## 模块A：大数据平台搭建（容器环境）

**环境说明：**

|  |
| --- |
| **服务端登录地址详见各模块服务端说明。**  **补充说明：**必须使用SSH工具连接宿主机（也就是大数据平台搭建实训任务的那台虚拟机）的直连ip进行操作  进入Master节点的方式为  docker exec -it master /bin/bash  进入Slave1节点的方式为  docker exec -it slave1 /bin/bash  进入Slave2节点的方式为  docker exec -it slave2 /bin/bash  所有相关安装包在容器Master节点的/opt/software目录下 |

### 任务一：Hadoop 完全分布式安装配置

本环节需要使用root用户完成相关配置，安装Hadoop需要配置前置环境。命令中要求使用绝对路径，具体要求如下:

1. 将master节点JDK安装包解压到/opt/module路径中(若路径不存在，则需新建)，将JDK解压命令复制并粘贴至对应报告中;
2. 修改容器中/etc/profile文件，设置JDK环境变量并使其生效，配置完毕后在master节点分别执行“java -version”和“javac”命令，将命令行执行结果分别截图并粘贴至对应报告中;
3. 请完成host相关配置，将三个节点分别命名为master、slave1、slave2，并做免密登录，用scp命令并使用绝对路径从master复制JDK解压后的安装文件到slave1、slave2节点（若路径不存在，则需新建），并配置slave1，slave2相关环境变量，将全部与JDK的scp复制命令复制并粘贴至对应报告中;
4. 在master将Hadoop解压到/opt/module(若路径不存在，则需新建)目录下，并将解压包分发至slave1、slave2中，其中master，slave1，slave2节点均作为datanode,配置好相关环境，初始化Hadoop环境namenode，将初始化命令及初始化结果截图（截取初始化结果日志最后20行即可）粘贴至对应报告中；
5. 启动hadoop集群（包括hdfs和yarn），使用jps命令查看master节点与slave1节点的java进程，将jps命令与结果截图粘贴至对应报告中。

### 任务二：ClickHouse单机安装配置

本环节需要使用root用户完成相关配置，具体要求如下：

1. 将Master节点ClickHouse相关安装包解压到/opt/module/clickhouse目录下（若路径不存在，则需新建），将命令复制并粘贴至对应报告中；
2. 执行启动各个相关脚本，将命令复制并粘贴至对应报告中；
3. 设置远程访问并移除默认监听文件（listen.xml），同时由于9000端口被hadoop占用，需要将clickhouse的端口更改为9001，将上述要求的配置或命令复制并粘贴至对应报告中；
4. 启动clickhouse，启动后查看clickhouse运行状态，并将命令和结果截图粘贴至对应报告中。

### 任务三：HBase分布式安装配置

本环节需要使用root用户完成相关配置，安装HBase需要配置前置环境。命令中要求使用绝对路径，具体要求如下:

1. 确认是否完成Hadoop和Zookeeper的分布式（Hadoop伪分布式不可以）安装部署，没有的话请进行安装部署并启动。完成部署后在三个节点使用jps命令，并将结果截图粘贴至对应报告中；
2. 将Master节点Hbase安装包解压到/opt/module目录下，将命令复制并粘贴至对应报告中；
3. 将解压包分发至slave1、slave2中，并修改相关配置和环境。配置好环境变量，在容器中/opt目录下运行命令hbase version，将命令与结果至对应报告中;
4. 启动HBase后在各节点使用jps命令查看，并将结果截图粘贴至对应报告中；正常启动后在hbase shell中查看命名空间，将命令和结果截图粘贴至对应报告中；

## 模块B：数据抽取

**环境说明：**

|  |
| --- |
| **服务端登录地址详见各模块服务端说明。**  **补充说明：**集群启动命令为在Master上执行allstart.sh；  各主机可通过SSH客户端进行SSH访问（需通过直连IP访问）；  Master节点MySQL数据库用户名/密码：root/123456（已配置远程连接，电商离线数据库为ds\_db01）；  Hive的配置文件位于/opt/module/hive-3.1.2/conf/  Spark任务在Yarn上用Client运行，方便观察日志。  Flink任务在Yarn上用per job模式（即Job分离模式，不采用Session模式），方便Yarn回收资源  **注：该Spark版本无法进行本地调试，请打包上传集群调试。** |

