

Topic is Relationship?

Clearly, the power of relational databases lies in relating tables to each other. Django offers ways to define the three most common types of database relationships: many-to-one, many-to-many and one-to-one.

Django type of Relationship?

1. Many-to-One
2. Many-to-Many
3. One-to-One

Many-to-one relationships → For example, if a user model has a post – that is, a post makes multiple user but each user only has one post – use the following definitions:

Concept → One User creates to multiple posts that set.

Many Too Many Relationships?

To define a many-to-many relationship, use `ManyToManyField`. You use it just like any other Field type: by including it as a class attribute of your model.

For example, if a Pizza has multiple Topping objects – that is, a Topping can be on multiple pizzas and each Pizza has multiple toppings – here's how you'd represent that:

Extra fields on many-to-many relationships →

When you're only dealing with many-to-many relationships such as mixing and matching pizzas and toppings, a standard `ManyToManyField` is all you need. However, sometimes you may need to associate data with the relationship between two models.

