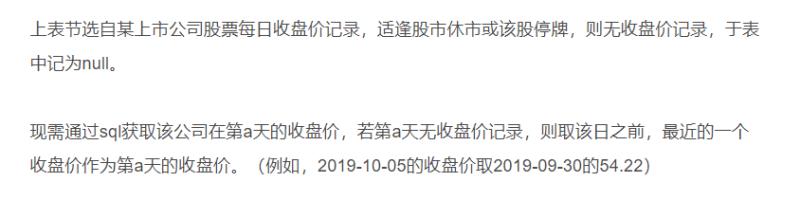
上海校区大数据企业面试真题

版本：V1.1

# 米哈游

1. 什么是Flink的非barrier对齐，如何实现？
2. flink的内存管理？
3. flink的序列化机制？
4. flink提交job的方式以及参数如何设置？ 页面提交和客户端提交有什么区别？
5. 你们flink集群规模？
6. flink提交作业的流程，以及与yarn是如何交互的？
7. flink的checkpoint机制以及精准一次性消费如何实现？
8. flink的状态是什么，分为几种？
9. SparkContext里面主要做了哪些工作？
10. ConcurrentHashMap的底层实现原理？
11. 什么是Watermark及主要作用？
12. flink是如何管理kakfa的offset，使用什么类型的状态保存offset？



# 美团外包

1. 实时方面
   1. 主攻哪个方向
   2. 我说的实时
   3. 具体介绍一下具体做了哪些工作.
   4. 为什么要做sparkstreaming到Flink的转化.
   5. 在什么场景下需要这么高的实时性.
   6. 既然是开窗为什么一定要转FLink.
   7. 遇到SparkStreaming不太能解决的问题.(我说的是手动维护Kafka的offset实现一致性消费的问题)
   8. 必需要手动维护offset吗?(我转到了Flink去解决这个问题)
   9. 遇到Flink不太能解决的问题.(我没多说,其实应该说大数据量使用redis布隆过滤器实现UV去重的)
   10. 实时指标出来后的应用场景.(我提到了ES预警)
   11. 预警是怎么做到的.预警的条件.(公司内部的预警和用户行为的预警)
2. 数仓方面:
   1. 当初建模的时候应用场景是什么样的.
   2. 建模的流程(我是从数据源开始讲)
   3. 都有哪些数据同步到数仓里面(我说大概20多张表)
   4. 对这些表有过什么分类吗.(我说的同步策略)
   5. 哪些表是相应的同步策略.
   6. 跨天支付数据是怎么处理的.
   7. 用户表为什么一定是拉链表.
   8. 数仓的分层.
   9. 如何找出来用户的一天的行为轨迹.(这个问题出现在描述dws层)
   10. dws和dwt的宽表都有哪些,并且都是什么!!!(详细都说出来!!!)
   11. 出口对应的指标.(我太菜了.没说出来几个)
   12. 你们是怎么保证数据质量的.(居然没想起来监控框架.我太菜了.)
   13. 数据质量监控的角度,(我只记得数据量)
   14. 有制定一些量化的'含量指标'(应该是这个词)也就是说怎么衡量这个表好用不好用.
   15. 有没有一张表实现所有的分析.
3. SQL题(很简单)
   1. 外卖的配送ID
   2. 外卖员的ID
   3. 订单配送的City
   4. 时间的CT
   5. 一整年中,每个月每个城市订单量Top10.
4. 其他
   1. 3.1.为什么考虑换一份工作.
   2. 3.2.离线和实时更偏向哪些(我说的实时),为什么?
   3. 3.3.工作后做的最有成就感的一件事是什么.
5. flink，ck机制，内存管理，出现反压怎么处理的？
6. kylin如何直接构建cube？

# 华为

1. spark内存管理
2. hive分区表中，单值分区和范围分区的区别
3. 你们公司执行spark任务时，资源怎么设置的（需要直接说出来）
4. 介绍一下kafka水位线（其实就是leo和Hw）
5. 说几个指标，分别从什么数据层拿取了数据，需要直接说出来
6. 数仓采用了什么模型？为什么？
7. hive分区表，单值分区和范围分区的区别
8. spark任务切分，怎么判断有没有执行shuffle
9. 你们公司拉链表都有什么字段，拉链表出错怎么办
10. 列举几张表的同步策略
11. flink Sql 了解吗

# 四、吉贝克

1. 你做了哪些项目
2. 你主要负责哪个项目
3. 你怎么建模的，你们的数仓架构（你们用到了哪些框架），你们的数据量，你们的集群规模，用的是Apache还是CDH，怎么维护集群的（集群监控）
4. 你们用的hql，还是sparksql？
5. hive中主要用到了哪些函数
6. 你们数仓遇到的问题
7. 你们用的脚本还是jar
8. 你们azkaban的版本
9. 你们遇到过hive与mysql间的字符集乱码问题吗？怎么解决的
10. 你们hive数据倾斜遇到过吗？怎么解决的
11. 你在离线数仓中做了什么

# 五、美团到店

1. 自我介绍
2. 具体介绍一下具体做了哪些工作.
3. 为什么要做sparkstreaming到Flink的转化.
4. Sparkstreaming和Flink消耗资源具体数据对比
5. 在什么场景下需要这么高的实时性.
6. 遇到SparkStreaming不太能解决的问题
7. 建模的流程
8. 数仓的分层.
9. 宽表都有哪些,？
10. 三范式知道吗，说一下？
11. 项目中遇到什么难解决的问题？
12. 有小文件和数据倾斜，这个怎么处理？
13. 空值key加随机数是一种数据倾斜解决方案，如果有单个key是热点值呢？又如果有多个key是热点值呢？用参数和代码分别怎么解决？
14. 调度工具用到哪些；
15. 数据可视化怎么做；
16. Flink怎么优化？举实际例子，数据对比
17. OLAP引擎用过哪些？
18. 用过什么工具进行数据迁移，导入导出。
19. 行存和和列存的区别？
20. OLTP和OLAP的区别
21. Flink的JobManger？
22. Flink的TaskManager
23. 为什么选择ElasticSearc，ClickHouse?
24. Spark Streaming和Flink的区别，包括计算实时指标的一个逻辑是怎样的？
25. 假设有些数据，延时了10分钟20分钟才过来，想这种数据在Spark Streaming和Flink分别做怎么处理的？
26. 算从0点累计到当前时间的DAU（日活），像这种数据，用Flink如何实现？
27. 布隆过滤器有什么缺点，哪些场景用不了？
28. 你们离线数仓是跑在什么引擎上的？
29. MapReduce从提交到最后执行大概是一个什么过程？
    1. Shuffle和Reduce有什么区别？
30. 一个任务，平常10分钟20分钟就完成了，今天1,2个小时都没完成，我们需要怎么解决？
31. 算过去30天有哪些用户是连续7天登录我们APP的，如何写SQL，思路？
32. 开窗函数有哪些？
33. 开窗函数什么情况下会有order by，什么情况下order by是必须要写的？
34. 数据报表存储这块用过哪些产品，用过哪些存储引擎？--没答上来，后来提醒的我说的HBase
35. OLAP引擎用过哪些？
36. 如何设计数据报表的存储，MySQL已经不能用了，查询效率太低，你们这时候如何存储？
37. 拉链表有什么缺点？拉链表有哪些字段必须要有的？
38. 数据和业务是怎么协作的？比如说数据对业务做一些反馈和支持？

# 六、润和

1. 负责几个topic
2. 每个topic有多少个分区
3. hbase列族有多少个
4. 看你Kafka写的挺多的，kafka装了几台，把你知道的关于kafka的全部说一遍
5. 我把生产者到broker到消费者给他说了一通
6. spark解决数据倾斜的手段
7. spark做了哪些优化
8. 项目中使用redis实现什么功能
9. sqoop导出一致性问题，通过什么参数增加map个数
10. hbase的组件
11. 实际中rowkey怎么设计的
12. HBASE的读写流程，如果数据已经写到了WAL还没写到MemStore挂机了，会怎么处理，有什么影响 （hbase是有事务吗求解）

# 七、声网

1. 怎么修改正在运行的Flink程序？如果有新的实时指标你们是怎么上线的？
2. 使用flink统计订单表的GMV，如果mysql中的数据出现错误，之后在mysql中做数据的修改操作，那么flink程序如何保证GMV的正确性，你们是如何解决？
3. 开发人员和测试人员如何保证SQL的正确性？假如这条sql就是写错误了，那么用这条SQL统计mysql中的数据，肯定也是无法发现错我，你们是如何解决？
4. 如何区分事实表和维度表？有度量值就一定是事实表吗？什么是描述性/修饰性维度？
5. 用sqoop在00：10分将mysql中的数据导入到HDFS，对于新增及变化的数据，由于存在窗口期，比如每天在00:05分的时候这条数据都修改，那么就会一直无法拉取到这条数据，怎么解决？
6. ETL清洗的规则是不超过1/10000，你们是怎么发现超过万分之一的？

