

掌握 SQL 子查询：让你成为查询优化高手

原创 张慧源 源话编程 2025年01月20日 09:49 吉林

👋 热爱编程的小伙伴们，欢迎来到我的编程技术分享公众号！在这里，我会分享编程技巧、实战经验、技术干货，还有各种有趣的编程话题！



源话编程

星光不问赶路人，时光不负有心人。

166篇原创内容

公众号

1. 引言

在 SQL 查询中，子查询是一种嵌套查询，它可以作为一个查询的一部分，通常嵌套在 **SELECT**、**FROM**、**WHERE** 等 SQL 语句中。子查询的主要作用是为主查询提供额外的数据或条件，从而简化复杂的查询逻辑。掌握子查询的使用方法，不仅能帮助你写出更简洁的 SQL 语句，还能提升你的查询效率。

2. 子查询的基本概念

什么是子查询？

子查询是嵌套在另一个查询中的查询，它的执行结果可以作为主查询的一部分。子查询通常用在以下几个地方：

- **WHERE 子句**：用于作为条件来筛选数据。
- **FROM 子句**：作为一个虚拟表来提供数据源。
- **SELECT 子句**：作为计算的结果返回。

子查询的作用

- **提供动态的数据源**：子查询可以返回不同的数据结果，主查询可以根据这些结果进行进一步的筛选。
- **简化复杂查询**：子查询可以将复杂的查询逻辑拆分成多个部分，使查询语句更简洁。
- **增强查询的灵活性**：子查询可以处理复杂的条件和计算，增强 SQL 的表达能力。

3. 子查询的分类

根据不同的使用场景和返回结果，子查询可以分为以下几类：

3.1 根据位置分类

1. **WHERE 子查询**：在 **WHERE** 子句中使用子查询来进行数据筛选。

2. **FROM 子查询**：将子查询作为 FROM 子句的一部分，作为一个临时的虚拟表。
3. **SELECT 子查询**：在 SELECT 子句中使用子查询，返回计算结果。

3.2 根据返回值分类

1. **标量子查询**：返回一个单一值，常用于 WHERE 或 SELECT 中。
2. **行子查询**：返回一行数据，常用于 WHERE 子句中。
3. **表子查询**：返回多行多列的数据，通常用于 FROM 子句。
4. **相关子查询**：在子查询中引用外部查询的列。
5. **非相关子查询**：独立于外部查询的子查询。

4. 子查询的常见应用场景

4.1 用于筛选数据

子查询常用于从一个查询结果中筛选出符合条件的记录。比如，我们可以使用子查询来查找某个部门的员工信息。

示例：通过子查询筛选数据

```
SELECT name
FROM employees
WHERE department_id = (SELECT id FROM departments WHERE name = 'HR');
```

在这个例子中，子查询 (SELECT id FROM departments WHERE name = 'HR') 返回 HR 部门的 ID，然后主查询根据这个 ID 筛选出 employees 表中属于该部门的员工。

4.2 用于聚合数据

子查询还可以用来进行聚合操作，如计算最大值、平均值等。

示例：计算某部门员工的平均工资

```
SELECT avg(salary)
FROM (SELECT salary FROM employees WHERE department_id = 1) AS dept_salaries;
```

在这个例子中，子查询首先筛选出部门 ID 为 1 的员工的工资数据，然后计算这些数据的平均值。

4.3 多表联合查询

子查询可以用于多表联合查询，解决一些复杂的查询需求。例如，我们可以在子查询中联接多张表。

示例：查询员工及其所在部门的信息

```
SELECT name, salary
FROM employees
WHERE department_id IN (SELECT id FROM departments WHERE location = 'New York');
```

在这个例子中，子查询返回 **New York** 位置的部门 ID，主查询根据这些部门 ID 筛选出符合条件的员工信息。

4.4 更新与删除操作中的子查询

子查询也可以在 **UPDATE** 和 **DELETE** 操作中使用，从而根据查询结果进行数据更新或删除。

示例：更新员工的工资

```
UPDATE employees
SET salary = salary * 1.1
WHERE department_id = (SELECT id FROM departments WHERE name = 'HR');
```

