

Provided with a dataset Sales Case Study which indicates the daily trading information for a large retail store, aggregated . The data is only for one of the products sold at this store. Used the dataset, to develop metrics and derive subsequent insights. Initially data had to be sorted and filtered in a structured manner, date format had to be updated to make it easier to use the data.

```
CSSALES.PUBLIC  Settings
1  SELECT * from CSSALES.PUBLIC.CSDATASET;
2
3  ---UPDATE CSDATASET SET DATE = REPLACE (DATE, '/', '/');
4  UPDATE CSDATASET
5  SET DATE = REPLACE (DATE, '/', '-');
6
7  ---CREATING UPDATED TABLE.
8  CREATE OR REPLACE TABLE SALES AS (
9  SELECT TO_DATE (DATE, 'DD-MM-YYYY') AS DATE_, *
10 FROM CSDATASET);
11
12 SELECT* FROM SALES;
13
14 ALTER TABLE SALES DROP COLUMN date;
15
16 --- MAKING COST OF SALES TO HAVE 2 DECIMAL POINTS
17
18 SELECT TRUNCATE (SALES,2) AS SALES,
19 TRUNCATE (COST_OF_SALES,2) AS COSTOFSALE,
20 FROM SALES;
21
```

	🕒 DATE_	📅 MONTH_NAME	📅 DAY	# SALES	# COST_OF_SALES	# QUANTITY_SOLD	# PRICE	# SALES	# COST_OF_SALES	# QUANTITY_SOLD	# PRICE_PER_UNIT	# GROSS_PROFIT
1	2013-12-30	Dec 2013	Mon	223937.96	230079.62	6827		223937.96	230079.62	6827	32.80	-6141.66
2	2013-12-31	Dec 2013	Tue	300345.48	306986.12	9268		300345.48	306986.12	9268	32.41	-6640.64
3	2014-01-01	Jan 2014	Wed	86782.46	87986.31	2678		86782.46	87986.31	2678	32.41	-1203.85
4	2014-01-02	Jan 2014	Thu	200173.11	202881.17	6175		200173.11	202881.17	6175	32.42	-2708.06
5	2014-01-03	Jan 2014	Fri	326906.07	333806.29	10084		326906.07	333806.29	10084	32.42	-6900.22
6	2014-01-04	Jan 2014	Sat	307043.93	313652.34	9470		307043.93	313652.34	9470	32.42	-6608.41
7	2014-01-05	Jan 2014	Sun	179188.88	183083.28	5524		179188.88	183083.28	5524	32.44	-3894.40
								306351.95	313446.06	9448	32.43	-7094.11

1. The daily sales price per unit?

```
---PRICE PER UNIT
SELECT ROUND(SALES/QUANTITY_SOLD, 2) AS PRICE_PER_UNIT
FROM SALES;

CREATE OR REPLACE TABLE SALES AS
  (SELECT DATE_,
    TO_CHAR(DATE_, 'MON YYYY') MONTH_NAME,
    TO_CHAR(DATE_, 'DY') AS DAY ,
    TRUNCATE(SALES,2) AS SALES,
    TRUNCATE(COST_OF_SALES,2) COST_OF_SALES ,
    QUANTITY_SOLD, ROUND(SALES/QUANTITY_SOLD,2) AS PRICE_PER_UNIT,
    (SALES-COST_OF_SALES) GROSS_PROFIT
  FROM SALES);

SELECT * FROM SALES;
--- AVERAGE UNIT SALE PRICE
SELECT SUM(SALES) AS TOTAL_SALES
FROM SALES;

SELECT SUM(QUANTITY_SOLD) AS TOTAL_UNIT_SOLD
FROM SALES;
```

