BlueprintBI

SAMPLE DESIGN PACK

Retail Trading & Merchandising Performance

A Power BI trading report combining SQL Server store EPOS and ERP data with Shopify online orders in a Fabric Lakehouse.

Retail | Power BI semantic model (Fabric) | Direct Lake | 3 report pages | 27 measures | 7 tables

CONTENTS

1 Scope and inputs
2 Data architecture
3 Data model
4 DAX measures
5 KPIs and RAG rules
6 Report design
7 Assumptions and open questions

About this sample. Generated by the Blueprint BI pipeline on 2026-11-19 from the inputs in section 1, and not hand-edited. Figures shown in the KPI cards and report pages are illustrative sample data, not real results.

SECTION 1

Scope and inputs

Purpose. How are sales, margin and basket value trending against target and last year, by channel, store and product category? Which stores and categories need action?

Audience	Retail directors, merchandising and buying teams, regional and store managers
Default period	Current month to date, compared with the same period last year; refreshed daily by 06:00.
Industry	Retail
Report type	Retail & Merchandising
BI tool	Power BI
Currency	GBP (£)
Data sources	SQL Server, Shopify
Style and pages	Corporate Clean, 3 pages

Columns provided

9 columns supplied, and where each one lands in the model.

Source column	Model destination	Notes
`OrderID`	`FactSales[OrderID]`	Degenerate key. Shopify IDs are prefixed 'SH-' so they cannot collide with EPOS transaction numbers.
`OrderDate`	`FactSales[DateKey], DimDate[Date]`	Converted to a yyyymmdd integer key. Times are truncated to the UK local date.
`StoreCode`	`FactSales[StoreKey], DimStore[StoreCode]`	Looked up to a surrogate key. Shopify orders map to the virtual store code 'WEB01'.
`SKU`	`FactSales[ProductKey], DimProduct[SKU]`	Looked up to a surrogate key. Shopify variant SKUs are matched to ERP SKUs.
`Quantity`	`FactSales[Quantity]`	Returns are kept as negative quantities on the original order.
`UnitPrice`	`FactSales[UnitPrice], FactSales[GrossAmount]`	Selling price excluding VAT. GrossAmount = Quantity × UnitPrice.
`Discount`	`FactSales[DiscountAmount]`	Assumed to be a line-level £ value excluding VAT. A percentage is converted to £ in Silver.
`CustomerID`	`FactSales[CustomerKey], DimCustomer[CustomerID]`	Anonymous EPOS sales map to the 'Unknown' member, CustomerKey -1.
`Channel`	`FactSales[ChannelKey], DimChannel[ChannelName]`	Standardised to Store, Online and Click & Collect.

KPIs provided

KPI	Definition	Target	Green	Amber	Red
Sales	Net sales excluding VAT after discounts and returns, against the £5M monthly target.	£5M per month	≥ £5,000,000	≥ £4,500,000	< £4,500,000
Gross Margin %	Net sales minus standard cost, divided by net sales.	40%	≥ 40.0%	≥ 37.0%	< 37.0%
Average Basket Value	Net sales divided by distinct baskets with positive net sales.	£42	≥ £42.00	≥ £39.00	< £39.00

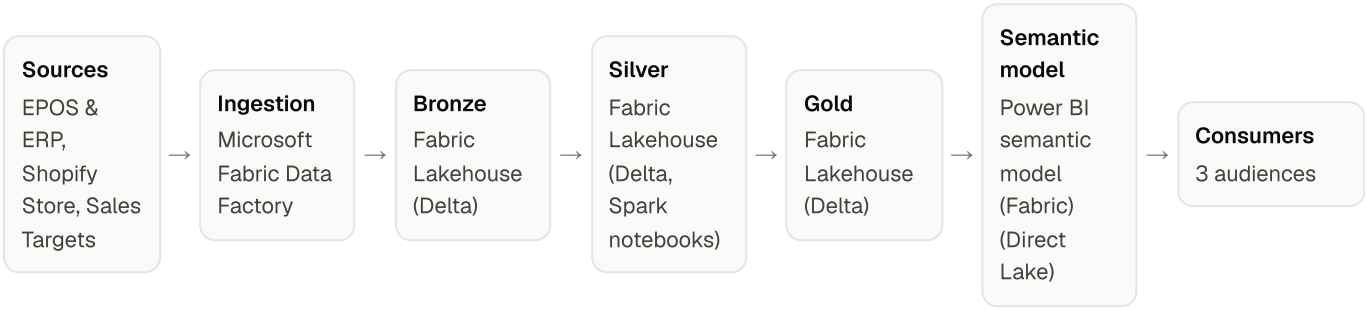
Business rules

Rule	Definition	Calculated in
Net Sales	NetSales = Quantity × UnitPrice − DiscountAmount, excluding VAT, with returns netted on the original order line.	Gold layer, every load
Cost of Sales	CostAmount = Quantity × the ERP standard unit cost effective on the order date.	Gold layer, every load
Order (basket)	A distinct OrderID with positive net sales. Fully returned orders do not count as baskets.	Semantic model measure [Orders]
Like-for-like store	A store trading for the full current period and the same period last year, excluding refits closed for 7 or more days.	Gold layer, DimStore[IsLikeForLike], refreshed monthly

SECTION 2

Data architecture

Direct Lake over the Gold Delta tables, with a dedicated _Measures table.



