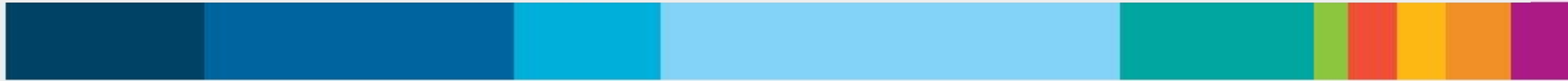


IBM Brighter Blue

Best Practices of Logical Database Design



Module Overview

In the previous module, you learnt in detail about the Record based logical model and the most popular of the models – Relational data model and its structure.

Let us now familiarize ourselves with the best practices of the logical database design as every database must be designed using the industry best practices.

This module should take about 45 minutes of your time and at the end of this module, you should be able to:

- Identify the **best practices** of logical database design.
- Recognize that the entities in an ER diagram can be mapped to tables.

01 Components of Database Design

Why the **Need** for a Logical Database Design?

Logical Database Design

The logical database design is a conceptual, abstract design. This type of design:

- ✓ Provides the foundation for designing a database.
- ✓ Views the logical relationships among the objects.
- ✓ Defines the types of information that you need, rather than implementing the details.



The logical database design is the process of constructing a model of information used in an enterprise based on a specific data model, but independent of a particular DBMS or other physical considerations.

It involves arranging data into a series of logical relationships and includes entities (tables), attributes (columns/fields) and keys (relationships).

Key Components of Logical Database Design

Let us learn about the key components used in the logical database design.

Click each database component to learn more.



Attributes

- ✓ A database field or an instance in a row of a database
- ✓ Defines the uniqueness of an entity



Keys

- ✓ Used to establish and identify relations between tables
- ✓ Vital part of the structure of a table

