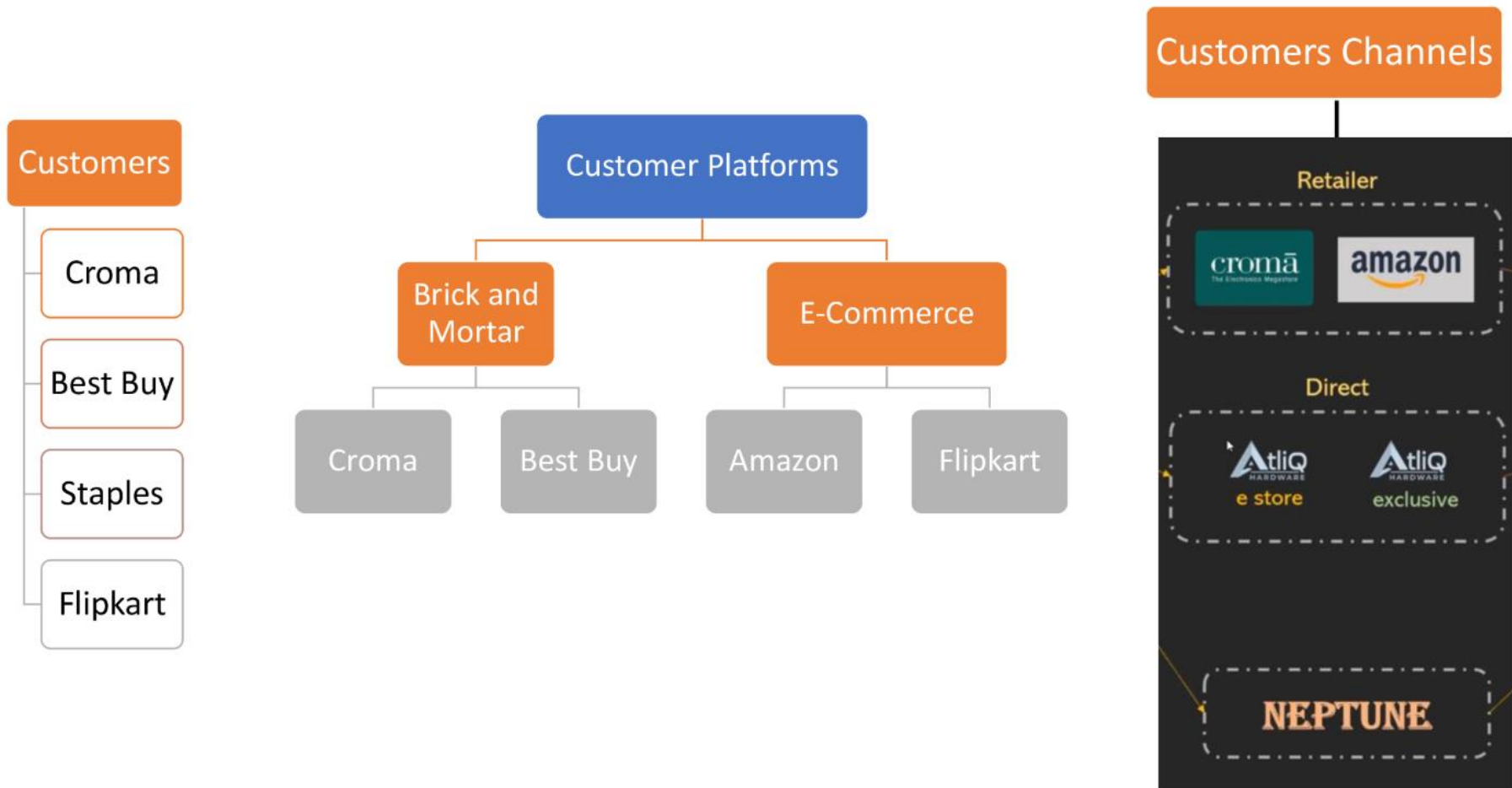# AtliQ Hardware Ad-hoc Analysis

- By Vipul Kachave

Problem Statement

- Atliq Hardware is a hardware retail business that operates in multiple markets and offers a wide range of personal computers, mice, printers, and various other peripherals.

- The business faces the challenge of analyzing large volumes of sales data across different branches and regions to gain actionable insights for decision-making. Currently, the business lacks a comprehensive data analysis solution to evaluate its financial performance and optimize its sales strategy.

- The objective of this project is to leverage SQL to analyze Atliq Hardware's sales data and generate key business insights that will help the management team make data-driven decisions.

