SWA261

The Analysis Phase: Data Modelling

BELGIUM CAMPUS iTversity

Outcomes

Students should understand the following outcomes, upon successful completion of this module:

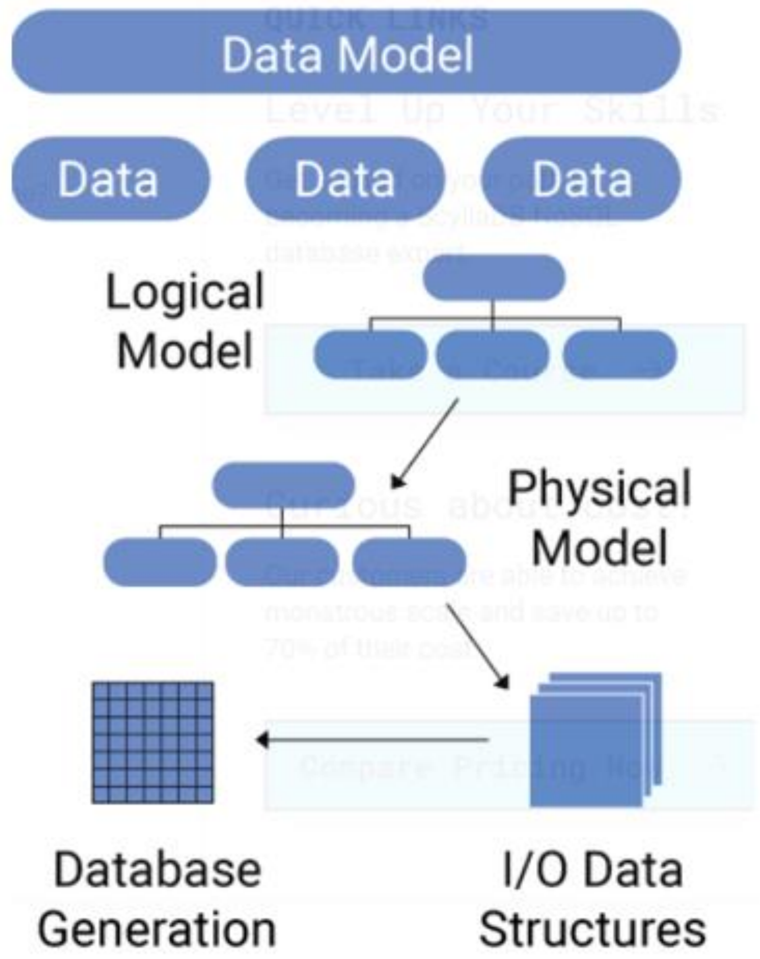
- Explain the rules and style guidelines for creating entity relationship diagrams (ERDs).
- Create an ERD.
- Describe the use of a data dictionary and metadata

Introduction

- A data model describes the data that flow through the business processes in an organization.

- During the analysis phase, the data model presents the logical organization of data without indicating how the data are stored, created, or manipulated so that analysts can focus on the business without being distracted by technical details.

- Later, during the design phase, the data model is changed to reflect exactly how the data will be stored in databases and files.



Data Model

- A data model is a formal way of representing the data that are used and created by a business system;
  - it illustrates people, places, or things about which information is captured and how they are related to each other.

- The data model is drawn by an iterative process in which the model becomes more detailed and less conceptual over time.

- During the *analysis phase*, analysts draw a *logical data model*, which shows the logical organization of data without indicating how data are stored, created, or manipulated.

- Because this model is free of any implementation or technical details, the analysts can focus more easily on matching the diagram to the real business requirements of the system.

- In the *design phase*, analysts draw a physical data model to reflect how the data will physically be stored in databases and files.

