QUESTION 1

Generate a report of individual product sales (aggregated on a monthly basis at the product code level) For Chroma India customer for FY = 2021

QUERY

```
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 asc;
```

QUESTION 2

Generate an aggregated monthly gross sales report for Croma India customer

QUERY

```
SELECT
    s.date,
    SUM(g.gross_price*sold_quantity) as gross_price_total
FROM fact_sales_monthly s
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
GROUP BY s.date
ORDER BY s.date
;
```

QUESTION 3

Create a stored procedure that can determine the market badge based on the following logic:

If total sold quantity > 5 million That market is considered gold else it is silver

QUERY:

```
CREATE PROCEDURE `get_market_badge`
(
    IN in_market VARCHAR(20),
    IN in_fiscal_year YEAR,
    OUT out_badge VARCHAR(10)
)
BEGIN

    DECLARE qty INT DEFAULT 0;

    #set default market to be India
        IF in_market = "" THEN
            SET in_market = "India";
        END IF;

    #retrieve total quantity for a given market and fiscal year

    SELECT
        sum(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
    GROUP BY c.market;

    #determine market badge
    IF qty > 5000000 THEN
        SET out_badge = "Gold";
    ELSE
        SET out_badge = "Silver";
    END IF;

END
```

QUESTION 4

Create a report for TOP MARKETS by net sales for a given financial year

QUERY

```
CREATE PROCEDURE `get_top_n_markets`(
in_fiscal_year INT,
in_top_n INT
)
BEGIN
    SELECT
        market,
        ROUND(SUM(net_sales)/1000000,2) as net_sales_mn
    FROM net_sales
    WHERE fiscal_year = in_fiscal_year
    GROUP BY market
    ORDER BY net_sales_mn desc
    LIMIT in_top_n;
END
```

QUESTION 5

Create a report for TOP PRODUCTS by net sales for a given financial year

QUERY

```
CREATE PROCEDURE `get_top_n_products` (
    in_fiscal_year int,
    in_top_n int
)
BEGIN
    SELECT
        product,
        ROUND(SUM(net_sales)/1000000,2) as net_sales_mn
    FROM net_sales
    WHERE fiscal_year = in_fiscal_year
    GROUP BY product
    ORDER BY net_sales_mn desc
    LIMIT in_top_n;
END
```

QUESTION 6

Create a report for TOP CUSTOMERS by net sales for a given financial year

QUERY

```
CREATE PROCEDURE `get_top_n_customers` (
    in_market VARCHAR(40),
    in_fiscal_year int,
    in_top_n int
)
BEGIN
    SELECT
    customer,
    ROUND(SUM(net_sales)/1000000,2) as net_sales_mn
    FROM net_sales s
    JOIN dim_customer c
        ON c.customer_code = s.customer_code
    WHERE
        s.fiscal_year = in_fiscal_year AND
        s.market = in_market
    GROUP BY customer
    ORDER BY net_sales_mn desc
    LIMIT in_top_n;
END
```

QUESTION 7

Create a report for FY 2021 for top 10 markets by % net sales

QUERY

```sql
WITH cte1 AS
(
    SELECT
        customer,
        ROUND(SUM(net_sales)/1000000,2) as net_sales_mn
    FROM net_sales s
    JOIN dim_customer c
        ON c.customer_code = s.customer_code
    WHERE
        s.fiscal_year = 2021
    GROUP BY customer
)
SELECT
    *,
    net_sales_mn*100/sum(net_sales_mn) over() as pct
FROM cte1
ORDER BY net_sales_mn desc
```

QUESTION 8

Generate a region wise net sales % breakdown by customers

QUERY

```sql
WITH cte1 AS
(
  SELECT
      c.customer,
      c.region,
      ROUND(SUM(net_sales)/1000000,2) as net_sales_mn
      FROM net_sales s
      JOIN dim_customer c
          ON c.customer_code = s.customer_code
      WHERE
          s.fiscal_year = 2021
      GROUP BY c.customer, c.region

)
SELECT
    *,
    net_sales_mn*100/sum(net_sales_mn) over(partition by region) as pct_share_region
FROM cte1
ORDER BY region, net_sales_mn desc
```

QUESTION 9

Write a stored procedure for getting top n products in each division by their quantities sold in a given FY

QUERY

```sql
CREATE PROCEDURE `get_top_n_products_per_division_by_qty_sold` (
    in_fiscal_year INT,
    in_top_n INT
)
BEGIN
    WITH
    cte1 AS
    (
      SELECT
          p.division,
          p.product,
          SUM(sold_quantity) as total_qty
      FROM dim_product p
      JOIN fact_sales_monthly s
      ON p.product_code = s.product_code
      WHERE fiscal_year=in_fiscal_year
      GROUP BY p.product, p.division
    ),
    cte2 AS
    (
      SELECT
          *,
          dense_rank() over(partition by division order by total_qty desc) as d_rnk
      FROM cte1
    )
    SELECT
        *
    FROM cte2
    WHERE d_rnk <=in_top_n;
END
```

