CHAPTER-1

Database Management System Introduction

Introduction

1. **Data** - Fact that can be recorded or stored

   e.g. Person Name, Age, Gender and Weight etc.

2. **Information**

   When data is **processed,** organized, structured or presented in a given context so as to make it useful, it is called information.

3. **knowledge**

   It is the appropriate collection of information

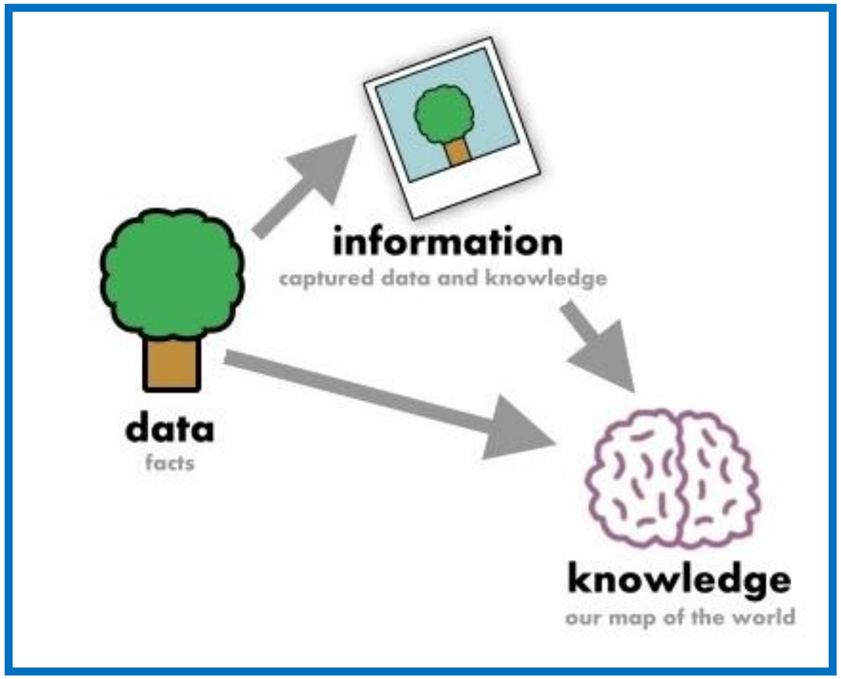


Figure: 1.1 An example of Data,information and knowledge

(**Image Source :** https://images.app.goo.gl/cbtcdV1EMn3SoWzi7)

What is DBMS?

- ***DBMS (Database Management System***= Database + Set of programs

- ***Database:-***A Collection of **logically related data.**
  - e.g. Books Database in Library, Student Database in University etc.
- **Management** - Manipulation, Searching and Security of data
  - e.g. Viewing result in GTU website, Searching exam papers in GTU website etc.
- **System** - **Programs** or **tools** used to manage database
  - e.g. SQL Server Studio Express, Oracle etc.
- **DBMS -** Database Management System (DBMS) is a **software designed to define, manipulate, retrieve and manage data in a database**.
  - e.g. MS SQL Server, Oracle, My SQL, SQLite, MongoDB etc.

Examples of DBMS

- Online Telephone Directory
- Electricity Service provider
- Facebook
- Whatsapp
- etc

Figure: 1.2 Online Telephone Directory

(Image Source : https://images.app.goo.gl/cbtcdV1EMn3SoWzi7)

Figure:1.5 Whatsapp

(Image Source

https://images.app.goo.gl/cbtcdV1EMn3SoWzi7)

Figure: 1.4 Facebook

(Image Source :

ttps://images.app.goo.gl/cbtcdV1EMn3SoWzi7)

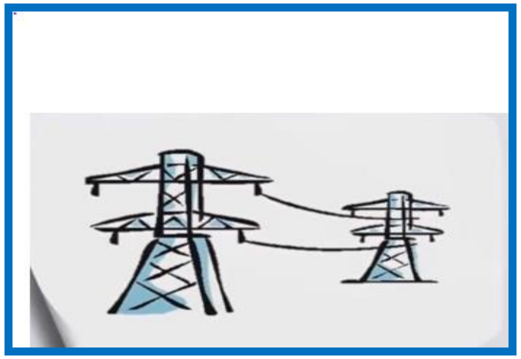


Figure: 1.3 Electricity Service provider

(**Image Source** : https://images.app.goo.gl/cbtcdV1EMn3SoWzi7)

Database System Applications

- DBMS is a computerized record-keeping system.
- DBMS is required where ever data need to be stored.
  - E-Commerce (**Flikart, Amazon, Shopclues, eBay** etc...)
  - Online Television Streaming (**Hotstar, Amazon Prime** etc...)
  - Social Media (**WhatsApp, Facebook, Twitter, LinkedIn** etc...)
  - Banking & Insurance
  - Airline & Railway
  - Universities and Colleges/Schools
  - Library Management System
  - Human Resource Department
  - Hospitals and Medical Stores
  - Government Organizations

File Processing System (FPS) or File System

- In Computer Science, File Processing System (FPS) is a way of storing, retrieving and manipulating data which is present in various files.

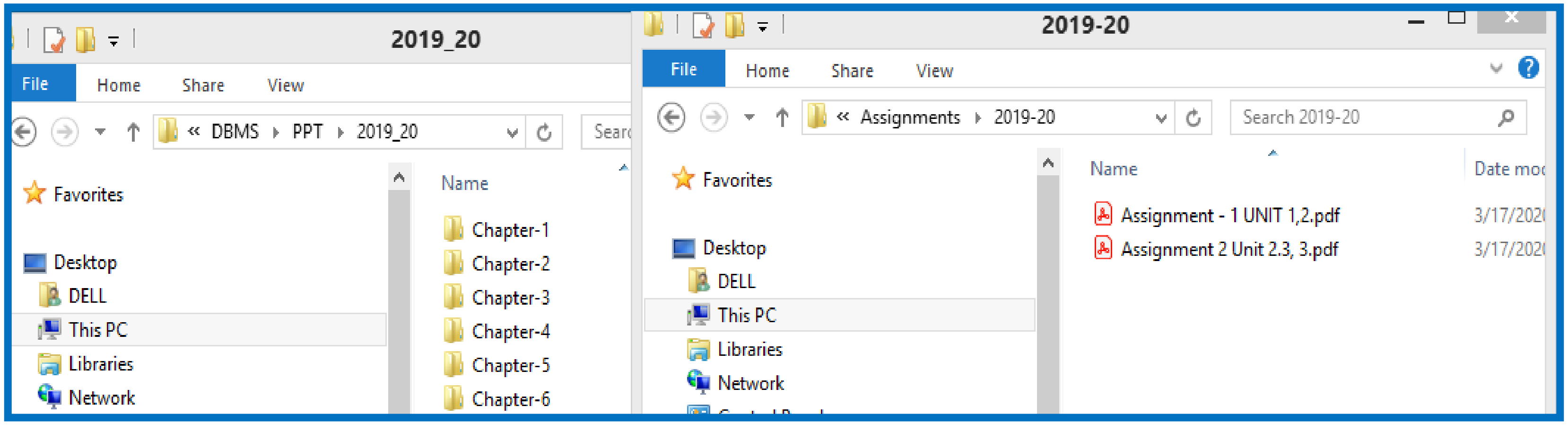


Figure: 1.6 An example of File Processing System

Advantages of File Processing System

- **Cost friendly –**
  There is a very minimal to no set up and usage fee for File Processing System. (In most cases, free tools are inbuilt in computers.)
- **Easy to use –**
  File systems require very basic learning and understanding, hence, can be easily used.
- **High scalability –**
  One can very easily switch from smaller to larger files as per his needs.

Disadvantages of File Processing System

- **Data redundancy and inconsistency**
  - Multiple file formats, duplication of information in different files
- **Difficulty in accessing data**
  - Need to write a new program to carry out each new task
- **Data isolation-** data are scattered in various files, and files may be in different formats, writing new application programs to retrieve the appropriate data is difficult.
- **Integrity problems-** when new constraints are added, it is difficult to change the programs to enforce them.

Disadvantages of File Processing System

- **Atomicity problems**
  - Failures may leave database in an inconsistent state with partial updates carried out
  - Example: Transfer of funds from one account to another should either complete or not happen at all
- **Concurrent-access by multiple users**
  - Example: Two people reading a balance (say 100) and updating it by withdrawing money (say 50 each) at the same time
- **Security problems**
- **Poor data control**
- **Limited data sharing**
- **Excessive programming effort**

Application of DBMS

- Providing Application Flexibility with Relational Databases.
- Object oriented Applications and the need for more complex Databases.
- Early DB Applications
- Extending DB capabilities for new applications

Advantages of DBMS over file management system.

- Minimal data redundancy
- Program data independence
- Efficient data access
- Improved data sharing
- Improved security
- Economy of scale
- Reduced program maintenance
- Improved Backup
- Improved data quality

Purpose of DBMS

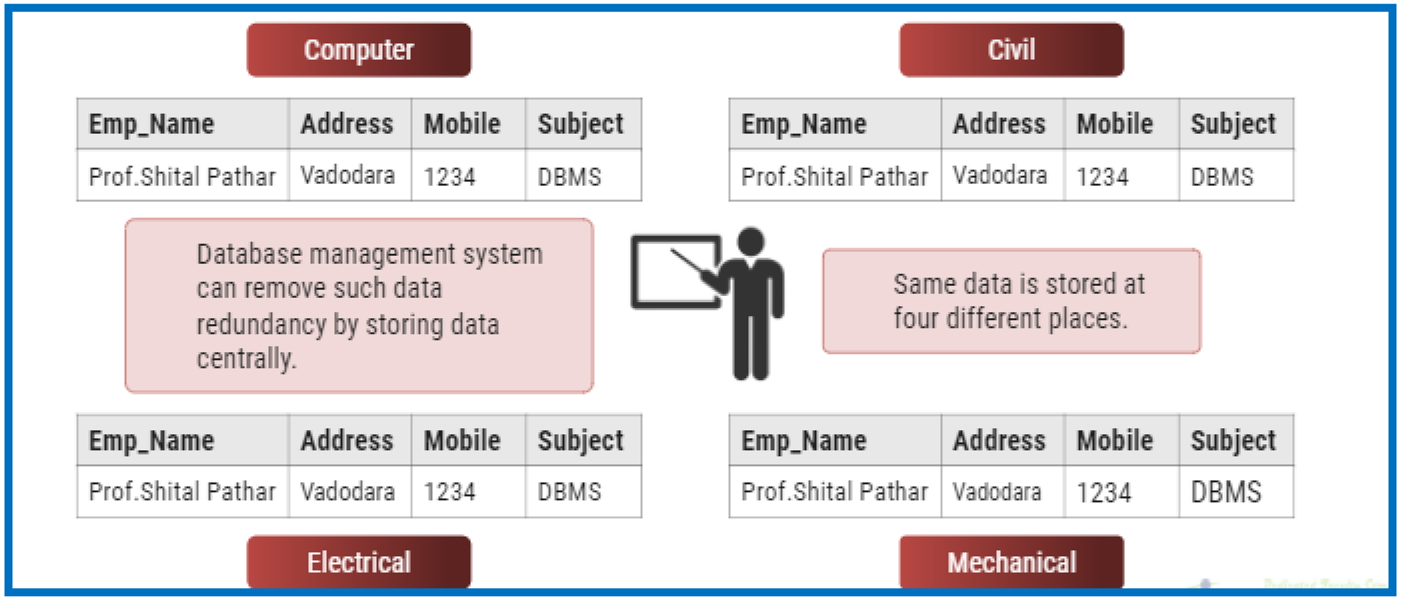
- Compactness- no need of paper work
- Speed
- Accuracy
- Protection

Benefits of DB Approach

- Data can be Shared
- Redundancy can be reduced
- Inconsistency can be avoided
- Security can be enforced
- Conflicting requirements can be balanced
- Integrity can be maintain

Advantages of DBMS

Reduce data redundancy (duplication)



Computer

Emp_Name	Address	Mobile	Subject
Prof.Shital Pathar	Vadodara	1234	DBMS

Civil

Emp_Name	Address	Mobile	Subject
Prof.Shital Pathar	Vadodara	1234	DBMS

Database management system can remove such data redundancy by storing data centrally.

Same data is stored at four different places.

Emp_Name	Address	Mobile	Subject
Prof.Shital Pathar	Vadodara	1234	DBMS

Electrical

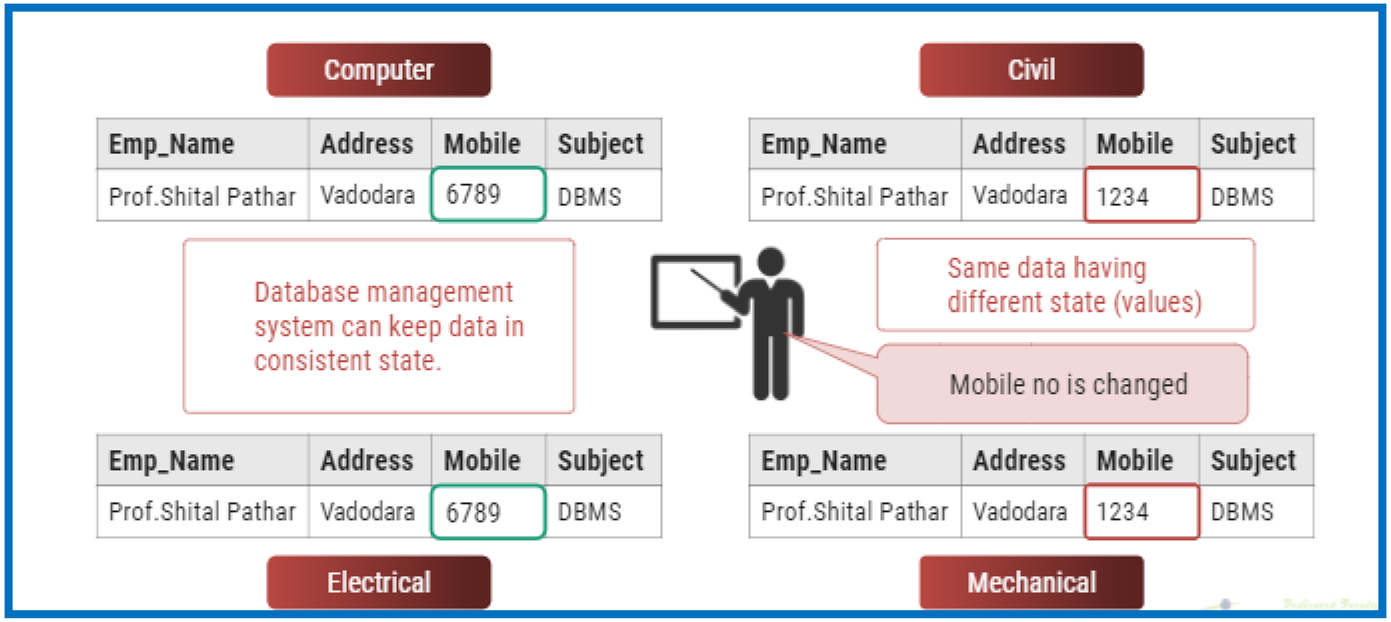
Emp_Name	Address	Mobile	Subject
Prof.Shital Pathar	Vadodara	1234	DBMS

Mechanical

Figure: 1.7 An example of data redundancy

(**Image Source :** https://www.researchgate.net/figure/An-example-of-redundancy-of-data-and-functionality_fig1_266550024)

Remove data inconsistency



Computer

Emp_Name	Address	Mobile	Subject
Prof.Shital Pathar	Vadodara	6789	DBMS

Civil

Emp_Name	Address	Mobile	Subject
Prof.Shital Pathar	Vadodara	1234	DBMS

Database management system can keep data in consistent state.

Same data having different state (values)

Mobile no is changed

Electrical

Emp_Name	Address	Mobile	Subject
Prof.Shital Pathar	Vadodara	6789	DBMS

Mechanical

Emp_Name	Address	Mobile	Subject
Prof.Shital Pathar	Vadodara	1234	DBMS

Figure: 1.8 An example of data inconsistency

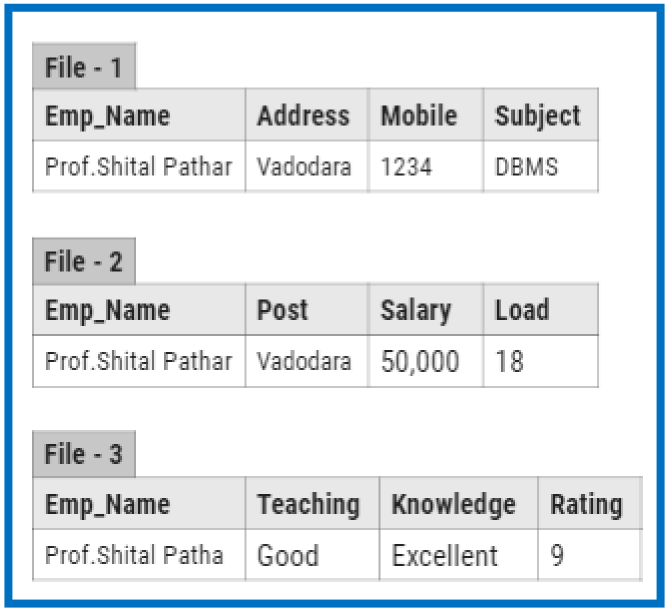
(Image Source : ttps://www.quora.com/What-is-data-inconsistency-1)

Data isolation

- Data are **scattered** in various files.
- Files may be in **different formats**.
- **Difficult to retrieve** the appropriate data.

DBMS allow us to access (retrieve) appropriate data easily.

Data isolation is a property that determines when and how changes made by one operation become visible to other concurrent users and systems.
This issue occurs in a concurrency situation.



File - 1

Emp_Name	Address	Mobile	Subject
Prof.Shital Pathar	Vadodara	1234	DBMS

File - 2

Emp_Name	Post	Salary	Load
Prof.Shital Pathar	Vadodara	50,000	18

File - 3

Emp_Name	Teaching	Knowledge	Rating
Prof.Shital Patha	Good	Excellent	9

Figure: 1.9 An example Data isolation
(Image Source : https://www.researchgate.net/figure/Example-of-data-isolation-problem_fig1_278658528)

Guaranteed atomicity

Atomicity: Either transaction **execute 0% or 100%**.

