

Customer & Product Sales Analysis Using SQL

A data-driven project to extract business insights
from a retail dataset using advanced SQL queries

 Gmail: srushtigavandi@gmail.com

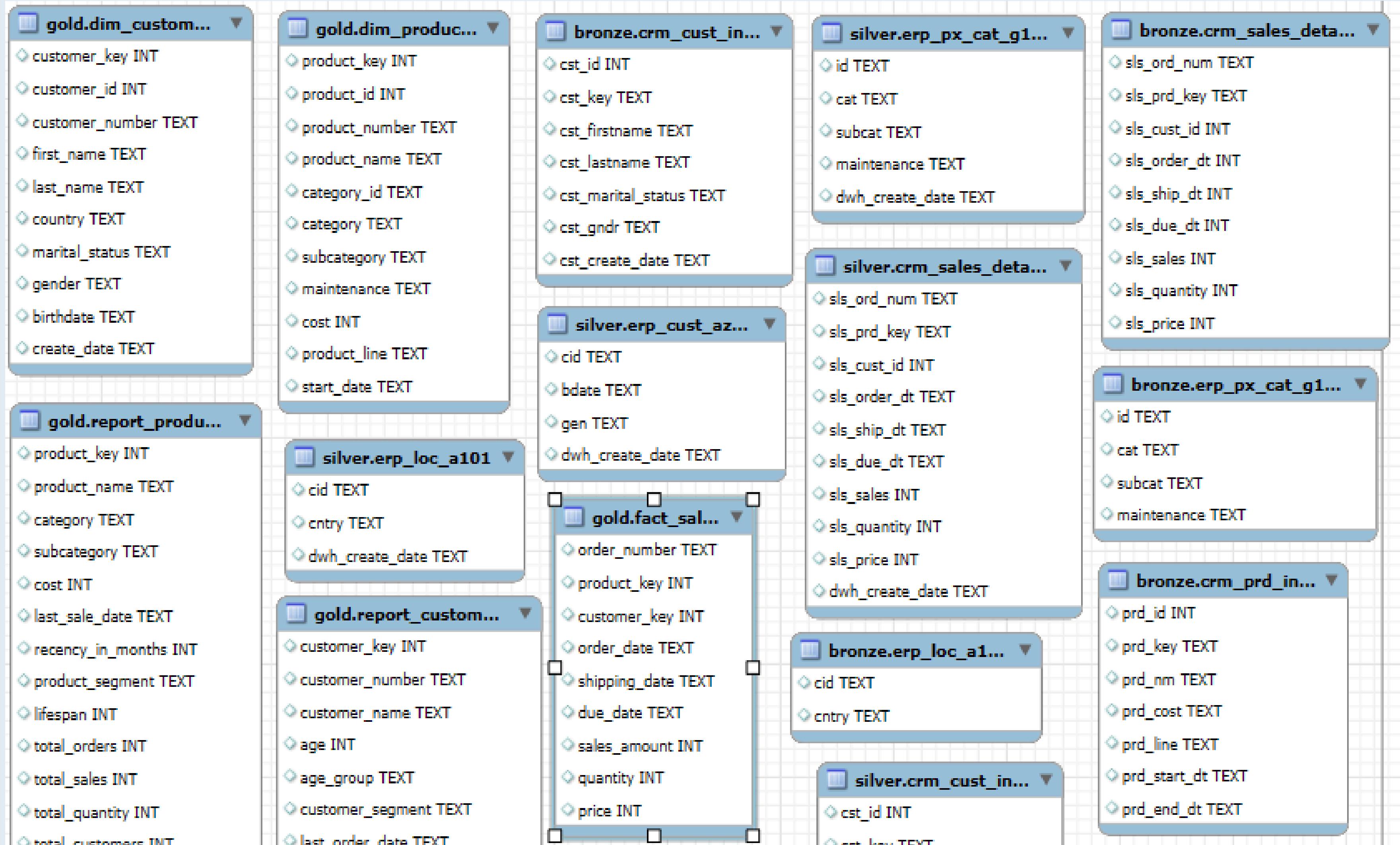
 LinkedIn: linkedin.com/in/srushti-gavandi

