Business 360 SQL Queries

##-Get all the sales transaction data from fact_sales_monthly table for that customer(croma: 90002002) in the fiscal_year 2021 --##

```
SELECT * FROM fact_sales_monthly
        WHERE customer_code = 90002002 AND
   year(DATE_ADD(DATE, INTERVAL 4 MONTH))= 2021
ORDER BY date ASC
LIMIT 100000;
```

##- a) Perform Join to pull product information : Monthly Product Transaction##

```
SELECT s.date, s.product_code, p.product, p.variant, s.sold_quantity
        FROM fact_sales_monthly s JOIN
   dim_product p ON
   s.product_code = p.product_code
WHERE customer_code     = 90002002 AND
get_fiscal_year(date) = 2021
LIMIT 1000000;
```

b) gross price total

```
SELECT s.date, s.product_code,
        p.product, p.variant, s.sold_quantity,
   g.gross_price,
   round(s.sold_quantity * g.gross_price,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
        WHERE customer_code = 90002002 AND
   get_fiscal_year(s.date) = 2021
   limit 1000000;
```

c) Gross price total by Month(Croma Monthly total sales)

```
SELECT s.date,
        sum(ROUND(g.gross_price*s.sold_quantity)) 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 =
90002002get_monthly_gross_sales_for_customerget_monthly_gross_sales_for_customer
GROUP BY s.date
ORDER BY s.date ASC;
```

Generate a yearly report for Croma India where there are 2 columns 1) Fiscal Year 2)Total Gross Sales amount in that year from Croma

```
SELECT get_fiscal_year(date) AS fiscal_year ,
 SUM(sold_quantity * gross_price) AS yearly_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 = 90002002
GROUP BY get_fiscal_year(date)
ORDER BY fiscal_year;
```

Pre-Invoice Discount Report

```
SELECT s.date, 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 1000000;
```

Net Invoice Sales

```
WITH CTE1 AS(
SELECT s.date, 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 1000000
)
```

Post Invoice Discount and Net Sales

```
SELECT *,
        ROUND(((1 - pre_invoice_discount_pct )* gross_price_total),2) 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 s.date = po.date AND
   s.product_code = po.product_code AND
   s.customer_code = po.customer_code;
```

Get TOP 5 markets by net sales for the 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;
```

Get Top N Customers by Net Sales for the fiscal year

```
 SELECT c.customer,
        round(sum(net_sales)/1000000,2) as net_sales_mln
FROM gdb0041.net_sales s
        JOIN dim_customer c ON
c.customer_code = s.customer_code
        WHERE fiscal_year=2021
        GROUP BY c.customer
        ORDER BY net_sales_mln desc
        LIMIT 5;
```

Get Top n Products by net sales for the fiscal year

```
SELECT p.product,
        ROUND(SUM(net_sales) /1000000,2) AS net_sales_mln
FROM gdb0041.net_sales s
JOIN dim_product p ON
        p.product_code = s.product_code
   WHERE fiscal_year = 2021
   GROUP BY p.product
   ORDER BY net_sales_mln
   LIMIT 5;
```

Bar Chart for FY 2021 for top 10 markets by % net sales.

```
WITH CTE1 AS (
SELECT c.customer,
        round(sum(net_sales)/1000000,2) as net_sales_mln
FROM gdb0041.net_sales s
        JOIN dim_customer c ON
c.customer_code = s.customer_code
        WHERE s.fiscal_year = 2021
        GROUP BY c.customer
)
SELECT *, net_sales_mln * 100/sum(net_sales_mln) over() as pct_net_sales
        FROM cte1
        ORDER BY net_sales_mln desc
```

Region Wise(APAC,EU,LATAM) %net sales breakdown by customers in a respective region for the FY 2021

```
WITH CTE2  AS (
SELECT c.customer, c.region,
        ROUND(sum(net_sales)/1000000,2) AS net_sales_mln
FROM gdb0041.net_sales s
```

```
JOIN dim_customer c ON
            c.customer_code = s.customer_code
   WHERE fiscal_year = 2021
   GROUP BY c.region, c.customer
  )
SELECT *, net_sales_mln*100/sum(net_sales_mln) over(partition by region) AS pct_share
            FROM CTE2
   ORDER BY  region, pct_share DESC
```

Get Top n products in each division by thier quantity sold

```
WITH CTE1 AS (
   SELECT
      p.division,
      p.product,
      SUM(sold_quantity) AS Qty_sold
   FROM dim_product p
   JOIN fact_sales_monthly s
      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 Qty_sold ) AS drank
   FROM CTE1
)
SELECT *
FROM CTE2
WHERE drank <= 3;
```

Retrieve top 2 markets in every region by thier gross sales amount in FY=2021

```
WITH CTE1 AS (
SELECT c.region,
        c.market,
   ROUND(SUM(gross_price_total)/1000000,2) AS Gross_Sales_Mln
FROM dim_customer c JOIN
gross_sales g ON
        c.customer_code = g.customer_code
        WHERE fiscal_year = 2021
GROUP BY c.region, c.market
),
CTE2 AS(
SELECT *,
        dense_rank() over(partition by region order by Gross_Sales_Mln DESC) AS gross_sales
   FROM CTE1
)
SELECT * FROM CTE2
        WHERE gross_sales <=2;
```

Joining the table for Forecast Accuracy

```
create table fact_act_est
        (
        select
           s.date as date,
           s.fiscal_year as fiscal_year,
           s.product_code as product_code,
           s.customer_code as customer_code,
           s.sold_quantity as sold_quantity,
           f.forecast_quantity as forecast_quantity
        from
           fact_sales_monthly s
```

```
        left join fact_forecast_monthly f
        using (date, customer_code, product_code)
        )
        union
        (
        select
           f.date as date,
           f.fiscal_year as fiscal_year,
           f.product_code as product_code,
           f.customer_code as customer_code,
           s.sold_quantity as sold_quantity,
           f.forecast_quantity as forecast_quantity
        from
                    fact_forecast_monthly  f
        left join fact_sales_monthly s
        using (date, customer_code, product_code)
        );


 set sql_safe_updates = 1;
UPDATE fact_act_est
        SET forecast_quantity = 0
        WHERE forecast_quantity is null;

UPDATE fact_act_est
        SET sold_quantity = 0
    WHERE sold_quantity is null;
```

Forecast Accuracy Report-Get Forecast Accuracy of FY 2021 and store that in a temporary table

```
CREATE temporary table Forecast_error_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-s.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_error_table
ORDER BY forecast_accuracy DESC;
```

Get Forecast Accuracy of FY 2020 and store that in a temporary table

```
WITH forecast_accuracy AS (
SELECT c.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 forecast_qty,
    sum(s.forecast_quantity-s.sold_quantity) AS net_error,
    sum(s.forecast_quantity-s.sold_quantity)*100/sum(s.forecast_quantity) AS net_error_pct,
    sum(abs(s.forecast_quantity-s.sold_quantity)) AS abs_error,
    sum(abs(s.forecast_quantity-s.sold_quantity))*100/sum(s.forecast_quantity) AS Abs_error_pct
FROM fact_act_est s
JOIN dim_customer c
        ON c.customer_code = s.customer_code
    WHERE fiscal_year = 2020
```

```
    GROUP BY customer_code
)
SELECT *,
        IF(abs_error_pct>100, 0, 100.0-abs_error_pct) AS forecast_accuracy
FROM forecast_accuracy
ORDER BY forecast_accuracy DESC;
```