

## 07 | 同事老打脸说数据有问题，该怎么彻底解决？

2020-04-17 郭忆

数据中台实战课

[进入课程 >](#)



讲述：郭忆

时长 20:49 大小 19.08M



你好，我是郭忆。

上一节课，我带你从模型设计层面，逐步将分散、杂乱、烟囱式的小数仓整合成了可复用、可共享的数据中台，数据研发效能提升了一倍。那是不是交付数据足够快，使用数据的人就满意了？当然不是，来看发生在我身边的一件事儿。

供应链部门运营陈英俊（化名）每天上班第一件事情，就是打开供应链辅助决策系统（数据产品），根据系统上给出的商品库存数据、区域下单数据，制订商品的采购计划，然后给供货商。可是今天当他准备工作时，突然发现系统中部分商品库存数据显示为 0，他第一时间将问题反馈给了数据部门。与此同时，与他一样无法工作的还有供应链部门的其他 50 多名运营。

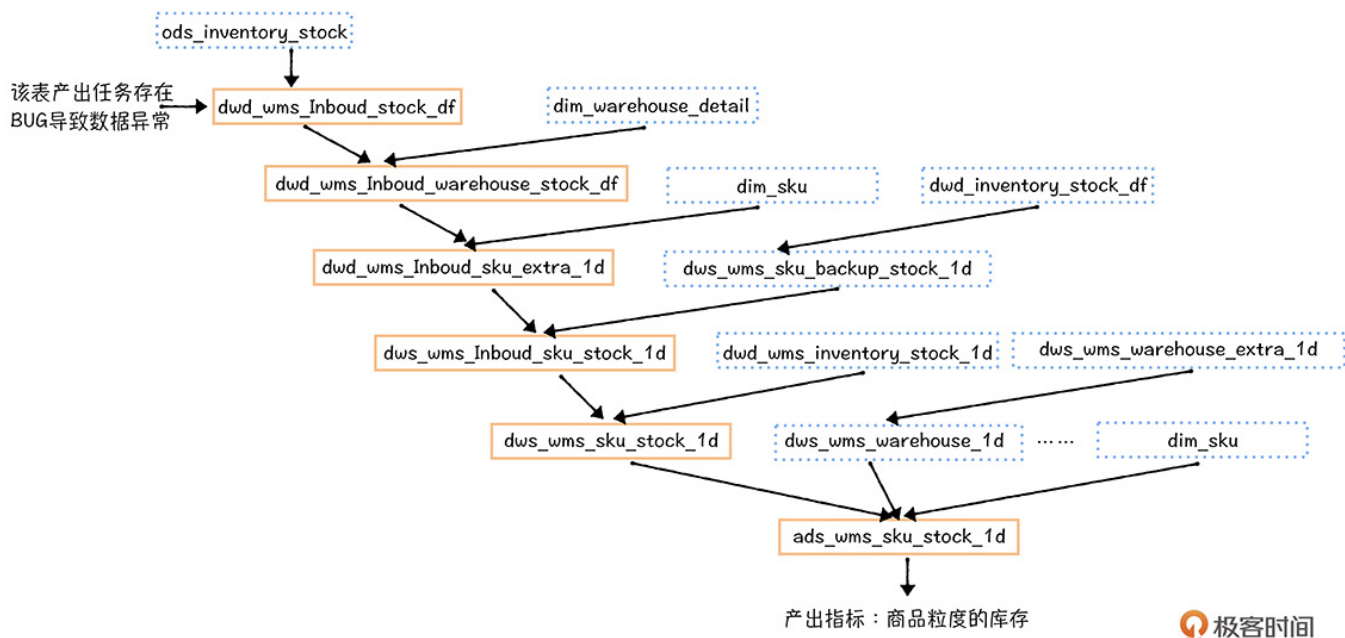


图1 ADS层商品库存表加工链路图

接到投诉后，负责库存域的数据开发郝建（化名）立即开始定位，首先就“商品库存”指标的产出表 `ads_wms_sku_stock_1d` 进行排查，确认它产出任务没有问题，是这个任务的上游输入表数据有问题，引发下游表数据异常。

从数据血缘图中你可以看到，`ads_wms_sku_stock_1d` 上游有 20 多张表，郝建逐层校验是哪个表的数据出现问题，结果锁定在了 `dwd_wms_inbound_stock_df`。这张表的产出任务在前一天有一次线上变更，任务代码存在漏洞，对部分商品入库数据格式解析异常，但是没有将异常抛出，导致产出数据表 `dwd_wms_inbound_stock_df` 数据异常，进而影响了所有下游表。

排查问题用了近 3 个小时。

既然问题定位清楚，就要开始修复的流程。修改好代码后，郝建重新跑了 `dwd_wms_inbound_stock_df` 的产出任务，确认数据没有问题，然后需要重跑该任务下游链路上的 5 个任务（图中红色表的产出任务）。

运行完任务用了 5 个小时。

经过数据验证，确认没有问题，此时已经过去了将近 9 个小时。对于像陈英俊这样的运营来说，一天都无法工作。如果你是陈英俊，对数据会满意吗？**所以，光快还不够，还要保证质量。**

