AtliQ Hardwares: Finance & Supply Chain Analysis

A Comprehensive Finance and Supply Chain Data Analysis Project Using SQL

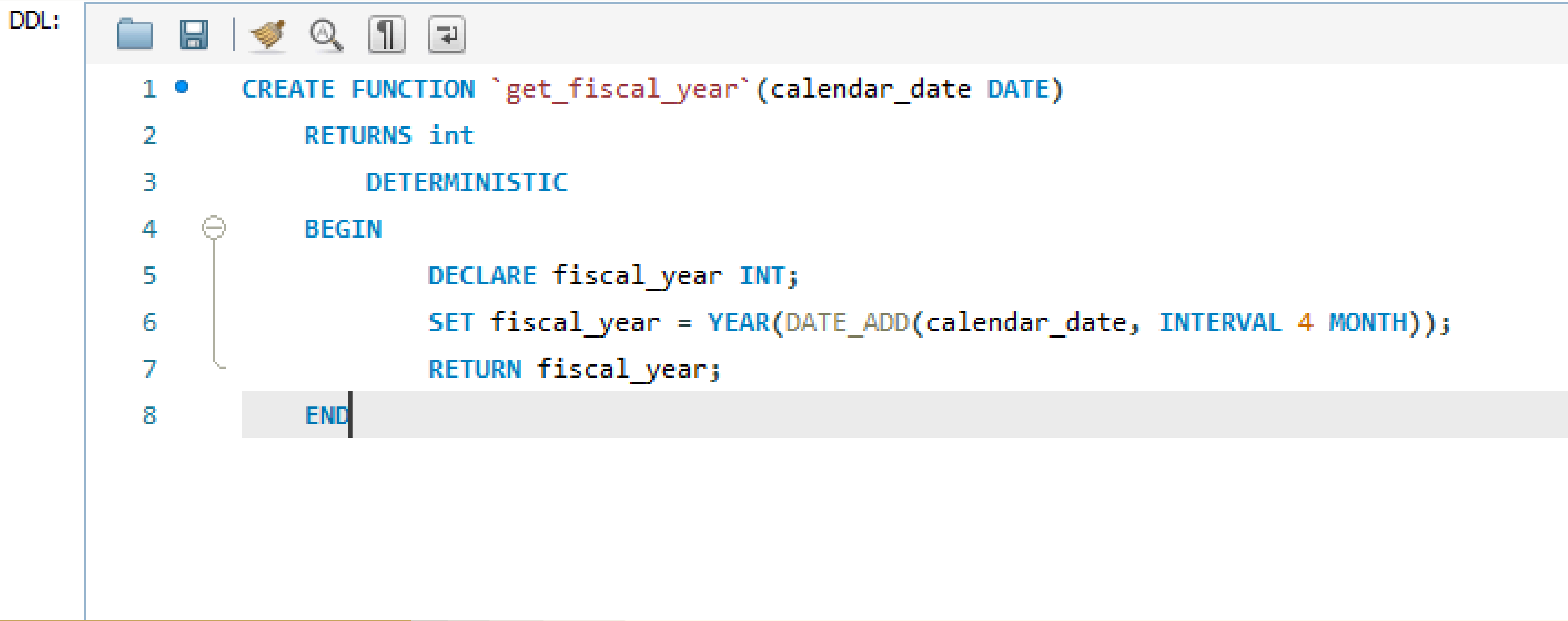
📊 Built with: SQL ⚙️ Industry: Hardware

🧠 Client: AtliQ Hardwares ✅ Built by: Vivek M

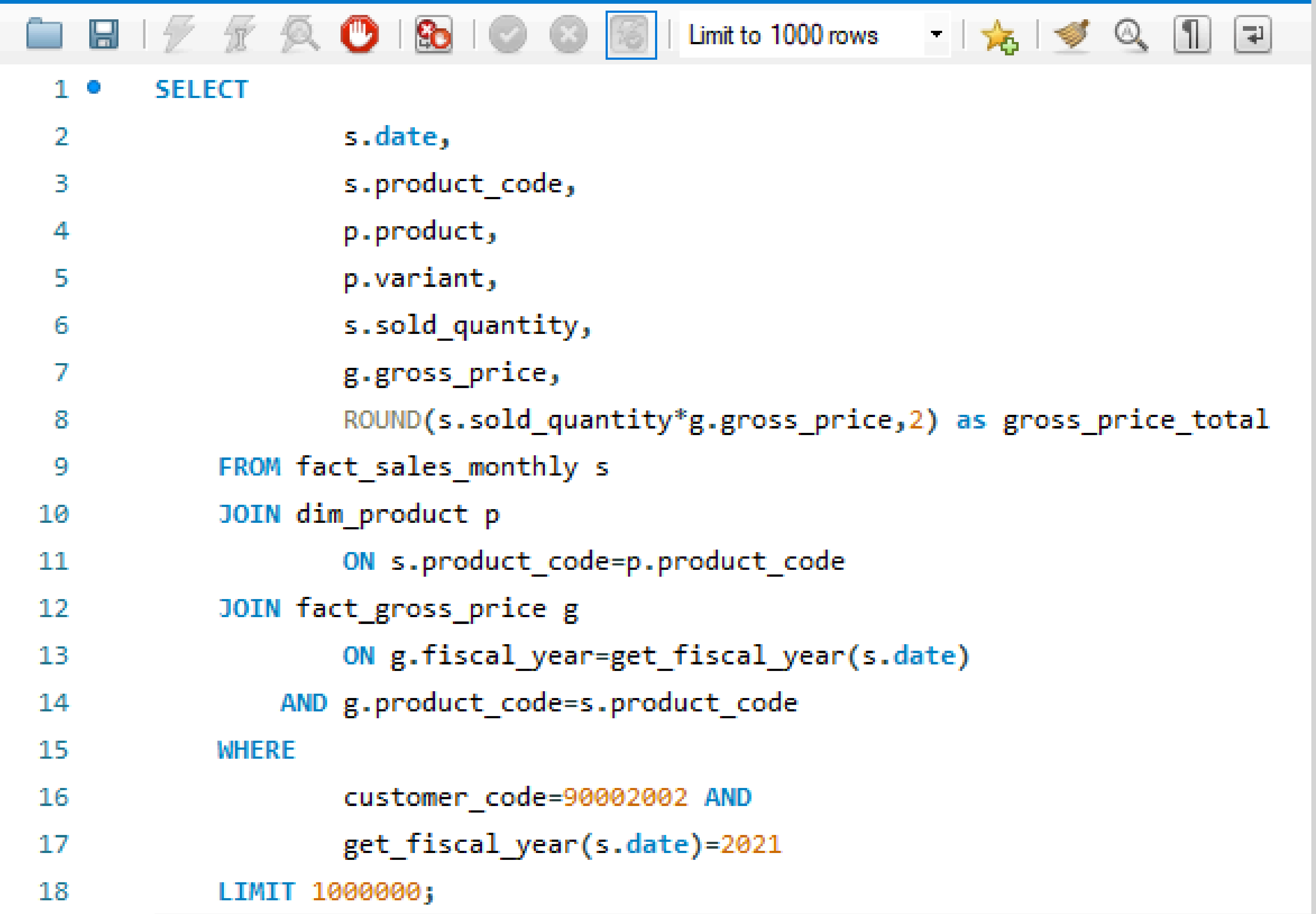
Finance View

PROBLEM-1. Generate a product-wise monthly sales report for Croma India for the fiscal year 2021 .

