

2024 年河南省职业院校 技能大赛高职组

“大数据分析与应用” 赛项

任 务 书 4

参赛队编号：_____

大数据分析与应用赛项竞赛试题

在大数据时代，电商的经营模式经历了显着的变化。与传统运营模式相比，过去的运营往往依赖个人经验和直觉，决策过程缺乏数据支持，发展路径封闭。而如今，大数据为电商提供了全新的视角，通过大量的数据分析，商家能够做出更科学、准确的决策。借助大数据，商家能够收集和整理患者的消费行为详细数据，包括消费者的购买金额、喜欢的产品渠道、偏好的产品种类、采购周期、购买动机，以及患者的家庭、工作与生活背景环境、消费观念和价值观等多维度信息。通过对这些数据的追踪，商家可以了解顾客的来源渠道，是通过网站广告、朋友推荐链接，还是其他途径；是新访客还是真实顾客；顾客浏览过哪些产品、购买通过这些准确的数据，商家可以根据顾客的年龄、收入、兴趣等锁定标签进行分组，并针对不同的目标群体进行定位。请以 scala 作为整个项目的基础开发语言，基于大数据平台综合利用大数据组件，对数据进行处理、分析及可视化呈现。

任务一：Hadoop 完全分布式安装配置（25 分）

本任务需要使用 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、从宿主机 /opt 目录下将文件 `hadoop-3.1.3.tar.gz`、`jdk-8u212-linux-x64.tar.gz` 复制到容器 Master 中的 /opt/software 路径中(若路径不存在,则需新建),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客户端桌面【Release\任务 A 提交结果.docx】中对应的任务序号下；
- 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客户端桌面【Release\任务 A 提交结果.docx】中对应的任务序号下；

- 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 scp 复制 JDK 的命令复制并粘贴至客户端桌面【Release\任务 A 提交结果.docx】中对应的任务序号下；
- 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）粘贴至客户端桌面【Release\任务 A 提交结果.docx】中对应的任务序号下；
- 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客户端桌面【Release\任务 A 提交结果.docx】中对应的任务序号下。

任务二：离线数据处理（25 分）

子任务一：数据抽取

编写 Scala 代码，使用 Spark 将 MySQL 的 user 库中表 user_info、order_info 的数据增量抽取到 **hive** 的 ods_mysql 库（路径为/user/hive/warehouse/）的 user_info、order_info 中。

抽取 user 库中 user_info 的增量数据进入 hive 的 ods_mysql 库中表 user_info。根据 ods_mysql.user_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，同时添加分区，若 operate_time 为空，则用 create_time 填充，分区字段为 etl_date，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) 。 id 作为 primaryKey , operate_time 作为 preCombineField 。 使用 spark-shell 执行 show partitions ods_mysql.user_info 命令。

抽取 user 库中 order_info 的增量数据进入 Hive 的 ods_mysql 库中表 order_info, 根据 ods_mysql.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, 同时添加分区, 分区字段为 etl_date, 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)。id 作为 primaryKey, operate_time 作为 preCombineField。使用 spark-shell 执行 show partitions ods_mysql.order_info 命令。

子任务二：数据清洗

编写 Scala 代码, 使用 Spark 将 ods 库中相应表数据全量抽取到 **Hive** 的 dwd_ds_hive 库 (路径为 /user/hive/warehouse/dwd_ds_hive.db) 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。(若 dwd_ds_hive 库中部分表没有数据, 正常抽取即可)

- 1、抽取 ods_mysql 库中 user_info 表中昨天的分区 (子任务一生成的分区) 数据, 并结合 dim_user_info 最新分区现有的数据, 根据 id 合并数据到 dwd_ds_hive 库中 dim_user_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 operate_time 排序取最新的一条), 分区字段为 etl_date 且值与 ods_mysql 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”。若该条记录第一次进入数仓 dwd 层则 dwd_insert_time、dwd_modify_time 均存当前操作时间, 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。id 作为 primaryKey, operate_time 作为 preCombineField。使用 spark-shell 执行 show partitions dwd_ds_hive.dim_user_info 命令,将结果截图粘贴至客户端桌面【Release\任务 B 提交结果.docx】中对应的任务序号下;

- 2、将 ods_mysql 库中 order_info 表昨天的分区(子任务一生成的分区)数据抽取到 dwd_ds_hive 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。id 作为 primaryKey, operate_time 作为 preCombineField。使用 spark-shell 执行 show partitions dwd.fact_order_info 命令,将结果截图粘贴至客户端桌面【Release\任务 B 提交结果.docx】中对应的任务序号下;

任务三：数据采集与实时计算（20 分）

任务一：实时数据采集

- 1、在主节点使用 Flume 采集实时数据生成器 10050 端口的 socket 数据,将数据存入到 Kafka 的 Topic 中(Topic 名称为 order,分区数为 4),使用 Kafka 自带的消费者消费 order (Topic) 中的数据。
- 2、采用多路复用模式,Flume 接收数据注入 kafka 的同时,将数据备份到 HDFS 目录/user/test/flumebakup 下。

子任务二：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,同时计算中使用 order_info 表中 create_time 或 operate_time 取两者中值较大者作为

EventTime, 若 operate_time 为空值或无此列, 则使用 create_time 填充, 允许数据延迟 5s, 订单状态 order_status 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)。

- 1、使用 Flink 消费 Kafka 中的数据, 统计商城实时订单数量 (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数量, 其他状态则累加), 将 key 设置成 totalcount 存入 Redis 中。
- 2、在任务 1 进行的同时, 使用侧边流, 统计每分钟申请退回订单的数量, 将 key 设置成 refundcountminute 存入 Redis 中。
- 3、在任务 1 进行的同时, 使用侧边流, 计算每分钟内状态为取消订单占有所有订单的占比, 将 key 设置成 cancelrate 存入 Redis 中, value 存放取消订单的占比 (为百分比, 保留百分比后的一位小数, 四舍五入, 例如 12.1%)。

任务四：数据可视化（10 分）

子任务一：用柱状图展示各省份消费额的中位数

编写 Vue 工程代码, 根据接口, 用柱状图展示 2020 年部分省份所有订单消费额的中位数 (前 10 省份, 降序排列, 若有小数则四舍五入保留两位), 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。

子任务二：用玫瑰图展示各地区消费能力

编写 Vue 工程代码, 根据接口, 用基础南丁格尔玫瑰图展示 2020 年各地区的消费总额占比, 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。

任务五：综合分析（20 分）

子任务一：Kafka 中的数据如何保证不丢失？

在任务 D 中使用到了 Kafka，将内容编写至客户端桌面中对应的任务序号下。

子任务二：请描述 HBase 的 rowkey 设计原则。

请简要概述 HBase 的 rowkey 的重要性并说明在设计 rowkey 时应遵循哪些原则，将内容编写至客户端桌面中对应的任务序号下。