当然，这个例子暴露出这样几个问题：

数据部门晚于业务方发现数据异常，被投诉后才发现问题。

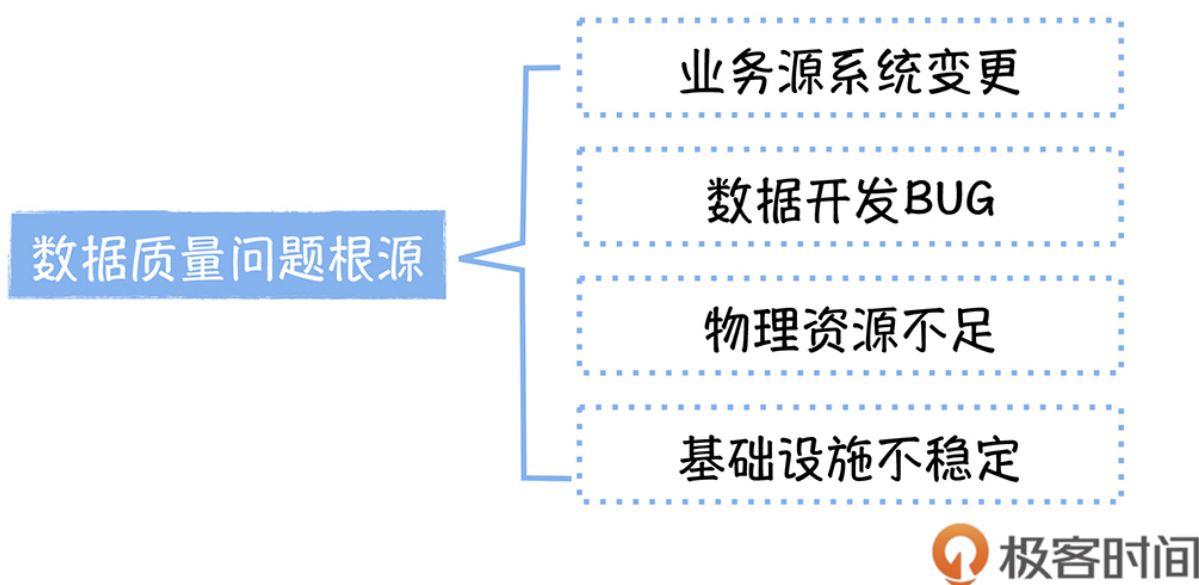
出现问题后，数据部门无法快速定位到数据异常的根源，排查用了较长的时间。

故障出现在数据加工链路的上游顶端，出现问题没有第一时间报警处理，导致问题修复时，所有下游链路上的任务都要运行，修复时间成本非常高。

**这些问题最终导致了数据长时间不可用。**那如何解决这些问题，确保数据高质量的交付呢？首先，你要了解产生这些问题的根源，毕竟认识问题才能解决问题。

## 数据质量问题的根源

在网易电商业务数据中台构建之初，我对数据团队一年内，记录在案的 321 次数据质量事件做了逐一分析，对这些事件的原因进行了归纳，主要有下面几类。这里多说一句，如果你想改进数据质量，不妨也对过去踩过的坑做一次复盘，归一下类，看看问题都出在哪里，然后制定针对性的改进计划。



### 业务源系统变更

数据中台的数据来源于业务系统，而源系统变更一般会引发 3 类异常情况，

**首先是源系统数据库表结构变更。**例如业务系统新版本发布上线，对数据库进行了表结构变更，增加了一个字段，同时对部分字段的类型、枚举值进行了调整。这种表结构变更没有通

知到数据团队，导致数据同步任务或者数据清洗任务异常，进而影响了下游数据产出。

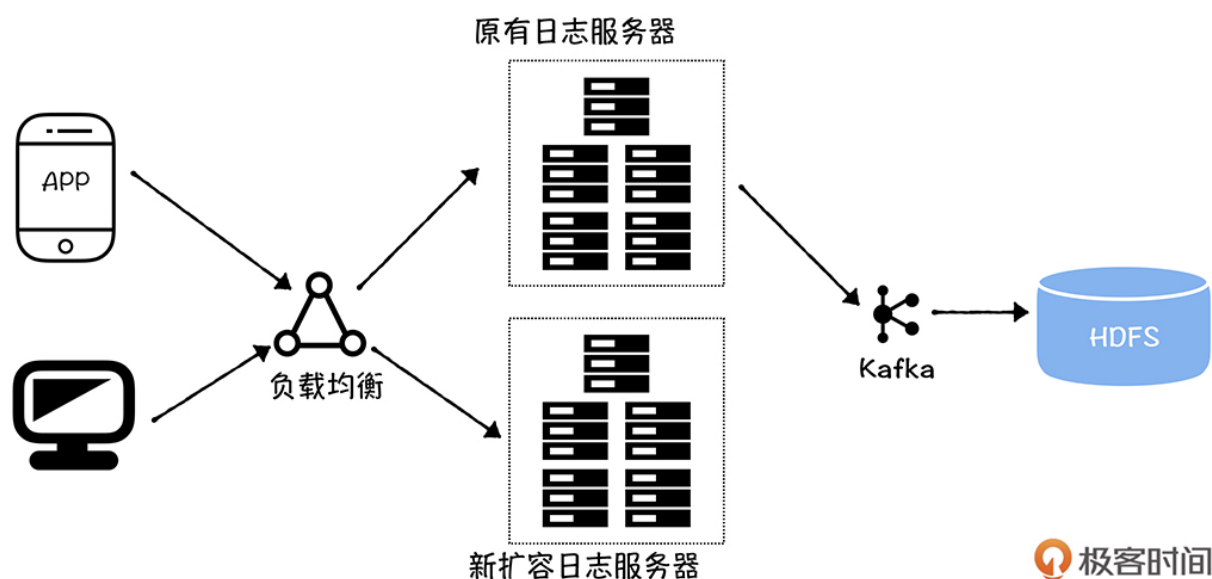


图2 日志服务器扩容误操作引发数据异常

**第二个是源系统环境变更。**我经常在大促期间见到这种情况，其中的典型是前端用户行为埋点日志量暴增，系统管理员紧急对服务器进行扩容，上线了 5 台新的服务器，但是没有配置这 5 台服务器的日志同步任务，结果导致数据侧少了这 5 台服务器的数据，最终影响了数据计算结果的准确性。