# 八、新潮传媒集团

1. 熟悉哪些组建
2. rdd.df.ds区别
3. flink与spark区别
4. Java写99乘法表（电脑直接敲）计时
5. 字符串，首字母大写（电脑敲）
6. 手写hql（体现实力）
7. 离线架构分层
8. kafka为什么快
9. hdfs了解多少
10. hbase部分内容
11. 问Java和Scala谁更熟的时候，大家一定要说Scala，说了Java会被问死，我说的Scala，就拿我没辙了
12. flink窗口5种
13. flink时间语意
14. 整体数据走向，分开来说
15. 有没有hive相关的API，我说我没做过，不清楚
16. Java中GC原理
17. Java中无符号状态如何有效处理
18. 还有很多简单一点的问题，面试太久了，我没录音，主要看第11点，我给大家的建议，除非单独Java开发过的

# 九、卓钢链

1. flink怎么提交
2. flink提交有多少jobmanger
3. flink与spark区别
4. flink反压
5. 5flink监控，如何有效处理数据积压
6. 离线那些很简单，随便说下海哥整理的那一套就行了
7. 最后说是不是培训，我果断否定，flink火了别的公司用，我们也能用，我们就是半年前换了flink

# 十、鸭嘴兽网络科技有限公司

1. flink窗口
2. spark手动提交offset
3. flink有什么问题
4. flink反压，如何解决
5. 为什么flink替换spark
6. spark优化
7. flink优化
8. 在flink项目中做了什么9.flink开发哪个窗口用的最多

# 十一、齐数科技

1. canal传输数据这么保证不丢失
2. flink配合redis以及布隆过滤器具体怎么实现大数据量的去重
3. flume你们公司允许丢多少数据，说个范围区间
4. spark shuffle讲一下
5. azkaban任务调度怎么使用，让我说一下yml语法，懵逼
6. canal到kafka到sparkstreaming怎么精准一次消费
7. spark的checkpoint怎么使用的，你之前公司有没有用过
8. 问我es的端口号，紧张忘了，我反问他canal端口号，我还提示说双十一相关，他也不会[666]
9. sparkstreaming怎么消费kafka（就是2种方式）
   1. 做过数据治理吗？有这种场景c表依赖于b表，b表依赖于a表，如果a表数据出错，就比如说a表是javaEE的订单表，javaEE修改了里面的字段，比如100万GMV，就可能出现只统计10万GMV，你们怎么处理？（答：javaEE改表为什么不跟我们说，谁给他们勇气的）
   2. 你们哪些表使用拉链表？拉链表建立分区表了吗？怎么建立的？（答：我们用户表建立了分区表，拉链表与分区表之间没有关系，拉链表可以建立分区表，但是我们没有建立，比如说可以按照省份进行建立分区表）

# 十二、喜马拉雅

1. Hive中七种join.每种区别
2. Hive的炸裂函数
3. Hive项目中遇到过哪些问题，具体业务中怎么解决的(要讲的很详细)
4. kafka的延迟队列的时间轮算法(不知道他当时问的是不是这个，我讲的这个)。
5. 建模的原因
6. 分层的原因
7. 命名表的时候的特殊性(比如有个表用户特别关注，如果是你怎么命名)，还有各种表的同步策略
8. 另外还有两道sql，当她的面写，一道侧写炸开，一道连续，还不算太难，但是不能写错一点，不然她会觉得写的不是特别好。由这个炸裂她会引出业务中的炸裂
9. 问为什么自己定义udtf而不是直接用炸裂

# 十三、斗象

1. flink checkpoint的实现原理
2. spark checkpoint的实现原理
3. jvm如何调优
4. rdd的作用主要是干什么的
5. hive和hbase的区别
6. flink开窗五分钟过来一亿条数据你是怎么处理的
7. flink开窗5分钟被同一用户连续访问60次，需要把他的访问信息调出来 你是怎么做的
8. spark有1000个分区 他们的数据是怎么交互的
9. spark 有10万条数据 你将这些数据怎么分配到集群的

# 十四、金大师

## 学长一

1. java基本数据类型
2. 讲一下hashmap
3. hashmap为什么用到红黑树
4. 链表的时间复杂度
5. 红黑树的时间复杂度
6. ArrayList与linkedList的区别，其中链表是什么链表
7. MySQL的事务隔离级别
8. 数据库的索引，索引结构是什么？
9. ①聊一聊kafka，②zookeeper中存储的kafka中的信息的格式，③ack，④副本个数，⑤ISR，⑥kafka的存储在哪，⑦kafka的读写流程，⑧分区个数
10. HBASE的读写流程，如果数据已经写到了WAL还没写到MemStore挂机了，会怎么处理，有什么影响
11. 说一下布隆过滤器怎么实现的，数据结构是什么
12. 业务中HBASE的RowKey怎么设计的
13. watermark处理迟到数据，怎么实现的
14. redis的数据类型，怎么用来去重的，存储的是什么数据
15. 说一下slot，业务中一个TaskManager设置几个slot，连接的kafka的分区数是多少

## 学长二

1. hashMap和hashSet底层实现原理
2. stringbuffer和stringbuilder的区别
3. spark如何保证精准一次消费
4. hive的两个表join的工作机制
5. kafka的精准一次消费 幂等性+事务 kafka版本 事务如何实现的
6. 事务的分类
7. 如何实现多线程，线程怎么关闭
8. 你知道什么二叉树
9. 红黑树的结构
10. es如何实现更新数据 可以更新部分属性吗

# 十五、齐数科技

1. canal传输数据这么保证不丢失
2. flink配合redis以及布隆过滤器具体怎么实现大数据量的去重
3. flume你们公司允许丢多少数据，说个范围区间
4. spark shuffle讲一下
5. azkaban任务调度怎么使用，让我说一下yml语法，懵逼
6. canal到kafka到sparkstreaming怎么精准一次消费
7. spark的checkpoint怎么使用的，你之前公司有没有用过
8. 问我es的端口号，紧张忘了，我反问他canal端口号，我还提示说双十一相关，他也不会
9. sparkstreaming怎么消费kafka（就是2种方式）

# 十六、海致星图

1. 双流join,left join,左流数据先来，右流一直没来，左流会这么样（1.5版本之后就又flink sql了，1.11多了hive）
2. 左流数据已经输出到sink了，此时右流数据来了，可以join又会这么样
3. flink故障恢复，说我概念记得挺牢的（我没理他）
4. Savapoint了解多少
5. 作业挂掉了，恢复上一个Checkpoint，用什么命令
6. 为什么用yarn-session
7. 说一下状态编程
8. 使用Mapstage，group by id 如何设计
9. 继续上面的Mapstage，id不放在key行不行
10. 数据积压问题
11. Kafka数据很多，内存很少，读取数据都是问题，现在想要写，怎么控制写速率（上面都是flink）
12. Spark哪一块用的多，实时，spark streaming用的是结构流还是什么，后面说到df
13. df与ds区别，课上讲的没够用
14. Map与mapPartition，我说完说我让我看源码
15. Task与 partition 有什么关系
16. Stage,宽依赖
17. Kafka一直说
18. 一个topic有3个分区，两个消费者，会怎么样
19. 一个topic有2个分区，三个消费者，会怎么样
20. Kafka怎么处理大量数据（为什么这么快），说我概念背的好，但是细节不是很注意
21. Hdfs小文件处理，spark处理小文件
22. arraylist在 Scala有什么可以做到同样功能，比较像的
23. Hbase,redis,es选一个，我选hbase,又谈到凤凰，凤凰和hbase这么放一起的，rowkey
24. 最后闲聊，数仓分层

# 十七、赢时胜

## 学长一

1. 自我介绍
2. 详细的讲一下实时项目.
3. 为什么要用Flink替代SparkStreaming(应该深入的去讲一下Flink)
4. 你们公司都处理过什么业务.
5. 公司的类型.公司的电商的服务还有吗?地址?(应该是想问网站的网址?).大数据分析对外部提供服务是在哪个web或者app(直接告诉他公司内部使用!)
6. 公司都卖些什么?买的最好的商品?
7. 分模块对指标进行分析.(具体分哪些模块)
8. 这个项目里面日常都会做哪些.
9. 指标中印象最深的就是什么?
10. 网络波动导致的支付先到了怎么办?
11. process用的种类.
12. 10个int以数组的形式保存,那么保存在什么状态好?VlaueState还是ListState?存在哪个的性能比较好?
13. 广告在没有人点击的(也就是没有数据流的时候)窗口,这个窗口存在吗?有没有对这些窗口进行校验的窗口.
14. 1小时的滚动窗口,一小时处理一次的压力比较大,想让他5分钟处理一次.怎么办?(自定义触发器)
15. 数仓项目处理的业务?
16. dws都有什么维度和和字段?
17. 维表的数据量扩大十倍会有什么问题?
18. 维度的地区增大(比如上海划分成具体的某些区),也就是改变维表的粒度会出现问题吗?

