

# 可重复读简介



group by 一定要使用到SELECT中未被聚合函数包括的属性列吗？



在SQL查询中，使用 `GROUP BY` 子句时，**所有在 `SELECT` 语句中出现但未被聚合函数（如 `COUNT()`、`SUM()`、`AVG()` 等）包裹的列**，必须出现在 `GROUP BY` 子句中。这是为了确保查询结果的逻辑一致性和可理解性。

## 为什么必须如此？

`GROUP BY` 的作用是将结果集按某些列进行分组，然后对每个分组进行聚合操作。聚合函数用于计算每个分组中的汇总信息，如总和、平均值等。但是，如果 `SELECT` 中有未被聚合函数包裹的列，而这些列没有出现在 `GROUP BY` 中，就会出现混淆：数据库无法确定应该为哪个分组的行返回该列的值。

例如，假设我们有一个名为 `employees` 的表，包含员工的 `department\_id` 和 `salary` 信息，我们想要按部门汇总工资总额：

sql

```
SELECT department_id, SUM(salary)
FROM employees
GROUP BY department_id;
```

这里 `department\_id` 是 `GROUP BY` 中的分组列，`SUM(salary)` 是聚合函数的计算结果。因为 `department\_id` 在 `SELECT` 中没有被聚合函数包裹，所以它必须出现在 `GROUP BY` 中，确保我们在每个部门上计算工资总和。

## 错误示例

如果我们尝试查询如下 SQL：

sql

```
SELECT department_id, employee_name, SUM(salary)
FROM employees
GROUP BY department_id;
```

这会导致错误，因为 `employee\_name` 既没有被聚合函数包裹，也没有出现在 `GROUP BY` 子句中。SQL无法知道应该返回哪个员工的名字（`employee\_name`），因为在一个部门内可能有多个员工。如果你只按 `department\_id` 分组，无法明确从该分组的多个员工中选出一个具体的 `employee\_name`。

## 正确的做法

要解决这个问题，有两种方法：

1. 将 `employee\_name` 包含在 `GROUP BY` 中：

sql

```
SELECT department_id, employee_name, SUM(salary)
FROM employees
GROUP BY department_id, employee_name;
```

这样，每个部门内的每个员工都会作为单独的一组进行聚合。

2. 对不参与分组的列使用聚合函数（如 `MAX()` 或 `MIN()`）：

sql

```
SELECT department_id, MAX(employee_name), SUM(salary)
FROM employees
GROUP BY department_id;
```

这种情况下，`MAX(employee\_name)` 会返回每个部门中的名字排序最靠后的员工（根据数据库排序规则），但由于 `employee\_name` 被聚合函数包裹，查询是合法的。

## 总结

在 `GROUP BY` 查询中，所有出现在 `SELECT` 中但没有被聚合函数包裹的列，都必须出现在 `GROUP BY` 子句中，以确保分组和聚合的逻辑一致性。如果这些列没有在 `GROUP BY` 中定义，SQL将无法处理该查询，通常会抛出语法错误。