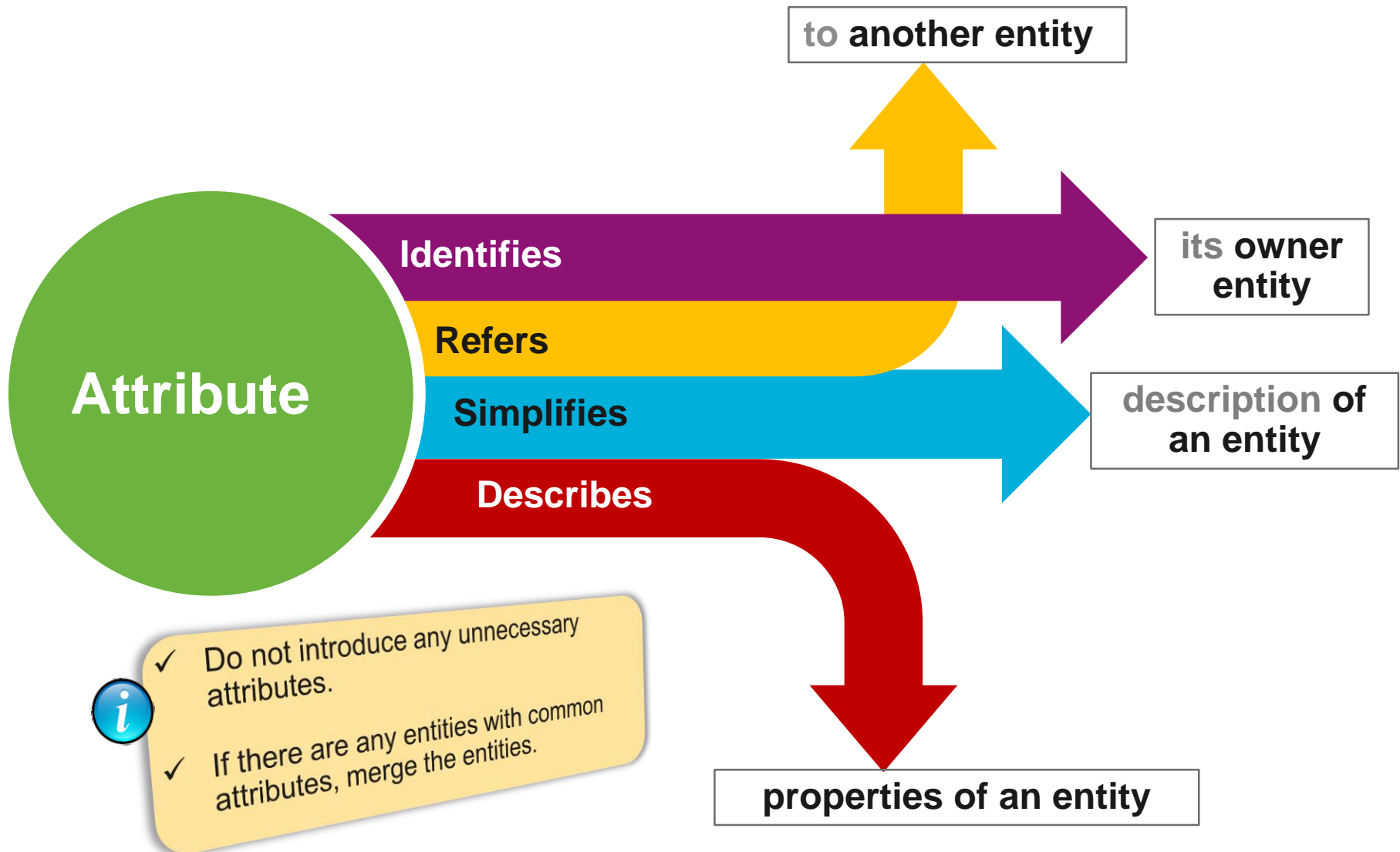


Entities & Sub-entities

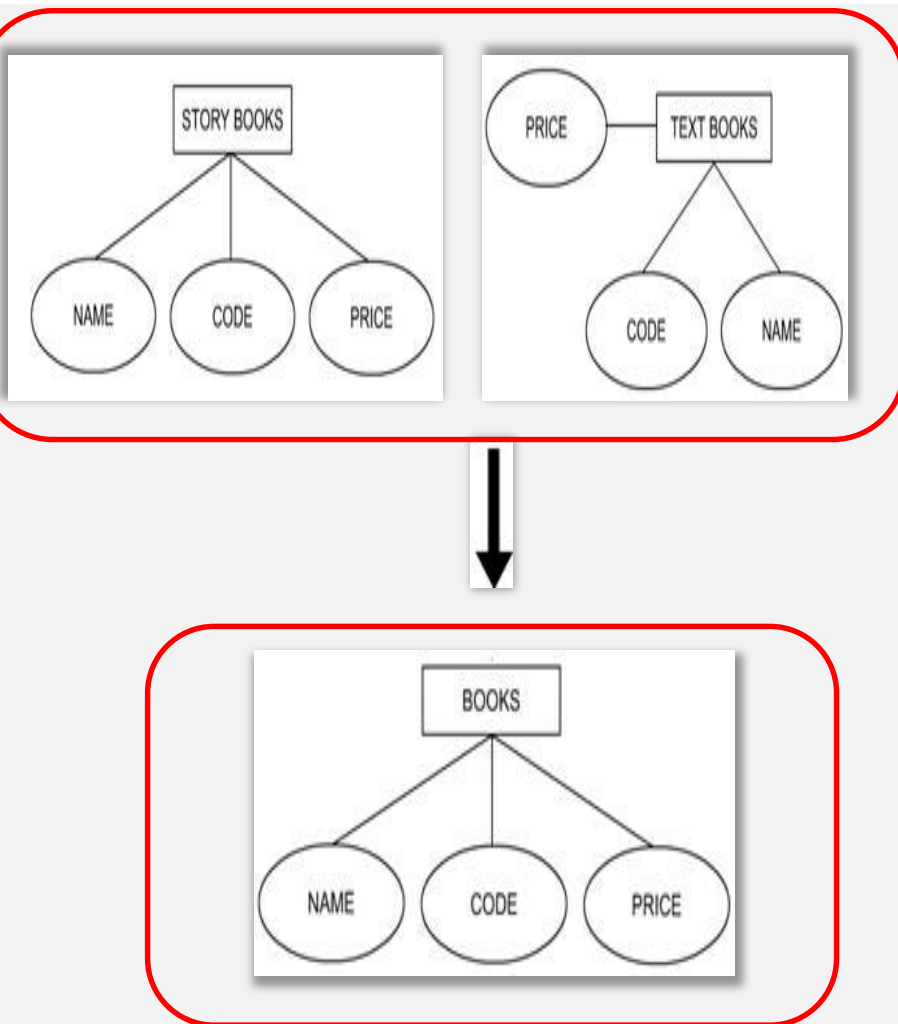
- ✓ A single person, place, or thing about which data can be stored
- ✓ Represents a chunk of information

02 Attributes

What is an **Attribute**?