## 学长二

1. yarn 资源队列
2. Hbase region 自动切分为什么不好
3. yarn session 和 pre job 区别
4. hive join 发生数据倾斜如何解决
5. 只让讲了一下采集部分，后面的是提问
6. spark sql 和 hive sql 区别
7. rdd 和 spark sql区别
8. 出了到flink实际场景的题 类似咱的Uv那个指标 ，如何解决乱序数据，后来又加了个
9. 迟到数据怎么办
10. flume 如何分渠道，就是那个mutil方式 ，记得说出拦截器加头信息
11. hashmap

# 十八、花旗

1. Java多线程
2. 多线程的创建方式
3. Java线程池
4. 德鲁伊连接池的特点，如果我连接突然断了，会发生什么
5. Java锁，怎么加锁，用过见过哪些锁，加锁有哪些影响
6. Java数据结构，hashmap 和arraylist
7. spark client cluster
8. spark shuffle
9. spark rdd
10. mr底成和spark stage的区别，mr也是有stage的？mr stage是什么
11. spark提交参数
12. 怎么开启压缩
13. 压缩的效率有多高
14. lzo压缩以后，传输量提高了多少，把具体值说一下
15. hive表优化
16. hive各种参数
17. hive去重
18. spark数据倾斜

# 十九、微盟

## 学长一

1. groupby和count（distinct）的底层机制和区别是什么
2. Spark和flink的双流join的底层原理
3. sparkstream统计每天营销额的时候，系统崩溃后，如何处理已经聚合后的数据，数据保存在哪里（这里我解释了幂等性和事物和ack-1，面试官揪着这里问了很久，他关心的事出了问题之后怎么解决）
4. 各种表怎么导入的，sqoop倒导表的详细步骤，累积型快照事实表，拉链表，现场写代码展示等
5. 数仓里面建的各种表，都建了哪些表，数仓每层之间同事都会有数据进行导入导出和计算，如何保证每层计算间有序状态不干涉
6. 精确一次，至多一次，至少一次对checkpoint有什么影响
7. flink里面异步IO代码具体怎么写的，每一步具体描述出来
8. 都用实时做了哪些任务
9. 都是根据你的简历，然后给你场景，让你分析，不然你介绍项目，他很抠细节，面试官年龄不大，2,30岁的样子。

## 学长二

1. 讲一下kafka中的各个组件？
2. 讲一下kafka中的分区？
3. isr的作用？
4. 数据在kafka中是怎么被处理的？
5. habse的架构？
6. 怎么获取hbase的数据？
7. 怎么设置redis中的过期时间以及hbase中怎么设置？
8. kafka中的ack级别？
9. hbase中wal的作用？以及怎么写数据？
10. 怎么设计rowkey
11. 单例模式（手敲）
12. kafka的leader挂了怎么办？

# 二十、贝壳找房

1. 公司是否有做生命周期管理
2. 为什么要做生命周期管理
3. 为什么使用parquet列式存储？为什么不用别的？
4. orc,rc,parquet列式存储有什么区别，底层存储的内存是否是连续的？
5. 为什么orc有索引就一定快？
6. 我答了orc的构成，他随后问到的
7. hive的优化
8. 说提前使用combinehiveinputformat，那么具体是怎么实现的？这个inputformat是什么东西？有几种格式？
9. 你刚刚说开启数据倾斜时负载均衡，那么具体是怎么实现的？不能只说个大概，要说用mr是怎么实现的
10. 什么是维度建模，为什么要维度建模
11. 为什么要维度退化，维度退化有什么好处？
12. kylin的构建算法
13. 光说个概念不行，会问你逐层构建每轮mr做了什么，要讲清楚，否则会一直问，我说不会，他还是追问了下。
14. 拉链表 也问的很细

面试官问的很细，都是离线的，而且每涉及到一个知识点，都会问你底层用mr是怎么实现的。不能只回答表面，会一直追问。

# 二十一、宝尊电商

1. spring 生命周期
2. 线程生命周期
3. flink的数据量
4. azkb几个工程
5. azkb的任务挂了怎么办的
6. flink双流join数据延迟怎么解决？？？？？
7. hive调优

# 二十二、食哼

1. flume的故障转移.使用的是filechenal，新的flume替换掉老的flume.（使用同一个共享组，然后注释掉原flume的source.之后新的flume无缝对接）
2. sparkstreaming数据量级别.开窗1小时.和开窗1分钟系统能不能撑得住.
3. 数据质量的问题.怎么样判断各个阶段数据量是对的.比如采集判断采集的数据是对的.判断的标准是什么.
4. foreach中向kafka中发送数据.解决方案.
5. 懒加载lazy.
6. hdfs的危害.
7. datenode的迁移.
8. 需求6小时内的wc.如果6小时后没有到来则清空之前的数据.
9. ss能承载的数据量和什么有关.
10. spark处理大数据场景怎么办?
11. sparkStreaming中处理线程安全问题.
12. MR的一些优化
13. sparkStreaming处理速度太大怎么办?
14. Hive的并行度由什么决定?
15. sparkStreaming怎么处理延时数据?
16. Kylin的构建cube与什么有关?如果数据量大的话对集群有什么影响吗?

# 二十三、字节跳动

1. 算法题，排列组合

示例：[1,2,3]

期望：[1,2,3],[1,3,2],[2,1,3],[2,3,1],[3,1,2],[3,2,1]

1. kafka如果创建大量的topic，对kafak会有什么影响？
2. kafka一个topic有3个partition，但是出现了数据倾斜，有2个partition中数据量很少，下游用flink消费kakfa，5秒钟一个窗口会有什么影响？
3. 谈谈你对spark Shuffle的理解？ 描述一下大表与大表join时的Shuffle机制或者过程？
4. 谈一下flink如何保证精准一次性？
5. barrier对齐会有什么危害？
6. 你在项目中都遇到过哪些问题？怎么解决的？有没有转化为经验，分享给其他同事，保证这个错误不会再犯？
7. 平时用用java代码多吗？知道线程池吗？你可以手写出一个线程池吗？

# 二十三、滴滴

1. redis底层map实现的原理
2. 为什么从SparkStreaming转移到Flink？就是因为flink功能强大吗？
3. SparkStreaming的checkpoint与flink的checkpoint的区别？
4. 4.5亿用户如何海量去重，我说使用redis自定义布隆过滤？他说这么高的并发，redis抗的住吗？
5. 你懂HBase吗？那么lsm树是什么？
6. 网络传输分几层
7. TCP三次握手
8. redis为key设置过期时间的底层原理是什么？

# 二十四、上海睿民

1. Hive的任务时会不会有任务的卡顿，无法完成？问数据倾斜问题？
2. 很奇怪的问题就是代码里面有没有发生什么问题造成卡顿的？
3. count distinct / group by 他是想问这个
4. 业务方面说出数据倾斜的场景
5. 有没有什么东西可以让你缩小错误的范围？如何去缩小的？
6. 程序流程图问我有没有看过，内存怎么样

# 二十五、软通

1. 介绍一下你们公司的数仓分层。
2. 印象最深的指标，或者最难的。
3. 使用的是hive还是sparkSql
4. 常用的算子
5. 常用的行动算子。
6. foreach 和foreachPartitions 的区别
7. reduce 和 reduceBykey的区别
8. reduce 是行动算子还是转换算子
9. 广播变量（结合公司业务）
10. 你们公司的数据量（实时）
11. executor 的内存这么设置
12. 你们公司用了多大内存（他应该是指你的一个executor设置了多大内存）
13. 消费kafka的两种模式
14. 你们用了多少个executor

# 二十六、信也科技

## 学长一

1. Flink CEP实现了那些需求？用那些算子具体每一步怎么实现
2. HBase的架构，数据热点怎么解决，介绍下一Phoenix的协处理器机制，
3. flink各个模式，job提交流程
4. 单链表
5. flink遇到哪些问题？怎么解决的
6. 为什么选择flink？
7. flink优化？具体说几个参数优化
8. Java对象创建的几种方式？
9. ES数据库选择得什么引擎？

## 学长二

1. canal搭建时候client和server装一起还是分开？
2. canal从MySQL拉过来数据什么格式
3. flink的重启策略

# 二十七、海尔集团

1. hive都进行了哪些优化
2. 你们数仓是怎么分层的
3. 如何建模
4. 数据怎么清洗的
5. Shell中有哪些变量
6. 写个shell脚本（启/停 群发的不行）
7. 有没有哪些sql经过你的优化之后效率有大幅的提高，请详述一下内容以及前后性能对比
8. hive的小文件的处理

# 二十八、中软

## 学长一

（国际外包太平洋）

1. flink sql怎么定义的，
2. rdd与dataset的转化，
3. 双流join，数据延迟来了怎么办，
4. 你具体负责哪些业务 ，
5. 你们的业务数据加了一张表，然后再用canal倒入的时候要做什么？
6. java和scala的区别

（华腾）

1. 做了哪些业务
2. hive和mysql的区别

## 学长二

1. hive优化
2. Sparkstreaming精准一致性消费
3. Map和mappartition
4. Hbase只能用phoneix建立索引吗？（我那知道）
5. Spark背压机制
6. Spark中使用那些算子？（我说map，flatmap的时候他说你还用这个，我人傻了）
7. Spark1.0和spark2.0初始化sparkcontext有什么不同？
8. 项目中遇到那些问题？
9. Canal的作用
10. 知道datax吗？

## 学长三

（中软外包平安医疗）

1. Hive行列过滤是什么意思

答：列处理：在SELECT中，只拿需要的列，如果有，尽量使用分区过滤，少用SELECT \*。

行处理：在分区剪裁中，当使用外关联时，如果将副表的过滤条件写在Where后面，那么就会先全表关联，之后再过滤。

1. 列式存储区别

答：常用的包括：textFile，rcFile，ORC，Parquet，一般企业里使用ORC或者Parquet，因为是列式存储，且压缩比非常高，所以相比于textFile，查询速度快，占用硬盘空间少

1. ORC与Parquet有什么区别

答：1.orc不支持嵌套结构(但可通过复杂数据类型如map<k,v>间接实现)，parquet支持嵌套结构。

2.orc与hive的兼容性强，作为hive的常用存储格式。  
3.orc相比parquet的存储压缩率较高。  
4.orc导入数据和数据查询的的速度比parquet快。

