Database Relationships

Relationship:

A relationship is a link that relates two entities that share one or more attributes.

Introduction

When creating a database, common sense dictates that we use separate tables for different types of entities. Some examples are: customers, orders, items, messages etc… But we also need to have relationships between these tables. For instance, customers make orders, and orders contain items. These relationships need to be represented in the database. Also, when fetching data with SQL, we need to use certain types of JOIN queries to get what we need.

There are several types of database relationships. Today we are going to cover the following:

- One to One Relationships
- One to Many and Many to One Relationships
- Many to Many Relationships
- Self Referencing Relationships

When selecting data from multiple tables with relationships, we will be using the JOIN query. There are several types of joins and these are as given bellows:

- Cross Joins
- Natural Joins
- Inner Joins
- Left (Outer) Joins
- Right (Outer) Joins

Relational database design supports many different types of relationships between tables. These relationships are in place to prevent the entry of inconsistent data and enforce referential integrity.

One-to-many relationships:

The most common relationship used when creating relational databases. A row in a table in a database can be associated with one or (likely) more rows in another table. An example of a one-to-many relationship is a single order has many items on that order. And since relationships work both ways it is not uncommon to hear reference to many-to-one-relationships as well.

One-to-one relationship:

A row in a table is associated to one and only one row in another table. An example of a one-to-one relationship is a person can have one social security number and a social security number can only be assigned to one person.

In most cases there is no need for a one-to-one relationship as the contents of the two tables can be combined into one table.

Many-to-many relationships:

When one or more rows in a table are associated with one or more rows in another table. An example of a many-to-many relationship is a table of customers who can purchase many different products and a table of products that can be purchased by many different customers.

In relational database design, a many-to-many relationship is not allowed.

Types of Relationships

One-to-many

The one-to-many relationship is the workhorse of relational databases as well as being the easiest relationship to understand. Let's say you need to build a shopping cart application for an e-commerce site. Your first draft of the database has columns for Item1, Item2, and Item3 with the corresponding Quantity1, Quantity2, and Quantity3 fields.

One-to-one

One-to-one table relationships are a little more interesting and more underused than either of the other two types of relationships. The key indicator of a possible need for a one-to-one relationship is a table that contains fields that are only used for a certain subset of the records in that table.

Many-to-many

Finally, there is the many-to-many table. This relationship is a little more complex than the one-to-many because, in addition to the two tables of data, we need another table to join the two tables of interest together. That's right, we're adding a table to the database -- but it is a simple table and saves us lots of effort down the road. As an example, let's say you want to add the ability to search for CDs by the musicians on any given song. From the musician side, you have one musician related to many songs.

Relationships Types

A relationship works by matching data in key columns — usually columns with the same name in both tables. In most cases, the relationship matches the primary key from one table, which provides a unique identifier for each row, with an entry in the foreign key in the other table.

There are three types of relationships between tables. The type of relationship that is created depends on how the related columns are defined.

- One-to-Many Relationships
- Many-to-Many Relationships
- One-to-One Relationships

One-to-Many Relationships

A one-to-many relationship is the most common type of relationship. In this type of relationship, a row in table A can have many matching rows in table B, but a row in table B can have only one matching row in table A. For example, the publishers and titles tables have a one-to-many relationship: each publisher produces many titles, but each title comes from only one publisher.

Make a one-to-many relationship if only one of the related columns is a primary key or has a unique constraint.

The primary key side of a one-to-many relationship is denoted by a key symbol. The foreign key side of a relationship is denoted by an infinity symbol.

Many-to-Many Relationships

In a many-to-many relationship, a row in table A can have many matching rows in table B, and vice versa. You create such a relationship by defining a third table, called a junction table, whose primary key consists of the foreign keys from both table A and table B. For example, the authors table and the titles table have a many-to-many relationship that is defined by a one-to-many relationship from each of these tables to the title authors table. The primary key of the title authors table is the combination of the au_id column (the authors table's primary key) and the title_id column (the titles table's primary key).

One-to-One Relationships

In a one-to-one relationship, a row in table A can have no more than one matching row in table B, and vice versa. A one-to-one relationship is created if both of the related columns are primary keys or have unique constraints.

This type of relationship is not common because most information related in this way would be all in one table. You might use a one-to-one relationship to:

- Divide a table with many columns.
- Isolate part of a table for security reasons.
- Store data that is short-lived and could be easily deleted by simply deleting the table.
- Store information that applies only to a subset of the main table.

The primary key side of a one-to-one relationship is denoted by a key symbol. The foreign key side is also denoted by a key symbol.

One to One Relationships

Now we have a relationship between the Customers table and the Addresses table. If each address can belong to only one customer, this relationship is "One to One". Keep in mind that this kind of relationship is not very common. Our initial table that included the address along with the customer could have worked fine in most cases.

One to Many and Many to One Relationships

This is the most commonly used type of relationship. Consider an e-commerce website, with the following:

- Customers can make many orders.
- Orders can contain many items.
- Items can have descriptions in many languages.

Many to Many Relationships

In some cases, you may need multiple instances on both sides of the relationship. For example, each order can contain multiple items. And each item can also be in multiple orders. For these relationships, we need to create an extra table.

Database Relationships

Relationships allow you to describe the connections between different database tables in powerful ways. Once you've described the relationships between your tables, you can later leverage that information to perform powerful cross-table

Types of Database Relationships

There are three different types of database relationships, each named according to the number of table rows that may be involved in the relationship. Each of these three relationship types exists between two tables.

One-to-one relationships occur when each entry in the first table has one, and only one, counterpart in the second table. One-to-one relationships are rarely used because it is often more efficient to simply put all of the information in a single table.

One-to-many relationships are the most common type of database relationship. They occur when each record in the first table corresponds to one or more records in the second table but each record in the second table corresponds to only one record in the first table. For example, the relationship between a Teachers table and a Students table in an elementary school database would likely be a one-to-many relationship, because each student has only one teacher, but each teacher may have multiple students.

Many-to-many relationships occur when each record in the first table corresponds to one or more records in the second table and each record in the second table corresponds to one or more records in the first table. For example, the relationship between a Teachers and a Courses table would likely be many-to-many because each teacher may instruct more than one course and each course may have more than one instructor.