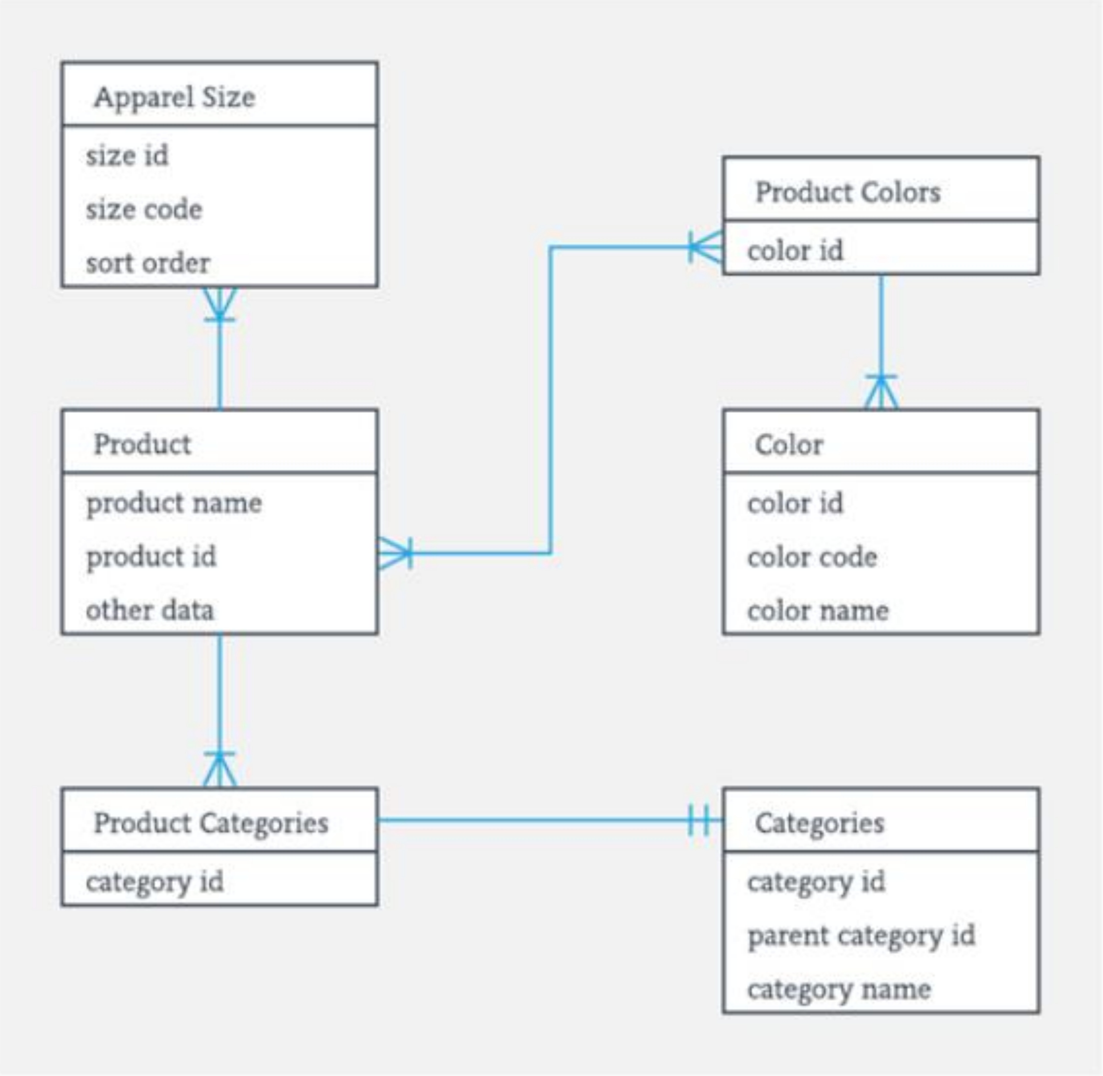
Entity Relationship Diagram - ERD

- An entity relationship diagram (ERD) is a picture which shows the information that is created, stored, and used by a business system.

