

什么是n+1查询？

比如说，一辆汽车有4个轮子.现在要查询品牌为丰田的汽车

```
public class Car{
    int id;
    string brand;// 品牌
    List<Wheel> wheelList;// 轮子
}

List<Car> list = listCar('丰田');
for(Car car : list){
    List<Wheel> wheelList = listWheel(car);
    car.setWheelList(wheelList);
}
```

相对应的sql操作就是

```
-- N+1 中的1
select * from tb_car where brand = '丰田'
```

然后遍历返回的结果，假设数量为N

```
-- N+1 中的N
select * from tb_car_wheel where car_id = ?
```

当N值很大时，性能会很低。即使查询tb_car_wheel的单条sql只要5ms，但是当N=1000时，耗时就长达5s。

如何避免N+1？

1. 批量查询

```
select * from tb_car_wheel where car_id in (?, ?, ?, ?, ?...)
```

然后在内存中组装汽车轮子

2. 关联表

```
select t.*,t2.* from tb_car t
left join tb_car_wheel t2
on t.id = t2.car_id
where brand = '丰田';
```

orm框架会自动将查询结果组装汽车轮子。

注意 通过配置单独查询sql的方式使用mybatis的association元素，会引起N+1问题。

```
<!-- 不要这么用 -->
<resultMap id = "" type = "">
    <association property="wheel" ,select="selectWheel"
</>
```