Results

Chart

	🕒 DATE_	📅 MONTH_NAME	📅 DAY	# SALES	# COST_OF_SALES	# QUANTITY_SOLD	# PRICE
1	2013-12-30	Dec 2013	Mon	223937.96	230079.62	6827	
2	2013-12-31	Dec 2013	Tue	300345.48	306986.12	9268	
3	2014-01-01	Jan 2014	Wed	86782.46	87986.31	2678	
4	2014-01-02	Jan 2014	Thu	200173.11	202881.17	6175	
5	2014-01-03	Jan 2014	Fri	326906.07	333806.29	10084	
6	2014-01-04	Jan 2014	Sat	307043.93	313652.34	9470	
7	2014-01-05	Jan 2014	Sun	179188.88	183083.28	5524	

2. What is the average unit sales price of this product?

CSSALES.PUBLIC Settings Code Versions

```
38 SELECT * FROM SALES;  
39 --- AVERAGE UNIT SALE PRICE  
40 SELECT SUM(SALES) AS TOTAL_SALES  
41 FROM SALES;  
42  
43 SELECT SUM(QUANTITY_SOLD) AS TOTAL_UNIT_SOLD  
44 FROM SALES;  
45  
46 SELECT SUM(Sales) / SUM(Quantity_Sold) AS Average_Unit_Sale_Price  
47 FROM SALES;  
48
```

Results Chart

	# AVERAGE_UNIT_SALE_PRICE
1	35.40044590

Query Details

Query duration 113ms

Rows 1

Query ID 01bc75a3-0000-89df-0...

### 3. What is the daily % gross profit?

CSSALES.PUBLIC ▾ Settings ▾

Code Versions 🔍

```
49 ---3.DAILY % GROSS PROFIT
50 SELECT DATE_,
51        SALES,
52        COST_OF_SALES,
53        ((SALES - COST_OF_SALES) / SALES) *100 AS DAILY_PERCENT_GROSS_PROFIT
54 FROM SALES;
55
```

🔍 Results Chart

🔍 🗑️ ⬇️ 📄 ⌚

	🕒 DATE_	# SALES	# COST_OF_SALES	# DAILY_PERCENT_GROSS_PROFIT
1	2013-12-30	223937.96	230079.62	-2.74257200
2	2013-12-31	300345.48	306986.12	-2.21100000
3	2014-01-01	86782.46	87986.31	-1.38720400
4	2014-01-02	200173.11	202881.17	-1.35285900
5	2014-01-03	326906.07	333806.29	-2.11076500

Query Details ⋮

Query duration 102ms

Rows 1.1K

Query ID 01bc75a5-0000-89ea-0...

Show more ▾

#### 4. What is the daily % gross profit per unit?

CSSALES.PUBLIC ▾ Settings ▾

Code Versions 🔍

```
55  
56  
57 ---4. What is the daily % gross profit per unit?  
58 SELECT DATE_,  
59        SALES,  
60        COST_OF_SALES,  
61        ((SALES - COST_OF_SALES) / QUANTITY_SOLD) AS GROSS_PROFIT_PER_UNIT  
62 FROM SALES;  
63
```

↶ Results

~ Chart

🔍 🗑️ ⬇️ 📄 ⌚

	🕒 DATE_	# SALES	# COST_OF_SALES	# GROSS_PROFIT_PER_UNIT
1	2013-12-30	223937.96	230079.62	-0.89961330
2	2013-12-31	300345.48	306986.12	-0.71651273
3	2014-01-01	86782.46	87986.31	-0.44953323
4	2014-01-02	200173.11	202881.17	-0.43855223
5	2014-01-03	326906.07	333806.29	-0.68427410

Query Details ⋮

Query duration 133ms

Rows 1.1K

Query ID 01bc75a8-0000-89df-0...

