

Database Management System

IT 261

Bachelor of Information Technology

Prepared by Dr. Khaled Eskaf

Lecture 2





Course Outlines

Week	Topic	Objective	MySQL Practical Activities
	Introduction to Database		Install MySQL, explore the
1	Concepts	Understand basic database concepts.	Interface, create a basic database.
2	Relational Data Model	Learn about the relational data model.	Create relational tables in MySQL.
	Relational		
3	Integrity Rules	Understand relational integrity constraints.	Implement primary and foreign key constraints in MySQL.
4	Relational Algebra and Basic SQL	Explore relational algebra and basic SQL commands.	Use basic SQL queries in MySQL.
5	Advanced SQL Queries	Delve into complex SQL operations.	Execute advanced SQL commands in MySQL.
6	Introduction to Database Normalization	Begin learning database normalization principles.	Apply the first normal form (1NF) in MySQL.
7	Advanced Database Normalization	Explore advanced normalization techniques.	Normalize a database in MySQL to 2NF and 3NF.
8	Data Modeling and ER Diagrams	Understand data modeling and ER diagrams.	Create an ER diagram and implement it in MySQL.
9	Database Design for Real-World Apps	Apply database design to real-world scenarios.	Develop a database schema based on a real-world example.
10	Project and Review	Apply all learned concepts in a comprehensive project.	Complete a project in MySQL incorporating all learned skills.

Revision

Definitions

Database:

- A very large, integrated collection of data.

- Models real-world:

Entities (e.g., students, courses),

Relationships (Ex., Eva is taking Biology)

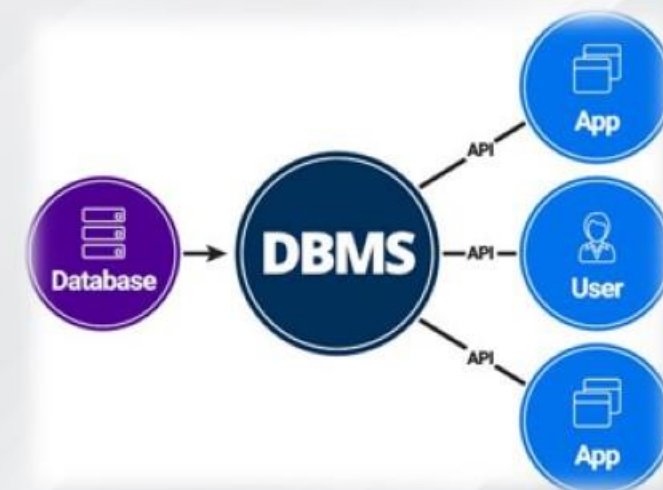
Database Management System (DBMS)

a software package designed to store and manage databases.

- Examples of Database Applications(Banking)

Roll No.	Name	Course	
CS08	Steive	Comp. Sci.	→ Entity 1
EE54	Jhoson	Electronics	→ Entity 2
B12	Eva	Biology	→ Entity 3
F32	Jhoson	Finance	→ Entity 4
M26	Erica	Maths	→ Entity 5

Student Table



Purpose of Database Systems

❑ In the early days, database applications were built directly on top of file systems

❑ Drawbacks of using file systems to store data:

- 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 — multiple files and formats



Drawbacks of File System

- = Data redundancy
- = Require more primary memory
- = Processor has to do lot of work
- = More Computations

Why Use a DBMS?



- Separation of the Data definition and the Program
- Abstraction into a simple model
- Data independence and efficient access.
- Reduced application development time – ad-hoc queries
- Data integrity and security.
- Uniform data administration.
- Concurrent access, recovery from crashes.
- Support for multiple different views



- ❑ **Shift from computation to information:**

- at the “low end”: scramble to webspace (a mess!)
- at the “high end”: scientific applications

- ❑ **Datasets increasing in diversity and volume.**

- Digital libraries, interactive video, Human Genome project, EOS project.
- ... need for DBMS exploding

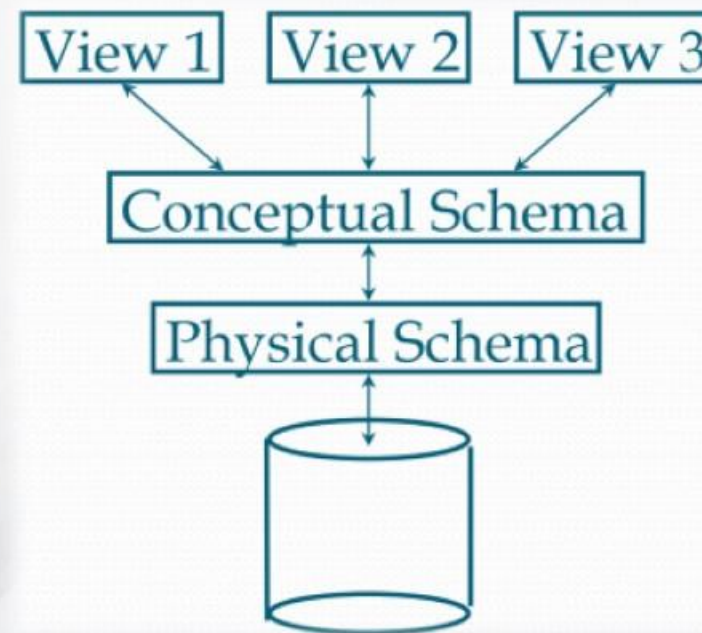
- ❑ **DBMS encompasses most of CS**

- OS, languages, theory, “AI”, multimedia, logic

Levels of Abstraction

Many **views**, single **conceptual (logical) schema** and **physical schema**.

- Views describe how users see the data.
- Conceptual schema defines logical structure. Sometime we separate between conceptual level and logical level
- Physical schema describes the files and indexes used
- *Schemas are defined using DDL (Data Definition Language)*
- *data is modified/queried using DML (Data Manipulation Language)*





CHAPTER 2

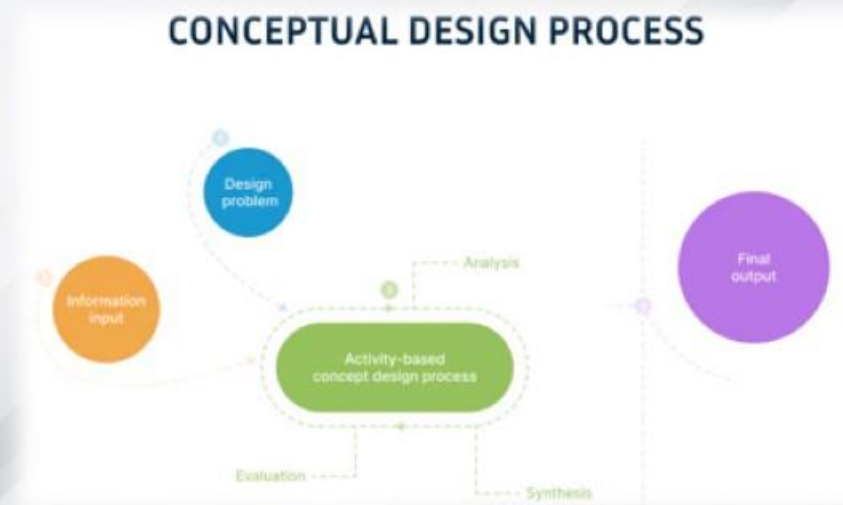


Week	Topic	Objective	MySQL Practical Activities
	Introduction to Database		Install MySQL, explore the
1	Concepts	Understand basic database concepts.	interface, create a basic database.
2	Relational Data Model	Learn about the relational data model.	Create relational tables in MySQL.
	Relational		

The Entity-Relationship Model

Overview of Database Design

- Conceptual design: (*ER Model is used at this stage.*)
 - What are the *entities* and *relationships* in the enterprise?
 - What information about these entities and relationships should we store in the database?
 - What are the *integrity constraints* or *business rules* that hold?
 - A database 'schema' in the ER Model can be represented pictorially (*ER diagrams*).
 - Can map an ER diagram into a relational schema.



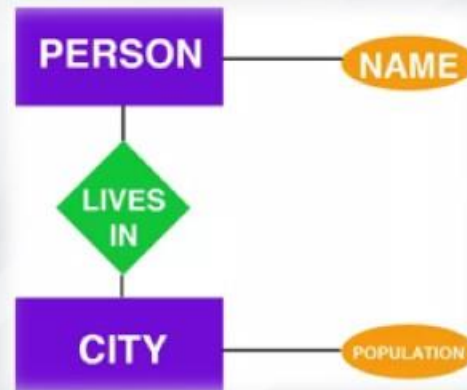
ER Model Basics



- **Entity**: Real-world object distinguishable from other objects. An entity is described (in DB) using a set of **attributes**.
- **Entity Set**: A collection of similar entities. E.g., all employees.
 - All entities in an entity set have the same set of attributes. (Until we consider ISA hierarchies, anyway!)
 - Each entity set has a **key**.
 - Each attribute has a **domain**.

What is an ER Diagram?

- Definition:** An ER Diagram (Entity-Relationship Diagram) is a **graphical representation of an ER model**. It uses symbols and notation to visualize the entities, attributes, and relationships defined in the model. ER diagrams are tools that help to detail and communicate the design based on the ER model.
- Components:** ER diagrams consist of rectangles (representing entities), ovals (representing attributes), diamonds (representing relationships), and lines connecting these shapes (indicating the nature of the relationship between entities).
- Purpose:** The main goal of an ER diagram is to provide a visual tool for examining and communicating the database's logical structure, making it easier for stakeholders to understand the design and for developers to implement the database schema.



What is an ER Model?

- **Definition:** The Entity-Relationship (ER) Model is a conceptual tool for modeling data in a structured way. It is **used to represent the data architecture and its components at a high level**, without getting into the details of physical storage. The ER model provides a way to describe the data types, relationships, and constraints that should be applied to the data.
- **Components:** The ER model includes entities (which represent real-world objects or concepts), attributes (which describe properties of entities), and relationships (which define how entities are connected to each other).
- **Purpose:** Its primary purpose is to establish a clear and comprehensive blueprint for the logical structure of the database. It helps in understanding the data requirements of an organization and how different data elements interact with one another.

Feature	ER Model	ER Diagram
What is it?	A plan for how to organize data.	A picture showing the plan.
Purpose	To think about what data you need and how it fits together.	To show others your data plan in an easy-to-understand way.
Contains	Ideas about data, like what information you're keeping track of and how it's linked.	Shapes and lines that represent your data ideas.
Use	To make a clear plan for your database's structure.	To help people see and understand your data plan.
Level	Thinking about what data is needed.	Showing the plan with pictures.
For Who?	People making the database plan.	People making the database and others who need to understand the plan.
Result	A detailed plan for your data.	A drawing that shows your data plan.

