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

```sql
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.

```sql
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

```sql
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	A5419110207	90002002	5
2020-09-01	A0118150102	90002002	162
2020-09-01	A4218110207	90002002	26
2020-09-01	A5519110301	90002002	3
2020-09-01	A2918150102	90002002	296
2020-09-01	A0118150103	90002002	193
2020-09-01	A5621110408	90002002	14
2020-09-01	A3220150403	90002002	781
2020-09-01	A0118150104	90002002	146

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 H…	Standard	202
2020-09-01	A0118150102	AQ Dracula H…	Plus	162
2020-09-01	A0118150103	AQ Dracula H…	Premium	193
2020-09-01	A0118150104	AQ Dracula H…	Premium Plus	146
2020-09-01	A0219150201	AQ WereWol…	Standard	149
2020-09-01	A0219150202	AQ WereWol…	Plus	107
2020-09-01	A0220150203	AQ WereWol…	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 H…	Standard	202	19.057	3849.57
2020-09-01	A4419110403	AQ Elite	Standard Red	16	288.050	4608.80
2020-09-01	A2720150701	AQ Trigger Ms	Standard 1	822	17.092	14049.38
2020-09-01	A4218110204	AQ Digit	Plus Grey	27	232.104	6266.80
2020-09-01	A5419110205	AQ Gamer 2	Plus Cool Blue	7	570.758	3995.30
2020-09-01	A5419110206	AQ Gamer 2	Plus Black	4	601.640	2406.56
2020-09-01	A3220150401	AQ Lite	Standard 1	197	18.494	3643.38
2020-09-01	A5419110204	AQ Gamer 2	Plus Firey Red	5	602.920	3014.60
2020-09-01	A2620150606	AQ Qwerty Ms	Premium 2	688	16.785	11548.08

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

```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 = 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

```sql
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.

```sql
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.

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

Output Snippet #8.1:

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

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:

```sql
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
Philipines	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;
```

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

@out_level
Silver

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'.

```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 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	70002017	A0118150101	AQ Dr...	Standard	248	19.0573	4726.21	0.0703
2020-09-01	70002018	A0118150101	AQ Dr...	Standard	240	19.0573	4573.75	0.2061
2020-09-01	70003181	A0118150101	AQ Dr...	Standard	31	19.0573	590.78	0.0974
2020-09-01	70003182	A0118150101	AQ Dr...	Standard	37	19.0573	705.12	0.2065
2020-09-01	70004069	A0118150101	AQ Dr...	Standard	7	19.0573	133.40	0.1068
2020-09-01	70004070	A0118150101	AQ Dr...	Standard	12	19.0573	228.69	0.2612
2020-09-01	70005163	A0118150101	AQ Dr...	Standard	17	19.0573	323.97	0.2471
2020-09-01	70006157	A0118150101	AQ Dr...	Standard	60	19.0573	1143.44	0.0858
2020-09-01	70006158	A0118150101	AQ Dr...	Standard	34	19.0573	647.95	0.2450
2020-09-01	70007198	A0118150101	AQ Dr...	Standard	24	19.0573	457.38	0.0736

DATABASE VIEWS

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

```sql
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	fiscal	customer_	market	product_	product	variar	sold_quantity	gross_price	gross_price_tc	pre_invoice_discount_pct	net_invoice_sales
20...	2018	70002017	India	A011...	AQ D...	Sta...	51	15.3952	785.16	0.0824	616411.528416
20...	2018	70002018	India	A011...	AQ D...	Sta...	77	15.3952	1185.43	0.2956	1404893.871792
20...	2018	70003181	Indo...	A011...	AQ D...	Sta...	17	15.3952	261.72	0.0536	68483.330208
20...	2018	70003182	Indo...	A011...	AQ D...	Sta...	6	15.3952	92.37	0.2378	8510.251314
20...	2018	70006157	Philip...	A011...	AQ D...	Sta...	5	15.3952	76.98	0.1057	5917.783614
20...	2018	70006158	Philip...	A011...	AQ D...	Sta...	7	15.3952	107.77	0.1875	11594.166025
20...	2018	70007198	Sout...	A011...	AQ D...	Sta...	29	15.3952	446.46	0.0700	199295.279400
20...	2018	70007199	Sout...	A011...	AQ D...	Sta...	34	15.3952	523.44	0.2551	273855.904056
20...	2018	70008169	Austr...	A011...	AQ D...	Sta...	22	15.3952	338.69	0.0953	114678.638943
20...	2018	70008170	Austr...	A011...	AQ D...	Sta...	5	15.3952	76.98	0.1896	5911.324992