Understanding Attributes

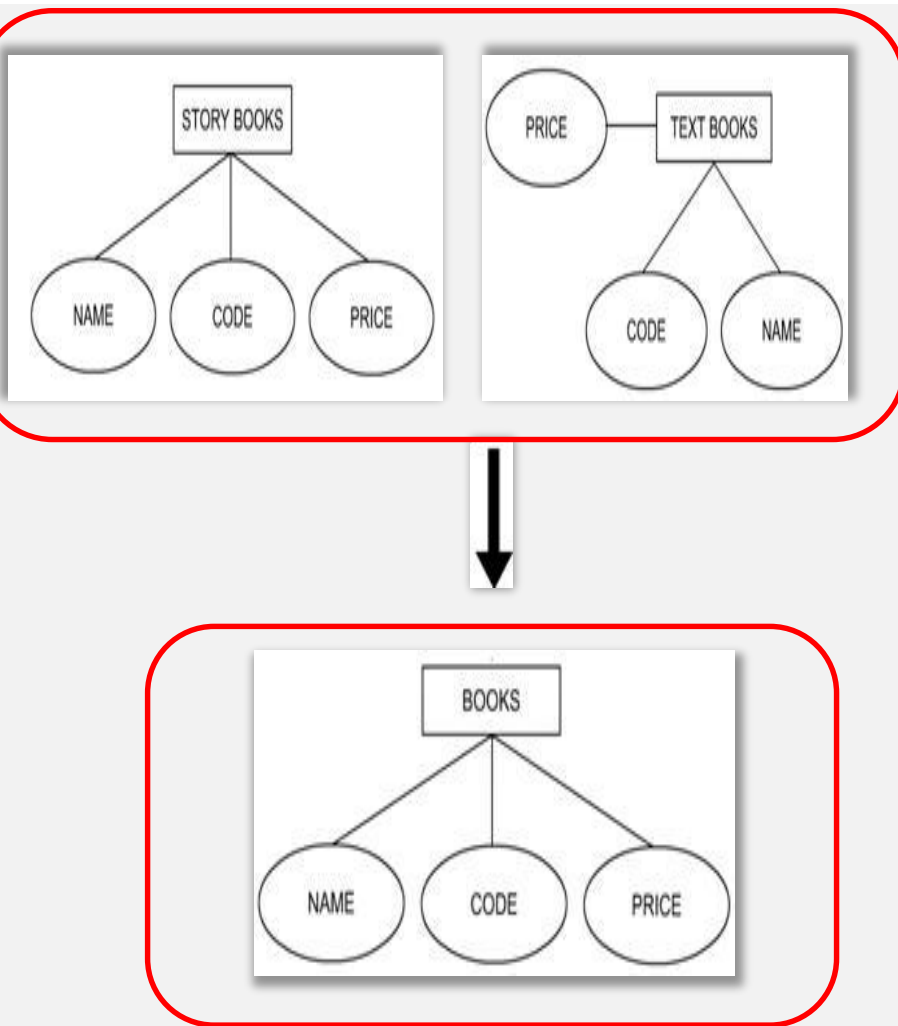


Let us understand Attributes with the help of a diagram. This diagram shows how you can merge two entities that contain common attributes.

Click the highlighted area on the diagram to learn more.

Every category of book will have NAME, CODE, and PRICE as its attributes.

Understanding Attributes



Let us understand Attributes with the help of a diagram. This diagram shows how you can merge two entities that contain common attributes.

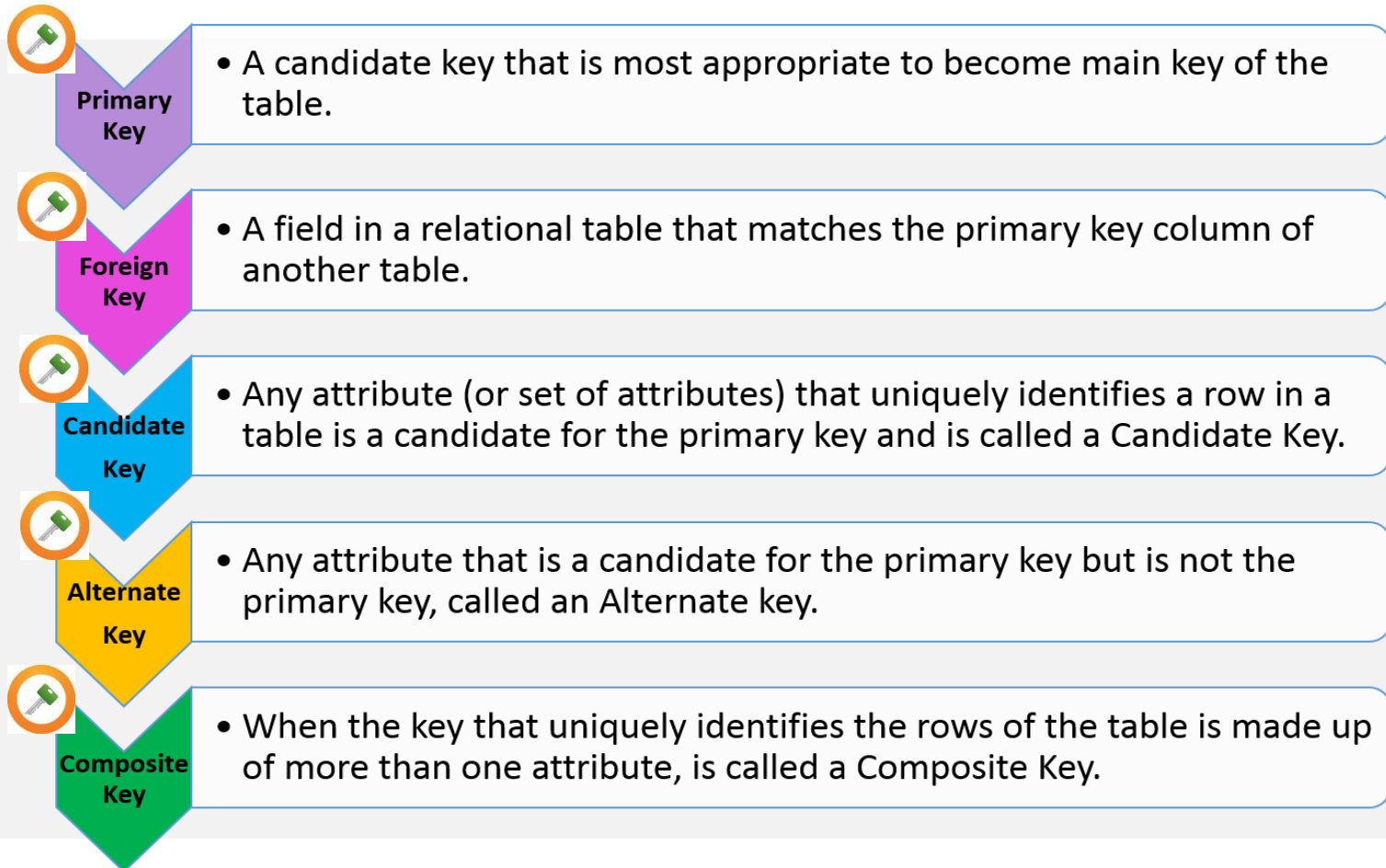
Click the highlighted area on the diagram to learn more.

Instead of making STORY BOOKS and TEXT BOOKS as different entities, you can create one entity, BOOKS.