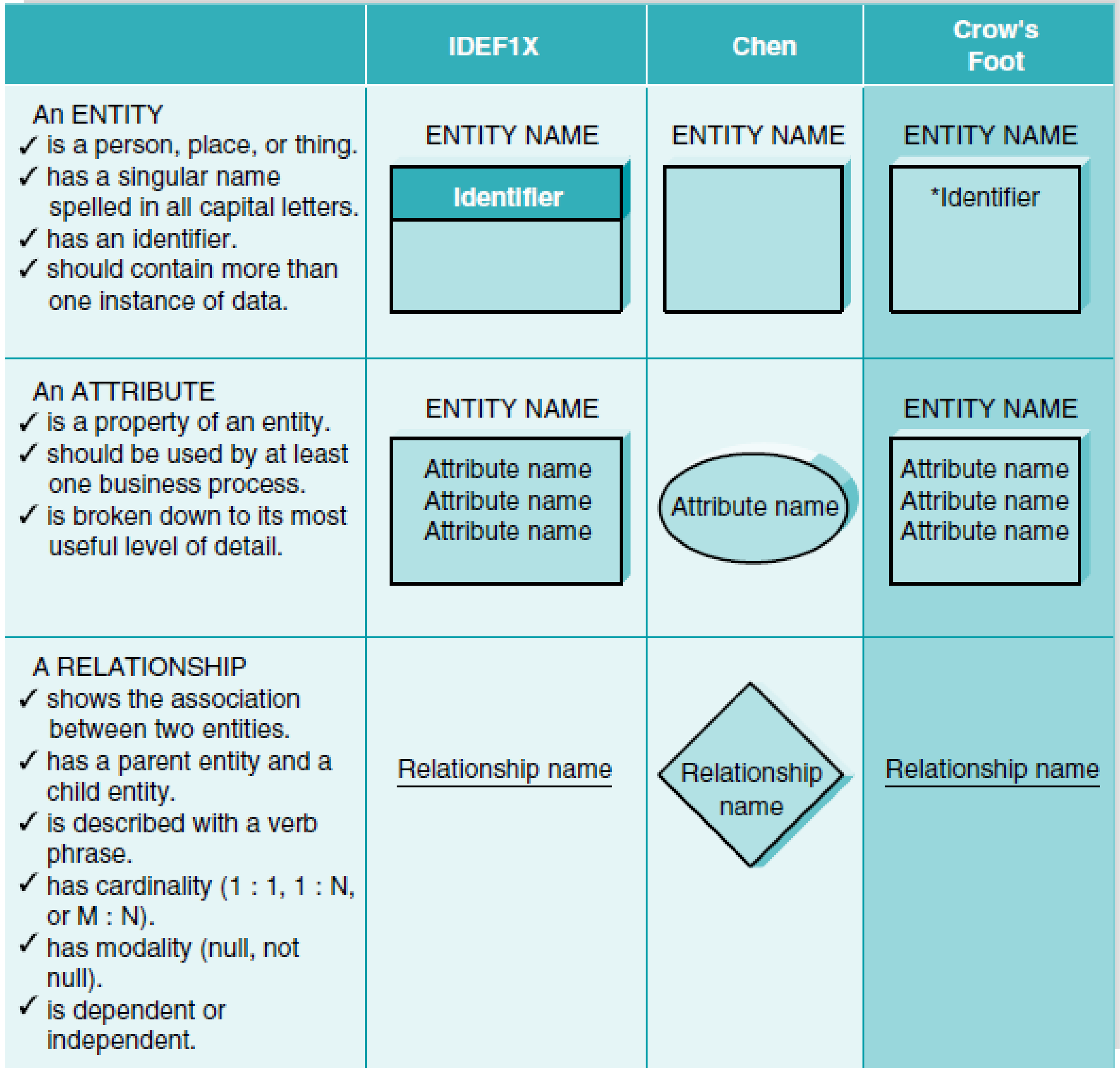
- An analyst can read an ERD to discover the individual pieces of information in a system and how they are organized and related to each other.

