Flink SQL 功能解密系列 —— 数据去重的技巧和思考

概述

去重逻辑在业务处理中使用广泛，大致可以分两类：DISTINCT去重和FIRST_VALUE主键去重，两者的区别是DISTINCT去重是对整行数据进行去重，比如tt里面数据可能会有重复，我们要去掉重复的数据；FIRST_VALUE是根据主键进行去重，可以看成是一种业务层面的去重，但是真实的业务场景使用也很普遍，比如一个用户有多次点击，业务上只需要取第一条。本文重点介绍这两种去重的应用。

1. DISTINCT 去重

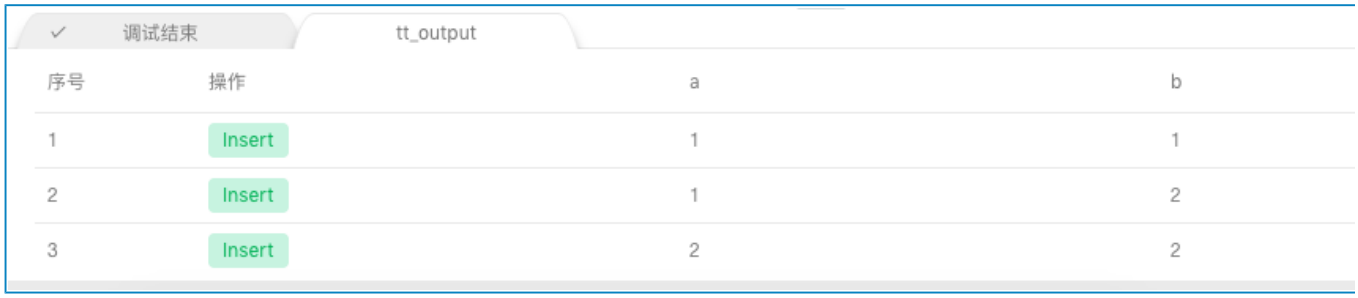
blink sql支持标准sql的DISTINCT去重。假如我们有如下输入数据，并希望对相同的行进行去重。

sql可以这么写：`select distinct * from tt_source;` 完整的blink sql如下，

```
create table tt_source(
  a varchar,
  b varchar
)with(
  type='tt',
  topic='se_taobao_wireless_click',
  accessId='08061416466YCN3FIU',
  accessKey='xxxxx'
  lengthCheck='PAD'
);
```

```
create table tt_output(
  a varchar,
  b varchar
)with(
  type='tt',
  topic='blink_test_32_1',
  accessKey='xxxx'
);

insert into tt_output
select distinct * from tt_source;
```

输出时，会对第一行(1,1)和第二行(1,1)数据进行去重。输出结果如下



2. FIRST_VALUE udaf去重

还有一种情况是根据primary key字段进行去重，即如果两行数据主键相同，即使其他非主键字段不一样，还是只取第一行数据。这种情况，我们可以使用FIRST_VALUE udaf函数来达到去重的目的。
对于如下输入，并希望根据主键a来去重数据：

sql可以这么写：

```
INSERT INTO tt_output
SELECT
  a,
```

```
  FIRST_VALUE(b)
FROM tt_source
GROUP BY a;
```

完整的blink sql如下，

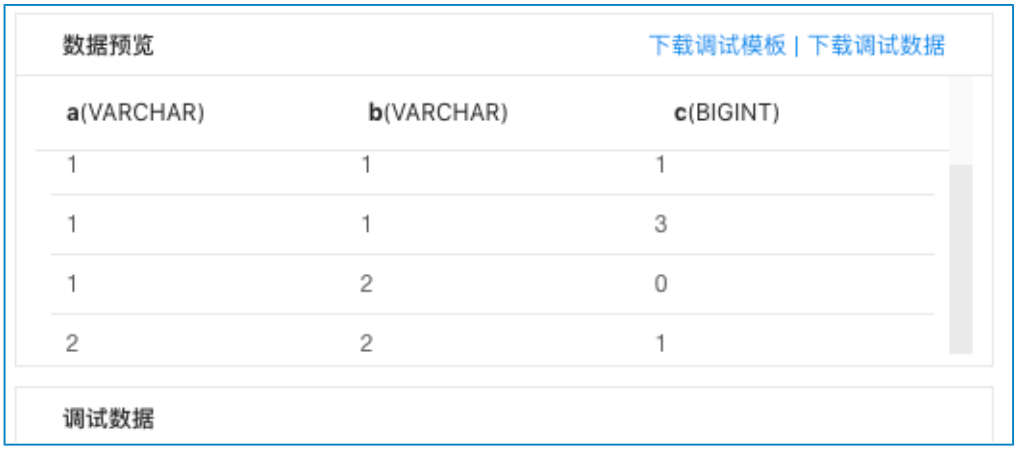
```
CREATE TABLE tt_source(
  a VARCHAR,
  b VARCHAR
)WITH(
  type='tt',
  topic='se_taobao_wireless_click',
  accessId='08061416466YCN3FIU',
  accessKey='xxx',
  lengthCheck='PAD'
);

CREATE TABLE tt_output(
  a VARCHAR,
  b VARCHAR
)WITH(
  type='tt',
  topic='blink_test_32_1',
  accessKey='xxx'
);

INSERT INTO tt_output
SELECT
  a,
  FIRST_VALUE(b)
FROM tt_source
GROUP BY a;
```

输出结果：

可以看到主键a相同的3行，只取了第一行。

FIRST_VALUE还支持传一个order参数，根据order来决定first是哪行，使用的方法是`FIRST_VALUE(b, c)`，但是要注意，c字段只能是BIGINT。假如我们有如下输入，对于相

同的主键，我们希望取c最小的记录（实际场景c一般是时间字段）。



完整的blink sql如下，

```
CREATE TABLE tt_source(
  a VARCHAR,
  b VARCHAR,
  c BIGINT
)WITH(
  type='tt',
  topic='se_taobao_wireless_click',
  accessId='08061416466YCN3FIU',
  accessKey='xxx',
  lengthCheck='PAD'
);

CREATE TABLE tt_output(
  a VARCHAR,
  b VARCHAR
)WITH(
  type='tt',
  topic='blink_test_32_1',
  accessKey='xxx'
);

INSERT INTO tt_output
SELECT
  a,
  FIRST_VALUE(b, c)
FROM tt_source
GROUP BY a;
```

输出结果：

序号	操作	a	b
1	Insert	1	1
2	Insert	1	2
3	Insert	2	2

可以看到当输出（1,1,1）后，由于又来了（1,2,0），0比1要小，所以又更新了主键为1的记录，输出（1，2）