N+1 Query Problem

Topic Cheat Sheet

DATABASE STAR

What is it?

One query fetches a list, then a second query runs for each row. N rows = N+1 total queries.

Why It's Bad

- Scales linearly with data. 10,000 rows = 10,001 queries
- Hides in dev (small dataset), explodes in production
- Each query is fast, so slow query logs won't catch it
- Symptoms appear gradually as data grows, not from a code change

How It Happens

It can happen with either:

- A query inside a loop (the manual version)
- ORM lazy loading: order.customer.name looks innocent, but queries the database on every iteration

Example: inside a loop

```
orders = db.query("SELECT id, customer_id, total FROM orders")

for order in orders:
    customer = db.query(
        "SELECT name FROM customers WHERE id = ?",
        order.customer_id
    )
    print(f"{customer.name}: {order.total}")
```

How to Fix It

- JOIN: fetch both tables in one query (best default fix)
- Eager loading: tell your ORM to load related records upfront (2 queries total, not N+1)
- Batching: collect all IDs first, then WHERE id IN (...) in one query (useful when JOINs aren't practical, e.g. cross-service data)

How to Detect It

- Enable query logging: look for the same query structure repeating with only the ID changing
- ORM debug/query counter mode: count queries per page load
- Search code for DB calls inside loops
- APM tools in production: flag total query count per request, not just slow individual queries