Data sources

Source	System	Refresh	Notes
EPOS & ERP	SQL Server	Daily 02:00	Store transaction lines, product master, standard costs and store hierarchy.
Shopify Store	Shopify Admin API	Daily 02:30	Online and click-and-collect orders, line items, discounts and customers.
Sales Targets	SQL Server (Finance planning schema)	Daily 02:00	Monthly sales targets by store, signed off by Finance.

Ingestion and layers

Stage	Technology	What happens
Ingestion	Microsoft Fabric Data Factory	Incremental load using a modified-date watermark, with a 7-day lookback for returns. Copy activities land SQL Server and Shopify data in Bronze; notebooks transform it to Silver and Gold.
Bronze	Fabric Lakehouse (Delta)	Raw, append-only copies of the source tables and Shopify JSON.
Silver	Fabric Lakehouse (Delta, Spark notebooks)	Cleaned, deduplicated and conformed orders, SKUs, stores and channels.
Gold	Fabric Lakehouse (Delta)	Star schema facts and dimensions with surrogate keys and derived amounts.

Semantic model and operations

Tool and storage mode	Power BI semantic model (Fabric), Direct Lake
Refresh and alerts	Ready by 06:00 daily; failures alert the BI support Teams channel and the on-call data engineer.
History kept	3 complete years plus the current year, covering YoY comparisons for the default period.
Environments	Dev, Test and Prod Fabric workspaces
Deployment	Fabric deployment pipelines with Test sign-off by Merchandising
Source control	Semantic model and report stored as PBIP in Azure DevOps Git; notebooks under Fabric Git integration

- **Storage mode caveat:** Queries fall back to DirectQuery if RLS is defined on a SQL endpoint, so RLS is defined in the semantic model only.
- **Storage mode caveat:** Calculated columns are not supported, so derived columns are built in Gold.
- **Storage mode caveat:** Exceeding capacity guardrails on rows per table forces fallback; size the capacity at F64 or higher.

Consumers

- Retail directors - Power BI App
- Merchandising team - Power BI App and Excel Analyse
- Store and regional managers - Power BI mobile

Governance

- Row-level security on DimStore[AreaManagerUPN] limits managers to their stores.
- Data quality checks run in Silver: unmatched SKUs, unmatched stores and negative prices are logged and alerted.
- A daily reconciliation compares Gold net sales with EPOS Z-reads and Shopify payouts, with a tolerance of 0.5%.
- Sensitivity label 'Confidential' applies; customer IDs are pseudonymised.
- Refresh monitoring uses the Fabric Monitoring hub with failure alerts.

SECTION 3

Data model

Star schema (galaxy with target fact): 2 fact tables and 5 dimensions. Measures are stored on a dedicated _Measures table.

FactSales

fact

Grain: One row per order line (OrderID and SKU) · Source: SQL Server EPOS and Shopify

Column	Type	Key and role	Description
OrderID	NVARCHAR(40)	Degenerate ID, hidden	Order or transaction number.
DateKey	INT	FK, → DimDate[DateKey], hidden	Order date key.
StoreKey	INT	FK, → DimStore[StoreKey], hidden	Store surrogate key.
ProductKey	INT	FK, → DimProduct[ProductKey], hidden	Product surrogate key.
CustomerKey	INT	FK, → DimCustomer[CustomerKey], hidden	Customer surrogate key.
ChannelKey	INT	FK, → DimChannel[ChannelKey], hidden	Channel surrogate key.
Quantity	INT	Additive, hidden	Units sold, negative for returns.
UnitPrice	DECIMAL(18,2)	Non-additive, hidden	Unit selling price excluding VAT.
GrossAmount	DECIMAL(18,2)	Additive, hidden	Quantity × UnitPrice.
DiscountAmount	DECIMAL(18,2)	Additive, hidden	Line discount in £, excluding VAT.
NetSales	DECIMAL(18,2)	Additive, hidden	Gross amount less discount.
CostAmount	DECIMAL(18,2)	Additive, hidden	Standard cost of units sold.

FactSalesTarget

fact

Grain: One row per store per calendar month · Source: SQL Server Finance planning

Column	Type	Key and role	Description
DateKey	INT	FK, → DimDate[DateKey], hidden	First day of the target month.
StoreKey	INT	FK, → DimStore[StoreKey], hidden	Store surrogate key.
TargetSales	DECIMAL(18,2)	Additive, hidden	Monthly net sales target in £.