1.User Defined Funtion- “get_fiscal_year”



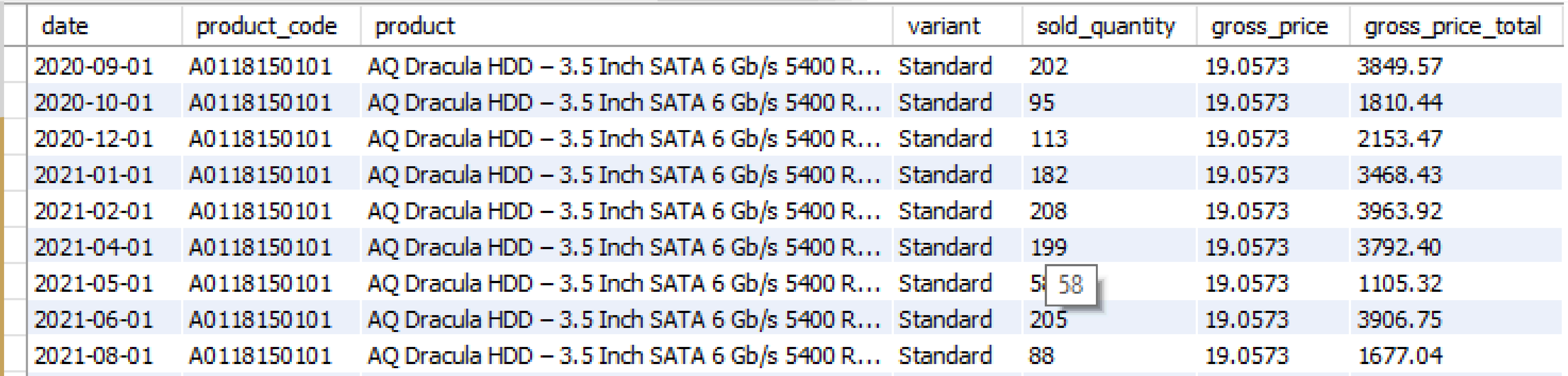
```
CREATE FUNCTION `get_fiscal_year`(calendar_date DATE)
    RETURNS int
        DETERMINISTIC
    BEGIN
            DECLARE fiscal_year INT;
            SET fiscal_year = YEAR(DATE_ADD(calendar_date, INTERVAL 4 MONTH));
            RETURN fiscal_year;
    END
```



```sql
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.fiscal_year=get_fiscal_year(s.date)
        AND g.product_code=s.product_code
    WHERE
            customer_code=90002002 AND
            get_fiscal_year(s.date)=2021
    LIMIT 1000000;
```

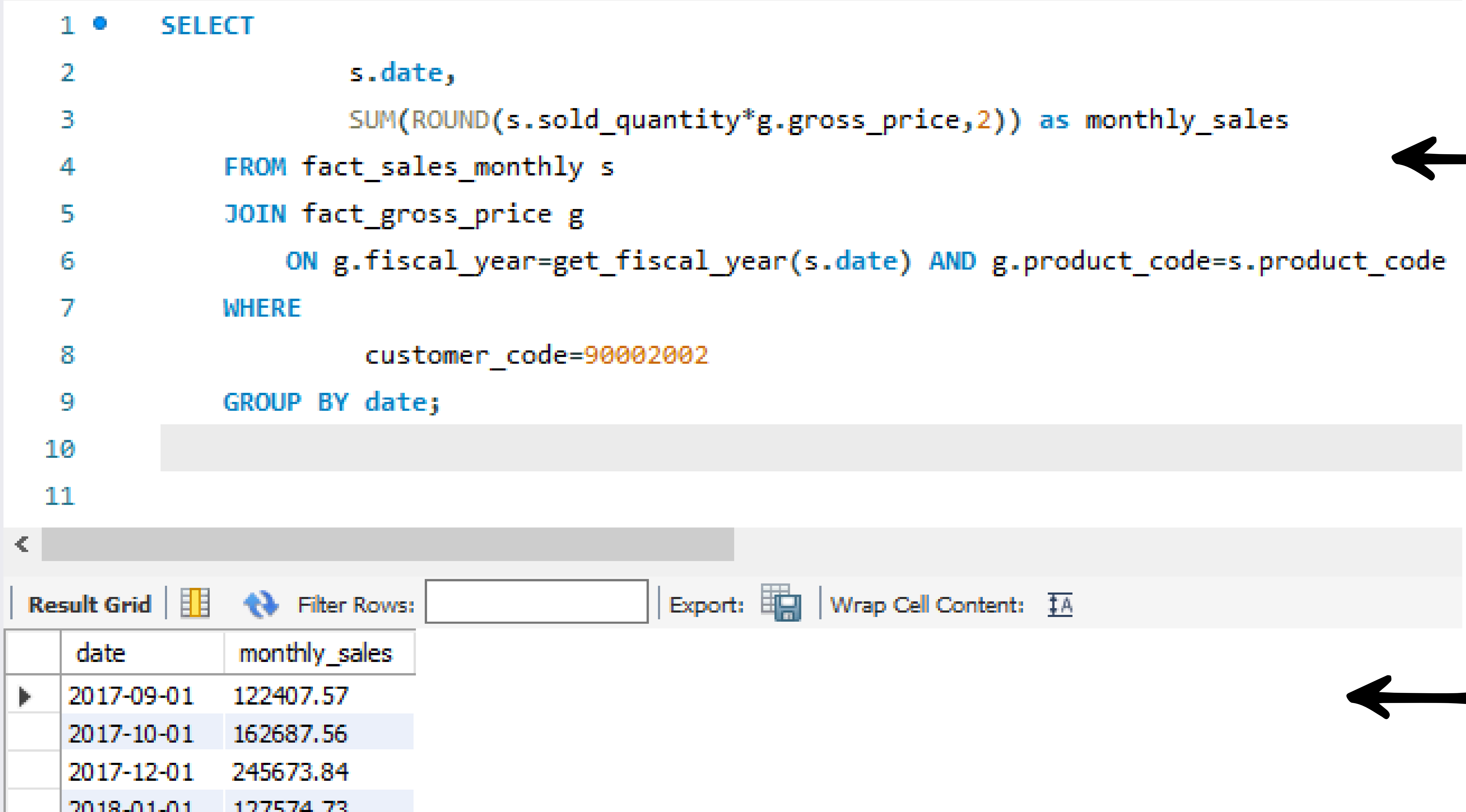
SQL Query

Result



date	product_code	product	variant	sold_quantity	gross_price	gross_price_total
2020-09-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	202	19.0573	3849.57
2020-10-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	95	19.0573	1810.44
2020-12-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	113	19.0573	2153.47
2021-01-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	182	19.0573	3468.43
2021-02-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	208	19.0573	3963.92
2021-04-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	199	19.0573	3792.40
2021-05-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	58	19.0573	1105.32
2021-06-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	205	19.0573	3906.75
2021-08-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	88	19.0573	1677.04

