Finance and Supply Chain Analytics

About AtliQ Hardwares and Problem Statement

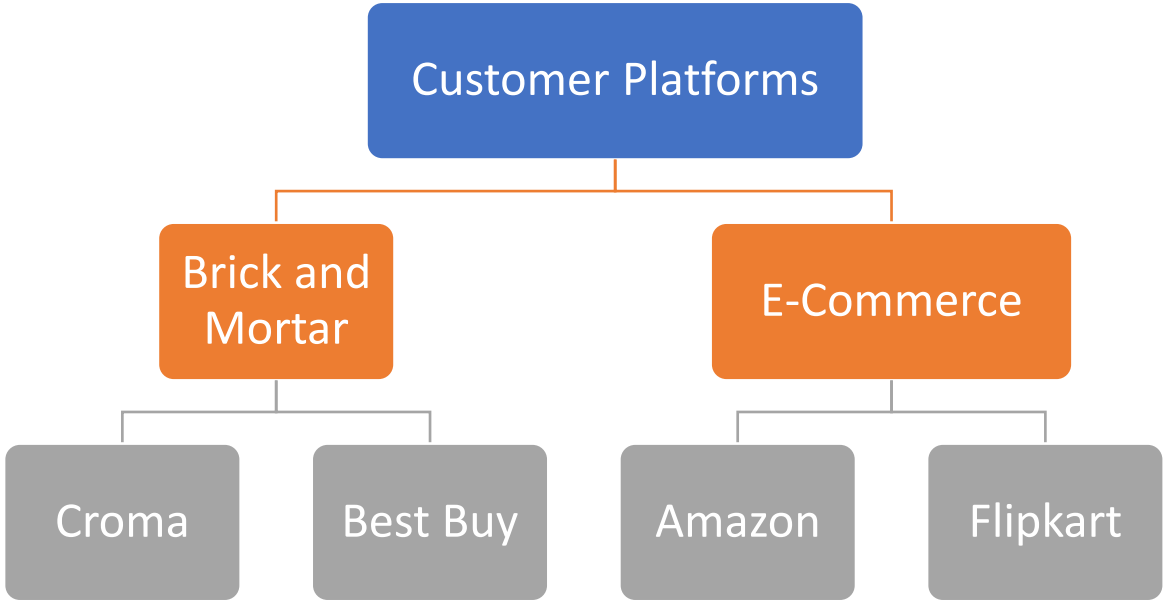
- AtliQ Harawares, a leading hardware company specializing in PCs, printers, mice, and computers with a global reach.

- The expanding size of Excel files has led to performance problems, resulting in unresponsiveness and inefficiency. AtliQ Hardware has launched a project to tackle this issue by assembling a team of data analysts. They will utilize MySQL as their database management system to extract meaningful insights from the data. These insights will empower the company to enhance decision-making and optimize operations, ultimately boosting overall performance.