DimDate

dimension · marked as date table

Grain: One row per calendar day · Source: Generated in Gold

Column	Type	Key and role	Description
DateKey	INT	PK, hidden	Date key in yyyymmdd format.
Date	DATE	Date	Calendar date.
Year	INT	Attribute	Calendar year.
MonthNumber	INT	Attribute, hidden	Month number from 1 to 12.
MonthName	NVARCHAR(10)	Attribute, sorted by MonthNumber	Short month name.
YearMonthNumber	INT	Attribute, hidden	Year-month in yyyymm format, used for sorting.
YearMonth	NVARCHAR(10)	Attribute, sorted by YearMonthNumber	Year-month label, e.g. Mar 2025.
DayOfWeekNumber	INT	Attribute, hidden	Day of week, Monday = 1.
DayOfWeek	NVARCHAR(10)	Attribute, sorted by DayOfWeekNumber	Day name.
IsComplete	BIT	Attribute	True for dates before today.

DimStore

dimension · SCD type 2

Grain: One row per store version · Source: SQL Server ERP

Column	Type	Key and role	Description
StoreKey	INT	PK (surrogate), hidden	Surrogate key.
StoreCode	NVARCHAR(10)	Business key	ERP store code; WEB01 is the online store.
StoreName	NVARCHAR(100)	Attribute	Store name.
Region	NVARCHAR(50)	Attribute	Trading region.
StoreFormat	NVARCHAR(30)	Attribute	High Street, Retail Park, Outlet or Online.
IsLikeForLike	BIT	Attribute	Like-for-like flag.
AreaManagerUPN	NVARCHAR(100)	Attribute, hidden	Area manager login, used for RLS.
ValidFrom	DATE	Date, hidden	SCD2 start date.
ValidTo	DATE	Date, hidden	SCD2 end date.

DimProduct

dimension · SCD type 1

Grain: One row per SKU · Source: SQL Server ERP

Column	Type	Key and role	Description
ProductKey	INT	PK (surrogate), hidden	Surrogate key.
SKU	NVARCHAR(30)	Business key	Stock keeping unit.

Column	Type	Key and role	Description
ProductName	NVARCHAR(150)	Attribute	Product description.
Department	NVARCHAR(50)	Attribute	Merchandise department.
Category	NVARCHAR(50)	Attribute	Merchandise category.
Brand	NVARCHAR(50)	Attribute	Brand.
PriceBandSort	INT	Attribute, hidden	Sort order for the price band.
PriceBand	NVARCHAR(20)	Attribute, sorted by PriceBandSort	Entry, Core or Premium.

DimCustomer

dimension · SCD type 1

Grain: One row per customer · Source: Shopify and SQL Server loyalty

Column	Type	Key and role	Description
CustomerKey	INT	PK (surrogate), hidden	Surrogate key; -1 is Unknown.
CustomerID	NVARCHAR(40)	Business key, hidden	Pseudonymised customer ID.
CustomerType	NVARCHAR(20)	Attribute	Loyalty, Registered or Anonymous.
HomeRegion	NVARCHAR(50)	Attribute	Region from the postcode district.

DimChannel

dimension · SCD type 1

Grain: One row per sales channel · Source: Gold reference data

Column	Type	Key and role	Description
ChannelKey	INT	PK (surrogate), hidden	Surrogate key.
ChannelCode	NVARCHAR(10)	Business key, hidden	Channel code.
ChannelName	NVARCHAR(30)	Attribute	Store, Online or Click & Collect.

Relationships

From (many)	To (one)	Cardinality	Filter	Active	Reason
FactSales.DateKey	DimDate.DateKey	many-to-one	single	Yes	
FactSales.StoreKey	DimStore.StoreKey	many-to-one	single	Yes	
FactSales.ProductKey	DimProduct.ProductKey	many-to-one	single	Yes	
FactSales.CustomerKey	DimCustomer.CustomerKey	many-to-one	single	Yes	
FactSales.ChannelKey	DimChannel.ChannelKey	many-to-one	single	Yes	
FactSalesTarget.DateKey	DimDate.DateKey	many-to-one	single	Yes	
FactSalesTarget.StoreKey	DimStore.StoreKey	many-to-one	single	Yes	

SECTION 4

DAX measures

27 measures in 6 display folders, stored on a dedicated `_Measures` table. Each block shows the measure name, format string, DAX and description.

Automated checks: 0 errors and 4 warnings found (see the Measures tab in the Studio for details).

Sales

Sales

`£#,0`

```
Sales =
SUM ( FactSales[NetSales] )
```

Net sales excluding VAT, after discounts and returns.

Gross Sales

`£#,0`

```
Gross Sales =
SUM ( FactSales[GrossAmount] )
```

Sales before discounts.

Discount Amount

`£#,0`

```
Discount Amount =
SUM ( FactSales[DiscountAmount] )
```

Total discounts given.

Discount %

`0.0%`

```
Discount % =
VAR g = [Gross Sales] RETURN IF ( g <= 0, BLANK (), DIVIDE ( [Discount Amount], g ) )
```

Discount as a share of gross sales.

Units Sold

`#,0`

```
Units Sold =
SUM ( FactSales[Quantity] )
```

Net units sold.