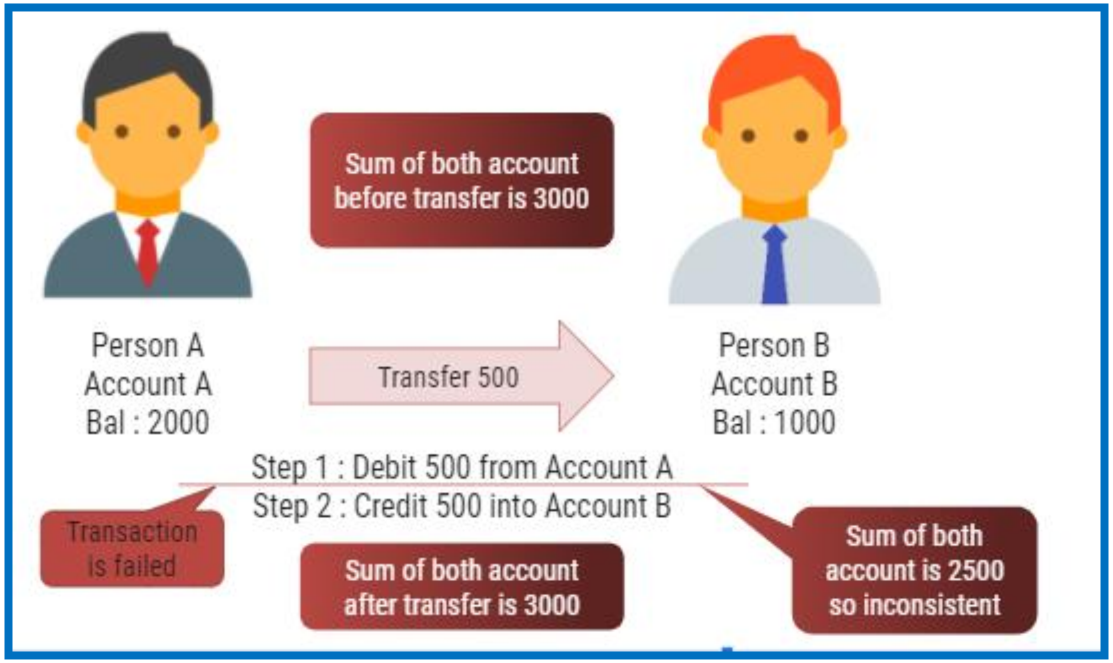
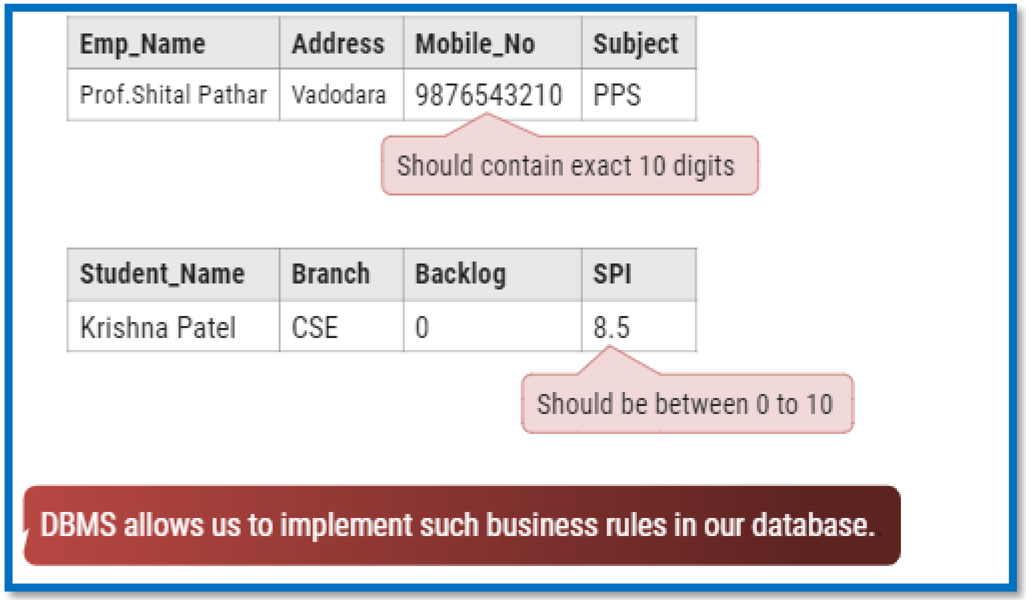

Sum of both account before transfer is 3000
Person A
Account A
Bal : 2000
Transfer 500
Person B
Account B
Bal : 1000
Step 1 : Debit 500 from Account A
Step 2 : Credit 500 into Account B
Transaction is failed
Sum of both account after transfer is 3000
Sum of both account is 2500 so inconsistent

Figure: 1.10 An example of atomicity

(Image Source : https://encrypted-tbn0.gstatic.com/images?q=tbn%3AANd9GcSVFDVbwXtsq94f6HGuntRnVsOaz27Xnw-Lew&usqp=CAU)

Allow to implement integrity constraints



Emp_Name	Address	Mobile_No	Subject
Prof.Shital Pathar	Vadodara	9876543210	PPS

Should contain exact 10 digits

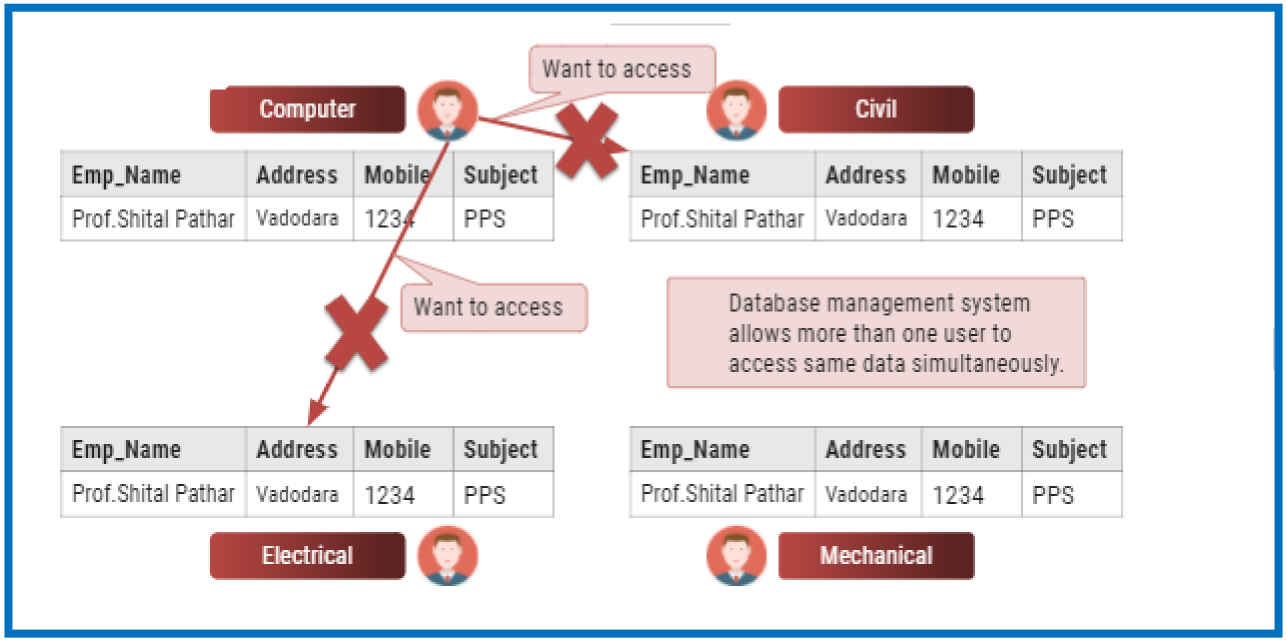
Student_Name	Branch	Backlog	SPI
Krishna Patel	CSE	0	8.5

Should be between 0 to 10
DBMS allows us to implement such business rules in our database.

Figure: 1.11 An example integrity constraints

(Image Source : http://www.allfordrugs.com/data-integrity/)

Sharing of data among multiple users



Emp_Name	Address	Mobile	Subject
Prof.Shital Pathar	Vadodara	1234	PPS

Emp_Name	Address	Mobile	Subject
Prof.Shital Pathar	Vadodara	1234	PPS

Database management system allows more than one user to access same data simultaneously.

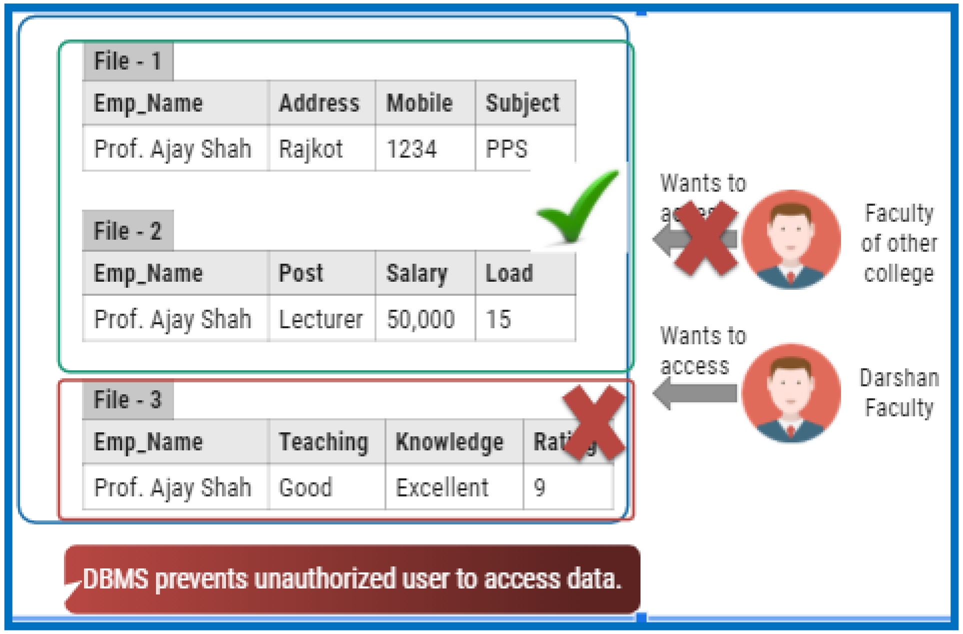
Emp_Name	Address	Mobile	Subject
Prof.Shital Pathar	Vadodara	1234	PPS

Emp_Name	Address	Mobile	Subject
Prof.Shital Pathar	Vadodara	1234	PPS

Figure: 1.12 An example of **Sharing**

(**Image Source** : http://www.allfordrugs.com/data-integrity/)

Restricting unauthorized access to data



File - 1

Emp_Name	Address	Mobile	Subject
Prof. Ajay Shah	Rajkot	1234	PPS

File - 2

Emp_Name	Post	Salary	Load
Prof. Ajay Shah	Lecturer	50,000	15

File - 3

Emp_Name	Teaching	Knowledge	Rating
Prof. Ajay Shah	Good	Excellent	9

Figure: 1.13 An example of unauthorized access to data

(**Image Source** : ttps://www.mageplaza.com/kb/your-web-server-set-up-incorrectly-allows-unauthorized-access-sensitive-files-magento-2.html)

Providing backup and recovery services

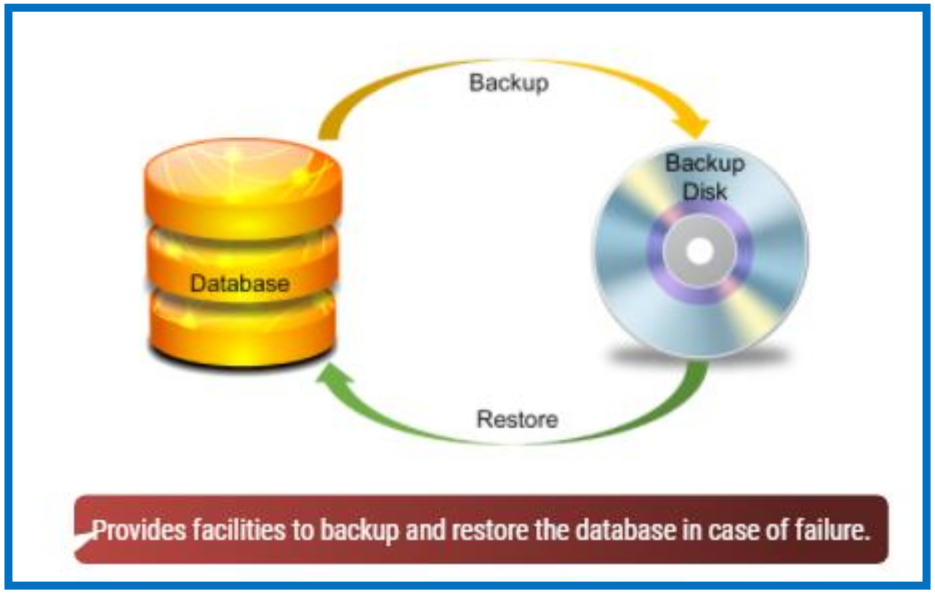

Backup
Backup Disk
Database
Restore
Provides facilities to backup and restore the database in case of failure.

Figure: 1.14 An example of backup and recovery services

(**Image Source** : https://www.enterprisestorageforum.com/backup-recovery/enterprise-backup-and-recovery-software.html)

Advantages of DBMS (Summary)

- Reduce data redundancy (duplication)
  - **Avoids unnecessary duplication** of data by storing data centrally.
- Remove data inconsistency
  - By **eliminating redundancy**, data **inconsistency can be removed**.
- Data isolation
  - A user can **easily retrieve proper data** as per his/her requirement.
- Guaranteed atomicity
  - Either transaction **executes 0% or 100%.**

Advantages of DBMS (Summary)

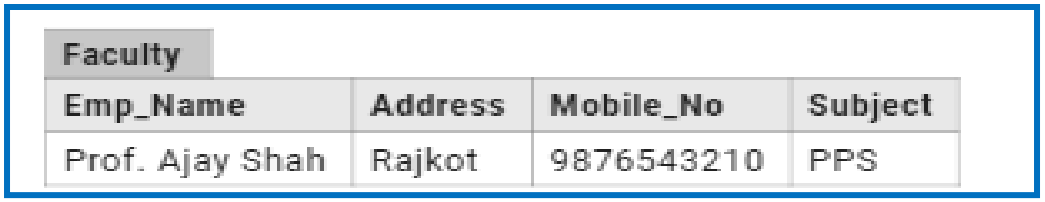
- Allow implementing integrity constraints
  - **Business rules can be implemented** such as do not allow to store amount less than Rs. 0 in balance.
- Sharing of data among multiple users
  - **More than one users can access** same data at the same time.
- Restricting unauthorized access to data
  - A user can **only access data which is authorized** to him/her.
- Providing backup and recovery services
  - Can **take a regular auto or manual backup** and **use it to restore** the database if it corrupts.

Basic Terms of DBMS

- **Data**
  - Data is **raw, unorganized facts** that need to be processed.
  - Example: Marks of students
  - Student_1 = 50/100, Student_2 = 25/100.
- **Information**
  - When data is **processed, organized, structured** or presented in a given context so as to make it useful, it is called information.
  - Example: Result of students (Pass or Fail)
  - Student_1 = Pass, Student_2 = Fail.

Basic Terms of DBMS

- **Metadata**
  - Metadata is **data about data**.
  - Data such as table name, column name, data type, authorized user and user access privileges for any table is called metadata for that table.



Faculty

Emp_Name	Address	Mobile_No	Subject
Prof. Ajay Shah	Rajkot	9876543210	PPS

Figure: 1.15 An example of Metadata

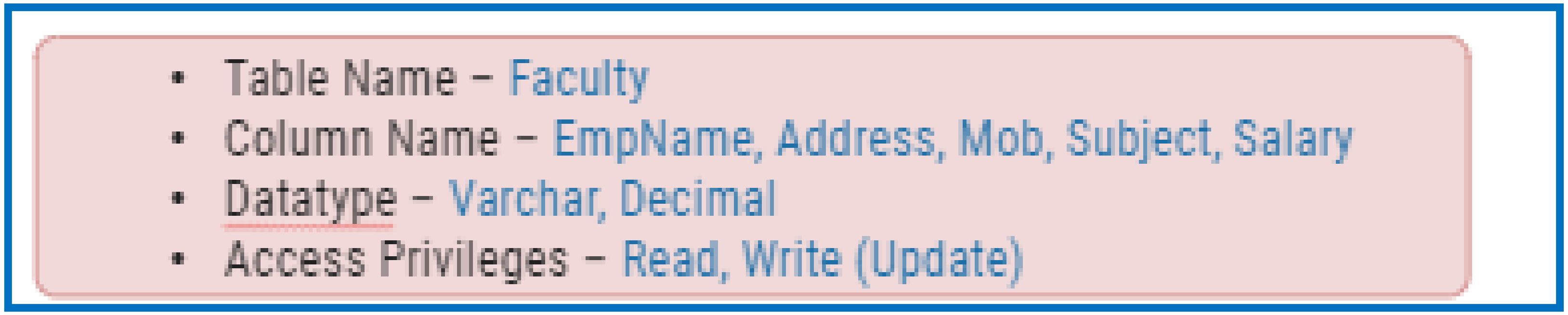
(**Image Source** : https://images.app.goo.gl/cbtcdV1EMn3SoWzi7)

Metadata of above table is:

- Table name such as Faculty
- Column name such as Emp_Name, Address, Mobile_No, Subject
- Datatype such as Varchar, Decimal
- Access privileges such as Read, Write (Update)

Basic Terms of DBMS

- **Data dictionary:-**A data dictionary is an information repository which **contains metadata**.

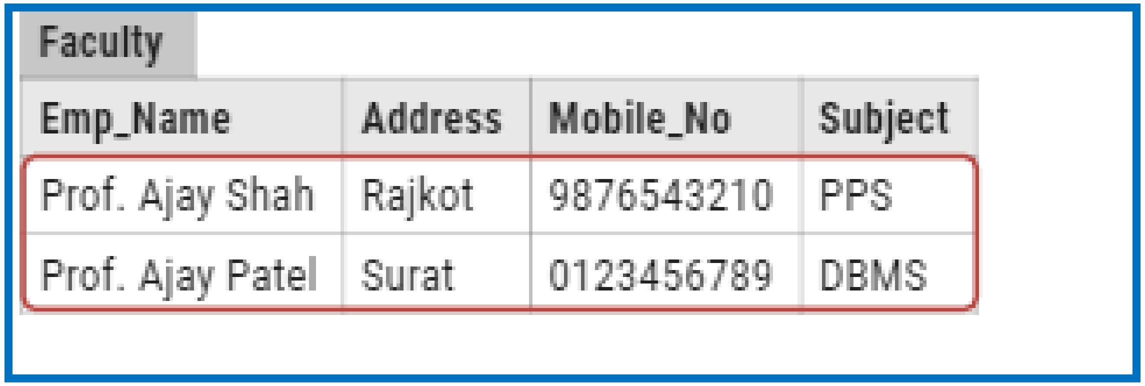


- Table Name – Faculty
- Column Name – EmpName, Address, Mob, Subject, Salary
- Datatype – Varchar, Decimal
- Access Privileges – Read, Write (Update)

Figure: 1.16 An example of data dictionary
(**Image Source** : https://dataedo.com/kb/data-glossary/what-is-data-dictionary)

Basic Terms of DBMS

- **Data warehouse**
  - A data warehouse is an information repository which **stores data**.