AtliQ Hardware - Business Model

Retrieve the Customer Code for customer "Flipkart" operating in Indian Market.

```sql
SELECT *
FROM dim_customer
WHERE customer LIKE 'Flipkart' AND market = "India";
```

	customer_code	customer	platform	channel	market	sub_zone	region
▶	90002009	Flipkart	E-Commerce	Retailer	India	India	APAC
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Create a user-defined function `get_fiscal_year` to get corresponding fiscal year by passing the calendar date.

```sql
CREATE FUNCTION `get_fiscal_year`(calender_date DATE)
RETURNS INT
    DETERMINISTIC
BEGIN
    DECLARE  fiscal_year INT;
    SET fiscal_year = YEAR(DATE_ADD(calender_date, INTERVAL 4 MONTH));
RETURN fiscal_year;
END
```

Retrieve all sales transaction for customer "Flipkart" for the Fiscal year 2021.

```sql
SELECT * FROM fact_sales_monthly

WHERE customer_code = 90002009 AND get_fiscal_year(DATE) = 2021

ORDER BY DATE ASC

LIMIT 100000;
```

date	fiscal_year	product_code	customer_code	sold_quantity
2020-09-01	2021	A0118150101	90002009	58
2020-09-01	2021	A0118150102	90002009	137
2020-09-01	2021	A0118150103	90002009	247
2020-09-01	2021	A0118150104	90002009	201
2020-09-01	2021	A0219150201	90002009	146
2020-09-01	2021	A0219150202	90002009	183
2020-09-01	2021	A0220150203	90002009	237
2020-09-01	2021	A0320150301	90002009	186
2020-09-01	2021	A0321150302	90002009	46
2020-09-01	2021	A0321150303	90002009	274
2020-09-01	2021	A0418150103	90002009	166
2020-09-01	2021	A0418150104	90002009	34
2020-09-01	2021	A0418150105	90002009	146
2020-09-01	2021	A0418150106	90002009	56
2020-09-01	2021	A0519150201	90002009	87
2020-09-01	2021	A0519150202	90002009	34
2020-09-01	2021	A0519150203	90002009	205
2020-09-01	2021	A0519150204	90002009	139

Create a user-defined function `get_fiscal_quarter` to get Quarterly sales transactions for customer

```sql
CREATE DEFINER=`root`@`localhost` FUNCTION `get_fiscal_quarter`(calender_date DATE)
RETURNS char(2)
    DETERMINISTIC
BEGIN
        DECLARE m TINYINT;
        DECLARE qtr CHAR(2);
        SET m = MONTH(calender_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;
```

Using this function retrieved the sold_quantity for Flipkart for fiscal year 2021 in Quarter-4

```sql
SELECT
    date,
    sold_quantity
FROM fact_sales_monthly
    WHERE customer_code = 90002009
        AND get_fiscal_year(date) = 2021
        AND get_fiscal_quarter(date) = "Q4"
ORDER BY DATE ASC;
```

date	sold_quantity
2021-06-01	97
2021-06-01	80
2021-06-01	43
2021-06-01	228
2021-06-01	204
2021-06-01	105
2021-06-01	112
2021-06-01	142
2021-06-01	304
2021-06-01	241
2021-06-01	135
2021-06-01	180
2021-06-01	103
2021-06-01	70
2021-06-01	189
2021-06-01	33
2021-06-01	112
2021-06-01	212

JOINS

Gross Monthly sales report:
Monthly sales transaction data for Flipkart by Joining two Tables

```sql
SELECT s.date,
       p.product_code, p.product, p.variant,
       s.sold_quantity
   FROM fact_sales_monthly AS s
   JOIN dim_product AS p
       ON s.product_code = p.product_code
   WHERE customer_code = 90002009
       AND get_fiscal_year(date) = 2021
ORDER BY DATE ASC;
```

date	product_code	product	variant	sold_quantity
2020-09-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	58
2020-09-01	A0118150102	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Plus	137
2020-09-01	A0118150103	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Premium	247
2020-09-01	A0118150104	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Premium Plus	201
2020-09-01	A0219150201	AQ WereWolf NAS Internal Hard Drive HDD – 8....	Standard	146
2020-09-01	A0219150202	AQ WereWolf NAS Internal Hard Drive HDD – 8....	Plus	183
2020-09-01	A0220150203	AQ WereWolf NAS Internal Hard Drive HDD – 8....	Premium	237
2020-09-01	A0320150301	AQ Zion Saga	Standard	186
2020-09-01	A0321150302	AQ Zion Saga	Plus	46
2020-09-01	A0321150303	AQ Zion Saga	Premium	274
2020-09-01	A0418150103	AQ Mforce Gen X	Standard 3	166
2020-09-01	A0418150104	AQ Mforce Gen X	Plus 1	34
2020-09-01	A0418150105	AQ Mforce Gen X	Plus 2	146
2020-09-01	A0418150106	AQ Mforce Gen X	Plus 3	56
2020-09-01	A0519150201	AQ Mforce Gen Y	Standard 1	87
2020-09-01	A0519150202	AQ Mforce Gen Y	Standard 2	34
2020-09-01	A0519150203	AQ Mforce Gen Y	Standard 3	205
2020-09-01	A0519150204	AQ Mforce Gen Y	Plus 1	139

