Table	Group
broadcasts	Broadcast
sequences	Sequence
sequence_mails	Sequence
sequence_mail_schedules	Sequence
sent_mail	Broadcast or Sequence?

This is one way of doing it. We create a group for Broadcast and another one for Sequence. This approach has several benefits:

- Domains remain small.
- We don't mix up different concepts, for example, broadcasts and sequences.
- Sequences are complicated, so I feel it's a good thing to have a separate domain.

And several disadvantages:

- Maybe some domains will be too small? Such as the Broadcast. Right now, we only have one model in it.
- What about `sent_mails`? Both sequences and broadcasts are using it, so we cannot put it in one group. We can solve this using a Shared domain. It's kind of like a "joker" domain where you can put classes used by multiple domains. It's not the best solution, but it definitely works. And believe me, every project has a Shared or Common domain. Literally, every larger application has one of these.
- What about other shared classes between broadcasts and sequences? We are in the design phase, so we only know about one shared class. But I'm 100% absolutely sure that we need other classes. These all will be placed inside the Shared domain and will be used by sequences and broadcasts. And remember, right now, we only have tables and models. What about DTOs, value objects, actions, and so on? For example, both broadcasts and sequences have filters that require a DTO or a value object (or maybe both?). This will also live in the Shared domain.

I think this solution is not that bad, but broadcasts and sequences can be grouped in another way: