Role and Duties of Database Administrator (DBA)

Decides hardware

Manages data integrity and security

Database design

Database implementation

Query processing performance

DBMS Architecture

Types of DBMS Architecture

There are three types of DBMS architecture:

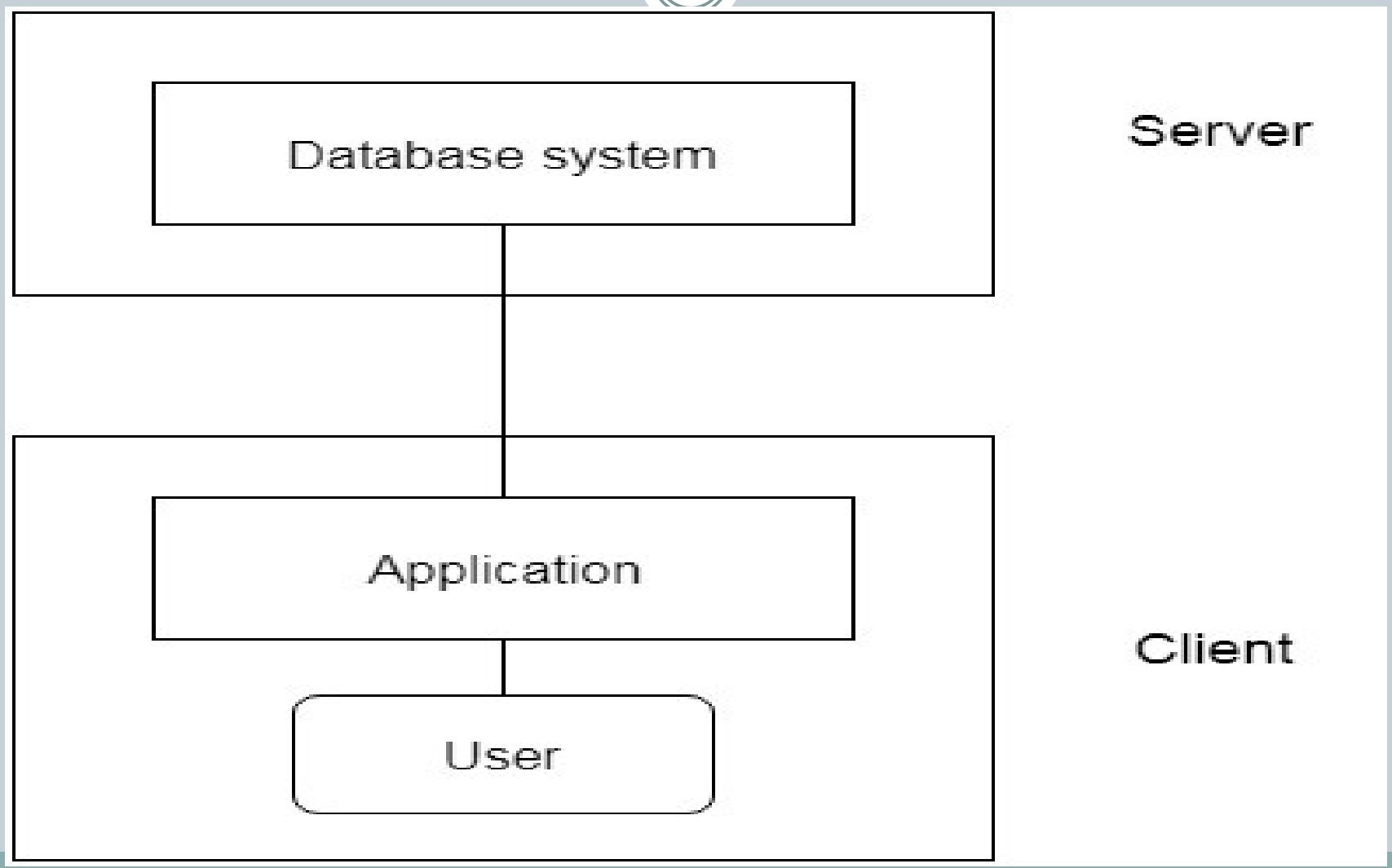
1. Single tier architecture
2. Two tier architecture
3. Three tier architecture

1. Single tier architecture

In this type of architecture, the database is readily available on the client machine, any request made by client doesn't require a network connection to perform the action on the database.

Example –Own System

2. Two tier architecture



2. Two tier architecture

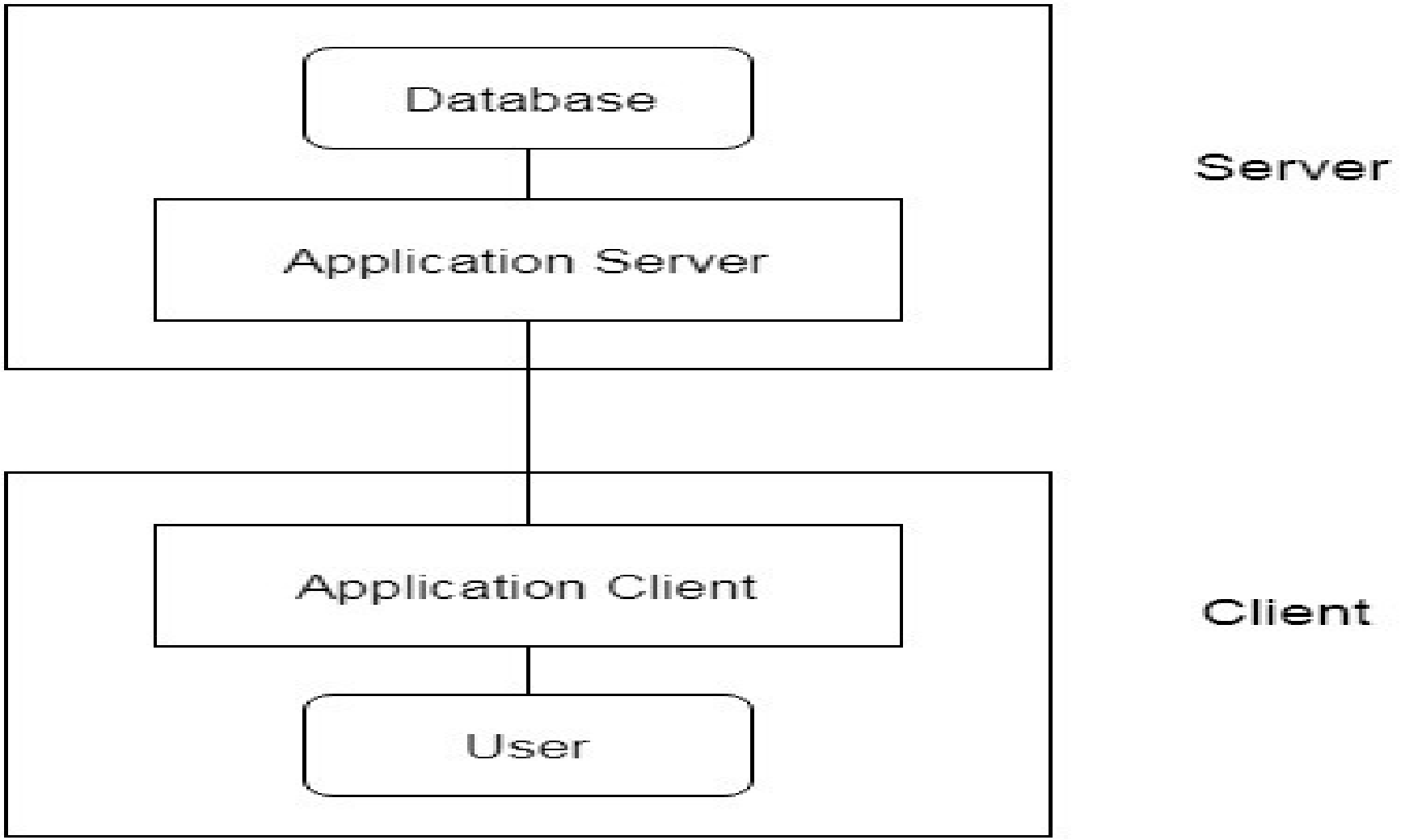
The 2-Tier architecture is same as basic client-server.

In the two-tier architecture, applications the client end can directly communicate with on the database at the server side. For this interaction, API's like:**ODBC**,**JDBC**are used.

ODBC =Open Database Connectivity

JDBC = Java Database Connectivity

3. Three tier Architecture



3. Three tier Architecture

The 3-Tier architecture contains another layer between the client and server.

In this architecture, client can't directly communicate with the server.

The application on the client-end interacts with an application server which further communicates with the database system.

Types of Database Users

Database users are the one who really use and take the benefits of database.

There are different types of users depending on their need and way of accessing the database.

1. **Application Programmers**
2. **Sophisticated Users**
3. **Specialized Users**
4. **Naive Users**

.

Application Programmers

Application Programmers –They are the developers who interact with the database by means of DML queries.

These DML queries are written in the application programs like C, C++,JAVA, Pascal etc.

These queries are converted into object code to communicate with the database.

For example, writing a C program to generate the report of employees who are working in particular department will involve a query to fetch the data from database. It will include a embedded SQL query in the C Program.

Sophisticated Users

They are database developers, who write SQL queries to select/insert/delete/update data.

They do not use any application or programs to request the database. .

They directly interact with the database by means of query language like SQL. These users will be scientists, engineers, analysts who thoroughly study SQL and DBMS to apply the concepts in their requirement.

In short, we can say this category includes designers and developers of DBMS and SQL.

Specialized Users

These are also sophisticated users, but they write special database application programs.

They are the developers who develop the complex programs to the requirement.

Naive Users

These are the users who use the existing application to interact with the database.

