18CSC303J -Database Management Systems

UNIT-II

Outline of the Presentation

S-1 SLO-1 :Database Design

SLO-2 :Design process

S-2 SLO-1 & SLO-2: Entity Relationship Model

S-3 SLO-1 & SLO2 : ER diagram

S 4-5 SLO-1 & SLO-2 : Lab4 : Inbuilt functions in SQL on sample exercise

S-6 SLO-1 & SLO-2 : Keys , Attributes and Constraints

S-7 SLO-1 & SLO-2 : Mapping Cardinality

S-8 SLO-1 & SLO-2 : Extended ER - Generalization, Specialization and Aggregation

S-9-10 SLO-1 & SLO-2 : Lab 5: Construct a ER Model for the application to be constructed to a Database

S-11 SLO-1 : ER Diagram Issues

SLO-2 : Weak Entity

S-12 SLO-1 & SLO-2 : Relational Model

S-13 SLO-1 & SLO-2 : Conversion of ER to Relational Table

S-14-15 SLO-1 & SLO-2 : Lab 6: Nested Queries on sample exercise

S-1 SLO-1 : Database Design

- Database systems are developed to manage large amount of specific domain and related domain.
- Database design involves the design database schema.
- The complete design of the database of application environment based on the requirements given by the environment/business
- The database design to be thoroughly discussed and designed by both database designer / developer and authorities from the enterprise (domain).

S-1 SLO-2 :Design process

- Design Process
- Conceptual design
- ✔A high level data model provides the database designer with a conceptual frame work which includes
    - What kind of data required by the database users?
    - How the database to be designed to fulfill the requirements?
- ✔Database designer should choose the appropriate data model and translate these requirements into a conceptual schema.
- ✔The schema developed at this conceptual-design phase provides a detailed overview of the enterprise.
- ✔The designer review the schema to confirm that all data requirements.
- ✔The designer can review the design to remove the redundant features
- ✔The focus at this point is on describing the data and their relationships, rather than on specifying physical storage details.

S-1 SLO-2 :Design process

- Design Process
- Specification of functional requirements
- The fully developed conceptual schema provides the functional requirements of the enterprise.
- Functional requirements describe about what kind of operation / transaction to be performed on the data.
  - The operations are:

    •Updating or Modifying data • Retrieval of data for processing • Deleting the data
- Designer can review the schema to ensure it meets all the functional requirements.

S-1 SLO-2 :Design process

- Design Process

  ✔The process of moving from an abstract data model to the implementation of the database proceeds in two final design phases.

- Logical Design Phase

  ✔The designer maps the high level conceptual schema onto the data model

- Physical Design Phase

  ✔The physical features of database are specified

  - File organization
  - Internal storage structures

S-1 SLO-2 :Design process

Database Design for a University

- ✔The initial specifications of the user requirements may collected
  - Discussion with database users
  - Designer's own analysis
- ✔It helps to design the conceptual structure of database

S-1 SLO-2 :Design process

Major Characteristics of the University

- ✔ The University is organized into departments, Each department has
  - Identified by Unique_name
  - Located in a Building
  - Budget
  - etc.,
- ✔ Each department has a list of courses and it associated with
  - Course_id
  - Title
  - Dept_name
  - Credits
  - etd.,
- ✔ Each department has faculty and they are identified by
  - Faculty_id
  - Name
  - Dept_name
  - Salary

S-1 SLO-2 :Design process

Major Characteristics of the University

- ✔ Each department has students and are identified by
  - Unique_id
  - Name
  - Depatment_name
  - etc.,
- ✔ University maintenance department, maintains list of classrooms
  - Room_number
  - Located in a building
  - Room_capacity
- ✔ University maintains a list of all classes (sections) taught, each section is identified by
  - Course_id
  - Section_id
  - Year
  - Semester
  - Room_number
  - Located in a buliding
  - Time_slot_id

S-1 SLO-2 :Design process

Major Characteristics of the University

- ✔ The department has a list of teaching assignments specifying, for each faculty, the sections the faculty is teaching.
- ✔ The university has a list of all student course registrations, specifying, for each student, the courses and the associated sections that the student has taken (registered for).
- ✔ A real university database would be much more complex than the preceding design.

S-1 SLO-2 :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)

S-1 SLO-2 :Design process

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

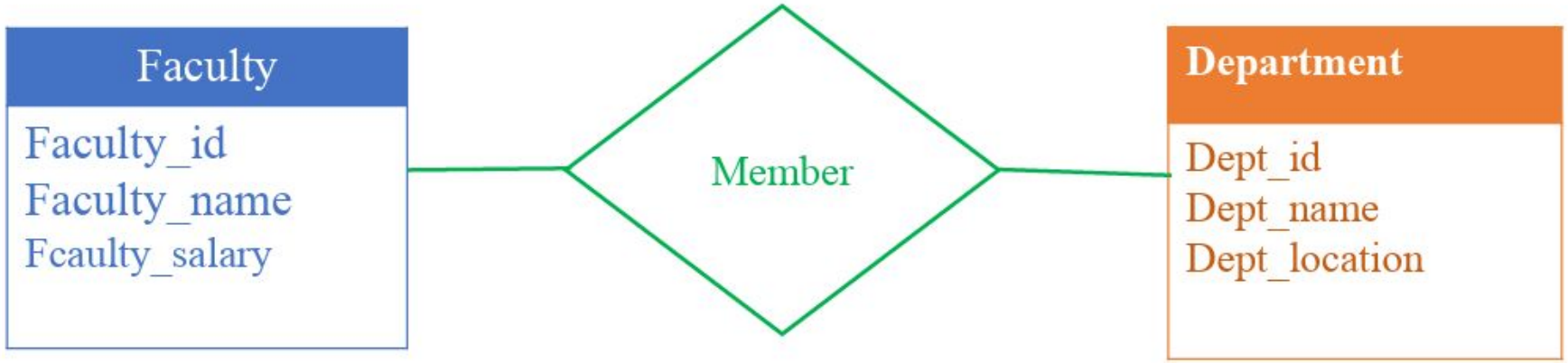
- ✔ The overall logical structure of a database can be represented using graphical notations by an E-R diagram.

- ✔ One of the most popular model is to use UML (Unified Modeling Language)

S-1 SLO-2 :Design process

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

S-1 SLO-2 :Design process

Normalization

- ✔ Normalization is a method to design a relational database
- ✔ It is a process to avoid redundant information and also inability to represent certain information
- ✔ It is used to design a good database without redundant information
- ✔ The most common approach is to use functional dependencies
- ✔ There are several normal forms are available , each normal forms designed using various functional dependencies

S-2 SLO-2 & SLO-2 :Entity Relationship Model

- ✔ 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

S-2 SLO-2 & SLO-2 :Entity Relationship Model

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

S-2 SLO-2 & SLO-2 :Entity Relationship Model

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.

S-2 SLO-2 & SLO-2 :Entity Relationship Model

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.,

S-2 SLO-2 & SLO-2 :Entity Relationship Model

- Values

✔ Each entity has a value for each attribute

✔ For instance , the particular faculty entity may have the following values :

- faculty_id = 123
- faculty_name= ‘Bill’
- faculty_ dept= = ‘IT’
- 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)

S-2 SLO-1 & SLO-2: Entity Relationship Model

- ✔ 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.

S-2 SLO-2 & SLO-2 :Entity Relationship Model

- Entity sets faculty and student

Faculty Entity Set:

Faculty_ID	Faculty_Name	Faculty_Department
101	John Smith	Computer Science
102	Emily Johnson	Mathematics
103	David Lee	Physics

Entity set : Faculty

Student Entity Set:

Student_ID	Student_Name	Student_Major
201	Sarah Parker	Biology
202	Michael Chen	Economics
203	Lisa Davis	Psychology

Entity set : Student

S-2 SLO-2 & SLO-2 :Entity Relationship Model

- Relationship Sets

✔ A relationship is an association among several entities.

✔ For example, we can define a relationship "counselor" that associates faculty "John Smith" with the student "Sarah Parker".

✔ The relationship specifies that "John Smith" is a counselor to student "Sarah Parker".

✔ A relationship set is a set of relationships of the same type.

✔ Formally, it is a mathematical relation on $n \geq 2$ (possibly non-distinct) entity sets.

✔ If E1, E2,..., En are entity sets, then a relationship set R is a subset of

$\{(e1,e2,...,en) \mid e1 \in E1, e2 \in E2,..., en \in En\}$ where (e1,e2,...,en) is a relationship.

S-2 SLO-2 & SLO-2 :Entity Relationship Model

- Relationship Sets

- In this case, the relationship set "counselor" would be represented as:

- Counselor = {(John Smith, Sarah Parker)}

- This indicates that "John Smith" is a counselor to "Sarah Parker".

S-2 SLO-2 & SLO-2 :Entity Relationship Model

- ✔ Consider the two entity sets Faculty and Student (Ref : Slide No 21)
- ✔ We define the relationship set counselor to denote the association between faculty and students.
- ✔ The following figure represents this association

Faculty Entity Set:

Faculty_ID	Faculty_Name	Faculty_Department
101	John Smith	Computer Science
102	Emily Johnson	Mathematics
103	David Lee	Physics

Student Entity Set:

Student_ID	Student_Name	Student_Major
201	Sarah Parker	Biology
202	Michael Chen	Economics
203	Lisa Davis	Psychology

S-2 SLO-2 & SLO-2 :Entity Relationship Model

- Relationship Sets
- The association between entity sets is referred to as participation.
- The entity sets E1, E2,..., En 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 John Smith, who has faculty_id 101, and the student entity Sarah Parkerwho has student_regno 201 participate in a relationship instance counselor/advisor.
- This relationship instance represents that in the university, the faculty "John Smith" is advising student "Sarah Parker".
- Advisor = {(Faculty_ID: 101, Student_ID: 201)}
- This indicates that "John Smith" with Faculty_ID 101 is advising "Sarah Parker" with Student_ID 201.

S-2 SLO-2 & SLO-2 :Entity Relationship Model

- 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

S-2 SLO-2 & SLO-2 :Entity Relationship Model

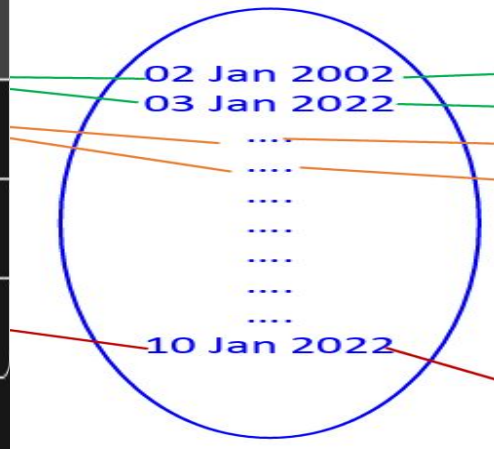
- A relationship may also have attributes called descriptive attributes.
- Consider a relationship set “advisor” with entity sets Faculty and Student.
- The attribute "date" can be associated with that relationship to specify the date when the faculty became the advisor of a student.
- For example, the advisor relationship among the entities corresponding to faculty "John Smith" and student "Sarah Parker" has the value “3 Jan 2022” for attribute "date", which means that "John Smith" became "Sarah Parker"’s advisor on January 3, 2022.
- In this case, we can represent this as:
- Advisor = {(Faculty_ID: 101, Student_ID: 201, Date: 3 Jan 2022)}
- This indicates that "John Smith" with Faculty_ID 101 started advising "Sarah Parker" with Student_ID 201 on January 3, 2022.

S-2 SLO-2 & SLO-2 :Entity Relationship Model

- ✔ Consider a relationship set "counselor" with a descriptive attribute "date".
- ✔ The figure shows that the faculty "John Smith" counsels two students with two different counseling dates.

Faculty Entity Set:

Faculty_ID	Faculty_Name	Faculty_Department
101	John Smith	Computer Science
102	Emily Johnson	Mathematics
103	David Lee	Physics



Student Entity Set:

Student_ID	Student_Name	Student_Major
201	Sarah Parker	Biology
202	Michael Chen	Economics
203	Lisa Davis	Psychology

S-2 SLO-1 & SLO-2: Entity Relationship Model

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.

S-2 SLO-1 & SLO-2: Entity Relationship Model

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, 101), (Faculty_Name, John Smith), (Dept_Name, Computer Science), (Salary, $70,000)}

S-2 SLO-1 & SLO-2: Entity Relationship Model

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

S-2 SLO-1 & SLO-2: Entity Relationship Model

Types of Values	Description
Single value	• Only one value can be stored • Example : Faculty_id, DOB
Multiple value	• More values are possible • Example : Faculty_Phone_no
Derived value	• The values which is derived from existing value • Example : AGE • The values keep on changing is not advisable to store in the database • Normally the values will be derived from existing value of another attribute. • AGE will be changing continuously. • It can be derived from DOB (DOB never change)
Null value	• NULL values are unknown undeclared • An attribute does not have a value for a particular entity in an entity set

S-2 SLO-1 & SLO-2: Entity Relationship Model

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

S-2 SLO-1 & SLO-2: Entity Relationship Model

Mapping Cardinalities

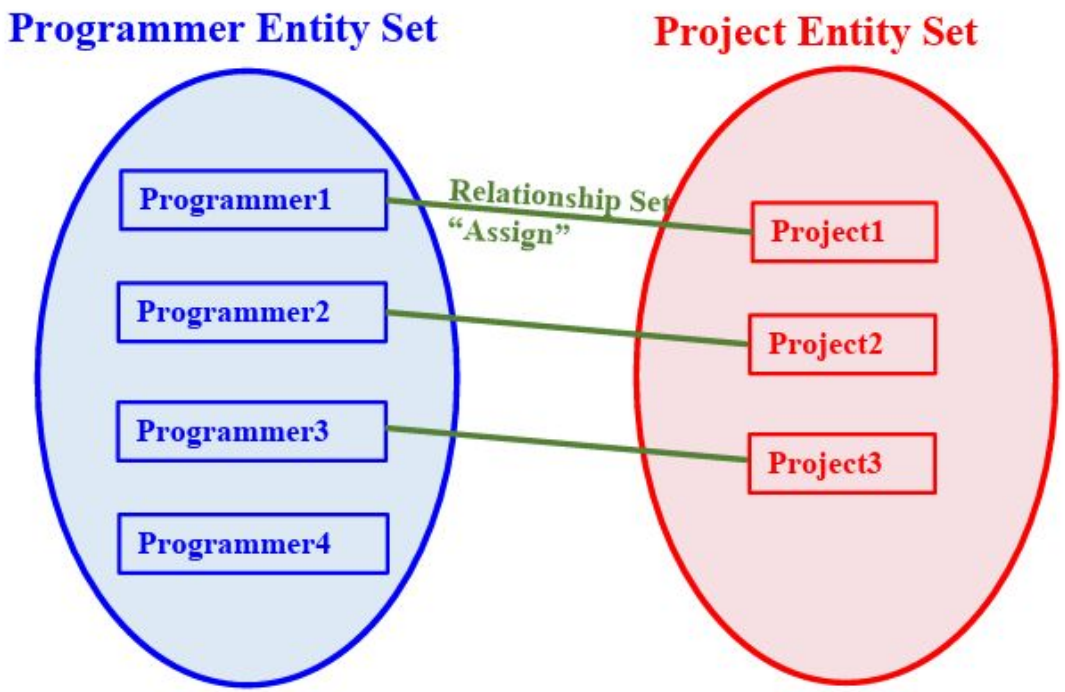
- ✔ Mapping cardinalities, or cardinality ratios, express the number of entities to which another entity can be associated via a relationship set.
- ✔ Mapping cardinalities are most useful in describing binary relationship sets.
- ✔ For a binary relationship set “Assign” between entity sets Programmer 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)

S-2 SLO-1 & SLO-2: Entity Relationship Model

Mapping Cardinalities

One-to-One (1:1)

- ✔ An entity in Programmer is associated with at most one entity in Project, and an entity in Project is associated with at most one entity in Programmer.
- ✔ The following figure depicts 1:1 mapping cardinality

