- ✓ The values stored in one or more attributes will identify an entity uniquely in an entity sets
- ✓ For example , faculty\_id is an attribute hold a unique value of a faculty, similarly the student\_Register\_no is unique for all students

# Entity Sets

- ✓ An entity set is a set of entities of the same type that shares the same attributes.
- ✓ The set of people who are faculties at a given university, can be defined as entity set “faculty”
- ✓ Similarly the entity set “student” represent all the students in the university.
- ✓ The entity sets do not need to be disjoint.
- ✓ For example we can create an entity set called “person” can have faculty entity , student entity, both or neither.



# Attributes

- ✓ Attributes are descriptive properties possessed by each member of an entity set.
- ✓ Each entity is represented by a set of attributes.
- ✓ Each attribute of an entity set will store the similar information.
- ✓ Each entity must have its own value for each attribute.
- ✓ Possible attributes for faculty entity set are
  - faculty\_id ( unique )
  - faculty\_name
  - faculty\_dept
  - faculty\_salary
  - etc.,

# Values

- ✓ Each entity has a value for each attribute
- ✓ For instance , the particular faculty entity may have the following values :
  - faculty\_id = 100186
  - faculty\_name = ‘Nantha’
  - faculty\_dept = ‘Computing Technologies’
  - faculty\_salary = 123456
  - faculty\_mobile = 9999955555
- ✓ The faculty\_id attribute is used to identity the faculty uniquely , because there is a possibility for more number of faculties will have the same name
- ✓ In general the university use to assign unique id for faculty and students (Reg. No)

- ✓ A database for a university may include a number of entity sets.
- ✓ For example , to keeping track of faculty and students , the university also has the information about **courses**.
- ✓ The entity set has the following attributes
  - course\_id
  - course\_title
  - department\_id
  - credits
- ✓ In a real setting , university database may keep more number of entity sets.

## ENTITY SETS FACULTY AND STUDENT

100186	<u>Nantha</u>
100181	<u>Murugan</u>
100199	<u>Ganesh</u>
100201	<u>Senthil</u>
100210	<u>Pradeep</u>
100212	<u>Sivakumar</u>
100300	<u>Chirsty</u>

Entity set : Faculty

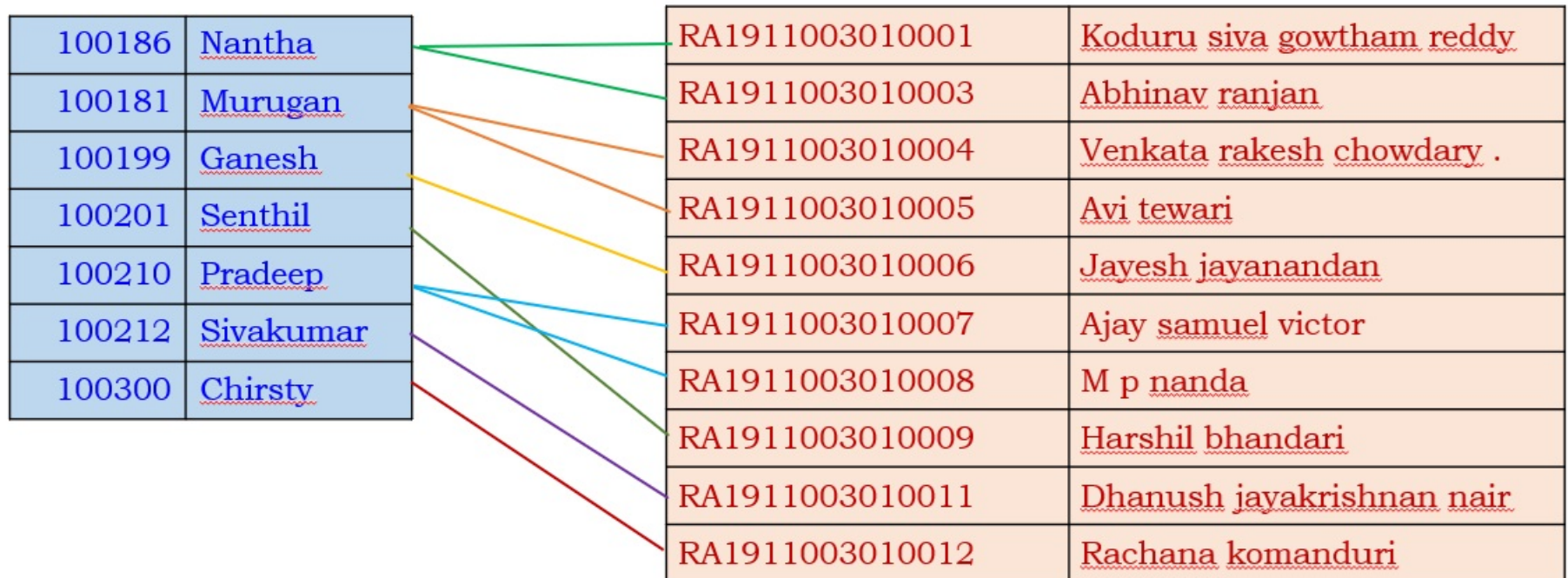
RA1911003010001	<u>Koduru siva gowtham reddy</u>
RA1911003010003	<u>Abhinav ranjan</u>
RA1911003010004	<u>Venkata rakesh chowdary .</u>
RA1911003010005	<u>Avi tewari</u>
RA1911003010006	<u>Jayesh jayanandan</u>
RA1911003010007	<u>Ajay samuel victor</u>
RA1911003010008	<u>M p nanda</u>
RA1911003010009	<u>Harshil bhandari</u>
RA1911003010011	<u>Dhanush jayakrishnan nair</u>
RA1911003010012	<u>Rachana komanduri</u>

Entity set : Student

## Relationship Sets

- ✓ A relationship is an association among several entities.
- ✓ For example , we can define a relationship counselor that associates faculty Nantha with the student Abhinav ranjan
- ✓ The relationship specifies that Nantha is a counselor to student Abhinav ranjan.
- ✓ A relationship set is a set of relationships of the same type.
- ✓ Formally, it is a mathematical relation on  $n \geq 2$  (possibly nondistinct) entity sets.
- ✓ If  $E_1, E_2, \dots, E_n$  are entity sets, then a relationship set  $R$  is a subset of  $\{(e_1, e_2, \dots, e_n) \mid e_1 \in E_1, e_2 \in E_2, \dots, e_n \in E_n\}$  where  $(e_1, e_2, \dots, e_n)$  is a relationship.

- ✓ Consider the two entity sets Faculty and Student.
- ✓ We define the relationship set counselor to denote the association between faculty and students.
- ✓ The following figure represents this association



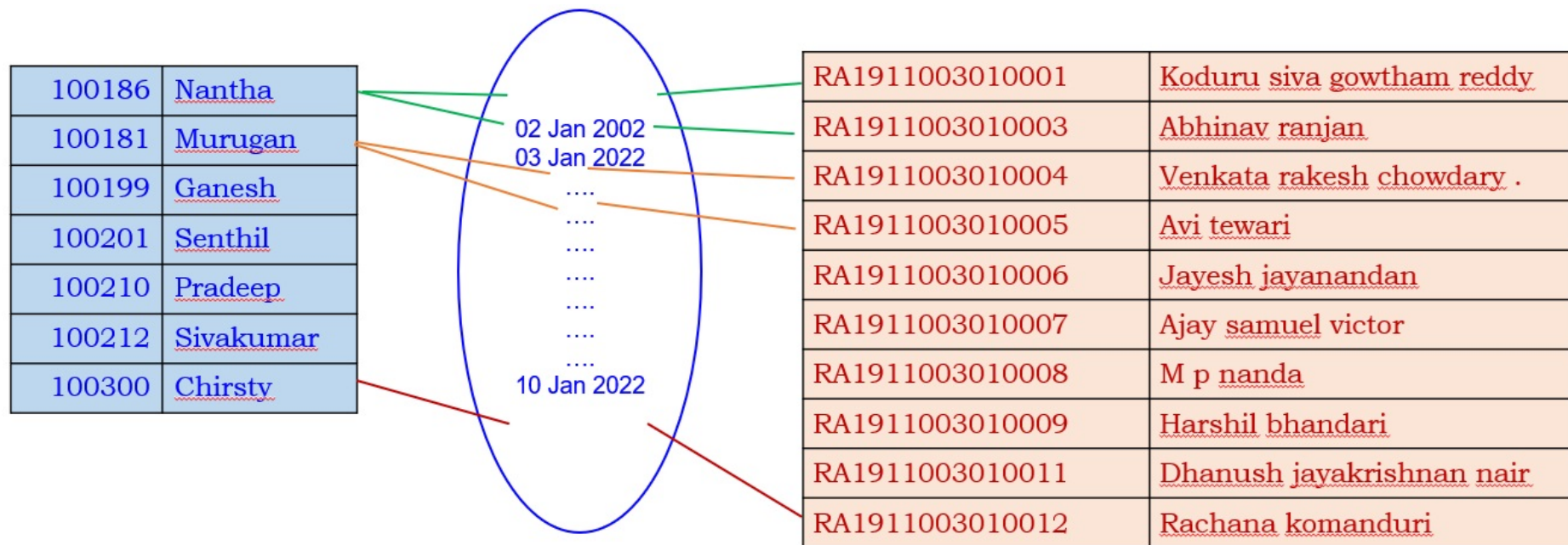
- ✓ The association between entity sets is referred to as participation.
- ✓ The entity sets  $E_1, E_2, \dots, E_n$  participate in relationship set  $R$ .
- ✓ A relationship instance in an E-R schema represents an association between the named entities in the real-world enterprise that is being modeled.
- ✓ To explain this, the individual faculty entity Nantha, who has faculty\_id 100186, and the student entity Abhinav ranjan who has student\_regno RA1911003010003 participate in a relationship instance counselor.
- ✓ This relationship instance represents that in the university, the faculty Nantha is counseling student Abhinav ranjan.

- ✓ The function that an entity plays in a relationship is called that entity's **role**.
- ✓ Since entity sets participating in a relationship set are generally distinct, roles are implicit and are not usually specified.
- ✓ The same entity set participates in a relationship set more than once, in different roles.
- ✓ In this type of relationship set, sometimes called a **recursive relationship set**, explicit role names are necessary to specify how an entity participates in a relationship instance.
- ✓ Example:
  - Consider the “course” entity set, which contains all about the courses offered in the university.
  - One course C2 , has a prerequisite course C1
  - The relationship set prereq that is modeled by pairs of course entities.
  - All relationships of prereq are characterized by (C1,C2) pairs, but (C2,C1) pairs are excluded



- ✓ A relationship may also have attributes called descriptive attributes.
- ✓ Consider a relationship set “counselor” with entity sets Faculty and Student.
- ✓ The attribute date can be associate with that relationship to specify the date when the faculty became the counselor of a student.
- ✓ The advisor relationship among the entities corresponding to faculty Nantha and student Abhinav ranjan has the value “3 Jan 2022” for attribute date, which means that Nantha became Abhinav ranjan’s counselor on 3 Jan 2022.