**最后一个源系统日志数据格式异常。**这种情况通常出现在前后端埋点日志中。业务系统发布上线引入埋点 BUG，导致 IP 格式出现了不符合约定的格式（比如，我们一般约定的 IP 格式是 166.111.4.129，结果出现了 166.111.4.null），最终也会导致计算结果错误。

## 数据开发任务变更

这种情况在数据质量事件中占到了 60% 以上，而它大多数是由于数据开发的纰漏引发的，来看几个你比较熟悉的例子：

任务发布上线，代码中引用的测试库没有修改为线上库，结果污染了线上数据；

任务发布上线，代码中使用了固定分区，没有切换为 “\${azkaban.flow.1.days.ago}”，导致数据异常；

前面例子中，数据格式处理错误，代码忽略了异常，导致数据错误；

任务配置异常，它通常表现在任务没有配置依赖，前一个任务没有运行完，后一个任务就开始运行，输入数据不完整，导致下游数据产出错误。



物理资源不足

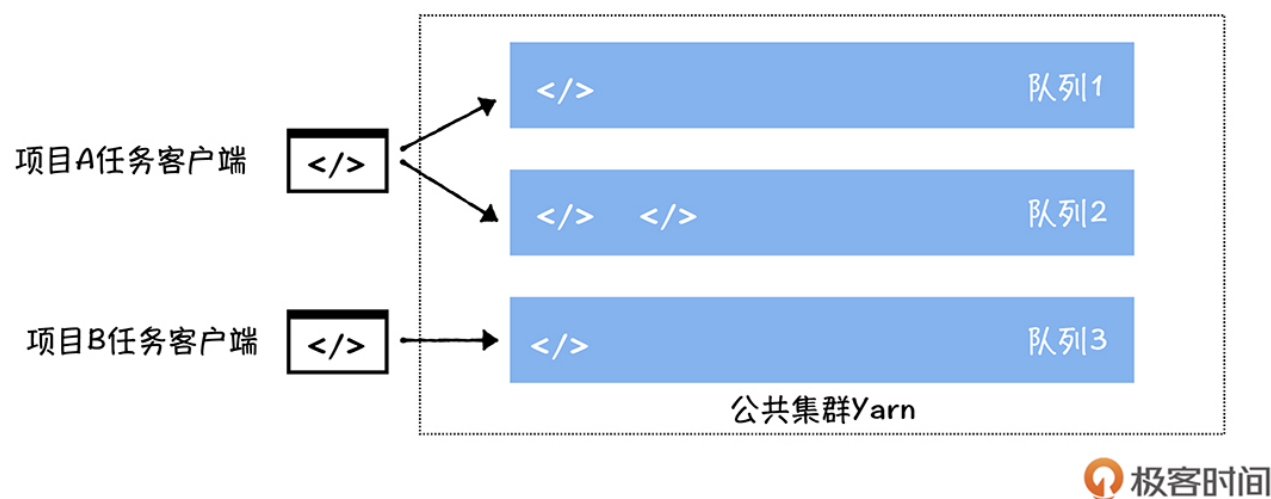


图3 Hadoop多队列任务提交

在多租户下，Hadoop 生态的大数据任务（MR，Hive，Spark）一般运行在 yarn 管理的多个队列上（调度器为 CapacityScheduluer），每个队列都是分配了一定大小的计算资源（CPU、内存）。

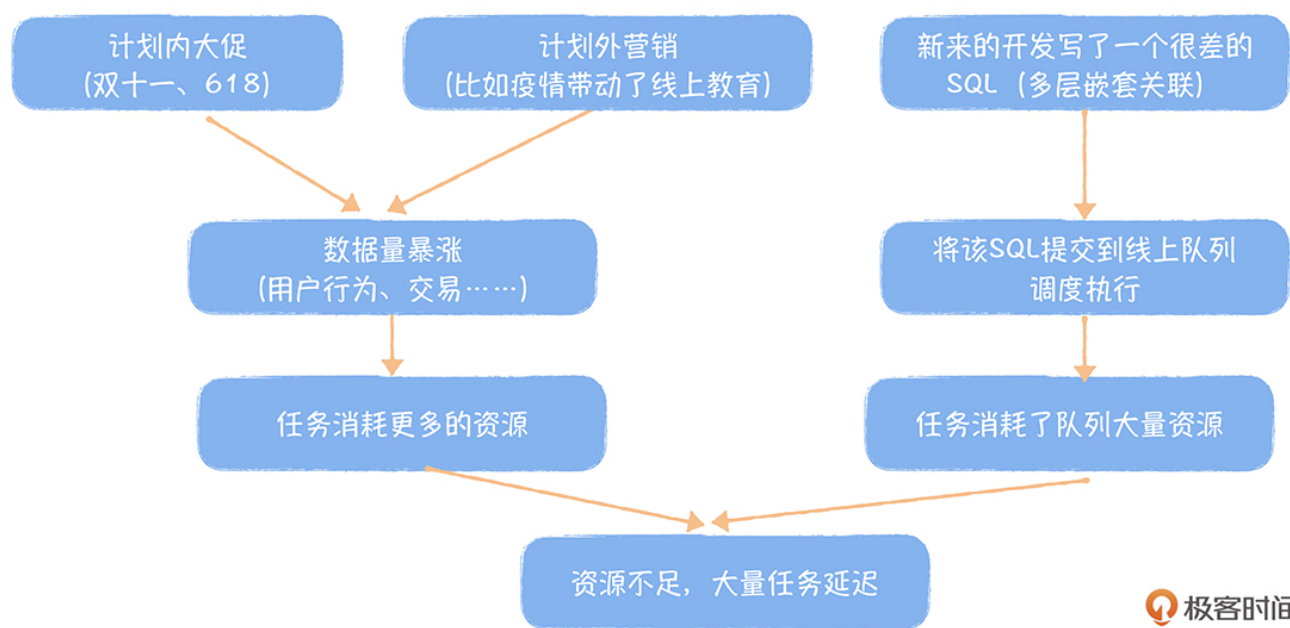


图4 物理资源不足导致任务延迟

我展示了两种常见的物理资源不足，导致任务延迟产出的情况。

基础设施不稳定

从数量上来看，这类异常不算多，但影响却是全局性的。我们曾经在大促期间，碰到了一个 [Hadoop 2.7 NameNode 的 BUG](#)，造成 HDFS 整个服务都停止读写，最终通过临时