PROBLEM-2. Generate monthly gross sales report for Croma India for all the years



```sql
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
       customer_code=90002002
  GROUP BY date;
```

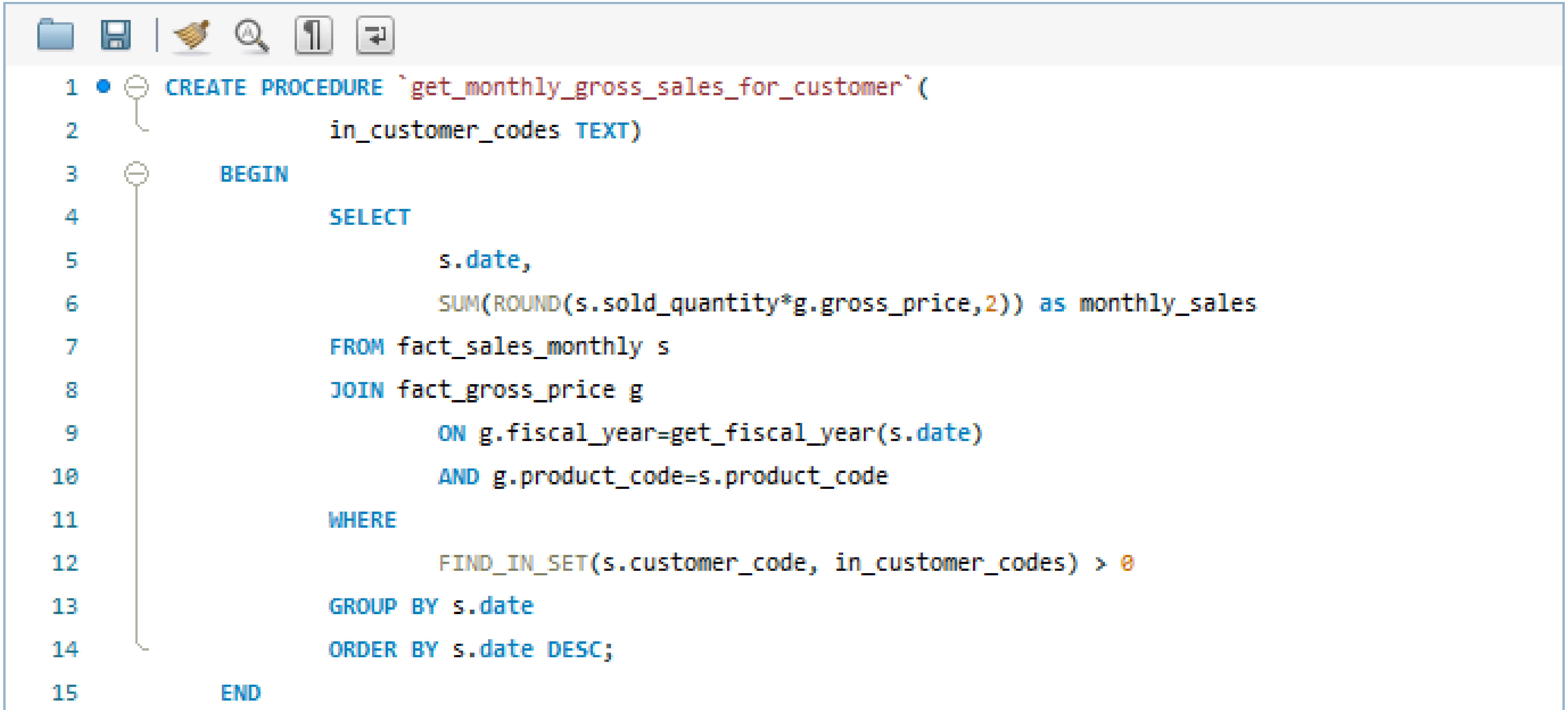
SQL Query

date	monthly_sales
2017-09-01	122407.57
2017-10-01	162687.56
2017-12-01	245673.84
2018-01-01	127574.73

Result

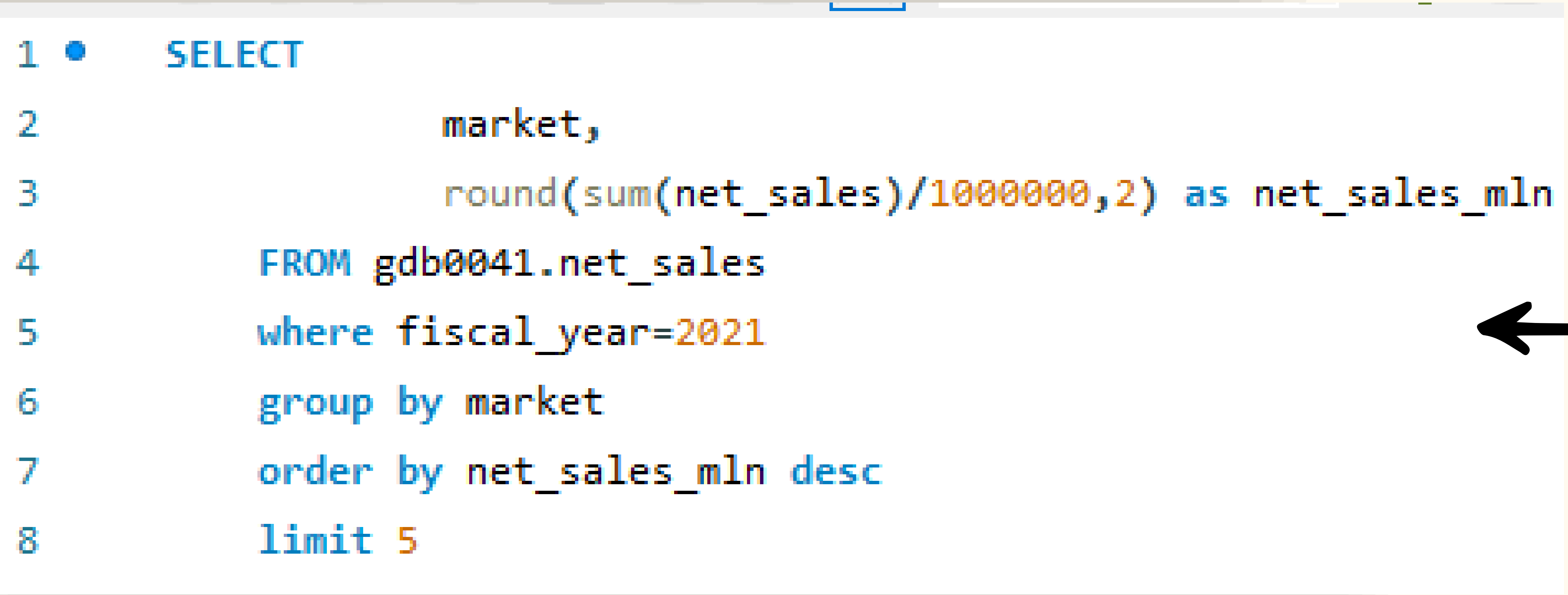
PROBLEM-3. Generate monthly gross sales report for any customer using stored procedure