- ✓ The following figure shows the relationship set counselor with a descriptive attribute date.
- ✓ Faculty Nantha counsel two students with two different counseling dates.



## Binary relationship set

- ✓ One entity set involves in two entity sets is known as Binary relationship set.

### Example

- ✓ The faculty and student entity sets participate in relationship set counselor.
- ✓ In addition each student must have another faculty who works as department counselor ( Co-ordinator )
- ✓ Then the faculty and student entity sets may participate in another relationship set, dept counselor.

## Attributes

- ✓ For each attribute, there is a set of permitted values, called the domain, or value set, of that attribute.
- ✓ For example the domain attribute of student\_regno might be the set of all text strings of a certain length.
- ✓ Similarly the domain attribute of dept\_name might be strings from the set { CSE,IT, MECH,ECE, EEE, BT,....}
- ✓ An attribute of an entity set is a function that maps from the entity set into a domain.
- ✓ An entity set may have several attributes, Each entity is described by a set of ( Attribute, Data Value) Pairs.
- ✓ For example , A particular ,the Faculty entity may be described by a set { (faculty\_id, 100186), (faculty\_name, Nantha), (dept\_name, cse), (salary, 123456) }

# Attribute types

**Simple** : Values can not be divided into subparts  
Example : Faculty\_salary, Dept\_name, etc.,  
Attributes like salary, deptname can't be divided further

**Composite** : Values can be divided into subparts  
Example : Faculty\_name, Faculty\_address

Faculty\_name can be divided into first\_name,  
middle\_name, last\_name

Faculty\_address can be divided into Door\_no,  
Street\_name, City\_name, State\_name, Pincode

TYPES OF VALUES	DESCRIPTION
Single value	<ul style="list-style-type: none"><li>• Only one value can be stored</li><li>• Example : <u>Faculty id</u>, DOB</li></ul>
Multiple value	<ul style="list-style-type: none"><li>• More values are possible</li><li>• Example : <u>Faculty Phone no</u></li></ul>
Derived value	<ul style="list-style-type: none"><li>• The values which is derived from existing value</li><li>• Example : AGE</li><li>• The values keep on changing is not advisable to store in the database</li><li>• Normally the values will be derived from existing value of another attribute.</li><li>• AGE will be changing continuously.</li><li>• It can be derived from DOB ( DOB never change)</li></ul>
Null value	<ul style="list-style-type: none"><li>• NULL values are unknown undeclared</li><li>• An attribute does not have a value for a particular entity in an entity set</li></ul>

# Constraints

- ✓ An E-R enterprise schema may define certain constraints to which the contents of a database must conform.
- ✓ This is achieved using
  - Mapping Cardinalities
  - Participation Constraints

## Mapping Cardinalities

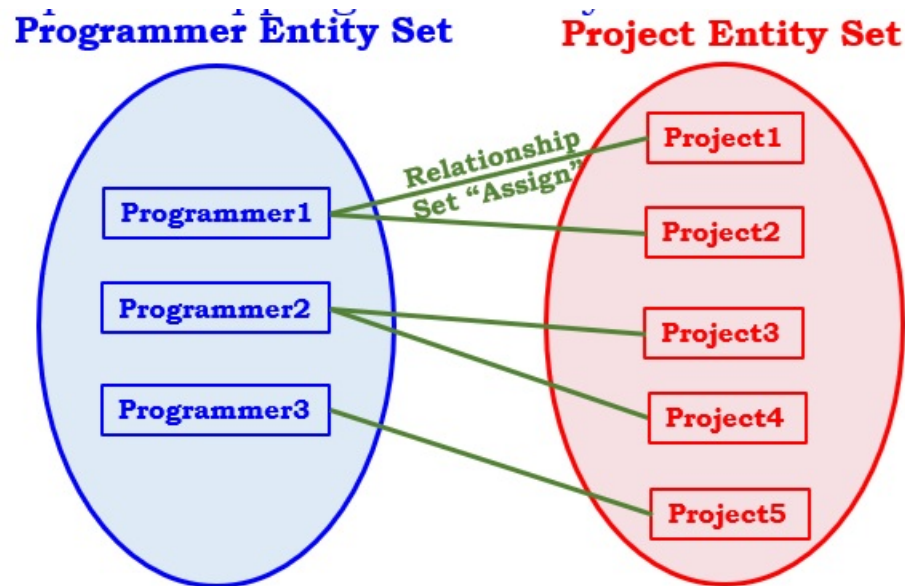
- ✓ Mapping cardinalities, or cardinality ratios, express the number of entities to which another entity can be associated via a relationship set.
- ✓ Mapping cardinalities are most useful in describing binary relationship sets.
- ✓ For a binary relationship set “Assign” between entity sets Programmer and Project the mapping cardinality must be one of the following.
  - One-to-One (1:1)
  - One-to-Many (1:M)
  - Many-to-One (M:1)
  - Many-to-Many (M:M)



# Mapping Cardinalities

## One-to-Many (1:M)

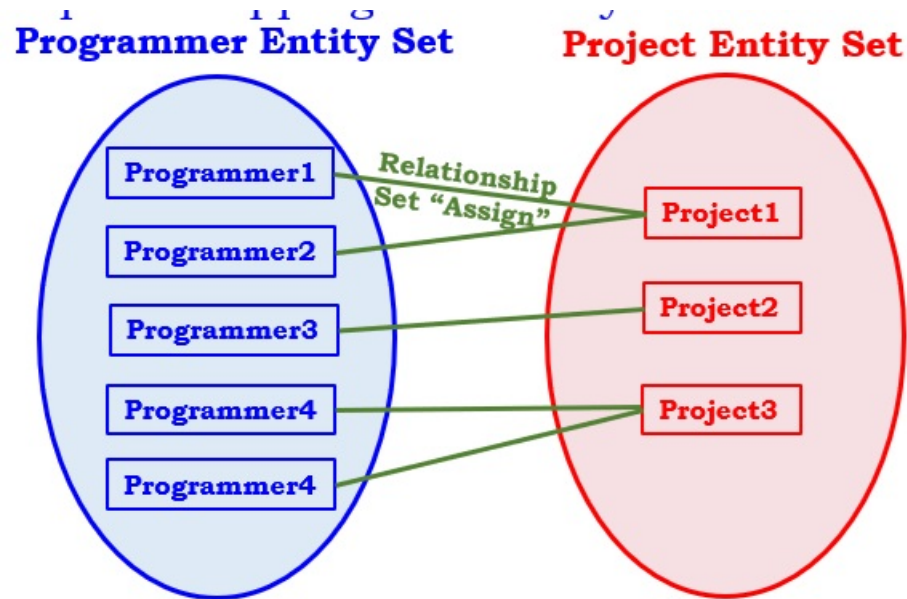
- ✓ One-to-many. An entity in Programmer is associated with any number (zero or more) of entities in Project. An entity in Project, however, can be associated with at most one entity in Programmer.
- ✓ The following figure depicts mapping cardinality 1:M



# Mapping Cardinalities

## Many-to-One (M:1)

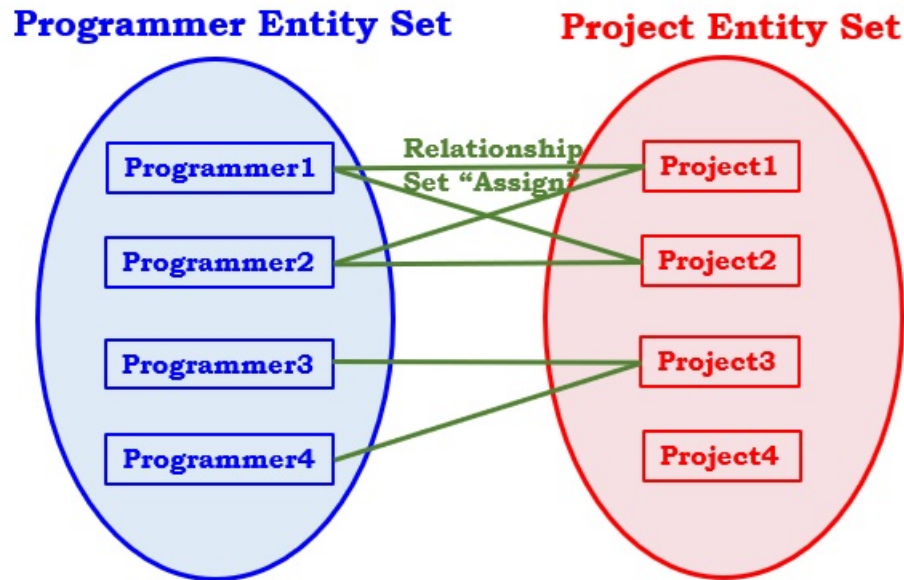
- ✓ An entity in Programmer is associated with at most one entity in Project. An entity in Project, however, can be associated with any number (zero or more) of entities in Programmer.
- ✓ The following figure depicts mapping cardinality M:1



# Mapping Cardinalities

## Many-to-Many (M:M)

- ✓ An entity in Programmer is associated with any number (zero or more) of entities in Project, and an entity in Project is associated with any number (zero or more) of entities in Programmer.
- ✓ The following figure depicts mapping cardinality M:M