For example, online library system, ticket booking systems, ATMs etc. which has existing application and users use them to interact with the database to fulfill their requests

Summary of Database Users

Users are differentiated by the way they expect to interact with the system

Application programmers –interact with system through DML calls

Sophisticated users –form requests in a database query language

Specialized users –write specialized database applications that do not fit into the traditional data processing framework

E.g. people accessing database over the web, bank tellers, clerical staff

Naïve users –invoke one of the permanent application programs that have been written previously

Data Dictionary

A Data Dictionary contains metadata i.e data about the database.

The data dictionary is very important as it contains information such as what is in the database, who is allowed to access it, where is the database physically stored etc.

The users of the database normally don't interact with the data dictionary, it is only handled by the database administrators.

Contents of Data Dictionary

Names of all the database tables and their schemas.

Details about all the tables in the database, such as their owners, their security constraints, when they were created etc.

Physical information about the tables such as where they are stored and how.

Table constraints such as primary key attributes, foreign key information etc.

Information about the database views that are

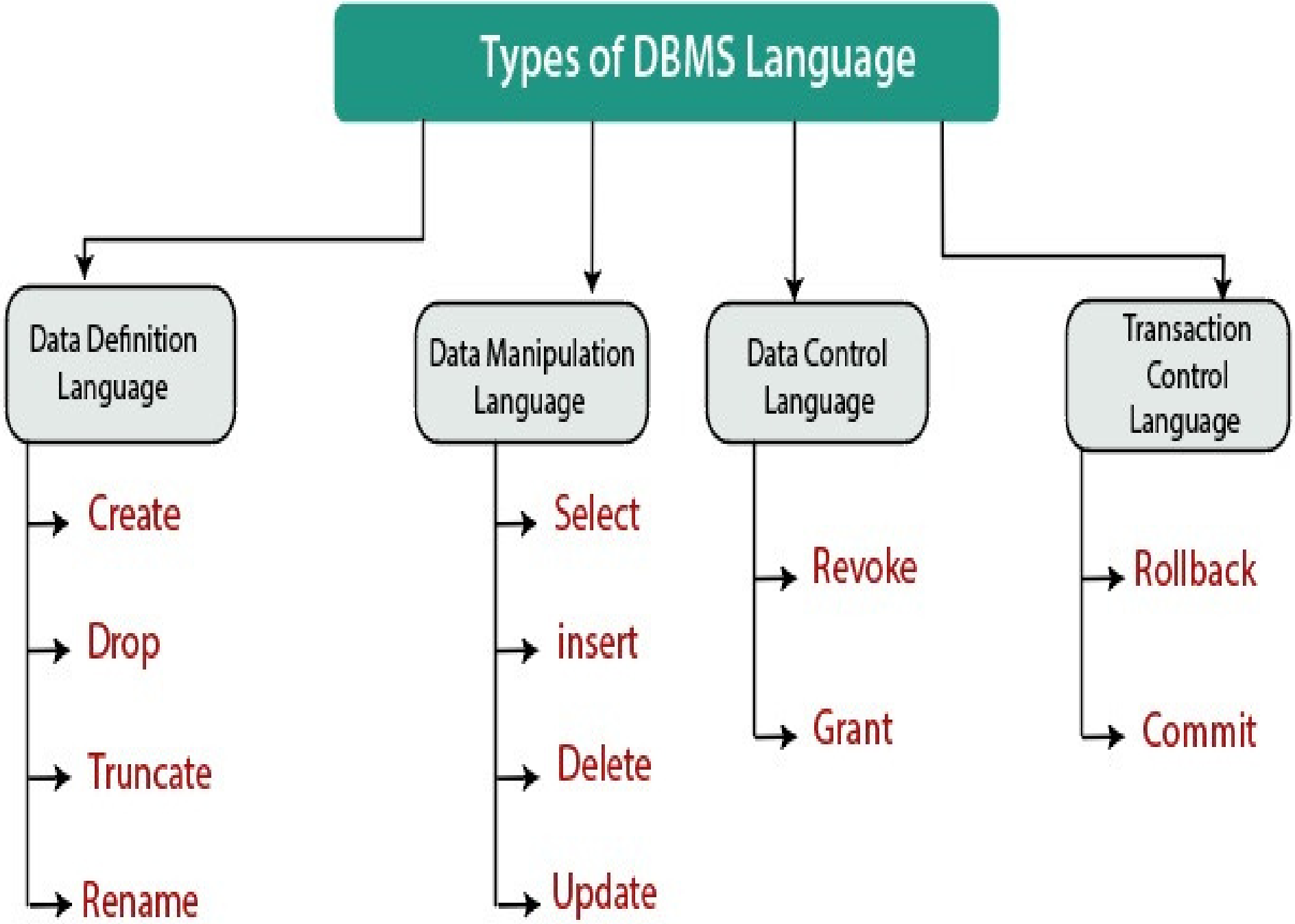
This is a Data Dictionary describing a table that contains employee details.

Field Name	Data Type	Field Size for display	Description	Example
Employee Number	Integer	10	Unique ID of each employee	1645000001
Name	Text	20	Name of the employee	David Heston
Date of Birth	Date/Time	10	DOB of Employee	08/03/1995
Phone Number	Integer	10	Phone number of employee	6583648648

Database Language

A DBMS has appropriate languages and interfaces to express database queries and updates.

Database languages can be used to read, store and update the data in the database.


Types of DBMS Language
Data Definition Language
Data Manipulation Language
Data Control Language
Transaction Control Language
Create
Drop
Truncate
Rename
Select
insert
Delete
Update
Revoke
Grant
Rollback
Commit

1. Data Definition Language

DDLstands for**D**ata**D**efinition**L**anguage. It is used to define database structure or pattern.

It is used to create schema, tables, indexes, constraints, etc. in the database.

Using the DDL statements, you can create the structure of the database.

Data definition language is used to store the information of metadata like the number of tables and schemas, their names, indexes, columns in each table, constraints, etc.

Commands under DDL

Create: It is used to create objects in the database.

Alter: It is used to alter the structure of the database.

Drop: It is used to delete objects from the database.

Truncate: It is used to remove all records from a table.

Rename: It is used to rename an object.

These commands are used to update the database schema that's why they come under Data definition language.

2. Data Manipulation Language

DMLstands for**D**ata**M**anipulation**L**anguage.

It is used for accessing and manipulating data in a database.

It handles user requests.

Commands under DML

Select: It is used to retrieve data from a database.

Insert: It is used to insert data into a table.

Update: It is used to update existing data within a table.

Delete: It is used to delete all records from a table.

3. Data Control Language

DCLstands for**D**ata**C**ontrol**L**anguage. It is used to retrieve the stored or saved data.

The DCL execution is transactional. It also has rollback parameters.

Commands under DCL

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

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

4. Transaction Control Language

TCL is used to run the changes made by the DML statement.

TCL can be grouped into a logical transaction.

Commands under TCL

Commit: It is used to save the transaction on the database.

Rollback: It is used to restore the database to original since the last Commit.

Database Model

Database model is a logical frame in which data is stored.

The model also describes the relationship between different parts of the data.

Logical vs Physical (ex-Cafeteria)

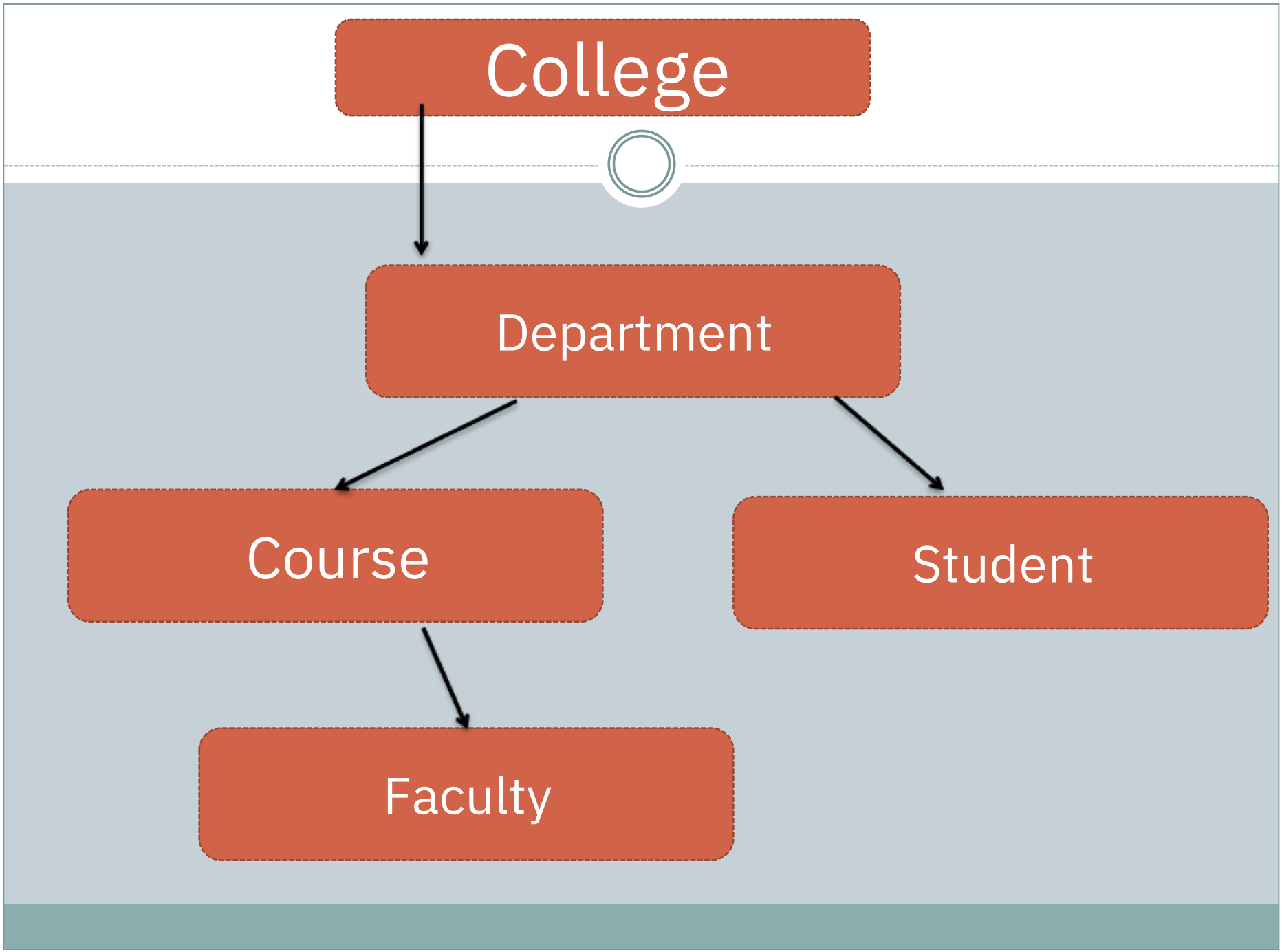
Types of Data Model

1. Hierarchical Data Model
2 Network Model
. Relational Data Model
3
.

Hierarchical Data Model

In this model each entity has only one parent but can have several children.

