SQL PROJECT
Sales analytics

Objective

- To generate report of individual product sales (aggregated on monthly basis at product code level) for croma india customer for FY 2021 to track sales and further product analysis can be performed on it.

- Creating net sales report

tables

```
1· SELECT * FROM
2      gdb0041.dim_customer;
```

customer_code	customer	platform	channel	market	sub_zone	region
70002017	Atliq Exclusive	Brick & Mortar	Direct	India	India	APAC
70002018	Atliq e Store	E-Commerce	Direct	India	India	APAC
70003181	Atliq Exclusive	Brick & Mortar	Direct	Indonesia	ROA	APAC
70003182	Atliq e Store	E-Commerce	Direct	Indonesia	ROA	APAC
70004069	Atliq Exclusive	Brick & Mortar	Direct	Japan	ROA	APAC

```
1· SELECT * FROM gdb0041.dim_product;
```

product_code	division	segment	category	product	variant
A1118150201	P & A	Peripherals	Processors	AQ 5000 Series Electron 8 5900X Desktop Proce...	Standard
A1118150202	P & A	Peripherals	Processors	AQ 5000 Series Electron 8 5900X Desktop Proce...	Plus
A1119150203	P & A	Peripherals	Processors	AQ 5000 Series Electron 8 5900X Desktop Proce...	Premium
A1018150101	P & A	Peripherals	Processors	AQ 5000 Series Electron 9 5900X Desktop Proce...	Standard
A1018150102	P & A	Peripherals	Processors	AQ 5000 Series Electron 9 5900X Desktop Proce...	Plus
A1018150103	P & A	Peripherals	Processors	AQ 5000 Series Electron 9 5900X Desktop Proce...	Premium

```
1· SELECT * FROM
2   gdb0041.fact_sales_monthly;
```

date	fiscal_year	product_code	customer_code	sold_quantity
2017-09-01	2018	A0118150101	70002017	51
2017-09-01	2018	A0118150101	70002018	77
2017-09-01	2018	A0118150101	70003181	17
2017-09-01	2018	A0118150101	70003182	6
2017-09-01	2018	A0118150101	70006157	5

```
1· SELECT * FROM
2   gdb0041.fact_gross_price;
```

product_code	fiscal_year	gross_price
A0118150101	2018	15.3952
A0118150101	2019	14.4392
A0118150101	2020	16.2323
A0118150101	2021	19.0573
A0118150102	2018	19.5875

Customer code for croma india

```
6 •  SELECT
7        *
8    FROM
9        dim_customer
10   WHERE
11       customer LIKE '%croma%'
12           AND market = 'India'
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

customer_code	customer	platform	channel	market	sub_zone	region
90002002	Croma	Brick & Mortar	Retailer	India	India	APAC
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Creating function for date

dim_customer | fact_sales_monthly

Limit to 1000 rows

```
3 •  SELECT * FROM fact_sales_monthly
4        WHERE
5                customer_code=90002002 AND
6                get_fiscal_year(date)=2021
7        ORDER BY date asc
8        LIMIT 100000;
9
```

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

date	fiscal_year	product_code	customer_code	sold_quantity
2020-09-01	2021	A0118150101	90002002	202
2020-09-01	2021	A0118150102	90002002	162
2020-09-01	2021	A0118150103	90002002	193
2020-09-01	2021	A0118150104	90002002	146
2020-09-01	2021	A0219150201	90002002	149
2020-09-01	2021	A0219150202	90002002	107
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

Name: get_fiscal_year

The name of the routine is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

DDL:

```
1 •  CREATE DEFINER=`root`@`localhost` FUNCTION `get_fiscal_year`(calendar_date DATE) RETURNS int
2        DETERMINISTIC
3    BEGIN
4            DECLARE fiscal_year INT;
5            SET fiscal_year = YEAR(DATE_ADD(calendar_date, INTERVAL 4 MONTH));
6            RETURN fiscal_year;
7        END
```

Gross sales report –monthly product transactions

