FINANCE, AND SUPPLY CHAIN ANALYTICS FOR ATLIQ HARDWARE

SQL PROJECT
BY AKHIL

PROJECT OVERVIEW

OPTIMIZING ATLIQ HARDWARE'S OPERATIONS THROUGH SQL

In this project, we aim to address the growing challenges **AtliQ Hardware** faces due to the size and complexity of their current data management in Excel. As a leading global manufacturer of hardware products, AtliQ Hardware requires efficient, data-driven insights to remain competitive in the market.

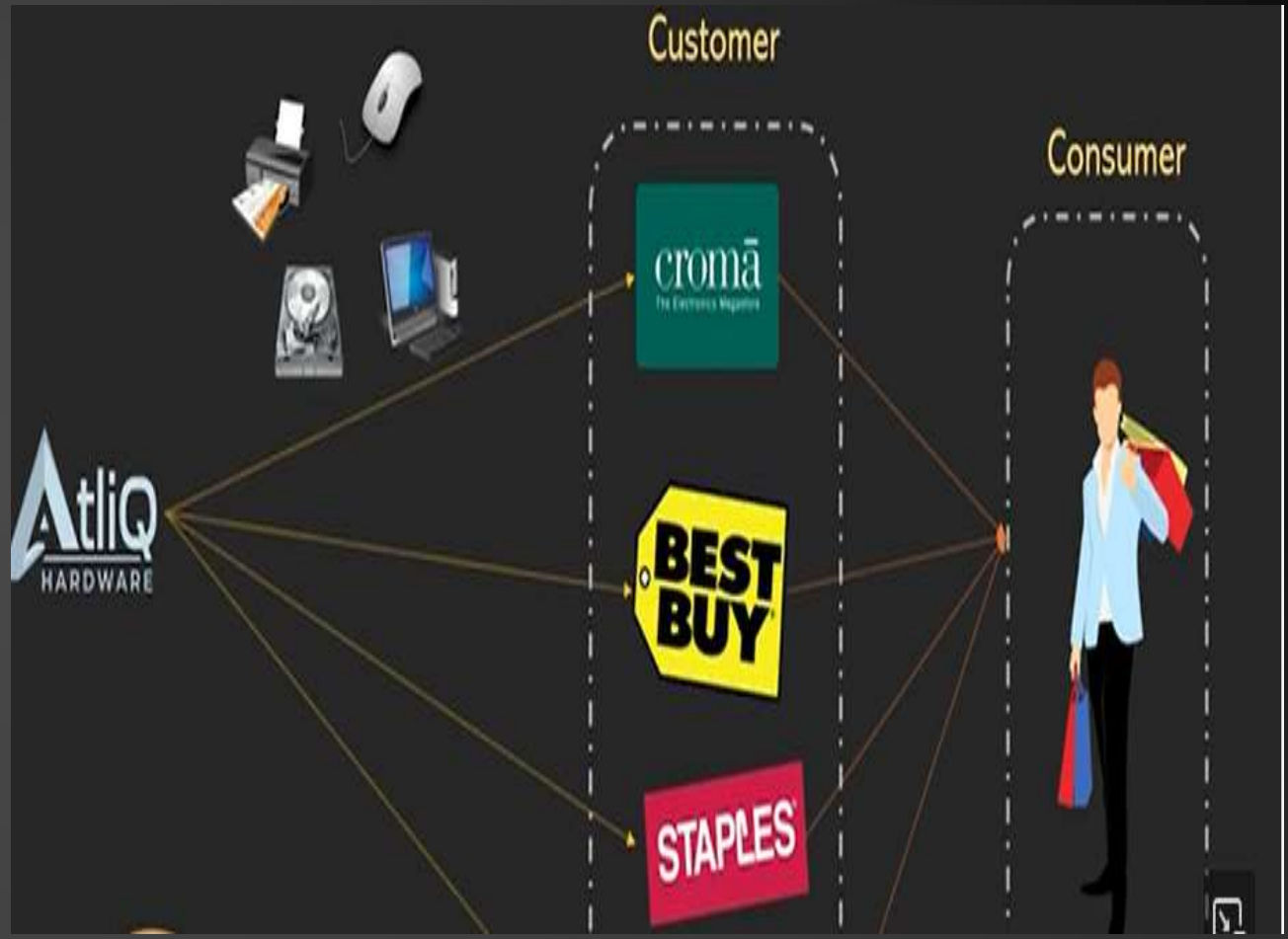
Our approach leverages **SQL to analyze key datasets**—spanning sales performance, customer behaviour, market trends, and supply chain operations—allowing us to transform raw data into actionable insights. By utilizing SQL for structured querying and analysis, we will streamline data processing, forecast trends, and provide recommendations to improve operational efficiency, increase sales, and optimize inventory management.

TABLE OF CONENTS

ATLIQ HARDWARE AND BUSINESS MODEL

AtliQ Hardware is a renowned global leader in electronics manufacturing, known for its expertise in producing and distributing a wide range of high-quality hardware products.

Our portfolio includes personal computers, printers, mice, and various other computer peripherals, designed to meet the diverse needs of customers around the world.



PROBLEM STATEMENT AND PROBLEM OVERVIEW

PROBLEM STATEMENT

AtliQ Hardware is experiencing performance inefficiencies driven by the growing size and complexity of its Excel-based data management system. To resolve these challenges, the company has established a specialized team of data analysts tasked with utilizing MySQL to extract meaningful insights and enhance overall operational efficiency. This transition aims to streamline data processes and enable more informed, data-driven decision-making.

PROBLEM OVERVIEW

The Scope of this project entails conduction an in-depth analysis of the dataset provided by AtliQ Hardware. The primary goal is to derive actionable insights regarding sales performance, market dynamics, customer behaviour, and to forecast supply chain trends

DATA SET

- **DIMENSION TABLES**
  - DIM_CUSTOMER
  - DIM_DATE
  - DIM_PRODUCT
- **FACT TABLES**
  - FACT_FORCAST_MONTHLY
  - FACT_FREIGHT_COST
  - FACT_GROSS_PRICE
  - FACT_MANUFACTURING_COST
  - FACT_POST_INVOICE_DEDUCTION
  - FACT_PRE_INVOICE_DEDUCTION
  - FACT_SALES_MONTHLY

PART 1: FINANCE ANALYTICS

CROMA INDIA PRODUCT WISE SALES REPORT FOR FISCAL YEAR 2021

- WE HAVE TO GENERATE A REPORT OF INDIVIDUAL PRODUCT SALES(AGGREGATED ON MONTHLY BASIS AT THE PRODUCT CODE LEVEL) FOR CROMA INDIA CUSTOMER FOR FY=2021 SO THAT WE CAN TRACK PRODUCT SALES AND RUN FURTHUR PRODUCT ANALYTICS

- REPORT SHOULD HAVE THE FOLLOWING FIELDS:

1. Month
2. Product Name
3. Variant
4. Sold Quantity
5. Gross Price Per Item
6. Gross Price Total

- **Module: User-Defined SQL Function**

a. first grab customer codes for Croma india

```
4   -- a. first grab customer codes for Croma india
5
6   SELECT * FROM dim_customer WHERE customer like "%croma%" AND market="india";
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

customer_code	customer	platform	channel	market	sub_zone	region
90002002	Croma	Brick & Mortar	Retailer	India	India	APAC
NULL	NULL	NULL	NULL	NULL	NULL	NULL

b. Get all the sales transaction data from fact_sales_monthly table for that customer(croma: 90002002) in the fiscal_year 2021

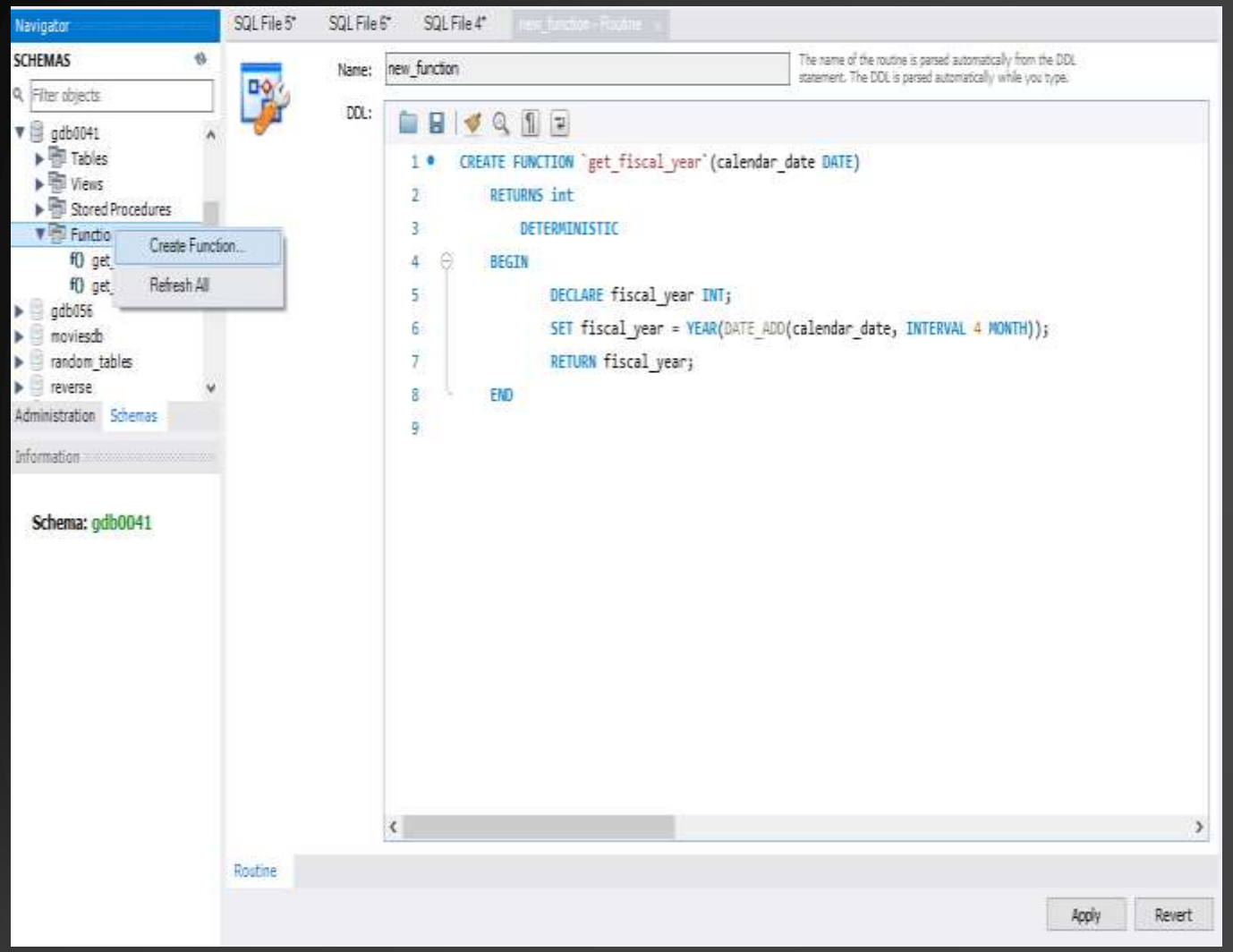
```
3
4     -- b. Get all the sales transaction data from fact_sales_monthly table for that customer(croma: 90002002) in the fiscal
5 •       SELECT * FROM fact_sales_monthly
6         WHERE
7                 customer_code=90002002 AND
8                 YEAR(DATE_ADD(date, INTERVAL 4 MONTH))=2021
9         ORDER BY date asc
0         LIMIT 100000;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

date	fiscal_year	product_code	customer_code	sold_quantity
2020-09-01	2021	A0118150101	90002002	202
2020-09-01	2021	A0118150102	90002002	162
2020-09-01	2021	A0118150103	90002002	193
2020-09-01	2021	A0118150104	90002002	146
2020-09-01	2021	A0219150201	90002002	149
2020-09-01	2021	A0219150202	90002002	107
2020-09-01	2021	A0220150203	90002002	123
2020-09-01	2021	A0320150301	90002002	146
2020-09-01	2021	A0321150302	90002002	236
2020-09-01	2021	A0321150303	90002002	137
2020-09-01	2021	A0418150103	90002002	23
2020-09-01	2021	A0418150104	90002002	82
2020-09-01	2021	A0418150105	90002002	86
2020-09-01	2021	A0418150106	90002002	48
2020-09-01	2021	A0519150201	90002002	138

Result Grid
Form Editor
Field Types
Query Stats

c. create a function 'get_fiscal_year' to get fiscal year by passing the date



d. Replacing the function created in the step:b

```
4 •  SELECT * FROM fact_sales_monthly
5        WHERE
6                customer_code=90002002 AND
7                get_fiscal_year(date)=2021
8        ORDER BY date asc
9        LIMIT 100000;
```

Result Grid | Filter Rows: | Edit:

date	fiscal_year	product_code	customer_code	sold_quantity
2020-09-01	2021	A0118150101	90002002	202
2020-09-01	2021	A0118150102	90002002	162
2020-09-01	2021	A0118150103	90002002	193
2020-09-01	2021	A0118150104	90002002	146
2020-09-01	2021	A0219150201	90002002	149
2020-09-01	2021	A0219150202	90002002	107
2020-09-01	2021	A0220150203	90002002	123
2020-09-01	2021	A0320150301	90002002	146
2020-09-01	2021	A0321150302	90002002	236
2020-09-01	2021	A0321150303	90002002	137
2020-09-01	2021	A0418150103	90002002	23
2020-09-01	2021	A0418150104	90002002	82
2020-09-01	2021	A0418150105	90002002	86
2020-09-01	2021	A0418150106	90002002	48
2020-09-01	2021	A0519150201	90002002	138

