《大数据综合项目实训》练习04

一、业务背景

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

日志文件：支付数据文件 `test04.log`

数据格式如下：

支付ID	订单ID	支付方式	支付金额	支付时间
PAY5001	O4001	数字人民币	198	2025-02-11 09:06:30
PAY5002	O4002	信用卡	359	2025-02-11 09:20:10
PAY5003	O4003	储蓄卡	128	2025-02-11 09:28:45
PAY5004	O4004	信用卡	159	2025-02-11 09:45:30
PAY5005	O4005	数字人民币	269	2025-02-11 10:05:18
PAY5006	O4006	储蓄卡	198	2025-02-11 10:18:40
PAY5007	O4007	信用卡	199	2025-02-11 10:30:55
PAY5008	O4008	数字人民币	299	2025-02-11 10:48:20
PAY5009	O4009	储蓄卡	89	2025-02-11 11:08:36
PAY5010	O4010	信用卡	228	2025-02-11 11:25:50

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_04`

```
kafka-topics.sh \
--create \
--topic test_topic_04 \
--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_04 \
--from-beginning
```

第 3 题：Kafka 生产者发送支付数据

3.1 准备支付数据文件

将日志文件 `test04.log` 复制到 slave2 的指定目录`/opt/data/flume/logs`：

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

3.2 在slave2将支付数据发送到 Kafka 主题

```
cat /opt/data/flume/logs/test04.log | \
kafka-console-producer.sh \
--broker-list master:9092,slave1:9092,slave2:9092 \
--topic test_topic_04
```

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_04` 支付信息表

表含义说明：
支付信息表用于记录订单的支付明细数据，包括支付方式、支付金额及支付时间等内容，用于后续对支付情况进行查询与统计分析。

字段说明：

中文字段	字段名	数据类型	说明
支付ID	payment_id	STRING	支付记录唯一标识
订单ID	order_id	STRING	关联订单ID
支付方式	pay_type	STRING	微信支付 / 支付宝 / 银联支付
支付金额	pay_amount	DOUBLE	实际支付金额
支付时间	pay_time	STRING	支付完成时间

完成下面的 Hive 任务

4.1 在 master 启动 Hive 环境：

```
hive --service metastore &
hive
```

4.2 选择数据库

```
use default;
```

4.3 删除数据表

```
DROP TABLE IF EXISTS test_table_04;
```

4.4 创建支付信息表：`test_table_04`

```
CREATE TABLE IF NOT EXISTS test_table_04 (
  payment_id   STRING,
  order_id     STRING,
  pay_method   STRING,
  pay_amount   DOUBLE,
  pay_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 读取支付数据

```
val kafkaDF = (
    spark.read
  .format("kafka")
  .option("kafka.bootstrap.servers", "master:9092")
  .option("subscribe", "test_topic_04")
  .load()
)
```

5.2 提取 Kafka 中的 value 字段

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

5.3 解析支付 CSV 数据（5 列）

```
val dataDF = lines.selectExpr(
  "split(line, ',')[0] AS payment_id",
  "split(line, ',')[1] AS order_id",
  "split(line, ',')[2] AS pay_method",
  "cast(split(line, ',')[3] AS double) AS pay_amount",
  "split(line, ',')[4] AS pay_time"
)
```

5.4 切换数据库并查看表

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

上传截图05：要求能看到支付表

5.5 将 DataFrame 写入 Hive 表

```
dataDF.coalesce(1).write.mode("append").insertInto("test_table_04")
```

5.6 查询 Hive 表前 5 行数据，验证写入结果

```
spark.sql("SELECT * FROM test_table_04 LIMIT 5").show(false)
```