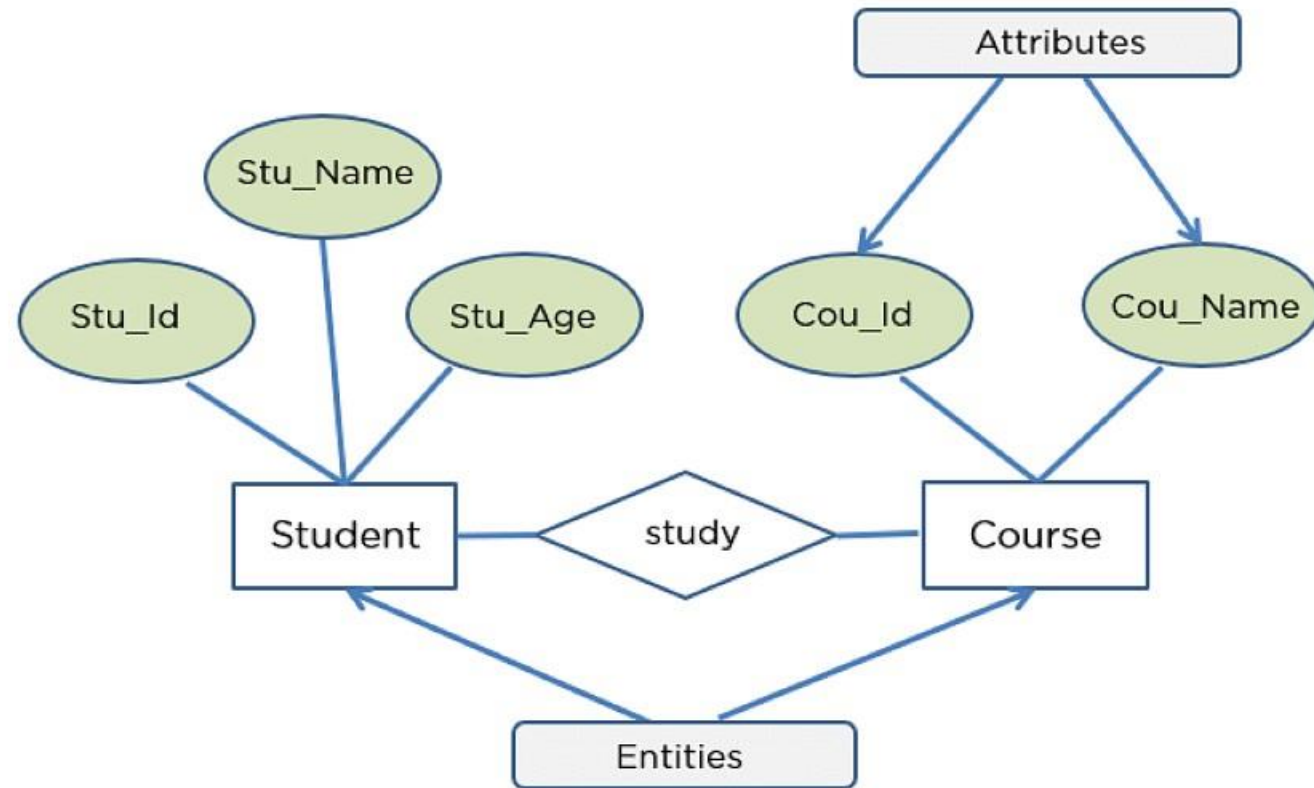


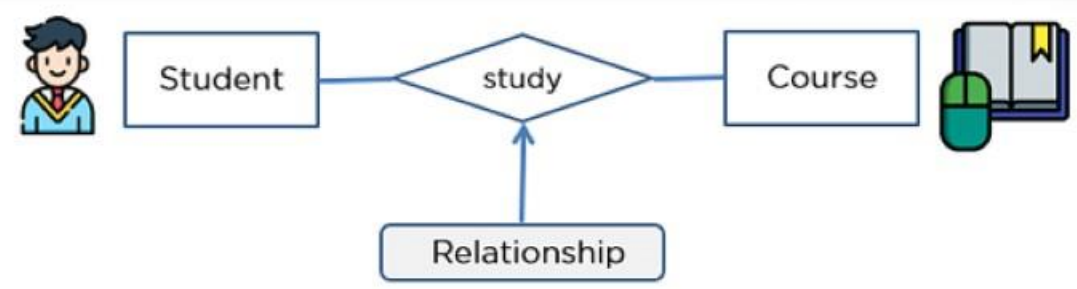
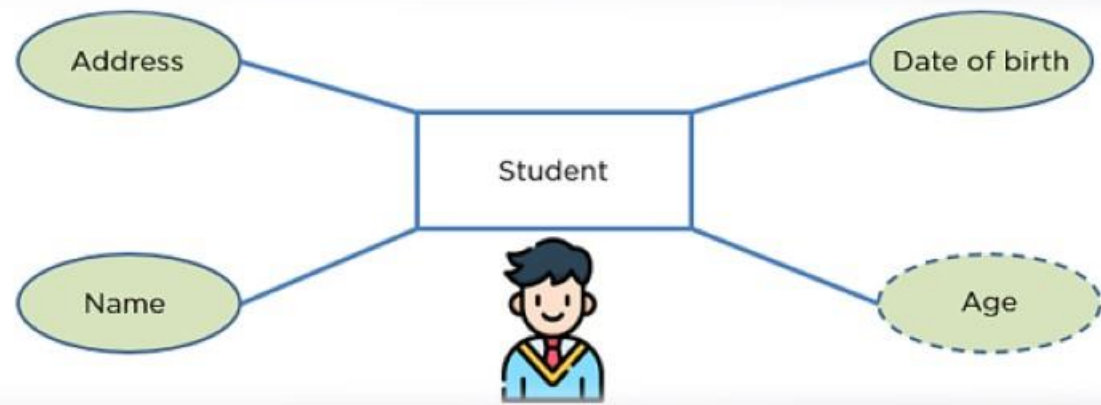
The following diagram showcases two entities - Student and Course, and their relationship.

The relationship described between student and course is many-to-many, as a course can be opted by several students, and a student can opt for more than one course.

Student entity possesses attributes - Stu_Id, Stu_Name & Stu_Age.

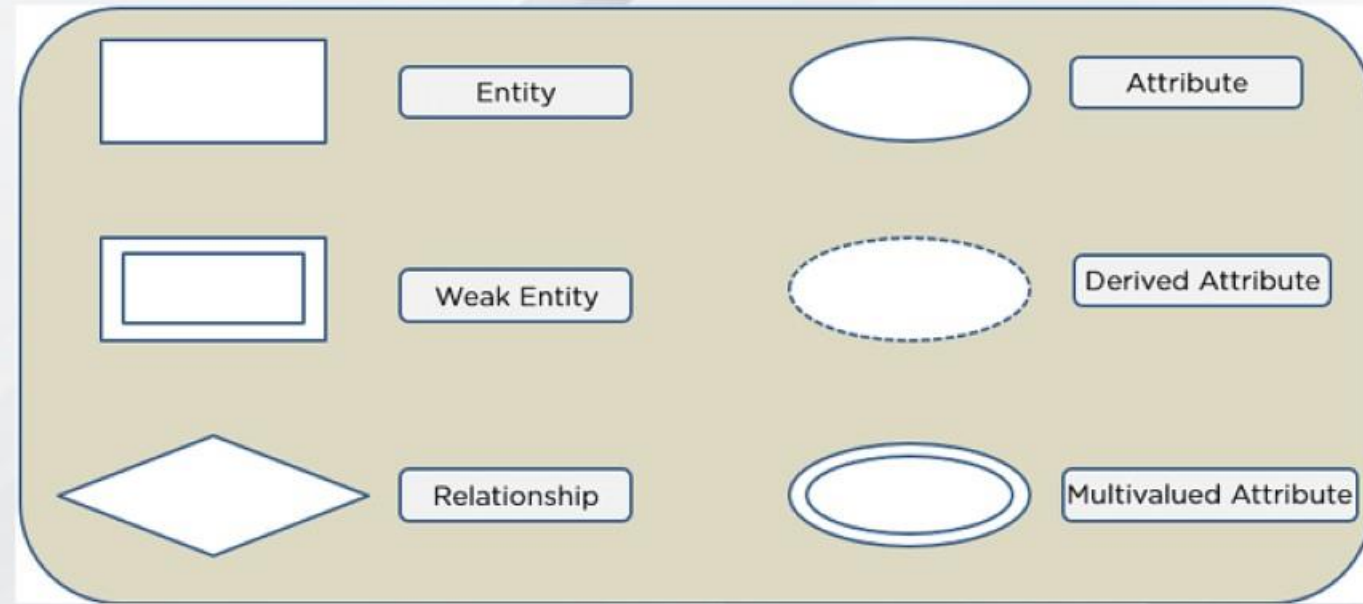
The course entity has attributes such as Cou_ID & Cou_Name.





Symbols Used in ER Diagrams

- Rectangles: This Entity Relationship Diagram symbol represents entity types
- Ellipses: This symbol represents attributes
- Diamonds: This symbol represents relationship types
- Lines: It links attributes to entity types and entity types with other relationship types
- Primary key: Here, it underlines the attributes
- Double Ellipses: Represents multi-valued attributes



Components of ER Diagram

You base an ER Diagram on three basic concepts:

- Entities
- Attributes
 - Key Attribute
 - Composite Attribute
 - Multivalued Attribute
 - Derived Attribute
- Relationships
 - One-to-One Relationships
 - One-to-Many Relationships
 - Many-to-One Relationships
 - Many-to-Many Relationships

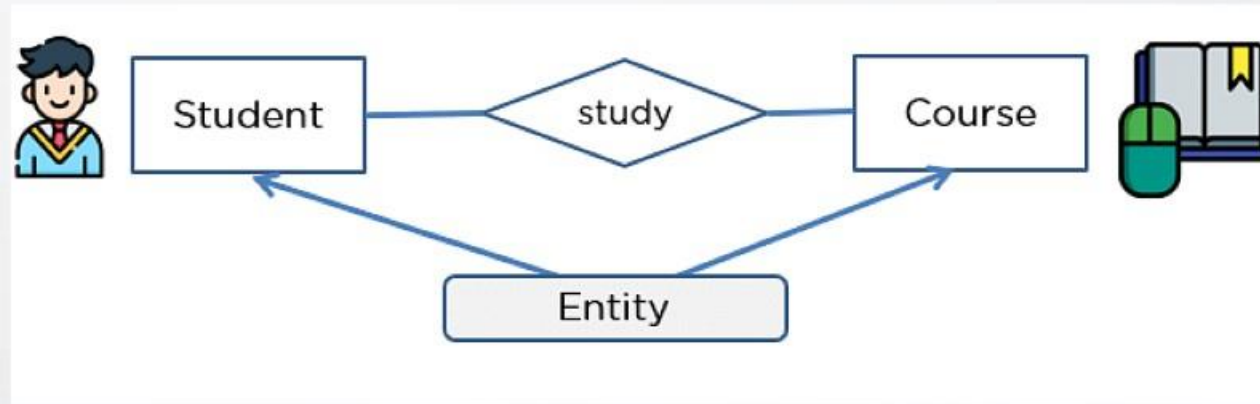
Component	Symbol
Entity	
Relationship	
Attribute	
Multivalued-Attribute	
Key Attribute	
Composite Attribute	
Weak Entity	
Weak Entity Relationship	

Entities

An entity can be either a living or non-living component.

