SQL Query

1. **Croma India Product wise sales report for fiscal year 2021.**

   The Report should have the following fields.

   a.) Month
   b.) Product Name & Variant
   c.) Sold Quantity
   d.) Gross Price Per item
   e.) Gross Price Total

Query:

Step 1: create a function 'get_fiscal_year' to get fiscal year by passing the 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
```

Step 2: query

```
select
        s.date,s.product_code,
p.product,p.variant,s.sold_quantity,
g.gross_price,(gross_price*sold_quantity) as gross_price_total
from fact_sales_monthly s
join dim_product p
on p.product_code=s.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
```

2. **Gross monthly total sales report for Croma**

   The Report should have the following fields.

   a.) Month
   b.) Total Gross Sales amount to Croma India in this month

Query:

```
select
        s.date,s.product_code,
p.product,p.variant,s.sold_quantity,
g.gross_price,(gross_price*sold_quantity) as gross_price_total
from fact_sales_monthly s
join dim_product p
on p.product_code=s.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
```

3. **Create a stored proc that can determine the market badge based on the following logic.**

   If total sold quantity > 5 million that market is connected Gold else it is Silver.

   My Input will be
   - **Market**
   - **Fiscal Year**

   Output
   - **Market badge**

<u>Query:</u>

Stored Procedure:

```
CREATE DEFINER=`root`@`localhost` 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 to me india
if in_market="" then
        set in_market="india";
end if ;
# retrive total qty for a given market+fyear
        SELECT
        SUM(sold_quantity) into qty
FROM fact_sales_monthly s
JOIN dim_customer c
ON c.customer_code=s.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
```

4. **Write a Stored proc for**
   a.) Top Market by net sales
   b.) Top Product by net sales
   c.) Top Customers by net sales

<u>Query:</u>

Step1: Database View for 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 as pre
        ON pre.customer_code = s.customer_code AND
                pre.fiscal_year=s.fiscal_year
```

Step 2: Now generate net_invoice_sales using the above created view "sales_preinv_discount"

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

Step 3: Database View for 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,
(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;
```

Step 4: Now generate net_sales using the above created view "sales_postinv_discount"

```
 SELECT

*,

 net_invoice_sales*(1-post_invoice_discount_pct) as net_sales

FROM gdb0041.sales_postinv_discount;
```

Step 5: Finally creating the view `net_sales` which inbuiltly use/include all the previous created view and gives the final result

```
 CREATE VIEW `net_sales` AS

	SELECT

*,

	 net_invoice_sales*(1-post_invoice_discount_pct) as net_sales

	FROM gdb0041.sales_postinv_discount;
```

Step 6: Stored proc 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
```

Step 7: stored procedure that takes market, fiscal_year and top n as an input and returns top n customers by net sales in that given fiscal year and market

```
 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
```

Step 8 : top n products by net sales

```
 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 gdb041.net_sales
where fiscal_year=in_fiscal_year
group by product
order by net_sales_mln desc
limit in_top_n;
        END
```

5. **Net sales % share Global**

   As a product owner , I want to see a bar chart report for FY-2021 for top 10 markets by % net sales.

<u>Query:</u>

```
 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
```

6. **Net Sales % share by region**

As a product owner , I want to see region wise (APAC, EU, LTAM etc)% net sales breakdown by customers in a respective region so that I can perform my regional analysis on financial performance of the company.0
FY=2021

Query:

```
with cte1 as (
                select
           c.customer,
                c.region,
                round(sum(net_sales)/1000000,2) as net_sales_mln
            from gdb0041.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
```

Supply Chain

1. Forecast Accuracy for all customers for given fiscal year

   a.) Customer Code, Name, Market
   b.) Total Sold Quantity
   c.) Total Forecast Quantity
   d.) Net Error
   e.) Absolute Error
   f.) Forecast Accuracy %

<u>Query</u>:

```
 create temporary table forecast_err_table

select

s.customer_code as customer_code,

c.customer as customer_name,

c.market as market,

sum(s.sold_quantity) as total_sold_qty,

sum(s.forecast_quantity) as total_forecast_qty,

sum(s.forecast_quantity-s.sold_quantity) as net_error,

round(sum(s.forecast_quantity-s.sold_quantity)*100/sum(s.forecast_quantity),1) as
net_error_pct,
sum(abs(s.forecast_quantity-s.sold_quantity)) as abs_error,

round(sum(abs(s.forecast_quantity-sold_quantity))*100/sum(s.forecast_quantity),2
) as abs_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_error_pct > 100, 0, 100.0 - abs_error_pct) as forecast_accuracy

        from forecast_err_table

order by forecast_accuracy desc;
```