- On an ERD, similar kinds of information are listed together and placed inside boxes called entities.

- Lines are drawn between entities to represent relationships among the data, and special symbols are added to the diagram to communicate high-level business rules that need to be supported by the system.



Elements of an ERD



	IDEF1X	Chen	Crow's Foot
An ENTITY ✓ is a person, place, or thing. ✓ has a singular name spelled in all capital letters. ✓ has an identifier. ✓ should contain more than one instance of data.	ENTITY NAME Identifier	ENTITY NAME	ENTITY NAME *Identifier
An ATTRIBUTE ✓ is a property of an entity. ✓ should be used by at least one business process. ✓ is broken down to its most useful level of detail.	ENTITY NAME Attribute name Attribute name Attribute name	Attribute name	ENTITY NAME Attribute name Attribute name Attribute name
A RELATIONSHIP ✓ shows the association between two entities. ✓ has a parent entity and a child entity. ✓ is described with a verb phrase. ✓ has cardinality (1 : 1, 1 : N, or M : N). ✓ has modality (null, not null). ✓ is dependent or independent.	Relationship name	Relationship name	Relationship name

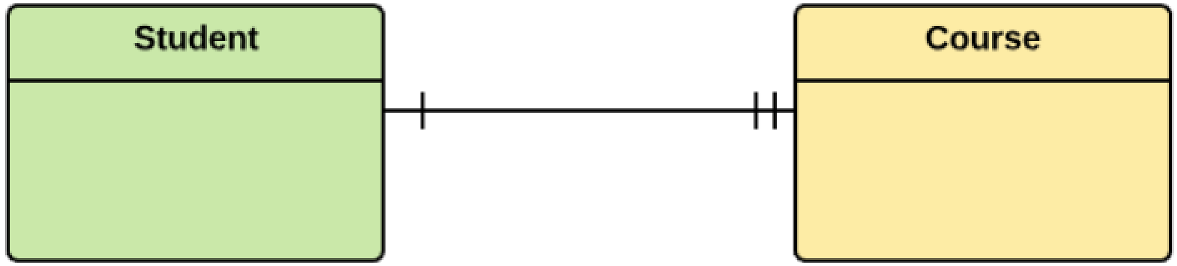
The *entity* is the basic building block for a data model. It is a person, place, event, or thing about which data is collected—for example, an employee, an order, or a product.

An ***attribute*** is some type of information that is captured about an entity. For example, last name, home address, and e-mail address are all attributes of an employee.

Relationships are associations between entities, and they are shown by lines that connect the entities together. Every relationship has a parent entity and a child entity, the parent being the first entity in the relationship, and the child being the second.

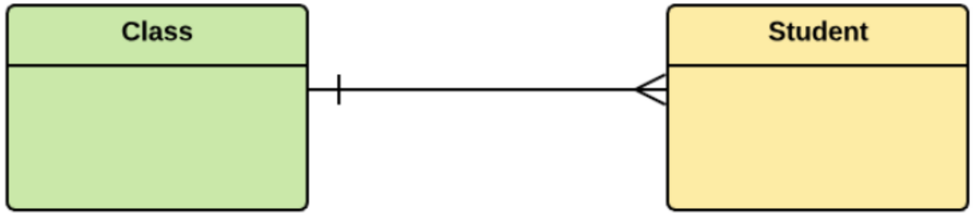
Cardinality

- Relationships have two properties that convey meaning about the nature of the entity relationship.
- First, a relationship has *cardinality*, which is the ratio of parent instances to child instances.
- To determine the cardinality for a relationship, we ask ourselves: "*How many instances of one entity are associated with an instance of the other?*"
- The cardinality is for binary relationships (i.e., relationships between two entities) is defined in the following ways:
  - ✓ 1:1 - (read as "one to one") relationship means that one instance of the parent entity is associated with exactly one instance of the child entity:

