《大数据综合项目实训》练习03

一、背景

某电商平台需要对**订单基础数据**进行集中采集与处理，需要完成一次完整流程：
订单数据采集 → Kafka 缓冲 → Spark on YARN 处理 → 写入 Hive → 查询验证

日志文件：订单数据文件 `test03.log`

数据格式如下：

订单ID	用户ID	商品ID	订单金额	下单时间
O4001	U001	P3001	198	2025-02-10 09:05:12
O4002	U002	P3003	359	2025-02-10 09:18:45
O4003	U003	P3002	128	2025-02-10 09:26:30
O4004	U001	P3004	159	2025-02-10 09:40:10
O4005	U004	P3005	269	2025-02-10 10:02:18
O4006	U002	P3001	198	2025-02-10 10:15:40
O4007	U005	P3006	199	2025-02-10 10:28:55
O4008	U006	P3008	299	2025-02-10 10:45:10
O4009	U003	P3009	89	2025-02-10 11:05:36
O4010	U002	P3010	228	2025-02-10 11:22:50

二、内容

第 1 题：启动与检查环境

1.1 启动 HDFS

在 master 执行：

```
start-dfs.sh
```

1.2 启动 YARN

在 slave1 执行：

```
start-yarn.sh
```

1.3 启动 Zookeeper

在三台节点执行：

```
zkServer.sh start
```

1.4 启动 Kafka

在三台节点执行：

```
kafka-server-start.sh -daemon /opt/apps/kafka/config/server.properties
```

第 2 题：Kafka 创建 Topic 并验证

2.1 创建 Topic

在 slave1 节点创建 Kafka 主题，主题名称：`test_topic_03`

```
kafka-topics.sh \
--create \
--topic test_topic_03 \
--bootstrap-server master:9092 \
--partitions 3 \
--replication-factor 1
```

2.2 验证 Topic

```
kafka-topics.sh --list --bootstrap-server master:9092
```

2.3 启动 Kafka 消费者

在 slave1 节点执行：

```
kafka-console-consumer.sh \
--bootstrap-server master:9092,slave1:9092,slave2:9092 \
--topic test_topic_03 \
--from-beginning
```

第 3 题：Kafka 生产者发送订单数据

3.1 准备订单数据文件

将日志文件 `test03.log` 复制到 slave2 的指定目录 `/opt/data/flume/logs`：

```
mkdir -p /opt/data/flume/logs
```

3.2 在slave2将用户数据发送到 Kafka 主题

```
cat /opt/data/flume/logs/test03.log | \
kafka-console-producer.sh \
--broker-list master:9092,slave1:9092,slave2:9092 \
--topic test_topic_03
```

3.3 在slave1节点查看 Kafka 消费主题结果

第 4 题：Hive 建表

任务：创建 `test_table_03` 订单信息表

表含义说明：
订单信息表用于记录用户在电商平台中产生的订单数据，包括下单用户、商品信息、订单金额及下单时间等内容。

字段说明：

中文字段	字段名	数据类型	说明
订单ID	order_id	STRING	订单唯一标识
用户ID	user_id	STRING	下单用户ID
商品ID	product_id	STRING	订单商品ID
订单金额	order_amount	DOUBLE	订单支付金额
下单时间	order_time	STRING	订单创建时间

完成下面的 Hive 任务

4.1 在 master 启动 Hive 环境：

```
# 启动元数据服务
hive --service metastore &
# 启动Hive
hive
```

4.2 选择数据库

```
use default;
```

4.3 删除数据表

```
DROP TABLE IF EXISTS test_table_03;
```

4.4 创建订单信息表：`exam_action_table_03`

```
CREATE TABLE IF NOT EXISTS test_table_03 (
  order_id    STRING,
  user_id     STRING,
  product_id  STRING,
  order_amount DOUBLE,
  order_time  STRING
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;
```

4.5 查看数据表

```
show tables;
```

第 5 题：在 spark-shell 中完成 Kafka → Spark → Hive 数据处理任务

在 FinalShell 新建一个master节点窗口，启动 spark-shell：

```
spark-shell \
--master yarn \
--deploy-mode client \
--driver-memory 512m \
--conf spark.driver.maxResultSize=256m \
--conf spark.executor.instances=1 \
--conf spark.executor.cores=1 \
--conf spark.executor.memory=512m \
--conf spark.sql.shuffle.partitions=2 \
--conf spark.sql.catalogImplementation=hive \
--conf spark.driver.extraClassPath=/opt/apps/hive/conf \
--conf spark.executor.extraClassPath=/opt/apps/hive/conf \
--packages org.apache.spark:spark-sql-kafka-0-10_2.12:3.1.1
```

5.1 从 Kafka 读取订单数据

```
val kafkaDF = (
    spark.read
  .format("kafka")
  .option("kafka.bootstrap.servers", "master:9092")
  .option("subscribe", "test_topic_03")
  .load()
)
```

5.2 提取 Kafka 中的 value 字段

```
val lines = kafkaDF.selectExpr("CAST(value AS STRING) AS line")
```

5.3 解析订单 CSV 数据（5 列）

```
val dataDF = lines.selectExpr(
  "split(line, ',')[0] AS order_id",
  "split(line, ',')[1] AS user_id",
  "split(line, ',')[2] AS product_id",
  "cast(split(line, ',')[3] AS double) AS order_amount",
  "split(line, ',')[4] AS order_time"
)
```

5.4 切换数据库并查看表

```
spark.sql("USE default")
spark.sql("show tables").show(false)
```

5.5 将 DataFrame 写入 Hive 表

```
dataDF.coalesce(1).write.mode("append").insertInto("test_table_03")
```

5.6 查询 Hive 表前 5 行数据，验证写入结果

```
spark.sql("SELECT * FROM test_table_03 LIMIT 5").show(false)
```