1. sqoop导出的事务

答：–staging-table方式（建立临时表，通过sqoop导入到临时表，成功之后再把临时表的数据通过事务导入到mysql的业务数据表）

1. 导入HDFS中map任务挂了怎么办
2. dws层和dwt的区别（怎么实现的，dwt是否分区）,分别是怎么做的
3. DWS层统计各个主题对象的当天行为，服务于DWT层的主题宽表
4. Dwt不分区
5. 在dwd层这些表是怎么处理的 （新增及变化表：优惠券领用表 用户表 订单表）
6. 在离线中，手动修改了数据
7. 数仓中是怎么做ETL的
8. sqoop导入到hdfs（使用几个map）又没有遇到数据不一致（比如说map失败了） 我我回答直接删掉重新导一次
9. 新增及变化是怎么区分开的（what）
10. 新增表、新增及变化表处理的方式有什么不一样的地方（怎么处理）
11. 数仓中分了哪些主题

会员

设备

商品

地区

活动

1. 如何保证数仓中数据的准确性（hive在处理数仓的过程，好多逻辑，在处理过程中会不会出现误差，是怎么处理的）
2. 数仓怎么去掉错误数据（异常），在Mysql中将数据写错了（原本是1000，人为的改成了1001），错误数据应该怎么处理
3. 使用sqoop导入Hdfs,但是业务库物理删除了一点（删掉了一条不需要的订单），相邻层之间进行监控
4. cannel可以识别delete(删除语句)语句
5. 什么情况下使用拉链表

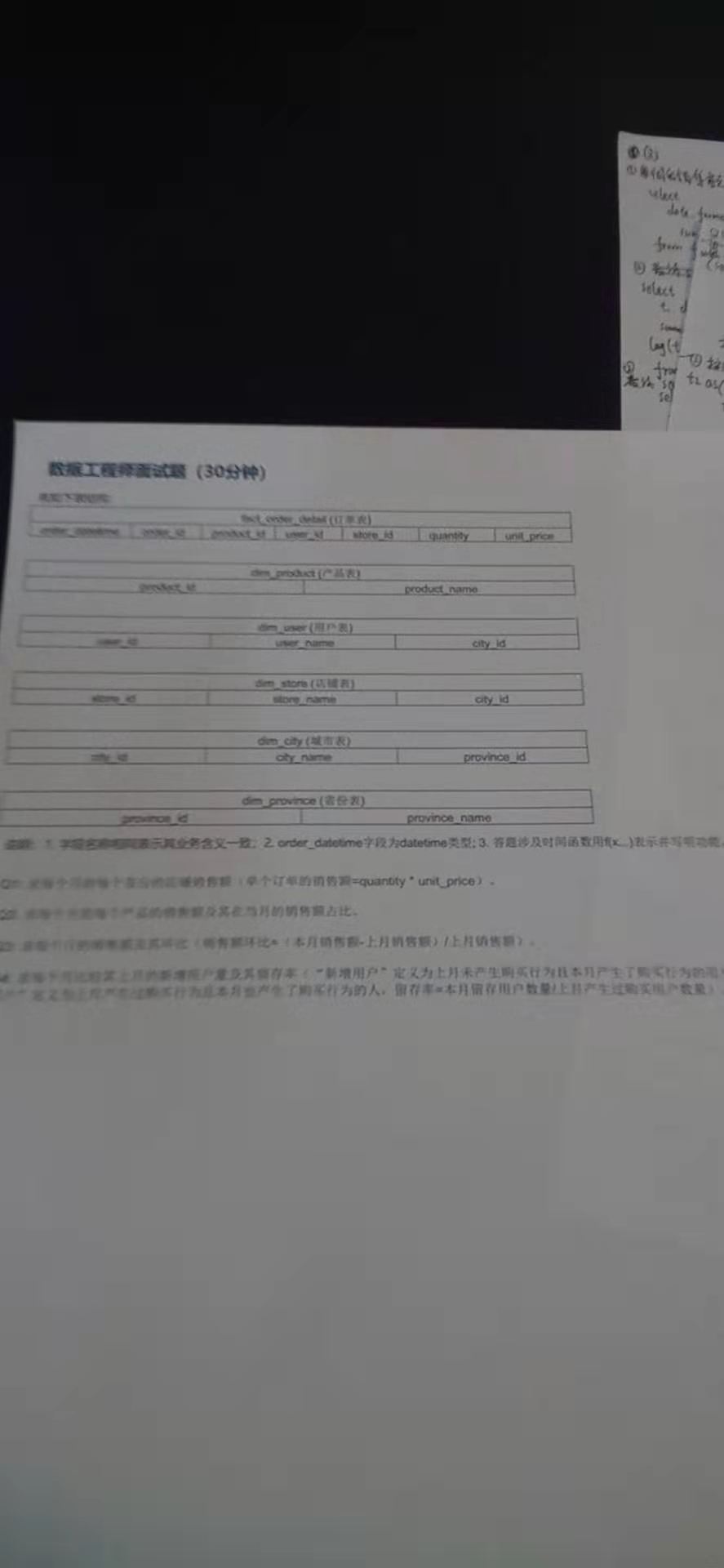
# 二十九、博彦科技

1. orderby 和sort by
2. having的作用
3. hive中插入数据的方式
4. spark和MR的区别
5. 在spark中union会经过shuffle吗？
6. 三范式的第三个

# 三十、驰骛信息科技有限公司

## 学长一

1. 用户行为数据有那些字段
2. 如何指定hql输出的内容到一个文件（我答MR写入到指定reduce，他问我hql怎么实现，spark怎么实现，我说不知道，请教他，他也没说个啥）
3. 如何判断计算是否正确？
4. 维度建模（我答维度建模的4步，他说这是比较简单的，然后说了一堆深奥的说根据相应的业务过程，没有表格啥的，就知道怎么建模，神经兮兮的）



1. 这不就是尚硅谷的数仓原题吗？

## 学长二

1. hive中udf函数中的方法？
2. a表往b表中写数据，如何避免小文件？

答：设置reduce的个数

1. 如何做数据的校验在hive中？
2. 还有就是业务的问题？有哪些表？表的数据量？那些维度表事实表？怎么的同步策略？业务表和事实表是否有交叉？

# 三十一、精锐教育集团

## 学长一

1. 介绍一下经历，离线数仓中分层怎么分
2. kafka怎么消费不丢失数据？
3. spark优化做过哪些？
4. spark处理任务来不及怎么办？（背压，参数记得吗？）
5. 公司的数据量大小？
6. spark的双流join和flink的双流join
7. 怎么使用布隆过滤器的？拓展（在布隆过滤器前提下，要知道具体是谁活跃过？设计bitmap的具体使用）
8. kylin怎么用的？
9. ES的写入过快问题怎么办？（他说要涉及到异步写入的场景，我没见过）
10. 场景题：实时场景的是什么什么率怎么做？分子分组都开做
11. 分组topn问题，手写sql
12. HBase协处理器了解吗？我只回答了phoenix使用二级索引的时候？

## 学长二

1. 自我介绍
2. mysql索引的底层用的什么实现的 b+树
3. mysql索引的最左匹配原则是什么
4. zookeeper选举机制，选举过程中某一台挂了会怎么选举
5. 你知道哪些二叉树 讲讲红黑树结构
6. 当场出了一道sql 很简单的
7. kafka分区 副本数
8. 你有什么想问的吗

# 三十二、奇利匙

1. 数仓从采集到数仓，每层干了什么？
2. 可视化对接的哪里？是hive吗？（肯定不是，是在mysql里面）
3. 你们dws层详细说一下他具体分析每天的什么东西？
4. kylin用过吗？说一下 （cube的聚合，剪枝）
5. flume优化做了哪些，kafka优化做了哪些，都是你做的吗？（废话，都是我）
6. 你们dwd维度建模工作是怎么分配的？按照业务线去分配

# 三十三、蚂蚁

1. Hbase如何读取数据？
2. （其实想问的是怎么设计rowkey获取数据）
3. 离线数仓分层
4. flink的精准一次性消费
5. flink的一个流的数据错了怎么处理？
6. 有哪些业务线？
7. 为什么用spark？

# 三十四、普洛斯

1. orc与parquet的区别？
2. 查询性能方面orc要高。
3. kakfa中的一条业务数据流？（从数据开始到结束）
4. window与processWindow中的process什么时候触发？

