Week05 | 课时 3 | 口径能不能负责，不看 SQL 多炫：tests、docs、lineage 与变更影响分析

Table of contents

负责的口径必须带证据：测试、文档、血缘和影响面

一个 mart 能跑出来，还不等于可以交付。

没有 tests、docs、lineage 和 impact notes 的指标，不应该进入 Agent 工具或正式看板。

进入课时 4 回看课时 2 返回 Week05 总览

下载讲义

提供适合离线阅读的 PDF 版和适合批注整理的 Word 版。

PDF 版 · 打印 / 离线阅读 Word 版 · 批注 / 二次整理

这节课解决什么问题

真正的问题不是“SQL 能不能跑”，而是：

> **当口径被别人使用、被 Agent 调用、被运营拿去决策时，你能不能解释它为什么可信、改动会影响谁、出了问题怎么定位？**

这节课关注口径证据链。

参考学习时间

建议按一节标准课安排：读完正文后，对照当前项目 analytics/models/marts/schema.yml 和 reports/week05/dbt_build_evidence.md 做一次证据复盘。

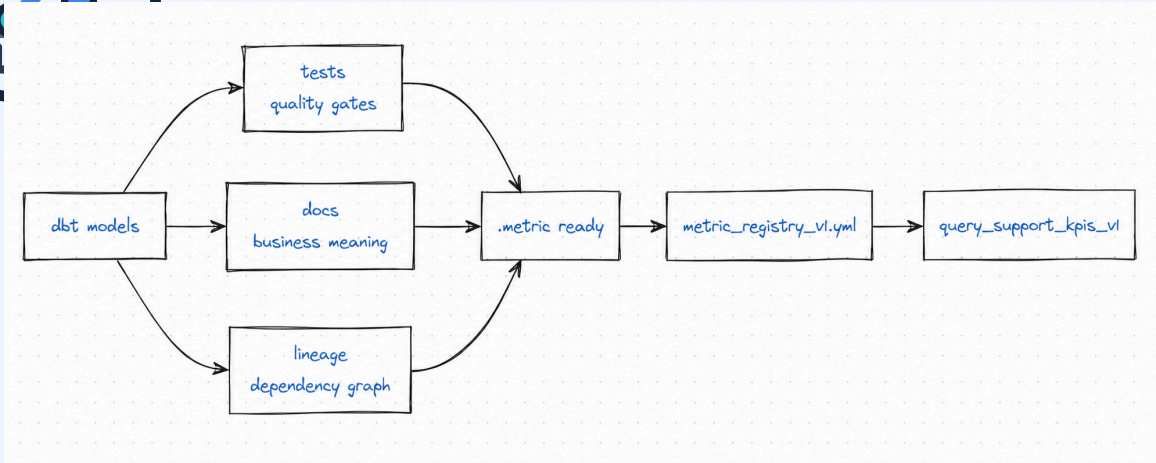
学完这一讲，你应该能做到什么

1. 区分 data tests、singular tests、unit tests、docs、lineage 和 artifacts。
2. 判断哪些测试保护口径，哪些只是形式化存在。
3. 解释 manifest.json、catalog.json、run_results.json、sources.json 的作用。
4. 写出一次口径变更的影响分析。
5. 说明口径进入工具前必须满足哪些准入门槛。

本课产出

- analytics/models/marts/schema.yml
- analytics/tests/no_pii_columns_in_agent_tool_input_view.sql
- analytics/target/manifest.json、catalog.json、run_results.json
- reports/week05/dbt_build_evidence.md
- reports/week05/lineage-impact-notes.md

先看一张总图



这张图说明：口径进入工具，不是从 SQL 直接跳过去，而是要先经过测试、文档、血缘和 registry。

先看一个事故：SQL 改对了，但系统答错了

数据团队发现 first_response_minutes 原来把“第一次任意回复”当成首响，包括系统消息、机器人消息和客户自己的补充说明。于是 SQL 被改成“第一次人工客服回复”。从业务角度看，这个修改是正确的。

但如果没有 docs、lineage 和 impact notes，事故会马上出现：

角色	看到什么	为什么会误判
运营	本周首响时间突然变慢	不知道是服务变差，还是口径从任意回复改成人工回复
BI	dashboard 数字跳变	图表没有同步说明字段定义变化
Agent	继续按旧解释回答	工具或回答模板没有拿到口径变更
数据团队	SQL 已经改对	但不知道下游谁受影响

