خط لوله یکپارچه‌سازی داده‌های موجودی کالاهای مصرفی سریع‌گردش (FMCG)

گزارش پروژه

Python · Pandas · SQLite — یک پروژه خودآموز ده‌روزه

تهیه‌شده برای: امیررضا باباحمدی

تاریخ داده‌ها: ۱ سپتامبر ۲۰۲۶

تاریخ گزارش: ۵ سپتامبر ۲۰۲۶

فهرست مطالب

۱. خلاصه مدیریتی

۲. مسئله کسب‌وکار و اهداف

۳. منابع داده

۴. معماری خط لوله

۵. یافته‌های کیفیت داده (Q1)

۶. اعتبارسنجی قوانین کسب‌وکار (Q2)

۷. استانداردسازی و تبدیل داده (Q3)

۸. یکپارچه‌سازی داده (Q4)

۹. نمای تحلیلی آماده کسب‌وکار (Q5)

۱۰. پرچم‌های عملیاتی (Q6)

۱۱. بارگذاری و بازتولیدپذیری (Q7)

۱۲. ثبت رخداد و مسیر ممیزی (Q8)

۱۳. محدودیت‌ها

۱۴. نتیجه‌گیری و پیشنهادها

۱. خلاصه مدیریتی

این پروژه یک خط لوله ETL کامل و بازتولیدپذیر برای داده‌های فروش، سفارش و موجودی یک توزیع‌کننده منطقه‌ای کالاهای مصرفی سریع‌گردش (FMCG) می‌سازد. چهار فایل خام CSV — فروش، سفارش‌ها، عکس‌های لحظه‌ای موجودی و فایل مرجع انبارها — پروفایل‌بندی، در برابر هشت قانون صریح کسب‌وکار اعتبارسنجی، پاک‌سازی و استانداردسازی، به‌صورت ایمن در یک جدول تحلیلی آماده کسب‌وکار یکپارچه، از نظر چهار وضعیت ریسک عملیاتی بررسی و در نهایت در یک پایگاه‌داده SQLite بارگذاری شدند. هر مرحله برای ممیزی ثبت شده است.

این خط لوله ۱۸۷۷ ردیف خام فروش، ۷۶۷ ردیف خام سفارش و ۳۴۷۰ ردیف خام عکس لحظه‌ای موجودی را در ۸ انبار پردازش کرد. پس از اعتبارسنجی و پاک‌سازی، ۱۴۲۷ ردیف فروش، ۶۳۳ ردیف سفارش و ۲۹۳۳ ردیف موجودی بارگذاری شدند که نمای آماده کسب‌وکاری شامل ۳۶۰ ترکیب کالا/انبار را پدید آوردند. تعداد ردیف‌های هر جدول SQLite در برابر دیتافریم منبع آن راستی‌آزمایی شد و هیچ ردیفی از جداول واقعیت به انباری اشاره نمی‌کند که در فایل مرجع انبارها وجود نداشته باشد.

یافته‌های کلیدی: ۳۶.۹٪ از ترکیب‌های کالا/انبار نسبت به سرعت فروش اخیر موجودی مازاد نشان می‌دهند، ۱۳.۱٪ در ۳۰ روز گذشته هیچ فروشی نداشته‌اند (موجودی راکد) و ۶.۴٪ مقدار سفارش باز بیشتری از موجودی فعلی دارند. این موارد برای بررسی کسب‌وکاری پرچم‌گذاری شده‌اند، نه برای اقدام خودکار — به بخش ۱۰ مراجعه کنید.

۲. مسئله کسب‌وکار و اهداف

یک توزیع‌کننده، داده‌های فروش، سفارش خرید و موجودی انبار را در سه سیستم جداگانه نگهداری می‌کند که با زمان‌بندی‌های متفاوت و بدون استاندارد مشترک برای ورود داده، فایل‌های مسطح (flat file) خروجی می‌گیرند. پیش از پاسخ به سؤالات عملیاتی پایه — کدام کالاها در آستانه اتمام موجودی هستند، کدام انبارها بیش از حد موجودی دارند، آیا سفارش‌های باز برای پوشش تقاضای اخیر کافی هستند — باید این چهار منبع پروفایل‌بندی، به یک استاندارد مشترک پاک‌سازی، به‌صورت ایمن ترکیب و از نظر سازگاری درونی بررسی شوند.