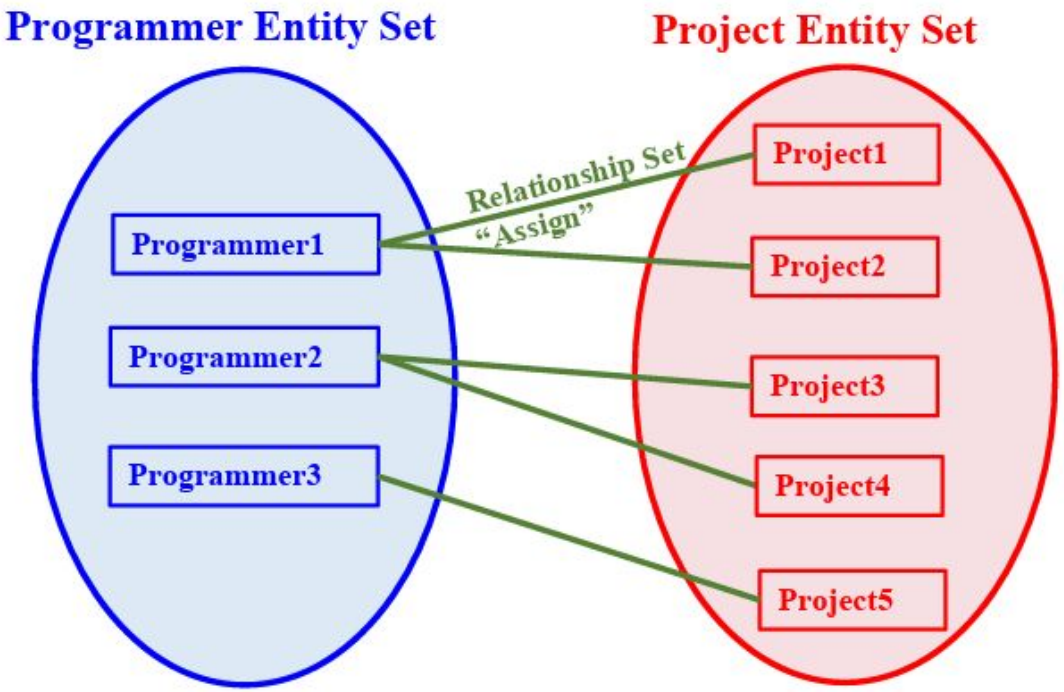


S-2 SLO-1 & SLO-2: Entity Relationship Model

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

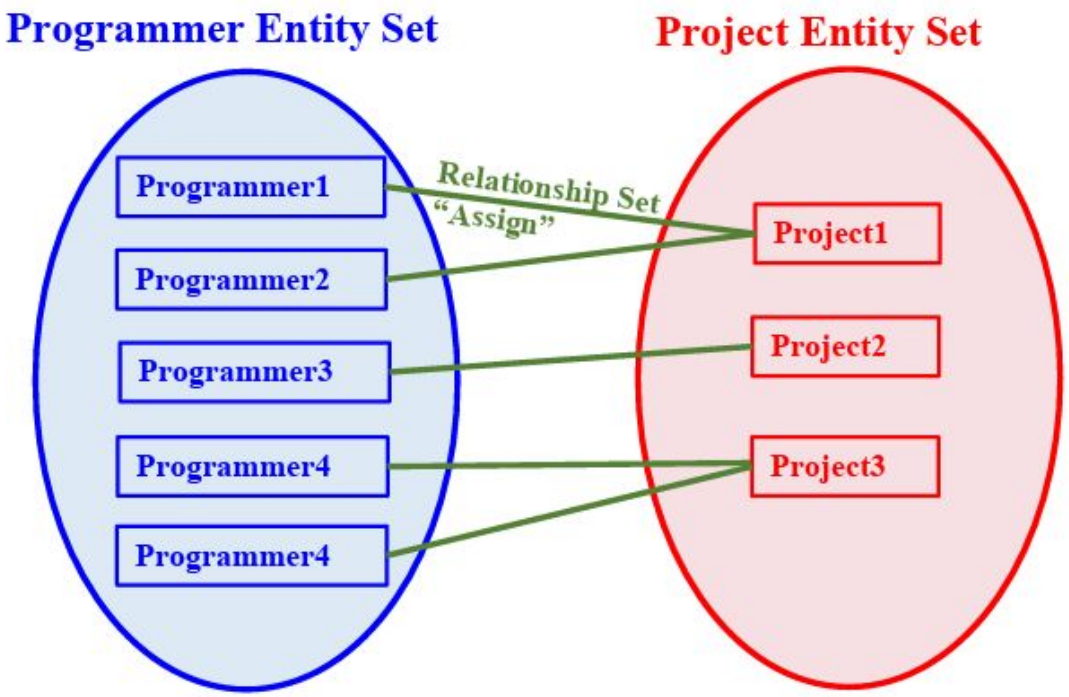


S-2 SLO-1 & SLO-2: Entity Relationship Model

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 1:M

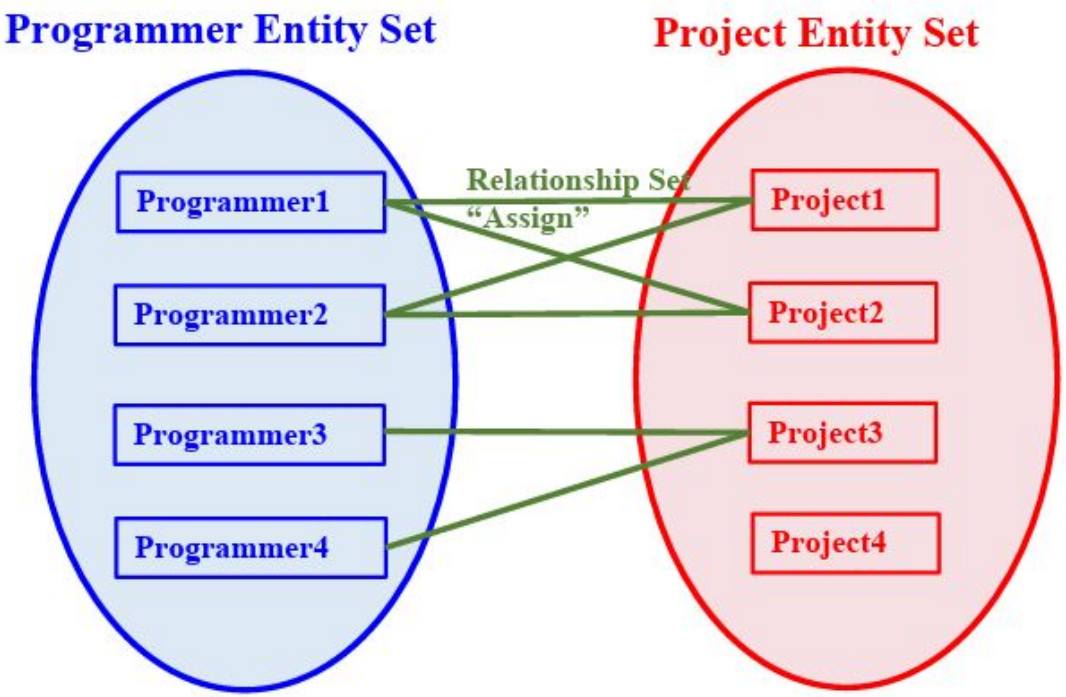


S-2 SLO-1 & SLO-2: Entity Relationship Model

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



S-2 SLO-1 & SLO-2: Entity Relationship Model

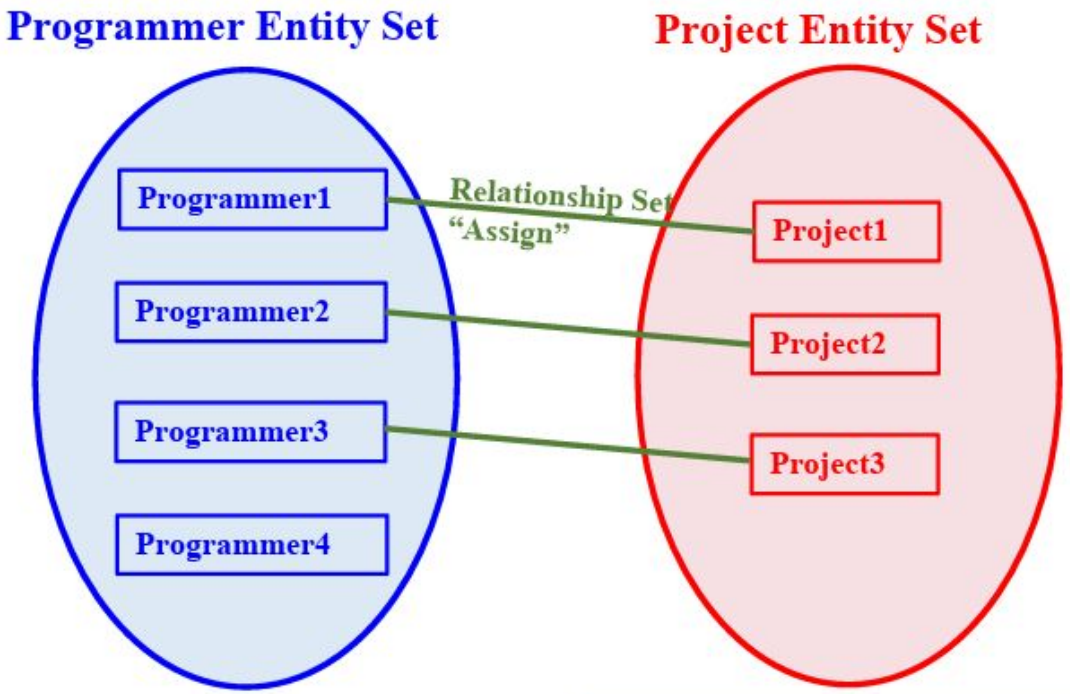
Participation Constraints

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.



S-2 SLO-1 & SLO-2: Entity Relationship Model

Participation Constraints

Example :

- ✔ In Figure : A, the participation of Project Entity Set in the relationship set is total while the participation of A 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.

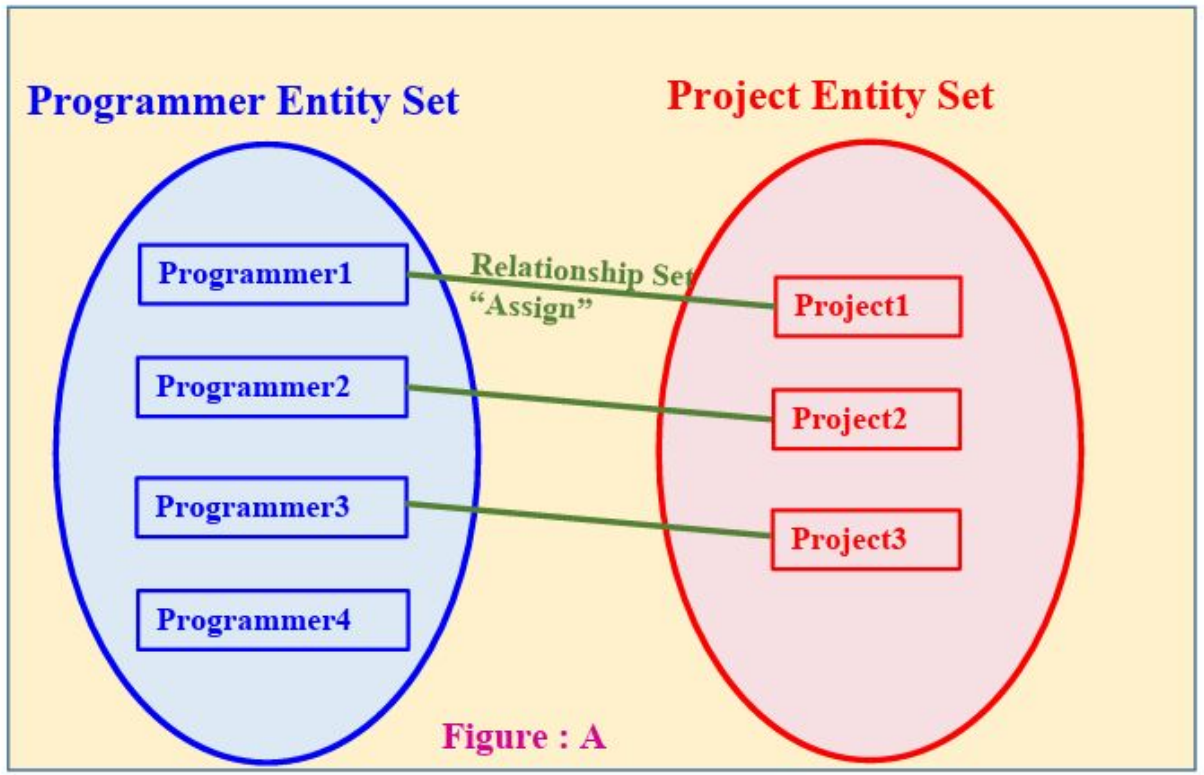


Figure : A

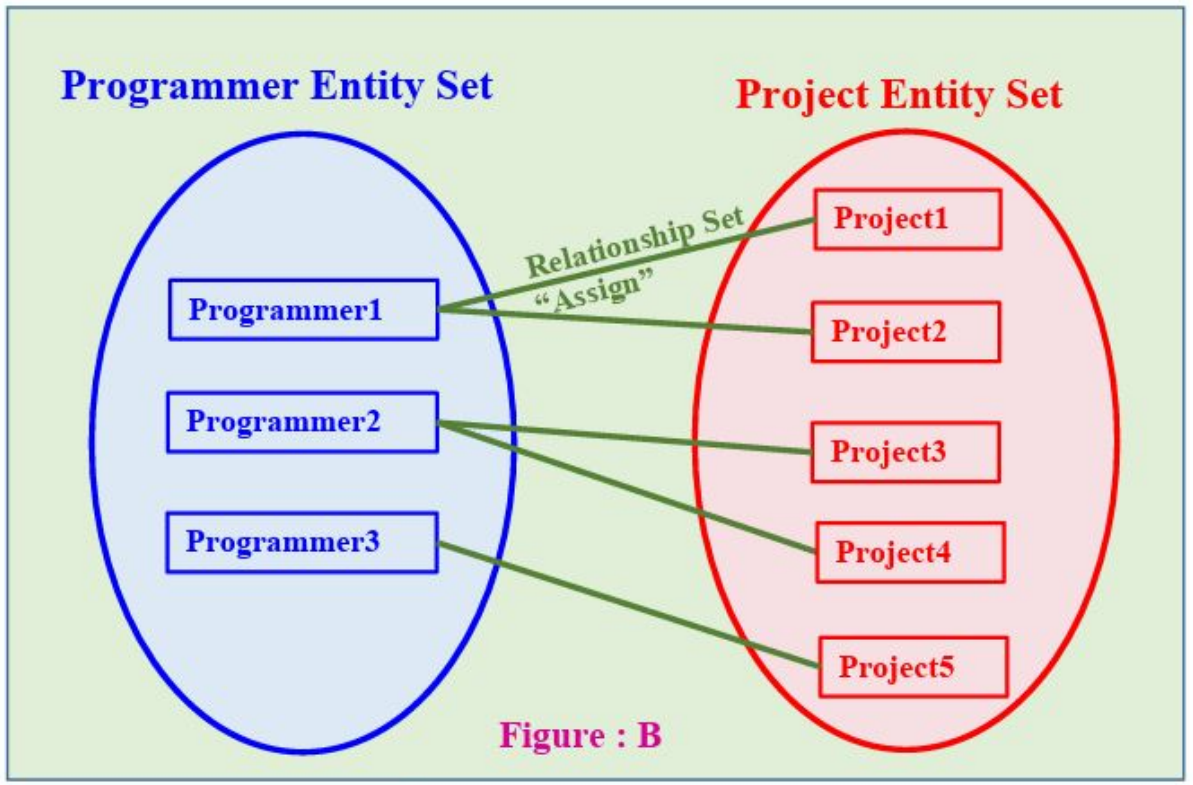


Figure : B

S-3 SLO-1 & SLO2 : ER diagram

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

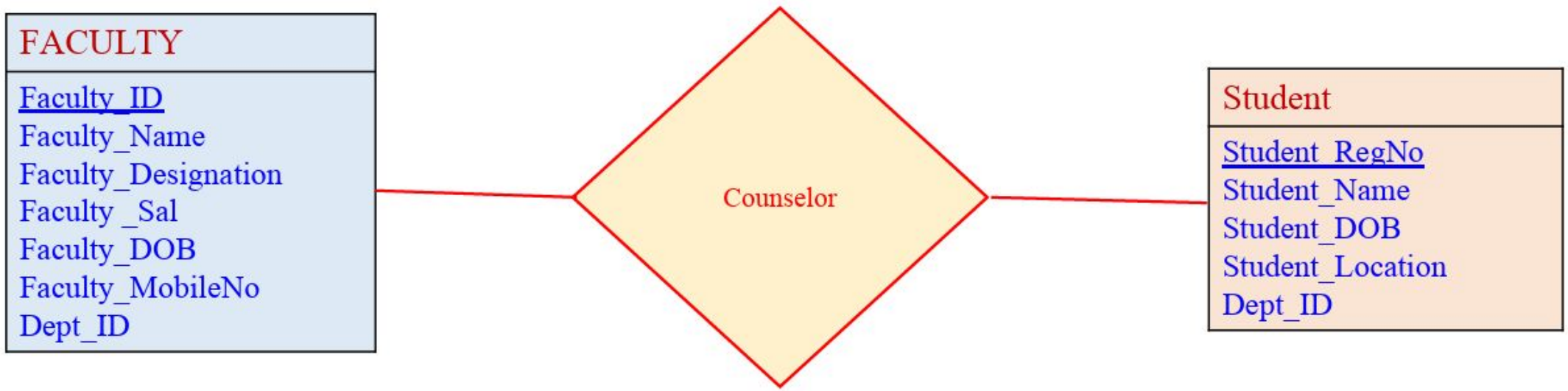
Basic Structure

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

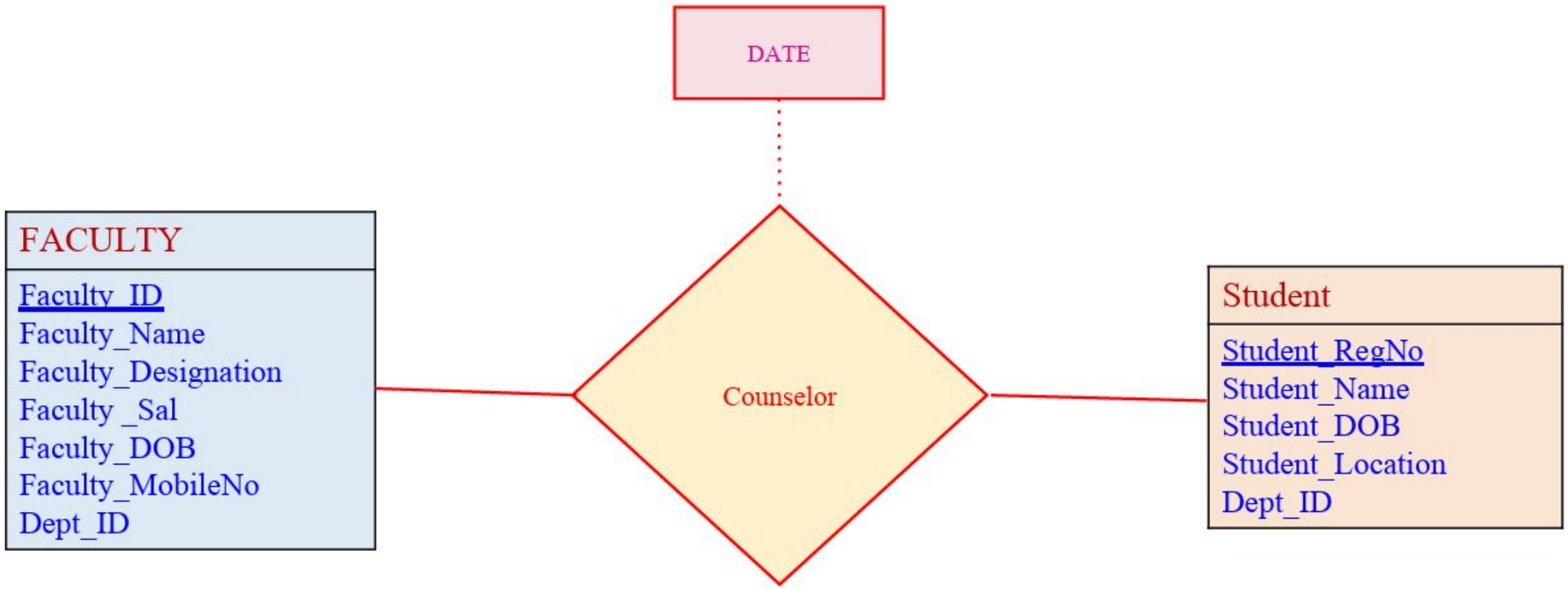
S-3 SLO-1 & SLO2 : ER diagram

- 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.



S-3 SLO-1 & SLO2 : ER diagram

- 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.



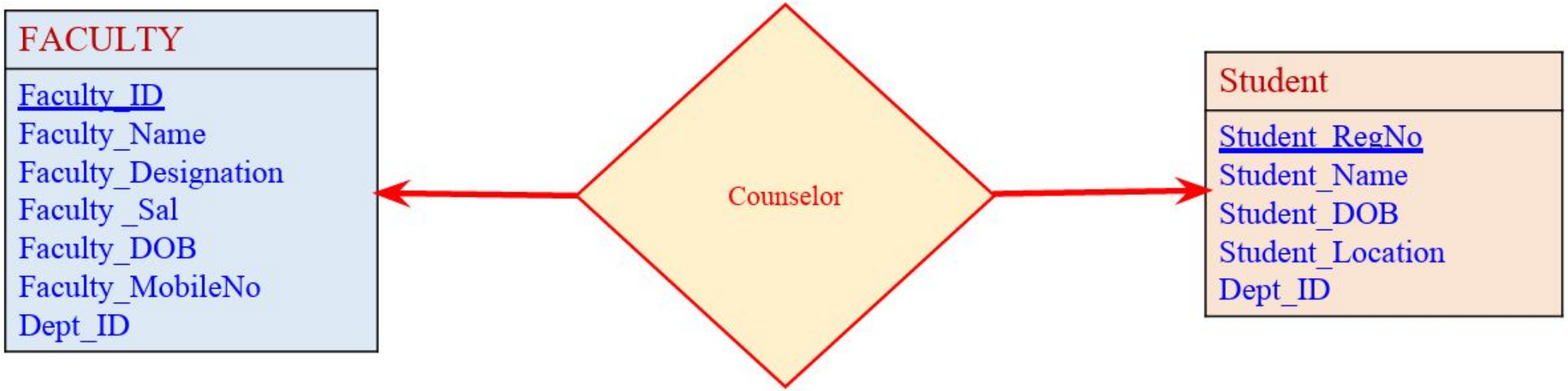
S-3 SLO-1 & SLO2 : ER diagram

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 (→) 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.

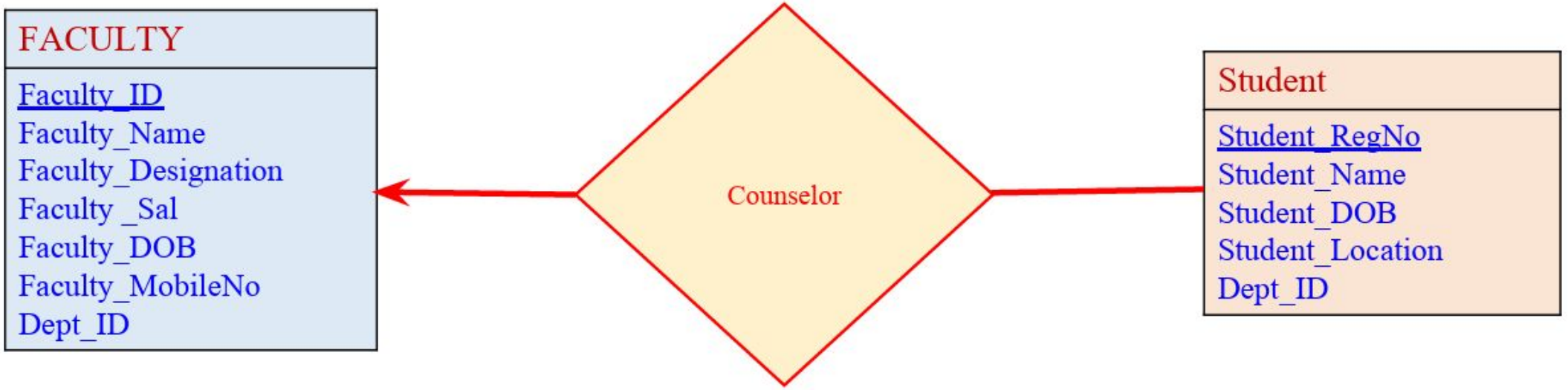


S-3 SLO-1 & SLO2 : ER diagram

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.