结论：没有证据链时，正确 SQL 也会造成协作事故。

1. tests 不是为了凑覆盖率

data tests 是对数据结果或 source 假设的断言。它们不是“看起来专业”的装饰，而是在告诉下游：这个字段或模型最低限度没有坏。dbt 内置的 unique、not_null、accepted_values、relationships 是最常见的 data tests。unit tests 则更适合复杂 SQL 逻辑的小样本验证，例如 reopen、escalation、first response 这种边界规则。[1]

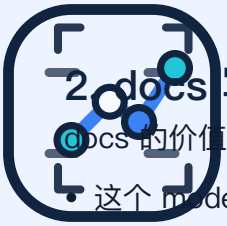
类型	保护什么	什么时候用	Week5 例子	不适合
data test: not_null	关键字段不能为空	字段是口径或 join 的硬前提	metric_name, metric_date, metric_value	可选描述字段
data test: unique	粒度不重复	一行应该唯一代表一个对象	ticket_id in support_case_mart	明细事件表
data test: relationships	外键关系存在	需要证明维表或上游引用可追	org_id 对齐 org dim	无上游维表时硬测
data test: accepted_values	枚举合法	状态、优先级、指标名必须在固定集合内	metric_name, status, priority	高频变化自由文本
singular test	复杂数据断言	内置测试表达不了，需要写 SQL 检查	safe view 不含 PII 字段	简单非空
unit test	SQL 逻辑边界	复杂逻辑要用小样本固定预期	reopen / escalation / first response 规则	替代大规模数据质量测试

当前项目已在 analytics/models/marts/schema.yml 中保护：

- support_case_mart.ticket_id 非空且唯一；
- support_kpi_mart.metric_date、metric_name、metric_value 非空；
- support_kpi_mart.metric_name 只能是 registry 支持的核心指标；
- agent_tool_input_view 的核心输出字段非空。

还单独加了 analytics/tests/no_pii_columns_in_agent_tool_input_view.sql，防止工具视图泄漏 PII 或正文列。

[1] dbt 里 data tests 是对数据集结果的断言；unit tests 是用小输入样本验证 SQL model 逻辑。两者保护的问题不同，不能互相替代。



2. docs 写给未来消费者，不是写给自己

docs 的价值不是生成一个好看的网页，而是让未来消费者知道：

- 这个 model 的粒度是什么；
- 关键字段怎么解释；
- 哪些字段不能被 Agent 查询；
- 某个指标改动会影响哪些下游；
- 当前版本和 release / evidence 怎么绑定。

docs 不是只写给数据团队自己看的。BI 要知道字段能不能直接做维度，运营要知道数字为什么跳变，Agent 工具开发要知道哪些字段能进入 schema，助教和未来的你要知道这个模型为什么这样设计。description、column descriptions、model descriptions 和 source descriptions 都是口径证据。[2]

一个 support_kpi_mart 的字段描述至少应该回答：

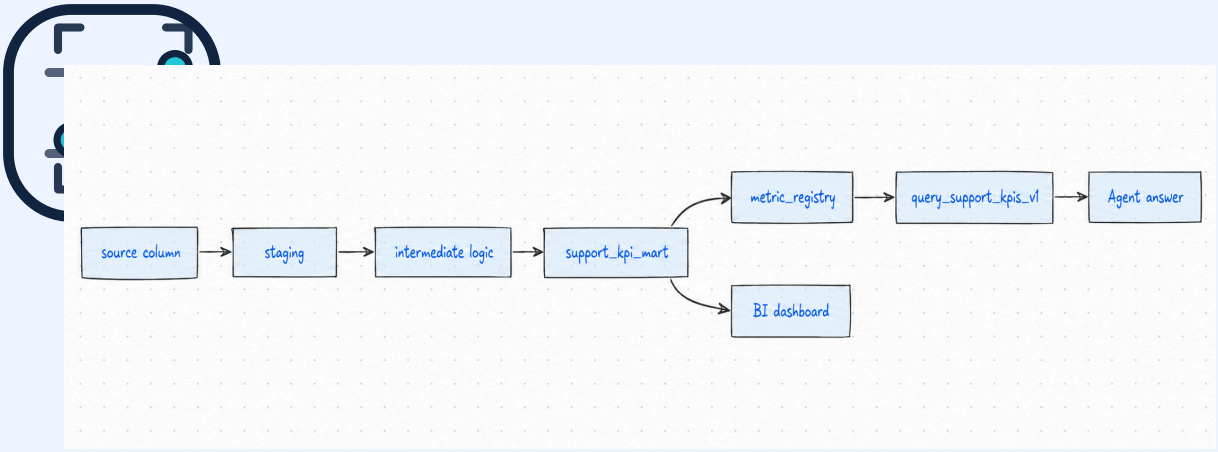
字段	文档里要讲清什么
metric_date	指标归属日期，不等于任何任意更新时间字段
metric_name	必须来自 registry，不允许自由命名
metric_value	聚合后的数值，不能反推 raw record
product_line / priority / org_id / category	允许维度，必须和 registry 白名单一致
data_release_id	用于把查询结果绑定到当前数据 release

