Topics Covered in Todays Class

Unit 1:
- Characteristics of the Database Approach
- Advantages of Using the Database Approach
- When not to use a DBMS

Main Characteristics of the Database Approach

1. **Self-describing nature** of a database system.
2. **Insulation** between programs and data, and data manipulation.
3. Support of **multiple views** of the data.
4. **Sharing** of data and **multi-user** transaction processing

Main Characteristics of the Database Approach

1. Self-describing nature of a database system

What will be your description of the data stored in the following tables ?

1BM14CS001	Aditya	3	WP
1BM14CS002	Bharath	3	DS

1BM14CS001	1000
1BM14CS002	1000

1BM14CS001	S
1BM14CS002	B

Main Characteristics of the Database Approach

1. Self-describing nature of a database system.

Student_Details

USN	Name	Sem	Sub
1BM14CS001	Aditya	3	WP
1BM14CS002	Bharath	3	DS

Student_ExamFee_Details

USN	Amount
1BM14CS001	1000
1BM14CS002	1000

Student_Grade_Details

USN	Grade
1BM14CS001	S
1BM14CS002	B

Main Characteristics of the Database Approach

1. Self-describing nature of a database system.

Student_Details

USN	Name	Sem	Sub
1BM14CS001	Aditya	3	WP
1BM14CS002	Bharath	3	DS

Student_ExamFee_Details

USN	Amount
1BM14CS001	1000
1BM14CS002	1000

Student_Grade_Details

USN	Grade
1BM14CS001	S
1BM14CS002	B

Column_name	Data_Type	Belongs_to
USN	Char(10)	Student_Details
Name	Char(30)	Student_Details
Sem	Integer	Student_Details
Sub	Char(2)	Student_Details
Amount	Float	Student_ExamFee_Details
Grade	Char(1)	Student_Grade_Details

Catalog
Information stored in catalog is called **meta-data** which Describes the structure Of primary database

Main Characteristics of the Database Approach

2. Insulation between programs and data, and data manipulation.

File Approach

student.txt

```
1BM14CS001 Aditya LA,Java,DBMS,OS,DC
1BM14CS002 Baharth DBMS,OS,DC
```

```
fp=fopen("student.txt","r");
while(fscanf(fp,"%s %s %s",USN,name,subjects)!=EOF)
{
printf("USN: %s Name: %s Subjects: %s",USN,name,subjects);
}
```

What will be the Output of the above program statements ?

Main Characteristics of the Database Approach

2. Insulation between programs and data, and data manipulation.

File Approach

student.txt

```
Aditya 1BM14CS001  LA,Java,DBMS,OS,DC
Baharth 1BM14CS002  DBMS,OS,DC
```

```
fp=fopen("student.txt","r");
while(fscanf(fp,"%s %s %s",USN,name,subjects)!=EOF)
{
printf("USN: %s Name: %s Subjects: %s",USN,name,subjects);
}
```

What will be the Output of the above program statements ?

Main Characteristics of the Database Approach

2. Insulation between programs and data, and data manipulation.

DBMS Approach

student

USN	Name	Subjects
1BM14CS001	Aditya	LA, Java, DBMS,OS,DC
1BM14CS002	Bharath	DBMS, OS, DC

```
select USN, Name, Subjects from student;
```

SQL **query** to retrieve and display table information

```
1BM14CS001 Aditya LA,Java,DBMS,OS,DC
1BM14CS002 Bharath DBMS,OS,DC
```

Main Characteristics of the Database Approach

2. Insulation between programs and data, and data manipulation.

DBMS Approach

student

Name	USN	Subjects
Aditya	1BM14CS001	LA, Java, DBMS,OS,DC
Bharath	1BM14CS002	DBMS, OS, DC

```
select USN, Name, Subjects from student;
```

```
1BM14CS001 Aditya LA,Java,DBMS,OS,DC
1BM14CS002 Bharath DBMS,OS,DC
```

SQL **query** to retrieve and display table informat

Main Characteristics of the Database Approach

3. Support of multiple views of the data

Student database

Student_Details

USN	Name	Sem	Sub
1BM14CS001	Aditya	3	WP
1BM14CS002	Bharath	3	DS

Student_ExamFee_Details

USN	Amount
1BM14CS001	1000
1BM14CS002	1000

Student_Grade_Details

USN	Grade
1BM14CS001	S
1BM14CS002	B

Accounts Section

1BM14CS001 Aditya 3 WP 1000
1BM14CS002 Bahart 3 DS 1000

Examination Section

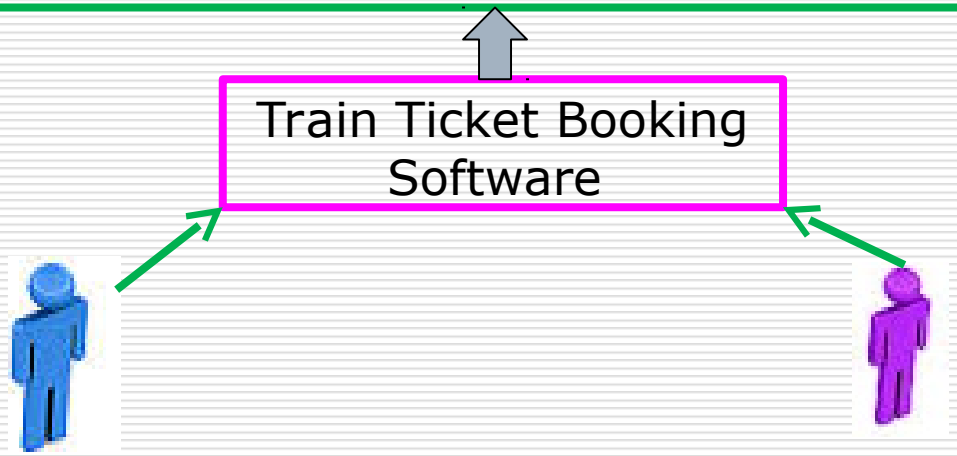
1BM14CS001 Aditya 3 WP S
1BM14CS002 Bahart 3 DS B

Main Characteristics of the Database Approach

4. Sharing of data and **multi-user** transaction processing

Train Reservation Database

Source	Destination	Train Name	Start Time	Start Date	Number of Seats Availability
Bangalore	Mysore	Chamundi Express	18:15	14-1-2016	2



Main Characteristics of the Database Approach

- **Self-describing nature of a database system:**
  - A DBMS **catalog** stores the description of a particular database (e.g. data structures, types, and constraints)
  - The description is called **meta-data**.
  - This allows the DBMS software to work with different database applications.
- **Insulation between programs and data, and data manipulation:**
  - Called **program-data independence**.
  - Allows changing data structures and storage organization without having to change the DBMS access programs.
  - A **data model** is used to hide storage details and present the users with a conceptual view of the database.
  - Programs refer to the data model constructs rather than data storage details

Main Characteristics of the Database Approach

- **Support of multiple views of the data:**
  - Each user may see a different view of the database, which describes **only** the data of interest to that user.
- **Sharing of data and multi-user transaction processing:**
  - Allowing a set of **concurrent users** to retrieve from and to update the database.
  - *Concurrency control* within the DBMS guarantees that each **transaction** is correctly executed or aborted
  - *Recovery* subsystem ensures each completed transaction has its effect permanently recorded in the database
  - **OLTP** (Online Transaction Processing) is a major part of database applications. This allows hundreds of concurrent transactions to execute per second.

Advantages of Using the Database Approach

1. **Controlling redundancy** in data storage and in development and maintenance efforts.
2. Restricting **unauthorized** access to data.
3. Providing **persistent storage** for program Objects (data structures provided by DBMS & the programming languages were incompatible)
4. Providing Storage Structures (e.g. indexes) for **efficient Query Processing**
5. Providing **backup** and **recovery** services.
6. Providing **multiple** interfaces to **different classes of users**.
7. Representing **complex relationships** among data.
8. Enforcing **integrity constraints** on the database.
9. Drawing **inferences and actions** from the stored data using deductive and active rules

Understanding **integrity constraints** on the database

Database **Catalog** for following database tables

Column_name	Data_Type	Belongs_to
USN	Char(10)	Student_Details
Name	Char(30)	Student_Details
Sem	Integer	Student_Details
Sub	Char(2)	Student_Details
Amount	Float	Student_ExamFee_Details
Grade	Char(1)	Student_Grade_Details

Which of the following tables data storage is correct as per the above catalog definition ?

Student_Grade_Details

USN	Grade
1BM14CS001	S
1BM14CS002	B

Student_Grade_Details

USN	Grade
1BM14CS001	S
1BM14CS002	8

Understanding **integrity constraints** on the database

Student_Details

USN	Name	Sem	Sub
1BM14CS001	Aditya	3	WP
1BM14CS002	Bharath	3	DS

What is wrong in the following table data as per the **Student_Details** table ?

Student_Grade_Details

USN	Grade
1BM14CS001	S
1BM14CS002	B
1BM14CS003	C

Understanding drawing of "**inferences and actions**" from the stored data

Student_Grade_Details

USN	Grade
1BM14CS001	S
1BM14CS002	B
1BM14CS003	C
1BM14CS004	S
1BM14CS005	F
1BM14CS006	S

Using the above table data, Can you **infer**
How many Students have scored **S grade** and What are their USN's?

Understanding drawing of "**inferences and actions**" from the stored data

Student_Grade_Details

USN	Grade
1BM14CS001	S
1BM14CS002	B
1BM14CS003	C
1BM14CS004	S
1BM14CS005	F
1BM14CS006	S

Catalog

Column_name	Data_Type	Constraint
USN	Char(10)	
Grade	Char(1)	Should not be empty

Using the above Catalog information, Can you tell Whether the following insert action on Student_Grade_Details table is **Right or Wrong** ?

insert into **Student_Grade_Details** values (1BM14CS007);

When not to use a DBMS ?

When not to use a DBMS

- Main inhibitors (**costs**) of using a DBMS:
  - High initial investment and possible need for additional hardware.
  - Overhead for providing generality, security, concurrency control, recovery, and integrity functions.
- When a DBMS may be unnecessary:
  - If the database and applications are **simple**, well defined, and **not expected to change**.
  - If access to data by **multiple users is not required.**

Data Definition language and Interfaces

- DBMS Languages
  - Data Definition Language (DDL)
  - Storage Definition Language (SDL)
  - View Definition language (VDL)
  - Data manipulation Language (DML)
- Data Interfaces
  - Menu-based
  - Form-Based
  - Graphical User Interface
  - Natural language Interface
  - Speech Input and Output
  - Interfaces for Parametric users
  - Interfaces for DBA

Topics Covered in Todays Class

Unit 1: Database System Concepts and Architecture
- Data Models, Schemas and instances
- Three Schema Architecture and Data Independence
- Database language and interfaces
- The Database System Environment

Objective of todays class

Understanding the basic terminologies and definitions involved in building Architecture of Database Systems.

In this regard, First we will understand
Data Models, Schemas and Instances

Data Models

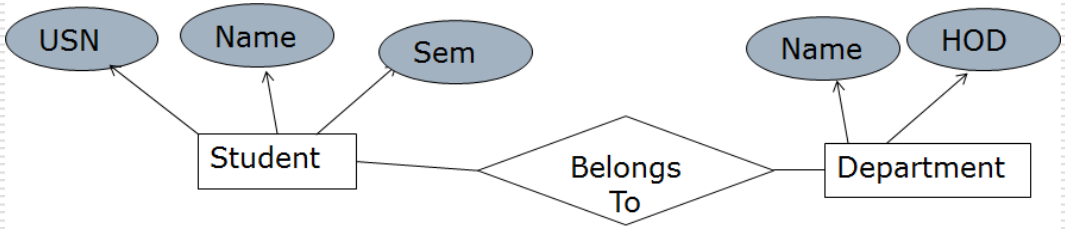
- **Data Abstraction** generally refers to the suppression of details of data organization and storage and the highlighting of the essential features for improved understanding of data.
- **Data Model:**
  - A set of concepts to describe the ***structure*** of a database, the ***operations*** for manipulating these structures, and certain ***constraints*** that the database should obey.
  - Data Models provides the necessary means to achieve data abstraction.

Categories of Data Models

- **Conceptual (high-level, semantic) data models:**
  - Provide concepts that are close to the way many users perceive data.
    - (Also called ***entity-based*** or ***object-based*** data models.)
- **Implementation (representational) data models:**
  - Provide concepts that fall in between high and low level, used by many commercial DBMS implementations (e.g. relational data models used in many commercial systems).
- **Physical (low-level, internal) data models:**
  - Provide concepts that describe details of how data is stored in the computer. These are usually specified in an ad-hoc manner through DBMS design and administration manuals

Categories of Data Models

Conceptual (high-level)



Implementation (Representational)

USN	Name	Sem	Dep
1BM14CS001	Aditya	3	CSE
1BM14IS002	Bharath	3	ISE

Dep	HOD
CSE	Dr. H S Guruprasad
ISE	Dr. Gowrishankar

Physical (low-level)

Schemas, Instance and Database State

- Database Schema:
  - The ***description*** of a database.
  - Includes descriptions of the database structure, data types, and the constraints on the database.
  - Schema Diagram:
  - An ***illustrative*** display of (most aspects of) a database schema
- Database State:
  - The actual data stored in a database at a ***particular moment in time***. This includes the collection of all the data in the database.
  - Also called **database instance** (or occurrence or snapshot).
    - The term *instance* is also applied to individual database components, e.g. *record instance, table instance, entity instance*

Schemas, Instance and Database State

Schema Diagram

Student

USN	Name	Sem	Dep

Department

Dep	HOD

Database State at time "X"

USN	Name	Sem	Dep
1BM14CS001	Aditya	3	CSE
1BM14IS002	Bharath	3	ISE

Database Instance

Database State at time "Y"

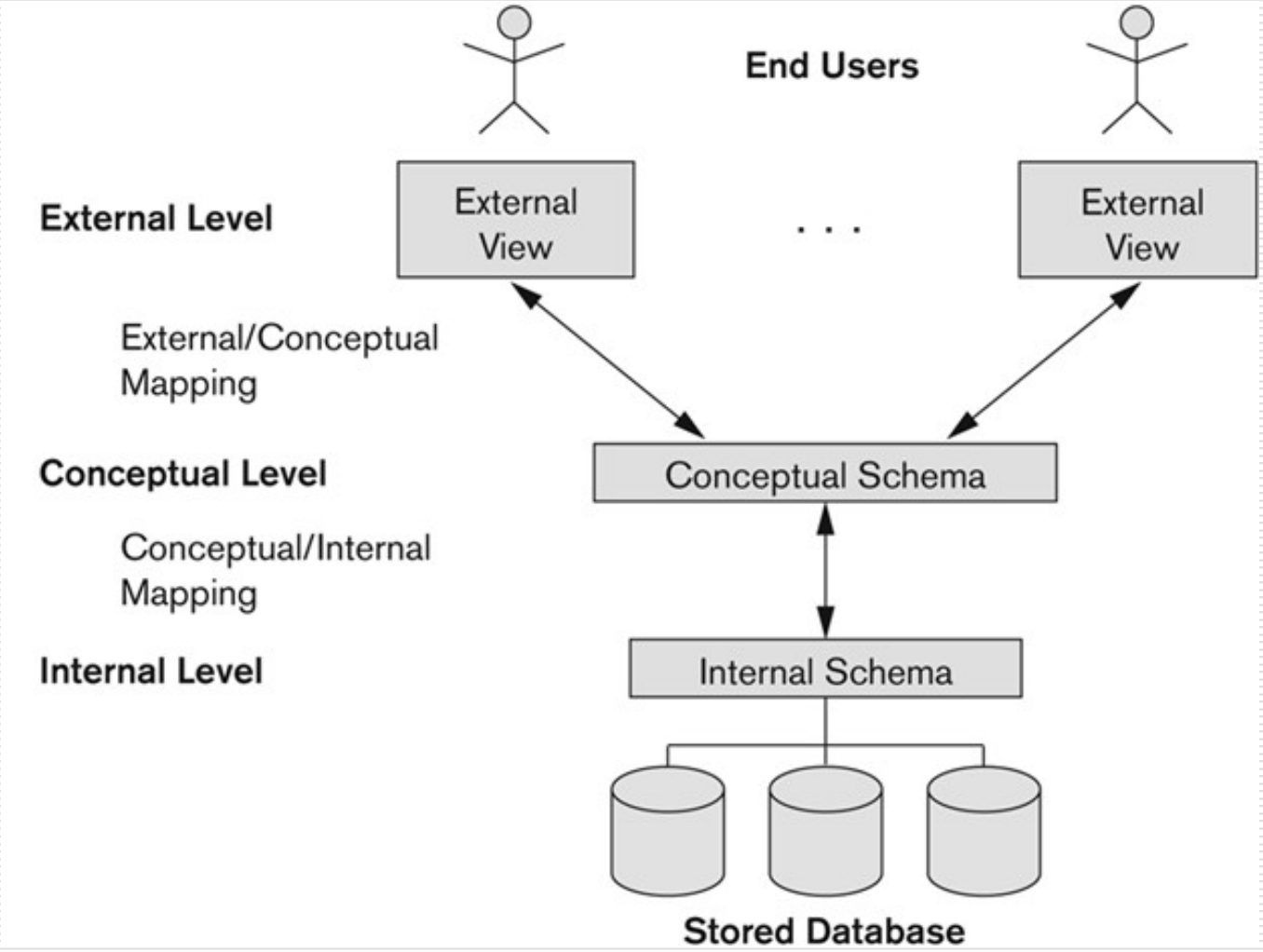
USN	Name	Sem	Dep
1BM14CS001	Aditya	3	CSE
1BM14IS002	Bharath	3	ISE
1BM14CS002	Chandan	3	CSE

Database Instance

Three-Schema Architecture

- Defines DBMS schemas at ***three*** levels:
  - **Internal schema** at the internal level to describe physical storage structures and access paths (e.g indexes).
    - Typically uses a **physical** data model.
  - **Conceptual schema** at the conceptual level to describe the structure and constraints for the whole database for a community of users.
    - Uses a **conceptual** or an **implementation** data model.
  - **External schemas** at the external level to describe the various user views.
    - Usually uses the same data model as the conceptual schema.

The three-schema architecture



Data Independence

- **Logical Data Independence:**
  - The capacity to change the conceptual schema without having to change the external schemas and their associated application programs.
- **Physical Data Independence:**
  - The capacity to change the internal schema without having to change the conceptual schema.

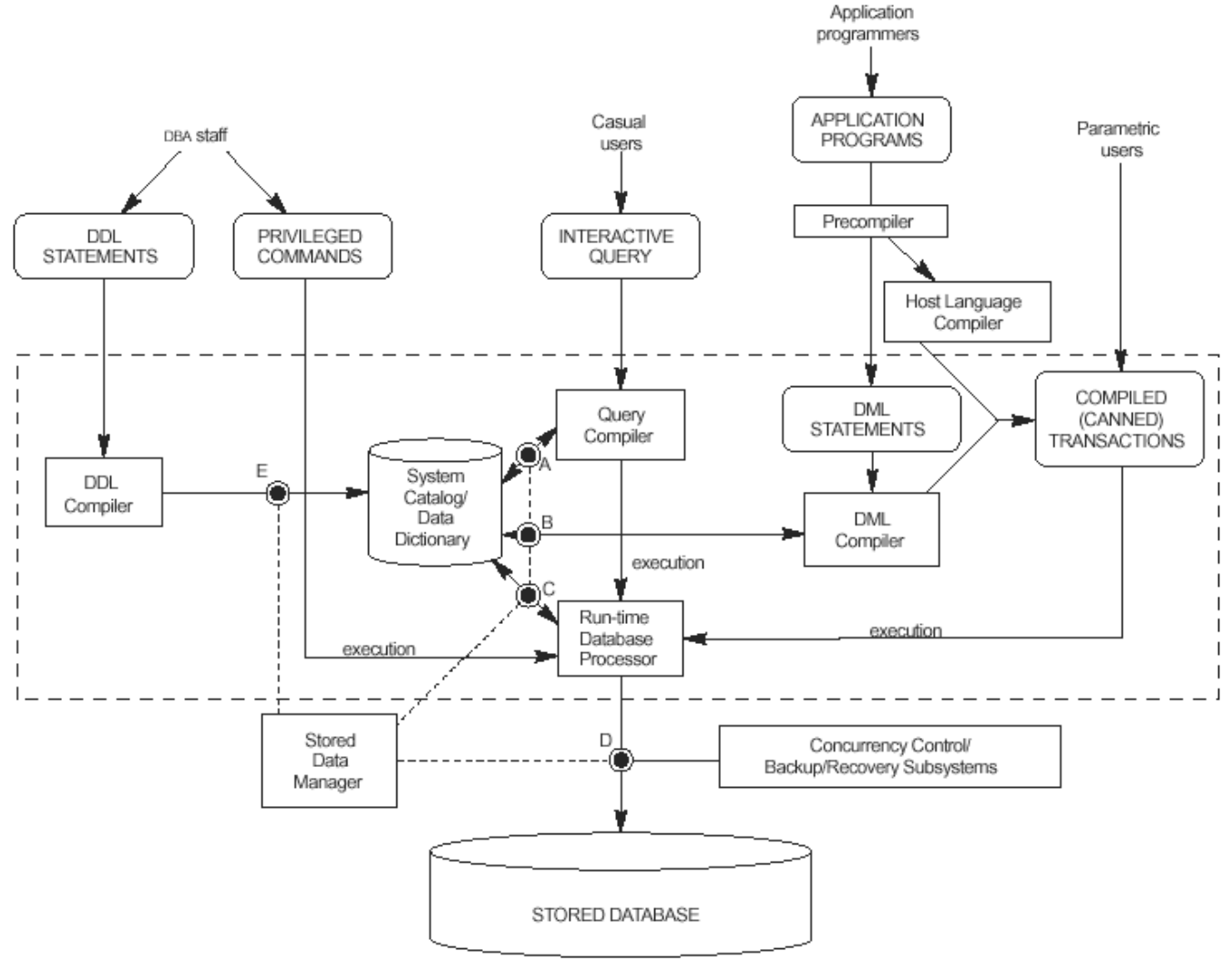
Review on Your Own

- Database language and interfaces
- The Database System Environment

Data Definition language and Interfaces

- DBMS Languages
  - Data Definition Language (DDL)
  - Storage Definition Language (SDL)
  - View Definition language (VDL)
  - Data manipulation Language (DML)
- Data Interfaces
  - Menu-based
  - Form-Based
  - Graphical User Interface
  - Natural language Interface
  - Speech Input and Output
  - Interfaces for Parametric users
  - Interfaces for DBA

The Database System Environment



Activity - Questionnaire

1. Mention three different levels in Three-Schema Architecture

Thanks for Listening

Summary

Student database

Student_Details

USN	Name	Sem	Sub
1BM14CS001	Aditya	3	WP
1BM14CS002	Bharath	3	DS

Student_ExamFee_Details

USN	Amount
1BM14CS001	1000
1BM14CS002	1000

Student_Grade_Details

USN	Grade
1BM14CS001	S
1BM14CS002	B

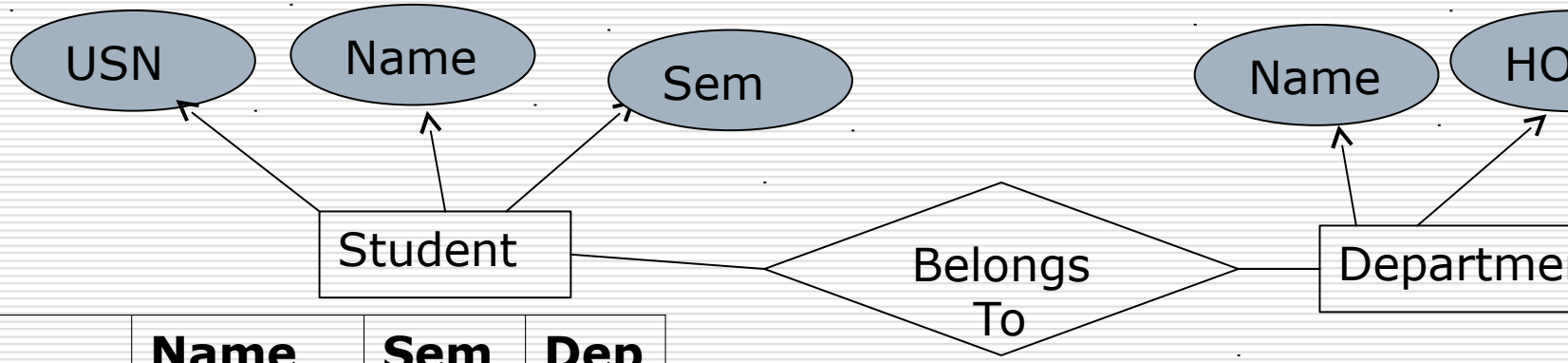
Accounts Section

Examination Section

1BM14CS001 Aditya 3 WP 1000
1BM14CS002 Bahart 3 DS 1000

1BM14CS001 Aditya 3 WP S
1BM14CS002 Bahart 3 DS B

Categories of Data Models



USN	Name	Sem	Dep
1BM14CS001	Aditya	3	CSE
1BM14IS002	Bharath	3	ISE

Dep	HOD
CSE	Dr. H S Guruprasad
ISE	Dr. Gowrishankar

USN	Name	Sem	Dep
1BM14CS001	Aditya	3	CSE
1BM14IS002	Bharath	3	ISE
1BM14CS002	Chandan	3	CSE

Topics Covered in Todays Class

Unit 2: Data Modeling using the Entity-Relationship(ER) diagram

- Entity Types, Entity Sets, Attributes and Keys

Sample Database Application

- **Example:** HOD of CSE department calls you and asks to develop the following application

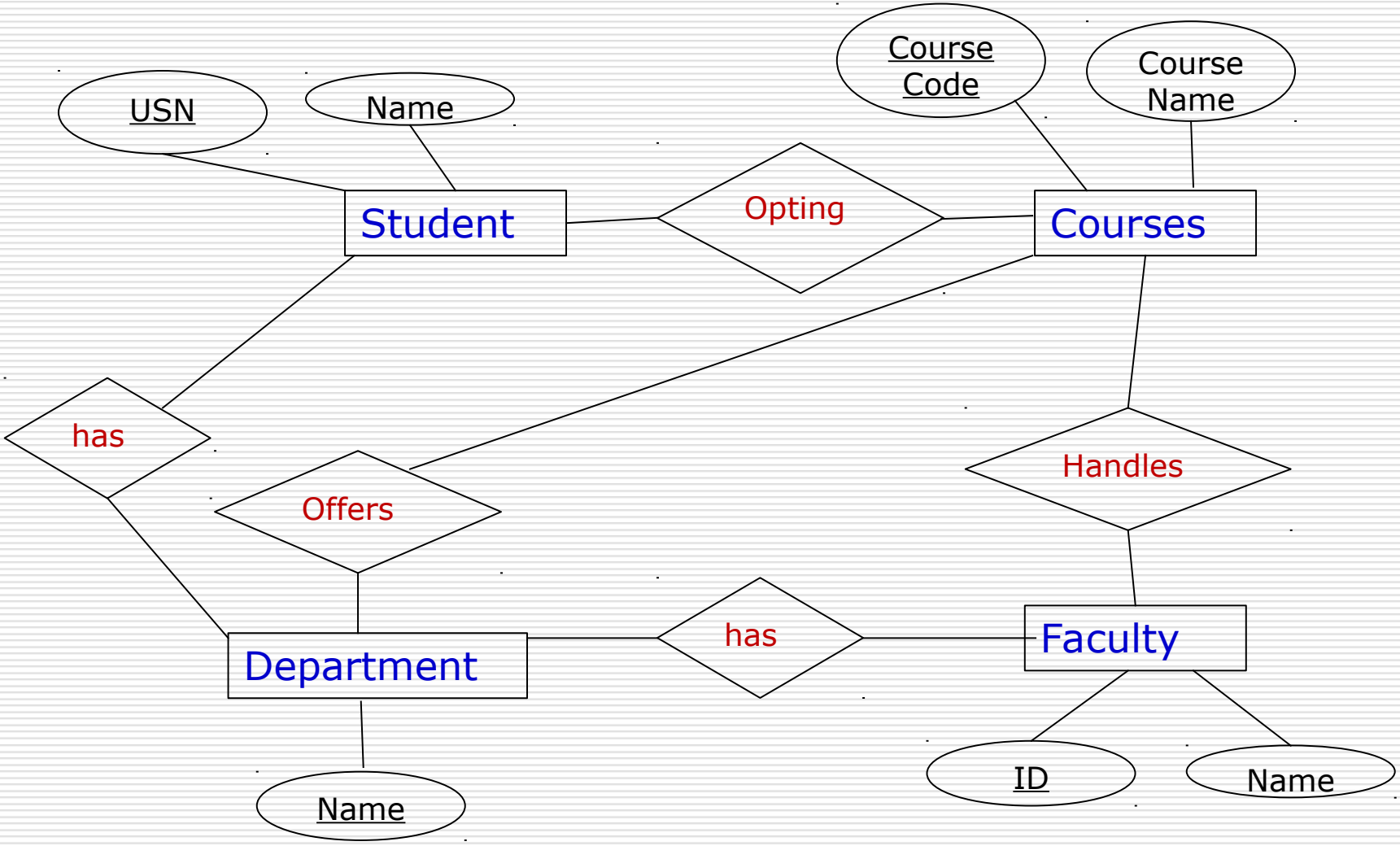
"Develop a database application to automate the process of course registration"

- Your **first task** should be to have a discussion with your client i.e HOD, to identify the requirements of the application.
  - Requirements are as follows
    - Department has students and faculty
    - Department will offer a set of courses during each semester
    - Each student in the department during course registration will be opting for courses offered by the department
    - Faculty will be handling courses
- Your **second task** should be to prepare a ER diagram which shows the Design plan of the database application to be developed.