Gross Monthly sales report:

Retrieve Gross_price and Gross_price_total by joining another table.

```sql
SELECT s.date, p.product_code, p.product, p.variant, s.sold_quantity,
        ROUND (g.gross_price,2) AS gross_price,
        ROUND (g.gross_price*s.sold_quantity,2) AS gross_price_total
    FROM fact_sales_monthly AS s
    JOIN dim_product AS p
        ON s.product_code = p.product_code
    JOIN fact_gross_price AS g
        ON g.product_code = s.product_code AND g.fiscal_year = get_fiscal_year(s.date)
WHERE customer_code = 90002009
    AND get_fiscal_year(date) = 2021
ORDER BY DATE ASC;
```

OUTPUT

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	58	19.06	1105.32
2020-09-01	A0118150102	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Plus	137	21.46	2939.54
2020-09-01	A0118150103	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Premium	247	21.78	5379.54
2020-09-01	A0118150104	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Premium Plus	201	22.97	4617.55
2020-09-01	A0219150201	AQ WereWolf NAS Internal Hard Drive HDD – 8...	Standard	146	23.70	3460.01
2020-09-01	A0219150202	AQ WereWolf NAS Internal Hard Drive HDD – 8...	Plus	183	24.73	4525.81
2020-09-01	A0220150203	AQ WereWolf NAS Internal Hard Drive HDD – 8...	Premium	237	23.62	5596.85
2020-09-01	A0320150301	AQ Zion Saga	Standard	186	23.72	4412.35
2020-09-01	A0321150302	AQ Zion Saga	Plus	46	27.10	1246.72
2020-09-01	A0321150303	AQ Zion Saga	Premium	274	28.01	7673.62
2020-09-01	A0418150103	AQ Mforce Gen X	Standard 3	166	19.52	3240.90
2020-09-01	A0418150104	AQ Mforce Gen X	Plus 1	34	19.92	677.41
2020-09-01	A0418150105	AQ Mforce Gen X	Plus 2	146	20.08	2931.18
2020-09-01	A0418150106	AQ Mforce Gen X	Plus 3	56	19.94	1116.44
2020-09-01	A0519150201	AQ Mforce Gen Y	Standard 1	87	22.40	1948.66
2020-09-01	A0519150202	AQ Mforce Gen Y	Standard 2	34	24.93	847.61
2020-09-01	A0519150203	AQ Mforce Gen Y	Standard 3	205	26.59	5450.36
2020-09-01	A0519150204	AQ Mforce Gen Y	Plus 1	139	26.11	3629.03

Retrieve Monthly Gross_sales_amount for Flipkart

```sql
SELECT s.date,
    ROUND(SUM(g.gross_price*s.sold_quantity),2) AS gross_sales_amount
FROM fact_sales_monthly AS s
    JOIN fact_gross_price AS g
        ON g.product_code = s.product_code AND g.fiscal_year = get_fiscal_year(s.date)
    WHERE customer_code = 90002009
    GROUP BY s.date
    ORDER BY s.date ASC;
```

date	gross_sales_amount
2017-09-01	198752.68
2017-10-01	220761.88
2017-11-01	327835.33
2018-01-01	178745.69
2018-02-01	169139.72
2018-03-01	199598.66
2018-05-01	175089.40
2018-06-01	189916.13
2018-07-01	187504.11
2018-09-01	457258.22
2018-10-01	602250.17
2018-11-01	827571.63
2019-01-01	464502.16
2019-02-01	447545.87
2019-03-01	494013.14
2019-05-01	477017.01
2019-06-01	469872.08
2019-07-01	486274.55

Generate Yearly sales report for customer

```sql
SELECT get_fiscal_year(s.date) AS fiscal_year,
    ROUND(SUM(g.gross_price*s.sold_quantity),2) AS total_yearly_sales
FROM fact_sales_monthly AS s
    JOIN fact_gross_price AS g
        ON g.product_code = s.product_code AND g.fiscal_year = get_fiscal_year(s.date)
    WHERE customer_code = 90002009
    GROUP BY fiscal_year
    ORDER BY fiscal_year ASC;
```

| fiscal_year	total_yearly_sales
2018	1847343.61
2019	4726304.84
2020	12995383.21
2021	30682137.39
2022	57590480.87

