[](http://www.tutorialspoint.com/index.htm)

Top of Form



Bottom of Form

* [Quizzes](http://www.tutorialspoint.com/questions_and_answers.htm)
* [Shared](http://www.tutorialspoint.com/shared-tutorials.php)
* [Articles](http://www.tutorialspoint.com/articles/index.php)



* DBMS Tutorial
* [DBMS - Home](http://www.tutorialspoint.com/dbms/index.htm)
* [DBMS - Overview](http://www.tutorialspoint.com/dbms/dbms_overview.htm)
* [DBMS - Architecture](http://www.tutorialspoint.com/dbms/dbms_architecture.htm)
* [DBMS - Data Models](http://www.tutorialspoint.com/dbms/dbms_data_models.htm)
* [DBMS - Data Schemas](http://www.tutorialspoint.com/dbms/dbms_data_schemas.htm)
* [DBMS - Data Independence](http://www.tutorialspoint.com/dbms/dbms_data_independence.htm)
* Entity Relationship Model
* [DBMS - ER Model Basic Concepts](http://www.tutorialspoint.com/dbms/er_model_basic_concepts.htm)
* [DBMS - ER Diagram Representation](http://www.tutorialspoint.com/dbms/er_diagram_representation.htm)
* [DBMS - Generalization, Aggregation](http://www.tutorialspoint.com/dbms/dbms_generalization_aggregation.htm)
* Relational Model
* [DBMS - Codd's Rules](http://www.tutorialspoint.com/dbms/dbms_codds_rules.htm)
* [DBMS - Relational Data Model](http://www.tutorialspoint.com/dbms/relational_data_model.htm)
* [DBMS - Relational Algebra](http://www.tutorialspoint.com/dbms/relational_algebra.htm)
* [DBMS - ER to Relational Model](http://www.tutorialspoint.com/dbms/er_model_to_relational_model.htm)
* [DBMS- SQL Overview](http://www.tutorialspoint.com/dbms/sql_overview.htm)
* Relational Database Design
* [DBMS - Database Normalization](http://www.tutorialspoint.com/dbms/database_normalization.htm)
* [DBMS - Database Joins](http://www.tutorialspoint.com/dbms/database_joins.htm)
* Storage and File Structure
* [DBMS - Storage System](http://www.tutorialspoint.com/dbms/dbms_storage_system.htm)
* [DBMS - File Structure](http://www.tutorialspoint.com/dbms/dbms_file_structure.htm)
* Indexing and Hashing
* [DBMS - Indexing](http://www.tutorialspoint.com/dbms/dbms_indexing.htm)
* [DBMS - Hashing](http://www.tutorialspoint.com/dbms/dbms_hashing.htm)
* Transaction And Concurrency
* [DBMS - Transaction](http://www.tutorialspoint.com/dbms/dbms_transaction.htm)
* [DBMS - Concurrency Control](http://www.tutorialspoint.com/dbms/dbms_concurrency_control.htm)
* [DBMS - Deadlock](http://www.tutorialspoint.com/dbms/dbms_deadlock.htm)
* Backup and Recovery
* [DBMS - Data Backup](http://www.tutorialspoint.com/dbms/dbms_data_backup.htm)
* [DBMS - Data Recovery](http://www.tutorialspoint.com/dbms/dbms_data_recovery.htm)
* DBMS Useful Resources
* [DBMS - Quick Guide](http://www.tutorialspoint.com/dbms/dbms_quick_guide.htm)
* [DBMS - Useful Resources](http://www.tutorialspoint.com/dbms/dbms_useful_resources.htm)
* [DBMS - Discussion](http://www.tutorialspoint.com/dbms/dbms_discussion.htm)
* Selected Reading
* [Developer's Best Practices](http://www.tutorialspoint.com/developers_best_practices/index.htm)
* [Questions and Answers](http://www.tutorialspoint.com/questions_and_answers.htm)
* [Effective Resume Writing](http://www.tutorialspoint.com/effective_resume_writing.htm)
* [HR Interview Questions](http://www.tutorialspoint.com/hr_interview_questions/index.htm)
* [Computer Glossary](http://www.tutorialspoint.com/computer_glossary.htm)
* [Who is Who](http://www.tutorialspoint.com/computer_whoiswho.htm)

ER Model - Basic Concepts

Advertisements

[Previous Page](http://www.tutorialspoint.com/dbms/dbms_data_independence.htm)

[Next Page](http://www.tutorialspoint.com/dbms/er_diagram_representation.htm)

The ER model defines the conceptual view of a database. It works around real-world entities and the associations among them. At view level, the ER model is considered a good option for designing databases.