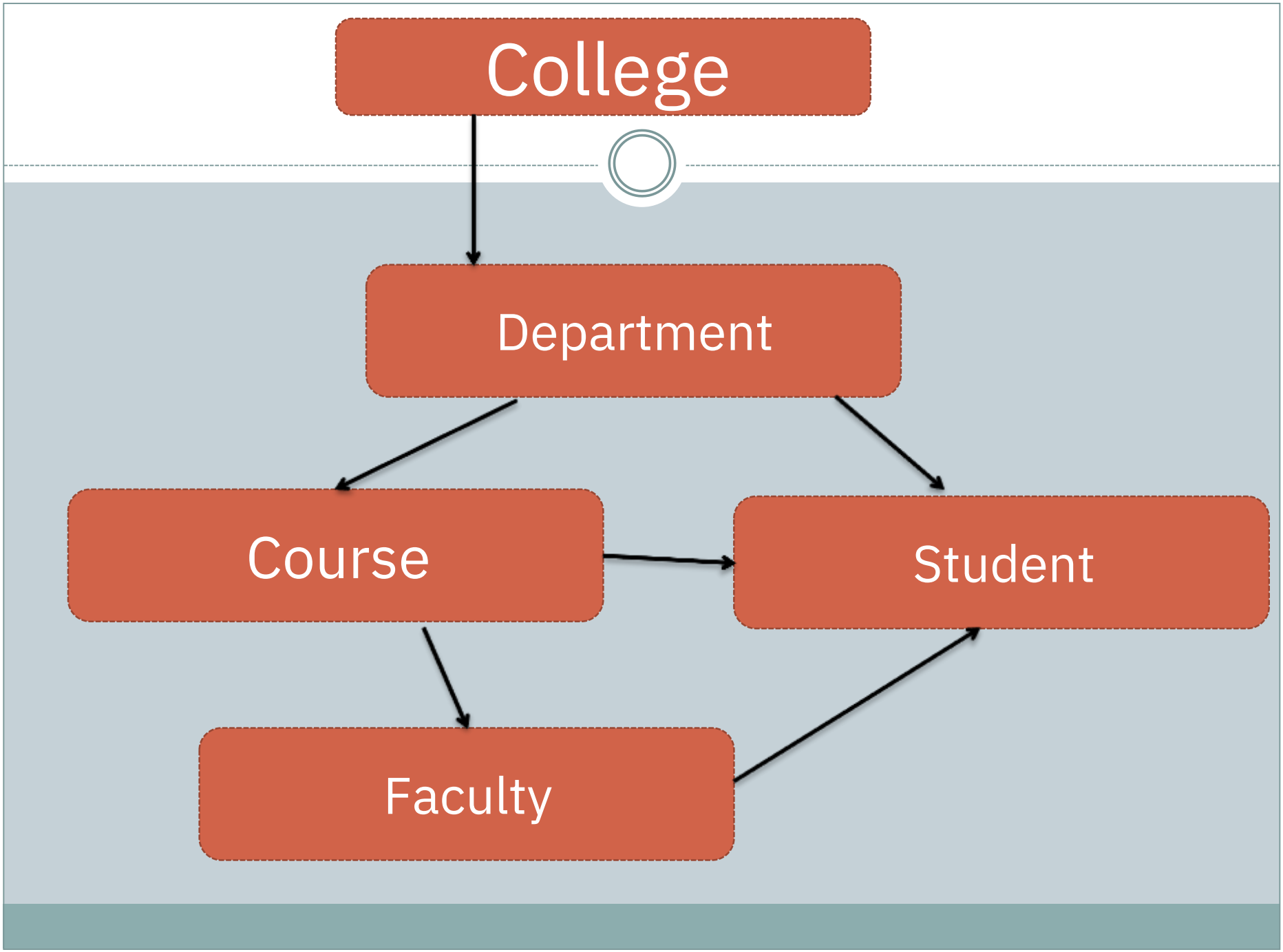
At the top of the hierarchy there is only one entity which is called Root. Example - College


College
Department
Course
Student
Faculty

Network Model

In this model, Entities are organized in a graph, in which entities can be accessed using several paths.

There may be multiple relationship between entities.


College
Department
Course
Student
Faculty

Relational Model

In this model, data is organized in the form of two dimension tables called Relations.

Here we have entities, relationship between entities.

Relational Model

Example of tabular data in the relational model

Attributes

Customer-id	*customer-name*	*customer-street*	*customer-city*	*account-number*
192-83-7465	Johnson	Alma	Palo Alto	A-101
019-28-3746	Smith	North	Rye	A-215
192-83-7465	Johnson	Alma	Palo Alto	A-201
321-12-3123	Jones	Main	Harrison	A-217
019-28-3746	Smith	North	Rye	A-201

A Sample Relational Database

customer-id	*customer-name*	*customer-street*	*customer-city*
192-83-7465	Johnson	12 Alma St.	Palo Alto
019-28-3746	Smith	4 North St.	Rye
677-89-9011	Hayes	3 Main St.	Harrison
182-73-6091	Turner	123 Putnam Ave.	Stamford
321-12-3123	Jones	100 Main St.	Harrison
336-66-9999	Lindsay	175 Park Ave.	Pittsfield
019-28-3746	Smith	72 North St.	Rye

(a) The *customer* table

account-number	*balance*
A-101	500
A-215	700
A-102	400
A-305	350
A-201	900
A-217	750
A-222	700

(b) The *account* table

customer-id	*account-number*
192-83-7465	A-101
192-83-7465	A-201
019-28-3746	A-215
677-89-9011	A-102
182-73-6091	A-305
321-12-3123	A-217
336-66-9999	A-222
019-28-3746	A-201

(c) The *depositor* table

Distributed Database

A **Distributed Database** is a collection of multiple interconnected databases, which are physically across various locations that communicate

spread via a computer network.

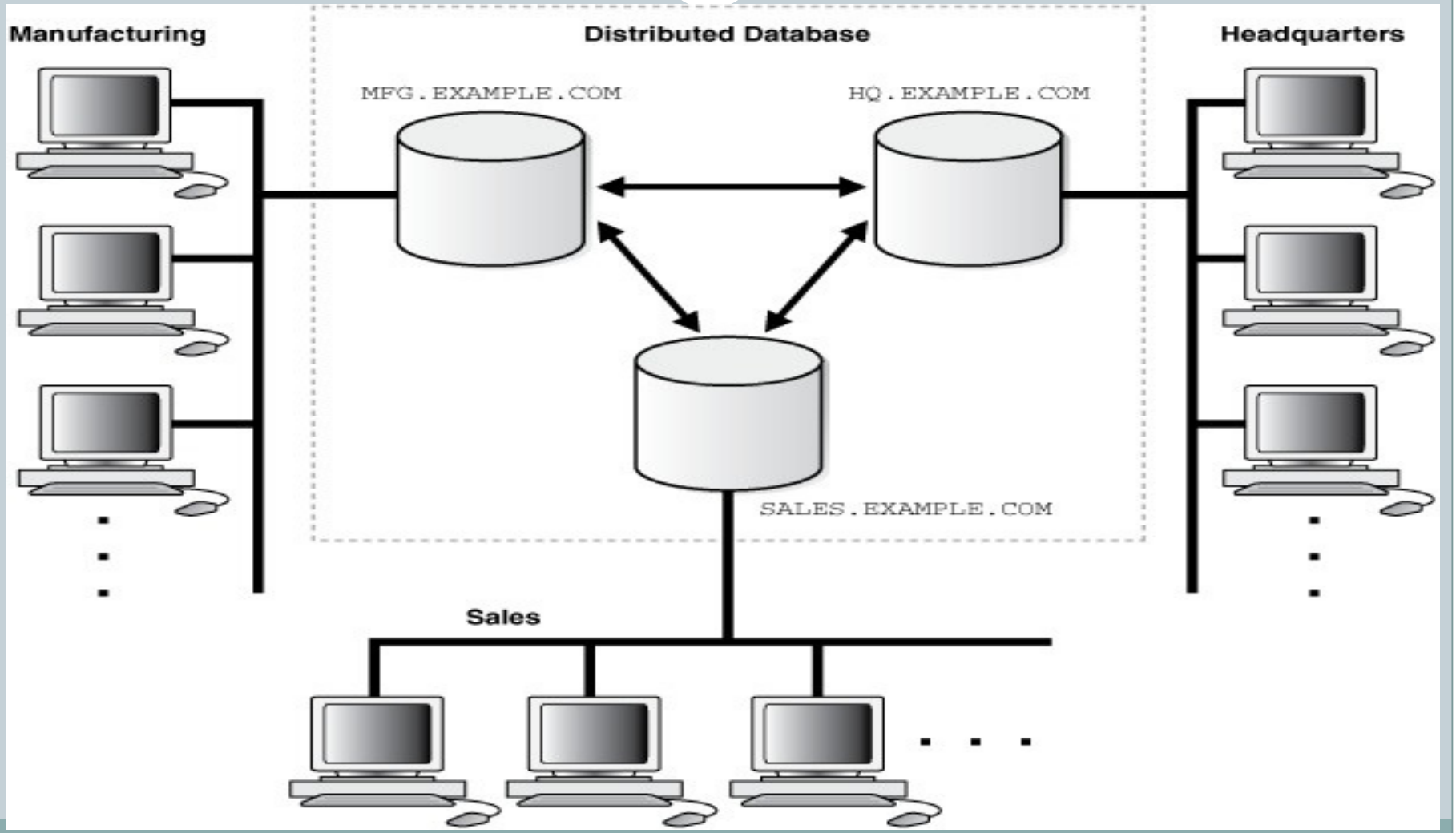
Features

Databases in the collection are logically interrelated with each other. Often they represent a single logical database.

in each site can be managed by a DBMS independent

Data is physically stored across multiple sites. Data of the other sites.