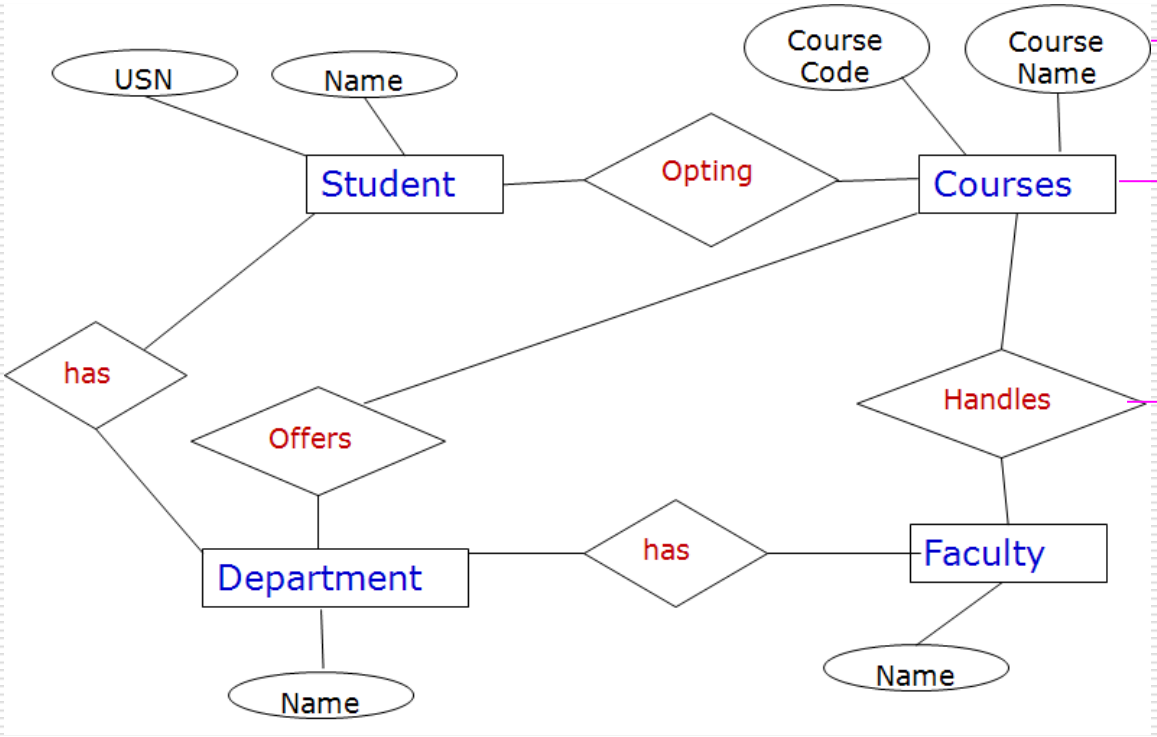
Example: House design plan

During plan preparation
Various notations and
Terminologies

Design plan or ER diagram before actual development of the application



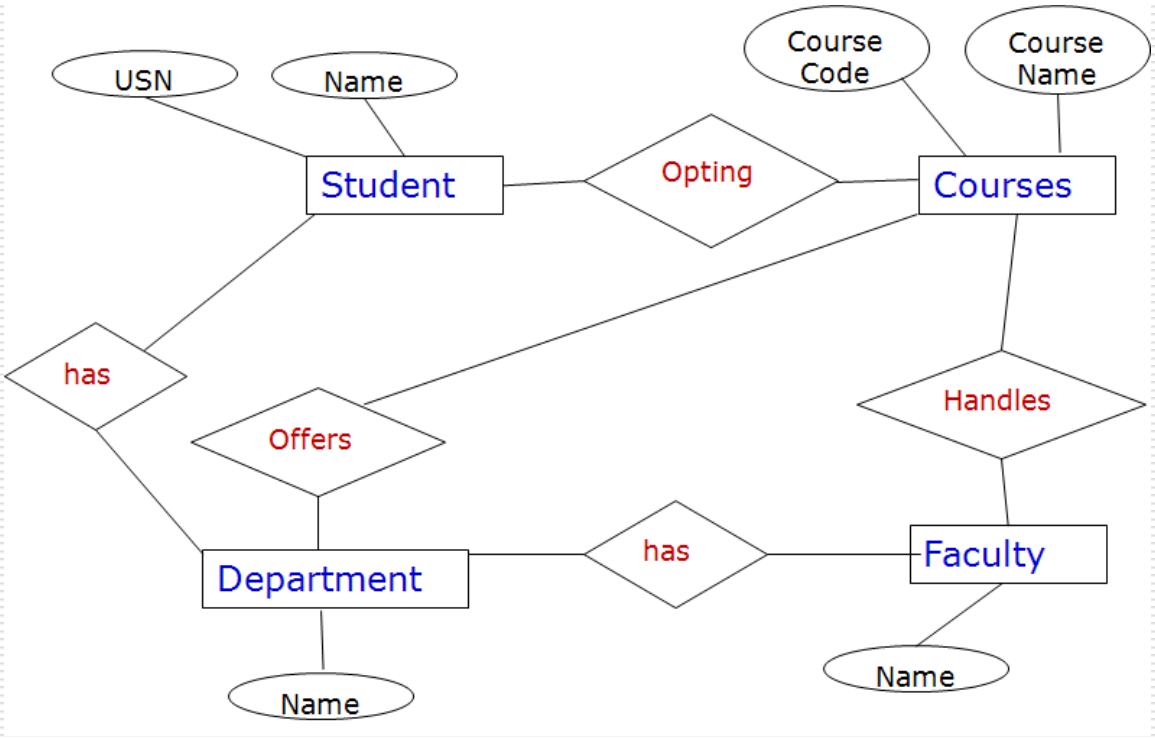
Design plan or ER diagram before actual development of the application



Requirements Were

- Department has students and faculty
- Department will offer a set of courses during each semester
- Each student in the department during course registration will be opting for courses offered by the department
- Faculty will be handling courses

Design plan or ER diagram before actual development of the application



Constraints

-Each semester student should register for minimum of 20 credits and maximum of 30 credits

-Each faculty can handle a maximum two courses During each semester

Requirements Were

- Department has students and faculty
- Department will offer a set of courses during each semester
- Each student in the department during course registration will be opting for courses offered by the department
- Faculty will be handling courses

What is Entity, Entity Types, Entity sets ?

- **Entity** is object in real world which has independent existence.
  - Example: Student, Course, Faculty, Car, House, College, Book, Food

ER Diagram Notation for Entity: Rectangle

Activity - Questionnaire

List out the Entities which you will come across in real world.

What is Entity Types, Entity sets ?

- Entity type is collection of entities with common attributes
- Entity set collection of one or more attributes

Student <– – – Entity Type

USN	Name	Email ID	Mobile No.	DOB
1BM14CS001	Aditya	aditya@bmsce.ac.in	9448444160	1-1-1997
1BM14CS002	Bharath	bharath@bmsce.ac.in	8762244699	31-12-1996

Entity Set

What is Attribute ?

- Attribute is a property that describes Entity.

Attributes:
USN, Name, Email ID, Mobile Number, DOB

Student <– – – Entity Type

USN	Name	Email ID	Mobile No.	DOB
1BM14CS001	Aditya	aditya@bmsce.ac.in	9448444160	1-1-1997
1BM14CS002	Bharath	bharath@bmsce.ac.in	8762244699	31-12-1996

Entity Set

What is **Domain** ?
Set of permitted values for an Attribute.

What is Attribute ?

- Attribute is a property that describes Entity.

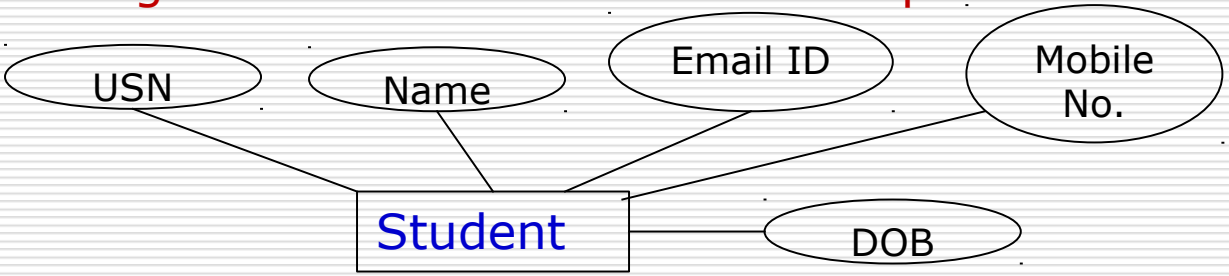
Attributes:
USN, Name, Email ID, Mobile Number, DOB

Student <--- Entity Type

USN	Name	Email ID	Mobile No.	DOB
1BM14CS001	Aditya	aditya@bmsce.ac.in	9448444160	1-1-1997
1BM14CS002	Bharath	bharath@bmsce.ac.in	8762244699	31-12-1996

Entity Set

ER Diagram Notation for Attribute: Ellipse



Activity - Questionnaire

List attributes for

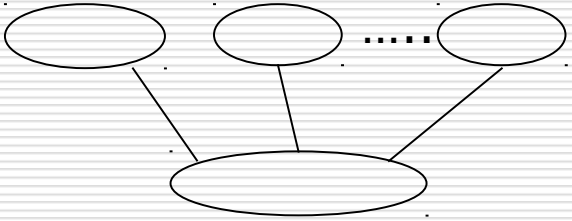
1. Car
2. Book

Different categories of attributes

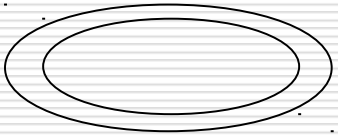
- Simple (Atomic) vs Composite
  - Attributes that are not divisible are called **Simple**.
  - Attributes that can be divided into smaller parts are called **Composite**.
  - Ex.: Simple- USN, Composite- Name(First Name, Last Name), Address(Street name, Area Name, Place)
- Single value vs Multiple valued
  - Ex.: Single – USN, Multiple – Mobile numbers
- Stored vs Derived
  - Ex.: Stored – DOB, Derived – Age
- Complex Attributes
  - Ex.: {Address(Street name, Area Name, Place)} → Office address and Residence Address
- Null values
  - Ex. : Middle Name is Optional i.e. Gautham Sharma or Gautham K Sharma

Attribute: ER diagram notations

Attribute



Composite Attribute



Multivalued Attribute

Derived Attribute

What is Key ?

- Key attribute is a attribute or a combination of attributes which will uniquely identify remaining attributes of entity.
- What are the Key attributes in the following student table ?

USN	Name	Email ID	Mobile No.	DOB
1BM14CS001	Aditya	aditya@bmsce.ac.in	9448444160	1-1-1997
1BM14CS002	Bharath	bharath@bmsce.ac.in	8762244699	31-12-1996

What is Key ?

- Key attribute is a attribute or a combination of attributes which will uniquely identify remaining attributes of entity.
- What are the Key attributes in the following student table ?

USN	Name	Email ID	Mobile No.	DOB
1BM14CS001	Aditya	aditya@bmsce.ac.in	9448444160	1-1-1997
1BM14CS002	Bharath	bharath@bmsce.ac.in	8762244699	31-12-1996

ER Diagram Notation for Entity: Rectangle

What is Key ?

- Key attribute is a attribute or a combination of attributes which will uniquely identify remaining attributes of entity.
- What are the Key attributes in the following student table ?

USN	Name	Email ID	Mobile No.	DOB
1BM14CS001	Aditya	aditya@bmsce.ac.in	9448444160	1-1-1997
1BM14CS002	Bharath	bharath@bmsce.ac.in	876244699	31-12-1996
1BM14CS003	Dinesh K	dineshk@bmsce.ac.in	8444160944	6-6-1998
1BM14CS004	Dinesh K	Dineshk_4@bmsce.ac.in	7446998762	6-6-1998

What is Key ?

- Key attribute is a attribute or a combination of attributes which will uniquely identify remaining attributes of entity.
- What are the Key attributes in the following student table ?

Faculty	Department	Salary
A	CSE	20K
B	CSE	21K
A	EC	30K
B	EC	27K
C	CSE	22K

What is Key ?

- Key attribute is a attribute or a combination of attributes which will uniquely identify remaining attributes of entity.
- What are the Key attributes in the following student table ?

Faculty	Department	Salary
A	CSE	20K
B	CSE	21K
A	EC	30K
B	EC	27K
C	CSE	22K

What is Key ?

- Key attribute is a attribute or a combination of attributes which will uniquely identify remaining attributes of entity.
- What are the Key attributes in the following student table ?

Faculty	Department	Salary
A	CSE	20K
B	CSE	21K
A	EC	30K
B	EC	27K
C	CSE	22K

What is the Salary of Employee whose name is **A** ?

What is Key ?

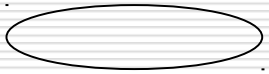
- Key attribute is a attribute or a combination of attributes which will uniquely identify remaining attributes of entity.
- What are the Key attributes in the following student table ?

Faculty	Department	Salary
A	CSE	20K
B	CSE	21K
A	EC	30K
B	EC	27K
C	CSE	22K

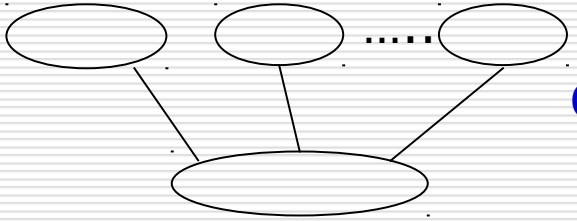
What is the Salary of Employee whose name is **A** ?

Note:
Before determining the Key attribute, look at the values present but it will not be consistent

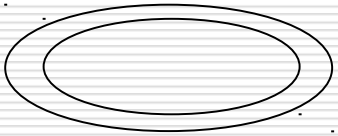
Activity To DO



Attribute

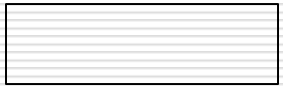


Composite Attribute

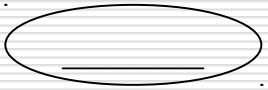


Multivalued Attribute

Derived Attribute



Entity



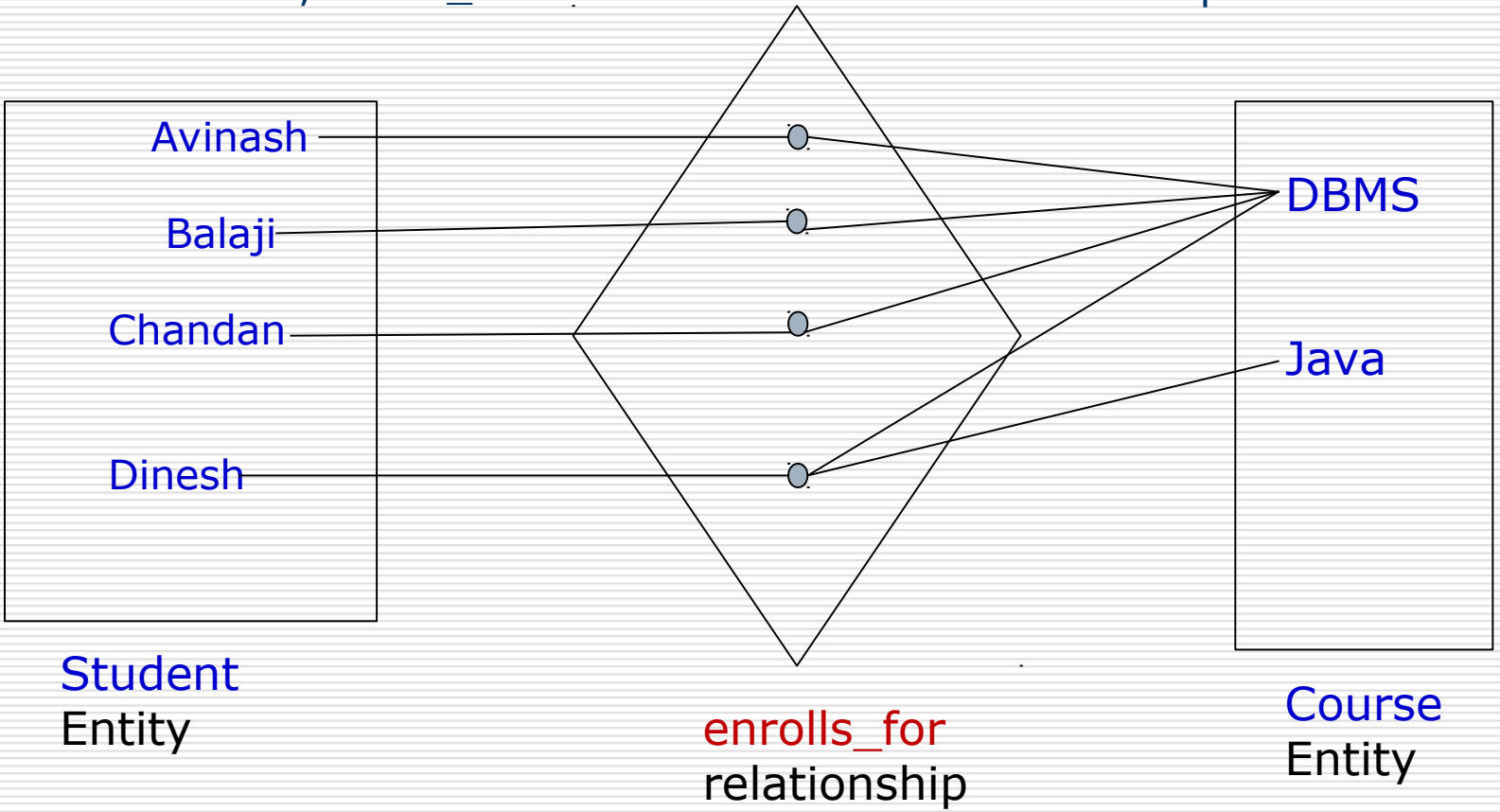
Key Attribute

Represent Employee entity using ER diagram notation Which has following attributes

-Employee ID (Key)
-Name
-Address(House No., Street name, Area name, Place)
-Mobile Number (Can have more than one)
-DOB
-Age

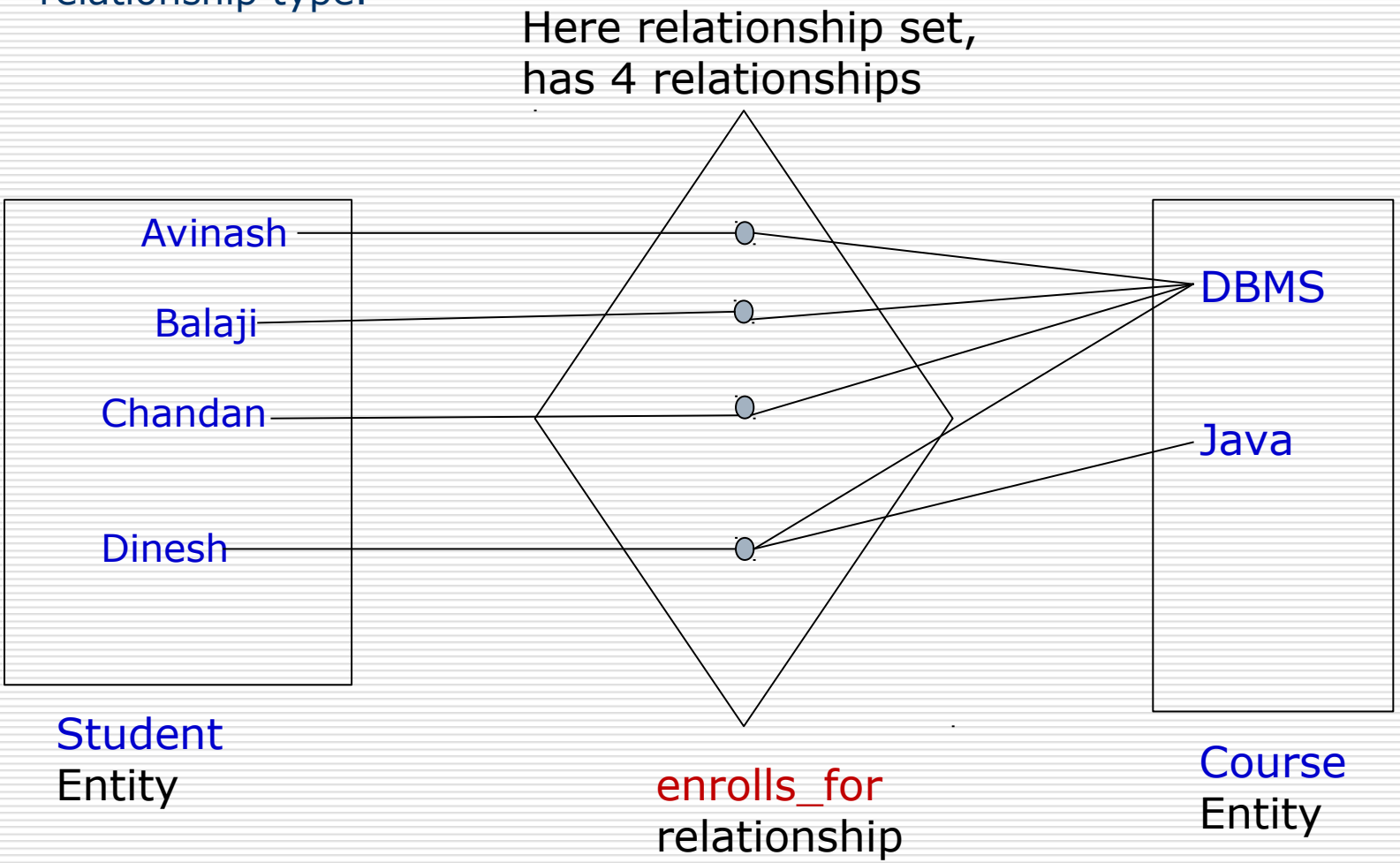
Relationship Type

- A Relationship Type defines a relationship set among entities of certain entity types.
- Example, an Faculty works_for a department, a student enrolls_for in a course. Here, works_for and enrolls are called relationships.



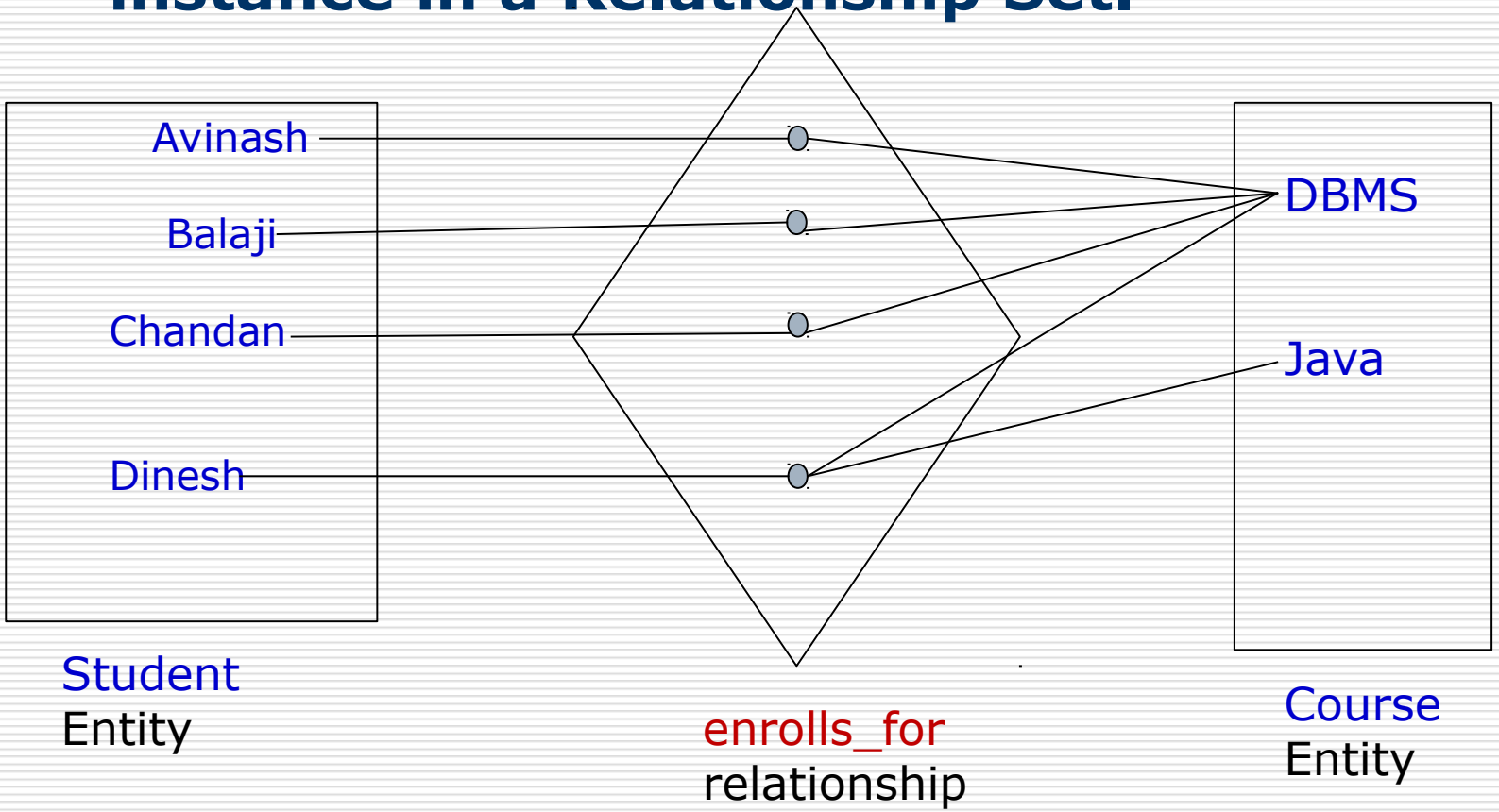
Relationship Set

- An Relationship Set is a collection of relationships all belonging to one relationship type.