```sql
SELECT
        s.date,
        s.product_code,
        p.product,
        p.variant,
        s.sold_quantity,
        g.gross_price,
        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 fact_gross_price g
        ON g.fiscal_year=get_fiscal_year(s.date)
       AND g.product_code=s.product_code
    WHERE
        customer_code=90002002 AND
        get_fiscal_year(s.date)=2021
    LIMIT 1000000;
```

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	3849.57
2020-09-01	A0118150102	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Plus	162	21.4565	3475.95
2020-09-01	A0118150103	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Premium	193	21.7795	4203.44
2020-09-01	A0118150104	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Premium Plus	146	22.9729	3354.04
2020-09-01	A0219150201	AQ WereWolf NAS Internal Hard Drive HDD – 8....	Standard	149	23.6987	3531.11
2020-09-01	A0219150202	AQ WereWolf NAS Internal Hard Drive HDD – 8....	Plus	107	24.7312	2646.24
2020-09-01	A0220150203	AQ WereWolf NAS Internal Hard Drive HDD – 8....	Premium	123	23.6154	2904.69
2020-09-01	A0320150301	AQ Zion Saga	Standard	146	23.7223	3463.46
2020-09-01	A0321150302	AQ Zion Saga	Plus	236	27.1027	6396.24
2020-09-01	A0321150303	AQ Zion Saga	Premium	137	28.0059	3836.81
2020-09-01	A0418150103	AQ Mforce Gen X	Standard 3	23	19.5235	449.04
2020-09-01	A0418150104	AQ Mforce Gen X	Plus 1	82	19.9239	1633.76
2020-09-01	A0418150105	AQ Mforce Gen X	Plus 2	86	20.0766	1726.59
2020-09-01	A0418150106	AQ Mforce Gen X	Plus 3	48	19.9365	956.95
2020-09-01	A0519150201	AQ Mforce Gen Y	Standard 1	138	22.3984	3090.98
2020-09-01	A0519150202	AQ Mforce Gen Y	Standard 2	72	24.9298	1794.95
2020-09-01	A0519150203	AQ Mforce Gen Y	Standard 3	38	26.5871	1010.31
2020-09-01	A0519150204	AQ Mforce Gen Y	Plus 1	149	26.1081	3890.11
2020-09-01	A0519150205	AQ Mforce Gen Y	Plus 2	29	29.7008	861.32

Gross monthly total sales for croma

```sql
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=90002002
    GROUP BY date;
```

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

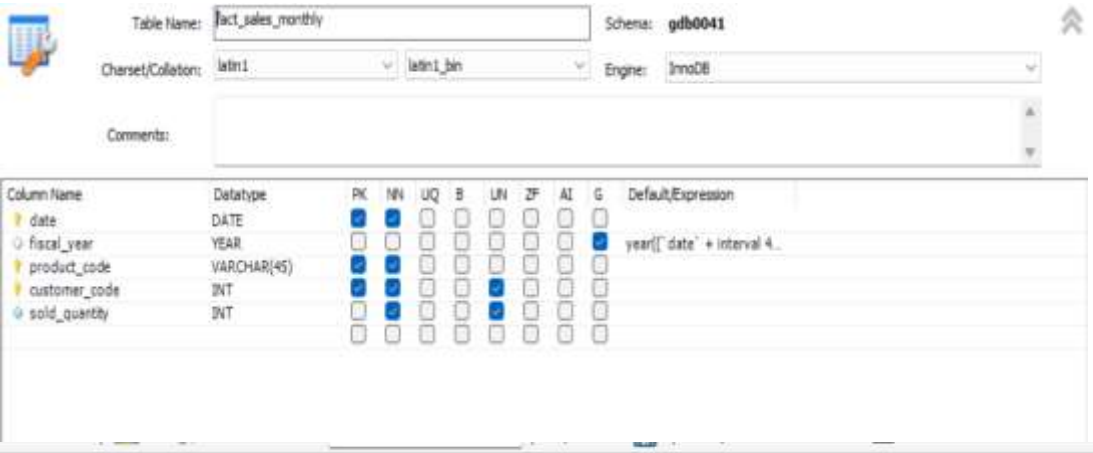
date	monthly_sales
2017-09-01	122407.57
2017-10-01	162687.56
2017-12-01	245673.84
2018-01-01	127574.73
2018-02-01	144799.54
2018-04-01	130643.92
2018-05-01	139165.06
2018-06-01	125735.36
2018-08-01	125409.90
2018-09-01	343337.14
2018-10-01	440562.10
2018-12-01	653944.72

Include pre-invoice deductions for all customers

