Task #1: Extract the Customer Code for a specific customer (CROMA) operating in India from 'dim_customer' table.

```
SELECT *
FROM dim_customer
WHERE customer LIKE "%croma%" AND market = "India"
```

Output Snippet #1:

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

USER DEFINED FUNCTIONS

Task #2: Create a user-defined function `get_fiscal_year` to get corresponding fiscal year (which begins in September for this company), by passing the calendar 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
```

Using this function, get all the sales transactions from 'fact_sales_monthly' table for that customer(Croma: 90002002) for fiscal_year: 2021

```
SELECT * FROM fact_sales_monthly
WHERE customer_code=90002002 AND get_fiscal_year(DATE)=2021
ORDER BY DATE ASC LIMIT 100000;
```

Output Snippet #2: Fiscal Year 2021 starts in September 2020 for this company, so we get all output beginning from date: 2020-09-01.

date	product_code	customer_code	sold_quantity
2020-09-01	A0118150101	90002002	202
2020-09-01	A0118150102	90002002	162
2020-09-01	A0118150103	90002002	193
2020-09-01	A0118150104	90002002	146
2020-09-01	A0219150201	90002002	149
2020-09-01	A0219150202	90002002	107
2020-09-01	A0220150203	90002002	123
2020-09-01	A0320150301	90002002	146
2020-09-01	A0321150302	90002002	236
2020-09-01	A0321150303	90002002	137

Created by: SRISHTI

Task #3: Create another user-defined function `get_fiscal_quarter` to generate quarter-wise sales data for Croma.

```
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
```

Use this function to get 'sold_quantity' data for Croma for Fiscal Year 2021 for Quarter 4.

```
SELECT date, sold_quantity
FROM fact_sales_monthly
WHERE customer_code = 90002002
AND get_fiscal_year(date) = 2021
AND get_fiscal_quarter(date) = "Q4"
ORDER BY DATE ASC;
```

Output Snippet #3:

date	sold_quantity
2021-06-01	205
2021-06-01	78
2021-06-01	48
2021-06-01	126
2021-06-01	40
2021-06-01	102
2021-06-01	31
2021-06-01	91
2021-06-01	70

JOINING TABLES

Task #4: Get Gross Sales Report: Monthly Product-level sales data for Croma, by joining two tables – 'fact_sales_monthly' and 'dim_product'.

```
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 = 90002002
AND get_fiscal_year(date) = 2021
ORDER BY DATE ASC;
```

Output Snippet #4:

date	product_code	product	variant	sold_quantity
2020-09-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	202
2020-09-01	A0118150102	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Plus	162
2020-09-01	A0118150103	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Premium	193
2020-09-01	A0118150104	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Premium Plus	146
2020-09-01	A0219150201	AQ WereWolf NAS Internal Hard Drive HDD – 8....	Standard	149
2020-09-01	A0219150202	AQ WereWolf NAS Internal Hard Drive HDD – 8....	Plus	107
2020-09-01	A0220150203	AQ WereWolf NAS Internal Hard Drive HDD – 8....	Premium	123
2020-09-01	A0320150301	AQ Zion Saga	Standard	146
2020-09-01	A0321150302	AQ Zion Saga	Plus	236

Task #5: Get Gross Price and Gross Price Total amounts from another table, 'fact_gross_price' using another Join.

```
SELECT s.date, p.product_code, p.product, p.variant, s.sold_quantity,
            ROUND (g.gross_price,3) 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 = 90002002
AND get_fiscal_year(date) = 2021
ORDER BY DATE ASC;
```

Output Snippet #5:

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	19.057	3849.57
2020-09-01	A0118150102	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Plus	162	21.457	3475.95
2020-09-01	A0118150103	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Premium	193	21.780	4203.44
2020-09-01	A0118150104	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Premium Plus	146	22.973	3354.04
2020-09-01	A0219150201	AQ WereWolf NAS Internal Hard Drive HDD – 8....	Standard	149	23.699	3531.11
2020-09-01	A0219150202	AQ WereWolf NAS Internal Hard Drive HDD – 8....	Plus	107	24.731	2646.24
2020-09-01	A0220150203	AQ WereWolf NAS Internal Hard Drive HDD – 8....	Premium	123	23.615	2904.69
2020-09-01	A0320150301	AQ Zion Saga	Standard	146	23.722	3463.46
2020-09-01	A0321150302	AQ Zion Saga	Plus	236	27.103	6396.24

Task #6: Get a Gross Sales Report for Croma by joining 'fact_sales_monthly' and 'fact_gross_price' tables.