STORED PROCEDURE

Create a Stored procedure to get monthly gross sales report for any customer passing any customer code into stored procedure.

```sql
CREATE DEFINER=`root`@`localhost` PROCEDURE `get_monthly_gross_sales_for_customer`(c_code INT)
BEGIN
    SELECT s.date,
    SUM(ROUND(s.sold_quantity*g.gross_price,2)) AS monthly_sales
    FROM fact_sales_monthly s
    JOIN fact_gross_price g
        ON g.fiscal_year = get_fiscal_year(s.date)
            AND g.product_code = s.product_code
    WHERE customer_code = c_code
    GROUP BY s.date;
END
```

```sql
call gdb0041.get_monthly_gross_sales_for_customer(90002009);
```
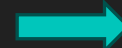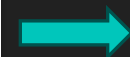
date	monthly_sales
2021-12-01	19849946.98
2021-11-01	19289758.70
2021-10-01	14634931.44
2021-08-01	2316472.95
2021-07-01	2380093.58
2021-06-01	2406323.75
2021-04-01	2276252.96
2021-03-01	2300637.79
2021-02-01	2337095.49
2020-12-01	4277979.70
2020-11-01	3861012.58
2020-10-01	3145927.24

Sometimes customer has two or more codes, so create stored procedure to get monthly gross sales report for customer passing customer codes into list fromat in stored procedure.

```sql
CREATE DEFINER=`root`@`localhost` PROCEDURE `get_monthly_gross_sales_for_amazon`(in_customer_codes TEXT)
BEGIN
    SELECT s.date,
           SUM(ROUND(s.sold_quantity*g.gross_price,2)) AS monthly_sales
    FROM fact_sales_monthly s
    JOIN fact_gross_price g
        ON g.fiscal_year=get_fiscal_year(s.date)
        AND g.product_code=s.product_code
    WHERE FIND_IN_SET(s.customer_code, in_customer_codes) > 0
    GROUP BY s.date
    ORDER BY s.date DESC;
END
```

date	monthly_sales
2017-09-01	198752.68
2017-10-01	220761.87
2017-11-01	327835.33
2018-01-01	178745.67
2018-02-01	169139.74
2018-03-01	199598.67
2018-05-01	175089.44
2018-06-01	189916.12
2018-07-01	187504.11
2018-09-01	457258.31
2018-10-01	602250.22
2018-11-01	827571.65
2019-01-01	464502.17
2019-02-01	447545.87
2019-03-01	494013.15
2019-05-01	477016.98
2019-06-01	469872.06
2019-07-01	486274.56

```sql
call gdb0041.get_monthly_gross_sales_for_amazon('90002016, 90002008');
```

Create a Join between two tables to get Market wise sold quantities.

```sql
SELECT c.market, SUM(sold_quantity) AS total_qty
FROM fact_sales_monthly s
    JOIN dim_customer c
        ON s.customer_code = c.customer_code
    WHERE get_fiscal_year(s.date) = 2021
    GROUP BY c.market;
```

market	total_qty
India	13751429
Indonesia	1434929
Japan	529487
Pakistan	454393
Philiphines	2422641
South Korea	3947794
Australia	1782354
Newzealand	835190
Bangladesh	575892
France	2047367
Germany	1243839
Italy	1150038

Create a Stored procedure to categorized market in three level badge based on their sold_quantity.

```sql
CREATE DEFINER=`root`@`localhost` PROCEDURE `get_market_badge`(
    IN in_market VARCHAR(50),
    IN in_fiscal_year YEAR,
    OUT out_badge VARCHAR(10)
)
BEGIN
    DECLARE qty INT DEFAULT 0;
        # set Default market as "India"
    IF in_market ="" THEN
    SET in_market = "India";
    END IF;
        # Retrieve total sold quantity for a given market in a given year
    SELECT
    sum(sold_quantity) INTO qty
    FROM fact_sales_monthly s
    JOIN dim_customer d
    ON s.customer_code = d.customer_code
    WHERE get_fiscal_year(s.date)=in_fiscal_year
        AND d.market = in_market
    GROUP BY d.market;
        # Determine Gold vs Silver status
    IF qty > 5000000 THEN
     SET out_badge = "Gold";
    ELSE
     SET out_badge = "Silver";
    END IF;
END
```

	@out_badge
▶	Gold

```sql
set @out_badge = '0';
call gdb0041.get_market_badge('India', 2020, @out_badge);
select @out_badge;
```

market	total_qty
India	5381959
Indonesia	576133
Japan	202306
Pakistan	427543
Philiphines	1160524
South Korea	1970860
Australia	917945
Newzealand	275390
Bangladesh	219328
France	759101
Germany	537802
Italy	555442

