AtliQ Hardware's Business Model

INTRODUCTION

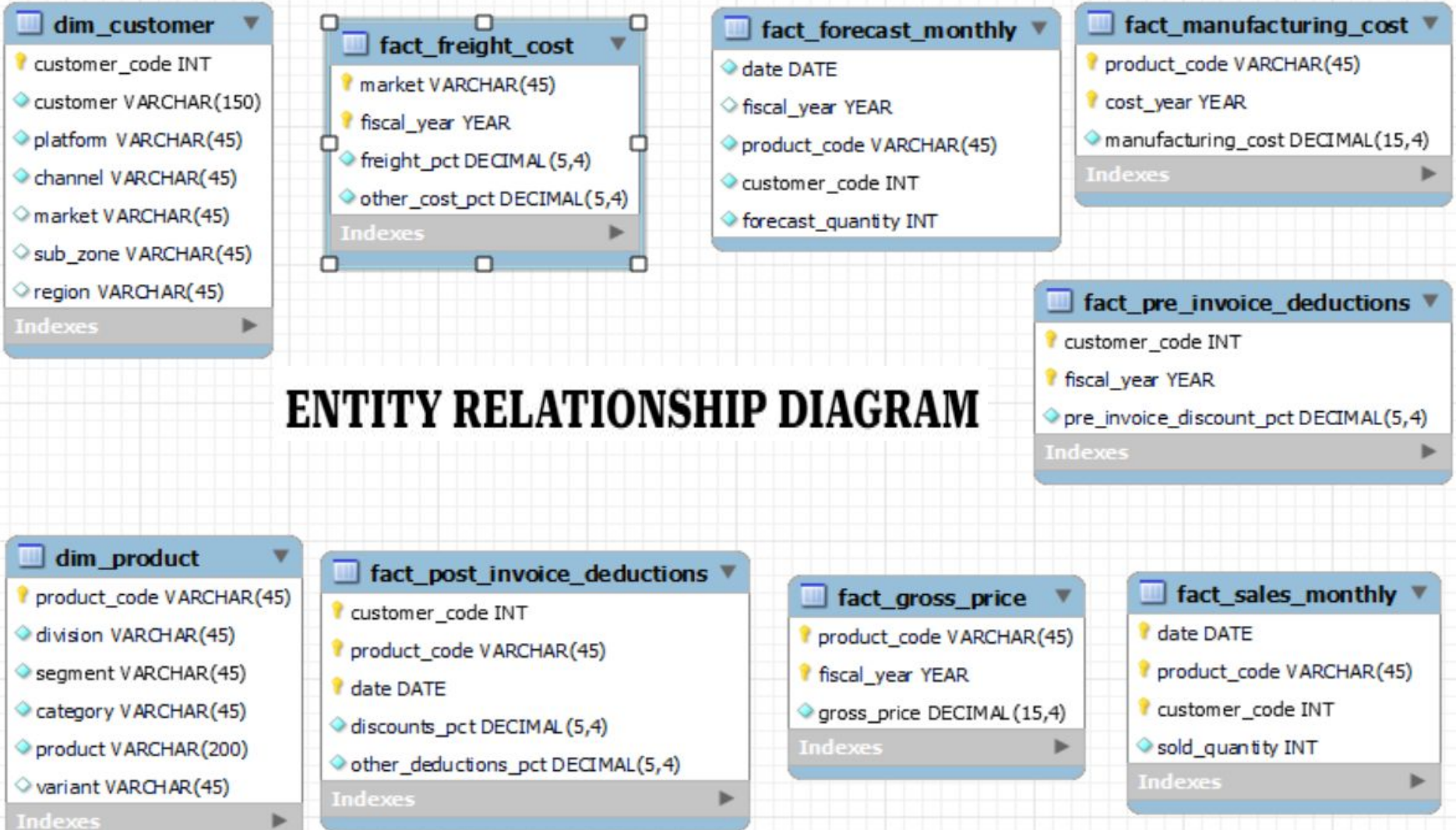
AtliQ is a company that sells Hardware like PC, Mouse, Printers etc to different customers like Chroma, Staples, BestBuy, Flipkart etc and then from these stores the hardwares are sold to the end consumer person
Customer is store
Consumer is the person who is consuming the product

Company has a manufacturing facility where they build all this hardware and send it ot the Warehouse distribution centres, they have business in different countries so they ship it to that location
And from there the hardware items goes to the individual customers

PROFIT AND LOSS TERMINOLOGIES

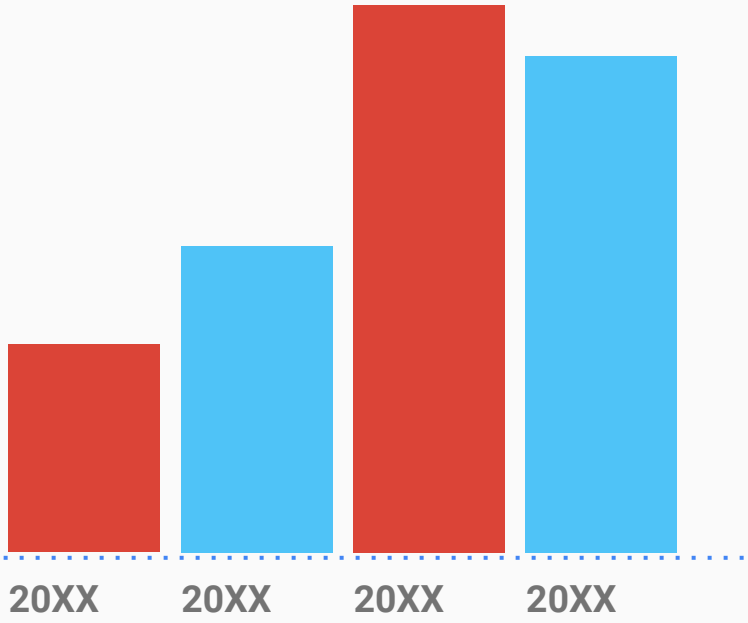
- Net Invoice Sales = Gross Price - Pre Invoice Deduction
- Post Invoice Deductions = Promotional Offers + Placement Fees + Performance Rebate
- Net Sales= Net Invoice Sales - Post Invoice Deductions
- Cost Of Goods Sold (COGS) = Manufacturing Cost + Freight(Transportation) + Other Cost
- Gross Margin on net sales(GM) = Net Sales - COGS
- %of GM= (GM/NS)*100
- Net Sales is basically the Revenue of AltiQ

ENTITY RELATIONSHIP DIAGRAM



PROBLEM STATEMENT

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-

- Month
- Product Name
- Variant
- Sold Quantity
- Gross Per Item
- Gross Per Total

Creating User Defined Function

```
CREATE  FUNCTION  get_fiscal_year (calendar_date DATE)

RETURNS int

DETERMINISTIC

BEGIN

DECLARE  fiscal_year INT;

SET fiscal_year = YEAR(DATE_ADD(calendar_date, INTERVAL 4 MONTH));

RETURN fiscal_year;

END
```

As The Fiscal_Year at AtliQ starts in September, so that acts as the 1st month of Fiscal_Year 2021 AND the total interval between Fiscal_Year and actual date year is 4 Months.

Hence, The ADDDATE() function is used that adds a time/date interval to a date and then returns the date.

CODE FUSION

```
SELECT s.date, s.product_code, p.product, p.variant,  s.sold_quantity, g.gross_price,
g.gross_price*s.sold_quantity AS gross_price_total
FROM fact_sales_monthly s

JOIN dim_product p
ON p.product_code=s.product_code
JOIN fact_gross_price g
ON g.product_code= s.product_code AND
      g.fiscal_year=get_fiscal_year(s.date)

WHERE customer_code=90002002
AND get_fiscal_year(date)=2021
ORDER BY date ASC
```

OUTPUT

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

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	3849.5746
2020-09-01	A3819150204	AQ LION x2	Plus 3	28	21.1942	593.4376
2020-09-01	A3220150403	AQ Lite	Plus 1	781	18.5844	14514.4164
2020-09-01	A3119150301	AQ Gamers	Standard 1	560	13.5061	7563.4160
2020-09-01	A1320150402	AQ Electron 5 3600 Desktop Processor	Plus	112	150.2450	16827.4400

Create a Report for the year 2021 and Q4 for Croma

```
SELECT * FROM fact_sales_monthly

WHERE customer_code=90002002

        AND get_fiscal_year(date)=2021
        AND get_fiscal_quarter(date)="Q4"          { User-Defined Function }

ORDER BY date ASC
```

Creating User Defined Function

```
CREATE FUNCTION get_fiscal_quarter(calendar_date date) RETURNS char(2)
DETERMINISTIC
BEGIN
DECLARE m tinyint;
DECLARE qtr CHAR(2);
SET m=MONTH(calendar_date);
CASE
        WHEN m IN (9,10,11) THEN SET qtr="Q1";
        WHEN m IN (12,1,2) THEN SET qtr="Q2";
        WHEN m IN (3,4,5) THEN SET qtr="Q3";
        ELSE SET qtr="Q4";
END CASE;
RETURN qtr;
END
```

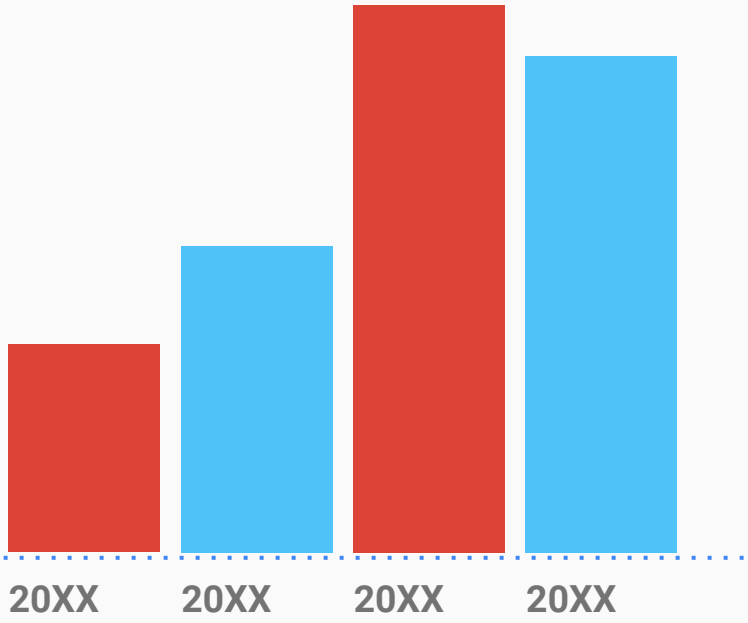
OUTPUT

Result Grid | Filter Rows: | Edit: | Export/Imp

date	product_code	customer_code	sold_quantity	fiscal_year
2021-06-01	A0118150101	90002002	205	2021
2021-06-01	A2620150603	90002002	996	2021
2021-06-01	A0118150102	90002002	78	2021
2021-06-01	A3920150303	90002002	69	2021
2021-06-01	A0118150103	90002002	48	2021
2021-06-01	A3220150403	90002002	95	2021

PROBLEM STATEMENT

As a Product Owner, I need an aggregate monthly Gross sales report for Croma India customer so that I can track how much sales this particular customer is generating for AltiQ and manage our relationships accordingly



The Report should have the following fields-

- Month
- Total Gross sales amount to Croma India in this Month

CODE FUSION

```
SELECT s.date ,
SUM( ROUND(sold_quantity*gross_price,2)) AS
gross_price_total
FROM fact_sales_monthly s

JOIN fact_ gross_price g
ON s.product_code=g.product_code
AND g.fiscal_year=get_fiscal_year(s.date)

WHERE customer_code=90   002002

GROUP BY s.date
ORDER BY s.date DESC
```

OUTPUT

OUTPUT

Result Grid | Filter Rows:

date	gross_price_total
2021-12-01	19537146.58
2021-10-01	13908229.35
2021-09-01	11192823.18
2021-08-01	2349478.81
2021-06-01	2288587.49
2021-05-01	2181587.87

Generate a Yearly Report for Croma India-

- **Fiscal Year**
- **Total Gross sales amount in that year from Croma India**

CODE FUSION

```
SELECT get_fiscal_year(date)  AS Fiscal_Year
SUM( ROUND(sold_quantity*gross_price,2)) AS
gross_sales_total
FROM fact_sales_monthly s

JOIN fact_ gross_price g
ON s.product_code=g.product_code
AND g.fiscal_year=get_fiscal_year(s.date)

WHERE customer_code=90   002002

GROUP BY get_fiscal_year(date)
ORDER BY Fiscal_Year DESC
```

OUTPUT

OUTPUT

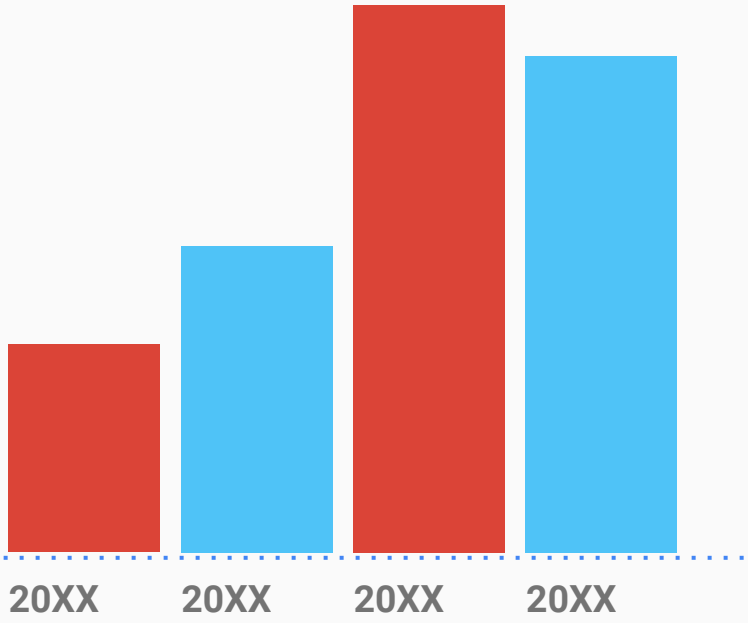
Result Grid | Filter Rows:

fiscal_year	yearly_sales
2018	6400351.36
2019	16499328.31
2020	24657811.19
2021	71287812.63
2022	121145442.85

PROBLEM STATEMENT

Create a Stored Procedure that can create a market badge based on the following logic,
If total sold quantity>5 million that market is considered Gold else it is Silver.

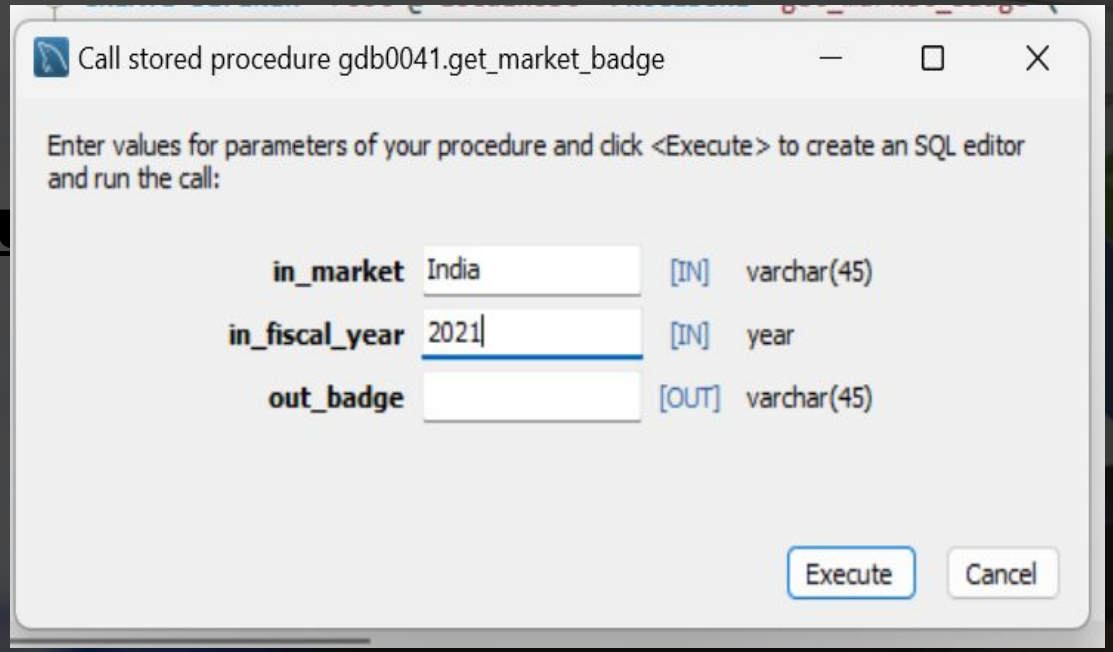
INPUT- Market,Fiscal_year,
OUTPUT- Market Badge



CODE FUSION

```
CREATE get_market_badge(
IN in_market varchar(45),
IN in_fiscal_year year,
OUT out_badge varchar(45)
)
BEGIN
DECLARE qty int default 0;
SELECT SUM(sold_quantity) into qty
FROM dim_customer c
JOIN fact_sales_monthly s
ON s.customer_code=c.customer_code
WHERE get_fiscal_year(s.date)=in_fiscal_year
 AND market=in_market
GROUP BY market;
if qty>5000000 then
set out_badge="Gold";
else
set out_badge="Silver";
end if;
END
```

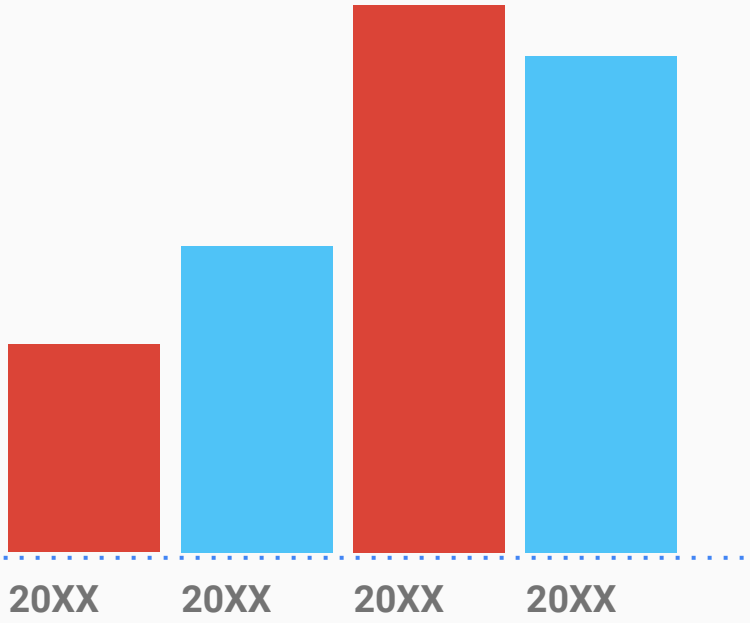
OUTPUT


Call stored procedure gdb0041.get_market_badge
Enter values for parameters of your procedure and click <Execute> to create an SQL editor and run the call:
in_market
India
[IN]
varchar(45)
in_fiscal_year
2021
[IN]
year
out_badge
[OUT]
varchar(45)
Execute
Cancel


Result Grid
@out_badge
Gold

PROBLEM STATEMENT

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 action to address any potential issues.



The Report should have the following -

- **Top Markets**
- **Top Products**
- **Top Customers**

{ Write Stored Procedure for each }

Creating View Table (For Net_Invoice_Sales)

```
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, g.gross_price*s.sold_quantity AS gross_price_total,
pre.pre_invoice_discount_pct

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 g   ON g.product_code= s.product_code AND  g.fiscal_year=s.fiscal_year

JOIN fact_pre_invoice_deductions pre   ON pre.customer_code=s.customer_code AND
pre.fiscal_year=s.fiscal_year
```

CODE FUSION

```
SELECT *,
(gross_price_total - gross_price_total*pre_invoice_discount_pct)
AS net_invoice_sales

FROM sales_preinv_discount;
```

OR

```
SELECT *,
(1 - pre_invoice_discount_pct)
AS net_invoice_sales

FROM sales_preinv_discount;
```

OUTPUT

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

	variant	sold_quantity	gross_price	gross_price_total	pre_invoice_discount_pct	net_invoice_sales
) – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	51	15.3952	785.1552	0.0824	720.45841152
) – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	77	15.3952	1185.4304	0.2956	835.01717376
) – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	17	15.3952	261.7184	0.0536	247.69029376
) – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	6	15.3952	92.3712	0.2378	70.40532864
) – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	5	15.3952	76.9760	0.1057	68.83963680

Creating View Table (For gross_sales)

```
CREATE  VIEW gross sales AS

SELECT s.date,s.fiscal_year,s.customer_code,c.customer,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

FROM fact_sales_monthly s

JOIN dim_product p ON s.product_code=p.product_code

JOIN dim_customer c ON s.customer_code=c.customer_code

JOIN fact_gross_price g ON g.fiscal_year=s.fiscal_year AND g.product_code=s.product_code;
```

Creating View Table (For Net_Sales)

```
CREATE VIEW sales_postinv_discount AS

SELECT spre.date,spre.fiscal_year, spre.customer_code, spre.market,spre.product_code, spre.product,
spre.variant,  spre.sold_quantity, spre.gross_price,  spre.pre_invoice_discount_pct,

(spre.gross_price_total - spre.gross_price_total*spre.pre_invoice_discount_pct)  AS  net_invoice_sales,
(spro.discounts_pct + spro.other_deductions_pct)  AS  post_invoice_discount_pct

FROM sales_preinv_discount  spre

JOIN fact_post_invoice_deductions  spro

ON spro.customer_code=spre.customer_code

AND spro.product_code=spre.product_code

AND spro.date=spre.date
```

CODE FUSION

```
SELECT *,
(gross_price_total -
gross_price_total*pre_invoice_discount_pct)*net_invoice_sales
AS  net_sales
FROM sales_postinv_discount;
```

OR

```
SELECT *,
(1 - pre_invoice_discount_pct)*net_invoice_sales
AS  net_sales
FROM sales_postinv_discount;
```

OUTPUT

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
tandard	51	785.1552	0.0824	720.45841152	0.3379	661.092638410752
tandard	77	1185.4304	0.2956	835.01717376	0.4013	588.186097196544
tandard	17	261.7184	0.0536	247.69029376	0.3752	234.414094014464
tandard	6	92.3712	0.2378	70.40532864	0.3446	53.662941489408
tandard	5	76.9760	0.1057	68.83963680	0.3065	61.563287190240

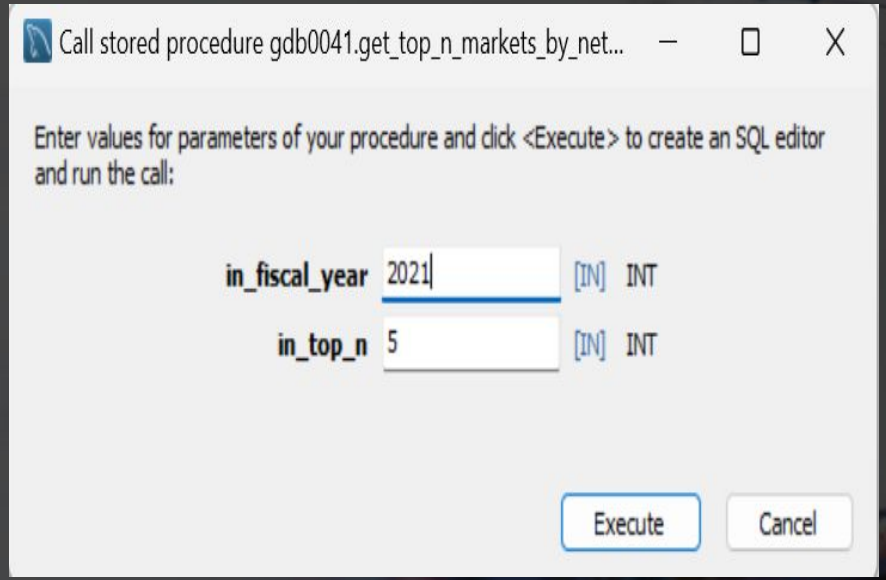
CODE FUSION- FOR TOP MARKET

```
SELECT market,
ROUND(sum(net_sales)/10000000,2) AS  net_sales_mln

FROM net_sales

WHERE fiscal_year=2021

GROUP BY market

ORDER BY net_sales_mln DESC
```

Creating Stored Procedure (For Top Markets)

```
CREATE PROCEDURE get_top_n_markets_by_net_sales ( in_fiscal_year INT, in_top_n INT)

BEGIN

SELECT market, ROUND(sum(net_sales)/10000000,2) 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;
```

OUTPUT



Result Grid | Filter Rows:

market	net_sales_mln
India	26.27
USA	15.87
South Korea	7.61
Canada	5.19
Philiphines	4.74

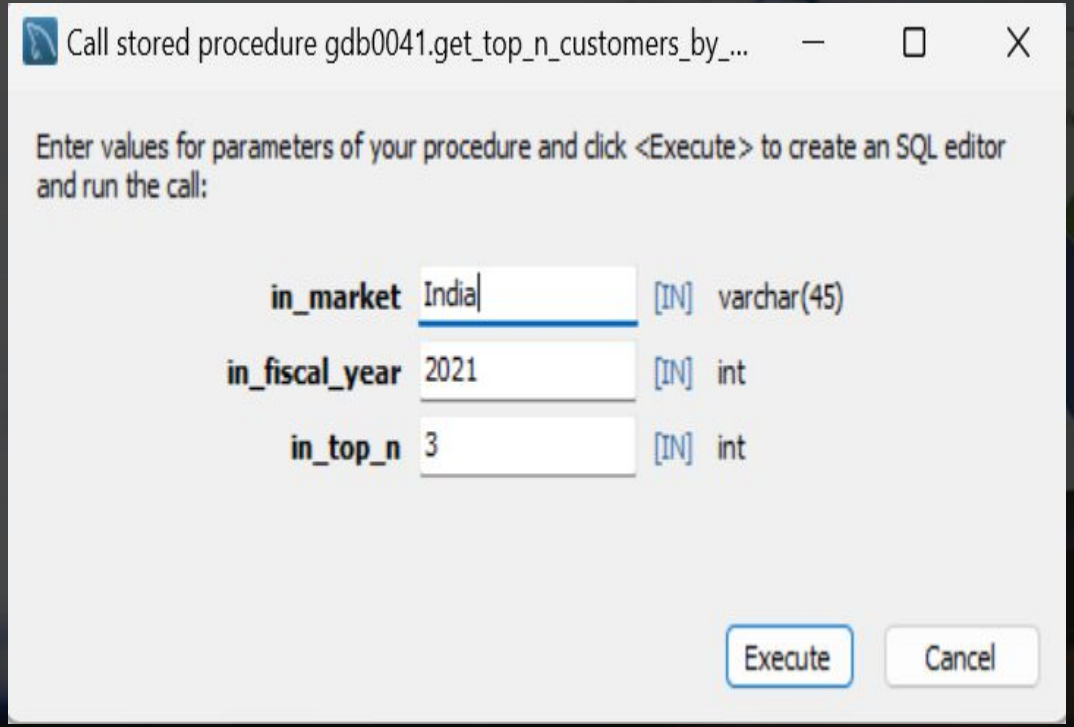
CODE FUSION- FOR TOP CUSTOMERS

```
SELECT c.customer,
ROUND(sum(net_sales)/10000000,2) AS  net_sales_mln

FROM net_sales n

JOIN dim_customer c
ON n.customer_code=c.customer_code

WHERE fiscal_year=2021

GROUP BY c.customer

ORDER BY net_sales_mln DESC
```

Creating Stored Procedure (For Top Customers)

```
CREATE PROCEDURE get_top_n_customers_by_net_sales ( in_market varchar(45), in_fiscal_year int, in_top_n int )

BEGIN

SELECT c.customer, ROUND(sum(net_sales)/10000000,2) AS net_sales_mln

FROM net_sales n

JOIN dim_customer c   ON n.customer_code=c.customer_code

WHERE n.fiscal_year=in_fiscal_year   AND n.market=in_market

GROUP BY c.customer

ORDER BY net_sales_mln desc

LIMIT in_top_n;

END
```

OUTPUT



Result Grid | Filter Rows:

customer	net_sales_mln
Amazon	3.70
Atliq Exclusive	3.28
Electricalsocity	1.62

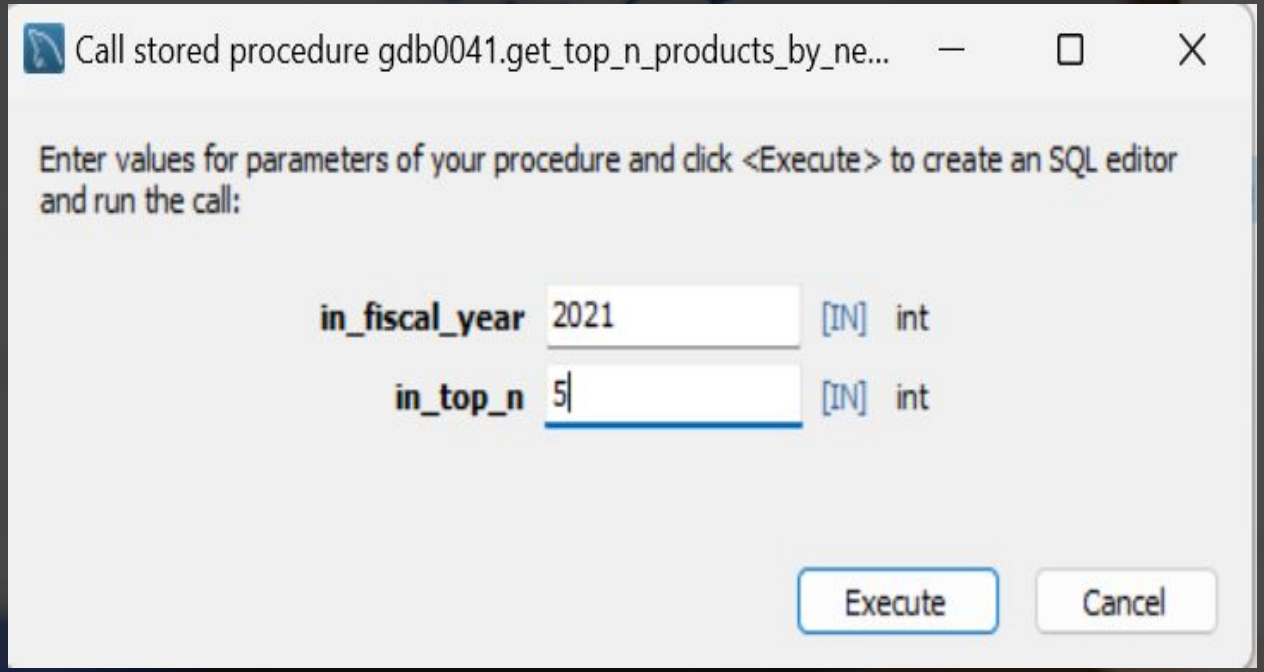
CODE FUSION- FOR TOP PRODUCTS

```
SELECT product,
ROUND(sum(net_sales)/10000000,2) AS  net_sales_mln

FROM net_sales n

WHERE fiscal_year=2021

GROUP BY product

ORDER BY net_sales_mln DESC
```

Creating Stored Procedure (For Top Products)

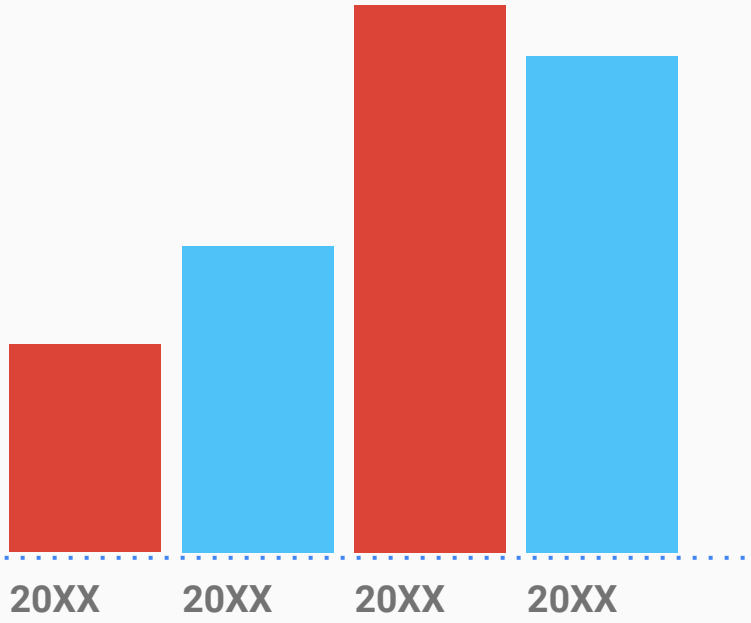
```
CREATE PROCEDURE get_top_n_products_by_net_sales ( in_fiscal_year int, in_top_n int )

BEGIN

SELECT product, ROUND(sum(net_sales)/10000000,2) AS net_sales_mln

FROM net_sales n

WHERE n.fiscal_year=in_fiscal_year

GROUP BY product

ORDER BY net_sales_mln desc

LIMIT in_top_n;

END
```

OUTPUT


Call stored procedure gdb0041.get_top_n_products_by_ne...
Enter values for parameters of your procedure and click <Execute> to create an SQL editor and run the call:
in_fiscal_year
2021
[IN]
int
in_top_n
5
[IN]
int
Execute
Cancel

PROBLEM STATEMENT

As a product owner I want to see a bar chart report for FY=2021 and for top 10 markets by % net sales.



CODE FUSION

```
WITH cte1 AS
(SELECT c.customer,
ROUND(sum(net_sales)/10000000,2) as net_sales_mln
FROM net_sales n
JOIN dim_customer c
ON n.customer_code=c.customer_code
WHERE n.fiscal_year=2021
GROUP BY c.customer)

SELECT *,
net_sales_mln*100/sum(net_sales_mln) over() AS pct
FROM cte1
ORDER BY net_sales_mln desc;
```

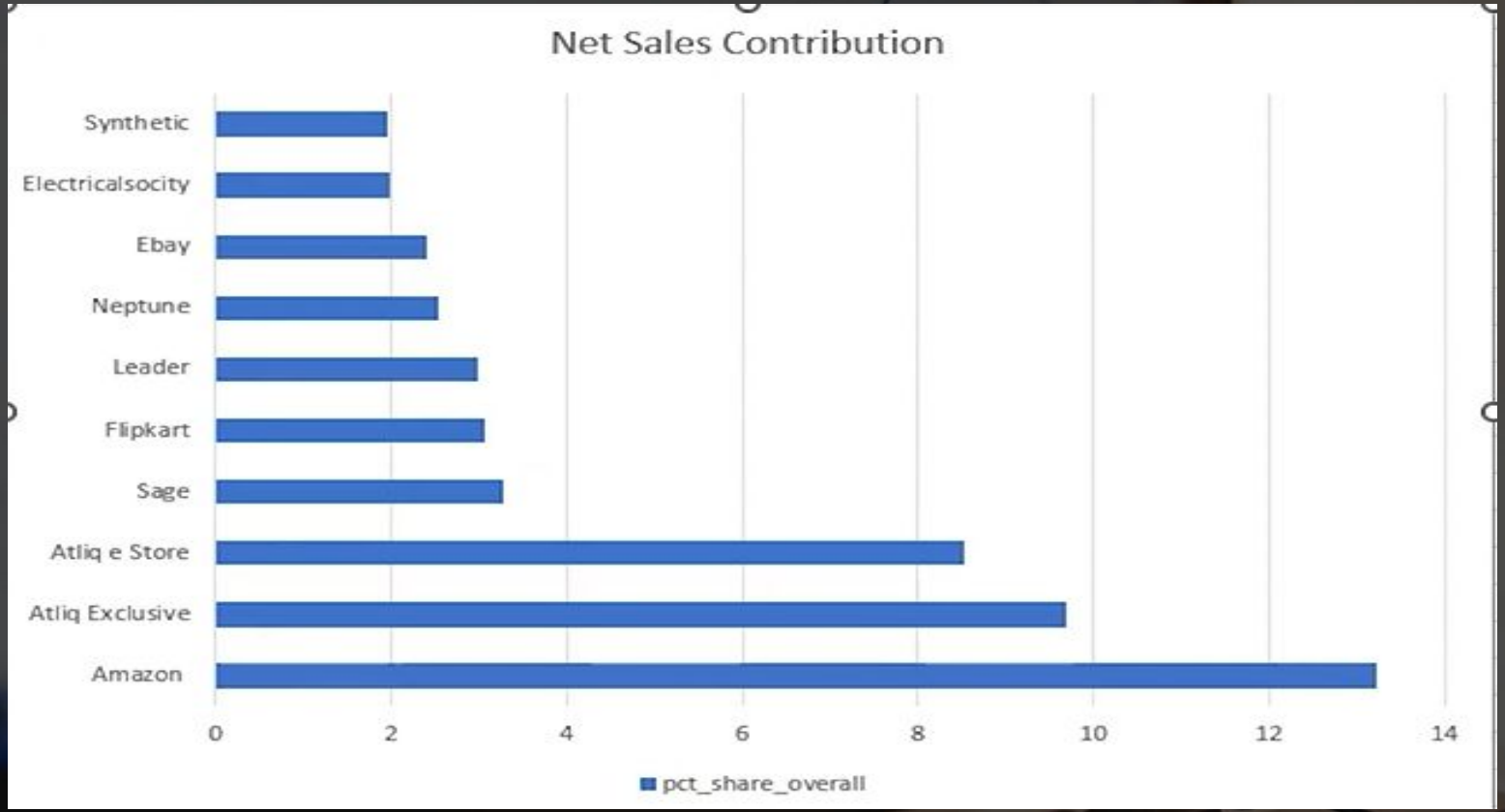
OUTPUT

Result Grid | Filter Rows: | Export:

customer	net_sales_mln
Amazon	109.03
Atliq Exclusive	79.92
Atliq e Store	70.31
Sage	27.07
Flipkart	25.25
Leader	24.52
Neptune	21.01

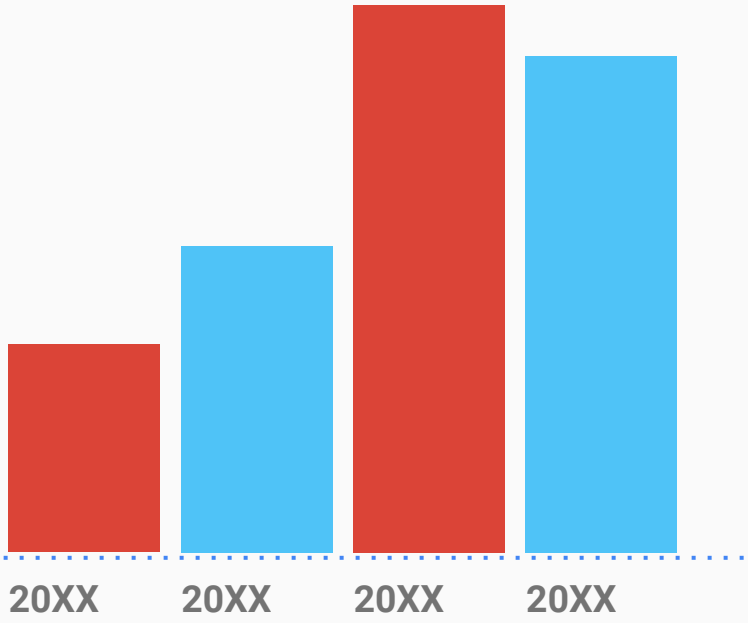
Result 1

OUTPUT


Net Sales Contribution
Synthetic
Electricalsocity
Ebay
Neptune
Leader
Flipkart
Sage
Atliq e Store
Atliq Exclusive
Amazon
0
2
4
6
8
10
12
14
pct_share_overall

PROBLEM STATEMENT

As a product owner I want to see region wise (APAC,EU,LTAM) % net sales breakdown by customers in a respective region so that I can perform my regional analysis on financial performance (FY=2021) of the company.



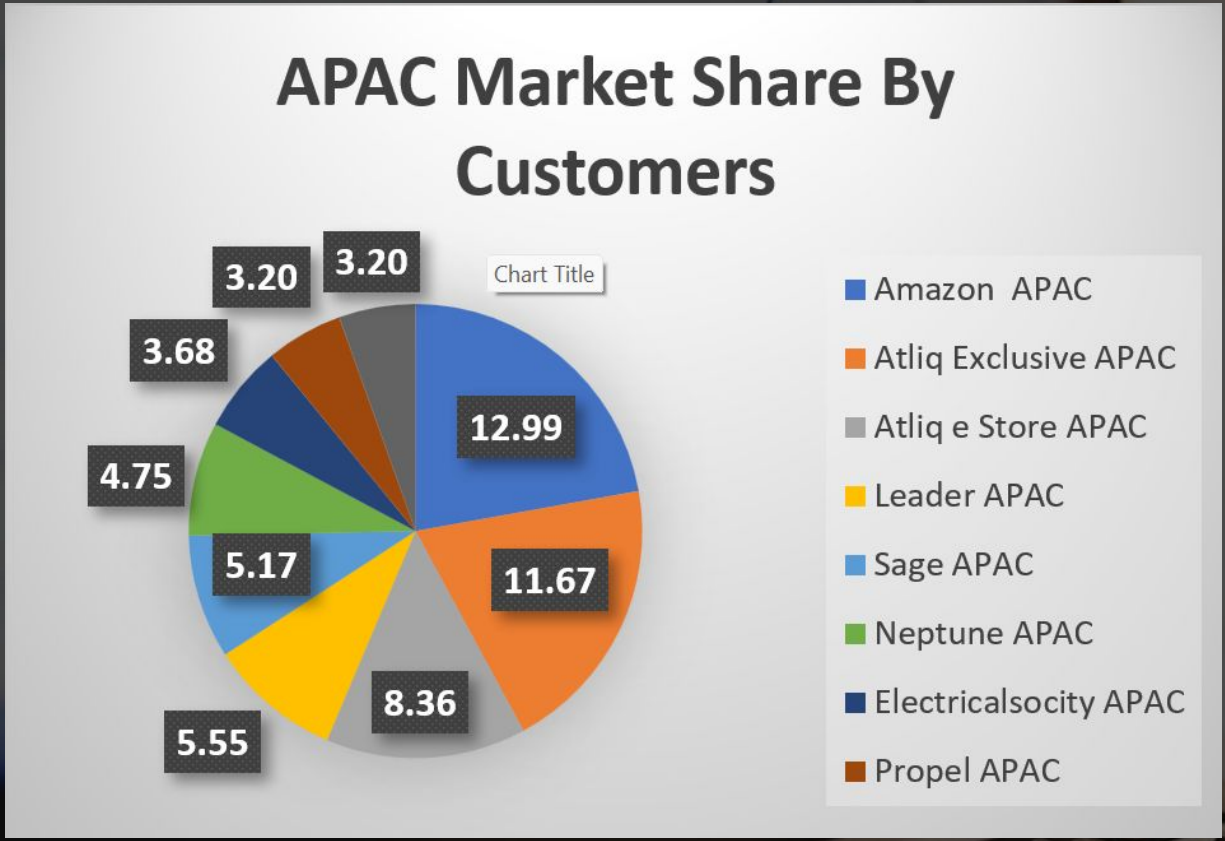
CODE FUSION

```
WITH cte1 as
(SELECT c.customer, c.region,
 ROUND(sum(net_sales)/10000000,2) as net_sales_mln
 FROM net_sales n
 JOIN dim_customer c
 ON n.customer_code=c.customer_code
 WHERE n.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, net_sales_mln desc;
```

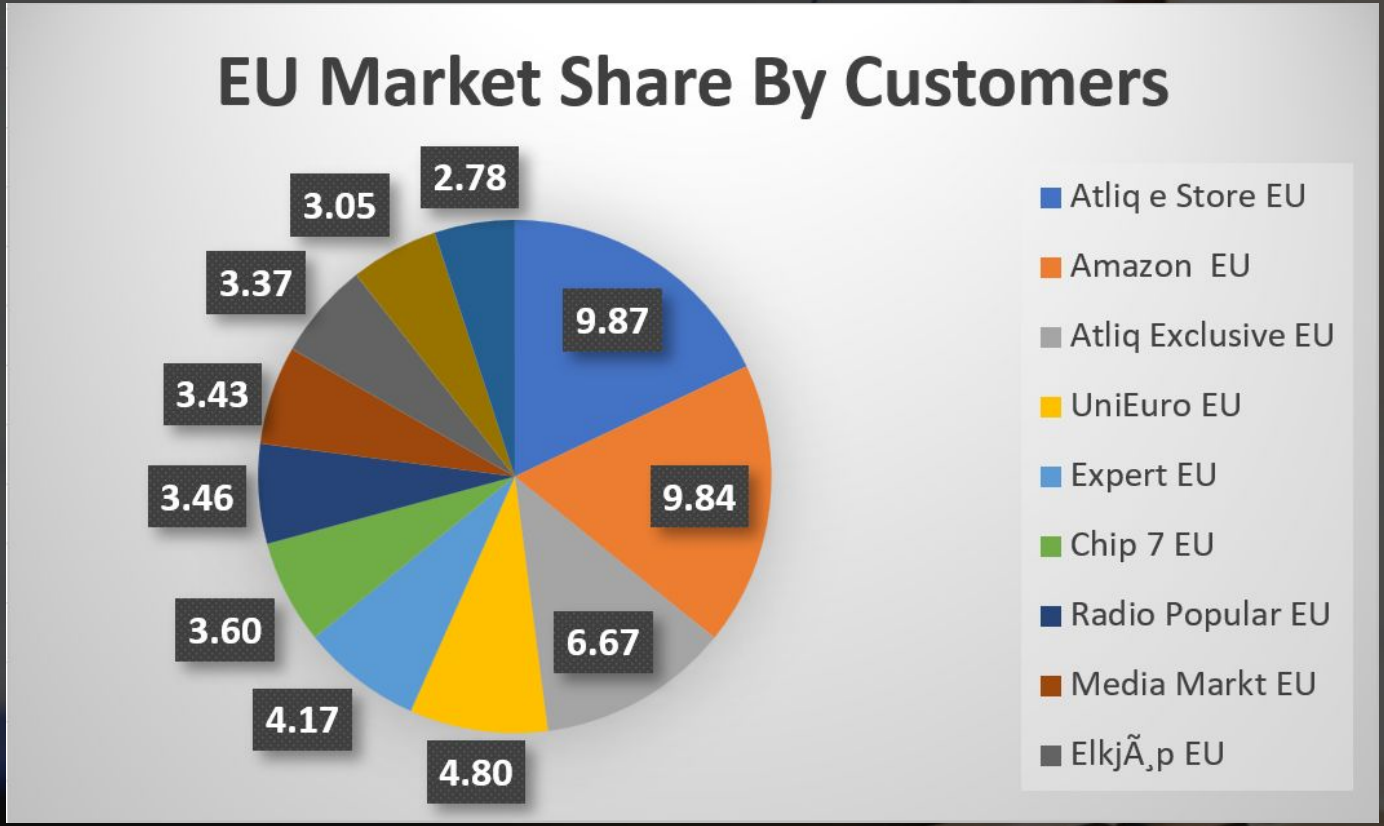
OUTPUT

customer	region	net_sales_mln	pct_share_region
Amazon	APAC	57.41	12.988688
Atliq Exclusive	APAC	51.58	11.669683
Atliq e Store	APAC	36.97	8.364253
Leader	APAC	24.52	5.547511
Sage	APAC	22.85	5.169683
Neptune	APAC	21.01	4.753394
Electricalsocity	APAC	16.25	3.676471
Propel	APAC	14.14	3.199095
Synthetic	APAC	14.14	3.199095
Flipkart	APAC	12.96	2.932127
Novus	APAC	12.91	2.920814
Expression	APAC	12.90	2.918552

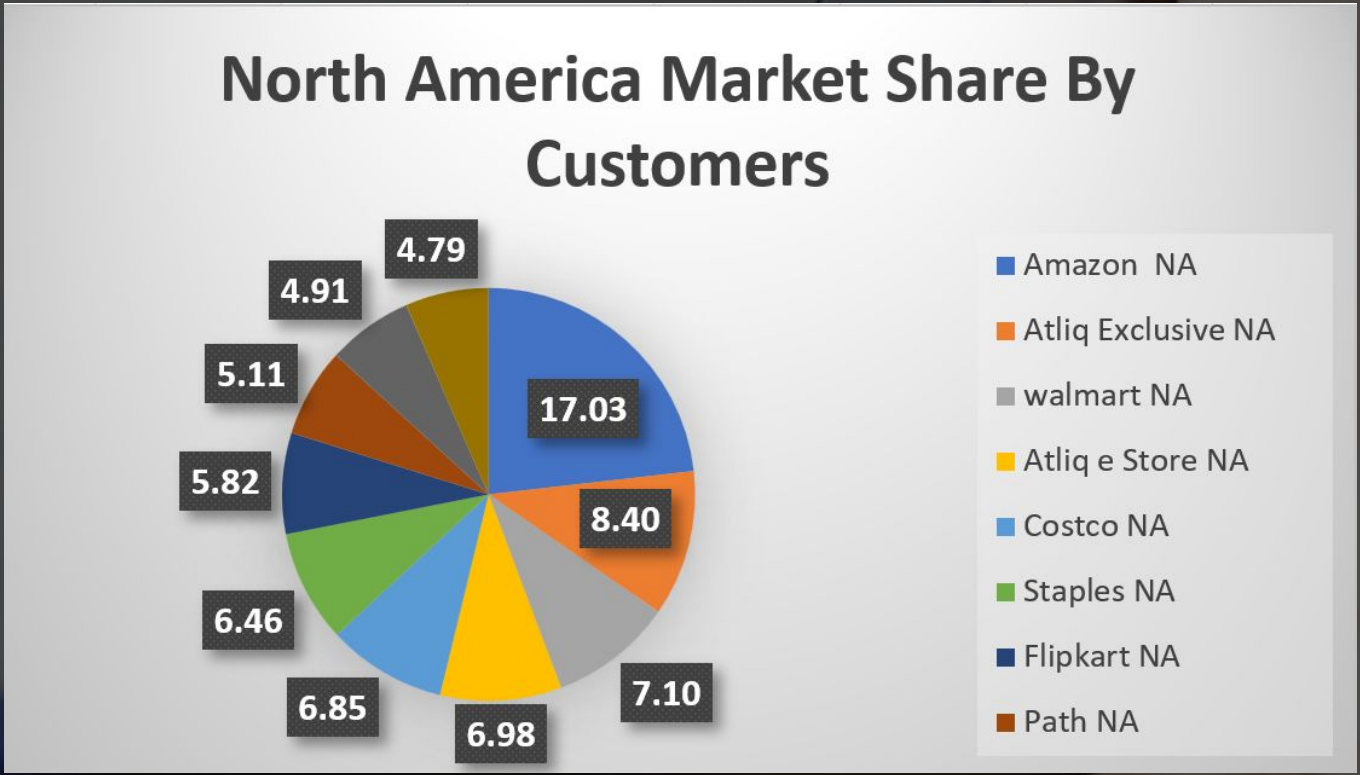
OUTPUT


APAC Market Share By Customers
Chart Title
12.99
11.67
8.36
5.55
5.17
4.75
3.68
3.20
3.20
Amazon APAC
Atliq Exclusive APAC
Atliq e Store APAC
Leader APAC
Sage APAC
Neptune APAC
Electricalsocity APAC
Propel APAC

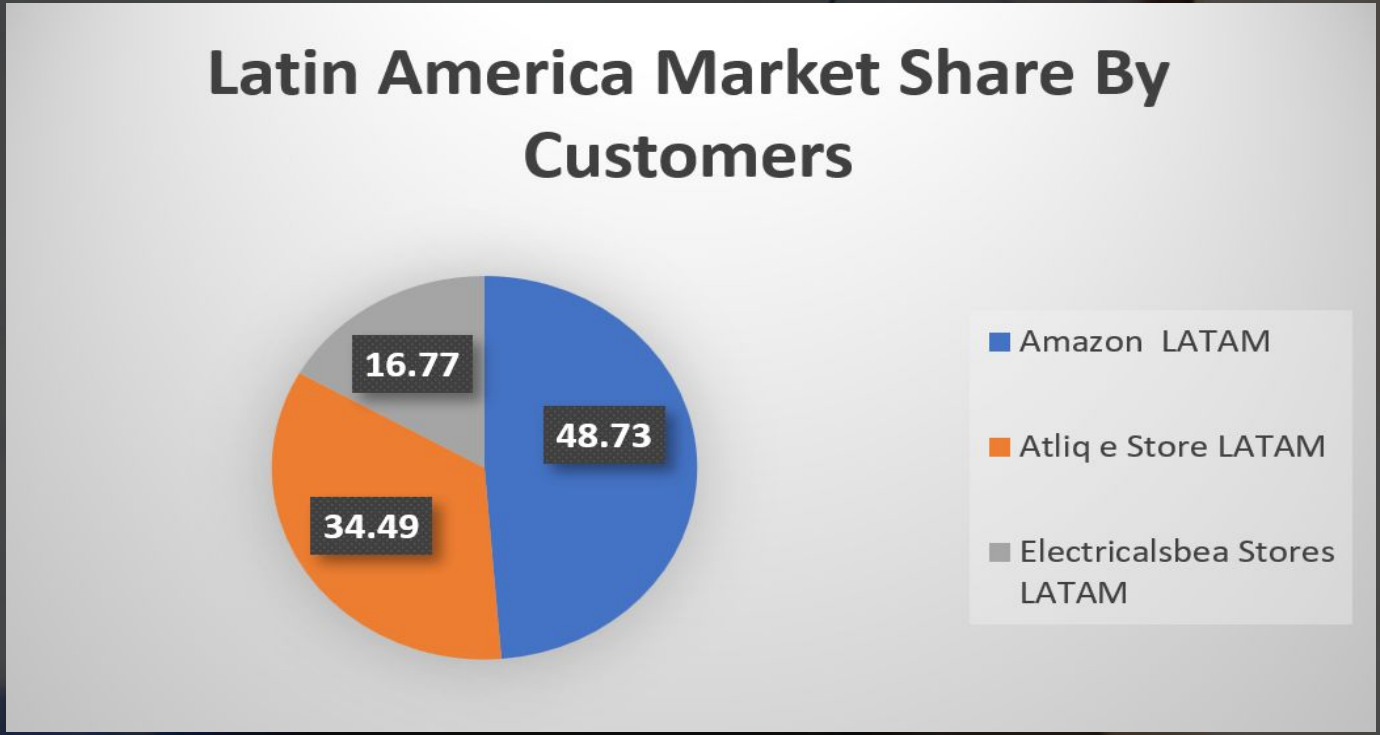
OUTPUT


EU Market Share By Customers
2.78
3.05
3.37
3.43
3.46
3.60
4.17
4.80
6.67
9.84
9.87
Atliq e Store EU
Amazon EU
Atliq Exclusive EU
UniEuro EU
Expert EU
Chip 7 EU
Radio Popular EU
Media Markt EU
ElkjÃ¸p EU

OUTPUT

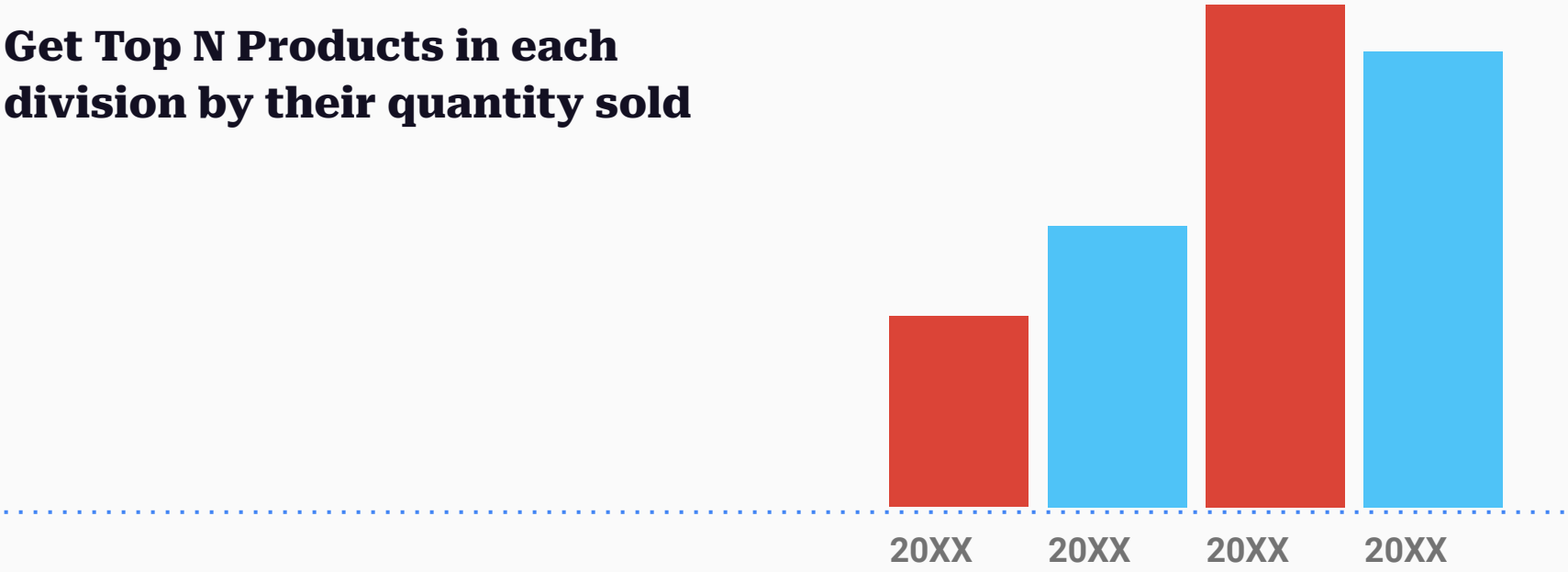

North America Market Share By Customers
4.79
4.91
5.11
5.82
6.46
6.85
6.98
7.10
8.40
17.03
Amazon NA
Atliq Exclusive NA
walmart NA
Atliq e Store NA
Costco NA
Staples NA
Flipkart NA
Path NA

OUTPUT


Latin America Market Share By Customers
16.77
48.73
34.49
Amazon LATAM
Atliq e Store LATAM
Electricalsbea Stores LATAM

PROBLEM STATEMENT

Get Top N Products in each division by their quantity sold



CODE FUSION

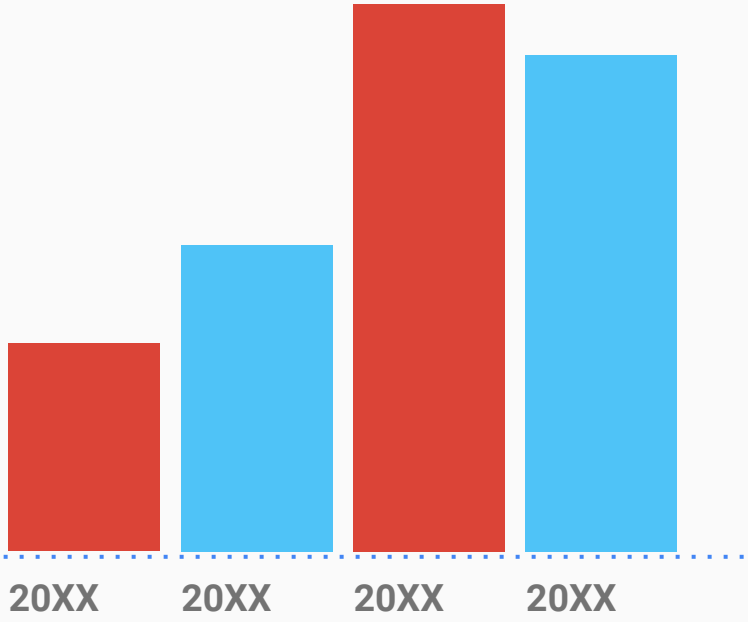
```
WITH CTE1 as(
                    SELECT p.division, p.product,
                    sum(sold_quantity) AS total_qty
                    FROM fact_sales_monthly s
                    JOIN dim_product p
                    ON p.product_code=s.product_code
                    WHERE fiscal_year=2021
                    GROUP BY p.product),
        CTE2 AS(
                    SELECT *,
                    dense_rank() over(partition by division order by
total_qty DESC) as drnk
                     FROM CTE1)
SELECT * FROM CTE2   WHERE drnk<=3;
```

OUTPUT

	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

PROBLEM STATEMENT

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



CODE FUSION

```
with cte1 as (
        select
            c.market,c.region,
            round(sum(gross_price_total)/1000000,2) as
gross_sales_mln
            from net_sales s
            join dim_customer c
            on c.customer_code=s.customer_code
            where fiscal_year=2021
            group by market
            order by gross_sales_mln desc),
        cte2 as (select *,
            dense_rank() over(partition by region order by
gross_sales_mln desc) as drnk
            from cte1)
    select * from cte2 where drnk<=2
```

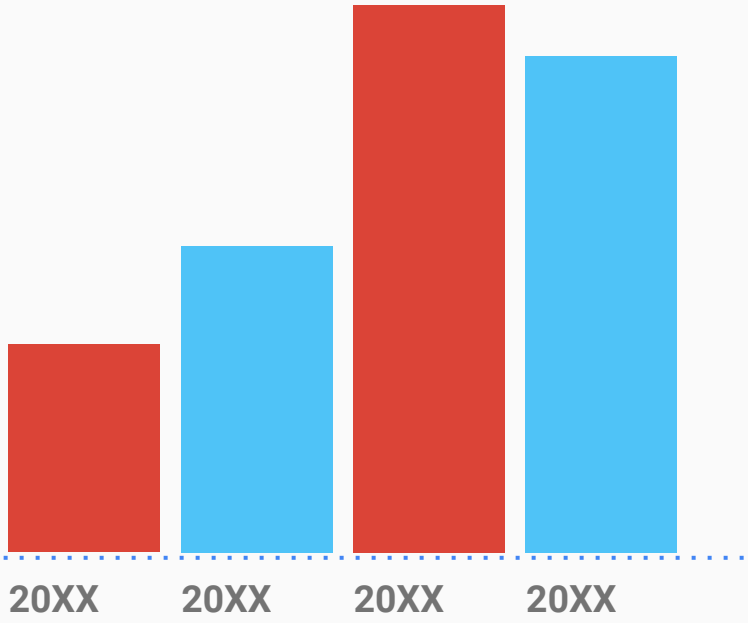
OUTPUT

	market	region	gross_sales_mln	drnk
▶	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

SUPPLY CHAIN ANALYTICS

PROBLEM STATEMENT

As a Product Owner I need an aggregate forecast accuracy report for all the customers for a given fiscal year so that I can track the accuracy of the forecast we make for these customers.



The Report should have the following fields-

- Customer_code
- Name, Market
- Total Forecast Quantity
- Total Sold Quantity
- Net Error
- Absolute Error
- Forecast Accuracy%

CODE FUSION

```
CREATE TABLE fact_act_est(
                    SELECT s.date AS Date,
                           s.fiscal_year AS fiscal_year,
                           s.customer_code AS customer_code,
                           s.product_code AS product_code,
                           s.sold_quantity AS sold_quantity,
                           f.forecast_quantity AS forecast_quantity
                  FROM fact_sales_monthly s
          LEFT JOIN fact_forecast_monthly f
          USING(date,customer_code,product_code)
UNION
          SELECT f.date AS Date,
                 f.fiscal_year AS fiscal_year,
                 f.customer_code AS customer_code,
                 f.product_code AS product_code,
                 s.sold_quantity AS sold_quantity,
                 f.forecast_quantity AS forecast_quantity
        FROM fact_forecast_monthly f
     LEFT JOIN fact_sales_monthly s
     USING(date,customer_code,product_code));
```

CODE FUSION

```
#Adding Value as O to Null Queries In New Created Table fact_act_est#

SELECT * FROM fact_act_est;

UPDATE fact_act_est
SET sold_quantity=0
WHERE sold_quantity IS NULL;

SET SQL_SAFE_UPDATES = 0;          #To disable safe updates in MySql#

UPDATE fact_act_est
SET forecast_quantity=0
WHERE forecast_quantity IS NULL;
```

CODE FUSION

#Calculating the forecast_accuracy#

```
WITH forecast_error_table AS(
  SELECT s.customer_code,
  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 absolute_error,
  sum(abs(forecast_quantity-sold_quantity))*100/sum(forecast_quantity) as
absolute_error_pct
  FROM fact_act_est s
  WHERE s.fiscal_year=2021
  GROUP BY s.customer_code
)
  SELECT c.customer,c.market, e.*,
  if(absolute_error_pct>100, 0, 100-absolute_error_pct) as forecast_accuracy
  FROM forecast_error_table e
  JOIN dim_customer c
  USING(customer_code)
  ORDER BY forecast_accuracy DESC;
```

OUTPUT

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

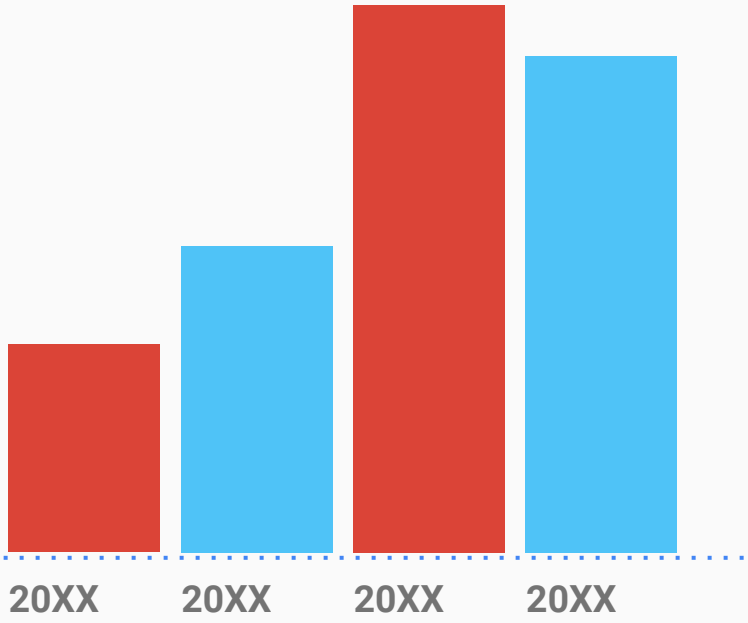
customer	market	customer_code	net_error	net_error_pct	absolute_error	absolute_error_pct	forecast
Coolblue	Italy	90013120	23985	17.9620	70467	52.7716	47.2284
Atliq e Store	Bangladesh	70010048	22571	15.8940	75711	53.3139	46.6861
Costco	Canada	90023027	43773	15.6353	149303	53.3297	46.6703
Relief	Canada	90023026	44504	16.2725	146948	53.7303	46.2697
Forward Stores	Portugal	90017051	31244	26.4629	63568	53.8406	46.1594
Mbit	Portugal	90017058	23335	21.1761	59473	53.9707	46.0293

Creating Stored Procedure (For Forecast_accuracy)

```
CREATE PROCEDURE ǧet_forecast_accuracỳ(in_fiscal_year int)

BEGIN

WITH forecast_error_table AS (SELECT s.customer_code, 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 absolute_error, sum(abs(forecast_quantity-sold_quantity))*100/sum(forecast_quantity) as absolute_error_pct FROM fact_act_est s  WHERE s.fiscal_year=in_fiscal_year

GROUP BY s.customer_code)SELECT c.customer,c.market, e.*, if(absolute_error_pct>100, 0, 100-absolute_error_pct) as forecast_accuracy

FROM forecast_error_table e

JOIN dim_customer c USING(customer_code) ORDER BY forecast_accuracy DESC;    END
```

PROBLEM STATEMENT

The supply chain business manager wants to see which customers' forecast accuracy has dropped from 2020 to 2021. Provide a complete report with these columns: customer_code, customer_name, market, forecast_accuracy_2020, forecast_accuracy_2021



CODE FUSION

```
# step 1: Get forecast accuracy of FY 2021 and store that in a temporary table#

create temporary table forecast_accuracy_2021
with forecast_err_table as (
      select
            s.customer_code as customer_code,
            c.customer as customer_name,
            c.market as market,
            sum(s.sold_quantity) as total_sold_qty,
            sum(s.forecast_quantity) as total_forecast_qty,
            sum(s.forecast_quantity-s.sold_quantity) as net_error,
            round(sum(s.forecast_quantity-s.sold_quantity)*100/sum(s.forecast_quantity),1) as net_error_pct,
            sum(abs(s.forecast_quantity-s.sold_quantity)) as abs_error,
            round(sum(abs(s.forecast_quantity-sold_quantity))*100/sum(s.forecast_quantity),2) as abs_error_pct
      from fact_act_est s
      join dim_customer c
      on s.customer_code = c.customer_code
      where s.fiscal_year=2021
      group by customer_code
)
select
      *,
   if (abs_error_pct > 100, 0, 100.0 - abs_error_pct) as forecast_accuracy
from
         forecast_err_table
order by forecast_accuracy desc;
```

CODE FUSION

```
# step 2: Get forecast accuracy of FY 2020 and store that also  in a temporary table#

create temporary table forecast_accuracy_2020
with forecast_err_table as (
      select
            s.customer_code as customer_code,
            c.customer as customer_name,
            c.market as market,
            sum(s.sold_quantity) as total_sold_qty,
            sum(s.forecast_quantity) as total_forecast_qty,
            sum(s.forecast_quantity-s.sold_quantity) as net_error,
            round(sum(s.forecast_quantity-s.sold_quantity)*100/sum(s.forecast_quantity),1) as net_error_pct,
            sum(abs(s.forecast_quantity-s.sold_quantity)) as abs_error,
            round(sum(abs(s.forecast_quantity-sold_quantity))*100/sum(s.forecast_quantity),2) as abs_error_pct
      from fact_act_est s
      join dim_customer c
      on s.customer_code = c.customer_code
      where s.fiscal_year=2020
      group by customer_code
)
select
      *,
   if (abs_error_pct > 100, 0, 100.0 - abs_error_pct) as forecast_accuracy
from
         forecast_err_table
order by forecast_accuracy desc;
```

CODE FUSION

```
# step 3: Join forecast accuracy tables for 2020 and 2021 using a customer_code      #

select
        f_2020.customer_code,
        f_2020.customer_name,
        f_2020.market,
        f_2020.forecast_accuracy as forecast_acc_2020,
        f_2021.forecast_accuracy as forecast_acc_2021
from forecast_accuracy_2020 f_2020
join forecast_accuracy_2021 f_2021
on f_2020.customer_code = f_2021.customer_code
where f_2021.forecast_accuracy < f_2020.forecast_accuracy
order by forecast_acc_2020 desc;
```

OUTPUT

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

customer_code	customer_name	market	forecast_acc_2020	forecast_acc_2021
70006158	Atliq e Store	Philiphines	42.65	24.49
70008170	Atliq e Store	Australia	40.96	38.74
90005161	Zone	Pakistan	40.08	37.10
90014140	Radio Popular	Netherlands	38.53	0.00
90008166	Sound	Australia	38.51	36.79
70014143	Atliq e Store	Netherlands	38.32	0.00
90004062	Flawless Stores	Japan	38.22	32.56
90014137	Media Markt	Netherlands	37.85	0.00
90014138	Mbit	Netherlands	37.83	0.00
70004069	Atliq Exclusive	Japan	37.62	32.09
90014136	Reliance Digital	Netherlands	37.59	0.00
80006154	Synthetic	Philiphines	37.49	24.63
70014142	Atliq Exclusive	Netherlands	37.43	0.00
90014141	Amazon	Netherlands	37.39	0.00
90005160	Nomad Stores	Pakistan	37.30	37.29
90006156	Amazon	Philiphines	37.21	27.94
90008164	Digimarket	Australia	37.15	36.01

Result 2

THANK YOU !