业务方面

 指标，来源维度表，用到哪些技术，逻辑等

以下是技术方面的

0，使用kettle遇到哪些异常，或者直接问kettle异常的

在做抽数的时候遇到的异常种类会有很多，（自己查找两个例子说出来就可以，但是一定要包含自己知道的），常见的就是空指针异常：一般是参数或者常量没设定好引起的

1. 性能优化，有一些问题是引申出来的

性能优化有三方面：索引优化，分区优化，执行计划hints优化

索引优化或者怎么建索引，哪些情况可以使用索引，哪些不适合索引，索引失效

一、索引优化—索引能够带来数据查询的效率提升，但是它会损失数据库性能

1. ，单表查询就是创建索引，也就是给经常做为查询条件的列创建索引
2. ，关联查询的时候（left join，或者inner join），在从表中建立索引，索引的字段是查询的条件，主表中不用建立索引
3. 子查询优化：子查询不要放在被驱动表；子查询不能放在select后面；先过滤大量数据之后在进行子查询
4. 使用where条件能过滤完的条件，就不要使用having，因为where的执行有与having

使用索引的场景

① 经常作为查询条件的

② 多表查询时，与其他表关联的的字段，在外键建立索引

③ 经常作为排序，统计，或者分组的字段

1. 数据可的数据量一定要大，数据量小的时候不如不建立

不适合索引的条件

① 频繁更新的字段

② 经常增删改，或者数据量较小的表

③ where条件中使用不到的情况

④ 列中null值太多

索引失效

① 没有使用到索引最左原则

② like查询 %前置，如like ‘%张’

③ 使用不等于符号，或者not in 或者or

④ 出现了隐式转换，如字符串作为查询条件时候，没有使用单引号

分区优化：range list hash 组合这四种，使用最多的就是range范围分区，range分区里面使用最多的就是时间分区，其他的自己查询记一下。

Range范围分区时候，有几个原则

1. 每个分区都会有一个value less then 字句，不包括上限值
2. 索引分区除了第一个，其他的都会有隐藏的下限，就是前一个的分区值
3. 在最高的分区中，最大值被定义

List 分区：适用于某列的值只有几个

Hash 分区：使用于没有合适的条件下的

执行计划（hints）优化-----怎么看执行计划，或者在执行计划里面有哪些连接方式

执行计划或者也叫explan解析计划，一般是f5或者是在select前面加上explan，在解析计划中，主要看有没有使用到索引，判断索引有没有失效，去判断sql的耗费或者耗时，hints优化使用的最多的是并行，在说一下命令结构

连接方式自己去查一下

帆软的使用流程

百度一下，在说写sql报表展示的时候提一下报表的sql不要写的关联太多，否则会耗时，用户体验不好

两者模型的差别

星星模型使用的最多，因为它允许数据冗余，在查询的时候不用做过多的条件连接，查询效率比雪花模型的高（雪花不允许数据冗余），然后在介绍两者模型的定义

存储过程使用的技术，逻辑有哪些

我说了一下业务的指标计算逻辑，他说我没理解他的问题，下来查了一下不知道是不是要回答游标，动态sql这些，如果是的话就结合业务去说

缓慢变化维怎么使用的，在前端怎么展现的

我说了一下上一个的结束时间作为下一个的开始时间，面试官说就不是这么用的

什么是高水平线

一个作业脚本昨天还运行好好的，今天无法运行，怎么分析？sql一直在执行，无法运行出结果怎么分析

数据分层，他们有什么区别

分析函数，行转列（除了case when和decode）

面试的时候，回答问题，要知道自己的答案为什么是这个，防止面试官深挖，不知道原因的情况下，看能不能绕开这个问题，回答一些侧面的。实在绕不开就委婉的说一下自己不会（或者说自己工作中用的多，具体的原理没看过；或者选则其他的方式去表达）

比如问表分区的时候，说完四种之后，就说我们常用的range分区，在说一下range分区原则或者场景就可以了，其他的三个知道就好，不要在面试中回答上来