Orders

`#,0`

```
Orders =
CALCULATE ( DISTINCTCOUNT ( FactSales[OrderID] ), FILTER ( VALUES ( FactSales[OrderID] ), CALCULATE ( SUM ( FactSales[NetSales] ) ) > 0 ) )
```

Distinct baskets with positive net sales.

Average Basket Value

`£#,0.00`

```
Average Basket Value =
VAR o = [Orders] RETURN IF ( o <= 0, BLANK (), DIVIDE ( [Sales], o ) )
```

Net sales per basket.

Margin

Cost of Sales

`£#,0`

```
Cost of Sales =
SUM ( FactSales[CostAmount] )
```

Standard cost of units sold.

Gross Margin

`£#,0`

```
Gross Margin =
[Sales] - [Cost of Sales]
```

Sales less cost of sales.

Gross Margin %

`0.0%`

```
Gross Margin % =
VAR s = [Sales] RETURN IF ( s <= 0, BLANK (), DIVIDE ( [Gross Margin], s ) )
```

Gross margin as a share of sales.

Time Intelligence

Sales PY

`£#,0`

```
Sales PY =
CALCULATE ( [Sales], SAMEPERIODLASTYEAR ( DimDate[Date] ) )
```

Sales for the same calendar period last year.

Sales YoY %

+0.0%;-0.0%;0.0%

```
Sales YoY % =
VAR py = [Sales PY] RETURN IF ( py <= 0, BLANK (), DIVIDE ( [Sales] - py, py ) )
```

Sales growth versus last year.

Gross Margin % PY

0.0%

```
Gross Margin % PY =
CALCULATE ( [Gross Margin %], SAMEPERIODLASTYEAR ( DimDate[Date] ) )
```

Gross margin % for the same period last year.

Gross Margin % YoY pts

+0.0%;-0.0%;0.0%

```
Gross Margin % YoY pts =
VAR cur = [Gross Margin %] VAR py = [Gross Margin % PY] RETURN IF ( ISBLANK ( cur ) || ISBLANK ( py ), BLANK (), cur - py )
```

Change in margin percentage points versus last year.

Average Basket Value PY

£#,0.00

```
Average Basket Value PY =
CALCULATE ( [Average Basket Value], SAMEPERIODLASTYEAR ( DimDate[Date] ) )
```

Average basket value for the same period last year.

Average Basket Value YoY %

+0.0%;-0.0%;0.0%

```
Average Basket Value YoY % =
VAR py = [Average Basket Value PY] RETURN IF ( py <= 0, BLANK (), DIVIDE ( [Average Basket Value] - py, py ) )
```

Basket value growth versus last year.

Targets

Sales Target

£#,0

```
Sales Target =
SUM ( FactSalesTarget[TargetSales] )
```

Monthly store sales target from Finance.

Sales vs Target %

0.0%

```
Sales vs Target % =
VAR t = [Sales Target] RETURN IF ( ISBLANK ( t ) || t <= 0, BLANK (), DIVIDE ( [Sales], t ) )
```

Sales as a share of target.

Sales Variance

£#,0;-£#,0

```
Sales Variance =
VAR t = [Sales Target] RETURN IF ( ISBLANK ( t ), BLANK (), [Sales] - t )
```

Sales minus target.

Gross Margin % Target

0.0%

```
Gross Margin % Target =
0.40
```

Company gross margin target of 40%.

Gross Margin % Variance pts

+0.0%;-0.0%;0.0%

```
Gross Margin % Variance pts =
VAR cur = [Gross Margin %] RETURN IF ( ISBLANK ( cur ), BLANK (), cur - [Gross Margin % Target] )
```

Margin points above or below target.

Average Basket Value Target

£#,0.00

```
Average Basket Value Target =
42
```

Company basket value target of £42.

Average Basket Value Variance

£#,0.00;-£#,0.00

```
Average Basket Value Variance =
VAR cur = [Average Basket Value] RETURN IF ( ISBLANK ( cur ), BLANK (), cur - [Average Basket Value Target] )
```

Basket value minus target.

Labels

Selected Period Label

Text

```
Selected Period Label =
VAR d1 = MIN ( DimDate[Date] ) VAR d2 = MAX ( DimDate[Date] ) RETURN "Period: " & FORMAT ( d1, "d mmm yyyy" ) & " to " & FORMAT ( d2, "d mmm yyyy" )
```

Selected date range for page subtitles.

RAG

Sales RAG

0

```
Sales RAG =
VAR _Value = [Sales]
RETURN
    SWITCH (
        TRUE (),
        ISBLANK ( _Value ), BLANK (),
        _Value >= 5000000, 1,    -- Green
        _Value >= 4500000, 2,    -- Amber
        3                   -- Red
    )
```

1 Green (≥ £5,000,000), 2 Amber (≥ £4,500,000), 3 Red (< £4,500,000). Drives conditional formatting for Sales.

Gross Margin RAG

0

```
Gross Margin RAG =
VAR _Value = [Gross Margin %]
RETURN
    SWITCH (
        TRUE (),
        ISBLANK ( _Value ), BLANK (),
        _Value >= 0.4, 1,    -- Green
        _Value >= 0.37, 2,    -- Amber
        3                   -- Red
    )
```

