ABC Food Market

From Spreadsheets to Real-Time Insights

Client Profile

1993

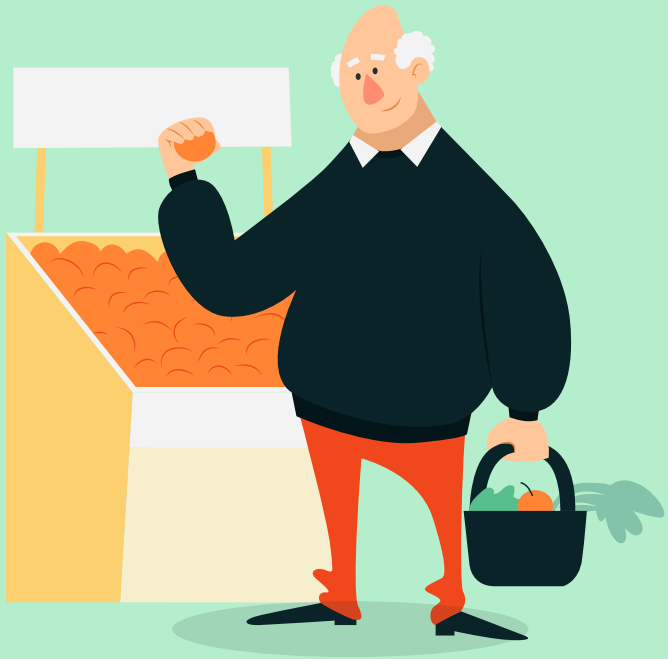
ABC Foodmart founded in Queens, NY. Started as a neighborhood grocery and has grown a loyal customer base over 30+ years.

2025 (Mid-Year)

Opened 3 new stores in Brooklyn, expanding from 2 to 5 locations across the city.

2025 (Now)

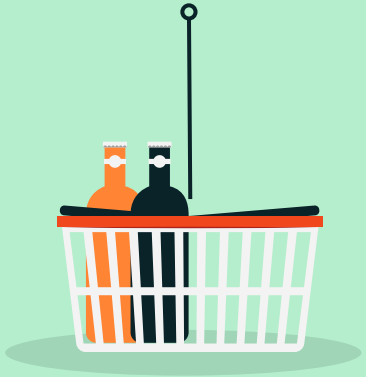
Hired our team to design a new data system and real-time dashboards.



Project Goals

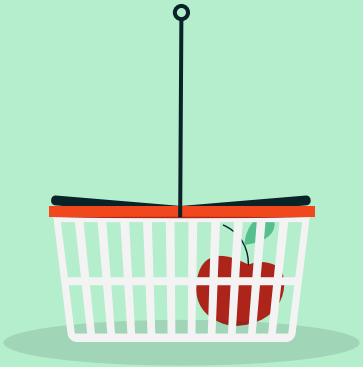
Objective

Build an integrated relational database, streamline ETL, deploy real-time dashboards



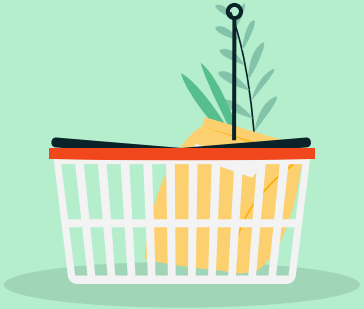
Scope

Centralize data across staffing, inventory, vendors, deliveries, sales, accounting



Approach

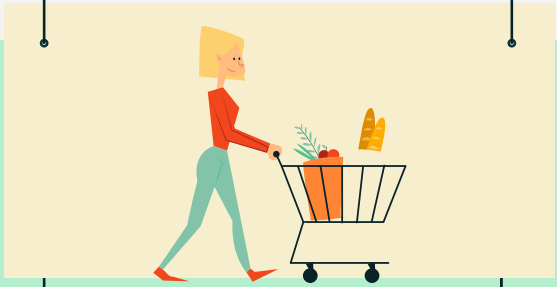
- Scalable data model (5 locations)
- Standardize & clean data
- Key metrics in 1–2 clicks



Original Data Overview

Operational Data

- Stores
- Employees
- Inventory Lots
- Purchase Orders



Transaction Data

- Transactions
- Sales Detailed Items
- Refunds

Other Linking Tables

Vendors, **Products**, **Payment Methods**, and **Shifts…**

connect operational and transactional data, enabling full traceability from supplier to customer.

Database Design

Master Data

Procure & Stock

Sell & Return

ETL Process

- **Extract**: Cleaned or synthetically generated data using Python Faker for reproducibility.
- **Transform**: Normalize data, fix column names, handle missing values.
- **Load**: Sequential loading into PostgreSQL via SQLAlchemy, maintaining referential integrity.

Loading Data Logic:

- Load master tables first (stores, products, vendors, etc.) to satisfy foreign keys.
- Load transactional tables in order such as: purchase orders → purchase products → inventory lots → transactions → sales details → refunds.
- Assign realistic dates, quantities, costs, and link sales to inventory lots for traceability.

Why: customer needs → insights we deliver

Managers need **fast, trusted answers** (sales, margins, inventory, refunds, labor)

Analysts need **repeatable** Python+SQL workflows (not ad-hoc spreadsheets)

Insights provided (examples):

- **Category/Store P&L** (revenue, COGS, GP, trend)
- **Low-stock & reorder** list; **near-expiration** FEFO monitor/markdowns
- **Refund rate & reasons** with drill-through to transactions
- **Revenue per labor hour** vs. store benchmark; **vendor fill/on-time**
- **Basket pairs** for cross-sell; operating expenses by store/type

One source of truth → **consistent KPIs** in Metabase for decision-making

How: tools, workflows, and operating model

Stack: **PostgreSQL** (tables + views/MVs), **Python** (SQLAlchemy + pandas), **Metabase**

Workflow: Python jobs run 10+ procedures, persist results as **views/MVs** → Metabase "Questions" power **auto-refreshing dashboards**

Examples: SQL window functions for **category P&L rank**; Python-parameterized **low-stock** + reorder quantities (lead time, sell-through); **refund hotspots** drill-downs

Performance & trust: indexed foreign keys; composite PK on inventory(store_id, product_sku); workload separation (OLTP writes vs. analytics reads); ingest **data-quality checks**; curated views for non-technical users

Access & security: roles for **analysts** (SQL + staging where appropriate) and **executives** (read-only dashboards); optional **row-level security**; PII minimized/aggregated in dashboards

Dashboard Showcase