03 Keys

Keys and their Types

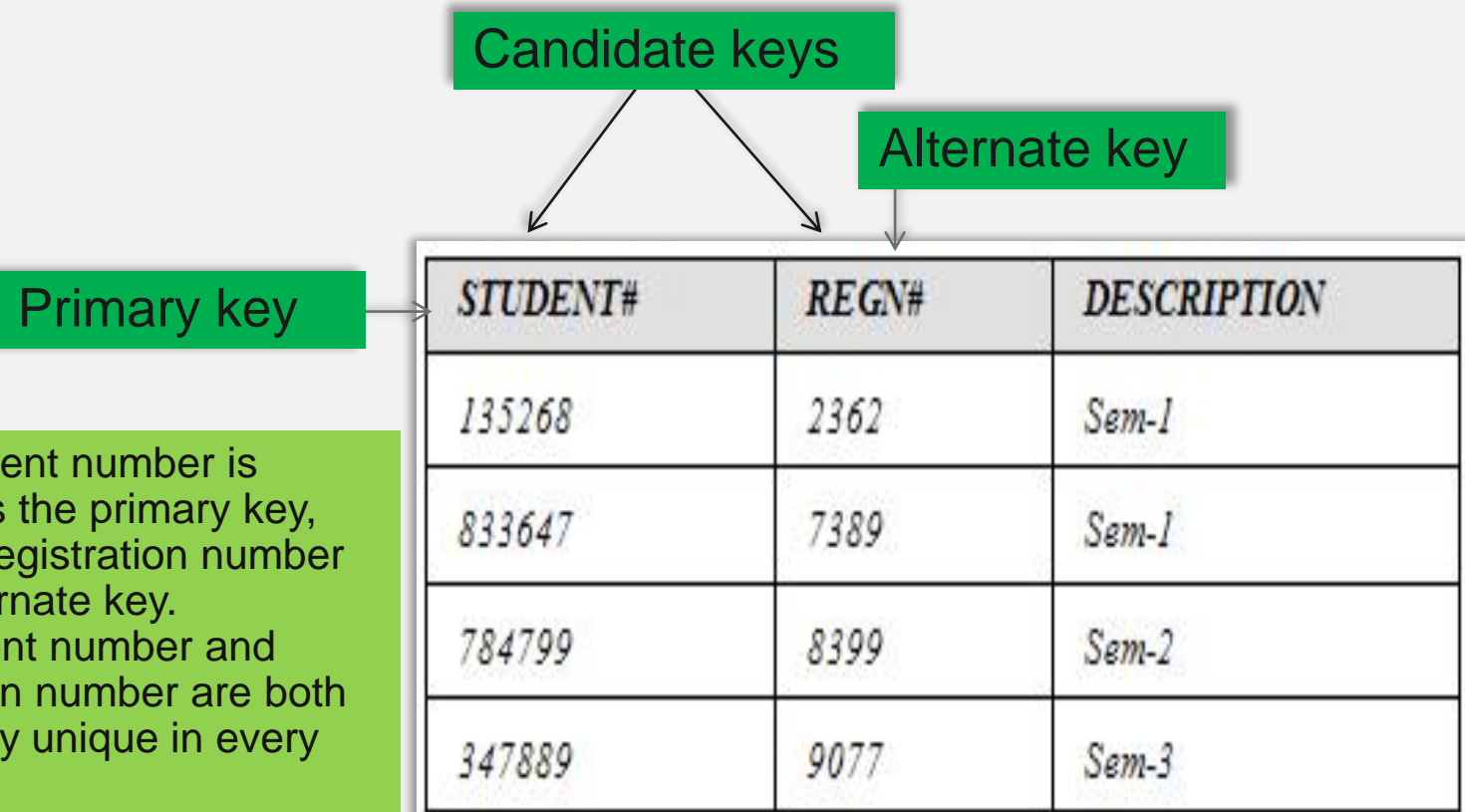
- Database Keys are used to establish and identify relations between tables. Let us learn about the different types of Keys.



Keys – Diagrammatic Representation

The diagram below depicts a STUDENT table.

Roll the mouse cursor over the diagram to know more.

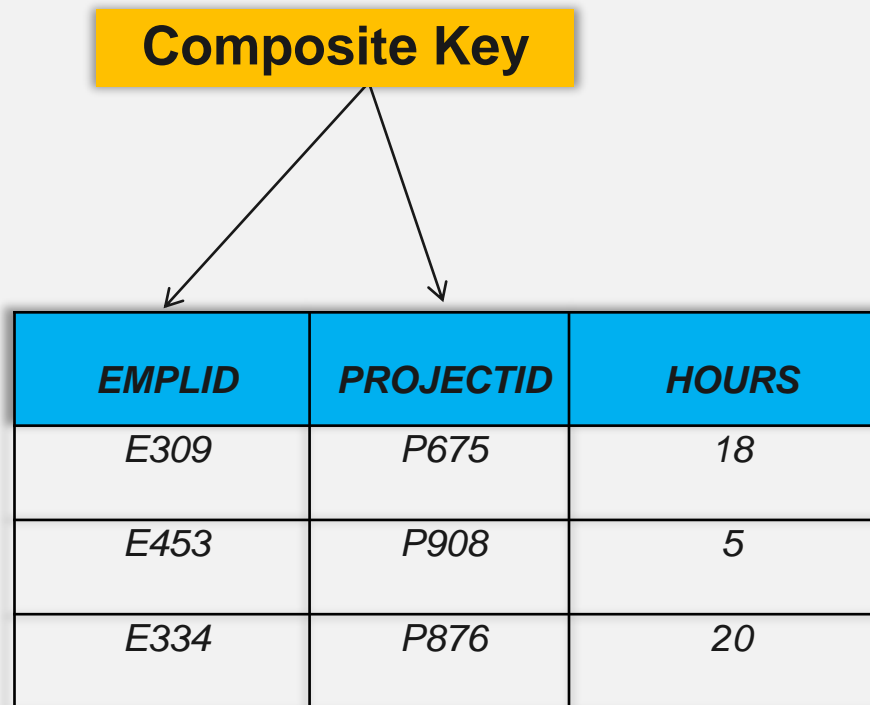


If the student number is chosen as the primary key, then the registration number is the alternate key. The student number and registration number are both individually unique in every row.