补丁的方式才修复。

总的来说，出现问题并不可怕，可怕的是，我们没有及时发现问题，尽快恢复服务，举一反三地通过流程和技术手段，降低问题出现的概率。所以接下来我们就来看一看，如何提高数据质量？

## 如何提高数据质量？

我认为，要想提升数据质量，最重要的就是“早发现，早恢复”：

早发现，是要能够先于数据使用方发现数据的问题，尽可能在出现问题的源头发现问题，这样就为“早恢复”争取到了大量的时间。

早恢复，就是要缩短故障恢复的时间，降低故障对数据产出的影响。

那具体如何做到这两个早呢？我总结了一套数据质量建设的方法，包括这样几个内容。

### 添加稽核校验任务

在数据加工任务中，对产出表按照业务规则，设计一些校验逻辑，确保数据的完整性、一致性和准确性，这是提升数据质量最行之有效的方法。

通常建议你在数据产出任务运行结束后，启动稽核校验任务对数据结果进行扫描计算，判断是否符合规则预期。如果不符合，就根据提前设定的强弱规则，触发不同的处理流程。

如果是强规则，就立即终止任务加工链路，后续的任务不会执行，并且立即发出电话报警，甚至我们要求，关键任务还要开启循环电话报警，直到故障被认领；如果是弱规则，任务会继续执行。但是存在风险，这些风险会通过邮件或者短信的方式，通知到数据开发，由人来进一步判断风险严重程度。

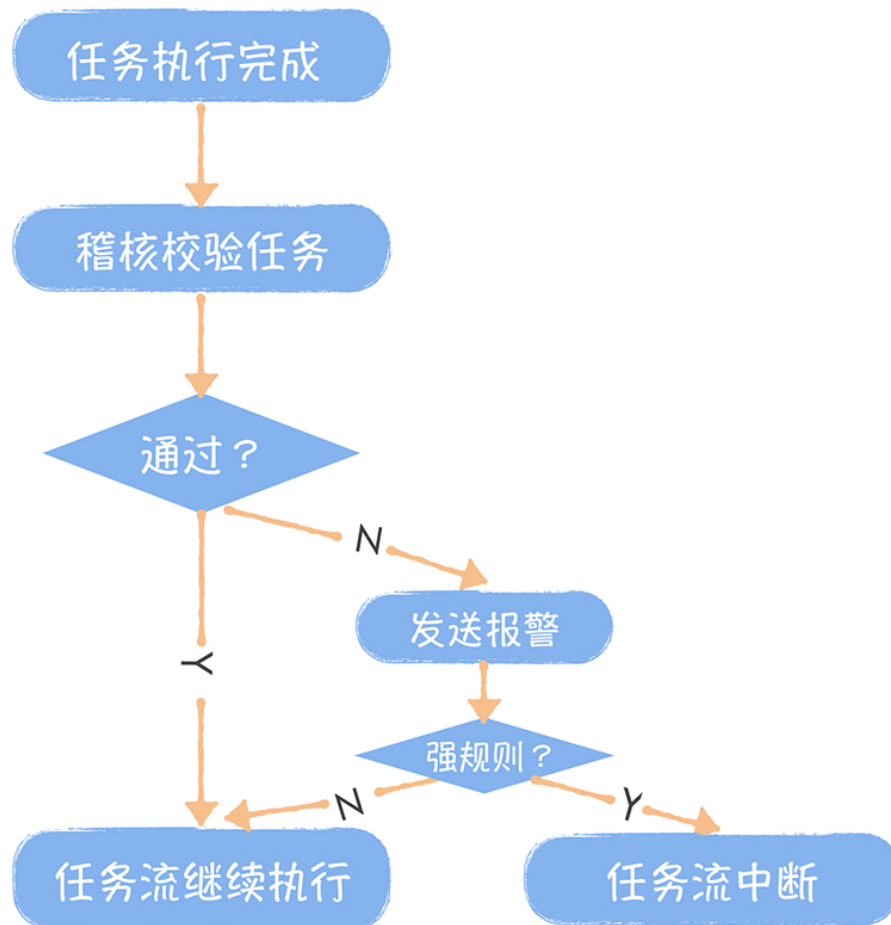


图5 稽核校验执行流程图

那具体要加哪些稽核规则呢？

**完整性规则。**主要目的是确保数据记录是完整的，不丢失。常见的稽核规则有表数据量的绝对值监控和波动率的监控（比如表波动超过 20%，就认为是异常）。还有主键唯一性的监控，它是判断数据是否有重复记录的监控规则，比较基础。除了表级别的监控，还有字段级别的监控（比如字段为 0、为 NULL 的记录）。

**一致性规则。**主要解决相关数据在不同模型中一致性的问题。商品购买率是通过商品购买用户数除以商品访问 uv 计算而来的，如果在不同的模型中，商品购买用户数是 1W、商品访问 uv10W，商品购买率 20%，那这三个指标就存在不一致。

**准确性规则。**主要解决数据记录正确性的问题。常见的稽核规则有，一个商品只能归属在一个类目，数据格式是不是正确的 IP 格式，订单的下单日期是还没有发生的日期等等。

它们是强规则还是弱规则，取决于业务对上述异常的容忍度（比如涉及到交易、支付跟钱相关的，一般都会设置为强规则，对于一些偏行为分析的，一般都是弱规则）。

## 建立全链路监控

在 06 讲中，我强调数据中台的模型设计是分层的，确保中间结果可以被多个模型复用。不过这会导致数据加工的链路变长，加工链路的依赖关系会非常复杂，最终当下游表上的某个指标出现问题，排查定位问题的时间都会比较长。**所以，我们有必要基于数据血缘关系，建立全链路数据质量监控。**

