

Menswear Insights Hub

* Tech Stack	AppSource custom visuals	Azure SQL	DAX	Power BI Desktop & Service
= Brief Summary	Built a two-page Power BI report on top of Azure SQL Database using a men's clothing dataset. Work covered cloud setup, secure connectivity, data preparation, report design (custom page size, branding), DAX enhancements, and governed sharing through Power BI Apps .			
🔗 Link	https://github.com/khanhmdinh/khanhmdinh.github.io/tree/main/06_Menswear%20Insights%20Hub			



Navigation bar

Table of Contents

[Table of Contents](#)

[Summary](#)

[Data Assessment & Cleaning Tools](#)

[Dataset Information](#)

[Data Cleaning \(Azure SQL Database\)](#)

[Data Cleaning: Price Standardization \(Power Query\)](#)

[DAX Functions](#)

[Project Showcase](#)

Summary

Scope of Work

- **Environment & Access**
 - Provisioned Azure SQL DB; configured **firewall/IP allowlist**.
 - Loaded source data and validated schema; documented connection details.
- **Connectivity to Power BI:** Established two connection patterns to Azure SQL: **Database connector** & Azure AD.
- **Data Preparation**
 - Performed **data cleaning** and profiling in Azure SQL.
 - Created **calculated columns** with **DAX** (business formulas applied to selected fields).
- **Report Design**
 - Custom **page settings & size** for a wide, presentation-style layout.
 - Branding (brands list on landing page); added **AppSource custom visual (ticker/scroller)** for marquee text.
 - Built a second page with prominent company banner and focused visuals.
- **Publishing & Sharing:** Published to **Power BI Service**



Deliverables

- **PBIX report** (desktop file) with custom page size, visuals, navigation.
- **Azure SQL scripts:** data load & cleanup (DDL/DML).
- **DAX assets:** list of calculated columns/measures with definitions.
- **Connection runbook:** steps for Database vs. Microsoft Account (AAD) connections.
- **Power BI App in Service:** app setup, audiences/permissions, and share settings.
- **Light data dictionary:** tables, key columns, business definitions.

Success Criteria

- **Secure connectivity:** Azure SQL reachable from Power BI after firewall/IP config; both connection methods tested.
- **Data quality:** no null/format anomalies in critical fields post-cleaning; sample queries match report aggregates.
- **DAX correctness:** calculated columns validate against hand-checks or SQL equivalents.
- **UX & design:** pages render at custom size with consistent branding; AppSource **scroller** functions as expected.

Data Assessment & Cleaning Tools

▼ Dataset Information

Column Name	Description
Brand	The brand names of the products
Title	The description of the products
Original Price	The market price or the list price
Sale Price	The actual price at which the product was sold

▼ Data Cleaning (Azure SQL Database)

A screenshot of the Azure Data Studio interface. On the left, the Object Explorer shows a database structure with tables like 'dbo.Men+Tshirt' containing columns 'Brand', 'Title', 'Original_Price', and 'Sale_Price'. In the center, the Query Editor window displays a T-SQL script for cleaning prices:

```
3 UPDATE [dbo].[Men+Tshirt]
4 SET original_price = TRIM(REPLACE(CAST(original_price) AS VARCHAR(MAX), '?', '')) 
5 WHERE original_price LIKE '%?%' 
6 
7 UPDATE [dbo].[Men+Tshirt]
8 SET sale_price = TRIM(REPLACE(CAST(sale_price) AS VARCHAR(MAX), '?', '')) 
9 WHERE sale_price LIKE '%?%'
```

The results pane shows a table with sample data:

Brand	Title	Original_Price	Sale_Price
The Indian Garage Co	Men Striped Slim Fit Shirt with Pat...	1,749	525

At the bottom, a message indicates the query succeeded.

▼ Data Cleaning: Price Standardization (Power Query)

Normalize pricing fields so every record has a reliable **Mark Price** (list/original price) while keeping the model lean and analytics-ready.

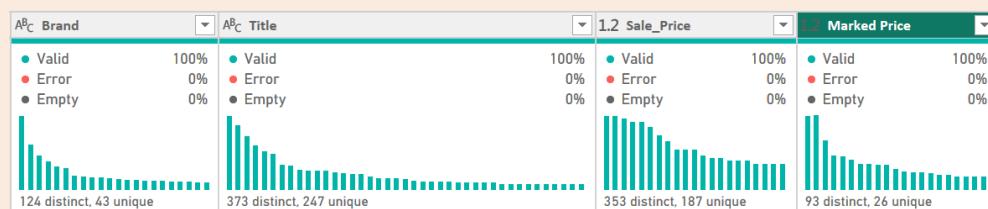
- Row quality:** Drop rows where both `Original_Price` and `Sales_Price` are missing (`NA / N/A /null`).
- Imputation logic:** If `Original_Price` is missing but `Sales_Price` exists → set `Marked_Price = Sales_Price × 1.5` (list price assumed 50% above sales). Otherwise → `Marked_Price = Original_Price`.

Two side-by-side screenshots of the Power Query 'Add Conditional Column' dialog. Both dialogs show a similar structure:

- New column name: `Marked_Price`
- Condition: `If [Original_Price] equals [NA]`
- Value: `[Sales_Price] * 1.5`
- Output: `SelectFactor`

The right dialog also includes an 'Else' clause with the value `[Original_Price]`.

- Types:** Enforce **Decimal Number** for all price columns.
- Tidy model:** Remove helper columns used exclusively for the imputation step.



DAX Functions

Discount % = DIVIDE('Tshirt'[Marked Price]-'Tshirt'[Sales Price],'Tshirt'[Marked Price])*100



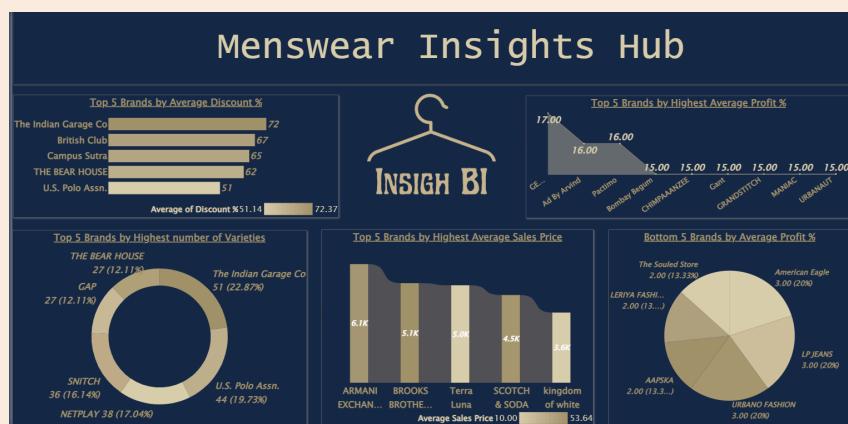
Cost Price = DIVIDE(100*'Tshirt'[Sales Price],100+'Tshirt'[Profit %])

Profit % = RANDBETWEEN(2,17)

Project Showcase

↗ [Detailed Report \(For More Information\)](#)

View the Live Dashboard: <https://app.powerbi.com/reportEmbed?reportId=d6338b69-bee6-402e-a5b1-6322f673a3eb&autoAuth=true&ctid=9f40849d-a657-43a5-85dc-4bd96886bad5>





Navigation bar