### 任务一：离线数据抽取

使用Scala编写spark工程代码，将MySQL的shtd\_store库中表user\_info、sku\_info、base\_province、base\_region、order\_info、order\_detail的数据抽取到Hive的ods库中对应表user\_info、sku\_info、base\_province、base\_region、order\_info、order\_detail中。

1. 抽取shtd\_store库中order\_info的增量数据进入Hive的ods库中表order\_info，根据ods.order\_info表中operate\_time或create\_time作为增量字段(即MySQL中每条数据取这两个时间中较大的那个时间作为增量字段去和ods里的这两个字段中较大的时间进行比较)，只将新增的数据抽入，字段名称、类型不变，同时添加静态分区，分区字段类型为String，且值为当前比赛日的前一天日期（分区字段格式为yyyyMMdd）。使用hive cli执行show partitions ods.order\_info命令，将结果截图粘贴至对应报告中；
2. 抽取shtd\_store库中order\_detail的增量数据进入Hive的ods库中表order\_detail，根据ods.order\_detail表中create\_time作为增量字段，只将新增的数据抽入，字段名称、类型不变，同时添加静态分区，分区字段类型为String，且值为当前比赛日的前一天日期（分区字段格式为yyyyMMdd）。使用hive cli执行show partitions ods.order\_detail命令，将结果截图粘贴至对应报告中。
3. 抽取shtd\_store库中user\_info的全量数据进入Hive的ods库中表user\_info。字段名称、类型不变，同时添加动态分区，分区字段类型为String，且值为birthday列数据，格式为（yyyyMM）。使用hive cli执行select count(distinct(etldate)) from ods.user\_info命令，将结果截图粘贴至对应报告中；
4. 抽取shtd\_store库中sku\_info的增量数据进入Hive的ods库中表sku\_info。根据ods.sku\_info表中create\_time作为增量字段，只将新增的数据抽入，字段名称、类型不变，同时添加静态分区，分区字段类型为String，且值为当前比赛日的前一天日期（分区字段格式为yyyyMMdd）。使用hive cli执行show partitions ods.sku\_info命令，将结果截图粘贴至对应报告中；
5. 抽取shtd\_store库中base\_province的全量数据进入Hive的ods库中表base\_province。字段名称、类型不变并添加字段create\_time取当前时间，同时添加静态分区，分区字段类型为String，且值为当前比赛日的前一天日期（分区字段格式为yyyyMMdd）。使用hive cli执行show partitions ods.base\_province命令，将结果截图粘贴至对应报告中；
6. 抽取shtd\_store库中base\_region的全量数据进入Hive的ods库中表base\_region。字段名称、类型不变并添加字段create\_time取当前时间，同时添加静态分区，分区字段类型为String，且值为当前比赛日的前一天日期（分区字段格式为yyyyMMdd）。使用hive cli执行show partitions ods.base\_region命令，将结果截图粘贴至对应报告中；

### 任务二：实时数据采集

1. 在Master节点使用Flume采集实时数据生成器25001端口的socket数据（实时数据生成器脚本为Master节点/data\_log目录下的gen\_ds\_data\_to\_socket脚本，该脚本为Master节点本地部署且使用socket传输），将数据存入到Kafka的Topic中（Topic名称为ods\_mall\_log，分区数为4），使用Kafka自带的消费者消费ods\_mall（Topic）中的数据，查看前2条数据的结果；

**注：需先启动已配置好的Flume再启动脚本，否则脚本将无法成功启动，启动方式为进入/data\_log目录执行./gen\_ds\_data\_to\_socket (如果没有权限，请执行授权命令chmod 777 /data\_log/gen\_ds\_data\_to\_socket)脚本会生成csv格式及将数据插入mysql**