این پروژه به هشت پرسش مشخص، با تکیه بر شواهد و نه فرض، پاسخ می‌دهد:

- Q1 — آیا می‌توان به داده‌های خام اعتماد کرد و دقیقاً کجای آن‌ها مشکل دارد؟
- Q2 — کدام رکوردها قوانین داده‌ای خود کسب‌وکار را نقض می‌کنند؟
- Q3 — آیا چهار فایل بدون ویرایش دستی در صفحه‌گسترده می‌توانند به‌صورت خودکار استاندارد شوند؟
- Q4 — آیا منابع بدون از دست رفتن یا تکرار پنهانِ رکوردها قابل ادغام‌اند؟
- Q5 — یک نمای واحد و آماده کسب‌وکار از موجودی چه شکلی دارد؟
- Q6 — کدام ترکیب‌های کالا/انبار همین حالا نیاز به توجه عملیاتی دارند؟
- Q7 — آیا بارگذاری در SQLite بازتولیدپذیر و از نظر درونی سازگار است؟
- Q8 — آیا مسیر ممیزی کاملی از رخدادها و زمان آن‌ها وجود دارد؟

۳. منابع داده

برای این پروژه هیچ خروجی واقعی از سامانه تولید در دسترس نبود؛ بنابراین یک مجموعه داده مصنوعی اما واقع‌گرایانه تولید شد (اسکریپت scripts_generate_raw_data.py، با بذر تصادفی numpy برای بازتولیدپذیری کامل). تولیدکننده رویکردی ترکیبی را دنبال می‌کند: ساختارهای جدولی معقول برای تراکنش/سفارش/موجودی همراه با یک شبکه توزیع شبیه‌سازی‌شده هشت‌انباره، که عمداً با همان دسته‌های مشکلاتی که یک خروجی واقعی دارد آلوده شده است؛ به این ترتیب خط لوله چیزی واقعی برای پاک‌سازی دارد، نه یک مجموعه داده تمیز و ساختگی.

مشکلات عمداً واردشده:

- سرستون‌های ناهماهنگ در فایل‌ها ("SKU"، "Warehouse_ID"، "OrderedQty"، "SnapshotDate" و...)
- فرمت‌های تاریخ مختلط (ISO، آمریکایی، "Jan-2026-15") و تاریخ‌های کاملاً خالی
- کدهای کالا (SKU) و مقادیر خالی
- مقادیر منفی بدون علامت‌گذاری — سامانه‌های منبع هیچ نشانگری برای «این یک مرجوعی مستند است» ندارند
- شناسه انبار با نگارش/بزرگی‌وکوچکی حروف ناهماهنگ ("WH_05"، "wh-01")
- شناسه‌های انباری که اصلاً وجود ندارند (WH-99) یا در تراکنش‌ها هستند اما در فایل مرجع انبارها نیستند (WH-09) — یک کلید خارجی آویزان
- ردیف‌های کاملاً تکراری و کلیدهای اصلی تکراری (تراکنش‌ها/سفارش‌های دوباره‌ارسال‌شده)
- متن وضعیت سفارش با غلط تایپی و بزرگی‌وکوچکی ناهماهنگ ("Shiped"، "PENDING"، ""cancelled")

۴. معماری خط لوله

خط لوله در پنج ماژول زیرِ پوشه src/ سازمان‌دهی شده که به ترتیبی ثابت توسط src/main.py اجرا می‌شوند:

- extract.py — چهار فایل CSV خام را دقیقاً به همان شکلی که دریافت شده‌اند می‌خواند
- profile.py — پروفایل کامل کیفیت داده برای هر فایل خام (Q1)
- transform.py — استانداردسازی، قوانین اعتبارسنجی، ادغام‌های ایمن، نمای آماده کسب‌وکار و پرچم‌های عملیاتی (Q2 تا Q6)