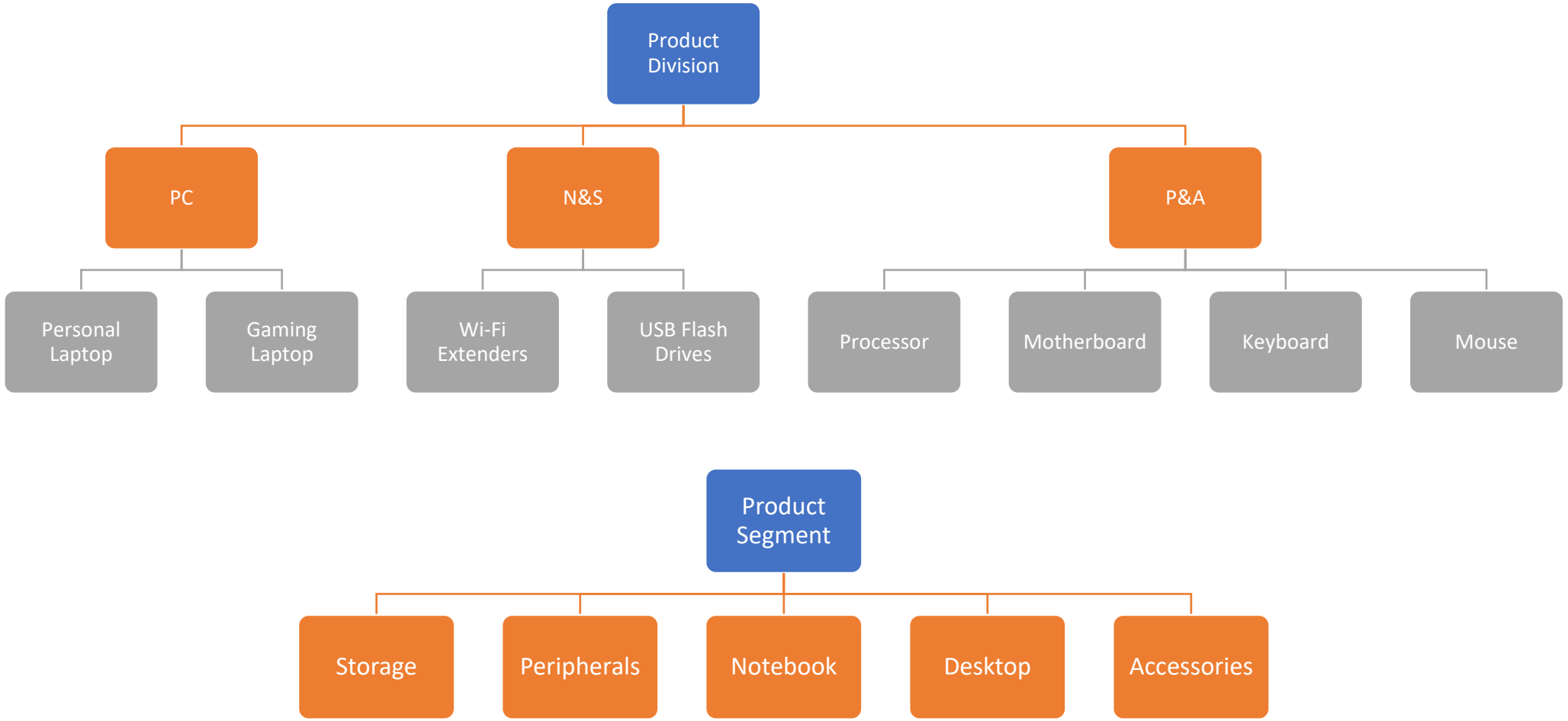
Project Overview

- This project is designed to analyze and extract valuable insights from the provided database. The database contains information about sales, products, customers, and regions for Atliq Hardware. I aim to address specific questions related to sales reports, market analysis, customer behavior, and supply chain forecasting.

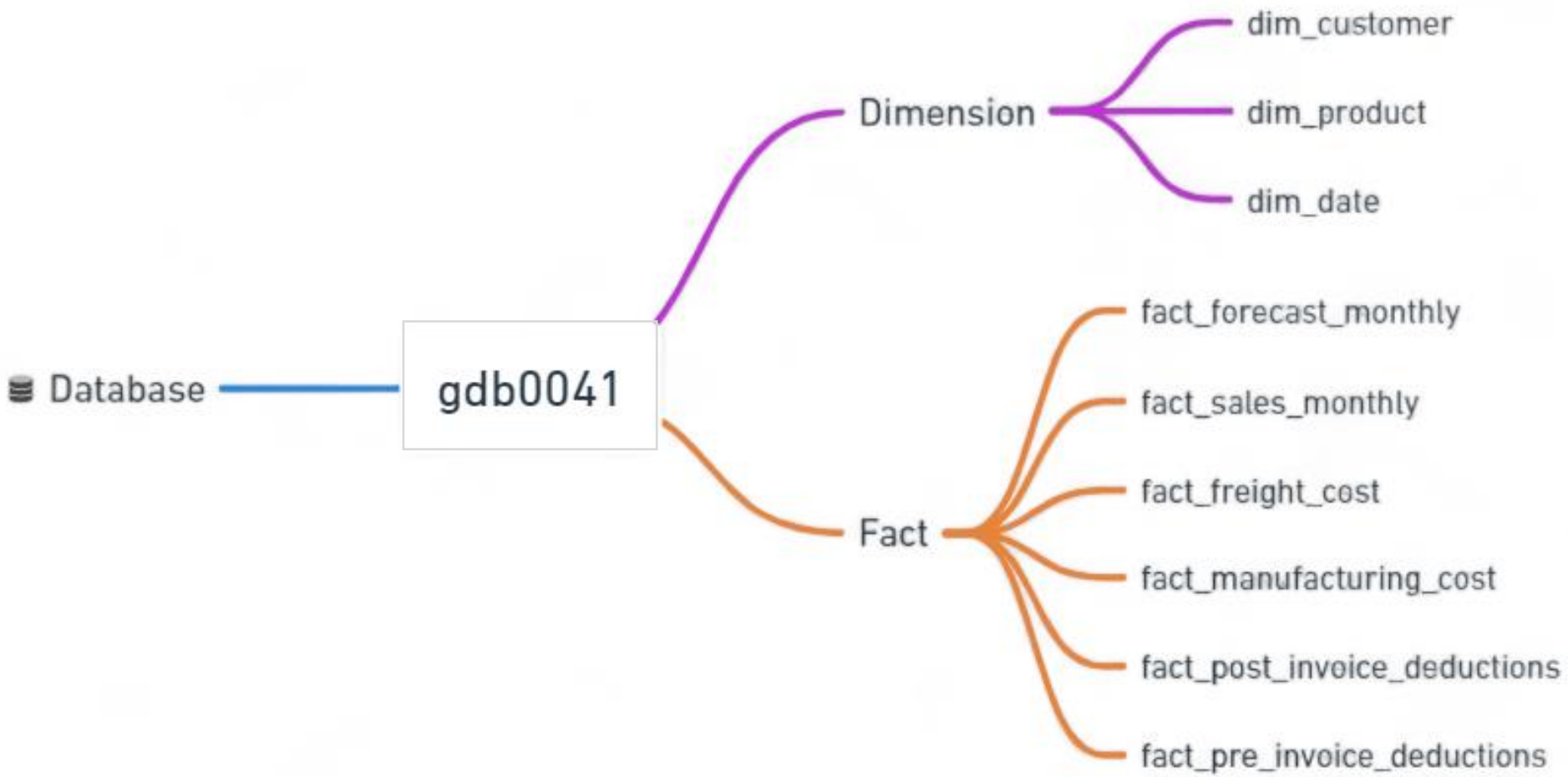
AtliQ Hardware – Business Model



AtliQ Hardware – Business Model



Data Sets


Database
gdb0041
Dimension
dim_customer
dim_product
dim_date
Fact
fact_forecast_monthly
fact_sales_monthly
fact_freight_cost
fact_manufacturing_cost
fact_post_invoice_deductions
fact_pre_invoice_deductions

