Atliq Hardware Ad-Hoc Insights

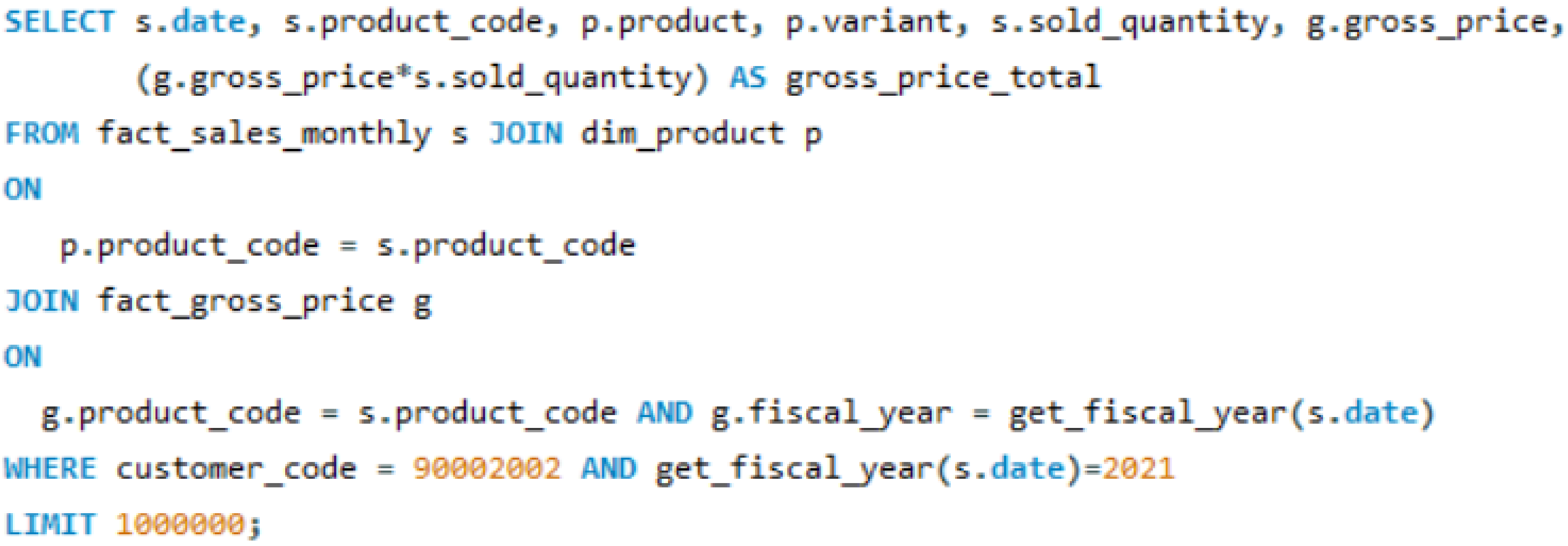
Report 1: Croma India product wise sales for fiscal year 2021

As a product owner, generate a report of individual product sales for Croma India customer for FY – 2021.

The report should have following fields:

1. Month
2. Product name
3. Variant
4. Sold quantity
5. Gross Price Per Item
6. Gross Price Total

Report 1:



```
SELECT s.date, s.product_code, p.product, p.variant, s.sold_quantity, g.gross_price,
       (g.gross_price*s.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(s.date)=2021
LIMIT 1000000;
```

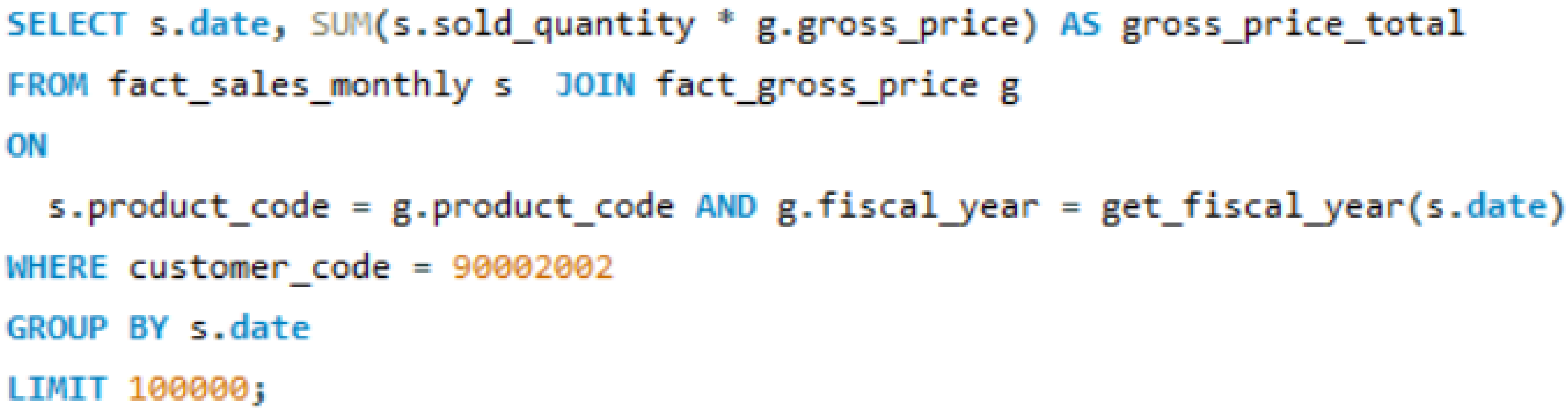
Report 2: Gross Monthly total sales report for Croma

As a product owner, generate a aggregate monthly gross sales report for Croma India customer so that I can track how much sales this particular customer is generating for Atliq and manage our relationship accordingly.