Module: Gross Sales Report: Monthly Product Transactions

- Till now we have find 1.Month and 4.sold quantity column now we have to find 2.Product Name and 3.Variant to find these column we need to go for dim_product table because these column present in product table

```
1 •  SELECT * FROM gdb0041.dim_product;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

product_code	division	segment	category	product	variant
A0118150101	P & A	Peripherals	Internal HDD	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard
A0118150102	P & A	Peripherals	Internal HDD	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Plus
A0118150103	P & A	Peripherals	Internal HDD	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Premium
A0118150104	P & A	Peripherals	Internal HDD	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Premium Plus
A0219150201	P & A	Peripherals	Internal HDD	AQ WereWolf NAS Internal Hard Drive HDD – 8....	Standard
A0219150202	P & A	Peripherals	Internal HDD	AQ WereWolf NAS Internal Hard Drive HDD – 8....	Plus
A0220150203	P & A	Peripherals	Internal HDD	AQ WereWolf NAS Internal Hard Drive HDD – 8....	Premium
A0320150301	P & A	Peripherals	Internal HDD	AQ Zion Saga	Standard
A0321150302	P & A	Peripherals	Internal HDD	AQ Zion Saga	Plus
A0321150303	P & A	Peripherals	Internal HDD	AQ Zion Saga	Premium
A0418150101	P & A	Peripherals	Graphic Card	AQ Mforce Gen X	Standard 1
A0418150102	P & A	Peripherals	Graphic Card	AQ Mforce Gen X	Standard 2
A0418150103	P & A	Peripherals	Graphic Card	AQ Mforce Gen X	Standard 3
A0418150104	P & A	Peripherals	Graphic Card	AQ Mforce Gen X	Plus 1
A0418150105	P & A	Peripherals	Graphic Card	AQ Mforce Gen X	Plus 2

a. Perform joins to pull product information

```
4 •   SELECT s.date, s.product_code, p.product, p.variant, s.sold_quantity
5     FROM fact_sales_monthly s
6     JOIN dim_product p
7         ON s.product_code=p.product_code
8     WHERE
9             customer_code=90002002 AND
10            get_fiscal_year(date)=2021
11    LIMIT 1000000;
```

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

date	product_code	product	variant	sold_quantity
2020-09-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	202
2020-09-01	A0118150102	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Plus	162
2020-09-01	A0118150103	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Premium	193
2020-09-01	A0118150104	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Premium Plus	146
2020-09-01	A0219150201	AQ WereWolf NAS Internal Hard Drive HDD – 8....	Standard	149
2020-09-01	A0219150202	AQ WereWolf NAS Internal Hard Drive HDD – 8....	Plus	107
2020-09-01	A0220150203	AQ WereWolf NAS Internal Hard Drive HDD – 8....	Premium	123
2020-09-01	A0320150301	AQ Zion Saga	Standard	146
2020-09-01	A0321150302	AQ Zion Saga	Plus	236
2020-09-01	A0321150303	AQ Zion Saga	Premium	137
2020-09-01	A0418150103	AQ Mforce Gen X	Standard 3	23
2020-09-01	A0418150104	AQ Mforce Gen X	Plus 1	82
2020-09-01	A0418150105	AQ Mforce Gen X	Plus 2	86
2020-09-01	A0418150106	AQ Mforce Gen X	Plus 3	48
2020-09-01	A0519150201	AQ Mforce Gen Y	Standard 1	138

Till now we have four column :

1. Month
2. Product Name
3. Variant
4. Sold quantity

We require two more column:

5. Gross price per item
6. Gross price total

```
1 •   SELECT * FROM gdb0041.fact_gross_price;
```

Result Grid | Filter Rows: | Edit:

product_code	fiscal_year	gross_price
A0118150101	2018	15.3952
A0118150101	2019	14.4392
A0118150101	2020	16.2323
A0118150101	2021	19.0573
A0118150102	2018	19.5875
A0118150102	2019	18.5595
A0118150102	2020	19.8577
A0118150102	2021	21.4565
A0118150103	2018	19.3630
A0118150103	2019	19.3442
A0118150103	2020	22.1317
A0118150103	2021	21.7795
A0118150103	2022	23.9920
A0118150104	2018	19.5743
A0118150104	2019	18.5072

b. Performing join with 'fact_gross_price' table with the above query and generating required fields

Query:

```
SELECT
        s.date,
        s.product_code,
        p.product,
        p.variant,
        s.sold_quantity,
        g.gross_price,
        ROUND(s.sold_quantity*g.gross_price,2) as gross_price_total
FROM fact_sales_monthly s
JOIN dim_product p
        ON s.product_code=p.product_code
JOIN fact_gross_price g
        ON g.fiscal_year=get_fiscal_year(s.date)
    AND g.product_code=s.product_code
WHERE
        customer_code=90002002 AND
        get_fiscal_year(s.date)=2021
```

Output:

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

date	product_code	product	variant	sold_quantity	gross_price	gross_price_total
2020-09-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	202	19.0573	3849.57
2020-09-01	A0118150102	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Plus	162	21.4565	3475.95
2020-09-01	A0118150103	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Premium	193	21.7795	4203.44
2020-09-01	A0118150104	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Premium Plus	146	22.9729	3354.04
2020-09-01	A0219150201	AQ WereWolf NAS Internal Hard Drive HDD – 8....	Standard	149	23.6987	3531.11
2020-09-01	A0219150202	AQ WereWolf NAS Internal Hard Drive HDD – 8....	Plus	107	24.7312	2646.24
2020-09-01	A0220150203	AQ WereWolf NAS Internal Hard Drive HDD – 8....	Premium	123	23.6154	2904.69
2020-09-01	A0320150301	AQ Zion Saga	Standard	146	23.7223	3463.46
2020-09-01	A0321150302	AQ Zion Saga	Plus	236	27.1027	6396.24
2020-09-01	A0321150303	AQ Zion Saga	Premium	137	28.0059	3836.81
2020-09-01	A0418150103	AQ Mforce Gen X	Standard 3	23	19.5235	449.04
2020-09-01	A0418150104	AQ Mforce Gen X	Plus 1	82	19.9239	1633.76
2020-09-01	A0418150105	AQ Mforce Gen X	Plus 2	86	20.0766	1726.59

Then we will export this report by click on export button and will save it as csv file which we can open in Excel and can give it to business manager and product owner.

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

date	product_code	product	variant	sold_quantity	gross_price	gross_price_total
2020-09-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	202	19.0573	3849.57
2020-09-01	A0118150102	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Plus	162	21.4565	3475.95
2020-09-01	A0118150103	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Premium	193	21.7795	4203.44
2020-09-01	A0118150104	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Premium Plus	146	22.9729	3354.04

After export we will convert above report into .csv file and we will open in excel which will look like below

	A	B	C	D	E	F	G
1	date	product_code	product	variant	sold_quantity	gross_price	gross_price_total
2	6/1/2021	A0118150101	AQ Dracula HDD â€“ 3.5 Inch SATA 6 Gb/s 5400 RPM 256 MB Cache	Standard	205	19.0573	3906.7465
3	6/1/2021	A0118150102	AQ Dracula HDD â€“ 3.5 Inch SATA 6 Gb/s 5400 RPM 256 MB Cache	Plus	78	21.4565	1673.607
4	6/1/2021	A0118150103	AQ Dracula HDD â€“ 3.5 Inch SATA 6 Gb/s 5400 RPM 256 MB Cache	Premium	48	21.7795	1045.416
5	6/1/2021	A0118150104	AQ Dracula HDD â€“ 3.5 Inch SATA 6 Gb/s 5400 RPM 256 MB Cache	Premium Plus	126	22.9729	2894.5854
6	6/1/2021	A0219150201	AQ WereWolf NAS Internal Hard Drive HDD â€“ 8.89 cm	Standard	40	23.6987	947.948
7	6/1/2021	A0219150202	AQ WereWolf NAS Internal Hard Drive HDD â€“ 8.89 cm	Plus	102	24.7312	2522.5824
8	6/1/2021	A0220150203	AQ WereWolf NAS Internal Hard Drive HDD â€“ 8.89 cm	Premium	31	23.6154	732.0774
9	6/1/2021	A0320150301	AQ Zion Saga	Standard	91	23.7223	2158.7293
10	6/1/2021	A0321150302	AQ Zion Saga	Plus	70	27.1027	1897.189
11	6/1/2021	A0321150303	AQ Zion Saga	Premium	145	28.0059	4060.8555
12	6/1/2021	A0418150103	AQ Mforce Gen X	Standard 3	108	19.5235	2108.538
13	6/1/2021	A0418150104	AQ Mforce Gen X	Plus 1	76	19.9239	1514.2164
14	6/1/2021	A0418150105	AQ Mforce Gen X	Plus 2	84	20.0766	1686.4344
15	6/1/2021	A0418150106	AQ Mforce Gen X	Plus 3	114	19.9365	2272.761
16	6/1/2021	A0519150201	AQ Mforce Gen Y	Standard 1	58	22.3984	1299.1072
17	6/1/2021	A0519150202	AQ Mforce Gen Y	Standard 2	107	24.9298	2667.4886
18	6/1/2021	A0519150203	AQ Mforce Gen Y	Standard 3	156	26.5871	4147.5876
19	6/1/2021	A0519150204	AQ Mforce Gen Y	Plus 1	78	26.1081	2036.4318
20	6/1/2021	A0519150205	AQ Mforce Gen Y	Plus 2	144	29.7008	4276.9152
21	6/1/2021	A0519150206	AQ Mforce Gen Y	Plus 3	161	31.2439	5030.2679
22	6/1/2021	A0519150207	AQ Mforce Gen Y	Premium 1	126	32.4427	4087.7802
23	6/1/2021	A0519150208	AQ Mforce Gen Y	Premium 2	94	30.5816	2874.6704

Module: Gross Sales Report: Total Sales Amount

Generate monthly gross sales report for Croma India for all the years

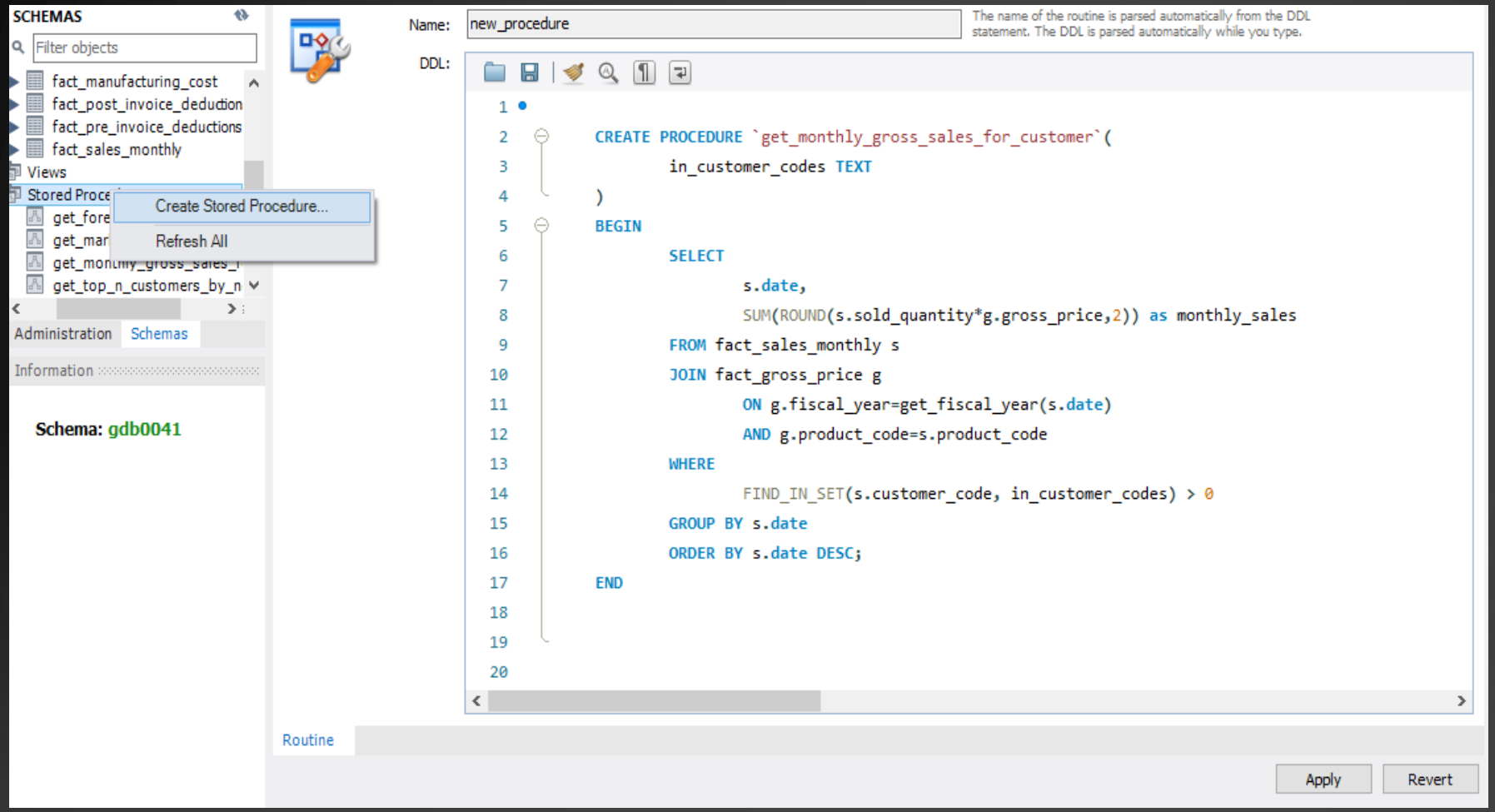
```
3  •  SELECT
4             s.date,
5             SUM(ROUND(s.sold_quantity*g.gross_price,2)) as monthly_sales
6      FROM fact_sales_monthly s
7      JOIN fact_gross_price g
8          ON g.fiscal_year=get_fiscal_year(s.date) AND g.product_code=s.product_code
9      WHERE
10              customer_code=90002002
11     GROUP BY date;
```

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

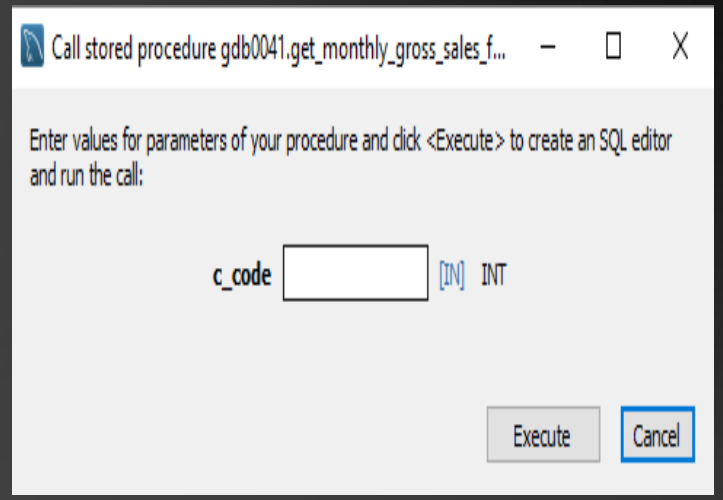
date	monthly_sales
2017-09-01	122407.57
2017-10-01	162687.56
2017-12-01	245673.84
2018-01-01	127574.73
2018-02-01	144799.54
2018-04-01	130643.92
2018-05-01	139165.06
2018-06-01	125735.36
2018-08-01	125409.90
2018-09-01	343337.14
2018-10-01	440562.10
2018-12-01	653944.72
2019-01-01	359025.06

Module: Stored Procedures: Monthly Gross Sales Report

Generate monthly gross sales report for any customer using stored procedure



After click on execution icon of Stored procedure monthly Gross sales we will find a new Screen where we can enter Customer code and can



Module: Stored Procedure: Market Badge

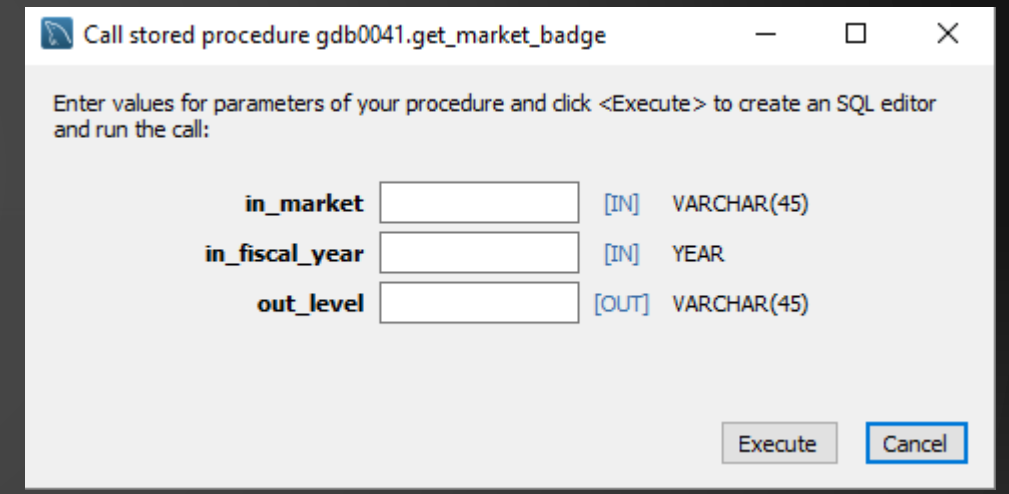
Write a stored proc that can retrieve market badge. i.e. if total sold quantity > 5 million that market is considered "Gold" else "Silver"