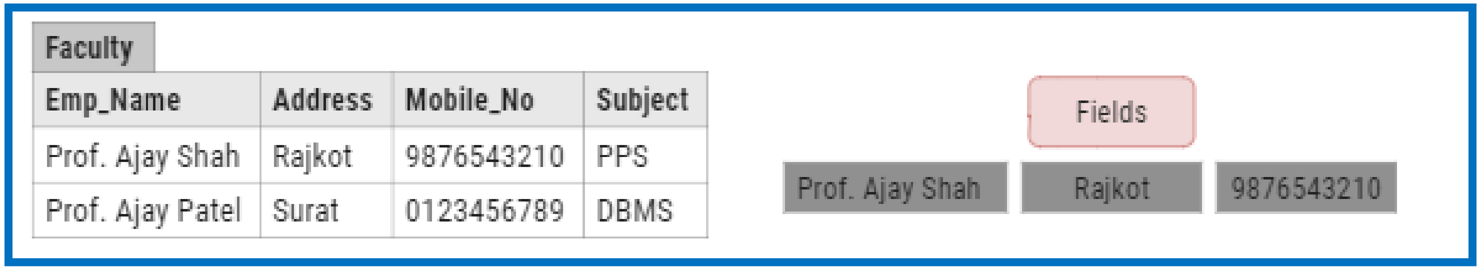
Faculty

Emp_Name	Address	Mobile_No	Subject
Prof. Ajay Shah	Rajkot	9876543210	PPS
Prof. Ajay Patel	Surat	0123456789	DBMS

Figure: 1.17 An example of warehouse
(**Image Source** : https://images.app.goo.gl/cbtcdV1EMn3SoWzi7)

Basic Terms of DBMS

- **Field**
  - A field is a **character or group of characters** that have a specific meaning.
  - E.g, the value of Emp_Name, Address, Mobile_No etc are all fields of Faculty table.



Faculty

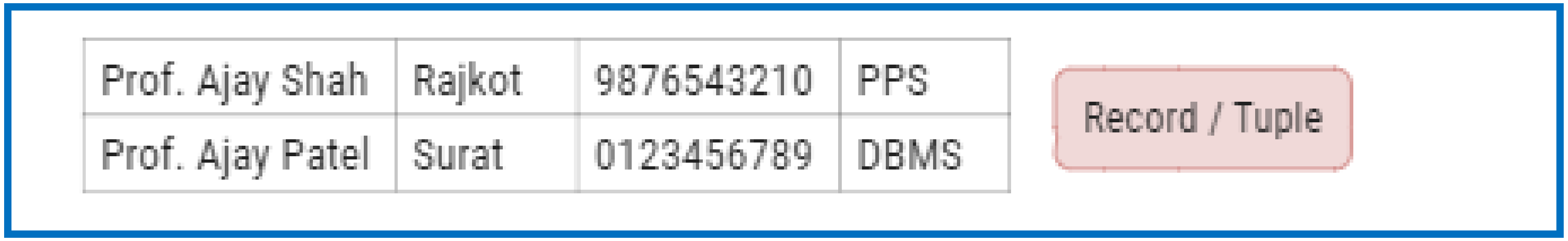
Emp_Name	Address	Mobile_No	Subject
Prof. Ajay Shah	Rajkot	9876543210	PPS
Prof. Ajay Patel	Surat	0123456789	DBMS

Figure: 1.18 An example of field

(**Image Source :** https://images.app.goo.gl/cbtcdV1EMn3SoWzi7)

Basic Terms of DBMS

- Record / Tuple
  - A record is a **collection of logically related fields**.
  - E.g, the collection of fields (Emp_Name, Address, Mobile_No, Subject) forms a record for the Faculty.



Prof. Ajay Shah	Rajkot	9876543210	PPS
Prof. Ajay Patel	Surat	0123456789	DBMS

Record / Tuple

Figure: 1.19 An example of Record
(**Image Source :** https://images.app.goo.gl/cbtcdV1EMn3SoWzi7)

3 Levels ANSI SPARC Database System

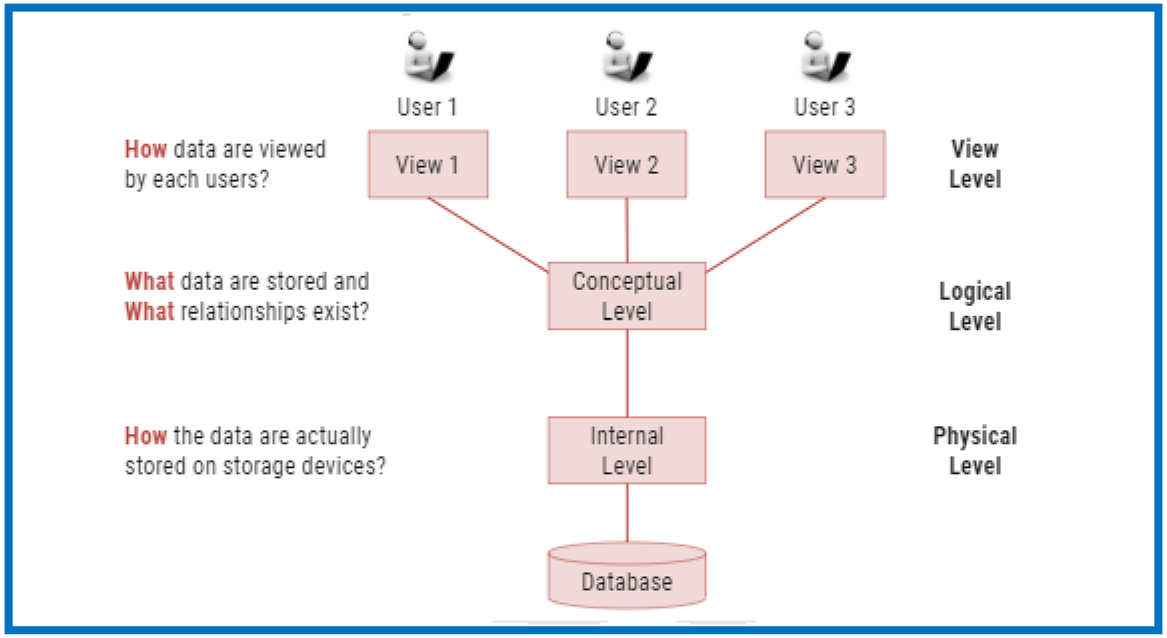

User 1
User 2
User 3
How data are viewed by each users?
View 1
View 2
View 3
View Level
What data are stored and What relationships exist?
Conceptual Level
Logical Level
How the data are actually stored on storage devices?
Internal Level
Physical Level
Database

Figure: 1.20 ANSI SPARC Database System

(**Image Source** : https://pt.slideshare.net/NurHidayahKhazali/chapter-2-database-environment/13)

Internal Level

- The internal level has an internal schema which describes the physical storage structure of the database.
- The internal schema is also known as a physical schema.
- It uses the physical data model. It is used to define that how the data will be stored in a block.
- The physical level is used to describe complex low-level data structures in detail.

Conceptual Level

- The conceptual schema describes the design of a database at the conceptual level. Conceptual level is also known as logical level.
- The conceptual schema describes the structure of the whole database.
- The conceptual level describes what data are to be stored in the database and also describes what relationship exists among those data.
- In the conceptual level, internal details such as an implementation of the data structure are hidden.
- Programmers and database administrators work at this level.

External Level

- At the external level, a database contains several schemas that sometimes called as subschema. The subschema is used to describe the different view of the database.
- An external schema is also known as view schema.
- Each view schema describes the database part that a particular user group is interested and hides the remaining database from that user group.
- The view schema describes the end user interaction with database systems.

Data Abstraction in DBMS

- Database systems are made-up of complex data structures.
- To ease the user interaction with database, the developers hide internal irrelevant details from users.
- This **process of hiding irrelevant details** from user is called data abstraction.

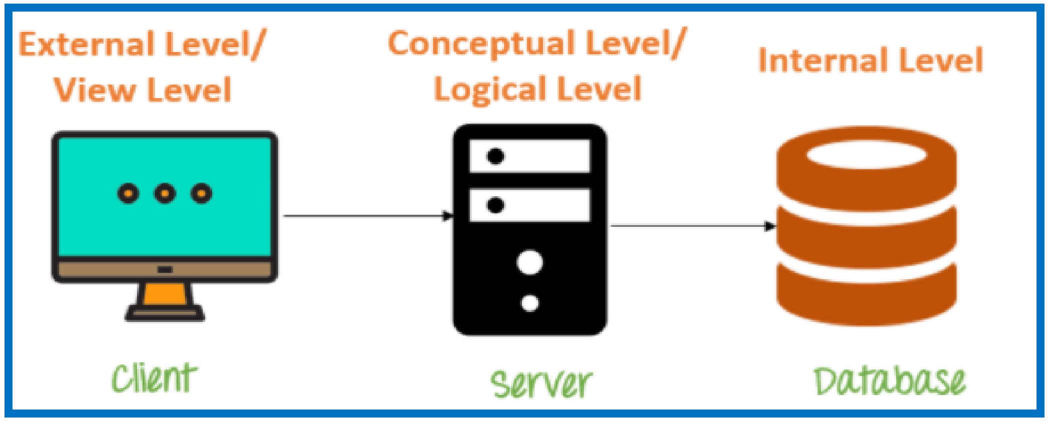


Figure: 1.21 An example of Data Abstraction

(**Image Source** : https://prepinsta.com/dbms/data-abstraction-and-data-independence/)

Mapping and Data Independence

Process of transforming requests and results between the three levels is called mapping.

Ability to modify a schema definition in one level without affecting a schema definition in the next higher level.

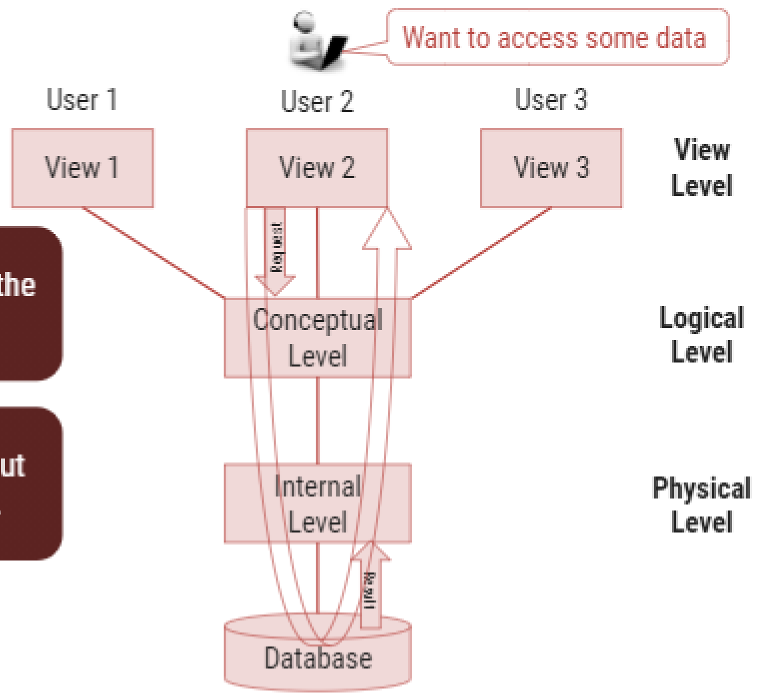


Figure1.22: Mapping and Data Independence

(**Image Source** : https://images.app.goo.gl/cbtcdV1EMn3SoWzi7)

Types of Data Independence

- **Physical Data Independence**
  - Physical Data Independence is the ability to modify the physical schema without requiring any change in logical (conceptual) schema and application programs.
  - Modifications at the internal levels are occasionally necessary to improve performance.
  - Possible modifications at internal levels are changes in file structures, compression techniques, hashing algorithms, storage devices, etc.

Types of Data Independence

- **Logical Data Independence**
  - Logical data independence is the ability to modify the conceptual schema without requiring any change in application programs.
  - Modification at the logical levels is necessary whenever the logical structure of the database is changed.
  - Application programs are heavily dependent on logical structures of the data they access. So any change in logical structure also requires programs to change.

Types of Database Users

- **Naive Users (End Users)**
  - **Unsophisticated users** who have zero knowledge of database system
  - End user interacts to database via sophisticated software or tools
  - e.g. Clerk in bank
- **Application Programmers**
  - **Programmers** who write software using tools such as Java, .Net, PHP etc…
  - e.g. Software developers

Types of Database Users

- Sophisticated Users
  - **Interact with database system** without using an application program
  - Use query tools like SQL
  - e.g. Analyst
- Specialized Users (DBA)
  - User **write specialized** database applications program
  - Use administration tools
  - e.g. Database Administrator

Role of DBA (Database Administrator)

- Schema Definition
  - DBA **defines the logical schema** of the database.
- Storage Structure and Access Method Definition
  - DBA **decides how the data is to be represented** in the database & how to access it.
- Defining Security and Integrity Constraints
  - DBA **decides on various security and integrity constraints**.
- Granting of Authorization for Data Access
  - DBA **determines which user needs access to which part** of the database.
- Liaison with Users
  - DBA **provide necessary data** to the user.

Role of DBA

- Assisting Application Programmer
  - DBA **provides assistance to application programmers** to develop application programs.
- Monitoring Performance
  - DBA **ensures that better performance is maintained** by making a change in the physical or logical schema if required.
- Backup and Recovery
  - DBA **backing up the database** on some storage devices such as DVD, CD or magnetic tape or remote servers and **recover the system in case of failures**, such as flood or virus attack from this backup.

Database System Architecture

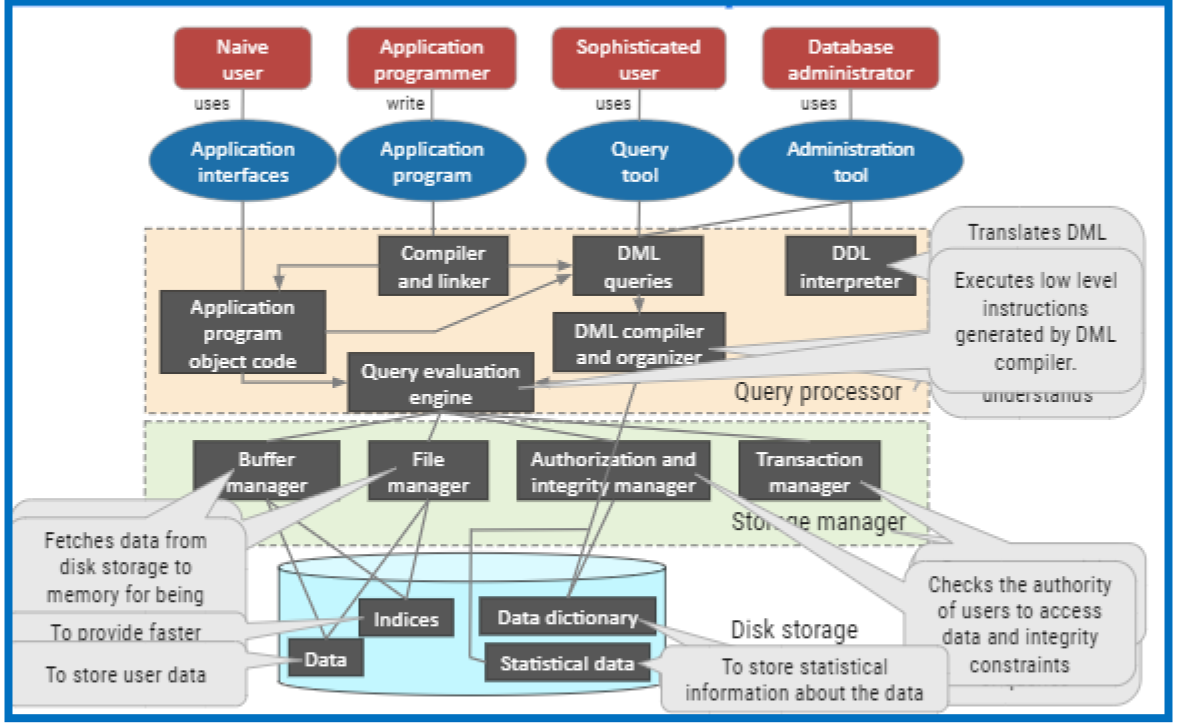


Figure: 1.23 Database System Architecture

(**Image Source** : https://www.researchgate.net/figure/The-general-system-architecture-of-federated-database-systems-with-database-wrappers-as_fig3_272293869)

Structured Query Language(SQL)

- SQL stands for Structured Query Language
- SQL lets you access and manipulate databases
- SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987

What Can SQL do?

- SQL can execute queries against a database
- SQL can retrieve data from a database
- SQL can insert records in a database
- SQL can update records in a database
- SQL can delete records from a database
- SQL can create new databases
- SQL can create new tables in a database
- SQL can create stored procedures in a database
- SQL can create views in a database
- SQL can set permissions on tables, procedures, and views

SQL Commands

- SQL commands are instructions. It is used to communicate with the database. It is also used to perform specific tasks, functions, and queries of data.

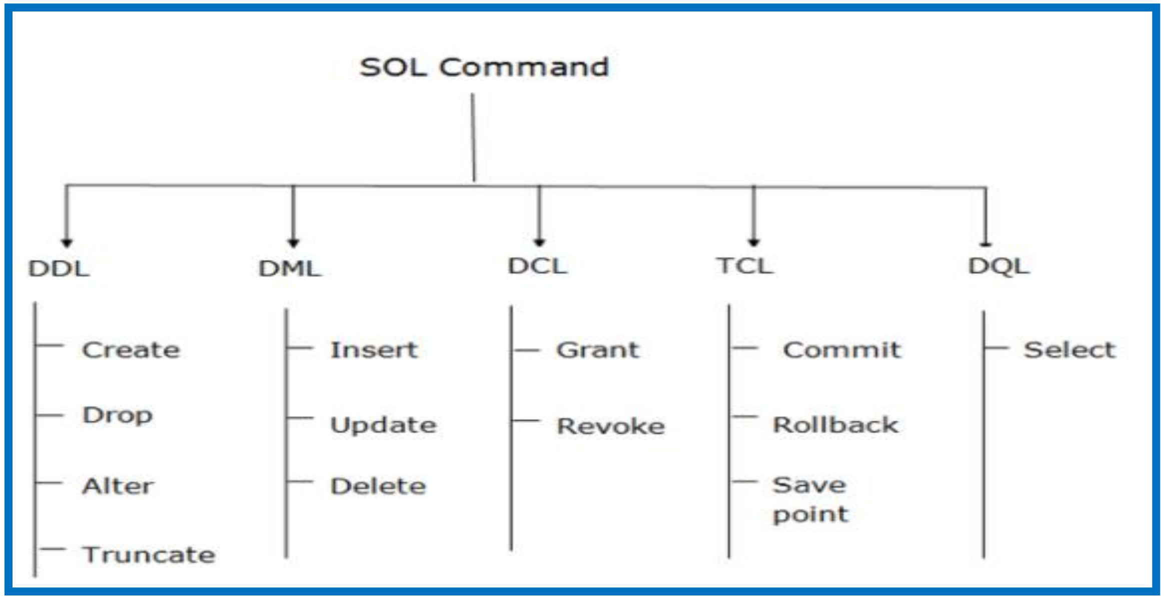


Figure: 1.24 SQL Commands

(**Image Source** : https://www.google.com/imgres?imgurl=https%3A%2F%2Fstatic.javatpoint.com%2Fdbms%2Fimages%2Fdbms-sql-command.png&imgrefurl=http)

Data Definition Language (DDL)

- DDL changes the structure of the table like creating a table, deleting a table, altering a table, etc.
- All the command of DDL are auto-committed that means it permanently save all the changes in the database.
- Here are some commands that come under DDL:

1. CREATE
2. ALTER
3. DROP
4. TRUNCATE

Data Definition Language (DDL Command)

1. **CREATE** It is used to create a new table in the database.