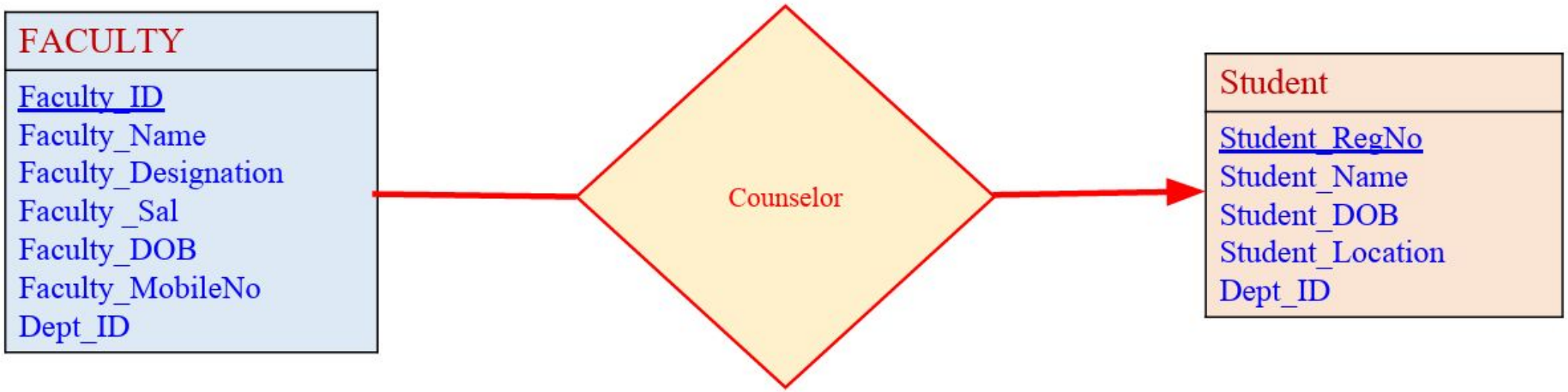


S-3 SLO-1 & SLO2 : ER diagram

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.

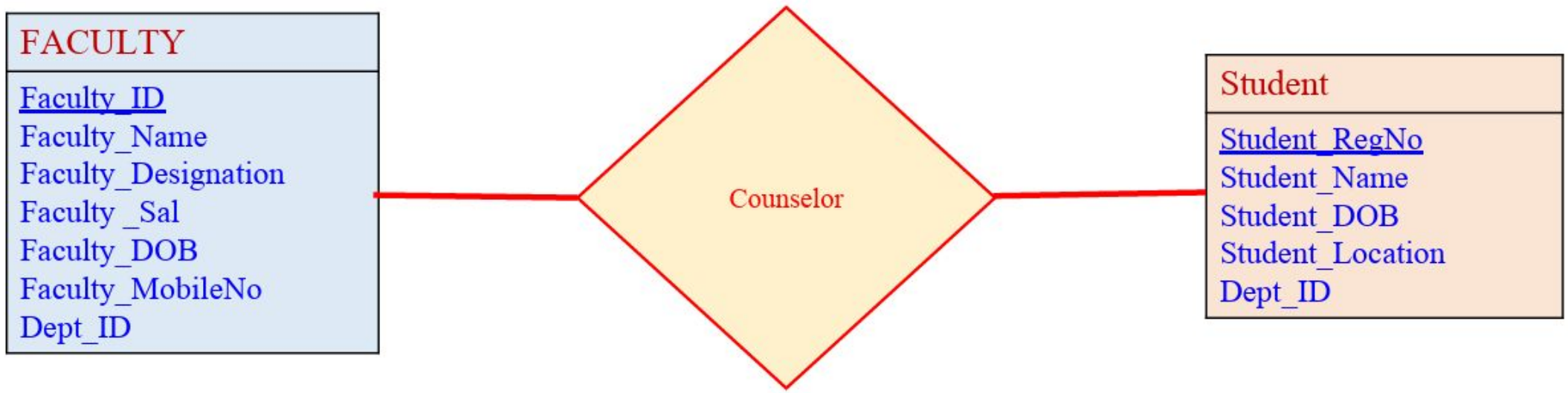


S-3 SLO-1 & SLO2 : ER diagram

Mapping Cardinality

Many-to-many:

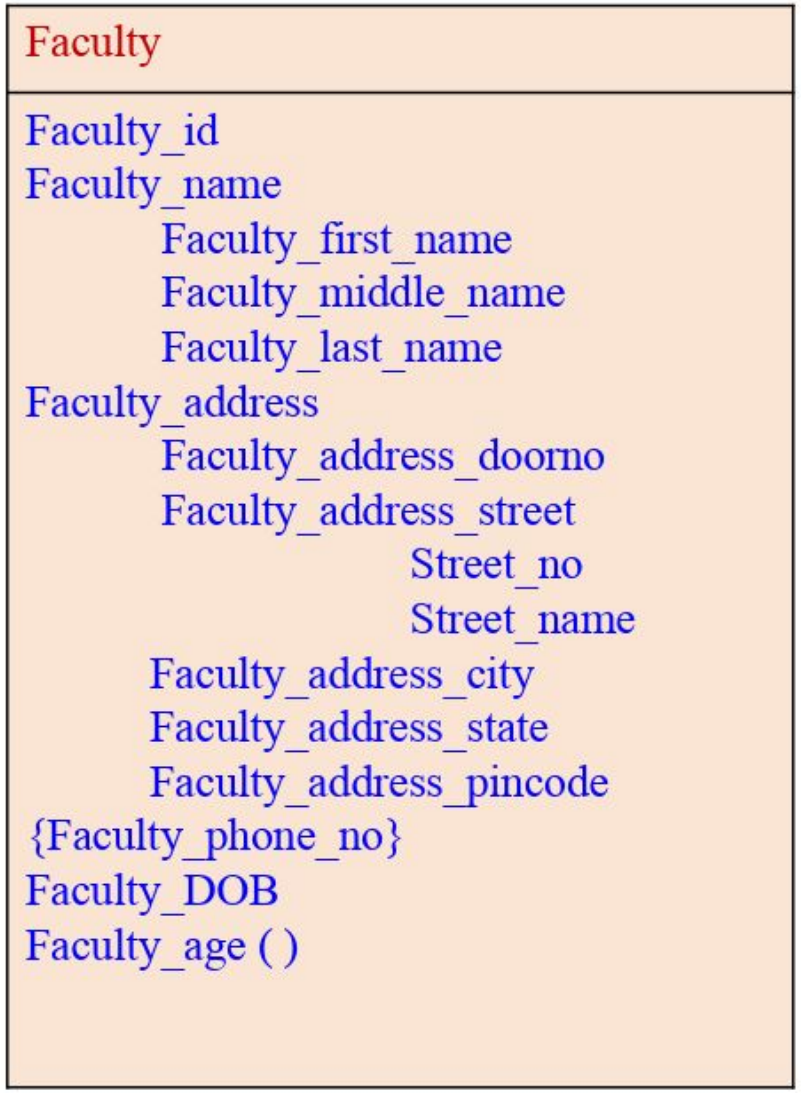
- We draw an undirected line from the relationship set counselor to both entity sets faculty and student as shown in the below figure, indicates that a faculty may counsel many students, and a student may have many counselor.



S-3 SLO-1 & SLO2 : ER diagram

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 ( )".



S 4-5 : SLO-1 & SLO-2 : Lab4 : Inbuilt functions in SQL on sample exercise

SQL Inbuilt functions are divided into the following categories

- ✔ Date Functions
- ✔ Character Functions
- ✔ Conversion functions
- ✔ Numeric functions
- ✔ Miscellaneous functions

S 4-5 : SLO-1 & SLO-2 : Lab4 : Inbuilt functions in SQL on sample exercise

Date Functions

Functions	Value Returned	Input	Output
add_months(d,n)	'n' months added to date 'd'.	Select add_months(sysdate,2) from dual;	
last_day(d)	Date corresponding to the last day of the month	Select last_day(sysdate) from dual;	
to_date(str,'format')	Converts the string ina given format into Oracle date.	Select to_date('10-02-09','dd-mm-yy') from dual;	
to_char(date,'format')	Reformats date according to format	Select to_char(sysdate,'dy dd mon yyyy') from dual;	
months_between(d1,d2)	No. of months between two dates	Select months_between(sysdate, to_date('10-10-07','dd-mm-yy')) from dual;	
next_day(d,day)	Date of the 'day' that immediately follows the date 'd'	Select next_day(sysdate,'wednesday') from dual;	

S 4-5 : SLO-1 & SLO-2 : Lab4 : Inbuilt functions in SQL on sample exercise

Date Functions

Functions	Value Returned	Input	Output
round(d,'format')	Date will be the rounded to nearest day.	Select round(sysdate,'year') from dual;	
		Select round(sysdate,'month') from dual;	
		Select round(sysdate,'day') from dual;	
		Select round(sysdate) from dual;	
trunc(d,'format');	Date will be the truncated to nearest day.	Select trunc(sysdate,'year') from dual;	
		Select trunc(sysdate,'month') from dual;	
		Select trunc(sysdate,'day') from dual;	
		Select trunc(sysdate) from dual;	
greatest(d1,d2,…)	Picks latest of list of dates	Select greatest(sysdate, to_date('02-10-06','dd-mm-yy'),to-date('12-07- 12','dd-mm-yy')) from dual;	
Date Arithmetic	Add /Subtract no. of days to a date	Select sysdate+25 from dual;	
		Select sysdate-25 from dual;	
	Subtract one date from another, producing a no. of days	Select sysdate - to_date('02-10-06','dd- mm-yy') from dual;	

S 4-5 : SLO-1 & SLO-2 : Lab4 : Inbuilt functions in SQL on sample exercise

Character Functions

Functions	Value Returned	Input	Output
initcap(char)	First letter of each word capitalized	Select initcap('database management') from dual;	
lower(char)	Lower case	Select lower('WELCOME') from dual;	
upper(char)	Upper case	Select upper('srmist') from dual;	
ltrim(char, set)	Initial characters removed up to the character not in set.	Select ltrim('muruganantham','murug') from dual;	
rtrim(char, set)	Final characters removed after the last character not in set.	Select rtrim('muruganantham','antham') from dual;	
translate(char, from, to)	Translate 'from' by 'to' in char.	Select translate('jack','j','b') from dual;	
replace(char, search, repl)	Replace 'search' string by 'repl' string in 'char'.	Select replace('jack and jue','j','bl') from dual;	
substr(char, m, n)	Substring of 'char' at 'm' of size 'n' char long.	Select substr('muruganantham',7,6) from dual;	

S 4-5 : SLO-1 & SLO-2 : Lab4 : Inbuilt functions in SQL on sample exercise

Conversion Functions

Functions	Value Returned	Input	Output
to_date(str,'format')	Converts the string ina given format into Oracle date.	Select to_date('10-02-09','dd-mm-yy') from dual;	
to_char(date,'format')	Reformats date according to format	Select to_char(sysdate,'dy dd mon yyyy) from dual;	
to_char(number,'format')	Display number value as a char.	Select to_char(12345.5,'L099,999.99') from dual;	
to_number(char)	Char string to number form	Select to_number('123') from dual;	

S 4-5 : SLO-1 & SLO-2 : Lab4 : Inbuilt functions in SQL on sample exercise

Numeric Functions

Functions	Value Returned	Input	Output
Abs(n)	Absolute value of n	Select abs(-15) from dual;	
Ceil(n)	Smallest int >= n	Select ceil(33.645) from dual;	
Cos(n)	Cosine of n	Select cos(180) from dual;	
Cosh(n)	Hyperbolic cosine of n	Select cosh(0) from dual;	
Exp(n)	en	Select exp(2) from dual;	
Floor(n)	Largest int <= n	Select floor(100.2) from dual;	
Ln(n)	Natural log of n (base e)	Select ln(5) from dual;	
Log(b,n)	Log n base b	Select log(2,64) from dual;	
Mod(m,n)	Remainder of m divided by n	Select mod(17,3) from dual;	

S 4-5 : SLO-1 & SLO-2 : Lab4 : Inbuilt functions in SQL on sample exercise

Numeric Functions

Functions	Value Returned	Input	Output
Power(m,n)	m power n	Select power(5,3) from dual;	
Round(m,n)	m rounded to n decimal places	Select round(125.67854,2) from dual;	
Sign(n)	If n<0, -1 if n=0, 0 otherwise 1.	Select sin(-19) from dual;	
Sin(n)	Sin of n	Select sin(90) from dual;	
Sinh(n)	Hyperbolic sin of n	Select sinh(45) from dual;	
Sqrt(n)	Square root of n	Select sqrt(7) from dual;	
Tan(n)	Tangent of n	Select tan(45) from dual;	
Tanh(n)	Hyperbolic tangent of n	Select tanh(60) from dual;	
Trunc(m,n)	m truncated to n decimal places	Select trunc(125.5764,2) from dual;	

S 4-5 : SLO-1 & SLO-2 : Lab4 : Inbuilt functions in SQL on sample exercise

Miscellaneous Functions

Functions	Value Returned	Input	Output
Uid	User id	Select uid from dual;	
User	User name	Select user from dual;	
Vsize(n)	Storage size of v	Select vsize('hello') from dual;	
NVL(exp1,exp2)	Returns exp1 if not null, otherwise returns exp2.	Select nvl(comm,50) from emp where empno=7369;	

S 4-5 : SLO-1 & SLO-2 : Lab4 : Inbuilt functions in SQL on sample exercise

GROUP FUNCTIONS

- ✔ AVG : Average value of a set
- ✔ COUNT : Numbers of non null values
- ✔ MAX : Maximum of a set
- ✔ MIN : Minimum of a set
- ✔ STDDEV : Standard Deviation of a set
- ✔ SUM : Sum of a set
- ✔ VARIANCE : Variance of a set

Note:

- ✔Group functions ignore null values
- ✔*Group by* Clause is used to modularize rows in a table into smaller groups
- ✔Columns that are not a part of the Group Functions should be included in the Group by clause
- ✔Any column or expression in the SELECT list that is not an aggregate function must be in the GROUP BY clause
- ✔Group Functions cannot be placed in the where clause
- ✔HAVING clause is to restrict groups Groups satisfying the HAVING condition are displayed

S-6 SLO-1 & SLO-2 : Keys , Attributes and Constraints

Keys

- ✔ An entity should be identified in an entity set uniquely.
- ✔ It is expressed in terms of their attributes
- ✔ The values hold by attributes must identify the record / tuple uniquely.
- ✔ No two records in relation are not allowed to hold exactly the same values for all attributes.

S-6 SLO-1 & SLO-2 : Keys , Attributes and Constraints

Superkey

- ✔ A superkey is a set of one or more attributes that, taken collectively, allow us to identify uniquely a record in the relation.
- ✔ For example, the Faculty_ID attribute of the relation faculty is sufficient to distinguish one faculty record from another.
- ✔ Here Faculty_ID is the superkey.
- ✔ The Faculty_name attribute of Faculty, on the other hand, is not a superkey, because many faculty might have the same name.

S-6 SLO-1 & SLO-2 : Keys , Attributes and Constraints

Superkey

- Let R denote the set of attributes in the schema of relation r. If we say that a subset K of R is a superkey for r.
- We are restricting consideration to instances of relations r in which no two distinct tuples have the same values on all attributes in K.
- That is, if t1 and t2 are in r and t1 = t2, then t1.K = t2.K.
- A superkey may contain extraneous attributes. For example, the combination of Faculty_ID and Faculty_name is a superkey for the relation Faculty.
- Minimal of Superkeys are called as Candidate key.
- It is possible that several distinct set of attributes could serve as a Candiadate key

S-6 SLO-1 & SLO-2 : Keys , Attributes and Constraints

Superkey

- Suppose that a combination of Faculty_name and Dept_name is sufficient to distinguish among members of the Faculty relation.
- Then, both {Faculty_ID} and {Faculty_name, Dept_name} are candidate keys.
- Although the attributes Faculty_ID and Faculty_name together can distinguish faculty tuples, their combination, {Faculty_ID, Faculty_name}, does not form a candidate key, since the attribute Faculty_ID alone is a candidate key.
- The term primary key is to denote a candidate key.
- A key (whether primary, candidate, or super) is a property of the entire relation, rather than of the individual tuples.
- The designation of a key represents a constraint in the real-world enterprise being modeled.

S-6 SLO-1 & SLO-2 : Keys , Attributes and Constraints

Superkey

- ✔ The Primary key should be selected with special care.
- ✔ As we discussed the name of the person is obviously not sufficient to identify uniquely a person , because many persons can have the same name.
- ✔ In India , now the Aadhar card number attribute would be a primary key / candidate key.
- ✔ Non resident of India will not have the Aadhar number .
- ✔ An alternative is to use some unique combination of other attributes as a key.

S-6 SLO-1 & SLO-2 : Keys , Attributes and Constraints

Superkey

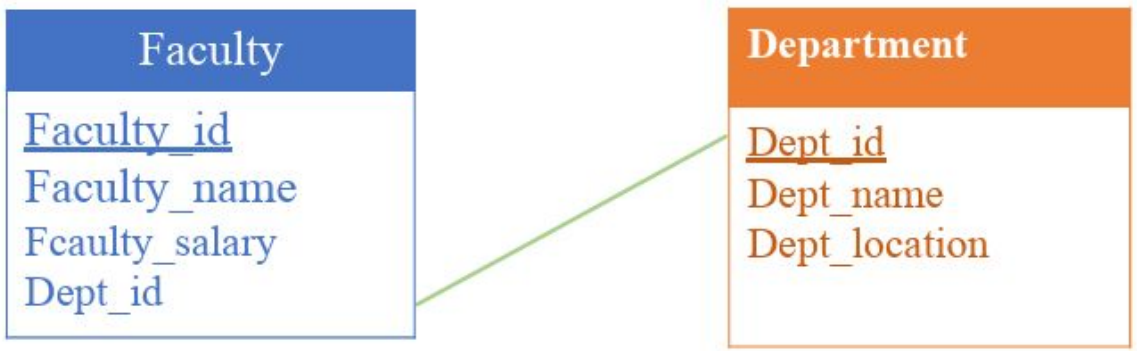
- ✔ The primary key should be chosen such that its attribute values are never,or very rarely, changed.
- ✔ For example , the address field should not be a primary key or part of primary key, since it is likely to change but, Aadhar number guaranteed never to change.
- ✔ To represent the primary key , the primary key attributes are underlined
- ✔ A relation, say r1, may include among its attributes the primary key of an other relation, say r2. This attribute is called a foreign key from r1, referencing r2.
- ✔ The relation r1 is also called the referencing relation of the foreign key dependency, and r2 is called the referenced relation of the foreign key.

