

Retail Insights Framework

January 2024

Prepared by KC Rakam and Kundan Sarkar

Reviewed by Vish Ganapathy, Keerthi Chidambaram, C.J. Gunn, Baseer Siddiqui



Retail Insights Framework

Data insights framework to accelerate the development of consistent and interoperable retail applications

Why Retail Insights Framework?

The need for a retail insights framework is driven by the increasing complexity of retail businesses. As retailers expand their product offerings, open new stores, and enter new markets, they need to be able to manage and analyze large amounts of data in order to make informed business decisions. A framework to derive these insights can help retailers to consolidate and standardize their data, making it easier to access and analyze.

An insight framework can also help to improve the efficiency of retail operations. By providing a common understanding of the data, the framework can help to break down data silos and improve communication between different departments within a retail business. This can lead to improved decision-making, reduced costs, and increased profits.

Solution Approach

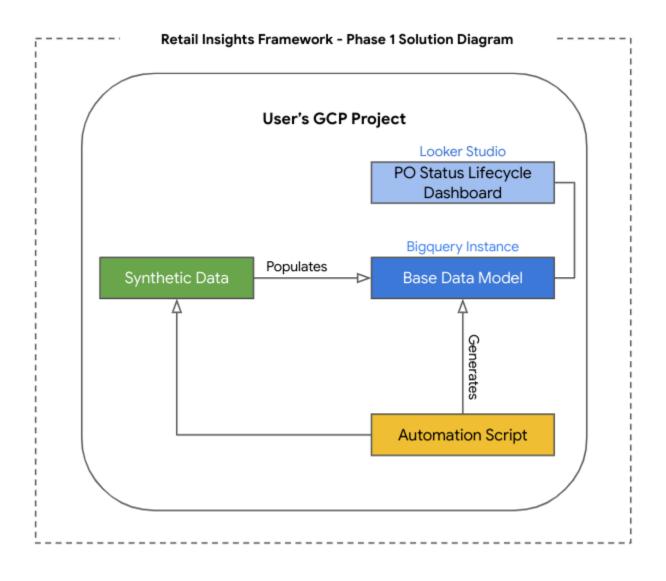
As our Retail and CPG customers are going through their digital supply chain transformation journeys, they are asking us how they can leverage Google's scale and AI/ML expertise to build an intelligent supply chain. Asks are primarily around how they can improve upon their visibility use cases and develop their predictive and prescriptive capabilities along their journey. Although we have a few data models already existing (e.g. Twin data model, Cortex data model, Data model for our productized solutions), there is a need for an insight framework that provides retail specific understandings and enables analytics over and above the company's master data and transaction data. To meet that requirement, the Retail/CPG team has defined a retail insights framework that combines master data dimensions with facts from transaction data and enhances it with aggregation views to be used to resolve analytic use cases and generate business insights.

The framework will initially focus on the PO visibility based on PO status updates. The goal is to position the framework as as an accelerator with the following components:

- Reference architecture and data model
- Pre-built automation scripts to create the data model and populate it with synthetic data. This script can be easily deployed by customers or partners in a Google Cloud project
- Looker Studio dashboards focusing on common use cases



Solution diagram





Retail Insight Framework (ERD View for Phase 1)

