Presto 中支持的七种 Join 类型

SQL Join 是最重要和最昂贵的 SQL 操作之一，需要数据库工程师深入理解才能编写高效的 SQL 查询。 从数据库工程师的角度来看，了解 JOIN 操作的工作原理有助于他们优化 JOIN 以实现高效执行。 本文介绍了开源分布式计算引擎 Presto SQL 支持的 join 操作。

几乎所有众所周知的数据库都支持以下五种类型的 JOIN 操作：Cross Join, Inner Join, Left Join, Right Join 以及 Full Join。另外，Presto 内部还有两种类型的 JOIN 操作：Semi Join 和 Anti Join。本文从使用者的角度介绍了这七种类型的 JOIN 。

Cross Join

Cross Join 返回两张表的笛卡尔积，其 JOIN 过程不带任何条件。换句话说，左表中的每一行都将与右表中的每一行连接起来。 Cross Join 比其他类型的联接更昂贵，因为它在两个表之间进行笛卡尔积。下面就是一个 Cross Join 的例子：

```
presto:iteblog> SELECT * FROM (VALUES 1, 2) t("left"), (VALUES 3, 4) u("right");
 left | right
------+-------
    1 |     3
    2 |     3
    1 |     4
    2 |     4
(4 rows)
```

Inner Join

Inner join 在一定条件下连接两张表。只有在满足给定条件时，表中的每一行才会与另一个表中的每一行连接。 在 Presto SQL 中，INNER JOIN、JOIN 和带有 WHERE 子句的分隔表都被视为 Inner join。如果查询包含带有 where 子句的逗号分隔表，如下所示，它将被读取为 Cross Join，并在连接操作后应用过滤器，后面会由 Presto 优化器重写为 Inner join 查询。

```
presto:iteblog> SELECT * FROM (VALUES 1, 2) t("left"), (VALUES 1, 1, 2) u("right")
           WHERE t."left" = u."right";
 left | right
------+-------
    1 |     1
    1 |     1
    2 |     2
```

(3 rows)

Left Outer Join

与 Inner join 一样，Left Outer Join 将左表中的每一行与右表中的匹配行连接起来。如果左表中的行在右表中没有找到，那么右表中对应的列用空值来填充。 在 Presto SQL 中，关键字 OUTER 在 LEFT OUTER JOIN 操作中是可选的。换句话说，LEFT JOIN 和 LEFT OUTER JOIN 的意思是一样的。

```
presto:iteblog> SELECT * FROM (VALUES 1, 2) t("left")
           LEFT OUTER JOIN (VALUES 1, 1) u("right")
           ON t."left" = u."right";
 left | right
------+-------
    1 |     1
    1 |     1
    2 | NULL
(3 rows)
```

Right Outer Join

Right Outer Join 将右表中的每一行与左表中的匹配行连接起来。如果右表中的行在左表中没有找到，那么左表中对应的列用空值来填充。 在 Presto SQL 中，关键字 OUTER 在 RIGHT OUTER JOIN 操作中是可选的。换句话说，RIGHT JOIN 和 RIGHT OUTER JOIN 的意思是一样的。

```
presto:iteblog> SELECT * FROM (VALUES 1, 2) t("left")
           RIGHT OUTER JOIN (VALUES 1, 2, 3) u("right")
           ON t."left" = u."right";
 left | right
------+-------
    1 |     1
    2 |     2
 NULL |     3
(3 rows)
```

Full Outer Join

Full Outer Join 可以看作是左外连接和右外连接的组合。它将每一侧的每一行与另一侧的匹配行连接起来。如果左表中的行在右表中没有找到，那么右表中对应的列用空值来填充。同样，如果右表中的行在左表中没有找到，那么左表中对应的列用空值来填充。在 Presto SQL 中，关键字 OUTER 在 FULL OUTER JOIN 操作中是可选的。 换句话说，FULL JOIN 和 FULL OUTER JOIN 的意思是一样的。

```
presto:iteblog> SELECT * FROM (VALUES 1, 2) t("left")
           FULL OUTER JOIN (VALUES 1, 3) u("right")
           ON t."left" = u."right";
 left | right
------+-------
    1 |     1
    2 | NULL
 NULL |     3
(3 rows)
```

除了上面提到的 JOIN 类型之外，还有两种 JOIN 类型：Semi Join 以及 Anti Join。 这两个 JOIN 一般是 Presto 改写用户 SQL 产生的。

Semi Join

在 Presto 内部，IN/EXISTS 中带子查询就会被翻译成 Semi Join，Semi Join 有两种变体：Left Semi Join 和 Right Semi Join。顾名思义，Left Semi Join
计算条件，如果左表中有匹配的行，则返回左表中的行。类似地，Right Semi Join
只返回右边表中匹配的行。以下使用 IN 谓词的查询被 Presto 优化器转换成 Semi Join：

```
presto:iteblog> SELECT o_orderkey,o_custkey FROM orders
           WHERE o_orderkey IN (SELECT o_orderkey
                    FROM orders
                    WHERE o_orderstatus = 'o' AND o_orderdate = '1996-01-02');
 o_orderkey | o_custkey
------------+-----------
     454791 |         1
   48313376 |         1
   22402209 |         1
   49312295 |         1
   29469061 |         1
   58877861 |         1
   27843492 |         1
    7855620 |         1
   17904289 |         1
   23747713 |         1
```

(10 rows)

Anti Join

Anti Join 是 Presto 另一种内部使用的 JOIN，用于在联接条件不匹配时从表中返回行。它与 Semi Join 类似，只返回一个表中的行，但 Join 条件与 Semi Join 相反。 例如，可以使用 Anti Join 评估以下 NOT IN 查询。

```
presto:iteblog> SELECT o_orderkey,o_custkey FROM orders
          WHERE o_orderkey IN (SELECT o_orderkey
                    FROM orders
                    WHERE o_orderstatus = 'o' AND o_orderdate = '1996-01-02');

 o_orderkey | o_custkey
------------+-----------
   58781825 |   1002496
   11692451 |    479269
   45393222 |    145972
   53041377 |    371935
   19746180 |    730093
     485186 |    653602
   46238592 |     47099
   43451713 |    152060
   43661861 |     62653
(9 rows)
```

尽管 Anti Join 可以被视为一种不同的 JOIN 类型，但在 Presto 内部通过否定连接条件将其视为 Semi Join。 不仅是 Semi Join，任何 JOIN 操作都可以使用其他 JOIN 组成。例如，通过交互参与 JOIN 的表的位置 ，可以将 left outer join 重写为 right outer join。