Database Overview

Name	Engine	Version	Row Format	Rows	Avg Row Length	Data Length
dim_customer	InnoDB	10	Dynamic	209	78	16.0 KiB
dim_date	InnoDB	10	Dynamic	86	190	16.0 KiB
dim_product	InnoDB	10	Dynamic	397	165	64.0 KiB
fact_act_est	InnoDB	10	Dynamic	1911762	49	90.7 MiB
fact_forecast_monthly	InnoDB	10	Dynamic	1805707	53	92.6 MiB
fact_freight_cost	InnoDB	10	Dynamic	135	121	16.0 KiB
fact_gross_price	InnoDB	10	Dynamic	1182	69	80.0 KiB
fact_manufacturing_cost	InnoDB	10	Dynamic	1182	69	80.0 KiB
fact_post_invoice_deductions	InnoDB	10	Dynamic	2041650	47	91.7 MiB
fact_pre_invoice_deductions	InnoDB	10	Dynamic	1045	62	64.0 KiB
fact_sales_monthly	InnoDB	10	Dynamic	1436708	51	70.7 MiB

Tables

Name	Type	Definer	Modified	Created	Security Type
get_forecast_accuracy	PROCEDURE	root@localhost	2024-04-22 01:1...	2024-04-22 01:1...	DEFINER
get_market_badge	PROCEDURE	root@localhost	2024-04-17 01:0...	2024-04-17 01:0...	DEFINER
get_monthly_gross_reports_for_c...	PROCEDURE	root@localhost	2024-04-17 00:3...	2024-04-17 00:3...	DEFINER
get_top_n_customers_by_net_sales	PROCEDURE	root@localhost	2024-04-19 00:4...	2024-04-19 00:4...	DEFINER
get_top_n_market_by_net_sales	PROCEDURE	root@localhost	2024-04-19 00:3...	2024-04-19 00:3...	DEFINER
get_top_n_products_by_net_sales	PROCEDURE	root@localhost	2024-04-19 23:0...	2024-04-19 23:0...	DEFINER
top_n_product_per_division_by_qt...	PROCEDURE	root@localhost	2024-04-20 23:5...	2024-04-20 23:5...	DEFINER

Stored Procedures

Name	Type	Definer	Modified	Created	Security Type
get_fiscal_quarter	FUNCTION	root@localhost	2024-04-16 00:0...	2024-04-16 00:0...	DEFINER
get_fiscal_year	FUNCTION	root@localhost	2024-04-15 23:4...	2024-04-15 23:4...	DEFINER

Functions

Name
net_sales
sales_postinv_discounts
sales_preinv_discount

Views

Croma India product wise sales report for fiscal year - 2021

```
SELECT monthname(s.date) as month,
p.product,p.variant,s.sold_quantity,
round(g.gross_price,2) as gross_price,
round(s.sold_quantity * g.gross_price,2) as gross_price_total
FROM fact_sales_monthly s
join dim_product p
using (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
limit 1000000;
```

month	product	variant	sold_quantity	gross_price	gross_price_total
September	AQ Dracula HDD – 3.5 Inc...	Standard	202	19.06	3849.57
September	AQ Dracula HDD – 3.5 Inc...	Plus	162	21.46	3475.95
September	AQ Dracula HDD – 3.5 Inc...	Premium	193	21.78	4203.44
September	AQ Dracula HDD – 3.5 Inc...	Premium Plus	146	22.97	3354.04
September	AQ WereWolf NAS Interna...	Standard	149	23.70	3531.11
September	AQ WereWolf NAS Interna...	Plus	107	24.73	2646.24
September	AQ WereWolf NAS Interna...	Premium	123	23.62	2904.69
September	AQ Zion Saga	Standard	146	23.72	3463.46
September	AQ Zion Saga	Plus	236	27.10	6396.24
September	AQ Zion Saga	Premium	137	28.01	3836.81
September	AQ Mforce Gen X	Standard 3	23	19.52	449.04
September	AQ Mforce Gen X	Plus 1	82	19.92	1633.76
September	AQ Mforce Gen X	Plus 2	86	20.08	1726.59

Gross monthly total sales for Croma

```sql
SELECT monthname(s.date) as month,
round(sum(s.sold_quantity * g.gross_price),2) as gross_price_total
FROM fact_sales_monthly s
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
group by s.date
order by date asc;
```