1 Green (≥ 40.0%), 2 Amber (≥ 37.0%), 3 Red (< 37.0%). Drives conditional formatting for Gross Margin %.

Average Basket Value RAG

```
Average Basket Value RAG =
VAR _Value = [Average Basket Value]
RETURN
    SWITCH (
        TRUE (),
        ISBLANK ( _Value ), BLANK (),
        _Value >= 42, 1,    -- Green
        _Value >= 39, 2,    -- Amber
        3                    -- Red
    )
```

1 Green (≥ £42.00), 2 Amber (≥ £39.00), 3 Red (< £39.00). Drives conditional formatting for Average Basket Value.

SECTION 5

KPIs and RAG rules

Each KPI target becomes a RAG measure (section 4) that returns 1 for Green, 2 for Amber and 3 for Red. Visuals use these codes for colour, and every status is shown as colour, icon and word together, so it never relies on colour alone.

KPI	Measure	Target	Green	Amber	Red	RAG measure
Sales	`Sales`	£5M per month	≥ £5,000,000	≥ £4,500,000	< £4,500,000	`Sales RAG`
Gross Margin %	`Gross Margin %`	40%	≥ 40.0%	≥ 37.0%	< 37.0%	`Gross Margin RAG`
Average Basket Value	`Average Basket Value`	£42	≥ £42.00	≥ £39.00	< £39.00	`Average Basket Value RAG`

Sample values

Illustrative values used in the report pages, with the status each one gets from the thresholds above.

KPI	Sample value	Change	Status	Direction
Sales	£4.7M	+3.8%	⚠ Amber	Higher is better
Gross Margin %	41.2%	+0.6 pts	✓ Green	Higher is better
Average Basket Value	£37.80	-2.1%	✗ Red	Higher is better

Status colours

✓ Green	⚠ Amber	✗ Red
#15803D	#B45309	#B91C1C

SECTION 6

Report design

3 report pages. Layouts below use the theme and the illustrative KPI values from section 5; charts show placeholder shapes.

Theme: Corporate Clean Indigo

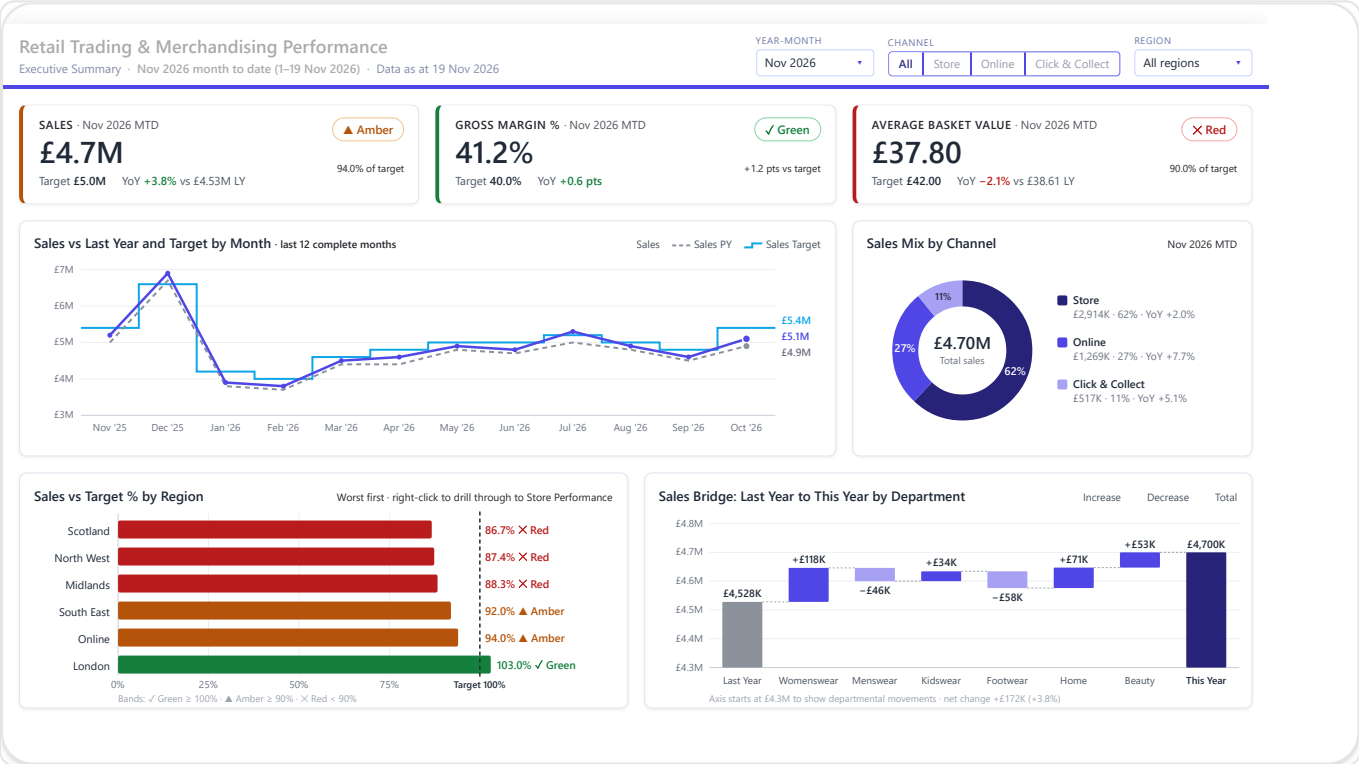
Primary #4F46E5 | Secondary #312E81 | Accent #0EA5E9 | Background #FFFFFF | Text #1F2937

Data colours, in order: Font: Segoe UI

Page 1: Executive Summary

Company trading position against target and last year. Default filter: Current month to date.

