

1. Why a massive **JOIN** query is an anti-pattern for a high-traffic system:

In a massive JOIN is bad because in this approach, the database ends up doing way too much work at once. Each extra table you join makes the query more expensive, it locks CPU, I/O operations, and slows down the full system in a large dataset.

2. **Nested loop join:**

In a nested loop join, our database first takes data from one table and then tries to match it with other tables. This is ok for small projects, but when it comes to large-scale applications, it has huge drawbacks, especially in large datasets. Let's have an example: We have 1000000 drivers and 100000 trips now. If we just want to find drivers with their trips, it will be $1000000 * 100000 = 100000000000$ (100 billion comparisons), even in the indexed table, it also needs a huge calculation.