Stored Procedure



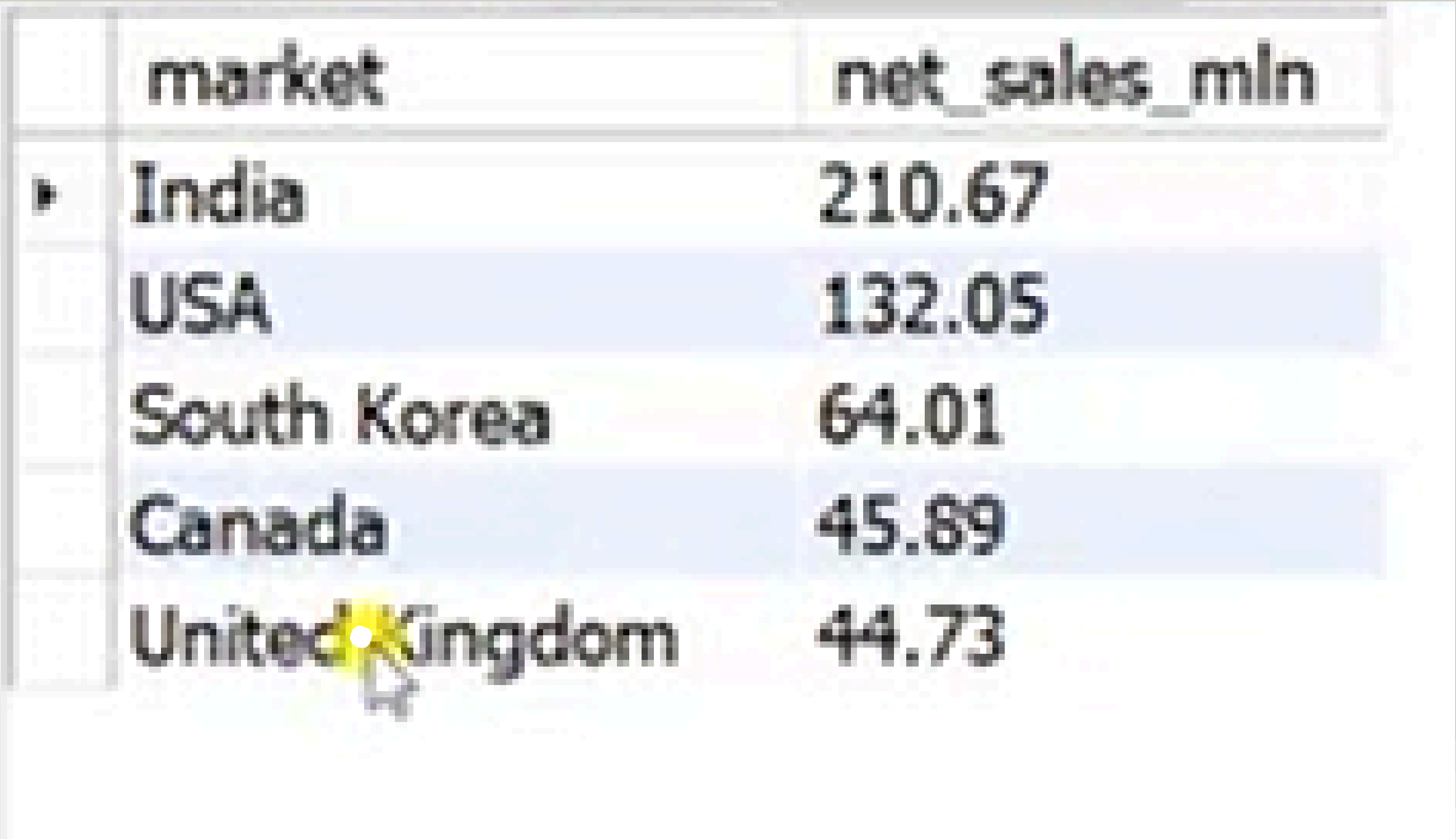
```sql
CREATE PROCEDURE `get_monthly_gross_sales_for_customer`(
            in_customer_codes TEXT)
    BEGIN
            SELECT
                    s.date,
                    SUM(ROUND(s.sold_quantity*g.gross_price,2)) as monthly_sales
            FROM fact_sales_monthly s
            JOIN fact_gross_price 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
```