1. 在Master节点进入到maxwell-1.29.0的解压后目录下（在/opt/module），配置相关文件并启动，读取MySQL数据的binlog日志（mysql的binlog相关配置已完毕）到Kafka的Topic中（Topic名称为ods\_mall\_data，分区数为4）。使用Kafka自带的消费者消费ods\_mall\_data（Topic）中的数据，查看前2条数据的结果；

## 模块C：数据采集与实时计算

**环境说明：**

|  |
| --- |
| **服务端登录地址详见各模块服务端说明。**  **补充说明：**集群启动命令为在Master上执行allstart.sh；  各主机可通过SSH客户端进行SSH访问（需通过直连IP访问）；  Master节点MySQL数据库用户名/密码：root/123456（已配置远程连接，电商离线数据库为ds\_db01）；  Hive的配置文件位于/opt/module/hive-3.1.2/conf/  Spark任务在Yarn上用Client运行，方便观察日志。  Flink任务在Yarn上用per job模式（即Job分离模式，不采用Session模式），方便Yarn回收资源  **注：该Spark版本无法进行本地调试，请打包上传集群调试。**  **推荐使用gosn解析json 要求打包运行** |

### 任务一：实时数据处理

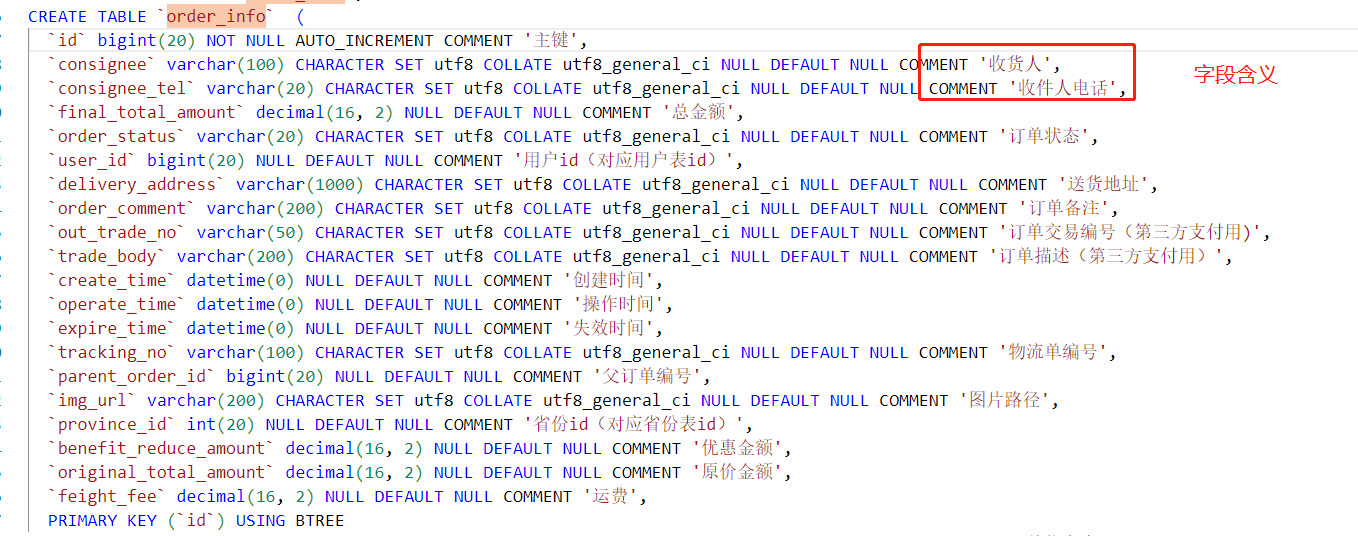
1. 使用Flink消费Kafka中topic为ods\_mall\_log的数据，根据数据中不同的表将数据分别分发至kafka的DWD层的fact\_order\_master、fact\_order\_detail的Topic中（将脚本数据格式转为json格式），其他的表则无需处理，其中请对进入到fact\_order\_master的内容进行排序，同时迟到容忍度设置为2分钟；使用kakfa自带的消费者，查看前两条数据
2. 使用Flink消费Kafka中topic为ods\_mall\_data的数据，根据数据中不同的表前缀区分（只获取data的内容，具体的内容格式考生请自查），将数据分别分发至kafka的DWD层的dim\_customer\_login\_log的Topic中，其他的表则无需处理；使用kakfa自带的消费者，查看前两条数据
3. 在任务1，2进行的同时，需要将order\_info、order\_detail、customer\_login\_log备份至Hbase中，同时建立Hive外表(表结构与离线数据ods层一致，timestamp格式可用string或者long代替，rowkey使用随机数（0-9）+yyyyMMddHHmmssSSS，其中对于customer\_login\_log缺失主键，请用随机数（0-9）+用户id+登陆时间代替)，同时在Hive中查询每个表前5条数据；

