例子：

甲、乙、丙三人各自下了一个订单，如下：

其中，括号里的是规格，括号前面的是spu名称

经过拆单处理并且运营人员手动操作后，现在的发货订单列表如下：

|  |  |  |
| --- | --- | --- |
| 1 | 甲（**自动拆单后取消所有拆单**） | 衣服(红色)×1; 衣服(绿色)×1；裤子(白色)×1 |
| 2 | 乙（自动拆单子订单） | 衣服(红色)×1 |
| 3 | 乙（**自动拆单后还手动拆了数量**） | 裤子(卡其色)×1 |
| 4 | 乙（**上面拆了数量的副产物**） | 裤子(卡其色)×1 |
| 5 | 丙（自动拆单结果） | 鞋子(荧光绿)×1 |

导出订单时，

导出总商品数量为

1(甲衣红) + 1(甲衣绿) + 1(甲裤白) + 1(乙衣红) + 2(乙裤卡其) + 1(丙鞋荧光绿) = 7

导出总订单数为

1(甲单) + 1(乙衣服拆单) + 1(乙裤子拆单) + 1(乙裤子拆单2) + 1(丙鞋子单) = 5

对于订单商品数据统计如下：

总订单数只与拆单前的订单相关，不受拆单影响

与衣服相关的订单有 [甲订单，乙订单] ，所以衣服导出总订单数为 2

与裤子相关的订单有 [甲订单，乙订单] ，所以裤子导出总订单数为 2

与鞋子相关的订单只有 [丙订单] ，所以鞋子导出总订单数为 1

导出总订单数跟发货订单相关，受拆单影响

与衣服相关的订单有 [订单1，订单2] ，所以衣服导出总订单数为 2

与裤子相关的订单有 [订单1，订单3，订单4] ，所以裤子导出总订单数为 3

与鞋子相关的订单有 [订单5] ，所以裤子导出总订单数为 1

sku订单数只与商品在订单中出现的次数相关（数量一般等于自动拆单后的相关订单数）。它不受合单、拆单的影响。举个例子，一个订单P中存在两个sku(sku1, sku2)，这两个sku都归属于同一个spu，是同款产品的不同规格，计算这个商品的sku订单数时，sku1在订单P中出现1次，sku2在订单P也出现1次。无论它们的数量是多少，还是最后拆分了成多少个订单。对于这个spu，sku订单数都是 1(订单P有sku1) + 1(订单P有sku2) = 2

**对于上述 衣服 这个商品**

与衣服(红色)相关的订单有 [甲订单，乙订单] ，衣服(红色) sku订单数为 2

与衣服(绿色)相关的订单有 [甲订单] ，衣服(绿色) sku订单数为 1

衣服sku订单数 = 2(红衣服) + 1(绿衣服) = 3

**对于上述 裤子 这个商品**

与裤子(白色)相关的订单有 [甲订单] ，裤子(白色) sku订单数为 1

与裤子(卡其色)相关的订单有 [乙订单] ，裤子(卡其色) sku订单数为 1

裤子sku订单数 = 1(白裤) + 1(卡其裤) = 2

不受拆数量后乙订单中卡其色裤子的订单3，订单4影响

**对于上述 鞋子 这个商品**

与鞋子(荧光绿)相关的订单只有 [丙订单]，鞋子(荧光绿)sku订单数为 1

鞋子sku订单数 = 1(荧光绿鞋) = 1

总销售数跟发货订单里的sku数量相关，不受拆单影响

衣服的销售量为

1(甲衣红) + 1(甲衣绿) + 1(乙衣红) = 3

裤子的销售量为

1(甲裤白) + 2(乙裤卡其) = 3

鞋子的销售量为

1(丙鞋荧光绿) = 1

在上述例子中，导出详情为

衣服： 衣服(红色)×2，衣服(绿色)×1

裤子： 裤子(白色)×1，裤子(卡其色)×2

鞋子： 鞋子(荧光绿)×1

导出详情中的数据只与商品数量相关，不受订单数目影响

最终结果为：

导出总商品数：7  导出总订单数：5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 商品名 | 导出总订单数 | sku订单数 | 导出总销售量 | 导出总销售详情 |
| 衣服 | 2 | 3 | 3 | 衣服(红色)×2，衣服(绿色)×1 |
| 裤子 | 3 | 2 | 3 | 裤子(白色)×1  裤子(卡其色)×2 |
| 鞋子 | 1 | 1 | 1 | 鞋子(荧光绿)×1 |

具体数值请查看上述详解。