Note : A primary key for a particular relation/ table is act as an referential key in another table (s) is called foreign key , it known as referential integrity constraints

S-6 SLO-1 & SLO-2 : Keys , Attributes and Constraints

Superkey

- ✔ Consider the two entitiy 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)



S-6 SLO-1 & SLO-2 : Keys , Attributes and Constraints

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

S-6 SLO-1 & SLO-2 : Keys , Attributes and Constraints

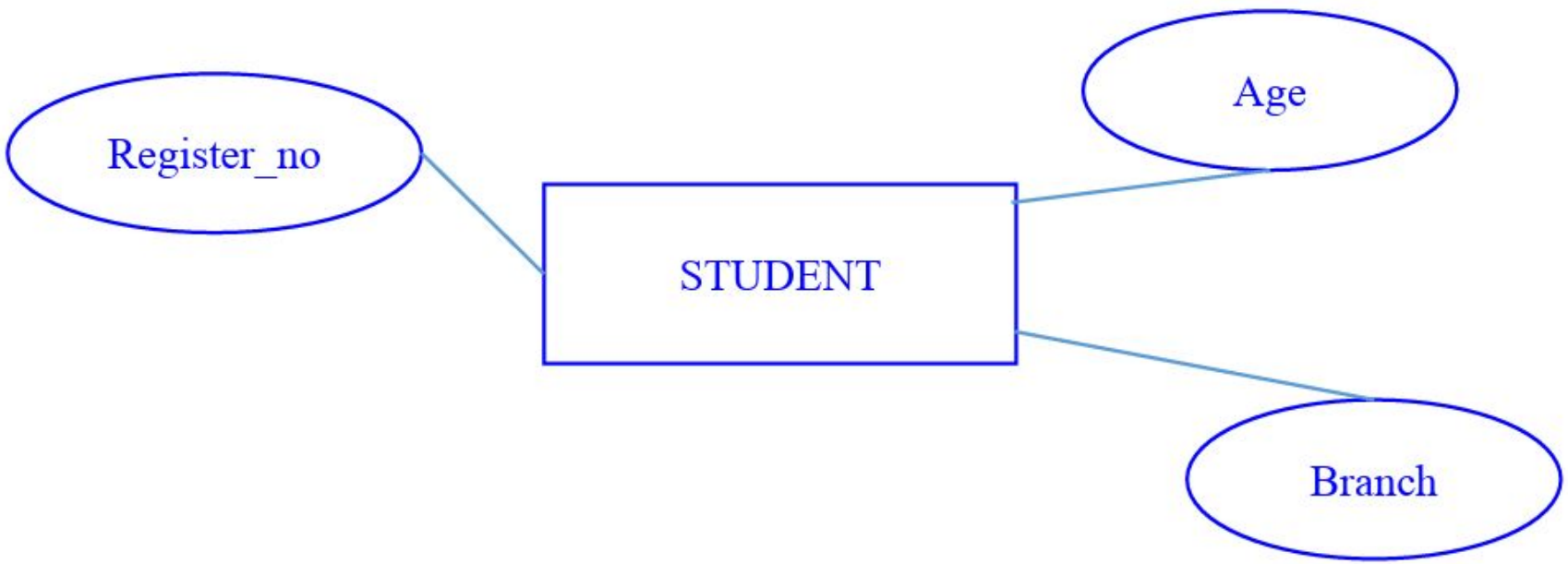
Simple attributes

It can not be divided further

All the simple attributes will hold the atomic values

Example :

Student = { Register_no, Name, ……. }

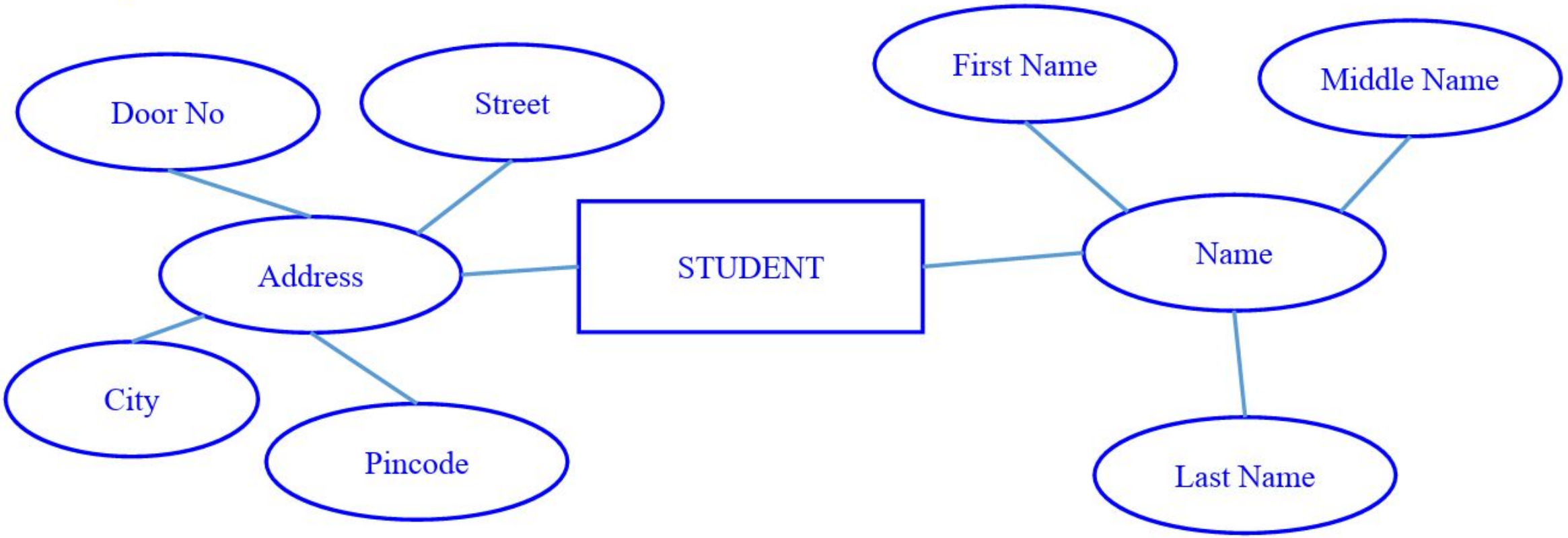


S-6 SLO-1 & SLO-2 : Keys , Attributes and Constraints

Composite attributes

Composed by many other simple attributes

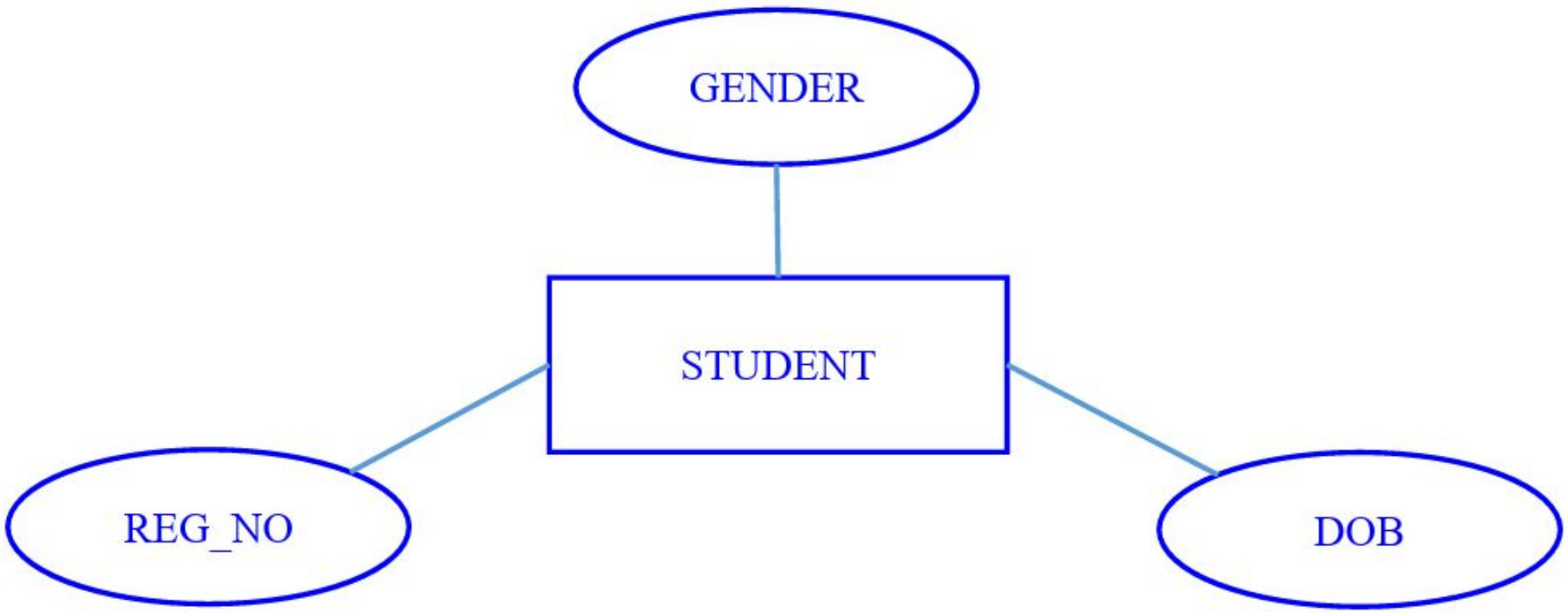
Example : Address , Name , etc.,



S-6 SLO-1 & SLO-2 : Keys , Attributes and Constraints

Single valued attributes

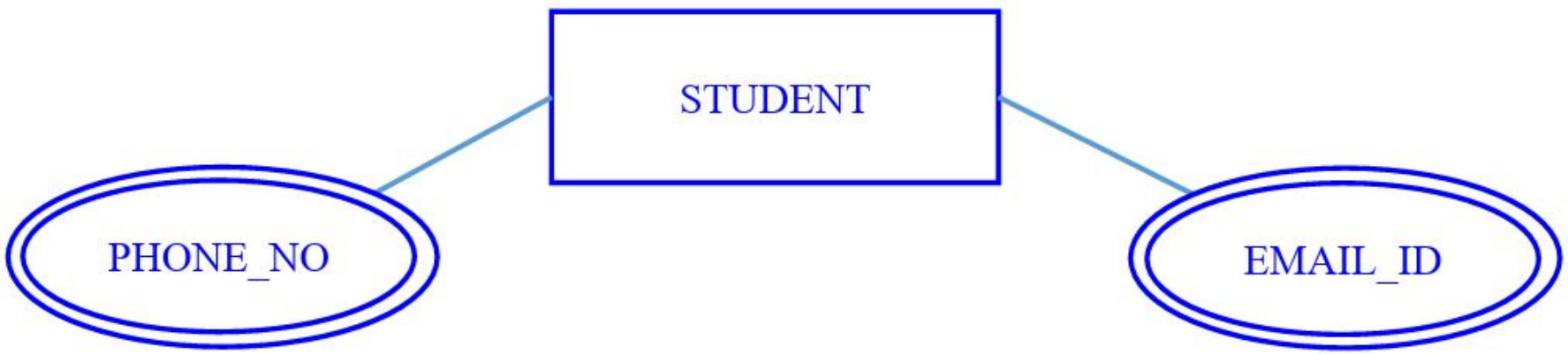
- ✔ Single valued attributes are those attributes which can take only one value for a given entity from an entity set.
- ✔ Example : Gender , DOB, Reg_No



S-6 SLO-1 & SLO-2 : Keys , Attributes and Constraints

Multi valued attributes

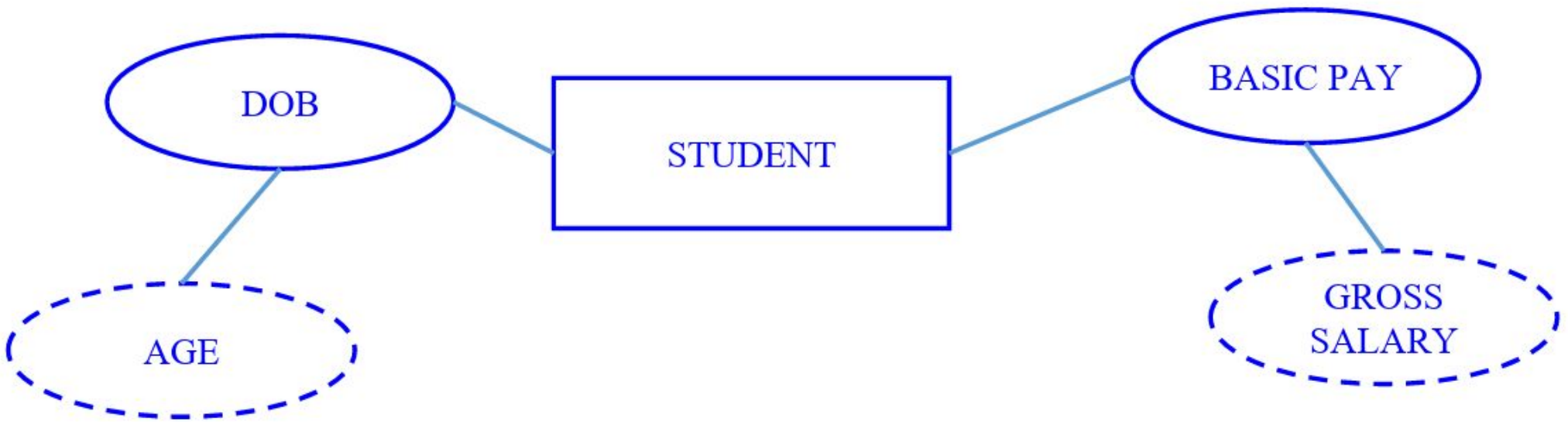
- ✔ Attributes can hold more than one values are called multi valued attribute
- ✔ Example : Phone_no, Email_id



S-6 SLO-1 & SLO-2 : Keys , Attributes and Constraints

Derived attributes

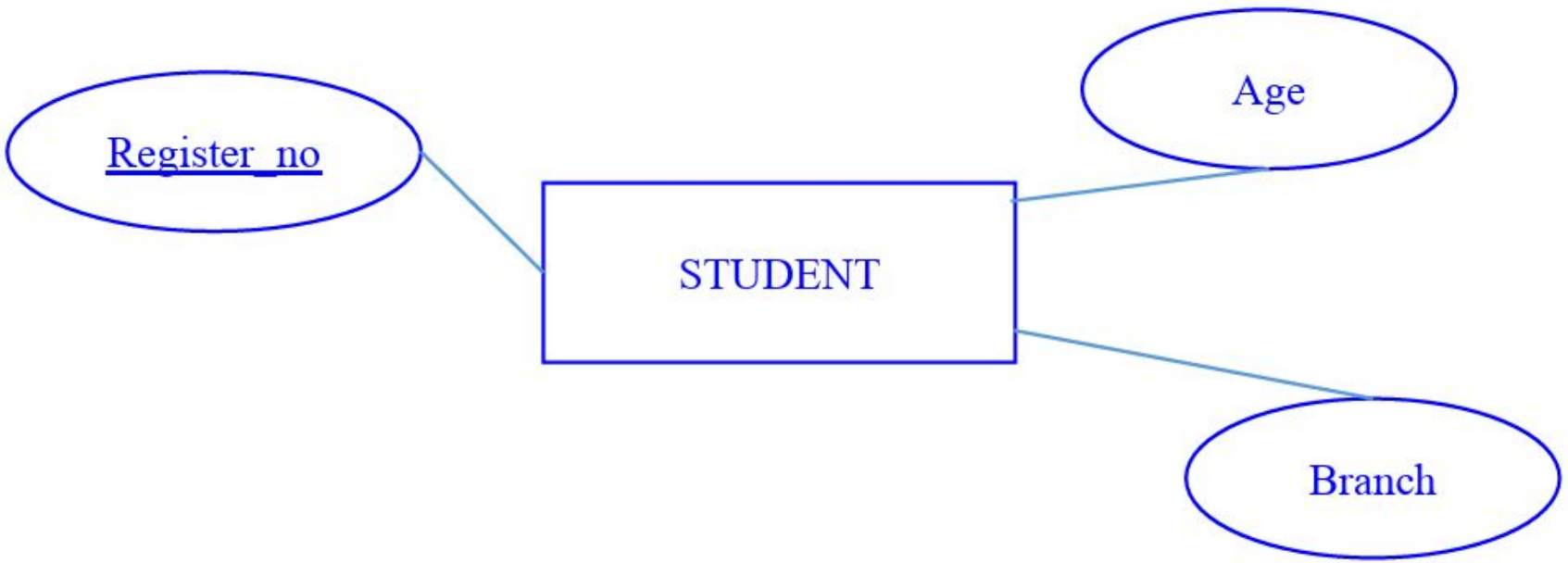
- ✔ A value which is derived from already existing value.
- ✔ It is not advisable to store such kind of values in database.
- ✔ The derived attributes represented by ellipse using dotted lines
- ✔ Example : Age , Gross Salary
- ✔ In the given figure below, Age is derived from DOB and Gross Salary derived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



S-6 SLO-1 & SLO-2 : Keys , Attributes and Constraints

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.



S-6 SLO-1 & SLO-2 : Keys , Attributes and Constraints

- ✔ 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

S-6 SLO-1 & SLO-2 : Keys , Attributes and Constraints

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 );
```

S-6 SLO-1 & SLO-2 : Keys , Attributes and Constraints

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);
```