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':

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

Output Snippet #12:

date	fiscal_	customer	market	product	product	variant	sold	gross_price	pre_invc	net_invoice_sale	post_invoice_discount_	net_sales
20...	2018	70002...	India	A011...	AQ D...	Stand...	51	785.16	0.0824	720.462816	0.3379	477.0184304736
20...	2018	70002...	India	A011...	AQ D...	Stand...	77	1185.43	0.2956	835.016892	0.4013	499.9246132404
20...	2018	70003...	Indo...	A011...	AQ D...	Stand...	17	261.72	0.0536	247.691808	0.3752	154.7578416384
20...	2018	70003...	Indo...	A011...	AQ D...	Stand...	6	92.37	0.2378	70.404414	0.3446	46.1430529356
20...	2018	70006...	Phili...	A011...	AQ D...	Stand...	5	76.98	0.1057	68.843214	0.3065	47.7427689090
20...	2018	70006...	Phili...	A011...	AQ D...	Stand...	7	107.77	0.1875	87.563125	0.3587	56.1542320625
20...	2018	70007...	Sout...	A011...	AQ D...	Stand...	29	446.46	0.0700	415.207800	0.3343	276.4038324600
20...	2018	70007...	Sout...	A011...	AQ D...	Stand...	34	523.44	0.2551	389.910456	0.4168	227.3957779392
20...	2018	70008...	Aust...	A011...	AQ D...	Stand...	22	338.69	0.0953	306.412843	0.3129	210.5362644253
20...	2018	70008...	Aust...	A011...	AQ D...	Stand...	5	76.98	0.1896	62.384592	0.3189	42.4901456112

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

```sql
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

```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;
```

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.

```sql
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 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
```

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	0.76
Atliq e Store	0.32
Atliq Exclusive	0.23
All-Out	0.21
Surface Stores	0.19

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 BZ Allin1	33.75
AQ Qwerty	27.84
AQ Trigger	26.95
AQ Gen Y	23.58
AQ Maxima	22.32

**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

Visual Chart of Top 10 Customers by Net Sales Contribution in % (graphed after extracting the data in Excel)
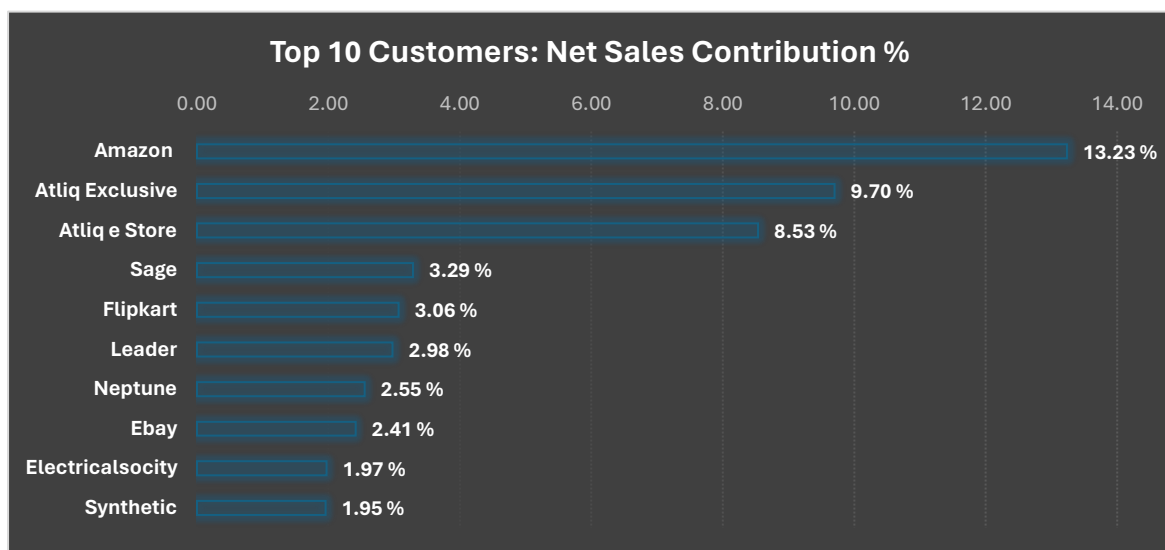
Task #18: Find Customer-wise Net Sales distribution per region for Fiscal Year 2021.

