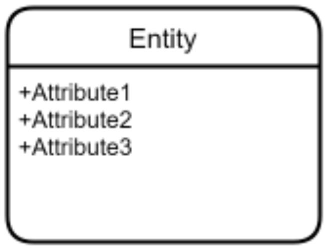
This reading covers the Entity Relationship Diagram (ER-D) and demonstrates how to use it to support the design of a relational database.
The relational database model organizes information into tables to ensure a good data structure to maintain consistency and accuracy, which makes the design of the tables and their relationships very crucial. The relational database design is very well connected with the entity relationship modelling process including entities, attributes and relationship identification and definition. The entity-relationship diagram (ER-D) is commonly used to represent and document the entity relationship models.
The use of entity relationship diagrams helps to provide the big picture of your database. It also ensures the data requirements and operations are well defined and documented in your project. In addition, the ER-D represents a blueprint that guides database developers through the implementation of the actual database in a relevant database management system such as Oracle and MySQL.
The entities, attributes and relationships between entities can be shown in a variety of diagrammatic formats in the ER diagrams. In this reading, you'll review the most used shapes and symbols.

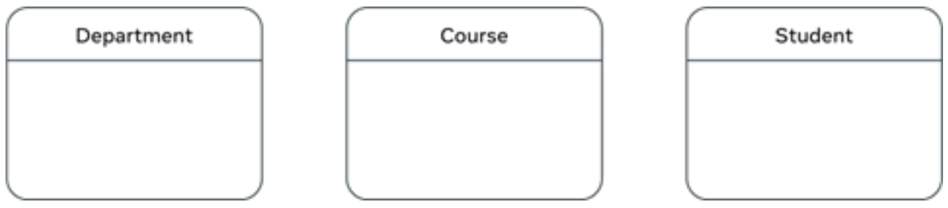
Entity representation

In the ER-D, a box with two compartments is used to represent the entity and its related attributes. The top compartment represents the entity name, and the bottom compartment includes the related attributes.



For example, a college enrollment system contains a database with information about the students, and the courses available in each department.
In this case, you can have three entities represented in three separate boxes:

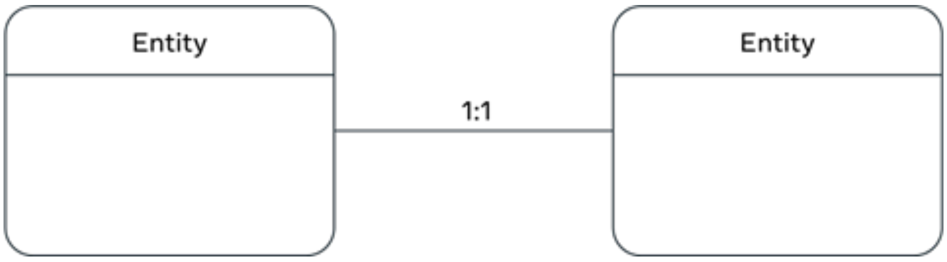
- the student entity,
- the course entity,
- and the department entity.



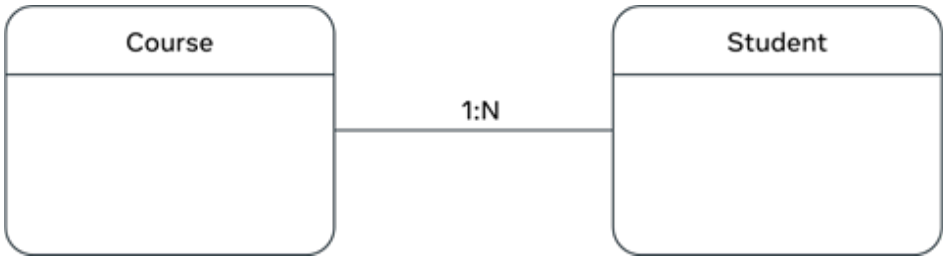
There's no point in considering entities or attributes that will not be used in your project. You should only capture data that helps the users of your database system to complete certain tasks and activities.

Relationship representation
The ER diagram uses different styles of lines to define the distinct types of relationships

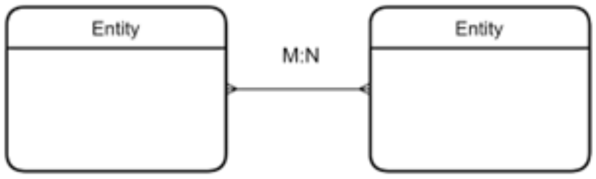
between entities. The line style depends on the cardinality of the relationship, which refers to the number of elements in a set of data as clarified in the following three cases.
1:1 (one-to-one): The ER-D uses a straight-line representation for a one-to-one cardinality relationship. For example, each passenger on a train should have only one ticket.



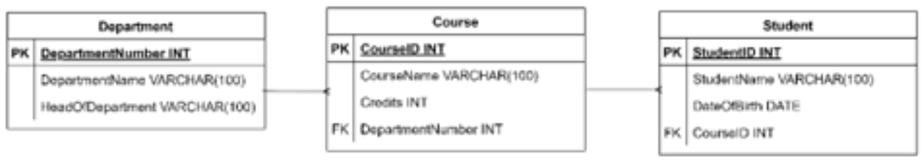
1:N (one-to-many): The ER-D is a straight line with a crow's foot notation on one side only to represent a one-to-many cardinality relationship. For example, one parent can have many children.



M:N (many-to-many): The ER-D is a straight line with crow's foot notations on both sides of entities to represent a many-to-many cardinality relationship. For example, many players play many games.



Based on this explanation, how would you depict the relationship between the student, course, and department entities introduced earlier in the college enrolment system example? Remember that many students may enroll in one course, and one department may offer many courses.
In this example, you can use a one-to-many relationship where the crows-foot notation is used to show that "many students" are enrolled in one specific "course", and "many courses" belong to one specific department.

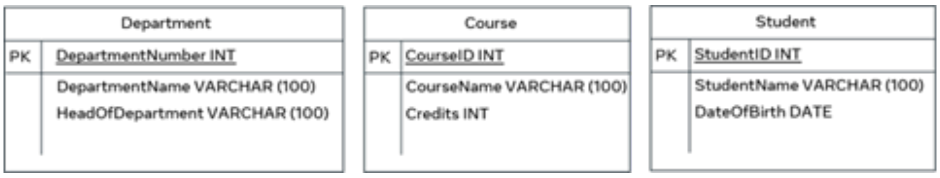


Attributes representation
Each entity has a set of attributes that hold relevant information about it. Each attribute must

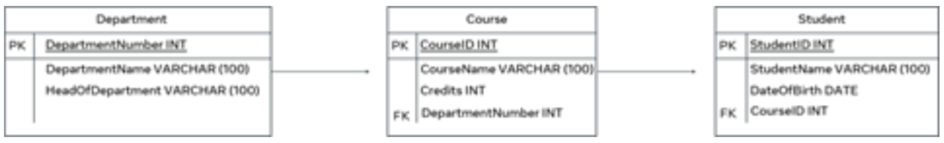
be defined with a data type.
In the college enrolment example, you can list the following attributes followed by relevant data types:

- The department attributes: department number, department name and head of department.
- The course attributes: course ID, course name, and course credits.
- The student attributes: student ID, name, and date of birth.

The three entities can be listed as three separate tables.



However, you need to add the department number to the course table as a foreign key in order to link the courses with the departments.
Similarly, you need to add the course ID to the student table as a foreign key in order to link the students with the courses. The final college enrolment entity relationship diagram is then three separate tables connected by the relevant keys.



In this reading, you learned how to use the ER diagram to illustrate and document a relational database system.