The report should have following fields,

1. Month
2. Total gross sales to Croma India in this month

Report 2:



```
SELECT s.date, SUM(s.sold_quantity * g.gross_price) 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 = 90002002
GROUP BY s.date
LIMIT 100000;
```

Report 3: Stored Procedure for market badge

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

If **total sold quantity > 5** million then market is considered **Gold** else it is **Silver**

My input will be

-Market

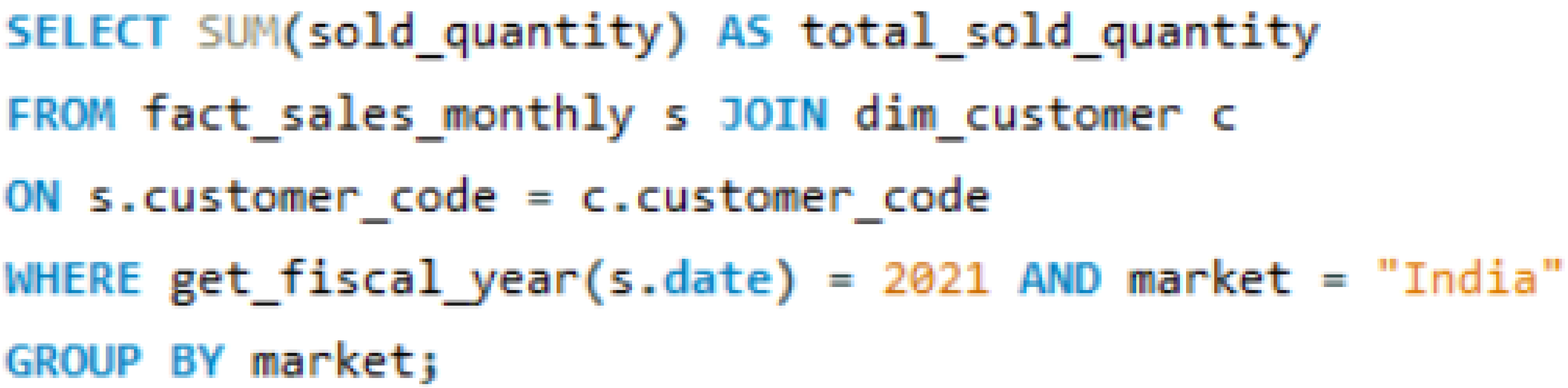
-Fiscal Year

Output:

Market badge

Report 3:

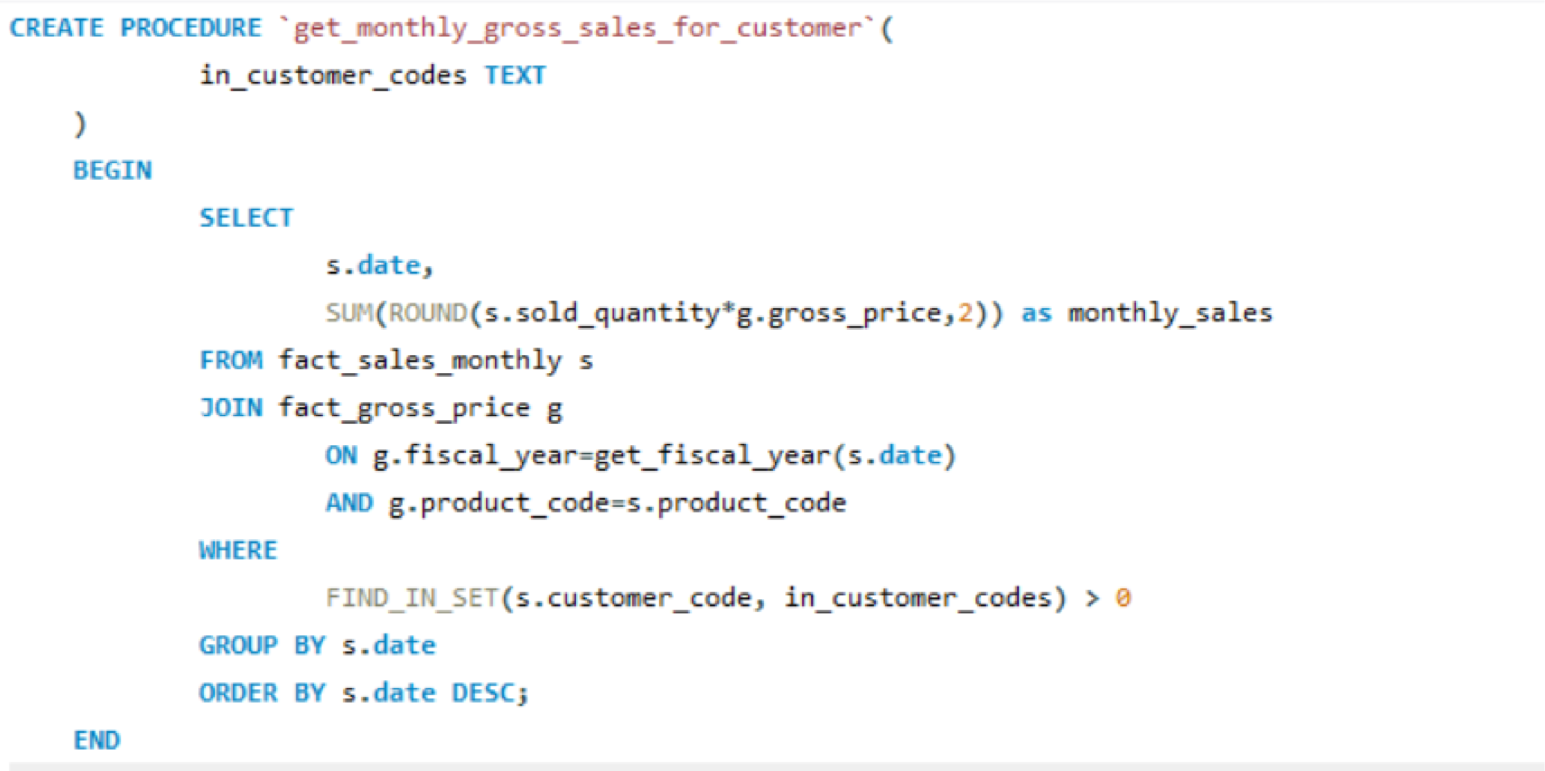
Step-1: Generate monthly gross sales report for Croma India for all the years



```sql
SELECT SUM(sold_quantity) AS total_sold_quantity
FROM fact_sales_monthly s JOIN dim_customer c
ON s.customer_code = c.customer_code
WHERE get_fiscal_year(s.date) = 2021 AND market = "India"
GROUP BY market;
```