PROBLEM-4. Get top 5 Market by net sales in 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
```

SQL Query

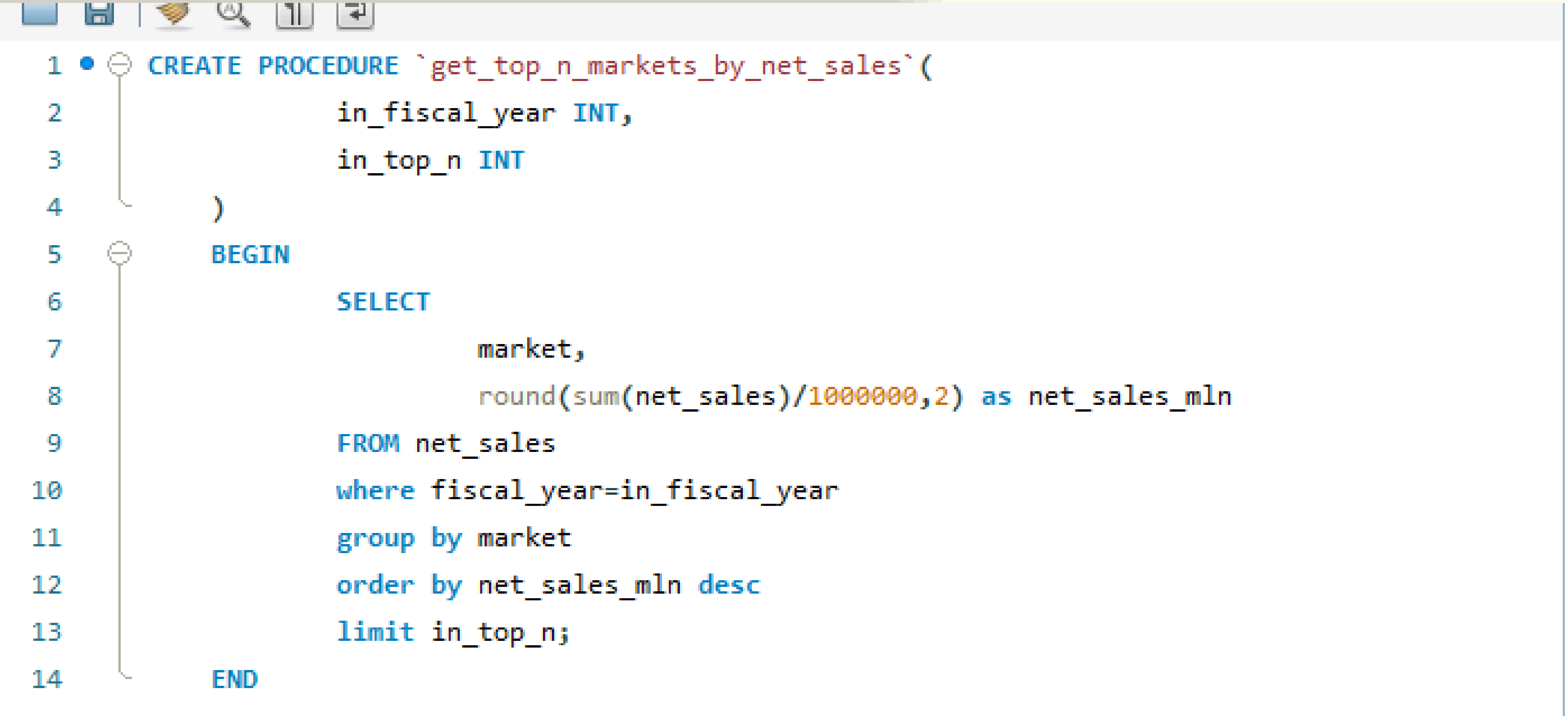


market	net_sales_mln
India	210.67
USA	132.05
South Korea	64.01
Canada	45.89
United Kingdom	44.73

Result

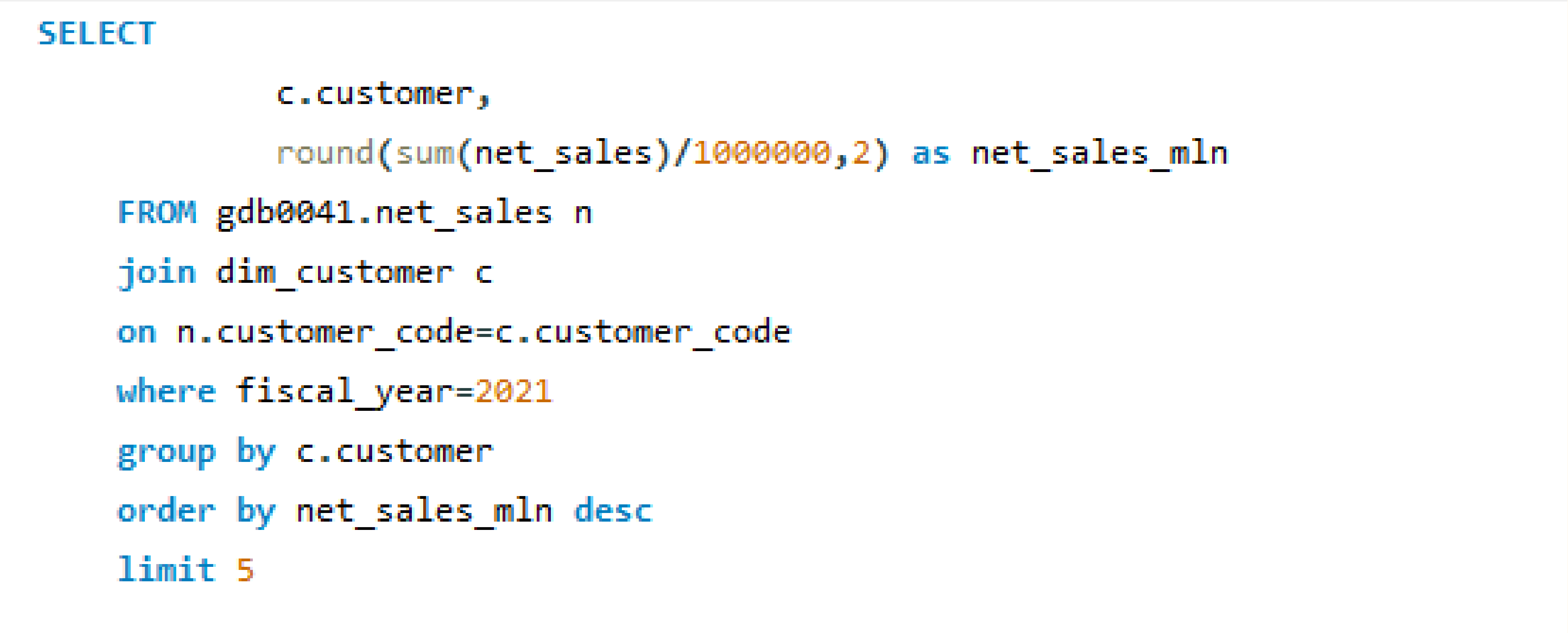
PROBLEM-5. Stored procedure to get top n markets by net sales for a given year

Stored Procedure



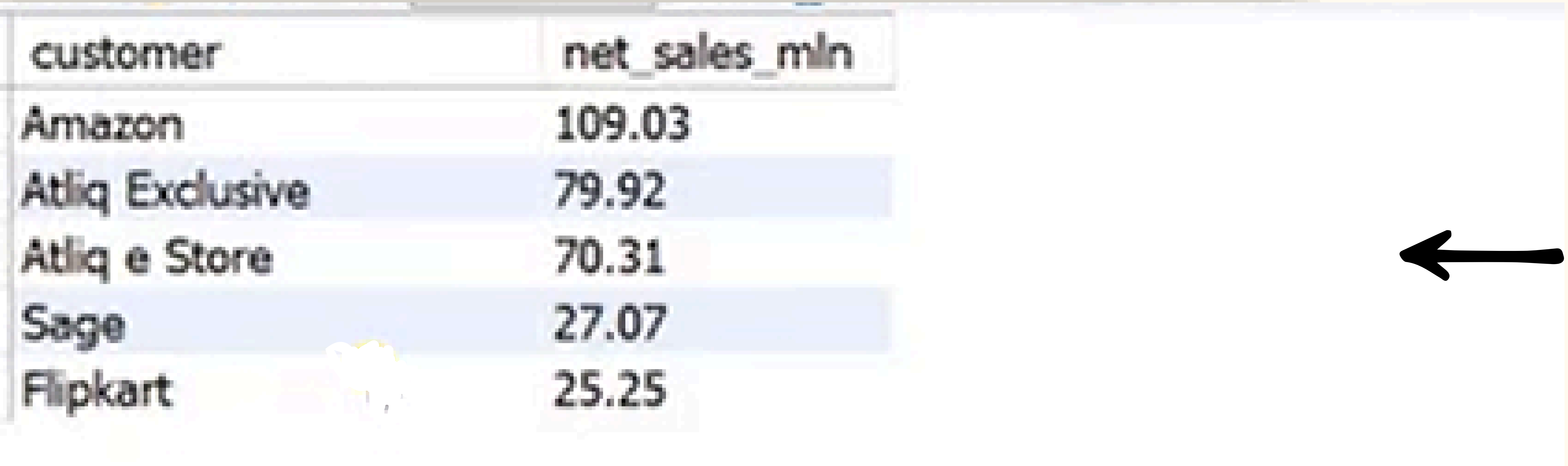
```sql
CREATE PROCEDURE `get_top_n_markets_by_net_sales`(
            in_fiscal_year INT,
            in_top_n INT
    )
    BEGIN
            SELECT
                     market,
                     round(sum(net_sales)/1000000,2) as net_sales_mln
            FROM net_sales
            where fiscal_year=in_fiscal_year
            group by market
            order by net_sales_mln desc
            limit in_top_n;
    END
```

PROBLEM-6. Get top 5 Customers by net sales in fiscal year 2021