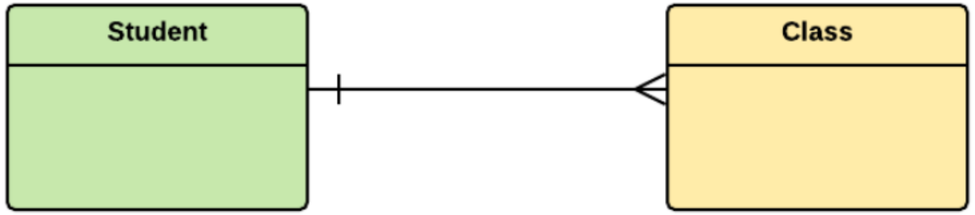


Cardinality

- ✓ 1:N – read as "one to many" relationship: a single instance of a parent entity is associated with many instances of a child entity; however, the child entity instance is related to only one instance of the parent.



- ✓ M:N - read as "many to many" relationship: many instances of an entity can relate to many instances of another entity. As a result, it is difficult to determine the parent and the child entities in the relationship.



Modality

- The second important property that characterizes the nature of the relationship between two entities is termed modality.

- As cardinality tells you the maximum number of times that an instance of an entity can be associated with instances of the other entity, *Modality* tells you the minimum number of times that the instance can be associated with instances of the other entity.

- Relationships have a modality of either “required” or “optional,” which refers to whether an instance of an entity can exist without a related instance in the related entity.

- Basically, the modality of a relationship indicates whether one entity instance is required to participate in the relationship.

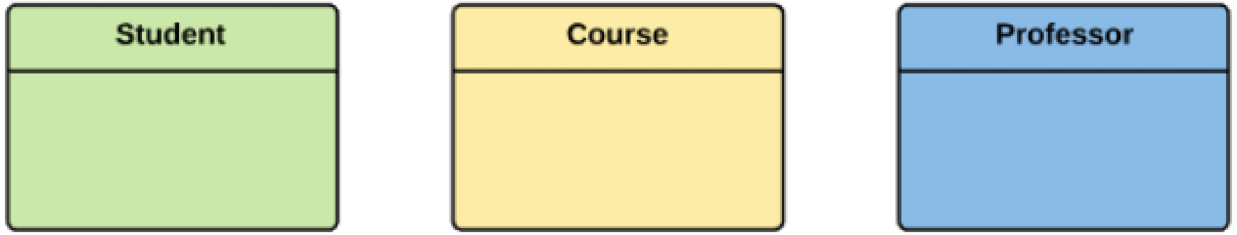
- Modality indicates whether the relationship between an entity instance and an instance of the related entity is “null” (optional) or “not null” (required).