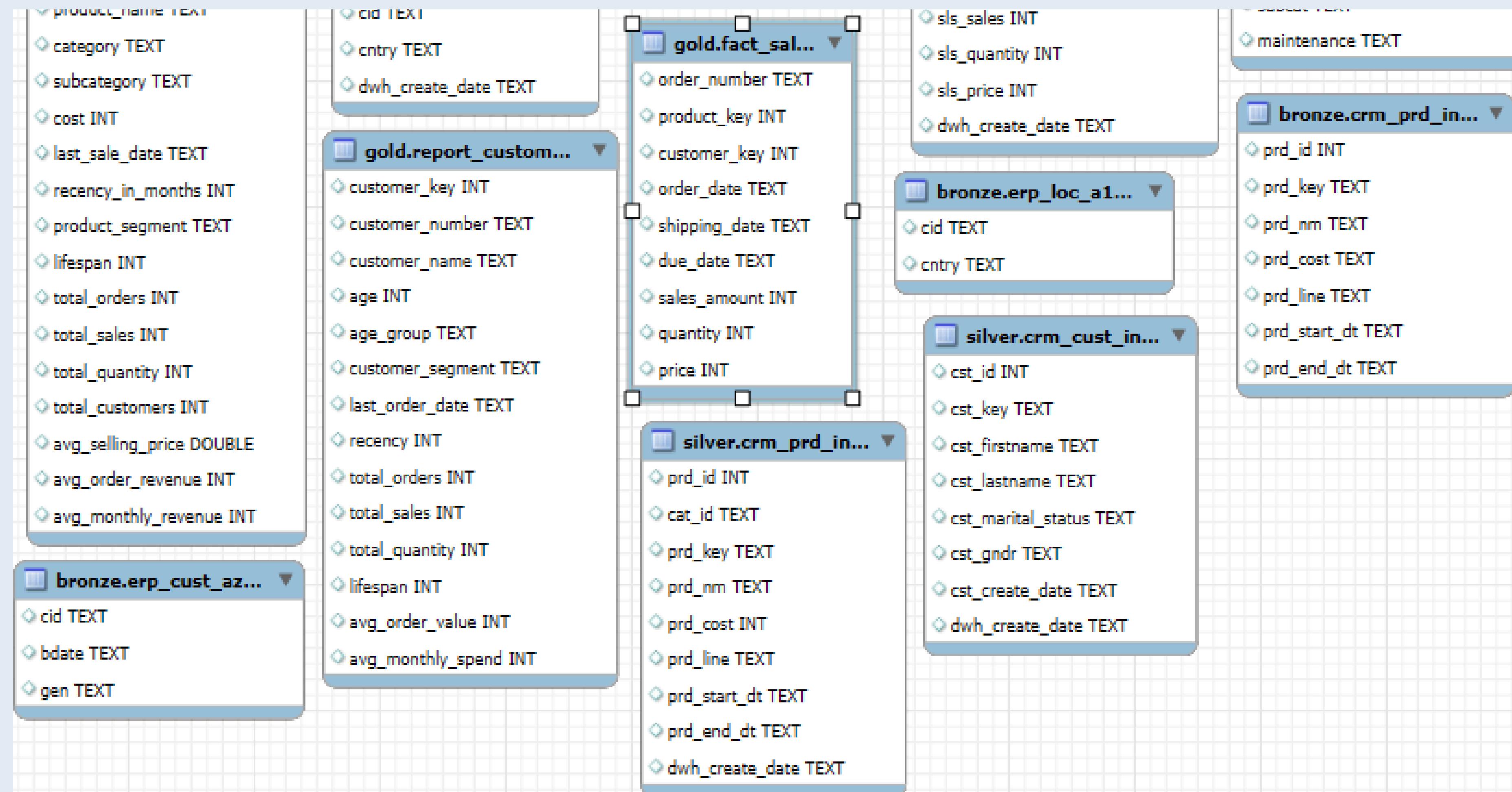
 GitHub: github.com/SrushtiGavandi



While this project is inspired by an existing source, the SQL queries have been adapted to suit MySQL Workbench, resulting in a different query structure and implementation approach







1 . CHANGE OVER TIME

Analyze how a measure evolves over time
- total sales by year

SELECT

```
YEAR(ORDER_DATE) AS YEAR ,  
SUM(SALES_AMOUNT) AS TOTAL_SALES ,  
SUM(QUANTITY) AS TOTAL_QUANTITY ,  
COUNT(DISTINCT ORDER_NUMBER) AS ORDER_NO
```

FROM `GOLD.FACT_SALES`

WHERE ORDER_DATE IS NOT NULL

GROUP BY YEAR

ORDER BY YEAR ;