rcim.PO 🗊					rcim.ASN 🖸			
PONumber 🖉 🛛	VARCHAR NN	10			ASNNumber 🖉	VARCHAR		
РОТуре D	VARCHAR NN				ASNDate	DATE		
PODate 💭	TIMESTAMP				PONumber	VARCHAR		
EarliestShipDate D	DATE		rcim.POLine 🖸		ShipDate	DATE	rcim.ASNLine 🖸	
LatestShipDate D	DATE		PONumber 🖉 🛛	VARCHAR NN	 Carrierld	VARCHAR	ASNNumber 🖉	VARCHAR
SupplierId D	VARCHAR		POLine Ø	INTEGER	TransportMode D	VARCHAR NN	ASNLine 🖉	INTEGER
rcim.POStatusUpda	tes 🖸 💦 🖓		ShipToNode D	VARCHAR	TransportId	VARCHAR	Productid	VARCHAR
PONumber D	VARCHAR NN AR		ProductId	VARCHAR	TransportRouteld	VARCHAR	ContainerId	VARCHAR
POLine 🖸	INTEGER 2)		OrderQuantity	NUMERIC(8,2)	TransportStopId	VARCHAR	ContainerPosition	VARCHAR
Status D	VARCHAR NN 2)		OrderPrice	NUMERIC(8,2)	DestinationNode	VARCHAR	PONumber D	VARCHAR
EntryTime D	datetime 2)		CreateDate D	TIMESTAMP	BOLNumber D	VARCHAR	POLine	INTEGER
CreateDate D	TIMESTAMP 2)		UpdateDate D	TIMESTAMP	CertOrigin D	VARCHAR	OrderQty D	NUMERIC(8,2)
UpdateDate 🗖	TIMESTAMP (P		UpdateCount D	INTEGER	CustomsDecl D	VARCHAR	PromiseQty D	NUMERIC(8,2)
UpdateCount D	INTEGER 1P		(PONumber, POLine)		GrossAmount	NUMERIC(8,2)	ShippedQty D	NUMERIC(8,2)
(PONumber, POLine	e) ER				TaxAmount	NUMERIC(8,2)	GrossAmount	NUMERIC(8,2)
					DiscountAmount	NUMERIC(8,2)	TaxAmount	NUMERIC(8,2)
					NetAmount	NUMERIC(8,2)	DiscountAmount	NUMERIC(8,2)
					CreateDate D	TIMESTAMP	NetAmount	NUMERIC(8,2)
					UpdateDate D	TIMESTAMP	CreateDate D	TIMESTAMP
					UpdateCount D	INTEGER	UpdateDate 💭	TIMESTAMP
							UpdateCount D	INTEGER
							(PONumber, POLine))

For phase 1, this framework will support the PO Visibility dashboard. The dashboard will be based on views to be generated from the following reference and transaction tables:

Reference Tables

- RefNode
- RefTradePartner
- RefProduct

Transaction Tables

- PO
- POLine
- ASN
- ASNLine
- POStatusUpdates

Please refer to this github link for the complete ERD

Framework Overview

The automation script included with this framework will deploy the data model in a BigQuery warehouse in the customer's GCP instance. The script will also populate the tables of this data model with synthetic data. The Looker Studio dashboard will reference this data and provide the



lifecycle of purchase order status updates along with detailed metrics of POs, PO lines, ASNs and ASN lines.

This framework can be further extended by the customer to leverage BigQuery as its platform to ingest data from existing master data applications (e.g. Product Master, Customer Master, Supplier Master, Location Master, etc) and transactional data applications (e.g. ERP, WMS, TMS, etc). It will further ingest (if available as input) or aggregate (calculate based on historical data) some data elements in BigQuery views.

These tables and views will then be consumed by:

- The visualization engine to generate visibility use cases
- AutoML and Vertex AI platforms to build and train predictive models

From the data flow perspective, the fully developed enterprise solution will:

- Land the raw dataset in the staging tier of the framework
- Transform the data to the framework schema in the processing tier
- Generate views and dashboards for customers, products, nodes and inventory activities in the reporting tier

Framework design principles and data model

This data model has been designed with the following design principles in mind:

- 1. Focus on model structures that can be used for business Analytics.
- 2. Setup data in order to feed AI/ML and time series models for forecasting.
- 3. Allow multiple financial years' worth of data to coexist.
- 4. Allow for maintaining history of price changes and zone-based pricing.
- 5. Pll data is not necessary. However, if provided will be maintained separately.
- 6. Incorporate reference tables with metadata to support optimizations.
- 7. Create models for developing scenario modeling and discrete event simulations.
- 8. Take an 80/20 approach to only model key datasets rather than be overly prescriptive.
- 9. Make the models ISV agnostic so that retailers can develop their own vernacular.
- 10. Design in "many-to-many" flexibility for supply chain, sales & purchase orders.

Table Groups

All of the tables are grouped under the following categories. These groups are described in a subsequent section of this writing.

- 1. Reference Data
- 2. Master Data
- 3. Transaction Data Sales
- 4. Transaction Data Purchases



5. Analytical data - Activity Summary

Table Group - Reference Data

This table group maintains metadata that is used to describe store & DC layouts, replenishment schedules and associate tasks. These tables are designed to provide the necessary metadata required for any "digital twin" simulations such as scenario modeling or discrete event simulations.