It showcases an entity as a rectangle in an ER diagram.

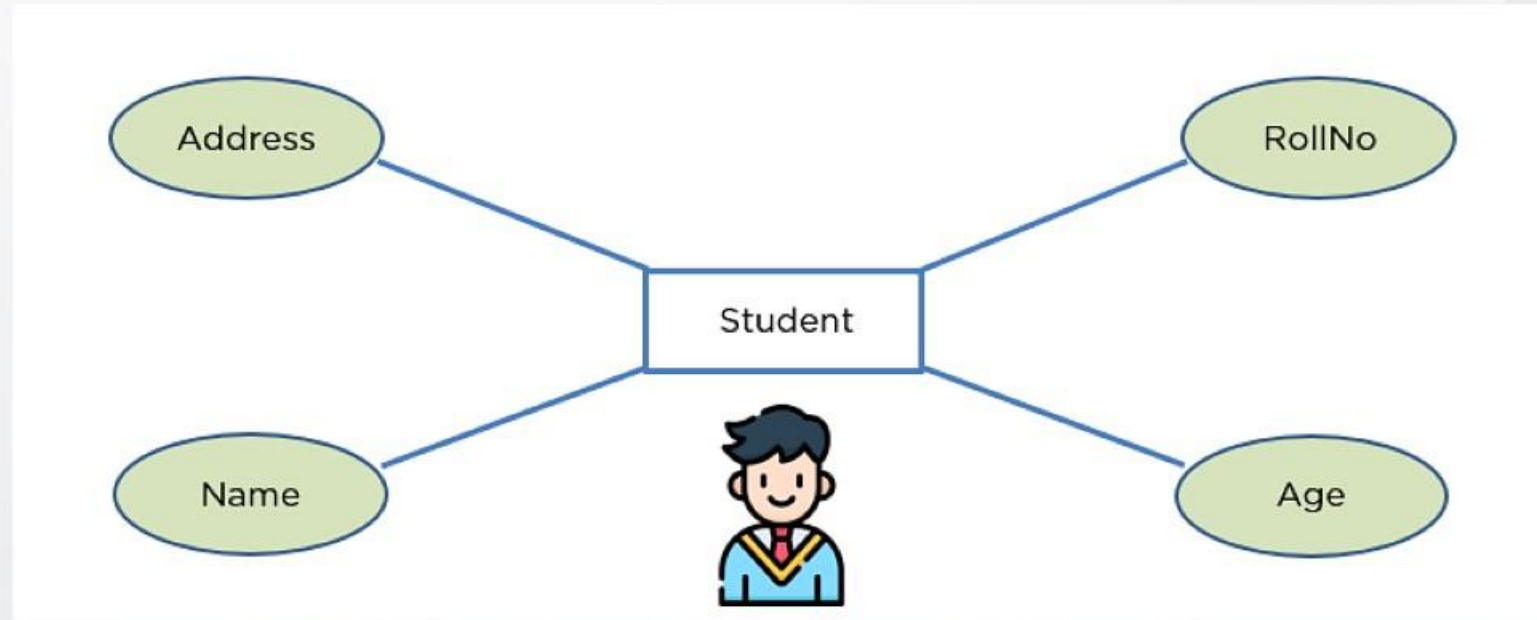
For example, in a student study course, both the student and the course are entities.



Attribute

An attribute exhibits the properties of an entity.

You can illustrate an attribute with an oval shape in an ER diagram.

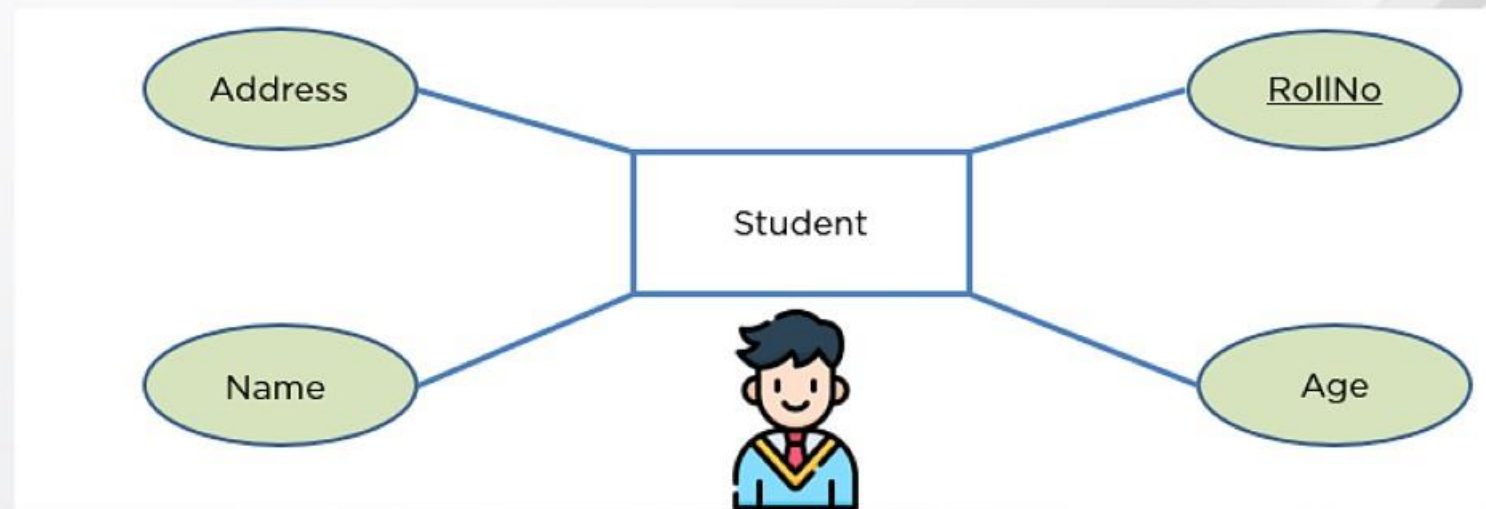


Key Attribute

Key attribute uniquely identifies an entity from an entity set.

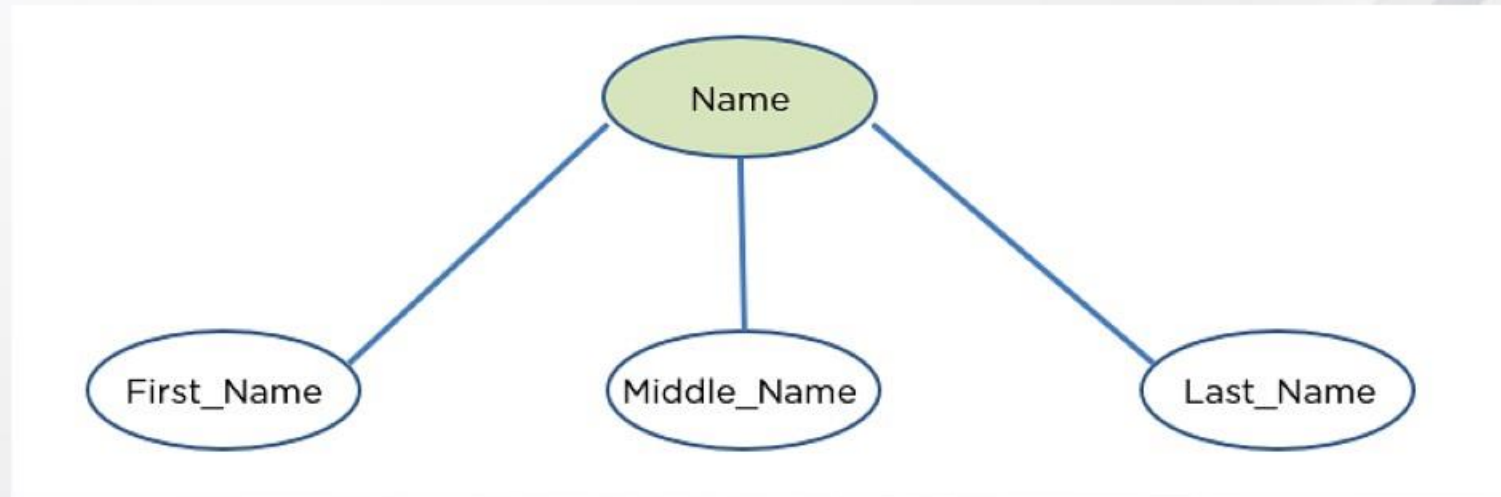
It underlines the text of a key attribute.

For example: For a student entity, the roll number can uniquely identify a student from a set of students.