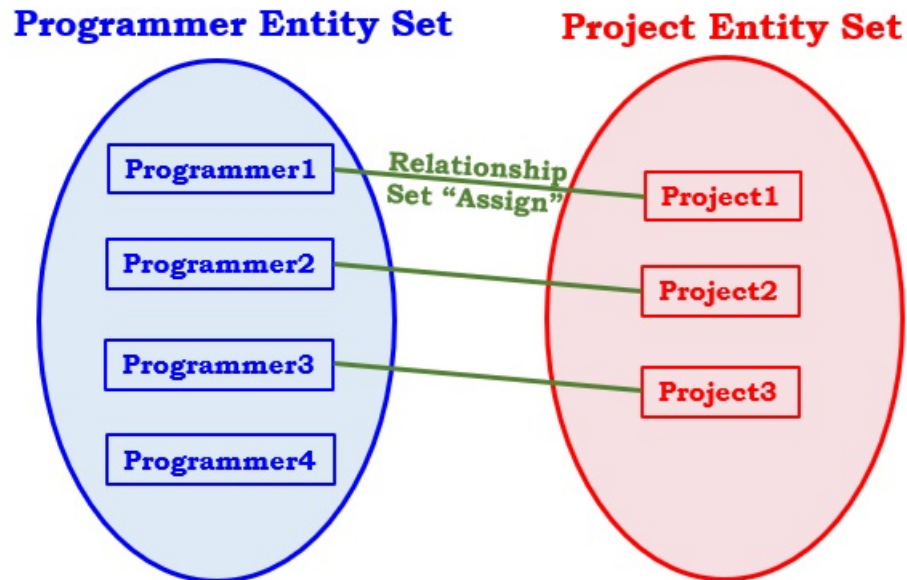
# Participation Constraints

## Total Participation :

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

## Partial Participation :

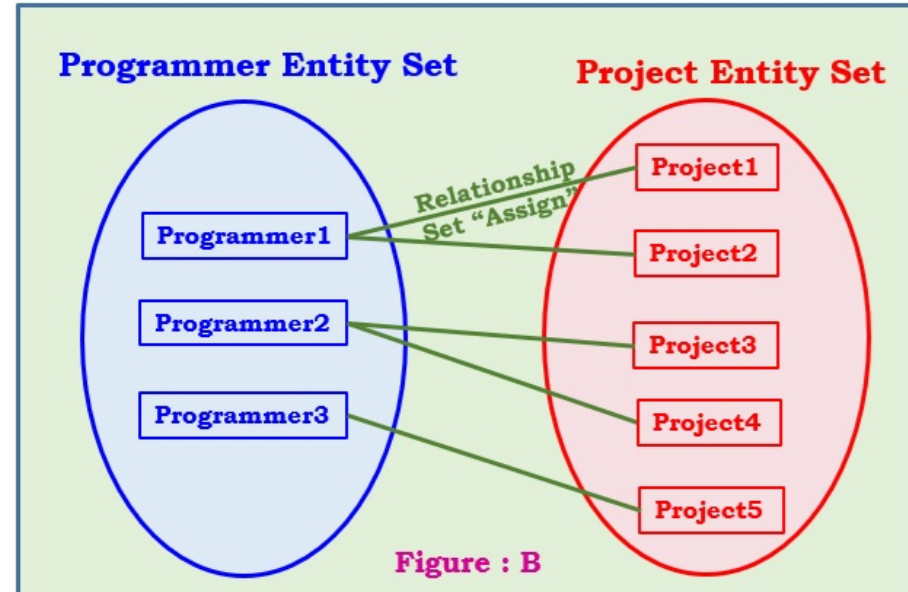
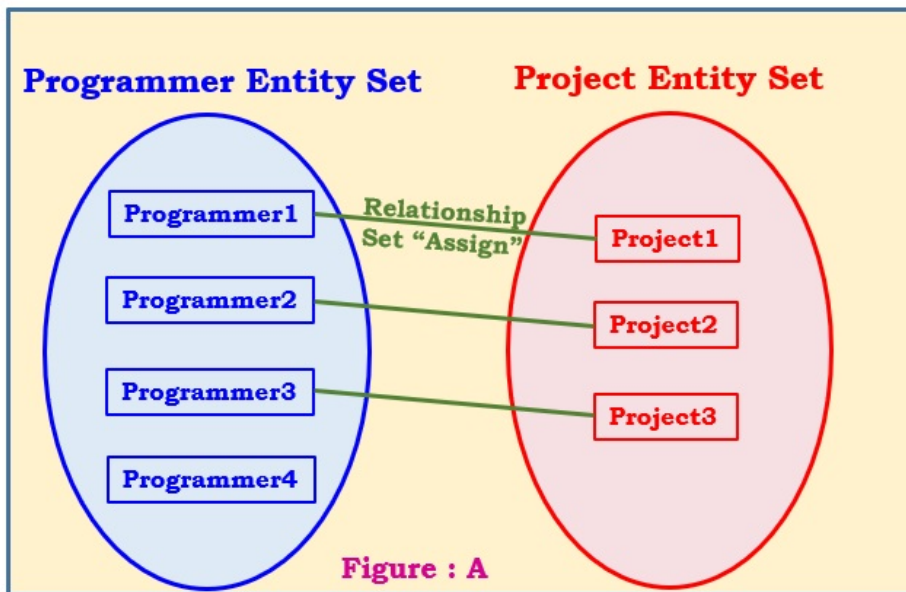
If only some entities in E participate in relationships in R, the participation of entity set E in relationship R is said to be partial.



# Participation Constraints

Example :

- ✓ In Figure : A, the participation of Project Entity Set in the relationship set is total while the participation of programmer entity set in the relationship set is partial.
- ✓ In Figure : B, the participation of both Programmer Entity Set and Project Entity Set in the relationship set are total.



**ER diagram**

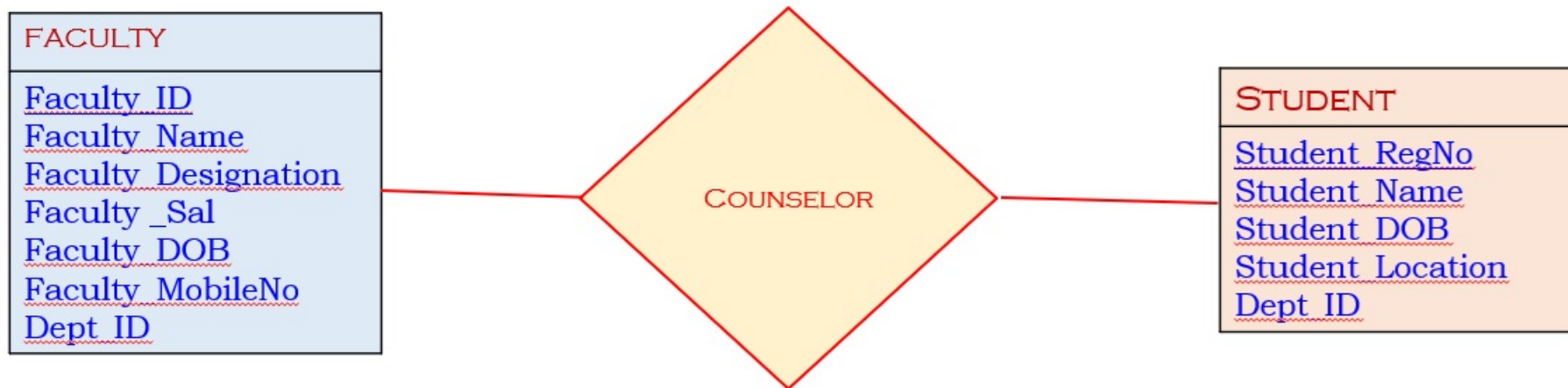
- ✓ E-R diagram can express the overall logical structure of a database graphically.
- ✓ E-R diagrams are simple and easy to understand

### Basic Structure

E-R Diagram consists of following major components

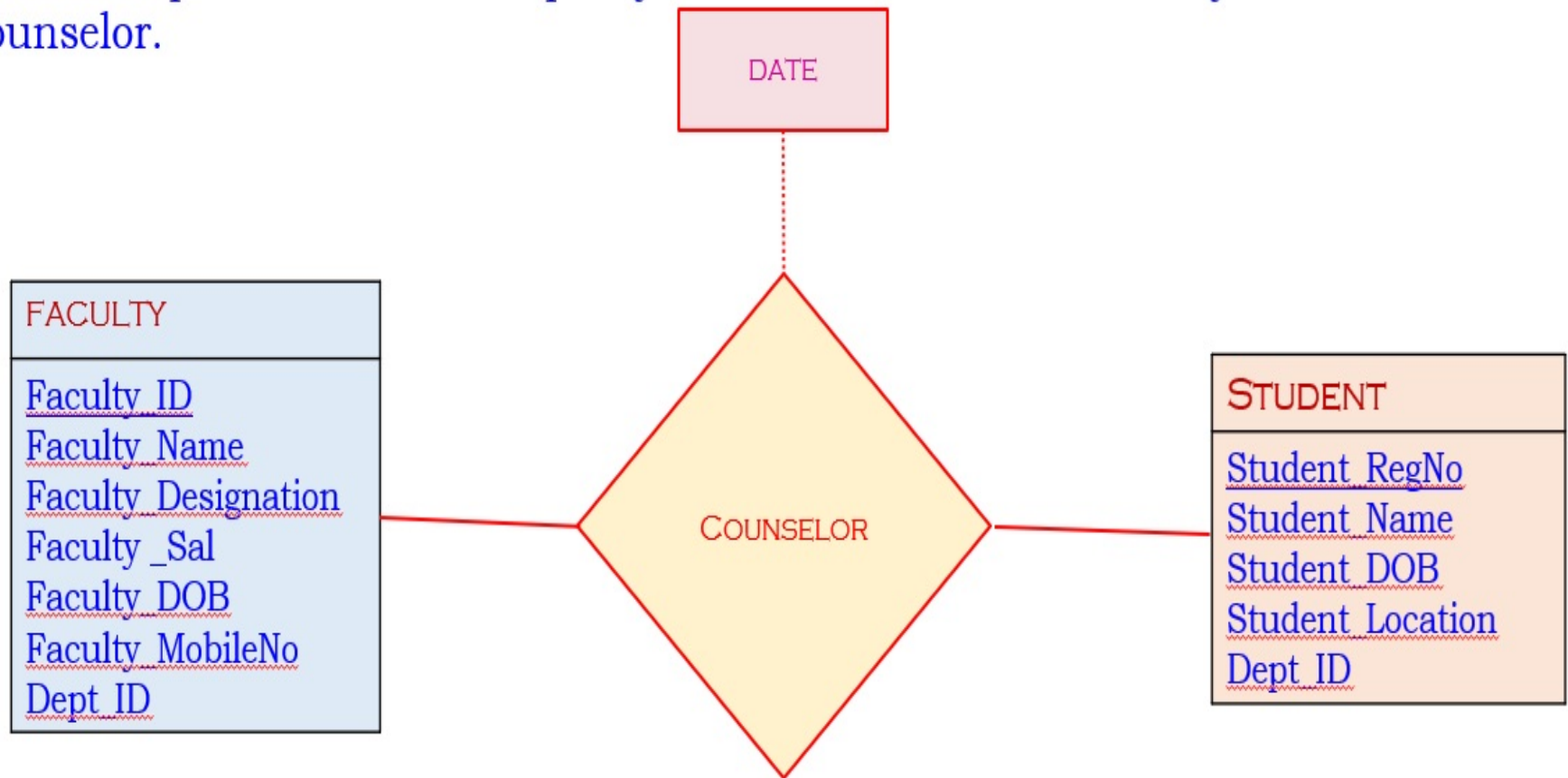
- ✓ **Rectangles** divided into two parts represent entity sets. The first part contains the name of the entity set. The second part contains the names of all the attributes of the entity set.
- ✓ **Diamonds** represent relationship sets.
- ✓ **Undivided rectangles** represent the attributes of a relationship set. Attributes that are part of the primary key are underlined.
- ✓ **Lines** link entity sets to relationship sets.
- ✓ **Dashed lines** link attributes of a relationship set to the relationship set.
- ✓ **Double lines** indicate total participation of an entity in a relationship set.
- ✓ **Double diamonds** represent identifying relationship sets linked to weak entity sets

- ✓ Consider the E-R diagram in following figure, which consists of two entity sets, faculty and student related through a binary relationship set counselor.
- ✓ The attributes associated with faculty are Faculty\_ID, Faculty\_Name, Faculty\_Designation, Faculty\_Sal, Faculty\_DOB, Faculty\_MobileNo, Dept\_ID
- ✓ The attributes associated with student are Student\_RegNo, Student\_Name, Student\_DOB, Student\_Location, Dept\_ID
- ✓ Attributes of an entity set that are members of the primary key are underlined.