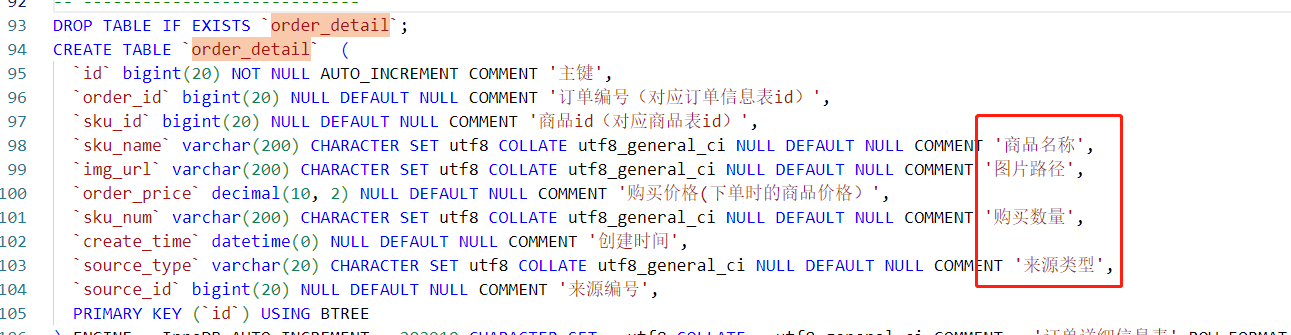
### 任务二：使用Flink处理Kafka中的数据

编写Scala代码，使用Flink消费Kafka中Topic为order的数据并进行相应的数据统计计算（订单信息对应表结构order\_info,订单详细信息对应表结构order\_detail（来源类型和来源编号这两个字段不考虑，所以在实时数据中不会出现），同时计算中使用order\_info或order\_detail表中create\_time或operate\_time取两者中值较大者作为EventTime，若operate\_time为空值或无此属性，则使用create\_time填充，允许数据延迟5S，订单状态分别为1001:创建订单、1002:支付订单、1003:取消订单、1004:完成订单、1005:申请退回、1006:退回完成。另外对于数据结果展示时，不要采用例如：1.9786518E7的科学计数法）。

**注：**与订单金额计算相关使用order\_money字段，同一个订单无需多次重复计算，需要考虑退款或者取消的订单。

数据结构：order\_info



order\_detail

1. 使用Flink消费Kafka中的数据，统计商城每分钟订单实收金额（需要考虑订单状态，若有取消订单、申请退回、退回完成则不计入订单实收金额，其他状态的则累加），将key设置成totalprice存入Redis中。使用redis cli以get key方式获取totalprice值，将结果截图粘贴至对应报告中，需两次截图，第一次截图和第二次截图间隔1分钟以上，第一次截图放前面，第二次截图放后面；
2. 编写Scala工程代码，使用Flink消费kafka中的数据，统计所有用户每分钟订单总数与订单总额，然后计算出每分钟订单平均金额，将key设置成avgprice存入redis中。使用redis cli以get key方式获取avgprice值，将结果截图粘贴至对应报告中，需两次截图，第一次截图和第二次截图间隔1分钟以上，第一次截图放前面，第二次截图放后面；