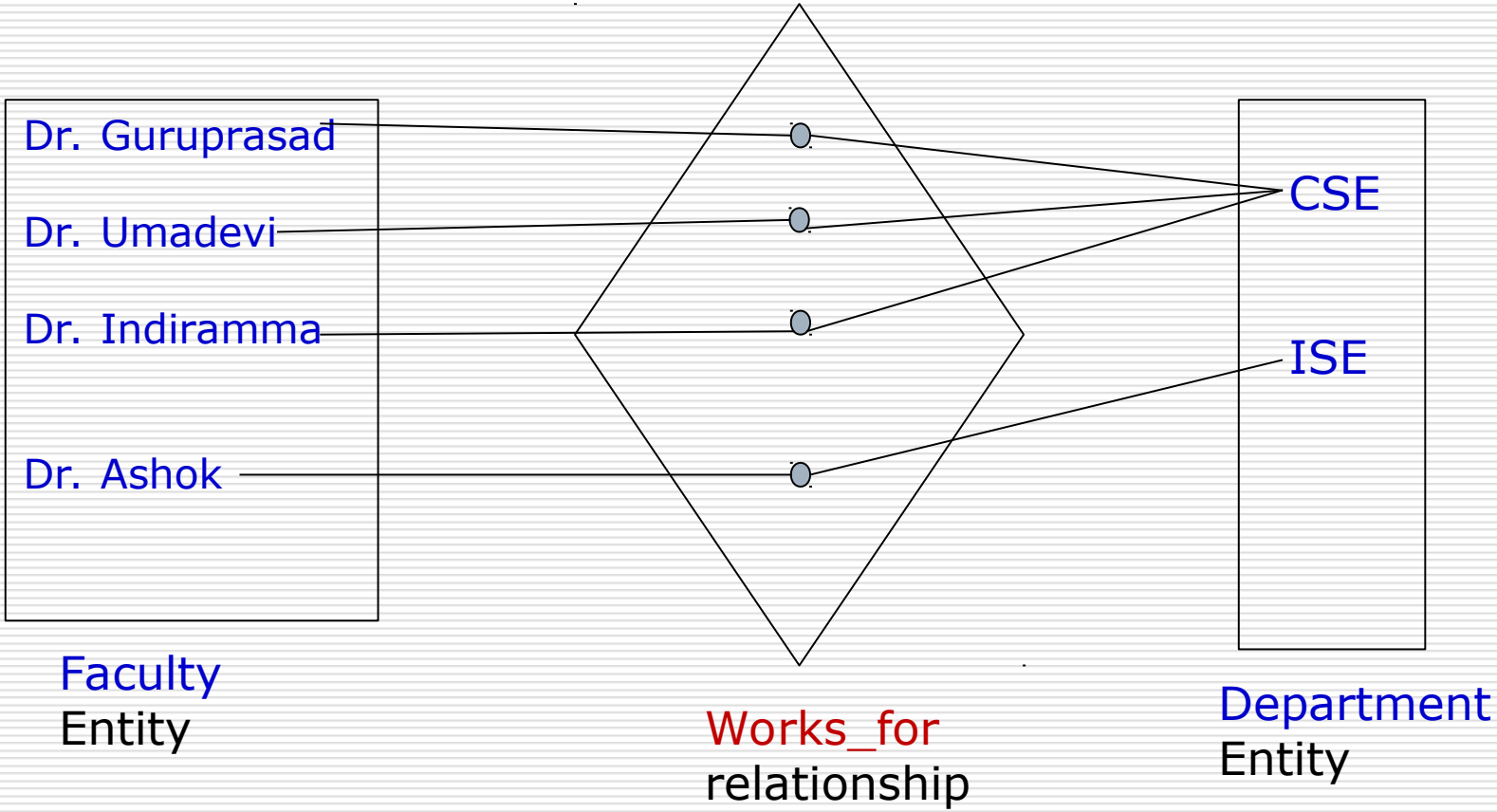
Relationship

- The association among entities is called a relationship. or **A Relationship is one instance in a Relationship Set.**



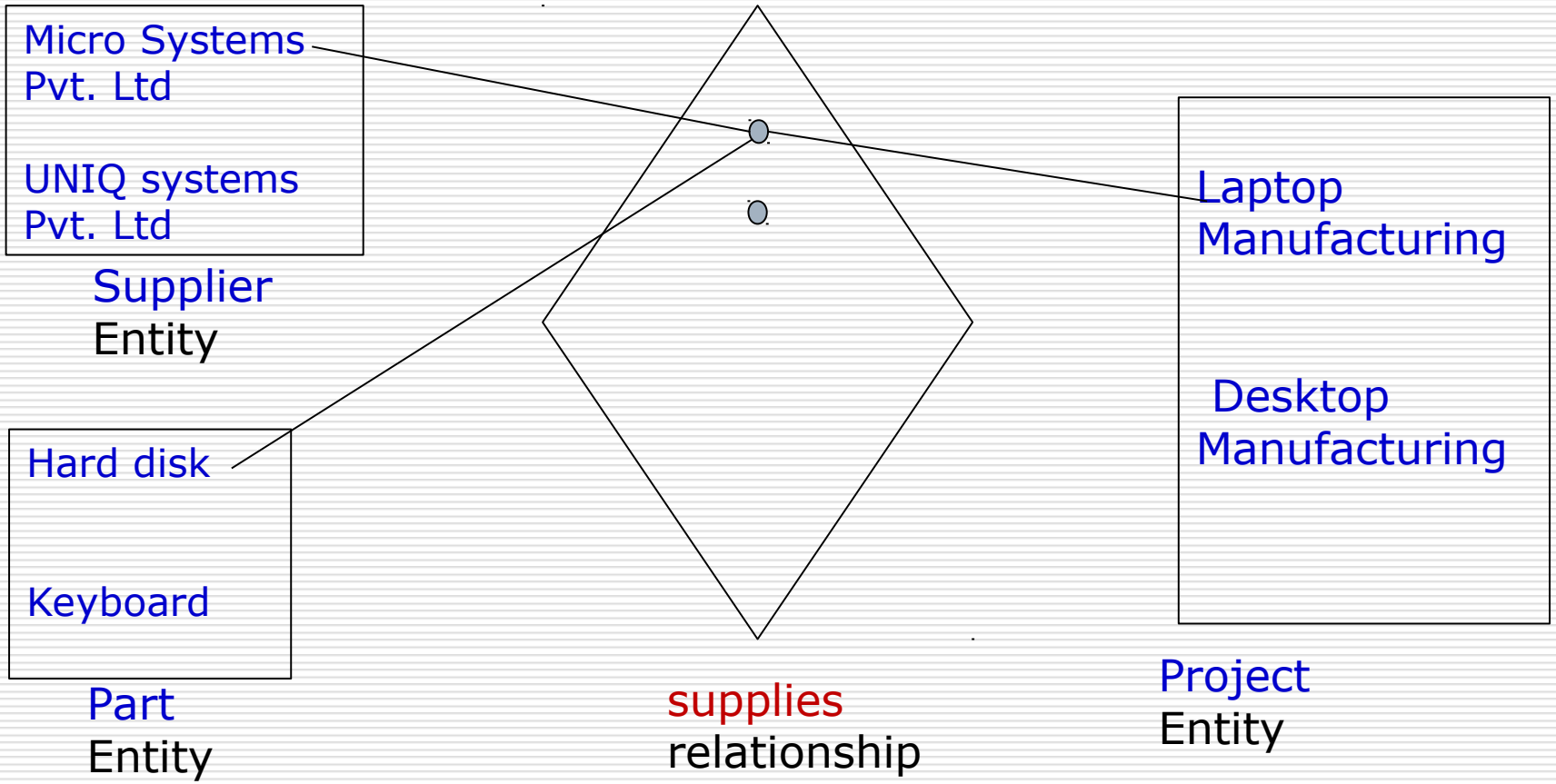
Relationship Degree

- Binary Relationship: Degree two, two entities are participating



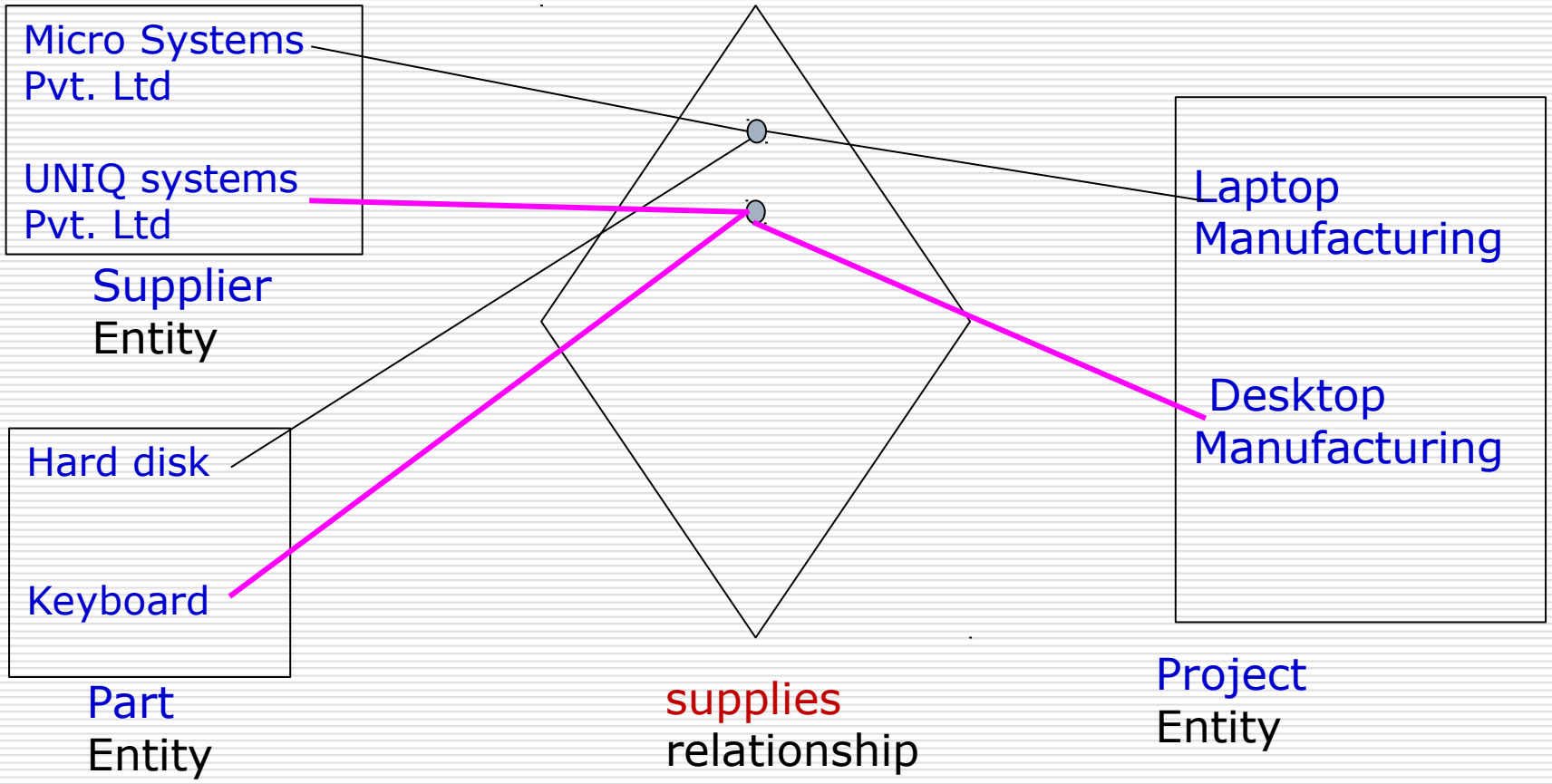
Relationship Degree

- Ternary Relationship: Degree three, three entities are participating



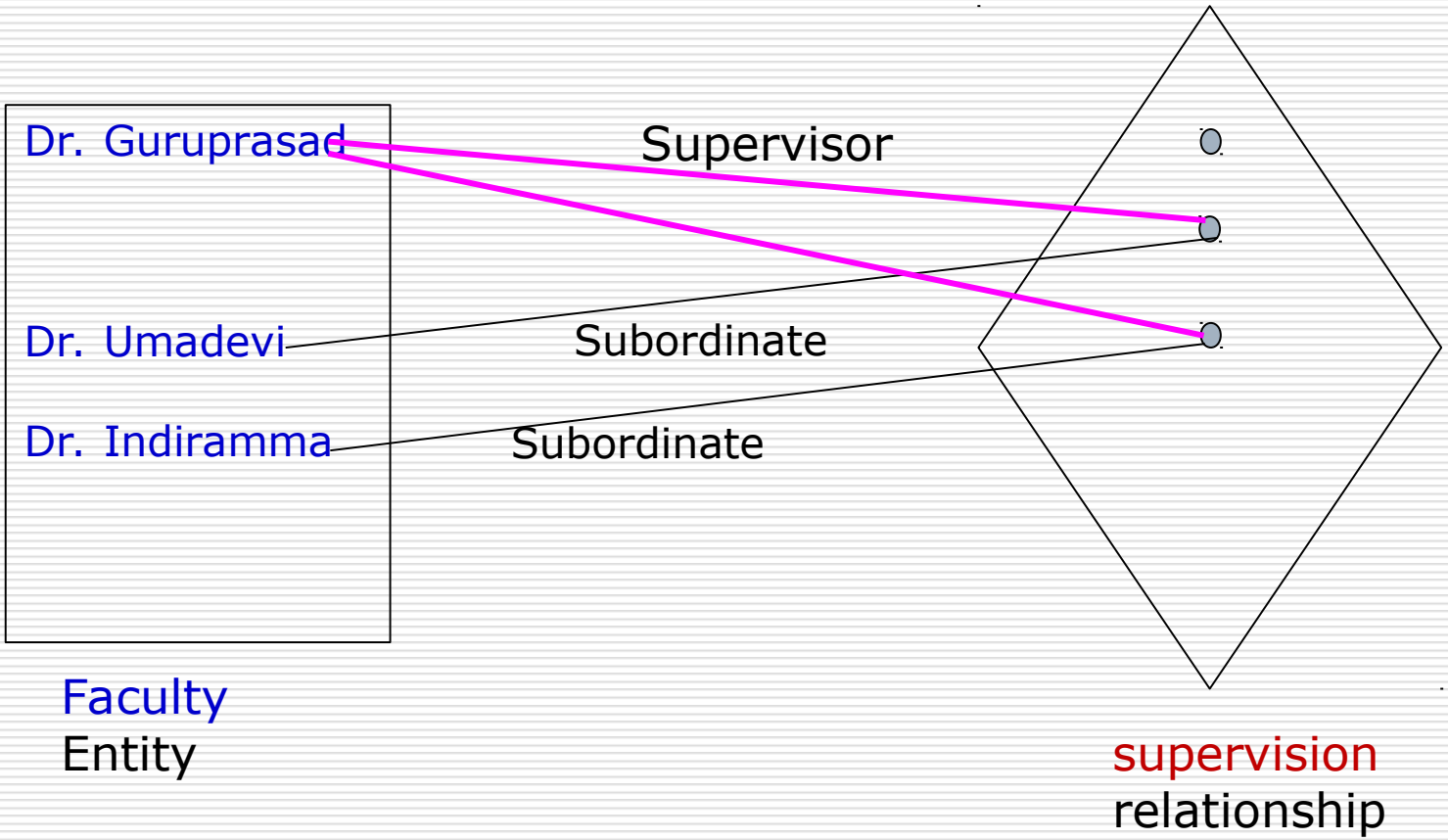
Relationship Degree

- Ternary Relationship: Degree three, three entities are participating



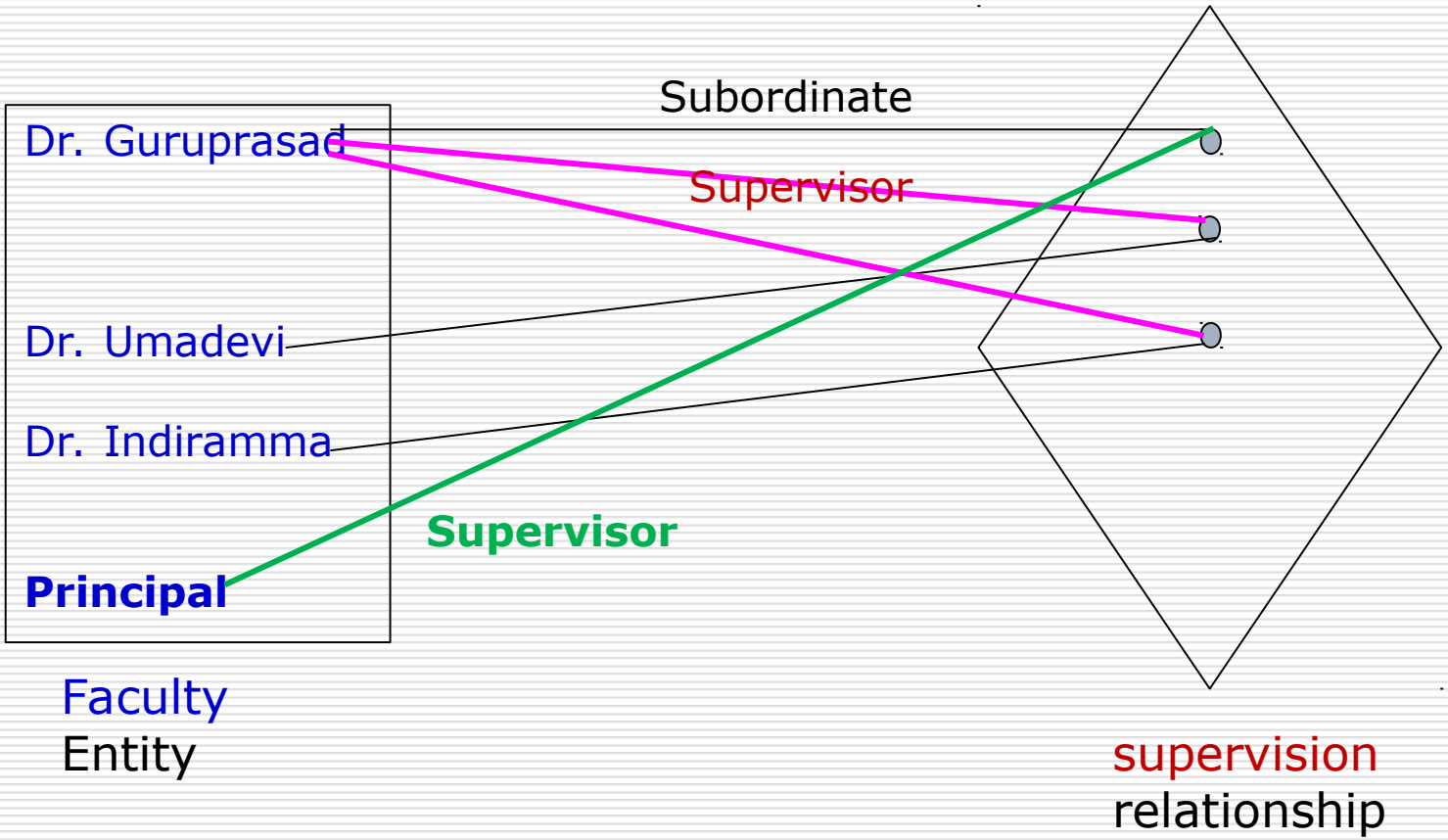
Recursive Relationship

- In some cases the **same entity** type **participates** in **more than once** in a relationship type in **different roles**.



Recursive Relationship

- In some cases the **same entity** type **participates** in **more than once** in a relationship type in **different roles**.



Relationship Constraints or Structural Constraints

Two Types

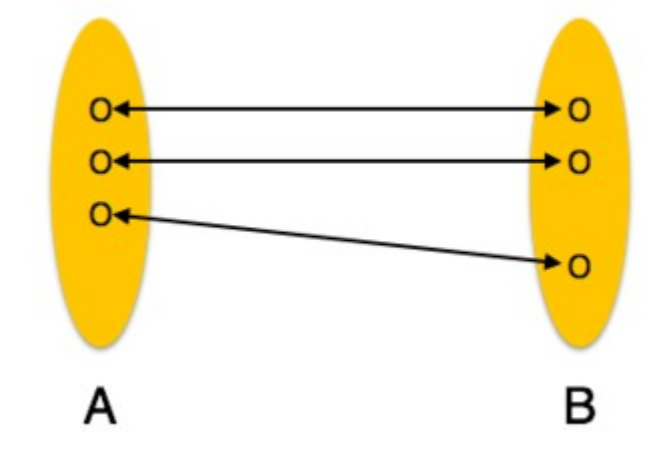
1. Cardinality Ratios
   a. One to one (1:1)
   b. One to Many (1:M)
   c. Many to Many (N:M)
2. Participation Constraints
   a. Total
   b. Partial

Cardinality Ratios

- Cardinality is a constraint on a relationship specifying the number of entity instances that a specific entity may be related to via the relationship.

One to One:

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

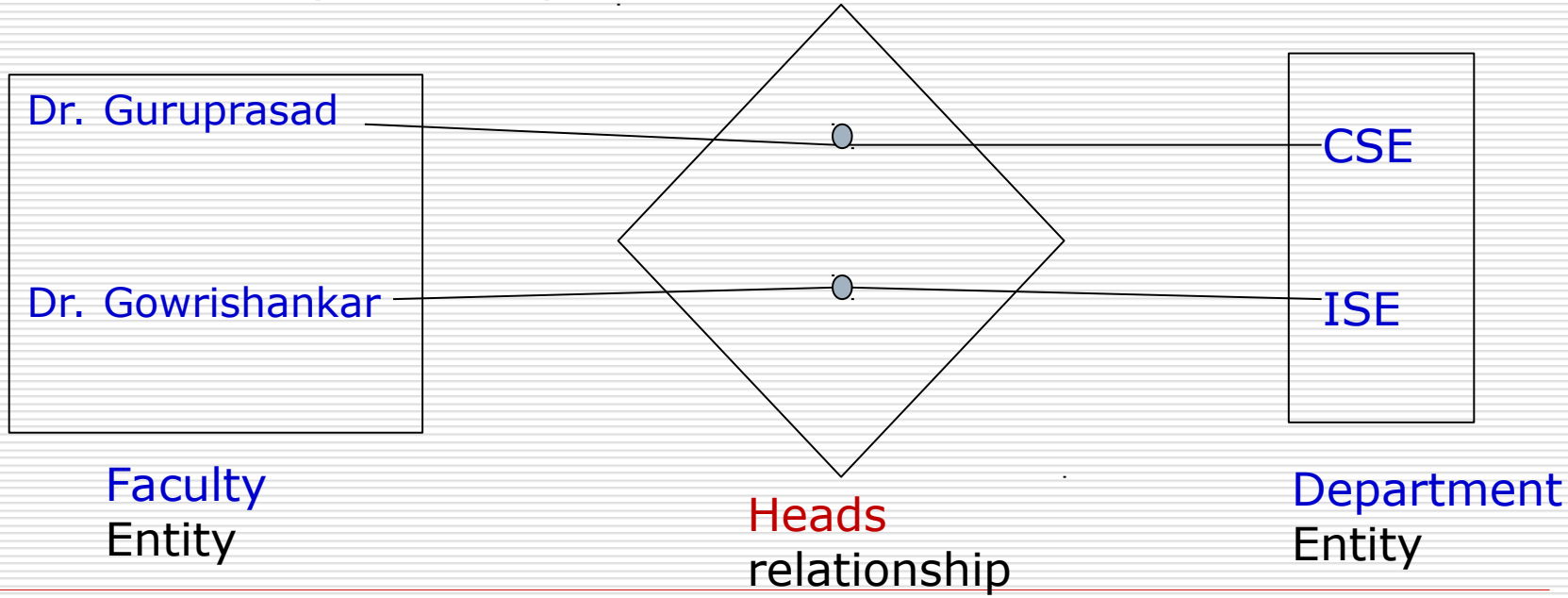


Cardinality Ratios

- Cardinality is a constraint on a relationship specifying the number of entity instances that a specific entity may be related to via the relationship.

One to One:

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

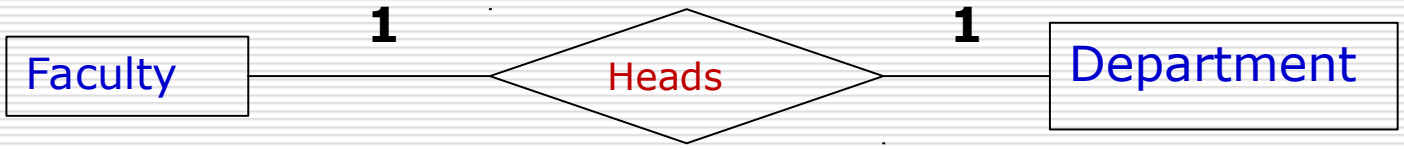


Cardinality Ratios

- Cardinality is a constraint on a relationship specifying the number of entity instances that a specific entity may be related to via the relationship.

One to One:

One instance of one entity type can participate in one instance of other entity type.

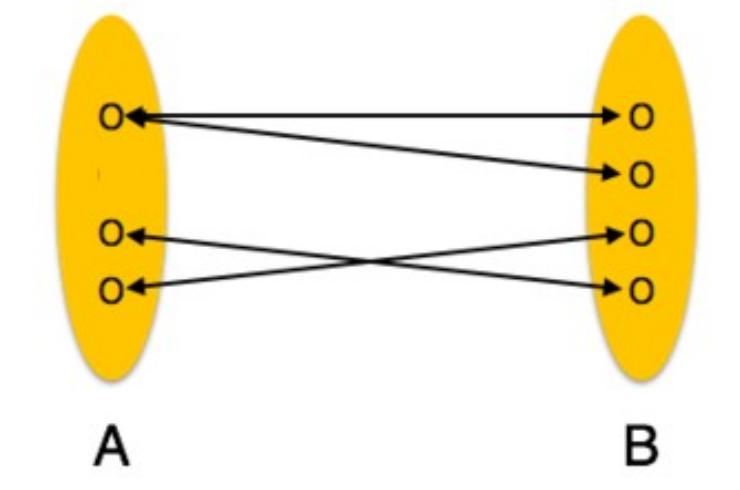


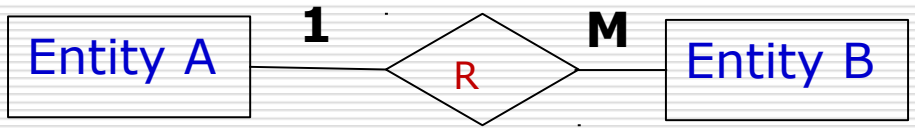
Cardinality Ratios

- Cardinality is a constraint on a relationship specifying the number of entity instances that a specific entity may be related to via the relationship.

One to Many:

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



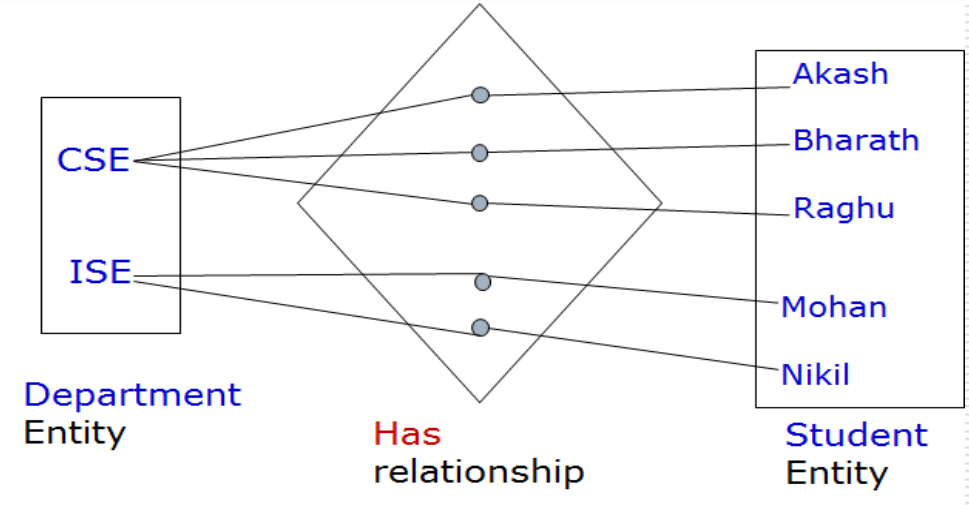


Cardinality Ratios

- Cardinality is a constraint on a relationship specifying the number of entity instances that a specific entity may be related to via the relationship.

One to Many:

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

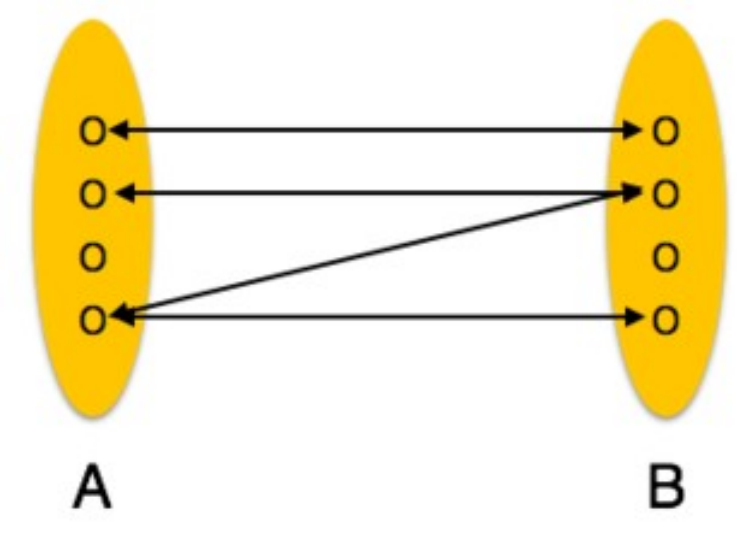


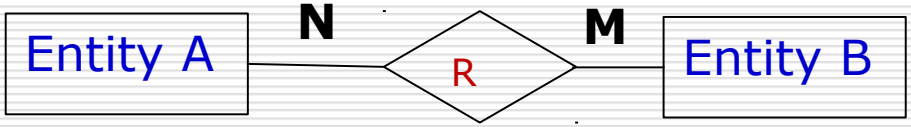
Cardinality Ratios

- Cardinality is a constraint on a relationship specifying the number of entity instances that a specific entity may be related to via the relationship.

Many to Many:

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



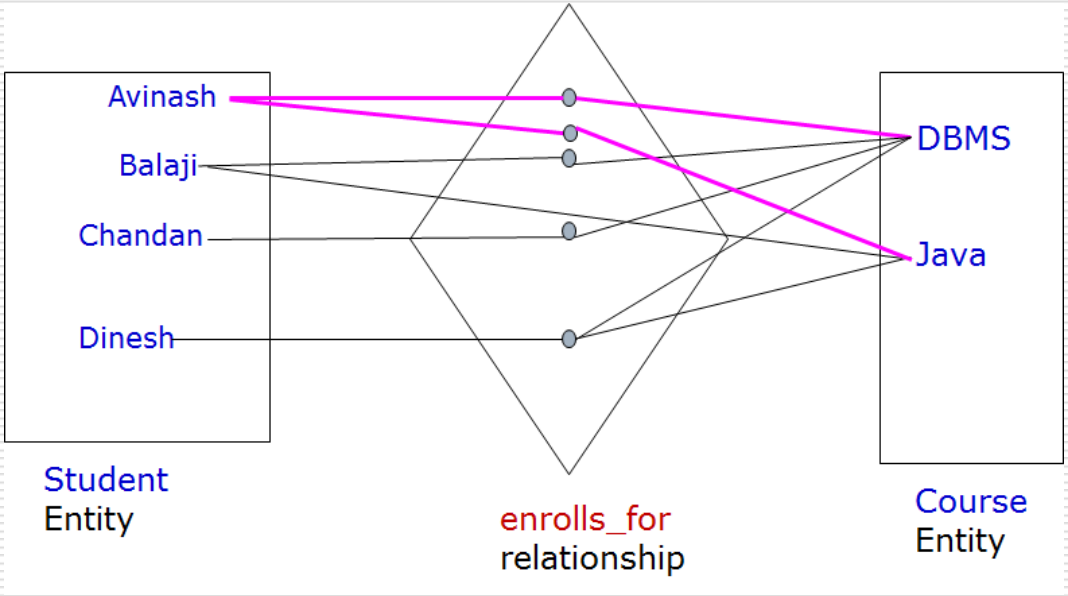


Cardinality Ratios

- Cardinality is a constraint on a relationship specifying the number of entity instances that a specific entity may be related to via the relationship.

Many to Many:

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

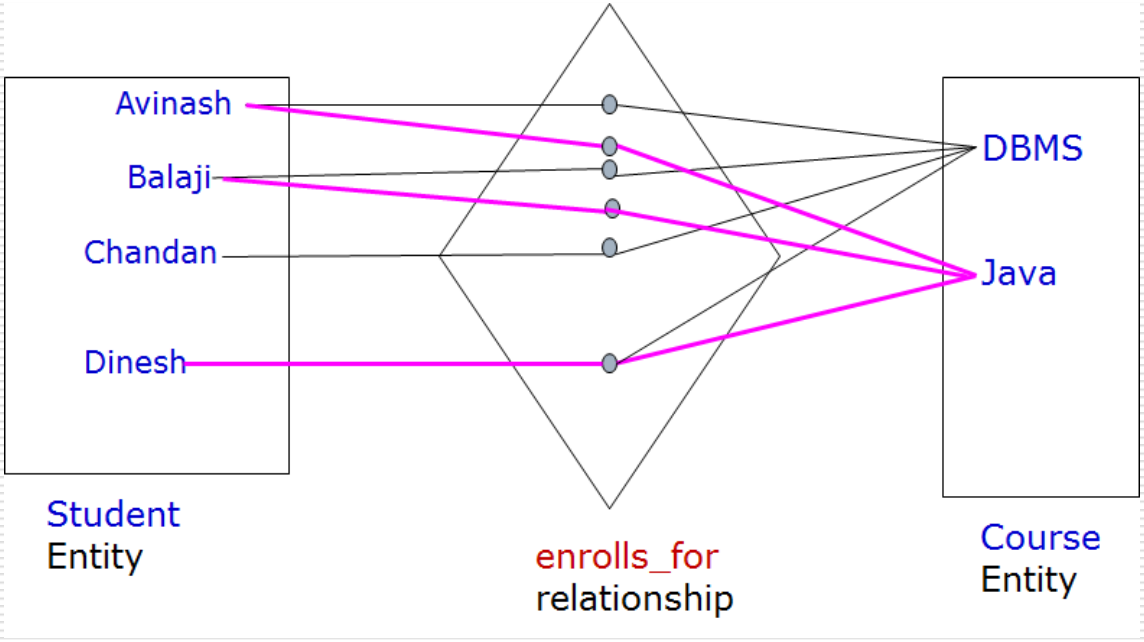


Cardinality Ratios

- Cardinality is a constraint on a relationship specifying the number of entity instances that a specific entity may be related to via the relationship.