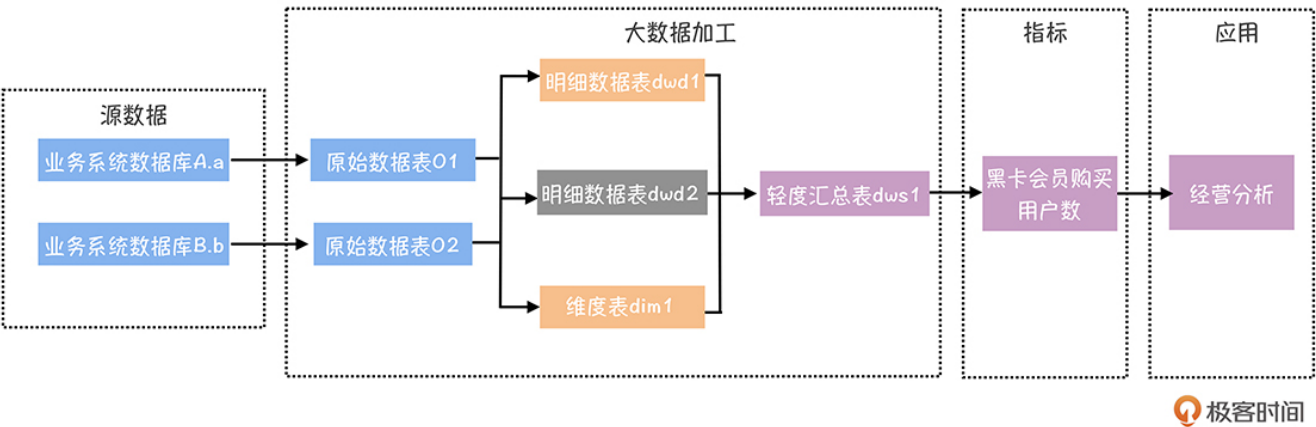


图6 全链路数据质量监控

从这个图中你可以看到，业务系统的源数据库表是起点，经过数据中台的数据加工链路，产出指标“黑卡会员购买用户数”，数据应用是链路的终点。

对链路中每个表增加稽核校验规则之后，当其中任何一个节点产出的数据出现异常时，你能够第一时间发现，并立即修复，做到早发现、早修复。另外，即使是使用方反馈经营分析上的黑卡会员购买用户数，相较于昨天数据大幅下降超过 30%，你也可以快速判定整个指标加工链路上节点是否运行正常，产出任务是否有更新，提高了问题排查速度。

## 通过智能预警，确保任务按时产出

在数据质量问题中，我提到会存在物理资源不足，导致任务产出延迟的情况。在网易，所有数据中台产出的指标要求 6 点前产出。为了实现这个目标，我们需要对指标加工链路中的每个任务的产出时间进行监控，基于任务的运行时间和数据血缘，对下游指标产出时间进行实时预测，一旦发现指标无法按时产出，则立即报警，数据开发可以终止一些低优先级的任务，确保核心任务按时产出。



## 通过应用的重要性区分数据等级，加快恢复速度

稽核校验会消耗大量的资源，所以只有核心任务才需要。核心任务的定义是核心应用（使用范围广、使用者管理级别高）数据链路上的所有任务。

## 规范化管理制度

讲到这儿，你可能会问：数据质量取决于稽核规则的完善性，如果数据开发没有添加，或者添加的规则不全，是不是就达不到早发现、早恢复？

这个问题戳中了要害，就是规则的完备性如何来保障。在我看来，这不仅仅是一个技术问题，也涉及管理。在网易，我们会制定一些通用的指导规则（比如，所有数据中台维护的表都需要添加主键唯一性的监控规则），但这些通用规则往往与业务关系不大。如果涉及业务层面，就要由数据架构师牵头，按照主题域、业务过程，对每个表的规则进行评审，决定这些规则够不够。

那我想建议你，如果要做稽核校验，可以通过组建数据架构师团队，由这个团队负责核心表的规则审核，确保规则的完备性。

那么当你按照这几个方法建立了数据质量体系之后，要如何验证体系是否有效呢？

## 如何衡量数据质量？

做数据治理，我一直奉行“效果可量化”的原则，否则这个治理做到什么程度，很难衡量。那么如何评价数据质量是否有改进呢？除了故障次数，你还可以有这样几个指标。

4 点半前数据中台核心任务产出完成率。这个指标是一个综合性指标，如果任务异常，任务延迟，强稽核规则失败，都会导致任务无法在规定时间内产出。

基于稽核规则，计算表级别的质量分数。根据表上稽核规则的通过情况，为每个表建立质量分数，对于分数低的表，表负责人要承担改进责任。

需要立即介入的报警次数，通常以开启循环报警的电话报警次数为准。对于核心任务，任务异常会触发循环电话报警，接到报警的数据开发需要立即介入。

数据产品 SLA。每个数据产品上所有指标有没有在 9 点产出，如果没有，开始计算不可用时间，整体可以按照不同数据产品的重要性进行折算，99.8% 是数据产品一个相对比较好的 SLA。

不过，技术和规范最终需要依靠产品来帮助落地，在网易内部，有一个数据质量中心的产品，通过介绍这个产品，我希望能给你一个参考，如何去设计一个数据质量中心，或者在选型的时候，数据质量中心必须具备的功能。

## 数据质量中心

数据质量中心（以下简称 DQC）的核心功能是稽核校验和基于数据血缘的全链路数据质量监控。

数据质量中心

数据管理

质量大屏

质量评估

表质量评分

监控列表

监控执行历史

执行实例

执行趋势

新建规则

模板规则

自定义规则

规则类型

☒ 字段级 ☐ 表级

\* 字段名

submit\_gorder\_cnt...