- ✓ If a relationship set has some attributes associated with it, then we enclose the attributes in a rectangle and link the rectangle with a dashed line to the diamond representing that relationship set.
- ✓ For example, in the given figure, the date descriptive attribute attached to the relationship set counselor to specify the date on which the faculty became the counselor.

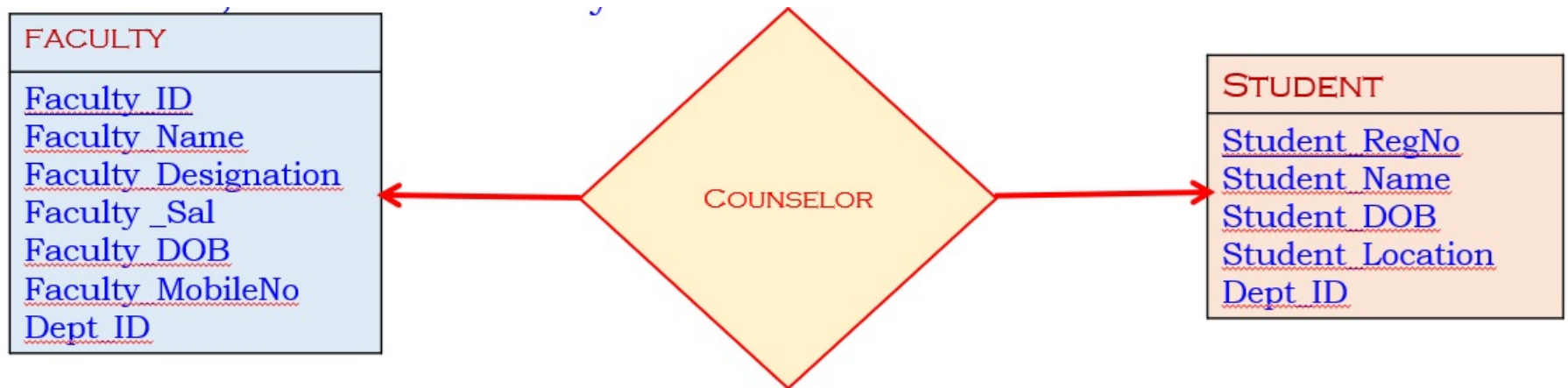


## Mapping Cardinality

- ✓ The relationship set counselor, between the faculty and student entity sets may be one-to-one, one-to-many, many-to-one, or many-to-many.
- ✓ To distinguish among these types, we draw either a directed line (  $\rightarrow$  ) or an undirected line ( — ) between the relationship set and the entity.

### One-to-one:

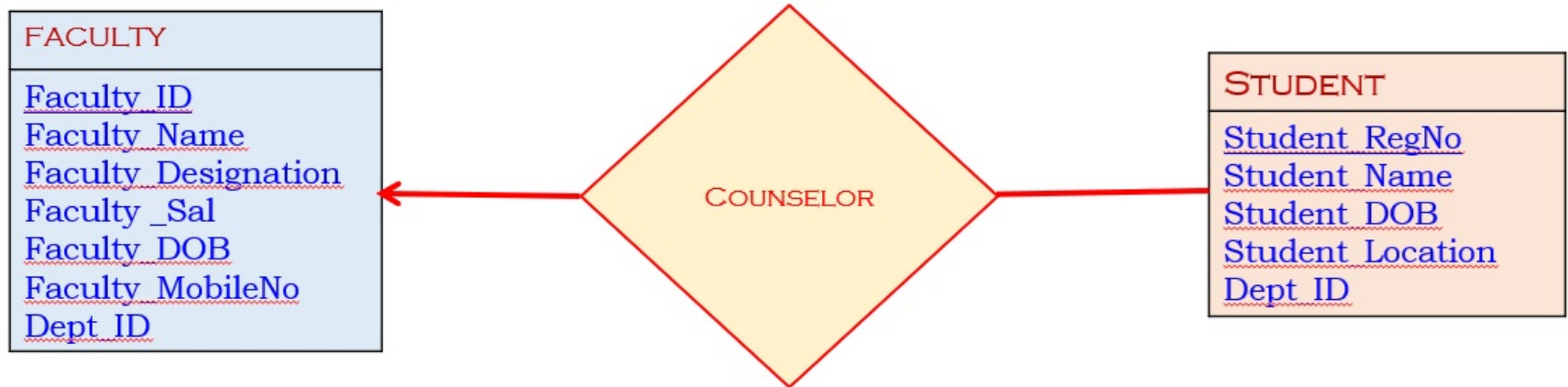
Line from the relationship set counselor to both entity sets faculty and student as given in the figure below. This indicates that a faculty may counsel at most one student, and a student may have at most one counselor.



# Mapping Cardinality

## One-to-many:

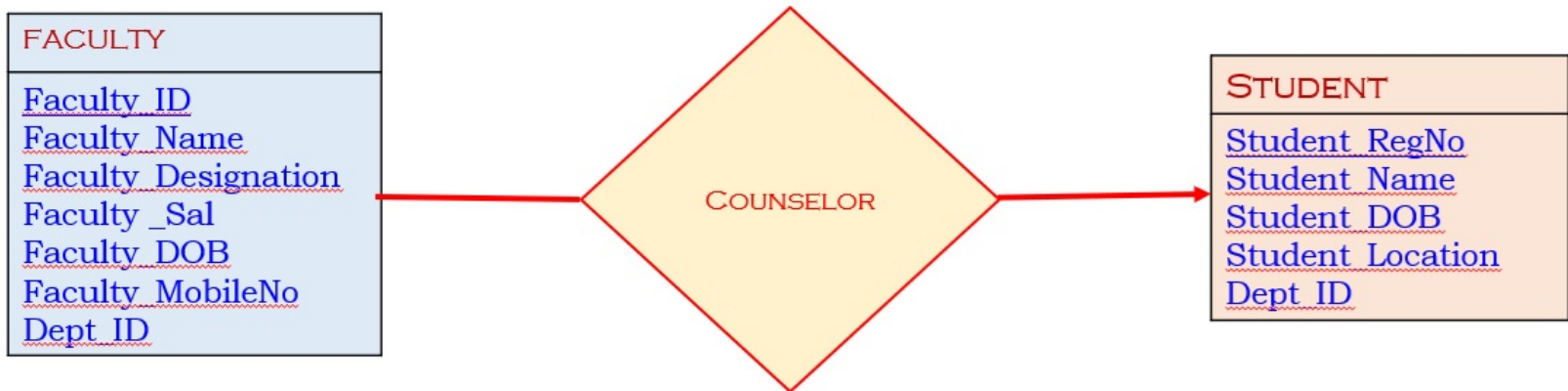
A directed line from the relationship set counselor to the entity set faculty and an undirected line to the entity set student as shown in the below figure, indicates that a faculty may counsel many students, but a student may have at most one counselor.



# Mapping Cardinality

## Many-to-one:

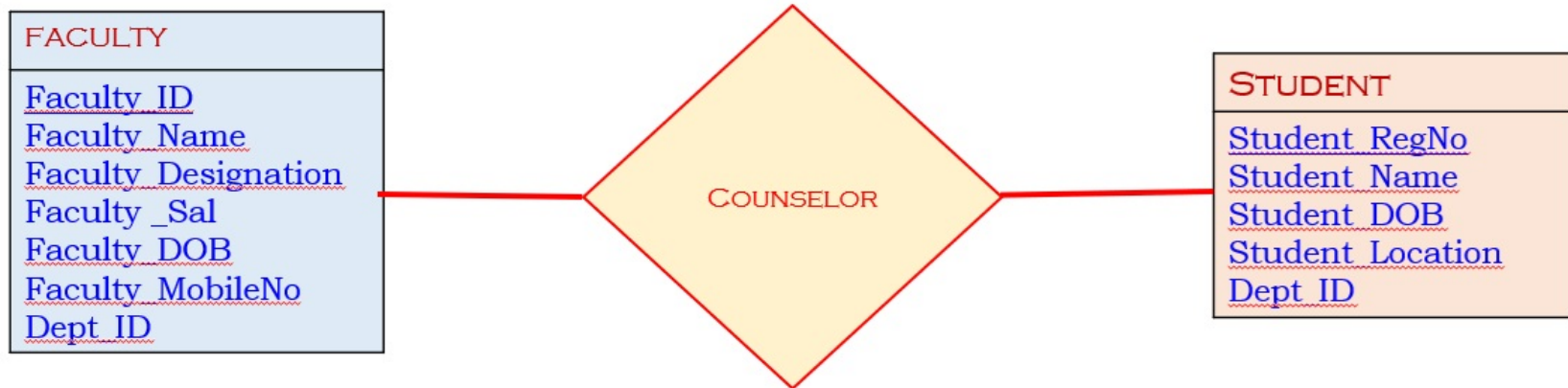
An undirected line from the relationship set counselor to the entity set faculty and a directed line to the entity set student as shown in the below figure, indicates that a faculty may counsel at most one student, but a student may have many counselors.



# Mapping Cardinality

Many-to-many:

- ✓ We draw an undirected line from the relationship set counselor to both entity sets faculty and student as shown in the below figure, indicates that a faculty may counsel many students, and a student may have many counselor.



# Complex Attributes

- ✓ Figure shows how composite attributes can be represented in the E-R notation.
- ✓ Here, a composite attribute Faculty\_name, with component attributes Faculty\_first\_name, Faculty\_middle\_name, and Faculty\_last\_name replaces the simple attribute name of Faculty.
- ✓ As another example, An address to the Faculty entity-set. The address can be defined as the composite attribute Faculty\_address with the attributes street, city, state, and pincode.
- ✓ The attribute street is itself a composite attribute whose component attributes are Faculty\_street\_no and Faculty\_street name.
- ✓ The given figure also illustrates a multivalued attribute phone number, denoted by “{ Faculty\_phone\_no }”.
- ✓ A derived attribute age, depicted by a “Faculty\_age ( )”.