Distributed Database



Advantages

Modular Development

More Reliable

Better Response

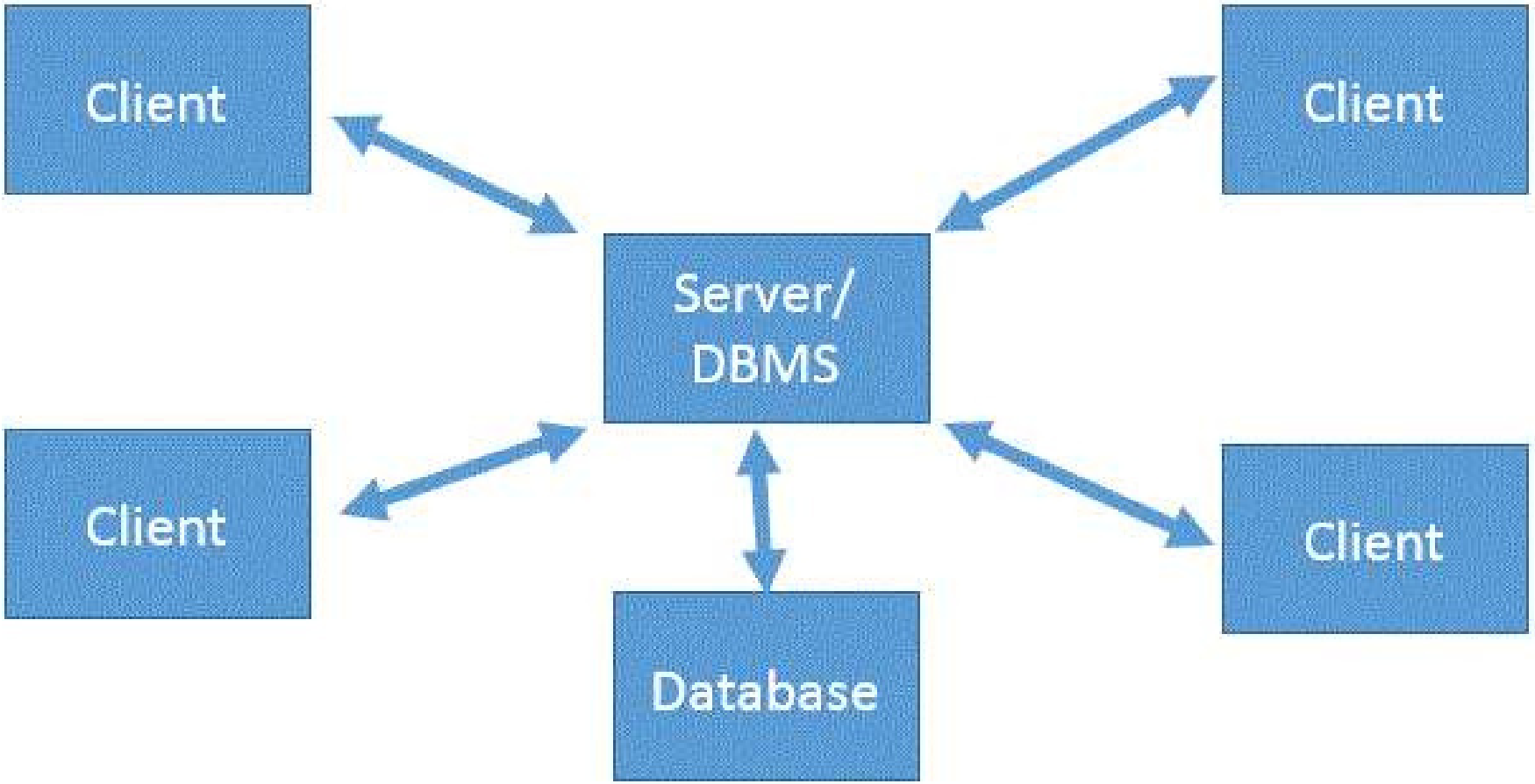
Lower Communication Cost

Client-Server Database

Client-server architecture,**architecture**of a computer network in which many**clients**(remote processors) request and receive service from a centralized**server**(host computer).

Clientcomputers provide an interface to allow a user to request services of the **server** and to display the results the **server** returns.

Client-Server Database



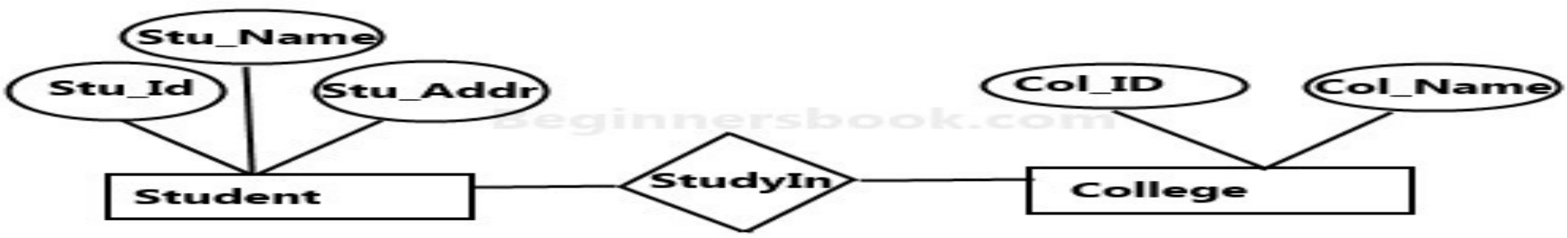
Advantages of Client/Server Database System

- Client/Server system has less expensive platforms to support applications that had previously been running only on large and expensive mini or mainframe computers.
- Client offer icon-based menu-driven interface, which is superior to the traditional command-line, dumb terminal interface typical of mini and mainframe computer systems.
- Client/Server environment facilitates in more productive work by the users and making better use of existing data

ER Model -Basic Concepts

An **Entity–relationship model (ER model)** describes the structure of a database with the help of a diagram, which is known as **Entity Relationship Diagram (ER Diagram)**.

An ER model is a design or blueprint of a database that can later be implemented as a database.



Sample E-R Diagram

Entity

An entity can be a real-world object, that can be easily identifiable.

For example, in a school database, students, teachers, classes, and courses offered can be considered as entities.

All these entities have some attributes or properties that give them their identity.

An entity set is a collection of similar types of entities.

Example –Group of Students.

Attributes

Entities are represented by means of their properties, called **attributes**. All attributes have values.

For example, a student entity may have name, class, and age as attributes.

Types of Attributes

1. **Simple attribute**
2. **Composite attribute**
3. **Derived attribute**
4. **Single-value attribute**
5. **Multi-value attribute**

Relationship

The association among entities is called a relationship.

For example, an employee **works at** a department, a student **enrolls** in a course. Here, Works at and Enrolls are called relationships.

Relationship Set

A set of relationships of similar type is called a relationship set. Like entities, a relationship too can have attributes.

Mapping Cardinalities

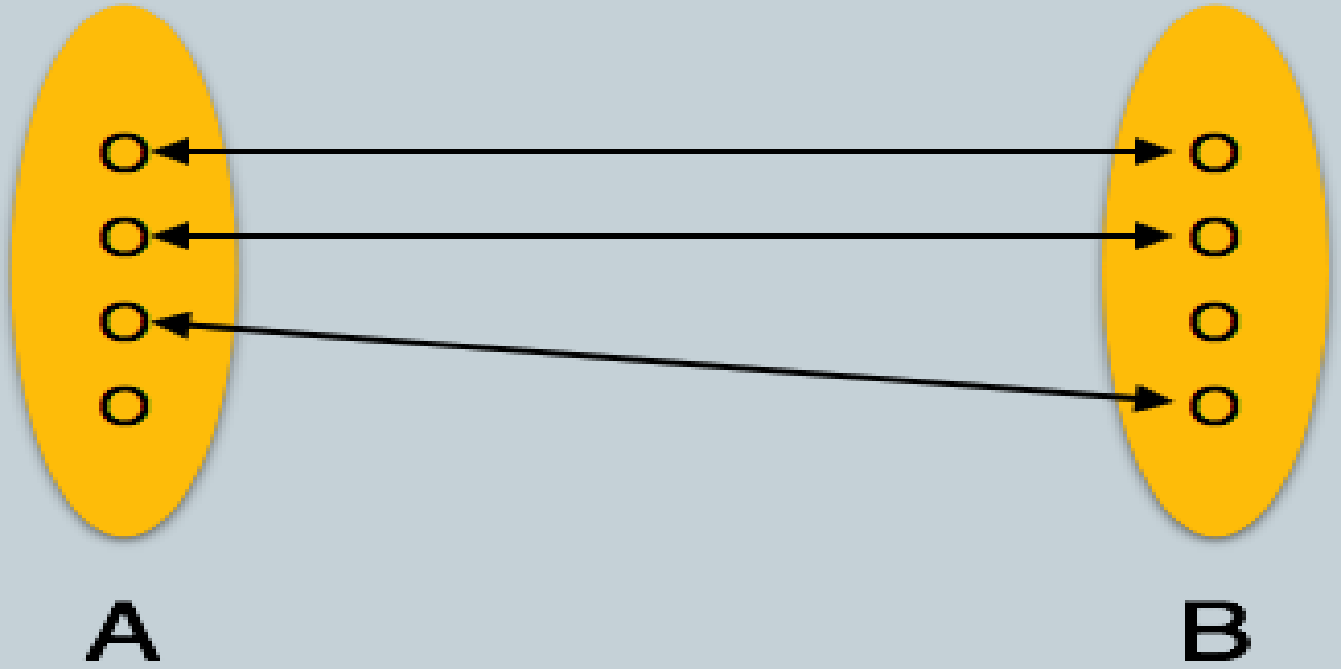
Cardinality defines the number of entities in one entity set, which can be associated with the number of entities of other set via relationship set.