YEAR	TOTAL_SALES	TOTAL_QUANTITY	ORDER_NO
NULL	4992	19	15
2010	43419	14	14
2011	7075088	2216	2216
2012	5842231	3397	3269
2013	16344878	52807	21287
2014	45642	1970	871

CUMULATIVE ANALYSIS

Aggregate the data progressively over time .Helps to understand whether our business is growing or declining

QUE - Calculate the total sales per month and the running total of sales over time

```

SELECT DATE_FORMAT(ORDER_DATE , '%y-%m') AS MONTH ,
SUM(SALES_AMOUNT) AS MONTHLY_SALE ,
SUM(SUM(SALES_AMOUNT)) OVER(ORDER BY DATE_FORMAT(ORDER_DATE , '%y-%m'))AS CUMULATIVE_MONTHLY_SALE
FROM `GOLD.FACT_SALES`
WHERE ORDER_DATE IS NOT NULL
GROUP BY MONTH
ORDER BY MONTH;
    
```

MONTH	MONTHLY_SALE	CUMULATIVE_MONTHLY_SALE
13-02	771218	14594706
13-03	1049732	15644438
13-04	1045860	16690298
13-05	1284456	17974754
13-06	1642948	19617702
13-07	1371595	20989297
13-08	1545910	22535207
13-09	1447324	23982531
13-10	1673261	25655792
13-11	1780688	27436480
13-12	1874128	29310608
14-01	45642	29356250

	MONTH	MONTHLY_SALE	CUMULATIVE_MONTHLY_SALE
►	NULL	4992	4992
	10-12	43419	48411
	11-01	469795	518206
	11-02	466307	984513
	11-03	485165	1469678
	11-04	502042	1971720
	11-05	561647	2533367
	11-06	737793	3271160
	11-07	596710	3867870
	11-08	614516	4482386
	11-09	603047	5085433
	11-10	708164	5793597

	MONTH	MONTHLY_SALE	CUMULATIVE_MONTHLY_SALE
	11-11	660507	6454104
	11-12	669395	7123499
	12-01	495363	7618862
	12-02	506992	8125854
	12-03	373478	8499332
	12-04	400324	8899656
	12-05	358866	9258522
	12-06	555142	9813664
	12-07	444533	10258197
	12-08	523887	10782084
	12-09	486149	11268233
	12-10	535125	11803358

	MONTH	MONTHLY_SALE	CUMULATIVE_MONTHLY_SALE
	12-11	537918	12341276
	12-12	624454	12965730
	13-01	857758	13823488
	13-02	771218	14594706
	13-03	1049732	15644438
	13-04	1045860	16690298
	13-05	1284456	17974754
	13-06	1642948	19617702
	13-07	1371595	20989297
	13-08	1545910	22535207
	13-09	1447324	23982531
	13-10	1673261	25655792

PERFORMANCE ANALYSIS

Comparing current value to targeted value
helps measure success and compare success