3. lineage 是变更影响分析入口

lineage 不只是“看模型图”。它回答的是：

> 如果我改了某个 source 字段或 intermediate 逻辑，谁会被影响?

[2] dbt docs generate 不是形式主义。它把 model、column、source descriptions 和 catalog 信息组织给未来消费者看，是口径可解释性的基础。



如果 first_response_minutes 的计算从“第一次客服回复”改成“第一次人工客服回复”，影响面包括：

- int_support_cases 的派生逻辑；
- support_case_mart 的首响字段；
- 未来如果 avg_first_response_minutes 进入 registry，会影响 support_kpi_mart；
- dashboard 中所有首响指标；
- Agent 对“客服响应是否变慢”的回答；
- docs、tests、registry examples、tool examples；
- reports/week05/lineage-impact-notes.md 中的变更记录。

4. artifacts 是可交付证据

dbt artifacts 不是临时垃圾文件。它们通常不提交到 main，但在实验和作业里要被理解和引用。

artifact	它证明什么	Week05 怎么用
manifest.json	项目资源、依赖、配置和 parent/child map	看模型、source、tests、metrics、parent/child map，支撑 lineage 和 impact notes
catalog.json	warehouse 中的列、类型、表信息	帮助 docs site 展示真实列信息，支撑字段解释
run_results.json	哪些节点跑过、耗时、状态	交付 build evidence、失败复盘和课程验收
sources.json	source freshness 结果	判断上游是否可信或过期

当前项目的证据文件 reports/week05/dbt_build_evidence.md 已记录：

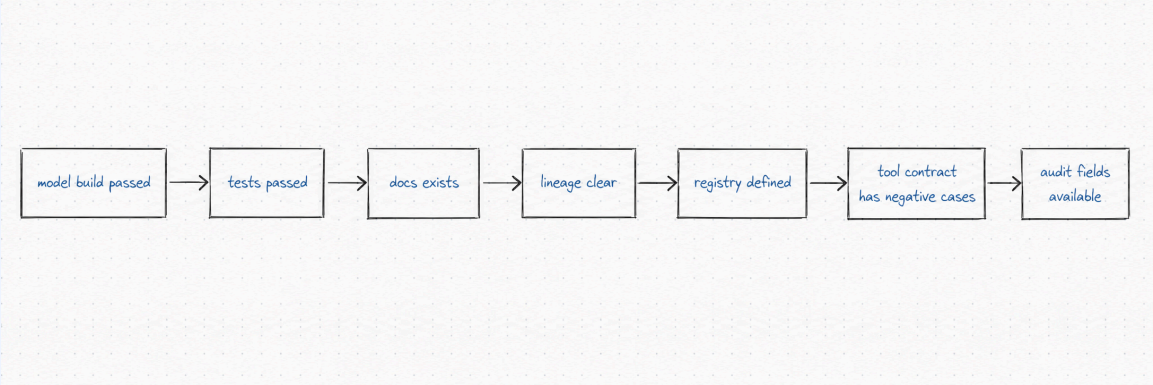
Check	Status
dbt debug	passed
dbt build --select tag:week05	passed: 37/37
dbt docs generate	passed
registry validator	passed: valid=true, metric_count=6
KPI query positive	passed: allowed=true, row_count=16
KPI query negative	passed: METRIC_DENIED, ROLE_DENIED
full pytest	passed: 65 passed, 2 skipped

run_results.json 不是完整 lineage，但它能证明某次命令到底跑了哪些节点、状态如何、失败在哪里。[3]