rcim.RefTask D		rcim.NodeLay	yout 🗘	rcim.RefReplenishme	nt D
Taskid 🖉 🖸	VARCHAR NN	LayId 🖉 D	VARCHAR NN	FromNode 🖉 🖸	VARCHAR
EffectiveDate 🖉 D	TIMESTAMP NN	Zone 🖉	VARCHAR(3)	ToNode 🖉 🖸	VARCHAR
+13 hidden fields		Aisle 🖉	VARCHAR(2)	ProductId 🖸	VARCHAR
		Bay 🖉	VARCHAR(2)	EffectiveDate 🖉 D	TIMESTAMP NN
		Level 🖉	VARCHAR(1)	+11 hidden fields	
		Slot 🖉	VARCHAR(2)		
		+8 hidden field			
rcim.RefProcess D				rcim.RefSimula	te 💭

Table	Purpose
RefTask	Definitions of associate tasks.
RefProcess	Definition of processes.
RefResource	Definition of constraint resources.
RefReplenishment	Schedule of replenishments
NodeLayout	Definition of the physical store & DC layouts
RefSimulate	Definition of discrete event simulation models.

Table Group - Master Data



This group of tables contains the reference tables for master data such as product, customer, location/node, trading partners such as suppliers, carriers etc. They in turn are referenced in many of the transaction tables.

Par		VARCHAR NN VARCHAR NN BYTES	1	AssociateId P D	VARCHAR NN		
PIIT	oken 🖸						
		BYTES		EffectiveDate 🖉 🖸	TIMESTAMP NN		
+10	hidden fields			PIIToken 🗊	BYTES -		
				+5 hidden fields		rcim.RefNode D	
						Nodeld 🖉 🖸	VARCHAR NN
rcim.RefProduct 🖸						Layld D	VARCHAR
ProductId D	VARCHAR NN					+14 hidden fields	
EffectiveDate 🖉 💭	TIMESTAMP NN		rcim.PII 💭				
+63 hidden fields			PIIToken 🖉 💭	BYTES NN			
rcim.RefProductF	rice D		+40 hidden fields		rcim.RefCustomer	D	
ProductId 🖉 D	VARCHAR NN				Customerid 🖉 D	VARCHAR NN	1
NodeId 🖉 D	VARCHAR NN				CustomerType 🖉 🗄	VARCHAR(3) NN	l
EffectiveDate 🖉 🛛	TIMESTAMP NN					BYTE	S

Table	Purpose
RefTradingPartner	Definition of all trading partners denoted by type.
RefAssociate	Definition of all the frontline associates; Store, DC & MFG.
RefNode	Definition of all supply and demand nodes.
RefCustomer	Definition of all known customers.
RefProduct	Definition of all products.
RefProductPrice	Product pricing at the product/node level.
RefPII	Vault containing any PII data, identified by a PII token.

Table Group - Transactional Data - Sales

All B2C sales; in the store (i.e. POS) and digitally are defined here; irrespective of if it is a "known" customer or not.



rcim.SalesOrde	rD	rcim.SalesOrd	erLine D
Orderld 🖸	VARCHAR NN	 Orderld 🖸	VARCHAR NN
lodeld 🖸	VARCHAR	ProductId	VARCHAR
ustomerld D	VARCHAR	+12 hidden fiel	ds
AssociateId D	VARCHAR		

Table	Purpose		
SalesOrder	All customer sales across all channels.		
SalesOrderLine	Line items associated with each sales order.		

Table Group - Transactional data - Purchases

All purchase orders and their related ASNs are defined here.

٣	PURCHASES								
	rcim.PO 🗊					rcim.ASN D			
	PONumber 🖉 🖸	VARCHAR NN	- 			ASNNumber 🖉	VARCHAR		
	SupplierId 🖸	VARCHAR				PONumber	VARCHAR		
	ShipToNode D	VARCHAR				CarrierId	VARCHAR		
	+12 hidden fields			rcim.POLine 🖸		DestinationNode	VARCHAR	rcim.ASNLine D	
				PONumber 🖉 D	VARCHAR NN	+16 hidden fields		ASNNumber 🖉	VARCHAR
				POLine Ø	INTEGER			ASNLine 🖉	INTEGER
	rcim.POStatusUpdate	es 🖸		ShipToNode D	VARCHAR			ProductId	VARCHAR
	PONumber 🗊	VARCHAR NN		ProductId	VARCHAR			PONumber D	VARCHAR
	POLine D	INTEGER		(PONumber, POLine)				POLine	INTEGER
	(PONumber, POLine)			+5 hidden fields				(PONumber, POLin	ie)
	+5 hidden fields							+12 hidden fields	

Table	Purpose
PO	All purchase orders
POLine	All line items associated with a PO
POStatusUpdates	Records the full life cycle of PO & POLine statuses including dwell times.
ASN	All advance ship notices received from suppliers.