Audience: Retail directors and the executive team · Slicers: DimDate[YearMonth] (dropdown), DimChannel[ChannelName] (buttons), DimStore[Region] (dropdown)



Page 1 as drawn by the AI Design Studio, unedited. Illustrative sample data.

Page 1 visuals

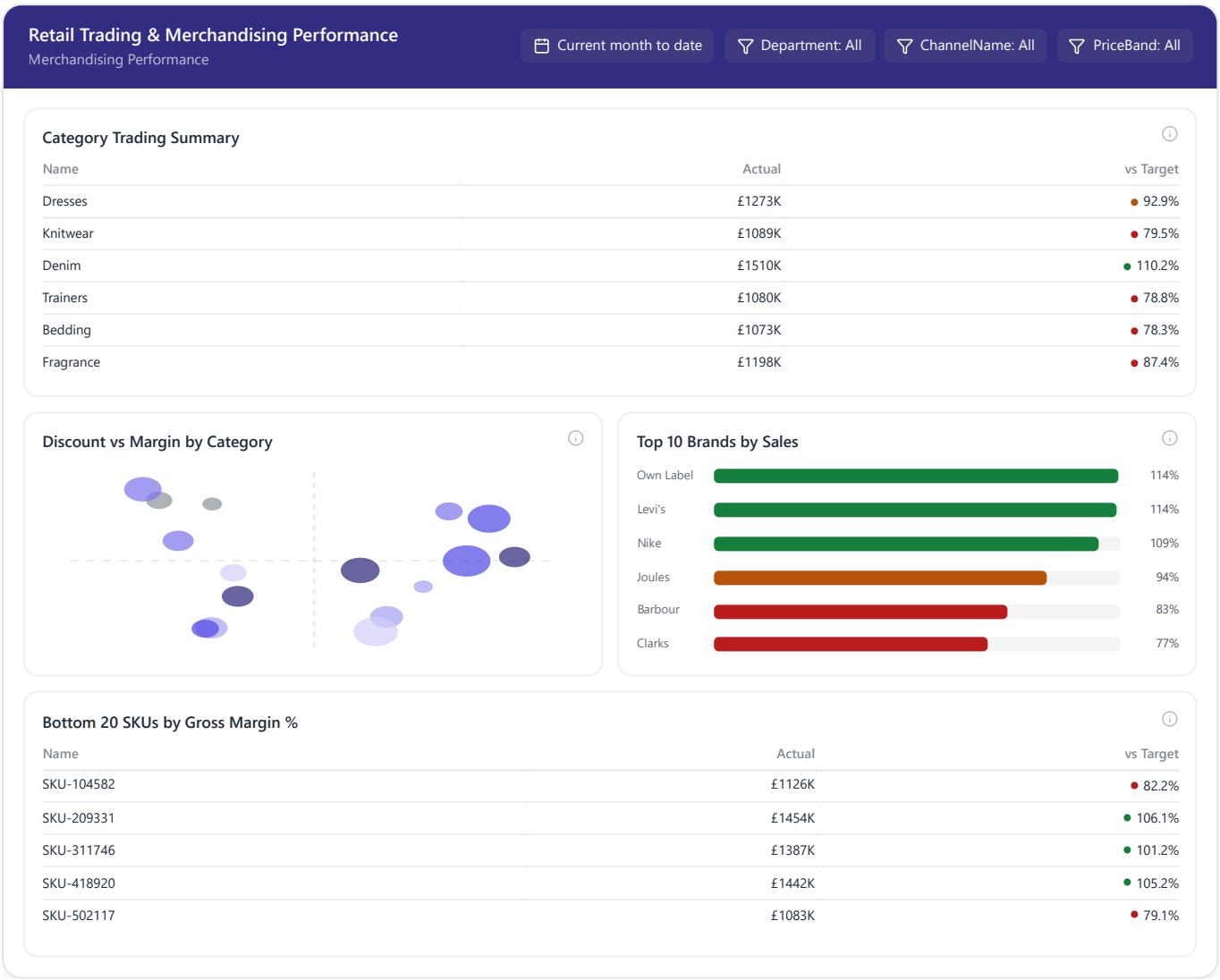
Visual	Type	Fields	Formatting and interaction
Trading KPIs	kpi strip	**Values:** `[Sales]`, `[Gross Margin %]`, `[Average Basket Value]` **Tooltip:** `[Sales YoY %]`, `[Sales vs Target %]`	Card status from [Sales RAG], [Gross Margin RAG] and [Average Basket Value RAG], shown as an icon plus the status word. Each card shows its YoY change underneath.