```
CREATE PROCEDURE `get_market_badge`(
            IN in_market VARCHAR(45),
            IN in_fiscal_year YEAR,
            OUT out_level VARCHAR(45)
    )
    BEGIN
             DECLARE qty INT DEFAULT 0;

             # Default market is India
             IF in_market = "" THEN
                  SET in_market="India";
             END IF;

             # Retrieve total sold quantity for a given market in a given year
             SELECT
                  SUM(s.sold_quantity) INTO qty
             FROM fact_sales_monthly s
             JOIN dim_customer c
             ON s.customer_code=c.customer_code
             WHERE
                  get_fiscal_year(s.date)=in_fiscal_year AND
                  c.market=in_market;

             # Determine Gold vs Silver status
             IF qty > 5000000 THEN
                  SET out_level = 'Gold';
             ELSE
                  SET out_level = 'Silver';
             END IF;
    END
```



BENEFITS OF STORED PROCEDURE

1. Convenience
2. Security
3. Maintainability
4. Performance
5. Developer Productivity

PART -2 : TOP CUSTOMERS, PRODUCTS, MARKETS

Module: Problem Statement and Pre-Invoice Discount Report

Till now we have found gross price and gross price total and we have to find top Customers, Products, Markets so we need to get pre invoice discount coloumn in last query where we get gross price and gross Discount

```
1  •    SELECT
2                   s.date,
3                   s.product_code,
4                   p.product,
5                   p.variant,
6                   s.sold_quantity,
7                   g.gross_price,
8                   ROUND(s.sold_quantity*g.gross_price,2) as gross_price_total
9           FROM fact_sales_monthly s
10          JOIN dim_product p
11                  ON s.product_code=p.product_code
12          JOIN fact_gross_price g
13                  ON g.fiscal_year=get_fiscal_year(s.date)
14              AND g.product_code=s.product_code
15          WHERE
16                  customer_code=90002002 AND
17                  get_fiscal_year(s.date)=2021
```

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

date	product_code	product	variant	sold_quantity	gross_price	gross_price_total
2020-09-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	202	19.0573	3849.57
2020-09-01	A0118150102	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Plus	162	21.4565	3475.95
2020-09-01	A0118150103	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Premium	193	21.7795	4203.44
2020-09-01	A0118150104	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Premium Plus	146	22.9729	3354.04
2020-09-01	A0219150201	AQ WereWolf NAS Internal Hard Drive HDD – 8....	Standard	149	23.6987	3531.11

INCLUDE PRE-INVOICE DEDUCTIONS IN CROMA DETAILED REPORT