S-6 SLO-1 & SLO-2 : Keys , Attributes and Constraints

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 constraints

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

S-6 SLO-1 & SLO-2 : Keys , Attributes and Constraints

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 );
```

S-6 SLO-1 & SLO-2 : Keys , Attributes and Constraints

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 geiven emp entity , primary key attribute is empno and foreign key is dept_id which is the primary key in dept entity.

S-6 SLO-1 & SLO-2 : Keys , Attributes and Constraints

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

S-7 SLO-1 & SLO-2 : Mapping Cardinality

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)
- ✔ Refer slide number 34 to 47 for a detailed note

S-8 SLO-1 & SLO-2 : Extended ER - Generalization, Specialization and Aggregation

Extended ER Features

- ✔ Basic ER Model is more than enough to model most of the Database Features.
- ✔ Extended ER model developed for some aspects of Database features more suitably expressed
- ✔ The followings are the Extended ER Features
  - Specialization
  - Generalization
  - Higher and lower level entity sets
  - Attribute inheritance
  - Aggregation
- ✔ To explain the above concepts, slightly more elaborate the schema for the university, by considering an entity set “person” with attributes “id”, “name”, and “address”

S-8 SLO-1 & SLO-2 : Extended ER - Generalization, Specialization and Aggregation

Specialization

- ✔ An entity set may include subgroupings of entities that are distinct in some way from other entities in the set.
- ✔ a subset of entities within an entity set may have attributes that are not shared by all the entities in the entity set.
- ✔ The E-R model provides a means for representing these distinctive entity groupings.
- ✔ The Entity set person may be further classified as one of the following:
  - Employee
  - Student

S-8 SLO-1 & SLO-2 : Extended ER - Generalization, Specialization and Aggregation

Specialization

- ✔ Both employee and student is described by a set of attributes that includes all the attributes of entity set person plus possibly additional attributes.
- ✔ For example, employee entities may be described further by the attribute salary, whereas student entities may be described further by the attribute fees.
- ✔ The process of designating subgroupings within an entity set is called specialization.
- ✔ The specialization of person allows us to distinguish among person entities according to whether they correspond to employees or students:
- ✔ In general, a person could be an employee, a student, both, or neither.

S-8 SLO-1 & SLO-2 : Extended ER - Generalization, Specialization and Aggregation

Specialization

- ✔ As another example, suppose the university divides students into two categories: Under graduate and Post graduate.
- ✔ Under graduate students have an office assigned to them. Post graduate students are assigned to a residential college.
- ✔ Each of these student types is described by a set of attributes that includes all the attributes of the entity set student plus additional attributes.
- ✔ The university could create two specializations of student, namely under graduate and post graduate.

S-8 SLO-1 & SLO-2 : Extended ER - Generalization, Specialization and Aggregation

Specialization

- ✔ We can apply specialization repeatedly to refine a design.
- ✔ For instance, university employees may be further classified as one of the following:
  - Faculty
  - Secretary
- ✔ Each of these employee types is described by a set of attributes that includes all the attributes of entity set employee plus additional attributes.
- ✔ For example, faculties entities may be described further by the attribute designation while secretary entities are described by the attribute hours per week.
- ✔ Further, secretary entities may participate in a relationship secretary for between the secretary and employee entity sets, which identifies the employees who are assisted by a secretary.

S-8 SLO-1 & SLO-2 : Extended ER - Generalization, Specialization and Aggregation

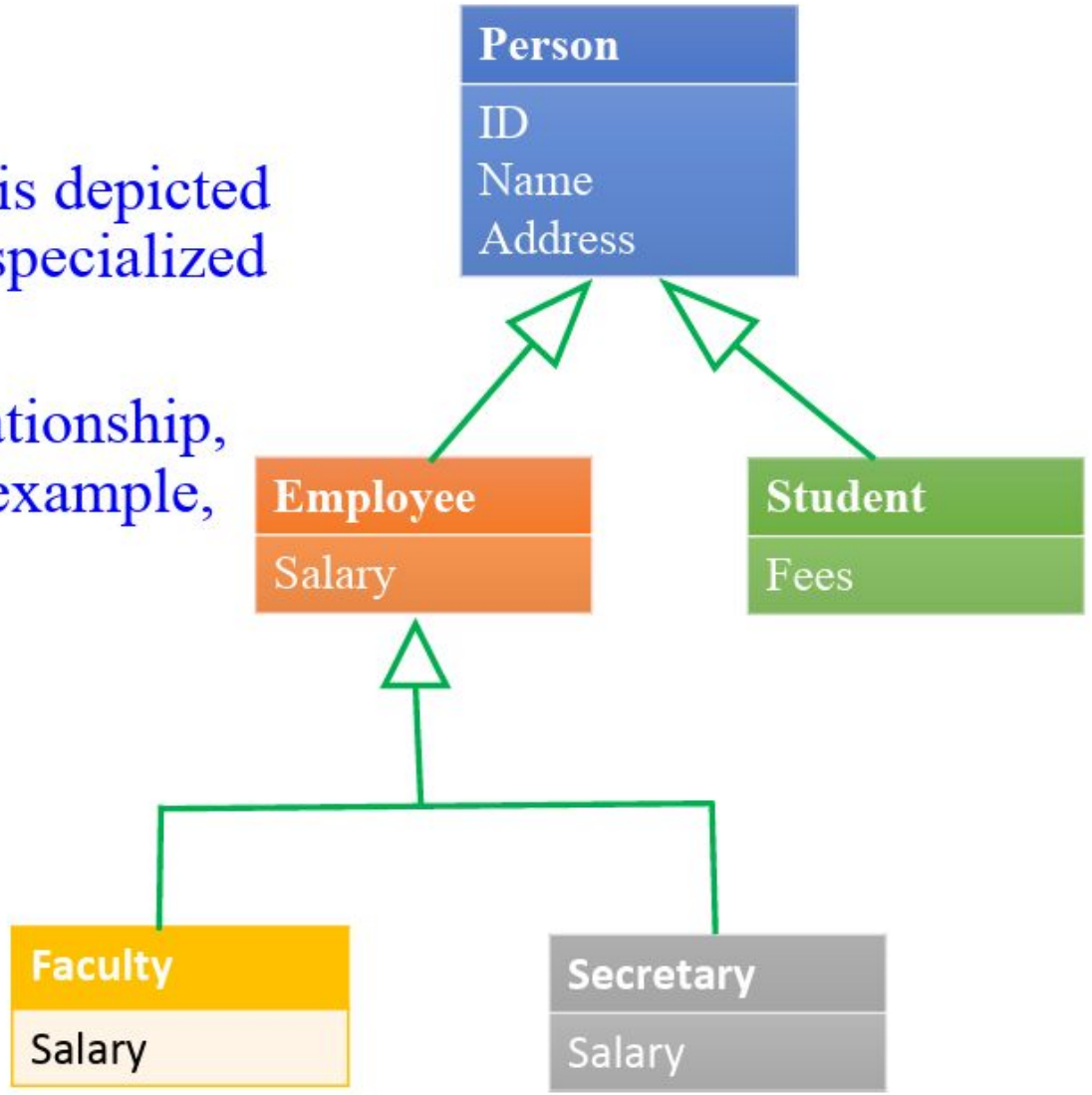
Specialization

- ✔ An entity set may be specialized by more than one distinguishing feature.
- ✔ In our example, the distinguishing feature among employee entities is the job the employee performs.
- ✔ Another, coexistent, specialization could be based on whether the person is a temporary employee or a permanent employee?
- ✔ Resulting in the entity sets temporary employee and permanent employee.

S-8 SLO-1 & SLO-2 : Extended ER - Generalization, Specialization and Aggregation

Specialization

- ✔ In terms of an E-R diagram, specialization is depicted by a hollow arrow-head pointing from the specialized entity to the other entity
- ✔ We refer to this relationship as the ISA relationship, which stands for "is a" and represents, for example, that an faculty "is a" employee.



S-8 SLO-1 & SLO-2 : Extended ER - Generalization, Specialization and Aggregation

Specialization

- ✔ Specialization represents in an E-R diagram depends on whether an entity may belong to multiple specialized entity sets or if it must belong to at most one specialized entity set.
- ✔ Multiple sets permitted is called overlapping specialization
- ✔ At most one permitted is called disjoint specialization.
- ✔ For an overlapping specialization (refer the figure in slide number 87 for student and employee as specializations of person), two separate arrows are used.
- ✔ For a disjoint specialization (refer the figure in slide number 87 for faculty and secretary as specializations of employee), a single arrow is used.
- ✔ The specialization relationship may also be referred to as a superclass-subclass relationship.

S-8 SLO-1 & SLO-2 : Extended ER - Generalization, Specialization and Aggregation

Generalization

- ✔ The refinement from an initial entity set into successive levels of entity subgroupings represents a top-down design process in which distinctions are made explicit.
- ✔ The design process may also proceed in a bottom-up manner, in which multiple entity sets are synthesized into a higher-level entity set on the basis of common features.
- ✔ The database designer may have first identified:
  - Faculty entity set with attributes Faculty_id, Faculty_name, Faculty_salary, and Faculty_Desig.
  - Secretary entity set with attributes secretary_id, secretary_name, secretary_salary, and hours_per_week.

S-8 SLO-1 & SLO-2 : Extended ER - Generalization, Specialization and Aggregation

Generalization

- ✔ There are some similarities are between the Faculty entity and Secretary entity, means several attributes that are conceptually the same across the two entity sets.
- ✔ For example, the identifier, name, and salary attributes are common between Faculty and Secretary entities.
- ✔ This commonality can be expressed by Generalization.
- ✔ Generalization is a containment relationship that exists between a higher-level entity set and one or more lower-level entity sets.
- ✔ In given example (slide number 85) ,employee is the higher-level entity set and faculty and secretary are lower-level entity sets.
- ✔ Higher- and lower-level entity sets also may be designated by the terms superclass and subclass, respectively.
- ✔ The person entity set is the superclass of the employee and student subclasses.

S-8 SLO-1 & SLO-2 : Extended ER - Generalization, Specialization and Aggregation

Attribute Inheritance

- ✔ An important property of the higher- and lower-level entities created by specialization and generalization is attribute inheritance.
- ✔ The attributes of the higher-level entity sets are said to be inherited by the lower-level entity sets.
