*SQL-_Consumer_Goods_Analysis*

*SQL-based data analytics solution for AtliQ Hardwares, a consumer goods company!!*

**1. To extract the following details and compute the Gross Price Total**

```
-- Month
-- Product Name
-- Variant
-- Sold Quantity
-- Gross Price Per Item
-- Gross Price Total


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 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
        AND GET_FISCAL_QUARTER(date) = 'Q4'
ORDER BY DATE ASC
LIMIT 1000000;
```

**2.**

```
-- Month
-- Total gross sales amount in India based on months
```

```
SELECT
    s.date,
    SUM(g.gross_price*s.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 ASC;

-- 3
-- Fiscal Year
-- Total Gross sales amount in specific year from Croma

SELECT
     get_fiscal_year(date) AS fiscal_year,
     ROUND(SUM(g.gross_price*s.sold_quantity),2)AS yearly_gross_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 g.fiscal_year
ORDER BY fiscal_year;

-- 4
-- Fiscal_year
-- Market
-- Market Badge

SELECT
    SUM(sold_quantity) AS total_qty
FROM fact_sales_monthly s
```

```
JOIN dim_customer c
ON c.customer_code = s.customer_code
WHERE get_fiscal_year(s.date)=2021 AND c.market = "India"
GROUP BY c.market;

-- 5
-- Report for Top Markets
-- Report for Top Products
-- Report for Top Customers

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 pre
ON pre.customer_code = s.customer_code AND
  pre.fiscal_year = s.fiscal_year
WHERE
   s.fiscal_year=2021
LIMIT 1000000;


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 pre
ON pre.customer_code = s.customer_code AND
  pre.fiscal_year = s.fiscal_year
WHERE
   s.fiscal_year=2021)
SELECT
        *,
   (gross_price_total - gross_price_total * pre_invoice_discount_pct) AS net_invoice_sales
FROM sales_preinv_discount;
```

**# Net_Invoice_Sales_**

```
SELECT
        *,
   (1 - 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
  s.date = po.date AND
  s.product_code = po.product_code AND
  s.customer_code = po.customer_code;

  SELECT
        s.date,
    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;
```

**# NET Sales**

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

**# Gross Sales**

```
SELECT
    s.date,
    s.fiscal_year,
    s.customer_code,
    c.customer,
    c.market,
    c.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
        FROM
    fact_sales_monthly s
        JOIN dim_product p
  ON s.product_code = p.product_code
  JOIN dim_customer c
  ON s.customer_code=c.customer_code
  JOIN fact_gross_price g
  ON s.fiscal_year = g.fiscal_year AND
    s.product_code = g.product_code;
```

**# Top 5 Market**

```
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;
```

**# Top Customers**

```
SELECT
    customer,
    ROUND(SUM(net_sales)/1000000,2) AS net_sales_mln
FROM net_sales
WHERE fiscal_year = 2021
GROUP BY customer
ORDER BY net_sales_mln DESC
LIMIT 5;
```

**# Top Products**

```
SELECT
    product,
    ROUND(SUM(net_sales)/1000000,2) AS net_sales_mln
FROM net_sales
WHERE fiscal_year = 2021
GROUP BY product
ORDER BY net_sales_mln DESC
LIMIT 5;
```

**#Net sales Global Market Share**

```
WITH cte1 AS(
SELECT
    customer,
```

```
    ROUND(SUM(net_sales)/1000000,2) AS net_sales_mln
FROM net_sales s
WHERE s.fiscal_year = 2021
GROUP BY customer)
SELECT *,
   net_sales_mln*100/sum(net_sales_mln) OVER()AS pct
FROM cte1
ORDER BY net_sales_mln DESC;
```

**# Net sales % share by region**

```
WITH cte1 AS(
SELECT
    customer,
    region,
    ROUND(SUM(net_sales)/1000000,2) AS net_sales_mln
FROM net_sales s
WHERE fiscal_year = 2021
GROUP BY customer, region
ORDER BY net_sales_mln DESC)

SELECT *,
          net_sales_mln*100/SUM(net_sales_mln) OVER(partition by region) AS pct_share_region
FROM CTE1
ORDER BY region, net_sales_mln DESC;

#

          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;
```

**# top 2 markets in every region by their gross sales amount**

```
WITH cte1 AS (
SELECT
        c.market,
        c.region,
        round(sum(gross_price_total)/1000000,2) as gross_sales_mln
        FROM gross_sales s
        JOIN dim_customer c
        ON c.customer_code=s.customer_code
        WHERE fiscal_year=2021
        GROUP BY market
        ORDER BY gross_sales_mln DESC
),
cte2 AS (
        SELECT *,
        dense_rank() over(PARTITION BY region ORDER BY gross_sales_mln DESC) AS drnk
        FROM cte1
)
SELECT * FROM cte2 WHERE drnk<=2;
```

**# Creating a New Table**

```
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, product_code, customer_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)
);

update fact_act_est
set sold_quantity = 0
where sold_quantity is null;

update fact_act_est
set forecast_quantity= 0
where forecast_quantity is null;
```

**# Get forecast accuracy of FY 2021 and store that in a temporary table**

```
create temporary table forecast_accuracy_2021
with forecast_err_table as (
      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;
```

**# Get forecast accuracy of FY 2020 and store that also in a temporary table**

```
drop table if exists forecast_accuracy_2020;
create temporary table forecast_accuracy_2020
with forecast_err_table as (
    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=2020
    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;
```

**# Join forecast accuracy tables for 2020 and 2021 using a customer_code**

```
select
		f_2020.customer_code,
		f_2020.customer_name,
		f_2020.market,
		f_2020.forecast_accuracy as forecast_acc_2020,
		f_2021.forecast_accuracy as forecast_acc_2021
from forecast_accuracy_2020 f_2020
join forecast_accuracy_2021 f_2021
on f_2020.customer_code = f_2021.customer_code
where f_2021.forecast_accuracy < f_2020.forecast_accuracy
order by forecast_acc_2020 desc;

select
		*,
    (forecast_quantity - sold_quantity) as net_err,
    (forecast_quantity - sold_quantity)*100/forecast_quantity as net_err_pct,
    abs(forecast_quantity - sold_quantity) as abs_err,
		abs(forecast_quantity - sold_quantity)*100/forecast_quantity as abs_err_pct
FROM fact_act_est;

with forecast_err_table as
(select
		s.customer_code,
		sum(s.sold_quantity) as total_sold_qty,
    sum((s.forecast_quantity - sold_quantity)) as net_err,
		sum((forecast_quantity - sold_quantity))*100/(forecast_quantity) as net_err_pct,
    sum(abs(forecast_quantity - sold_quantity)) as abs_err,
		sum(abs(forecast_quantity - sold_quantity))*100/(forecast_quantity) as abs_err_pct
```

```
FROM fact_act_est s
where s.fiscal_year = 2021
group by customer_code)
select
        e.*,
    c.customer,
    c.market,
    if (abs_err_pct > 100, 0, 100-abs_err_pct) as forecast_accuracy
from forecast_err_table e
join dim_customer c
using (customer_code)
order by forecast_accuracy ASC;
```