1--RDBMS

***Relational Database Management System***

A relational database management system (RDBMS) is a collection of programs and capabilities that enable IT teams and others to create, update, administer and otherwise interact with a [relational database](https://searchdatamanagement.techtarget.com/definition/relational-database). RDBMSes store data in the form of tables, with most commercial relational database management systems using [Structured Query Language](https://searchsqlserver.techtarget.com/definition/SQL) (SQL) to access the database

A database management system (DBMS) is a software interface that sits between a database and a user, or between a database and an application (such as an inventory system).  
Different DBMSs support different database models and different

1. DDLs [Data Definition Language] and
2. DMLs [Data Manipulation Language].

A typical DBMS creates, manipulates, modifies, and maintains a database.

A relational database management system (RDBMS) supports the relational model. It performs the same basic functions as those listed for a DBMS. SQL is the data definition and data manipulation language that is packaged with an RDBMS. The following SlideShow walks you through the major functions of a RDBMS. Use the forward and back arrows to navigate through the series of images.

[**RDBMS Functions**](https://www.relationaldbdesign.com/database-design/module2/rdbms-functions.php)

**RDBMS Management**

One RDBMS can manage many relational databases.  
Oracle, Microsoft, IBM, MySQL, and Sybase have comprehensive RDBMSs available on the market.  
Each company packages its product(s) with utilities to enhance RDBMS capabilities and performance. Product prices vary widely among these top competitors.  
The next lesson discusses the concept of database change management.

[Relational Database Design Implementation](https://amzn.to/3m8sO8i)

**Main functions of DBMS**

A database management system (DBMS) is a collection of programs that manages the database structure and controls access to the data stored in the database. In a sense, a database resembles a very well-organized electronic filing cabinet in which powerful software, known as a database management system, helps manage the cabinet’scontents.

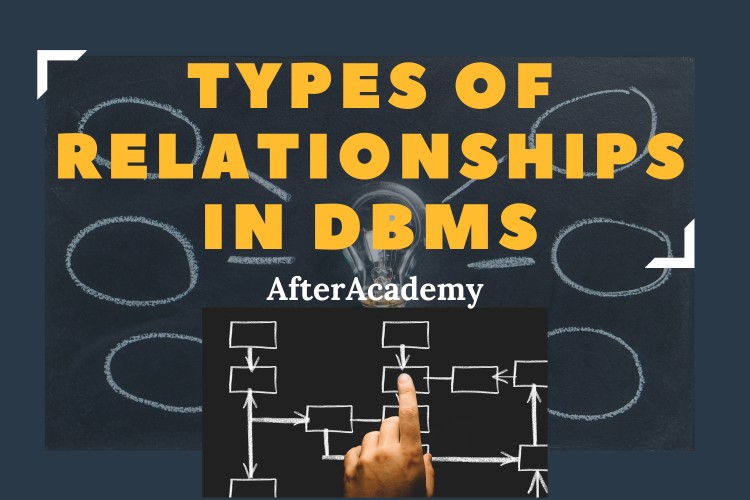
Types of relationships in a database

Four types of relationships exist in relational database design:

* one to one - where one table record relates to another record in another table
* one to many - where one table record relates to multiple records in another table
* many to one - where more than one table record relates to another table record
* many to many - where multiple records relate to more than one record in another table

These relations form functional dependencies within the database. Some common **examples of relational databases** include MySQL, Microsoft SQL Server, Oracle, etc.

2--What are the different types of relationships in DBMS?



We are related to things around us in some or the other way. We are related to our family, friends, etc and these relationships are of different kinds. ***For example***, you and your father are related. There is a parent-child relationship between you. You have only one father but he may have many children. You and your siblings are related. You and your teacher are related. The teacher teaches you as well as many other students. You also study from different teachers.

Database relationships are also very similar to such relationships. We are done with the ER model and we also know about the Entity. Now, in this blog, we will see the database relationships i.e. the second part of the ER model in details.

#### Relationship in DBMS

Any association between two entity types is called a relationship. Entities take part in the relationship. It is represented by a diamond shape.

***For example,*** A teacher teaches students. Here, "***teaches***" is a relationship and this is the relationship between a Teacher entity and a Student entity.

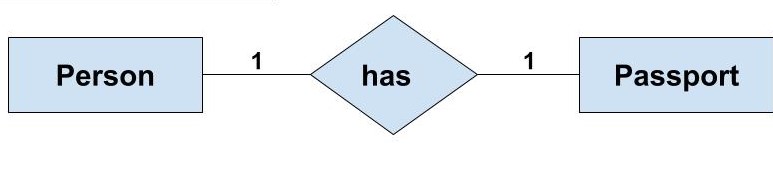
We have two entity types of 'Customer'(Customer\_id, Name, City, Phone) and 'Account'(Account\_no, Type, Balance). We store the data of 'Customer' in one table and his accounts details in the 'Account' table. Now, to link these two tables we need to insert the primary key 'Customer\_id' of the 'Customer' table in the 'Account' table. This key acts as a foreign key for the 'Account' table and refers to a column with the same name in the 'Customer' table. This is how a relationship between two tables is established. There are three types of relationships that can exist between two entities.

* One-to-One Relationship
* One-to-Many or Many-to-One Relationship
* Many-to-Many Relationship

#### One-to-One Relationship

Such a relationship exists when each record of one table is related to only one record of the other table.