```
SELECT
        s.date,
        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
        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=get_fiscal_year(s.date)
        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=get_fiscal_year(s.date)
WHERE
        get_fiscal_year(s.date)=2021
```

date	product_code	product	variant	sold_quantity	gross_price_per_item	gross_price_total	pre_invoice_disco
2020-09-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	248	19.0573	4726.21	0.0703
2020-09-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	240	19.0573	4573.75	0.2061
2020-09-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	31	19.0573	590.78	0.0974
2020-09-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	37	19.0573	705.12	0.2065
2020-09-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	7	19.0573	133.40	0.1068
2020-09-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	12	19.0573	228.69	0.2612
2020-09-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	17	19.0573	323.97	0.2471
2020-09-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	60	19.0573	1143.44	0.0858
2020-09-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	34	19.0573	647.95	0.2450
2020-09-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	24	19.0573	457.38	0.0736
2020-09-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	88	19.0573	1677.04	0.2105
2020-09-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	49	19.0573	933.81	0.0793
2020-09-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	60	19.0573	1143.44	0.1817
2020-09-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	7	19.0573	133.40	0.0861

Added the fiscal year in the fact_sales_monthly table itself




```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
```

date	customer_code	product_code	product	variant	sold_quantity	gross_price_per_item	gross_price_total	pre_invoice_discou
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

Creating the view `sales_preinv_discount` and store all the data in like a virtual table and then generating net_invoive_sales

Name: sales_preinv_discount

DDL:

```
    DEFINER = `root`@`localhost`
    SQL SECURITY DEFINER
VIEW `sales_preinv_discount` AS
    SELECT
        `s`.`date` AS `date`,
        `s`.`fiscal_year` AS `fiscal_year`,
        `s`.`customer_code` AS `customer_code`,
        `c`.`market` AS `market`,
        `s`.`product_code` AS `product_code`,
        `p`.`product` AS `product`,
        `p`.`variant` AS `variant`,
        `s`.`sold_quantity` AS `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` AS `pre_invoice_discount_pct`
    FROM
        ((((`fact_sales_monthly` `s`
        JOIN `dim_customer` `c` ON ((`s`.`customer_code` = `c`.`customer_code`)))
        JOIN `dim_product` `p` ON ((`s`.`product_code` = `p`.`product_code`)))
        JOIN `fact_gross_price` `g` ON (((`g`.`fiscal_year` = `s`.`fiscal_year`)
            AND (`g`.`product_code` = `s`.`product_code`))))
        JOIN `fact_pre_invoice_deductions` `pre` ON (((`pre`.`customer_code` = `s`.`customer_code`)
            AND (`pre`.`fiscal_year` = `s`.`fiscal_year`))))
```

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

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

product_code	product	variant	sold_quantity	gross_price_per_item	gross_price_total	pre_invoice_discount_pct	net_invoice_sales
A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	51	15.3952	785.16	0.0824	720.462816
A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	77	15.3952	1185.43	0.2956	835.016892
A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	17	15.3952	261.72	0.0536	247.691808
A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	6	15.3952	92.37	0.2378	70.404414
A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	5	15.3952	76.98	0.1057	68.843214

Database Views: Post Invoice Discount, Net Sales

Name: net_sales

The name of the view is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

DDL:

```
CREATE
    ALGORITHM = UNDEFINED
    DEFINER = `root`@`localhost`
    SQL SECURITY DEFINER
VIEW `net_sales` AS
    SELECT
        `sales_postinv_discount`.`date` AS `date`,
        `sales_postinv_discount`.`fiscal_year` AS `fiscal_year`,
        `sales_postinv_discount`.`customer_code` AS `customer_code`,
        `sales_postinv_discount`.`market` AS `market`,
        `sales_postinv_discount`.`product_code` AS `product_code`,
        `sales_postinv_discount`.`product` AS `product`,
        `sales_postinv_discount`.`variant` AS `variant`,
        `sales_postinv_discount`.`sold_quantity` AS `sold_quantity`,
        `sales_postinv_discount`.`gross_price_total` AS `gross_price_total`,
        `sales_postinv_discount`.`pre_invoice_discount_pct` AS `pre_invoice_discount_pct`,
        `sales_postinv_discount`.`net_invoice_sales` AS `net_invoice_sales`,
        `sales_postinv_discount`.`post_invoice_discount_pct` AS `post_invoice_discount_pct`,
        (`sales_postinv_discount`.`net_invoice_sales` * (1 - `sales_postinv_discount`.`post_invoice_discount_pct`)) AS `net_s
    FROM
        `sales_postinv_discount`
```