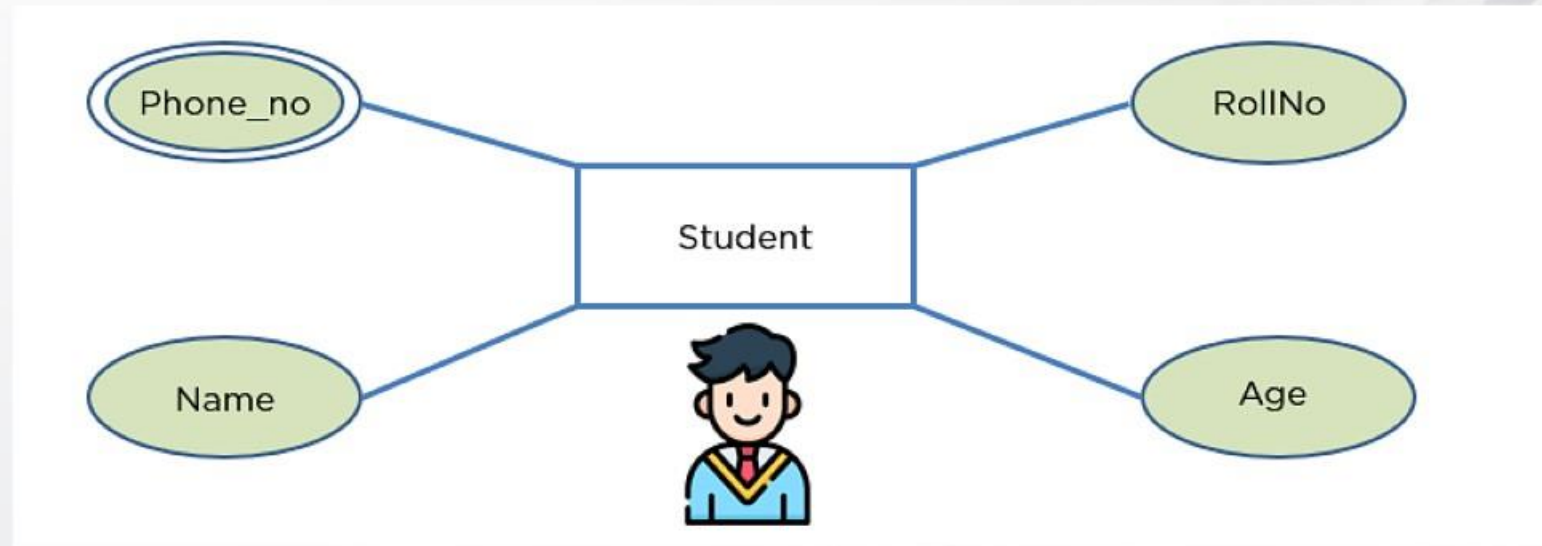
Composite Attribute

An attribute that is composed of several other attributes is known as a composite attribute. An oval showcases the composite attribute, and the composite attribute oval is further connected with other ovals.



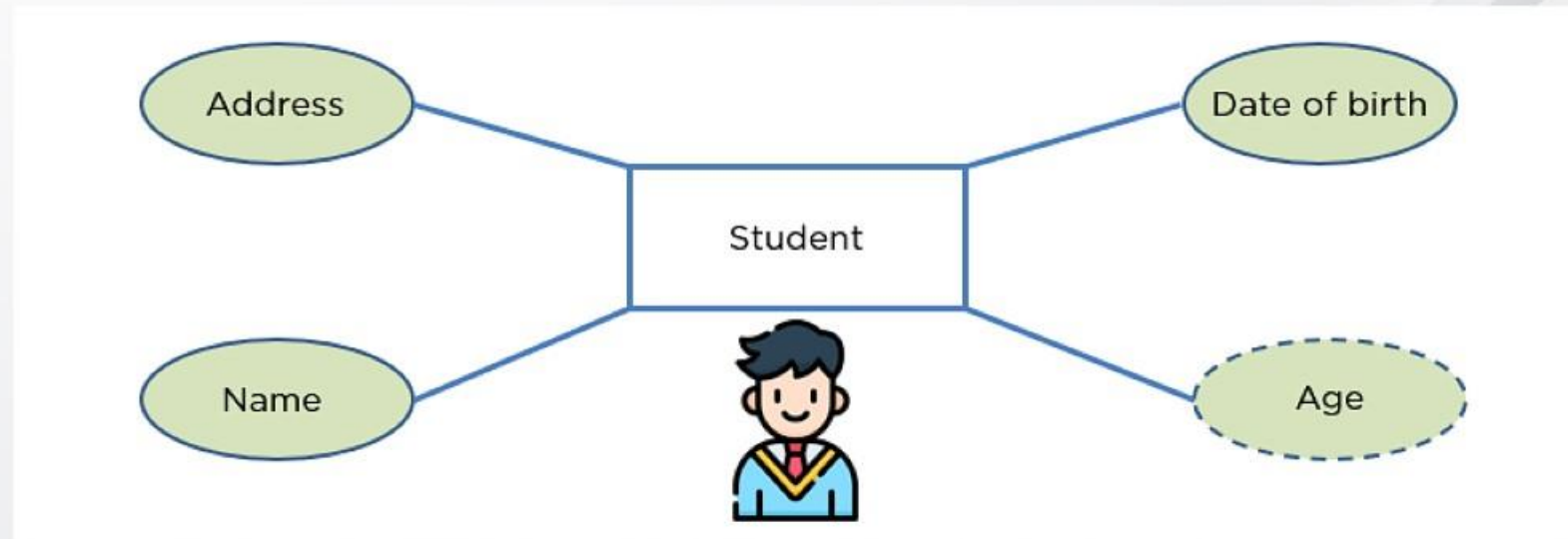
Multivalued Attribute

Some attributes can possess over one value, those attributes are called multivalued attributes. The double oval shape is used to represent a multivalued attribute.



Derived Attribute

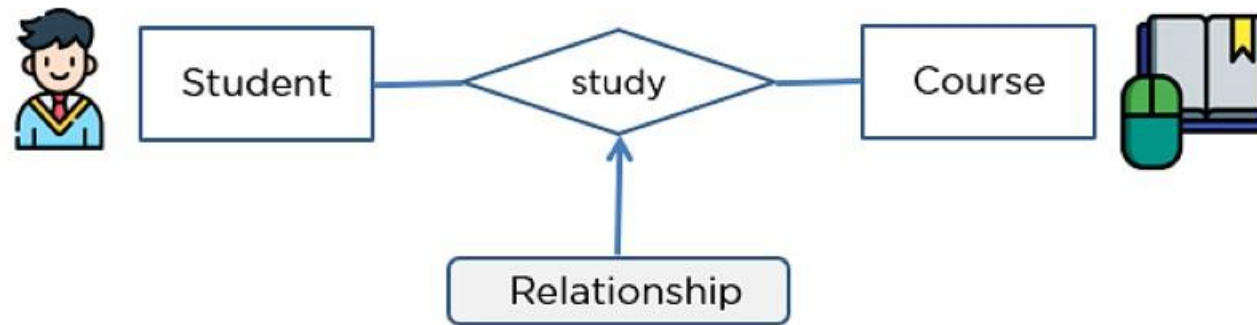
An attribute that can be derived from other attributes of the entity is known as a derived attribute. In the ER diagram, the dashed oval represents the derived attribute.



Relationship

The diamond shape showcases a relationship in the ER diagram.
It depicts the relationship between two entities.

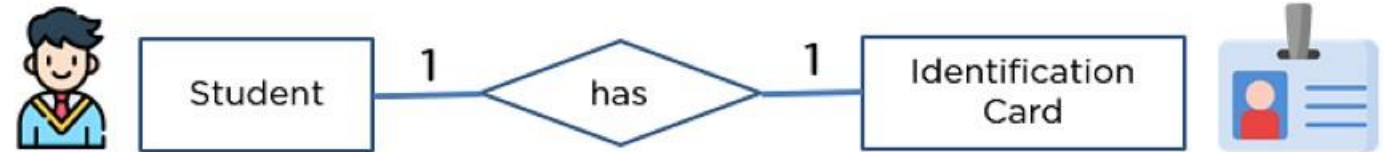
In the example below, both the student and the course are entities, and study is the relationship between them.



One-to-One Relationship

When a single element of an entity is associated with a single element of another entity, it is called a one-to-one relationship.

For example, a student has only one identification card and an identification card is given to one person.



One-to-Many Relationship

When a single element of an entity is associated with more than one element of another entity, it is called a one-to-many relationship

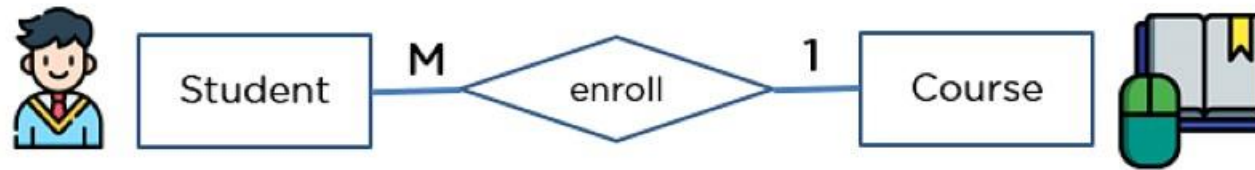
For example, a customer can place many orders, but an order cannot be placed by many customers.



Many-to-One Relationship

When more than one element of an entity is related to a single element of another entity, then it is called a many-to-one relationship.

For example, students have to opt for a single course, but a course can have many students.



Many-to-Many Relationship

When more than one element of an entity is associated with more than one element of another entity, this is called a many-to-many relationship.

For example, you can assign an employee to many projects and a project can have many employees.



How to Create an ER Diagram

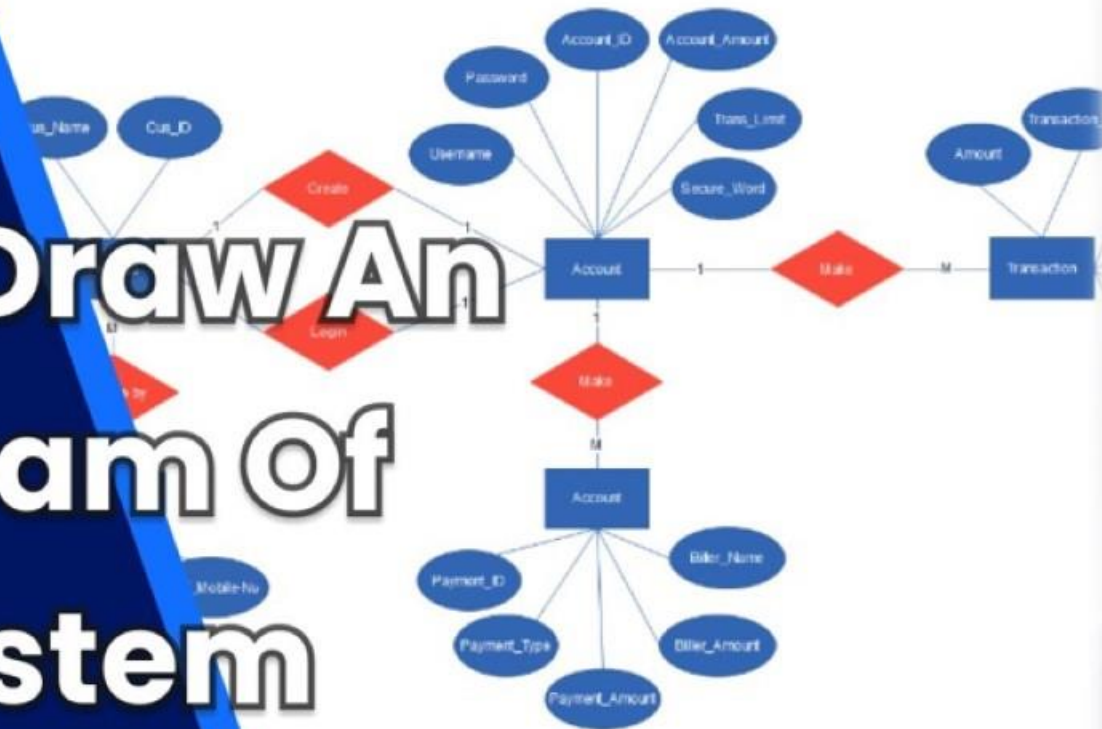
Before you design a relational database, it makes sense to create an ER diagram. You may have software that is equipped to handle this process. If not, Put pen (or pencil) to paper, or find a software program that can handle the charting requirements.

