Preppin Data 2022-WK29: C&BSco Meeting Targets?

Table of Content

- 1. Link to challenge
- 2. Requirements
- 3. Screenshot solution
- 4. Notebook cells
- 5. Final output
- 6. Reference materials

𝒪 Link to challenge

https://preppindata.blogspot.com/2022/07/2022-week-29-c-meeting-targets.html

© Requirements

- · Input both data sets
- Remove unnecessary values from the Product Name field to just leave the Product Type
- Total Sales for each Store and Product Type
- Change the Targets data set into three columns
 - Product
 - o Store
 - Sales Target (k's)
- Multiple the Sales Target (k's) by 1000 to create the full sales target number (i.e. 75 becomes 75000)
- Prepare your data sets for joining together by choosing your next step:
 - Easy make your Sales input Product Type and Store name UPPER CASE
 - Hard make your Targets' Store and Product fields TitleCase
- Join the data sets together and remove any duplicated fields
- Calculate whether each product in each store beats the target
- Output the results

Screenshot solution

The below is a screenshot of what the solution looks like in the notebook. You can view and download the final output here. Check my Github repository for the compiled SQL and a copy of the notebook markdown.

Notebook cells

all_data_cbsco

-- same data as for the challenge from week 27

SELECT * FROM preppindata.`2022w27_cbsco`;

Untitled

Sale_Date	Order_ID	Sale_Value	Product_Name	Store_Name	Region
2022-12-15T00:00:00.000Z	770	143.84	Liquid - 750ml	Chelsea	West
2022-12-15T00:00:00.000Z	770	143.84	Liquid - 1L	Chelsea	West
2022-12-15T00:00:00.000Z	770	143.84	Bar - 2x	Chelsea	West
2022-08-04T00:00:00.000Z	515	90.17	Liquid - 250ml	Chelsea	West
2022-08-04T00:00:00.000Z	515	90.17	Bar - 1x	Chelsea	West
☐ 7 columns · 4.063 rows ①	1.4 seconds (15 minutes ago)			

all_data_targets

SELECT * FROM preppindata.`2022w29_targets`;

Untitled

PRODUCT	CHELSEA	DULWICH	LEWISHAM	NOTTING_HILL	SHOREDITCH	WIMBLEDON
BAR	25	30	35	30	30	35
LIQUID	60	75	75	65	70	70

7 columns · 2 rows © 0.7 seconds (15 minutes ago)

step_1

-- clean products table and only keep the product type (liquid - bar) from product name

SELECT

*,

TRIM(SPLIT(Product_Name,"-")[ORDINAL(1)]) AS Product_Type

--^^ take the first value (ordinal position = 1) from the array created by the SPLIT funct:
leading and trailing spaces and rename the column as Product_Type
FROM preppindata.`2022w27_cbsco`;

Untitled

Sale_Date	Order_ID	Sale_Value	Product_Name	Store_Name	Region
2022-12-15T00:00:00.000Z	770	143.84	Liquid - 750ml	Chelsea	West
2022-12-15T00:00:00.000Z	770	143.84	Liquid - 1L	Chelsea	West
2022-12-15T00:00:00.000Z	770	143.84	Bar - 2x	Chelsea	West
2022-08-04T00:00:00.000Z	515	90.17	Liquid - 250ml	Chelsea	West
2022-08-04T00:00:00.000Z	515	90.17	Bar - 1x	Chelsea	West
8 columns · 4,063 rows	1.3 seconds (1	5 minutes ago)			

step_2

-- adding up sales values per store and product type

SELECT

Store_Name,

Product_Type,

Region,

SUM(Sale_Value) AS Total_Sales

FROM step_1

GROUP BY Store_Name, Product_Type, Region;

Store_Name	Product_Type	Region	Total_Sales	
Chelsea	Liquid	West	59,640.5	
Chelsea	Bar	West	29,245.81	
Dulwich	Liquid	East	76,457.58	
Dulwich	Bar	East	30,156.21	
Lewisham	Liquid	East	78,734.58	
4 columns	12 rows 🕒 0.7 se	conds (15 min	utes ago)	

step_3

-- reshaping the targets table

SELECT *

FROM all_data_targets

UNPIVOT (Target FOR Store IN (CHELSEA, DULWICH, LEWISHAM, NOTTING_HILL, SHOREDITCH, WIMBLE

PRODUCT	Target	Store
BAR	25	CHELSEA
BAR	30	DULWICH
BAR	35	LEWISHAM
BAR	30	NOTTING_HILL
BAR	30	SHOREDITCH
3 columns	· 12 rows	© 0.8 seconds (Just now

