

- GROUP BY odps\_orderkey, odps\_orderdate
- HAVING sum\_total > 400000.0;

由上可见，使用常见的 MaxCompute SQL 语法，访问 TableStore 的所有细节由 MaxCompute 内部处理。这包括在列名的选择上，比如上面的SQL中，使用的列名是 odps\_orderkey, odps\_totalprice 等，而不是原始 TableStore 里面的主键名 o\_orderkey 或属性列名 o\_totalprice，因为在创建 External Table 的 DDL 语句里，已经做了对应的 mapping。当然您也可根据自己的需求在创建 External Table 时选择保留原始的 TableStore 主键/列名。

如果需要对一份数据做 **多次计算**，相较每次从 TableStore 去远程读数据，有个更高效的办法是先一次性把需要的数据导入到 MaxCompute 内部成为一个 MaxCompute ( 内部 ) 表。比如：

```
<ol><li> CREATE TABLE internal_orders AS</li><li> SELECT odps_orderkey, odps_orderdate, odps_custkey, odps_totalprice</li><li> FROM ots_table_external</li><li> WHERE odps_orderkey > 5000 ;</li></ol>
```

现在 internal\_orders 就是一个 MaxCompute 表了，也拥有所有 MaxCompute 内部表的特性，包括：高效的压缩列存储数据格式，完整的内部宏数据以及统计信息等。同时因为存储在 MaxCompute 内部，访问速度会比访问外部的 TableStore 更快，尤其适用于需要进行多次计算的热点数据。

## 2. MaxCompute 导出数据到 TableStore

**注意：**MaxCompute 不会主动创建外部的 TableStore 表，所以在对 TableStore 表进行数据输出之前，必须保证该表已经在 TableStore 上创建过(否则将报错)。

根据上面的操作，您已创建了外部表 ots\_table\_external 来打通 MaxCompute 与 TableStore 数据表 ots\_tpch\_orders 的链路，同时还有一份存储在 MaxCompute 内部表 internal\_orders 的数据，现在希望对 internal\_orders 中的数据进行一定处理后再写回 TableStore 上。可通过对外部表做 **INSERT OVERWRITE TABLE** 操作来实现。如下所示：

```
<ol><li> INSERT OVERWRITE TABLE ots_table_external</li><li> SELECT odps_orderkey, odps_orderdate, odps_custkey, CONCAT ( odps_custkey, 'SHIPPED' ), CEIL ( odps_totalprice
```