```
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 = 90002002
GROUP BY s.date
ORDER BY s.date ASC;
```

Output Snippet #6:

date	gross_sales_amount
2017-09-01	122407.56
2017-10-01	162687.57
2017-12-01	245673.80
2018-01-01	127574.74
2018-02-01	144799.52
2018-04-01	130643.90
2018-05-01	139165.10
2018-06-01	125735.38

Task #7: Get Total Yearly Sales Report for Croma using Fiscal Year

```
SELECT get_fiscal_year(s.date) AS fiscal_year,
       ROUND(SUM(g.gross_price*s.sold_quantity),2) AS 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 = 90002002
GROUP BY fiscal_year
ORDER BY fiscal_year ASC;
```

Output Snippet #7:

fiscal_year	yearly_sales
2018	1324097.44
2019	3555079.02
2020	6502181.91
2021	23216512.22
2022	44638198.92

STORED PROCEDURE

Task #8: Create a stored procedure to get Monthly Gross Sales Report for any Customer using their customer code.

```
CREATE 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
```

Then call this stored procedure to get Gross Monthly Sales for Croma using its customer code.

```
CALL get_monthly_gross_sales_for_customer(90002002);
```

Output Snippet #8.1:

date	monthly_sales
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

Sometimes a customer has two or more codes, so the stored procedure needs to accommodate this need. For example, let's say Amazon has two product codes so we create a stored procedure for it:

```
CREATE 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
```

We can then call the stored procedure using the two product codes for Amazon:

```
CALL get_monthly_gross_sales_for_amazon('90002016, 90002008');
```

Output Snippet #8.2:

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

Task #9: Create a stored procedure to give Market Badges to customers based on their sales turnover.

First we need to create a join between 'fact_sales_monthly' and 'dim_customer' table to get total market-wise (country-wise) sales quantities.

```
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;
```

Output Snippet #9:

market	total_qty
India	13751429
Indonesia	1434929
Japan	529487
Pakistan	454393
Philiphines	2422641
South Korea	3947794
Australia	1782354
Newzealand	835190
Bangladesh	575892

Using this output we can develop a stored procedure such that if sold quantity for a market is greater than 5 million for the chosen year, that market is 'Gold', else it is 'Silver'.

```
CREATE 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 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(s.sold_quantity) INTO qty
     FROM fact_sales_monthly s
     JOIN dim_customer c
     ON s.customer_code=c.customer_code
     WHERE get_fiscal_year(s.date)=in_fiscal_year
     AND c.market=in_market;

  # Determine Gold vs Silver status
          IF qty > 5000000 THEN
        SET out_badge = 'Gold';
          ELSE
        SET out_badge = 'Silver';
          END IF;
END
```

Now upon calling this stored procedure, for market = Australia and year = 2020, we get the output as: Silver. This is correct, as Australia's Sales for this year is 917945.

```
SET @out_badge = '0';
CALL get_market_badge('Australia', 2020, @out_badge);
SELECT @out_badge;
```

@out_badge
Silver

market	total_qty
India	5381959
Indonesia	576133
Japan	202306
Pakistan	427543
Philiphines	1160524
South Korea	1970860
Australia	917945
Newzealand	275390

Task #10: Add a 'fiscal_year' column to 'fact_sales_monthly' table, and use it to join with 'pre invoice deductions' table to get 'pre_invoice_discount_pct'.