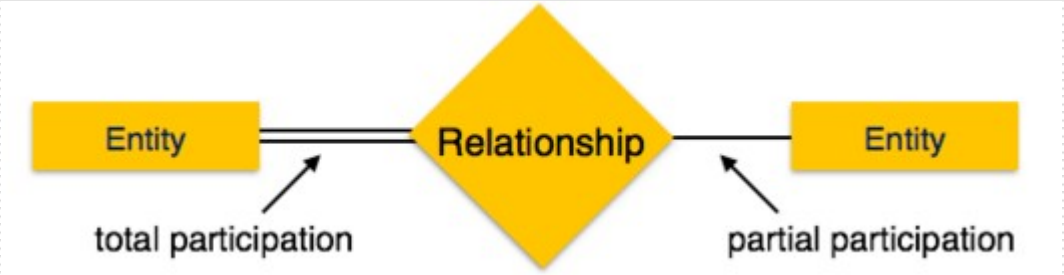
Many to Many:

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



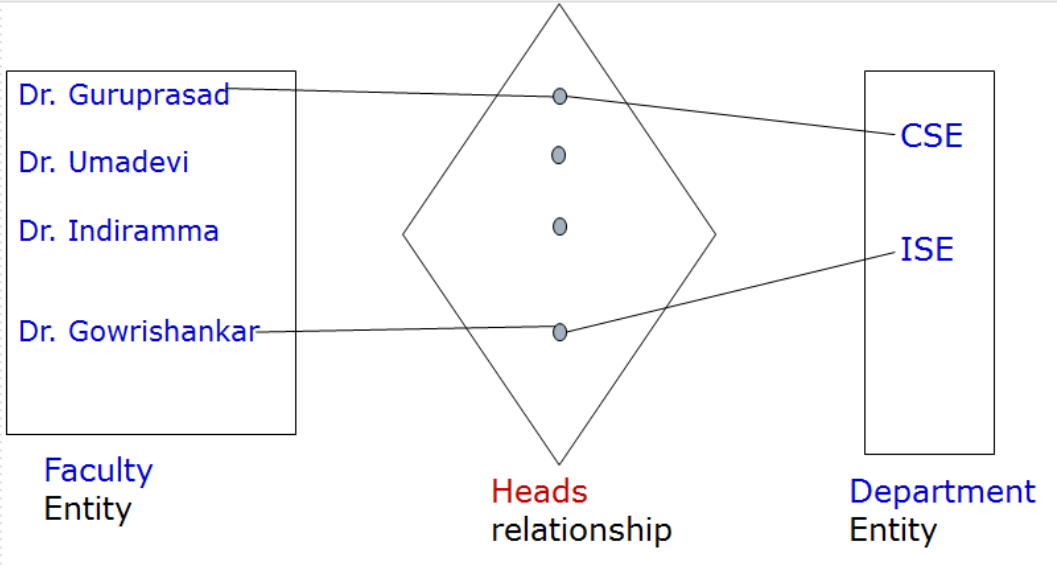
Participation Constraint

- Minimum number of relationship instance that each entity can participate in.
- **Total Participation** – Each entity is involved in the relationship. Total participation is represented by double lines.
- **Partial participation** – Not all entities are involved in the relationship. Partial participation is represented by single lines.



Participation Constraint

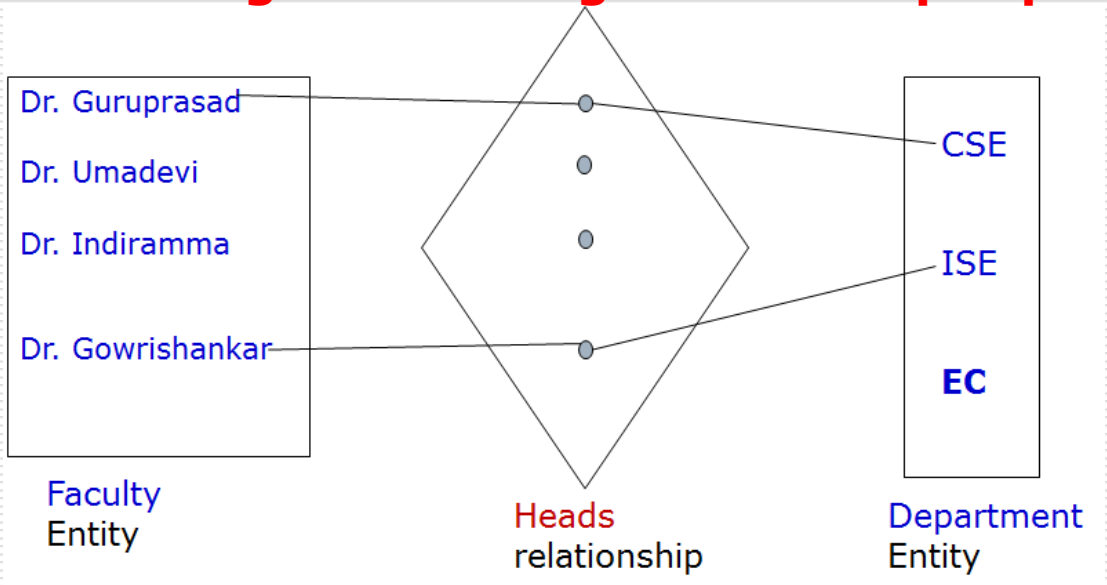
- Minimum number of relationship instance that each entity can participate in.
- **Total Participation** – Each entity is involved in the relationship. Total participation is represented by double lines.
- **Partial participation** – Not all entities are involved in the relationship. Partial participation is represented by single lines.



Participation Constraint

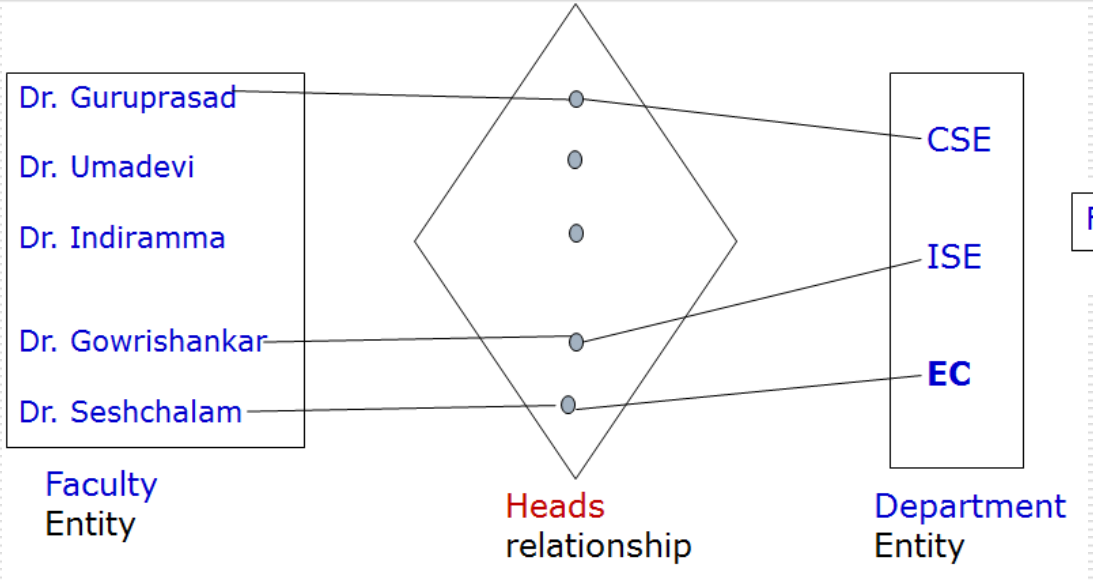
- Minimum number of relationship instance that each entity can participate in.
- **Total Participation** – Each entity is involved in the relationship. Total participation is represented by double lines.
- **Partial participation** – Not all entities are involved in the relationship. Partial participation is represented by single lines.

What is wrong in following relationship representation ?



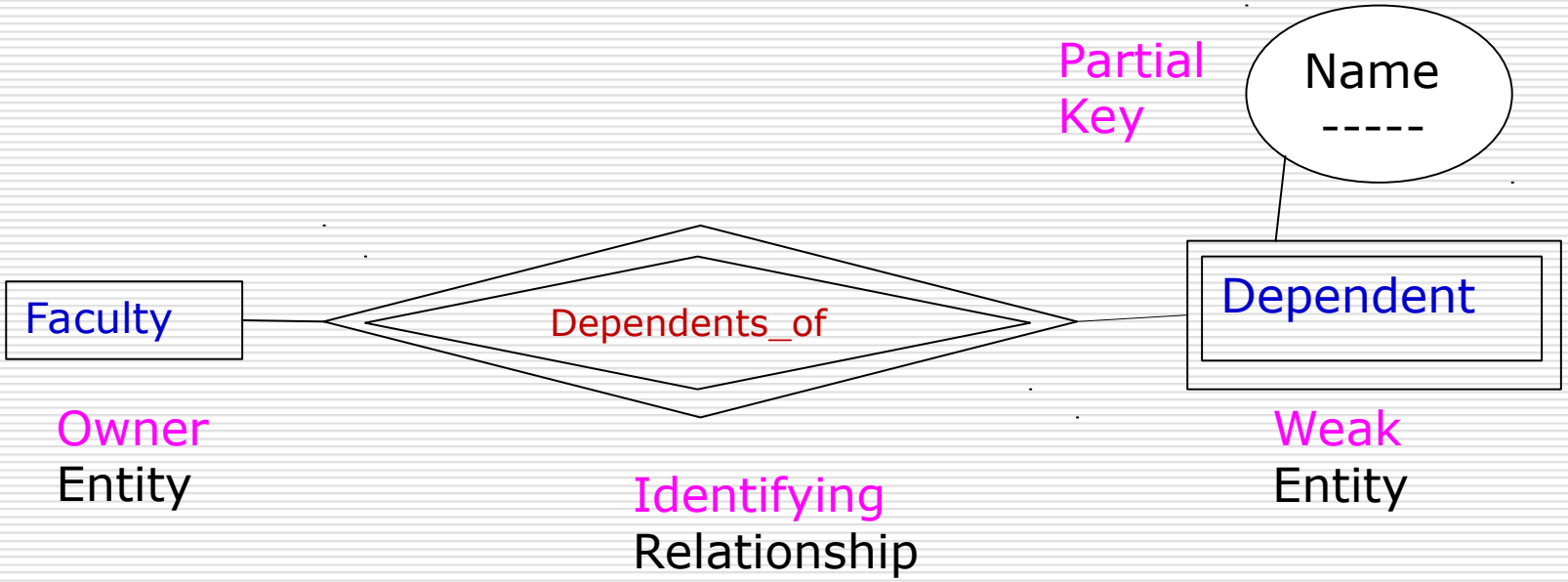
Participation Constraint

- Minimum number of relationship instance that each entity can participate in.
- **Total Participation** – Each entity is involved in the relationship. Total participation is represented by double lines.
- **Partial participation** – Not all entities are involved in the relationship. Partial participation is represented by single lines.



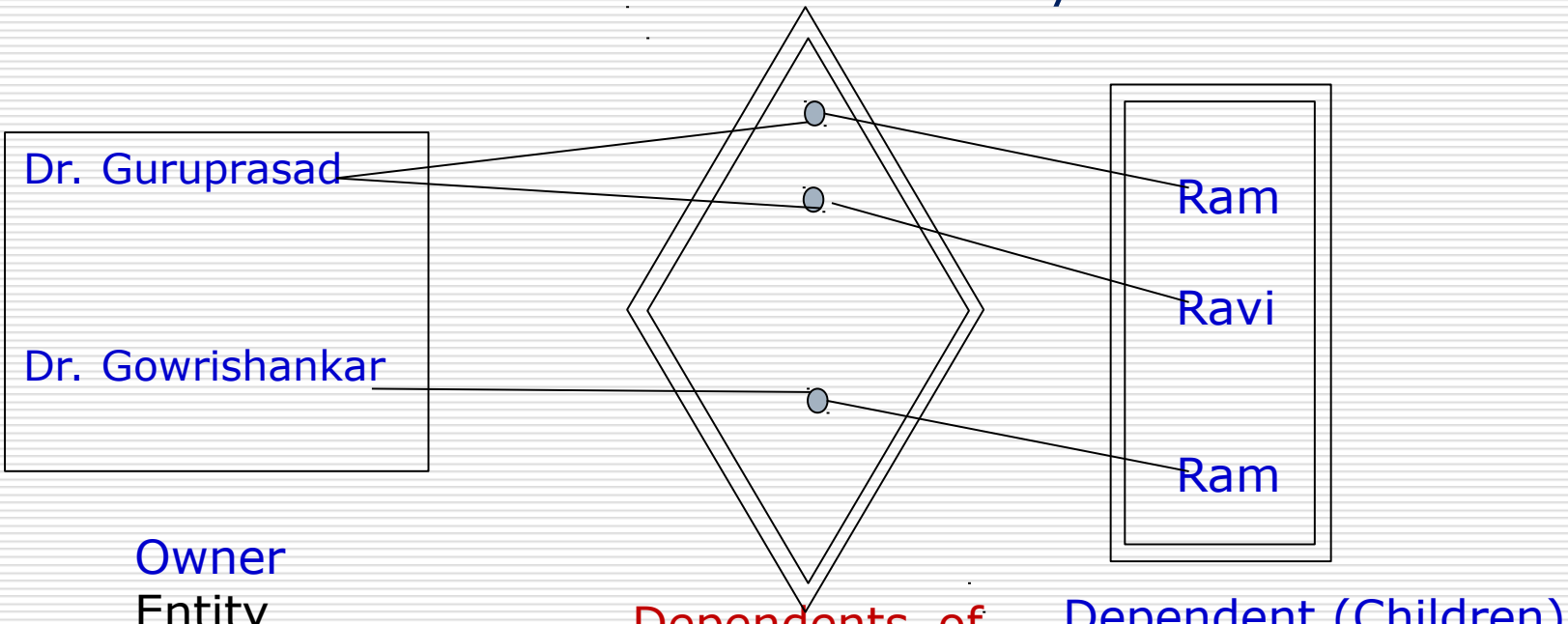
Weak Entity

- A weak entity set is one which **does not have any primary key** associated with it.
- A weak entity type normally has partial key which is the set of attributes that can uniquely identify weak entities that are related to same owner entity.

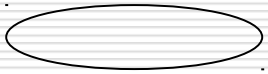


Weak Entity

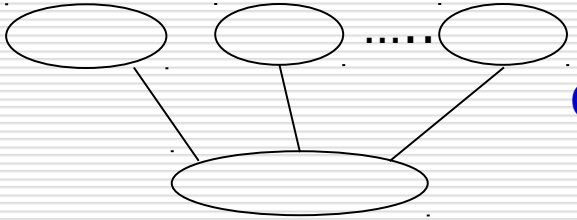
- A weak entity set is one which **does not have any primary key** associated with it.
- A weak entity type normally has partial key which is the set of attributes that can uniquely identify weak entities that are related to same owner entity.



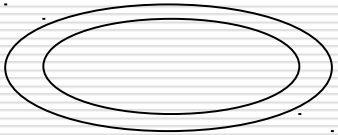
ER Diagram Symbols



Attribute

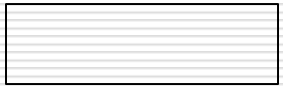


Composite Attribute

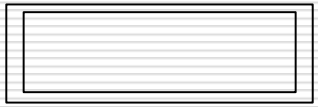


Multivalued Attribute

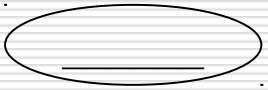
Derived Attribute



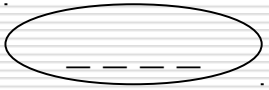
Entity



Weak Entity

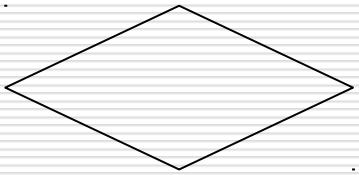


Key Attribute

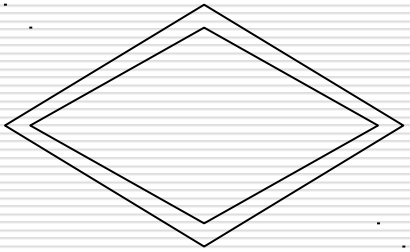


Partial Key

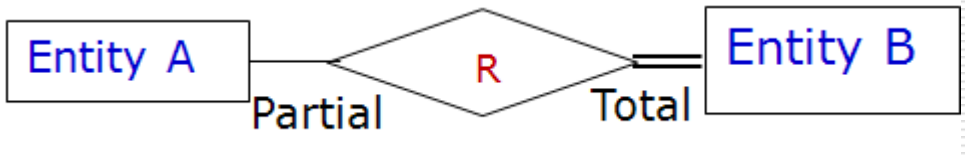
ER Diagram Symbols



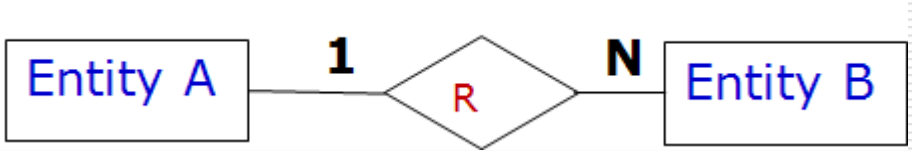
Relationship



Identifying Relationship



Total Participation of Entity A in R



Cardinality Ratio 1:N for Entity A : B in R

Topics Covered in Todays Class

Unit 2:
ER diagram design for a given requirements

Database Design

Database design: Why do we need it ?

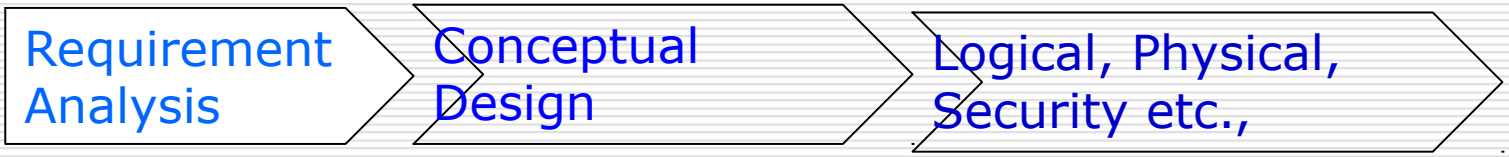
- Agree on structure of the database before deciding on a particular implementation

Consideration Issues such as:

- What entities to model
- How entities are related
- What constraints exist in the domain
- How to achieve good design

ER diagram is a conceptual design of database

Database Design Process

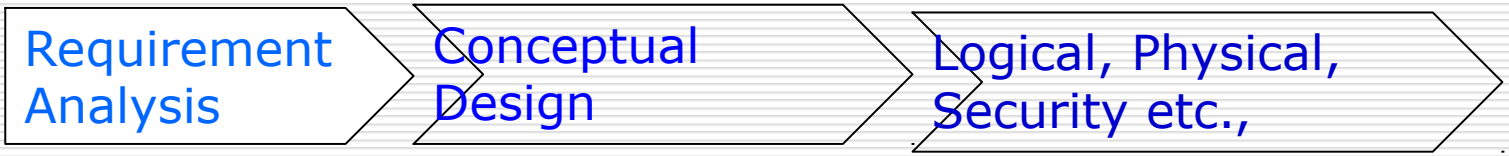


Requirement Analysis

- What is going to be stored ?
- How is it going to be used ?
- What are we going to do with the data ?
- Who should access the data ?

Technical and Non-technical people are involved

Database Design Process

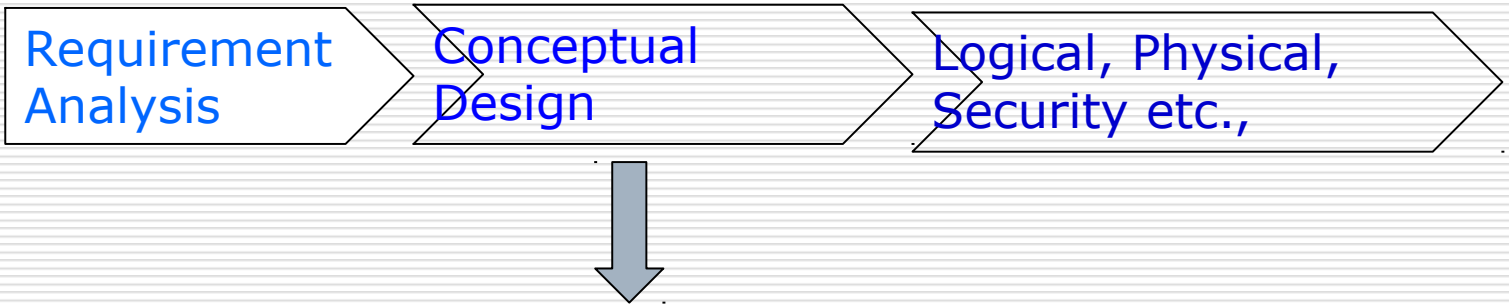


Conceptual Design

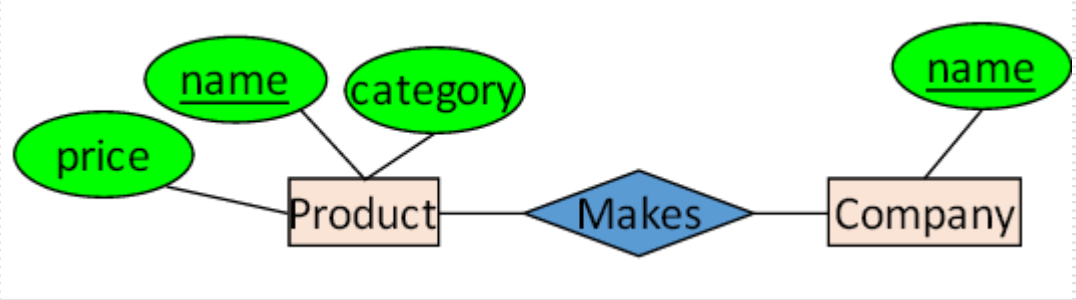
- A high-level description of the database.
- Sufficiently precise that technical people can understand it.
- But, not so precise that non-technical can not participate.

This is were Entity-Relationship (ER) Diagram fits in.

Database Design Process



ER diagram is **visual syntax for DB design** which is precise enough for technical points, but abstracted enough for non-technical people.



Main phases of Database Design

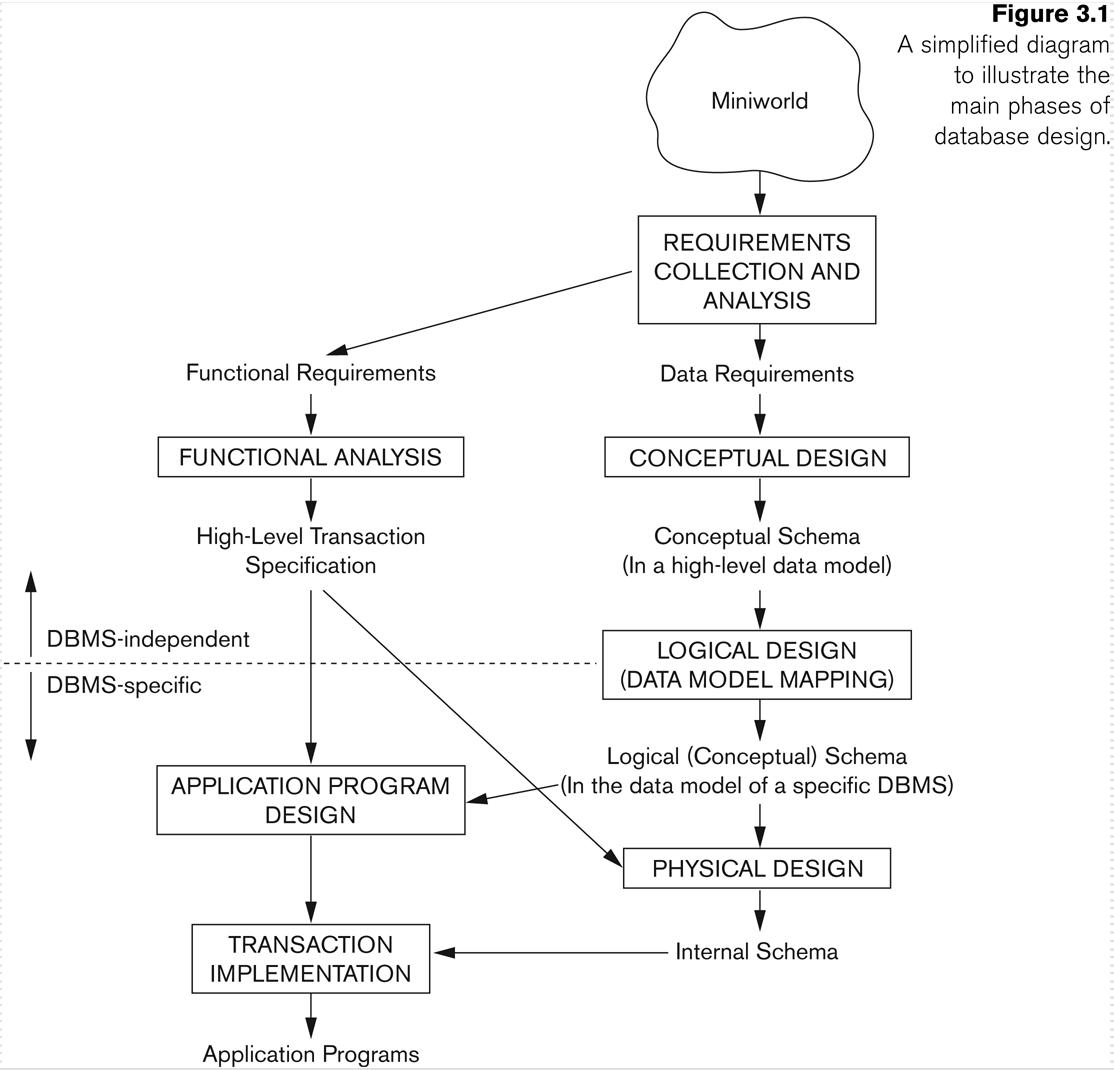
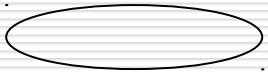
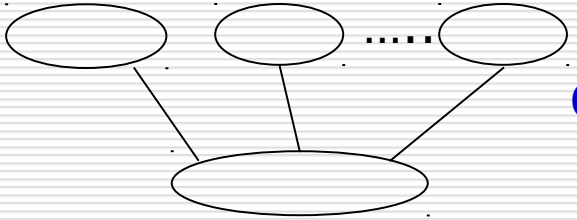


Figure 3.1
A simplified diagram to illustrate the main phases of database design.