3、采用双流JOIN的方式（本系统稳定，无需担心数据迟到与丢失的问题,建议使用滚动窗口），结合订单信息和订单详细信息（需要考虑订单状态，若有取消订单、申请退回、退回完成则不进行统计），拼接成如下表所示格式，其中包含订单id、订单总金额、商品数，将数据存入Clickhouse数据库shtd\_result的orderpostiveaggr表中（表结构如下），然后在Linux的Clickhouse命令行中根据id降序排序，查询出前5条，将SQL语句与执行结果截图粘贴至对应报告中。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 中文含义 | 备注 |
| id | int | 订单id |  |
| orderprice | double | 订单总金额 | 统计订单信息中 final\_total\_amount字段 |
| orderdetailcount | int | 商品数量总计 | 累加订单详细信息中 sku\_num字段 |

## 模块D：离线数据处理

**环境说明：**

|  |
| --- |
| **服务端登录地址详见各模块服务端说明。**  **补充说明：**集群启动命令为在Master上执行allstart.sh；  各主机可通过SSH客户端进行SSH访问（需通过直连IP访问）；  Master节点MySQL数据库用户名/密码：root/123456（已配置远程连接，电商离线数据库为ds\_db01）；  Hive的配置文件位于/opt/module/hive-3.1.2/conf/  Spark任务在Yarn上用Client运行，方便观察日志。  Flink任务在Yarn上用per job模式（即Job分离模式，不采用Session模式），方便Yarn回收资源  **注：该Spark版本无法进行本地调试，请打包上传集群调试。** |

**环境说明：**

**提示**：若完成了实时计算的任务二中的第3小题后，可以将HBase中备份的数据与ods中的离线数据进行合并，抽取到dwd（需自建）的对应表中。

### 任务一：数据清洗

使用Scala编写spark工程代码，将ods库中相应表数据全量抽取到Hive的dwd库中对应表中。表中有涉及到timestamp类型的，均要求按照yyyy-MM-dd HH:mm:ss，不记录毫秒数，若原数据中只有年月日，则在时分秒的位置添加00:00:00，添加之后使其符合yyyy-MM-dd HH:mm:ss。