***For example,*** If there are two entities ‘Person’ (Id, Name, Age, Address)and ‘Passport’(Passport\_id, Passport\_no). So, each person can have only one passport and each passport belongs to only one person.

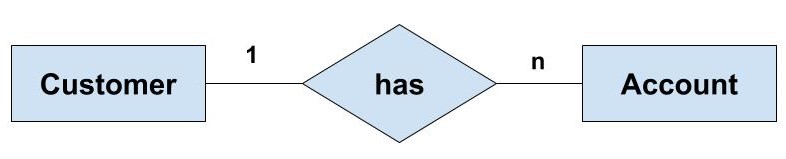


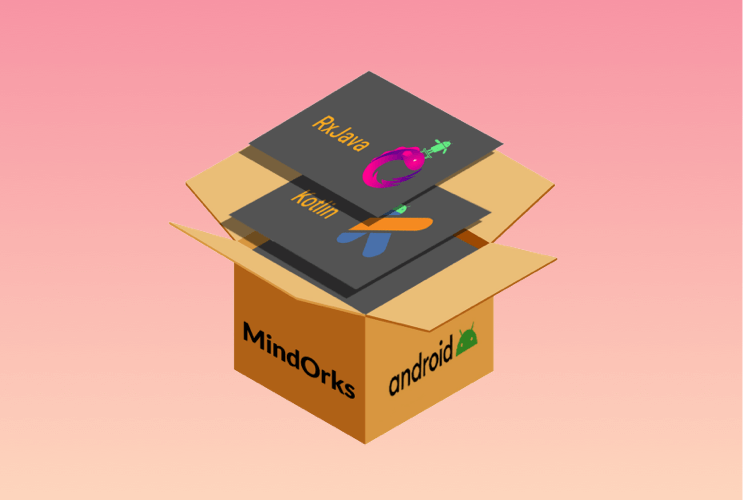
Such a relationship is not very common. However, such a relationship is used for security purposes. In the above example, we can easily store the passport id in the ‘Person’ table only. But, we make another table for the ‘Passport’ because Passport number may be sensitive data and it should be hidden from certain users. So, by making a separate table we provide extra security that only certain database users can see it.

#### One-to-Many or Many-to-One Relationship

Such a relationship exists when each record of one table can be related to one or more than one record of the other table. This relationship is the most common relationship found. A one-to-many relationship can also be said as a many-to-one relationship depending upon the way we view it.

***For example,***If there are two entity type ‘Customer’ and ‘Account’ then each ‘Customer’ can have more than one ‘Account’ but each ‘Account’ is held by only one ‘Customer’. In this example, we can say that each Customer is associated with many Account. So, it is a one-to-many relationship. But, if we see it the other way i.e many Account is associated with one Customer then we can say that it is a many-to-one relationship.





NEW

##### **Android App Development Online Course by MindOrks**

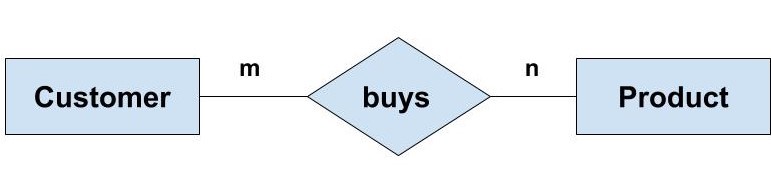
Start your career in Android Development. Learn by doing real projects.

[**CHECK NOW**](https://mindorks.com/android-app-development-online-course)

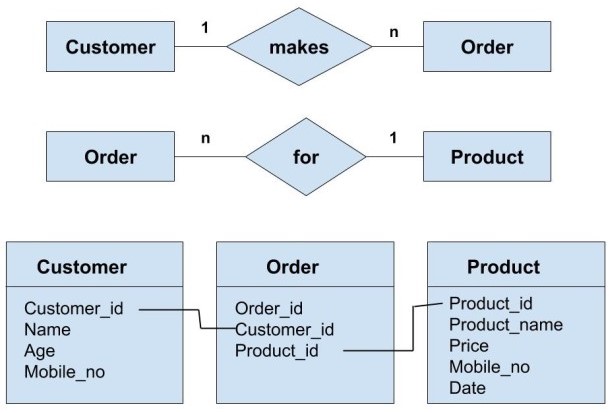
#### Many-to-Many Relationship

Such a relationship exists when each record of the first table can be related to one or more than one record of the second table and a single record of the second table can be related to one or more than one record of the first table. A many-to-many relationship can be seen as a two one-to-many relationship which is linked by a 'linking table' or 'associate table'. The linking table links two tables by having fields which are the primary key of the other two tables. We can understand this with the following example.

**Example:**If there are two entity type ‘Customer’ and ‘Product’ then each customer can buy more than one product and a product can be bought by many different customers.



Now, to understand the concept of the linking table here, we can have the ‘Order’ entity as a linking table which links the ‘Customer’ and ‘Product’ entity. We can break this many-to-many relationship in two one-to-many relationships. First, each ‘Customer’ can have many ‘Order’ whereas each ‘Order’ is related to only one ‘Customer’. Second, each ‘Order’ is related only one Product wheres there can many orders for the same Product.



In the above concept of linking can be understood with the help of taking into consideration all the attributes of the entities 'Customer', 'Order' and 'Product'. We can see that the primary key of both 'Customer' and 'Product' entity are included in the linking table i.e 'Order' table. These key act as foreign keys while referring to the respective table from the 'Order' table.