- load.py — پنج جدول را در SQLite می‌نویسد و بارگذاری را راستی‌آزمایی می‌کند (Q7)
- main.py — تمام مراحل بالا را هماهنگ کرده و لاگ‌ها، نمودارها و خلاصه کیفیت را می‌نویسد (Q8)

هر فایل دقیقاً از همان توالی ثابت عبور می‌کند: خواندن خام ← استانداردسازی نام ستون‌ها ← تبدیل نوع داده (تاریخ، عدد) ← اجرای قانون کلید خارجی انبار ← اعمال باقی قوانین اعتبارسنجی ← حذف تکراری‌های کامل ← حذف تکراری بر اساس کلید اصلی. استفاده مجدد از یک مجموعه توابع تبدیل در هر چهار فایل — به‌جای نوشتن کد پاک‌سازی اختصاصی برای هر فایل — همان چیزی است که خط لوله را قابل نگهداری و رفتار آن را قابل پیش‌بینی می‌کند.

۵. یافته‌های کیفیت داده (Q1)

هر فایل خام پیش از هرگونه پاک‌سازی پروفایل‌بندی شد: تعداد ردیف/ستون، نام دقیق ستون‌ها همان‌گونه که دریافت شده، درصد مقادیر گمشده، ردیف‌های کاملاً تکراری، کلیدهای اصلی تکراری، تاریخ‌های غیرقابل‌تجزیه، مقادیر منفی در ستون‌های مقداری، کدهای کالای خالی و شناسه‌های انباری که در فایل مرجع انبارها نیستند. جزئیات کامل در logs/pipeline.log ثبت شده؛ اعداد کلیدی:

فایل	تعداد ردیف	تکراری کامل	SKU خالی/گمشده	تاریخ غیرقابل‌تجزیه	مقدار منفی
sales.csv	۱۸۷۷	۲۷	۲۴	۳۲۹	۳۵
orders.csv	۷۶۷	۷	۰	۹۵	۲۰
inventory.csv	۳۴۷۰	۲۵	۰	۳۵۵	۱۱۸
warehouses.csv	۸	۰	—	—	—

جدول ۱. پروفایل کیفیت داده خام، پیش از هرگونه پاک‌سازی.

ناهنجاری‌های شناسه انبار (مقادیری که به هیچ ردیفی در warehouses.csv نگاشت نمی‌شوند) نیز برای هر فایل شمارش شد: ۲۷ مورد در فروش، ۱۴ مورد در سفارش‌ها و ۶۹ مورد در موجودی. این موارد در بخش بعد تحت قانون R2 رسیدگی می‌شوند، نه اینکه به‌صورت موردی حذف شوند.

سرستون‌های دریافتی نیز مستقیماً مشکل استانداردسازی را نشان می‌دهند: فایل sales.csv با سرستون‌های "SaleDate"، "SKU " و "Warehouse_ID" در کنار هم دریافت شد — سه قرارداد متفاوت فاصله‌گذاری و بزرگی‌وکوچکی حروف در یک فایل شش‌ستونی. این دقیقاً همان چیزی است که بخش ۷ (Q3) به‌صورت خودکار استاندارد می‌کند.

۶. اعتبارسنجی قوانین کسب‌وکار (Q2)

هشت قانون پیش از نوشتن هر کد پاک‌سازی تعریف شدند تا هر تصمیم حذف یا پرچم‌گذاری در ادامه مسیر، به یک قانون صریح و مستند بازگردد، نه یک قضاوت موردی هنگام نگاه‌کردن به داده:

قانون	بیانیه	نحوه اجرا
R1	کد کالا (SKU) نباید خالی باشد	ردیف‌ها حذف می‌شوند؛ تعداد ثبت می‌شود
R2	warehouse_id باید در جدول مرجع انبارها وجود داشته باشد	ردیف‌ها به‌جای حذف بی‌صدا، در data/processed/quarantine_*.csv قرنطینه می‌شوند
R3	ستون‌های تاریخ باید قابل تجزیه باشند	مقادیر غیرقابل‌تجزیه با pd.to_datetime(errors='coerce') خالی می‌شوند؛ ردیف‌های با تاریخ الزامی خالی حذف می‌شوند
R4	ستون‌های مقداری باید عددی باشند	مقادیر غیرعددی با pd.to_numeric(errors='coerce') خالی می‌شوند
R5	مقدار منفی فقط در صورت علامت‌گذاری به‌عنوان مرجوعی مستند معتبر است	چون هیچ پرچم مرجوعی در داده منبع نیست، هر مقدار منفی خطا در نظر گرفته و حذف می‌شود
R6	on_hand_qty نمی‌تواند منفی باشد	ردیف‌ها حذف می‌شوند؛ تعداد ثبت می‌شود
R7	کلیدهای اصلی نباید تکرار شوند	transaction_id / order_id با حفظ نخستین مورد از تکرار حذف می‌شوند
R8	ستون‌های الزامی نباید کاملاً خالی باشند	در مرحله پروفایل‌بندی بررسی می‌شود (بخش ۵)

جدول ۲. قوانین کسب‌وکاری اعتبارسنجی‌شده در src/transform.py.

قانون R2 نیازمند توضیح جداگانه است: به‌جای حذف بی‌صدای ردیف‌های دارای شناسه انبار ناشناخته، خط لوله آن‌ها را به یک فایل قرنطینه به ازای هر منبع هدایت می‌کند (برای نمونه data/processed/quarantine_sales_bad_warehouse_id.csv) و تمام ستون‌های اصلی را دست‌نخورده نگه می‌دارد تا یک مسئول داده بتواند آن‌ها را بررسی و در صورت لزوم اصلاح و دوباره وارد کند. هیچ‌چیز بدون ردپا ناپدید نمی‌شود.

۷. استانداردسازی و تبدیل داده (Q3)

شش تابع قابل‌استفاده‌مجدد فقط یک‌بار در src/transform.py نوشته و به‌طور یکسان روی همه فایل‌ها اعمال می‌شوند تا هر اصلاح یا بهبود فقط در یک نقطه لازم باشد:

- standardize_column_names — کوچک‌کردن حروف، حذف فاصله‌های اضافه، تبدیل فاصله‌های داخلی به زیرخط، حذف نویسه‌های عجیب
- normalize_ids — بزرگ‌کردن حروف، حذف فاصله، یکسان‌سازی جداکننده‌ها به خط تیره (به این ترتیب " WH-01 "، " wh-01 " و "WH_05" همگی به یک شکل استاندارد واحد می‌رسند)
- parse_dates — استفاده از pd.to_datetime(..., errors='coerce') به‌طور یکسان صرف‌نظر از ترکیب فرمت منبع
- to_numeric_safe — استفاده از pd.to_numeric(..., errors='coerce') برای هر ستون شبه‌مقداری
- clean_text — کوتاه‌کردن و کوچک‌کردن فیلدهای متنی آزاد (وضعیت سفارش)، همراه با یک نگاشت کوچک غلط تایپی ("shiped" → "shipped") پیش از اعتبارسنجی در برابر مجموعه وضعیت‌های استاندارد
- drop_exact_duplicates — حذف ردیف‌های کاملاً یکسان، با ثبت تعداد برای هر فایل

اعمال همان شش تابع در همه‌جا و به همان ترتیب، چیزی است که سه فایل CSV با فرمت‌بندی ناهماهنگ را به جداولی تبدیل می‌کند که می‌توان با اطمینان آن‌ها را ادغام کرد — جایگزین آن (نوشتن اسکریپت‌های پاک‌سازی اختصاصی برای هر فایل) نه مقیاس‌پذیر است و نه به‌راحتی قابل ممیزی.

