

Relational model

You are already familiar with table relationships, one of the main concepts behind the relational database model. In this reading, you'll find out more about the relational database model. Though it was introduced a long time ago, it is still the most widely used data model for commercial databases.

What is the relational model?

The relational model is built around three main concepts which are:

Data,

Relationships,

and constraints.

It describes a database as “a collection of inter-related relations (or tables)”. It is still a dominant model used for data storage and retrieval. In essence, it is a way of organizing or storing data in a database. SQL is the language that's used to retrieve data from a relational database.

Fundamental concepts of the relational model

Relation

A relation represents a file that stores data. It's also known as a table. Within a table there are rows and columns. Each row represents a group of related data values. A row, or record, is also known as a tuple. Columns in a table are also known as fields or attributes. These columns define or describe a row. Therefore, a record or a row consists of a set of attributes.

	Column 1	Column 2	Column 3
Row 1			
Row 1			
Row 1			

Column

A table stores pieces of data or facts as columns. In other words, the principal storage unit of a database is a column (attribute). When determining the columns for your table, think about the pieces of data that need to be stored within that table. Each column is a generic representation of the piece of data that needs to be stored. Each table cell that becomes a part of a row will have a specific instance of a piece of data.

ID	First Name	Last Name
1	John	Smith
2	Mish	Azerrad
3	Peter	Klien

For example, ID, First Name and Last Name are columns. The ID numbers, first names and last names are instances or pieces of data that are stored in those columns. Also, every column has a specific data type, which could be numeric, text, or date.

Domain

The domain is a set of acceptable values that a column is allowed to contain. The domain depends on the data type of the column. Namely whether it is numeric, or text based. The domain of ID has a set of acceptable and possible values that are numeric such as 1, 2 and 3. The domain of First Name has a set of acceptable and possible values that are text based, which is people's first names. In the ID column, it's not possible to store values such as “John” or “001”. Similarly, the First Name column can't accept any numeric pieces of data.

Record or tuple

A record, also known as a tuple, is a row within a table. If a table has columns for ID, First Name and Last Name, then one record or tuple would have one person's ID, first name and last name. Another record would have another person's full personal information.

Key

Each row or tuple has one or more attributes, known as a relation key, that can uniquely identify a specific row. This is also known as the primary key.

Degree

Degree is the number of columns or attributes within a relation. A student table that stores the student's name, address, phone number and email address would have a degree of four.

Cardinality

Cardinality refers to how many records there are within a particular table in a database. If you have 100 students in your student table, with all their information organized into individual rows, then that table has a cardinality of 100.

What are constraints?

In the relational model, every relation needs to meet three conditions. These three conditions must be met for a relation to be valid. They are called relational integrity constraints and they are:

- . Key constraints
- . Domain constraints
- . Referential integrity constraints

Key constraints

The key constraint revolves around the key attribute(s). In the relational model, a key attribute is an identifier that can be used to refer to a record. It must also be unique for each record. For example, you can use the Student ID in the student table as the key. This means that there can't be two students with the same Student ID. If so, it would be invalid and cause an issue when it comes to accessing or retrieving the data. Also, a key attribute cannot have NULL values. This is the requirement that should be met by the Key constraint.

Domain constraints

Domain constraints are all about the requirement of inserting values that have a valid data type. There are a variety of data types that can be included within a table, namely numeric, text and data, in the case of the example. If an attempt is made to store an incorrect data typed value to an attribute, it's declared a violation of domain constraints. For instance, if an attribute requires a numeric value to be entered, and the value you are attempting to enter uses letters instead of numbers, then it would be invalid.

Referential integrity constraints

A database has multiple tables that refer to one another. Referential integrity constraints are based on the concept of foreign keys. A foreign key is a key attribute present in a table, which is also a primary key of another table to which it needs to be linked. Through this key, it references the other table to which it's related. For example, the order ID is present in the Order_Item table as a foreign key, which is also the primary key of the order table. So, the order table and the Order_Item table are related to each other because the Order_Item table references the order table via the order ID attribute. The referential integrity constraint states that if a relation refers to a key attribute of another relation, then that key element must exist. In other words, there must be matching values in the two tables for that attribute.