Keys – Diagrammatic Representation

The diagram below depicts the WORKING HOURS of an employee.

Roll the mouse cursor over the diagram to know more.



Each row can be uniquely identified by a composite key composed of EMPLID and PROJECTID.

Spot Quiz

03

Which of the following statements is true about Attributes?

A

It identifies its owner entity

B

It refers to another key

C

It simplifies the description of an entity

D

It describes the properties of keys

Spot Quiz

01

When the key that uniquely identifies the rows of the table is made up of more than one attribute, it called a _____

A

Primary Key

B

Candidate Key

C

Composite Key

D

Alternate Key

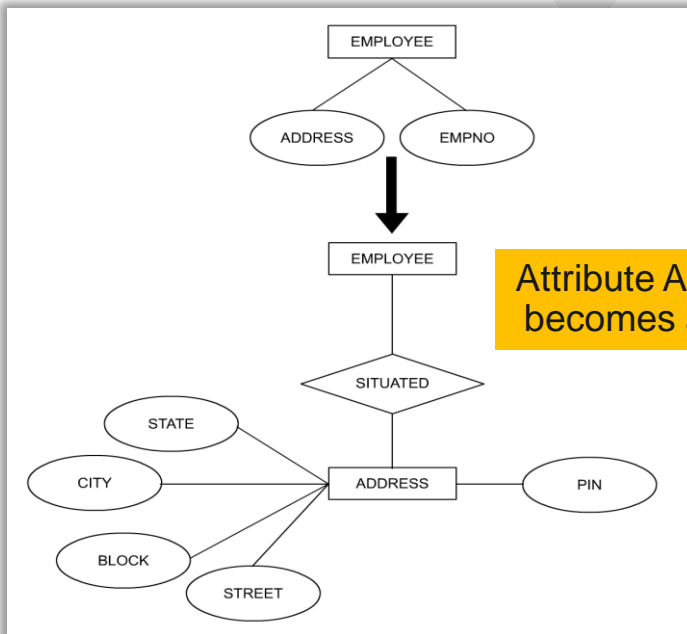
03 Entities & Sub-entities

What is an Entity?

Entity

An entity may be an object with a physical existence. It may refer to individuals, organizations, systems, bits of data, or even distinct system components.

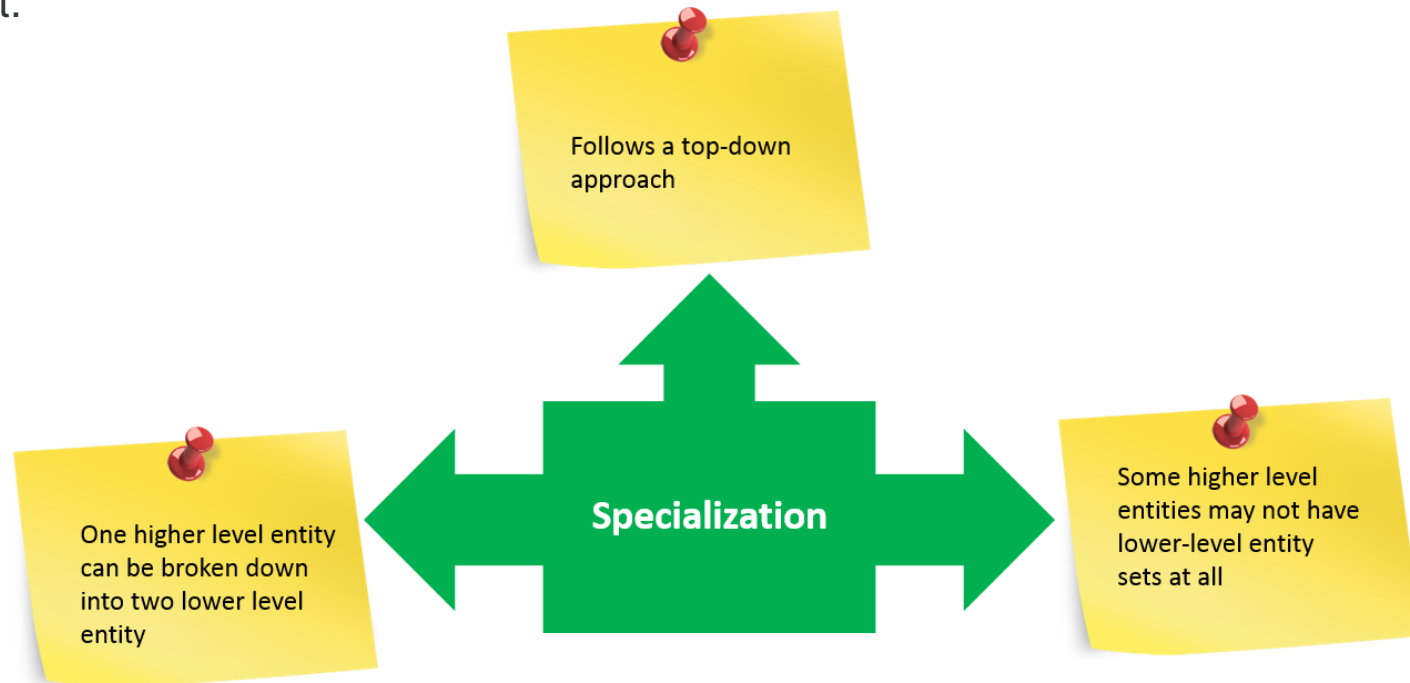
You can create a new entity to represent important recurring groups of attributes, as shown in the following diagram



Understanding Sub-entities – Specialization

Entities that have optional attributes can be replaced with sub-entities. This is called **Specialization**.