P&L report:
Retrieve pre invoice discount percent by joining two tables

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

OUTPUT

date	customer_code	product_code	product	variant	sold_quantity	gross_price_per_item	gross_price_total	pre_invoice_discount_pct
2020-09-01	70002017	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	248	19.0573	4726.21	0.0703
2020-09-01	70002018	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	240	19.0573	4573.75	0.2061
2020-09-01	70003181	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	31	19.0573	590.78	0.0974
2020-09-01	70003182	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	37	19.0573	705.12	0.2065
2020-09-01	70004069	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	7	19.0573	133.40	0.1068
2020-09-01	70004070	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	12	19.0573	228.69	0.2612
2020-09-01	70005163	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	17	19.0573	323.97	0.2471
2020-09-01	70006157	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	60	19.0573	1143.44	0.0858
2020-09-01	70006158	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	34	19.0573	647.95	0.2450
2020-09-01	70007198	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	24	19.0573	457.38	0.0736
2020-09-01	70007199	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	88	19.0573	1677.04	0.2105
2020-09-01	70008169	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	49	19.0573	933.81	0.0793
2020-09-01	70008170	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	60	19.0573	1143.44	0.1817
2020-09-01	70009133	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	7	19.0573	133.40	0.0961
2020-09-01	70009134	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	21	19.0573	400.20	0.2899
2020-09-01	70010047	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	37	19.0573	705.12	0.0552
2020-09-01	70011193	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	70	19.0573	1334.01	0.1056
2020-09-01	70011194	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	18	19.0573	343.03	0.1859
2020-09-01	70012042	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	28	19.0573	533.60	0.0936
2020-09-01	70012043	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	29	19.0573	552.66	0.2287
2020-09-01	70013125	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	21	19.0573	400.20	0.1041
2020-09-01	70013126	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	14	19.0573	266.80	0.2873
2020-09-01	70014142	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	6	19.0573	114.34	0.0962
2020-09-01	70014143	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	21	19.0573	400.20	0.2674
2020-09-01	70015151	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	26	19.0573	495.49	0.0679

Create view "sales_pre_invoice_discount".

```sql
CREATE VIEW `sales_pre_invoice_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
```

Using view sales_pre_invoice_discount generate the net_invoice_sales

```sql
SELECT *,(gross_price_total-pre_invoice_discount_pct*gross_price_total) AS
net_invoice_sales
FROM sales_pre_invoice_discount
```

date	product_code	customer_code	market	product	variant	sold_quantity	gross_price_per_item	gross_price_total	pre_invoice_discount_pct	net_invoice_sales
2017-09-01	A011815O101	70002017	India	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	51	15.3952	785.16	0.0824	720.462816
2017-09-01	A011815O101	70002018	India	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	77	15.3952	1185.43	0.2956	835.016892
2017-09-01	A011815O101	70003181	Indonesia	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	17	15.3952	261.72	0.0536	247.691808
2017-09-01	A011815O101	70003182	Indonesia	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	6	15.3952	92.37	0.2378	70.404414
2017-09-01	A011815O101	70006157	Philiphines	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	5	15.3952	76.98	0.1057	68.843214
2017-09-01	A011815O101	70006158	Philiphines	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	7	15.3952	107.77	0.1875	87.563125
2017-09-01	A011815O101	70007198	South Korea	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	29	15.3952	446.46	0.0700	415.207800
2017-09-01	A011815O101	70007199	South Korea	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	34	15.3952	523.44	0.2551	389.910456
2017-09-01	A011815O101	70008169	Australia	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	22	15.3952	338.69	0.0953	306.412843
2017-09-01	A011815O101	70008170	Australia	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	5	15.3952	76.98	0.1896	62.384592
2017-09-01	A011815O101	70011193	France	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	10	15.3952	153.95	0.0521	145.929205
2017-09-01	A011815O101	70011194	France	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	4	15.3952	61.58	0.2046	48.980732
2017-09-01	A011815O101	70012042	Germany	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	0	15.3952	0.00	0.0984	0.000000
2017-09-01	A011815O101	70012043	Germany	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	0	15.3952	0.00	0.2620	0.000000
2017-09-01	A011815O101	70013125	Italy	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	1	15.3952	15.40	0.0587	14.496020
2017-09-01	A011815O101	70013126	Italy	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	1	15.3952	15.40	0.2501	11.548460
2017-09-01	A011815O101	70016178	Poland	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	1	15.3952	15.40	0.1937	12.417020
2017-09-01	A011815O101	70022085	USA	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	20	15.3952	307.90	0.2025	245.550250
2017-09-01	A011815O101	70023031	Canada	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	4	15.3952	61.58	0.1008	55.372736
2017-09-01	A011815O101	80001019	China	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	10	15.3952	153.95	0.2815	110.613075
2017-09-01	A011815O101	80006154	Philiphines	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	10	15.3952	153.95	0.2728	111.952440
2017-09-01	A011815O101	80006155	Philiphines	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	28	15.3952	431.07	0.2797	310.499721
2017-09-01	A011815O101	80007195	South Korea	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	80	15.3952	1231.62	0.2990	863.365620
2017-09-01	A011815O101	90001021	China	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	1	15.3952	15.40	0.2798	11.091080