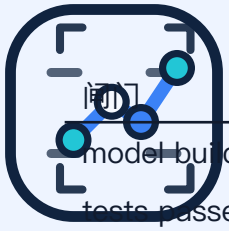
5. 口径进入工具前的准入门槛

一个指标要进入 query_support_kpis_v1，至少满足：

1. dbt build --select tag:week05 通过；
2. 关键字段有 data tests；
3. 复杂逻辑有 singular test、unit test 或样本逻辑测试；
4. schema.yml 有 model / column descriptions；
5. lineage 能追到 sources；
6. metric registry 有稳定定义；
7. tool contract 有白名单和负例；
8. 变更影响分析写入 report；
9. safe view 不暴露 PII、正文和任意 SQL 入口。



[3] run_results.json 记录节点执行状态和耗时，但它不是完整 lineage。lineage 主要依赖 manifest 的依赖关系，run results 更适合作为运行证据。



问	小白版解释
model build passed	模型至少能稳定构建出来
tests passed	关键字段和边界假设有自动检查
docs exists	未来消费者能读懂字段和粒度
lineage clear	改动影响谁可以追踪
registry defined	指标、维度、过滤器、角色已经白名单化
tool contract has negative cases	工具不仅能查成功，也知道哪些请求必须拒绝
audit fields available	后续能解释某次 Agent 答案来自哪次查询

> ! 核心判断
>
> 能跑出来的是 model；能解释、能测试、能复盘、能被工具安全消费的，才是可负责口径。

6. 怎么写 lineage-impact-notes.md

最小模板：

```
## Change

- changed_field:
- changed_logic:
- reason:

## Upstream

- source:
- staging:
- intermediate:

## Downstream impact

- marts:
- metrics:
- dashboards:
- tools:
- evals:

## Required updates
```

```
- docs:
- tests:
- registry:
- tool examples:
- delivery summary:
```

完整示例：

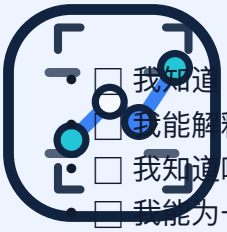
```
## Change

- changed_field: first_response_minutes
- changed_logic: exclude system and bot comments; only count first human support reply
- reason: align first response KPI with support operations definition

## Upstream

- source: omni_postgres.ticket_comment_fact
- staging: stg_ticket_comments
- intermediate: int_support_cases

## Downstream impact

- marts: support_case_mart, support_kpi_mart
- metrics: future avg_first_response_minutes
- dashboards: support response dashboard if metric is enabled
- tools: query_support_kpis_v1 if avg_first_response_minutes enters registry
- evals: any answer quality test that checks response-time explanations

## Required updates

- docs: update model and column descriptions for first_response_minutes
- tests: add sample cases for bot/system/customer vs human support replies
- registry: keep metric disabled until ownership and allowed roles are confirmed
- tool examples: add positive/negative cases only after registry enablement
- delivery summary: explain why historical values may change
```

影响分析要覆盖工具和评测，因为 Week05 的口径不是只给 dashboard 用，后续 Agent 回答、eval 和审计都会消费它。[4]

自检清单

- ☐ 我能说清 data tests 和 docs 分别保护什么。

[4]影响分析只写 dashboard 不够。Week05 的指标会进入 registry、tool contract、Agent answer 和 eval，任何口径变化都可能影响后续工具和评测。



- [ ] 我知道 lineage 不是图，而是变更影响分析入口。
- [ ] 我能解释 manifest.json、catalog.json、run_results.json 的作用。
- [ ] 我知道哪些条件满足后指标才能进入工具。
- [ ] 我能为一个首响口径变化写出影响面。

课后最小行动

选一个指标，写一段影响分析：

- 如果改时间字段，会影响谁?
- 如果改维度白名单，会影响谁?
- 如果把指标开放给 Agent，需要新增哪些 tests、docs、registry 和负例?

延伸阅读

主题	推荐资料	为什么读
dbt data tests	dbt Docs: Data tests	理解数据结果断言如何保护模型假设
dbt unit tests	dbt Docs: Unit tests	理解复杂 SQL 逻辑如何用小样本固定预期
dbt documentation	dbt Docs: Documentation	理解 docs 为什么服务未来消费者
dbt manifest	dbt Docs: manifest.json	理解 lineage 和依赖图的机器可读来源
dbt run results	dbt Docs: run_results.json	理解运行证据如何支撑交付和失败复盘

Footnotes