



# SQL PROJECT

## ATLIQ HARDWARE FINANCE AND SUPPLY CHAIN ANALYTICS



*Presented by: Vishal Kumar Gupta*

# OVERVIEW

01

Objective

02

About Company

03

QUERIES ,INSIGHTS &  
VISUALISATIONS





# OBJECTIVES

Atliq Hardwares (imaginary company) is one of the leading computer hardware producers in India and well expanded in other countries too.

However, the management noticed that they do not get enough insights to make quick and smart data-informed decisions. They want to expand their data analytics team by adding several junior data analysts.

Wanda Mishra, Product Owner wanted to hire someone who is good at both tech and soft skills. Hence, he decided to conduct a SQL challenge which will help him understand both the skills



# ABOUT COMPANY

Atliq Hardwares, a leading computer hardware producer in India with a significant presence in other countries, identified a need for better data insights to support quick and informed decision-making. It sells many types of hardware products to multiple customers in different countries, and regions.



# 1. Croma India Product wise sales report for fiscal year 2021.

The Report should have the following fields.

- a.) Month
- b.) Product Name & Variant
- c.) Sold Quantity
- d.) Gross Price Per item
- e.) Gross Price Total

- Step 1: create a function 'get\_fiscal\_year' to get fiscal year by passing the date

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
```

- Step 2: query

```
select
s.date,s.product_code,
p.product,p.variant,s.sold_quantity,
g.gross_price,
(gross_price*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
```

# Output

date	product	variant	sold_quantity	gross_price	gross_price_total
2020-09-01	AQ Dracula HDD â€“ 3.5 Inch SATA 6 Gb/s 5400 RPM 256 MB Cache	Standard	202	19.0573	3849.57
2020-09-01	AQ 5000 Series Electron 9 5900X Desktop Processor	Premium	159	123.9315	19705.10
2020-09-01	AQ Marquee P3	Plus 2	172	42.3841	7290.06
2020-09-01	AQ Mforce Gen Z	Plus 2	173	36.5679	6326.24
2020-09-01	AQ 5000 Series Electron 8 5900X Desktop Processor	Standard	106	124.264	13171.9
2020-09-01	AQ Mforce Gen X	Standard 3	23	19.5235	449.04
2020-09-01	AQ GT 21	Premium	182	39.1215	7120.1
2020-09-01	AQ 5000 Series Ultron 8 5900X Desktop Processor	Plus	168	145.1474	24384.76
2020-09-01	AQ WereWolf NAS Internal Hard Drive HDD â€“ 8.89 cm	Premium	123	23.6154	2904.69
2020-09-01	AQ Dracula HDD â€“ 3.5 Inch SATA 6 Gb/s 5400 RPM 256 MB Cache	Plus	162	21.4565	3475.9
2020-09-01	AQ Marquee P3	Premium	197	44.7044	8806.76
2020-09-01	AQ Mforce Gen Z	Standard 3	167	34.2412	5718.28
2020-09-01	AQ 5000 Series Electron 9 5900X Desktop Processor	Plus	51	123.4797	6297.46
2020-09-01	AQ Mforce Gen Z	Plus 3	182	37.5278	6830.05
2020-09-01	AQ Mforce Gen Y	Standard 3	38	26.5871	1010.30
2020-09-01	AQ Zion Saga	Plus	236	27.1027	6396.23
2020-09-01	AQ Marquee P4	Standard	56	45.0813	2524.55
2020-09-01	AQ Mforce Gen Y	Plus 3	28	31.2439	874.82
2020-09-01	AQ Dracula HDD â€“ 3.5 Inch SATA 6 Gb/s 5400 RPM 256 MB Cache	Premium	193	21.7795	4203.44
2020-09-01	AQ Marquee P3	Standard	120	40.5013	4860.1
2020-09-01	AQ Mforce Gen Z	Standard 1	51	30.4696	1553.94
2020-09-01	AQ Mforce Gen Y	Standard 2	72	24.9298	1794.94
2020-09-01	AQ Mforce Gen Z	Premium 1	177	36.2434	6415.08
2020-09-01	AQ 5000 Series Electron 8 5900X Desktop Processor	Plus	33	129.5836	4276.25
2020-09-01	AQ Mforce Gen Y	Premium 2	118	30.5816	3608.62
2020-09-01	AQ Mforce Gen Z	Premium 2	166	39.399	6540.2
2020-09-01	AQ 5000 Series Ultron 8 5900X Desktop Processor	Premium	153	142.9402	21869.85
2020-09-01	AQ Dracula HDD â€“ 3.5 Inch SATA 6 Gb/s 5400 RPM 256 MB Cache	Premium Plus	146	22.9729	3354.04
2020-09-01	AQ Electron 5 3600 Desktop Processor	Standard	152	149.1034	22663.71
2020-09-01	AQ Mforce Gen Z	Plus 1	122	37.5826	4585.07
2020-09-01	AQ Mforce Gen Y	Plus 1	149	26.1081	3890.10

## 2. Gross monthly total sales report for Croma

The Report should have the following fields.

a.) Month

b.) Total Gross Sales amount to Croma India in this month

select