Report 3:

Step-2: Creating stored procedure for the above report for all customers



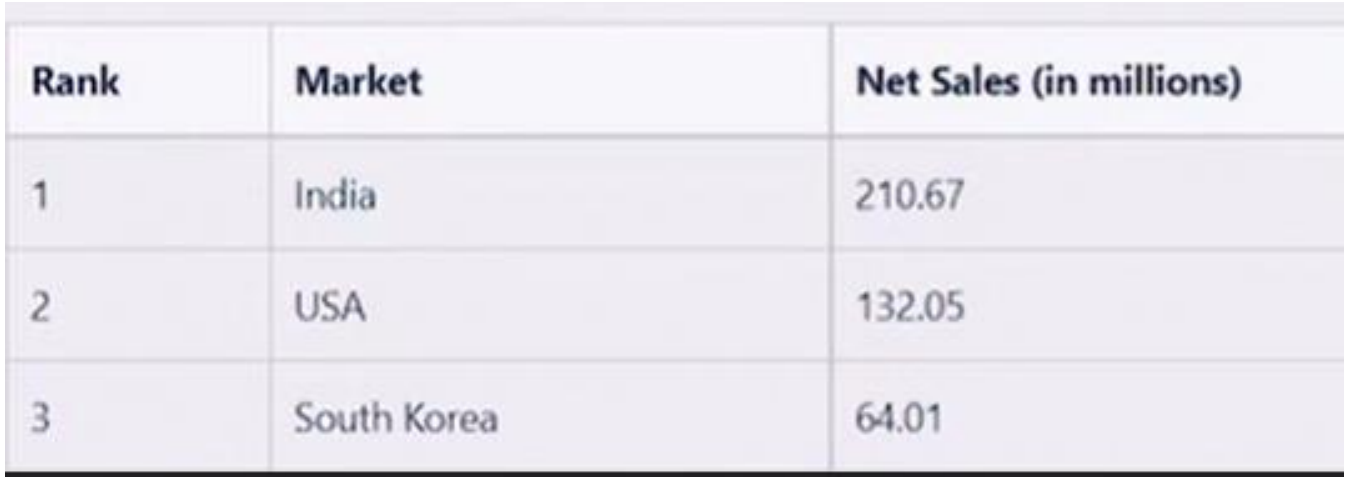
```
CREATE PROCEDURE `get_monthly_gross_sales_for_customer`(
            in_customer_codes TEXT
    )
    BEGIN
            SELECT
                    s.date,
                    SUM(ROUND(s.sold_quantity*g.gross_price,2)) as monthly_sales
            FROM fact_sales_monthly s
            JOIN fact_gross_price g
                    ON g.fiscal_year=get_fiscal_year(s.date)
                    AND g.product_code=s.product_code
            WHERE
                    FIND_IN_SET(s.customer_code, in_customer_codes) > 0
            GROUP BY s.date
            ORDER BY s.date DESC;
    END
```

Report 4: Top markets for a given fiscal year

As a product owner I want a report for top markets by net sales for given fiscal year so that I can have a holistic view of our financial performance.

We will probably write stored procedure for this as we will need this report going forward.

