



## Week06 | 课时 4 | Asset Checks / Metadata / Run

### Evidence: 把“跑过”升级成“可验收”

#### Table of contents

运行完成不是验收，证据完整才是验收 .....	1
这节课解决什么问题 .....	1
参考学习时间 .....	2
学完这一讲，你应该能做到什么 .....	2
本课产出 .....	2
先看质量门禁图 .....	2
1. pytest 和 asset checks 的分工 .....	2
2. Evidence schema 的 required / optional .....	3
3. Optional 依赖不能 fake passed .....	4
4. Metadata 放在哪里 .....	5
5. 最小 evidence 示例 .....	5
自检清单 .....	6
课后最小行动 .....	6

#### 运行完成不是验收，证据完整才是验收

这一讲收紧 Week06 的质量门禁：

asset checks 验证资产状态，run evidence 记录运行事实，二者一起决定下游能不能消费。

[进入课时 5 返回 Week06 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。

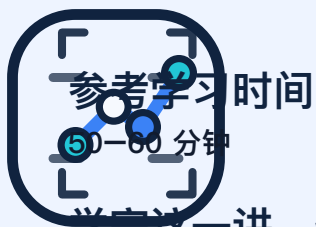
[PDF 版 · 打印 / 离线阅读](#) [Word 版 · 批注 / 二次整理](#)

#### 这节课解决什么问题

Week06 之前，很多系统只能回答“任务跑完了吗”。Week06 之后，我们要回答：

**这个资产在这个分区上，是否有足够证据让下游消费？**

这要求我们把 checks、metadata 和 evidence 分层。



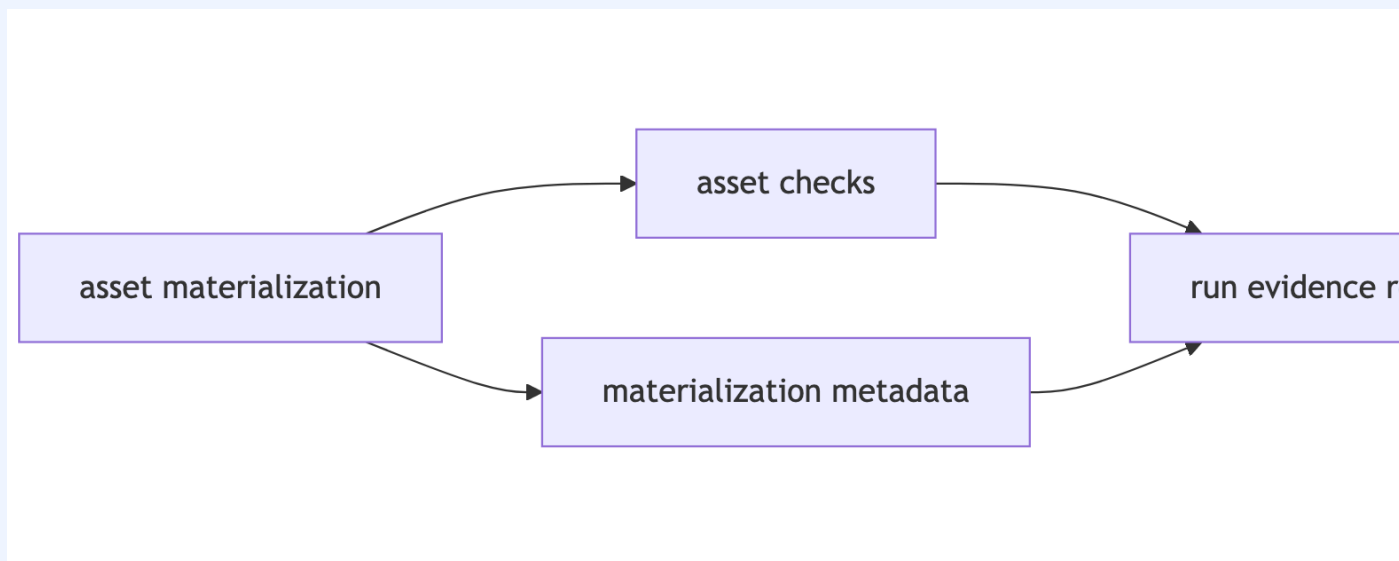
## 学完这一讲，你应该能做到什么

1. 区分 pytest 和 asset checks 的职责。
2. 设计 5 类 Student Core asset checks。
3. 定义 run evidence schema 的 required / optional 字段。
4. 解释 optional Week04 / Week05 依赖为什么要写 skipped。
5. 判断哪些 metadata 放 Dagster，哪些落 report 文件。

## 本课产出

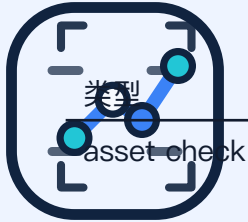
- [contracts/run\\_evidence/week06\\_run\\_evidence.schema.json](#)
- [docs/blueprints/week06/week06-run-evidence-spec.md](#)
- [reports/week06/run\\_evidence/](#)

## 先看质量门禁图



## 1. pytest 和 asset checks 的分工

类型	验什么	例子
pytest	代码逻辑、schema、helper、CLI	evidence schema validation、backfill plan 生成