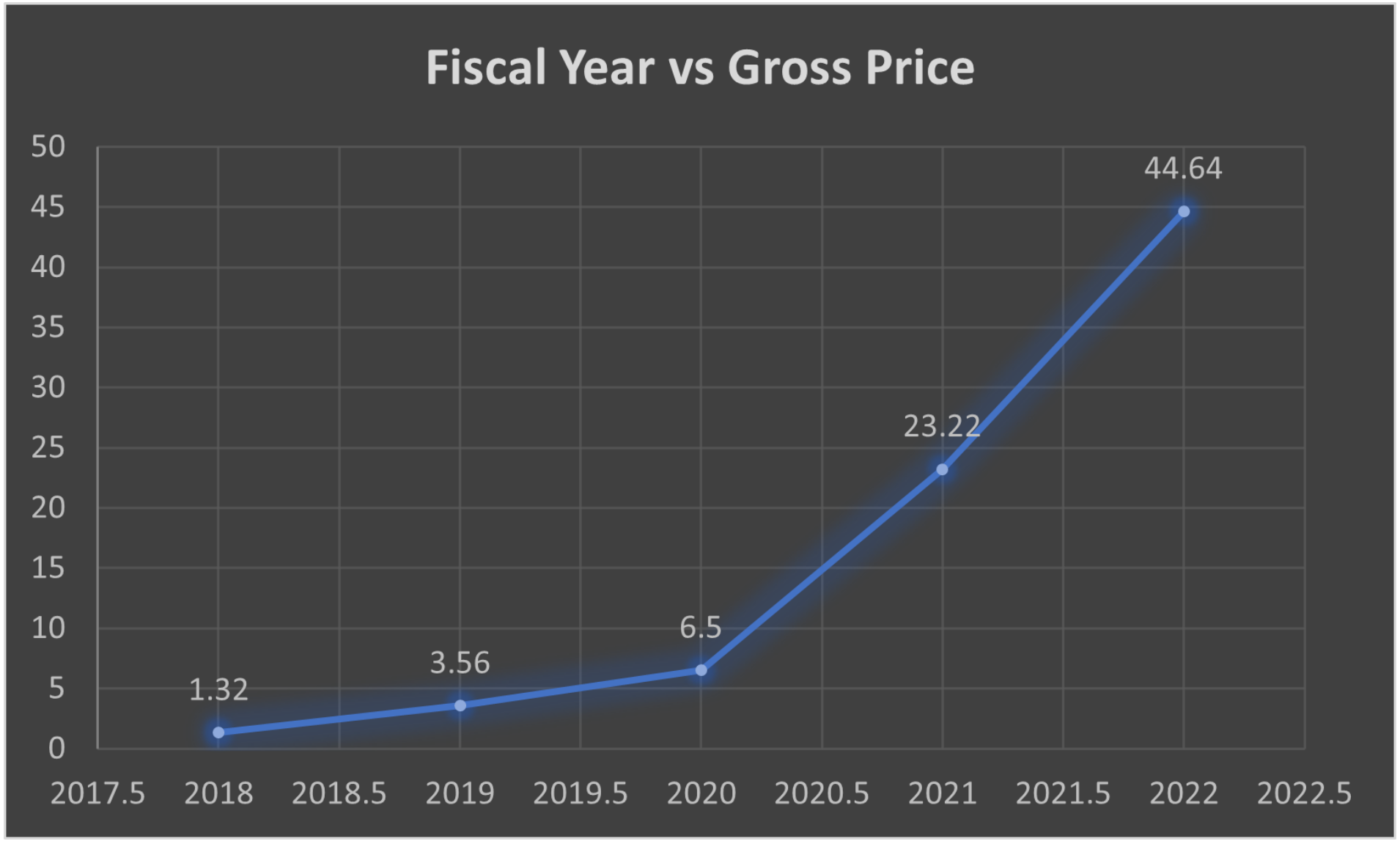
month	gross_price_total
September	122407.56
October	162687.57
December	245673.80
January	127574.74
February	144799.52
April	130643.90
May	139165.10
June	125735.38
August	125409.88
September	343337.17
October	440562.08
December	653944.75
January	359025.02
February	356607.17

Yearly gross sales report for Croma India

```sql
select g.fiscal_year,
round(sum(s.sold_quantity * g.gross_price)/1000000,2) as "gross_price_total(in mln)"
from fact_sales_monthly s
join fact_gross_price g
on s.product_code = g.product_code
and get_fiscal_year(s.date) = g.fiscal_year
where customer_code = 90002002
group by g.fiscal_year;
```

fiscal_year	gross_price_total(in mln)
2018	1.32
2019	3.56
2020	6.50
2021	23.22
2022	44.64