# 三十五、七牛云

1. 实时数仓的架构？
2. kakfa中ack级别？为什么0不用
3. kafka中isr的作用？
4. kafka中rebalance？

答：

* 1. 触发Rebalance的时机：
  2. Rebalance的触发条件有三个：
  3. （1）组员个数发生变化。例如有新的consumer实例加入该消费组或者离开组
  4. （2）订阅的Topic个数发生变化
  5. （3）订阅Topic的分区数发生变化
  6. 消费组成员正常的添加和停掉导致rebalance，这种情况无法避免，但是在某些情况下，consumer实例
  7. 会被coordinator错误的认为已停止从而被踢出group。从而导致rebalance。
  8. 参数解决：
  9. （1）当 Consumer Group 完成 Rebalance 之后，每个 Consumer 实例都会定期地向 Coordinator 发送心跳请求，表明它还存活着。
  10. 如果某个 Consumer 实例不能及时地发送这些心跳请求，Coordinator 就会认为该 Consumer 已经 “死” 了，从而将其从 Group 中移除，然后开启新一轮 Rebalance。
  11. 这个时间可以通过Consumer 端的参数 session.timeout.ms进行配置。默认值是 10 秒。
  12. （2）除了这个参数，Consumer 还提供了一个控制发送心跳请求频率的参数，就是 heartbeat.interval.ms。这个值设置得越小，Consumer 实例发送心跳请求的频率就越高。
  13. 频繁地发送心跳请求会额外消耗带宽资源，但好处是能够更加快速地知晓当前是否开启 Rebalance，因为，目前 Coordinator 通知各个 Consumer 实例开启 Rebalance 的方法，
  14. 就是将 REBALANCE\_NEEDED 标志封装进心跳请求的响应体中。
  15. （3）除了以上两个参数，Consumer 端还有一个参数，用于控制 Consumer 实际消费能力对 Rebalance 的影响，即 max.poll.interval.ms 参数。
  16. 它限定了 Consumer 端应用程序两次调用 poll 方法的最大时间间隔。它的默认值是 5 分钟，表示你的 Consumer 程序如果在 5 分钟之内无法消费完
  17. poll 方法返回的消息，那么 Consumer 会主动发起 “离开组” 的请求，Coordinator 也会开启新一轮 Rebalance。

1. hdfs中的小文件？
2. hdfs的优缺点？它支持高频率的读取数据吗？

答：

* 1. 优点：高容错性、适合批处理、适合大数据处理、流式数据访问（一次写入，多次读取，不能修改，只能追加）
  2. 缺点：不适合低延时数据访问、无法高效的对大量小文件进行存储、
  3. 并发写入、文件随机修改（一个文件只能有一个写，不允许多个线程同时写，仅支持数据sppend，不支持文件的随机修改

1. kakfa的高效读写数据？零复制零拷贝（解释清楚）
2. hdfs的架构？dn、nn、2nn
3. ES的倒排索引？

答：

* 1. 切词==>倒排索引（倒排表，查询相对应的id获取数据）
  2. 倒排表以字或词为关键字进行索引，表中关键字所对应的记录表项记录了出现这个字或词的所有文档，一个表项就是一个字表段，它记录该文档的ID和字符在该文档中出现的位置情况。
  3. 由于每个字或词对应的文档数量在动态变化，所以倒排表的建立和维护都较为复杂，但是在查询的时候由于可以一次得到查询关键字所对应的所有文档，所以效率高于正排表。

1. 快排、堆排？
2. GC
3. 环形缓冲区为什么设置成环形的？

答：

* 1. 环形缓冲区的特性：
  2. 当一个数据元素被用掉后，其余数据元素不需要移动其存储位置。相反，一个非圆形缓冲区（例如一个普通的队列）在用掉一个数据元素后，其余数据元素需要向前搬移。
  3. 换句话说，圆形缓冲区适合实现FIFO先进先出缓冲区，而非圆形缓冲区适合后进先出缓冲区。
  4. 适合于事先明确了缓冲区的最大容量的情形。扩展一个圆形缓冲区的容量，需要搬移其中的数据。因此一个缓冲区如果需要经常调整其容量，用链表实现更为合适

# 三十六、阳光午餐

1. 你们公司的是自营店还是平台
2. 你们公司的app有网页版本的吗
3. spark streaming 你会那些
4. 说一下kafka你会那些东西
5. watermark说一下
6. java自带的连接池是什么
7. 多线程说一下
8. 反射说一下。
9. 双流join说一下，对比flink的双流join
10. 你使用scala还是java多一些

# 三十七、好未来

1. 如何删除外部表 包括原始数据
2. 建模过程
3. 如何解决用实时系统分析一下 前几个月的数据（不会啊）
4. sqoop 同步策略
5. 拉链表如何实现的-> 缓慢变化维的数据还有其他方式吗
6. 建模过程DWD
7. hive 跟换引擎为Spark 运行的区别
8. 实时架构，spark精准一次性消费(我主动讲的)
9. 主题如何划分的
10. hive组成

# 三十八、韵达

1. 用过哪些调度框架，kylin刷入数据到hbase时，用的什么调度工具
2. flink的分区分配策略
3. flink各种窗口的区别
4. 时间语义
5. apply和process区别
6. flume采用什么类型组件
7. flume支持scv格式的数据么
8. 如果不想等窗口关闭才看结果，该怎么做

# 三十九、深圳市领星网络科技有限公司

笔试：

一、Redis

1. Redis工作原理，使用场景是什么？
2. 你在项目中Redis的存储有哪些？
3. Redis支持的最大数据量是多少？Redis集群下怎么从某一台集群查key-value
4. 列举一个常用的Redis客户端的并发模型
5. Redis，传统数据库，hbase，hive每个之间的区别
6. Redis的性能瓶颈在哪里
7. Redis支持的数据格式
8. 如何使用Redis高并发可以支持10万Qps+

二、Spark

1. sparksql介绍下（rdd dataframe）
2. udf和udaf都写过哪些
3. 介绍下udaf
4. spark提交流程
5. spark调优思路
6. 宽窄依赖是什么？区别是什么？
7. spark on yarn 和MapReduce中yarn有什么区别
8. spark支持的分布式部署方式

三、hbase

* 1. hbase最主要的特点是什么？
  2. 简单描述hbase的rowkey的设计原则
  3. 请描述hbase中scan和get的功能以及实现的异同
  4. 请描述如何处理hbase中region太多和region太大带来的冲突
  5. hbase的rowkey怎么创建比较好？列族怎么创建比较好
  6. hbase过滤器实现原则
  7. hbase宕机如何处理
  8. hbase怎么预分区
  9. 请减速hbase中compact用途是什么，什么时候触发，分哪两种compact，有何区别，有哪些相关配置参数
  10. 关系型数据库是怎么把数据导出到hbase里的
  11. 你们用hbase存储什么数据
  12. hbase如何实现模糊查询

四、kafka

1. kafka中zk起到什么作用，可以不用zookeeper么
2. kafka是如何做到高吞吐量的，请分别从读写两个方面介绍一些
3. kafka的消息持久性是如何实现
4. kafka数据是如何存储
5. kafka消息数据一致性是如何保障的
6. kafka的message格式是什么样的
7. kafka中消费者组是什么概念
8. kafka中的消息是否丢失和重复消费

五、flink

1. 怎么提交实时任务，有多少job manager
2. 怎么做压力测试和监控
3. 为什么用flink
4. checkpoint存在哪里
5. 如果下级存储不支持事务，flink怎么保证exctly-once
6. 说一下flink状态机制
7. flink中的Windows出现了数据倾斜，你有什么解决办法
8. flink在使用聚合函数groupby distinct keyby 等函数是出现数据热点该如何解决

# 四十、明略科技

1. hive静态分区和动态分区区别
2. 怎么建表的？命名规则？
3. 哪些用hiveSQL、哪些用sparkSQL？
4. 小表join大表怎么实现？优化？
5. Spark DataFrame的复用？？
6. Spark持久化策略？
7. 内部表和外部表？关键字区别？
8. Java的常用集合？哪些list？
9. Java常用设计模式？
10. 数据结构？哪些map？
11. hashMap?

# 四十一、平安银行（中国平安）

1. es读写
2. Hbase读写
3. Flink并行度设置
4. Redis悲观锁和乐观锁,又问你知道mysql的悲观和乐观锁不
5. 熟不熟悉Flink算子，像Flatmap,Map怎么用

# 四十二、天阳科技

1. 你们kafka数据量怎么这么大
2. 你们的kafka设多少topic？

答：多少个日志类型就多少个Topic。也有对日志类型进行合并的。 错误日志、启动日志、曝光日志、页面日志、动作日志 如果下一级都是离线，可以从一个topic取 如果一个业务部门只处理一种日志，那么就需要独立设topic 数据量大 =》 一张表一个topic

# 四十三、车轮互联

1. hdfs调优你都做过哪些
2. yarn调优
3. 求每小时的活跃用户，BI报表已经生成的情况下延迟数据怎么处理

# 四十四、叮咚买菜

1. 简单介绍下框架，手画框架，注意说法，挺懂的，可能随时提问
2. 实时和离线集群是搭建一起还是分开，占比
3. Hive如何实现去重
4. Azkaban版本，有没有二次编译过
5. HashMap
6. 有没有用springboot，编写代码流程
7. 用的框架版本（Hadoop，Saprk）
8. 快排，和归并的区别

# 四十五、上海小砖块网络科技有限公司

## 学长一

1. HBase中rowkey怎么找数据的，连接zookeeper后怎么找到元数据表
2. 二级索引除了phoenix外还有什么方法创建
3. spark报错序列化类找不到为什么
4. spark的shuffle
5. rowkey分几段，用个实际场景说明，如何查一个用户一个月的，一个月所以用户的
6. 用的什么调度框架，里面的sql语句如何触发
7. hbase中存储的数据量多大
8. 自己员工用app有优惠吗
9. 大数据有多少人，怎么分配，整个大数据组多少人，你们的gmv多少
10. 为什么离职，你认为怎样才算更好的发展