Entity

An entity can be a real-world object, either animate or inanimate, 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. An entity set may contain entities with attribute sharing similar values. For example, a Students set may contain all the students of a school; likewise a Teachers set may contain all the teachers of a school from all faculties. Entity sets need not be disjoint.

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.

There exists a domain or range of values that can be assigned to attributes. For example, a student's name cannot be a numeric value. It has to be alphabetic. A student's age cannot be negative, etc.

Types of Attributes

* **Simple attribute** − Simple attributes are atomic values, which cannot be divided further. For example, a student's phone number is an atomic value of 10 digits.
* **Composite attribute** − Composite attributes are made of more than one simple attribute. For example, a student's complete name may have first\_name and last\_name.
* **Derived attribute** − Derived attributes are the attributes that do not exist in the physical database, but their values are derived from other attributes present in the database. For example, average\_salary in a department should not be saved directly in the database, instead it can be derived. For another example, age can be derived from data\_of\_birth.
* **Single-value attribute** − Single-value attributes contain single value. For example − Social\_Security\_Number.
* **Multi-value attribute** − Multi-value attributes may contain more than one values. For example, a person can have more than one phone number, email\_address, etc.

These attribute types can come together in a way like −

* simple single-valued attributes
* simple multi-valued attributes
* composite single-valued attributes
* composite multi-valued attributes

Entity-Set and Keys

Key is an attribute or collection of attributes that uniquely identifies an entity among entity set.

For example, the roll\_number of a student makes him/her identifiable among students.

* **Super Key** − A set of attributes (one or more) that collectively identifies an entity in an entity set.
* **Candidate Key** − A minimal super key is called a candidate key. An entity set may have more than one candidate key.
* **Primary Key** − A primary key is one of the candidate keys chosen by the database designer to uniquely identify the entity set.

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. These attributes are called **descriptive attributes**.

Degree of Relationship

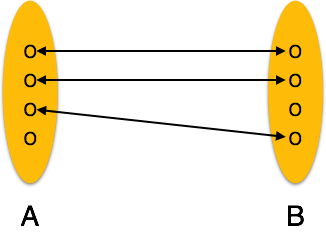
The number of participating entities in a relationship defines the degree of the relationship.

* Binary = degree 2
* Ternary = degree 3
* n-ary = degree

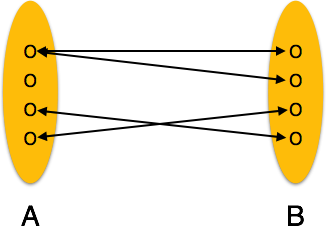
Mapping Cardinalities

**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.

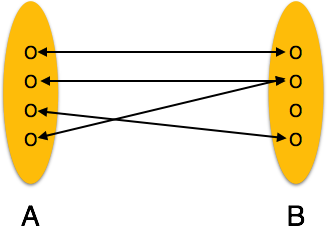
* **One-to-one** − One entity from entity set A can be associated with at most one entity of entity set B and vice versa.



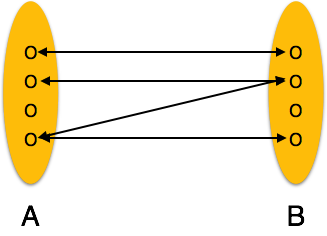
* **One-to-many** − One entity from entity set A can be associated with more than one entities of entity set B however an entity from entity set B, can be associated with at most one entity.



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



[Previous Page](http://www.tutorialspoint.com/dbms/dbms_data_independence.htm)

[Print](http://www.tutorialspoint.com/cgi-bin/printpage.cgi)

[PDF](http://www.tutorialspoint.com/dbms/pdf/er_model_basic_concepts.pdf)

[Next Page](http://www.tutorialspoint.com/dbms/er_diagram_representation.htm)

Advertisements

[[img](javascript:void(0)) [img](javascript:void(0)) [img](javascript:void(0)) [img](javascript:void(0)) [img](javascript:void(0)) img](javascript:void(0))

[](http://www.tutorialspoint.com/index.htm)

* [Write for us](http://www.tutorialspoint.com/about/tutorials_writing.htm)

* [FAQ's](http://www.tutorialspoint.com/about/faq.htm)

* [Helping](http://www.tutorialspoint.com/about/about_helping.htm)
* [Contact](http://www.tutorialspoint.com/about/contact_us.htm)

© Copyright 2015. All Rights Reserved.

 go