- **Syntax:**
  - CREATE TABLE TABLE_NAME (COLUMN_NAME DATATYPES[,....]);

- **Example:**
  - CREATE TABLE EMPLOYEE(Name VARCHAR2(20), Email VARCHAR2(100), DOB DATE);

Data Definition Language (DDL Command)

2. **DROP:** It is used to delete both the structure and record stored in the table.

- **Syntax**
  - DROP TABLE ;
- **Example**
  - DROP TABLE EMPLOYEE;

Data Definition Language (DDL Command)

3. **ALTER:** It is used to alter the structure of the database. This change could be either to modify the characteristics of an existing attribute or probably to add a new attribute.

- **Syntax:**
- **To add a new column in the table**
- ALTER TABLE table_name ADD column_name COLUMN-definition;
- **To modify existing column in the table:**
- ALTER TABLE MODIFY(COLUMN DEFINITION....);
- **EXAMPLE:**
- ALTER TABLE STU_DETAILS ADD(ADDRESS VARCHAR2(20));
- ALTER TABLE STU_DETAILS MODIFY (NAME VARCHAR2(20));

Data Definition Language (DDL Command)

4. **TRUNCATE:** It is used to delete all the rows from the table and free the space containing the table.

- **Syntax:**
  - TRUNCATE TABLE table_name;
- **Example:**
  - TRUNCATE TABLE EMPLOYEE;

Data Manipulation Language(DML)

- DML commands are used to modify the database. It is responsible for all form of changes in the database.
- The command of DML is not auto-committed that means it can't permanently save all the changes in the database. They can be rollback.
- Here are some commands that come under DML:

1. INSERT
2. UPDATE
3. DELETE

Data Manipulation Language(DML Commands)

1. **INSERT:** The INSERT statement is a SQL query. It is used to insert data into the row of a table.

- **Syntax:**
  –
  ```
  INSERT INTO TABLE_NAME  (col1, col2, col3,.... col N)
  VALUES (value1, value2, value3, .... valueN);
  Or
  INSERT INTO TABLE_NAME
  VALUES (value1, value2, value3, .... valueN);
  ```
- **For example:**
  –
  ```
  INSERT INTO BOOK (Author, Subject) VALUES ("Shital", "DBMS");
  ```

Data Manipulation Language(DML Commands)

2. **UPDATE:** This command is used to update or modify the value of a column in the table.

- **Syntax:**
  - UPDATE table_name SET [column_name1= value1,...column_nameN = value N] [WHERE CONDITION]

- **For example:**
  - UPDATE students   SET User_Name = 'Shital   WHERE Student_Id = '3'

Data Manipulation Language(DML Commands)

3. **DELETE:** It is used to remove one or more row from a table.

- **Syntax:**
  - DELETE FROM table_name [WHERE condition];
- **For example:**
  - DELETE FROM Book WHERE Author="Shital";

Data Control Language(DCL)

- DCL commands are used to grant and take back authority from any database user.
- Here are some commands that come under DCL:

1. Grant
2. Revoke

Data Control Language(DCL Commands)

1. **Grant:** It is used to give user access privileges to a database.

- **Example**
  – GRANT SELECT, UPDATE ON MY_TABLE TO SOME_USER, ANOTHER_USER;

2. **Revoke:** It is used to take back permissions from the user.

- **Example**
  – REVOKE SELECT, UPDATE ON MY_TABLE FROM USER1, USER2;

Transaction Control Language(TCL)

- TCL commands can only use with DML commands like INSERT, DELETE and UPDATE only.
- These operations are automatically committed in the database that's why they cannot be used while creating tables or dropping them.
- Here are some commands that come under TCL:

1. COMMIT
2. ROLLBACK
3. SAVEPOINT

Transaction Control Language(TCL Commands)

1. **Commit:** Commit command is used to save all the transactions to the database.

- **Syntax:-**COMMIT;
- **Example:**
  - DELETE FROM CUSTOMERS WHERE AGE = 25;
  - COMMIT;

Transaction Control Language(TCL Commands)

2. **Rollback:** Rollback command is used to undo transactions that have not already been saved to the database.

- **Syntax:-**ROLLBACK;
- **Example:**
  - DELETE FROM CUSTOMERS WHERE AGE = 25;
  - ROLLBACK;

Transaction Control Language(TCL Commands)

3. SAVEPOINT: It is used to roll the transaction back to a certain point without rolling back the entire transaction.

- **Syntax:-**SAVEPOINT SAVEPOINT_NAME;

DATA MODELS

Data Models

- Data models re how the logical structure of a database is designed.
- Data Models are fundamental entities to introduce abstraction in a DBMS.
- Data models define how data is connected to each other and how they are processed and stored inside the system.
- It defines how data can be stored, accessed and updated in database management systems.

Data Models

- There are different types of data models in DBMS:-
  - ER(Entity-Relationship)- Model
  - Relational Model
  - Network Model
  - Object Oriented Model

ER Model

Basic Concepts:-

- **What is E-R diagram?**
  - E-R diagram means Entity-Relationship diagram
  - It is a visual tool for **graphical (pictorial) representation** of database.
  - ER Model was proposed by Peter Chen in 1970’s to use it for a conceptual modelling/designing of database.

ER Model

Basic Concepts:-

- **What is E-R diagram?**
  - It is based on the view of real-world entities and relationships among them.
  - While expressing real-world scenario into the database model, the ER Model creates **entity set, relationship set, general attributes and constraints**.
  - ER Model mainly focuses on **Entities and their attributes** and **Relationships** among entities.
  - It uses various types of symbols to represent objects of database.

Entity

- Entity is real-world objects, place, person about which we collect data
- Example: For University database entity can be Student, Teacher, Department, Course, Result, Class, etc.
- Entity is denoted by a **Rectangle** containing name of entity.

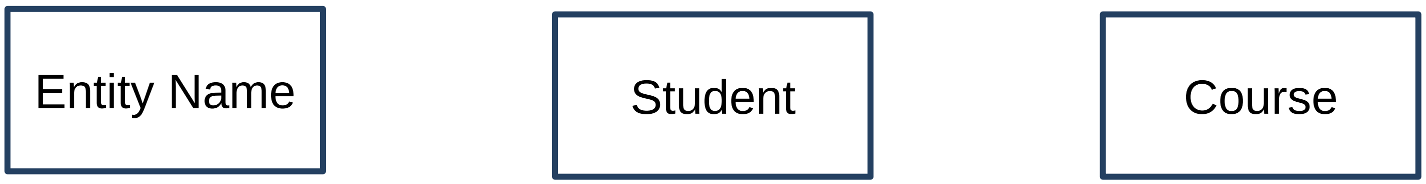


Figure: 1.25 Entity

Attributes

- Attribute represent **properties or details** about an entity.
- Example: For student entity, attributes can be name, enrollment no, address, date of birth, result, etc.
- It is denoted by an **oval** symbol having name of an attribute.

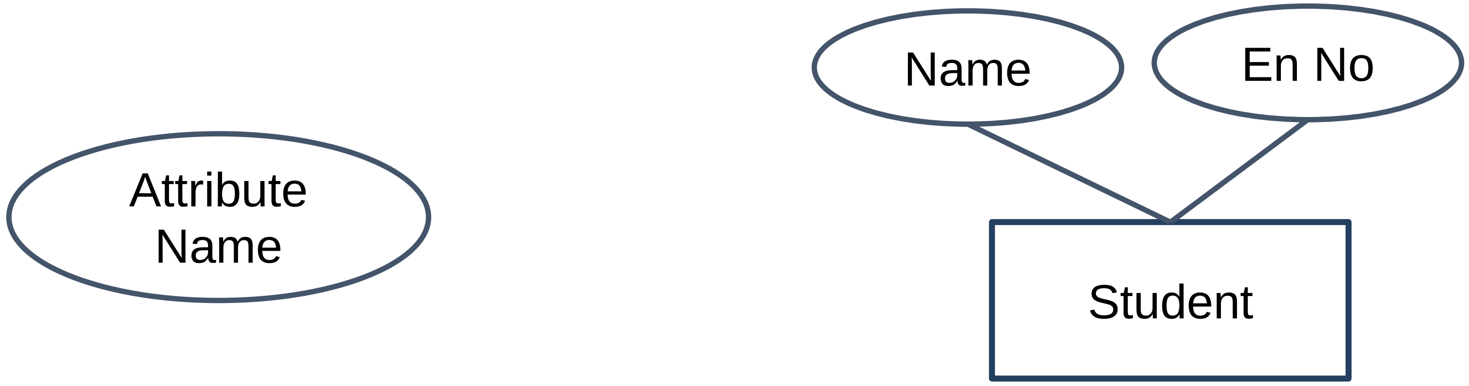
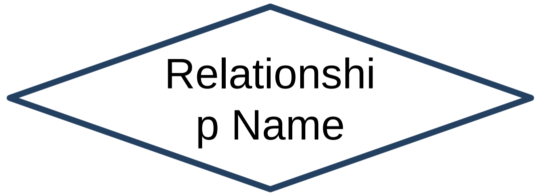


Figure: 1.26 Attributes

Relationship

- Relationship is an association/connection between several entities.
- It defines how entities are related to each other.
- It is denoted by a **diamond** containing relationship's name.
- Diamond should be placed between two entities and a line connecting to both entity.
- Example: Book from library is issued by student, here book & student are entities and issue is relationship.



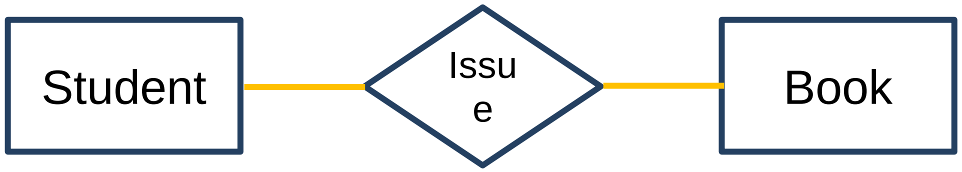


Figure:1.27 Relationship

Entity Set

- Entity Set is a set of entities of the same type having same properties or attributes.
- Example: set of all persons, companies, trees, holidays, all students studying in a university

➢ **Relationship Set:** When there are set of relationships of a same type is called Relationship set

E-R Diagram of Library Management System

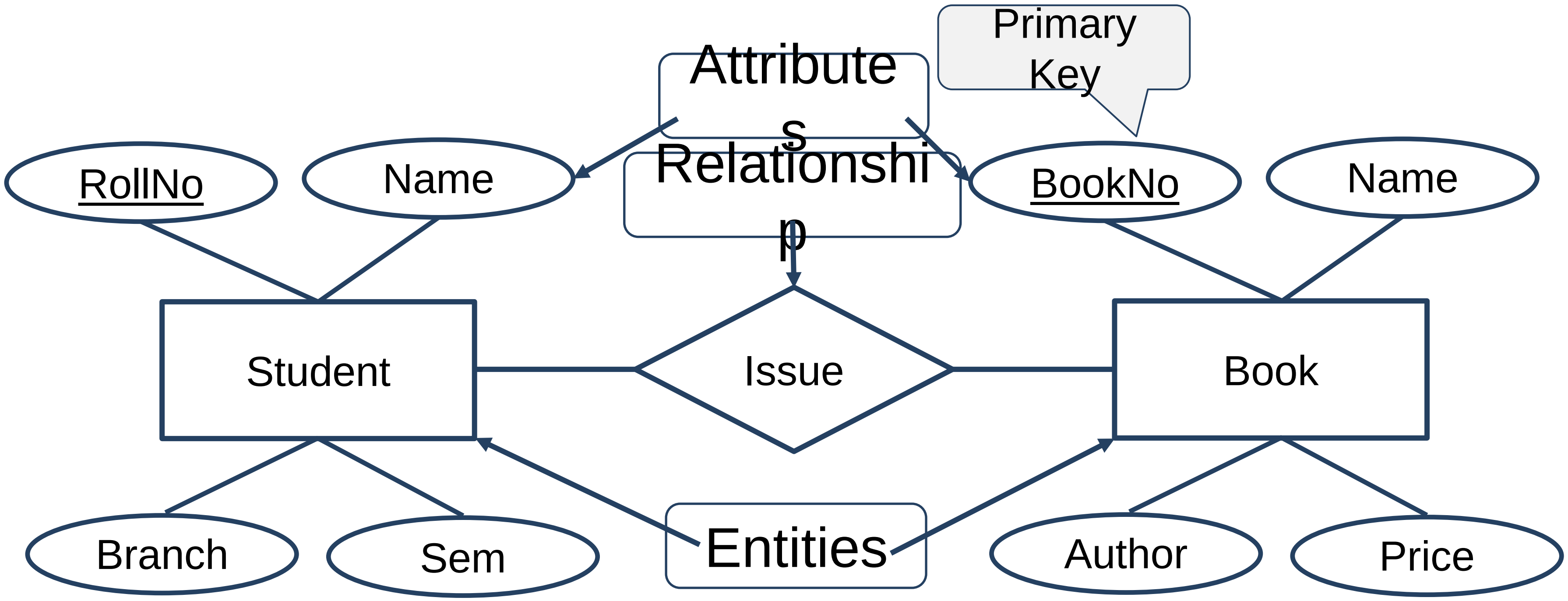


Figure: 1.28 An example of E-R Diagram

Types of Attributes

- Let’s understand different types of attributes
    1. Simple and composite attributes.
    2. Single-valued and multi-valued attributes
    3. Stored attribute and Derived attributes
    4. Complex Attribute
    5. Key Attribute

1. Simple and composite attributes

- **Simple Attribute:**

- It cannot be divided further in more subparts.
- It is like undivided atomic value.
- Example: Price, Year, Enno, CPI

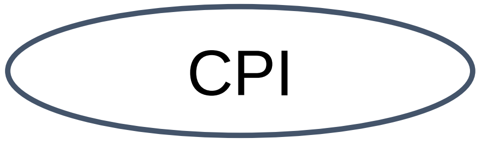


Figure: 1.29 Simple Attribute

1. Simple and composite attributes

➢ **Composite Attribute:**

- It can be divided further in more subparts.
- It is an attribute composed of many other attributes.
- Example: Name

  (first name, middle name, last name)

  Address

  (street, road, city)

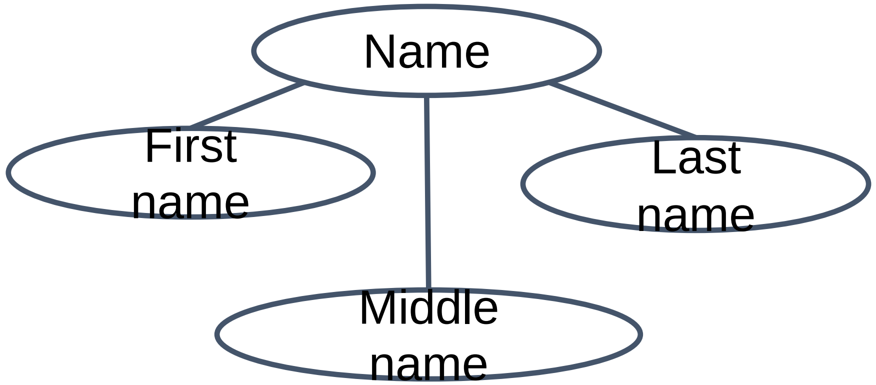


Figure: 1.30 Composite Attribute

2. Single-valued and multi-valued attributes

➢ **Single-Valued Attribute:**

- As name suggests it has single value only.
- Example: Enno, Birthdate

Figure: 1.31 Single-valued Attribute

2. Single-valued and multi-valued attributes

➢ **Multi-Valued Attribute:**

- It has multiple/more than one values.
- Example: PhoneNo

  (person may have multiple phone nos)

  EmailID

  (person may have multiple emails)

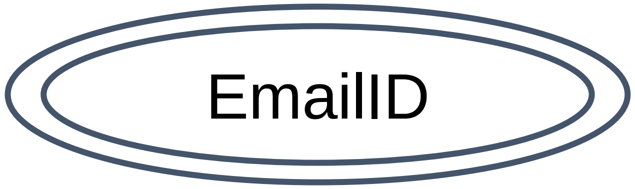


Figure: 1.32 Multi-valued Attribute

3. Stored attribute and Derived attributes

- **Stored Attribute:**

- In this attribute value needs to be stored/defined manually.
- Example: Birthdate, Height, Weight

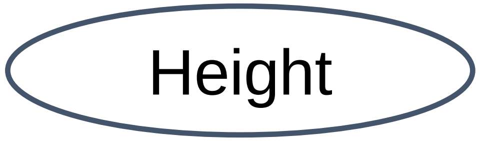


Figure: 1.33 Stored Attribute

3. Stored attribute and Derived attributes

➢ **Derived Attribute:**

- Derived attribute value can be calculated or derived from other attributes.
- Example: Age (Can be derived from current date and birth date)

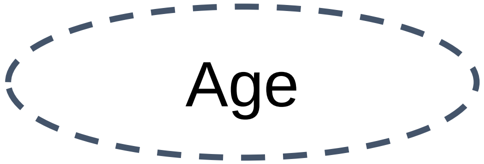


Figure: 1.34 Derived Attribute

4. Complex Attribute

- Attribute that are derived by nesting the composite and multivalued attributes are called complex attributes.

Address_ phone((phone),	address (H.no., city, state))
Complex Attribute	MV Attribute / Composite Attribute

Address_ phone((phone), address (H.no., city, state))

Complex Attribute MV Attribute Composite Attribute

5. Key Attribute

- Attribute which uniquely identifies each entity in the entity set is called key attribute.
- For example, RollNo will be unique for each student.
- It is denoted by an oval with underlying lines.

