회차와 수정본을 다른 표로 나눠 연결해봤습니다

지피터스 24기 · 자료 정리 2026-10-14 / 기존 실습 근거 2026-10-13 / 단독 재현 코드는 10-14 실행 확인

회차 하나에 원고가 여러 개 붙는 구조

10월 13일 보존한 연재 준비용 DB의 세 표를 설명합니다. SQLite로 수행한 대체 실습이며 Airtable이나 운영 봇에 연결한 결과는 아닙니다.

표	레코드	역할
episode	3개	회차 ID·제목·독자 질문
revision	6개	회차에 연결된 초안·수정본
action	3개	상태·다음 행동

E02라는 회차 ID에 E02-baseline과 E02-revised가 연결됩니다. 제목과 다음 행동을 바꿔도 회차 ID를 바꾸지 않으면 관계가 유지됩니다.

직접 확인

`python3 bboya-replay.py`를 실행합니다. 표준 sqlite3만 사용하며 메모리 DB에서 보존한 레코드를 재구성합니다. 운영 DB를 수정하지 않습니다. counts가 3/6/3이고 E02의 버전 두 개가 같은 회차를 가리키는지 확인합니다.

이 DB에는 10월 7일 초안·수정본까지만 넣었습니다. 지식관리 예제의 10월 13일 결과 원고는 이 DB에 아직 등록하지 않았습니다. next_action도 보존한 조회 시점 값입니다.

10월 14일 단독 재현 출력


```
{
  "counts": {
    "episode": 3,
    "revision": 6,
    "action": 3
  },
  "E02": [
    {
      "id": "E02",
      "title": "AI 더치페이 도구를 믿어도 될까?",
      "revision_id": "E02-baseline",
      "label": "baseline"
    },
    {
      "id": "E02",
      "title": "AI 더치페이 도구를 믿어도 될까?",
      "revision_id": "E02-revised",
      "label": "revised"
    }
  ]
}
```

단독 재현 코드: bboya-replay.py

긴 줄은 지면에 맞춰 자동 줄바꿈됩니다. 코드 복사 시 줄바꿈·들여쓰기를 확인하세요.

```
"""Rebuild the recorded sample in memory; no live DB is modified."""
import sqlite3, json
SCHEMA = """PRAGMA foreign_keys=ON;
CREATE TABLE IF NOT EXISTS episode(id TEXT PRIMARY KEY,
    title TEXT NOT NULL,
    question TEXT NOT NULL);
CREATE TABLE IF NOT EXISTS revision(id TEXT PRIMARY KEY,
    episode_id TEXT NOT NULL REFERENCES episode(id),
    label TEXT NOT NULL,
    source TEXT NOT NULL);
CREATE TABLE IF NOT EXISTS action(episode_id TEXT PRIMARY KEY REFERENCES episode(id),
    state TEXT NOT NULL CHECK(state IN ('draft',
    'review',
    'approved',
    'published')),
    next_action TEXT NOT NULL,
    evidence TEXT NOT NULL DEFAULT '',
    published_url TEXT NOT NULL DEFAULT '',
    CHECK(state!='published' OR published_url LIKE 'https://%'));
"""
EPISODES = [('E01', 'AI 여행 일정은 실제로 움직일 수 있을까?', '이동·예산·운영시간을 검증할 수 있는가'),
 ('E02', 'AI 더치페이 도구를 믿어도 될까?', '분할 금액 합계가 원래 총액과 같은가'),
 ('E03', 'AI 실습 안내만으로 페이지를 만들 수 있을까?', '초보자가 안내만 보고 이름 변경을 완료하는가')]
ACTIONS = [('E01', 'draft', '출발지와 여행 날짜 확인', '', ''),
 ('E02', 'review', '브라우저 계산 결과를 결과 원고에 반영', '', ''),
 ('E03', 'draft', '사용 운영체제와 시험 참가자 확인', '', '')]
c = sqlite3.connect(':memory:')
c.row_factory = sqlite3.Row
c.executescript(SCHEMA)
c.executemany('INSERT INTO episode VALUES(?,?,?)', EPISODES)
c.executemany('INSERT INTO action VALUES(?,?,?,?,?)', ACTIONS)
for i in range(1,4):
    for label in ['baseline','revised']:
        ident = f'E0{i}'
        source = f'../knowledge/source/{label}-{i:02d}.md'
        c.execute('INSERT INTO revision VALUES(?,?,?,?)',
            (ident+'-'+label, ident, label, source))
counts = {t:c.execute('SELECT COUNT(*) FROM '+t).fetchone()[0]
          for t in ['episode','revision','action']}
assert counts == {'episode':3,'revision':6,'action':3}
linked = [dict(r) for r in c.execute(
    "SELECT e.id,e.title,r.id AS revision_id,r.label "
    "FROM episode e JOIN revision r ON e.id=r.episode_id "
    "WHERE e.id='E02'")]
print(json.dumps({'counts':counts,'E02':linked},
                 ensure_ascii=False,indent=2))
```