设计思路

- 使用FlinkKafkaConsumer010获取kafka队列中的数据
- 自定义MysqlWriter类实现OutputFormat接口，对于每一个流入flink的数据进行处理
- 对于获取的数据在flink中使用JSONObject转化为结构化的BuyInfo，根据destination区分不同的数据，然后collect起来，交给MysqlWriter处理
- MysqlWriter获取数据后利用jdbc根据数据destination的不同将数据存放到不同的表

测试入口

- job4\src\main\java\com\bigdata\week3\job4\Kafka目录下的Consumer_java，@Test下执行run函数

核心代码

Consumer_java

```
@SpringBootTest
public class Consumer_java {
    MySqlWriter mySqlWriter=new MySqlWriter();
    private static String accessKey = "17E8AFD1271D6CF443EA";
    private static String secretKey =
"Wzc5NTVBQkEzRUE4MzlGM0RFNzQ4MkNCNjBDMDIy";
    private static String endpoint = "http://scut.depts.bingosoft.net:29997";
    private static String bucket = "chenzhuokun";
    private static String key = "demo.txt";
    private static String topic = "data_flka_chenzhuokun_job3";
    private static int period = 5000;
    private static String bootstrapServers =
"bigdata35.depts.bingosoft.net:29035,bigdata36.depts.bingosoft.net:29036,bigdata
37.depts.bingosoft.net:29037";
    private static String[] filtersDestination = {"德阳市", "湛江市", "佛山市", "乌
鲁木齐市", "沈阳市", "北京市"};
    @Test
    public  void run() throws Exception {
        StreamExecutionEnvironment env =
StreamExecutionEnvironment.getExecutionEnvironment();
        Properties properties = new Properties();
        properties.setProperty("bootstrap.servers", bootstrapServers);
        properties.setProperty("acks", "all");
        properties.setProperty("key.serializer",
"org.apache.kafka.common.serialization.StringSerializer");
        properties.setProperty("value.serializer",
"org.apache.kafka.common.serialization.StringSerializer");
        properties.setProperty("group.id", "flink-group");
        FlinkKafkaConsumer010<String> consumer = new
FlinkKafkaConsumer010<String>(topic, new SimpleStringSchema(), properties);
        consumer.setCommitOffsetsOnCheckpoints(true);
        DataStream<String> stream = env.addSource(consumer);
        List<DataStream<String>> streams = new ArrayList<>();
        for (String city : filtersDestination) {
            DataStream<String> tempStream = stream
                    .filter(new FilterFunction<String>() {
                        @Override
                        public boolean filter(String s) throws Exception {
```

```
                            return BuyInfo.getDesfronString(s).equals(city);
                        }
                    })
                    .flatMap((String line, Collector<String> collector) -> {
                        //System.out.println("get data with desitination " +
city + ": " + line);

collector.collect(BuyInfo.getDesfronString(line)+"@"+line);
                    })
                    .returns(Types.STRING);
            tempStream.writeUsingOutputFormat(mySqlWriter);
            streams.add(tempStream);
        }
        DataStream<String> tempStream = stream
                .filter(new FilterFunction<String>() {
                    @Override
                    public boolean filter(String s) throws Exception {
                        return
Arrays.asList(filtersDestination).indexOf(BuyInfo.getDesfronString(s)) == -1;
                    }
                })
                .flatMap((String line, Collector<String> collector) -> {
                    //System.out.println("get data with other destination : " +
line);
                    collector.collect(BuyInfo.getDesfronString(line)+"@"+line);
                })
                .returns(Types.STRING);
        tempStream.writeUsingOutputFormat(mySqlWriter);
        env.execute("kafka streaming word count");
    }
}
```

MySqlWriter

```
public class MySqlWriter implements OutputFormat<String> {
    Statement statement;
    @Override
    public void configure(Configuration configuration) {
        Connection connection= null;
        try {
            connection =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bigdata","caesar","1643
75");
        } catch (SQLException throwables) {
            throwables.printStackTrace();
        }
        try {
            statement=connection.createStatement();
        } catch (SQLException throwables) {
            throwables.printStackTrace();
        }
    }
    @Override
    public void open(int i, int i1) throws IOException {

    }
    @Override
```