To create an ER diagram by hand:

1. Create a rectangular box for each **entity** or concept relevant to your model.
2. Draw lines to connect related entities to model the **relationships**. Label the relationships using verbs inside diamond shapes.
3. Identify the relevant **attributes** for each entity, beginning with the most important attributes, and enter them in ovals in the diagram. Later, you can make the attribute lists more detailed.

Creating an Entity-Relationship (ER) Diagram for a simple bank system involving customers and their bank accounts can be a great way to understand how ER diagrams work.

How To Draw An ER Diagram Of Bank System



ER Diagram for Bank Database:

In an ER diagram for a Banking System, we will have to keep in my mind the following points while making the diagram:

- There are going to be a number of entities, namely customers, banks, bank accounts, and transactions.
- The various entities in our diagram will have many different attributes like the Customer has **First Name** and **Last Name**.
- There are going to be varying relationships between the various entities like the Bank account, and the Customer will have a relationship called owned by, where we show that the account is being held by the Customer.

Let's start with making our ER diagram:

Deciding Entities

We first decide the entities that we are going to have in our diagram :

- Bank**
- Bank Account**
- Transaction**
- Customer**

- Bank
- Bank Account
- Transaction
- Customer

Bank Account

Customer

Bank

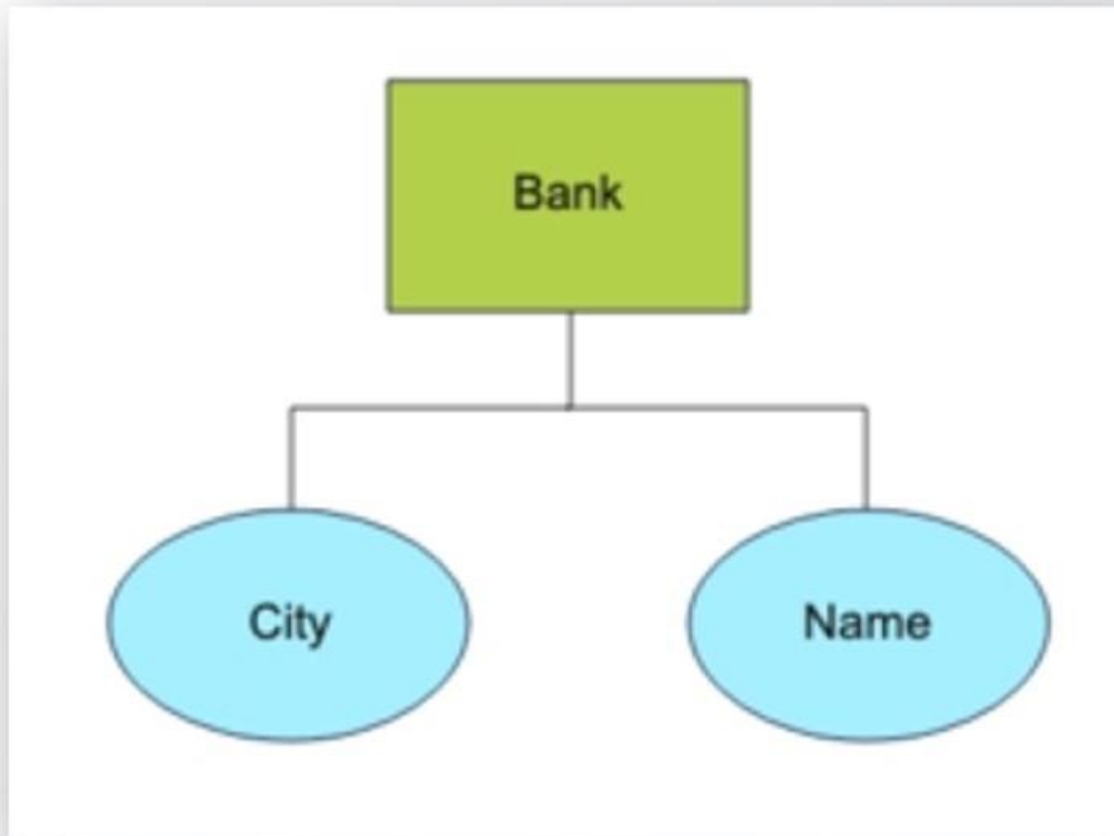
Transaction

Deciding Attributes :

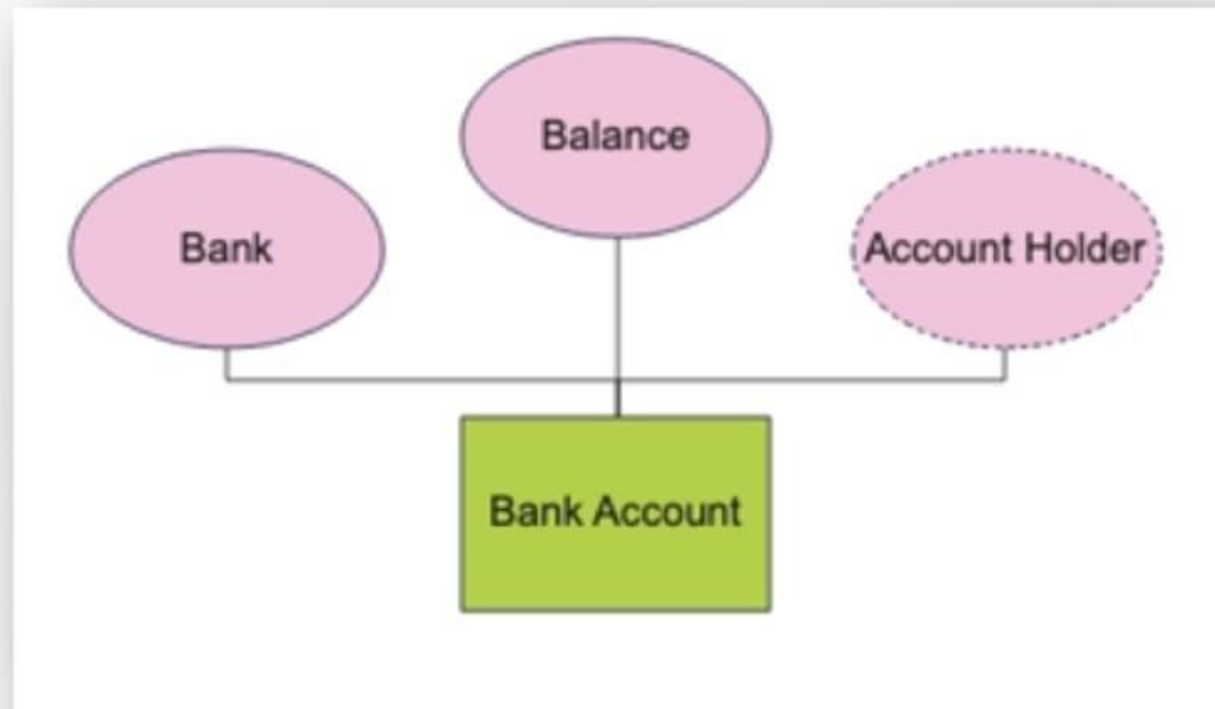
We then decide what attributes are associated with these entities:

- Bank: **Name, City**
- Bank Account: **Balance, Account holder, Bank**
- Customer: **First Name, Last Name, Phone Number**
- Transaction: **Deposit Account, Withdrawal Account, Timestamp, Currency, Amount**