How to Create an ERD

Step 1) Entity Identification

We have three entities

- Student
- Course
- Professor

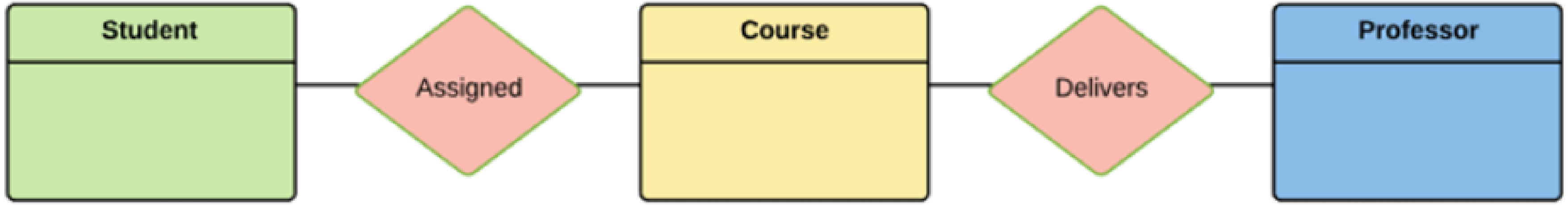


How to Create an ERD

Step 2) Relationship Identification

We have the following two relationships

- The student is **assigned** a course
- Professor **delivers** a course

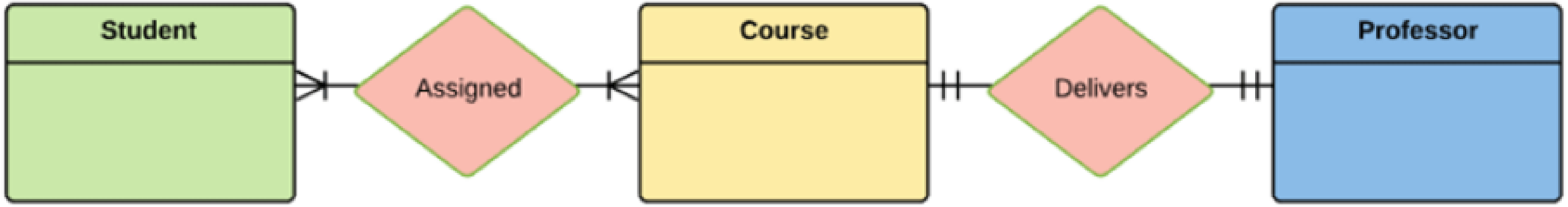


How to Create an ERD

Step 3) Cardinality Identification

For them problem statement we know that,

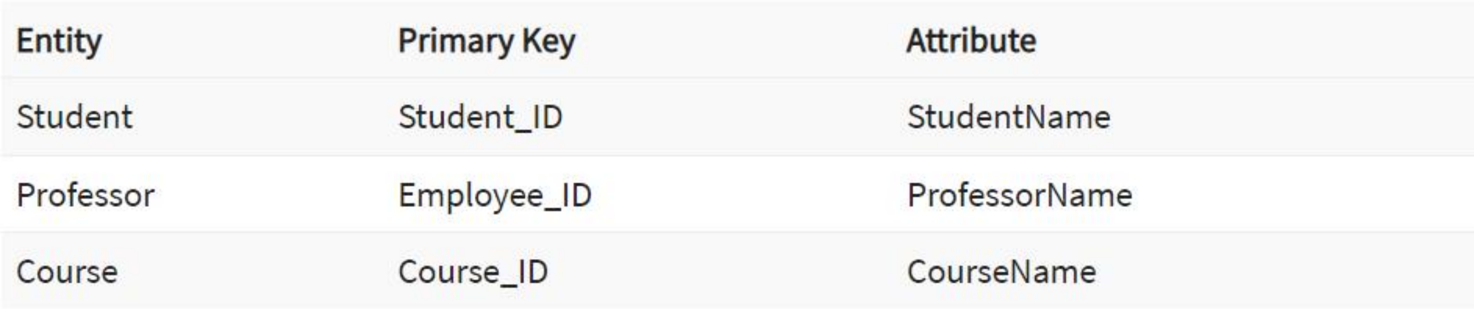
- A student can be assigned **multiple** courses
- A Professor can deliver only **one** course