## Faculty

Faculty\_id

Faculty\_name

Faculty\_first\_name

Faculty\_middle\_name

Faculty\_last\_name

Faculty\_address

Faculty\_address\_doorno

Faculty\_address\_street

Street\_no

Street\_name

Faculty\_address\_city

Faculty\_address\_state

Faculty\_address\_pincode

{Faculty\_phone\_no}

Faculty\_DOB

Faculty\_age ( )

**KEYS**



# Types of Keys

- In a database, various types of keys are used to uniquely identify records and establish relationships between tables.

Here are the key types:

1.Primary Key

2.Foreign Key

3.Candidate Key

4.Super key

5.Composite Key

6.Alternate Key

7.Unique Key

# Primary Key

- A unique identifier for a record in a table. No two rows can have the same primary key value, and it cannot contain “NULL” values.
- Example: In an "Employees" table, an "Employee\_ID" could be the primary key.

# Foreign Key

- A field (or collection of fields) in one table that uniquely identifies a row of another table, creating a link between the two.
- Example: In an "Orders" table, "Customer\_ID" could be a foreign key referencing the "Customer\_ID" in a "Customers" table.

# Candidate Key

- A column (or set of columns) that can uniquely identify any record in the table. A table can have multiple candidate keys, but only one is chosen as the primary key.
- Example: In a “Students” table, both “Student\_ID” and “Email” could be candidate keys.

# Super Key

- A set of one or more columns that can uniquely identify a record in the table.
- All candidate keys are super keys, but super keys may include unnecessary attributes.
- Example: In an "Employees" table, {"Employee\_ID", "Email"} is a super key, but "Employee\_ID" alone can also be a candidate key.

# Composite Key

- A key that consists of two or more columns to uniquely identify a row when no single column is sufficient.
- Example: In a "Course\_Enrollments" table, a combination of "Student\_ID" and "Course\_ID" could form a composite key.

# Alternate Key

- Any candidate key that is not chosen as the primary key.
- Example: If "Student\_ID" is the primary key, "Email" would be the alternate key in the "Students" table.

# Unique Key

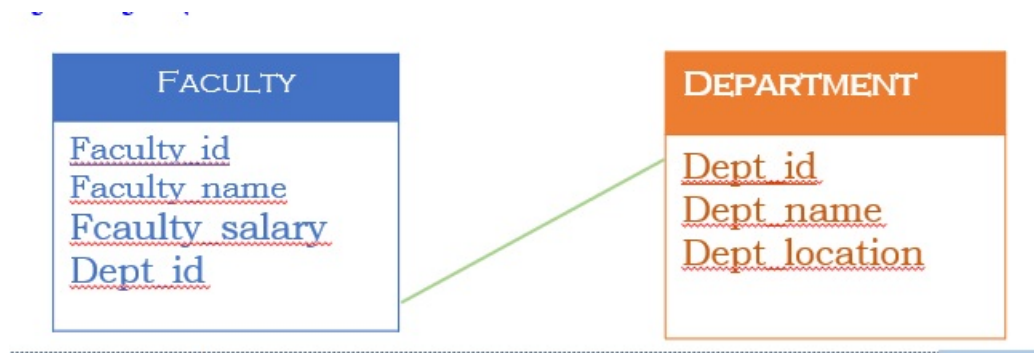
- A constraint that ensures all values in a column or a set of columns are distinct. It allows one NULL value , unlike the primary key.
- Example: A "Phone\_Number" column in a "Contacts" table might be a unique key.

These keys help maintain data integrity, ensure unique identification, and define relationships between tables in relational databases.



# PRIMARY key

- ✓ Consider the two entity sets named : Faculty and Department
- ✓ For Faculty entity set the primary key is : Faculty\_id
- ✓ For Department entity set the primary key is : Dept\_id
- ✓ In this relations, Dept\_id in the Department relation , is the referential key or foreign key for the Faculty relation.
- ✓ Primary key in a relations is underlined
- ✓ Only one primary key is possible for a relation
- ✓ One or more attributes can be combined and declared as a primary key , known as composite primary key. ( **Note** : Maximum 16 Columns are allowed )



# Attributes

- ✓ Attributes are the properties of an entity
- ✓ Attributes are used to describe about an entity
- ✓ The type of attributes are
  - Simple attributes
  - Composite attributes
  - Single valued attributes
  - Multi valued attributes
  - Derived attributes
  - Key attributes

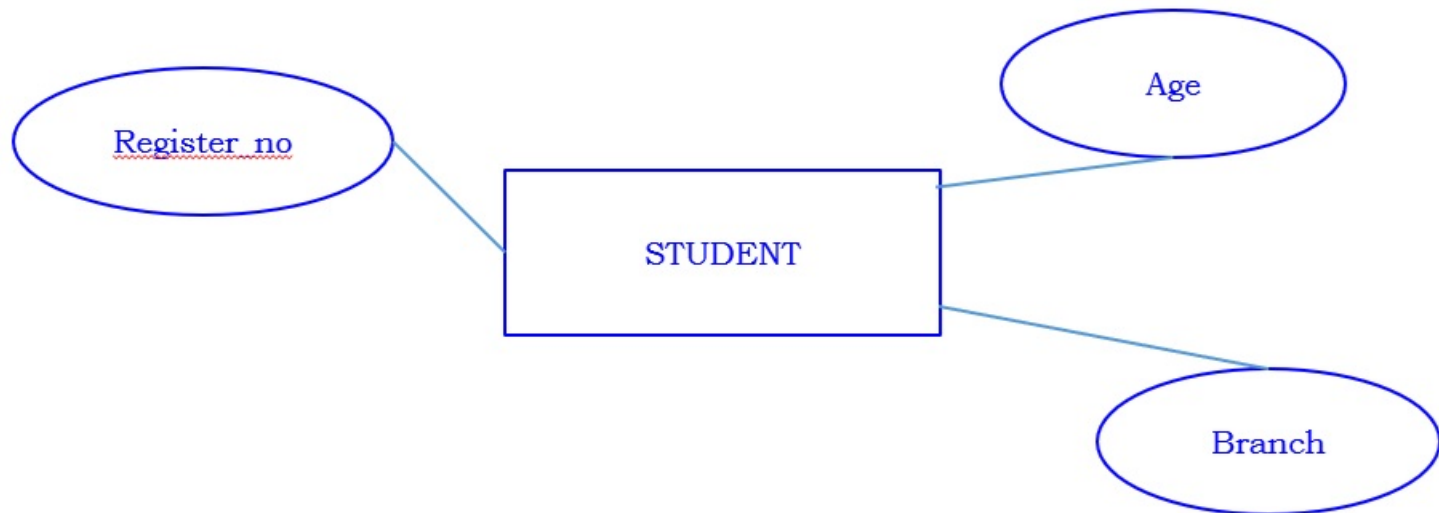
# Simple attributes

It can not be divided further

All the simple attributes will hold the atomic values

Example :