## 学长二

1. spark的shuffle流程？
2. hbase的rowkey设计？
3. 二级索引创建方法有哪些？

二级索引创建参考https://blog.csdn.net/weixin\_43892898/article/details/89249322

1. 整个数仓分层脚本如何调度执行的？
2. 脚本挂了怎么办？
3. 数仓分层之间数据处理通过什么技术实现？
4. 元数据管理是用的什么？
5. Hbase中get获取数据时是怎么个流程？

# 其他

1. 你们这边数据量有多少呢，就是从ods到ads层一共多少数据量？
2. 拉链表如果有一天的数据错了，比如说到12月15号，但是发现11月10号拉链的数据错了，导致后续拉链的结果都错了，这个应该怎么修正？
3. 真实项目中，spark的core和task数量该如何设置？

task数量，至少设置成和spark application的cpu core总数量一样（但这是最理想的情况，400 cpu core，分配了400个task，同时跑，不可能差不多同时跑完）

spark官方推荐：task数量，设置为spark application的cpu core总数量的2~3倍。怎么解释呢？因为如果400 cpu core，分配了400个task，同时跑，20个先跑完了，380个还在运行，这时，就有20个cpu core 空闲出来了，就导致了浪费。那如果task数量设置为cpu core总数的2-3倍，那么一个task跑完后，另一个task立马补上来，这就避免了cpu core空闲，提高spark作业速度。

1. 如何设置并行度？

SparkConf sparkConf = new SparkConf().set(“spark.default.parallelism”, “800”);

1. java的集合了解多少?
2. scala的集合了解多少?
3. scala集合里面是怎么扩容的?
4. scala常见的算子?
5. scala中reduce by和group by有什么区别?
6. reduce by和group by 返回什么类型的值,传的是什么参数?
7. foreache和coleasce什么区别?
8. 并发读取mysql需要关注那几个参数?(sparkcontext去读去连接)
9. spark共享变量?
10. 项目中怎么用的背压机制?
11. Flink的key By和Spark的 group by有什么区别?
12. Flink的watermark什么时候去触发计算?
13. 消息超过watermark的时间会丢失数据吗?
14. Kafka数据挤压怎么办?
15. Kafka的分区分配策略?
16. 为什么不用rabitMQ要用kafka?
17. mysql中的数据是全部导入还是增量导入到hdfs?
18. java的集合了解多少?
19. scala的集合了解多少?
20. scala集合里面是怎么扩容的?
21. scala常见的算子?
22. scala中reduce by和group by有什么区别?
23. reduce by和group by 返回什么类型的值,传的是什么参数?
24. foreache和coleasce什么区别?
25. 并发读取mysql需要关注那几个参数?(sparkcontext去读去连接)
26. spark共享变量?
27. 项目中怎么用的背压机制?
28. Flink的key By和Spark的 group by有什么区别?
29. Flink的watermark什么时候去触发计算?
30. 消息超过watermark的时间会丢失数据吗?
31. Kafka数据挤压怎么办?
32. Kafka的分区分配策略?
33. 为什么不用rabitMQ要用kafka?
34. mysql中的数据是全部导入还是增量导入到hdfs?
35. 一个任务，平常10分钟20分钟就完成了，今天1,2个小时都没完成，我们需要怎么解决？
36. 算过去30天有哪些用户是连续7天登录我们APP的，如何写SQL，思路？
37. 开窗函数有哪些？
38. 开窗函数什么情况下会有order by，什么情况下order by是必须要写的？
39. 数据报表存储这块用过哪些产品，用过哪些存储引擎？--没答上来，后来提醒的我说的HBase
40. OLAP引擎用过哪些？
41. 如何设计数据报表的存储，MySQL已经不能用了，查询效率太低，你们这时候如何存储？
42. 拉链表有什么缺点？拉链表有哪些字段必须要有的？
43. 数据和业务是怎么协作的？比如说数据对业务做一些反馈和支持？
44. HBASE的读写流程，如果数据已经写到了WAL还没写到MemStore挂机了，会怎么处理，有什么影响
45. 说一下布隆过滤器怎么实现的，数据结构是什么
46. 业务中HBASE的RowKey怎么设计的
47. watermark处理迟到数据，怎么实现的
48. redis的数据类型，怎么用来去重的，存储的是什么数据
49. 说一下slot，业务中一个TaskManager设置几个slot，连接的kafka的分区数是多少

# 精·Flink面试总结

## Flink提交

1. flink怎么提交
2. flink集群规模？flink的数据量？在flink项目中做了什么？
3. flink提交作业的流程，以及与yarn是如何交互的？
4. yarn-session与Per Job优缺点
5. flink提交job的方式以及参数如何设置？ 页面提交和客户端提交有什么区别？
6. Flink的JobManger，提交有多少jobmanger
7. Flink的TaskManager
8. 说一下slot，业务中一个TaskManager设置几个slot，连接的kafka的分区数是多少
9. 怎么修改正在运行的Flink程序？如果有新的实时指标你们是怎么上线的？

## 二．状态编程

1. 说一下状态编程（operator state，keyed state）
2. flink的状态是什么，分为几种？
3. 10个int以数组的形式保存,保存在什么状态好?VlaueState还是ListState?存在哪个的性能比较好?
4. 使用MapStage，group by id 如何设计
5. 继续上面的MapStage，id不放在key行不行
6. flink是如何管理kafka的offset，使用什么类型的状态保存offset？
7. 一个窗口，现在只取第一帧和最后一帧，怎么做？

## 反压（背压，数据积压）

<https://blog.csdn.net/weixin_49060400/article/details/112253767>

1. flink用什么监控，如何有效处理数据积压
2. 遇到Flink不太能解决的问题.(PV,UV放内存，OOM了，后面配合redis以及布隆过滤器)
3. 使用flink统计订单表的GMV，如果mysql中的数据出现错误，之后在mysql中做数据的修改操作，那么flink程序如何保证GMV的正确性，你们是如何解决？

## 四．Spark与Flink对比

1. Spark与Flink区别
2. Flink的key By和Spark的 group by有什么区别?
3. spark有哪些优化
4. Flink怎么优化
5. 遇到SparkStreaming不太能解决的问题.(我说的是手动维护Kafka的offset实现一致性消费的问题)
6. 必需要手动维护offset吗?(转到了Flink去解决这个问题)
7. Sparkstreaming和Flink消耗资源具体数据对比
8. 为什么要用Flink替代SparkStreaming(应该深入的去讲一下Flink)
9. 在什么场景下需要这么高的实时性

## 五．Checkpoint

1. flink checkpoint的实现原理（容错机制，故障恢复，分布式快照，checkpoint，）
2. flink的checkpoint机制以及精准一次性消费如何实现？
3. 精确一次，至多一次，至少一次对checkpoint有什么影响
4. Savapoint了解多少
5. 作业挂掉了，恢复上一个Checkpoint，用什么命令
6. 什么是Flink的非barrier对齐，如何实现？（1.11版本）

<https://blog.csdn.net/nazeniwaresakini/article/details/107954076>

总结：异步快照

## 六．窗口与Watermark

1. flink时间语义
2. 什么是Watermark及主要作用？什么时候去触发计算?
3. 消息超过watermark的时间会丢失数据吗?（允许迟到，侧输出）
4. 开窗函数有哪些？（五种）
5. flink开发哪个窗口用的最多（最好随手举一个例子表面怎么用的）
6. 既然是开窗为什么一定要转Flink（说时间语义）
7. 广告在没有人点击的(也就是没有数据流的时候)窗口,这个窗口存在吗?有没有对这些窗口进行校验的窗口.
8. 1小时的滚动窗口,一小时处理一次的压力比较大,想让他5分钟处理一次.怎么办?(自定义触发器)
9. flink开窗五分钟过来一亿条数据你是怎么处理的
10. flink开窗5分钟被同一用户连续访问60次，需要把他的访问信息调出来 你是怎么做的

## 七．双流join

1. Spark和flink的双流join的底层原理
2. A表left join B表
3. A表数据来了，B没来
4. A表数据来了，B在规定时间内到
5. A表数据来了，B在规定时间后面到(此处规定时间，就可以很好的利用起来说一下两种算子优缺点)

这个问题，process中两种算子（connect，join）分别说明，Flink SQL（撤回流）可以写两种风格，种类很多，需要细细品

## 八．杂七杂八

1. process用的种类(8个，最好中文名都记一下，不需要都掌握，可以把最熟悉的在项目在项目中怎么用说一下)
2. flink的内存管理？

[<https://blog.csdn.net/u012151684/article/details/109439590>](https://blog.csdn.net/u012151684/article/details/109439590)

1. flink的序列化机制？

<https://blog.csdn.net/xiaopeigen/article/details/108318530>

关键字：TypeInformation、TypeInfo、TypeInfoFactory、kryo

1. Kafka数据很多，内存很少，读取数据都是问题，现在想要写，怎么控制写速率
2. Rich Functions与Functions区别
3. flink里面异步IO代码具体怎么写的，每一步具体描述出来

# 精·花旗面试题

（据说有了这个可以直接入职）

## Java部分

### 1.==/equals/Hash code

==比较的是内存地址，equals比较的是值

1. 对于基本数据类型:只有==,没有equals.
2. 对于字符串:==比较的是内存地址，equals比较的是值（因为equals重写了equals方法）
3. 对于对象:==比较的是内存地址，equals比较的是两个对象值是否相等，如果没有被重写，比较的是对象的引用地址是否相同；

hashCode是jdk根据对象的地址或者字符串或者数字**算出来的int类型的数值**。对象放入集合中时，**先判断hashcode是否相等，再判断equals是否相等，都相等就算是同一个对象**，list则可以放入，set因为不允许重复所以不会放入