There are Four Types of Mapping Cardinalities

1. One to One
2. One-to-many
3. Many-to-one
4. Many-to-many

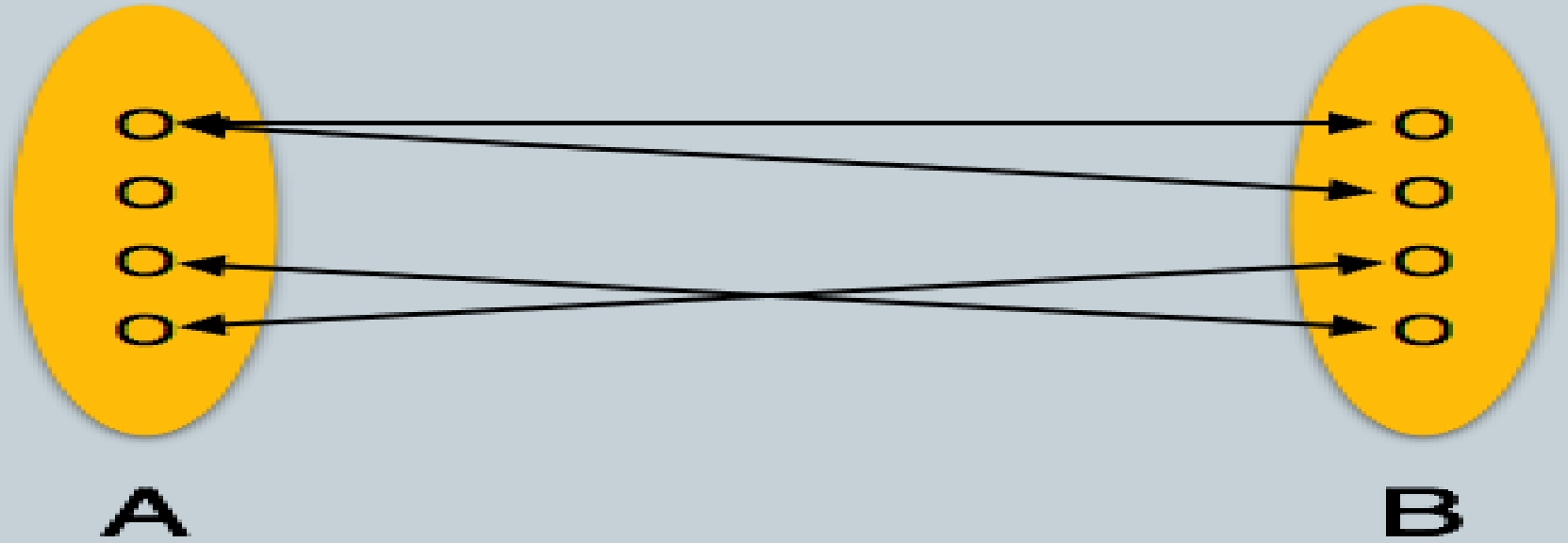
One-to-one Relationship

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



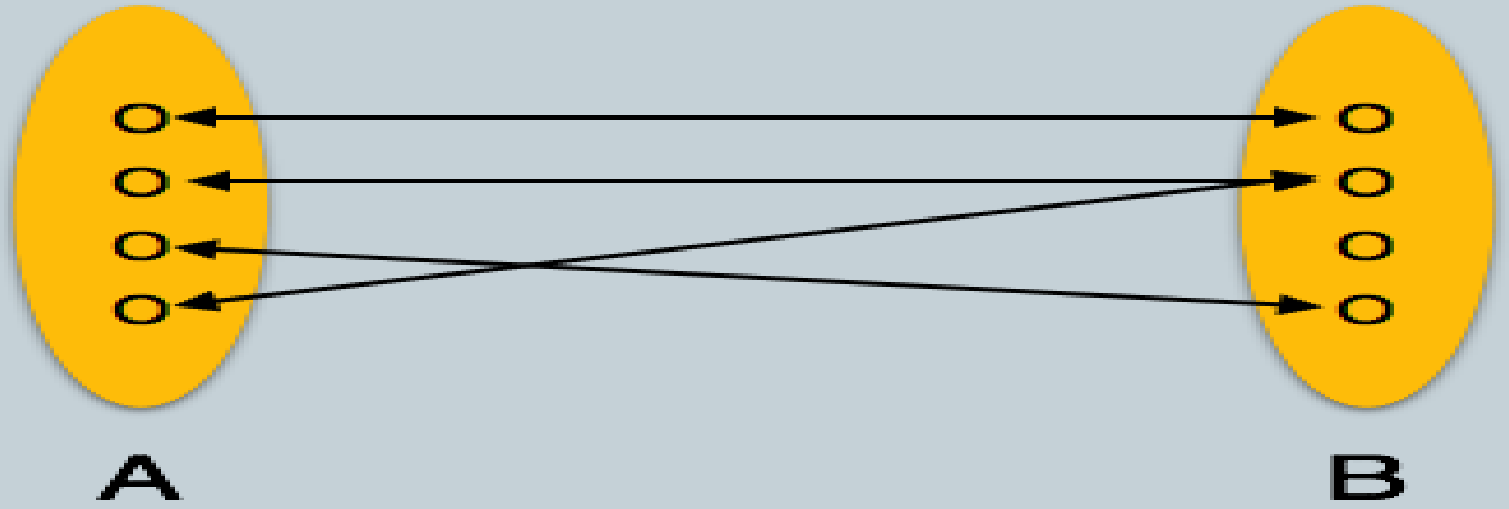
One-to-many Relationship

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.



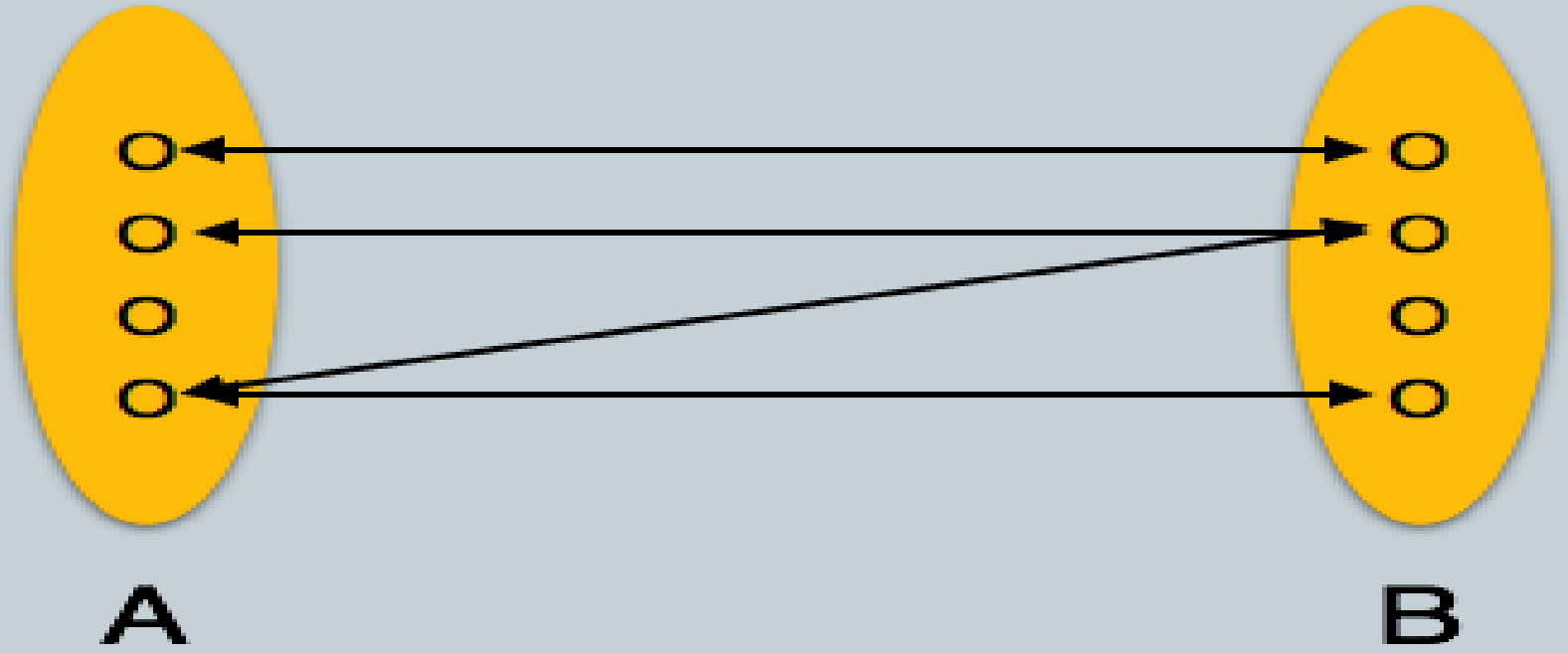
Many-to-one Relationship

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.



Many-to-many Relationship

Many-to-many– One entity from A can be associated with more than one entity from B and vice versa.