```
VIEW `sales_preinv_discount` AS
    SELECT
        `s`.`date` AS `date`,
        `s`.`fiscal_year` AS `fiscal_year`,
        `s`.`customer_code` AS `customer_code`,
        `c`.`market` AS `market`,
        `s`.`product_code` AS `product_code`,
        `p`.`product` AS `product`,
        `p`.`variant` AS `variant`,
        `s`.`sold_quantity` AS `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` AS `pre_invoice_discount_pct`
    FROM
        ((((`fact_sales_monthly` `s`
        JOIN `dim_customer` `c` ON ((`s`.`customer_code` = `c`.`customer_code`)))
        JOIN `dim_product` `p` ON ((`s`.`product_code` = `p`.`product_code`)))
        JOIN `fact_gross_price` `g` ON (((`g`.`fiscal_year` = `s`.`fiscal_year`)
            AND (`g`.`product_code` = `s`.`product_code`))))
        JOIN `fact_pre_invoice_deductions` `pre` ON (((`pre`.`customer_code` = `s`.`cus
            AND (`pre`.`fiscal_year` = `s`.`fiscal_year`))))
```

Net sales report

```
1 •    SELECT * FROM gdb0041.net_sales;
```

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

product	variant	sold_quantity	gross_price_total	pre_invoice_discount_pct	net_invoice_sales	post_invoice_discount_pct	net_sales
AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	4	61.58	0.2803	44.319126	0.3905	27.0125072970
AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	16	246.32	0.2803	177.276504	0.4139	103.9017589944
AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	4	61.58	0.2803	44.319126	0.3295	29.7159739830
AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	6	92.37	0.2803	66.478689	0.3244	44.9130022884
AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	9	138.56	0.2803	99.721632	0.3766	62.1664653888
AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	6	92.37	0.2803	66.478689	0.3615	42.4466429265
AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	7	107.77	0.2803	77.562069	0.3173	52.9516245063
AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	10	153.95	0.2803	110.797815	0.3501	72.0074999685
AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	6	92.37	0.2803	66.478689	0.3740	41.6156593140
AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	4	61.58	0.2117	48.543514	0.2863	34.6455059418
AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	2	30.79	0.2117	24.271757	0.2851	17.3518790793
AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	3	46.19	0.2117	36.411577	0.2882	25.9177605086
AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	5	76.98	0.2117	60.683334	0.3334	40.4515104444
AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	1	15.40	0.2117	12.139820	0.3296	8.1385353280
AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	1	15.40	0.2117	12.139820	0.2901	8.6180582180
AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	5	76.98	0.2117	60.683334	0.3233	41.0644121178
AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	1	15.40	0.2117	12.139820	0.3095	8.3825457100
AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	1	15.40	0.2117	12.139820	0.3209	8.2441517620

Get top 5 market and customers by net sales in fiscal year 2021

```sql
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
```

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

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

```sql
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
order by net_sales_mln desc
limit 5
```

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

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

customer wise net sales percentage contribution

```sql
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
```

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

customer	net_sales_mln	pct_net_sales
Amazon	109.03	13.233402
Atliq Exclusive	79.92	9.700206
Atliq e Store	70.31	8.533803
Sage	27.07	3.285593
Flipkart	25.25	3.064692
Leader	24.52	2.976089
Neptune	21.01	2.550067
Ebay	19.88	2.412914
Electricalsocity	16.25	1.972327
Synthetic	16.10	1.954121
Electricalslytical	15.64	1.898289
Acclaimed Sto...	14.32	1.738075
Propel	14.14	1.716228

Thank you.