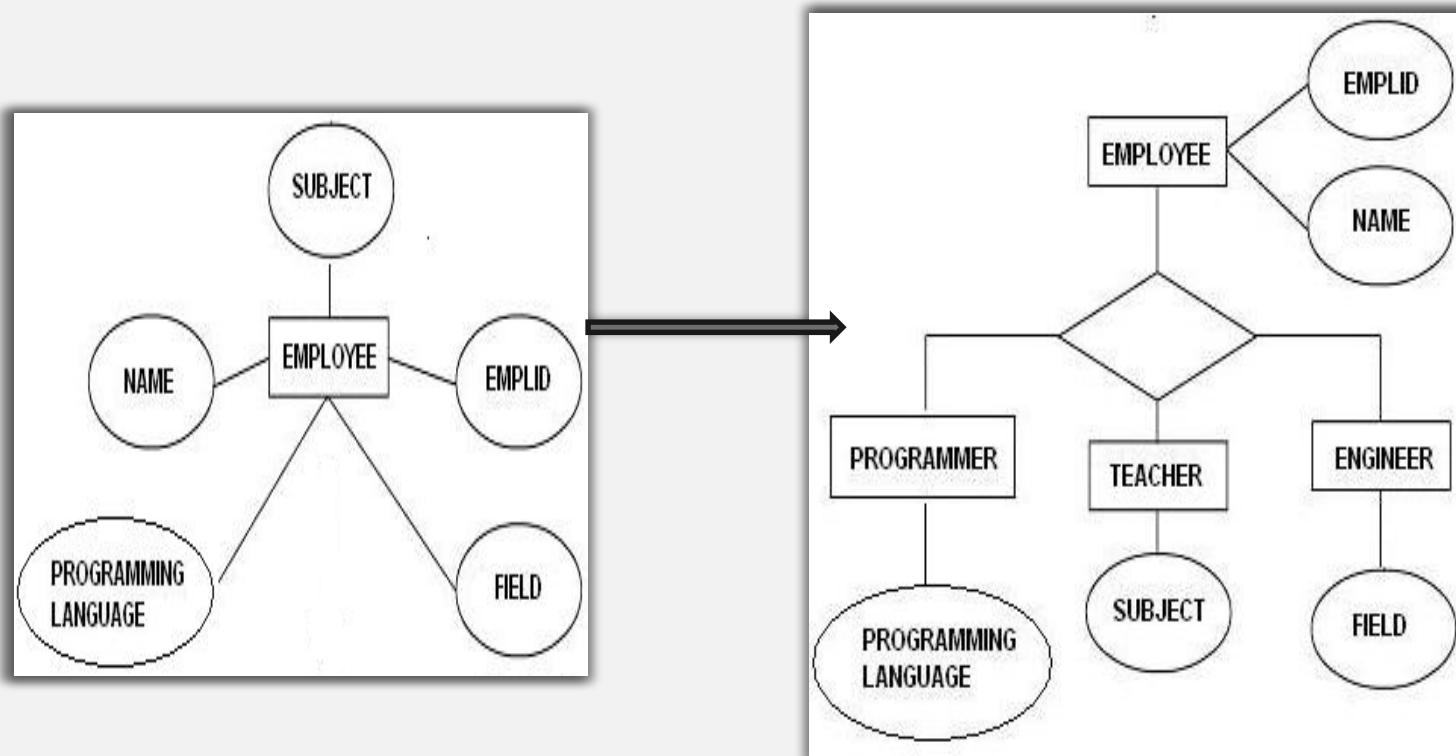
Specialization is the result of taking a subset of a higher-level entity set to form a lower-level entity set.



Specialization – Diagrammatic Representation

The diagram below depicts the concept of Specialization.