Yearly gross sales report for Croma India



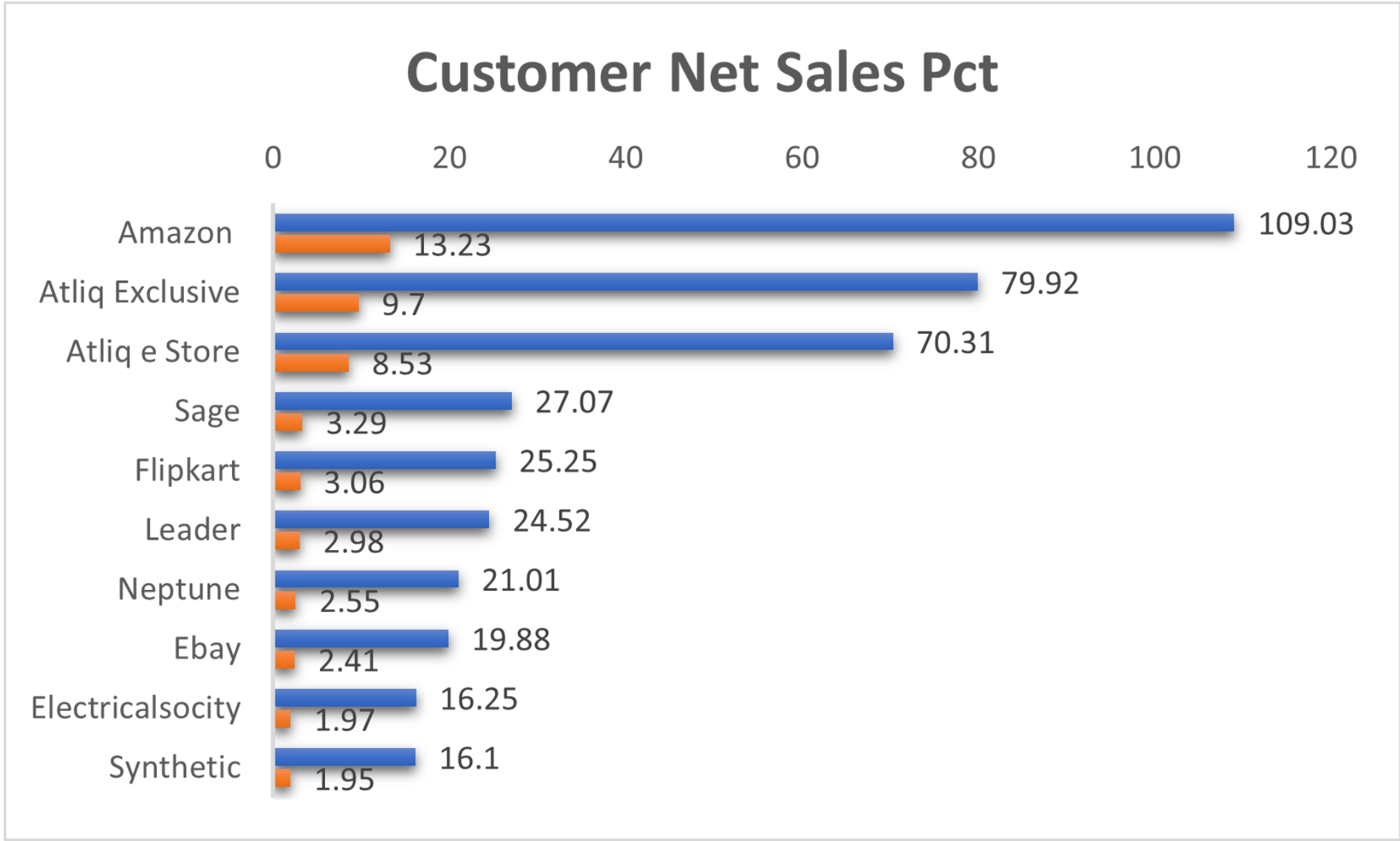
Top Market and Customers for a Financial Year 2021
Customer

```sql
SELECT c.customer,
round(sum(net_sales)/1000000,2) as net_sales_mln
FROM gdb0041.net_sales s
join dim_customer c
using (customer_code)
where fiscal_year = 2021
group by customer
order by net_sales_mln desc
limit 5;
```