### 2.Reflection反射

class.forName("MyClass") 会导致类加载，只执行静态代码块

(MyClass)class.newInstance() 加载类, 然后生成实例对象

### 3.IOC

IOC: inversion of Control 控制反转，是一种设计思想，指导我们如何设计出松耦合、更优良的程序。Ioc意味着**将你设计好的对象交给容器控制**，而不是传统的在你的对象内部直接控制

### 4.DI(依赖注入)

DI—Dependency Injection，即“依赖注入”**即由容器动态的将某个依赖关系注入到组件之中**

### 5.AOP

面向切面编程（Aspect Oriented Programming，AOP）其实就是一种关注点分离的技术。比如我们写业务逻辑代码的同时，还要写事务管理、缓存、日志等等通用化的功能。简单地说，就是将那些于业务无关，却为业务模块所共同调用的逻辑封装起来。

AOP技术的实现是建立在JAVA语言的反射机制与动态代理机制之上的（不要答，问到在说 ）。

### 6.overload/override

overload是重载：用于在一个对象中实现多个相同名称的方法，方法的参数不同

override是覆盖：是子类中实现一个和父类方法名，参数完全相同的方法, 子类的实例调用方法时调用的是子类的方法. private修饰的方法不可以重写

int和Integer区别

int 是基本数据类型，Integer包装类，可以new对象，契合java中万物皆对象的思想，int的默认值是0，Integer的默认值是NULL。

### 7.PreparedStatement/Statement

Statement每次执行sql语句，相关数据库都要执行sql语句的编译。对于**只执行一次的SQL语句**选择Statement是最好的。

因为PreparedStatement对象的开销比Statement大，对于一次性操作并不会带来额外的好处。相反,如果**SQL语句被多次执行**选用PreparedStatement是最好的.PreparedStatement的第一次执行消耗是很高的. 它的性能体现在后面的重复执行.

### 8.Select\*

我们应当避免select\*操作，应该采用行列过滤的方式（where），只选取对我们有用的字段

### 9.Merge(mysql)

1.Merge(MRG\_MyISAM)存储引擎类型允许你把许多**结构相同的表合并为一个表**。当从合并表中执行查询，从多个表返回的结果就像从一个表返回的结果一样。 2.要创建合并表的前提是每一个合并的表必须有同样的表定义（表结构、索引），并且**子表的存储引擎必须是Myisam**

对数据重复分割，直至子数组大小为1，然后原路执行合并操作。合并的时候比较大小。谁小谁在前，谁大谁在后，合并成一个新的数组 (MergeSort)

### 10.In/exists

mysql中，in和exists的区别

select \* from user where exists (select 1);

**exists的条件就像一个bool条件**,exists对外表用loop逐条查询，exists (select 1)就像条件，为true就返回外表该条数据，false则不返回。

select \* from user where user\_id = 1 or user\_id = 2 or user\_id = 3; (in查询的子条件返回结果必须**只有一个字段**), 而exists就没有这个限制

**in查询相当于多个or条件的叠加**, in查询就是**先将子查询条件的记录全都查出来**，假设结果集为B，共有m条记录，然后再将子查询条件的结果集分解成m个，再进行m次查询。

### 11.Union/Union all

纵向拼接查询结果，union去重，union all不去重

### 12.各种索引Index

MySQL 官方对索引的定义为：索引（Index）是帮助 MySQL 高效获取数据的数据结构。提取句子主干，就可以得到索引的本质：索引是数据结构。

* 聚簇索引，它并不是一种单独的索引类型，而是一种数据存储方式，数据行即索引，索引即数据
* 非聚簇索引，又称为辅助索引或二级索引。 InnoDB 的二级索引 data 域存储的是相应记录主键的值而不是物理位置的指针。

Unique index 唯一索引，不能重复。

index(普通索引可以重复) 比如说，因为人有可能同名，所以同一个姓名在同一个“员工个人资料”数据表里可能出现两次或更多次。

Covering Index。使用**覆盖索引**的好处是**辅助索引不包含整行记录的所有信息**，故其大小要远小于聚集索引，因此可以减少大量的 IO 操作。

### 13.Connection Pool

预先在缓冲池中放入一定数量的连接，当需要建立数据库连接时，只需从“缓冲池”中取出一个，使用完毕之后再放回去。**数据库连接很昂贵**，

从内存中获取和归还连接的效率，远远高高每次连接的创建和销毁，大大提高了服务器的性能，所以使用连接池是很用必要的

简单的获取一个连接，系统却要在背后做很多消耗资源的事情，大多时候，创建连接的时间比执行sql语句的时间还要长。

它的优点有：   1、Druid连接池对于SQL的性能监控很严格。   2、Druid连接池的扩展性很好，用户可以自己编写filter拦截JDBC的任意方法，在上面进行扩展，例如进行用户名密码加密，sql日志等等。   3、Druid连接池支持目前大部分数据库。

### 14.Link list/ArrayList/Vector/Set

### 15.Map/HashMap/CoHashMap

### 16.多线程实现方式

继承Thread类，优点：轻松实现，实例化对象之后，直接调用start方法即可。缺点：Java单继承局限，所以一般不用。

实现Runnable 接口，避免了继承的局限，实现优点繁琐

* **1.继承Thread类，重写run方法**
* **2.实现Runnable接口，重写run方法，实现Runnable接口的实现类的实例对象作为Thread构造函数的target**
* **3.通过Callable和FutureTask创建线程**
* **4.通过线程池创建线程**

前面两种可以归结为一类：无返回值，原因很简单，通过重写run方法，run方式的返回值是void，所以没有办法返回结果 后面两种可以归结成一类：有返回值，通过Callable接口，就要实现call方法，这个方法的返回值是Object，所以返回的结果可以放在Object对象中

### 17.Junit/unit test

Junit 单元测试

JUnit4**通过注解的方式来识别测试方法**。目前支持的主要注解有：

* @BeforeClass 全局只会执行一次，而且是第一个运行
* @Before 在测试方法运行之前运行
* @Test 测试方法
* @After 在测试方法运行之后允许
* @AfterClass 全局只会执行一次，而且是最后一个运行
* @Ignore 忽略此方法

unittest单元测试框架不仅可以适用于单元测试，还可以适用WEB自动化测试用例的开发与执行，该测试框架可组织执行测试用例，并且提供了丰富的断言方法，判断测试用例是否通过，最终生成测试结果

### 18.Socket网络通信

Socket是什么呢？ Socket是应用层与TCP/IP协议族通信的中间软件抽象层，它是一组接口。在设计模式中，Socket其实就是一个门面模式，它把复杂的TCP/IP协议族隐藏在Socket接口后面，对用户来说，一组简单的接口就是全部，让Socket去组织数据，以符合指定的协议。

### 19.Transactional

@Transactional可以说是spring中最常用的注解之一了，通常情况下我们在需要对一个service方法添加事务时，加上这个注解，如果发生unchecked exception，就会发生rollback

@Transactional 注解只能应用到 public 可见度的方法上。如果你在 protected、private 或者package-visible 的方法上使用@Transactional 注解，它也不会报错，但是这个被注解的方法将不会展示已配置的事务设置。

Spring团队建议在具体的类（或类的方法）上使用 @Transactional 注解, 因为注解是 不能继承的。或者只能当你设置了基于接口的代理时它才生效

## Spark部分

### 1.saveAsTextFile

saveAsTextFile, 一般而言，saveAsTextFile会按照执行task的多少生成多少个文件. 一般一个分区对应一个task，就会生成一个文件。可能导致小文件过多的问题。

在RDD上调用coalesce(1,true).saveAsTextFile()，意味着做完计算之后将数据汇集到一个分区，然后再执行保存的动作，显然，一个分区，Spark自然只起一个task来执行保存的动作，也就只有一个文件产生了。又或者，可以调用repartition(1)，它其实是coalesce的一个包装，默认第二个参数为true。

你虽然可以这么做，但代价是巨大的。因为Spark面对的是大量的数据，并且是并行执行的，如果强行要求最后只有一个分区，必然导致大量的磁盘IO和网络IO产生，并且最终执行reduce操作的节点的内存也会承受很大考验。Spark程序会很慢，甚至死掉。

### 2.saveAsTable

saveAsTable会利用hive API将Dataset持久化为表，其中表的元数据默认用derby存了一个数据库中，表的数据会存在spark.sql.warehouse.dir变量的文件夹下。