Entity Relationship diagram(ER Diagram)

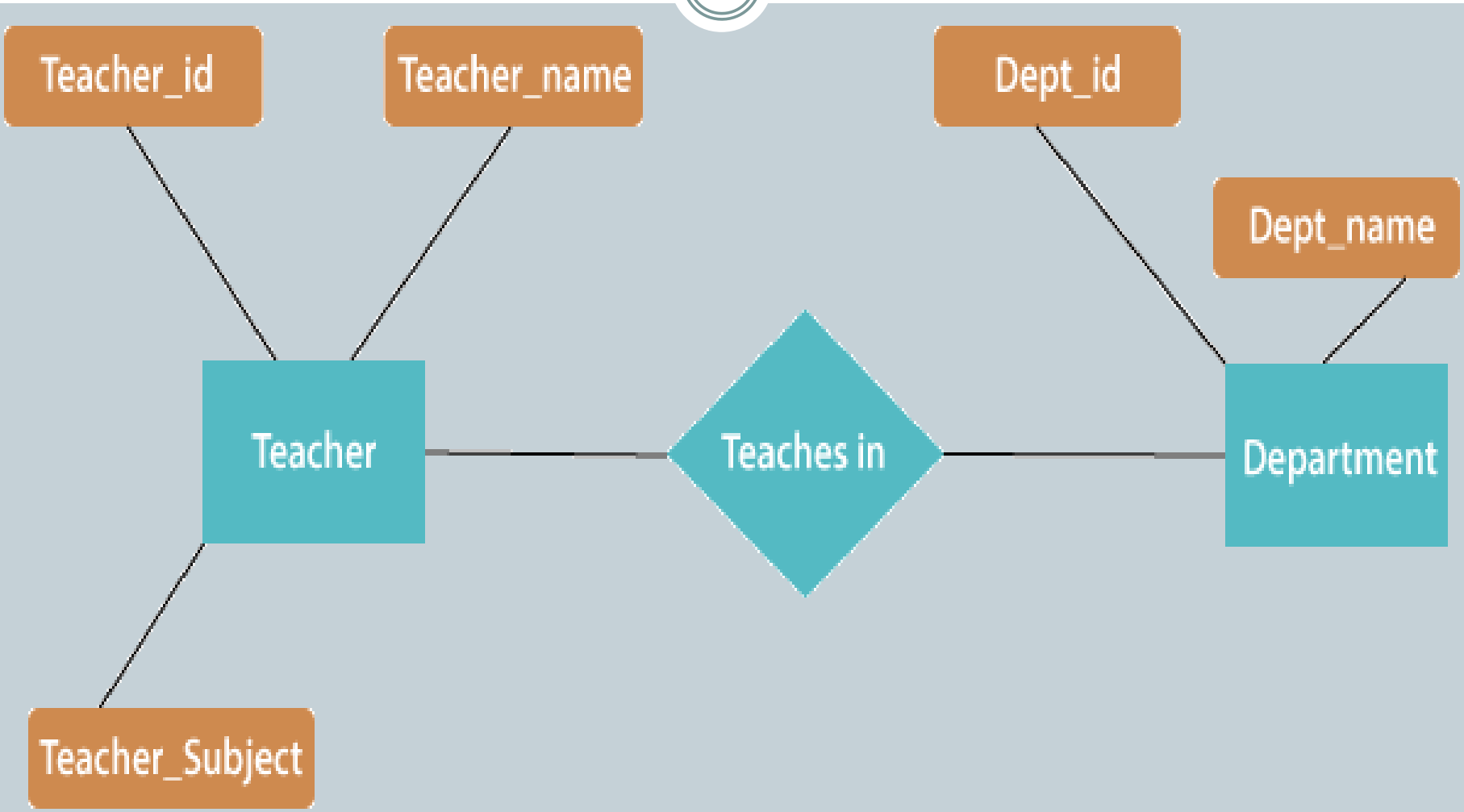
Introduced by Dr Peter Chen in 1976.

It is a Non-technical design method works on conceptual level based on the perception of real world.

Consists of entities, attributes and relationships.

Basically it is a diagrammatic representation of database.

Entity-Relationship Model



ER Diagram Representation

1. Entity

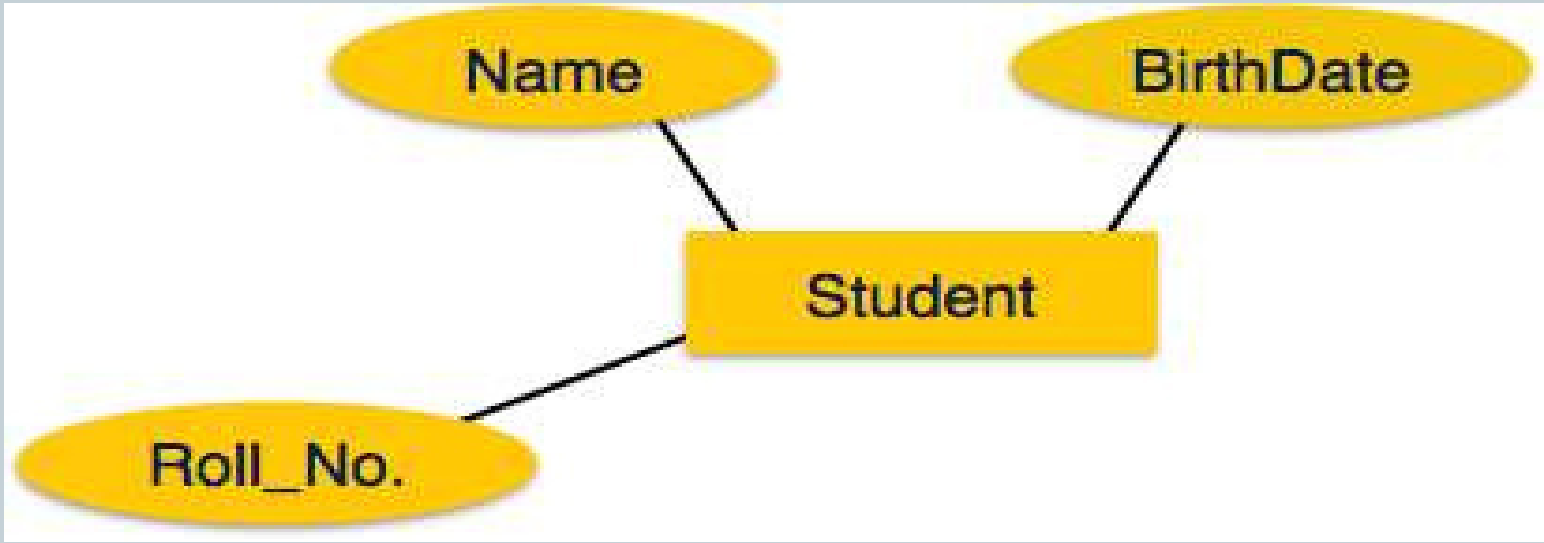
Entities are represented by means of rectangles.

Rectangles are named with the entity set they represent.

2. Attributes

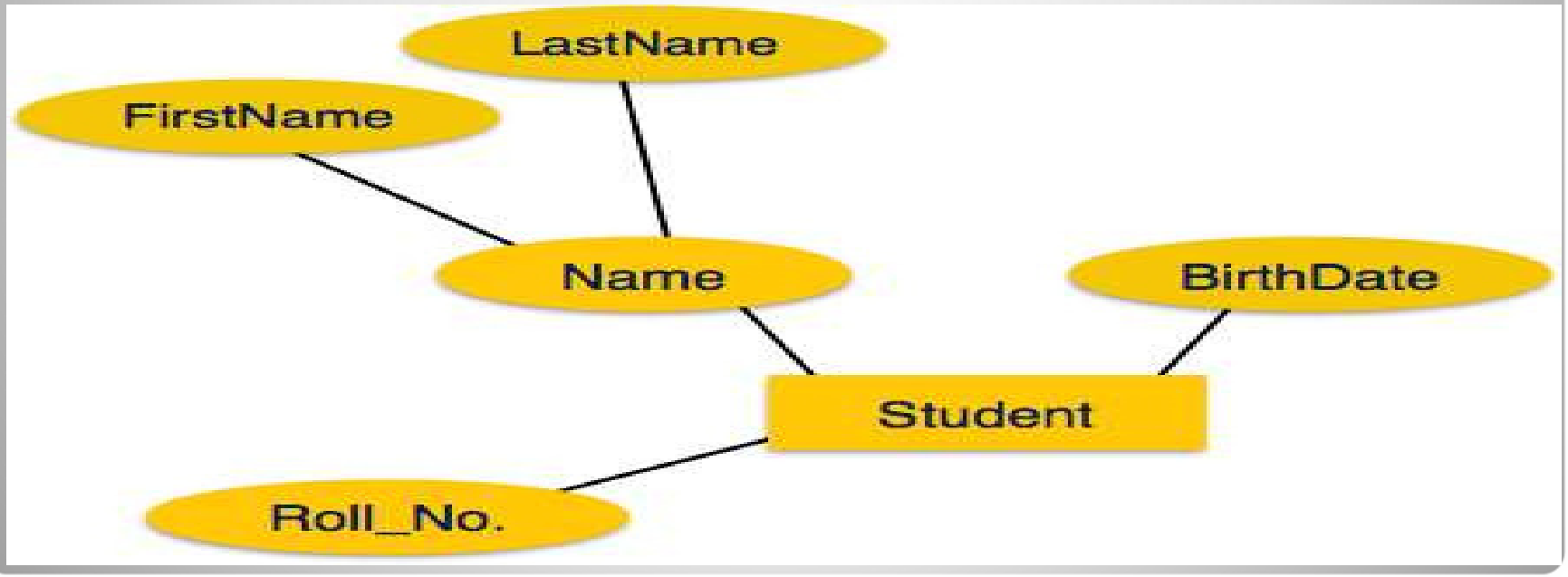
Attributes are the properties of entities. Attributes are represented by means of ellipses.