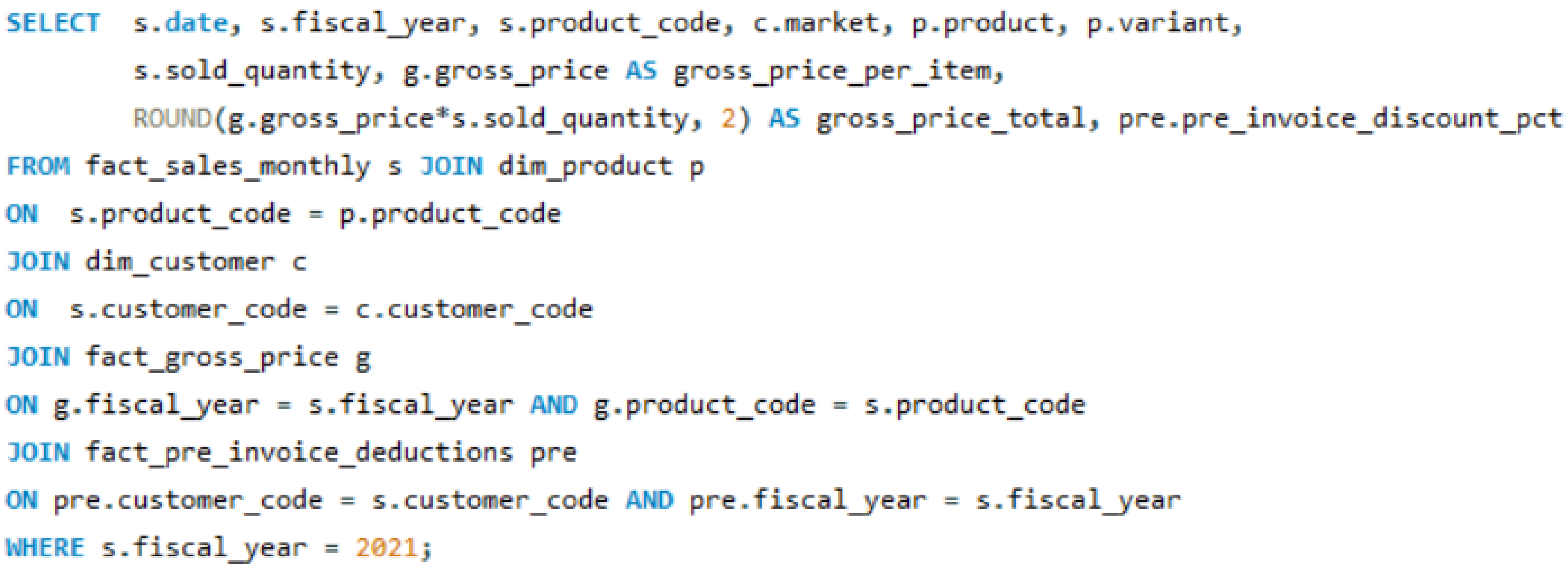
Report for **top markets**



Rank	Market	Net Sales (in millions)
1	India	210.67
2	USA	132.05
3	South Korea	64.01

Report 4:

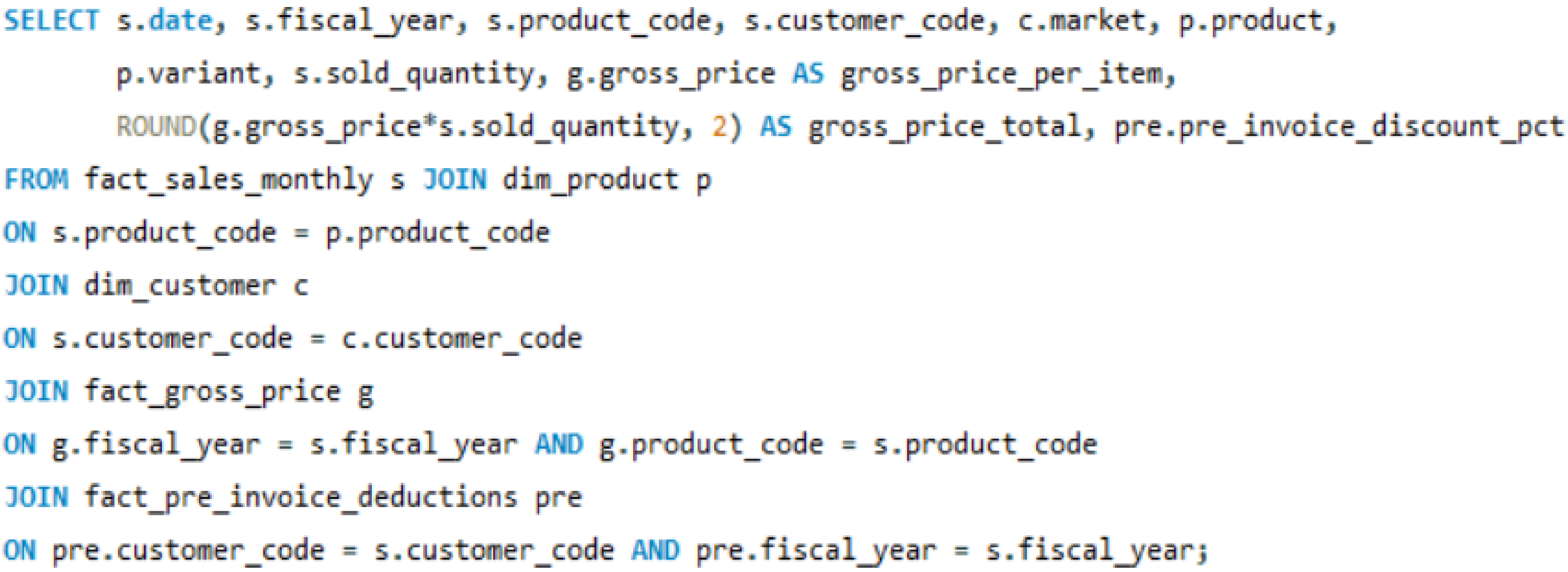
Step-1: Get the net invoice sales amount using the CTE's



```sql
SELECT  s.date, s.fiscal_year, s.product_code, c.market, p.product, p.variant,
        s.sold_quantity, g.gross_price AS gross_price_per_item,
        ROUND(g.gross_price*s.sold_quantity, 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 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
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;
```