```
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 Snippet #10:

date	customer_code	product_code	product	variant	sold_quantity	gross_price_per_item	gross_price_total	pre_invoice_discount_pct
2020-09-01	90001021	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5...	Standard	58	19.0573	1105.32	0.3095
2020-09-01	90001021	A0118150102	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5...	Plus	36	21.4565	772.43	0.3095
2020-09-01	90001021	A0118150103	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5...	Premium	77	21.7795	1677.02	0.3095
2020-09-01	90001021	A0118150104	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5...	Premium...	44	22.9729	1010.81	0.3095
2020-09-01	90001021	A0219150201	AQ WereWolf NAS Internal Hard Drive HD...	Standard	71	23.6987	1682.61	0.3095
2020-09-01	90001021	A0219150202	AQ WereWolf NAS Internal Hard Drive HD...	Plus	52	24.7312	1286.02	0.3095
2020-09-01	90001021	A0220150203	AQ WereWolf NAS Internal Hard Drive HD...	Premium	11	23.6154	259.77	0.3095
2020-09-01	90001021	A0320150301	AQ Zion Saga	Standard	54	23.7223	1281.00	0.3095
2020-09-01	90001021	A0321150302	AQ Zion Saga	Plus	76	27.1027	2059.81	0.3095
2020-09-01	90001021	A0321150303	AQ Zion Saga	Premium	80	28.0059	2240.47	0.3095

DATABASE VIEWS

Task #11: Create a view named `sales_preinv_discount` to store and access all current data captured for future reuse.

```
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
```

Use this view to generate 'net_invoice_sales' data:

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

Output Snippet #11:

date	product_code	customer_c	product	variant	fiscal_year	sold_quar	market	gross_price_p	gross_price_total	pre_invoice_discount_pct	net_invoice_sales
2017-09...	A0118150101	70002017	AQ Dracula H...	Standard	2018	51	India	15.3952	785.16	0.0824	720.462816
2017-09...	A0118150101	70002018	AQ Dracula H...	Standard	2018	77	India	15.3952	1185.43	0.2956	835.016892
2017-09...	A0118150101	70003181	AQ Dracula H...	Standard	2018	17	Indonesia	15.3952	261.72	0.0536	247.691808
2017-09...	A0118150101	70003182	AQ Dracula H...	Standard	2018	6	Indonesia	15.3952	92.37	0.2378	70.404414
2017-09...	A0118150101	70006157	AQ Dracula H...	Standard	2018	5	Philiphines	15.3952	76.98	0.1057	68.843214
2017-09...	A0118150101	70006158	AQ Dracula H...	Standard	2018	7	Philiphines	15.3952	107.77	0.1875	87.563125
2017-09...	A0118150101	70007198	AQ Dracula H...	Standard	2018	29	South Ko...	15.3952	446.46	0.0700	415.207800
2017-09...	A0118150101	70007199	AQ Dracula H...	Standard	2018	34	South Ko...	15.3952	523.44	0.2551	389.910456
2017-09...	A0118150101	70008169	AQ Dracula H...	Standard	2018	22	Australia	15.3952	338.69	0.0953	306.412843
2017-09...	A0118150101	70008170	AQ Dracula H...	Standard	2018	5	Australia	15.3952	76.98	0.1896	62.384592

Task #12: Create a view `sales_postinv_discount` to capture 'post_invoice_discount_pct':

```
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;
```

Using this view, create a report for 'net_sales':

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

Output Snippet #12:

date	fiscal_year	customer_cc	market	product_code	product	variant	sold_qu	gross_price_	pre_invoice_	net_invoice_sa	post_invoice	net_sales
2017-12-01	2018	80007196	South ...	A0118150101	AQ Dracula HDD – 3....	Standard	344	5295.95	0.2231	4114.423555	0.4628	2210.2683337...
2017-11-01	2018	80007195	South ...	A0118150101	AQ Dracula HDD – 3....	Standard	274	4218.28	0.2990	2957.014280	0.3430	1942.7583819...
2017-11-01	2018	80007196	South ...	A0118150101	AQ Dracula HDD – 3....	Standard	192	2955.88	0.2231	2296.423172	0.4423	1280.7152030...
2018-06-01	2018	80007196	South ...	A0118150101	AQ Dracula HDD – 3....	Standard	149	2293.88	0.2231	1782.115372	0.3591	1142.1577419...
2018-03-01	2018	80007195	South ...	A0118150101	AQ Dracula HDD – 3....	Standard	169	2601.79	0.2990	1823.854790	0.3881	1116.0167460...
2017-11-01	2018	90002009	India	A0118150101	AQ Dracula HDD – 3....	Standard	136	2093.75	0.2156	1642.337500	0.3539	1061.1142587...
2018-11-01	2019	90022081	USA	A0118150101	AQ Dracula HDD – 3....	Standard	158	2281.39	0.2483	1714.920863	0.4016	1026.2086444...
2018-03-01	2018	80007196	South ...	A0118150101	AQ Dracula HDD – 3....	Standard	140	2155.33	0.2231	1674.475877	0.3895	1022.2675229...
2018-04-01	2018	80007196	South ...	A0118150101	AQ Dracula HDD – 3....	Standard	138	2124.54	0.2231	1650.555126	0.3983	993.1390193142

Now save 'net_sales' as a view for future use:

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

Task #13: Get Top 5 Market (Countries) by Net Sales in Fiscal Year 2021

```
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;
```

Output Snippet #13:

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

Task #14: Create a stored procedure to get Top N Markets by Net Sales for a chosen 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
```

Use this stored procedure to see Top 5 markets in 2020:

```
CALL get_top_n_markets_by_net_sales(2020, 5);
```

Output Snippet #14:

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

Task #15: Create a stored procedure to get Top N Customers by Net Sales for a chosen year.

```
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
    JOIN dim_customer c
    ON s.customer_code=c.customer_code
    WHERE fiscal_year=in_fiscal_year AND s.market=in_market
    GROUP BY customer
    ORDER BY net_sales_mln DESC
    LIMIT in_top_n;
END
```

Call this stored procedure to see Top 5 customers in Japan in 2020:

```
CALL get_top_n_customers_by_net_sales('Japan', 2020, 5);
```

Output Snippet #15:

customer	net_sales_mln
Amazon	49.77
Atliq e Store	31.74
Atliq Exclusive	22.97
Flipkart	10.92
Sage	8.31

Task #16: Create a stored procedure to get the Top N Products by Net Sales for a chosen year.

```
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 net_sales
    WHERE fiscal_year=in_fiscal_year
    GROUP BY product
    ORDER BY net_sales_mln DESC
    LIMIT in_top_n;
END
```

Call this stored procedure to see data for Top 5 Products in 2021:

```
CALL get_top_n_products_by_net_sales(2021, 5);
```

Output Snippet #16:

product	net_sales_mln
AQ Wi Power Dx2	14.37
AQ BZ Gen Y	12.09
AQ Wi Power Dx1	11.84
AQ Lite	11.55
AQ BZ Compact	11.40

COMMON TABLE EXPRESSION & WINDOW FUNCTION

Task #17: Find out customer-wise Net Sales percentage contribution using the stored procedure for Top N Customers by modifying the query to create a Common Table Expression (CTE) on which we can create a Window Function.

```
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;
```

Output #17:

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

Task #18: Find Customer-wise Net Sales distribution per region for Fiscal Year 2021.

```
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 *,
       net_sales_mln*100/SUM(net_sales_mln) OVER (PARTITION BY region)
       AS pct_share_region
FROM cte1
ORDER BY region, pct_share_region DESC;
```

Output Snippet #18:

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

Task #19: Find Top 3 Products from each Division by total quantity sold in a given year using DENSE_RANK.

```
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.division,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 Snippet #19.1:

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 wirel...	2448784	3
PC	AQ Digit	135092	1
PC	AQ Gen Y	135031	2
PC	AQ Elite	134431	3

Task #20: Retrieve the Top 2 Markets in every Region by their gross sales amount in Fiscal Year 2021.

```
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;
```

Output Snippet #20:

market	region	gross_sales_mln	rnk
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

***