۸. یکپارچه‌سازی داده (Q4)

دانه داده (grain) پیش از نوشتن هر ادغام مشخص شد: یک ردیف به ازای هر (sku, warehouse_id) برای نمای آماده کسب‌وکار، و یک ردیف به ازای هر تراکنش/سفارش برای جداول واقعیت. همه ادغام‌ها از نوع 'how='left با indicator=True هستند؛ تعداد ردیف‌ها پیش و پس از هر ادغام ثبت می‌شود و ردیف‌های نامنطبق به‌جای حذف بی‌صدا، جداگانه ثبت می‌شوند.

ادغام	ردیف پیش از ادغام	ردیف پس از ادغام	نامنطبق	تفسیر
تحلیلی ← موجودی	۳۶۰	۳۶۰	۰	هر ترکیب کالا/انبار یک عکس لحظه‌ای موجودی جاری دارد
تحلیلی ← فروش	۳۶۰	۳۶۰	۴۷	بدون فعالیت فروش در ۳۰ روز گذشته — بی‌ضرر، نه یک خطا
تحلیلی ← سفارش	۳۶۰	۳۶۰	۲۱۴	بدون سفارش باز در ۳۰ روز گذشته — بی‌ضرر، نه یک خطا
تحلیلی ← انبارها	۳۶۰	۳۶۰	۰	صفر ناهنجاری، چون قانون R2 پیش از این مرحله شناسه‌های انباری نامعتبر را قرنطینه کرده است

جدول ۳. حسابرسی ادغام برای نمای تحلیلی آماده کسب‌وکار.

ثابت ماندن تعداد ردیف روی ۳۶۰ پیش و پس از هر ادغام، نبود fan-out (بزرگ‌شدن ناخواسته تعداد ردیف در اثر ادغام یک‌به‌چند) را تأیید می‌کند — بررسی‌ای که خط لوله به‌صورت خودکار انجام می‌دهد و در صورت شکست، هشدار صادر می‌کند.

۹. نمای تحلیلی آماده کسب‌وکار (Q5)

جدول analytic_sku_warehouse یک ردیف به ازای هر (sku, warehouse_id) دارد — در اجرای حاضر ۳۶۰ ردیف — با ستون‌های زیر:

- on_hand_qty — آخرین عکس لحظه‌ای موجودی برای آن کالا/انبار
- recent_sales_qty، last_sale_date — حجم فروش ۳۰ روز گذشته و آخرین فروش
- recent_orders_qty، last_order_date — حجم سفارش باز ۳۰ روز گذشته (به‌جز سفارش‌های لغوشده) و آخرین سفارش

- warehouse_name، city، region — پیوندشده از فایل مرجع انبارها
- چهار ستون پرچم عملیاتی — به بخش ۱۰ مراجعه کنید

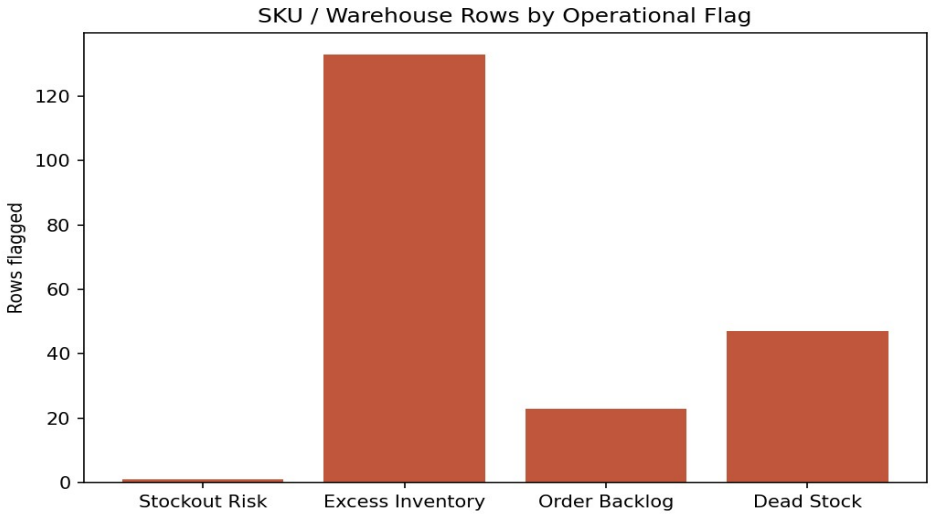
این همان جدولی است که یک مدیر دسته‌بندی یا تحلیل‌گر عملیات، روزانه واقعاً از آن پرس‌وجو می‌کند، به‌جای اینکه هر بار سه جدول واقعیت خام را دستی ادغام کند.

