Flink SQL base 整理

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 类别 | Function | Description | Example | Calcite support | Flink support |
| Scan | select | 查询 | **select user,product,amount from t\_order** | **Y** | **Y** |
| Aggregate | Group by | 聚合 | **select product,sum(amount) as total from t\_order group by product** | Y | Y |
|  | Having | 聚合条件过滤 | **select product,sum(amount) as total from t\_order group by product Having sum(amount) > 30** | Y | Y |
|  | Distinct | 去重 | **select distinct from t\_order** | **Y** | **Y** |
|  | Distinct aggregate | 聚合计数操作去重 | **select count(distinct user) from t\_order** | **Y** | **N** |
| Filter | Where | 条件过滤 | **select \* from t\_order where user = 2 or product = 'productB'** | **Y** | **Y** |
| Join | Inner join | 内连接 | **select \* from t\_order inner join t\_product on product = name** | Y | Y |
|  | Left join | 左外连接 | **select \* from t\_order left join t\_product on product = name** | Y | Y |
|  | Right join | 右外连接 | **select \* from t\_order right join t\_product on product = name** | Y | Y |
|  | Full join | 全连接 | **select \* from t\_order full join t\_product on product = name** | Y | Y |
|  | Cross join | 笛卡尔连接 | **select \* from t\_order cross join t\_product** | Y | N |
|  | Join without condition | 无条件的连接 | **Select \* from t\_order,t\_product**  **select \* from t\_order inner join t\_product on 1=1** | Y | N |
|  | NonEqui-join | 非等值连接 | **select \* from t\_order,t\_product where price > amount**  **select \* from t\_order inner join t\_product on product <> name** | Y | N |
| Other | Alias | 重命名as | **SELECT p.product, p.user FROM t\_order AS p** | **Y** | **Y** |
|  | union | 合并去重 | **select \* from t\_orderA union select \* from t\_orderB** | **Y** | **Y** |
|  | unionAll | 合并不去重 | **select \* from t\_orderA union all select \* from t\_orderB** | **Y** | **Y** |
|  | Order | 排序 | **select \* from t\_order order by user,product DESC** | **Y** | **Y** |
|  | Limit | 取前N条记录 | **select \* from t\_order order by user DESC limit 3** | **Y** | **Y** |
|  | With..as | Sql 片段 | **with t\_result as**  **(select case when amount < 10 then 'low' "** + **when amount < 20 then 'medium'**  **else 'high' end from t\_order)**  **select \* from t\_result** | **Y** | **Y** |
|  | Case when  Else  end | Case语句 | **select case when amount < 10 then 'low'**  **when amount < 20 then 'medium' else 'high' end from t\_order** ; | **Y** | **Y** |
| 操作符 | Like | 近似匹配 | **select product from t\_order where product like 'product\_'** | **Y** | **Y** |
|  | Not like | 近似匹配 | **select product from t\_order where product not like '%\_B'** | **Y** | **Y** |
|  | Between | 范围 | **select \* from t\_order where user between 2 and 4** | **Y** | **Y** |
|  | WildCard | 通配符 | **select product from t\_order where product like 'product\_'** | **Y** | **Y** |
|  | In | 判断是否在多个值中 | **select \* from t\_order where product in ('productA','productB')** | **Y** | **Y** |
|  | Not in | 判断是否不在多个值中 | **select \* from t\_order where product not in ('productA','productB')** | **Y** | **Y** |
|  | Exists | 判断子句是否值 | **select \* from t\_order where exists(select user from t\_order where user = 2)** | **Y** | **Y** |
|  | Not exists | 判断子句是否为空 | **select \* from t\_order where not exists(select user from t\_order where user = 6)** | **Y** | **N** |
| 函数 | Sum | 求和 | **select sum(amount) from t\_order** | **Y** | **Y** |
|  | Average | 平均 | **select avg(amount) from t\_order** | **Y** | **Y** |
|  | Max | 最大值 | **select max(amount) from t\_order** | **Y** | **Y** |
|  | Min | 最小值 | **select min(amount) from t\_order** | **Y** | **Y** |
|  | Count | 计数 | **select count(user) from t\_order** | **Y** | **Y** |
|  | cast | 类型转化 | **select EXTRACT(year from cast('2008-12-29' as date)) from t\_order** | **Y** | **Y** |
|  | substr | 求字符串的子串 | **select substring(product from 1 for 2) from t\_order**  **select substring(product,1,2) from t\_order** | **Y** | **Y** |
|  | Rank(),  Row\_number(),  Dense\_rank() | 排名函数 | **select user, rank() over(partition by user order by amount) from t\_order**  **select row\_number() over(order by amount DESC) as rowid,\* from t\_order** | **N** | **N** |
|  | Concat(s1,s2…) | 字符串连接 | **select concat(amount,product) from t\_order** | **N** | **N** |
|  | Coalesce(exp1,exp2,…) | 返回第一个非空值 | **select coalesce(null,null,null,0) from t\_order** | **Y** | **Y** |
|  | over | 开窗函数 | **"select sum(user)over(order by amount DESC) as rowid,\* from t\_order"**; | **N** | **N** |

PS：以上只是sql语法的一些基本要素，更多的语法规则使用会根据主流测试基准提供的Query进行测试然后添加。