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

Top Market and Customers for a Financial Year 2021

Customer



Presented by - Vijaya Mense

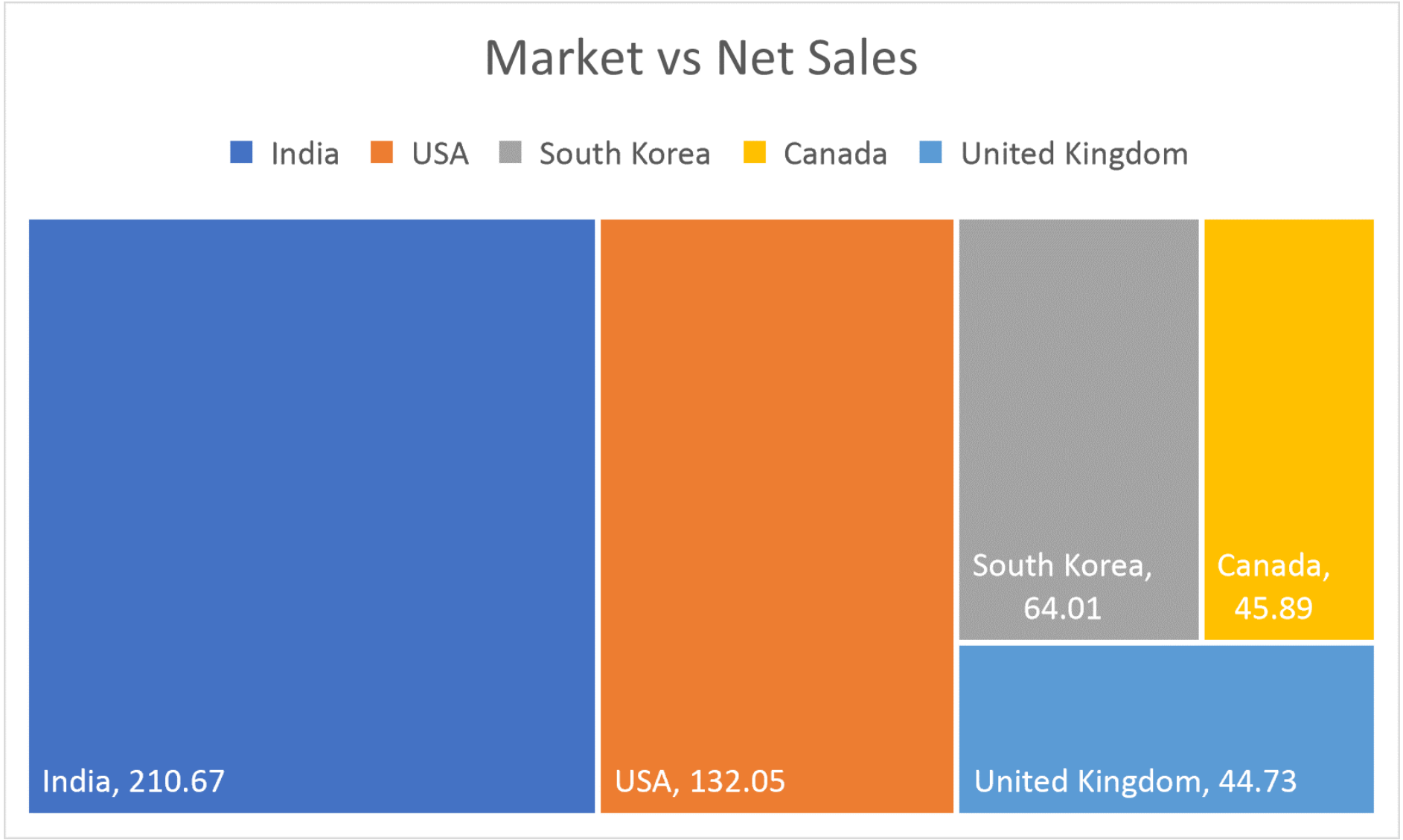
Top Market and Customers for a Financial Year 2021
Market

```
SELECT market,
round(sum(net_sales)/1000000,2) as net_sales_mln
FROM gdb0041.net_sales
where fiscal_year = 2021
group by market
order by net_sales_mln desc
limit 5;
```

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

Top Market and Customers for a Financial Year 2021

Market



Presented by - Vijaya Mense

Net sales % share by Customers

```sql
with cte as(
SELECT c.customer,
 round(sum(net_sales)/1000000,2) as net_sales_mln
 FROM gdb0041.net_sales s
 join dim_customer c
 using (customer_code)
 where s.fiscal_year = 2021
group by customer
 order by net_sales_mln desc
)
select *,
round(net_sales_mln*100/sum(net_sales_mln) over(),2) as net_sales_perc
from cte
order by net_sales_perc desc
limit 10;
```

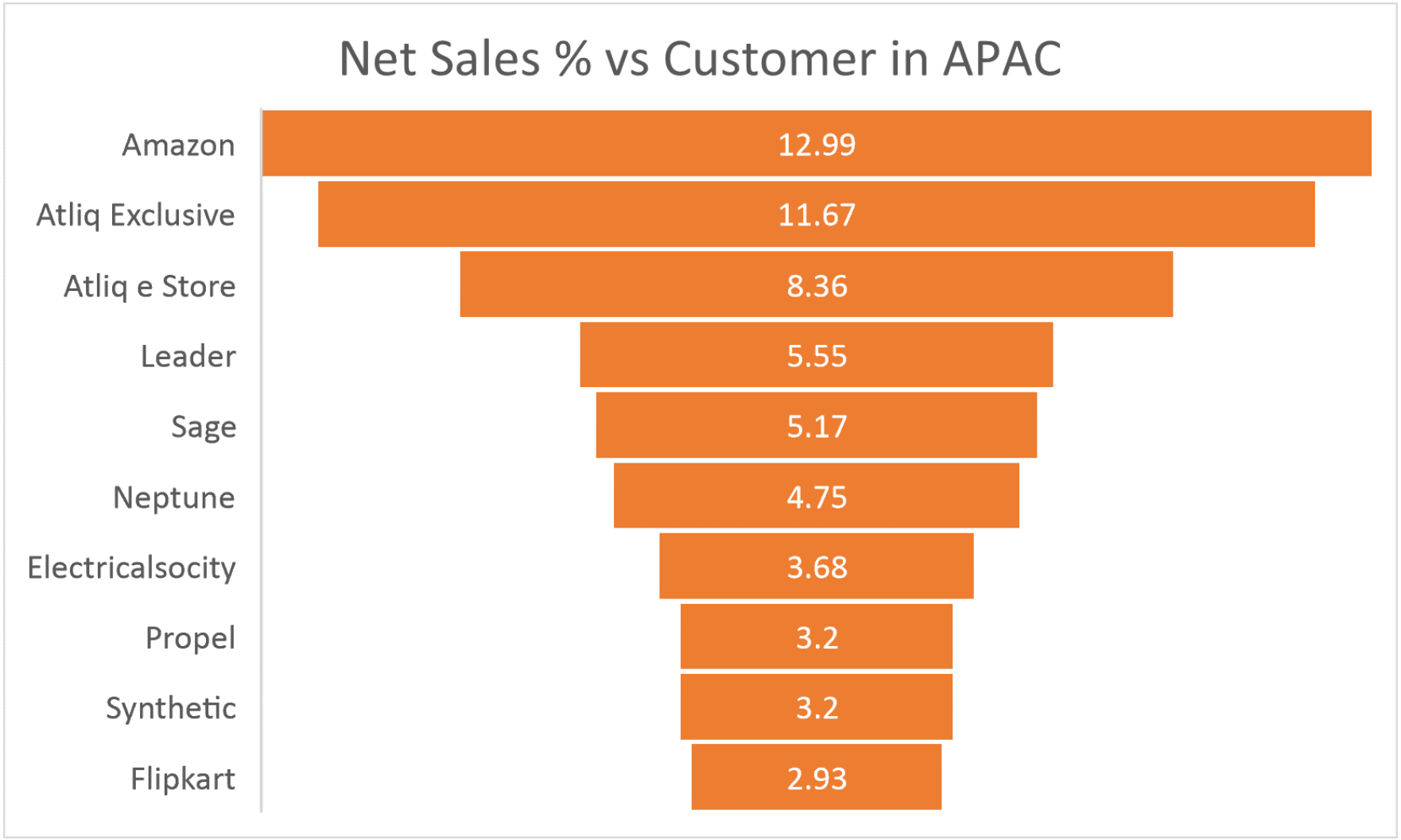
customer	net_sales_mln	net_sales_perc
Amazon	109.03	13.23
Atliq Exclusive	79.92	9.70
Atliq e Store	70.31	8.53
Sage	27.07	3.29
Flipkart	25.25	3.06
Leader	24.52	2.98
Neptune	21.01	2.55
Ebay	19.88	2.41
Electricalsocity	16.25	1.97
Synthetic	16.10	1.95