۱۰. پرچم‌های عملیاتی (Q6)

چهار پرچم ساده و بدون بهینه‌سازی مستقیماً روی نمای آماده کسب‌وکار محاسبه شدند، با اولویت عمدی شفافیت بر پیچیدگی در این نخستین گذر:

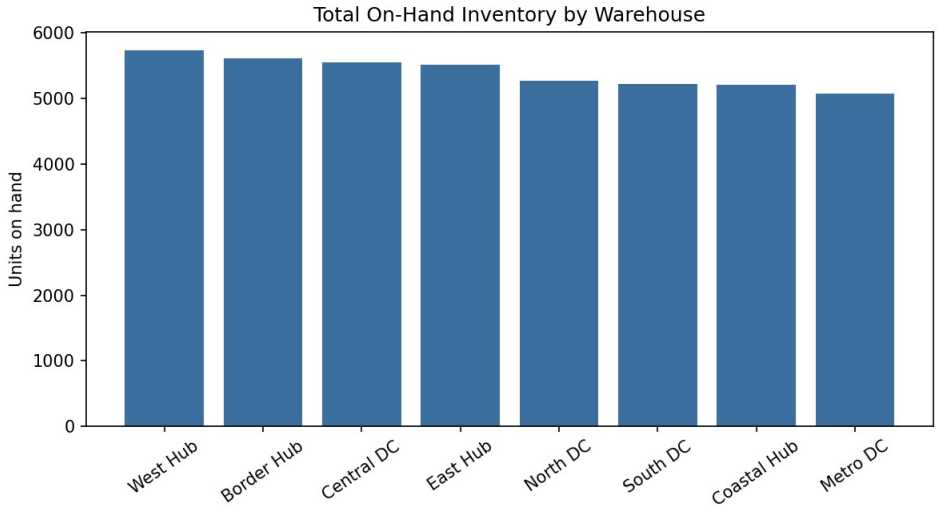
پرچم	تعریف	تعداد ردیف	درصد از ۳۶۰
ریسک اتمام موجودی (F1)	موجودی فعلی ≤ ۰ درحالی‌که تقاضای فروش اخیر وجود داشته	۱	۰.۳٪
موجودی مازاد (F2)	موجودی فعلی بیش از ۴ برابر مقدار فروش اخیر است	۱۳۳	۳۶.۹٪
تأخیر در تکمیل سفارش (F3)	مقدار سفارش باز از موجودی فعلی بیشتر است	۲۳	۶.۴٪
موجودی راکد (F4)	موجودی وجود دارد اما در ۳۰ روز گذشته هیچ فروشی ثبت نشده	۴۷	۱۳.۱٪

جدول ۴. پرچم‌های عملیاتی محاسبه‌شده در src/transform.add_operational_flags().