1. 将ods库中user\_info表数据抽取到dwd库中dim\_customer的分区表，分区字段为etldate且值与ods库的相对应表该值相等，并添加dwd\_insert\_user、dwd\_insert\_time、dwd\_modify\_user、dwd\_modify\_time四列,其中dwd\_insert\_user、dwd\_modify\_user均填写“user1”，dwd\_insert\_time、dwd\_modify\_time均填写当前操作时间，并进行数据类型转换。在hive cli中按照id顺序排序，查询dim\_customer前1条数据，将结果内容复制粘贴至对应报告中；
2. 抽取ods库sku\_info表中昨天的分区（任务一生成的分区）数据，并结合dim\_sku\_info最新分区现有的数据，根据id合并数据到dwd库中dim\_sku\_info的分区表（合并是指对dwd层数据进行插入或修改，需修改的数据以id为合并字段，根据create\_time排序取最新的一条），分区字段为etl\_date且值与ods库的相对应表该值相等，并添加dwd\_insert\_user、dwd\_insert\_time、dwd\_modify\_user、dwd\_modify\_time四列,其中dwd\_insert\_user、dwd\_modify\_user均填写“user1”。若该条数据第一次进入数仓dwd层则dwd\_insert\_time、dwd\_modify\_time均填写当前操作时间，并进行数据类型转换。若该数据在进入dwd层时发生了合并修改，则dwd\_insert\_time时间不变，dwd\_modify\_time存当前操作时间，其余列存最新的值。使用hive cli查询表dim\_sku\_info的字段id、sku\_desc、dwd\_insert\_user、dwd\_insert\_time、dwd\_modify\_user、dwd\_modify\_time、etl\_date，条件为最新分区的数据，id小于等于20，并且按照id升序排序，将结果截图粘贴至对应报告中；
3. 将ods库中base\_province表数据抽取到dwd库中dim\_customer的分区表，分区字段为etldate且值与ods库的相对应表该值相等，并添加dwd\_insert\_user、dwd\_insert\_time、dwd\_modify\_user、dwd\_modify\_time四列,其中dwd\_insert\_user、dwd\_modify\_user均填写“user1”，dwd\_insert\_time、dwd\_modify\_time均填写当前操作时间，并进行数据类型转换。在hive cli中按照region\_id顺序排序，查询dim\_customer前1条数据，将结果内容复制粘贴至对应报告中；
4. 将ods库中base\_region表数据抽取到dwd库中dim\_customer的分区表，分区字段为etldate且值与ods库的相对应表该值相等，并添加dwd\_insert\_user、dwd\_insert\_time、dwd\_modify\_user、dwd\_modify\_time四列,其中dwd\_insert\_user、dwd\_modify\_user均填写“user1”，dwd\_insert\_time、dwd\_modify\_time均填写当前操作时间，并进行数据类型转换。在hive cli中按照id顺序排序，查询dim\_customer前1条数据，将结果内容复制粘贴至对应报告中；
5. 获取hbase中ods:order\_info中的数据和ods库中order\_info表昨天的分区（任务一生成的分区）数据抽取到dwd库中fact\_order\_info的动态分区表，分区字段为etl\_date，类型为String，取create\_time值并将格式转换为yyyyMMdd，同时若operate\_time为空，则用create\_time填充，并添加dwd\_insert\_user、dwd\_insert\_time、dwd\_modify\_user、dwd\_modify\_time四列，其中dwd\_insert\_user、dwd\_modify\_user均填写“user1”，dwd\_insert\_time、dwd\_modify\_time均填写当前操作时间，并进行数据类型转换。使用hive cli执行show partitions dwd.fact\_order\_info命令，将结果截图粘贴至对应报告中；
6. 获取hbase中ods:order\_detail中的数据和ods库中order\_detail表昨天的分区（任务一中生成的分区）数据抽取到dwd库中fact\_order\_detail的动态分区表，分区字段为etl\_date，类型为String，取create\_time值并将格式转换为yyyyMMdd，并添加dwd\_insert\_user、dwd\_insert\_time、dwd\_modify\_user、dwd\_modify\_time四列，其中dwd\_insert\_user、dwd\_modify\_user均填写“user1”，dwd\_insert\_time、dwd\_modify\_time均填写当前操作时间，并进行数据类型转换。使用hive cli执行show partitions dwd.fact\_order\_detail命令，将结果截图粘贴至对应报告中。
7. 待任务6完成以后，需删除ods.user\_info中的分区，仅保留最近的三个分区。并在hive cli执行show partitions ods.user\_info命令，将结果截图粘贴至对应报告中；

### 任务二：指标计算

使用Scala编写spark工程代码，并计算相关指标。

**注：在指标计算中，不考虑订单信息表中order\_status字段的值，将所有订单视为有效订单。计算订单金额或订单总金额时只使用final\_total\_amount字段。**

1. 根据dwd层表统计每个省每月下单的数量和下单的总金额，并按照year，month，region\_id进行分组,按照total\_amount逆序排序，形成sequence值，将计算结果存入hbase的dws:province\_consumption\_day\_aggr表中（表结构如下），然后使用hive cli根据订单总数、订单总金额、省份表主键均为降序排序，再hive中查询出前5条，在查询时对于订单总金额字段将其转为bigint类型（避免用科学计数法展示），将SQL语句与执行结果截图粘贴至对应报告中;

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 中文含义 | 备注 |
| row\_key | string | Row\_key | 使用0-9随机数加上时间戳 |
| province\_id | int | 省份表主键 |  |
| province\_name | string | 省份名称 |  |
| region\_id | int | 地区主键 |  |
| region\_name | string | 地区名称 |  |
| total\_amount | double | 订单总金额 | 当月订单总金额 |
| total\_count | int | 订单总数 | 当月订单总数 |
| sequence | int | 次序 |  |
| year | int | 年 | 订单产生的年,为动态分区字段 |
| month | int | 月 | 订单产生的月,为动态分区字段 |

1. 根据dws层表来计算每个地区2020年订单金额前3省份，依次存入clickhouse数据库shtd\_result的regiontopthree表中（表结构如下），然后在Linux的clickhouse命令行中根据地区表主键升序排序，查询出前5条，将SQL语句与执行结果截图粘贴至对应报告中。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 中文含义 | 备注 |
| regionid | int | 地区表主键 |  |
| regionname | text | 地区名称 |  |
| provinceids | text | 省份表主键 | 用,分割显示前三省份的id |
| provincenames | text | 省份名称 | 用,分割显示前三省份的name |
| provinceamount | text | 省份名称 | 用,分割显示前三省份的订单金额（需要去除小数部分，使用四舍五入） |