QUESTION 10

Retrieve top 2 markets in every region by their gross sales amount in financial year 2021

QUERY

```sql
WITH
  cte1 AS
  (
    SELECT
        c.market,
        c.region,
        ROUND(SUM(gross_price_total/1000000),2) as gross_price_mn
    FROM dim_customer c
    JOIN gross_sales s
    ON c.customer_code = s.customer_code
    GROUP BY c.region,c.market
  ),
  cte2 AS
  (
    SELECT
        *,
        dense_rank() over(partition by region order by gross_price_mn desc) AS rnk
    FROM cte1
    )
  SELECT * FROM cte2 WHERE rnk⩽2;
```

QUESTION 11

Generate a report to see which customers' forecast accuracy has dropped from 2020 to 2021

QUERY

```
# Creating Temporary Table of Forecast Accuracy 2021

    CREATE TEMPORARY TABLE fa_2021
    WITH cte1 AS
    (
    SELECT
        s.customer_code,
        c.customer as customer_name,
        c.market,
        SUM(sold_quantity) as total_sold_qty,
        SUM(forecast_quantity) as total_forecast_qty,
        SUM((forecast_quantity - sold_quantity)) as Net_Error,
        SUM((forecast_quantity - sold_quantity))*100/SUM(forecast_quantity) as Net_Error_pct,
        SUM(abs(forecast_quantity - sold_quantity)) as Abs_Net_Error,
        SUM(abs(forecast_quantity-sold_quantity))*100/SUM(forecast_quantity) as
Abs_Net_Error_pct
    FROM fact_act_est s
    JOIN dim_customer c
    ON s.customer_code = c.customer_code
    WHERE s.fiscal_year = 2021
    GROUP BY customer_code
    )
    SELECT
        *,
        if(Abs_Net_Error_pct > 100, 0, 100-Abs_Net_Error_pct) as Forecast_Accuracy_2021
    FROM cte1
    ORDER BY Forecast_Accuracy_2021 desc;

# Creating Temporary Table of Forecast Accuracy 2020

  CREATE TEMPORARY TABLE fa_2020
  WITH cte2 AS
  (
  SELECT
      s.customer_code,
      c.customer as customer_name,
      c.market,
      SUM(sold_quantity) as total_sold_qty,
      SUM(forecast_quantity) as total_forecast_qty,
      SUM((forecast_quantity - sold_quantity)) as Net_Error,
      SUM((forecast_quantity - sold_quantity))*100/SUM(forecast_quantity) as Net_Error_pct,
      SUM(abs(forecast_quantity - sold_quantity)) as Abs_Net_Error,
      SUM(abs(forecast_quantity - sold_quantity))*100/SUM(forecast_quantity) as
Abs_Net_Error_pct
  FROM fact_act_est s
  JOIN dim_customer c
  ON s.customer_code = c.customer_code
  WHERE s.fiscal_year = 2020
  GROUP BY customer_code
  )
  SELECT
      *,
      if(Abs_Net_Error_pct > 100, 0, 100-Abs_Net_Error_pct) as Forecast_Accuracy_2020
  FROM cte2
  ORDER BY Forecast_Accuracy_2020 desc;

# Joining both Temporary tables

  SELECT
      fa_2021.customer_code,
      fa_2021.customer_name,
      fa_2021.market,
      Forecast_Accuracy_2020,
      Forecast_Accuracy_2021
  FROM fa_2021
  JOIN fa_2020
  ON fa_2021.customer_code=fa_2020.customer_code
  WHERE Forecast_Accuracy_2021 < Forecast_Accuracy_2020
  ORDER BY Forecast_Accuracy_2020
```

QUESTION 12

Generate an aggregated forecast accuracy report for all the customers for FY 2021

QUERY

```sql
WITH cte1 AS
(
  SELECT
      a.customer_code,
      SUM(a.sold_qty) as total_sold_qty,
      SUM(a.forecast_qty) as total_forecast_qty,
      SUM((forecast_qty - sold_qty)) as Net_Error,
      SUM((forecast_qty - sold_qty))*100/SUM(forecast_qty) as Net_Error_pct,
      SUM(abs(forecast_qty - sold_qty)) as Abs_Net_Error,
      SUM(abs(forecast_qty - sold_qty))*100/SUM(forecast_qty) as
Abs_Net_Error_pct
  FROM fact_act_est a
  WHERE a.fiscal_year = 2021
  GROUP BY customer_code
)
SELECT
    e.*,
    c.customer,
    c.market,
    if(Abs_Net_Error_pct > 100, 0, 100-Abs_Net_Error_pct) as Forecast_Accuracy
FROM cte1 e
JOIN dim_customer c
ON e.customer_code=c.customer_code
ORDER BY Forecast_Accuracy desc
```