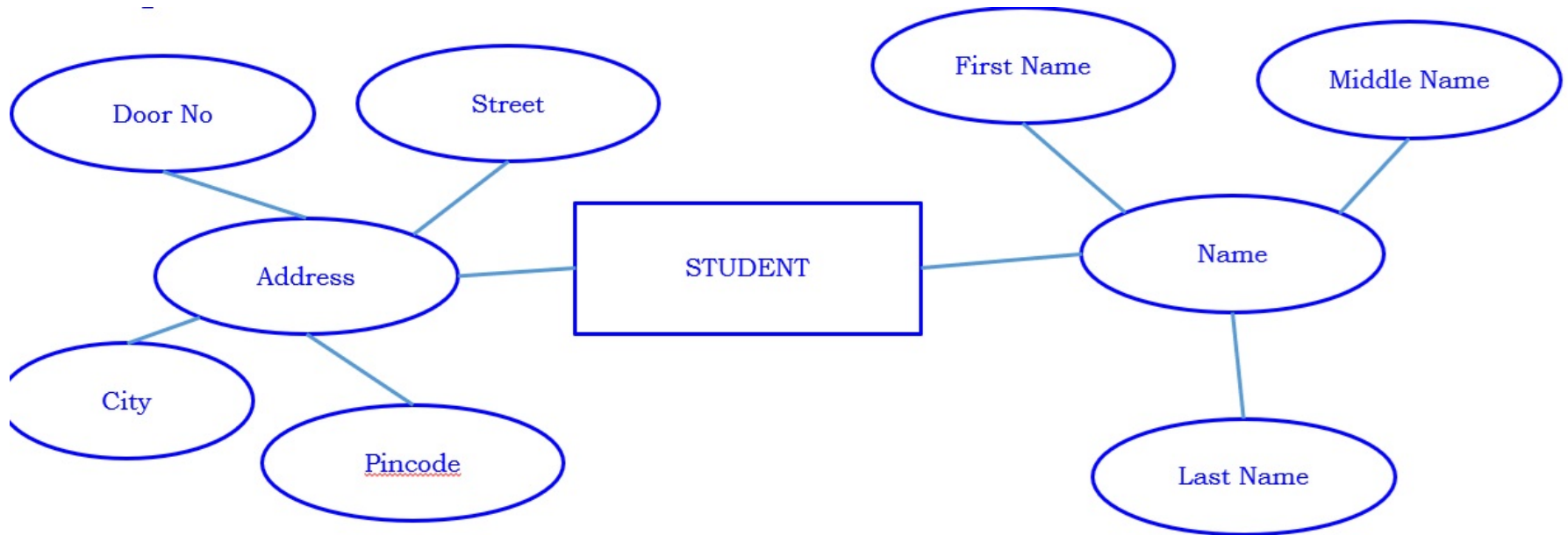
Student = { Register\_no, Name, ..... }



# Composite attributes

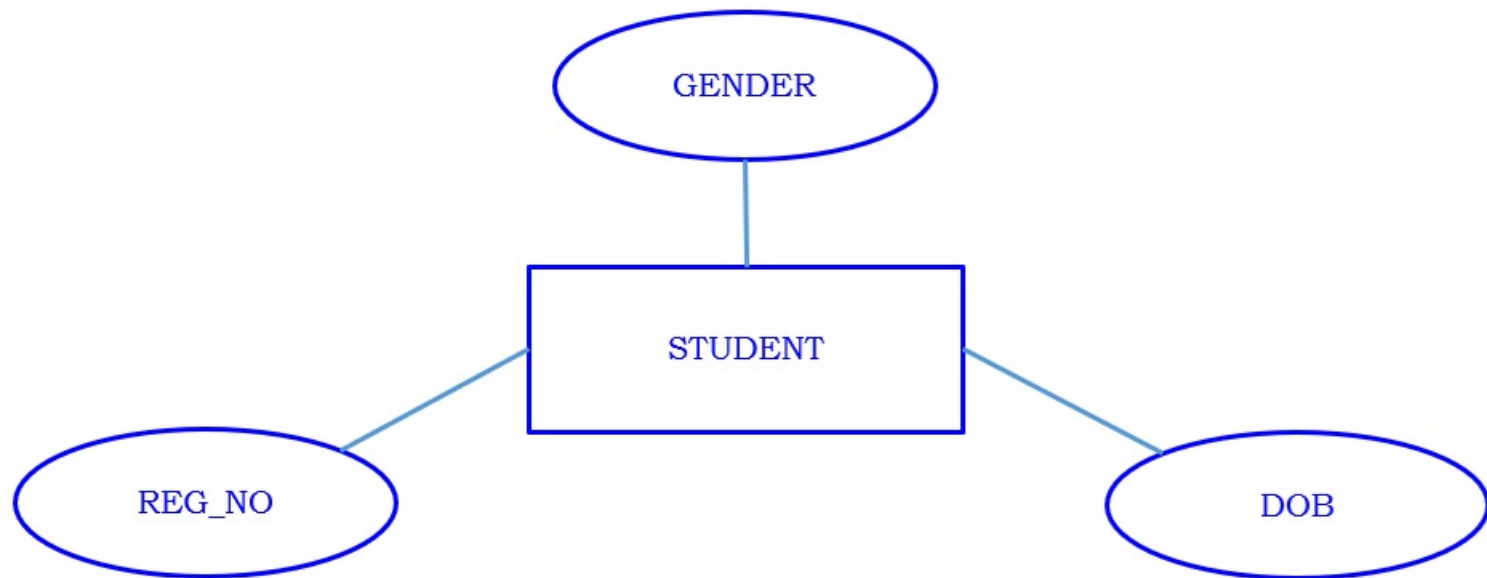
Composed by many other simple attributes

Example : Address , Name , etc.,



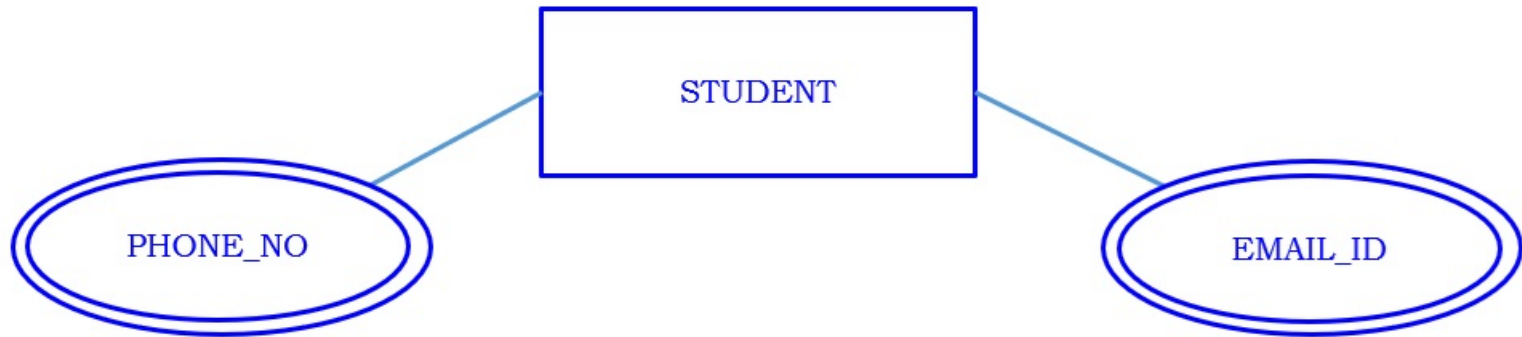
## Single valued attributes

- ✓ Single valued attributes are those attributes which can take only one value for a given entity from an entity set.
- ✓ Example : Gender , DOB, Reg\_No



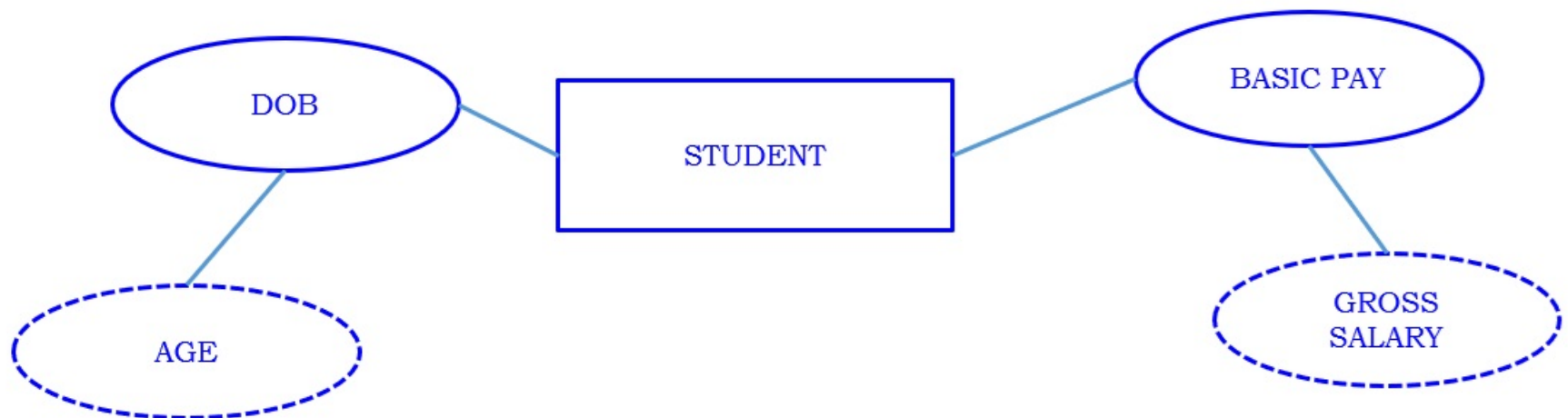
## Multi valued attributes

- ✓ Attributes can hold more than one values are called multi valued attribute
- ✓ Example : Phone\_no, Email\_id



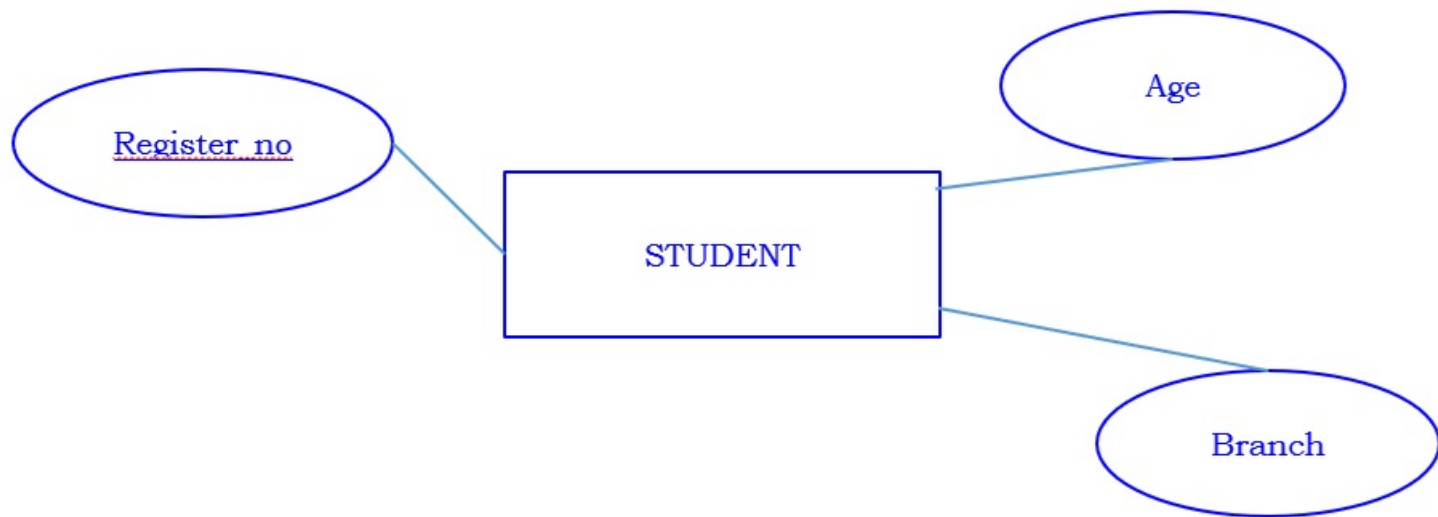
## Derived attributes

- ✓ A value which is derived from already existing value.
- ✓ It is not advisable to store such kind of values in database.
- ✓ The derived attributes represented by ellipse using dotted lines
- ✓ Example : Age , Gross Salary
- ✓ In the given figure below, Age is derived from DOB and Gross Salary derived from Basic Pay



## Key attributes

- ✓ Attributes which is used to identify an entity in an entity set is called Key attributes
- ✓ Key attributes are represented by underline the name of the attribute.
- ✓ In the given figure , In Student entity the attribute Register\_no is key attribute used to identity each student uniquely.