Create view "sales_post_invoice_discount".

```sql
CREATE VIEW `sales_post_invoice_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;
```

Using view sales_post_invoice_discount generate the Net sales

```sql
SELECT *, ROUND(net_invoice_sales*(1-post_invoice_discount_pct),2) AS net_sales
FROM sales_post_invoice_discount;
```

date	fiscal_year	customer_code	market	product_code	product	variant	sold_quant	gross_price	pre_invoice_discount_pct	net_invoice_sales	post_invoice_discount_pct	net_sales
2017-09-01	2018	90027207	Brazil	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s ...	Standard	4	61.58	0.2803	44.319126	0.3905	27.01
2017-11-01	2018	90027207	Brazil	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s ...	Standard	16	246.32	0.2803	177.276504	0.4139	103.90
2017-12-01	2018	90027207	Brazil	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s ...	Standard	4	61.58	0.2803	44.319126	0.3295	29.72
2018-01-01	2018	90027207	Brazil	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s ...	Standard	6	92.37	0.2803	66.478689	0.3244	44.91
2018-03-01	2018	90027207	Brazil	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s ...	Standard	9	138.56	0.2803	99.721632	0.3766	62.17
2018-04-01	2018	90027207	Brazil	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s ...	Standard	6	92.37	0.2803	66.478689	0.3615	42.45
2018-05-01	2018	90027207	Brazil	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s ...	Standard	7	107.77	0.2803	77.562069	0.3173	52.95
2018-07-01	2018	90027207	Brazil	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s ...	Standard	10	153.95	0.2803	110.797815	0.3501	72.01
2018-08-01	2018	90027207	Brazil	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s ...	Standard	6	92.37	0.2803	66.478689	0.3740	41.62
2017-09-01	2018	90023030	Canada	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s ...	Standard	4	61.58	0.2117	48.543514	0.2863	34.65
2017-10-01	2018	90023030	Canada	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s ...	Standard	2	30.79	0.2117	24.271757	0.2851	17.35
2017-12-01	2018	90023030	Canada	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s ...	Standard	3	46.19	0.2117	36.411577	0.2882	25.92
2018-01-01	2018	90023030	Canada	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s ...	Standard	5	76.98	0.2117	60.683334	0.3334	40.45
2018-02-01	2018	90023030	Canada	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s ...	Standard	1	15.40	0.2117	12.139820	0.3296	8.14
2018-04-01	2018	90023030	Canada	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s ...	Standard	1	15.40	0.2117	12.139820	0.2901	8.62
2018-05-01	2018	90023030	Canada	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s ...	Standard	5	76.98	0.2117	60.683334	0.3233	41.06
2018-06-01	2018	90023030	Canada	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s ...	Standard	1	15.40	0.2117	12.139820	0.3095	8.38
2018-08-01	2018	90023030	Canada	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s ...	Standard	1	15.40	0.2117	12.139820	0.3209	8.24
2017-09-01	2018	90023029	Canada	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s ...	Standard	2	30.79	0.2171	24.105491	0.3051	16.75
2017-10-01	2018	90023029	Canada	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s ...	Standard	4	61.58	0.2171	48.210982	0.3053	33.49
2017-11-01	2018	90023029	Canada	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s ...	Standard	3	46.19	0.2171	36.162151	0.3608	23.11
2018-01-01	2018	90023029	Canada	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s ...	Standard	1	15.40	0.2171	12.056660	0.2672	8.84
2018-02-01	2018	90023029	Canada	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s ...	Standard	4	61.58	0.2171	48.210982	0.2803	34.70
2018-03-01	2018	90023029	Canada	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s ...	Standard	2	30.79	0.2171	24.105491	0.2958	16.98
2018-05-01	2018	90023029	Canada	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s ...	Standard	1	15.40	0.2171	12.056660	0.3076	8.35

Retrieve Top 5 Markets by Net sales in Fiscal year 2021

```sql
SELECT market, ROUND(SUM(net_sales)/1000000, 2) AS net_sales_mln
    FROM 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 USA	132.05
South Korea	64.01
Canada	45.89
United Kingdom	44.73

Create stored procedure to get Top n Customers by Net sales for fiscal year

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

customer	net_sales_mln
Sage	22.85
Leader	22.38
Atliq Exclusive	6.85
Atliq e Store	6.06
Amazon	5.87

Top 5 Customers in South korea

```sql
call gdb0041.get_top_n_customers_by_net_sales('South Korea', 2021, 5);
```

Create stored procedure to get Top n Markets by Net sales for fiscal year

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

market	net_sales_mln
India	37.77
USA	16.16
South Korea	15.69
Philiphines	6.95
Canada	5.85
France	5.44
Australia	4.74
Italy	3.90
Germany	3.49
Indonesia	3.09

Top 10 Markets in FY 2019