```
SELECT
            c.customer,
            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
    order by net_sales_mln desc
    limit 5
```

SQL Query

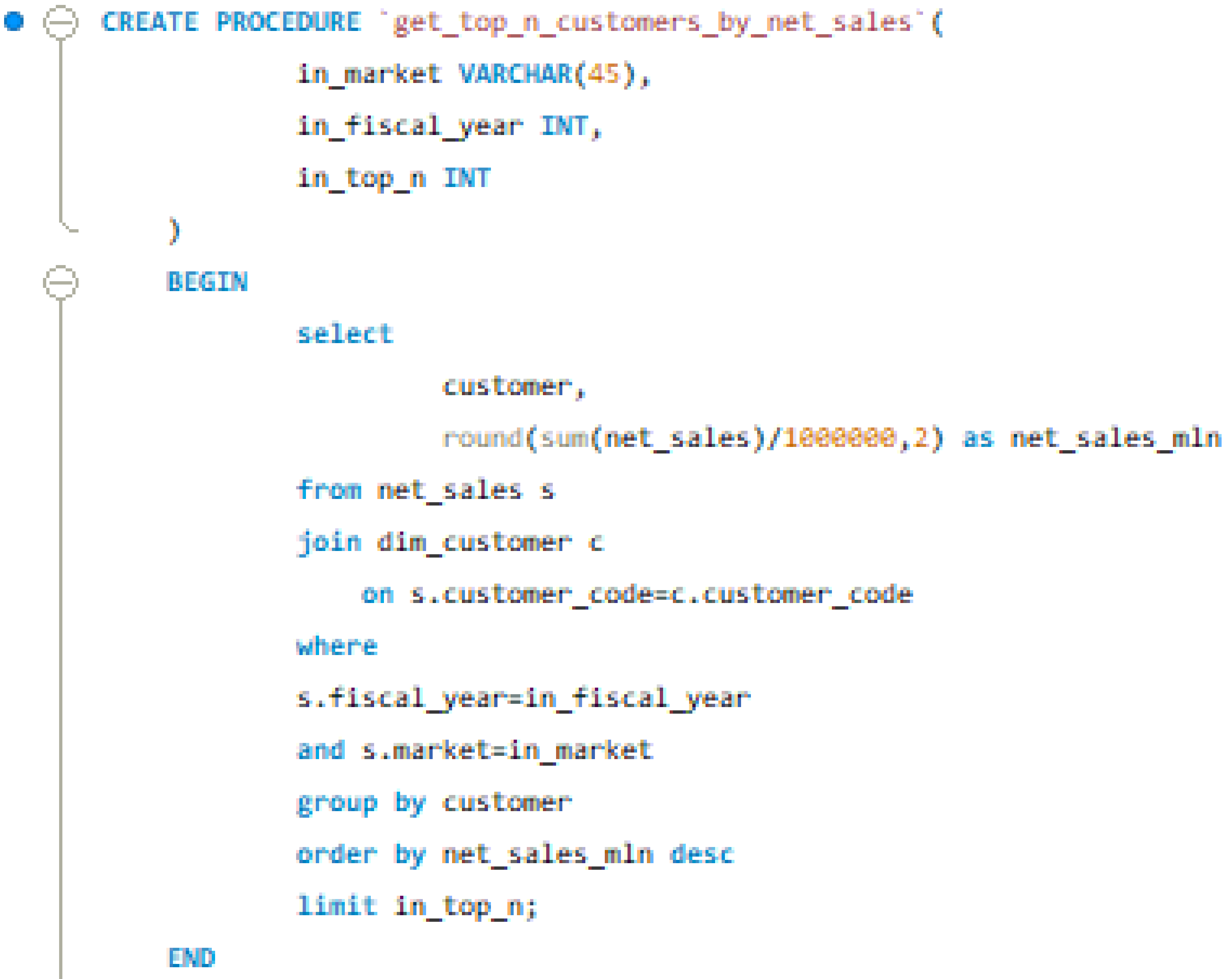


customer	net_sales_mln
Amazon	109.03
Atliq Exclusive	79.92
Atliq e Store	70.31
Sage	27.07
Flipkart	25.25

Result

PROBLEM-7. Stored procedure to get top n Customers by net sales for a given year

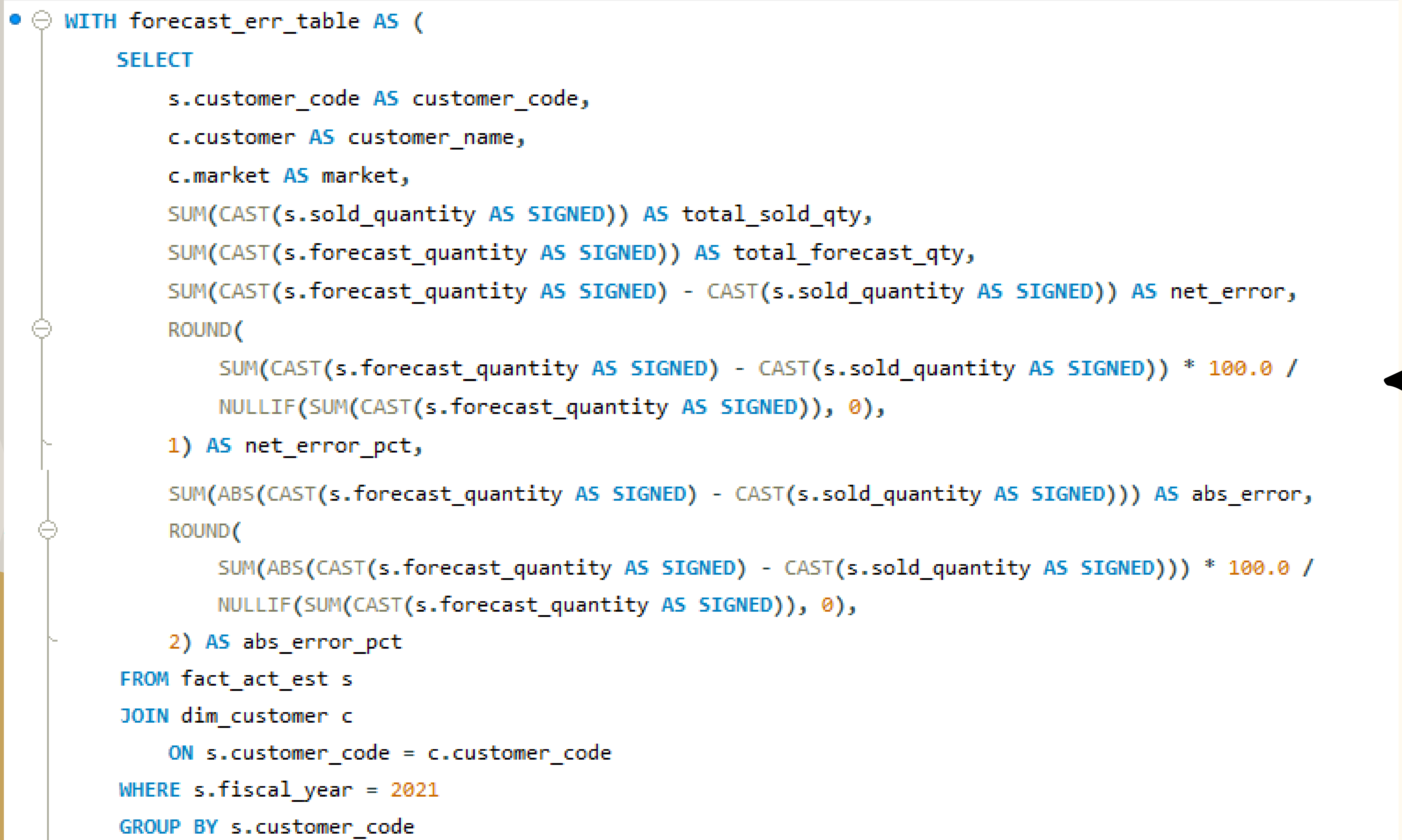
Stored Procedure