Analyze the yearly performance of the products by comparing each product's sales to both its average sales performance and previous year sales performance

```
33 • Ⓜ WITH PRODUCT_YEARLY_SALE AS (
34     SELECT S.PRODUCT_KEY ,
35         P.PRODUCT_NAME ,
36         YEAR(S.ORDER_DATE) AS SALES_YEAR ,
37         SUM(S.SALES_AMOUNT) AS YEARLY_SALES
38     FROM `GOLD.FACT_SALES` AS S
39     JOIN `GOLD.DIM_PRODUCTS` AS P ON
40         S.PRODUCT_KEY = P.PRODUCT_KEY
41     GROUP BY S.PRODUCT_KEY ,
42         P.PRODUCT_NAME ,
43         YEAR(S.ORDER_DATE)),
44
45 Ⓜ AVG_AND_PREVIOUS_YEAR_SALES AS(
46     SELECT
47         PRODUCT_KEY , PRODUCT_NAME , SALES_YEAR , YEARLY_SALES ,
48         AVG(YEARLY_SALES) OVER(PARTITION BY PRODUCT_KEY) AS AVG_YEARLY_SALES ,
49         LAG(YEARLY_SALES) OVER (PARTITION BY PRODUCT_KEY ORDER BY SALES_YEAR) AS PREVIOUS_YEAR_SALE
50     FROM PRODUCT_YEARLY_SALE )
51
```

```

45    AVG_AND_PREVIOUS_YEAR_SALES AS(
46        SELECT
47            PRODUCT_KEY , PRODUCT_NAME , SALES_YEAR , YEARLY_SALES ,
48            AVG(YEARLY_SALES) OVER(PARTITION BY PRODUCT_KEY) AS AVG_YEARLY_SALES ,
49            LAG(YEARLY_SALES) OVER (PARTITION BY PRODUCT_KEY ORDER BY SALES_YEAR) AS PREVIOUS_YEAR_SALE
50        FROM PRODUCT_YEARLY_SALE )
51
52        SELECT
53            PRODUCT_KEY , PRODUCT_NAME , SALES_YEAR , YEARLY_SALES , AVG_YEARLY_SALES ,PREVIOUS_YEAR_SALE ,
54        CASE
55            WHEN YEARLY_SALES >= AVG_YEARLY_SALES THEN 'ABOVE AVG'
56            ELSE 'BELOW AVG'
57        END AS SALES_VS_AVG
58        FROM AVG_AND_PREVIOUS_YEAR_SALES
59        ORDER BY PRODUCT_KEY , SALES_YEAR ;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	PRODUCT_KEY	PRODUCT_NAME	SALES_YEAR	YEARLY_SALES	AVG_YEARLY_SALES	PREVIOUS_YEAR_SALE	SALES_VS_AVG
▶	3	Mountain-100 Black- 38	2011	165375	165375.0000	NULL	ABOVE AVG
	4	Mountain-100 Black- 42	2011	151875	151875.0000	NULL	ABOVE AVG
	5	Mountain-100 Black- 44	2011	202500	202500.0000	NULL	ABOVE AVG
	6	Mountain-100 Black- 48	2010	3375	96187.5000	NULL	BELOW AVG
	6	Mountain-100 Black- 48	2011	189000	96187.5000	3375	ABOVE AVG
	7	Mountain-100 Silver- 38	2010	3400	98600.0000	NULL	BELOW AVG
	7	Mountain-100 Silver- 38	2011	193800	98600.0000	3400	ABOVE AVG
	8	Mountain-100 Silver- 42	2011	142800	142800.0000	NULL	ABOVE AVG
	9	Mountain-100 Silver- 44	2010	10200	83300.0000	NULL	BELOW AVG
	9	Mountain-100 Silver- 44	2011	156400	83300.0000	10200	ABOVE AVG
	10	Mountain-100 Silver- 48	2011	122400	122400.0000	NULL	ABOVE AVG

Note: The data shown below is for demonstration purposes only and includes sample entries to illustrate the query results. Additional data is present beyond what is displayed .

PART TO WHOLE ANALYSIS

Which category contributes the most to overall sales

```
SELECT P.CATEGORY ,  
SUM(S.SALES_AMOUNT) AS TOTAL_SALES ,  
ROUND((SUM(S.SALES_AMOUNT)/(SELECT SUM(SALES_AMOUNT) FROM `GOLD.FACT_SALES`)) * 100 , 2) AS PER_CONTRI  
FROM `GOLD.FACT_SALES` AS S JOIN  
`GOLD.DIM_PRODUCTS` AS P  
ON S.PRODUCT_KEY = P.PRODUCT_KEY  
GROUP BY P.CATEGORY  
ORDER BY TOTAL_SALES DESC;
```

Result Grid | Filter Rows: E

	CATEGORY	TOTAL_SALES	PER_CONTRI
▶	Bikes	28316272	96.46
	Accessories	700262	2.39
	Clothing	339716	1.16

DATA SEGMENTATION

GROUP THE DATA BASED ON SPECIFIC RANGE

HELPS UNDERSTAND THE CORRELATION BETWEEN TWO MEASURES

QUE - SEGMENT PRODUCTS INTO RANGES AND COUNT HOW MANY PRODUCTS FALL INTO EACH CATEGORY

SELECT

CASE

```
WHEN cost < 100 THEN 'BELOW 100'  
WHEN cost BETWEEN 100 AND 499 THEN '100 - 499'  
WHEN cost BETWEEN 500 AND 999 THEN '500 - 999'  
WHEN cost >= 1000 THEN '1000 and ABOVE'  
ELSE 'UNKNOWN'  
END AS cost_range,  
COUNT(*) AS total_products  
FROM `gold.dim_products`  
GROUP BY cost_range  
ORDER BY total_products DESC;
```

	cost_range	total_products
▶	BELOW 100	110
	100 - 499	101
	500 - 999	45
	1000 and ABOVE	39

CUSTOMER REPORT