Show more ▾



5. Pick any 3 periods during which this product was on promotion/special:

CSSALES.PUBLIC

Settings

Code Versions

75

76

77

78

79

80

81

82

83

```
SELECT DATE_,
        QUANTITY_SOLD,
        PRICE_PER_UNIT,
        MIN(PRICE_PER_UNIT)
FROM SALES
GROUP BY DATE_, QUANTITY_SOLD, PRICE_PER_UNIT
ORDER BY MIN(PRICE_PER_UNIT) ASC
LIMIT 3;
```

Results

Chart

Query Details

	🕒 DATE_	# QUANTITY_SOLD	# PRICE_PER_UNIT	# MIN(PRICE_PER_UNIT)	
1	2014-09-02	15291	30.70	30.70	Query duration 117ms
2	2014-08-28	8400	30.72	30.72	Rows 3
3	2014-09-01	19366	30.72	30.72	Query ID 01bc75ab-0000-89ea-0...



```
106  
107 ---ELASTIC OF DEMAND COEFFICIENT  
108  
109 SELECT * FROM SALES;  
110  
111 CREATE OR REPLACE TEMPORARY TABLE  
112 SELECT DATE_,  
113         QUANTITY_SOLD,  
114         SALES,  
115         (PERCENTAGE_Change_Quantity / PERCENTAGE_Change_SALES) AS ELASTIC_OF_DEMAND  
116 FROM SALES;  
117  
118 ---Price_Change_Percent < -10: Unit sale price dropped by more than 10%.  
119 -----Quantity_Change_Percent > 10: Quantity sold increased by more than 10%.  
120 ---Rows meeting either condition might indicate a promotion or special.|  
121 WITH UNIT_DATA AS (  
122     SELECT  
123         DATE_,  
124         SALES,  
125         QUANTITY_SOLD,  
126         SALES / QUANTITY_SOLD AS Unit_Sale_Price  
127     FROM SALES  
128 ),
```

```
129 price_changes AS (  
130     SELECT  
131         curr.DATE_,  
132         curr.UNIT_SALE_PRICE,  
133         prev.Unit_Sale_Price AS Prev_UNIT_SALE_PRICE,  
134         curr.QUANTITY_SOLD,  
135         prev.QUANTITY_SOLD AS Prev_QUANTITY_SOLD,  
136         ((curr.UNIT_Sale_PRICE - prev.UNIT_SALE_PRICE) / prev.UNIT_SALE_PRICE) * 100 AS Price_Change_Percent,  
137         ((curr.Quantity_Sold - prev.Quantity_Sold) / prev.Quantity_Sold) * 100 AS Quantity_Change_Percent  
138     FROM unit_data curr  
139     JOIN unit_data prev  
140         ON curr.DATE_ = DATEADD(DAY, 1, prev.DATE_)  
141 )  
142 SELECT *  
143 FROM price_changes  
144 WHERE Price_Change_Percent < -10 OR Quantity_Change_Percent > 10;  
145  
146 ---don't need explicit "specials" column-patterns in unit price and quantity sold reveal it.  
147 --"Possible Special": Unit price is more than 10% below average, indicating a possible discount.  
148  
149 You can adjust the 0.9 threshold to be more or less sensitive.  
150 WITH base AS (  
151     SELECT  
152         Date_,
```

Activate Windows

Go to Settings to activate Windows.

1	Date	DIFFERENCE % CHANGE= NEW- OLD/OLD	% CHANGE IN QTY	DIFF % CHANGE= NEW-OLD/OLD	% CHANGE IN PRICE	ELASTIC OF DEMAND	INELASTIC OR ELATIC
2	8/28/2014	-509	-5.713323605	-16646.0839	-6.059461746	0.942876421	INELASTIC
3	8/29/2014	10917	129.9642857	335941.0438	130.1763205	0.998371172	INELASTIC
4	8/30/2014	3993	20.67091163	123089.4996	20.72188634	0.997540054	INELASTIC
5	8/31/2014	-11664	-50.03861004	-358925.7165	-50.05262332	0.999720029	INELASTIC
6	9/1/2014	7720	66.28885454	236782.6386	66.10882541	1.002723224	ELASTIC
7	9/2/2014	-4075	-21.04203243	-125483.6939	-21.09133999	0.997662189	INELASTIC
8	9/3/2014	-191	-1.249100778	-5402.1417	-1.150689582	1.08552367	ELASTIC
9	9/4/2014	-4936	-32.68874172	-151755.8453	-32.70122287	0.999618328	INELASTIC
10							