Request 1) Finding customer_code for croma in india?
sol:

```
SELECT * FROM dim_customer
WHERE customer = "croma" AND market = "india";
```

Request 2) All sales_montly_transcations for croma for 2021 in india market?
(hint: customer_code of croma for india is 90002002)
sol:

```
SELECT * FROM fact_sales_monthly
WHERE customer_code = 90002002 AND YEAR(date) = 2021;
```

Request 3) User defined function for fiscal year (sep 2020 = 1st month of 2021)
sol:

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

Request 4) For croma 2021 fiscal_year give gross_price, product, variant, sold_quanitiy, gross_price_total
sol:

```
SELECT s.date, s.product_code, p.product, p.variant, s.sold_quantity, g.gross_price,
ROUND((g.gross_price * s.sold_quantity), 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.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 DESC;
```

Request 5) Gross monthly sales report of croma
sol:

```
SELECT s.date, SUM(ROUND((g.gross_price * s.sold_quantity), 2)) AS gross_price_total
FROM fact_sales_monthly s
JOIN fact_gross_price g ON s.product_code = g.product_code AND g.fiscal_year = get_fiscal_year(s.date)
WHERE customer_code = 90002002
GROUP BY s.date
ORDER BY date DESC;
```

Request 6) Gross year sales report of croma
sol:

```
SELECT get_fiscal_year(s.date) AS fiscal_year,
SUM(ROUND((g.gross_price * s.sold_quantity), 2)) AS gross_price_total
FROM fact_sales_monthly s
JOIN fact_gross_price g ON s.product_code = g.product_code AND g.fiscal_year = get_fiscal_year(s.date)
WHERE customer_code = 90002002
GROUP BY get_fiscal_year(s.date)
ORDER BY fiscal_year;
```

Request 7) Stored procedure for monthly_gross_sales_for_customer
sol:

```
CREATE    DEFINER=`root`@`localhost`    PROCEDURE    get_monthly_gross_sales_for_customer(in_customer_code
```

```
TEXT)
BEGIN
    SELECT get_fiscal_year(date) AS fiscal_year,
    SUM(sold_quantity * g.gross_price) AS total_gross_price
    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_code) > 0
    GROUP BY get_fiscal_year(date);
END;
```

Request 8) Stored procedure for market badge based on total sales quantity
sol:

```
CREATE DEFINER=`root`@`localhost` PROCEDURE get_market_badge(
    IN in_market VARCHAR(45),
    IN in_fiscal_year YEAR,
    OUT out_level VARCHAR(45)
)
BEGIN
    DECLARE qty INT DEFAULT 0;
    IF in_market = "" THEN
        SET in_market = "India";
    END IF;
    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;
    IF qty > 5000000 THEN
        SET out_level = 'Gold';
    ELSE
        SET out_level = 'Silver';
    END IF;
END;
```

Request 9) Top Customer, Product, Market by net_sales of a particular fiscal_year
Step 1: Calculate net_invoice_sales

```
WITH cte1 AS (
SELECT s.date, s.product_code, p.product, p.variant, s.sold_quantity, g.gross_price, s.customer_code,
ROUND((g.gross_price * s.sold_quantity), 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 dim_date dt ON dt.calendar_date = s.date
JOIN fact_gross_price g ON g.product_code = s.product_code AND g.fiscal_year = dt.fiscal_year
JOIN fact_pre_invoice_deductions pre ON pre.customer_code = s.customer_code AND pre.fiscal_year = dt.fiscal_year
WHERE dt.fiscal_year = 2021
LIMIT 1000000)
SELECT *, (gross_price_total - gross_price_total * pre_invoice_discount_pct) AS net_invoice_sales FROM cte1;
```

Step 2: Create view sales_preinv_discount

```
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 pre ON pre.customer_code = s.customer_code AND pre.fiscal_year = s.fiscal_year;
```

Step 3: Calculate net_invoice_sales using view

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

Step 4: Create view sales_postinv_discount

```
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,
(gross_price_total - pre_invoice_discount_pct * 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;
```

Step 5: Final net_sales view

```
CREATE VIEW net_sales AS
SELECT *, net_invoice_sales * (1 - post_invoice_discount_pct) AS net_sales
FROM gdb0041.sales_postinv_discount;
```

Top Markets and Customers Module:

Get top 5 markets by net sales in fiscal year 2021:

```
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;
```

Stored procedure to get top n markets by net sales for a given 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;
```

Stored procedure for top n customers by net sales:

```
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;
```