例如：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 | 华东地区 | 21,27,11 | 上海市,江苏省,浙江省 | 100000,100,10 |

1. 根据dwd层的数据，请计算每个省份累计订单量（订单信息表一条算一个记录），然后根据每个省份订单量从高到低排列，将结果打印到控制台（使用spark中的show算子，同时需要显示列名），将执行结果截图粘贴至对应报告中。

例如：可以考虑首先生成类似的临时表A：

|  |  |
| --- | --- |
| province\_name | Amount（订单量） |
| A省 | 10122 |
| B省 | 301 |
| C省 | 2333333 |

然后生成结果类似如下：其中C省销量最高，排在第一列，A省次之，以此类推。

|  |  |  |
| --- | --- | --- |
| C省 | A省 | B省 |
| 2333333 | 10122 | 301 |

**提示：可用str\_to\_map函数减轻工作量**

## 模块E：数据可视化

系统接口

### 任务一：用条形图展示消费额最高的国家

编写Vue工程代码，根据接口，用条形图展示2020年1月消费额最高的5个国家，同时将用于图表展示的数据结构在浏览器的console中进行打印输出，将图表可视化结果和浏览器console打印结果分别截图并粘贴至对应报告中。

### 任务二：用折线图展示2021年第一季度上架下单人数的变化

编写Vue工程代码，根据接口，用折线图展示2021年第一季度上架下单人数的变化情况，同时将用于图表展示的数据结构在浏览器的console中进行打印输出，将图表可视化结果和浏览器console打印结果分别截图并粘贴至对应报告中。

### 任务三：用折柱混合图展示国家平均消费额和地区平均消费额

编写Vue工程代码，根据接口，用折柱混合图展示2020年第一季度各国家平均消费额和地区平均消费额的对比情况，柱状图展示平均消费额最高的5个国家，折线图展示这5个国家所在的地区的平均消费额变化，同时将用于图表展示的数据结构在浏览器的console中进行打印输出，将图表可视化结果和浏览器console打印结果分别截图并粘贴至对应报告中。

## 模块F：综合分析

### 任务一：Spark 数据倾斜分析

我们查看 SparkUI 会发现，大部分 task 运行速度极快，其中极小部分 task 运行速度极慢，我们称这种情况为数据倾斜，请问造成数据倾斜的原因主要是什么？怎么定位问题出现在哪段代码呢？

### 任务二：数据仓库中怎么处理缓慢变化维，有哪几种方式？

在模块 B 的数据仓库的设计过程中，会出现缓慢变化维的问题，缓慢变化维是指 一些维度表的数据不是静态的，而是会随着时间而缓慢地变化，在数仓设计中有哪些方式应对这些问题？

### 任务三：Flink 有哪些重启策略？各个重启策略如何配置？

在模块 D 中使用到了 Flink，Flink 在运行 job 时可能会出现各种问题，从而会导致其失败或者重启，对于类似于网络波动造成的运行失败可以采取相对应重启策略来重试，请问 Flink 有几种重启策略（中文）？分别怎么配置这些重启策略？

### 任务四：hive调优可以从哪几个方面进行？

在使用hive进行清洗时，我们会发现有时候hive执行的是非常缓慢，出现这种情况的原因是什么？我们应该从哪几个方面进行调整？

### 任务五：Hadoop是什么？它的特征有哪些？

在模块A中，搭建必不可少的是hadoop，Hadoop为什么是必不可少的组件？它的特点是什么？为什么hadoop能够得到广泛应用呢？在模块A中我们会发现相关的一些组件都会依赖hadoop，例如spark on yarn，hiveServer2等，这些搭建模式的优点是什么？

### 任务六:vue的生命周期及computed 和 watch 的区别

computed 和 watch 的区别和运用的场景？它们的区别是什么？在哪个生命周期内调用异步请求？