Finance Analytics – AtliQ Hardware

1.Croma India Product wise sales report for fiscal year. [Query | CSV]

As a product owner, **I want to** generate a report of individual product sales (aggregated on a monthly basis at the product code level) for Croma India customer for FY=2021 **so that** I can track individual product sales and run further product analytics on it in excel.

The 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

```
SELECT
        MONTH(f.date) AS "Month",
        p.product AS "Product Name",
        p.variant AS "Variant",
        SUM(f.sold_quantity) AS "Sold Quantity",
         SUM(gp.gross_price) AS "Gross Price Per Item",
         SUM(ROUND((f.sold_quantity * gp.gross_price),2)) AS "Gross Price Total"
FROM fact_sales_monthly f
JOIN dim_product p
        ON f.product_code = p.product_code
JOIN fact_gross_price gp
        ON f.product_code = gp.product_code AND f.fiscal_year = gp.fiscal_year
WHERE  customer_code = 90002002 AND f.fiscal_year = 2021
GROUP BY 1,2,3;
```

Month	Product Name	Variant	Sold Quantity	Gross Price Per Item	Gross Price Total
9	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 RPM 256 MB Cache	Standard	202	19.0573	3849.57
9	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 RPM 256 MB Cache	Plus	162	21.4565	3475.95
9	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 RPM 256 MB Cache	Premium	193	21.7795	4203.44
9	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 RPM 256 MB Cache	Premium Plus	146	22.9729	3354.04
9	AQ WereWolf NAS Internal Hard Drive HDD – 8.89 cm	Standard	149	23.6987	3531.11
9	AQ WereWolf NAS Internal Hard Drive HDD – 8.89 cm	Plus	107	24.7312	2646.24
9	AQ WereWolf NAS Internal Hard Drive HDD – 8.89 cm	Premium	123	23.6154	2904.69
9	AQ Zion Saga	Standard	146	23.7223	3463.46
9	AQ Zion Saga	Plus	236	27.1027	6396.24
9	AQ Zion Saga	Premium	137	28.0059	3836.81

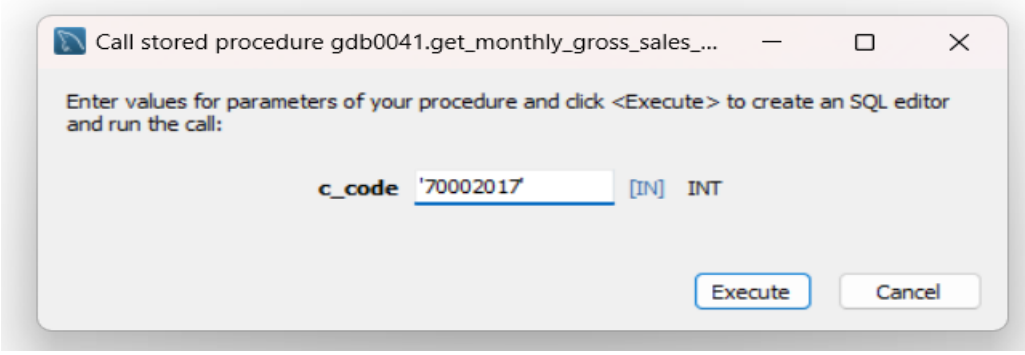
2. Gross Monthly total sales report for Croma. [Query | CSV]


```
SELECT
    s.date,
    ROUND(SUM(gp.gross_price * s.sold_quantity),2) AS  'total_gross_price'
FROM fact_sales_monthly s
JOIN fact_gross_price gp
        ON gp.product_code = s.product_code
                AND gp.fiscal_year = GET_FISCAL_YEAR(s.date)
WHERE customer_code = 90002002
GROUP BY s.date
ORDER BY s.date ASC;
```

date	total_gross_price
2017-09-01	122407.56
2017-10-01	162687.57
2017-12-01	245673.80
2018-01-01	127574.74
2018-02-01	144799.52
2018-04-01	130643.90
2018-05-01	139165.10
2018-06-01	125735.38
2018-08-01	125409.88
2018-09-01	343337.17
2018-10-01	440562.08

The above query is use in the various situation so that, we create stored procedure to the above condition using following code.

```
CREATE PROCEDURE `get_monthly_gross_sales_for_customer` (c_code INT)
BEGIN
SELECT
   s.date,
   ROUND(SUM(gp.gross_price * s.sold_quantity),2) AS  'total_gross_price'
          FROM fact_sales_monthly s
          JOIN fact_gross_price gp
                    ON gp.product_code = s.product_code
                              AND gp.fiscal_year = GET_FISCAL_YEAR(s.date)
          WHERE customer_code = c_code
          GROUP BY s.date
          ORDER BY s.date ASC;
END
```



70002017 – AtliQ Exclusive Store (We retrieve the total gross price for various customer by customer_id as input and click excute the below query had been run and shows as result below)

```
call gdb0041.get_monthly_gross_sales_for_customer('70002017');
```

date	total_gross_price
2017-09-01	145578.88
2017-10-01	170974.19
2017-11-01	213674.29
2018-01-01	135615.52
2018-02-01	144920.22
2018-03-01	131081.08
2018-05-01	132221.37
2018-06-01	138167.23
2018-07-01	120584.34
2018-09-01	348492.59
2018-10-01	494018.45

3.Generate a yearly report for Croma India. [Query | CSV]