Types of relationships

In the relational model, there are three types of relationships that can exist between tables.

- . One-to-one
- . One-to-many
- . Many-to-many

One-to-one

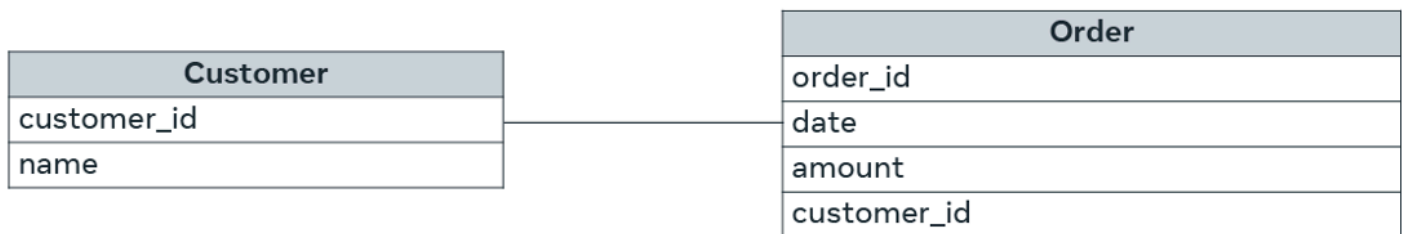
In order to understand one-to-one relationships, let's take the example of two tables: Table A and Table B. A one-to-one (1:1) relationship means that each record in Table A relates to one, and only one, record in Table B. Likewise, each record in Table B relates to one, and only one, record in Table A. Here is a diagram that illustrates the example:



Here, every country has one, and only one, capital. And every capital belongs to one and only one country.

One-to-many

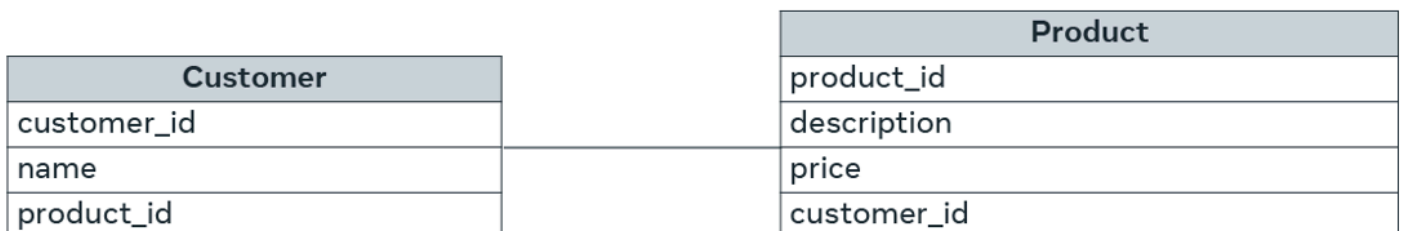
If there are two tables, Table A and Table B, a one-to-many (1:N) relationship means a record in Table A can relate to zero, one, or many records in Table B. Many records in Table B can relate to one record in Table A. Let's examine the following relationship between customers and orders.



Here, each customer can place many orders. Many records in the order table can relate to only one record in the customer table.

Many-to-many

If there are two tables, Table A and Table B, a many-to-many (N:N) relationship means many records in Table A can relate to many records in Table B. And many records in Table B can relate to many records in Table A. Let's examine this example of many-to-many relationship between customer and product with the use of a diagram. In the diagram, customers can purchase various products, and products can be purchased by many customers.



Usually, many-to-many relationships are not kept in a data model. They are broken down into two one-to-many relationships by introducing a junction or middle table.

To conclude, there are many benefits to the relational database model. This includes the ability to design and develop a meaningful system of information, and the ability to access and retrieve every single piece of data stored in the database.

Mark as completed



Like



Dislike

Report an
issue