Net sales % share by Region - APAC

```sql
with cte as(select customer,
sum(net_sales) as net_sales
from net_sales s
join dim_customer c
using (customer_code)
where s.fiscal_year = 2021 and region = "APAC"
group by customer
order by net_sales desc
)
select customer,round(net_sales*100/sum(net_sales) over(),2) as net_sales_perc
from cte
limit 10;
```

customer	net_sales_perc
Amazon	12.99
Atliq Exclusive	11.67
Atliq e Store	8.36
Leader	5.55
Sage	5.17
Neptune	4.75
Electricalsocity	3.68
Propel	3.20
Synthetic	3.20
Flipkart	2.93

Net sales % share by Region - APAC



Supply Chain – Forecast Quantity

```sql
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;
```

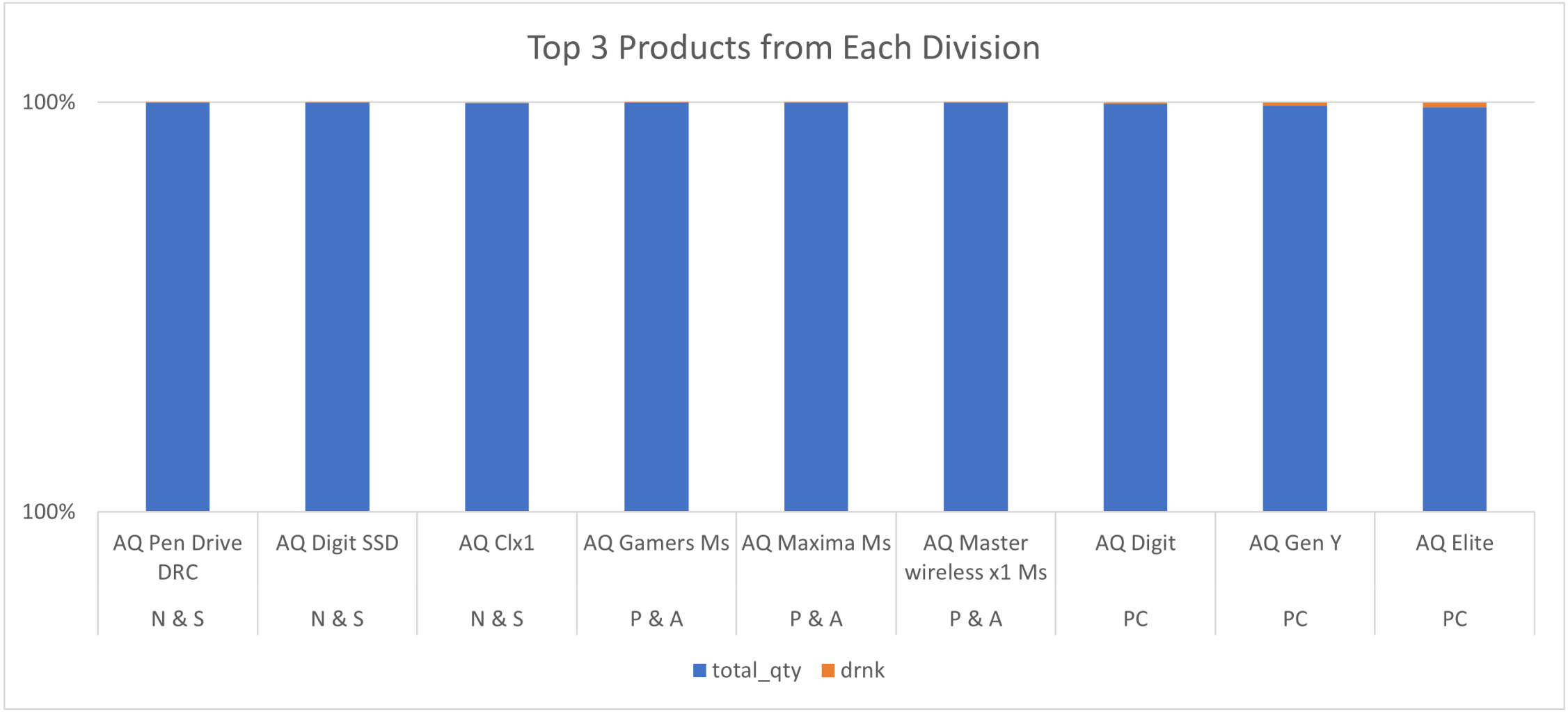
customer_code	customer_name	market	total_sold_qty	total_forecast_qty	net_error	net_error_pct	abs_error	abs_error_pct	forecast_
90013120	Coolblue	Italy	109547	133532	23920	17.9	70402	52.72	47.28
70010048	Atliq e Store	Bangladesh	119439	142010	22526	15.9	75666	53.28	46.72
90025209	Electricalsbea Stores	Columbia	13178	15428	1892	12.3	8224	53.31	46.69
90023027	Costco	Canada	236189	279962	43752	15.6	149282	53.32	46.68
90023026	Relief	Canada	228988	273492	44486	16.3	146930	53.72	46.28
90017051	Forward Stores	Portugal	86823	118067	31138	26.4	63462	53.75	46.25
70027208	Atliq e Store	Brazil	33713	47321	13282	28.1	25458	53.80	46.20
90017058	Mbit	Portugal	86860	110195	23226	21.1	59364	53.87	46.13
90023028	walmart	Canada	239081	283323	44228	15.6	153044	54.02	45.98
90023024	Sage	Canada	246397	287233	40823	14.2	155597	54.17	45.83
90013124	Amazon	Italy	110898	136116	25168	18.5	73776	54.20	45.80
90015146	Mbit	Norway	147152	210507	63320	30.1	114154	54.23	45.77
90017054	Flawless Stores	Portugal	84371	114698	30241	26.4	62397	54.40	45.60