  - Example, student and employee inherit the attributes of person.
- ✔ Student entity is described by its ID, name, and address attributes, and additionally a fees attribute.
- ✔ Employee is described by its ID, name, and address attributes, and additionally a salary attribute.
- ✔ Attribute inheritance applies through all tiers of lower-level entity sets.
  - Example : Faculty and Secretary, which are subclasses of employee, inherit the attributes ID, name, and address from person, in addition to inheriting the attribute salary from employee.

S-8 SLO-1 & SLO-2 : Extended ER - Generalization, Specialization and Aggregation

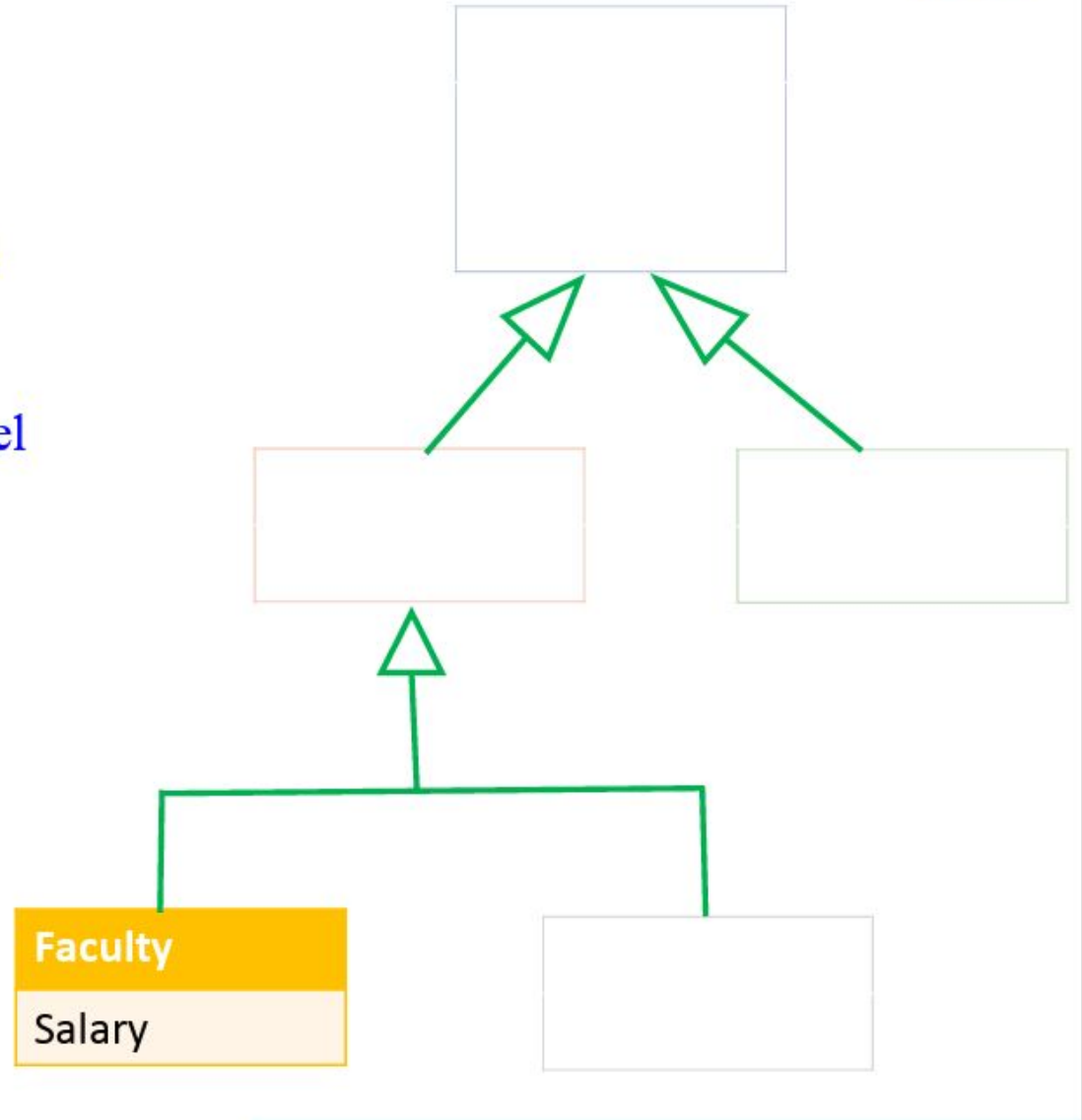
Attribute Inheritance

- ✔ An E-R model was arrived at by specialization or generalization, the outcome is basically the same:
  - A higher-level entity set with attributes and relationships that apply to all of its lower-level entity sets.
  - Lower-level entity sets with distinctive features that apply only within a particular lower-level entity set.

S-8 SLO-1 & SLO-2 : Extended ER - Generalization, Specialization and Aggregation

Attribute Inheritance

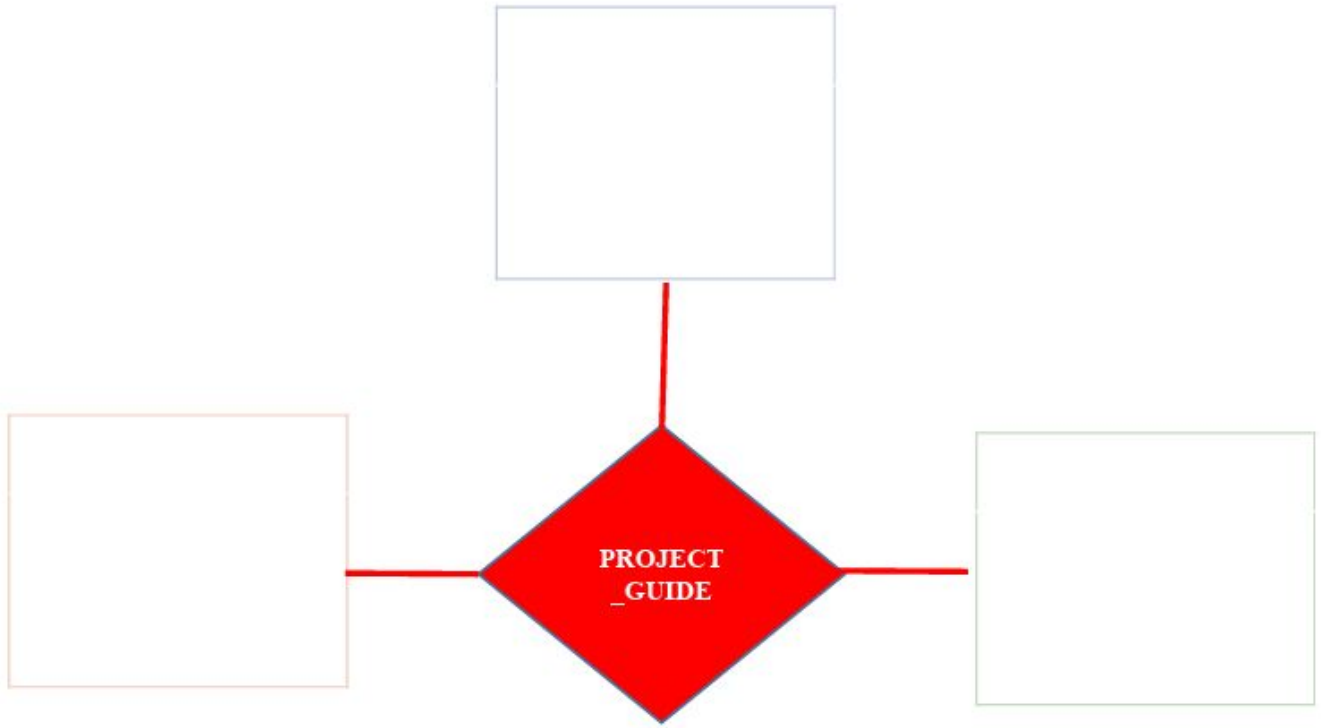
- The given Figure describes a hierarchy of entity sets.
- In the figure, employee is a lower-level entity set of person and a higher-level entity set of the faculty and secretary entity sets.
- In a hierarchy, a given entity set may be involved as a lower- level entity set in only one ISA relationship; that is, entity sets in this diagram have only single inheritance.
- If an entity set is a lower-level entity set in more than one ISA relationship, then the entity set has multiple inheritance, and the resulting structure is said to be a lattice.



S-8 SLO-1 & SLO-2 : Extended ER - Generalization, Specialization and Aggregation

Aggregation

- One limitation of the E-R model is that it cannot express relationships among relationships.
- To illustrate the need for such a construct, consider the ternary relationship project_guide, between an faculty, student and project



S-8 SLO-1 & SLO-2 : Extended ER - Generalization, Specialization and Aggregation

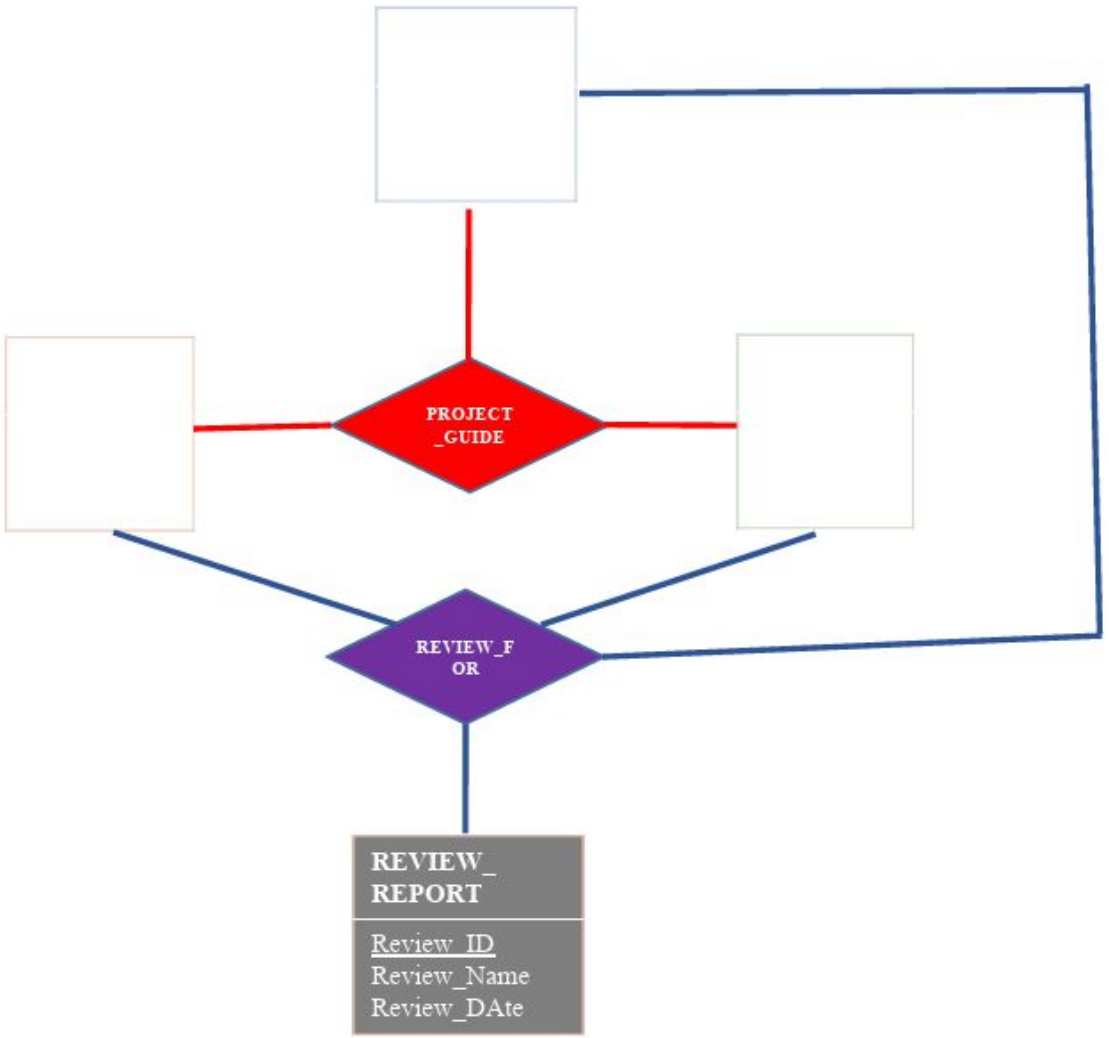
Aggregation

- ✔ Consider that , each faculty guiding a student on a project is required to file a monthly review report.
- ✔ We model the review report as an entity review_report, with a primary key review_id.
- ✔ One alternative for recording the (student, project, faculty) combination to which a review corresponds is to create a quaternary (4-way) relationship set review_for between faculty , student, project, and review_report evaluation.
- ✔ A quaternary relationship is required—a binary relationship between student and review report, for example, would not permit us to represent the (project, faculty) combination to which a review_ report corresponds.

S-8 SLO-1 & SLO-2 : Extended ER - Generalization, Specialization and Aggregation

Aggregation

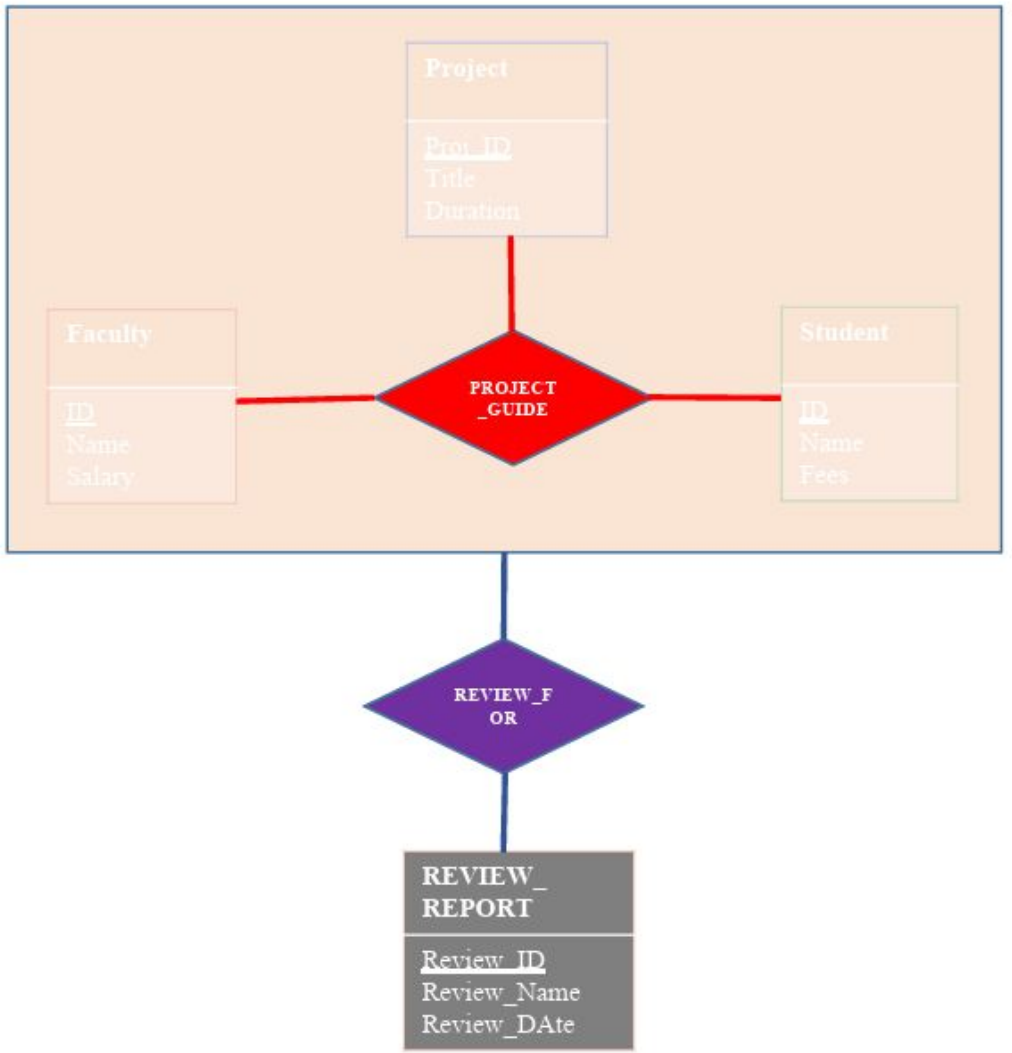
- ✔ Using the basic E-R modeling constructs, the following E-R diagram for the above constraints is obtained
- ✔ This diagram with redundant relationships



S-8 SLO-1 & SLO-2 : Extended ER - Generalization, Specialization and Aggregation

Aggregation

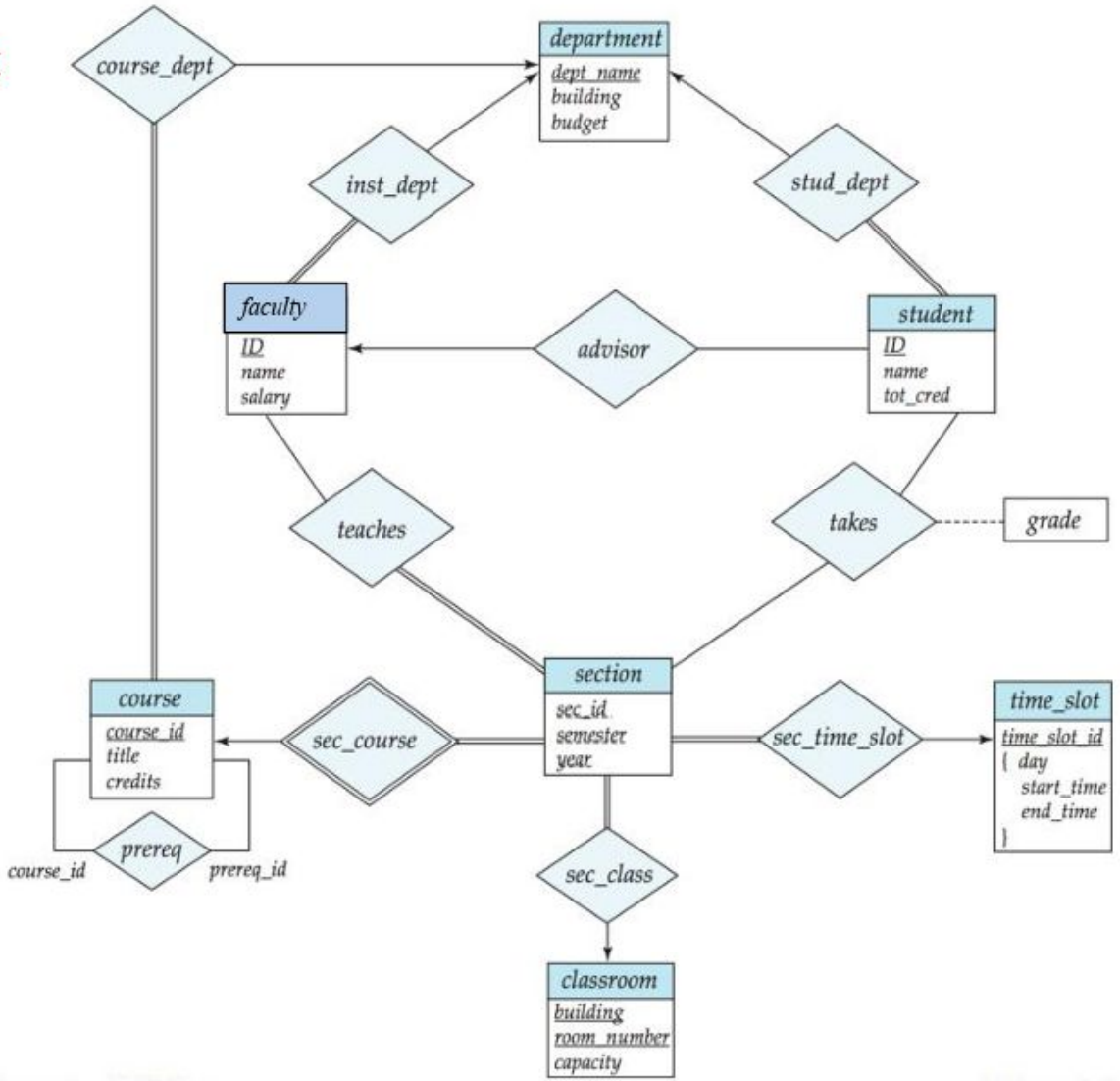
- ✔ The best way to model a situation such as the one just described is to use aggregation.
- ✔ Aggregation is an abstraction through which relationships are treated as higher-level entities.
- ✔ In the given example, the relationship set project_ guide (relating the entity sets faulty, student, and project) as a higher-level entity set called project_guide.
- ✔ Such an entity set is treated in the same manner as is any other entity set.
- ✔ We can then create a binary relationship review_for between project_guide and review report to represent which (student, project, faculty) combination an review_report is for.
- ✔ Figure shows a notation for aggregation commonly used to represent this situation.



S-9-10 SLO-1 & SLO-2 : Lab 5: Construct a ER Model for the application to be constructed to a Database

Sample ER Diagram for University Management System

- ✔Discuss briefly about E-R Diagram
- ✔Give the E-R Diagram Notations
- ✔List the schema participated in university
- ✔List the relationship sets
- ✔List the required constraints
- ✔Draw the E-R Diagram



S-11 SLO-1 : ER Diagram Issues

- ✔ The notions of an entity set and a relationship set are not precise.
- ✔ It is possible to define a set of entities and the relationships among them in a number of different ways.
- ✔ The followings are the basic issues in ER Diagram
  - Use of Entity Sets versus Attributes
  - Use of Entity Sets versus Relationship Sets
  - Binary versus n-ary Relationship Sets
  - Placement of Relationship Attributes

S-11 SLO-1 : ER Diagram Issues

Use of Entity Sets versus Attributes

- ✔ Consider the entity set faculty with the additional attribute phone_no , (Figure a)
- ✔ The considering phone as a separate entity , with attributes phone_no and location.
- ✔ The location may be office or home or mobile
- ✔ In this case , the attribute phone_no do not add to the faculty entity
- ✔ The following may consider
  - A phone entity set with attributes phone number and location.
  - A relationship set faculty_phone, denoting the association between faculty and the phones that they have. (Figure b)

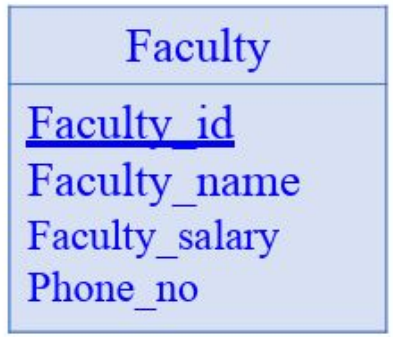


Figure a

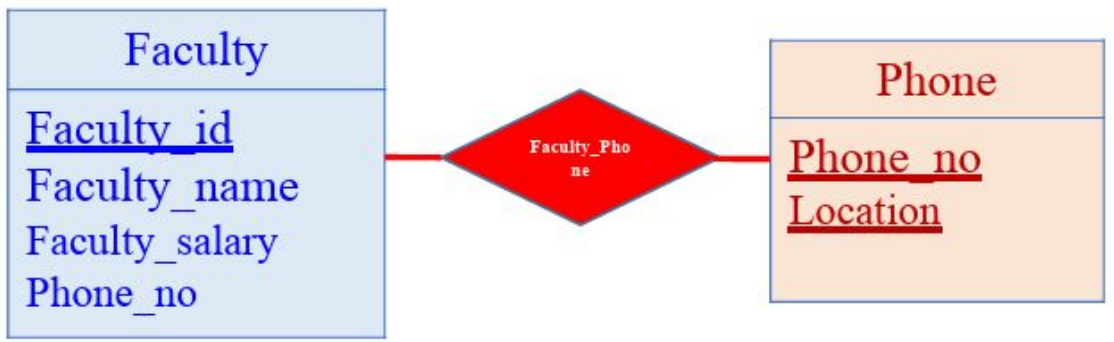


Figure b

S-11 SLO-1 : ER Diagram Issues

Use of Entity Sets versus Attributes

- ✔ Treating a phone as an attribute phone number implies that faculty have precisely one phone number each.
- ✔ Treating a phone as an entity phone permits faculty to have several phone numbers (including zero) associated with them.
- ✔ However, we could instead easily define phone number as a multivalued attribute to allow multiple phones per faculty.
- ✔ The main difference then is that treating a phone as an entity better models a situation where one may want to keep extra information about a phone, such as its location, or its type like mobile, office, old phone, etc.,

S-11 SLO-1 : ER Diagram Issues

Use of Entity Sets versus Relationship Sets

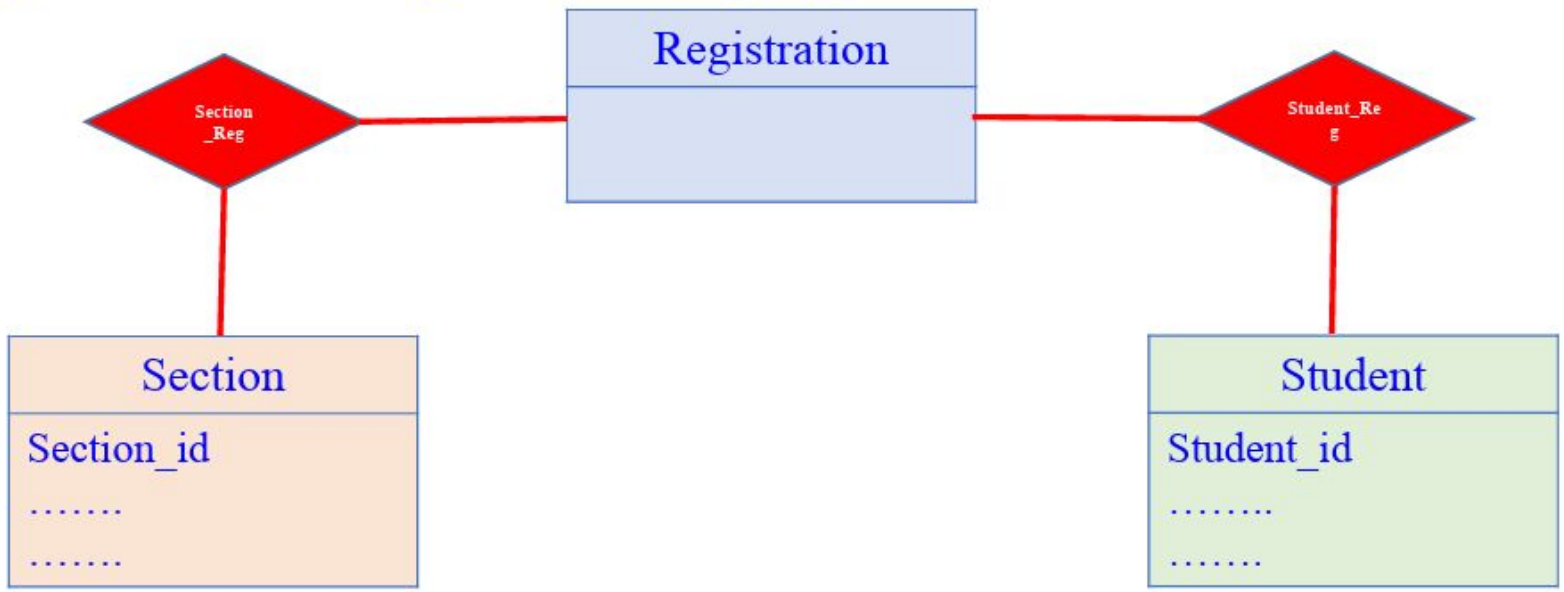
- ✔ It is not always clear whether an object is best expressed by an entity set or a relationship set.
- ✔ In ER diagram for University Management system, we used the takes relationship set to model the situation where a student takes a (section of a) course.
- ✔ An alternative is to imagine that there is a course-registration record for each course that each student takes.
- ✔ Then need to have an entity set to represent the course-registration record.
- ✔ Let us call that entity set registration. Each registration entity is related to exactly one student and to exactly one section,
- ✔ Have two relationship sets, one to relate course registration records to students and one to relate course-registration records to sections.

S-11 SLO-1 : ER Diagram Issues

Use of Entity Sets versus Relationship Sets

✔ In the given Figure , we show the entity sets section and student from ER diagram for University Management System with the takes relationship set replaced by one entity set and two relationship sets:

- registration, the entity set representing course-registration records.
- section reg, the relationship set relating registration and course.
- student reg, the relationship set relating registration and student.



S-11 SLO-1 : ER Diagram Issues

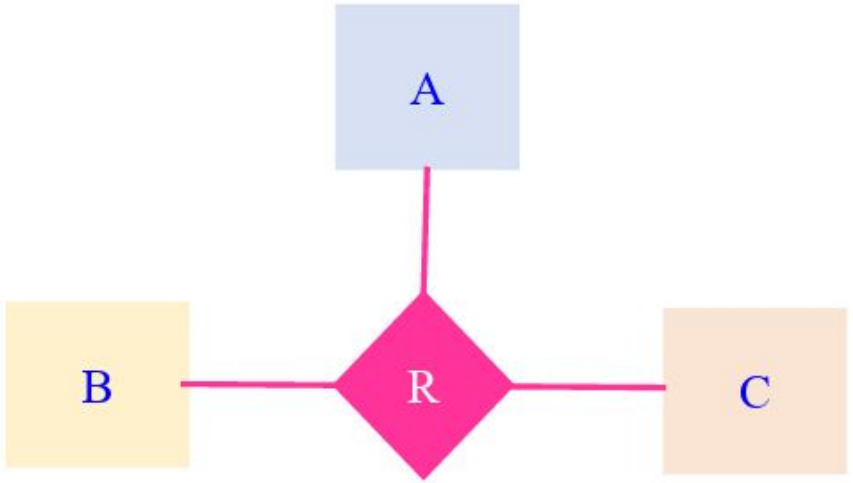
Use of Entity Sets versus Relationship Sets

- ✔ Relationships in databases are often binary.
- ✔ Some relationships that appear to be nonbinary could actually be better represented by several binary relationships.