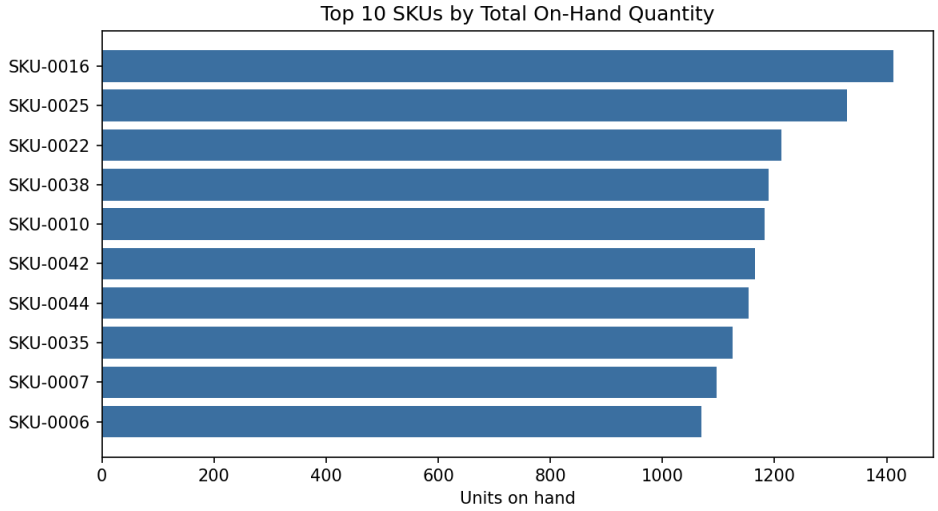


شکل ۱. تعداد ردیف‌های کالا/انبار به تفکیک نوع پرچم.

موجودی مازاد با ۳۶.۹٪ از ترکیب‌های کالا/انبار، بزرگ‌ترین پرچم منفرد است — سیگنالی معقول برای این نخستین گذر با آستانه ۴ برابر، اما باید پیش از استفاده در تصمیم‌گیری، در برابر اهداف واقعی سطح خدمت بازبینی شود (به بخش ۱۳، محدودیت‌ها مراجعه کنید). تنها یک ترکیب ریسک آشکار اتمام موجودی نشان می‌دهد و ۱۳.۱٪ در ۳۰ روز گذشته هیچ حرکت فروشی نداشته‌اند.



شکل ۲. مجموع موجودی فعلی به تفکیک انبار.



شکل ۳. ده کالای برتر بر اساس مجموع موجودی فعلی.

۱۱. بارگذاری و بازتولیدپذیری (Q7)

پنج جدول با 'if_exists='replace در database/inventory.db نوشته شدند: dim_warehouses، fact_inventory، fact_orders، fact_sales و analytic_sku_warehouse. پس از بارگذاری، خط لوله به‌جای فرض موفقیت‌آمیز بودن نوشتن، بررسی‌های کیفیت خود را اجرا می‌کند:

جدول	ردیف در دیتافریم	ردیف در SQLite	تطابق
dim_warehouses	۸	۸	بله
fact_sales	۱۴۲۷	۱۴۲۷	بله
fact_orders	۶۳۳	۶۳۳	بله
fact_inventory	۲۹۳۳	۲۹۳۳	بله
analytic_sku_warehouse	۳۶۰	۳۶۰	بله

جدول ۵. راستی‌آزمایی تعداد ردیف پس از بارگذاری (دیتافریم در برابر SELECT COUNT(*)).

بررسی دوم، یکپارچگی ارجاعی را سراسر مسیر تأیید می‌کند: ۰ ردیف در fact_sales به شناسه انباری اشاره می‌کند که در dim_warehouses وجود ندارد (از مجموع ۴۹۹۳ ردیف در سه جدول واقعیت).

خط لوله تنها از مسیرهای نسبی (pathlib) استفاده می‌کند، با یک دستور واحد (python -m src.main) سرتاسر اجرا می‌شود و ایدمپوتنت است — اجرای دوباره آن از یک /data/processed/ و /database/ تمیز، همان تعداد ردیف و همان تعداد پرچم را تولید می‌کند، چون تولیدکننده منبع دارای بذر تصادفی ثابت است و هر تبدیل قطعی (deterministic) است.

۱۲. ثبت رخداد و مسیر ممیزی (Q8)

logs/pipeline.log در هر اجرا بازنویسی می‌شود (نه افزوده) و با یک برچسب زمانی روی هر خط ثبت می‌کند: زمان شروع و پایان خط لوله؛ تعداد ردیف خوانده‌شده از هر فایل خام؛ هر آماره پروفایل‌بندی از بخش ۵؛ هر نقض قانون اعتبارسنجی و اقدام قرنطینه از بخش ۶؛ تعداد ردیف‌های حذف‌شده در پاک‌سازی و حذف تکراری کامل؛ تعداد پیش/پس/نامنطبق هر ادغام از بخش ۸؛ تعداد هر پرچم عملیاتی از بخش ۱۰؛ و نتایج راستی‌آزمایی پس از بارگذاری از بخش ۱۱. هر استثنای مدیریت‌نشده به‌جای شکست بی‌صدا، به همراه ردپای کامل آن ثبت می‌شود.