\* 规则模板

请选择规则模板

期望范围

指标	配置
规则命中行数	> <input type="text"/>
规则命中行数比率	> <input type="text"/> %

2个指标中至少配置1项

规则控制

☐ 强 规则不通过时，任务置为失败状态，停止运行

☒ 弱 规则不通过时，任务继续运行

描述

0/120

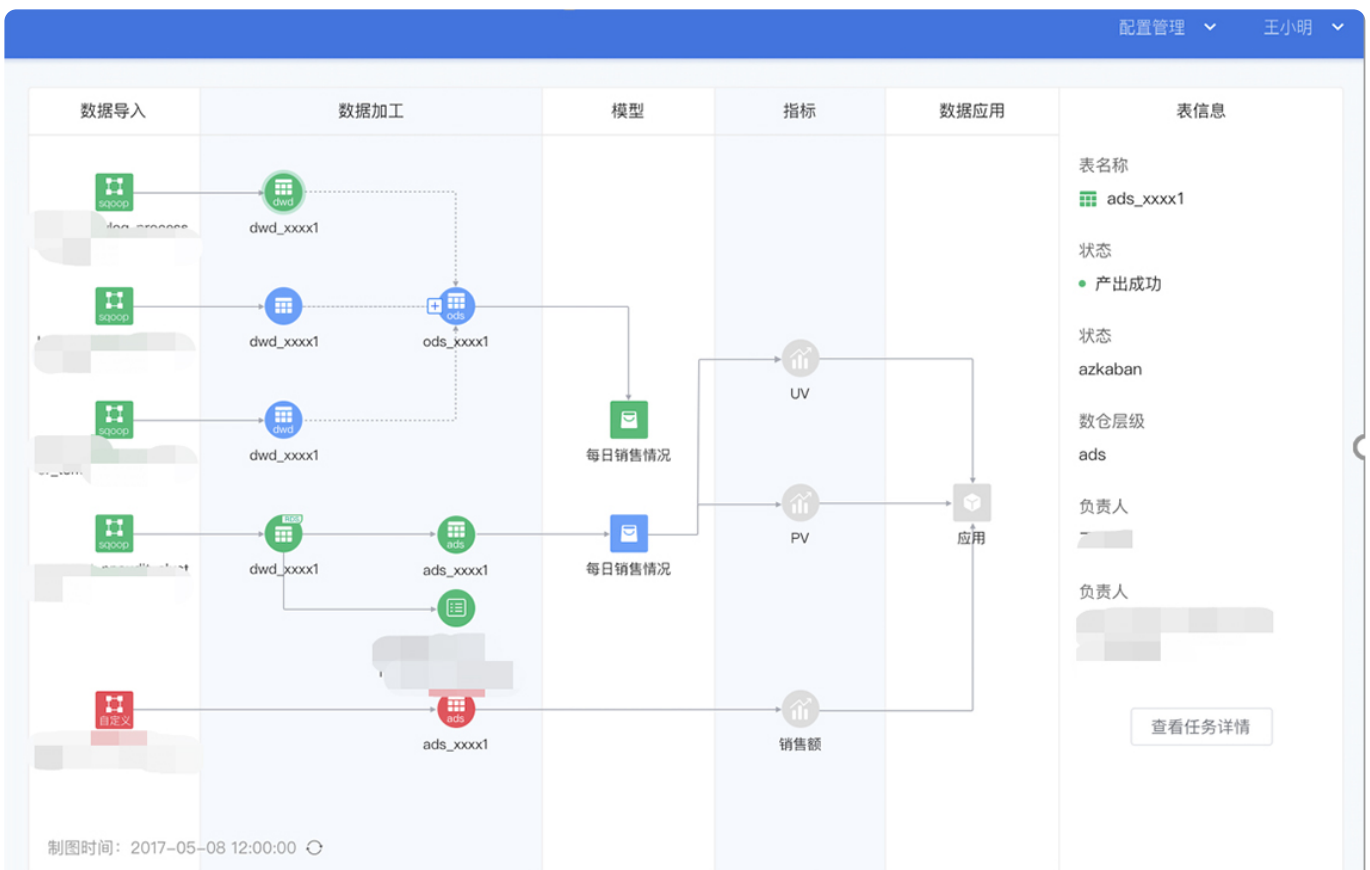
新建

取消

DQC 的首页是质量大屏，提供了稽核规则的数量、表的覆盖量以及这些规则的执行通过情况。通过这些数据，你就能跟你的老板讲清楚，目前数据质量水平建设如何？目标是多少？距离目标还有多少差距。



在 DQC 中创建稽核规则非常简单，DQC 内置了大量的基础规则，例如 IP 字段格式校验，主键唯一性校验，表行数波动率校验，同时还提供了自定义 SQL 的方式，允许业务层面的规则创建，例如我们前面提到的一致性规则中，两个指标相除等于第三个指标，就可以通过自定义 SQL 解决。



DQC 还提供了全链路监控的功能，覆盖了从数据导入、数据加工、模型产出、指标、到数据应用的完整链路。绿色节点代表数据正常，蓝色节点代表数据正在产出中，红色节点代表数据异常，灰色节点代表产出任务还未被调度到。通过这个监控，大幅提高了问题发现和定位的速度。