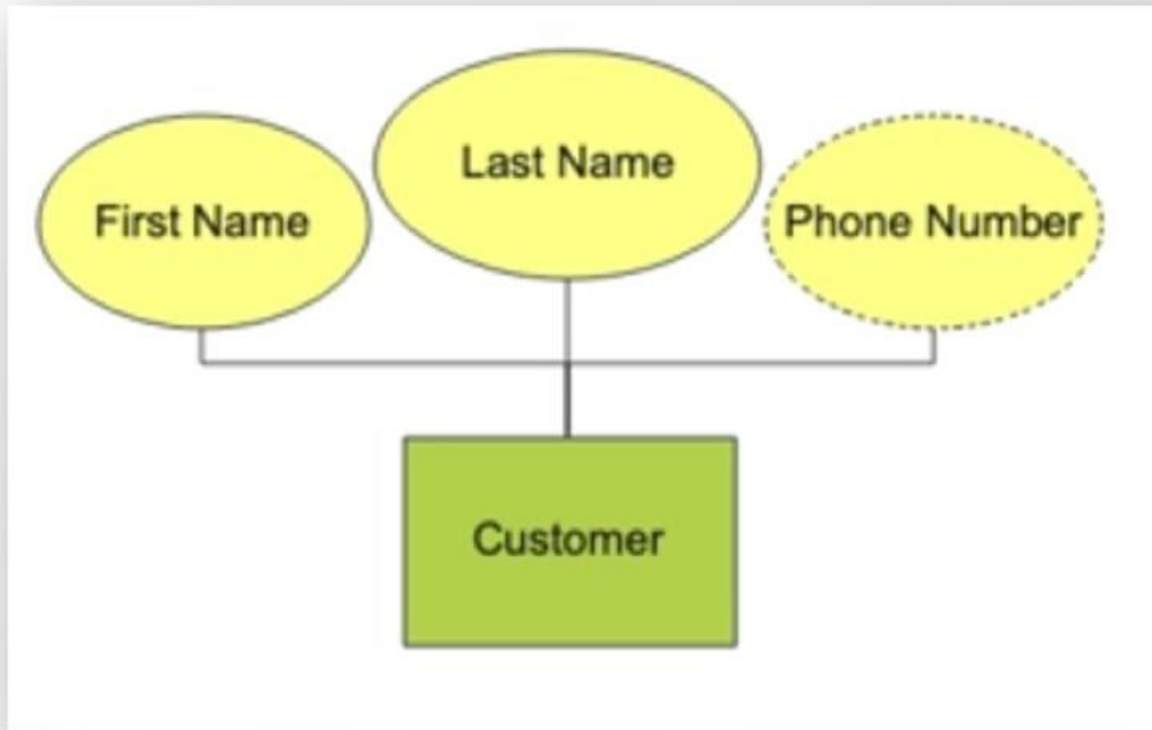
Bank: **Name, City**



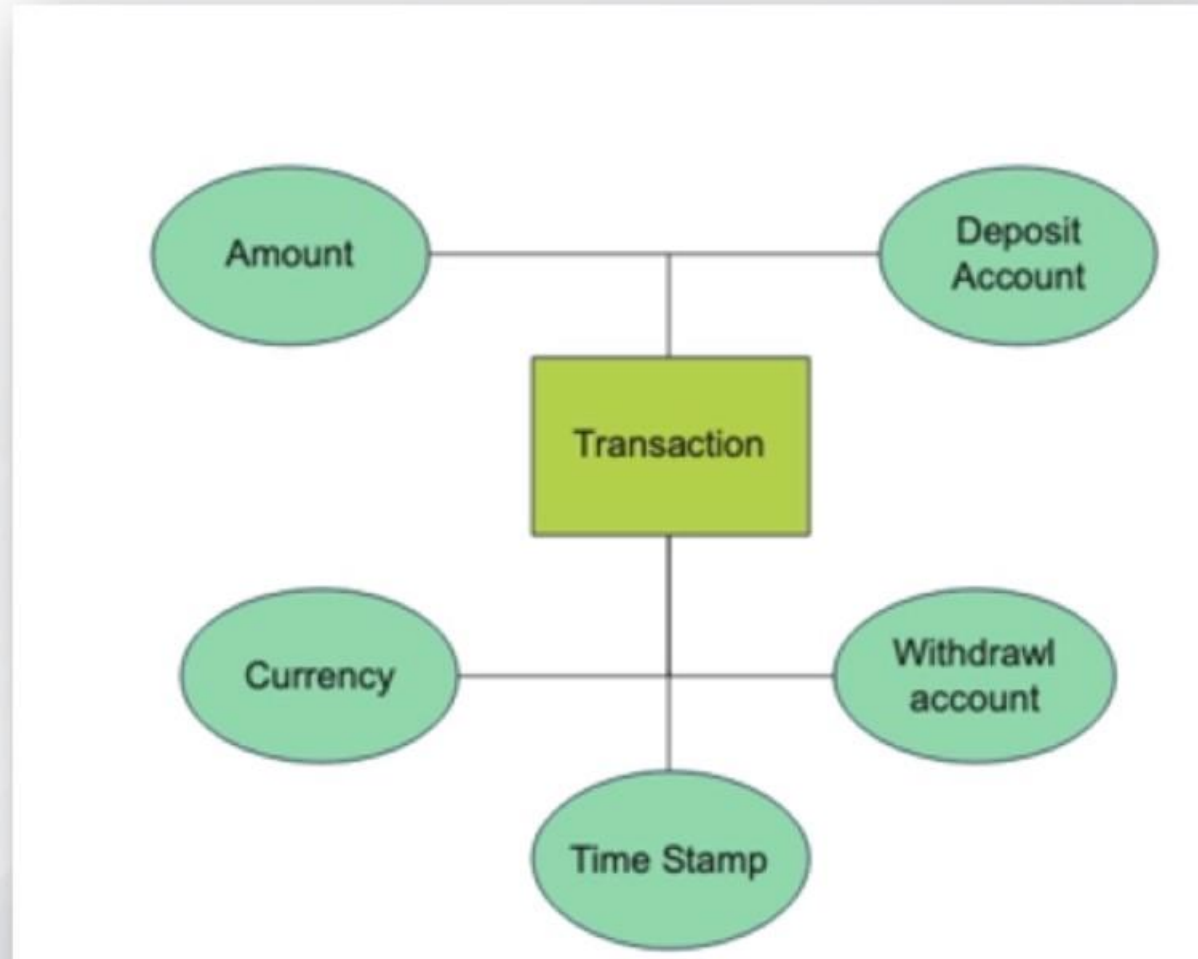
Bank Account: **Balance**, **Account holder**, **Bank**



Customer: **First Name, Last Name, Phone Number**



- Transaction: **Deposit Account, Withdrawal Account, Timestamp, Currency, Amount**

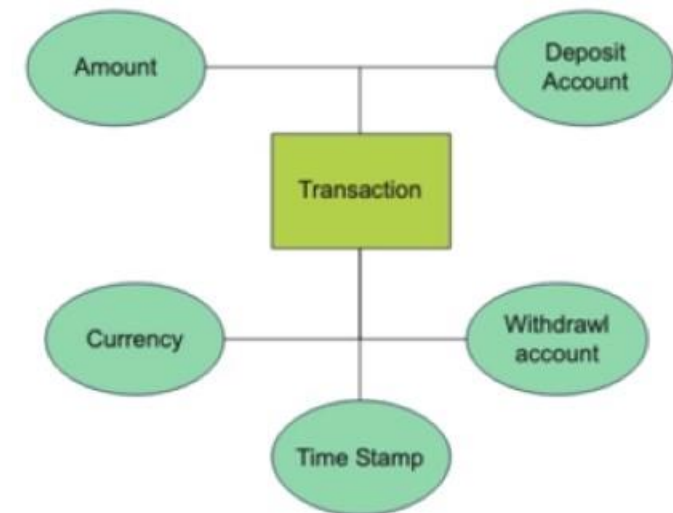
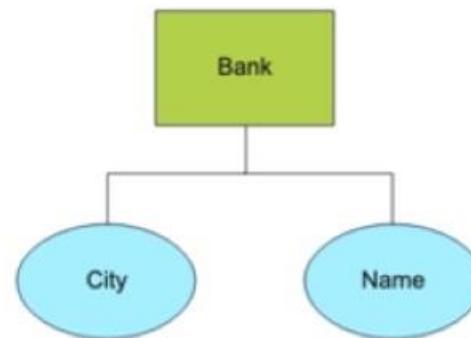
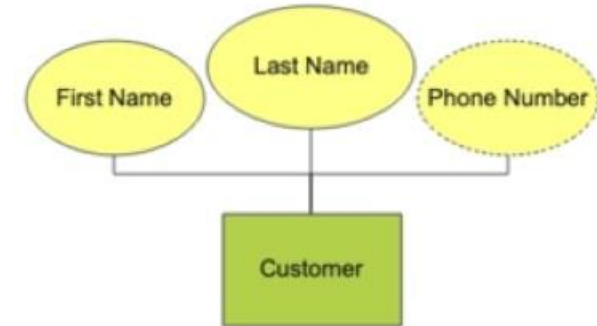
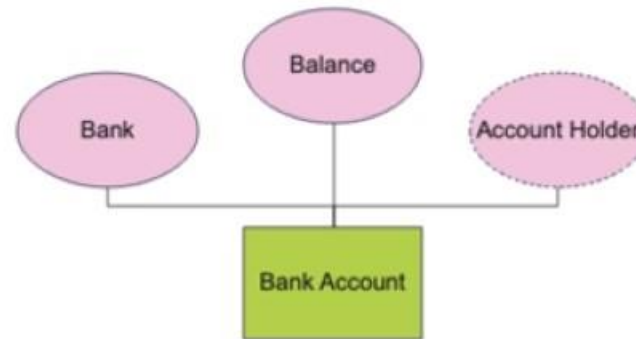


•Bank: **Name, City**

•Bank Account: **Balance, Account holder, Bank**

•Customer: **First Name, Last Name, Phone Number**

•Transaction: **Deposit Account, Withdrawal Account, Timestamp, Currency, Amount**

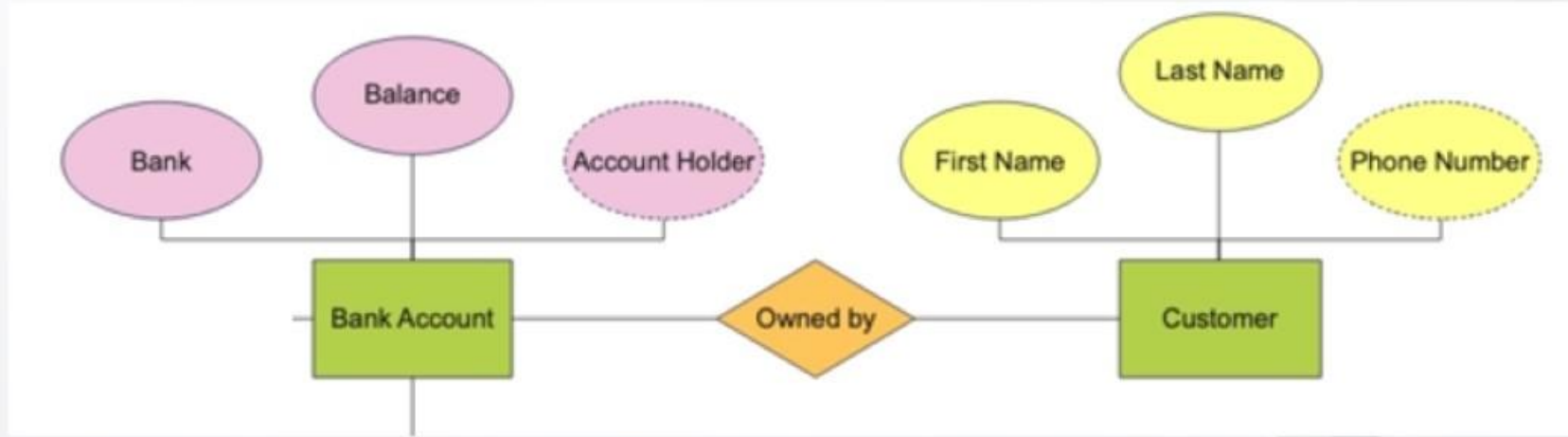


Deciding Relationships :