## ✓ Constraints

- ✓ It is a condition to manage the consistency as well integrity of the values stored in an attribute.
- ✓ Constraints specified at the time of designing relations is good choice
- ✓ There are two types of Constraints
  - ✓ Domain Constraints
    - Not Null
    - Check
    - Unique
    - Primary key
  - ✓ Integrity Constraints
    - Referential key or Foreign key

# Domain Constraints

## Not Null :

( **NOTE** : By default ,an attribute hold NULL values )

If an attribute holds not null constraint

- ✓ The value should be inserted
- ✓ It will not accept “NULL” values
- ✓ It will accept Duplicate values
- ✓ N number of not null constraints is possible in a relation
- ✓ While inserting a new record the not null must be entered otherwise , insertion of new record is not possible
- ✓ Example : Student entity defined with not null constraint for an attribute Register\_no

```
CREATE TABLE STUDENT (  
    Register_no Number(10) NOT NULL,  
    LastName varchar(25) ,  
    FirstName varchar(25),  
    DOB Date );
```

# Domain Constraints

## Check :

- ✓ Check Constraints check the condition specified in the create statement.
- ✓ If the condition satisfied then the value will be inserted , otherwise will not be permitted.
- ✓ It allows NULL values
- ✓ It allows duplicate values
- ✓ Example : The emp entity created with check constraint for an attribute “Salary” should be greater than 10000.

```
CREATE TABLE emp ( empno number (10) Not null,  
                    Ename varchar2(25),  
                    ..... ,  
                    ..... ,  
                    Salary number(10,2) Check (Salary > 10000);
```

# Domain Constraints

## Unique:

- ✓ To maintain the distinct values in an attribute of an entity set ,UNIQUE constraint is used.
- ✓ It will not accept duplicate values.
- ✓ It will accept NULL values .
- ✓ It will accept N number of null values , because two null values are always not equal.
- ✓ A relation can have N number of unique constraints.
- ✓ Example : A Student entity is created with unique constraint for an attribute Register\_no

```
CREATE TABLE STUDENT (  
    Register_no Number(10) Unique,  
    LastName varchar(25) ,  
    FirstName varchar(25),  
    DOB Date );
```

**Note : An attribute can hold one or more**

```
CREATE TABLE STUDENT (  
    Register_no Number(10) Not null Unique,  
    LastName varchar(25),FirstName varchar(25)
```

## Domain Constraints

### Primary key

- ✓ Minimal of super key is known as Candidate key.
- ✓ Candidate key represented as PRIMARY KEY
- ✓ A relation can have only one primary key
- ✓ Combination of one or more ( Maximum 16 Nos ) attributes can be declared as primary key.
- ✓ It will not accept both null values and duplicate values.
- ✓ Primary key is the combination of Not null and Unique constraints.
- ✓ Primary key can act as a referential key for another table called child table.
- ✓ Example: A Student entity created with primary key constraint for an attribute Register\_no

```
CREATE TABLE STUDENT (  
    Register_no Number(10) Primary key,  
    LastName varchar(25) ,  
    FirstName varchar(25),  
    DOB Date );
```

## Integrity Constraints

### Referential Integrity / Foreign key Constraints

- ✓ A primary key will be a referential key for another table is called as referential integrity / foreign key constraints.
- ✓ Foreign key allows only the values available in referential key ( Primary key).
- ✓ It allows duplicate values and null values.
- ✓ It allows N number of null values.
- ✓ Example : An entity emp created with foreign key constraint referencing dept entity primary key attribute dept\_id.

```
CREATE TABLE emp ( empno number (10) Primary key,  
                    Ename varchar2(25),  
                    .....  
                    .....  
                    Salary number(10,2) Check (Salary > 10000),  
                    Dept_id references DEPT (DEPT_ID);
```

Note : The given emp entity , primary key attribute is empno and foreign key is dept\_id which is the primary key in dept entity.

## An overview of Constraints

CONSTRAINTS	NULL VALUES	DUPLICATE VALUES	CHECKING THE CONDITION	REFERENTIAL KEY
NOT NULL	NO	YES	YES	NO
CHECK	YES	YES	YES	NO
UNIQUE	YES	NO	YES	NO
PRIMARY KEY	NO	NO	YES	YES
FOREIGN KEY	YES	YES	YES	NO

# Relational Model



- ✓ The relational model is today the primary data model for commercial data processing applications.
- ✓ It attained its primary position because of its simplicity, which eases the job of the programmer / developer.
- ✓ It is simple and easy to understand compared to earlier data models such as the network model or the hierarchical model.
- ✓ The followings should be consider for Relational Model
  - Structure of Relational Databases
  - Database Schema
  - Keys
  - Schema Diagrams
  - Relational Query Languages
  - Relational Operations

# Structure of Relational Databases

- ✓ A relational database consists of a collection of tables.
- ✓ Each table will have a unique name (unique identification)
- ✓ For example, consider the faculty table in the given figure, which stores information about faculty.
- ✓ This table contains four attributes (columns) named faculty\_id, faculty\_name, dept\_name and salary

<u>FACULTY ID</u>	<u>FACULTY NAME</u>	<u>DEPT NAME</u>	SALARY
100186	Nantha	CSE	12345
100181	Murugan	DSBS	23456
100199	Ganesh	DSBS	12456
100201	Senthil	CSE	34213
100210	Pradeep	BT	23457
100212	Sivakumar	MECH	12567
100300	Chirsty	ECE	23425

# Structure of Relational Databases

- ✓ Consider the following table Course, which stores the information about course details like course\_code, title, dept\_name, credits

<u>COURSE_CODE</u>	TITLE	<u>DEPT_NAME</u>	CREDITS
18CSC303J	Database Management Systems	Computing Technology	4
18CSE456T	Distributed Operating System	Computing Technology	3
18CSE390T	Computer Vision	Data Science and Business Systems	3
18CSC205J	Operating Systems	Data Science and Business Systems	4
18CSE344T	Cloud Architecture	Networking and Communications	3
18CSC305J	Artificial Intelligence	Computing Intelligence	4
18CSE459T	Service Oriented Architecture	Computing Intelligence	3

## Structure of Relational Databases

- ✓ Consider the table, prereq, which stores the prerequisite courses for each course.
- ✓ The table has two attributes, course\_code and prereq\_code.

<u>COURSE_CODE</u>	<u>PREREQ_CODE</u>
18CSC303J	18CSC161J
18CSE456T	18CSC205J
18CSE390T	18CSE353T
18CSC205J	18CSC161J
18CSE344T	18CSE378T
18CSC305J	18CSE388T
18CSE459T	18CSC302J

## Structure of Relational Databases

- ✓ A row in a table represents a relationship among a set of values.
- ✓ A table is a collection of such relationships,
- ✓ In mathematical terminology, a tuple is simply a sequence (or list) of values.
- ✓ A relationship between  $n$  values is represented mathematically by an  $n$ -tuple of values, i.e., a tuple with  $n$  values, which corresponds to a row in a table.
- ✓ In relational model the term relation is used to refer to a table
- ✓ The term tuple is used to refer to a row.
- ✓ The term attribute refers to a column of a table.
- ✓ For each attribute of a relation, there is a set of permitted values, called the Domain of that attribute.
- ✓ The domains of all attributes of relation be atomic.
- ✓ The null value is a special value that signifies that the value is unknown or does not exist.

## Database Schema

- ✓ The database schema, which is the logical design of the database.
- ✓ Database instance, which is a snapshot of the data in the database at a given instant in time.
- ✓ The concept of a relation corresponds to the programming-language notion of a variable.
- ✓ The concept of a relation schema corresponds to the programming-language notion of type definition.
- ✓ A relation schema consists of a list of attributes and their corresponding domains.

## Database Schema

✓ Consider the Department relation

<u>DEPT_NAME</u>	LOCATION	BUDGET
Computing Technologies	<u>Techpark</u>	7000000
Networking and Communication	<u>Techpark</u>	4000000
Computing Intelligence	University Building	6000000
Data Science and Business Systems	University Building	3000000
<u>Mechatronics</u>	<u>Hitech</u>	3500000
Electrical Engineering	Main Building	2000000

✓ The schema for that relation is  
department (dept\_name, location, budget)

## Database Schema

- ✓ Consider the university database example
- ✓ Each course in a university may be offered multiple times, across different semesters, or even within a semester.
- ✓ A relation to describe each individual offering, or section, of the class.
- ✓ The schema is:
  - section (course\_code, sec id, semester, year, location, room number, time slot id)
- ✓ To describe the association between faculty and the class sections that they teach.
  - teaches (faculty\_id, course id, sec id, semester, year)



# Database Schema

## Section relation

<u>COURSE_CODE</u>	<u>SEC_ID</u>	SEMESTER	YEAR	LOCATION	<u>ROOM_NO</u>	<u>TIME_SLOT_ID</u>
18CSC303J	A1	EVEN	2022	<u>Techpark</u>	TP801	A
18CSE456T	A1	ODD	2021	<u>Techpark</u>	TP706	B
18CSE390T	B1	EVEN	2022	University Building	UB4001	C
18CSC205J	B1	EVEN	2022	University Building	UB5002	B
18CSE344T	B1	ODD	2021	<u>Techpark</u>	TP403	D
18CSC305J	A1	ODD	2021	University Building	UB1201	E
18CSE459T	A1	EVEN	2022	University Building	UB1210	G

# Database Schema

## Teaches Relation

<u>FACULTY_ID</u>	<u>COURSE_CODE</u>	<u>SEC_ID</u>	SEMESTER	YEAR
100186	18CSC303J	A1	EVEN	2022
100181	18CSE456T	A1	ODD	2021
100199	18CSE390T	B1	EVEN	2022
100201	18CSC205J	B1	EVEN	2022
100210	18CSE344T	B1	ODD	2021
100212	18CSC305J	A1	ODD	2021
100300	18CSE459T	A1	EVEN	2022

## Database Schema

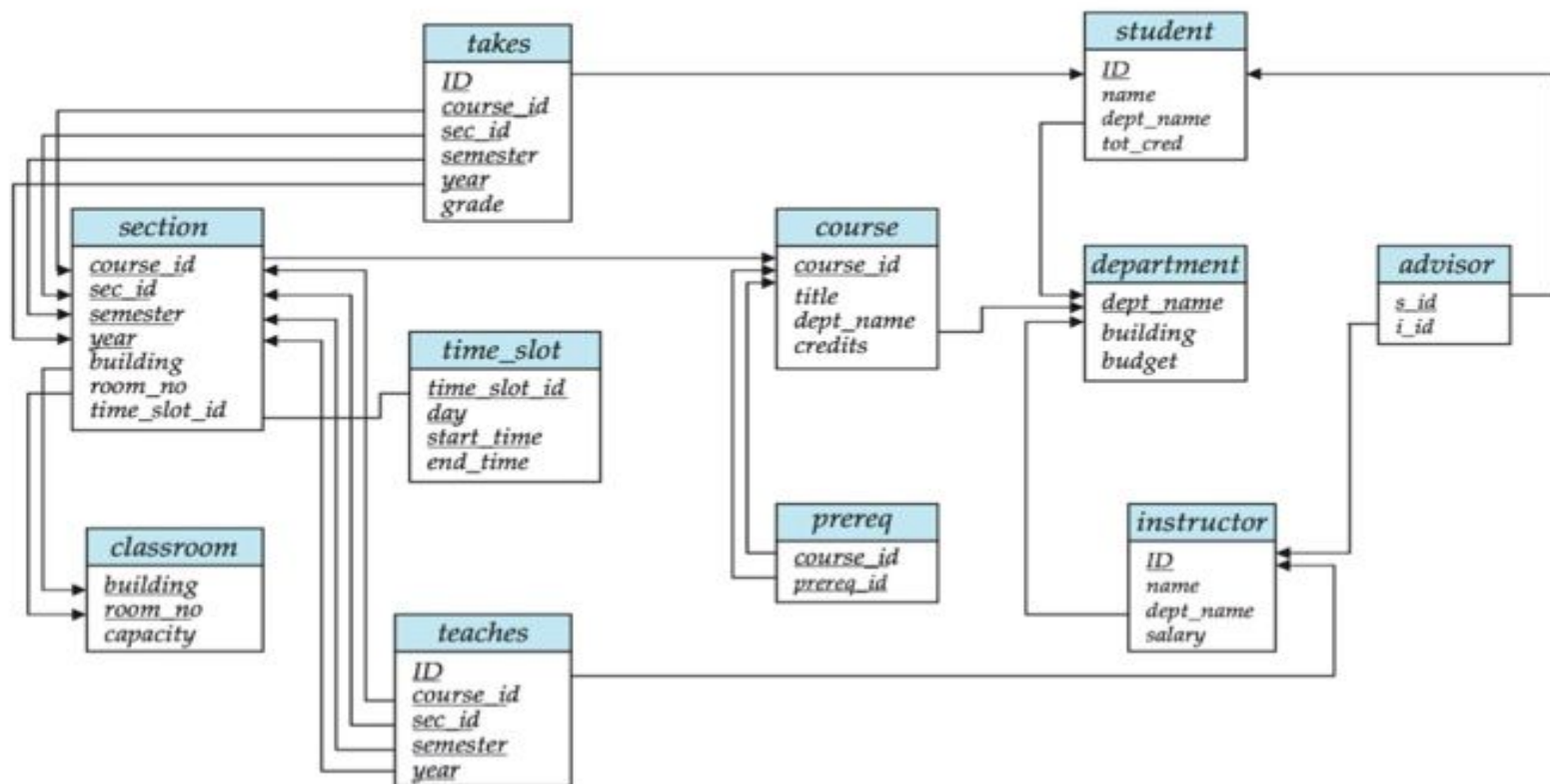
- ✓ The other relations of University database is given below
  - student (reg\_no, name, dept name, fees)
  - counselor (faculty\_id, reg\_no )
  - takes (reg\_no, course\_code, sec\_id, semester, year, credits)
  - classroom (location, room number, capacity)
  - time\_slot (time\_slot\_id, day\_order, start\_time, end\_time)