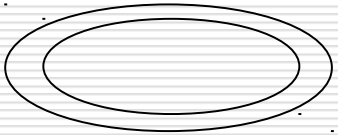
ER Diagram Symbols



Attribute

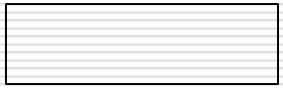


Composite Attribute

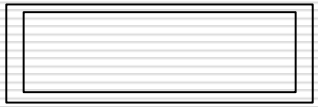


Multivalued Attribute

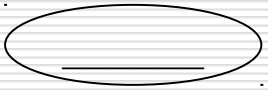
Derived Attribute



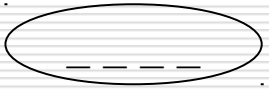
Entity



Weak Entity

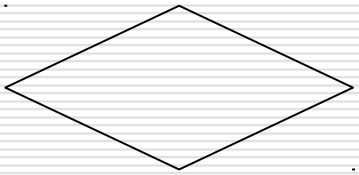


Key Attribute

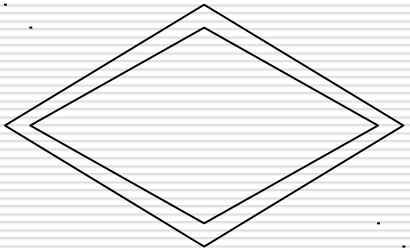


Partial Key

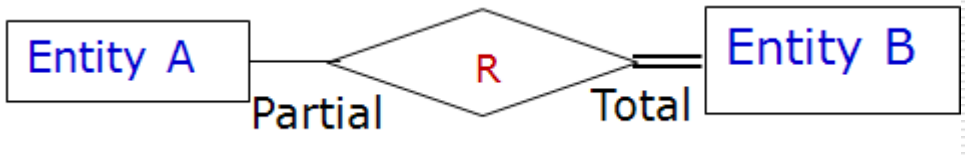
ER Diagram Symbols



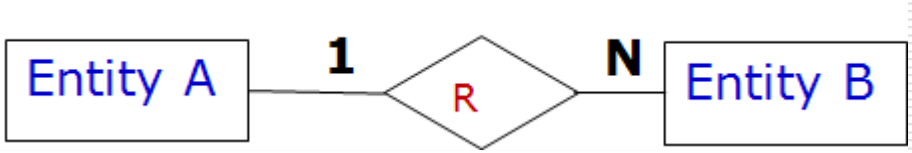
Relationship



Identifying Relationship



Total Participation of Entity A in R

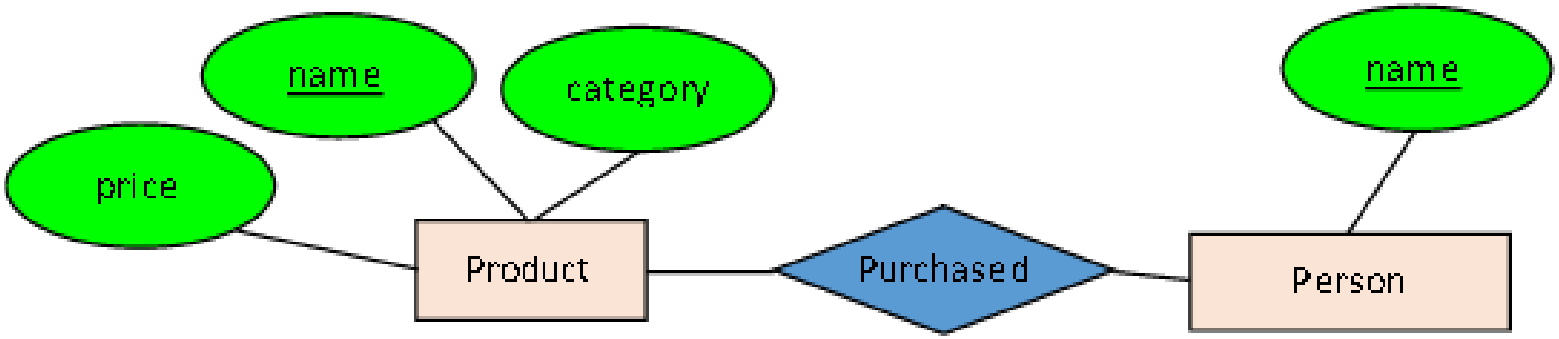


Cardinality Ratio 1:N for Entity A : B in R

Activity: ER diagram

Decision: Relationship vs Empty

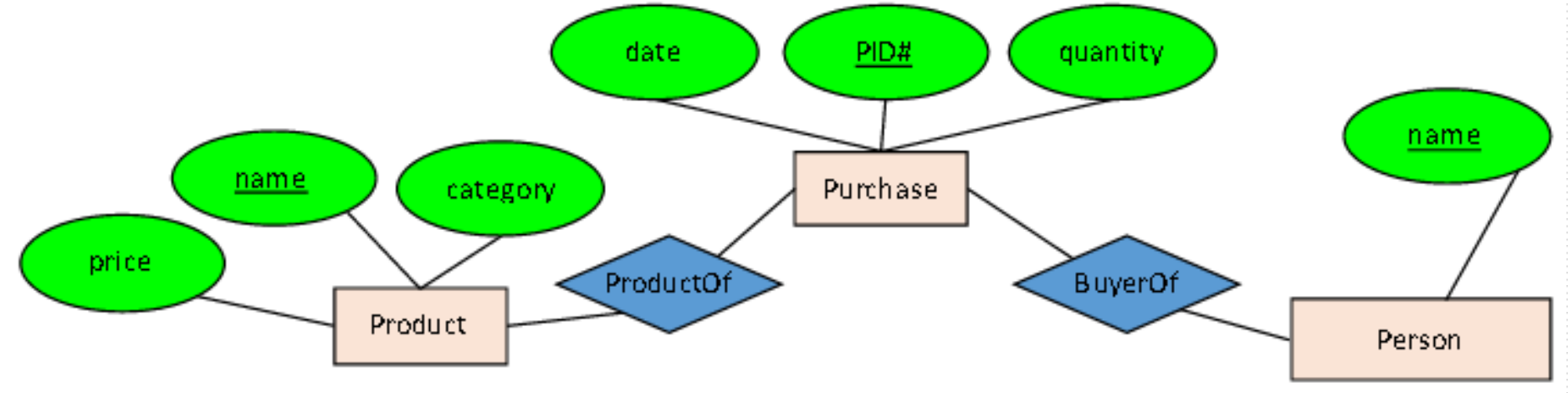
Question 1: What does the following ER diagram say ?



Activity: ER diagram

Decision: Relationship vs Empty

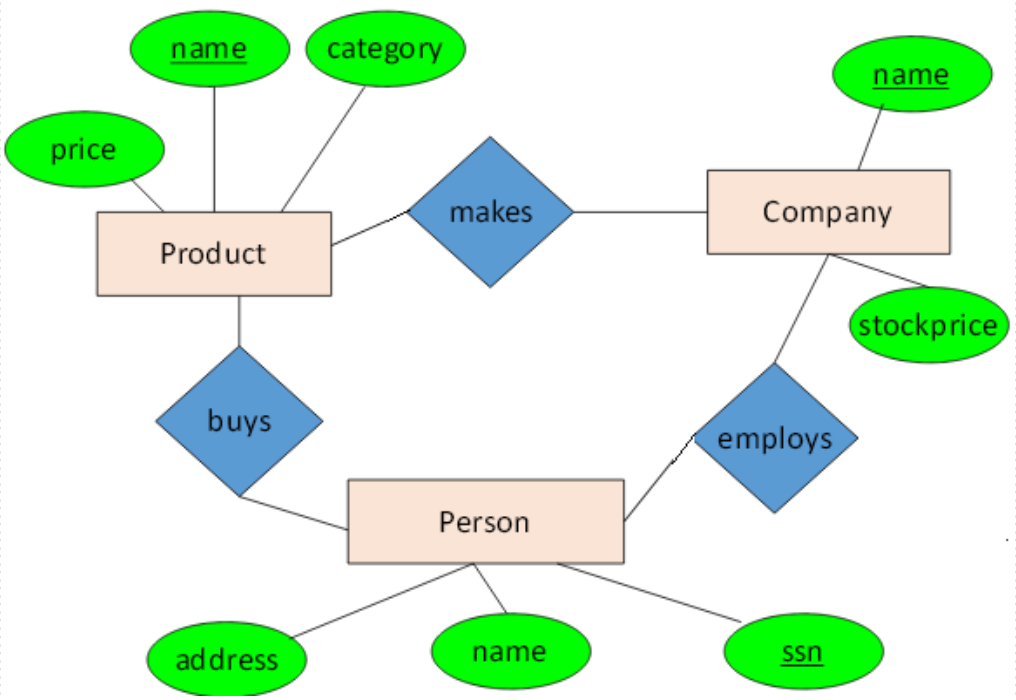
Question 2: What does the following ER diagram say ?



Activity: ER diagram

Decision: Relationship vs Empty

Question 3: What does the following ER diagram say ?



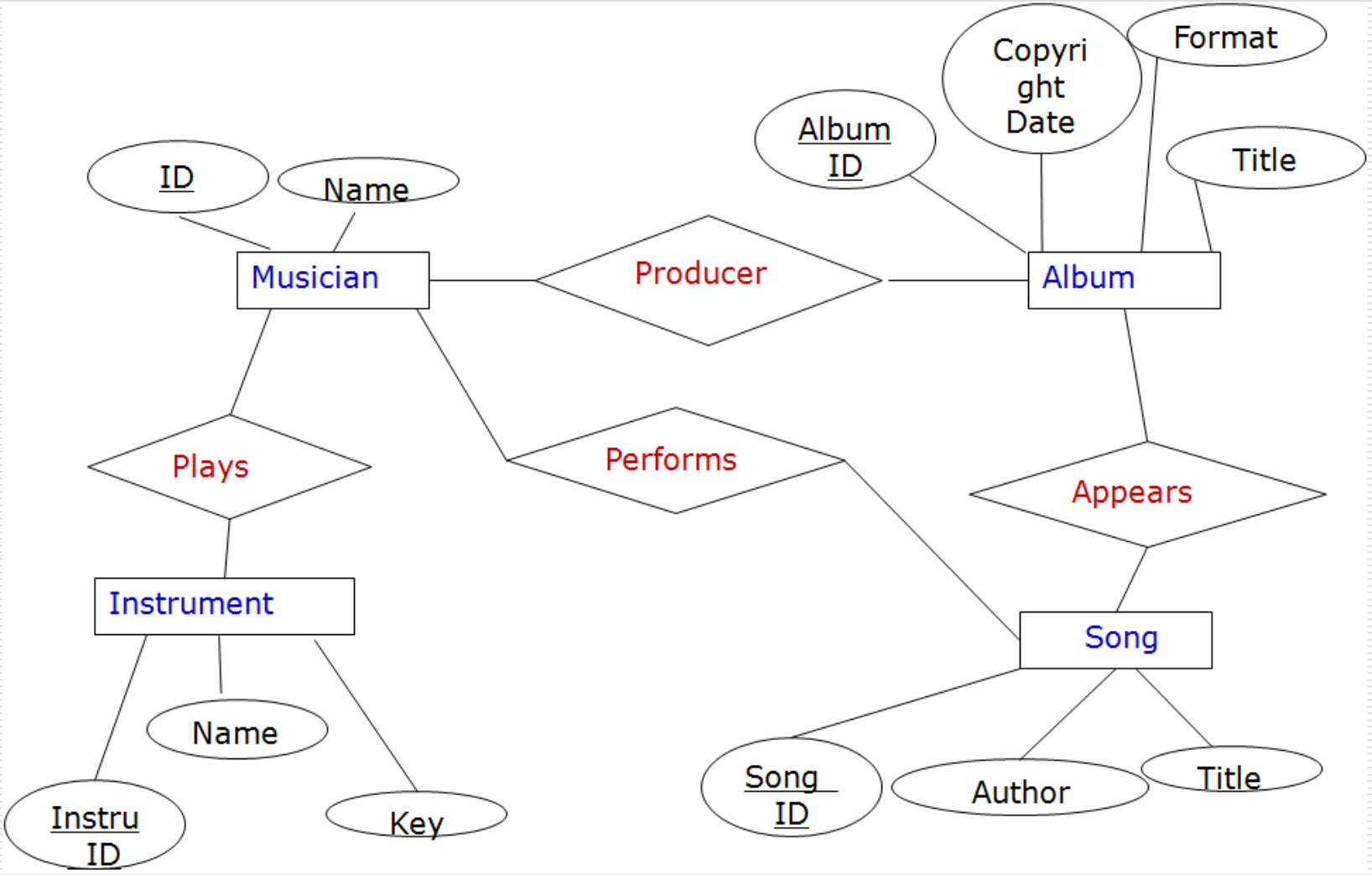
Activity: ER diagram design

Draw ER diagram for the following requirements

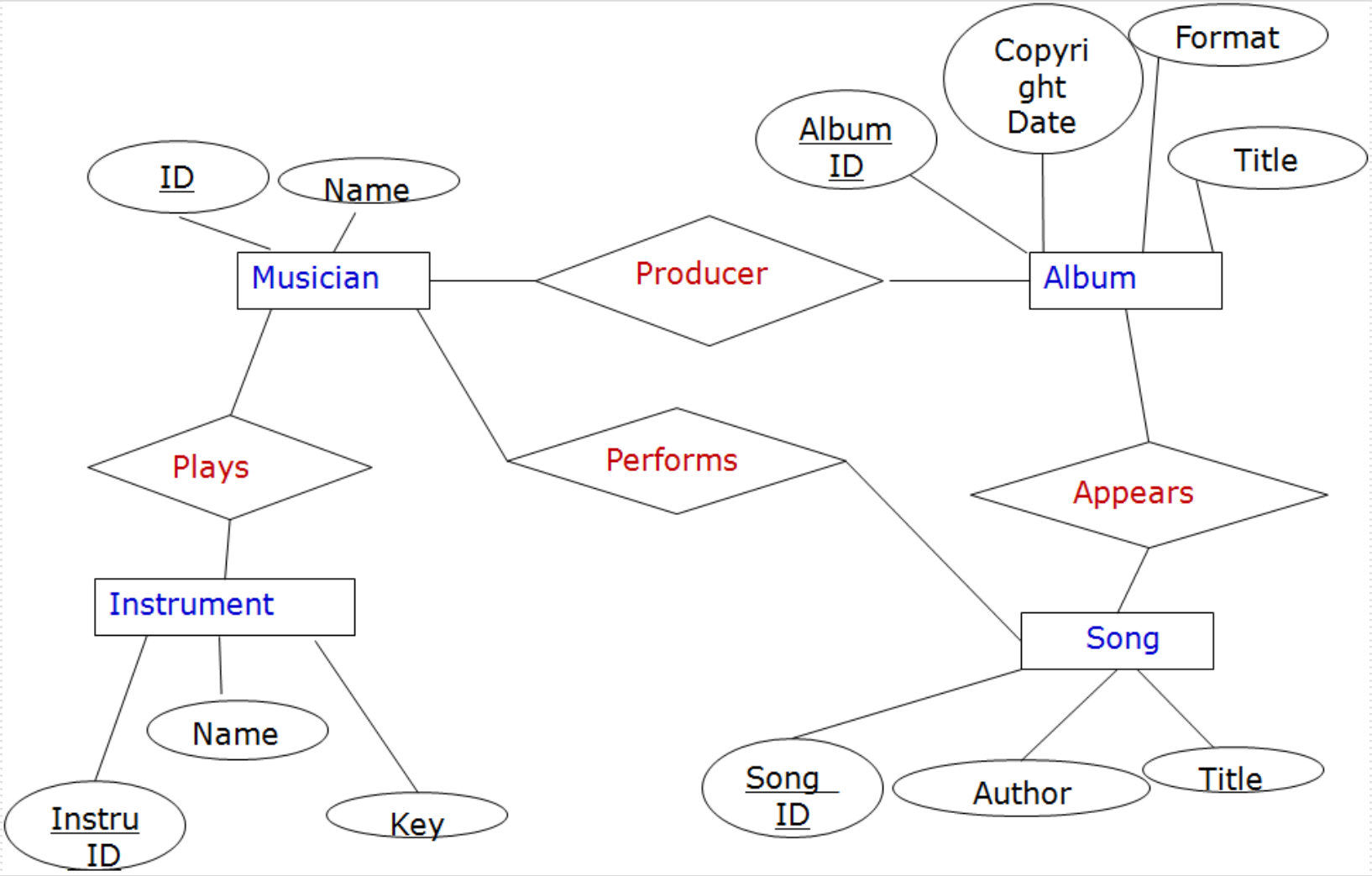
Universal Records has decided to store information about musicians who perform on its albums (as well as other company data) in a database.

- Each musician that records at Universal has an ID(Key) and name.
- Each instrument used in songs recorded at Universal has a unique identification number (Key) , a name and a musical key.
- Each album recorded on the Universal label has a unique identification number (Key) , a title, a copyright date and a format.
- Each song recorded at Universal has a Song ID (Key), a title and an author.
- Each musician may play several instruments, and a given instrument may be played by several musicians.
- Each album has a number of songs on it, but no song may appear on more than one album.
- Each song is performed by one or more musicians, and a musician may perform a number of songs.
- Each album has exactly one musician who acts as its producer. A musician may produce several albums, of course.

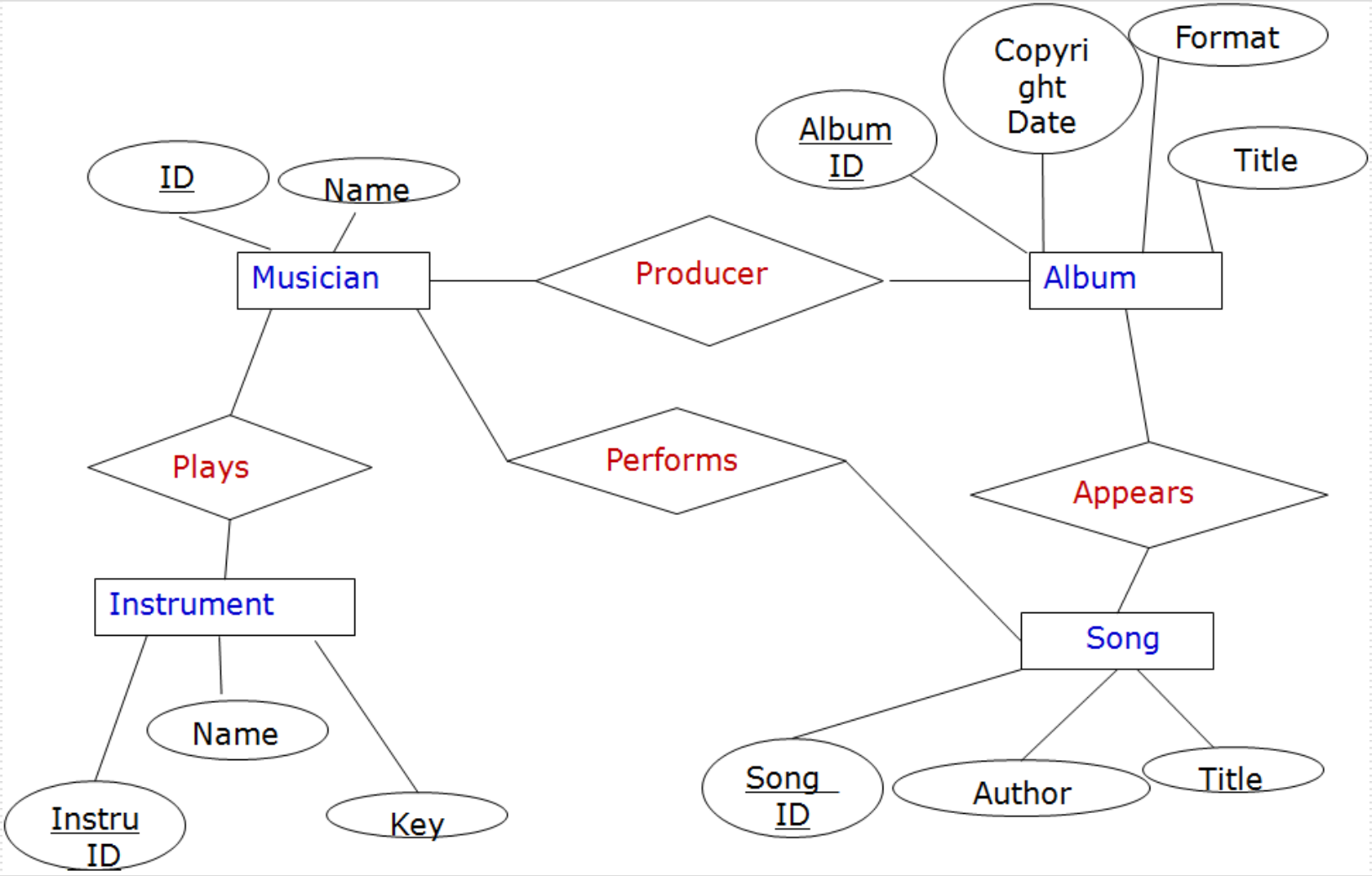
Each musician that records at Universal has an ID (Key) and name.



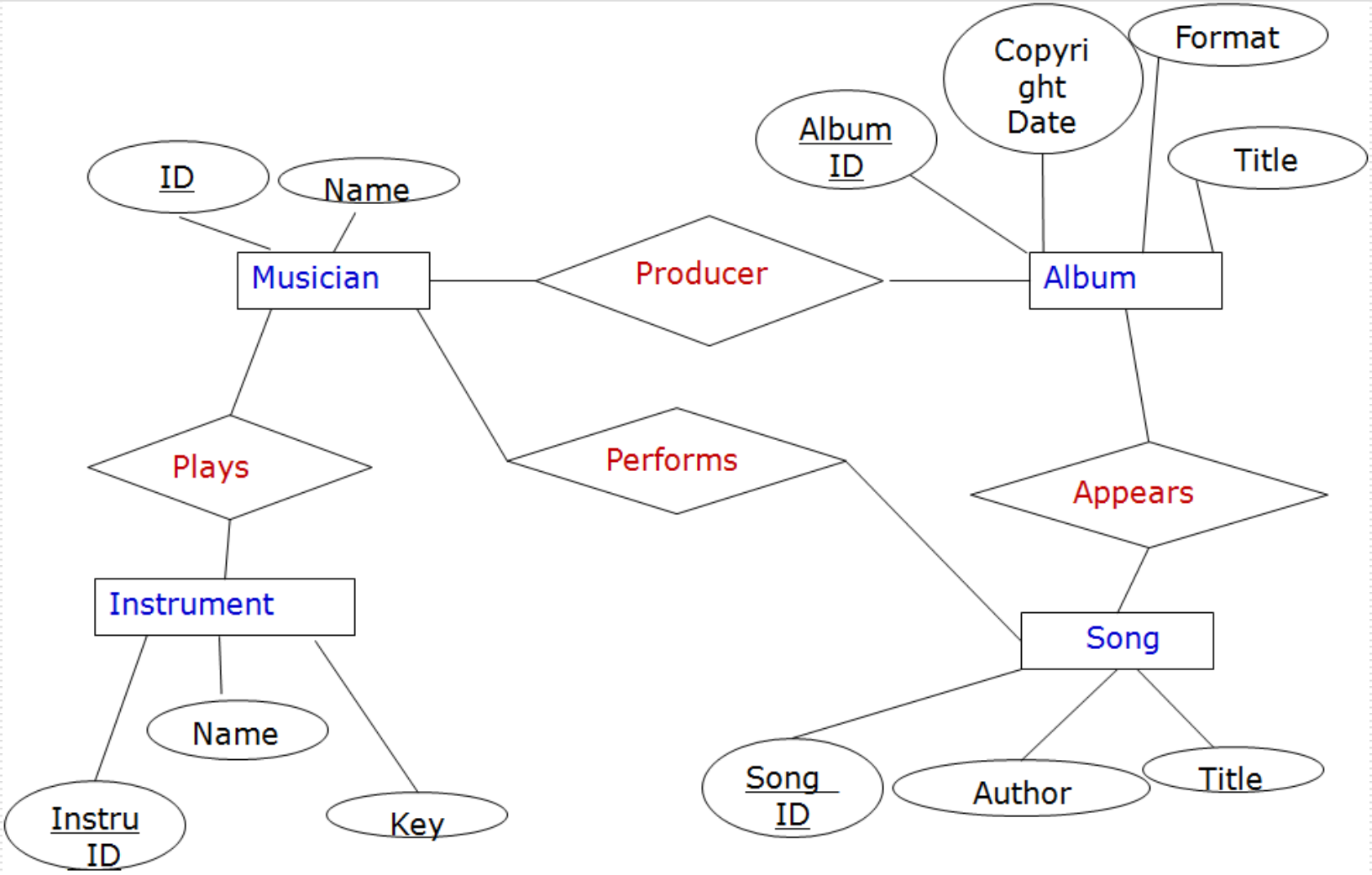
Each instrument used in songs recorded at Universal has a unique identification number (Key) , a name and a musical key.



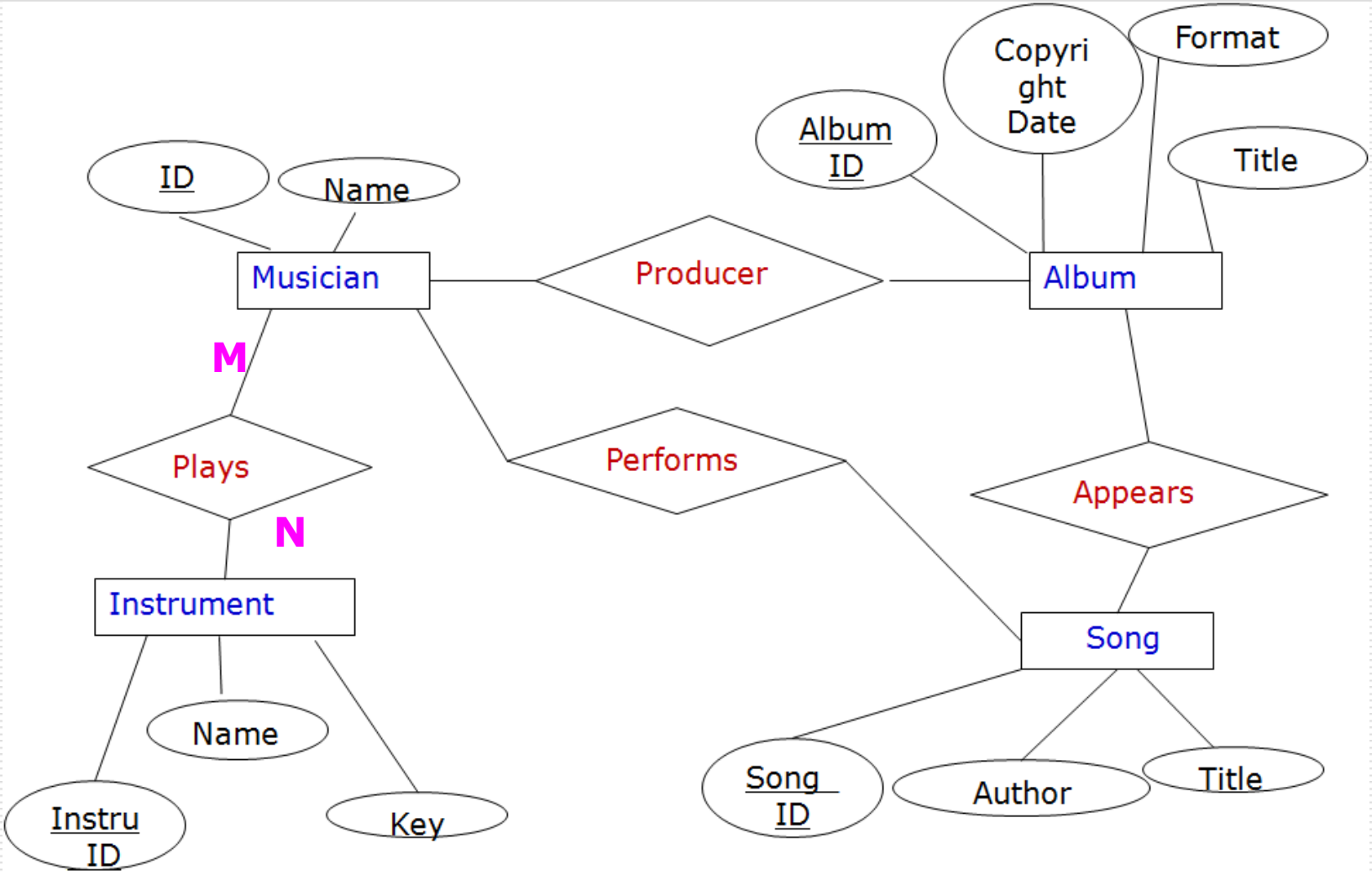
Each album recorded on the Universal label has a unique identification number (Key) , a title, a copyright date and a format.



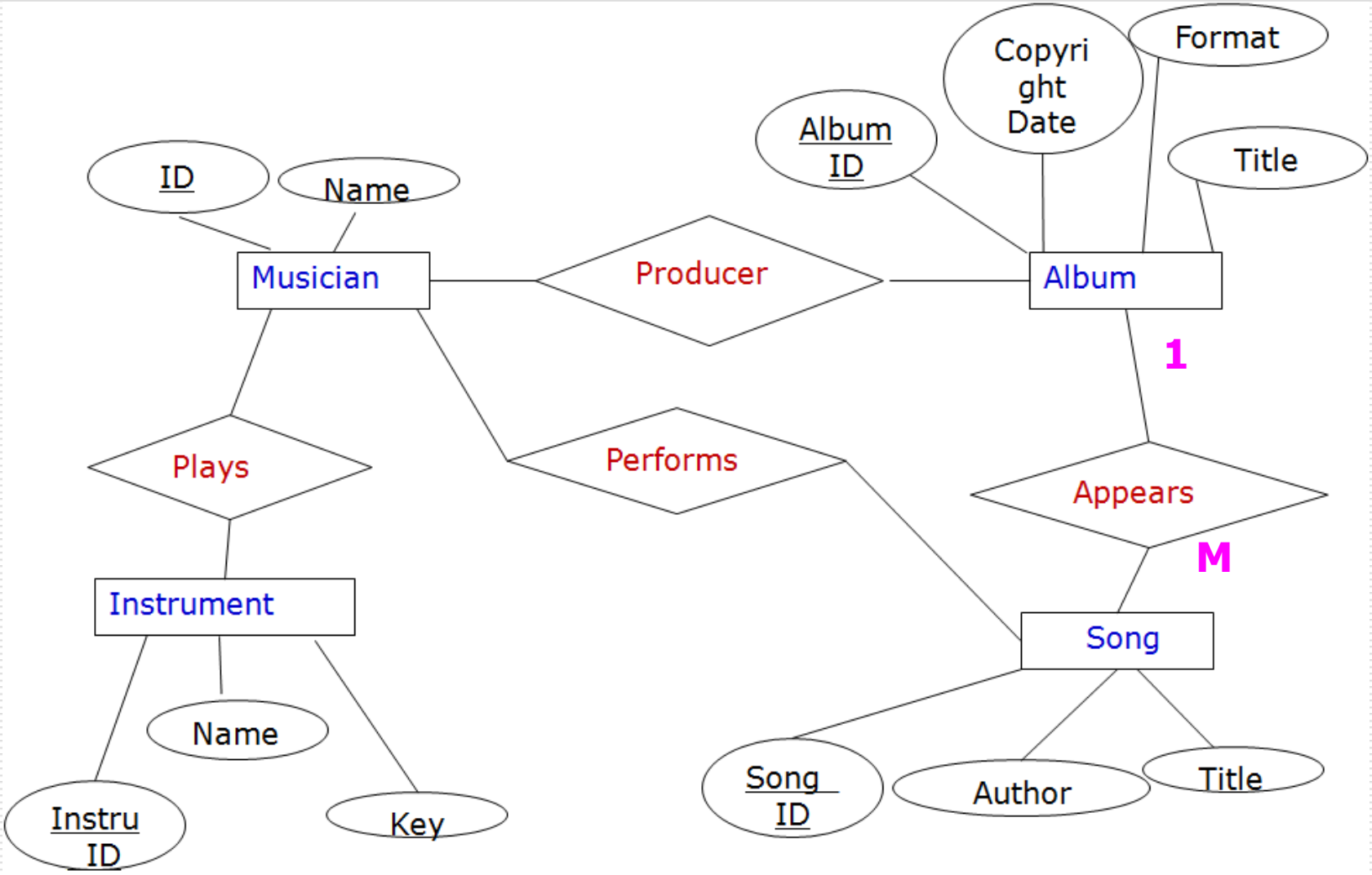
Each song recorded at Universal has a Song ID (Key), a title and an author.