```
s.date,s.product_code,  
p.product,p.variant,s.sold_quantity,  
g.gross_price,(gross_price*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
```

# Output

date	gross_price_total
2017-09-01	122407.5582
2017-10-01	162687.5716
2017-12-01	245673.8042
2018-01-01	127574.7372
2018-02-01	144799.5182
2018-04-01	130643.8976
2018-05-01	139165.0975
2018-06-01	125735.3786
2018-08-01	125409.8801
2018-09-01	343337.1651
2018-10-01	440562.0754
2018-12-01	653944.7486
2019-01-01	359025.0186
2019-02-01	356607.1729
2019-04-01	379549.685
2019-05-01	340152.2349
2019-06-01	343792.042
2019-08-01	338108.8774
2019-09-01	808250.4406
2019-10-01	1092622.1983
2019-12-01	1488174.0158
2020-01-01	812929.7497
2020-02-01	862762.7656
2020-04-01	130520.9209
2020-05-01	145049.0525
2020-06-01	362545.1364
2020-08-01	799327.6345
2020-09-01	2296919.6327
2020-10-01	3109316.8802

3Create a stored proc that can determine the market badge based on the following logic.

If total sold quantity > 5 million that market is connected Gold

else it is Silver.

My Input will be

- Market
- Fiscal Year

#### Output

- Market badge

Stored Procedure:

```
CREATE  
DEFINER='root'@'localhost'  
PROCEDURE  
'get_market_badge'  
(  
    IN in_market varchar(45),  
    IN in_fiscal_year year,  
    OUT out_badge varchar(45)  
)  
BEGIN  
declare qty int default 0;  
#set default market to me  
india  
if in_market="" then  
    set in_market="india";  
end if ;
```

```
# retrieve total qty for a given  
market+fyear  
    SELECT  
        SUM(sold_quantity) into qty  
    FROM fact_sales_monthly s  
    JOIN dim_customer c  
    ON  
        c.customer_code=s.customer_code  
    WHERE  
        get_fiscal_year(s.date)=in_fiscal_year  
        and c.market=in_market  
    group by c.market;  
  
# determine market badge  
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	2020	[IN] year
out_badge		[OUT] varchar(45)

```
1 • |set @out_badge = '0';  
2 • |call gdb0041.get_market_badge('india', 2020, @out_badge);  
3 • |select @out_badge;  
4
```

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

@out_badge
Gold

- 4. Write a Stored proc for**
- a.) **Top Market by net sales**
  - b.) **Top Product by net sales**
  - c.) **Top Customers by net sales**

**Step1: Database View for sales\_preinv\_discount**

```
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
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
```

**Step 2: Now generate net\_invoice\_sales using the above created view "sales\_preinv\_discount"**

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

### Step 3: Database View for sales\_postinv\_discount

```
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;
```

### Step 4: Now generate net\_sales using the above created view "sales\_postinv\_discount"

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

### Step 5: Finally creating the view `net\_sales` which inbuiltly use/include all the previous created view and gives the final result

