Assignment 2: Join Aggregate Count with Stores KTable

Add the stores information to the aggregation thats already have an aggregation that 's performed on the count.

Questions for this assignment:

1. Create a new domain that represents the TotalCountWithAddress. This will have two properties **count** and **store** that represents the total running count and store information.
2. Implement the join in the aggregateOrdersByCount function.

Download resource file:

OrdersTopology.java

Instructor's Example:

1. Create a new domain that represents the TotalCountWithAddress. This will have two properties **count** and **store** that represents the total running count and store information.

```
public record TotalCountWithAddress (Long count,
                                     Store store) {
}
```

2. Implement the join in the aggregateOrdersByCount function.

```
private static void aggregateOrdersByCount(KStream<String, Order>
generalOrdersStream, String storeName, KTable<String, Store> storesTable) {

    var ordersCountPerStore = generalOrdersStream
            //.map((key, value) -> KeyValue.pair(value.locationId(), value) )
            .groupByKey(Grouped.with(Serdes.String(),
SerdesFactory.orderSerdes()))
            .count(Named.as(storeName), Materialized.as(storeName));

    ordersCountPerStore
            .toStream()
            .print(Printed.<String, Long>toSysOut().withLabel(storeName));


    ValueJoiner<Long, Store, TotalCountWithAddress> valueJoiner =
TotalCountWithAddress::new;

    var revenueWithStoreTable = ordersCountPerStore
            .join(storesTable, valueJoiner);

    revenueWithStoreTable
            .toStream()
            .print(Printed.<String,
TotalCountWithAddress>toSysOut().withLabel(storeName+"-bystore"));


}
```

Download resource file

OrdersTopology.java