## Keys

- ✓ One or more attributes used to identify an entity uniquely in an entity set is known as key attributes
- ✓ Key attributes are called Super Key
- ✓ Minimal of Super key is Candidate Key
- ✓ Candidate key is also known as Primary key
- ✓ A primary key for a particular relation will be act as a referential key for another table is known as Foreign key

## Schema Diagram

- ✓ The given figure is for University Database
- ✓ A database schema, along with primary key and foreign key dependencies, can be depicted by schema diagrams.
- ✓ Each relation given as relation name and list of attributes
- ✓ Primary key attributes are underlined
- ✓ Foreign key dependencies appear as arrows from the foreign key attributes of the referencing relation to the primary key of the referenced relation.
- ✓ Referential integrity constraints other than foreign key constraints are not shown explicitly in schema diagrams.

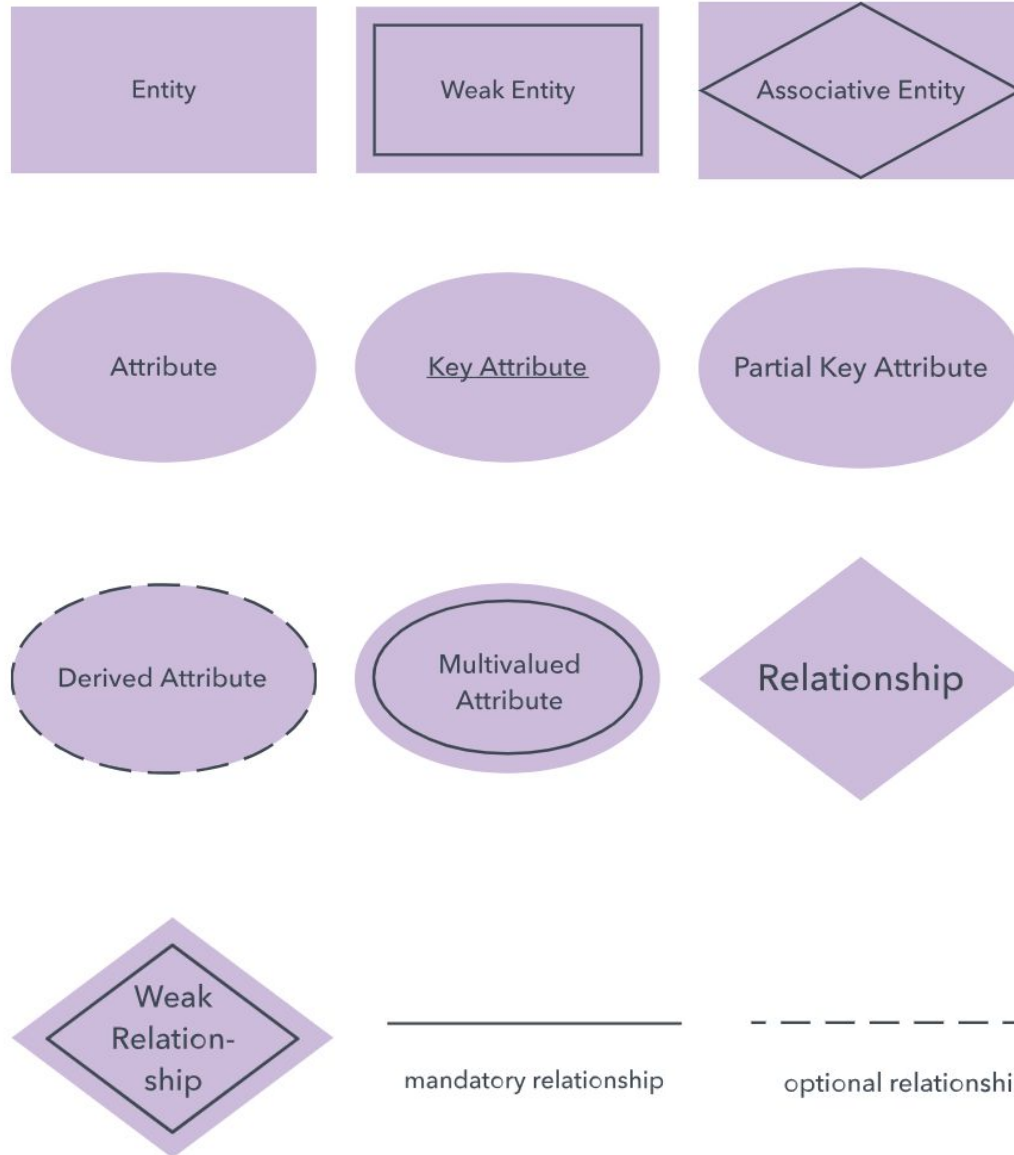


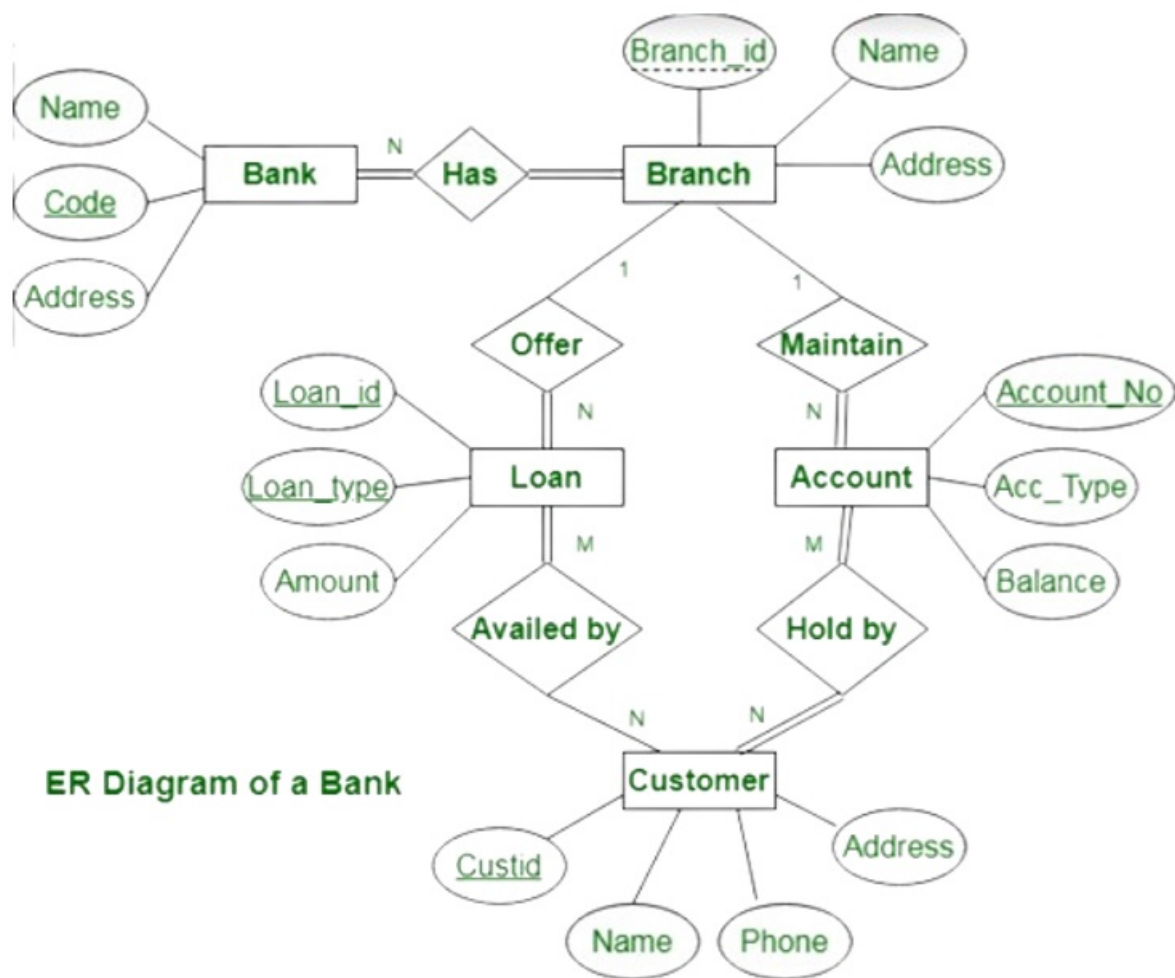
## Relational Query Languages

- ✓ A query language is a language in which a user requests information
- ✓ These languages are usually on a level higher than that of a standard programming.
- ✓ Query languages can be categorized as either procedural or nonprocedural.
- ✓ In a procedural language, the user instructs the system to perform a sequence of operations on the database to compute the desired result.
- ✓ In a nonprocedural language, the user describes the desired information without giving a specific procedure for obtaining that information.
- ✓ There are a number of “pure” query languages.
- ✓ The relational algebra is procedural.
- ✓ The tuple relational calculus and domain relational calculus are nonprocedural.

# ER Diagram Symbols







**ER Diagram of a Bank**