- ✔ For instance, one could create a ternary relationship parent, relating a child to his/her mother and father.
- ✔ However, such a relationship could also be represented by two binary relationships, mother and father, relating a child to his/her mother and father separately.
- ✔ it is always possible to replace a nonbinary (n-ary, for $n > 2$) relationship set by a number of distinct binary relationship sets.

S-11 SLO-1 : ER Diagram Issues

Use of Entity Sets versus Relationship Sets

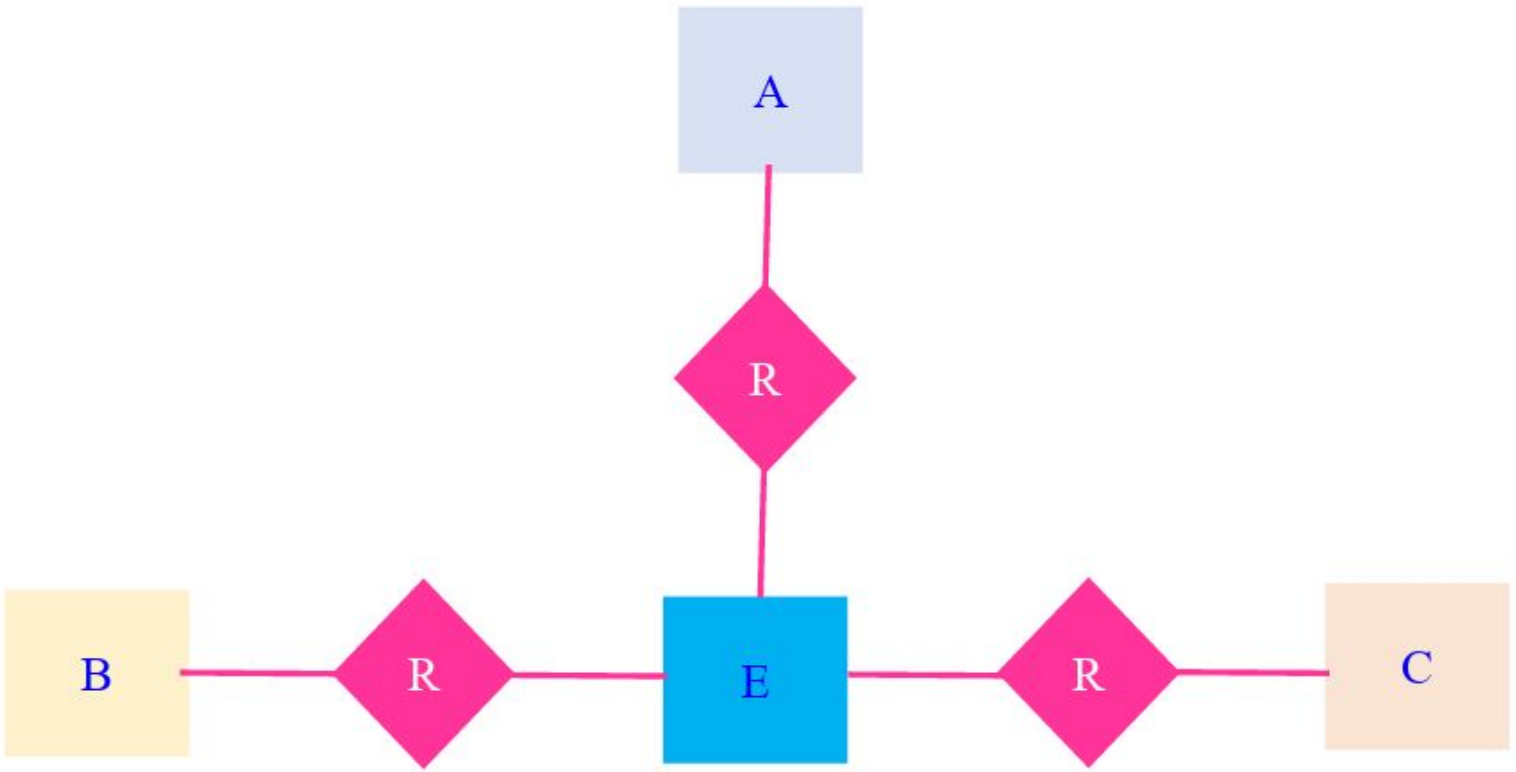
- Consider the abstract ternary (n = 3) relationship set R, relating entity sets A, B, and C. We replace the relationship set R by an entity set E, and create three relationship sets as shown in Figure below.
  - RA, relating E and A.
  - RB, relating E and B.
  - RC, relating E and C.



S-11 SLO-1 : ER Diagram Issues

Use of Entity Sets versus Relationship Sets

- ✔ If the relationship set R had any attributes, these are assigned to entity set E.
- ✔ Further, a special identifying attribute is created for E For each relationship (ai, bi, ci) in the relationship set R, we create a new entity ei in the entity set E.
- ✔ Then, in each of the three new relationship sets, we insert a relationship as follows:
  - (ei, ai) in RA.
  - (ei, bi) in RB.
  - (ei, ci) in RC.



S-11 SLO-1 : ER Diagram Issues

Placement of Relationship Attributes

- ✔ The cardinality ratio of a relationship can affect the placement of relationship attributes.
- ✔ Thus, attributes of one-to-one or one-to-many relationship sets can be associated with one of the participating entity sets, rather than with the relationship set.
- ✔ For instance, let us specify that counselor is a one-to-many relationship set such that one faculty may advise several students, but each student can be counseled only a single faculty.

S-11 SLO-1 : ER Diagram Issues

Placement of Relationship Attributes

- In this case, the attribute date, which specifies when the faculty became the counselor of a student, could be associated with the student entity set, as Figure below depicts.

Here's the updated Student table:

Student_ID	Student_Name	Student_Major	Student_Advisor	Counseling_Date
201	Sarah Parker	Biology	101	3 Jan 2022
202	Michael Chen	Economics	101	15 Feb 2022
203	Lisa Davis	Psychology	102	10 Mar 2022

In this updated table, the "Counseling_Date" column specifies when each student was counseled by their advisor. For example, "Sarah Parker" was counseled by their advisor on January 3, 2022, while "Michael Chen" was counseled on February 15, 2022, and "Lisa Davis" on March 10, 2022.

S-12 SLO-1 & SLO-2 : 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

S-12 SLO-1 & SLO-2 : Relational Model

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

faculty_id	faculty_name	dept_name	salary
101	John Smith	Computer Science	$70,000
102	Emily Johnson	Mathematics	$65,000
103	David Lee	Physics	$75,000

- ✔ This table represents information about faculty members in a university system. Each row corresponds to a different faculty member, with columns indicating their unique identifier (faculty_id), name (faculty_name), department (dept_name), and salary.

S-12 SLO-1 & SLO-2 : Relational Model

Structure of Relational Databases

- Consider the following table Course, which stores the information about course details like course_code, title, dept_name, credits

course_code	title	dept_name	credits
CS101	Introduction to Computer Science	Computer Science	4
MATH202	Calculus II	Mathematics	3
PHY301	Quantum Mechanics	Physics	5

- This table stores information about various courses offered by a university. Each row represents a different course, with columns indicating the course code (course_code), title (title), department offering the course (dept_name), and number of credits (credits).

S-12 SLO-1 & SLO-2 : Relational Model

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.

course_code	prereq_code
CS101	CS100
MATH202	MATH101
PHY301	PHY202
ENG101	ENG100
BIO202	BIO101
CHEM301	CHEM202

✔ This table represents the prerequisite courses for each course. Each row indicates a prerequisite relationship between a course and its prerequisite, with columns indicating the course code (course_code) and the prerequisite code (prereq_code). For example, CS101 has a prerequisite of CS100, MATH202 has a prerequisite of MATH101, and PHY301 has a prerequisite of PHY202.

S-12 SLO-1 & SLO-2 : Relational Model

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.

S-12 SLO-1 & SLO-2 : Relational Model

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.

S-12 SLO-1 & SLO-2 : Relational Model

Database Schema

✔ Consider the Department relation

dept_name	location	budget
CS	Building A	$500,000
Mathematics	Building B	$400,000
Physics	Building C	$600,000
Biology	Building D	$450,000
Chemistry	Building E	$550,000
History	Building F	$300,000

✔ The schema for that relation is

department (dept_name, location, budget)

S-12 SLO-1 & SLO-2 : Relational Model

- 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 slot id)
- ✔To describe the association between faculty and the class sections that they teach. teaches (faculty_id, course id, sec id, semester, year)

S-12 SLO-1 & SLO-2 : Relational Model

Database Schema

Section relation

Course_code	Sec_id	Semester	Year	Location	Room_no	Time_slot_id
CS101	1	Fall	2023	Building A	101	1
MATH202	2	Spring	2024	Building B	201	2
PHY301	1	Fall	2023	Building C	301	3
ENG101	1	Spring	2023	Building D	102	4
BIO202	2	Fall	2024	Building E	202	5
CHEM301	1	Spring	2023	Building F	103	6

S-12 SLO-1 & SLO-2 : Relational Model

Database Schema

Teaches Relation

Faculty_id	Course_Code	Sec_id	Semester	Year
101	CS101	1	Fall	2023
102	MATH202	2	Spring	2024
103	PHY301	1	Fall	2023
104	ENG101	1	Spring	2023
105	BIO202	2	Fall	2024
106	CHEM301	1	Spring	2023

S-12 SLO-1 & SLO-2 : Relational Model

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)

S-12 SLO-1 & SLO-2 : Relational Model

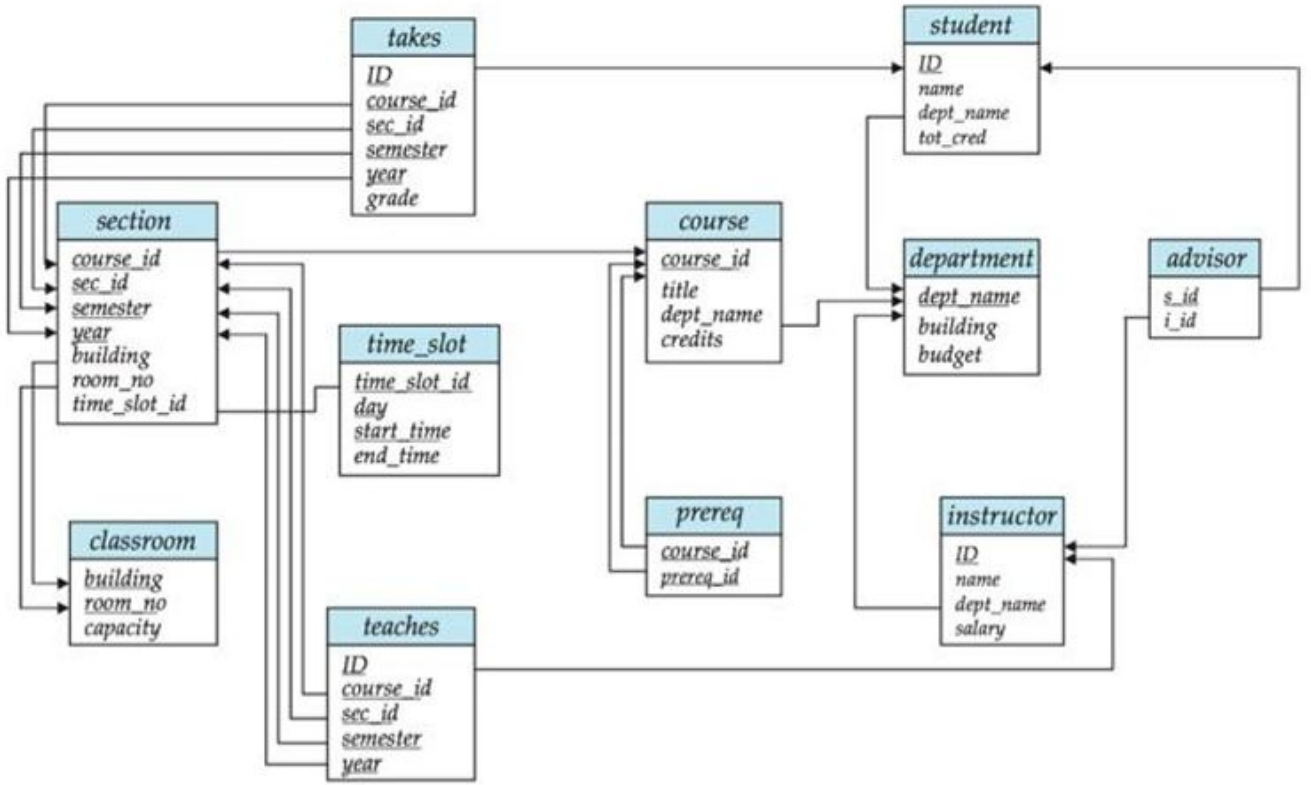
Keys

- ✔ One or more attributes used to identify an entity uniquely in an entity set if 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

S-12 SLO-1 & SLO-2 : Relational Model

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.



S-12 SLO-1 & SLO-2 : Relational Model

Relational Query Languages

- ✔ A query language is a language in which a user requests info
- ✔ 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.

S-13 SLO-1 & SLO-2 : Conversion of ER to Relational Table

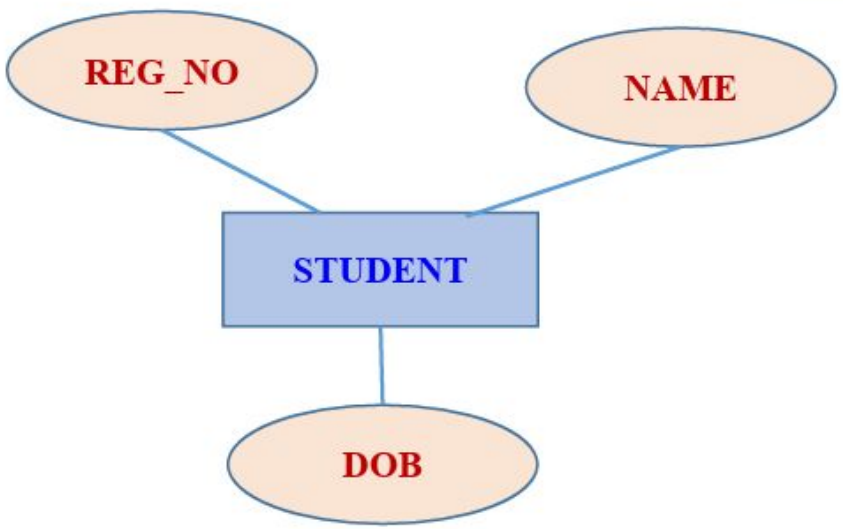
- ✔ After completing the ER diagram
- ✔ ER diagram is converted into the tables in relational model
- ✔ Relational model can be easily implemented in ORACLE, MYSQL , etc.,
- ✔ The below points to be considered for converting ER diagram into tables.
  - Strong Entity Set With Only Simple Attributes
  - Strong Entity Set With Composite Attributes
  - Strong Entity Set With Multi Valued Attributes
  - Translating Relationship Set into a Table
  - Binary Relationships With Cardinality Ratios
  - Binary Relationship With Both Cardinality Constraints and Participation Constraints
  - Binary Relationship With Weak Entity Set

S-13 SLO-1 & SLO-2 : Conversion of ER to Relational Table

Strong Entity Set With Only Simple Attributes

- ✔ A strong entity set with only simple attributes will require only one table in relational model.
- ✔ Attributes of the table will be the attributes of the entity set.
- ✔ The primary key of the table will be the key attribute of the entity set.

ER DIAGRAM



Er Diagram to relational table

RELATIONAL TABLE

REG_NO	NAME	DOB

Schema : student (reg_no, name, dob)

S-13 SLO-1 & SLO-2 : Conversion of ER to Relational Table