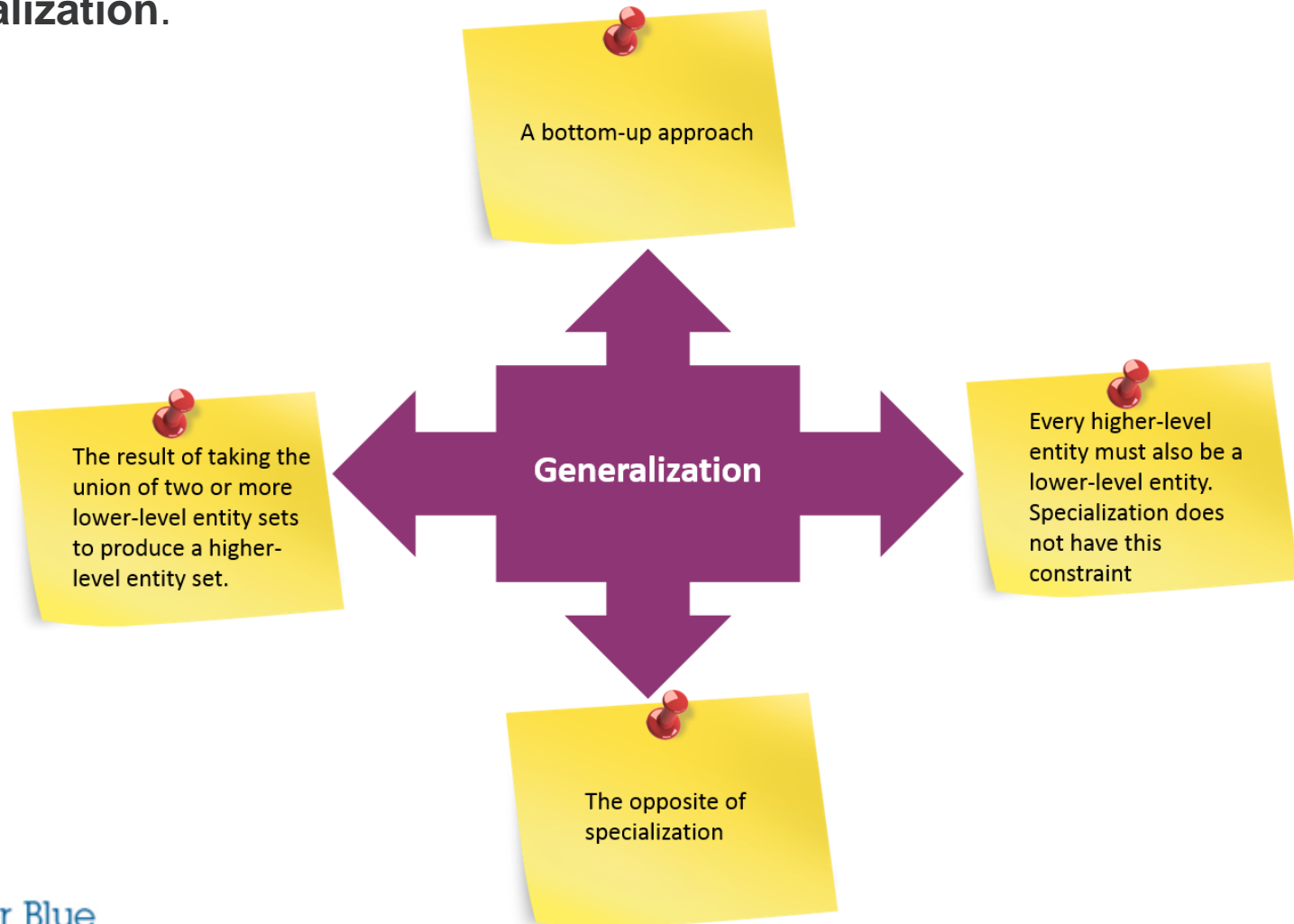
Roll the mouse cursor over the diagram to know more.



EMPLOYEE is the higher-level entity set, while PROGRAMMER, TEACHER, and ENGINEER are lower-level entity sets.

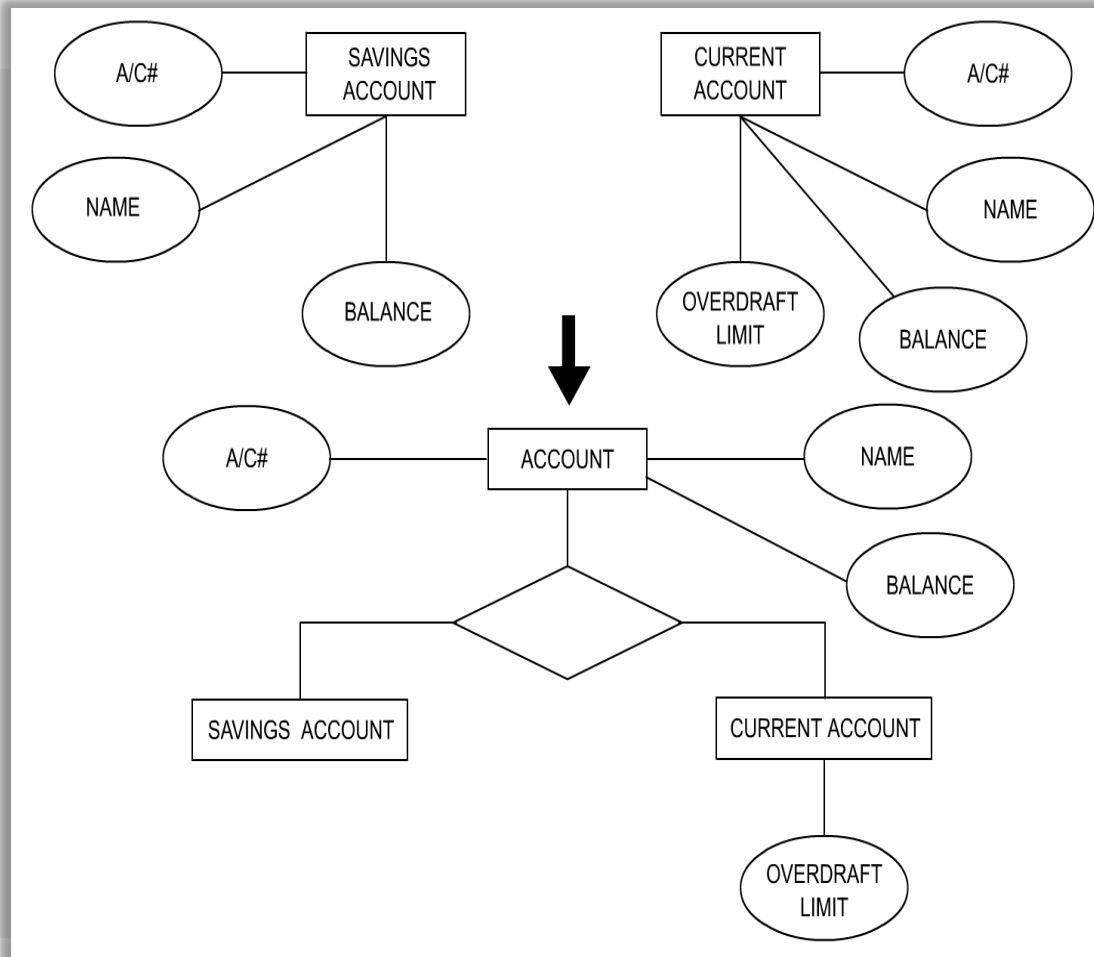
What is Generalization?