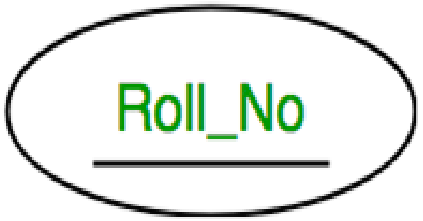


Figure: 1.35 Key Attribute

Example of All Attributes

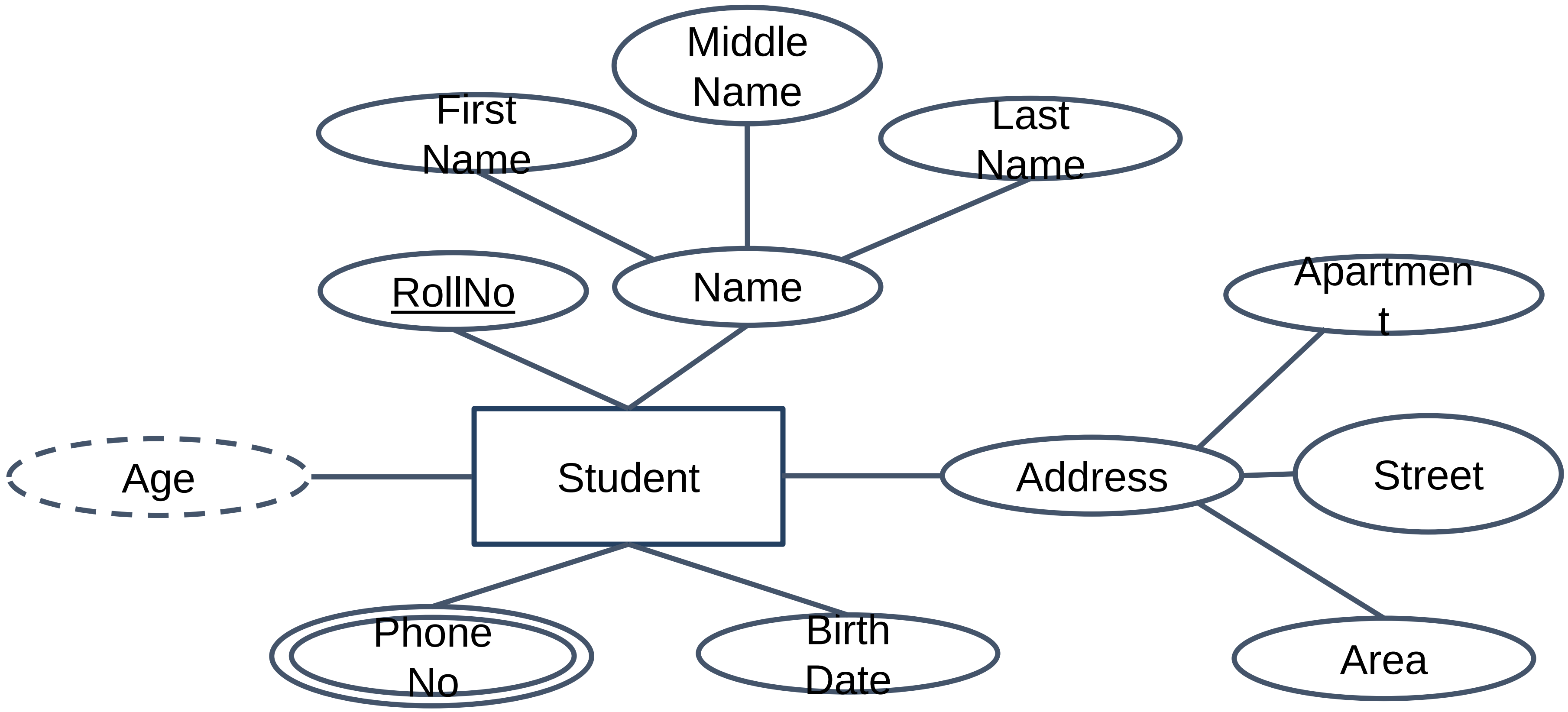


Figure: 1.36 An example of all Attributes

Descriptive Attributes

- If any relationship has a attribute like entity then its known as Descriptive Attributes.
- Example: Student had been issued degree certificate on 25/5/2020. Certificate date is an attribute of relationship.

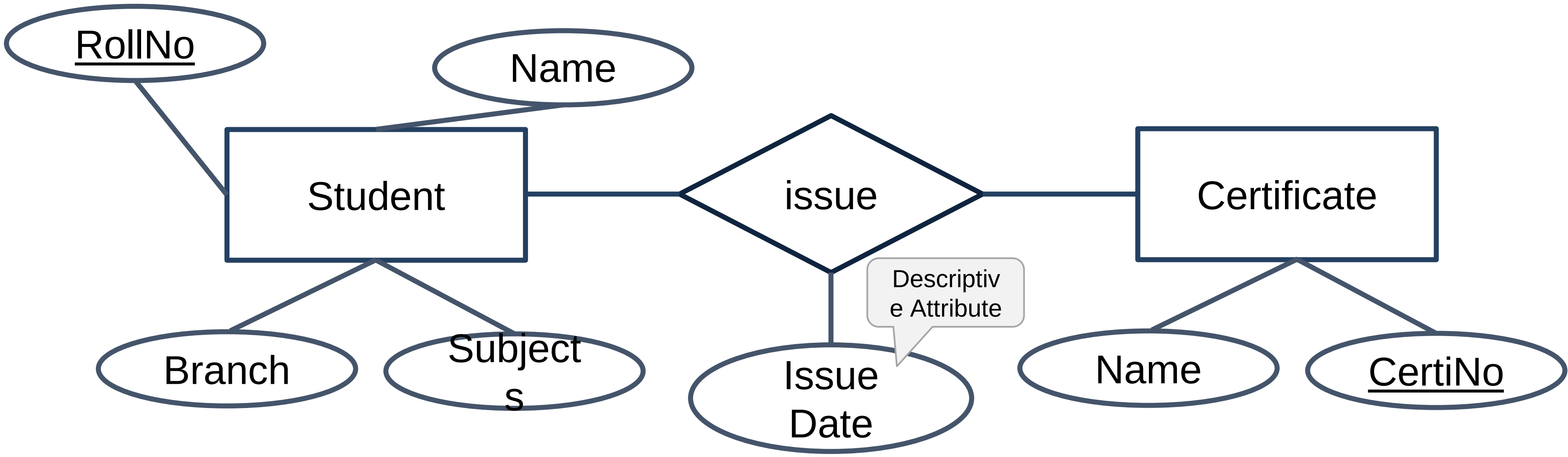


Figure: 1.37 Descriptive Attributes

Recursive Relationship Set

- When one entity set participate in a relationship for more than once then it is called recursive relationship set.

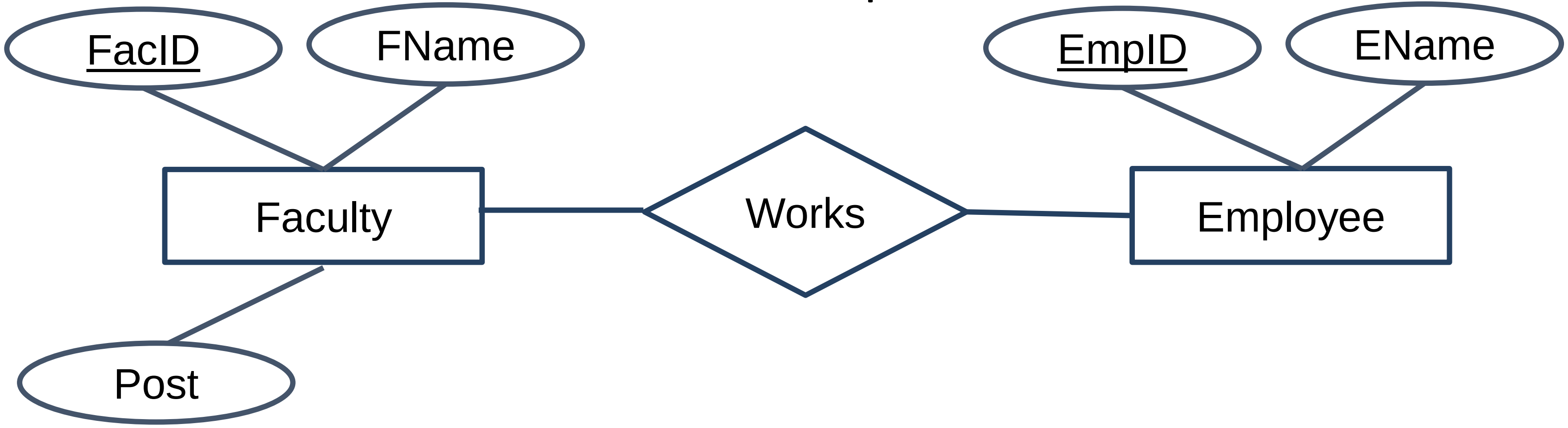


Figure: 1.38 Recursive Relationship set

Degree of a Relationship Set

- Number of entity sets joining in a relationship set is known as a degree of a relationship set.

1. Unary Relationship
2. Binary Relationship/Ternary Relationship
3. n-ary Relationship

Degree of a Relationship Set

1. Unary Relationship:

- When only one entity set participating in a relationship is called Unary Relationship.
- Example: One person is married to only one person.

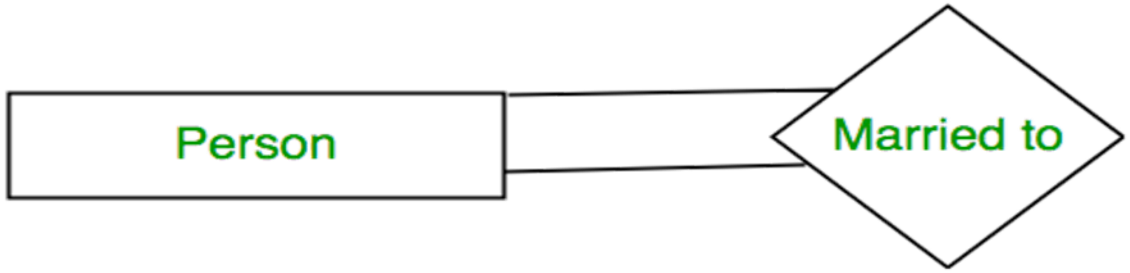


Figure: 1.39 Unary Relationship

Degree of a Relationship Set

2. Binary Relationship:

- When Two entity set participating in a one relationship is called Binary Relationship.
- Example: Student is Enrolled in Course

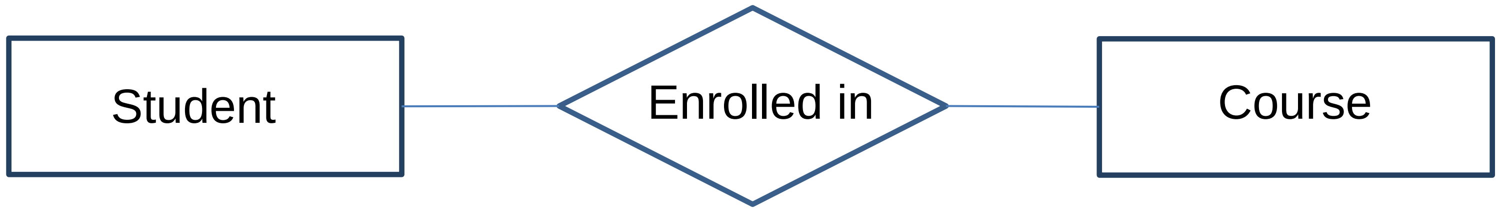


Figure: 1.40 Binary Relationship

Degree of a Relationship Set

3. N-array / Ternary Relationship:

- When there are 3 or N entity set participating in a relationship is called Ternary / N-array Relationship.

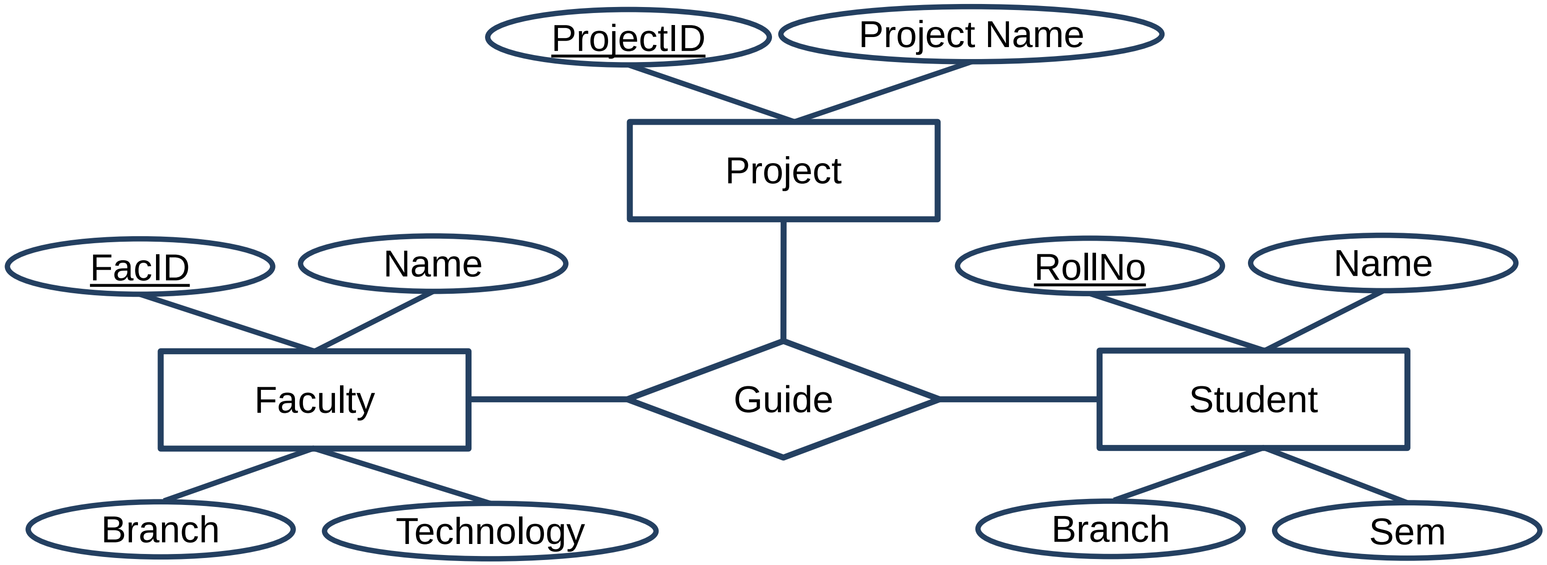


Figure: 1.41 Ternary Relationship

Cardinality Constraints (Mapping Cardinality)

- It defines numbers of times an entity of another entity set participate in a relationship set.
- It helps in defining binary relationship sets.
- Mapping Cardinality for binary relationship can be following types:
  1. One to One
  2. One to Many
  3. Many to One
  4. Many to Many

Cardinality Constraints (Mapping Cardinality)

1.One to One Relationship(1-1):

- An entity in A is associated with only one entity in B and an entity in B is associated with only one entity in A.

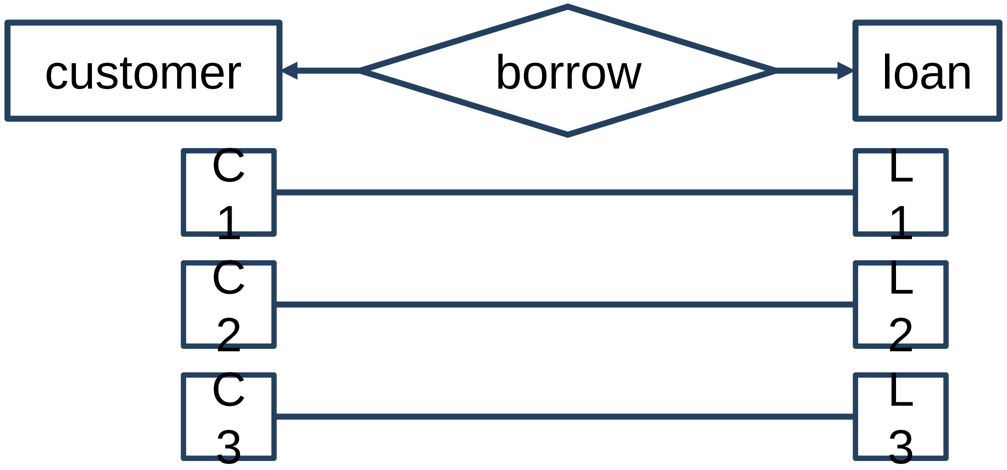


Figure: 1.42 1-1 Relationship

- **Example:** One customer is connected with only one loan through borrower relationship and loan is connected to one customer using borrower.

Cardinality Constraints (Mapping Cardinality)

2. One to Many Relationship(1-M):

- An entity in A is associated with more than one entities in B and an entity in B is associated with only one entity in A.

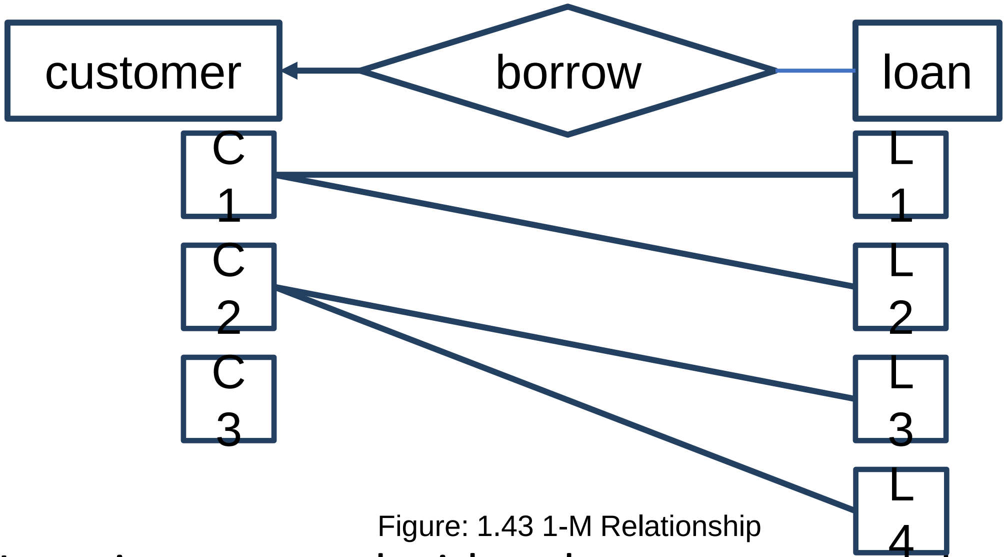


Figure: 1.43 1-M Relationship

- **Example:** Loan is connected with only one customer through borrower relationship and customer is connected with more than one loan.

Cardinality Constraints (Mapping Cardinality)

3. Many to One Relationship(M-1):

- An entity in A is associated with only one entity in B and an entity in B is associated with more than one entities in A.

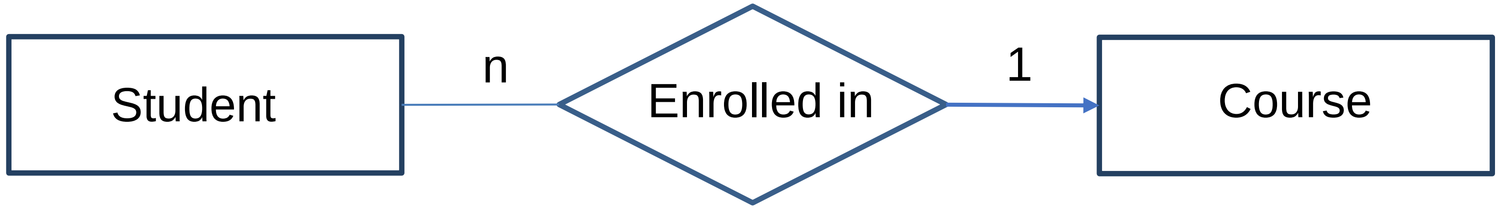


Figure: 1.44 M-1 Relationship

Cardinality Constraints (Mapping Cardinality)

4. Many to Many Relationship(M-M):

- An entity in A is associated with more than one entities in B and an entity in B is associated with more than one entities in A.