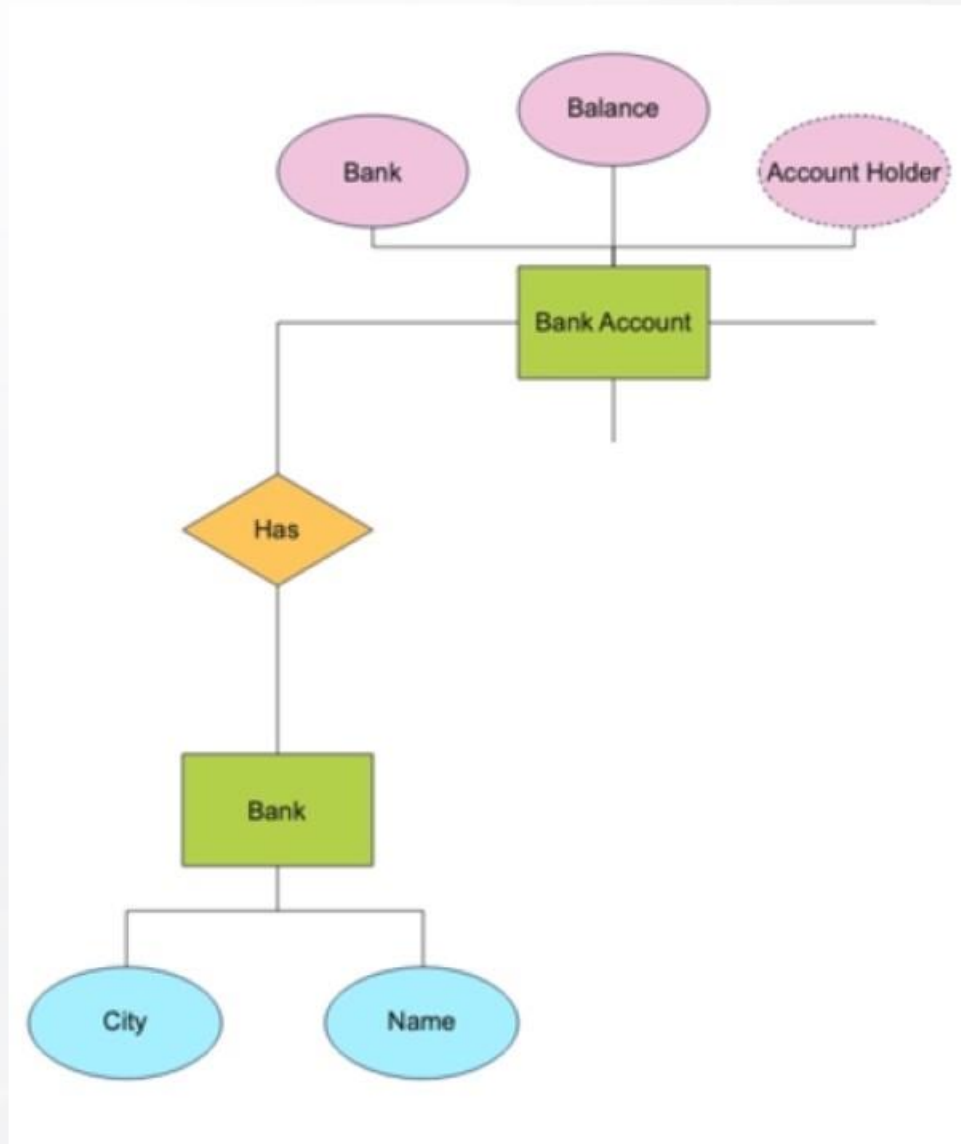
We then decide the relationships between the various entities:

- A Bank Account is owned by a Customer.
- A Bank has a Bank Account.
- A Transaction is linked to the Bank Account.
- The relationship between Bank Account and Customer is many to one because a Customer can have many Bank Accounts.
- The relationship between Bank and Bank Account is one too many as a Bank has many bank accounts.
- The Transaction to bank account relationship is many to one because multiple transitions can be linked to one bank account.

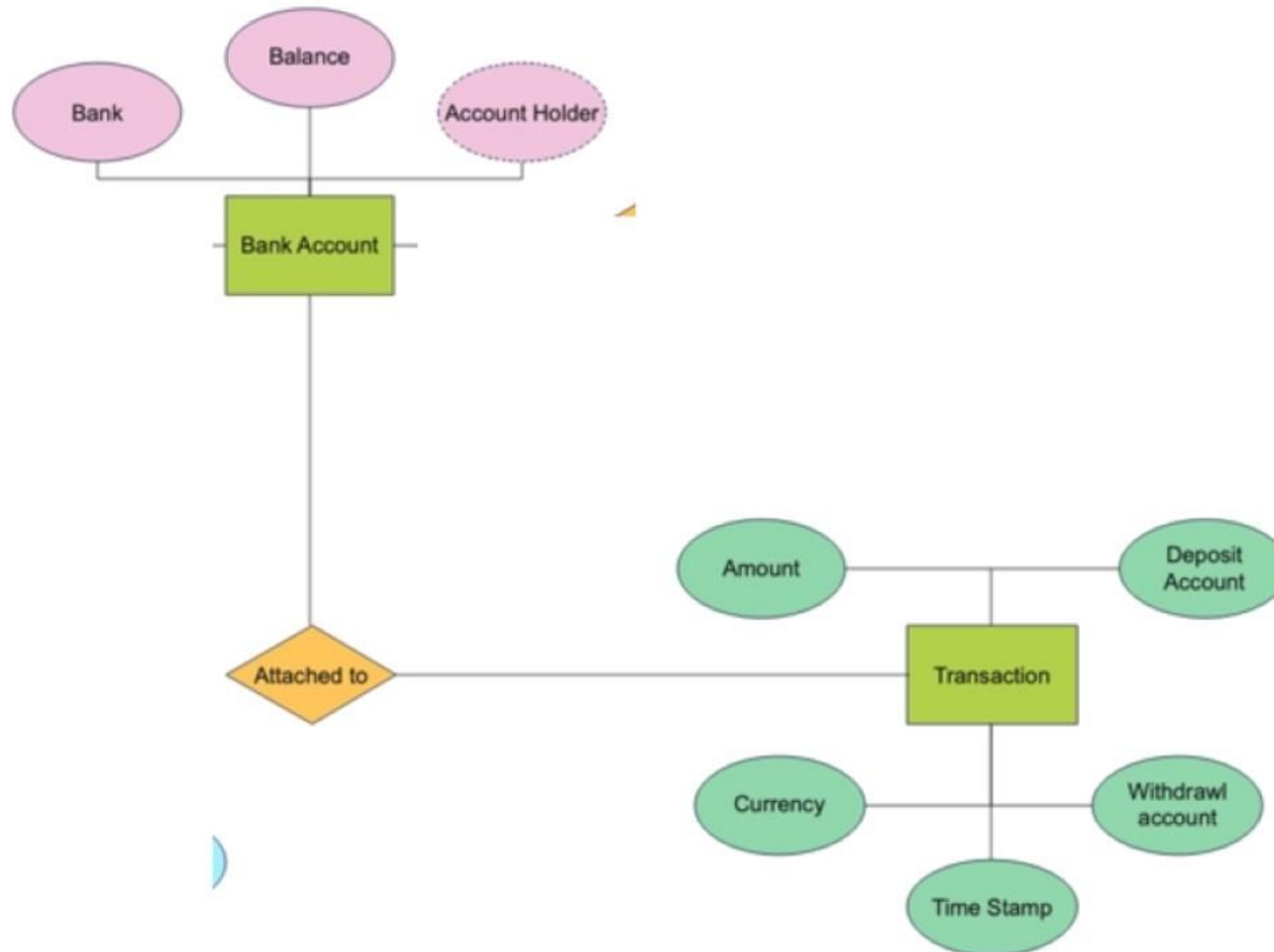
A Bank Account is owned by a Customer.



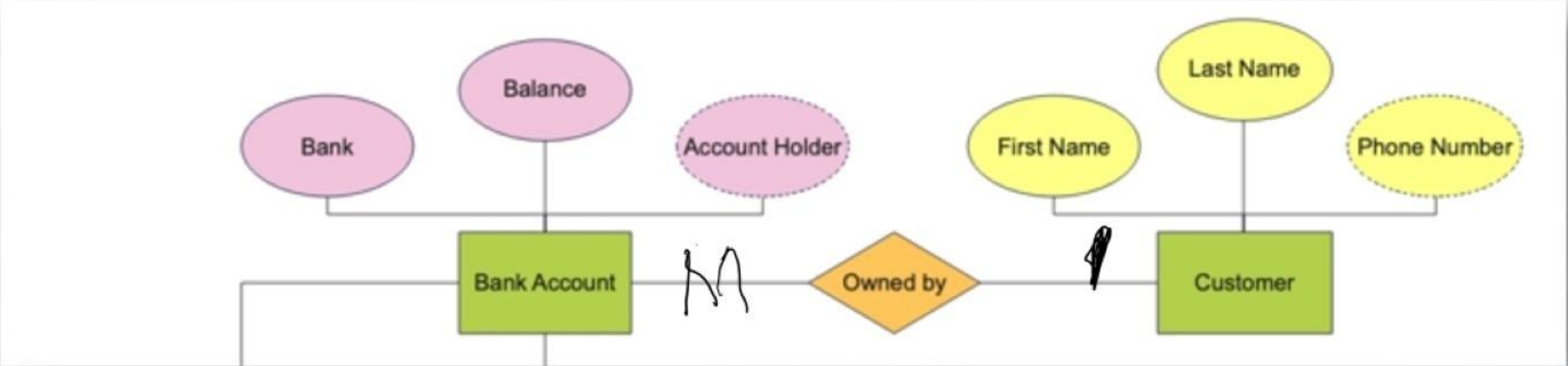
A Bank has a Bank Account.