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

# Output

Step 6: Stored proc to get top n markets by net sales for a given year

```
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
```

rocedure gdb0041.get\_top\_n\_market\_by\_net... — □ X

parameters of your procedure and click <Execute> to create an SQL editor

in_fiscal_year	2020	[IN] INT
in_top_n	5	[IN] INT

1 • call gdb0041.get\_top\_n\_market\_by\_net\_sales(2020, 5);  
2

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

	market	net_sales_mln
▶	India	64.73
	USA	46.35
	South Korea	22.38
	Philippines	17.45
	Canada	15.87

# Output

Step 7: 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

```
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
```

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	2020	[IN] int
in_top_n	3	[IN] int

```
1 • |call gdb0041.get_top_n_customers_by_net_sales('india', 2020, 3);
2
```

Result Grid		
	customer	net_sales_mln
▶	Amazon	12.68
	Atliq Exclusive	6.03
	Flipkart	5.61

# Output

Please enter values for parameters of your procedure and click <Execute> to create an SQL statement for the call:

in\_fiscal\_year  [IN] int  
in\_top\_n  [IN] INT

2

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

	product	net_sales_mln
▶	AQ Wi Power Dx2	14.37
	AQ BZ Gen Y	12.09
	AQ Wi Power Dx1	11.84

Step 8 : top n products by net sales

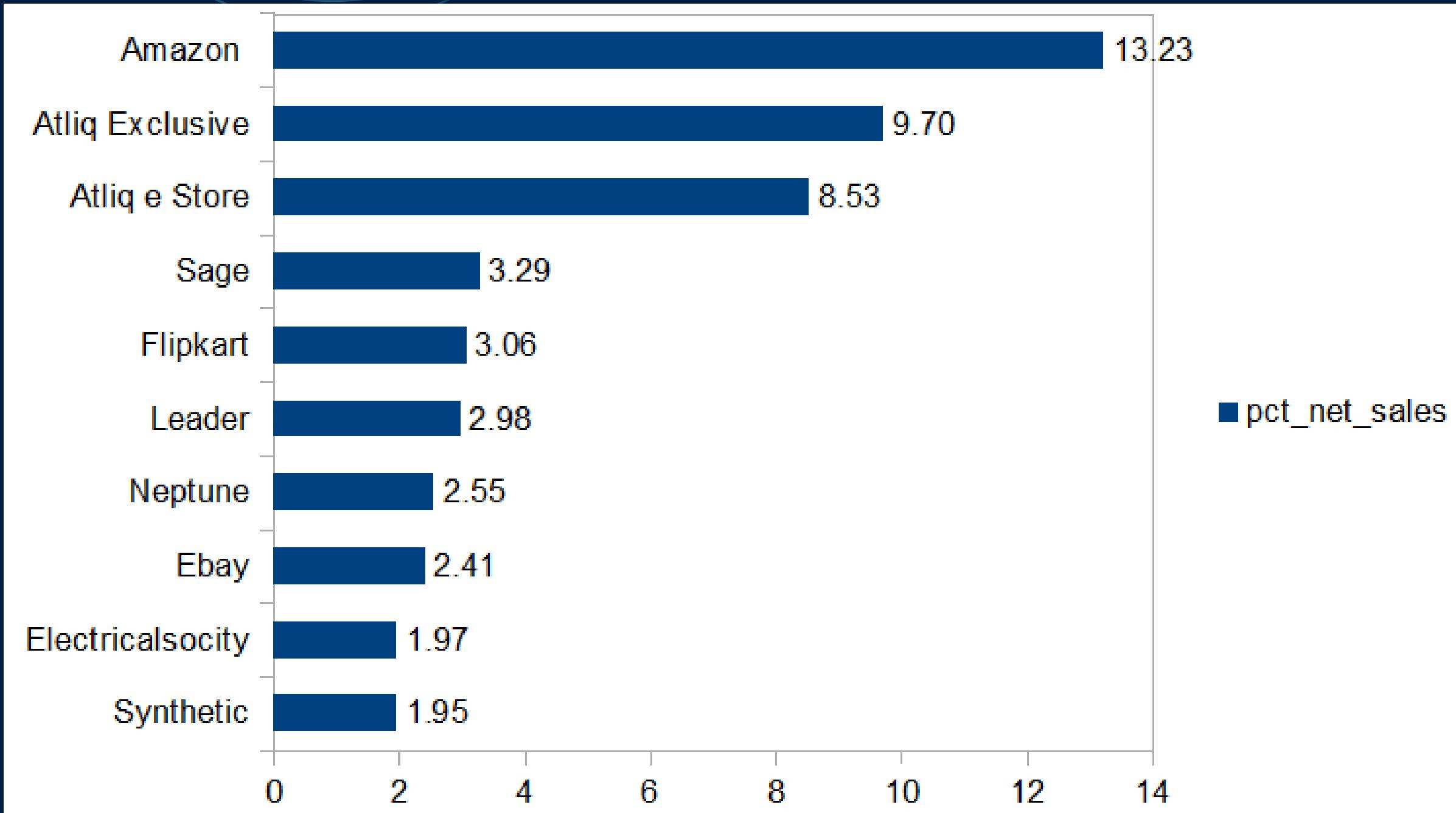
```
CREATE PROCEDURE
get_top_n_products_by_net_sales(
in_fiscal_year int,
in_top_n int
)
BEGIN
select
product,
round(sum(net_sales)/1000000,2) as
net_sales_mln
from gdb041.net_sales
where fiscal_year=in_fiscal_year
group by product
order by net_sales_mln desc
limit in_top_n;
END
```