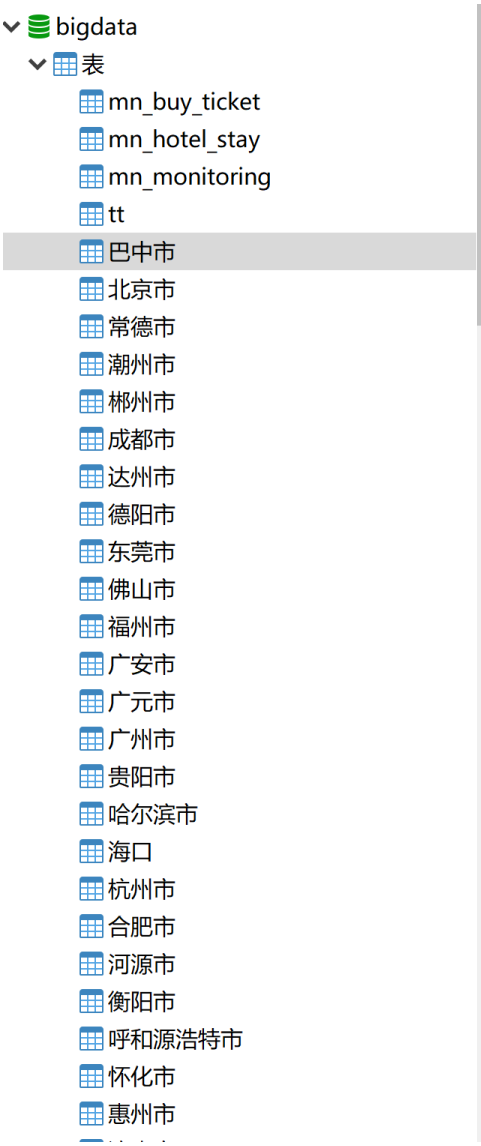
```
    public void writeRecord(String s) throws IOException {
        BuyInfo buyInfo= JSONObject.parseObject(s.split("@")[1],BuyInfo.class);
        try {
            Class.forName("com.mysql.jdbc.Driver");
            String sql="CREATE TABLE if not exists "+s.split("@")[0]+" (\n" +
                    "  `username` varchar(50) DEFAULT NULL COMMENT '姓名',\n" +
                    "  `buy_time` datetime DEFAULT NULL COMMENT '购票时间',\n" +
                    "  `buy_address` varchar(500) DEFAULT NULL COMMENT '购票地
址',\n" +
                    "  `origin` varchar(100) DEFAULT NULL COMMENT '出发地',\n" +
                    "  `destination` varchar(100) DEFAULT NULL COMMENT '目的地'\n"
+
                    ") ENGINE=InnoDB DEFAULT CHARSET=utf8;";
            statement.execute(sql);
            SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
            String time="str_to_date('"+sdf.format(buyInfo.getBuy_time())+"',
'%Y-%m-%d %H:%i:%s')";
            sql="insert into "+s.split("@")[0]+"
(username,buy_time,buy_address,origin,destination)
values('"+buyInfo.getUsername()+"',"+time+",'"+buyInfo.getBuy_address()+"','"+bu
yInfo.getOrigin()+"','"+buyInfo.getDestination()+"');";
            statement.execute(sql);
            System.out.println(sql);
        } catch (ClassNotFoundException | SQLException e) {
            e.printStackTrace();
        }

        //System.out.println("数据"+buyInfo+"已经存入表"+s.split("@")[0]+"中");
    }
    @Override
    public void close() throws IOException {

    }
}
```