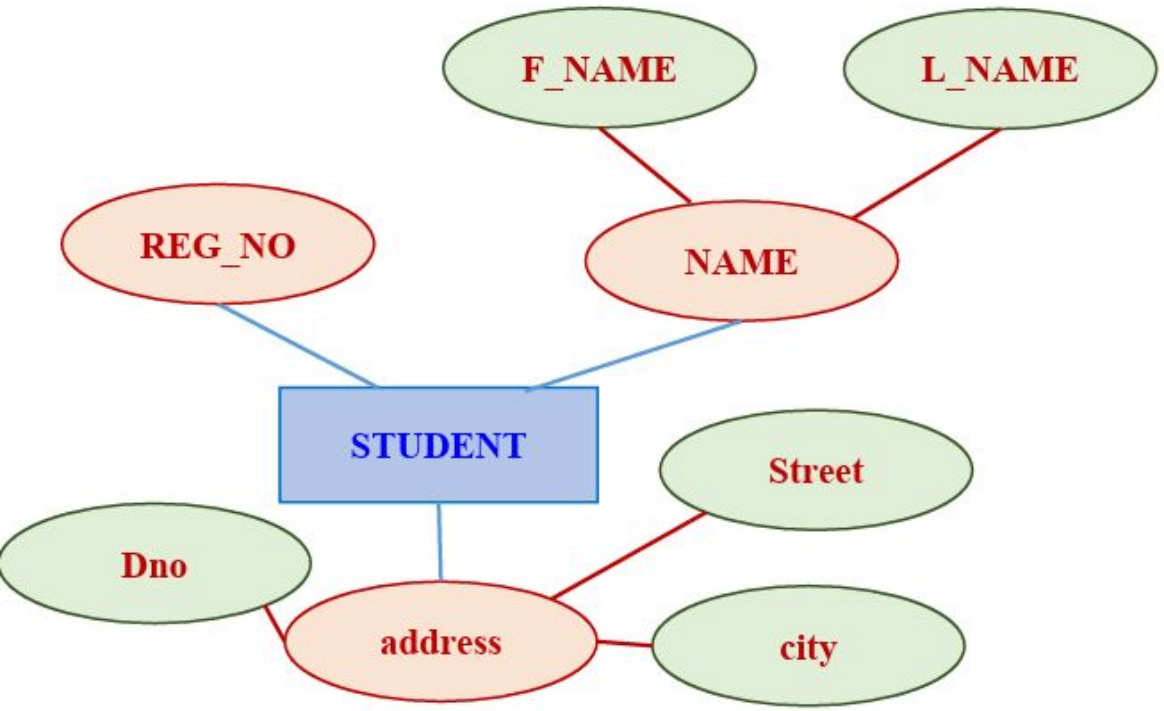
Strong Entity Set With Composite Attributes

- ✔ A strong entity set with any number of composite attributes will require only one table in relational model.
- ✔ While conversion, simple attributes of the composite attributes are taken into account and not the composite attribute itself.

ER DIAGRAM

Er Diagram to relational table

RELATIONAL TABLE



Reg_no	F_name	L_name	dno	street	city

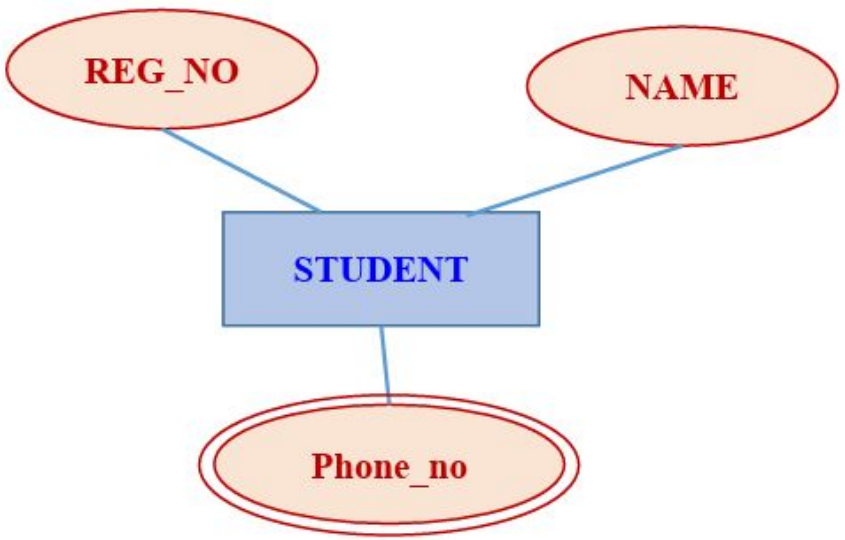
Schema : student (reg_no, f_name, l_name, dno, street, city)

S-13 SLO-1 & SLO-2 : Conversion of ER to Relational Table

Strong Entity Set With Multi Valued Attributes

- ✔ A strong entity set with any number of multi valued attributes will require two tables in relational model.
- ✔ One table will contain all the simple attributes with the primary key.
- ✔ Other table will contain the primary key and all the multi valued attributes.

ER DIAGRAM



Er Diagram to relational table

RELATIONAL TABLE

REG_NO	NAME

Schema : student (reg_no, name)

REG_NO	Phone_no

Schema : student (reg_no, phone_no)

S-13 SLO-1 & SLO-2 : Conversion of ER to Relational Table

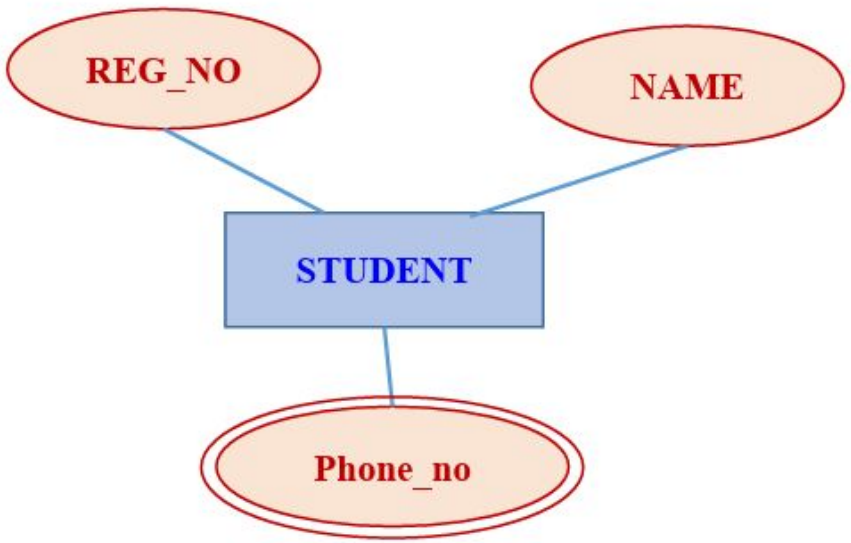
Strong Entity Set With Multi Valued Attributes

✔ Also the stronger entity set with any number of multi valued attributes may be converted as shown below

ER DIAGRAM

Er Diagram to relational table

RELATIONAL TABLE



reg_no	name	phone_no1	phone_no2	Phone_no3

Schema : student (reg_no, name, phone_no1 ,phone_no2, phone_no3)

S-13 SLO-1 & SLO-2 : Conversion of ER to Relational Table

Translating Relationship Set into a Table

- ✔ A relationship set will require one table in the relational model.
- ✔ Attributes of the table are :
  - Primary key attributes of the participating entity sets
  - Its own descriptive attributes if any.
- ✔ Set of non-descriptive attributes will be the primary key.

NOTE

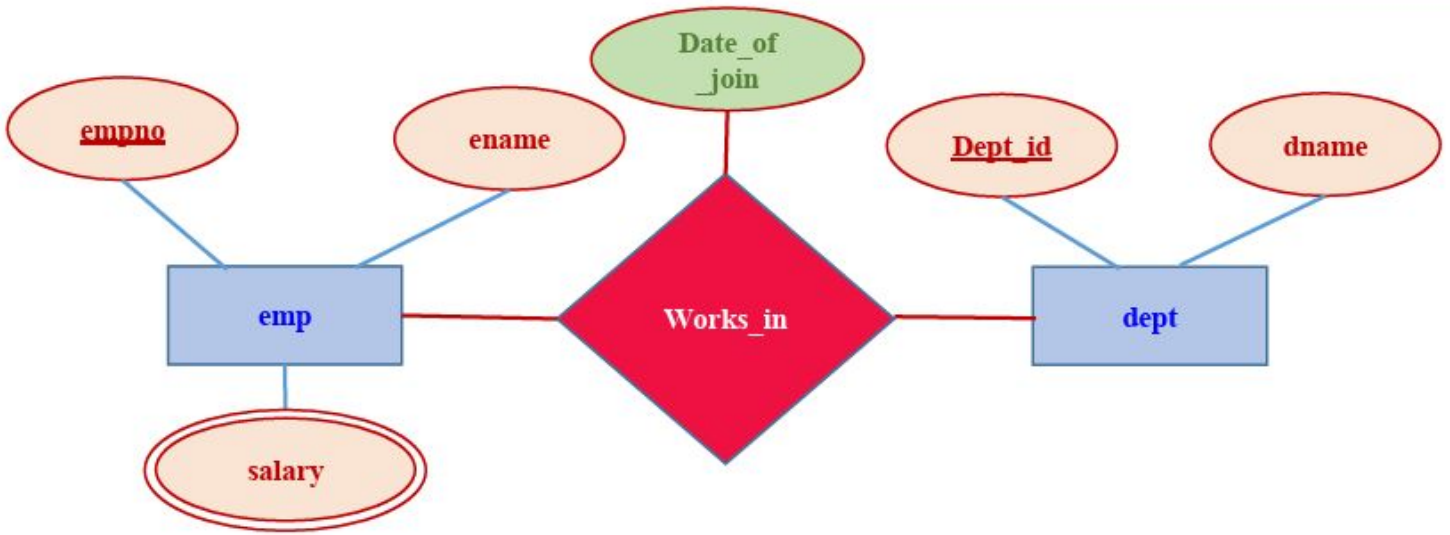
✔If we consider the overall ER diagram, three tables will be required in relational model

1. Emp
2. Dept
3. Works_in

ER DIAGRAM

Er Diagram to relational table

RELATIONAL TABLE



empno	dept_id	date_of_join

Schema : works_in (empno, dept_id, date_of_join)

S-13 SLO-1 & SLO-2 : Conversion of ER to Relational Table

Binary Relationships With Cardinality Ratios

✔ Four types are possible

1. Binary relationship with cardinality ratio 1:1
2. Binary relationship with cardinality ratio 1:m
3. Binary relationship with cardinality ratio m:1
4. Binary relationship with cardinality ratio m:m

S-13 SLO-1 & SLO-2 : Conversion of ER to Relational Table

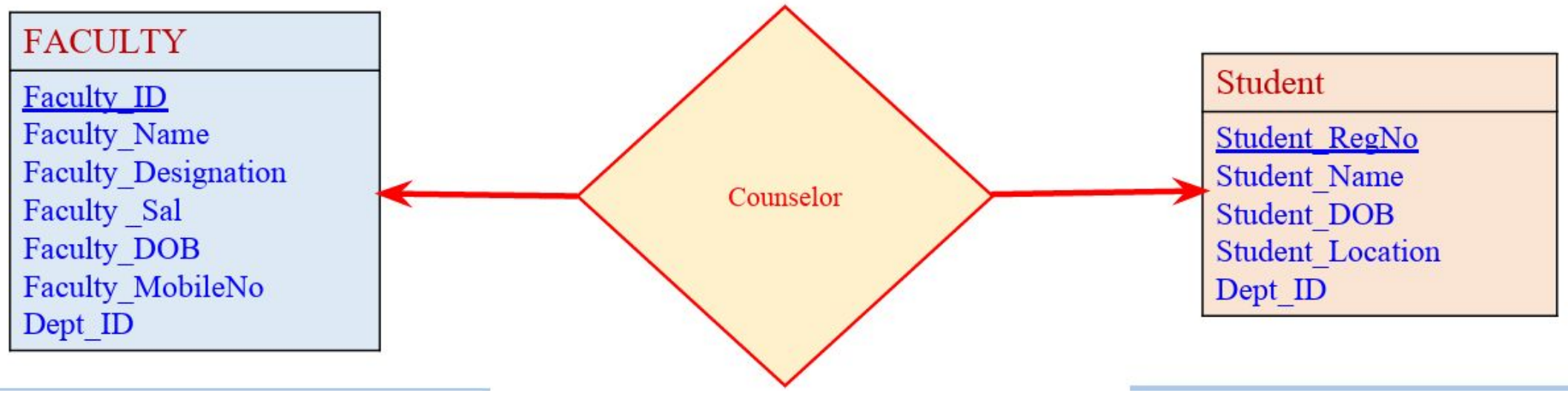
Mapping Cardinality

- ✔ 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 (→) or an undirected line (—) between the relationship set and the entity.

Binary Relationships With Cardinality Ratios

- ✔ Binary relationship with cardinality ratio 1:1

  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.

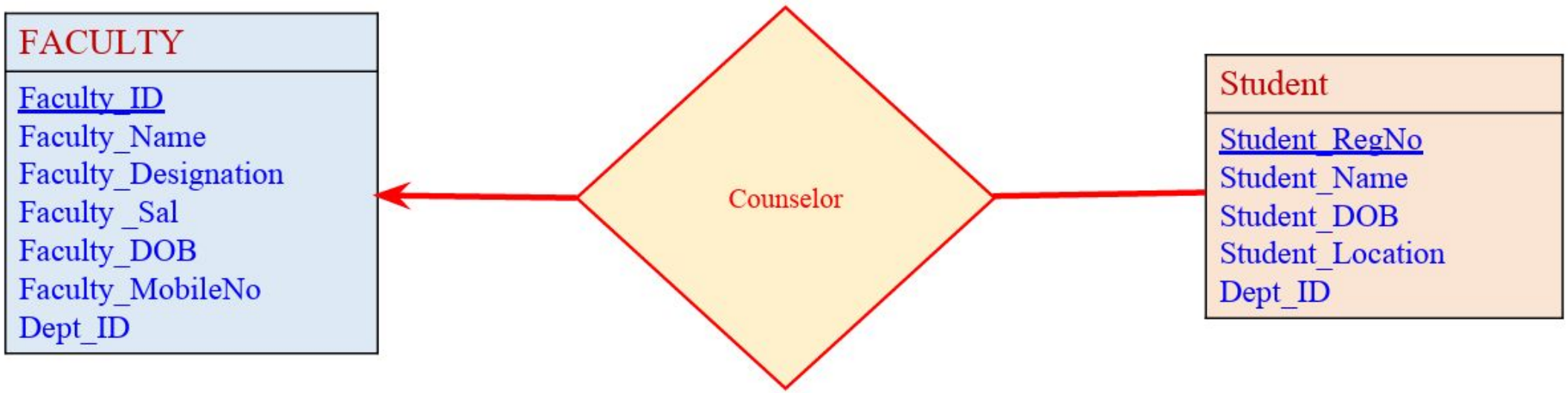


S-13 SLO-1 & SLO-2 : Conversion of ER to Relational Table

Binary Relationships With Cardinality Ratios

✔ Binary relationship with cardinality ratio 1:m

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.

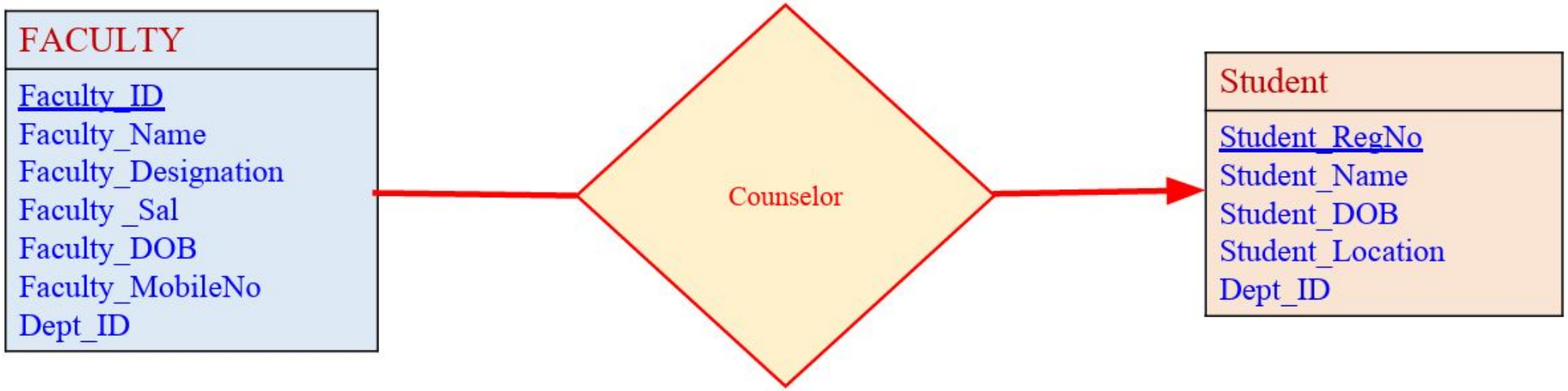


S-13 SLO-1 & SLO-2 : Conversion of ER to Relational Table

Binary Relationships With Cardinality Ratios

✔ Binary relationship with cardinality ratio m:1

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.

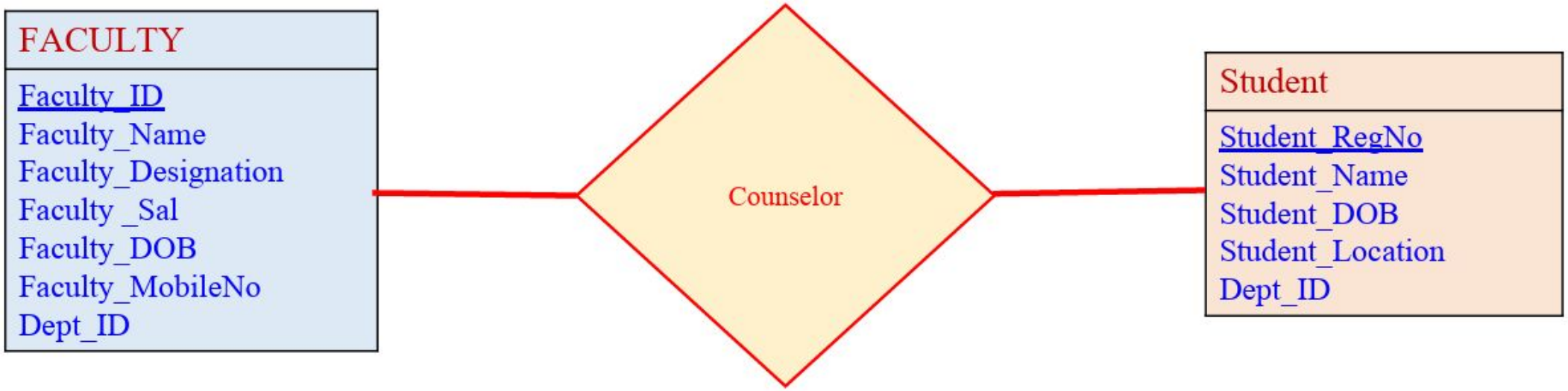


S-13 SLO-1 & SLO-2 : Conversion of ER to Relational Table

Binary Relationships With Cardinality Ratios

✔ Binary relationship with cardinality ratio m:m

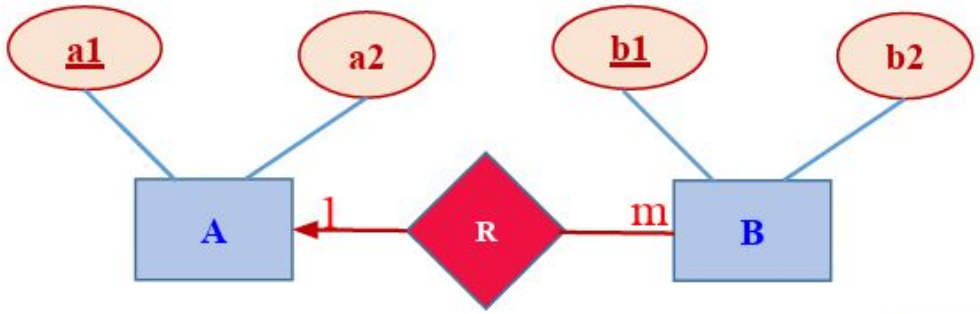
We draw an undirected line from the relationship set counselor to both entity sets faculty and student as shown in the below figure, indicates that a faculty may counsel many students, and a student may have many counselor.



S-13 SLO-1 & SLO-2 : Conversion of ER to Relational Table

Binary Relationship With Both Cardinality Constraints and Participation Constraints

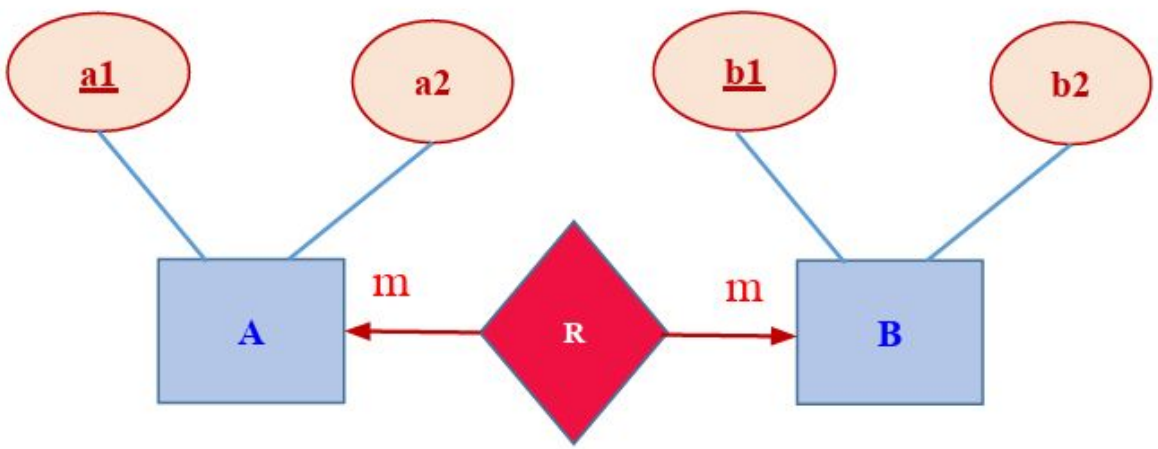
- Because of the total participation constraint, foreign key acquires NOT NULL constraint
- Now foreign key can not be null.
- Option 1: For Binary Relationship With Cardinality Constraint and Total Participation Constraint From One Side
  - Because cardinality ratio = 1 : n , so we will combine the entity set B and relationship set R.
  - Then, two tables will be required-
    - A (a1 , a2)
    - BR (a1 , b1 , b2)
  - Because of total participation, foreign key a1 has acquired NOT NULL constraint, so it can't be null now.



S-13 SLO-1 & SLO-2 : Conversion of ER to Relational Table

Binary Relationship With Both Cardinality Constraints and Participation Constraints

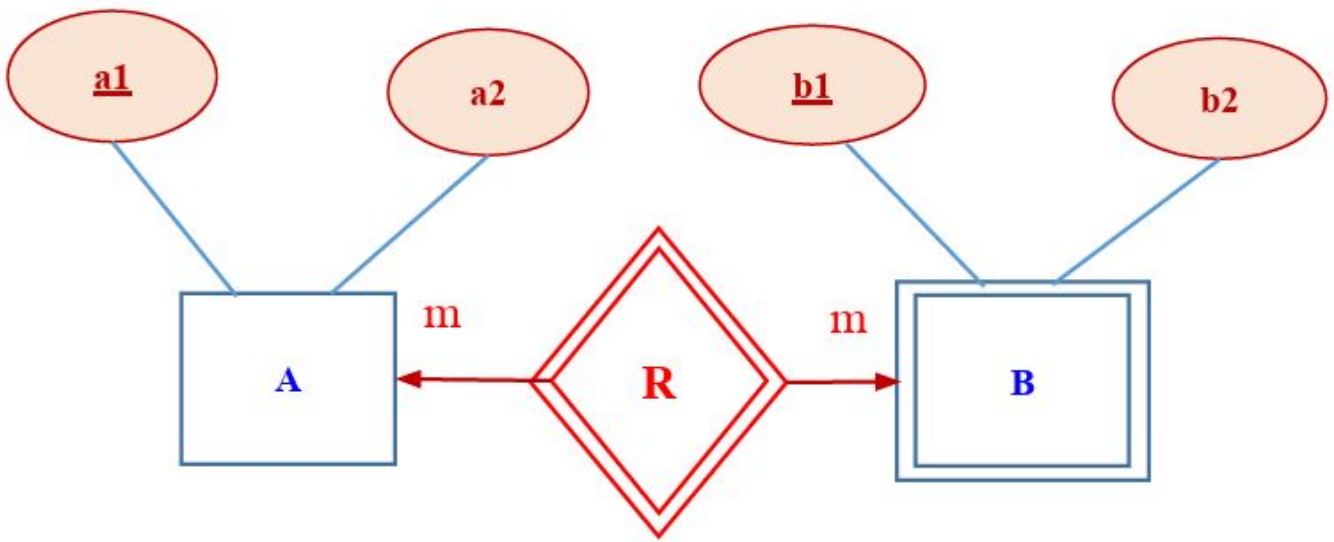
- ✔ Option 2: Binary Relationship With Cardinality Constraint and Total Participation Constraint From Both Sides
  - ✔ If there is a key constraint from both the sides of an entity set with total participation, then that binary relationship is represented using only single table.
    - ARB (a1 , a2 , b1 , b2)



S-13 SLO-1 & SLO-2 : Conversion of ER to Relational Table

Binary Relationship With Weak Entity Set

- ✔ Weak entity set always appears in association with identifying relationship with total participation constraint.
- ✔ Here, two tables will be required-
  - A (a1 , a2)
  - BR (a1 , b1 , b2)



S-14-15 SLO-1 & SLO-2 :

Lab 6: Nested Queries on sample exercise commands to the sample exercises

✔ Guidelines for Subqueries
- Enclose subqueries in parentheses.
- Place subqueries on the right side of the comparison operator.
- Do not add an ORDER BY clause to a subquery.
- Use single-row operators with single-row subqueries.
- Use multiple-row operators with multiple-row subqueries

✔ Single-Row Subqueries
- Return only one row
- Use single-row comparison operators (ie; relational operators)

✔ Multiple-Row Subqueries
- Return more than one row
- Use multiple-row comparison operators

S-14-15 SLO-1 & SLO-2 :
Lab 6: Nested Queries on sample exercise

Answer the following queries using EMP table

Q1) List the name of the employees whose salary is greater than that of employee with empno 7566.

Q2) List the name of the employees whose job is equal to the job of employee with empno 7369 and salary is greater than that of employee with empno 7876.

Q3) List the ename, job, sal of the employee who get minimum salary in the company

Q4) List deptno & min(salary) department wise, only if min(sal) is greater than the min(sal) of deptno 20.

Q5) List empno, ename, job of the employees whose job is not a 'CLERK' and whose salary is less than at least one of the salaries of the employees whose job is 'CLERK'.

Q6) List empno, ename, job of the employees whose salary is greater than the average salary of each department.

S-14-15 SLO-1 & SLO-2 :
Lab 6: Nested Queries on sample exercise

Q7) Display the name, dept. no, salary, and commission of any employee whose salary and commission matches both the commission and salary of any employee in department 30.

Q8) List ename sal, deptno, average salary of the dept where he/she works, if salary of the employee is greater than his/her department average salary.

Q9) List ename, job, sal of the employees whose salary is equal to any one of the salary of the employee 'SCOTT' and 'WARD'.

Q10) List ename, job, sal of the employees whose salary and job is equal to the employee 'FORD'.

Q11) List ename, job, deptno, sal of the employees whose job is same as 'JONES' and salary is greater than the employee 'FORD'.

Q12) List ename, job of the employees who work in deptno 10 and his/her job is any one of the job in the department 'SALES'.

THANK YOU