Generate a yearly report for Croma India where there are two columns

1. Fiscal Year
2. Total Gross Sales amount In that year from Croma

```
SELECT
    get_fiscal_year(s.date) AS 'fiscal_year',
    ROUND(SUM(gp.gross_price * s.sold_quantity), 2) AS 'total_gross_price'
FROM
    fact_sales_monthly s
        JOIN
    fact_gross_price gp ON gp.product_code = s.product_code
        AND gp.fiscal_year = GET_FISCAL_YEAR(s.date)
WHERE
    customer_code = 90002002
GROUP BY 1
ORDER BY 1 ASC;
```

	fiscal_year	total_gross_price
▶	2018	1324097.44
	2019	3555079.02
	2020	6502181.91
	2021	23216512.22
	2022	44638198.92

4.Stored Procedure for market Badge. [Query | CSV]

Stored proc for market badge

Attach Add a child issue Link issue

Description

Create a stored proc that can determine the market badge based on the following logic,

If **total sold quantity** > **5 million** that market is considered **Gold** else it is **Silver**

My input will be,

- market
- fiscal year

Output

- market badge

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `get_market_badge`(
mrt VARCHAR(25),
fy INT)
BEGIN
	SELECT
		IF((SUM(s.sold_quantity) > 5000000),
			'GOLD','SILVER') AS badge
	FROM fact_sales_monthly s
	JOIN dim_customer c
		ON c.customer_code = s.customer_code
	WHERE GET_FISCAL_YEAR(s.date) = fy
			AND c.market = mrt
	GROUP BY c.market;
END
```

Write a Market and Fiscal year you want find and click execute the following stored procedure call and respective result shown below.

```
call gdb0041.get_market_badge('india', 2021);
```

badge
GOLD

5.Top 5 markets for a given financial year. [Query | CSV]

Description

As a product owner, I want a report for top markets, products, customers by net sales for a given financial year so that I can have a holistic view of our financial performance and can take appropriate actions to address any potential issues.

We will probably write stored proc for this as we will need this report going forward as well.

1. Report for **top markets**,

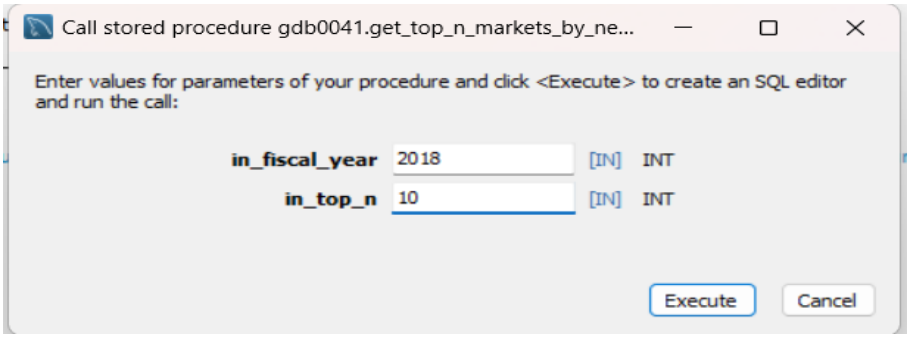
Rank	Market	Net Sales (in millions)
1	India	210.67
2	USA	132.05
3	South Korea	64.01

```
SELECT
        market,
        ROUND(SUM(net_sale)/1000000,2) AS 'total_net_sale_IN_MILLIONS'
FROM net_sale_table
WHERE fiscal_year=2021
GROUP BY market
ORDER BY total_net_sale_IN_MILLIONS DESC
LIMIT 5;
```

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

The above query is use in the various situation so that, we create stored procedure to the above condition using following code.

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




```
call gdb0041.get_top_n_markets_by_net_sales(2018, 10);
```

market	net_sales_mln
India	12.72
USA	5.94
South Korea	4.42
Australia	1.42
France	1.39
Philiphines	1.04
Indonesia	0.88
Canada	0.55
Italy	0.28
Brazil	0.20

6. Top products for a given financial year and market. [Query | CSV]

Description

As a product owner, I want a report for top markets, products, customers by net sales for a given financial year so that I can have a holistic view of our financial performance and can take appropriate actions to address any potential issues.

We will probably write stored proc for this as we will need this report going forward as well.

2. Report for **top products**,

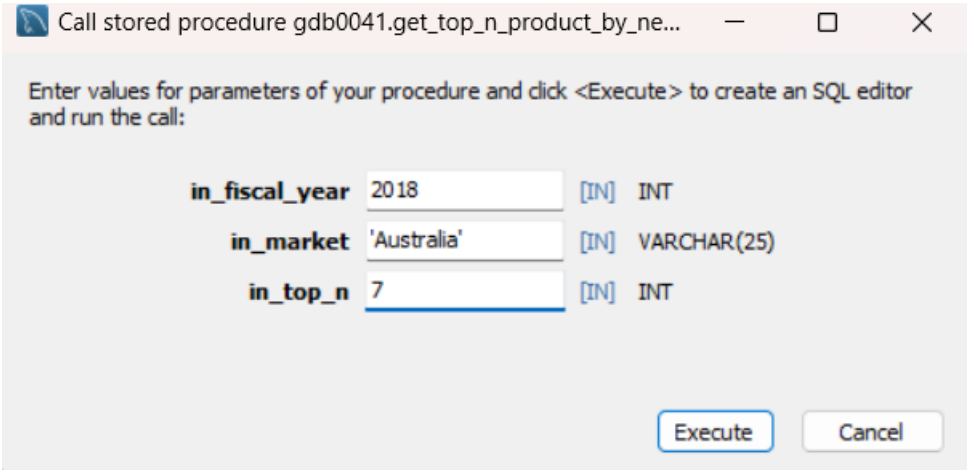
Rank	Product	Net Sales
1	AQ BZ Allin1	33.75
2	AQ Qwerty	27.84

```
SELECT product,
        ROUND(SUM(net_sale)/1000000,2) AS 'total_net_sale_IN_MILLIONS'
FROM net_sale_table
WHERE market='india'
        AND fiscal_year=2021
GROUP BY product
ORDER BY total_net_sale_IN_MILLIONS DESC
LIMIT 5;
```

product	total_net_sale_IN_MILLIONS
AQ BZ Allin1	8.54
AQ Qwerty	7.22
AQ Trigger	6.78
AQ Gen Y	6.02
AQ Trigger Ms	5.74

The above query is use in the various situation so that, we create stored procedure to the above condition using following code

```
CREATE PROCEDURE `get_top_n_product_by_netsale_`(
in_fiscal_year INT,
in_market VARCHAR(25),
in_top_n INT)
BEGIN
    SELECT product,
        ROUND(SUM(net_sale)/1000000,2) AS 'total_net_sale_IN_MILLIONS'
    FROM net_sale_table
    WHERE fiscal_year=in_fiscal_year
        AND market=in_market
    GROUP BY product
    ORDER BY total_net_sale_IN_MILLIONS DESC
    LIMIT in_top_n;
END
```




```
call gdb0041.get_top_n_product_by_netsale_(2018, 'Australia', 7);
```

product	total_net_sale_IN_MILLIONS
AQ Master wireless x1	0.17
AQ Wi Power Dx1	0.17
AQ 5000 Series Electron 8 5900X Desktop Proce...	0.13
AQ 5000 Series Electron 9 5900X Desktop Proce...	0.12
AQ Master wired x1	0.12
AQ BZ Compact	0.10
AQ BZ 101	0.10

7.Top customers for a given financial year. [Query | CSV]

Description

As a product owner, I want a report for top markets, products, customers by net sales for a given financial year so that I can have a holistic view of our financial performance and can take appropriate actions to address any potential issues.

We will probably write stored proc for this as we will need this report going forward as well.

3. Report for **top customers**

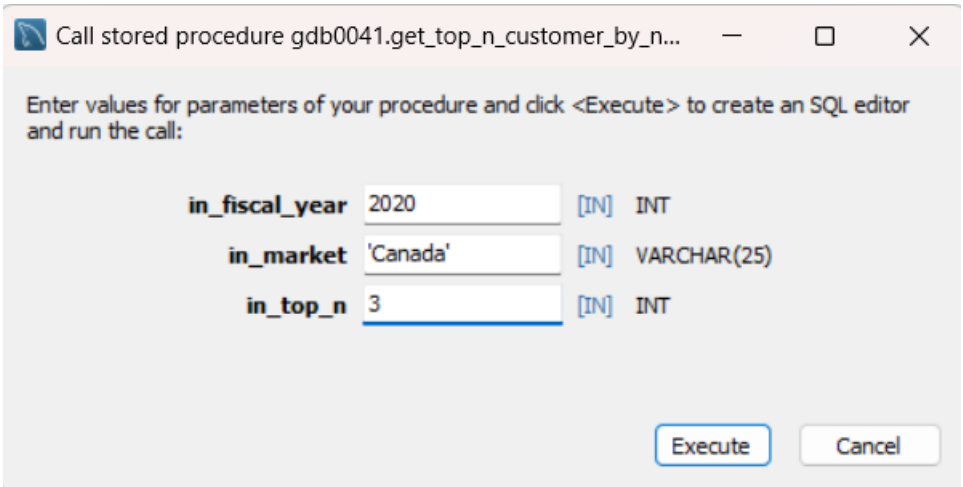
Rank	Customer	Net Sales
1	Amazon	109.03

```
SELECT customer,
        ROUND(SUM(net_sale)/1000000,2) AS 'total_net_sale_IN_MILLIONS'
FROM net_sale_table
WHERE market='india'
        AND fiscal_year=2021
GROUP BY customer
ORDER BY total_net_sale_IN_MILLIONS DESC
LIMIT 5;
```

customer	total_net_sale_IN_MILLIONS
Amazon	30.00
Atliq Exclusive	23.98
Flipkart	12.96
Electricalsocity	12.31
Propel	11.86

The above query is use in the various situation so that, we create stored procedure to the above condition using following code

```
CREATE PROCEDURE `get_top_n_customer_by_netsale_`(
in_fiscal_year INT,
in_market VARCHAR(25),
in_top_n INT)
BEGIN
  SELECT customer,
        ROUND(SUM(net_sale)/1000000,2) AS 'total_net_sale_IN_MILLIONS'
FROM net_sale_table
WHERE fiscal_year=in_fiscal_year
        AND market=in_market
GROUP BY customer
ORDER BY total_net_sale_IN_MILLIONS DESC
LIMIT in_top_n;
END
```




```
call gdb0041.get_top_n_customer_by_netsale_(2020, 'Canada', 3);
```

customer	total_net_sale_IN_MILLIONS
Amazon	4.66
Atliq e Store	1.74
Atliq Exclusive	1.45

8.Create View for Total Gross Sales. [Query | CSV]

It should have the following columns,

date, fiscal_year, customer_code, customer, market, product_code, product, variant,

sold_quanity, gross_price_per_item, gross_price_total

```
CREATE VIEW gross_sales_table AS
 SELECT
            s.date,
            s.fiscal_year,
            c.customer_code,
            c.customer,
            c.market,
            p.product_code,
            p.product,
            p.variant,
            s.sold_quantity,
            gp.gross_price AS gross_price_per_item,
            ROUND((s.sold_quantity * gp.gross_price),2) AS 'total_gross_price'
FROM fact_sales_monthly s
JOIN dim_customer c ON c.customer_code = s.customer_code
JOIN dim_product p ON p.product_code = s.product_code
JOIN fact_gross_price gp ON gp.product_code = s.product_code
AND gp.fiscal_year = s.fiscal_year;


SELECT * FROM gdb0041.gross_sales_table;
```

date	fiscal_year	customer_code	customer	market	product_code	product	variant	sold_quantity	gross_price_per_item	total_gross_price
2017-09-01	2018	70002017	Atliq Exclusive	India	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	51	15.3952	785.16
2017-09-01	2018	70002018	Atliq e Store	India	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	77	15.3952	1185.43
2017-09-01	2018	70003181	Atliq Exclusive	Indonesia	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	17	15.3952	261.72
2017-09-01	2018	70003182	Atliq e Store	Indonesia	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	6	15.3952	92.37
2017-09-01	2018	70006157	Atliq Exclusive	Philiphines	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	5	15.3952	76.98
2017-09-01	2018	70006158	Atliq e Store	Philiphines	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	7	15.3952	107.77
2017-09-01	2018	70007198	Atliq Exclusive	South Korea	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	29	15.3952	446.46
2017-09-01	2018	70007199	Atliq e Store	South Korea	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	34	15.3952	523.44

9.Create View for Pre-Invoice Deduction Amount. [Query | CSV]

```
CREATE VIEW pre_invoice_deduction_table AS
WITH cte AS (
SELECT
        s.date, s.fiscal_year,
        s.customer_code,
        c.market,
        s.product_code,p.product,p.variant,
        s.sold_quantity,
        ROUND(g.gross_price,2) as gross_price_per_item,
        ROUND(s.sold_quantity*g.gross_price,2) as gross_price_total,
        ROUND(pre.pre_invoice_discount_pct,2) as 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 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
ORDER BY s.date)

SELECT *, ROUND((total_gross_price * pre_invoice_deduction_pct),2) AS
                `pre_invoice_deduction_amt`
FROM cte;
```

date	fiscal_year	customer_code	customer	market	product_code	product	variant	sold_quantity	gross_price	total_gross_price	pre_invoice_deduction_pct	pre_invoice_deduction_amt
2017-09-01	2018	70002017	Atliq Exclusive	India	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	51	15.40	785.16	0.08	62.81
2017-09-01	2018	70002018	Atliq e Store	India	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	77	15.40	1185.43	0.30	355.63
2017-09-01	2018	70003181	Atliq Exclusive	Indonesia	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	17	15.40	261.72	0.05	13.09
2017-09-01	2018	70003182	Atliq e Store	Indonesia	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	6	15.40	92.37	0.24	22.17
2017-09-01	2018	70006157	Atliq Exclusive	Philiphines	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	5	15.40	76.98	0.11	8.47
2017-09-01	2018	70006158	Atliq e Store	Philiphines	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	7	15.40	107.77	0.19	20.48
2017-09-01	2018	70007198	Atliq Exclusive	South Korea	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	29	15.40	446.46	0.07	31.25
2017-09-01	2018	70007199	Atliq e Store	South Korea	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	34	15.40	523.44	0.26	136.09

10.Create View for Net Invoice Sales. [Query | CSV]

```
CREATE VIEW net_invoice_sale_table AS
SELECT *,
        ROUND((total_gross_price - pre_invoice_deduction_amt ),2) AS
                net_invoice_sale_amt
FROM pre_invoice_deduction_table;

SELECT * FROM gdb0041.net_invoice_sale_table;
```

date	fiscal_year	customer_code	customer	market	product_code	product	variant	sold_quantity	gross_price	total_gross_price	pre_invoice_deduction_pct	net_invoice_sale
2017-09-01	2018	70002017	Atliq Exclusive	India	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	51	15.40	785.16	0.08	722.35
2017-09-01	2018	70002018	Atliq e Store	India	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	77	15.40	1185.43	0.30	829.80
2017-09-01	2018	70003181	Atliq Exclusive	Indonesia	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	17	15.40	261.72	0.05	248.63
2017-09-01	2018	70003182	Atliq e Store	Indonesia	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	6	15.40	92.37	0.24	70.20
2017-09-01	2018	70006157	Atliq Exclusive	Philiphines	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	5	15.40	76.98	0.11	68.51
2017-09-01	2018	70006158	Atliq e Store	Philiphines	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	7	15.40	107.77	0.19	87.29
2017-09-01	2018	70007198	Atliq Exclusive	South Korea	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	29	15.40	446.46	0.07	415.21
2017-09-01	2018	70007199	Atliq e Store	South Korea	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	34	15.40	523.44	0.26	387.35
2017-09-01	2018	70008169	Atliq Exclusive	Australia	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	22	15.40	338.69	0.10	304.82
2017-09-01	2018	70008170	Atliq e Store	Australia	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	5	15.40	76.98	0.19	62.35

11.Create View for Post-Invoice Deduction Amount. [Query | CSV]

```
CREATE VIEW post_invoice_deduction_pct_table AS
SELECT
        nis.date, nis.fiscal_year,
        nis.customer_code,
        nis.market,
        nis.product_code, nis.product, nis.variant,
        nis.sold_quantity,
        nis.gross_price, nis.total_gross_price,
        nis.pre_invoice_deduction_pct, nis.pre_invoice_deduction_amt,
        nis.net_invoice_sale_amt,
        ROUND((discounts_pct + other_deductions_pct),2) AS
                post_invoice_deductions_pct
FROM net_invoice_sale_table nis
JOIN fact_post_invoice_deductions post
        ON nis.customer_code = post.customer_code
                AND nis.product_code = post.product_code;

CREATE VIEW post_invoice_deduction_table AS
SELECT *, ROUND((post_invoice_deductions_pct * net_invoice_sale_amt),2) AS
                                post_invoice_deduction_amt
FROM post_invoice_deduction_pct_table;
```

```
SELECT * FROM gdb0041.post_invoice_deduction_table;
```

date	fiscal_year	customer_code	market	product_code	product	variant	sold_quantity	gross_price	total_gross_price	pre_invoice_deduction_pct	pre_invoice_deduction_amt	net_invoice_sale_amt	post_invoice_deductions_pct	post_invoice_deduction_amt
2017-09-01	2018	70002017	India	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	51	15.40	785.16	0.08	62.81	722.35	0.34	245.60
2017-09-01	2018	70002017	India	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	51	15.40	785.16	0.08	62.81	722.35	0.41	296.16
2017-09-01	2018	70002017	India	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	51	15.40	785.16	0.08	62.81	722.35	0.41	296.16
2017-09-01	2018	70002017	India	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	51	15.40	785.16	0.08	62.81	722.35	0.37	267.27
2017-09-01	2018	70002017	India	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	51	15.40	785.16	0.08	62.81	722.35	0.40	288.94
2017-09-01	2018	70002017	India	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	51	15.40	785.16	0.08	62.81	722.35	0.36	260.05

12.Create View for Net Sales. [Query | CSV]

```
CREATE VIEW net_sales_table AS
SELECT *,
        net_invoice_sale_amt - post_invoice_deduction_amt AS  net_sale_table

FROM post_invoice_deduction_table;

SELECT * FROM gdb0041.net_invoice_sale_table;
```

date	fiscal_year	customer_code	market	product_code	product	variant	sold_quantity	gross_price	total_gross_price
2017-09-01	2018	70002017	India	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	51	15.40	785.16
2017-09-01	2018	70002017	India	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	51	15.40	785.16
2017-09-01	2018	70002017	India	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	51	15.40	785.16
2017-09-01	2018	70002017	India	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	51	15.40	785.16
2017-09-01	2018	70002017	India	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	51	15.40	785.16
2017-09-01	2018	70002017	India	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	51	15.40	785.16

total_gross_price	pre_invoice_deduction_pct	pre_invoice_deduction_amt	net_invoice_sale_amt	post_invoice_deductions_pct	post_invoice_deduction_amt	net_sale_table
785.16	0.08	62.81	722.35	0.34	245.60	476.75
785.16	0.08	62.81	722.35	0.41	296.16	426.19
785.16	0.08	62.81	722.35	0.41	296.16	426.19
785.16	0.08	62.81	722.35	0.37	267.27	455.08
785.16	0.08	62.81	722.35	0.40	288.94	433.41
785.16	0.08	62.81	722.35	0.36	260.05	462.30

13.Create CTE FY-2021 for TOP 10 customer BY %NET SALES. [Query | CSV]

```
WITH cte AS (
        SELECT
                customer,
                SUM(net_sale_amt)/1000000 AS tot_ns
        FROM net_sales_table
        WHERE fiscal_year =2021
        GROUP BY customer)

SELECT *,
        tot_ns*100/SUM(tot_ns) OVER() AS pct_ns FROM cte
ORDER BY tot_ns  DESC
LIMIT 10;
```

customer	tot_ns	pct_ns
Amazon	3041.914939	13.3495681691
Atliq Exclusive	2241.872387	9.8385486960
Atliq e Store	1923.023270	8.4392663005
Sage	755.035422	3.3135038415
Flipkart	726.631165	3.1888505976
Leader	659.801396	2.8955654220
Ebay	578.031955	2.5367168846
Neptune	575.611820	2.5260960231
Electricalsocity	482.411755	2.1170837246
Synthetic	430.089140	1.8874637879

14. Find customer wise net sales distribution per region for FY 2021. [Query | CSV]

```
with cte1 as (
                    select
                                c.customer,
                                c.region,
                                round(sum(net_sale_amt)/1000000,2) as net_sales_mln
          from net_sales_table ns
          join dim_customer c
              on ns.customer_code=c.customer_code
          where fiscal_year=2021
          group by c.customer, c.region)
select
          *,
          net_sales_mln*100/sum(net_sales_mln) over (partition by region)
                                                as pct_share_region
from cte1
order by region, pct_share_region desc
```

customer	region	net_sales_mln	pct_share_region
Amazon	APAC	1605.57	12.985237
Atliq Exclusive	APAC	1446.41	11.698012
Atliq e Store	APAC	1026.47	8.301697
Leader	APAC	659.80	5.336210
Sage	APAC	629.82	5.093744
Neptune	APAC	575.61	4.655314
Electricalsocity	APAC	482.41	3.901548
Propel	APAC	390.70	3.159832
Synthetic	APAC	383.08	3.098205

15.Find out top 3 products from each division by total quantity sold in a given year,

And create a stored procedure. [Query | CSV]

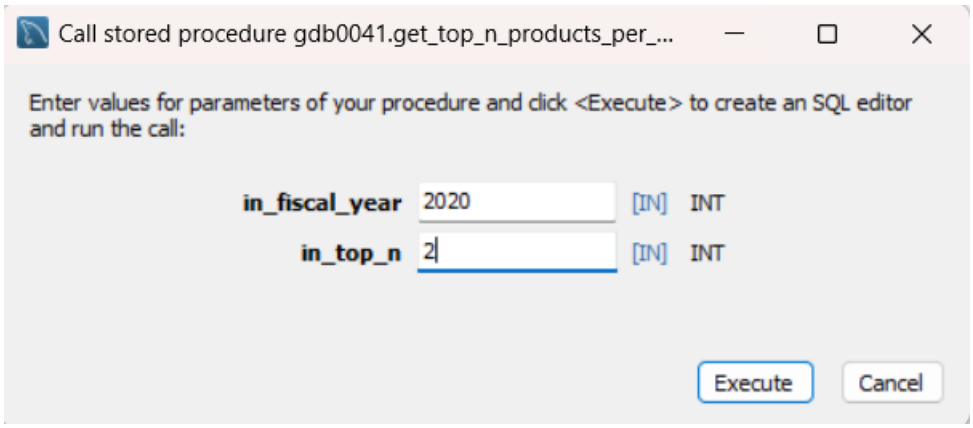
```
with cte1 as
(select
p.division,
p.product,
sum(s.sold_quantity) as total_qty
from fact_sales_monthly s
join dim_product p
on p.product_code=s.product_code
where fiscal_year=2021
group by p.product,p.division),
cte2 as
(select
*,
dense_rank() over (partition by division order by total_qty desc) as drnk
from cte1)

select * from cte2 where drnk<=3;
```

division	product	total_qty	drnk
N & S	AQ Pen Drive DRC	2034569	1
N & S	AQ Digit SSD	1240149	2
N & S	AQ Clx1	1238683	3
P & A	AQ Gamers Ms	2477098	1
P & A	AQ Maxima Ms	2461991	2
P & A	AQ Master wireless x1 Ms	2448784	3
PC	AQ Digit	135092	1
PC	AQ Gen Y	135031	2
PC	AQ Elite	134431	3

The above query is use in the various situation so that, we create stored procedure to the above condition using following code.

```
CREATE PROCEDURE `get_top_n_products_per_division_by_qty_sold`(
        in_fiscal_year INT,
                in_top_n INT)
BEGIN
        with cte1 as (
                select p.division, p.product,
            sum(sold_quantity) as total_qty
        from fact_sales_monthly s
        join dim_product p on p.product_code=s.product_code
        where fiscal_year=in_fiscal_year
        group by p.product,p.division),
      cte2 as (
                select *,
            dense_rank() over (partition by division order by total_qty desc) as drnk
        from cte1)
select * from cte2 where drnk <= in_top_n;
END
call gdb0041.get_top_n_products_per_division_by_qty_sold(2020, 2);
```



division	product	total_qty	drnk
N & S	AQ Clx1	935128	1
N & S	AQ Neuer SSD	924264	2
P & A	AQ Master wired x1 Ms	1578253	1
P & A	AQ Gamers Ms	1566445	2
PC	AQ Digit	68862	1
PC	AQ Elite	67841	2

16. Retrieve top 2 markets in every region by their gross sales amount in FY = 2021

[Query | CSV]

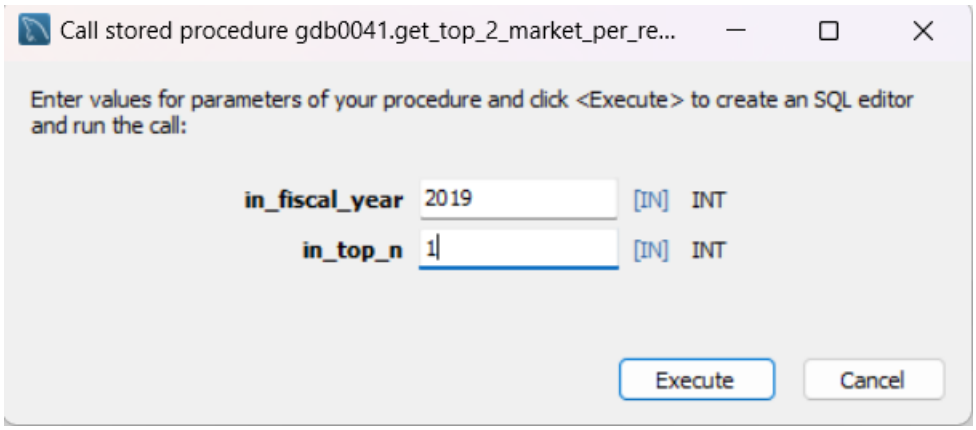
```
WITH cte AS (
        SELECT
                c.market,c.region,
                ROUND((SUM(gs.total_gross_price)) / 1000000,2) AS
                        "gross_sales_mln"
        FROM gross_sales_table gs
        JOIN dim_customer c
                ON  gs.customer_code = c.customer_code
                        AND gs.customer = c.customer
        WHERE gs.fiscal_year = 2021
        GROUP BY c.market,c.region ),
cte2 AS (
         SELECT * ,
                        dense_rank() over (partition by region order by
gross_sales_mln desc) as 't_rank'
         FROM cte)

SELECT * FROM cte2
WHERE t_rank<=2;
```

market	region	gross_sales_mln	t_rank
India	APAC	455.05	1
South Korea	APAC	131.86	2
United Kingdom	EU	78.11	1
France	EU	67.62	2
Mexico	LATAM	2.30	1
Brazil	LATAM	2.14	2
USA	NA	264.46	1
Canada	NA	89.78	2

The above query is use in the various situation so that, we create stored procedure to the above condition using following code.

```
CREATE PROCEDURE `get_top_2_market_per_region_by_gross_sales_mln`(
        in_fiscal_year INT,
                in_top_n INT)
BEGIN
 WITH cte AS (
        SELECT c.market,c.region,
                ROUND((SUM(gs.total_gross_price)) / 1000000,2) AS "gross_sales_mln"
        FROM gross_sales_table gs
        JOIN dim_customer c ON  gs.customer_code = c.customer_code
                        AND gs.customer = c.customer
        WHERE gs.fiscal_year = in_fiscal_year
        GROUP BY c.market,c.region ),
 cte2 AS (
  SELECT * ,dense_rank() over (partition by region order by gross_sales_mln desc) as
                                        't_rank'
  FROM cte)
SELECT  *  FROM cte2
WHERE t_rank<=in_top_n;
END
```




```
call gdb0041.get_top_2_market_per_region_by_gross_sales_mln(2019, 1);
```

market	region	gross_sales_mln	t_rank
India	APAC	69.52	1
France	EU	9.67	1
Brazil	LATAM	0.94	1
USA	NA	34.92	1

Supply Chain Analytics – AtliQ Hardware

17. Create a report for aggregated forecast accuract % over customer code for FY = 2021. [Query | CSV]

Report contains below columns

(customer code , customer name, markert , total sold quantity , forecast sold quantity,

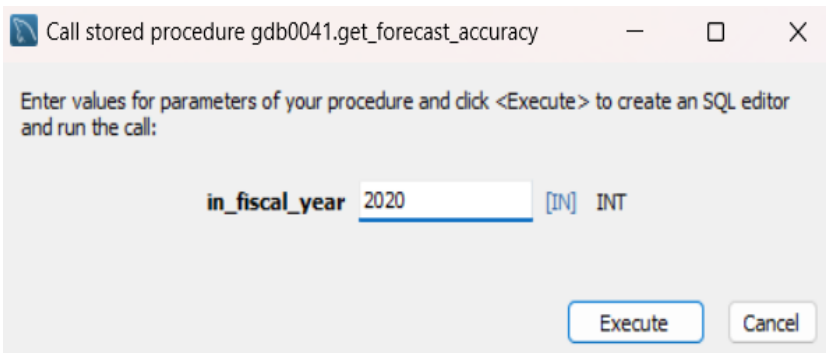
net error , net error , absolute error pct , absolute error pct , forecast accuracy pct)

```
WITH error_table AS (
        SELECT f.customer_code, c.customer AS "customer_name",c.market,
                SUM(sold_quantity) AS 'total_sold_qty',
                SUM(forecast_quantity) AS 'total_forecast_sold_qty',
                SUM((forecast_quantity - sold_quantity)) AS 'net_error',
                SUM((forecast_quantity - sold_quantity)) *100 /SUM(forecast_quantity)
                        AS 'net_error_pct',
                SUM(ABS(forecast_quantity - sold_quantity)) AS 'abs_error',
                SUM(ABS(forecast_quantity - sold_quantity)) *100 /SUM(forecast_quantity)
                        AS 'abs_error_pct'
        FROM fact_act_fore_monthly f
        JOIN dim_customer c
                USING(customer_code)
        WHERE fiscal_year = 2021
        GROUP BY customer_code)
SELECT  *,
    IF (et.abs_error_pct > 100, 0 , (100 - et.abs_error_pct)) AS forecast_accuracy_pct
FROM error_table et
ORDER BY forecast_accuracy_pct DESC;
```

customer_code	customer_name	market	total_sold_qty	total_forecast_sold_qty	net_error	net_error_pct	abs_error	abs_error_pct	forecast_accuracy_pct
90013120	Coolblue	Italy	109547	133532	23985	17.9620	70467	52.7716	47.2284
70010048	Atliq e Store	Bangladesh	119439	142010	22571	15.8940	75711	53.3139	46.6861
90023027	Costco	Canada	236189	279962	43773	15.6353	149303	53.3297	46.6703
90023026	Relief	Canada	228988	273492	44504	16.2725	146948	53.7303	46.2697
90017051	Forward Stores	Portugal	86823	118067	31244	26.4629	63568	53.8406	46.1594

The above query is use in the various situation so that, we create stored procedure to the above condition using following code.

```
CREATE PROCEDURE `get_forecast_accuracy`(
in_fiscal_year INT)
BEGIN
        WITH error_table AS (
                SELECT f.customer_code, c.customer AS "customer_name",c.market,
                SUM(sold_quantity) AS 'total_sold_qty',
                SUM(forecast_quantity) AS 'total_forecast_sold_qty',
                SUM((forecast_quantity - sold_quantity)) AS 'net_error',
                SUM((forecast_quantity - sold_quantity)) *100 /SUM(forecast_quantity)
                        AS 'net_error_pct',
                SUM(ABS(forecast_quantity - sold_quantity)) AS 'abs_error',
                SUM(ABS(forecast_quantity - sold_quantity)) *100 /SUM(forecast_quantity)
                        AS 'abs_error_pct'
FROM fact_act_fore_monthly f
JOIN dim_customer c USING(customer_code)
WHERE fiscal_year = in_fiscal_year
GROUP BY customer_code)
SELECT  *,
    IF (et.abs_error_pct > 100, 0 , (100 - et.abs_error_pct)) AS forecast_accuracy_pct
FROM error_table et
ORDER BY forecast_accuracy_pct DESC;
END
            call gdb0041.get_forecast_accuracy(2020);
```



18.The supply chain business manager wants to see which customers' forecast accuracy has dropped from 2020 to 2021. [Query | CSV]

Provide a complete report with these columns:

customer_code, customer_name, market, forecast_accuracy_2020, forecast_accuracy_2021

STEP 1:

```
-- FOR 2022
CREATE TEMPORARY TABLE forecast_accuracy_2020
  WITH error_table AS (
                        SELECT
                              f.customer_code,
                              c.customer AS "customer_name",
                              c.market,
                              SUM(sold_quantity) AS 'total_sold_qty',
                              SUM(forecast_quantity) AS 'total_forecast_sold_qty',
                              SUM((forecast_quantity - sold_quantity)) AS 'net_error',
                              SUM((forecast_quantity - sold_quantity)) *100 /SUM(forecast_quantity) AS 'net_error_pct',
                              SUM(ABS(forecast_quantity - sold_quantity)) AS 'abs_error',
                              SUM(ABS(forecast_quantity - sold_quantity)) *100 /SUM(forecast_quantity) AS 'abs_error_pct'
                        FROM fact_act_fore_monthly f
                        JOIN dim_customer c
                        USING(customer_code)
                        WHERE fiscal_year = 2020
                        GROUP BY customer_code
                  )
                  SELECT  *,
                              IF (et.abs_error_pct > 100, 0 , (100 - et.abs_error_pct)) AS forecast_accuracy_pct
                  FROM error_table et
                  ORDER BY forecast_accuracy_pct DESC;


SELECT * FROM forecast_accuracy_2020;
```

STEP 2:

```
-- FOR 2021


CREATE TEMPORARY TABLE forecast_accuracy_2021
   WITH error_table AS (
                        SELECT
                                f.customer_code,
                                c.customer AS "customer_name",
                                c.market,
                                SUM(sold_quantity) AS 'total_sold_qty',
                                SUM(forecast_quantity) AS 'total_forecast_sold_qty',
                                SUM((forecast_quantity - sold_quantity)) AS 'net_error',
                                SUM((forecast_quantity - sold_quantity)) *100 /SUM(forecast_quantity) AS 'net_error_pct',
                                SUM(ABS(forecast_quantity - sold_quantity)) AS 'abs_error',
                                SUM(ABS(forecast_quantity - sold_quantity)) *100 /SUM(forecast_quantity) AS 'abs_error_pct'
                        FROM fact_act_fore_monthly f
                        JOIN dim_customer c
                        USING(customer_code)
                        WHERE fiscal_year = 2021
                        GROUP BY customer_code
                    )
                SELECT  *,
                        IF (et.abs_error_pct > 100, 0 , (100 - et.abs_error_pct)) AS forecast_accuracy_pct
                FROM error_table et
                ORDER BY forecast_accuracy_pct DESC;


SELECT * FROM forecast_accuracy_2021;
```

STEP 3:

-- NOW JOIN THE ABOVE TWO TABLE WITH 2020 > 2021

```
SELECT
                f_20.customer_code,
                f_20.customer_name,
                f_20.market,
                f_20.forecast_accuracy_pct AS forecast_accuracy_2020,
                f_21.forecast_accuracy_pct AS forecast_accuracy_2021
FROM forecast_accuracy_2020 f_20
JOIN forecast_accuracy_2021 f_21
ON f_20.customer_code = f_21.customer_code
WHERE f_20.forecast_accuracy_pct > f_21.forecast_accuracy_pct
ORDER BY f_20.forecast_accuracy_pct DESC;
```

customer_code	customer_name	market	forecast_accuracy_2020	forecast_accuracy_2021
70006158	Atliq e Store	Philiphines	42.6505	24.4904
70008170	Atliq e Store	Australia	40.9573	38.7404
90005161	Zone	Pakistan	40.0813	37.0962
90014140	Radio Popular	Netherlands	38.5260	0.0000
90008166	Sound	Australia	38.5111	36.7911
70014143	Atliq e Store	Netherlands	38.3174	0.0000