# Visuals

## 5. Net sales % share Global

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

```
with cte1 as (  
    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=2021  
        group by customer)  
    select  
        *,  
        net_sales_mln*100/sum(net_sales_mln)  
        over() as pct_net_sales  
    from cte1  
    order by net_sales_mln desc
```



# Visuals

## 6. Net Sales % share by region

As a product owner , I want to see region wise (APAC, EU, LTAM etc)% net sales breakdown by customers in a respective region so that I can perform my regional analysis on financial performance of the company.0

FY=2021

with ctel 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

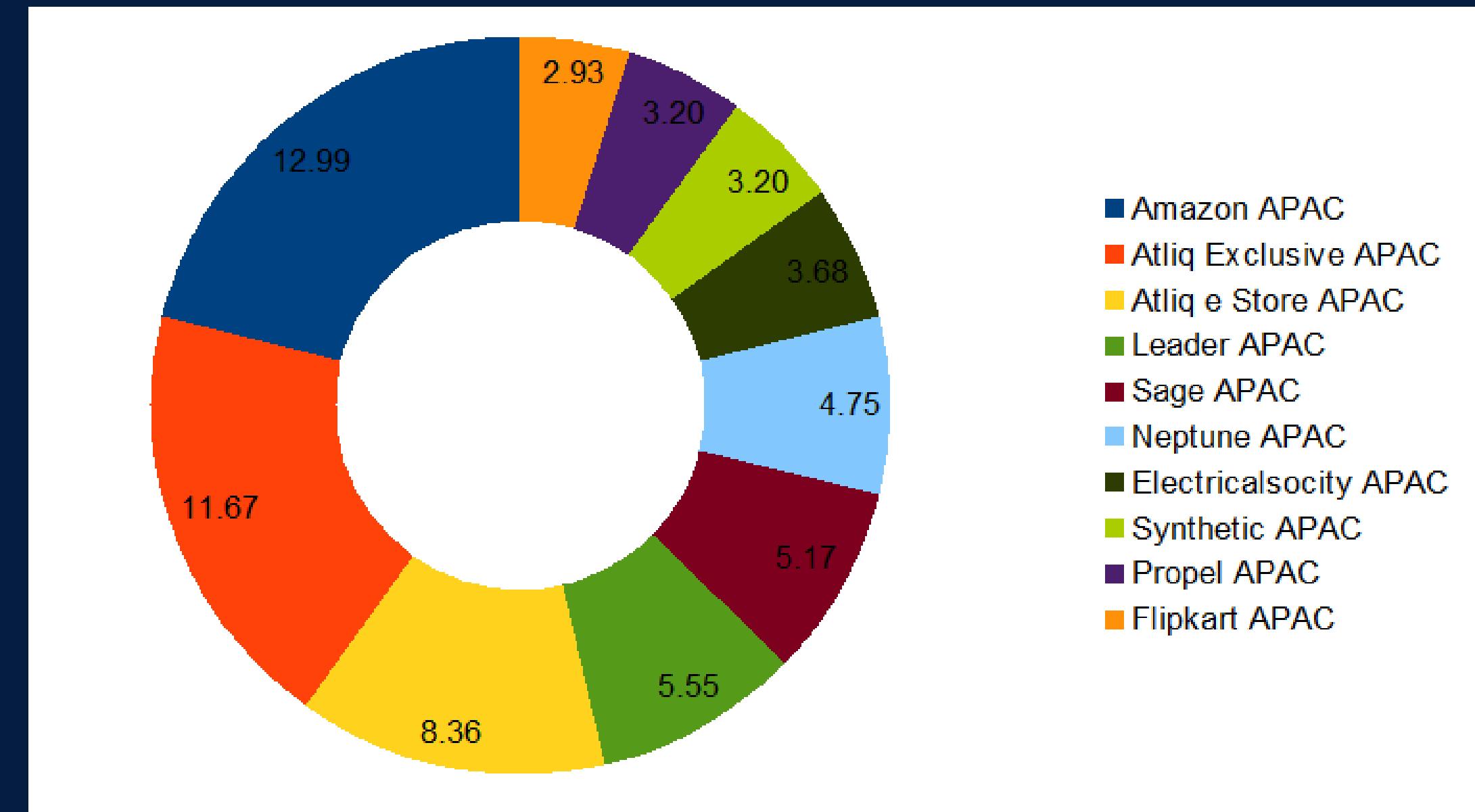
```
n.customer_code=c.customer_code  
where fiscal_year=2021
```

```
group by c.customer, c.region)select
```

