What is a Relationship in SQL

In the previous chapters, we have performed operations and discussions on a single table. But in most real-world situations, we will need more than one table, and there are many tasks that need to be completed by combining different tables.

So here comes the question: how do we define the relationship between two different tables?

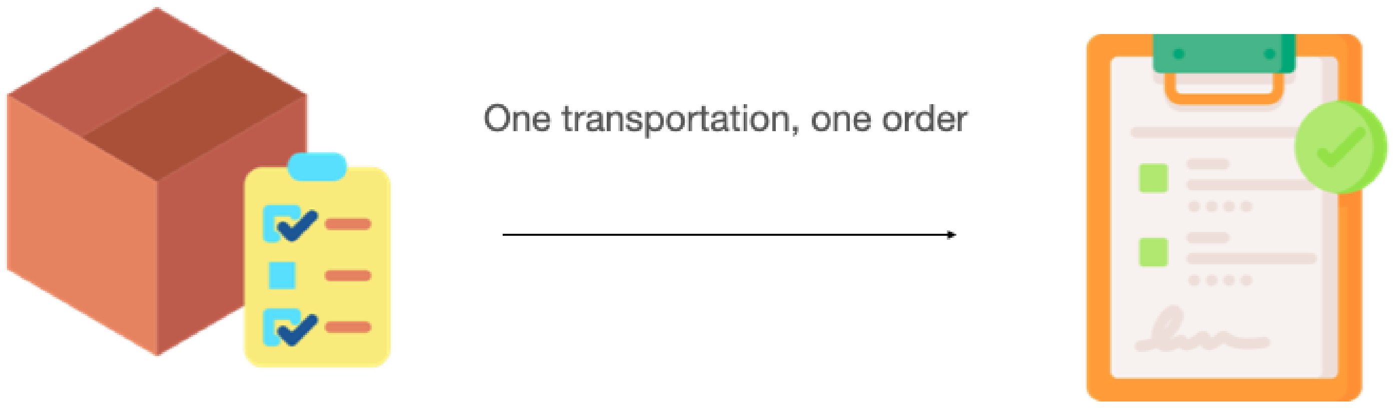
The answer is that we should depend on the technology of relationships in SQL. And like the MySQL used in our series of demonstrations, it is classified as Relational Database Management System, the most important thing is to have the design of Relationship, which is also one of the standard equipment of most mature commercial databases on the market.

In SQL, we don't need to use a new statement, the key point is to establish the correct field correspondence, such as the id column of the users table corresponds to the user_id column of the orders table

Now, let's take a look at how to design the correct relationship.

One to One

Imagine a situation where our database has two tables: `transportations` and `orders`. Each transportation is only responsible for one order, and each order is only be assigned to one transportation. Then, they have One to One relationship.



In the database, of course, there is no relationship without doing anything. We have to define the following two tables:

transportations

id	order_id	address
1	1	U.S
2	2	India

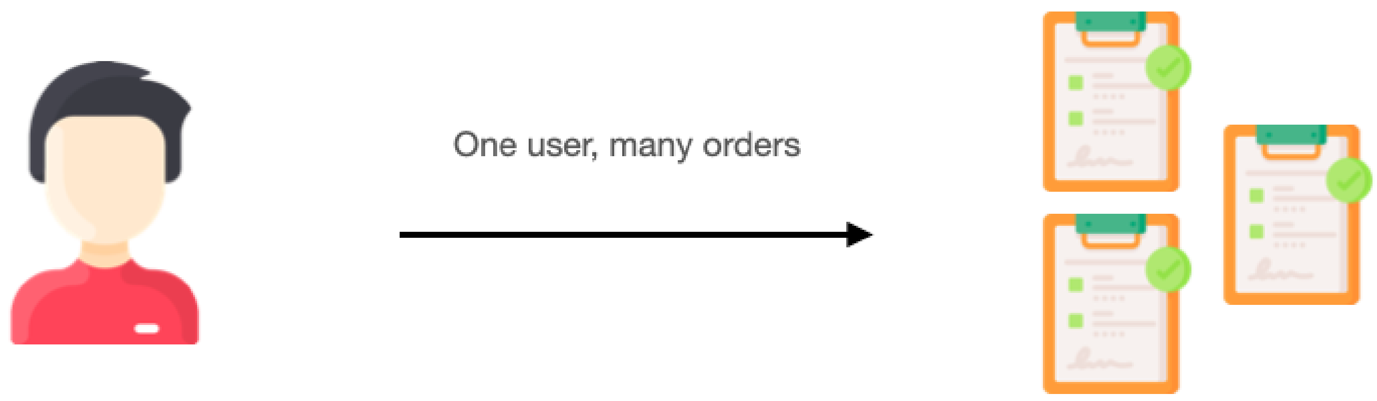
orders

id	note
1	some information
2	some comments

Please focus on `transportations`, we add a column `order_id`, and use this column to record which order corresponds to which transportation. In the one to one relationship, you can also put the responsibility for recording id to another table, which means to add a new column `transportation_id` to `orders`.

One to Many

The One to Many relationship is very common in the design of actual database table architecture. It can be imagined as the relationship between `users` and `orders`. Usually, a user can have many orders, but an order can only belong to one user.



In this case, we can design the following tables:

users

id	name	age
1	John	40
2	May	30

orders

id	user_id	note
1	1	some information
2	2	some comments
3	2	no comments

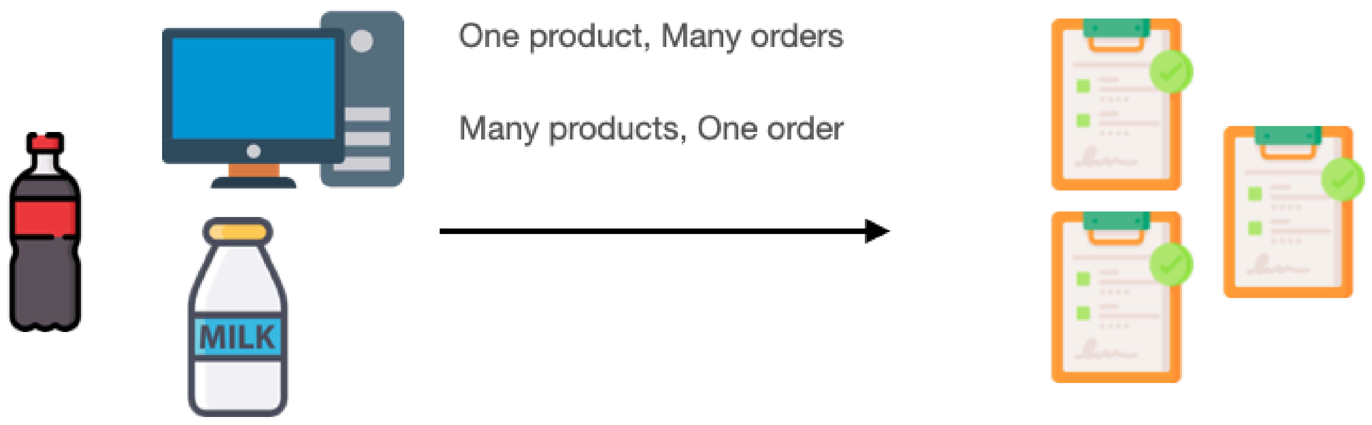
Let's focus on the `user_id` column in `orders` here. We set the id of another table in the table which belongs to the other table, and it is different from one to one, so the settings cannot be exchanged here.

The reason is that if we set `order_id` in `users`, then each user will be limited to only have one order, which does not meet the data relationship we want!

Many to Many

The Many to Many relationship is a relatively complex relationship. Usually, two tables with this relationship require a third new table be created just to record the relationship between the two original tables.

For example, the relationship between `orders` and `products`, an order can have many products, and a product can also belong to many orders.