Visual	Type	Fields	Formatting and interaction
Sales vs Last Year and Target by Month	line	**Axis:** `DimDate[YearMonth]` **Values:** `[Sales]`, `[Sales PY]`, `[Sales Target]`	Sales PY shown as a dashed grey line; target as a stepped line. Data labels on the last point only, in £M. Filter: Last 12 complete months
Sales Mix by Channel	donut	**Legend:** `DimChannel[ChannelName]` **Values:** `[Sales]` **Tooltip:** `[Sales YoY %]`	Data labels show the percentage of total.
Sales vs Target % by Region	bar	**Axis:** `DimStore[Region]` **Values:** `[Sales vs Target %]` **Tooltip:** `[Sales Variance]`	Constant line at 100%. Sorted ascending so the worst regions appear first. Drill-through to Store Performance
Sales Bridge: Last Year to This Year by Department	waterfall	**Axis:** `DimProduct[Department]` **Values:** `[Sales]` **Tooltip:** `[Sales PY]`	Breakdown versus Sales PY, with the starting bar labelled 'Last Year'. Increases and decreases are labelled with signed £K values.

Page 2: Merchandising Performance

Category, brand and SKU performance on sales, margin and discount. Default filter: Current month to date.

Audience: Merchandisers and buyers · Slicers: DimProduct[Department] (dropdown), DimChannel[ChannelName] (buttons), DimProduct[PriceBand] (buttons)



Page 2 visuals

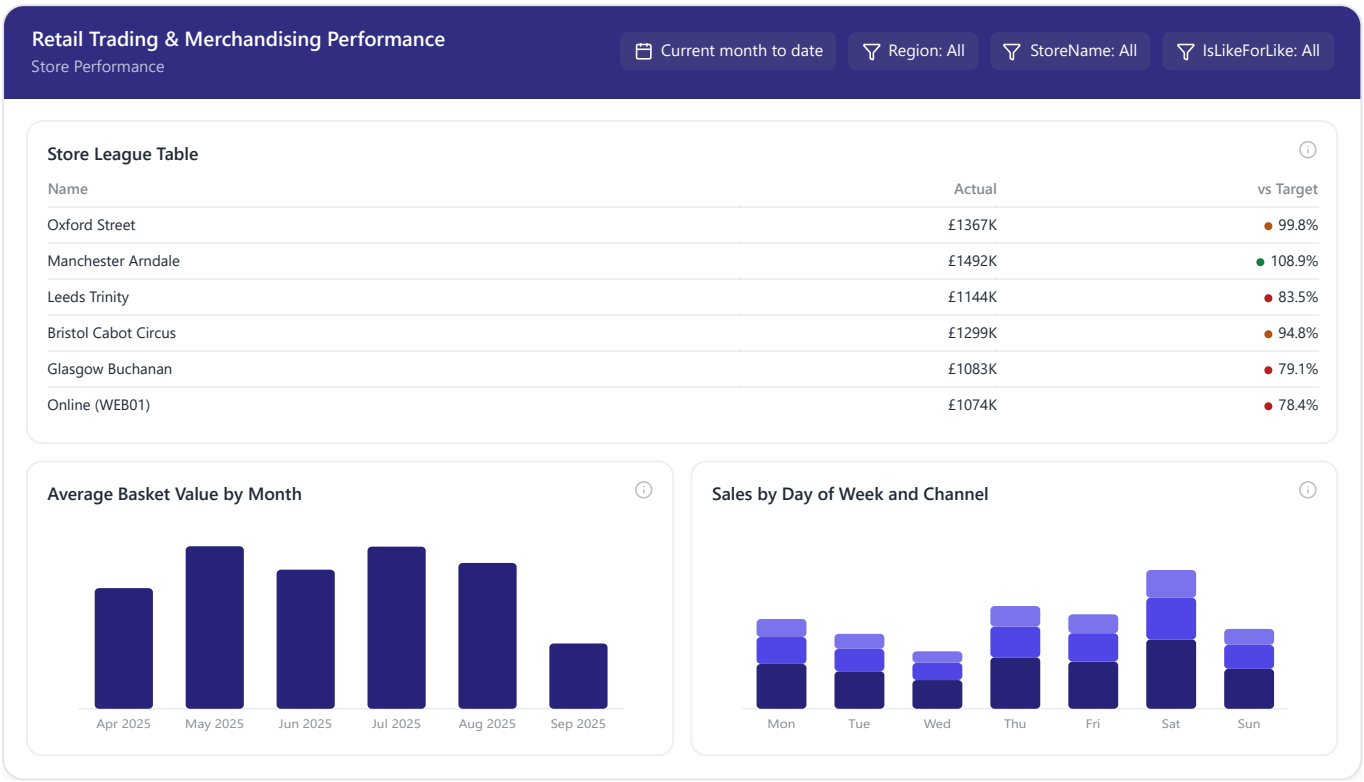
Visual	Type	Fields	Formatting and interaction
Category Trading Summary	matrix	**Rows:** `DimProduct[Department]`, `DimProduct[Category]` **Values:** `[Sales]`, `[Sales YoY %]`, `[Gross Margin %]`, `[Gross Margin % YoY pts]`, `[Discount %]`, `[Units Sold]`	Gross Margin % shows an icon and text from [Gross Margin RAG]. Sales YoY % uses a diverging font colour with +/- signs.
Discount vs Margin by Category	scatter	**Details:** `DimProduct[Category]` **X:** `[Discount %]` **Y:** `[Gross Margin %]` **Size:** `[Sales]`	Constant Y line at [Gross Margin % Target]. Category data labels shown.
Top 10 Brands by Sales	bar	**Axis:** `DimProduct[Brand]` **Values:** `[Sales]` **Tooltip:** `[Gross Margin %]`, `[Sales YoY %]`	Sorted descending; data labels in £K. Filter: Top 10 by [Sales]