```
 7              g.gross_price as gross_price_per_item,
 8              ROUND(s.sold_quantity*g.gross_price,2) as gross_price_total,
 9              pre.pre_invoice_discount_pct
10          FROM fact_sales_monthly s
11          JOIN dim_product p
12                  ON s.product_code=p.product_code
13          JOIN fact_gross_price g
14                  ON g.fiscal_year=get_fiscal_year(s.date)
15                  AND g.product_code=s.product_code
16          JOIN fact_pre_invoice_deductions as pre
17                  ON pre.customer_code = s.customer_code AND
18                  pre.fiscal_year=get_fiscal_year(s.date)
19          WHERE
20              s.customer_code=90002002 AND
21                  get_fiscal_year(s.date)=2021
22          LIMIT 1000000;
```

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

t_code	product	variant	sold_quantity	gross_price_per_item	gross_price_total	pre_invoice_discount_pct
150101	AQ Master wired x1 Ms	Standard 1	537	2.9434	1580.61	0.3025
150101	AQ Master wired x1 Ms	Standard 1	1036	2.9434	3049.36	0.3025
150101	AQ Master wired x1 Ms	Standard 1	921	2.9434	2710.87	0.3025
150101	AQ Master wired x1 Ms	Standard 1	306	2.9434	900.68	0.3025
150101	AQ Master wired x1 Ms	Standard 1	485	2.9434	1427.55	0.3025
150101	AQ Master wired x1 Ms	Standard 1	291	2.9434	856.53	0.3025

MODULE: DATABASE VIEWS: INTRODUCTION

Get the net_invoice_sales amount using the CTE's

```sql
WITH cte1 AS (
        SELECT
                s.date,
                s.customer_code,
                s.product_code,
                    p.product, p.variant,
                    s.sold_quantity,
                    g.gross_price as gross_price_per_item,
                    ROUND(s.sold_quantity*g.gross_price,2) as gross_price_total,
                    pre.pre_invoice_discount_pct
        FROM fact_sales_monthly s
        JOIN dim_product p
                ON s.product_code=p.product_code
         JOIN fact_gross_price g
                ON g.fiscal_year=s.fiscal_year
                AND g.product_code=s.product_code
         JOIN fact_pre_invoice_deductions as pre
                ON pre.customer_code = s.customer_code AND
                pre.fiscal_year=s.fiscal_year
         WHERE
                s.fiscal_year=2021)
    SELECT
            *,
            (gross_price_total-pre_invoice_discount_pct*gross_price_total) as net_invoice_sales
    FROM cte1
    LIMIT 1500000;
```

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

ct	variant	sold_quantity	gross_price_per_item	gross_price_total	pre_invoice_discount_pct	net_invoice_sales
cula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	248	19.0573	4726.21	0.0703	4393.957437
cula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	240	19.0573	4573.75	0.2061	3631.100125
cula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	31	19.0573	590.78	0.0974	533.238028
cula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	37	19.0573	705.12	0.2065	559.512720
cula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	7	19.0573	133.40	0.1068	119.152880
cula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	12	19.0573	228.69	0.2612	168.956172
cula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	17	19.0573	323.97	0.2471	243.917013

CREATING THE VIEW `SALES_PREINV_DISCOUNT` AND STORE ALL THE DATA IN LIKE A VIRTUAL TABLE

```sql
CREATE  VIEW `sales_preinv_discount` AS
    SELECT
            s.date,
            s.fiscal_year,
            s.customer_code,
            c.market,
            s.product_code,
            p.product,
            p.variant,
            s.sold_quantity,
            g.gross_price as gross_price_per_item,
            ROUND(s.sold_quantity*g.gross_price,2) as gross_price_total,
            pre.pre_invoice_discount_pct
    FROM fact_sales_monthly s
    JOIN dim_customer c
        ON s.customer_code = c.customer_code
    JOIN dim_product p
            ON s.product_code=p.product_code
    JOIN fact_gross_price g
```

Now generate net_invoice_sales using the above created view "sales_preinv_discount"

```sql
SELECT
        *,
        (gross_price_total-pre_invoice_discount_pct*gross_price_total) as net_invoice_sales
    FROM gdb0041.sales_preinv_discount
```

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

ct	variant	sold_quantity	gross_price_per_item	gross_price_total	pre_invoice_discount_pct	net_invoice_sales
cula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	51	15.3952	785.16	0.0824	720.462816
cula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	77	15.3952	1185.43	0.2956	835.016892
cula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	17	15.3952	261.72	0.0536	247.691808
cula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	6	15.3952	92.37	0.2378	70.404414
cula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	5	15.3952	76.98	0.1057	68.843214
cula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	7	15.3952	107.77	0.1875	87.563125
cula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	29	15.3952	446.46	0.0700	415.207800
cula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	34	15.3952	523.44	0.2551	389.910456
cula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	22	15.3952	338.69	0.0953	306.412843
cula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	5	15.3952	76.98	0.1896	62.384592
cula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	10	15.3952	153.95	0.0521	145.929205
cula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	4	15.3952	61.58	0.2046	48.980732
cula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	0	15.3952	0.00	0.0984	0.000000
cula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	0	15.3952	0.00	0.2620	0.000000
cula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	1	15.3952	15.40	0.0587	14.496020
cula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	1	15.3952	15.40	0.2501	11.548460
cula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	1	15.3952	15.40	0.1937	12.417020
cula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	20	15.3952	307.90	0.2025	245.550250

MODULE: DATABASE VIEWS: POST INVOICE DISCOUNT, NET SALES

Create a view for post invoice deductions: `sales_postinv_discount`

```sql
CREATE VIEW `sales_postinv_discount` AS
    SELECT
            s.date, s.fiscal_year,
            s.customer_code, s.market,
            s.product_code, s.product, s.variant,
            s.sold_quantity, s.gross_price_total,
            s.pre_invoice_discount_pct,
            (s.gross_price_total-s.pre_invoice_discount_pct*s.gross_price_total) as net_invoice_sales
            (po.discounts_pct+po.other_deductions_pct) as post_invoice_discount_pct
    FROM sales_preinv_discount s
    JOIN fact_post_invoice_deductions po
        ON po.customer_code = s.customer_code AND
        po.product_code = s.product_code AND
        po.date = s.date;
```

Create a report for net sales

```sql
SELECT
        *,
        net_invoice_sales*(1-post_invoice_discount_pct) as net_sales
    FROM gdb0041.sales_postinv_discount;
```

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

	variant	sold_quantity	gross_price_total	pre_invoice_discount_pct	net_invoice_sales	post_invoice_discount_pct	net_sales
▶ . 6 Gb/s 5400 R...	Standard	4	61.58	0.2803	44.319126	0.3905	27.0125072970
. 6 Gb/s 5400 R...	Standard	16	246.32	0.2803	177.276504	0.4139	103.9017589944
. 6 Gb/s 5400 R...	Standard	4	61.58	0.2803	44.319126	0.3295	29.7159739830
. 6 Gb/s 5400 R...	Standard	6	92.37	0.2803	66.478689	0.3244	44.9130022884
. 6 Gb/s 5400 R...	Standard	9	138.56	0.2803	99.721632	0.3766	62.1664653888
. 6 Gb/s 5400 R...	Standard	6	92.37	0.2803	66.478689	0.3615	42.4466429265
. 6 Gb/s 5400 R...	Standard	7	107.77	0.2803	77.562069	0.3173	52.9516245063
. 6 Gb/s 5400 R...	Standard	10	153.95	0.2803	110.797815	0.3501	72.0074999685
. 6 Gb/s 5400 R...	Standard	6	92.37	0.2803	66.478689	0.3740	41.6156593140
. 6 Gb/s 5400 R...	Standard	4	61.58	0.2117	48.543514	0.2863	34.6455059418
. 6 Gb/s 5400 R...	Standard	2	30.79	0.2117	24.271757	0.2851	17.3518790793

FINALLY CREATING THE VIEW `NET_SALES` WHICH INBUILTLY USE/INCLUDE ALL THE PREVIOUS CREATED VIEW AND GIVES THE FINAL RESULT