Report 4:

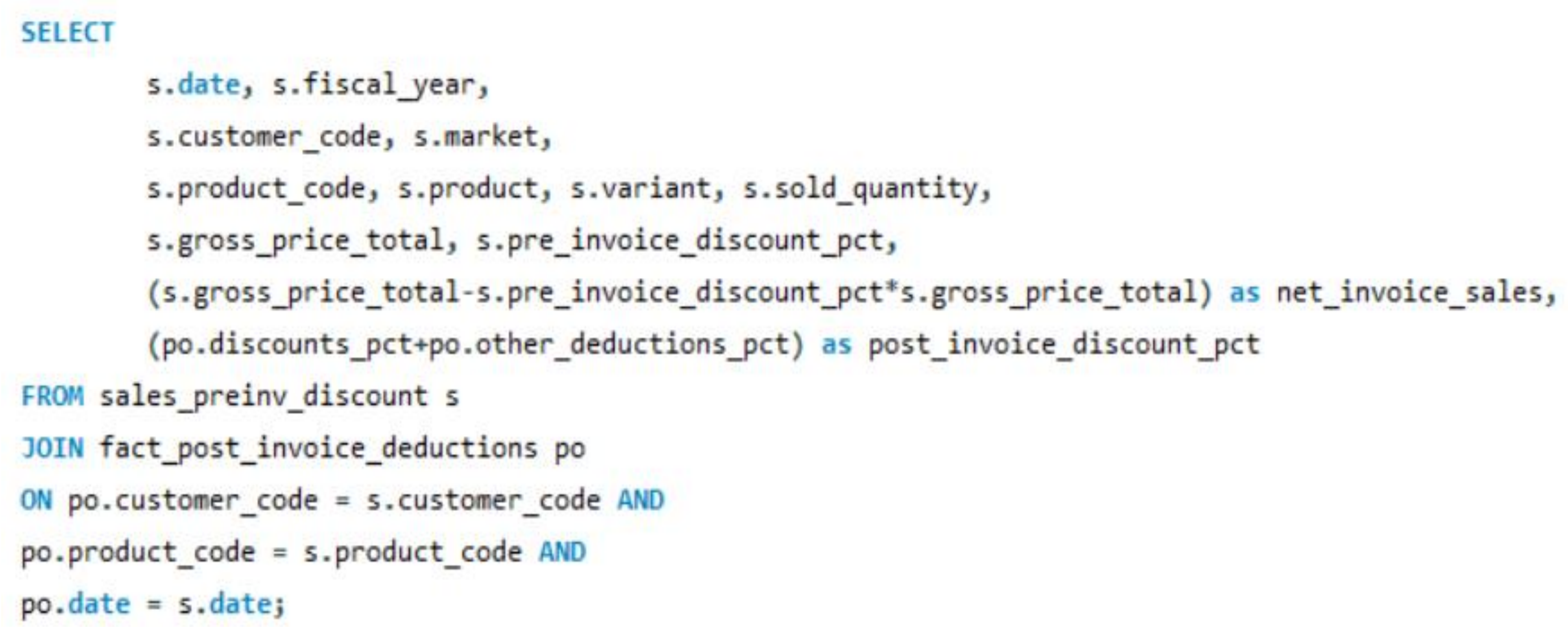
Step-2: Creating the view `sales preinv discount`



```
SELECT s.date, s.fiscal_year, s.product_code, s.customer_code, c.market, p.product,
       p.variant, s.sold_quantity, g.gross_price AS gross_price_per_item,
       ROUND(g.gross_price*s.sold_quantity, 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 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
JOIN fact_pre_invoice_deductions pre
ON pre.customer_code = s.customer_code AND pre.fiscal_year = s.fiscal_year;
```

Report 4:

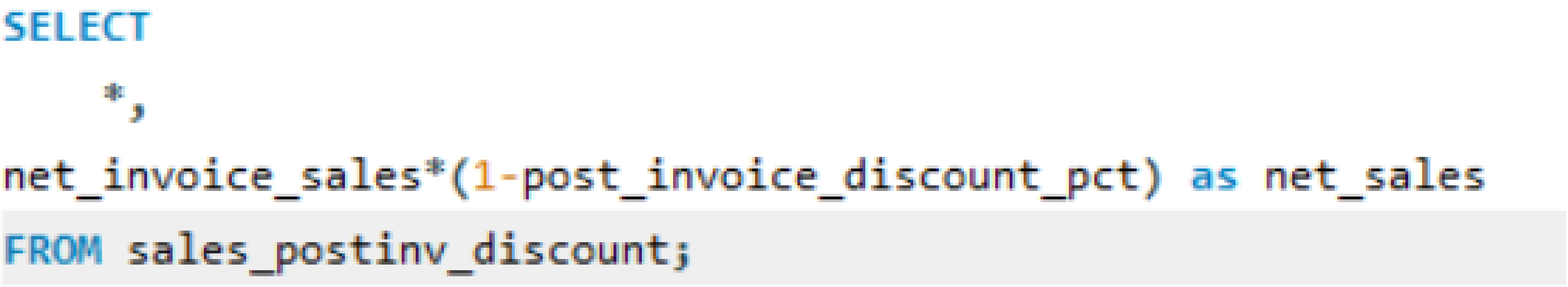
Step-3: Now generate ‘net invoice sales' and 'post invoice discount pct' using the above created view "sales preinv discount"`



```sql
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;
```

Report 4:

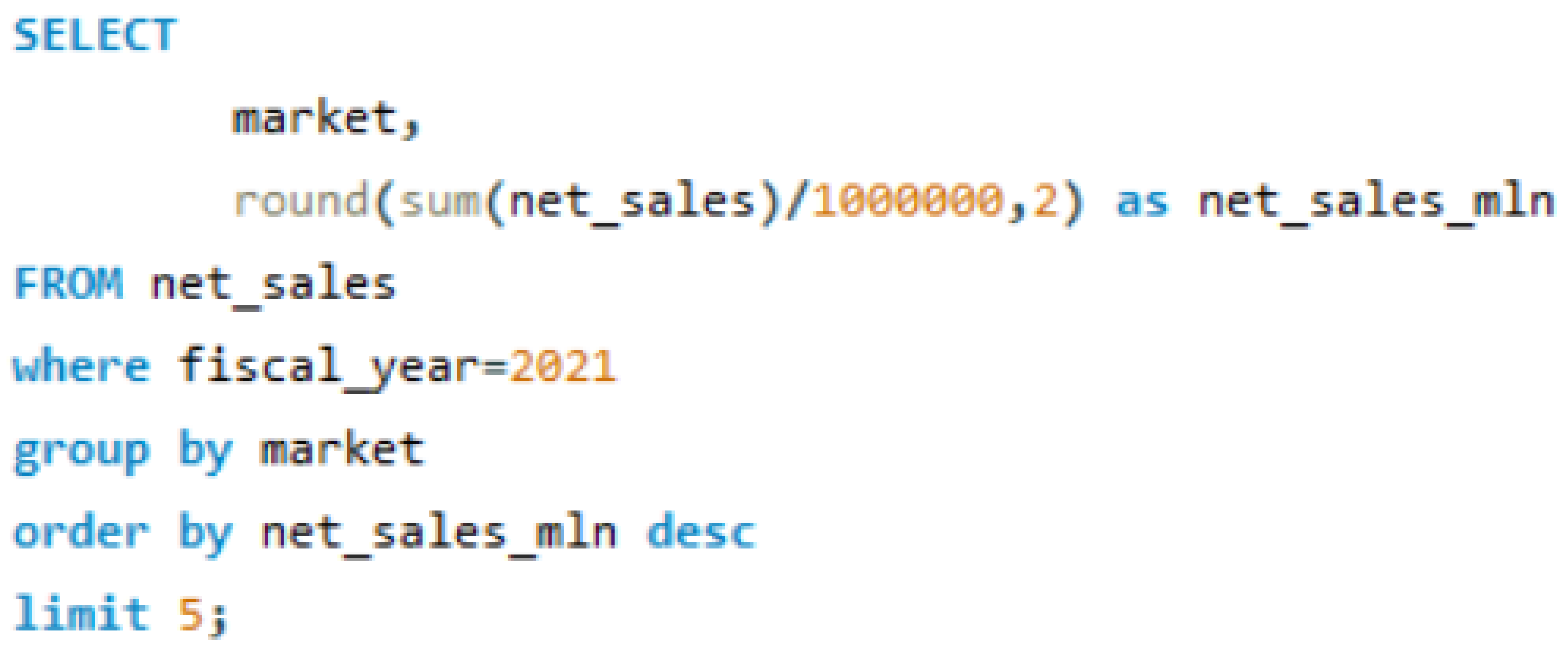
Step-4: Create a report for net sales



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

Report 4:

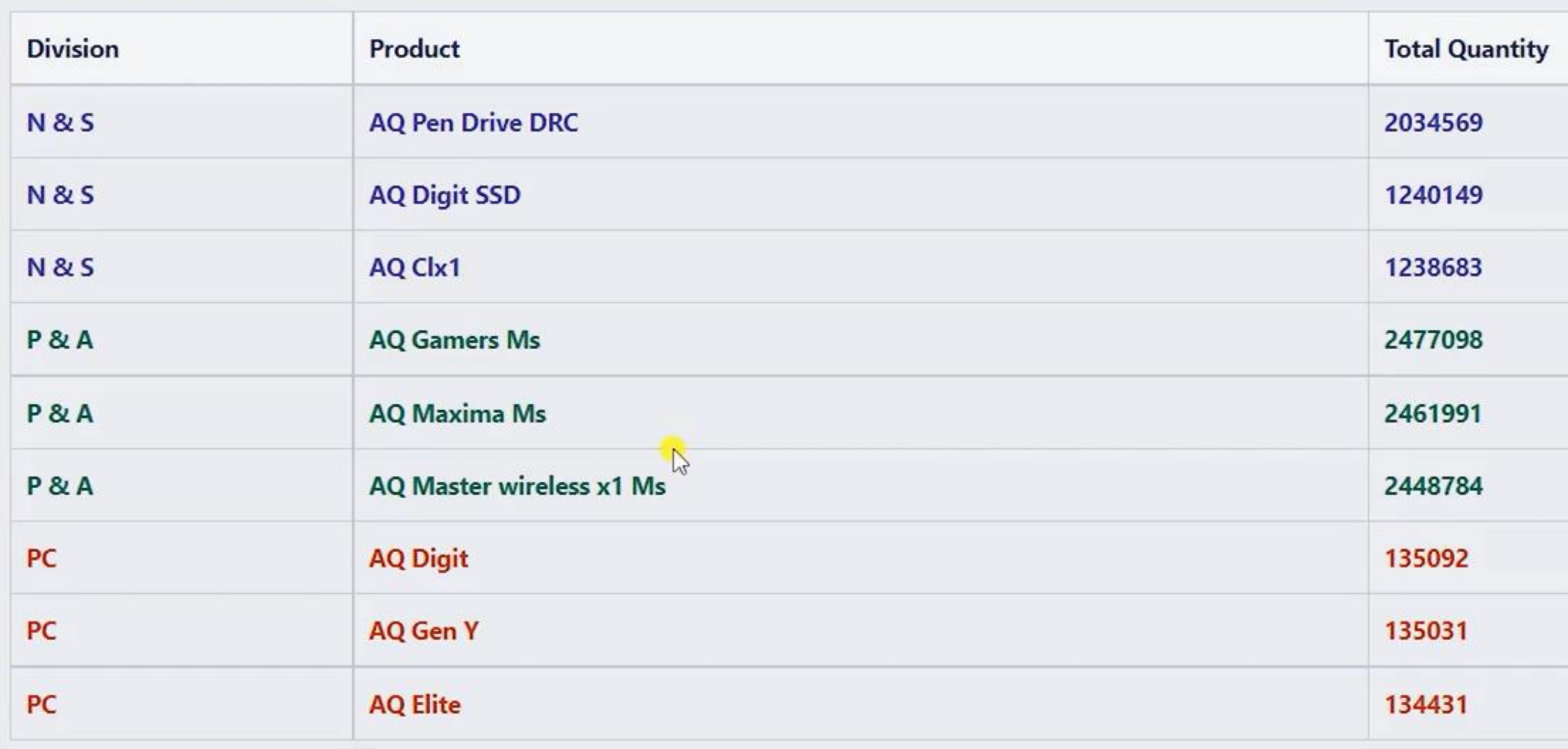
Step-5: Get top 5 market by net sales in fiscal year 2021



```
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;
```