```sql
call gdb0041.get_top_n_markets_by_net_sales(2019, 10);
```

Create stored procedure to get Top n Products by Net sales for fiscal year

```sql
CREATE DEFINER=`root`@`localhost` 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 net_sales
    WHERE fiscal_year=in_fiscal_year
    GROUP BY product
    ORDER BY net_sales_mln DESC
    LIMIT in_top_n;
END
```

product	net_sales_mln
AQ BZ Allin1 Gen 2	84.63
AQ HOME Allin1 Gen 2	78.84
AQ Smash 2	73.55
AQ Smash 1	67.98
AQ Electron 3 3600 Desktop Processor	65.65
AQ Gamer 2	65.00
AQ BZ Allin1	59.20

Top 7 Products in FY 2022

```sql
call gdb0041.get_top_n_products_by_net_sales(2022, 7);
```

COMMON TABLE EXPRESSION (CTE's)

Generate customer-wise Net sales % Distribution by creating CTE.

```sql
WITH net_sales_pct 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 *,
        ROUND((net_sales_mln*100)/SUM(net_sales_mln) OVER(),2) AS net_sales_pct
    FROM net_sales_pct
    ORDER BY net_sales_mln DESC;
```

customer	net_sales_mln	net_sales_pct
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		2.98
Neptune	21.01	2.55
Ebay	19.88	2.41
Electricalsocity	16.25	1.97
Synthetic	16.10	1.95
Electricalslytical	15.64	1.90
Acclaimed Sto...	14.32	1.74
Propel	14.14	1.72
Novus	12.91	1.57
Expression	12.90	1.57
Reliance Digital	12.75	1.55
walmart	12.63	1.53
Costco	12.19	1.48
Staples	11.49	1.39
Girias	11.30	1.37
Vijay Sales	11.27	1.37
Path	11.02	1.34
Lotus	10.53	1.28
Ezone	10.30	1.25
Control	10.10	1.23

Generate customer-wise % share per Region by creating CTE.

```sql
WITH cte1 AS (
    SELECT c.customer,
           c.region,
           ROUND(SUM(net_sales)/1000000,2) 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, c.region
)
SELECT *,
       ROUND((net_sales_mln*100)/SUM(net_sales_mln) OVER (PARTITION BY region), 2) AS share_region_pct
    FROM cte1
    ORDER BY region, share_region_pct DESC;
```

customer	region	net_sales_mln	share_region_pct
Amazon	APAC	57.41	12.99
Atliq Exclusive	APAC	51.58	11.67
Atliq e Store	APAC	36.97	8.36
Leader	APAC	24.52	5.55
Sage	APAC	22.85	5.17
Neptune	APAC	21.01	4.75
Electricalsocity	APAC	16.25	3.68
Propel	APAC	14.14	3.20
Synthetic	APAC	14.14	3.20
Flipkart	APAC	12.96	2.93
Expression	APAC	12.90	2.92
Novus	APAC	12.91	2.92
Girias	APAC	11.30	2.56
Vijay Sales	APAC	11.27	2.55
Ebay	APAC	11.14	2.52
Electricalslytical	APAC	11.08	2.51
Reliance Digital	APAC	11.10	2.51
Lotus	APAC	10.53	2.38
Ezone	APAC	10.30	2.33
Viveks	APAC	10.09	2.28
Croma	APAC	9.88	2.24
Zone	APAC	6.91	1.56
Acclaimed Sto...	APAC	5.79	1.31
Taobao	APAC	4.31	0.98
Digimarket	APAC	3.97	0.90

WINDOW FUNCTIONS

Rank the Products Division-wise based on sold quantity