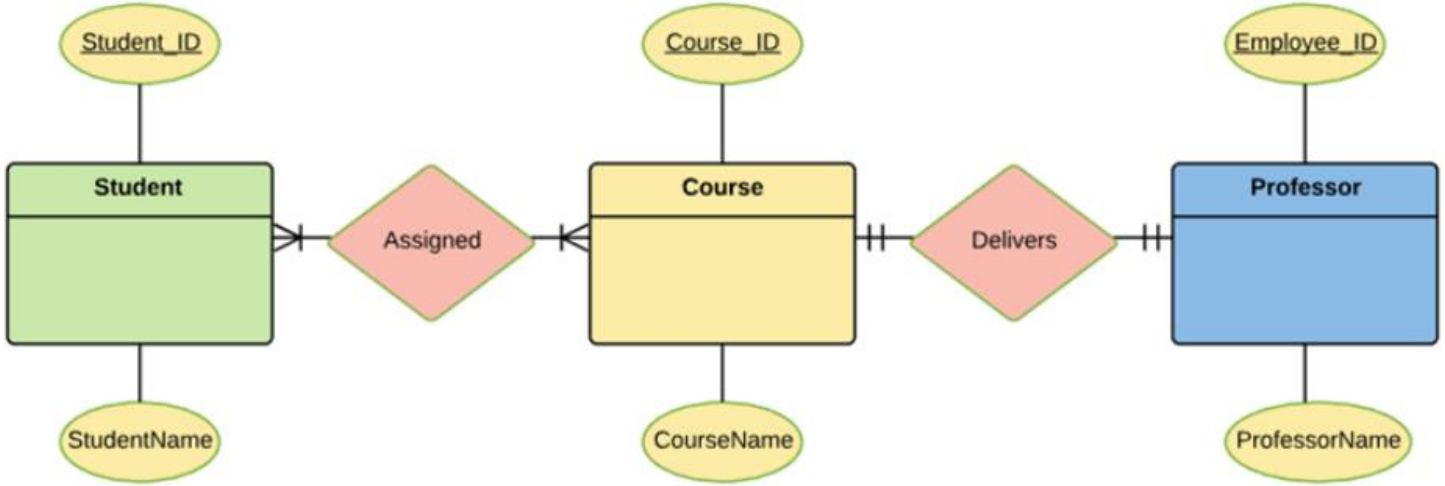
How to Create an ERD

Step 4) Identify Attributes

You need to study the files, forms, reports, data currently maintained by the organization to identify attributes. You can also conduct interviews with various stakeholders to identify entities. Initially, it's important to identify the attributes without mapping them to a particular entity.



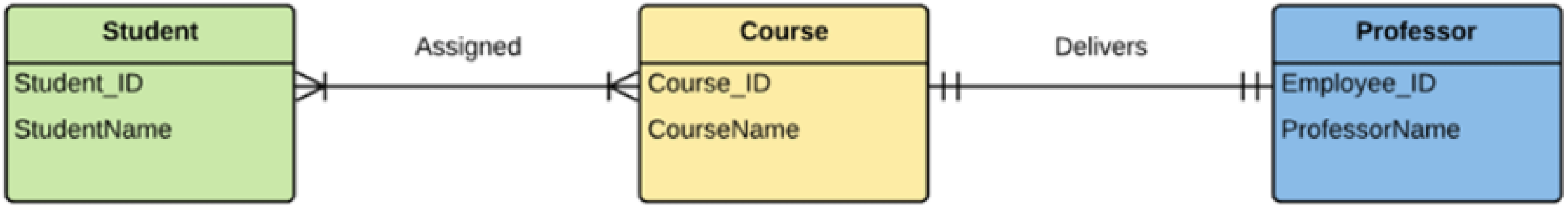
Entity	Primary Key	Attribute
Student	Student_ID	StudentName
Professor	Employee_ID	ProfessorName
Course	Course_ID	CourseName