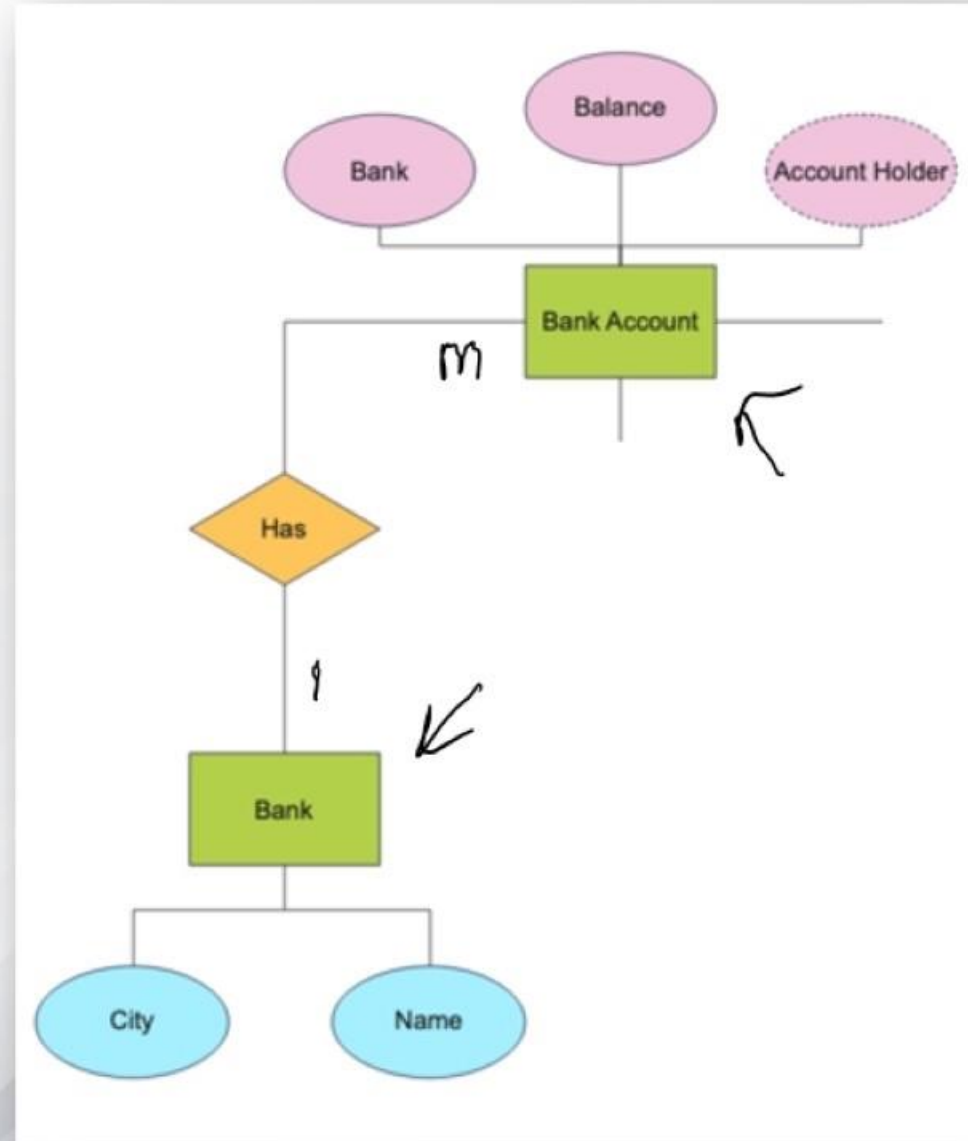
A Transaction is linked to the Bank Account



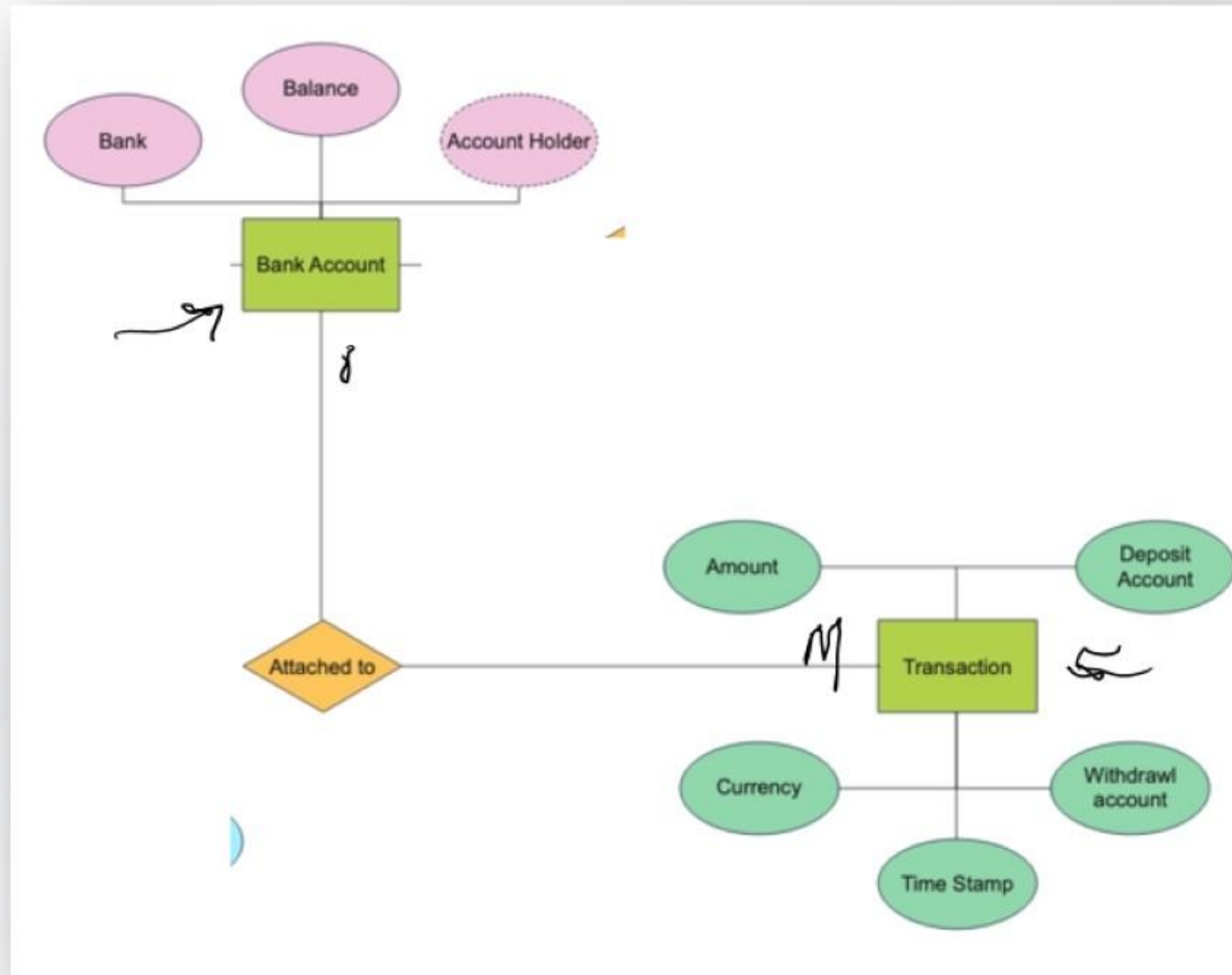
The relationship between Bank Account and Customer is many to one because a Customer can have many Bank Accounts.

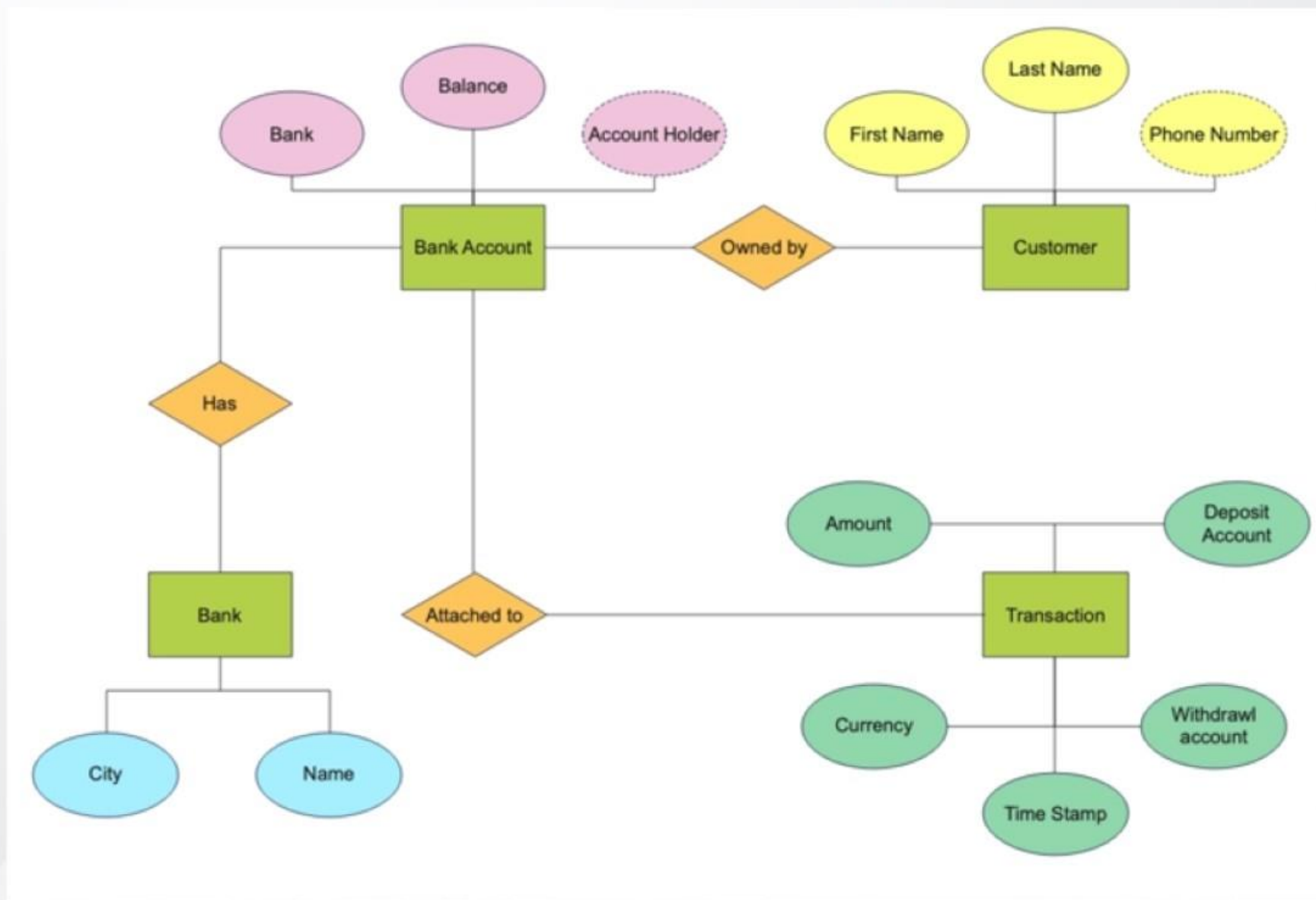


The relationship between Bank and Bank Account is one too many as a Bank has many bank accounts.



The Transaction to bank account relationship is many to one because multiple transitions can be linked to one bank account.





Advantages of Banking ER diagrams:

Let's discuss the advantages of using an ER model for designing your banking database.

- Conceptually, it is very simple:** ER models are very simple because if we know the relationship between entities and attributes, then we can easily draw an ER diagram. These help us easily understand the system.
- Effective communication tool:** It is an effective communication tool for database designers
- Increased efficiency:** By using an ER diagram, designers can identify inefficiencies and redundancies in the system, allowing them to streamline processes and reduce costs. This can result in a more efficient system that is easier to maintain and operate.
- Easier maintenance and updates:** By having a clear understanding of the system's components and relationships, designers can more easily make updates and modifications to the system as needed. ER diagrams can help to ensure that updates and modifications are made in a consistent and efficient manner, which can help to minimize errors and downtime.



مجموعة جامعة
الجميع الذكية
للتعليم القابضة

MIDOCEAN
UNIVERSITY



Thank You