alternative_for_unpivot_step_1

```
-- creating an array of structs
-- an array is a collection of variables of the same data type, individual variables within accessed by their index number.
-- a struct is a collection of variables from different data types, variables are accessed

SELECT
Product,
[
STRUCT('CHELSEA' AS Store, Chelsea AS Target),
STRUCT('DULWICH' AS Store, Dulwich AS Target),
STRUCT('LEWISHAM' AS Store, Lewisham AS Target),
STRUCT('NOTTING_HILL' AS Store, Notting_Hill AS Target),
STRUCT('SHOREDITCH' AS Store, Shoreditch AS Target),
STRUCT('WIMBLEDON' AS Store, Wimbledon AS Target)
] AS Target_Struct
FROM preppindata.`2022w29_targets`;
```

Untitled

Product	Target_Struct
BAR	[{"Store":"CHELSEA","Target":25},{"Store":"DULWICH","Ta
LIQUID	[{"Store":"CHELSEA","Target":60},{"Store":"DULWICH","Ta

2 columns · 2 rows © 0.7 seconds (Just now)

alternative_for_unpivot_step_2

SELECT *
FROM alternative_for_unpivot_step_1 AS t CROSS JOIN
UNNEST(t.Target_Struct) AS s;

Product	Target_Struct	Store	Target
BAR	[{"Store":"CHELSEA","Target":25},{"Store":"DULWICH","Ta	CHELSEA	25
BAR	[{"Store":"CHELSEA","Target":25},{"Store":"DULWICH","Ta	DULWICH	30
BAR	[{"Store":"CHELSEA","Target":25},{"Store":"DULWICH","Ta	LEWISHAM	35
BAR	[{"Store":"CHELSEA","Target":25},{"Store":"DULWICH","Ta	NOTTING_HILL	30
BAR	[{"Store":"CHELSEA","Target":25},{"Store":"DULWICH","Ta	SHOREDITCH	30
£ 4 colum	nns · 12 rows © 0.9 seconds (Just now)		

step_4

-- recalculating the target and fixing the spelling for product type and store

SELECT

INITCAP(Product) as Product_Type, Target * 1000 AS Target, INITCAP(Store) AS Store FROM step_3;

Untitled

Product_Type	Target	Store
Bar	25,000	Chelsea
Bar	30,000	Dulwich
Bar	35,000	Lewisham
Bar	30,000	Notting_Hill
Bar	30,000	Shoreditch

step_5

-- fixing the spelling for Notting Hill in the targets table so the join works correctly (Notting Hill to Notting_Hill)

SELECT

Product_Type,

CASE WHEN Store = "Notting_Hill" THEN "Notting Hill"

ELSE Store END AS Store_Name,

Target,

FROM step_4;

Untitled

Product_Type	Store_Name	Target
Bar	Chelsea	25,000
Bar	Dulwich	30,000
Bar	Lewisham	35,000
Bar	Notting Hill	30,000
Bar	Shoreditch	30,000

3 columns · 12 rows © 0.8 seconds (Just now)

A Final output

output

```
-- bringing it all together
```

-- showing which stores beat their targets

SELECT

- s.Store_Name,
- s.Region,
- s.Product_Type,
- s.Total_Sales,
- t.Target,

IF (s.Total_Sales > t.Target, "Yes", "No") AS Beats_Target

FROM step_2 AS s -- short table stores

LEFT JOIN step_5 AS t -- targets

ON s.Product_Type = t.Product_Type AND s.Store_Name = t.Store_Name;

Chelsea Wes Chelsea Wes Dulwich East	st Bar	59,640.5 29,245.81	60,000 25,000	No Yes	
		29,245.81	25,000	Yes	
Dulwich Fast					
Daiwicii Last	t Liquid	76,457.58	75,000	Yes	
Dulwich East	t Bar	30,156.21	30,000	Yes	
Lewisham East	t Liquid	78,734.58	75,000	Yes	

Reference materials

Functions & operators used

TRIM: trims the leading and trailing spaces from a string

<u>SPLIT</u>: splits the values in a string using the delimiter argument and returns an array of those values

UNPIVOT: an operator that rotates columns into rows, is part of the FROM clause

<u>INITCAP</u>: takes a string and returns it with the first character of each word in uppercase and all others in lowercase, non-alphabetic characters remain the same.

Blogs / Snippets

Using UNPIVOT in BigQuery

<u>Using UNNEST in BigQuery</u> (for alternative solution to reshape the targets table)

<u>Exploring ARRAYS and STRUCTS in Google BigQuery</u> (for alternative solution to reshape the targets table)