\*

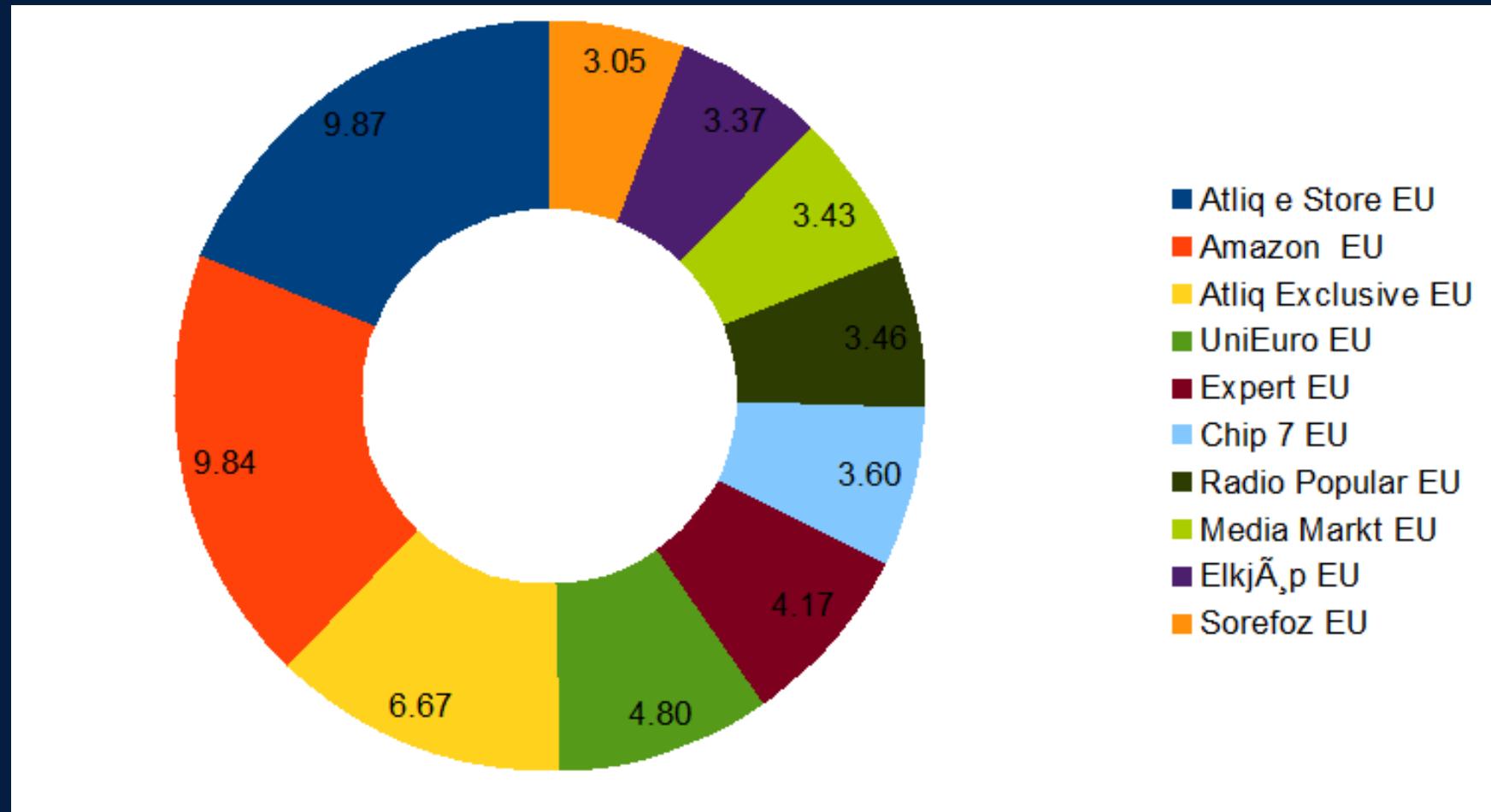
```
net_sales_mln*100/sum(net_sales_mln) over  
(partition by region) as pct_share_region  
from ctel  
order by region, pct_share_region desc
```