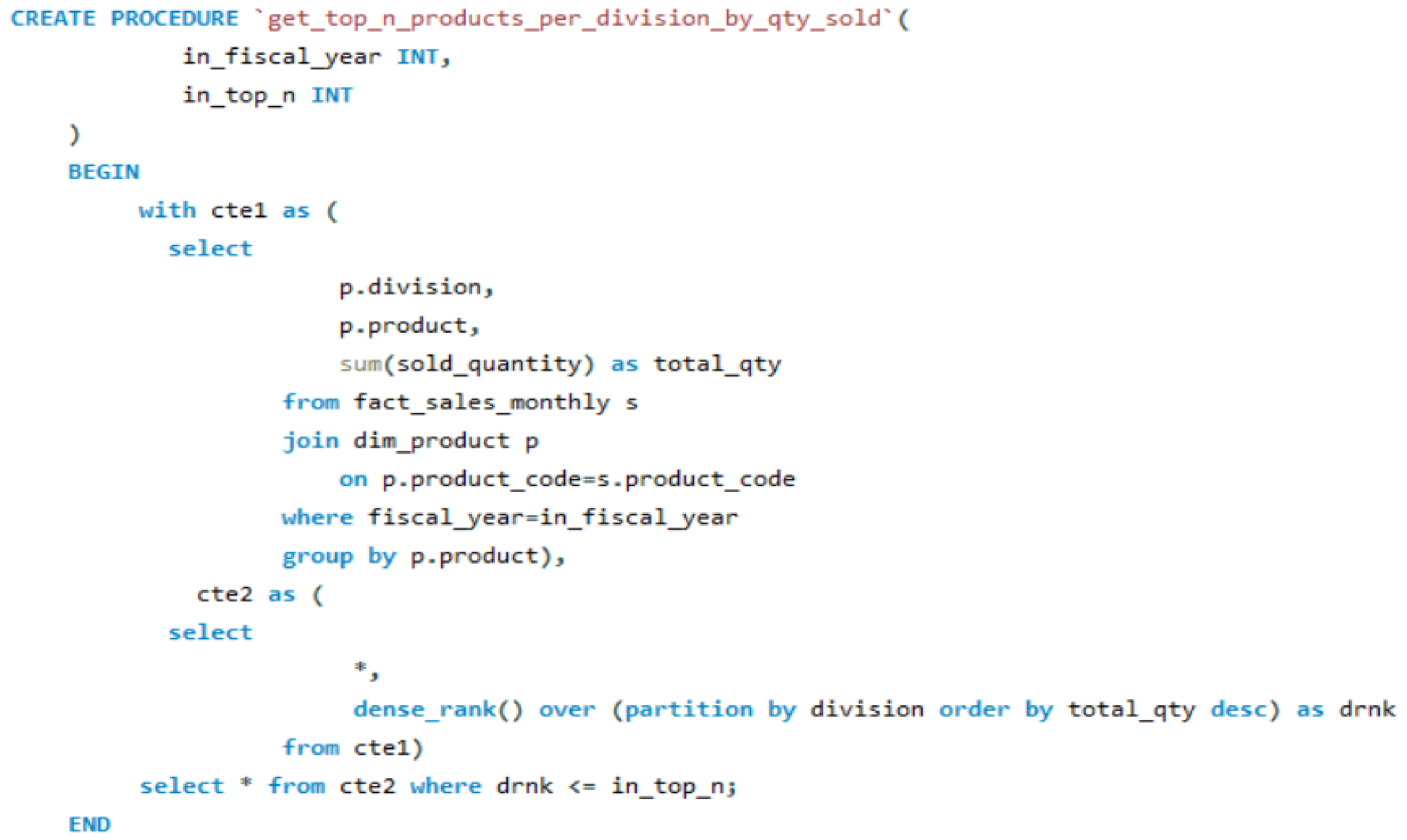
Report 5: Get top n products in each division by their quantity sold