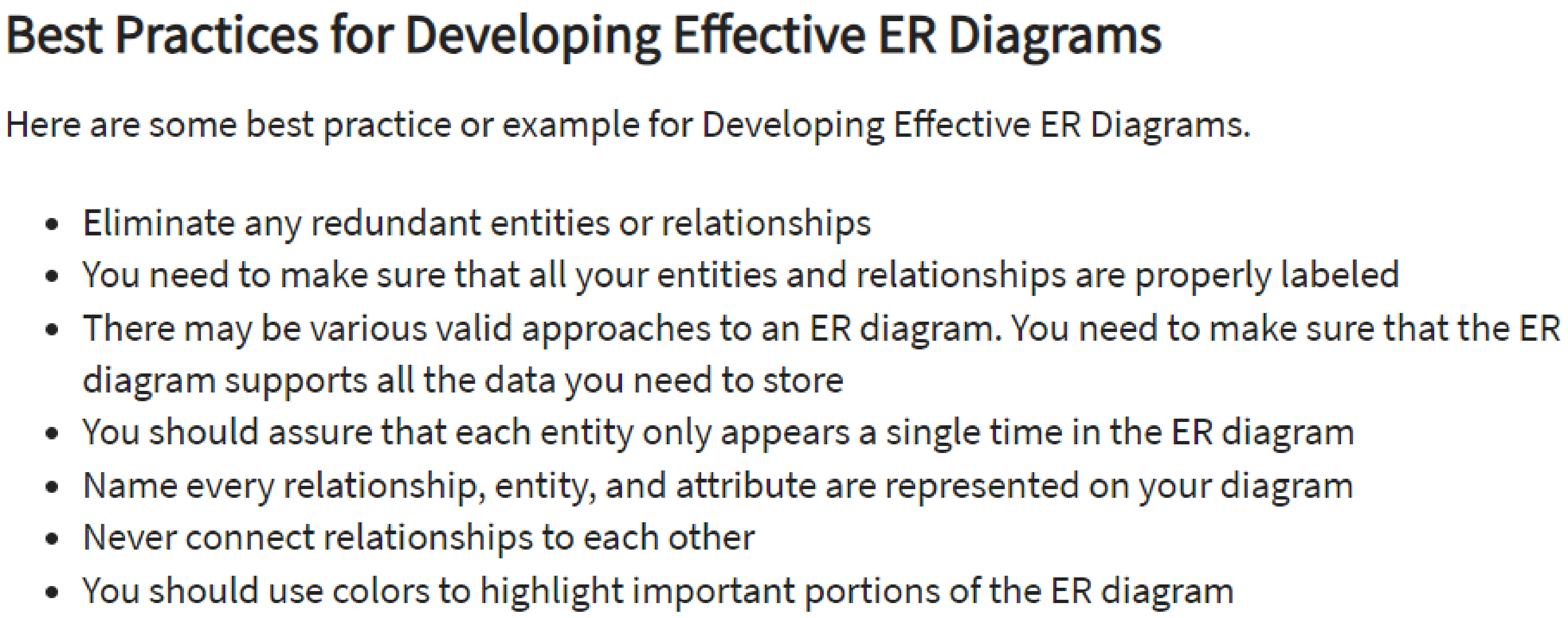


How to Create an ERD

Step 5) Create the ERD Diagram

A more modern representation of Entity Relationship Diagram Example





Best Practices for Developing Effective ER Diagrams

Here are some best practice or example for Developing Effective ER Diagrams.

- Eliminate any redundant entities or relationships
- You need to make sure that all your entities and relationships are properly labeled
- There may be various valid approaches to an ER diagram. You need to make sure that the ER diagram supports all the data you need to store
- You should assure that each entity only appears a single time in the ER diagram
- Name every relationship, entity, and attribute are represented on your diagram
- Never connect relationships to each other
- You should use colors to highlight important portions of the ER diagram

The Data Dictionary and Metadata

- CASE tools are often used to help build ERDs.
- Every CASE tool has something called a *data dictionary*, which defines where the analyst goes to define or look up information about the entities, attributes, and relationships on the ERD.
- The information you see in the data dictionary is called *metadata*, which, quite simply, is data about data.
- *Metadata* is anything that describes an entity, attribute, or relationship, such as entity names, attribute descriptions, and relationship cardinality, and it is captured to help designers better understand the system that they are building and to help users better understand the system that they will use.
- Metadata are stored in the data dictionary so that they can be shared and accessed by developers and users throughout the SDLC

Thank You!

THE END

info@belgiumcampus.ac.za

+27 10 593 53 68

/belgiumcampusSA

#Belgium Campus

/belgiumcampus