Atitit 常见sql和stream api转换表

目录

[1.1. Map运算 Select udf(fld) from xxx 1](#_Toc12477)

[1.2. Filter运算 条件查询wehre 1](#_Toc15521)

[1.3. 聚合运算sum 3](#_Toc2192)

## Map运算 Select udf(fld) from xxx

**public** **static** String toStrArr(String condFldVals) {

List li=Lists.*newArrayList*();

String[] a=condFldVals.split(",");

List<String> sList = Arrays.*asList*(a);

List li\_rzt=sList.stream().map(s-> "'"+s+"'").collect(Collectors.*toList*());

**return** Joiner.*on*(",").join(li\_rzt);

}

## Filter运算 条件查询wehre

// 查询税率 数据表选择运算

**private** **static** Map selectTaxrateFrom\_taxRateTable\_where\_loc\_and\_itemtype(Object loc, Object itemtype) {

// 税率表

List<Map> taxRateTable = **new** ArrayList() {

{

// loc,sale,type(food)

add(MapBldr.*newx*().put("loc地点", "CA").put("type", "other").put("tax rate税率", "9.75%")

.put("tax\_rate\_num税率数字格式", 0.0975).build());

add(MapBldr.*newx*().put("loc地点", "CA").put("type", "food").put("tax rate税率", "0").build());

add(MapBldr.*newx*().put("loc地点", "NY").put("type", "other").put("tax rate税率", " 8.875%")

.put("tax\_rate\_num税率数字格式", 0.08875).build());

add(MapBldr.*newx*().put("loc地点", "NY").put("type", "food").put("tax rate税率", "0").build());

add(MapBldr.*newx*().put("loc地点", "NY").put("type", "cloth").put("tax rate税率", "0").build());

}

};

List<Map> result = taxRateTable.stream().filter(map\_item -> {

**return** loc.equals(map\_item.get("loc地点").toString()) && itemtype.equals(map\_item.get("type").toString());

// return true;

}).collect(Collectors.*toList*());

**return** result.get(0);

}

// 查询物品类型

**private** **static** Object gettype(Object name) {

List<Map> itemTypetable = **new** ArrayList() {

{

// loc,sale,type(food)

add(MapBldr.*newx*().put("item物品名", "potato chips").put("type", "food").build()

);

add(MapBldr.*newx*().put("item物品名", "shirt").put("type", "cloth").build());

}

};

List<Map> result = itemTypetable.stream().filter(map\_item -> {

**return** name.equals(map\_item.get("item物品名").toString());

// return true;

}).collect(Collectors.*toList*());

**if** (result.size() == 0)

**return** "other";

**return** result.get(0).get("type");

}

## 聚合运算sum

// 内存数据表聚合运算 物品总价不含税

// select sum(物品价格不含税) from 物品列表

DecimalFormat df2 = **new** DecimalFormat("##0.00");// 这样为保持2位

**double** subtotal = ((List<Map>) input1.get("shoplist物品列表")).stream()

.mapToDouble(i -> (**double**) i.get("item\_total物品价格不含税")).sum();

input1.put("subtotal物品总价不含税", (df2.format(subtotal + 0)));