验什么

例子

某个资产当前数据状态

row count、duplicate、null rate、partition completeness

不要把 asset checks 写成大型数据质量平台。Student Core 先做 5 类：

1. manifest consistency
2. row count
3. duplicate / idempotency
4. required field null rate
5. partition completeness

项目侧当前对应的测试入口是：

```
# 可执行: 已与项目 runbook 对齐
docker compose --profile tools --env-file infra/env/.env.local -f infra/docker-compose.yml run --rm devbox \
  pytest tests/integration/test_week06_asset_checks.py tests/integration/test_week06_run_evidence_generation.py -
q
```

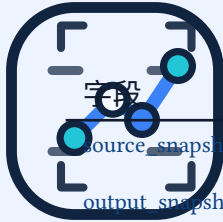
怎么看输出：只要其中一类 check 失败，就不能把 downstream decision 写成 release；如果是 optional 依赖缺失，要写 `skipped/not_available`，不是 `passed`。

## 2. Evidence schema 的 required / optional

必填字段建议：

字段	作用
<code>evidence_schema_version</code>	schema 版本
<code>run_id</code>	本次运行标识
<code>asset_key</code>	资产 key
<code>partition_key</code>	分区
<code>status</code>	<code>passed / failed / skipped / warning</code>
<code>started_at / finished_at</code>	时间边界
<code>report_path</code>	证据文件位置
<code>reason_codes</code>	决策原因

可选字段建议：



	为什么可选
source_snapshot_id	Week04 lakehouse 未启用时可能为空
output_snapshot_id	同上
dbt_invocation_id	Week05 analytics 未启用时可能为空
semantic_metric_count	semantic mart 未启用时可能为空
lakehouse_snapshot_id	Week04 lakehouse 状态未启用时可能为空
data_release_id	课程环境可先用 week06-dev-local
trace_id / git_sha	Week11 / Week14 会进一步使用
downstream_decision	根据 checks 和 evidence 决定是否放行
checks	保存结构化 check 结果

### 3. Optional 依赖不能 fake passed

如果 Week04 或 Week05 尚未完成，应写：

```
{
  "status": "skipped",
  "reason_codes": ["week05_analytics_not_available"],
  "dbt_invocation_id": null,
  "semantic_metric_count": null
}
```

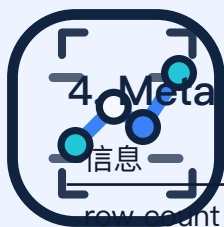
不要为了让图看起来完整，把没有跑过的依赖写成 passed。

同理，Week04 lakehouse 未启用时应写：

```
{
  "status": "skipped",
  "reason_codes": ["week04_lakehouse_not_available"],
  "lakehouse_snapshot_id": null
}
```

⚠️ skipped 是诚实状态，fake passed 是治理事故

Week06 允许 optional 依赖暂不可用，但必须把不可用原因写进 evidence。后续 Week08 / Week11 / Week14 宁愿看到清楚的 skipped，也不能消费一个伪装成 passed 的状态。



## 4. Metadata 放在哪里

	建议位置	理由
row count / partition key	Dagster metadata + report	UI 可见, 报告可归档
sample file path	Dagster metadata	方便演示和定位
完整 checks 结果	report file	避免 UI metadata 过长
failure reason / action	report file	方便 runbook 引用
release / trace / git sha	report file + summary metadata	供后续评测和治理追溯

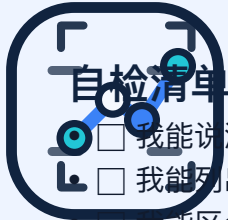
## 5. 最小 evidence 示例

```
{
  "evidence_schema_version": "week06_run_evidence_v1",
  "run_id": "week06::2026-04-17",
  "asset_key": "week06/ops/run_evidence_report",
  "partition_key": "2026-04-17",
  "status": "passed",
  "started_at": "2026-04-17T10:00:00Z",
  "finished_at": "2026-04-17T10:02:30Z",
  "report_path": "reports/week06/run_evidence/week06_ops_run_evidence_report_2026-04-17.json",
  "reason_codes": ["checks_passed"],
  "input_row_count": 3,
  "output_row_count": 3,
  "data_release_id": "week06-dev-local",
  "downstream_decision": "dry_run_only",
  "checks": [
    {"name": "manifest_consistency", "status": "passed"},
    {"name": "duplicate_idempotency", "status": "passed"}
  ]
}
```

常见错误: `evidence_schema_version` 写错、`reason_codes` 不是数组、`report_path` 指向不存在文件, 都会导致 schema validation 失败。

**! 核心判断**

可验收的数据资产必须同时有运行结果、质量判断和可追溯证据。



- 我能分清 pytest 和 asset checks 的分工
- 我能列出 5 类 Student Core checks
  - 我能区分 evidence required 和 optional 字段
  - 我知道 optional 依赖未启用时要写 skipped
  - 我能说明哪些 metadata 适合放在 UI, 哪些应该落 report

## 课后最小行动

补一份 evidence 字段表:

```
## Week06 run evidence fields
```

```
- Required:
```

```
- Optional:
```

```
- Status values:
```

```
- Reason code examples:
```

```
- Downstream decision rule:
```