

## WMS-DU Step6 调试与补丁全记录 (Canvas)

### 阶段目标

完成 Step5 后, 进入 Step6 调试: 1. 扩展数据库体检脚本 (索引/唯一约束/列序); 2. Putaway & Inbound 性能与日志完善; 3. 设计与实现周期快照 StockSnapshot 逻辑; 4. 确保四大验证测试 (入库幂等、Putaway、快照幂等、快照回灌) 全绿。

---

### 一、迁移问题与解决

#### 问题 1: Multiple head revisions

alembic upgrade head 报 multiple head。

解决: 创建 merge revision 6e6459c3169f, 合并多头 (add\_stock\_snapshots, perf\_indexes, uq\_batches\_composite)。

#### 问题 2: CREATE INDEX CONCURRENTLY 报错

PG 不允许 CONCURRENTLY 在事务块执行。

解决: 在 20251014\_perf\_indexes.py 迁移中使用 with context.autocommit\_block()。

#### 问题 3: UndefinedTable / UndefinedColumn

ledger 实际表名 stock\_ledger; columns 为 stock\_id, occurred\_at。

解决: 修正索引定义, 建立 (stock\_id, occurred\_at) 复合索引。

#### 问题 4: DuplicateTable

已存在 stock\_snapshots 表。

解决: 增加 \_has\_table 检查, 幂等建表。

#### 问题 5: as\_of\_ts / snapshot\_date 不一致

快照表列名不同。

解决: 迁移及脚本自适应列名; 最终统一 snapshot\_date。

#### 问题 6: qty / qty\_on\_hand 差异

schema 不同步。

解决: 自适应写入逻辑, 探测列名动态使用 qty 或 qty\_on\_hand。

#### 问题 7: warehouse\_id not null

快照插入遗漏 warehouse\_id。

解决：join locations 补 warehouse\_id。

---

## 二、体检修复

运行：

```
python scripts/pg_healthcheck.py --dsn postgresql://wms:wms@127.0.0.1:5433/wms
--spec scripts/db_spec.json
```

初期缺失约束：uq\_stocks\_item\_loc, uq\_batches\_item\_batch, 缺少表名对齐。

通过新增迁移 20251014\_fix\_uniques\_for\_stocks\_batches.py 修复，最终体检绿灯。

---

## 三、测试调试阶段

### 1) Inbound 幂等

- 修复 fixture：pytest\_asyncio.fixture 替换 pytest.fixture；
- 去掉重复事务 (async with session.begin() 冲突)；
- 修正 Location 模型无 code 字段；
- 最终状态：✅通过。

### 2) Putaway 出入对称

- 调整测试逻辑：不再使用 code 列；
- 兼容无 updated\_at 的 stocks 表；
- 最终状态：✅通过。

### 3) 快照幂等

- 多次迭代修正：
- 去除多命令 prepared 语句（拆为两步 delete+insert）；
- 兼容 snapshot\_date / warehouse\_id / qty\_on\_hand；
- 最终作业改为当日增量（不再累计 base）。
- 最终状态：✅通过。

### 4) 快照回灌

- 调整作业为仅算当日窗口 (prev\_end, cut\_end]；
- 结果符合预期：T-1=2, T=5。
- 最终状态：✅通过。

#### 四、四大补丁最终验证命令


```
# 0) 升级 + 体检
alembic upgrade head
python scripts/pg_healthcheck.py --dsn postgresql://wms:wms@127.0.0.1:5433/wms
--spec scripts/db_spec.json

# 1) 入库幂等
PYTHONPATH=. pytest -q tests/quick/test_inbound_idempotent_pg.py -s

# 2) Putaway 出入对称
PYTHONPATH=. pytest -q tests/quick/test_putaway_move_pg.py -s

# 3) 快照幂等
PYTHONPATH=. pytest -q tests/quick/test_stock_snapshot_pg.py -s

# 4) 快照回灌
PYTHONPATH=. pytest -q tests/quick/test_stock_snapshot_backfill_pg.py -s
```

当前结果：四项均绿 

#### 五、下一步

1. 在 CI 中增加独立 Schema 体检 job：

```
jobs:
  pg-check:
    name: Schema Healthcheck (PG)
    runs-on: ubuntu-latest
    continue-on-error: true
    steps:
      - uses: actions/checkout@v4
      - name: Run PG Healthcheck
        run: |
          python scripts/pg_healthcheck.py --dsn "$DATABASE_URL" --spec
scripts/db_spec.json
```

2. 等稳定后设为 Required Check。
3. 归档 Step6 canvas → 060\_SnapshotService\_v1\_CI体检全绿.md。

**结论：** Step6 四大补丁全部验证通过，PG 体检绿灯，快照逻辑与仓储模块幂等性校验完备。项目可安全进入下一阶段 CI 集成与性能监控。