### 3.saveAs

行动算子，每一个行动算子对应一个job，一个job对应多个stage，一个stage对应多个task

saveAsTextFile、saveAsSequenceFile、saveAsObjectFile

def saveAsTextFile(path: String): Unit

def saveAsTextFile(path: String, codec: Class[\_ <: CompressionCodec]): Unit

**saveAsTextFile**用于将RDD以文本文件的格式存储到文件系统中。

codec参数可以指定压缩的类名

**saveAsSequenceFile**用于将RDD以SequenceFile的文件格式保存到HDFS上。

用法同saveAsTextFile。

**saveAsObjectFile**用于将RDD中的元素序列化成对象，存储到文件中

### 4.Spark内存模型

Spark 1.6之后引入的统一内存管理机制，与静态内存管理的区别在于存储内存和执行内存共享同一块空间，可以动态占用对方的空闲区域。

默认情况下堆外内存并不启用，spark.memory.offheap.enable开启，spark.memory.offheap.size设置大小

### 5.partitionBy和Repartition

Spark中，repartition和partitionBy都是重新分区的算子，其中partitionBy只能作用于PairRDD. 但是，当作用于PairRDD时，repartition和partitionBy的行为是不同的。repartition是把数据随机打散均匀分布于各个Partition；而partitionBy则在参数中指定了Partitioner（默认HashPartitioner），将每个(K,V)对按照K根据Partitioner计算得到对应的Partition. 在合适的时候使用partitionBy可以减少shuffle次数，提高效率

repartition 其实使用了一个**随机生成的数**来当做 Key，而不是使用原来的 Key！！

### 6.Repartition和coalesce

它们两个都是RDD的分区进行重新划分，repartition只是coalesce接口中shuffle为true的简易实现。repartition底层调用的也是coalesce，只不过参数shuffle=true，可以用来增加分区。coalesce可以用来减少分区，默认不开启shuffle。

### 7.Cache/Persist

cache底成调用的是 Persist(Memory only), 在Persist中，存储级别可以更换，memory，memory+disk，disk，serilrize (序列化，以时间换空间)

### 8.CheckPoint

cache给算子缓存，rdd复用，ck，检查点，快照恢复，一般存储在hdfs高可用文件系统上

对于cache，若机器发生故障，内存或者磁盘中缓存的数据丢失时，就要根据lineage(血统)进行数据恢复，想象一下，如果在这之前有100个rdd，那么在要经过100次的转换，才能将数据恢复过来，这样效率非常低。

所以可以使用rdd的checkpoint机制(检查点，相当于快照)，将你认为很重要的rdd存放到一个公共的高可用的存储系统中去，如hdfs，下次数据丢失时，就可以从前面ck的rdd直接进行数据恢复，而不需要根据lineage去从头一个一个的去恢复，这样极大地提高了效率。

cache保留血缘关系，checkpoint切断血缘关系，使用checkpoint会重新计算一次，建议使用checkpoint前，先cache下。cache+checkpoint。建议先将rdd缓存一下，这样会直接对内存中的数据进行ck, 不然的话还要启动一个任务根据rdd的依赖关系去重新计算

### 9.RDD, DF, DS

1）RDD

**优点:**

编译时类型安全

编译时就能检查出类型错误

面向对象的编程风格

直接通过类名点的方式来操作数据

**缺点:**

序列化和反序列化的性能开销

无论是集群间的通信, 还是IO操作都**需要对对象的结构和数据进行序列化和反序列化**。

GC的性能开销，频繁的创建和销毁对象, 势必会增加GC

2）DataFrame

DataFrame引入了schema和off-heap

schema : RDD**每一行的数据, 结构都是一样的，这个结构就存储在schema中。 Spark通过schema就能够读懂数据, 因此在通信和IO时就只需要序列化和反序列化数据, 而结构的部分就可以省略了**。

DataFrame底层就是row类型的dataset：type DataFrame = Dataset[Row]

DataFrame也是懒执行的，但性能上比RDD要高，主要原因：

优化的执行计划，即查询计划通过Spark catalyst optimiser进行优化。（课件中的例子join

之前，优化器可以提前过滤）

**3）DataSet**

DataSet结合了RDD和DataFrame的优点，并带来的一个新的概念Encoder。

当序列化数据时，Encoder产生字节码与off-heap进行交互，**能够达到按需访问数据的效果，而不用反序列化整个对象**。Spark还没有提供自定义Encoder的API，但是未来会加入。

①　是DataFrame API的一个扩展，是SparkSQL最新的数据抽象；

②　用户友好的API风格，既具有类型安全检查也具有DataFrame的查询优化特性；

**③**　用样例类来对DataSet中定义数据的结构信息，样例类中每个属性的名称直接映射到DataSet中的字段名称；

**④**　**DataSt\***是强类型的。比如可以有 DataSet[Car]，DataSet[Person]

DataSet为什么会逐步取代DataFrame?

DataFrame是弱类型的，类似于jdbc的resultset，不仅要求开发人员熟知个字段的类型，还要明确个字段的顺序。DataSet是强类型的，我们可以直接通过字段名来操作数据。

**三者之间的转换：**

## Hive部分

### 1.四个By

OrderBy，SortBy, distributed by, cluster by

1）Sort By：分区内有序；

2）Order By：全局排序，只有一个Reducer；

3）Distrbute By：类似MR中Partition，进行分区，结合sort by使用。

4）Cluster By：当Distribute by和Sort by字段相同时，可以使用Cluster by方式。Cluster by除了具有Distribute by的功能外还兼具Sort by的功能。但是排序只能是升序排序，不能指定排序规则为ASC或者DESC。

### 2.自定义函数

**UDF:** 单行进入，单行输出

**UDAF:** 多行进入，单行输出

**UDTF:** 单行输入，多行输出

自定义UDF、UDTF

在项目中是否自定义过UDF、UDTF函数，以及用他们处理了什么问题，及自定义步骤？

1）自定义过。

2）用UDF函数解析**公共字段**；用UDTF函数解析**事件字段**。UDAF在sparksql中使用到。

自定义UDF：继承UDF，重写evaluate方法

自定义UDTF：继承自GenericUDTF，重写3个方法：initialize(自定义输出的列名和类型)，process（将结果返回forward(result)），close

为什么要自定义UDF/UDTF，因为自定义函数，可以自己埋点Log打印日志，出错或者数据异常，方便调试.

### 3.hive文件存储格式

1.textFile

默认格式；

存储方式为行存储；

磁盘开销大 数据解析开销大；

但使用这种方式，hive不会对数据进行切分，从而无法对数据进行并行操作。

2、sequencefile

二进制文件,以<key,value>的形式序列化到文件中； 存储方式：行存储； 可分割 压缩； 一般选择block压缩； 优势是文件和[Hadoop](http://lib.csdn.net/base/hadoop) api中的mapfile是相互兼容的

3、rcfile

存储方式：数据按行分块 每块按照列存储； 压缩快 快速列存取； 读记录尽量涉及到的block最少； 读取需要的列只需要读取每个row group 的头部定义； 读取全量数据的操作 性能可能比sequencefile没有明显的优势，

4、orcfile

存储方式：数据按行分块 每块按照列存储；

压缩快 快速列存取；

效率比rcfile高,是rcfile的改良版本。

5` Apache Parquet

比对三种主流的文件存储格式TEXTFILE 、ORC、PARQUET 压缩比：ORC > Parquet > textFile（textfile没有进行压缩） 查询速度：三者几乎一致

### 4.Hive中数据导出表的方式

**1.将查询的结果导出到本地** insert overwrite local directory ‘本地路径’ select 指定字段 from 表名; **2.将查询的结果格式化导出到本地** insert overwrite local directory ‘/export/servers/exporthive’ row format delimited fields terminated by ‘\t’ collection items terminated by ‘#’ select \* from student; **3.将查询的结果导出到HDFS上** insert overwrite directory ‘/export/servers/exporthive/a’ select \* from score; **4.Hadoop命令导出到本地** dfs -get /export/servers/exporthive/000000\_0 /export/servers/exporthive/local.txt; dfs -get 表中数据在HDFS上的存储位置 本地路径; **5.hive shell 命令导出到本地** hive -e “select \* from myhive.score;” > /export/servers/exporthive/score.txt hive -e “select 指定字段 from 表名;” > 本地路径； **6.export导出到HDFS上** export table score to ‘/export/exporthive/score’; export table 表名 to ‘本地路径’; **7.通过sqoop方式导出**

根据导出的地方不一样，将这些方式分为三种： （1）、导出到本地文件系统； （2）、导出到HDFS中； （3）、导出到Hive的另一个表中.

（4）、导出到其他数据库。

### 5.分区，分桶和Index

以上谈论的分区和分桶都是只能优化查询到某个群体，不能具体到的那个数据，为了实现查询到的那个数据的优化，就产生了索引，对hive中的数据，会产生一个新文件，这个文件会记录每个数据的位置。

create index index\_name on table tablename(col) as 'org.apache.hadoop.hive.ql.index.compact.CompactIndexHandler'  
with deferred rebuild;

### 6.hive view

hive表之视图 1、视图是一个虚表，一个逻辑概念，可以跨越多张表。表是物理概念，数据放在表中，视图是虚表，操作视图和操作表是一样的，所谓虚，是指视图下不存数据。 2、视图是建立在已有表的基础上，视图赖以建立的这些表称为基表 3、视图可以简化复杂的查询 create view 视图表名 as select 基表1.字段1，基表1.字段2，基表2.字段1 ...... from 库名1.表名 库名2.表名 where 基表1.字段=基表2.字段 在hive中，视图中是不存数据的，在oracle和mysql中，视图是可以存数据的，称之为物化，可提高查询速度

**创建视图**：create view view\_piaofang as select \* from t\_name;

**查看视图** ：show tables; #既有表又有视图 show views #只查看视图

**查看视图的详细信息** desc formatted|extended

**删除视图** drop view view\_name;

**使用视图的注意点**：

* Hive中的试图，仅仅是一个sql语句的快捷方式
* hive中的视图只是逻辑视图，没有物化视图
* hive的视图，不支持增删改，只支持查询
* hive的视图，只有的查询的时候，才会真正执行