```
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
```

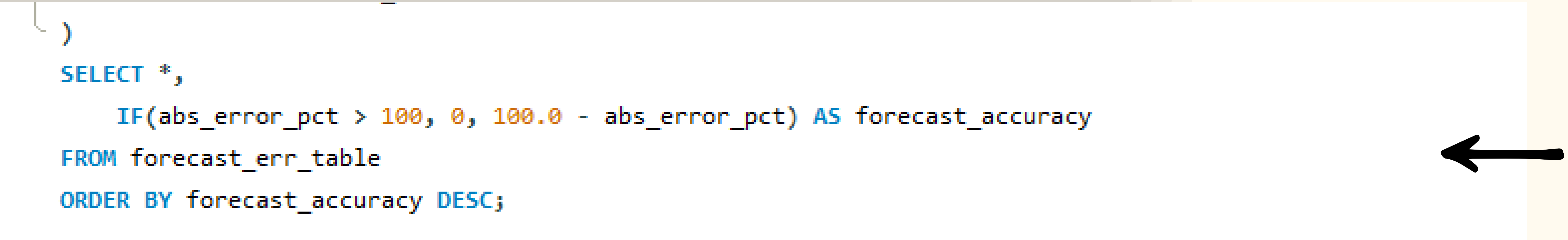
Supply Chain View

PROBLEM-8. Forecast accuracy report using CTE