Every ellipse represents one attribute and is directly connected to its entity (rectangle).



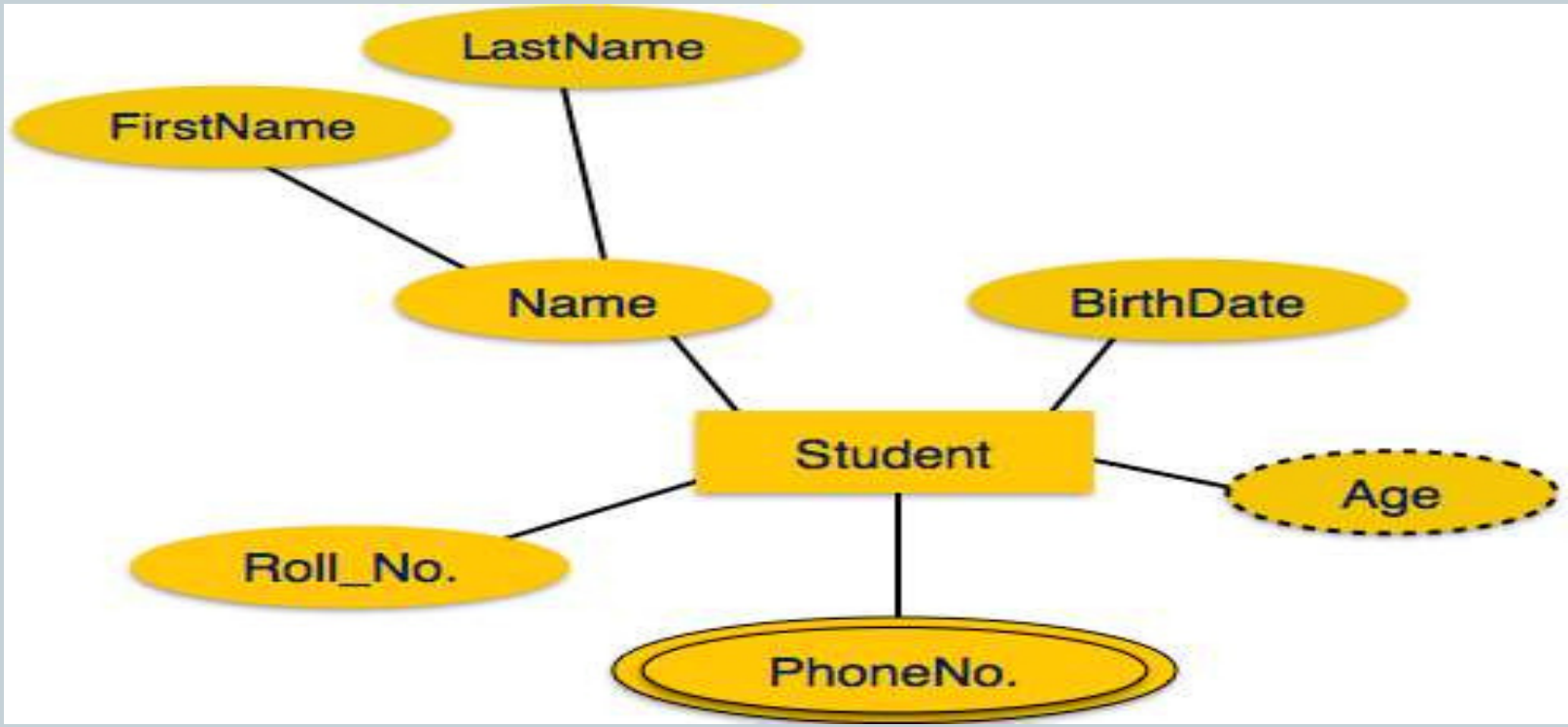
If the attributes are**composite**, they are further divided in a tree like structure. Every node is then connected to its attribute.

That is, composite attributes are represented by ellipses that are connected with an ellipse.

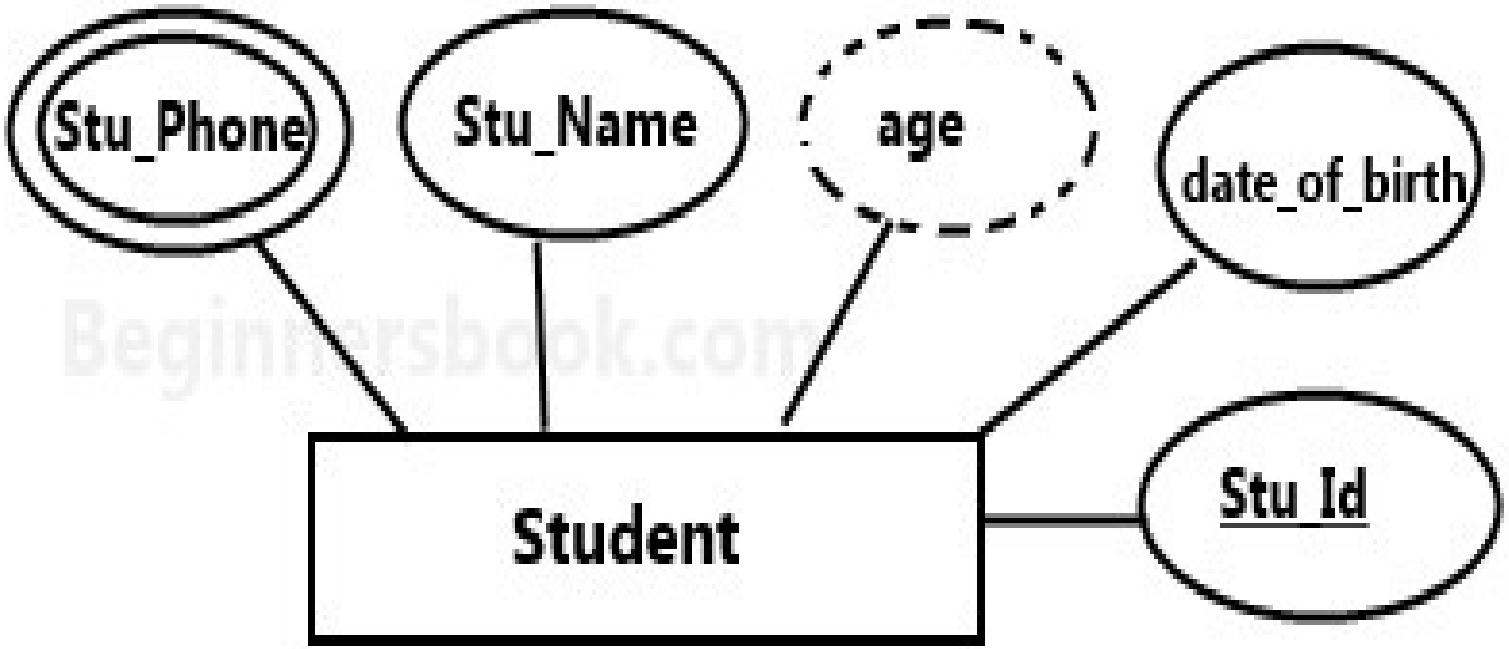


Derived attributes are depicted by dashed ellipse.

Multivalued attributes are depicted by double ellipse



Different Attributes



Relationship

Relationships are represented by diamond-shaped box. Name of the relationship is written inside the diamond-box.

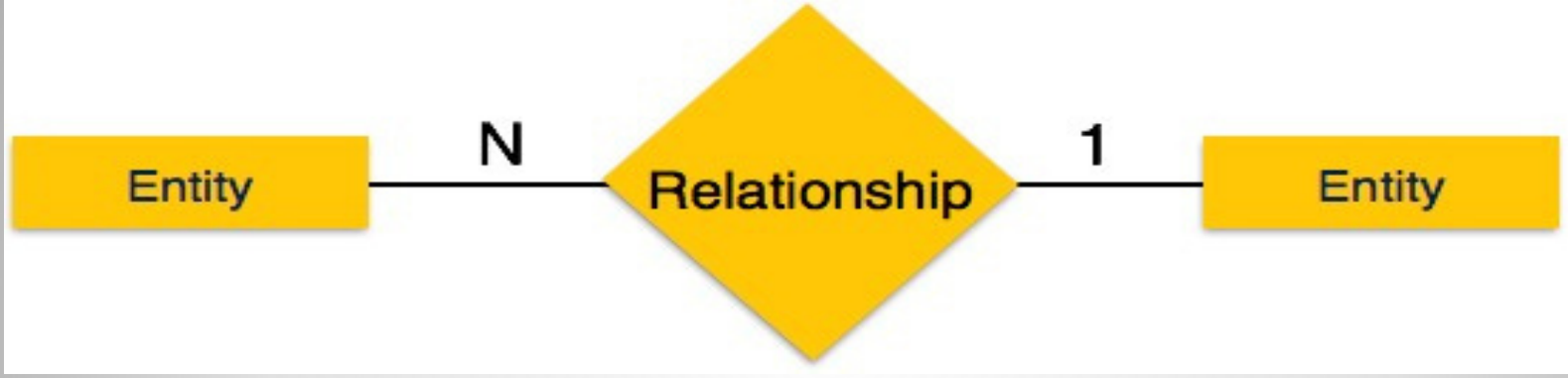
All the entities (rectangles) participating in a relationship, are connected to it by a line.