Visual	Type	Fields	Formatting and interaction
Bottom 20 SKUs by Gross Margin %	table	**Values:** `DimProduct[SKU]`, `DimProduct[ProductName]`, `DimProduct[Category]`, `[Units Sold]`, `[Sales]`, `[Discount %]`, `[Gross Margin %]`	Gross Margin % shows an icon and text from [Gross Margin RAG]. Sorted ascending by Gross Margin %. Filter: Bottom 20 by [Gross Margin %], where [Sales] > £1,000

Page 3: Store Performance

Store-level target delivery, basket value and channel trends. Default filter: Current month to date.

Audience: Regional and store managers · Slicers: DimStore[Region] (dropdown), DimStore[StoreName] (dropdown), DimStore[IsLikeForLike] (buttons) · Drill-through from Executive Summary



Page 3 visuals

Visual	Type	Fields	Formatting and interaction
Store League Table	table	**Values:** `DimStore[StoreName]`, `DimStore[Region]`, `[Sales]`, `[Sales Target]`, `[Sales vs Target %]`, `[Sales YoY %]`, `[Average Basket Value]`, `[Gross Margin %]`	Average Basket Value shows an icon and text from [Average Basket Value RAG]; Gross Margin % from [Gross Margin RAG]. Data bars on Sales vs Target %; sorted ascending.
Average Basket Value by Month	column	**Axis:** `DimDate[YearMonth]` **Values:** `[Average Basket Value]` **Tooltip:** `[Average Basket Value PY]`, `[Orders]`	Constant line at [Average Basket Value Target] (£42). Column colour from [Average Basket Value RAG], with data labels in £0.00. Filter: Last 12 complete months
Sales by Day of Week and Channel	stacked column	**Axis:** `DimDate[DayOfWeek]` **Legend:** `DimChannel[ChannelName]` **Values:** `[Sales]`	Sorted by DayOfWeekNumber.

SECTION 7

Assumptions and open questions

Confirm these with the business before build. Each one changes the model or measures if the answer differs.

#	Assumption or question	Impact if different
1	Prior-year comparisons use the calendar year; the business may report on a 4-4-5 or 52/53-week retail calendar.	Add fiscal week, period and year columns to DimDate and rewrite the PY measures to shift by 364 days or by fiscal period.
2	Sales and UnitPrice exclude VAT, and Discount is a line-level £ amount.	Silver must strip VAT or convert a percentage discount to £, otherwise Sales and margin are overstated.
3	Sales targets are set by store and calendar month only, so target measures are not sliced by product, channel or day.	A product or channel target grain needs extra keys on FactSalesTarget; weekly targets need a DateKey per week.
4	The £5M Sales target and RAG bands apply to the company total per month; store status uses Sales vs Target %.	If the bands should apply per store, the Sales KPI must move to [Sales vs Target %] with bands of 1.00 and 0.95.
5	Click & Collect orders are attributed to the online store WEB01, not to the collecting store.	Store attribution would need a collection StoreCode from Shopify, which would shift sales between stores and Online.
6	Cost of sales uses the ERP standard cost effective on the order date, not weighted-average or landed cost.	Gold must join a costing history table, and Gross Margin % values will change.
7	The RLS mapping of area managers to stores is maintained in the ERP store master.	If HR holds the mapping, add a security bridge table fed from HR, which also allows multi-store access.

Recommended next steps

1. Partition FactSales by month in Gold and use V-Order so Direct Lake stays within guardrails.
2. Pre-aggregate Shopify payloads in Silver; do not land line-item JSON in Gold.
3. Document every measure's description in the PBIP model and publish a data dictionary to the Power BI App.
4. Use consistent naming: Fact/Dim prefixes, PascalCase columns and measure folders matching KPI names.
5. Run Best Practice Analyzer and DAX query tests in the deployment pipeline before promoting to Prod.
6. Drive adoption with a weekly trading subscription email to store managers, sent at 07:00 on Mondays.