```
1 •  CREATE VIEW `net_sales` AS
2        SELECT
3               *,
4               net_invoice_sales*(1-post_invoice_discount_pct) as net_sales
5        FROM gdb0041.sales_postinv_discount;
6
```

MODULE: TOP MARKETS AND CUSTOMERS

Get top 5 market by net sales in fiscal year 2021

```
1  SELECT
2              market,
3              round(sum(net_sales)/1000000,2) as net_sales_mln
4      FROM gdb0041.net_sales
5      where fiscal_year=2021
6      group by market
7      order by net_sales_mln desc
8      limit 5
9
```

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

market	net_sales_mln
India	210.67
USA	132.05
South Korea	64.01
Canada	45.89
United Kingdom	44.73

STORED PROC TO GET TOP N MARKETS BY NET SALES FOR A GIVEN YEAR

```sql
CREATE PROCEDURE `get_top_n_markets_by_net_sales`(
                in_fiscal_year INT,
                in_top_n INT
        )
        BEGIN
                SELECT
                        market,
                        round(sum(net_sales)/1000000,2) as net_sales_mln
                FROM net_sales
                where fiscal_year=in_fiscal_year
                group by market
                order by net_sales_mln desc
                limit in_top_n;
        END
```

stored procedure that takes market, fiscal_year and top n as an input and returns top n customers by net sales in that given fiscal year and market

```sql
CREATE PROCEDURE `get_top_n_customers_by_net_sales`(
                in_market VARCHAR(45),
                in_fiscal_year INT,
                in_top_n INT
        )
        BEGIN
                select
                        customer,
                        round(sum(net_sales)/1000000,2) as net_sales_mln
                from net_sales s
                join dim_customer c
                    on s.customer_code=c.customer_code
                where
                s.fiscal_year=in_fiscal_year
                and s.market=in_market
                group by customer
                order by net_sales_mln desc
                limit in_top_n;
        END
```

GENERATE A REPORT GETTING TOP 5 CUSTOMERS BY NET SALES IN FISCAL YEAR 2021

```
SELECT
    c.customer,
    ROUND(SUM(Net_sales)/1000000,2) as net_sales_mln
FROM gdb0041.net_sales  n
JOIN dim_customer c
ON n.customer_code = c.customer_code
WHERE fiscal_year = 2021
GROUP BY customer
ORDER BY net_sales_mln desc
LIMIT 5;
```

Result Grid | Filter Rows:

customer	net_sales_mln
Amazon	109.03
Atliq Exclusive	79.92
Atliq e Store	70.31
Sage	27.07
Flipkart	25.25

GENERATE A REPORT GETTING TOP 5 PRODUCTS BY NET SALES IN FISCAL YEAR 2021