Write a stored procedure for getting **top n** products in each division by their quantity sold in a given fiscal year. For example below would be the result for **FY = 2021**



Division	Product	Total Quantity
N & S	AQ Pen Drive DRC	2034569
N & S	AQ Digit SSD	1240149
N & S	AQ Clx1	1238683
P & A	AQ Gamers Ms	2477098
P & A	AQ Maxima Ms	2461991
P & A	AQ Master wireless x1 Ms	2448784
PC	AQ Digit	135092
PC	AQ Gen Y	135031
PC	AQ Elite	134431

Report 5:



```
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 fact_sales_monthly s
                   join dim_product p
                       on p.product_code=s.product_code
                   where fiscal_year=in_fiscal_year
                   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 <= in_top_n;
    END
```

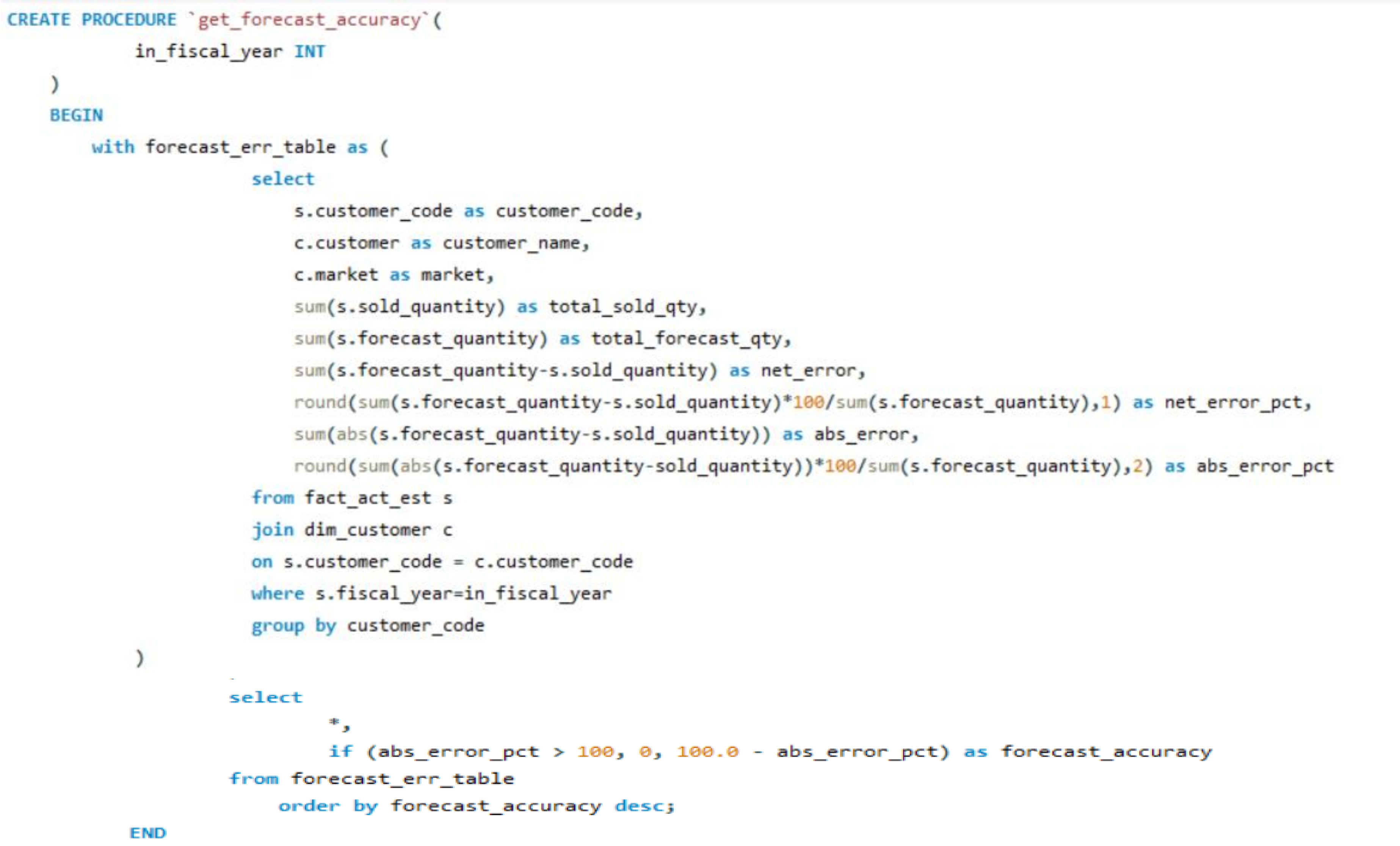
Report 6: Forecast Accuracy for all customers for a given fiscal year

As a product owner, I need an aggregate forecast accuracy report for all customers for a given fiscal year so that I can track the accuracy of the forecast we make for these customers.

The report should have the following fields,

1. Customer code, name, market
2. Total sold quantity
3. Total Forecast Quantity
4. Net Error
5. Absolute Error
6. Forecast Accuracy %

Report 6:



```
CREATE PROCEDURE `get_forecast_accuracy`(
            in_fiscal_year INT
    )
    BEGIN
        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=in_fiscal_year
                       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;
                END
```