Total Gross Sales Amount In that year from Croma

```
select
          get_fiscal_year(date) as fiscal_year,
          sum(round(sold_quantity*g.gross_price,2)) 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;
```

Stored Procedure: -

1. Top 10 Market by Net Sales

```
CREATE DEFINER=`root`@`localhost` 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 gdb0041.net_sales
where fiscal_year = in_fiscal_year
group by market
```

```
order by net_sales_mln desc

limit in_top_n;

END
```

2. Top 10 Products by Net Sales

```
CREATE DEFINER=`root`@`localhost` 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 gdb0041.net_sales
          where fiscal_year=in_fiscal_year
          group by product
          order by net_sales_mln desc
          limit in_top_n;

END
```

3. Top 10 Customers by Net Sales

```
CREATE DEFINER=`root`@`localhost` 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 fiscal_year = in_fiscal_year and s.market = in_market
group by customer
order by net_sales_mln desc
```

```
        limit in_top_n;
        END
```

Gross Sales: -

```
CREATE
    ALGORITHM = UNDEFINED
    DEFINER = `root`@`localhost`
    SQL SECURITY DEFINER
VIEW `gross sales` AS
    SELECT
        `s`.`date` AS `date`,
        `s`.`fiscal_year` AS `fiscal_year`,
        `s`.`customer_code` AS `customer_code`,
        `c`.`customer` AS `customer`,
        `c`.`market` AS `market`,
        `s`.`product_code` AS `product_code`,
        `p`.`product` AS `product`,
        `p`.`variant` AS `variant`,
        `s`.`sold_quantity` AS `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 (((`g`.`fiscal_year` = `s`.`fiscal_year`)
            AND (`g`.`product_code` = `s`.`product_code`))))
```

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

```
with cte1 as (
        select
            c.market,
            c.region,
            round(sum(gross_price)/1000000,2) as gross_sales_mln
            from fact_sales_monthly s
            join dim_customer c
            on c.customer_code=s.customer_code
    Join fact_gross_price p
    on p.product_code = s.product_code
            where s.fiscal_year = 2021
            group by c.market, c.region
            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;
```

Supply Chain Analytics: -

The supply chain business manager wants to see the customer’s forecast accuracy for 2021. Provide a complete report.

```
create temporary table forecast_err_table;
```

```
with forecast_err_tale as(
 select
 s.customer_code,
 sum(s.sold_quantity) as total_sold_quantity,
 sum(s.forecast_quantity) as total_forecast_quantity,
 sum(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 gdb0041.fact_act_est s
 where 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
  join dim_customer c
  using(customer_code)
  order by  forecast_accuracy desc;
```