Top 3 products from each division by total quantity sold in a given year

```sql
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, 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

Top 3 products from each division by total quantity sold in a given year



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

```
SELECT * FROM fact_sales_monthly
    WHERE
            customer_code=90002002 AND
            YEAR(DATE_ADD(date, INTERVAL 4 MONTH))=2021
    ORDER BY date asc
    LIMIT 100000;
```

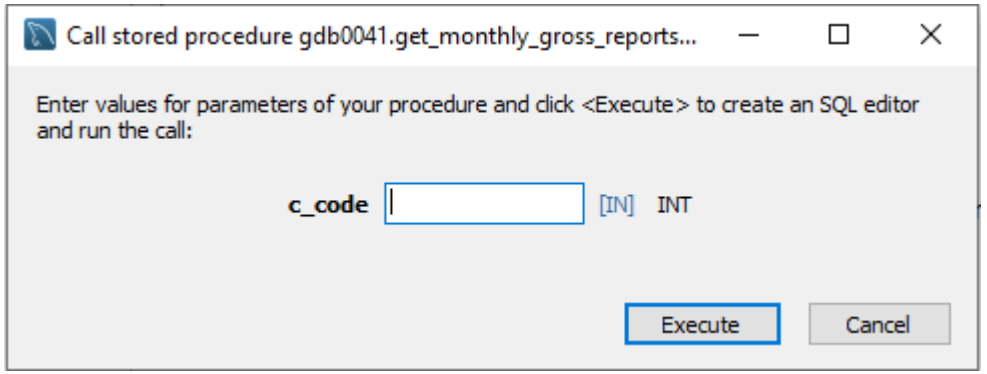
date	fiscal_year	product_code	customer_code	sold_quantity
2020-09-01	2021	A0220150203	90002002	123
2020-09-01	2021	A0320150301	90002002	146
2020-09-01	2021	A0321150302	90002002	236
2020-09-01	2021	A0321150303	90002002	137
2020-09-01	2021	A0418150103	90002002	23
2020-09-01	2021	A0418150104	90002002	82
2020-09-01	2021	A0418150105	90002002	86
2020-09-01	2021	A0418150106	90002002	48
2020-09-01	2021	A0519150201	90002002	138
2020-09-01	2021	A0519150202	90002002	72
2020-09-01	2021	A0519150203	90002002	38
2020-09-01	2021	A0519150204	90002002	149
2020-09-01	2021	A0519150205	90002002	29

Create function for fiscal_year

```
CREATE DEFINER=`root`@`localhost` 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
```

Generate monthly gross sales report for any customer using stored procedure

```
CREATE DEFINER=`root`@`localhost` PROCEDURE
`get_monthly_gross_reports_for_customer`(
 c_code INT
)
BEGIN
select
    s.date,
    sum(round(g.gross_price*sold_quantity,2)) as monthly_sales
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=c_code
group by s.date;
END
```



Thank You