```
SELECT
    product, ROUND(SUM(net_sales) / 1000000, 2) AS net_sales_mln
FROM
    gdb0041.net_sales
WHERE
    fiscal_year = 2021
GROUP BY product
ORDER BY net_sales_mln DESC
LIMIT 5;
```

Result Grid | Filter Rows:

product	net_sales_mln
AQ BZ Allin1	33.75
AQ Qwerty	27.84
AQ Trigger	26.95
AQ Gen Y	23.58
AQ Maxima	22.32

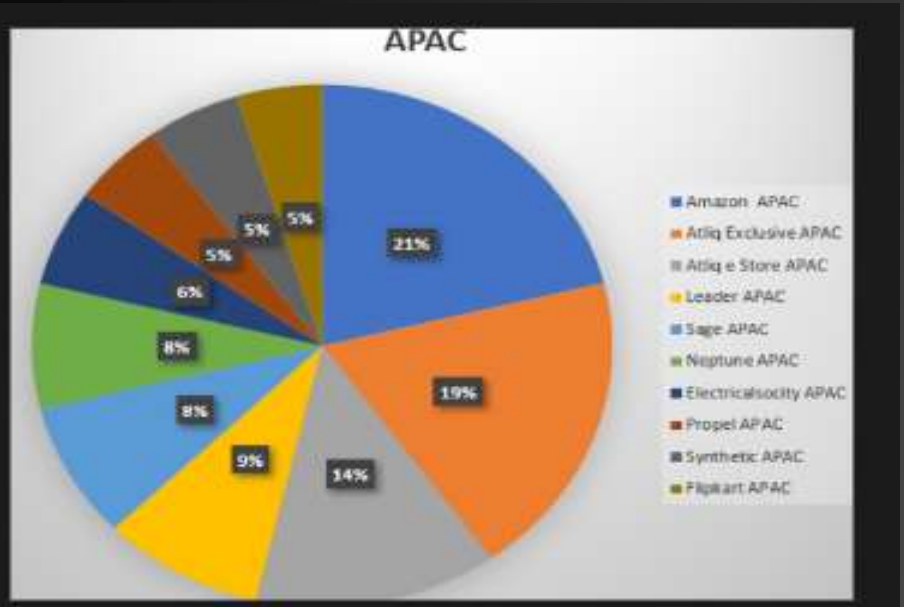
GENERATE A NET SALES % REPORT OF CUSTOMERS IN DIFFERENT REGIONS

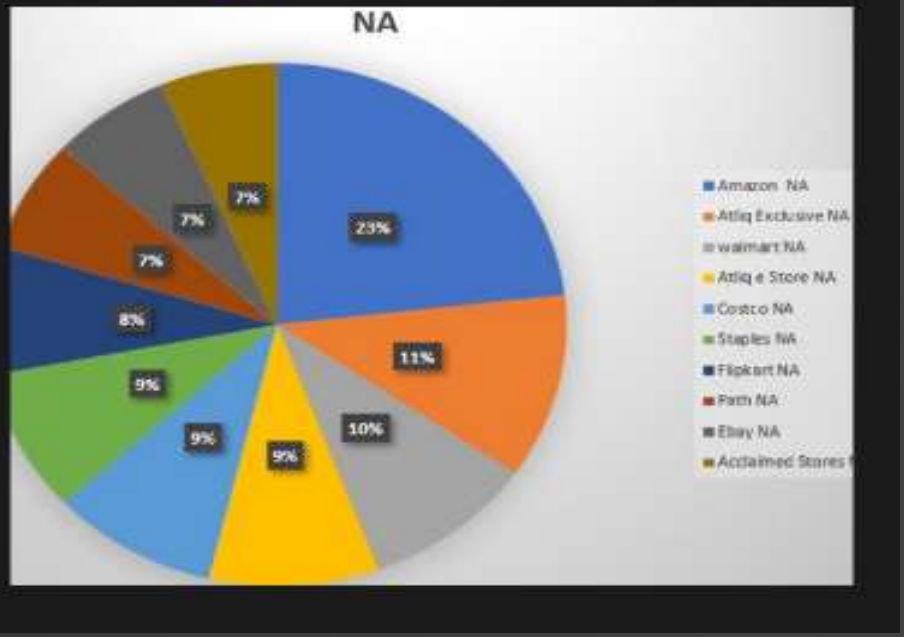
```
With cte1 as
(SELECT
    c.customer, c.region,
     ROUND(SUM(Net_sales)/1000000,2) as net_sales_mln
    FROM gdb0041.net_sales  n
    JOIN dim_customer c
  ON n.customer_code = c.customer_code
   WHERE fiscal_year = 2021
  GROUP BY c.customer, c.region)
 Select
     *,
     ROUND(net_sales_mln*100/sum(net_sales_mln)
     over (partition by region),2) as pct_share_region
     from cte1
 order by region, net_sales_mln desc;
```

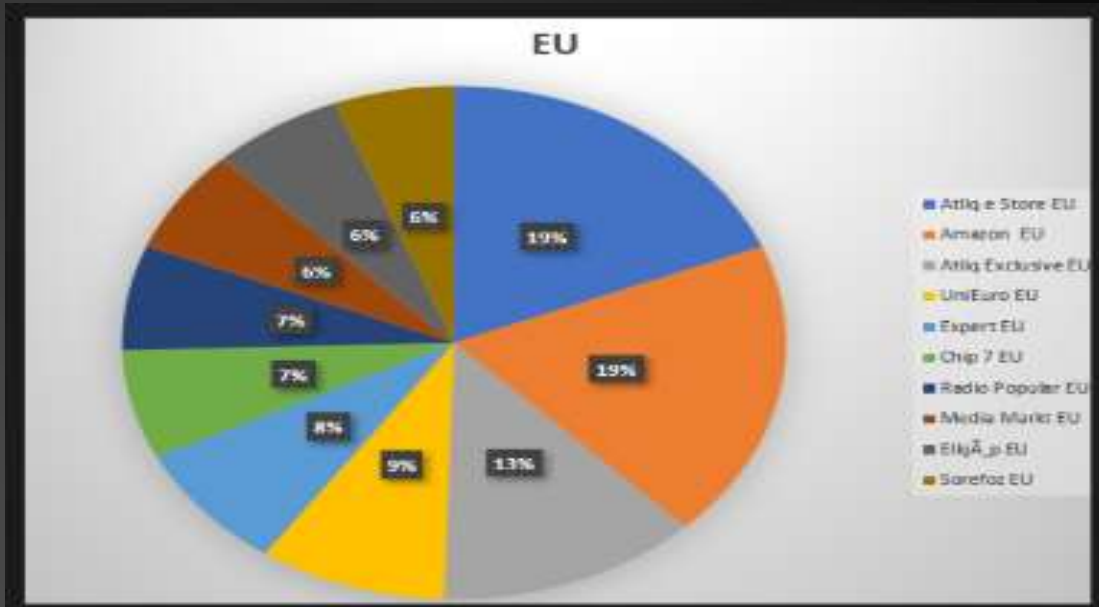
Result Grid | Filter Rows: | Export:

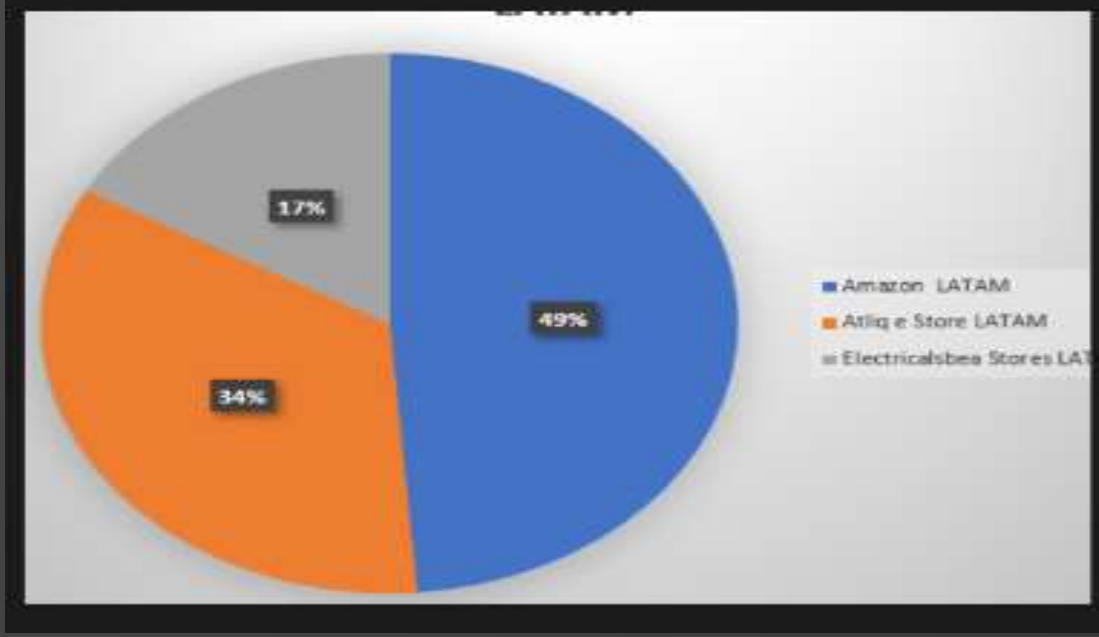
customer	region	net_sales_mln	pct_share_region
Amazon	APAC	57.41	12.99
Atliq Exclusive	APAC	51.58	11.67
Atliq e Store	APAC	36.97	8.36
Leader	APAC	24.52	5.55
Sage	APAC	22.85	5.17
Neptune	APAC	21.01	4.75
Electricalsocity	APAC	16.25	3.68
Propel	APAC	14.14	3.20
Synthetic	APAC	14.14	3.20
Flipkart	APAC	12.96	2.93
Novus	APAC	12.91	2.92
Expression	APAC	12.90	2.92

MARKET SHARE % - TOP CUSTOMERS IN DIFFERENT REGIONS









CONCLUSION

- ATLIQ HARDWARE ACHIEVED RECORD SALES IN 2022.
- INDIA WAS THE LARGEST MARKET IN 2021 WITH SALES OF $210.67M
- AMAZON GENERATED THE HIGHEST NET SALES IN 2021 WITH $109.03M
- THE AQ BZ ALL-IN-ONE WAS THE TOP-SELLING PRODUCT IN 2021 WITH THE SALES OF $33.75M
- AMAZON CAPTURES THE TOP MARKET SHARE% IN APAX, LATAM & NA REGIONS.
- ATLIQ ESTORE TOPPED THE CHART IN EU REGION.

THANK YOU