

# Object Relational Mysteries (ORM)

Author: Chris Conlon

Claps: 66

Date: Mar 5, 2018

Every year as Klaviyo approaches the holiday ramp up, it's been a common pattern to audit our codebase for inefficient and redundant queries, trying to draw blood from a database stone. I'd make a wager that doing an audit like this in any codebase is a chore, and not one that engineers get excited to spend hours on. That was the case at Klaviyo, so this year we decided to build some tooling to make it **much** less painful, and it turned out to be really helpful. If you're just here for the tool that came out of this holiday season's ramp up, you're looking for the [django-orm-profiler](#) which we've open sourced. Otherwise, read on!

We run a lot of our queries through the Django ORM at Klaviyo, and it can get hairy sometimes at scale. Even just changing a single line to stop loading an unused related object (which would invoke its foreign key and then in turn run an unnecessary query) can be a performance boon when you're running that line of code (and by extension the redundant query) thousands of times a second in an event processing pipeline, for example.

```
# runs a query
sandwich_id = sandwich_theory_model.sandwich.id
```

vs

```
# doesn't run a query
sandwich_id = sandwich_theory_model.sandwich_id
```

It's just not easy to catch these implicit query landmines that ORMs hide down under their shiny interfaces. Until you've become too familiar with the quirks and gotchas of your ORM and you can just spot them inline you need to audit your codebase to see what's really happening.