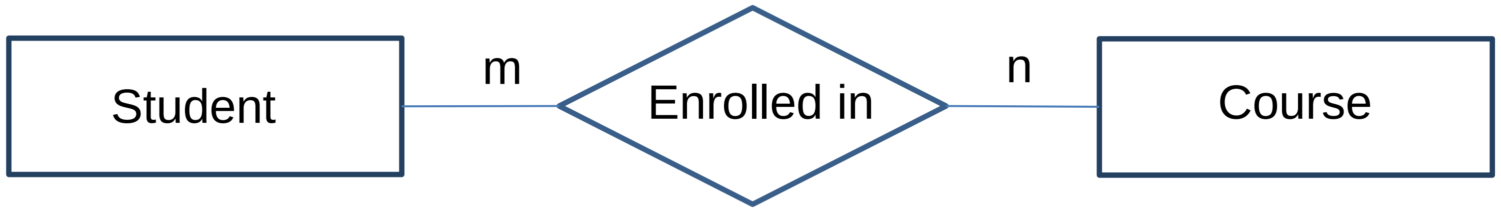


Figure: 1.45 M-M Relationship

Weak Entity Set

- Any Entity set that does not have a primary key of own is known as Weak Entity Set. This entity is known as Dependent Entity.
- Any Entity that has a key attribute is strong entity type and known as a Independent Entity.
- Existence of a weak entity set depends on the existence of a strong entity set.
- Weak Entity set is denoted by double rectangle.
- Weak Entity Relationship set is denoted by double diamond

Weak Entity Set

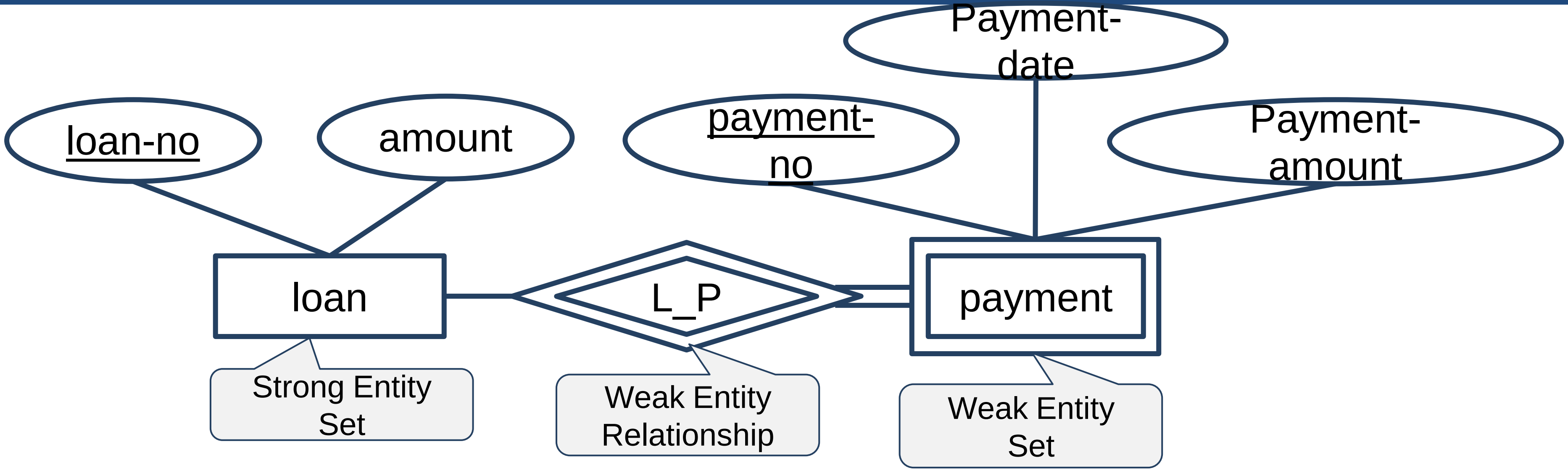


Figure: 1.46 Weak Entity Set

- Primary key of a weak entity set is known as discriminator (partial key), it's created by combining primary key of strong entity set.

Weak Entity Set

- Here Loan-No is primary key of a Loan entity.
- Payment-no is discriminator of payment entity.
- Primary key for Payment is (loan-no,payment-no)
- Discriminator attribute of weak entity set is denoted with dashed underline.

Super Class & Sub Class

- A **superclass** is an entity from which another entities can be derived.
- Example: an entity person has two subsets employee and teacher
- Here person is superclass.

- A **subclass** is an entity that is derived from another entity.
- employee and teacher entities are derived from entity account.
- Here employee and teacher are subclass.

Super Class & Sub Class

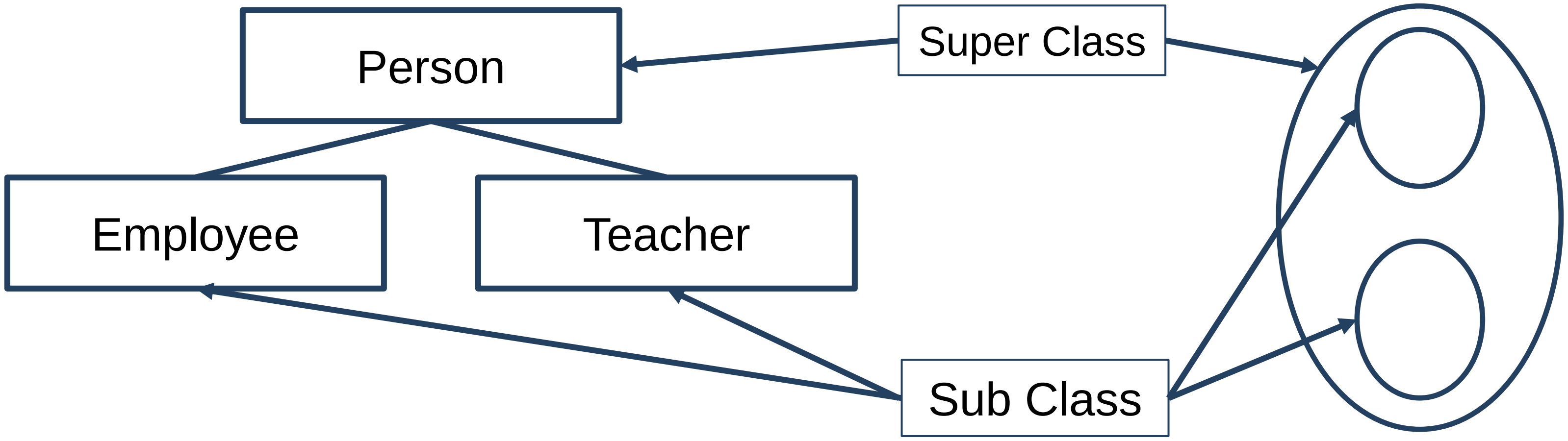


Figure: 1.47 Super & Sub class

Generalization & Specialization

- **Generalization:**

- It determines the common features of multiple entities to create a new entity.
- Generalization is a process of creating group from several entities.
- It follows a bottom-up approach.
- It’s like union of two or more lower level entity sets to make higher level entity set.

Generalization & Specialization

- **Generalization:**

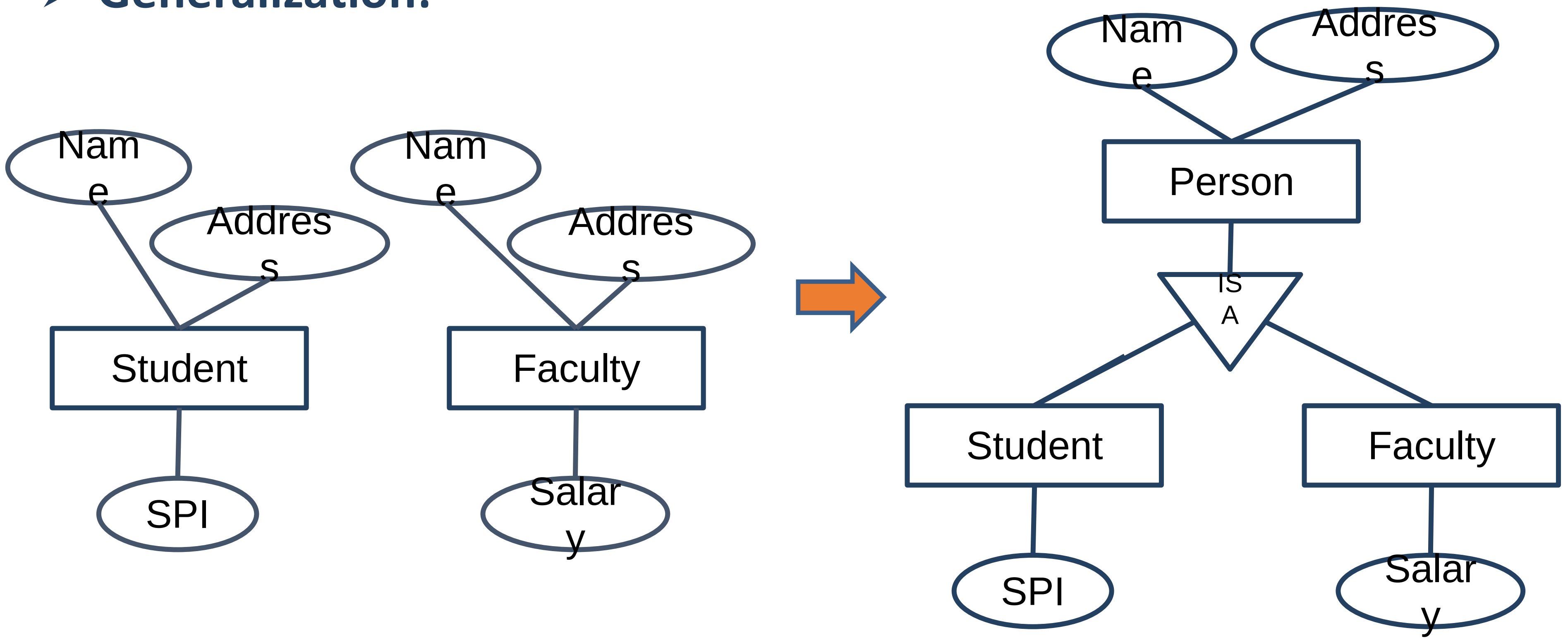


Figure: 1.48 Generalization

Generalization & Specialization

- **Specialization:**

- It divides entity to make multiple entities that inherits some features of splitting entity.
- It is a process of creating subgroups within entities.
- It follows a top-down approach.
- It’s like subset of higher level entity set to form a lower level entity set.

Generalization & Specialization

- **Specialization:**

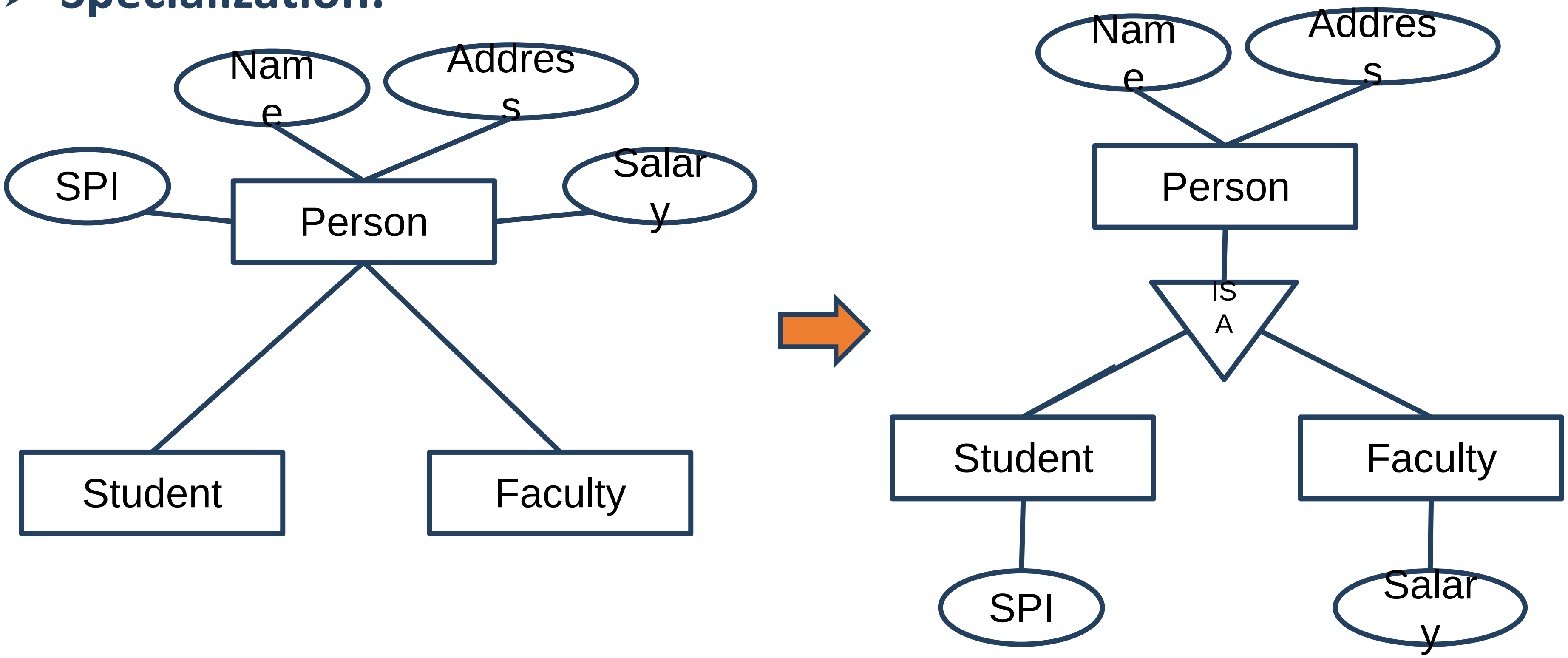


Figure: 1.49 Specialization

Example

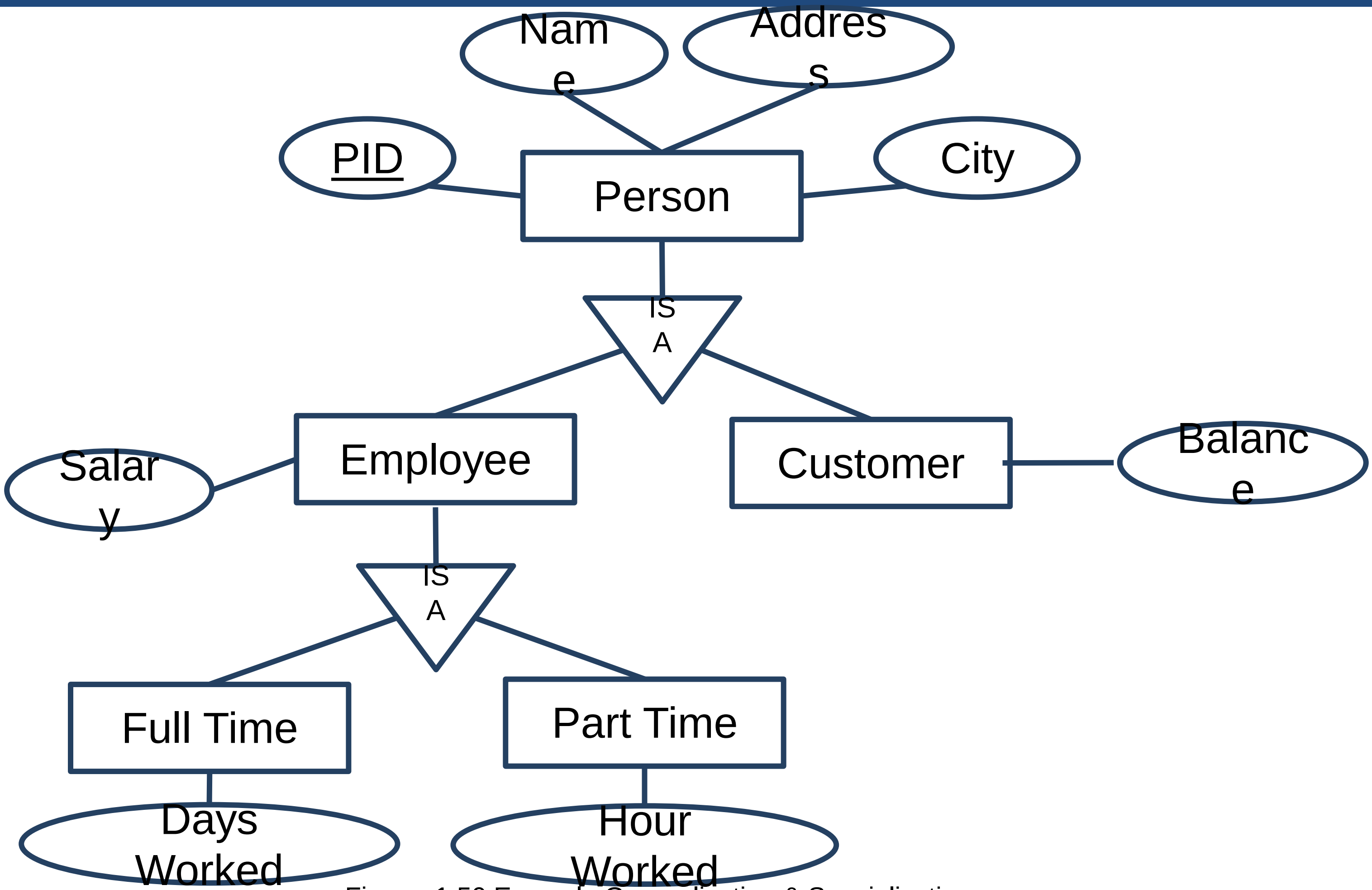


Figure: 1.50 Example Generalization & Specialization

Constraints on Specialization and Generalization

- Constraints are normally divided in two types

1. Disjoint Constraint
2. Participation Constraint

Disjoint Constraints

- It defines relationship of members of superclass and subclass and also indicates member of superclass can be a member of one or more subclass.
    1. Disjoint constraint
    2. Non-disjoint(Overlapping) constraint

Disjoint Constraint

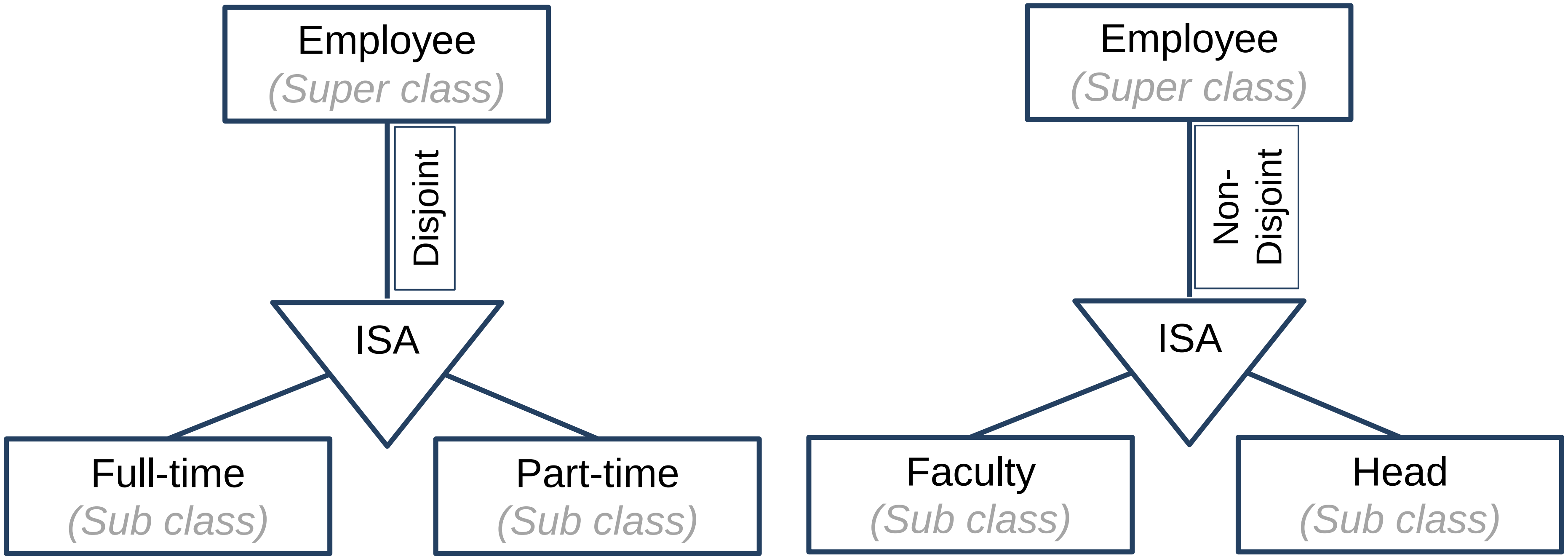
1. **Disjoint Constraint**

- It defines entity of a super class can belong to only one sub class entity set
- It is denoted by **d** or **disjoint** near ISA triangle.

2. **Non-Disjoint Constraint:**

- It defines entity of a super class can belong to more than one sub class entity set.
- It is denoted by **o** or **non-disjoint** near ISA triangle.

Example



All employees are associated with only one sub class either Full time or Part Time

One Employee can be associated with more than one sub class, like faculty and head.

Figure: 1.51 Example of Disjoint & Non-Disjoint Constraint

Participation(Completeness) Constraints

- It defines every member of super class must participate as a member of subclass or not.
- It defines how much entity set participates in a relationship set.
    1. Total(Mandatory) Participation
    2. Partial(Optional) Participation

Participation Constraints

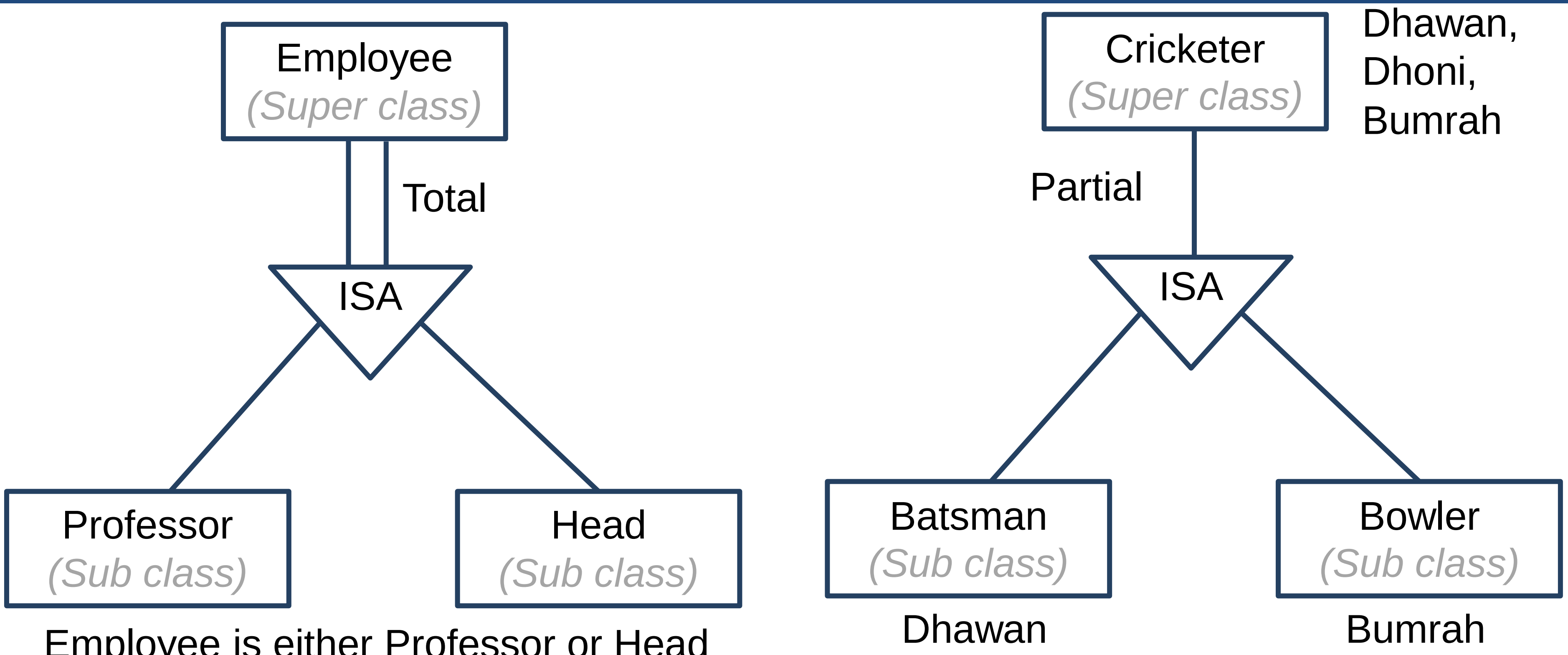
1. **Total Participation:**

- Every entity in the entity set participates in at least one relationship in the relationship set such participation is total.
- Every entity of superclass must be a member of subclass.
- It is denoted by double line connected with relationship

2. **Partial Participation:**

- Some entities in the entity set may not participate in any relationship in the relationship set such participation is partial.
- Every entity in super class does not belong to any of the subclass.
- It is denoted by single line connected with relationship.

Participation Constraints

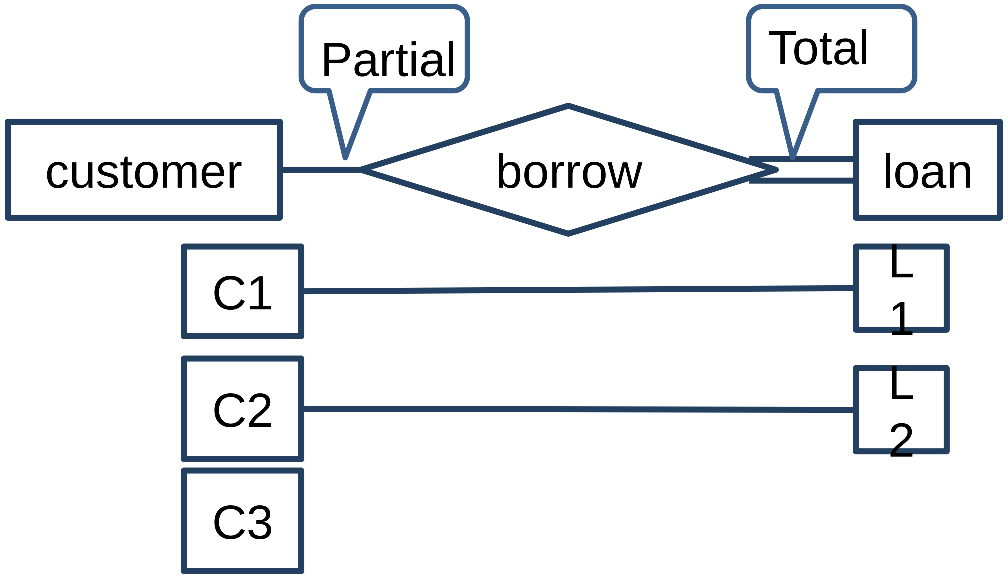


Employee is either Professor or Head

Dhoni is not associated with any subclass

Figure: 1.52 Example of Participation Constraint

Participation Constraints



Each customer has maximum one loan

Figure: 1.53 Example of participation Constraint

Aggregation

- Normally in E-R model relationship between entities are possible to define, but relationships between two relationships defining is not possible, that’s limitation here.
- Aggregation is kind of abstraction which treats relationships as entities.
- Aggregation is process of creating single entity by combining components & relationship between two entity of ER model.

Aggregation

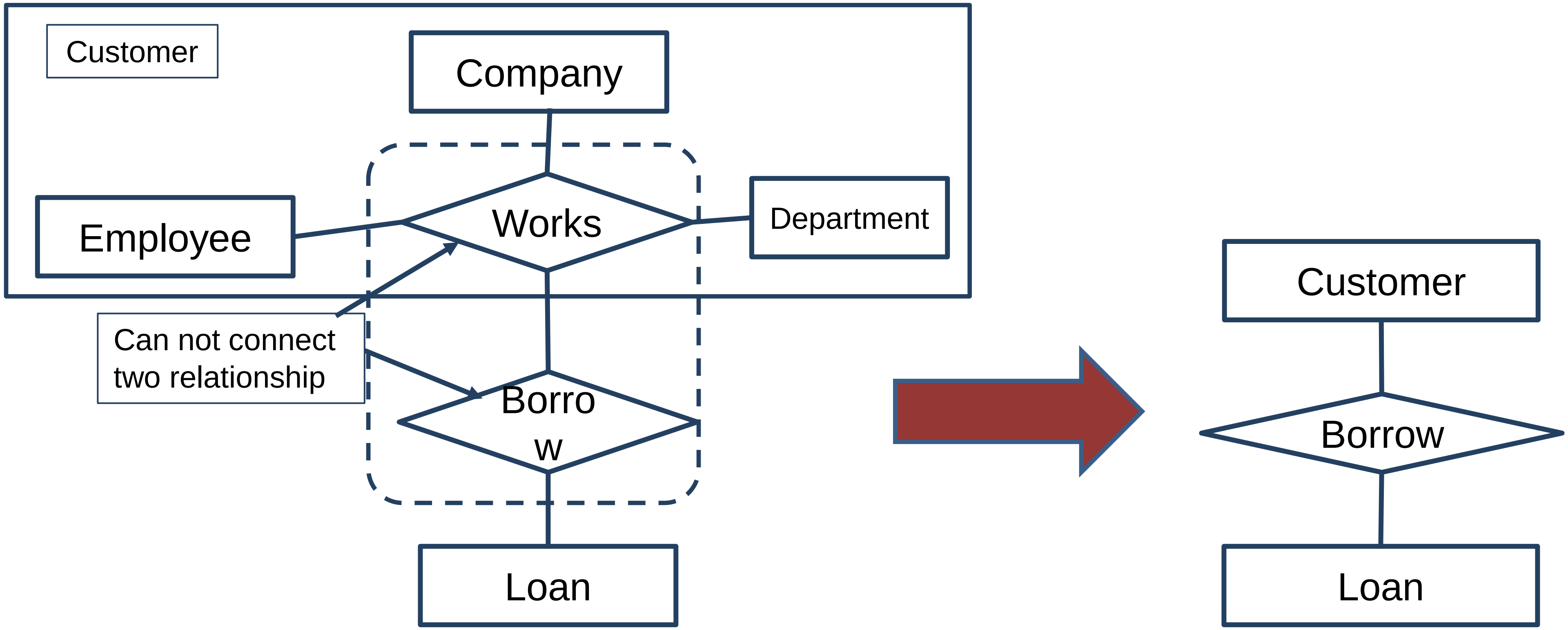


Figure: 1.54 Example of Aggregation

E-R Model Symbols

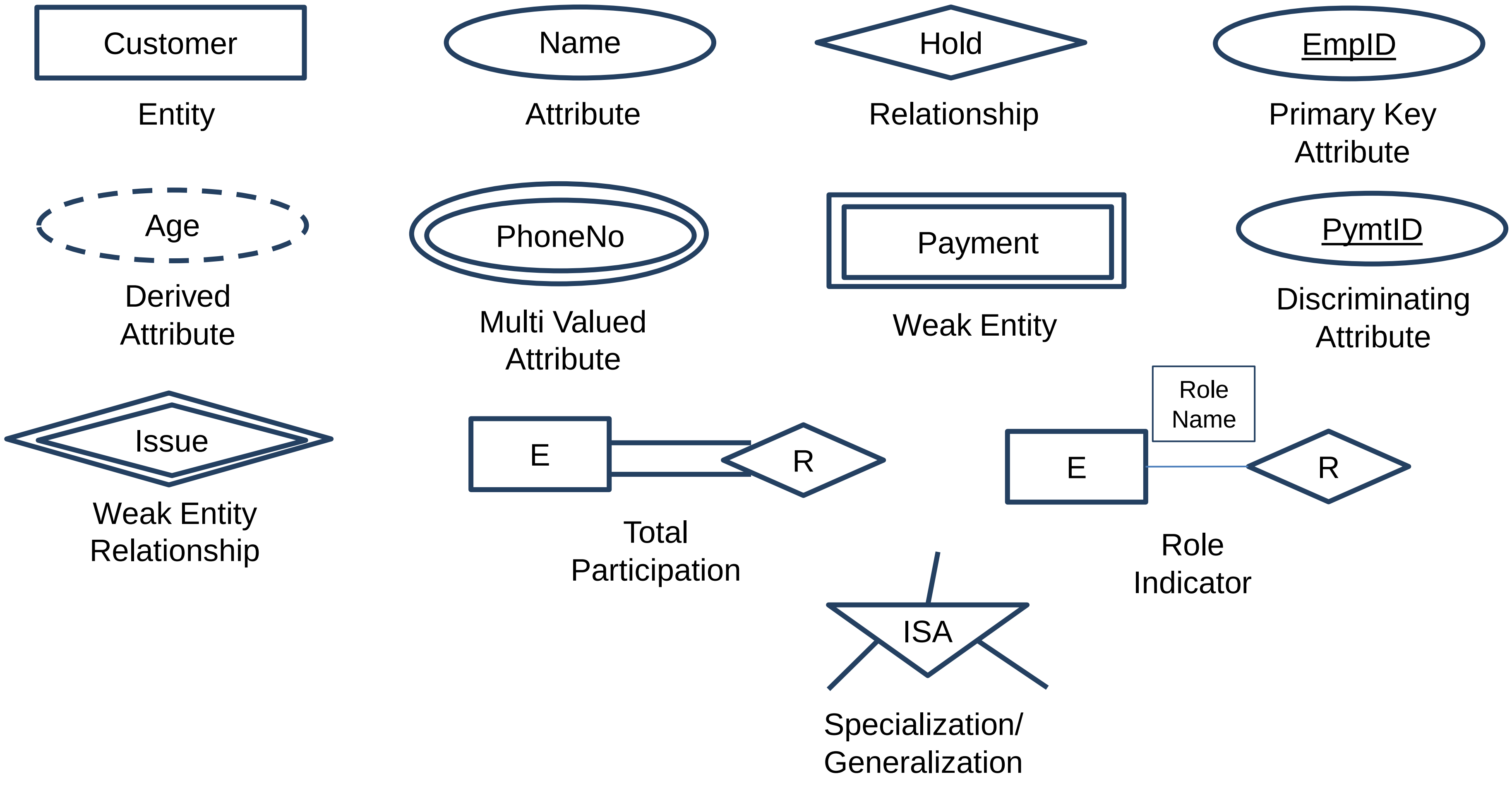


Figure: 1.55 E-R Model Symbols

Hierarchical Model

- It organize data into a tree like structure having one root or parent.
- Here data starts from room with one to many kind of relationship

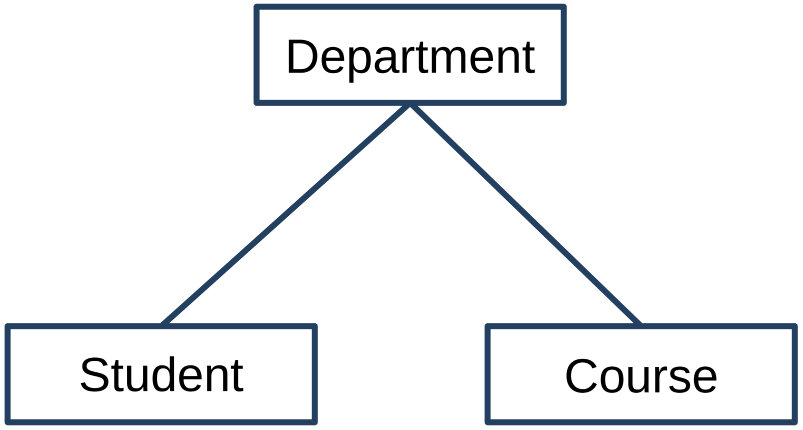


Figure: 1.56 Hierarchical Model

Network Model

- It is an extension of hierarchical model, with many to many kind of relationship in tree structure with multiple parents.

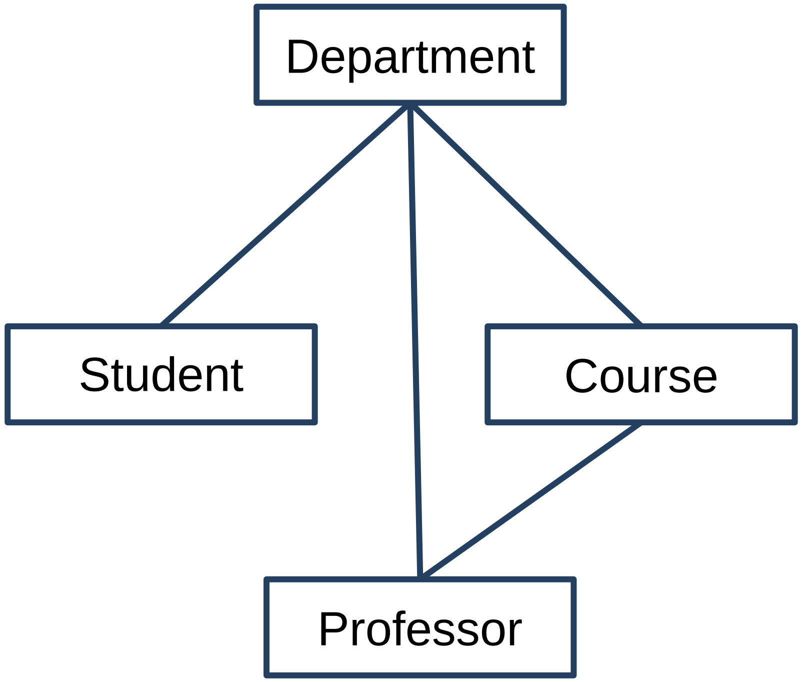


Figure: 1.57 Network Model

Relational Model

- In this model data is organized in two-dimensional tables and relationship is retained by storing a common attribute.
- **Tables:** Relations are saved in the format of tables. Table has rows and columns. Here rows denotes records and columns denotes attributes.
- **Tuple:** A single row of a table which has a single record value is known as tuple .