在这个例子中，子查询返回 **HR** 部门的 ID，主查询根据该部门 ID 更新该部门所有员工的工资。

4.5 性能优化

在某些情况下，子查询可以帮助优化查询性能，减少数据扫描的范围。通过子查询，我们可以限制主查询需要处理的数据量。

5. 子查询的具体实例

5.1 WHERE 子查询

```
SELECT name
FROM employees
WHERE department_id = (SELECT id FROM departments WHERE name = 'HR');
```

解释：在主查询的 **WHERE** 子句中，使用子查询来动态获取 **HR** 部门的 ID，然后筛选出属于该部门的员工。

5.2 FROM 子查询

```
SELECT avg(salary)
FROM (SELECT salary FROM employees WHERE department_id = 1) AS dept_salaries;
```

解释：子查询首先获取 `department_id` 为 1 的员工工资，然后计算这些员工的平均工资。

5.3 SELECT 子查询

```
SELECT (SELECT COUNT(*) FROM employees) AS employee_count;
```

解释：通过子查询，获取整个员工表中的员工总数，并返回一个计算结果。

5.4 相关子查询

```
SELECT name, salary
FROM employees e
WHERE salary > (SELECT avg(salary) FROM employees WHERE department_id = e.department_id);
```

解释：相关子查询通过引用外部查询中的 `department_id`，动态计算每个部门的平均工资，并筛选出工资高于该部门平均工资的员工。

6. 子查询与连接的比较

6.1 子查询 vs. JOIN

有时，子查询可以替代 `JOIN` 来实现相同的查询逻辑。然而，子查询和 `JOIN` 的性能差异是需要考虑的因素。

子查询示例：

```
SELECT name
FROM employees
WHERE department_id = (SELECT id FROM departments WHERE name = 'HR');
```

`JOIN` 示例：

```
SELECT e.name
FROM employees e
JOIN departments d ON e.department_id = d.id
WHERE d.name = 'HR';
```

性能分析：对于简单的查询，**JOIN** 通常比子查询更高效，尤其是在子查询返回大量数据时，**JOIN** 可以利用索引优化查询速度。

7. 子查询的优化技巧

7.1 避免嵌套子查询

嵌套过深的子查询会影响查询性能，尽量避免在 **WHERE** 子句中使用嵌套查询。如果查询变得复杂，可以考虑拆解查询，使用临时表或者优化查询结构。

7.2 使用 EXISTS 和 IN 的差异

- **EXISTS**：用于检查子查询是否返回任何结果，适用于检查某个条件是否成立。
- **IN**：用于检查某个值是否在子查询返回的结果中，适合用于多值比较。

```
-- EXISTS 示例
SELECT name
FROM employees e
WHERE EXISTS (SELECT 1 FROM departments d WHERE d.id = e.department_id AND d.name = 'HR')

-- IN 示例
SELECT name
FROM employees
WHERE department_id IN (SELECT id FROM departments WHERE name = 'HR');
```

7.3 将子查询转换为 JOIN

对于复杂查询，可以将子查询转化为 **JOIN**，以提高查询性能。例如，当子查询涉及多个表时，**JOIN** 的性能通常优于嵌套子查询。

7.4 使用索引优化子查询

确保在子查询所涉及的列上创建索引，以提高查询效率。例如，**WHERE** 子句中的列、**JOIN** 子句中的列以及 **ORDER BY** 子句中的列都应当建立索引。

8. 子查询常见问题及解决方案

8.1 子查询返回多个结果时如何处理？

当子查询返回多个结果时，可以使用 `IN` 来处理，而不是使用 `=`。如果子查询只需要返回一个值，确保它只返回单个结果。

8.2 子查询导致查询效率低怎么办？

可以考虑将子查询改写为 `JOIN`，或者使用临时表和索引来优化性能。

8.3 相关子查询的性能问题如何处理？

避免使用不必要的相关子查询，考虑拆解查询或者使用 `JOIN`。

结语

子查询是 SQL 中非常有用的工具，能够在复杂查询中提供灵活的解决方案。通过合理地使用子查询，可以简化查询结构并提高查询效率。然而，子查询也有性能瓶颈，特别是嵌套过深时。理解子查询的使用场景、性能优化技巧，并根据实际需求选择合适的查询方式，能够帮助你写出更高效、更简洁的 SQL 语句。

个人观点，仅供参考，非常感谢各位朋友们的支持与关注！

如果你觉得这个作品对你有帮助，请不吝[点赞](#)、[在看](#)，分享给身边更多的朋友。如果你有任何疑问或建议，欢迎在评论区[留言](#)交流。



张慧源

喜欢作者

数据库 · 目录

上一篇

提升效率的秘密武器：MySQL 常用命令速查宝典

下一篇

SQL 全文本搜索深度解读

个人观点，仅供参考