```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

Rank the Markets Region-wise based on gross sales

```sql
WITH cte1 AS (
    SELECT gs.market,
           c.region,
           ROUND(SUM(gs.gross_price_total)/1000000,2) AS gross_sales_mln
    FROM gross_sales AS gs
    JOIN dim_customer AS c
        ON gs.customer_code = c.customer_code
    WHERE fiscal_year=2021
    GROUP BY gs.market,c.region
    ORDER BY c.region),

cte2 AS (
    SELECT *,
           DENSE_RANK() OVER(PARTITION BY region ORDER BY gross_sales_mln DESC) AS rnk
    FROM cte1
    )

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

market	region	gross_sales_mln	rnk
India	APAC	1433.53	1
South Korea	APAC	439.60	2
United Kingdom	EU	235.58	1
France	EU	200.61	2
Brazil	LATAM	9.48	1
Mexico	LATAM	8.08	2
USA	NA	811.37	1
Canada	NA	272.38	2

Visualize the Insights using PowerBI
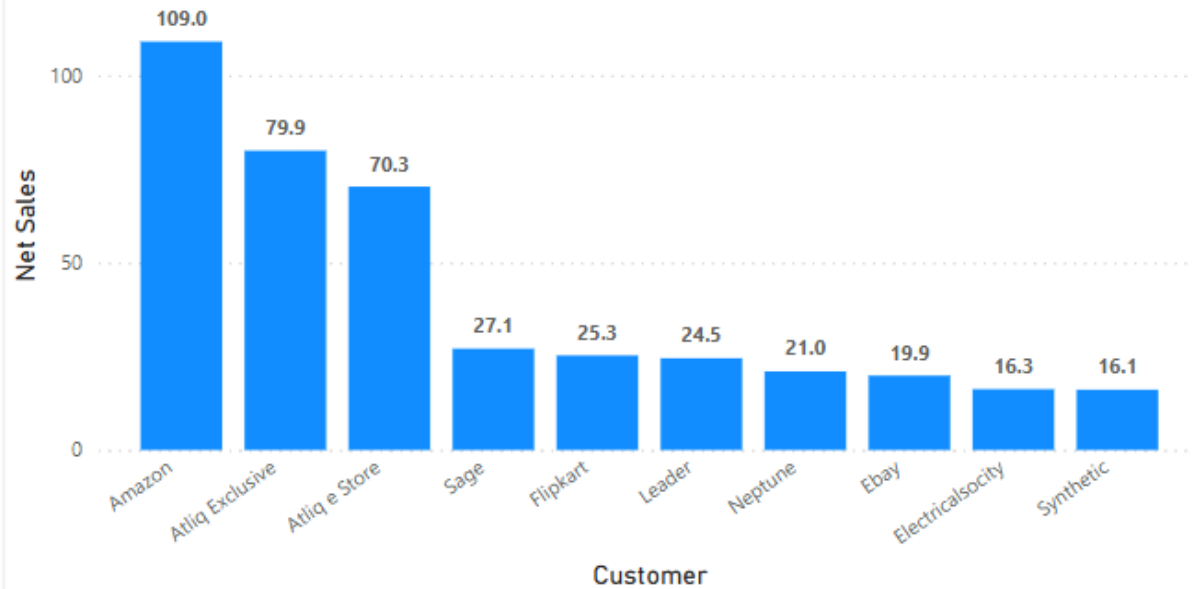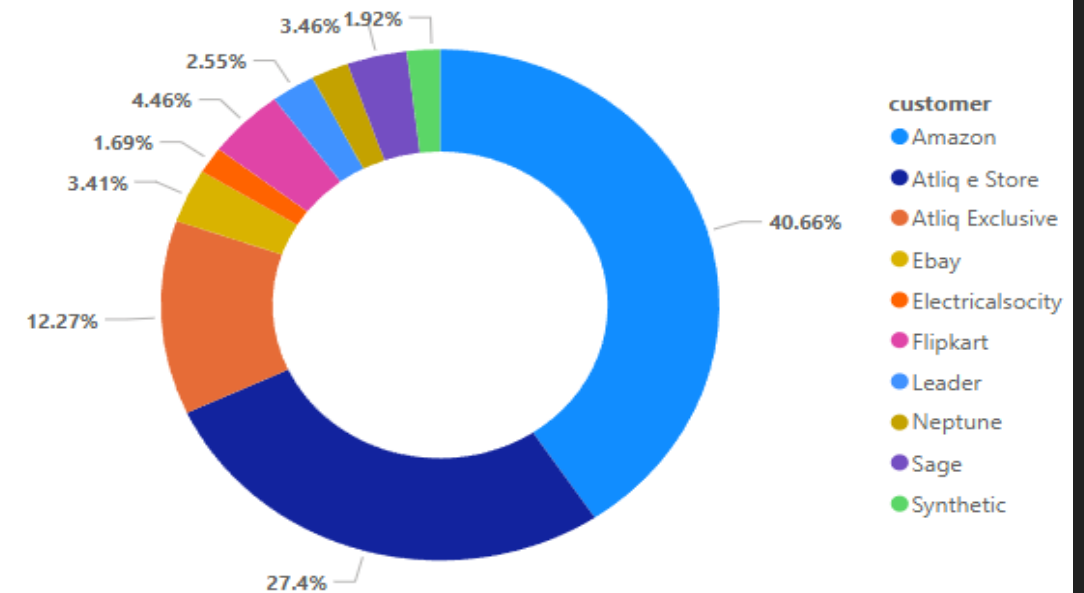
INSIGHTS

○ India leads in Gross sales, with a total sales of 1.4 Billion.

○ AQ Pen drive DRC has the highest quantity sold, reaching 2.03 Millions units.

○ Amazon has the highest net sales in fiscal year 2021 which is 109.3 million.

○ India earned Gold badge in 2021 having 5.3 Million quantity sold.