Net Sales % in APAC

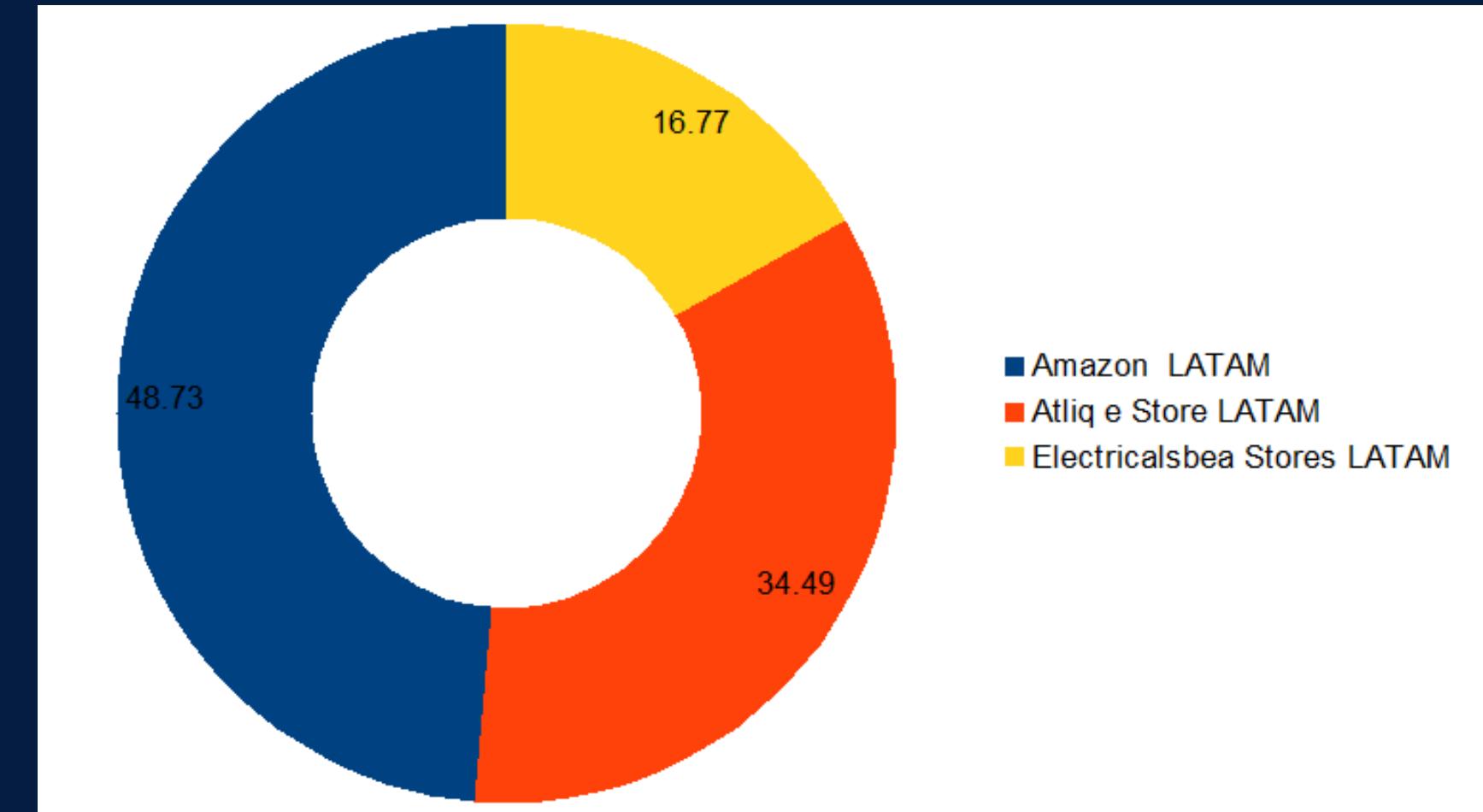


# Visuals

Net Sales % in EU



Net Sales % in LATAM



## 1. Forecast Accuracy for all customers for given fiscal year

- a.) Customer Code, Name, Market
- b.) Total Sold Quantity
- c.) Total Forecast Quantity
- d.) Net Error
- e.) Absolute Error
- f.) Forecast Accuracy %

```

create temporary table forecast_err_table
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_quan-
tity),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_quanti-
ty),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;
```

# Supply Chain

## Output

1	customer_code	customer_name	market	total_sold_qty	total_forecast_qty	net_error	net_error_pct	abs_error	abs_error_pct	forecast_accuracy
2	90013120	Coolblue	Italy	109547	133532	23985	18	70467	52.77	47.23
3	70010048	Atliq e Store	Bangladesh	119439	142010	22571	15.9	75711	53.31	46.69
4	90023027	Costco	Canada	236189	279962	43773	15.6	149303	53.33	46.67
5	90023026	Relief	Canada	228988	273492	44504	16.3	146948	53.73	46.27
6	90017051	Forward Stores	Portugal	86823	118067	31244	26.5	63568	53.84	46.16
7	90017058	Mbit	Portugal	86860	110195	23335	21.2	59473	53.97	46.03