最终效果

```
insert into 银川市 (username,buy_time,buy_address,origin,destination) values('
insert into 遂宁市 (username,buy_time,buy_address,origin,destination) values('
insert into 哈尔滨市 (username,buy_time,buy_address,origin,destination) values
insert into 银川市 (username,buy_time,buy_address,origin,destination) values('
insert into 攀枝花市 (username,buy_time,buy_address,origin,destination) values
insert into 南充市 (username,buy_time,buy_address,origin,destination) values('
insert into 娄底市 (username,buy_time,buy_address,origin,destination) values('
insert into 达州市 (username,buy_time,buy_address,origin,destination) values('
insert into 岳阳市 (username,buy_time,buy_address,origin,destination) values('
insert into 珠海市 (username,buy_time,buy_address,origin,destination) values('
insert into 永州市 (username,buy_time,buy_address,origin,destination) values('
insert into 杭州市 (username,buy_time,buy_address,origin,destination) values('
insert into 天津市 (username,buy_time,buy_address,origin,destination) values('
insert into 湛江市 (username,buy_time,buy_address,origin,destination) values('
insert into 武汉市 (username,buy_time,buy_address,origin,destination) values('
insert into 太原市 (username,buy_time,buy_address,origin,destination) values('
insert into 南宁市 (username,buy_time,buy_address,origin,destination) values('
insert into 常德市 (username,buy_time,buy_address,origin,destination) values('
insert into 云浮市 (username,buy_time,buy_address,origin,destination) values('
insert into 沈阳市 (username,buy_time,buy_address,origin,destination) values('
insert into 重庆市 (username,buy_time,buy_address,origin,destination) values('
insert into 内江市 (username,buy_time,buy_address,origin,destination) values('
insert into 怀化市 (username,buy_time,buy_address,origin,destination) values('
insert into 南京市 (username,buy_time,buy_address,origin,destination) values('
insert into 南京市 (username,buy_time,buy_address,origin,destination) values('
insert into 湛江市 (username,buy_time,buy_address,origin,destination) values('
insert into 泸州市 (username,buy_time,buy_address,origin,destination) values('
```



秦欣汝	2019-11-0	乌鲁木齐市	乌鲁木齐市	巴中市
潘伟	2019-11-0	资阳市	天河资阳市	巴中市
方佳怡	2019-10-1	攀枝花市	环攀枝花市	巴中市
鲁淳美	2019-07-0	湘潭市	小北湘潭市	巴中市
倪贺祥	2019-09-0	西安市	东湖西安市	巴中市
鲁淳美	2019-07-0	湘潭市	小北湘潭市	巴中市
秦欣汝	2019-11-0	乌鲁木齐市	乌鲁木齐市	巴中市
秦欣汝	2019-11-0	乌鲁木齐市	乌鲁木齐市	巴中市
潘伟	2019-11-0	资阳市	天河资阳市	巴中市
方佳怡	2019-10-1	攀枝花市	环攀枝花市	巴中市
鲁淳美	2019-07-0	湘潭市	小北湘潭市	巴中市
倪贺祥	2019-09-0	西安市	东湖西安市	巴中市
秦欣汝	2019-11-0	乌鲁木齐市	乌鲁木齐市	巴中市
潘伟	2019-11-0	资阳市	天河资阳市	巴中市
方佳怡	2019-10-1	攀枝花市	环攀枝花市	巴中市
鲁淳美	2019-07-0	湘潭市	小北湘潭市	巴中市
秦欣汝	2019-11-0	乌鲁木齐市	乌鲁木齐市	巴中市
潘伟	2019-11-0	资阳市	天河资阳市	巴中市
方佳怡	2019-10-1	攀枝花市	环攀枝花市	巴中市
鲁淳美	2019-07-0	湘潭市	小北湘潭市	巴中市
倪贺祥	2019-09-0	西安市	东湖西安市	巴中市
赵晨茜	2019-03-2	拉萨市	环市拉萨市	巴中市
孙涵涵	2019-09-1	合肥市	东华合肥市	巴中市
韩益辰	2019-12-1	武汉市	东风武汉市	巴中市
傅淼	2019-04-1	杭州市	流花杭州市	巴中市
李禹辰	2019-02-1	株洲市	流花株洲市	巴中市
赵雅晗	2019-09-0	眉山市	先烈眉山市	巴中市
李国栋	2019-01-1	内江市	环市内江市	巴中市
韩润莎	2019-02-0	湛江市	东湖湛江市	巴中市
钱雄霖	2019-11-2	韶关市	江南韶关市	巴中市