```
SELECT C.CUSTOMER_KEY, C.FIRST_NAME,C.LAST_NAME ,C.COUNTRY,  
COUNT(DISTINCT S.ORDER_NUMBER) AS TOTAL_ORDERS,  
SUM(S.SALES_AMOUNT) AS TOTAL_SALES,  
ROUND(AVG(S.SALES_AMOUNT), 2) AS AVG_ORDER_VALUE  
FROM  
`GOLD.FACT_SALES` AS S  
JOIN  
`GOLD.DIM_CUSTOMERS` AS C  
ON S.CUSTOMER_KEY = C.CUSTOMER_KEY  
GROUP BY  
C.CUSTOMER_KEY, C.FIRST_NAME,C.LAST_NAME , C.COUNTRY  
ORDER BY  
TOTAL_SALES DESC;
```

	CUSTOMER_KEY	FIRST_NAME	LAST_NAME	COUNTRY	TOTAL_ORDERS	TOTAL_SALES	AVG_ORDER_VALUE
▶	1133	Kaitlyn	Henderson	France	5	13294	949.57
	1302	Nichole	Nara	France	5	13294	1022.62
	1309	Margaret	He	France	5	13268	947.71
	1132	Randall	Dominguez	France	5	13265	1205.91
	1301	Adriana	Gonzalez	France	5	13242	1324.20
	1322	Rosa	Hu	France	5	13215	881.00
	1125	Brandi	Gill	France	5	13195	1099.58
	1308	Brad	She	France	5	13172	1197.45
	1297	Fransisco	Sara	France	5	13164	1097.00
	434	Maurice	Shan	France	6	12914	1076.17
	440	Janet	Munoz	France	6	12488	892.00
	242	Lisa	Cai	France	7	11468	458.72

PRODUCT PERFORMANCE REPORT

```
SELECT P.PRODUCT_KEY,P.PRODUCT_NAME,P.CATEGORY,P.SUBCATEGORY,
COUNT(DISTINCT S.ORDER_NUMBER) AS TOTAL_ORDERS,
SUM(S.QUANTITY) AS TOTAL_QUANTITY SOLD,
SUM(S.SALES_AMOUNT) AS TOTAL_SALES,
ROUND(SUM(S.SALES_AMOUNT) / NULLIF(SUM(S.QUANTITY), 0), 0) AS AVG_SELLING_PRICE
FROM
`GOLD.FACT_SALES` AS S
JOIN
`GOLD.DIM_PRODUCTS` AS P
ON S.PRODUCT_KEY = P.PRODUCT_KEY
GROUP BY P.PRODUCT_KEY,P.PRODUCT_NAME,P.CATEGORY,P.SUBCATEGORY
ORDER BY TOTAL_SALES DESC;
```

	PRODUCT_KEY	PRODUCT_NAME	CATEGORY	SUBCATEGORY	TOTAL_ORDERS	TOTAL_QUANTITY SOLD	TOTAL_SALES	Avg_Selling_Price
▶	122	Mountain-200 Black- 46	Bikes	Mountain Bikes	620	620	1373454	2215
	121	Mountain-200 Black- 42	Bikes	Mountain Bikes	614	614	1363128	2220
	123	Mountain-200 Silver- 38	Bikes	Mountain Bikes	596	596	1339394	2247
	125	Mountain-200 Silver- 46	Bikes	Mountain Bikes	580	580	1301029	2243
	120	Mountain-200 Black- 38	Bikes	Mountain Bikes	582	582	1294854	2225
	124	Mountain-200 Silver- 42	Bikes	Mountain Bikes	560	560	1257368	2245
	17	Road-150 Red- 48	Bikes	Road Bikes	337	337	1205786	3578
	20	Road-150 Red- 62	Bikes	Road Bikes	336	336	1202208	3578
	18	Road-150 Red- 52	Bikes	Road Bikes	302	302	1080556	3578
	19	Road-150 Red- 56	Bikes	Road Bikes	295	295	1055510	3578
	16	Road-150 Red- 44	Bikes	Road Bikes	281	281	1005418	3578

Note: The data shown below is for demonstration purposes only and includes sample entries to illustrate the query results. Additional data is present beyond what is displayed.