You learnt about Specialization in the previous screen. Let us now learn about **Generalization**.



Generalization – Diagrammatic Representation

The following diagram represents the concept of generalization.

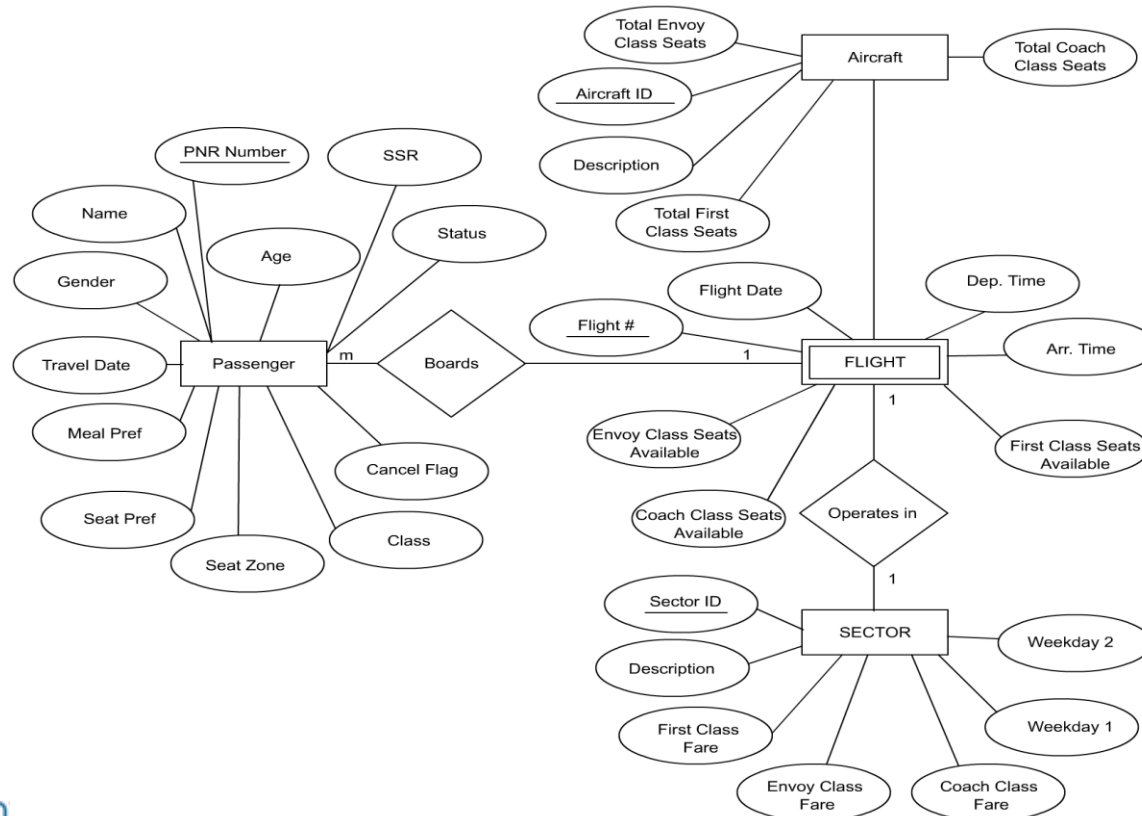


Lower-level entity sets (SAVING ACCOUNT and CURRENT ACCOUNT) produce a higher-level entity set (ACCOUNT).

04 Mapping Entity Relation Diagram (ERD) to Tables

Mapping ERD to Tables

Mapping an ER diagram to a table is one of the important stages when developing a database application. Ultimately, the relational model formed by mapping the ER diagram to the tables, will be implemented as a set of tables and columns in a database management system. An ER diagram is depicted below.



Spot Quiz

03

Which of the following is true regarding Generalization?

A

Follows a top-down approach

B

The result of taking the union of two or more lower-level entity sets to produce a higher-level entity set.

C

Some higher level entities may not have lower-level entity sets at all

D

Every higher-level entity must also be a lower-level entity.

Spot Quiz

01

In the three-layer DBMS architecture, which level defines how each group of end-users sees the organization of data in the database?

A

Internal level

B

External Level

C

Conceptual Level

D

Physical level

Module Summary

Now that you have completed this module on **Best Practices of Logical Database Design**, you should be able to:

- Identify the best practices of logical database design.
- Recognize that the entities in an ER diagram can be mapped to tables.