Rendered table associations such as:

orders

id	user_id	note
1	1	some information
2	2	some comments
3	2	no comments

products

id	name
1	table
2	chair
3	apple

orders_products

id	order_id	product_id

1	1	1
2	1	2
3	2	2
4	3	1
5	3	3

Let's focus on the `orders_products` table, this table is specially used to record the relationship between `orders` and `products`, so only the ids of `orders` and `products` can be seen. Through this table, we can understand The following associations: 1. order 1 has two products, table and chair. 2. order 2 has one product, chair. 3. order 3 has two products, table and apple.

And that's all there is to it!

Meaningful Transformation

Finally, readers who are familiar with the business field may notice that this `orders_products` table is very similar to `order_items`. Can we use this table to record order details?

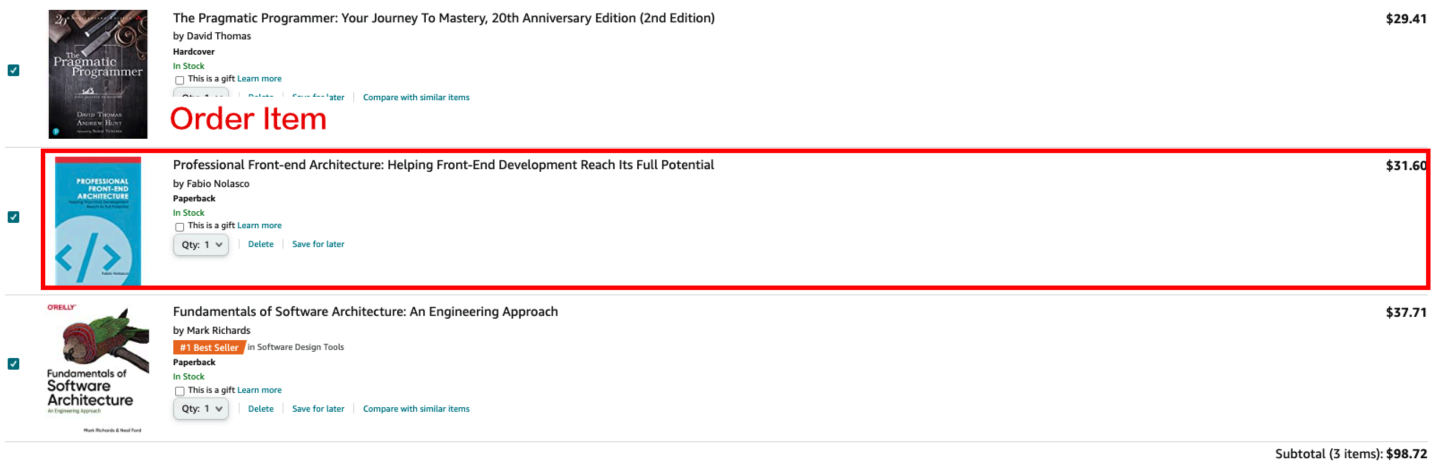
Of course! And `orders_products` can be reasonably expanded to:

order_items

id	order_id	product_id	quantity
1	1	1	1
2	1	2	3
3	2	2	5
4	3	1	3

5	3	3	8

Here, each row represents an ordered item. As a real-world example, here, a user purchased 3 items in their order which has a specific order id, and each item will have its own product id and the quantity purchased in this order.



The Pragmatic Programmer: Your Journey To Mastery, 20th Anniversary Edition (2nd Edition) $29.41
by David Thomas
Hardcover
In Stock
This is a gift Learn more
Compare with similar items

Professional Front-end Architecture: Helping Front-End Development Reach Its Full Potential $31.60
by Fabio Nolasco
Paperback
In Stock
This is a gift Learn more
Qty: 1 | Delete | Save for later

Fundamentals of Software Architecture: An Engineering Approach $37.71
by Mark Richards
#1 Best Seller in Software Design Tools
Paperback
In Stock
This is a gift Learn more
Qty: 1 | Delete | Save for later | Compare with similar items

Subtotal (3 items): $98.72

It becomes a table with practical use, and it can record the relationship between the other two tables at the same time. We call this an intermediate table of the other two tables.

This is a very common database design logic that is worth remembering, as it will be helpful for database system design in the future!