Table	Purpose
ASNLine	Line items associated with ASNLine.

Table Group - Analytical data - Activity Summary

These tables are designed to be derived tables which aggregate all the activities for a given entity; for e.g. Customer, Node/PO or Stock. These tables will be fine tuned to incorporate any aggregated metrics such as CLTV, AOV, Lead Times etc.

rcim.POActivity	/Summary 🖸	rcim.StockActivity	/Summary 🖸	rcim.CustomerActivityS	ummary D
FY 🖉 🖸	VARCHAR(4) NN	FY 🖉 🖸	VARCHAR(4) NN	FY 🖉	VARCHAR
FW 🖉 D	NUMERIC(2,0) NN	FW 🖉 🖵	NUMERIC(2,0) NN	CustomerId 🖉	VARCHAR
ShipToNode 🖉	VARCHAR	Nodeld 🖉	VARCHAR	+23 hidden fields	
POStatus 🖉 🖵	VARCHAR	ProductId \mathcal{O}	VARCHAR		
+5 hidden fields		StockStatus 🖉 D	VARCHAR NN		

Table	Purpose
POActivitySummary	Consolidated and summarized view into all inbound POS at a node level.
StockActivitySummary	A daily/weekly Clos-of-Business (COB) node/SKU level unit positions of inventory.
CustomerActivitySummary	Consolidated and summarized view into all customer sales orders.



Sample Dashboard



PO# 🛛 🔹	PO Upd Dt 😝 🔹		PO Upd St	PO Upd Line	ASN #	ASN PO Line	PO Line Order Qty	PO Order Amt	ASN Promi	ASN Shipp
po-94612172	Jan 17, 2024, 5:58:	alNVGuUcWaWohQlEnZ	Shipped	649317886	asn-GB60Q	5269231651065	89	\$34.8	94	90
po-94612172	Jan 17, 2024, 5:58:	VoSkwvSCBfNneJpszNcI	Shipped	649317886	asn-GB83S	5269231651065	39	\$34.8	94	90
po-94612172	Jan 17, 2024, 5:58:	VoSkwvSCBfNneJpszNcI	Shipped	649317886	asn-GB83S	8167090446713	39	\$34.8	51	46
po-94612172	Jan 17, 2024, 5:58:	tCIFYTjvegvvfoUPcJqG	Shipped	649317886	asn-GB86X	6493178862486	58	\$34.8	81	79
po-94612172	Jan 17, 2024, 5:58:	tCIFYTjvegvvfoUPcJqG	Shipped	649317886	asn-GB86X	5269231651065	58	\$34.8	94	90
po-94612172	Jan 17, 2024, 5:58:	tCIFYTjvegvvfoUPcJqG	Shipped	649317886	asn-GB86X	8167090446713	58	\$34.8	51	46
po-94612172	Jan 17, 2024, 5:58:	tCIFYTjvegvvfoUPcJqG	Shipped	649317886	asn-GB86X	4440235164300	58	\$34.8	76	71
po-94612172	Jan 17, 2024, 5:58:	tCIFYTjvegvvfoUPcJqG	Shipped	649317886	asn-GB86X	2558317312128	58	\$34.8	87	82
po-94612172	Jan 17, 2024, 5:58:	tCIFYTjvegvvfoUPcJqG	Shipped	649317886	asn-GB97M	1490532676025	58	\$34.8	42	38

- Filters : Dashboard information can be filtered with following variables
 - Period Financial Year, Weeks
 - ShippingTo ShipToNodes of POLines
 - POType This list can be dynamic based on customer specific POTypes
- Trend Lines
 - For each POStatus, there will be two trend lines, one for Order quantity (units) and the other for order amounts (\$\$\$)
- PO Information Table
 - Information on specific POs based on filter criteria
 - Can be nested information from PO, POLines, ASN, ASNLines and POStatuses

Please refer to this github link for detailed steps on how to deploy this framework.



Conclusion

As mentioned earlier, the current version of this framework is an accelerator with an automation script providing the deployable data model with associated synthetic data. It is also providing an out-of-the box Looker Studio dashboard focusing on a very specific visibility use case of the status update lifecycle of purchase orders. The base data model of this framework includes additional entities for customers, products, suppliers and tasks and the framework can be extended for use cases around customer experience, inventory visibility, supplier risk, associate productivity, etc in future releases. Google retail/CPG customers can deploy this framework standalone or with other GCP frameworks like Cortex and enable additional use cases with their complementary enterprise data.

Contact a Google Cloud sales rep to find out how you can get started today