همین موضوع، فایل لاگ را به یک پاسخ کامل و زمانی به این پرسش تبدیل می‌کند که «خط لوله در این اجرا واقعاً چه کاری انجام داد» — که هم برای اشکال‌زدایی و هم برای اثبات به یک بازبین که هیچ مرحله‌ای نادیده گرفته نشده، مفید است.

۱۳. محدودیت‌ها

- داده زیربنایی مصنوعی است، نه یک خروجی واقعی از سامانه تولید؛ مقادیر مطلق (واحد، درآمد) نمایشی هستند، نه ارقام واقعی فروش.

- هیچ سامانه منبعی نشانگر «این مقدار منفی یک مرجوعی مستند است» ندارد؛ بنابراین قانون R5 هر مقدار منفی را خطای ورود داده در نظر می‌گیرد. یک سامانه منبع واقعی باید پرچم مرجوعی/بازپرداخت داشته باشد که باید به‌جای حذف کلی، رعایت شود.
- آستانه‌های موجودی مازاد (۴ برابر) و تأخیر در تکمیل سفارش (بیش از موجودی فعلی) در بخش ۱۰، نقطه شروع معقولی برای این نخستین گذر هستند، نه آستانه‌هایی تنظیم‌شده در برابر توافق‌نامه‌های سطح خدمت واقعی — باید پیش از استفاده در تصمیمات خرید یا تخصیص، با کسب‌وکار بازبینی شوند.
- بازه زمانی اخیر به‌صورت ثابت ۳۰ روز از میان ۷۰ روز تاریخچه تعریف شده؛ هر دو در src/main.py قابل تنظیم‌اند و الزام کسب‌وکاری سخت نیستند.
- ردیف‌های قرنطینه‌شده (R2) برای بررسی روی دیسک نوشته می‌شوند اما به‌صورت خودکار اصلاح یا دوباره وارد نمی‌شوند — این مرحله عمداً یک تصمیم انسانی است.

۱۴. نتیجه‌گیری و پیشنهادها

خط لوله هر هشت هدف بیان‌شده در بخش ۲ را برآورده می‌کند: کیفیت داده خام به‌طور کامل پروفایل‌بندی و مستند شده، قوانین کسب‌وکاری صریح و همراه با مسیر ممیزی اجرا شده‌اند، چهار منبع بدون از دست رفتن بی‌صدای داده استاندارد و ادغام شده‌اند، یک نمای واحد آماده کسب‌وکار به پرسش‌های عملیاتی واقعی پاسخ می‌دهد، چهار وضعیت ریسک عملیاتی به‌صورت شفاف پرچم‌گذاری شده‌اند، بارگذاری SQLite به‌جای فرض‌شدن راستی‌آزمایی شده، و هر اجرا به‌طور کامل ثبت و بازتولیدپذیر است.

گام‌های پیشنهادی بعدی: اعتبارسنجی آستانه‌های موجودی مازاد و تأخیر در تکمیل سفارش در برابر اهداف واقعی سطح خدمت کسب‌وکار؛ افزودن نشانگر مستند مرجوعی/بازپرداخت به فایل فروش تا مرجوعی‌های واقعی دیگر از خطاهای ورود داده قابل‌تشخیص نباشند؛ و بازبینی ردیف‌های قرنطینه‌شده در data/processed/ با مسئول داده مرجع انبارها، چرا که یک شناسه انباری آویزان تکرارشونده معمولاً نشانه یک شکاف همگام‌سازی میان سامانه‌های منبع است، نه یک غلط تایپی یک‌باره.