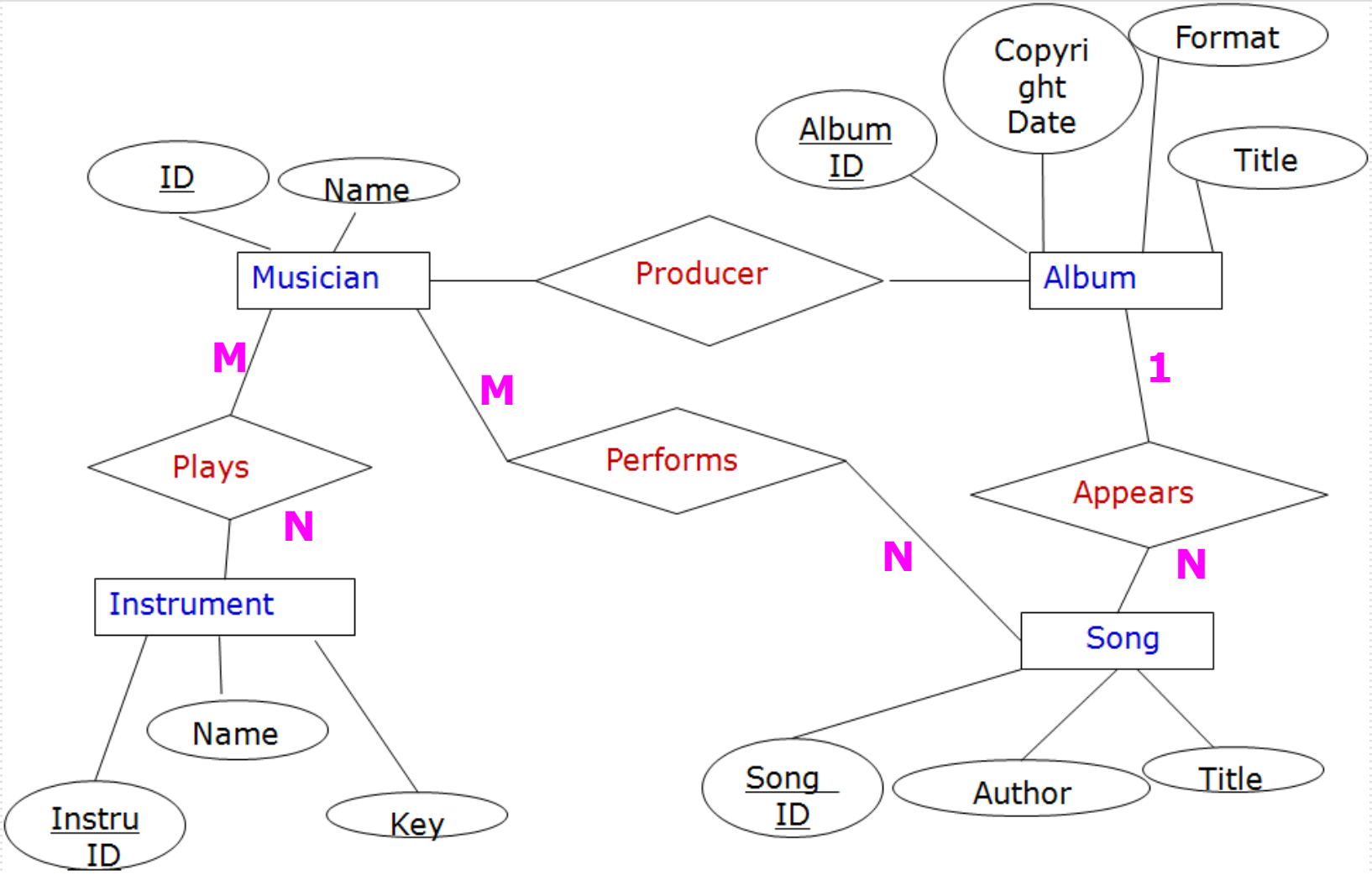
Each musician may play several instruments, and a given instrument may be played by several musicians.



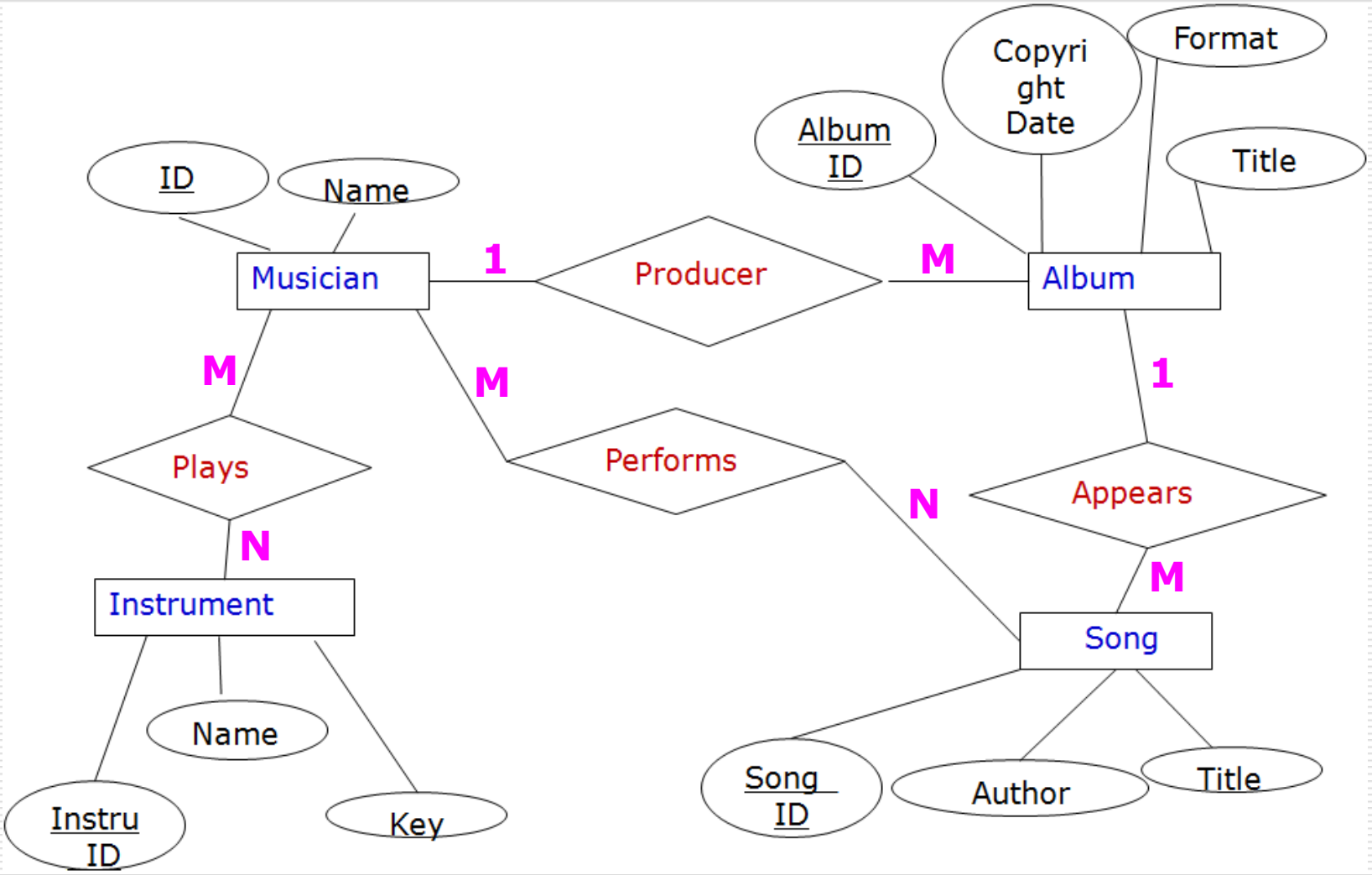
Each album has a number of songs on it, but no song may appear on more than one album.



Each song is performed by one or more musicians, and a musician may perform a number of songs.



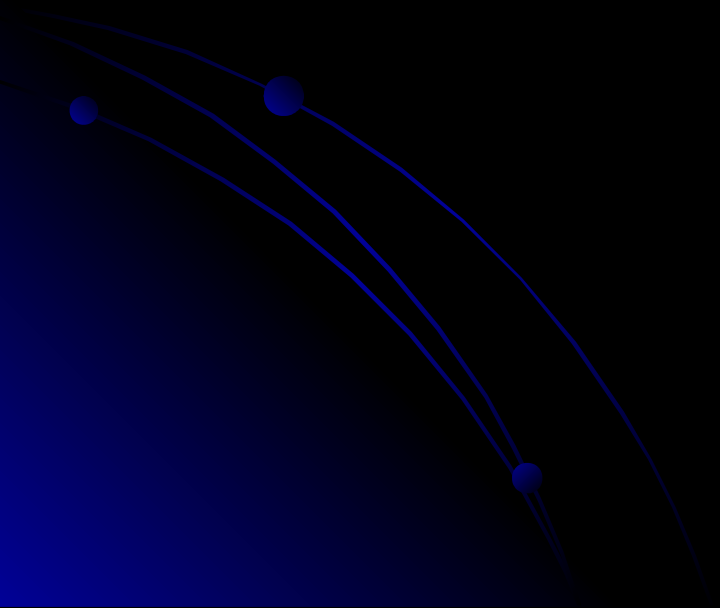
Each album has exactly one musician who acts as its producer. A musician may produce several albums, of course



Session 4

© 2017-18 KL University – The contents of this presentation are an intellectual and copyrighted property of KL University. ALL RIGHTS RESERVED

Enhanced Entity Relationship (EER) Model



The Entity (Review)

- Entity Relationship (ER) Model – represents an object
  - Physical – person, car
  - Conceptual – school, company
- ER model is based on the perception of the real world as a collection of objects with attributes
- Attributes – describe the entity
  - Single, Multi-value
  - Composite, Simple
  - Derived, Stored

What is an EER Model?

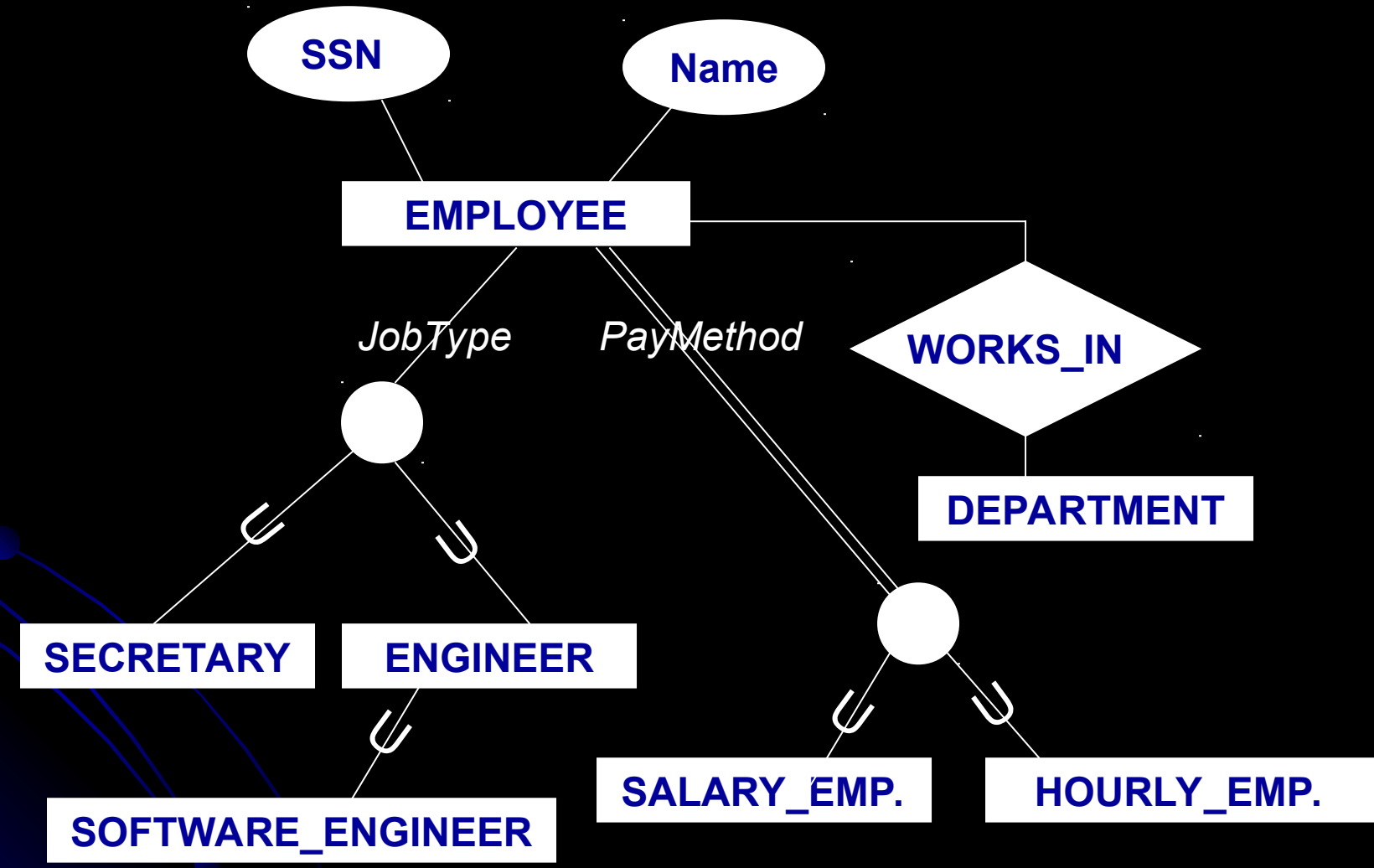
- Enhanced Entity Relationship (EER) – Data Modeling
- EER shows complex relationships between objects in a database (multimedia, geographical).
- Concepts of subclasses and superclasses, specializations and generalizations.
- Put concepts in diagram to form EER model

Specialization

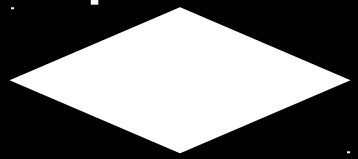
Subgrouping into subclasses (top-down approach)

- Example: EMPLOYEE -> SECRETARY MANAGER, etc.
- Inheritance – Inherit attributes and relationships from superclass (Name, Birthdate, etc.)
- Subclasses may have uniqe attributes
  - SECRETARY has TypingSpeed attribute, MANAGER has BusinessUnitManaged, etc.

Specialization (cont.)



Model Shapes

- When you have more than one subclass based on the same defining attribute (*JobType*), use 
- To show class/subclass relationships, use 
  - Used for relationships between entity types
- To show relationship between two different entity types, use 

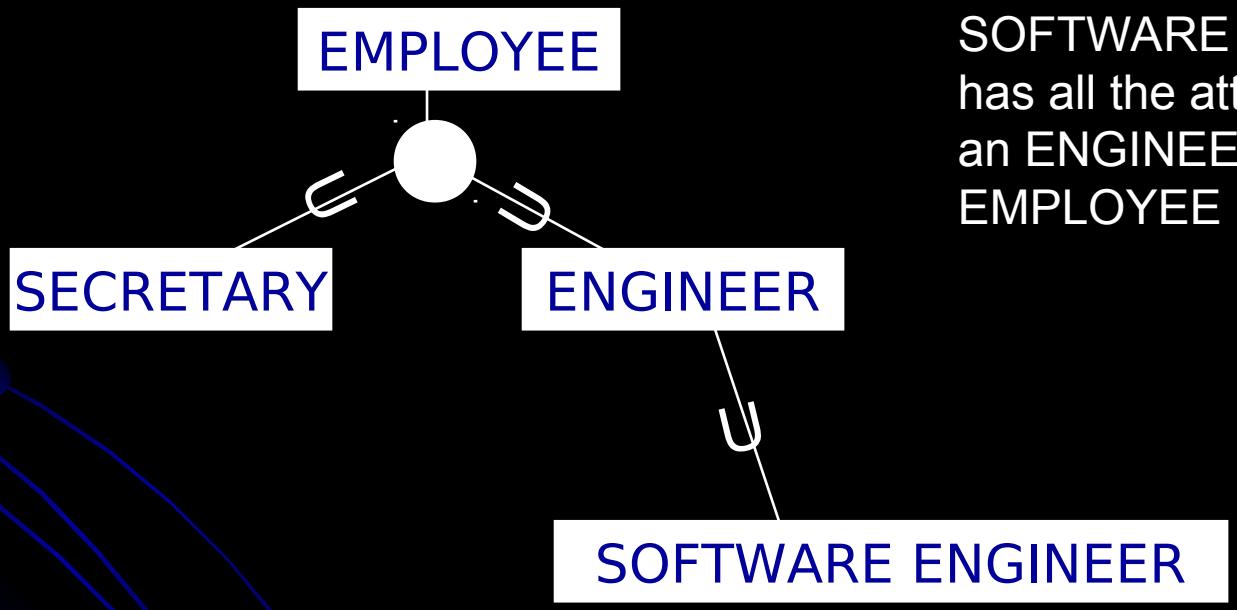
Generalization

Reverse processes of defining subclasses (bottom-up approach)

- Bring together common attributes in entities
- Example: CAR (with attributes color, price, max speed) and TRUCK (with attributes color, price, tonnage) can be generalized into VEHICLE (with attributes color and price).

Hierarchies and Lattices

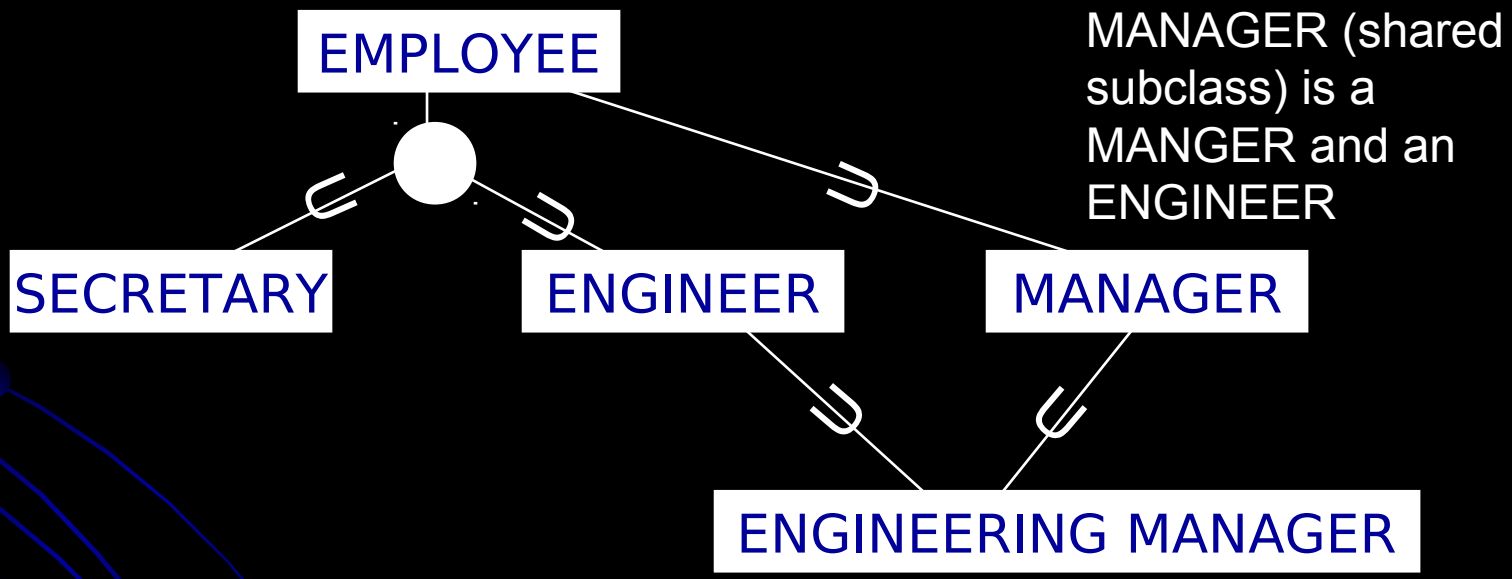
- Hierarchy – subclass participates in one class/subclass relationship



SOFTWARE ENGINEER has all the attributes of an ENGINEER and EMPLOYEE

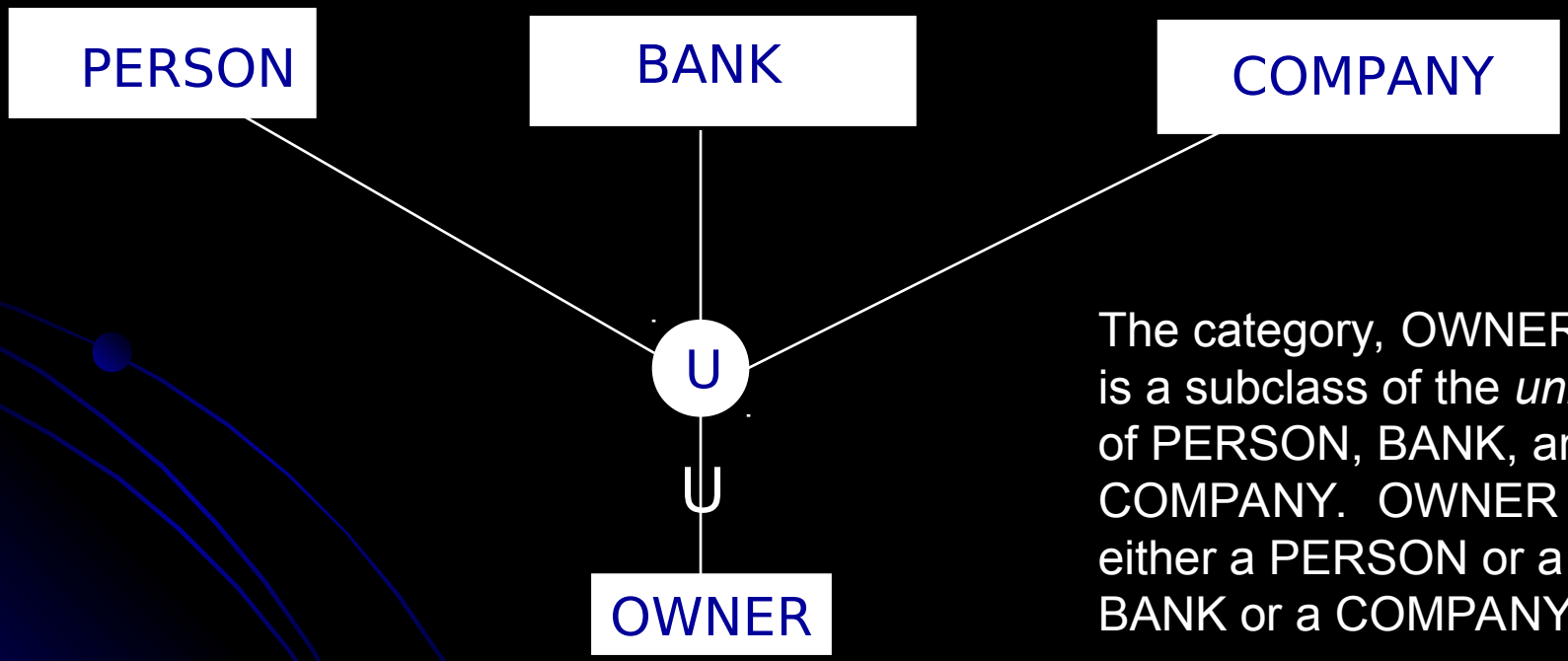
Hierarchies and Lattices

- Lattice – subclass participates in more than one class/subclass relationship



Categories

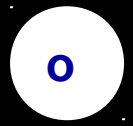
- Models a class/subclass with more than one superclass of *distinct* entity types. Attribute inheritance is selective.



The category, OWNER, is a subclass of the *union* of PERSON, BANK, and COMPANY. OWNER is either a PERSON or a BANK or a COMPANY

Constraints

- Disjoint – an entity can be a member of at most one subclass of a specialization 

- Overlap – an entity may belong to more than one subclass of a specialization 

- Total specialization – each entity of a superclass belongs to some subclass of a specialization 
- Partial specialization – each entity of a superclass does not have to belong to some subclass of a specialization 

Putting It All Together

Example (Figure 4.7 in textbook)

Specialization / Generalization Lattice Example (UNIVERSITY)

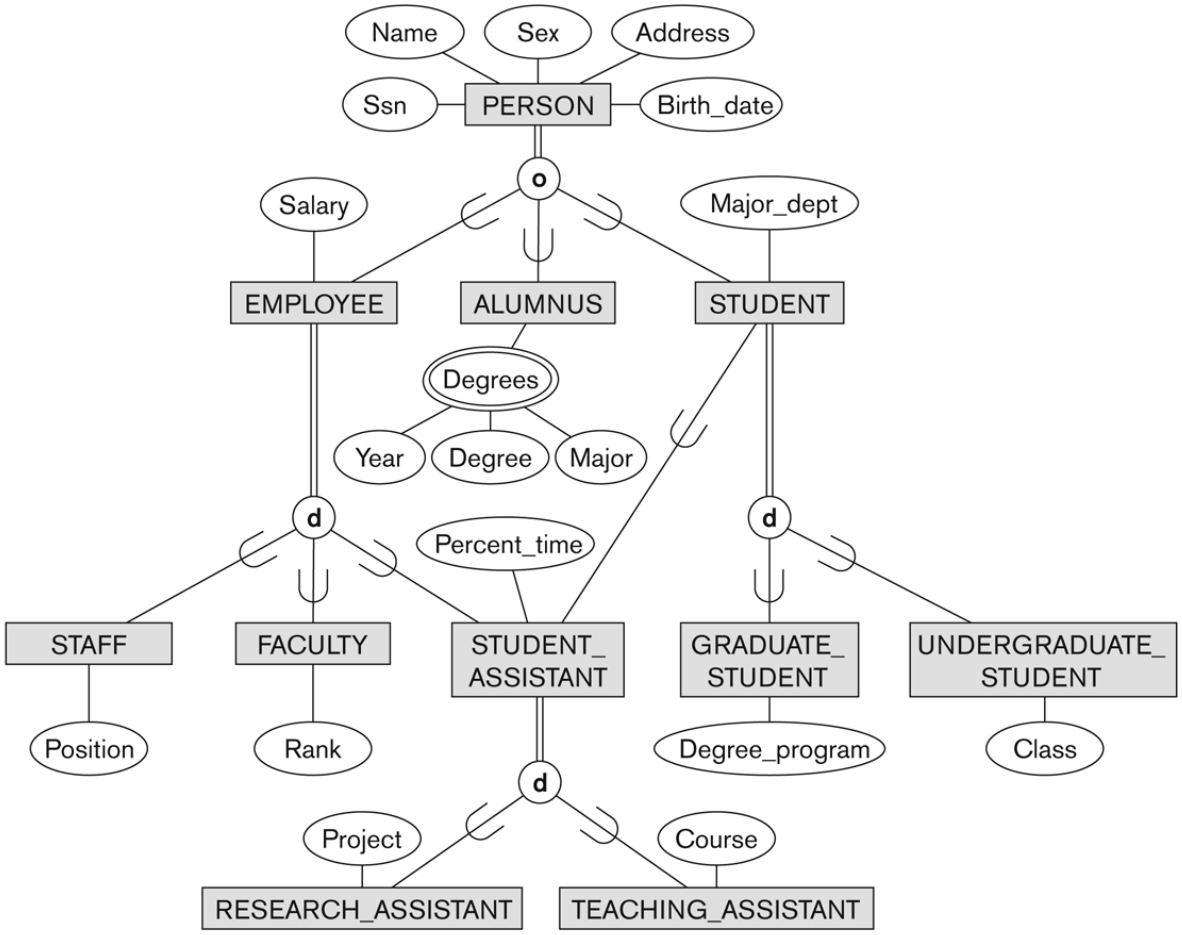


Figure 4.7
A specialization lattice with multiple inheritance for a UNIVERSITY database.