Relational Model

- **Relation Instance:** It is a finite set of tuples in a relation.
- **Relation schema:** It describes relation name (table name), attributes, and their names.
- **Relation Key:** Each tuple has one more attributes that identifies relation uniquely is known as relation key.
- **Attribute Domain:** It is a predefined value scope of a every attribute.

Relational Model

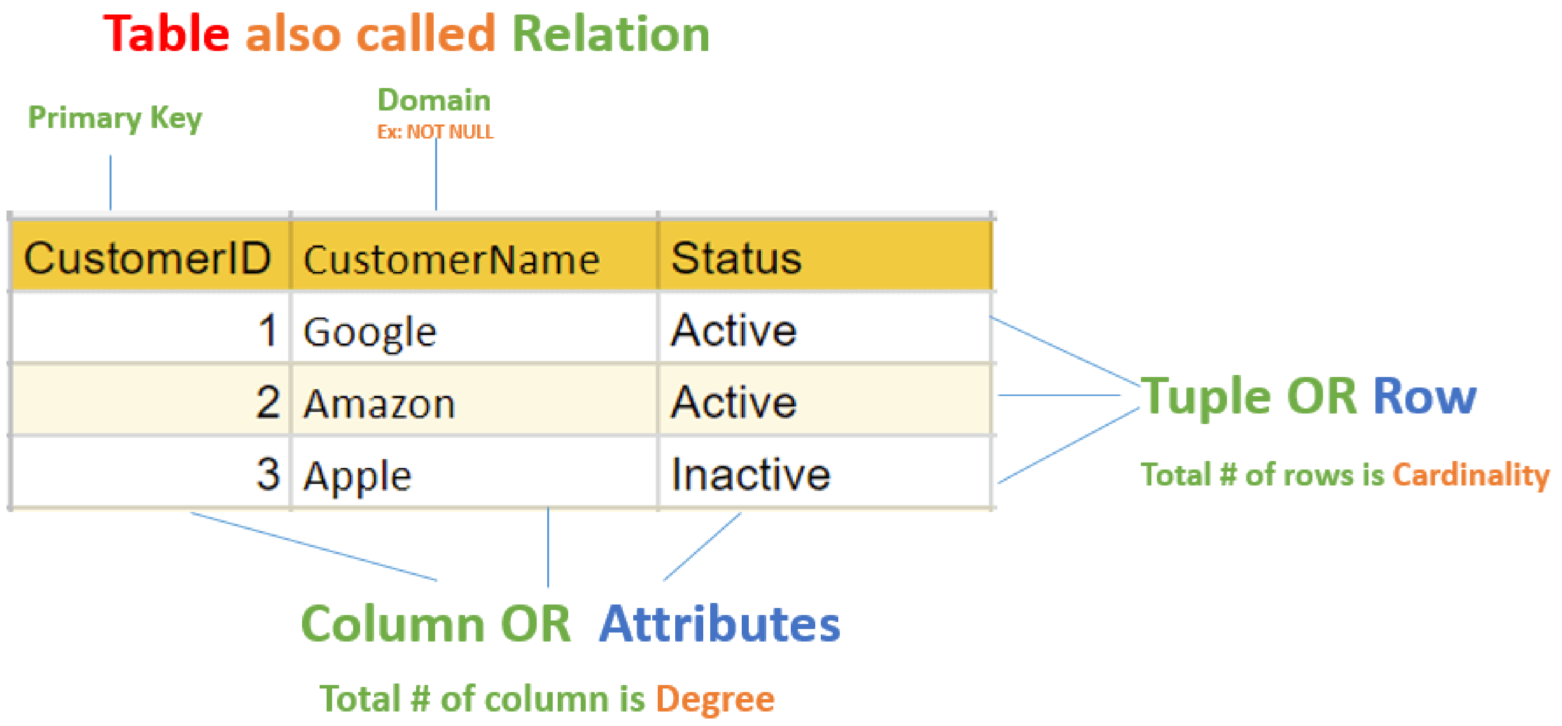


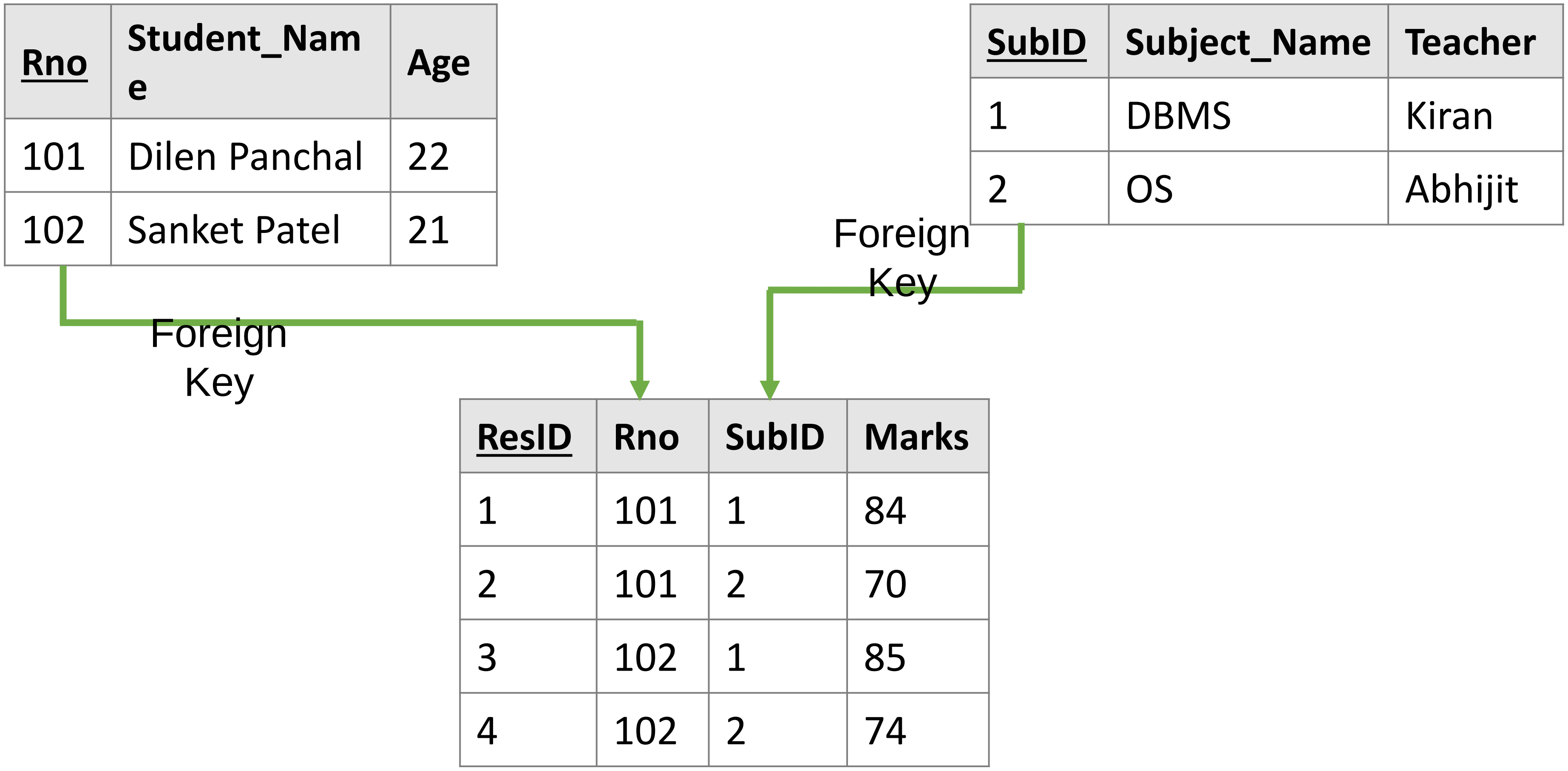
Table also called Relation
Primary Key
Domain
Ex: NOT NULL
Tuple OR Row
Total # of rows is Cardinality
Column OR Attributes
Total # of column is Degree

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

Figure: 1.58 Example Relational Model

(Image Source: https://www.guru99.com/relational-data-model-dbms.html**)**

Relational Model



Rno	Student_Name	Age
101	Dilen Panchal	22
102	Sanket Patel	21

SubID	Subject_Name	Teacher
1	DBMS	Kiran
2	OS	Abhijit

ResID	Rno	SubID	Marks
1	101	1	84
2	101	2	70
3	102	1	85
4	102	2	74

Figure: 1.59 Example Relational Model

Object-Oriented Model

- This model follows the method of representing real world objects.
- Objects are derived from real world entities and situations.
- Each object has their own properties that are defined as attribute and their behavior is defined as methods.
- Object is like an instance of class. So similar attributes and methods are grouped together as a class.

Object-Oriented Model

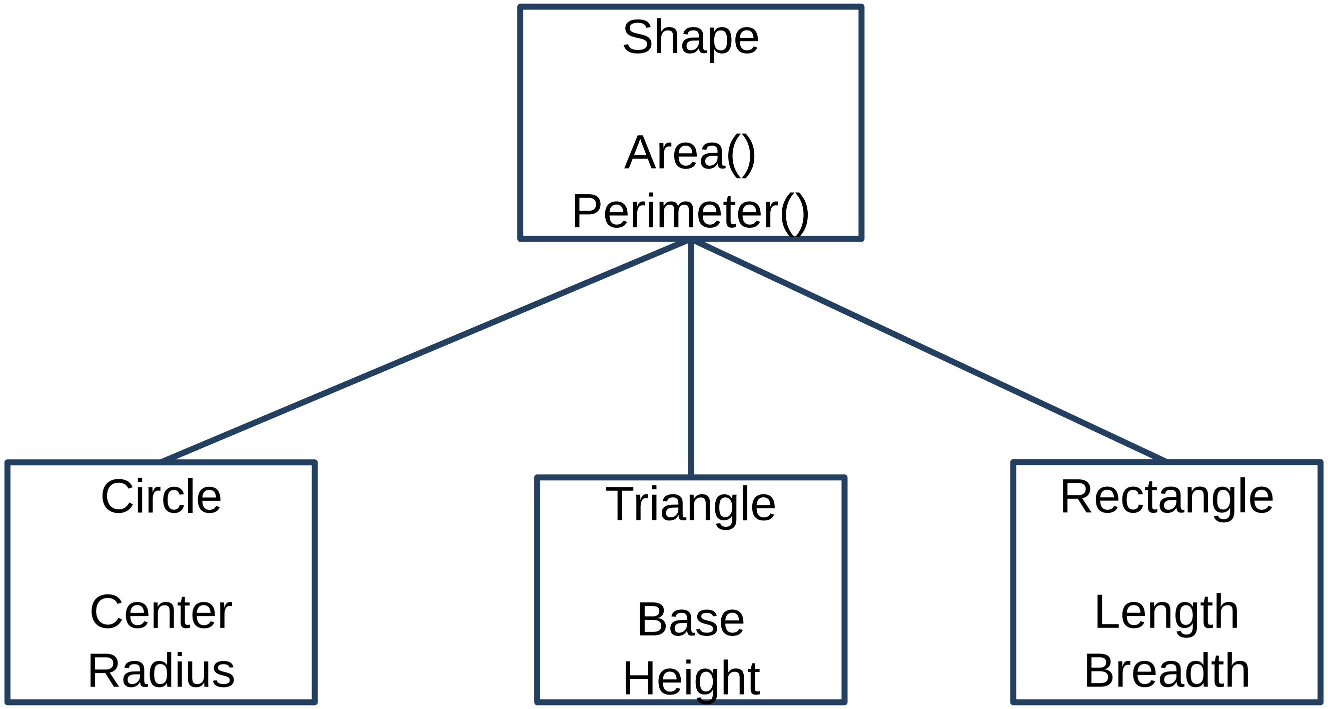


Figure: 1.60 Example Object-Oriented Model

Integrity Constraints

- Integrity constraints means set of rules defined in database system. It is used to maintain the quality of information.
- It ensures that any data related operation performed on database like, data insertion, updating does not affect integrity of data.
- It provide protection against any damage to database.
- Following are different types of Integrity Constraints:
    1. Check
    2. Not null
    3. Unique
    4. Primary key
    5. Foreign key

Integrity Constraints

1. Check:

- Check integrity constraint defines a specific rule for a column, all the rows of that column must have to satisfy it.
- Check controls the tuple values to some set, range or specific value.
- It can be applied on more than one column of a relation(table).
- Example: Tuple value of CGPA should be between 0 to 10

Integrity Constraints

2. Not Null

- As name suggests all the rows of relation(table) should be some definite value for the column on which Not null is applied.
- Simply when we apply this constraint on any column, it should have some value.
- Example: For relation *Student,* Name column should have some value.

Integrity Constraints

3. Unique

- As name defines it make sure that columns or group of columns in each row of table has unique (distinct) value.
- Columns can have null values, but they can’t be duplicated.
- This constraint sometimes also referred as Unique Key.
- Example: For relation *Student,* Enrollmentno column should have unique value.

Integrity Constraints

4. Primary Key

- Primary key applies to column or combination of columns to uniquely identifies each row in the table.
- Primary Key = Unique key + Not null
- There can be only one primary key per table
- Example: For student relation table, **EnrollmentNo** column should have unique value and can't be null.

Primary Key

EnrollNo	Student_Name	DeptID
001	Ramesh	1
002	Suresh	2
003	Mahesh	1

Figure: 1.61 Primary Key

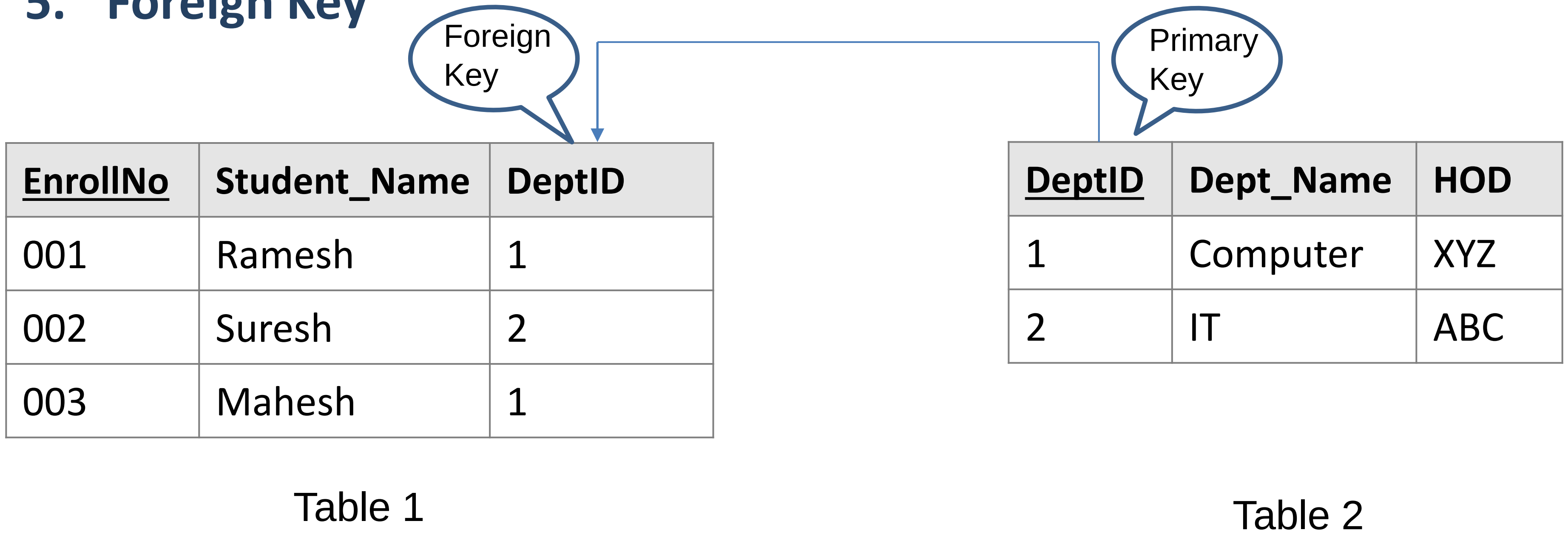
Integrity Constraints

5. Foreign Key

- Foreign key is defined to link two tables(relation).
- Foreign key is set of one or more attributes whose value is derived from the primary key of another relation.
- Foreign key is also known as referential integrity.
- Attribute or column which is declared as foreign key in Table 1 is derived from primary key of Table 2 and every value of foreign key column in table 1 must be null or available in primary key of table 2.

Integrity Constraints

5. Foreign Key



<u>EnrollNo</u>	Student_Name	DeptID
001	Ramesh	1
002	Suresh	2
003	Mahesh	1

<u>DeptID</u>	Dept_Name	HOD
1	Computer	XYZ
2	IT	ABC

Figure: 1.62 Example Foreign Key

Some other keys

- **Super Key:** It is a set of one or more columns that identifies each tuple record uniquely in a relation.

Employee

Emp_SSN	Emp_Number	Emp_Name
123	101	Ramesh
456	102	Suresh
789	103	Dinesh
791	104	Mahesh

Figure: 1.63 Super Key

- Here entity sets {Emp_SSN}, {Emp_Number}, {Emp_SSN, Emp_Number}, {Emp_SSN, Emp_Name}, {Emp_SSN, Emp_Number, Emp_Name}, {Emp_Number, Emp_Name} are uniquely identifies rows so they all are super keys.

Some other keys

- ➢ **Candidate Key:** It is subset or part of the super key.
- Sometimes they’re also known as minimal super key with no repeated attributes.
- Each table has minimum one candidate key, but there can be multiple candidate keys also.
- One Primary key is selected from candidate keys.
- Example: {Emp_SSN} , {Emp_Number}
- ➢ **Alternate Key:** Any candidate key that’s not chosen as a primary key is alternate key.

References

[1] Abraham Silberschatz, Henry F. Korth and S. Sudarshan, Database System Concepts, McGraw-Hill Education (Asia), Seventh Edition, 2019.

[2] C. J. Date, A. Kannan and S. Swamynathan, An Introduction to Database Systems, Pearson Education, Eighth Edition, 2009.

[3] Database Management Systems, CSE, DIET, https://www.darshan.ac.in/DIET/CE/GTU-Computer-Engineering-Study-Material

[4] Database management systems by Raghu Ramakrishnan and Johannes Gehrke http://pages.cs.wisc.edu/~dbbook/openAccess/thirdEdition/slides/slides3ed.html

[5] Database management system tutorial, https://www.tutorialspoint.com/dbms/index.htm