```sql
select count(*)
from sent_mails
where sent_at between "2022-01-01 00:00:00" and "2022-12-31 23:59:59";
```

So you've created indexes and written good queries, but your reports are still slow. What else can you do?

- You can always look for N+1 query problems. It's a very common problem when you're working with ORM.
- You can optimize select * queries. Most of the time, it's unnecessary to query every column.

Your next step can be database partitioning. I'm talking about horizontal partitioning. It's a way to tell MySQL how to store your data physically. You can create multiple portions (partitions) from a table, and the engine will store these portions in different files. For example, you can define rules such as:

- `sent_mails` between 2022-01 and 2022-06 should be a partition.
- `sent_mails` between 2022-07 and 2022-12 should be another partition.

This way, you can write queries that only touch a small number of partitions (fewer rows to fetch).

If you're still having problems, you can always use Redis. These historical reports are static by nature. If you want, you can replicate the whole `sent_mails` in Redis and serve the requests from there. There are a dozen different ways to do that.