8	90023028	walmart	Canada	239081	283323	44242	15.6	153058	54.02	45.98
9	90023024	Sage	Canada	246397	287233	40836	14.2	155610	54.18	45.82
10	90013124	Amazon	Italy	110898	136116	25218	18.5	73826	54.24	45.76
11	90015146	Mbit	Norway	147152	210507	63355	30.1	114189	54.24	45.76
12	90017054	Flawless Stores	Portugal	84371	114698	30327	26.4	62483	54.48	45.52
13	70027208	Atliq e Store	Brazil	33713	47321	13608	28.8	25784	54.49	45.51
14	90015147	Chiptec	Norway	154897	223867	68970	30.8	122100	54.54	45.46
15	80001019	Neptune	China	1113979	1275248	161269	12.6	695779	54.56	45.44
16	90015144	Sound	Norway	160074	225637	65563	29.1	123257	54.63	45.37
17	90009130	Logic Stores	Newzealand	103290	110175	6885	6.2	60225	54.66	45.34
18	90021088	Electricalslytical	United Kingdom	224350	323689	99339	30.7	176975	54.67	45.33
19	90015149	UniEuro	Norway	142086	212500	70414	33.1	116172	54.67	45.33
20	90017050	Electricalsara Stores	Portugal	85272	114688	29416	25.6	62760	54.72	45.28
21	70013125	Atliq Exclusive	Italy	101658	123428	21770	17.6	67546	54.73	45.27
22	90021094	Coolblue	United Kingdom	208512	301367	92855	30.8	165043	54.76	45.24
23	70009134	Atliq e Store	Newzealand	103747	110791	7044	6.4	60726	54.81	45.19