```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 *,
       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
Synthetic	APAC	14.14	3.199095
Propel	APAC	14.14	3.199095
Flipkart	APAC	12.96	2.932127

Pie Chart representing Top 10 Customers for APAC region: (graphed after extracting the data in Excel)

Top 10 Customers: Net Sales % by APAC Region
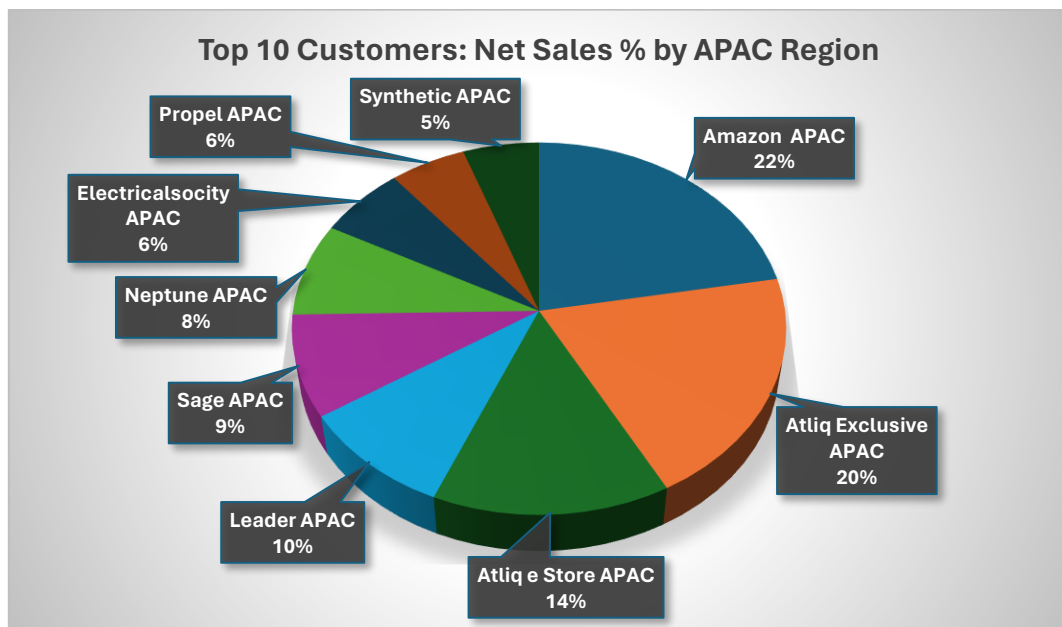
Task #19: Find Top 3 Products from each Division by total quantity sold in a given year using DENSE_RANK.

```
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.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 Dri...	2034569	1
N & S	AQ Digit SSD	1240149	2
N & S	AQ Clx1	1238683	3
P & A	AQ Gamers ...	2477098	1
P & A	AQ Maxima ...	2461991	2
P & A	AQ Master ...	2448784	3
PC	AQ Digit	135092	1
PC	AQ Gen Y	135031	2
PC	AQ Elite	134431	3

Create a stored procedure for the above query and call it to see Top 2 Products in each Division for 2020:

```
CALL get_top_n_products_per_division_by_qty_sold(2020, 2);
```

Output Snippet #19.2:

division	product	total_qty	drnk
N & S	AQ Clx1	935128	1
N & S	AQ Neuer SSD	924264	2
P & A	AQ Master ...	1578253	1
P & A	AQ Gamers ...	1566445	2
PC	AQ Digit	68862	1
PC	AQ Elite	67841	2

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

```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
            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

***