Copyright © 2007 Ramez Elmasri and Shamkant B. Navathe

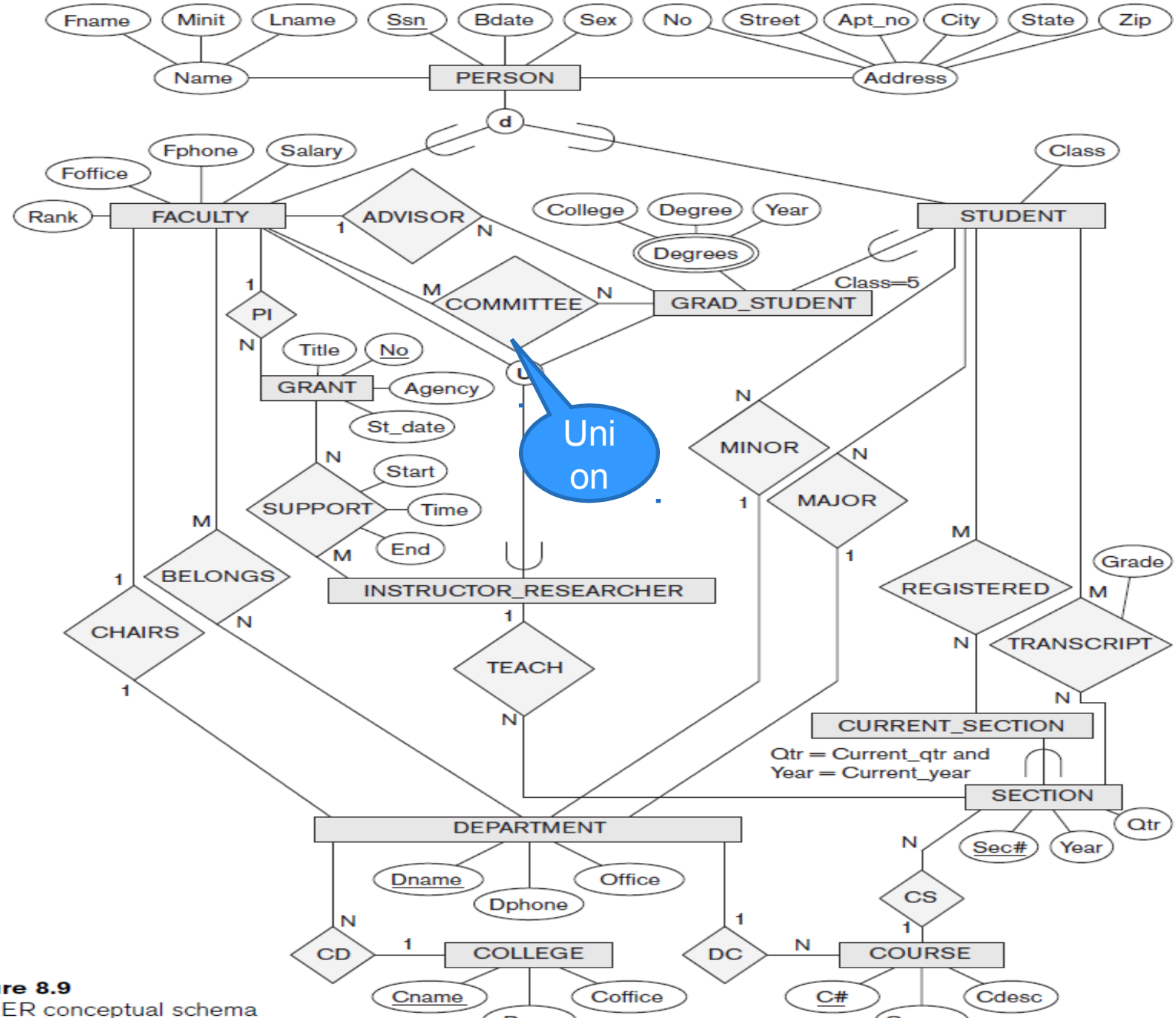


Figure 8.9
An EER conceptual schema

Topics Covered in Todays Class

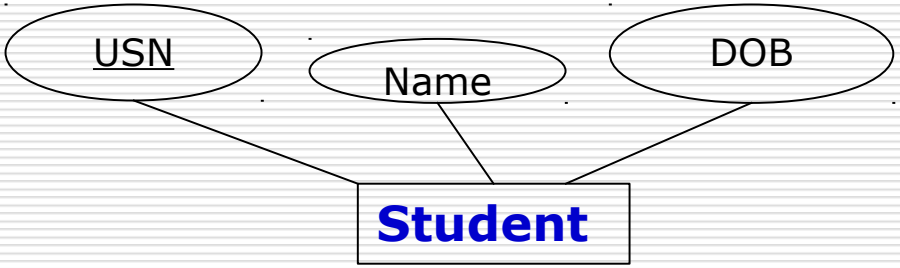
Unit 2:

-Converting ER Diagrams to Tables
or Database design using ER-to-Relational Mapping

- An ER diagram is a pictorial representation of the information that can be captured by a database. Such a "picture" serves two purposes:
  - It allows database professionals to describe an overall design concisely yet accurately.
  - (Most of) it can be easily transformed into the relational schema.
- The ER diagram represents the conceptual level of database design meanwhile the relational schema (or database table) is the logical level for the database design.

Converting ER diagram to Tables

- Entities and Simple Attributes
  - An entity type within ER diagram is turned into a table.
  - Each attribute turns into a column (attribute) in the table. The key attribute of the entity is the **primary key** of the table which is usually underlined.



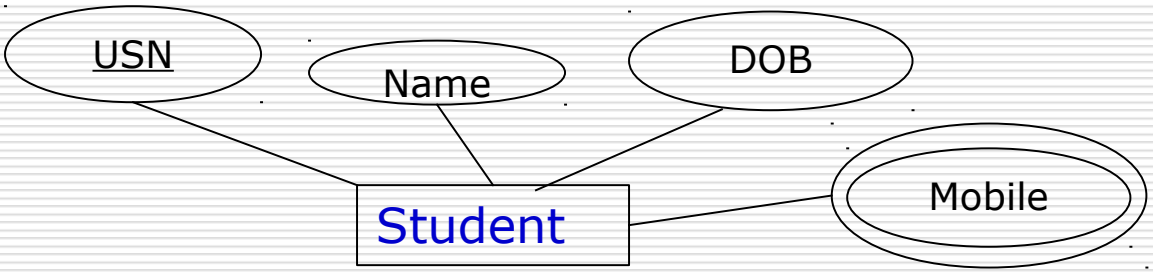
Student Table

Primary Key →

USN	**Name**	**DOB**
1BM14CS001	Aditya	1-1-1997
1BM14CS002	Bharath	31-12-1996

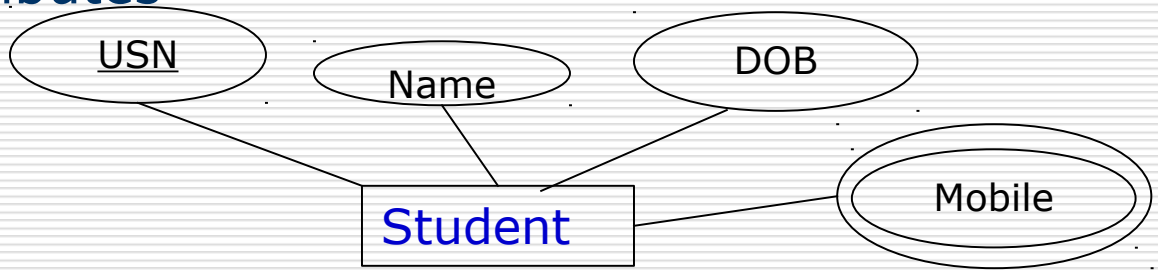
Converting ER diagram to Tables

- Multi-Valued Attributes



Converting ER diagram to Tables

- Multi-Valued Attributes



Which of the following representation of the table for Multivalued Attribute is best ?

USN	Name	DOB	Mobile 1	Mobile 2
1BM14CS001	Aditya	1-1-1997	8766655433	
1BM14CS002	Bharath	31-12-1996	9762255433	7066722433

Student Table

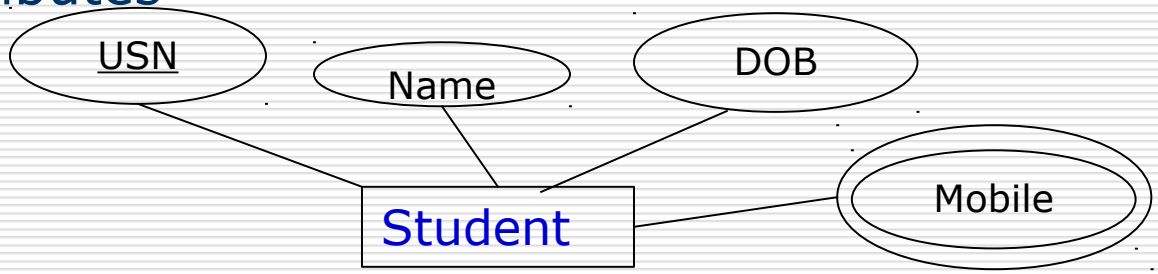
USN	Name	DOB
1BM14CS001	Aditya	1-1-1997
1BM14CS002	Bharath	31-12-1996

Mobile Table

USN	Mobile
1BM14CS001	8766655433
1BM14CS002	9762255433
1BM14CS002	7066722433

Converting ER diagram to Tables

- Multi-Valued Attributes



Primary Key

Student Table

USN	Name	DOB
1BM14CS001	Aditya	1-1-1997
1BM14CS002	Bharath	31-12-1996

Foreign Key

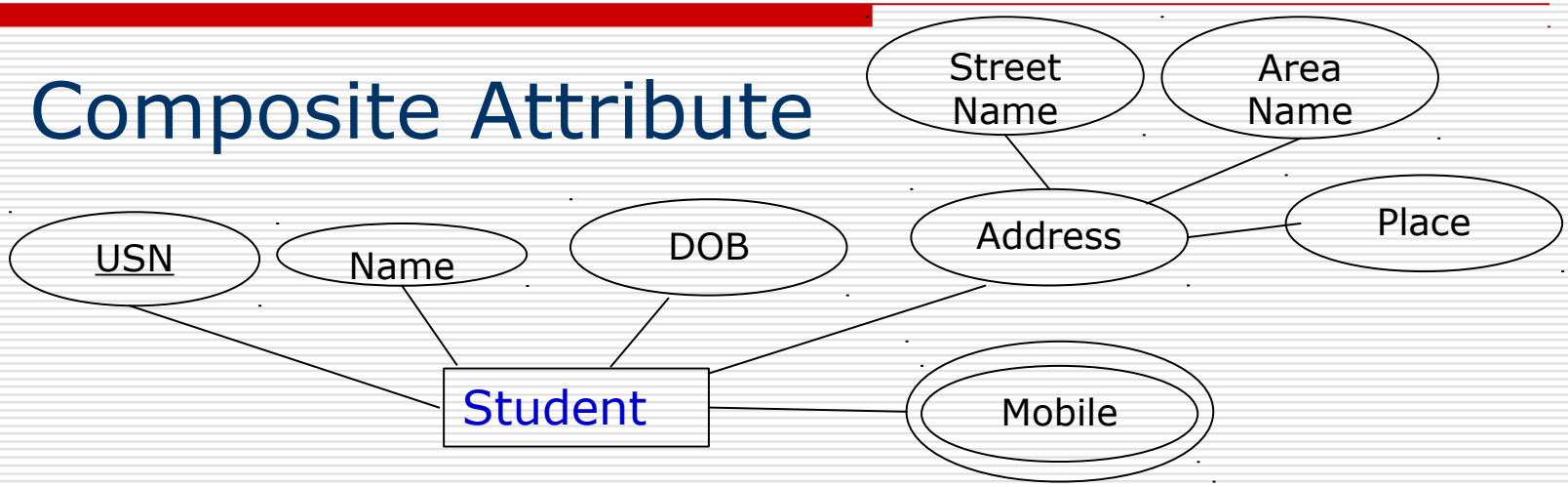
Mobile Table

USN	Mobile
1BM14CS001	8766655433
1BM14CS002	9762255433
1BM14CS002	7066722433

If you have a multi-valued attribute, take the attribute and turn it into a new entity or table of its own. Then make a 1:N relationship between the new entity and the existing one. In simple words,
1.Create a table for the attribute.
2.Add the primary (id) column of the parent entity as a foreign key within the new table

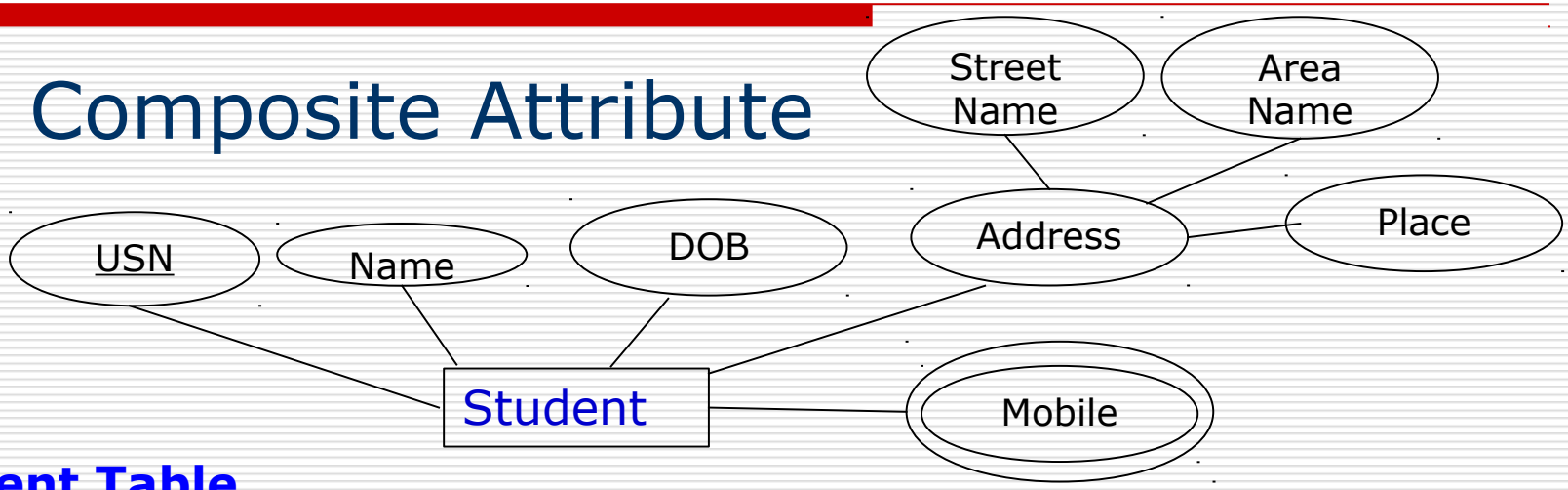
Converting ER diagram to Tables

- Composite Attribute



Converting ER diagram to Tables

Composite Attribute



Student Table

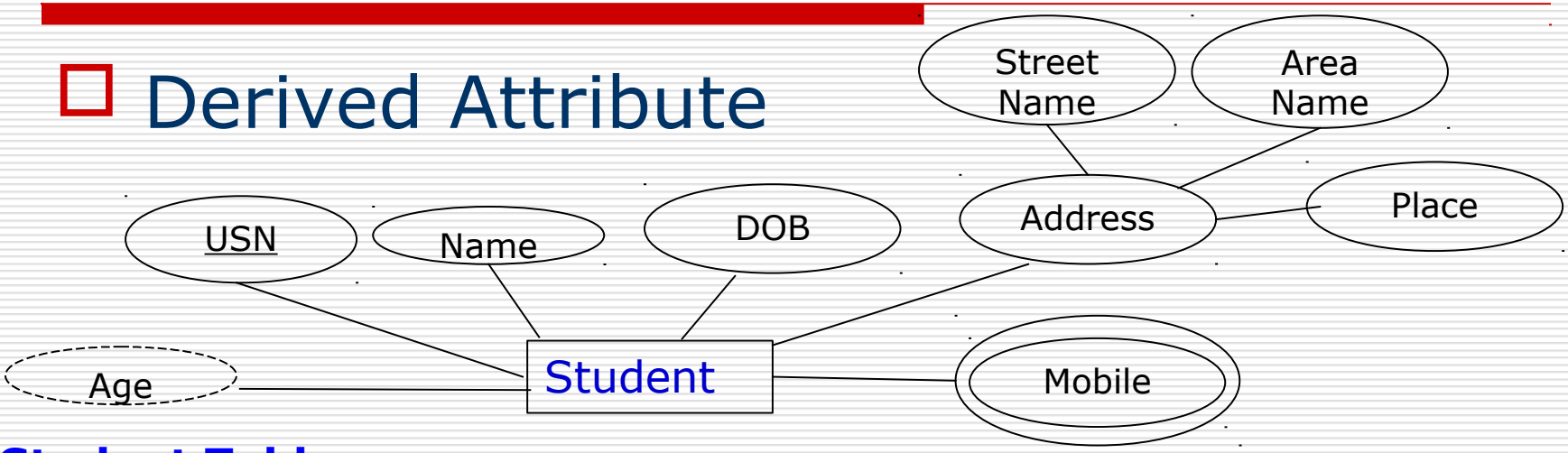
USN	Name	DOB	Street	Area	Place
1BM14CS001	Aditya	1-1-1997	RK Road	Nagar	Mandya
1BM14CS002	Bharath	31-7-1996	S V Road	Layout	Kolar

Mobile Table

USN	Mobile
1BM14CS001	8766655433
1BM14CS002	9762255433
1BM14CS002	7066722433

Converting ER diagram to Tables

Derived Attribute



Student Table

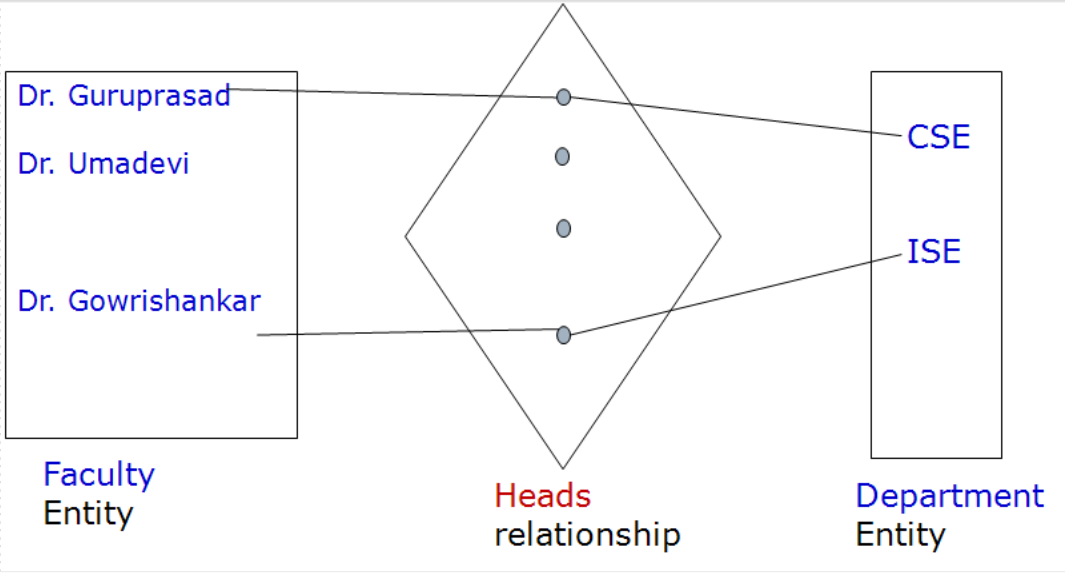
USN	Name	DOB	Street	Area	Place
1BM14CS001	Aditya	1-1-1997	RK Road	Nagar	Mandya
1BM14CS002	Bharath	31-7-1996	S V Road	Layout	Kolar

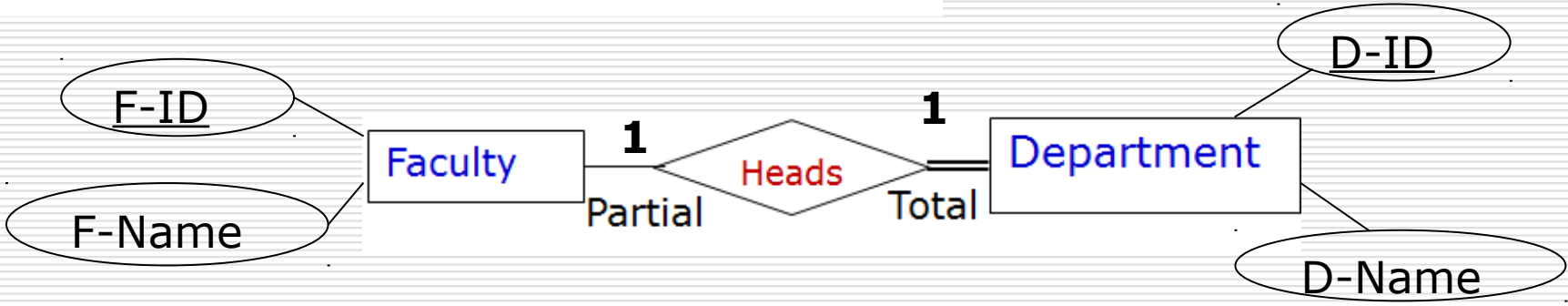
Mobile Table

USN	Mobile
1BM14CS001	8766655433
1BM14CS002	9762255433
1BM14CS002	7066722433

Converting ER diagram to Tables

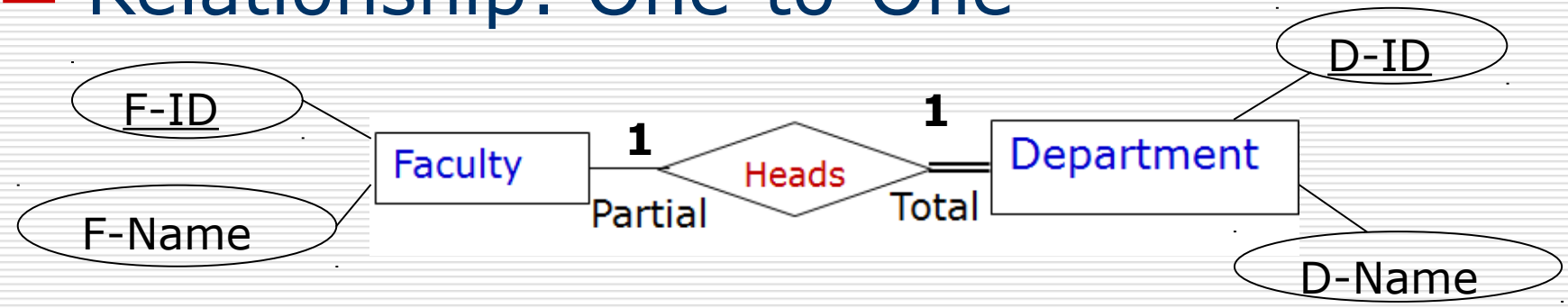
Relationship: One-to-One





Converting ER diagram to Tables

Relationship: One-to-One



Approach 1: Foreign Key

Faculty Table

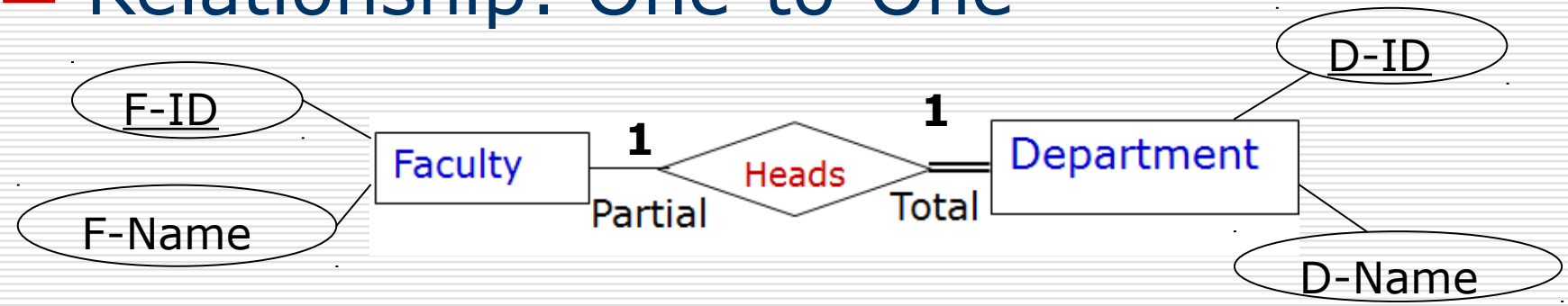
F-ID	**F-Name**
1	Dr. H S Guruprasad
2	Dr. Umadevi V
3	Dr. Gowrishankar

Department Table

D-ID	**D-Name**	**Heading F-ID**
10	CSE	1
20	ISE	3

Converting ER diagram to Tables

Relationship: One-to-One



Approach 1: Foreign Key

Primary Key

Faculty Table

F-ID	**F-Name**
1	Dr. H S Guruprasad
2	Dr. Umadevi V
3	Dr. Gowrishankar

Department Table

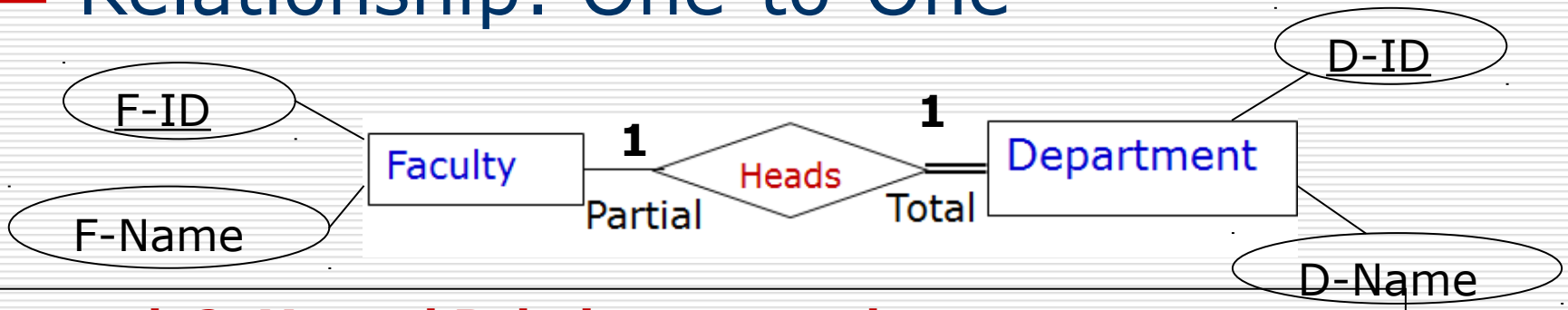
Primary Key

Foreign Key

D-ID	**D-Name**	**Heading F-ID**
10	CSE	1
20	ISE	3

Converting ER diagram to Tables

Relationship: One-to-One



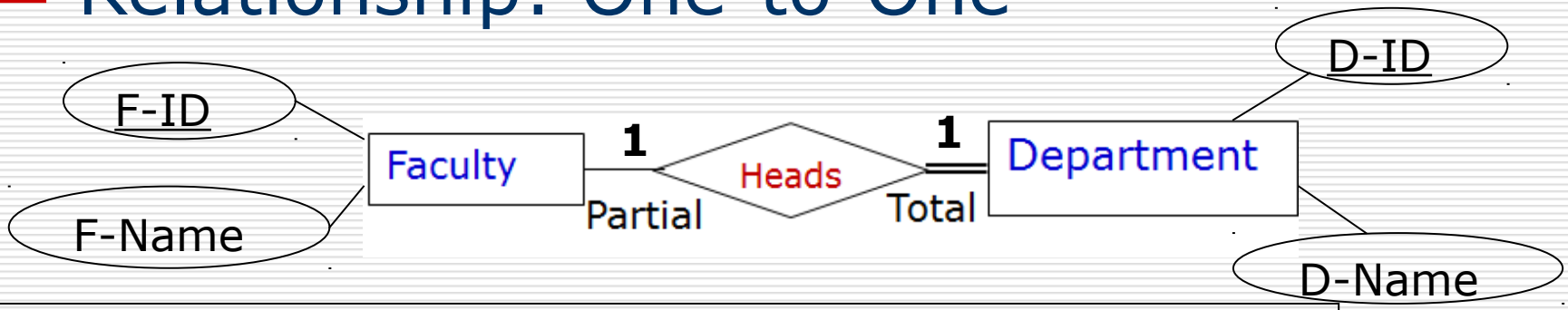
Approach 2: Merged Relation approach
Merging two entity types and relationship into one single relation.
This may be appropriate when **both participations are total**

Faculty –Department Table

F-ID	F-Name	D-ID	D-Name	HOD
1	Dr. H S Guruprasad	10	CSE	Yes
2	Dr. Umadevi V	10	CSE	No
3	Dr. Gowrishankar	20	ISE	Yes