所以你可以发现，一个好用的 DQC，必须要具备的功能就是质量度量、稽核规则管理以及全链路监控。

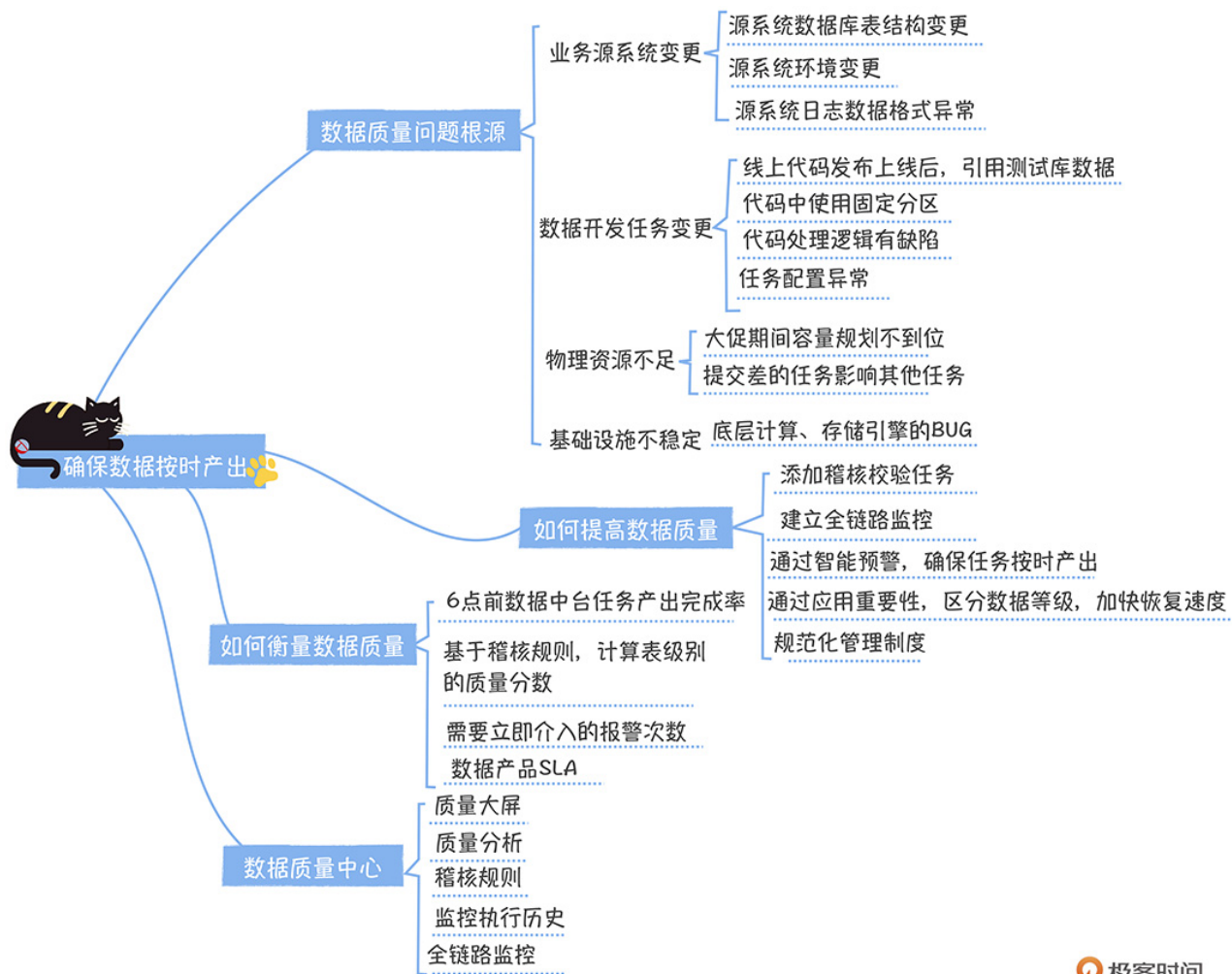
## 课堂总结

本节课，我从数据质量问题的根源入手，带你分析了背后的原因，和五种提高数据质量的方法，在课程结束前，我再强调几个重点：

数据质量治理必须要做到全链路，从业务系统的数据源到指标所在的应用，这样可以提前发现问题，将故障消灭在摇篮中；

根据应用的优先级和全链路血缘关系，圈定核心任务，要确保核心任务的稽核规则全覆盖，优先保障核心任务的按时产出，在资源紧缺时，有必要停止非核心任务；

稽核规则的完备性，可以通过数据架构师团队对每个域下的核心表进行评审的方式保障，同时问题回溯和复盘，也可以不断地完善。



## 思考时间

我提到数据完整性可以通过数据记录的波动率来监控，如果超过 20% 的波动，应该被视为异常。但是你有没有想过，这种也存在误判的情况，尤其是在双十一大促期间，大概率数据稽核规则都是异常的，此时你又该怎么办？

最后，感谢你的阅读，如果这节课让你有所收获，也欢迎你将它分享给更多的人。



点击参与 

# 和郭忆一起，落地数据中台



扫一扫参与小程序话题



新版升级：点击「 请朋友读」，20位好友免费读，邀请订阅更有**现金**奖励。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 06 | 数据模型无法复用，归根结底还是设计问题

下一篇 08 | 交付速度和质量问题解决了，老板说还得“省”

## 精选留言 (9)

 写留言



吴科

2020-04-17

数据部门作为下游系统，上游业务系统变更，往往不能第一时间通知，第二天跑批是才发现，早晨五点起来处理太平常不过了。

资源抢占也时常发生，分析师临时加了一个任务跑全量的数据，还加到了资源主列队中，第二天所有跑批都延迟了。

数据部门，要规范各种数据相关的变更才能解决。实际中，我们数据部门都是弱勢的。...

