



连接查询你一定不陌生，但你知道还有半连接和反连接吗？执行计划中的反半连接是指什么-CSDN博客



Beta



Beta



举报



不够发的Tony老师

已关注



18



46



1



专栏目录

文章目录

- 半连接
- 反连接
- 总结

大家好，我是只谈技术不剪发的 Tony 老师。在关系数据库中，连接查询（JOIN）可以从两个或多个表中获取相关的数据。我们熟悉的连接查询包括内连接、左/右/全外连接、交叉连接等。除此之外，还有两种特殊的连接查询：**半连接**（Semi Join）和**反连接**（Anti Join）。由于 SQL 标准没有定义这两种连接查询语法，而是通过子查询的方式实现相同的效果，因此，本文就来介绍一下它们的概念和作用。

📖 本文内容适用于各种数据库，包括 Oracle、MySQL、Microsoft SQL Server、PostgreSQL 以及 SQLite 等。

半连接

半连接返回左表中与右表至少匹配一次的数据行，通常体现为 EXISTS 或者 IN 子查询。半连接的示意图如下：



table1 中的 id = 2 在 table2 中没有对应的数据，所以连接的结果不包含该记录。

半连接只会返回左表中的数据，右表只用于条件判断。另外，即使右表中存在多个匹配的数据，左边中的数据只返回一次。半连接通常用于存在性判断，例如哪些顾客购买了产品，而不需要知道他们购买的具体产品和数量。

以下语句用于查找拥有员工的部门（示例数据）：

```
1 SELECT *
2 FROM department d
3 WHERE EXISTS (SELECT 1 FROM employee WHERE dept_id = d.dept_id);
4
5 dept_id|dept_name |
6 -----|-----/
7      1|行政管理部 |
8      2|人力资源部 |
9      3|财务部   |
10     4|研发部   |
11     5|销售部   |
```

以上语句也可以使用 IN 或者 =ANY 操作符加上子查询来实现：

```
1 SELECT *
2 FROM department d
3 WHERE dept_id IN (SELECT dept_id FROM employee);
4
5 SELECT *
6 FROM department d
7 WHERE dept_id = ANY (SELECT dept_id FROM employee);
```

虽然 SQL 没有定义 SEMI JOIN 关键字，但是我们可以通过数据库的执行计划查看相关的信息。以下是 MySQL 数据库中的执行计划：

```
1 | EXPLAIN ANALYZE
2 | SELECT *
3 | FROM department d
4 | WHERE EXISTS (SELECT 1 FROM employee WHERE dept_id = d.dept_id);
5 |
6 | -> Nested loop semijoin  (cost=5.36 rows=30) (actual time=0.056..0.097 rows=5 loops=1)
7 |   -> Table scan on d  (cost=0.85 rows=6) (actual time=0.034..0.042 rows=6 loops=1)
8 |   -> Index lookup on employee using idx_emp_dept (dept_id=d,dept_id)  (cost=1.67 rows=5) (actual time=0.008..0.008 rows=1 loops=6)
```

其中，Nested loop semijoin 表示这是一个嵌套循环的半连接查询。

📖 关于各种数据库执行计划的查看方式和结果解释，可以参考这篇文章。

半连接也可以使用内连接实现，例如：

```
1 SELECT DISTINCT d.*
2 FROM department d
3 JOIN employee e ON e.dept_id = d.dept_id;
```

首先通过内连接获取所有满足条件的数据，然后执行 DISTINCT 操作去除重复值；显然这种方式不如半连接效率高，不过大多数数据库可以实现这两者的等价转换。

反连接

反连接返回左表中与右表不匹配的数据行，通常体现为 NOT EXISTS 或者 NOT IN 子查询。反连接的逻辑与半连接正好相反，示意图如下：



table1 中只有 id = 2 在 table2 中没有对应的数据，所以连接的结果返回了该记录。

反连接只会返回左表中的数据，右表只用于条件判断。反查询常见的应用包括：查找没有员工的部门信息，或者没有购买任何产品的顾客信息等。

例如，以下语句返回了没有员工的部门：

```
1 SELECT *
2 FROM department d
3 WHERE NOT EXISTS (SELECT 1 FROM employee WHERE dept_id = d.dept_id);
4
5 dept_id|dept_name |
6 -----|-----/
7      6|保卫部   |
```

以上语句也可以使用 NOT IN 或者 !=ALL 操作符加上子查询来实现：

```
1 SELECT *
2 FROM department d
3 WHERE dept_id NOT IN (SELECT dept_id FROM employee);
4
5 SELECT *
6 FROM department d
7 WHERE dept_id !=ALL (SELECT dept_id FROM employee);
```

使用 NOT IN 或者 !=ALL 操作符时需要注意子查询中可能出现的 NULL 值。例如：

```
1 SELECT *
2 FROM department d
3 WHERE dept_id NOT IN (1, NULL, 2);
```

以上查询不会返回任何结果，因为它的查询条件实际上等价于：

```
1 SELECT *
2 FROM department d
3 WHERE dept_id !=1 AND dept_id != NULL AND dept_id != 2;
```

其中，dept_id != NULL 导致所有数据都不会满足条件；因为任何数值和 NULL 进行比较的结果都是未知（Unknown），也就是不为真（True）。

虽然 SQL 没有定义 ANTI JOIN 关键字，但是我们可以通过数据库的执行计划查看相关的信息。以下是 MySQL 数据库中的执行计划：

```
1 | EXPLAIN ANALYZE
2 | SELECT *
3 | FROM department d
4 | WHERE NOT EXISTS (SELECT 1 FROM employee WHERE dept_id = d.dept_id);
5 |
6 | -> Nested loop antijoin  (cost=5.36 rows=30) (actual time=0.245..0.255 rows=1 loops=1)
7 |   -> Table scan on d  (cost=0.85 rows=6) (actual time=0.135..0.151 rows=6 loops=1)
8 |   -> Index lookup on employee using idx_emp_dept (dept_id=d,dept_id)  (cost=1.67 rows=5) (actual time=0.016..0.016 rows=1 loops=6)
```

其中，Nested loop antijoin 表示这是一个嵌套循环的反连接查询。

📖 关于各种数据库执行计划的查看方式和结果解释，可以参考这篇文章。

反连接也可以使用外连接实现，例如：

```
1 SELECT DISTINCT d.*
2 FROM department d
3 LEFT JOIN employee e ON e.dept_id = d.dept_id
4 WHERE e.dept_id IS NULL;
```

首先通过左外连接获取所有满足条件的数据，然后使用 WHERE 条件找出右表中不存在的数据，最后执行 DISTINCT 操作去除重复值；显然这种方式不如反连接效率高，不过大多数数据库可以实现这两者的等价转换。

总结

数据库中除了常见的内连接、左/右/全外连接、交叉连接等类型之外，还有两种特殊的连接查询：半连接和反连接。本文介绍了这两种连接的原理和它们在数据库中的等价实现。

如果觉得文章对你有帮助，欢迎关注❤️、评论📄、点赞👍！