Converting ER diagram to Tables

Relationship: One-to-One



Approach 3: Cross-reference or relationship relation approach

Faculty Table

F-ID	F-Name
1	Dr. H S Guruprasad
2	Dr. Umadevi V
3	Dr. Gowrishankar

Department Table

D-ID	D-Name
10	CSE
20	ISE

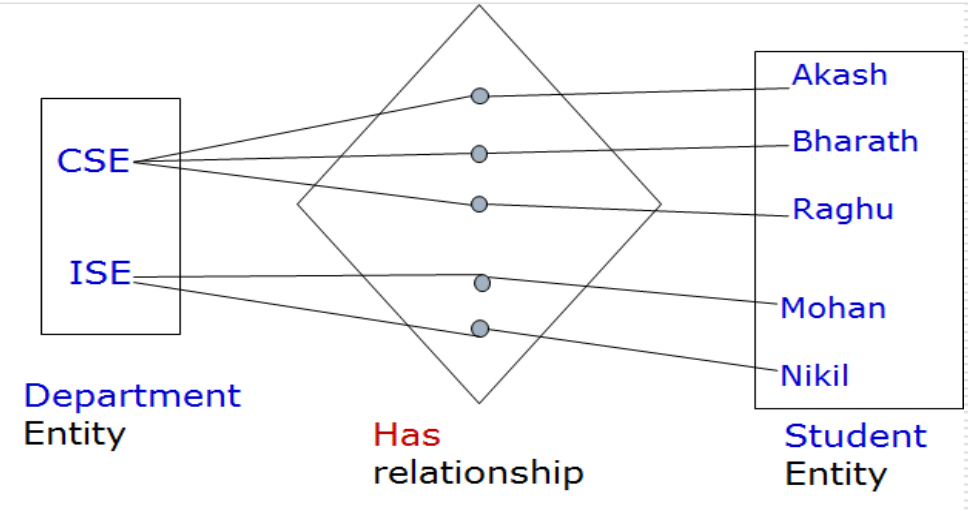
HOD Table

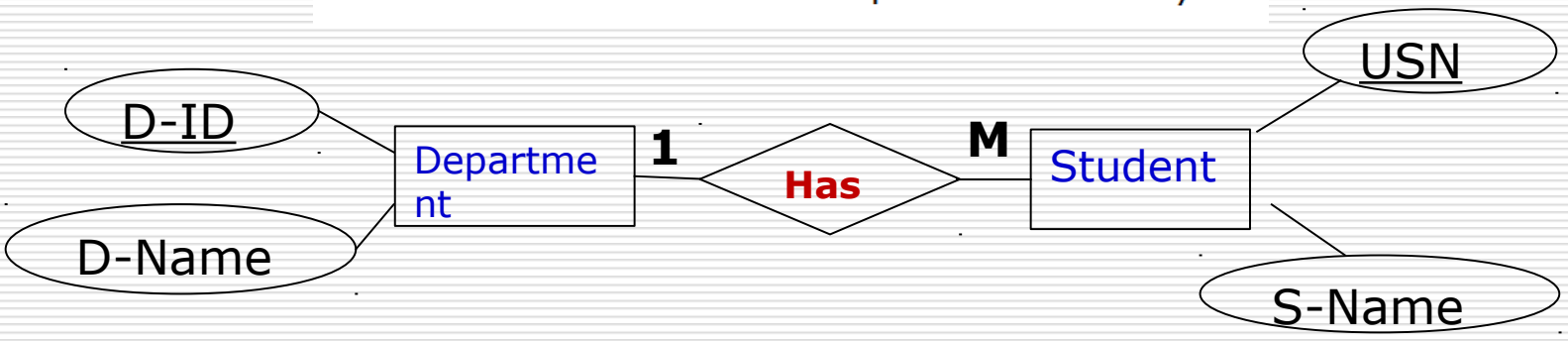
D-ID	F-ID
10	1
20	3

Lookup Table

Converting ER diagram to Tables

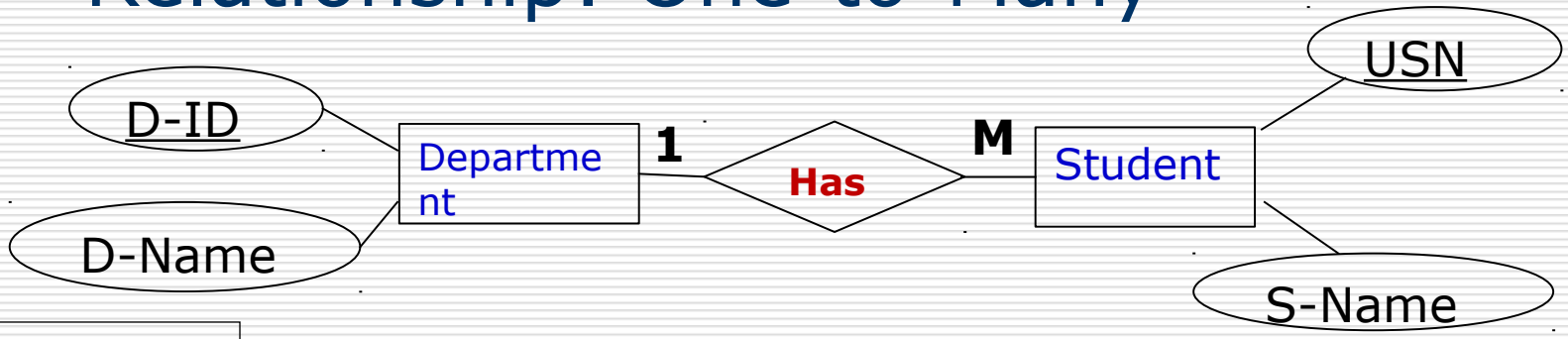
- Relationship: One-to-Many





Converting ER diagram to Tables

Relationship: One-to-Many



Approach 1

Department Table

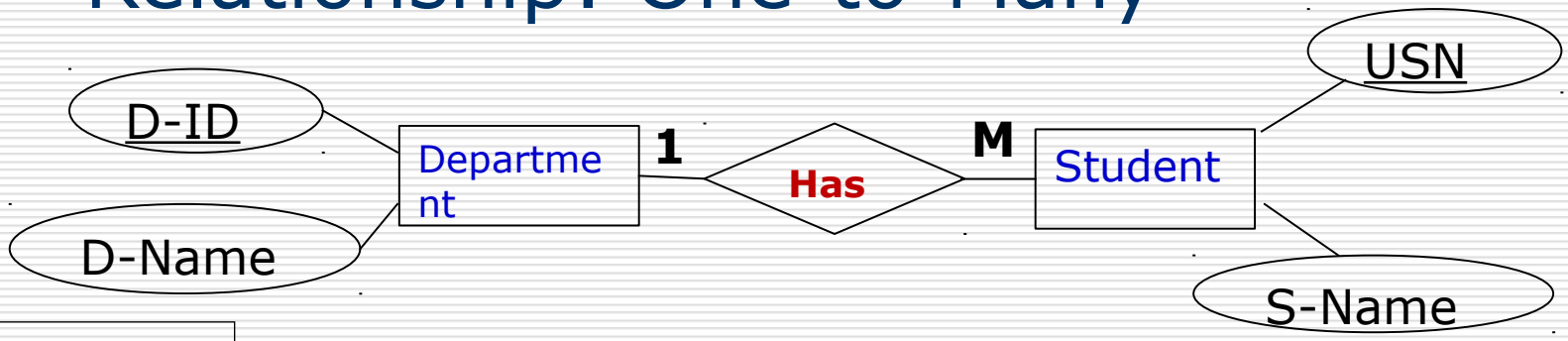
D-ID	D-Name
10	CSE
20	ISE

Student Table

USN	S-Name	D-ID
1BM14CS001	Akash	10
1BM14CS002	Bharath	10
1BM14CS003	Ragu	10
1BM14CS004	Mohan	20
1BM14CS005	Nikil	20

Converting ER diagram to Tables

Relationship: One-to-Many



Approach 2

Department Table

D-ID	D-Name
10	CSE
20	ISE

Student Table

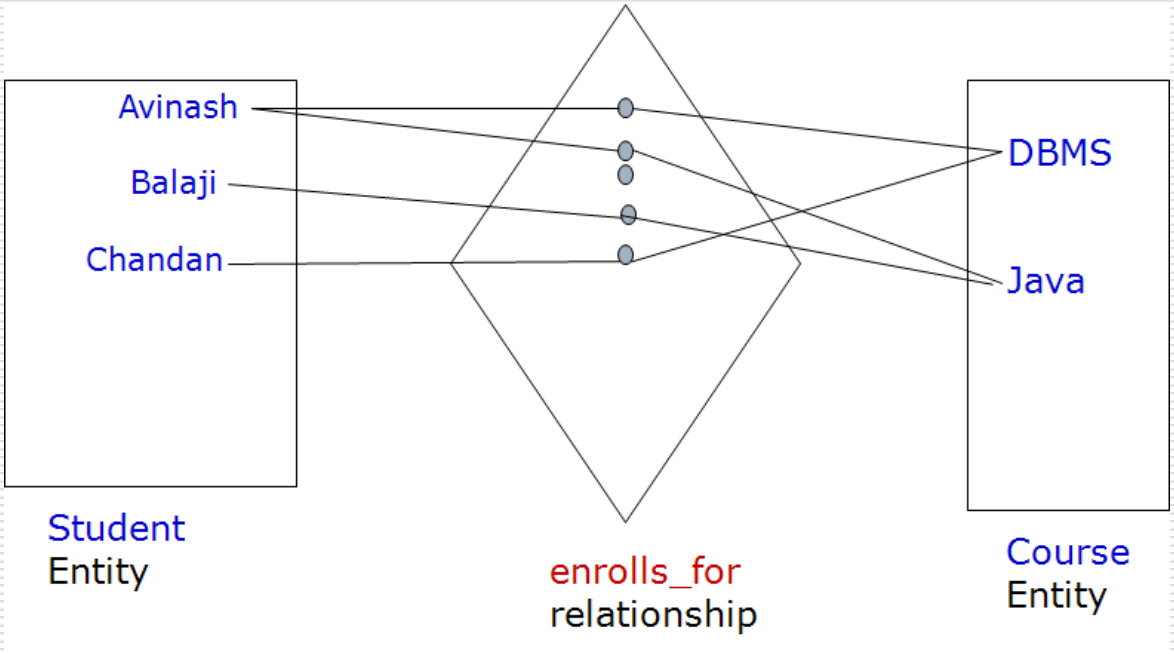
USN	S-Name
1BM14CS001	Akash
1BM14CS002	Bharath
1BM14CS003	Ragu
1BM14CS004	Mohan
1BM14CS005	Nikil

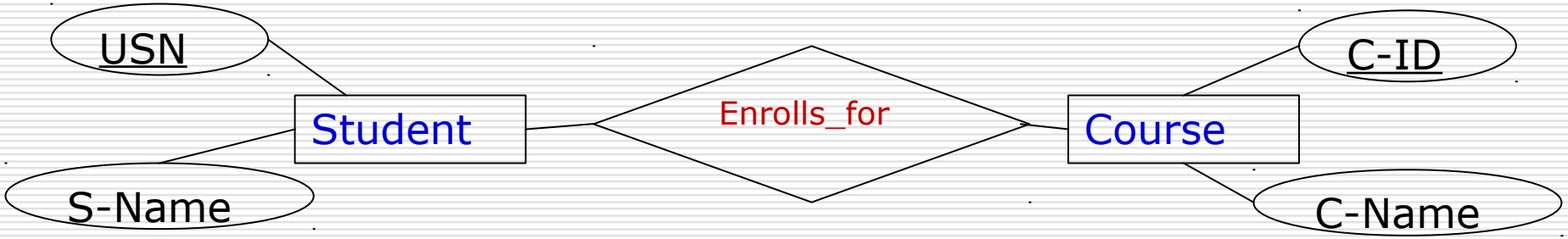
Department Student Table

USN	D-ID
1BM14CS001	10
1BM14CS002	10
1BM14CS003	10
1BM14CS004	20
1BM14CS005	20

Converting ER diagram to Tables

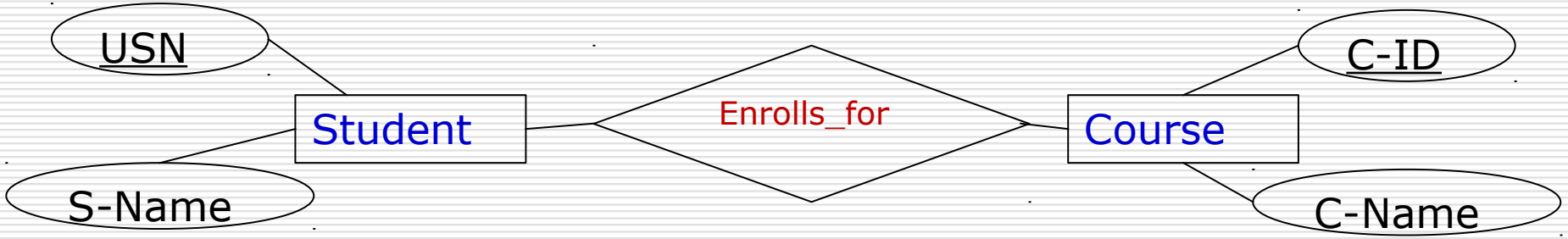
- Relationship: Many-to-Many





Converting ER diagram to Tables

Relationship: Many-to-Many



Student Table

USN	**S-Name**
1BM14CS001	Avinash
1BM14CS002	Balaji
1BM14CS003	Chandan

Course Table

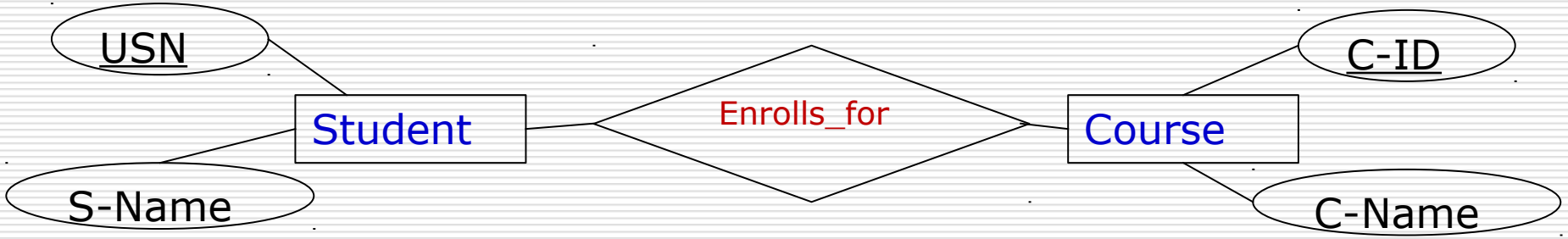
C-ID	**C-Name**
10	DBMS
20	Java

Student Course Table

USN	**C-ID**
1BM14CS001	10
1BM14CS001	20
1BM14CS002	20
1BM14CS003	10

Converting ER diagram to Tables

Relationship: Many-to-Many



Student Table

USN	**S-Name**
1BM14CS001	Avinash
1BM14CS002	Balaji
1BM14CS003	Chandan

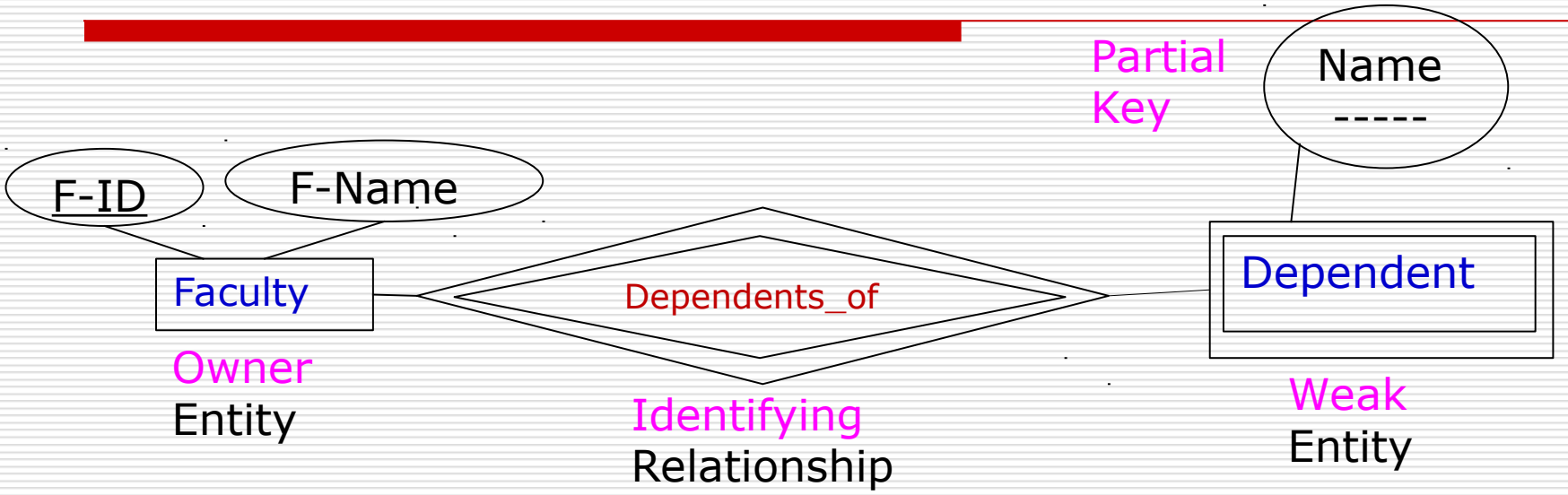
Course Table

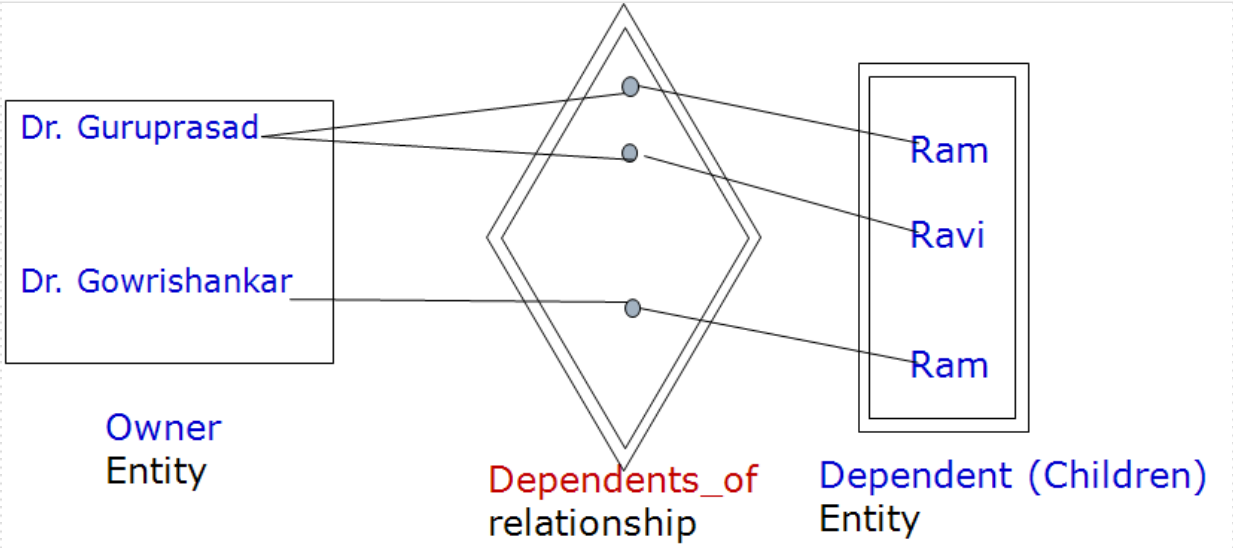
C-ID	**C-Name**
10	DBMS
20	Java

Student Course Table

USN	**C-ID**
1BM14CS001	10
1BM14CS001	20
1BM14CS002	20
1BM14CS003	10

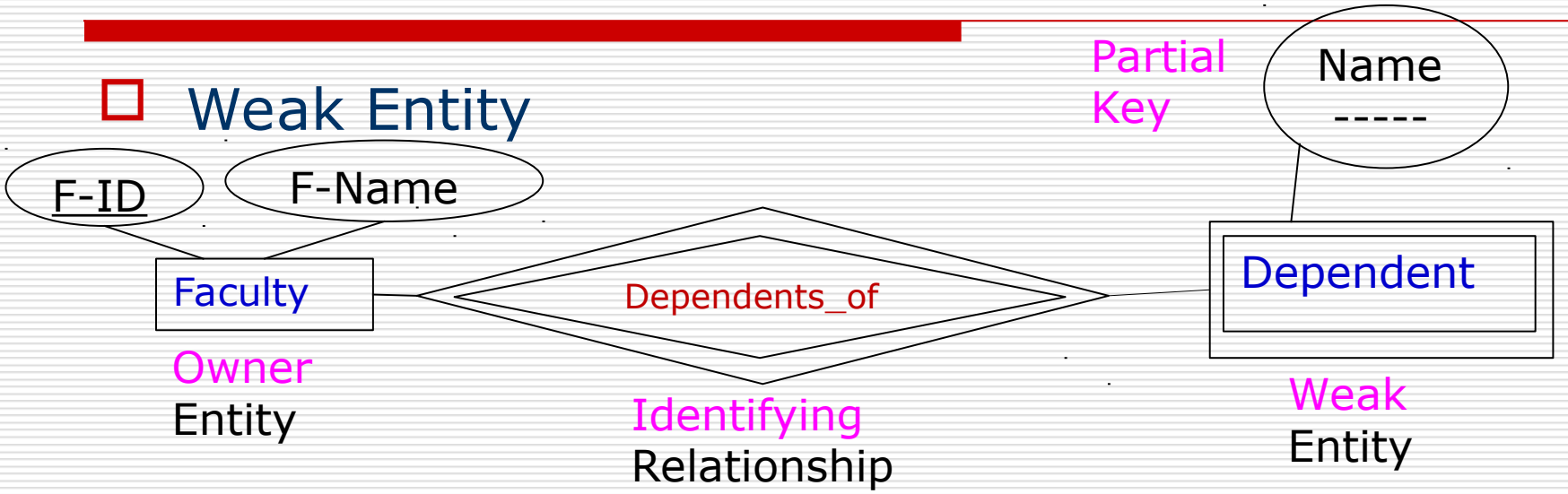
Converting ER diagram to Tables





Converting ER diagram to Tables

Weak Entity



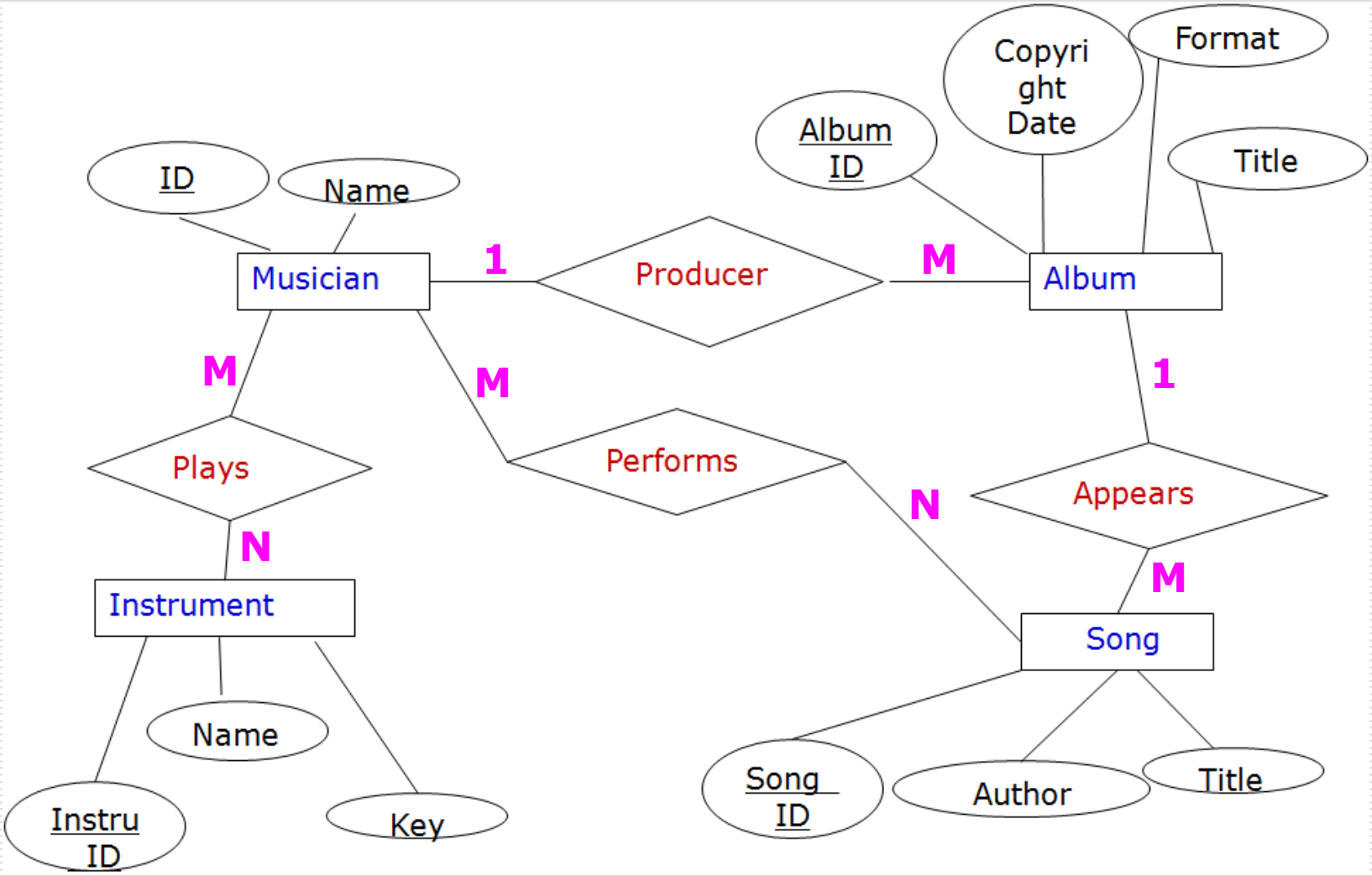
Faculty Table

F-ID	**F-Name**
10	Dr. Guruprasad
20	Dr. Gowrishankar

Dependent Table

F-ID	**Dependent Name**
10	Ram
10	Ravi
20	Ram

Activity To Do
Convert the Following ER diagram to Database table schema



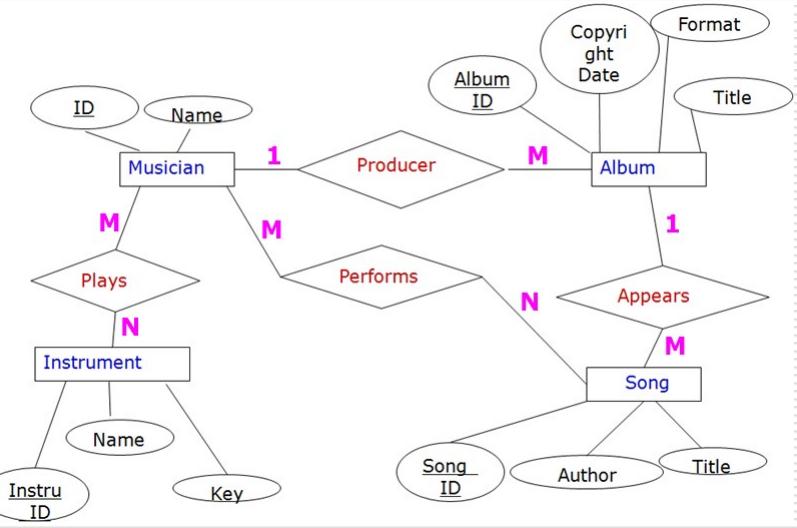
Database tables or Schema Diagram

Musician Table

Name	Muscian-ID

Album Table

Album-ID	Copyright Date	Format	Title	Producer ID



Song Table

Song-ID	Author	Title	Album-ID

Performing Table

Muscian-ID	Song-ID

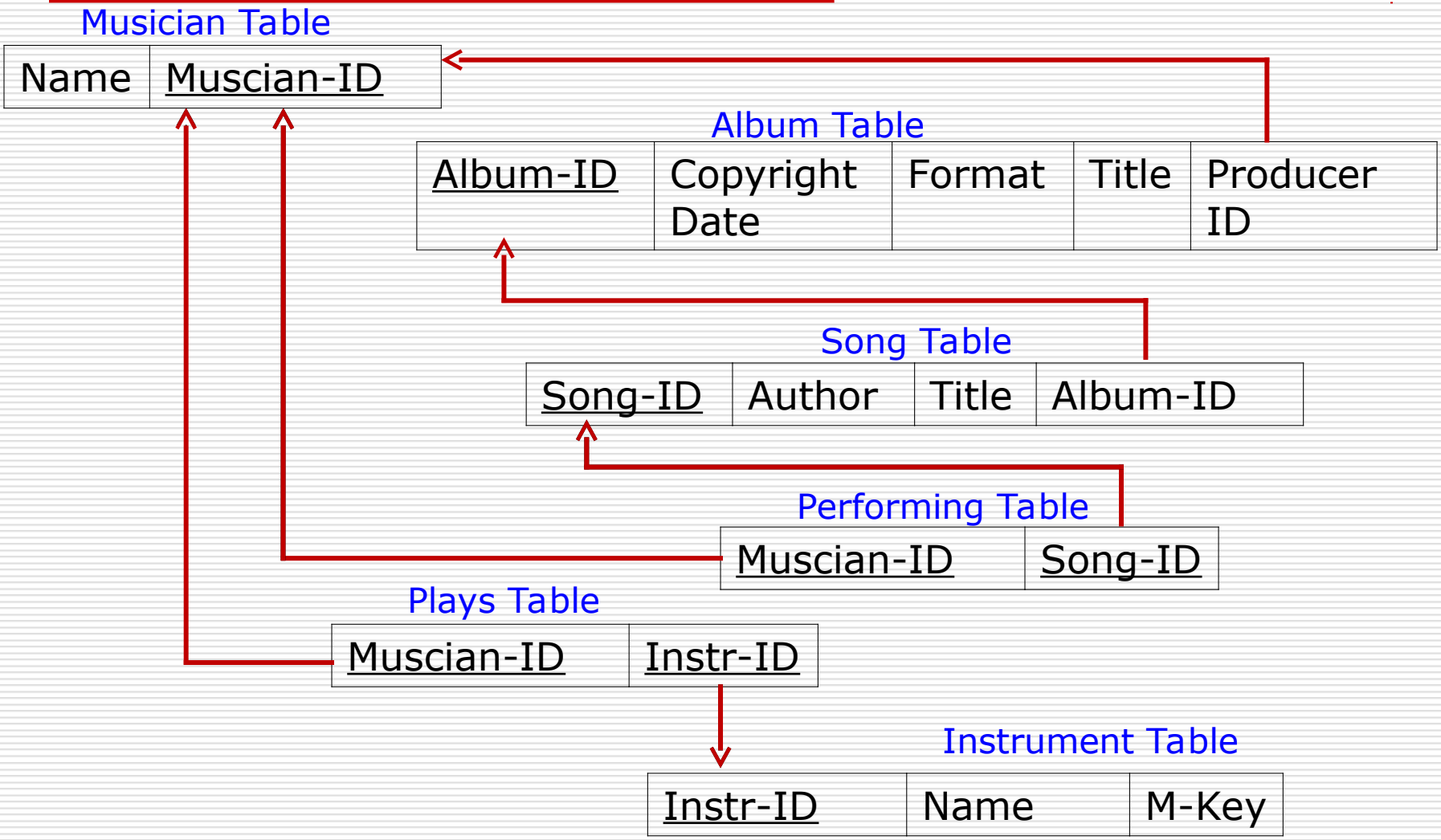
Plays Table

Muscian-ID	Instr-ID

Instrument Table

Instr-ID	Name	M-Key

Database tables or Schema Diagram



Thanks for Listening

Cardinality Ratios

- Cardinality is a constraint on a relationship specifying the number of entity instances that a specific entity may be related to via the relationship.

Many to One:

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