展开

作者回复: 你说的都是大实话。

我真心觉得数据开发真是苦逼的职位，白天忙需求，晚上接报警，因为指标业务口径不一致、数据质量问题天天被人怼，处于弱勢地位，工具产品也不到位，所以需要数据中台来拯救他们。

感谢你的阅读，期待与你在留言区再次相遇~

2

3



小桥流水

2020-04-18

考虑增加2个指标

- 1、不同时间点的任务完成率
- 2、当天累计任务完成率

展开

作者回复: 这两个指标, 对于预防一些大面积的故障, 还是有帮助的, 例如基础设施层面的故障。不过可能粒度还是太粗了, 最好是能够到任务粒度的, 才能起到预警的作用。如果只是衡量结果, 我觉得只要一个基线任务产出时间完成率就够了。

感谢你的阅读, 期待与你再次相遇~

1

1



leslie

2020-04-17

数据规则, 各自为战。最近刚碰到数据问题, 历史遗留下来的数据库设计造成的坑; 虽然之前指定了一堆相应的规则, 可是提交层还是出了不少隐患。

提交到平台其实是一个不错的方式与方法: 多方面皆有隐患, 今天的课程倒是引发了一些思路和操作方式。

现在来回答今天的问题: 误判的原因是源自对于某些表的数据量或者说设计上做复杂了...

展开

作者回复: 感谢你的肯定和鼓励, 看到能够引发你的一些思考, 我真的觉得这篇文章还是很有意义的。

再来谈谈我在课后留的这个问题。其实稽核监控的规则设定, 都是根据日常流量的正常波动范围来设定的, 如果遇到大规模的引流或者重大促销, 必然会不适用, 所以如果不调整, 大促期间, 稽核监控基本全部会触发报警, 也失去了早发现早恢复的用途。

所以一般在大促期间, 我们一般会根据历史大促的经验, 预留一些Buffer, 来调整稽核监控。

感谢你的阅读, 期待与你再次相遇~

1

1

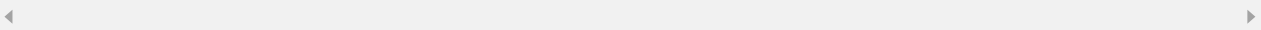


JohnT3e

2020-04-17

关于全链路监控深有体会。如果问题不早发现，而最终由业务反馈的话，可能会面临整个数据流程的数据回滚和重新加工，进而引发连锁反应，导致更多任务超时或者错误。

作者回复: 看来你也遇到了，哈哈。希望07数据质量的介绍对你有所帮助，能够解决你当前面临的问题~



1



Geek\_f071bc

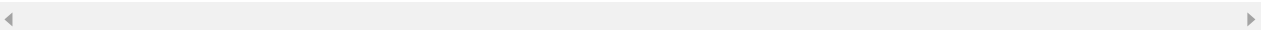
2020-04-19

老师，基于稽核规则，计算表级别的质量分数？能举个例子吗，这个质量分数怎么算？

作者回复: 你好，每个有稽核规则的表，会计算质量分数。质量分数的计算主要是结合规则是否成功，规则的数量，规则是强规则还是弱规则。比如我们约定，强规则失败一次扣4分，弱规则失败一个扣2分，然后我们就可以计算某一天的某个表的质量分。

你可能会有疑问，那加的规则越多，不是风险就越大了么？确实会有这样的问题，但是规则的完备性不是由数据开发单方面决定的，是经过由数据架构师，域负责人组成的评审小组确定的，所以会确保每个表的稽核规则尽可能的完备，另外事故回溯也会完善稽核规则。

感谢你的阅读，期待与你再次相遇~



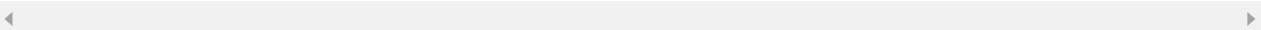
刘明

2020-04-18

数据中台之后，稽核监控应当是比较重要的问题了。但是在任务紧的情况下，这些质量方面的投入往往又会被忽视

展开 ∨

作者回复: 你好，要把数据质量稽核监控，纳入到数据研发的标准流程中，不能因为时间紧张，就忽视了，这样导致交付后会出现更大的问题。我在第12讲流程协作与规范中，会讲到数据研发的标准化流程，期望对你有所帮助。感谢你的阅读~期待与你再次相遇~



崔军伟

2020-04-17

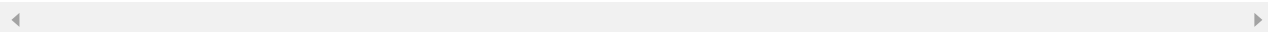
DQC 全链路监控功能不错，还需要在各个节点上增加执行时间和时长，监控各节点执行时长也很重要。

节日促销异常问题：可以将监控的所有指标作为数据，通过机器学习训练出稽核规则模型，这样每天每次任务跑完的指标，由稽核规则模型判别这次任务是否为异常。

展开 ∨

作者回复: 赞，确实基于AI去判断稽核失败到底是正常失败还是异常失败，是一条可行的路，至少可以降低人接入的频率。Good!

感谢你的阅读，期待与你再次相遇~



**volcano**

2020-04-17

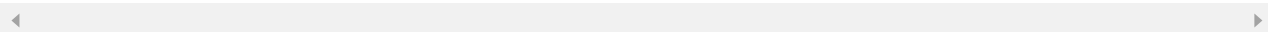
完整性规则中

如何确定业务库数据到ODS数据是不缺失的呢？有没有具体的稽核方案  
业务库是不会允许select count()这种需要大量扫表的行为

展开 ∨

作者回复: ODS 层的数据缺失，可以通过在数据传输工具中做校验来完成，读取了多少行记录，这个在数据传输工具中是可以知道的。

感谢你的阅读~期待与你再次相遇~



**Geek\_09c245**

2020-04-17

总结平常的数据质量问题很到位，对数据治理有帮助，大多数小公司都值得一学

作者回复: 感谢你的认可和鼓励，这确实是在实践过程中总结沉淀出来的一套方法，拿出来分享给大家，希望你有所帮助~