Types Of Relationship

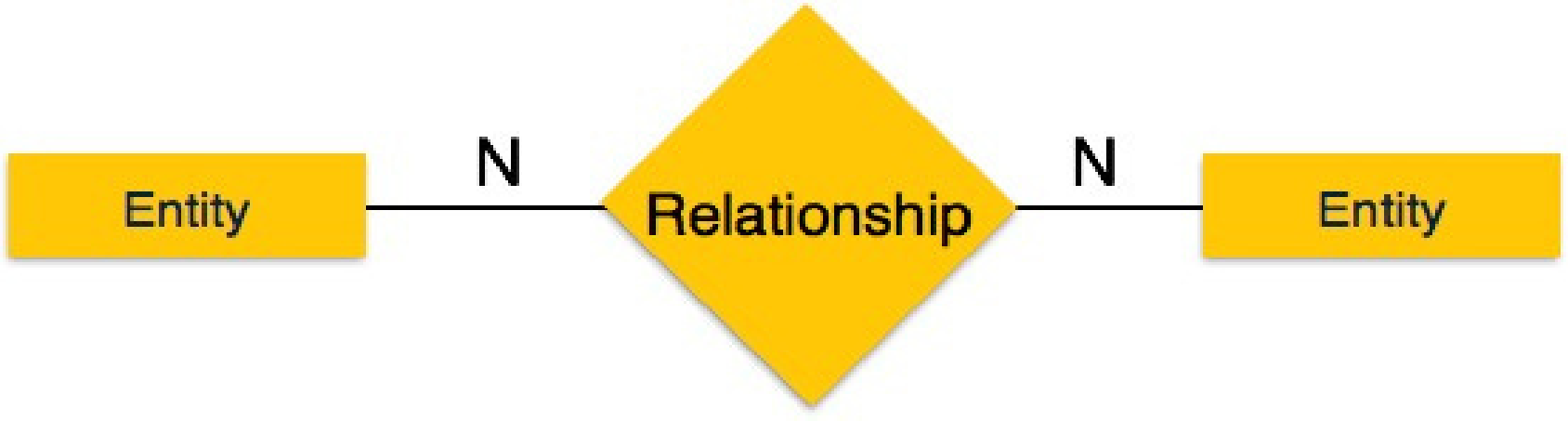
One to One

One to Many

Many to One



Many to Many



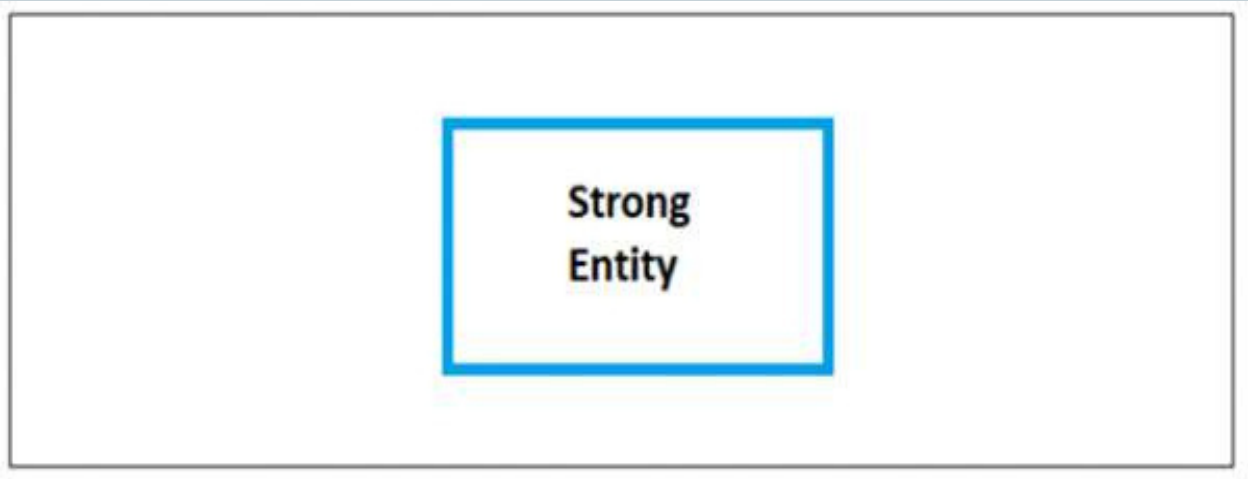
Strong and Weak Entity Types

1. Strong Entity

The strong entity has a primary key.

Strong Entity is represented by a single rectangle

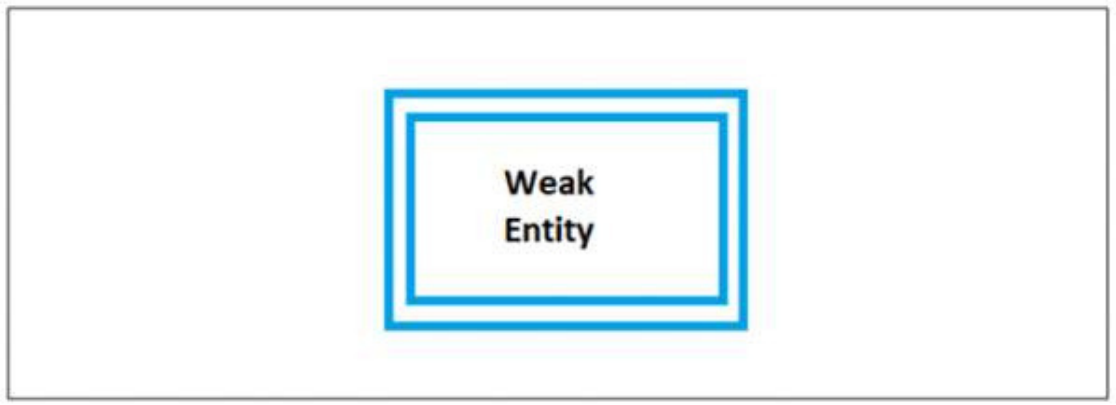
In Professor table we have **Professor_Name, Professor _ID and Professor _Salary**are attributes so **Professor_ID**is the primary key



2. Weak Entity

The weak entity in DBMS do not have a primary key and are dependent on the parent entity. It mainly depends on other entities.

Weak Entity is represented by double rectangle.



Continuing our previous example,**Professor**is a strong entity, and the primary key is **Professor_ID**. However, another entity is**Professor_Dependents**, which is our Weak Entity.

Table -Professor_Dependents

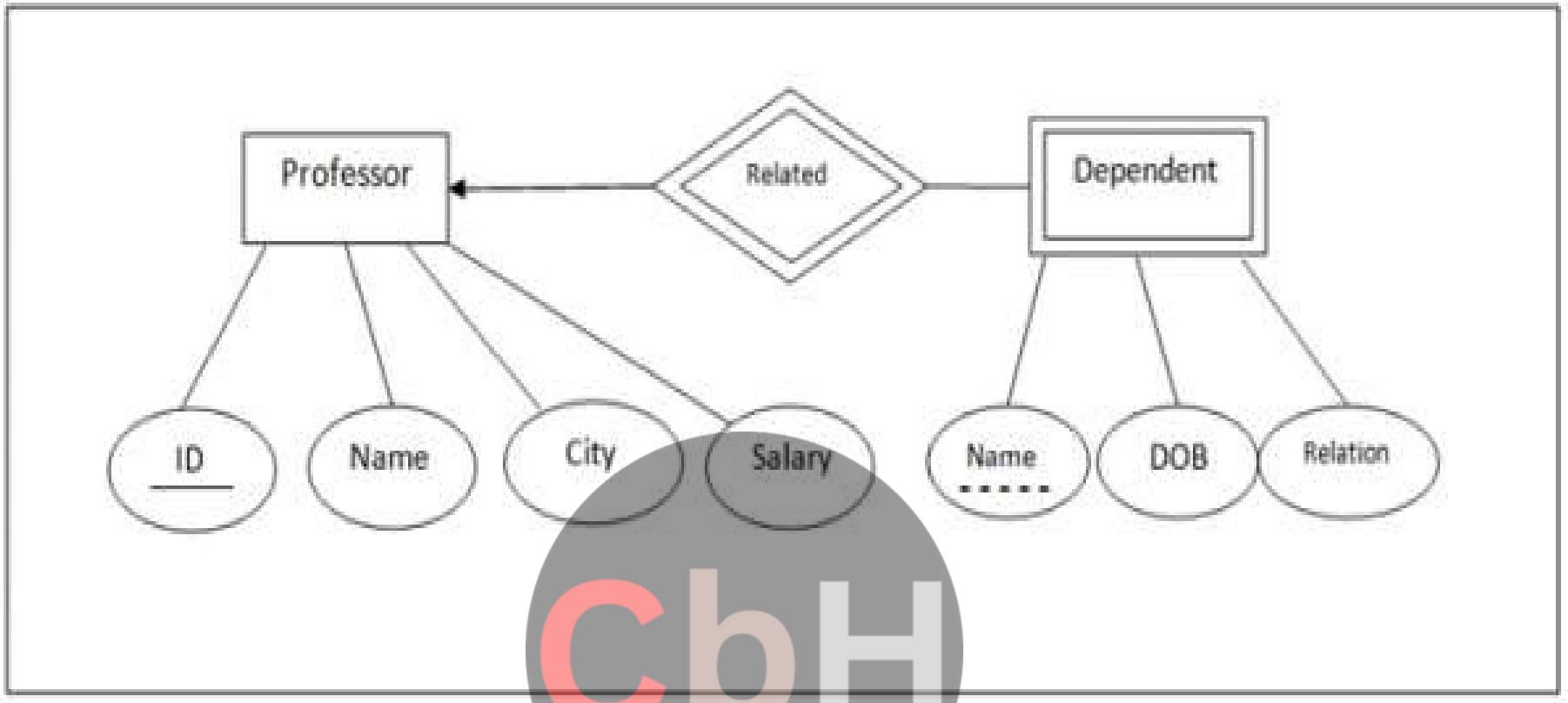
Name	DOB	Relation

This is a weak entity since its existence is dependent on another entity **Professor**, which we have shown as Professor_Dependents

.

Example of Strong and Weak Entity

The example of strong and weak entity can be understood by the below figure.



The Strong Entity is **Professor**, whereas **Dependent** is a Weak Entity. **ID** is the primary key (represented with a line) and Name in **Dependent** entity is called **Partial Key** (represented with a dotted line).