```sql
WITH forecast_err_table AS (
    SELECT
        s.customer_code AS customer_code,
        c.customer AS customer_name,
        c.market AS market,
        SUM(CAST(s.sold_quantity AS SIGNED)) AS total_sold_qty,
        SUM(CAST(s.forecast_quantity AS SIGNED)) AS total_forecast_qty,
        SUM(CAST(s.forecast_quantity AS SIGNED) - CAST(s.sold_quantity AS SIGNED)) AS net_error,
        ROUND(
            SUM(CAST(s.forecast_quantity AS SIGNED) - CAST(s.sold_quantity AS SIGNED)) * 100.0 /
            NULLIF(SUM(CAST(s.forecast_quantity AS SIGNED)), 0),
        1) AS net_error_pct,
        SUM(ABS(CAST(s.forecast_quantity AS SIGNED) - CAST(s.sold_quantity AS SIGNED))) AS abs_error,
        ROUND(
            SUM(ABS(CAST(s.forecast_quantity AS SIGNED) - CAST(s.sold_quantity AS SIGNED))) * 100.0 /
            NULLIF(SUM(CAST(s.forecast_quantity AS SIGNED)), 0),
        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 s.customer_code
```

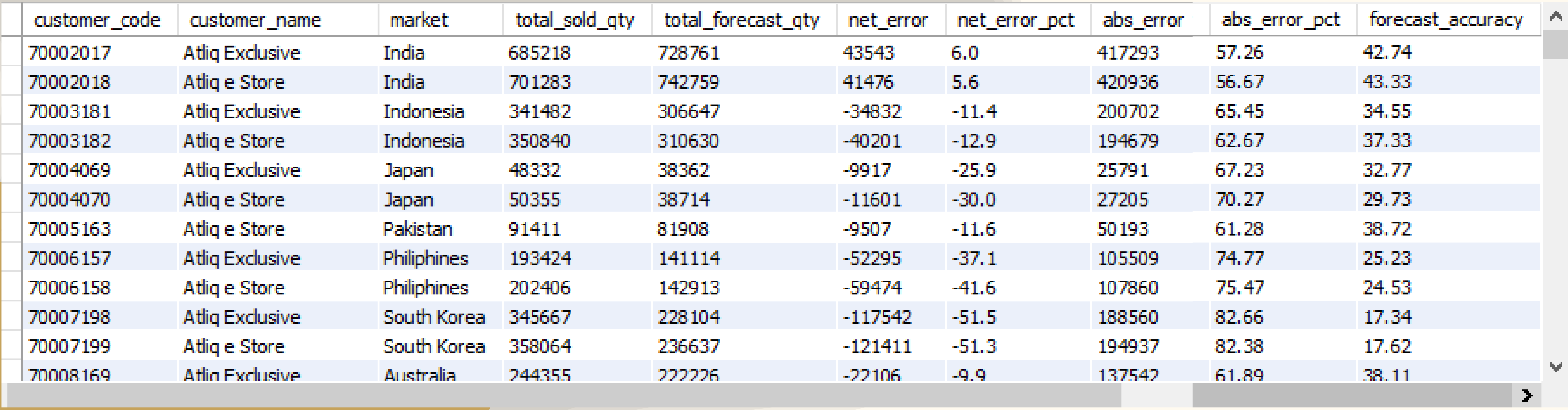
← SQL Query



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

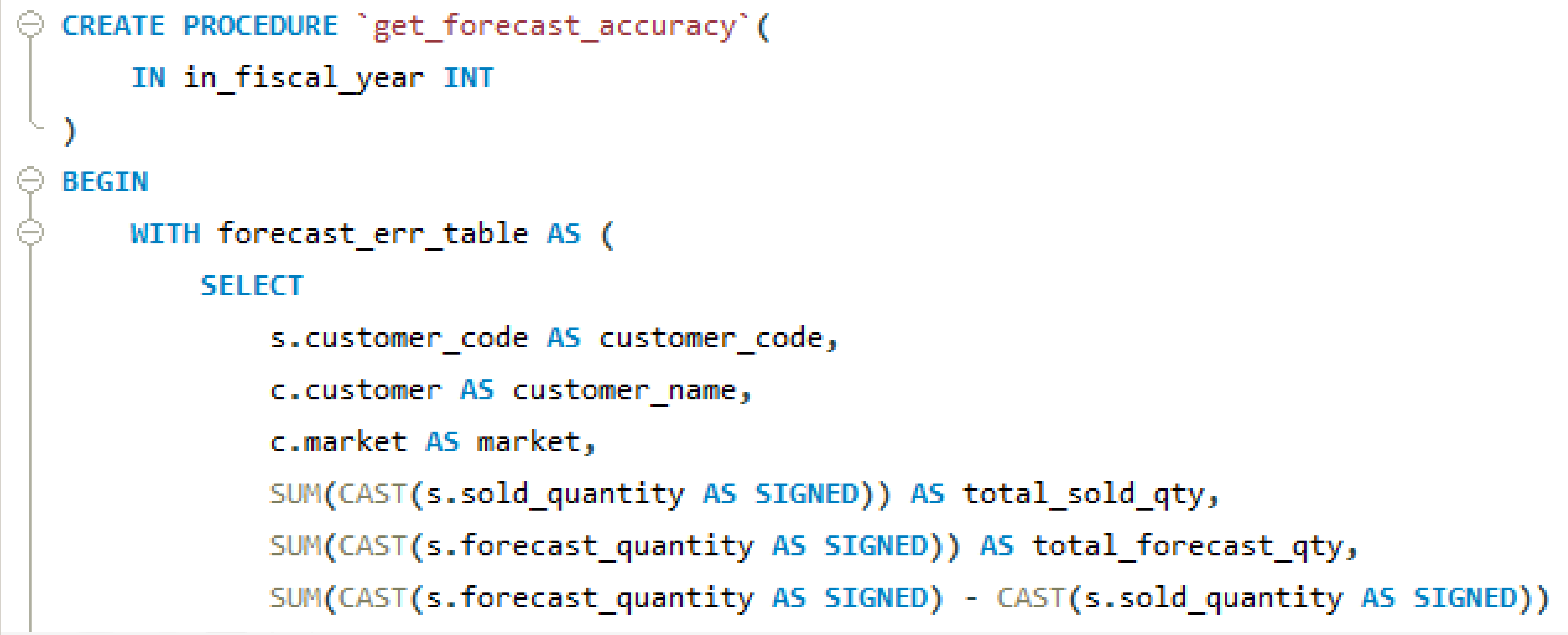
SQL Query

Result

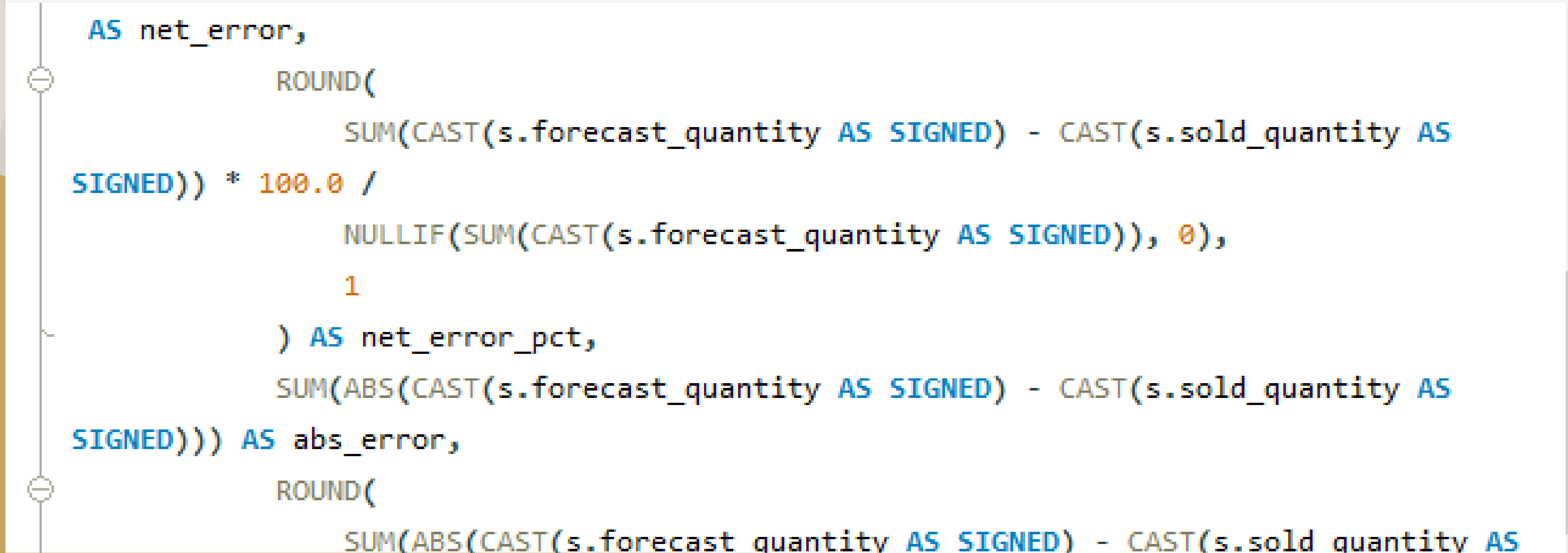


customer_code	customer_name	market	total_sold_qty	total_forecast_qty	net_error	net_error_pct	abs_error	abs_error_pct	forecast_accuracy
70002017	Atliq Exclusive	India	685218	728761	43543	6.0	417293	57.26	42.74
70002018	Atliq e Store	India	701283	742759	41476	5.6	420936	56.67	43.33
70003181	Atliq Exclusive	Indonesia	341482	306647	-34832	-11.4	200702	65.45	34.55
70003182	Atliq e Store	Indonesia	350840	310630	-40201	-12.9	194679	62.67	37.33
70004069	Atliq Exclusive	Japan	48332	38362	-9917	-25.9	25791	67.23	32.77
70004070	Atliq e Store	Japan	50355	38714	-11601	-30.0	27205	70.27	29.73
70005163	Atliq e Store	Pakistan	91411	81908	-9507	-11.6	50193	61.28	38.72
70006157	Atliq Exclusive	Philiphines	193424	141114	-52295	-37.1	105509	74.77	25.23
70006158	Atliq e Store	Philiphines	202406	142913	-59474	-41.6	107860	75.47	24.53
70007198	Atliq Exclusive	South Korea	345667	228104	-117542	-51.5	188560	82.66	17.34
70007199	Atliq e Store	South Korea	358064	236637	-121411	-51.3	194937	82.38	17.62
70008169	Atliq Exclusive	Australia	244355	222226	-22106	-9.9	137542	61.89	38.11

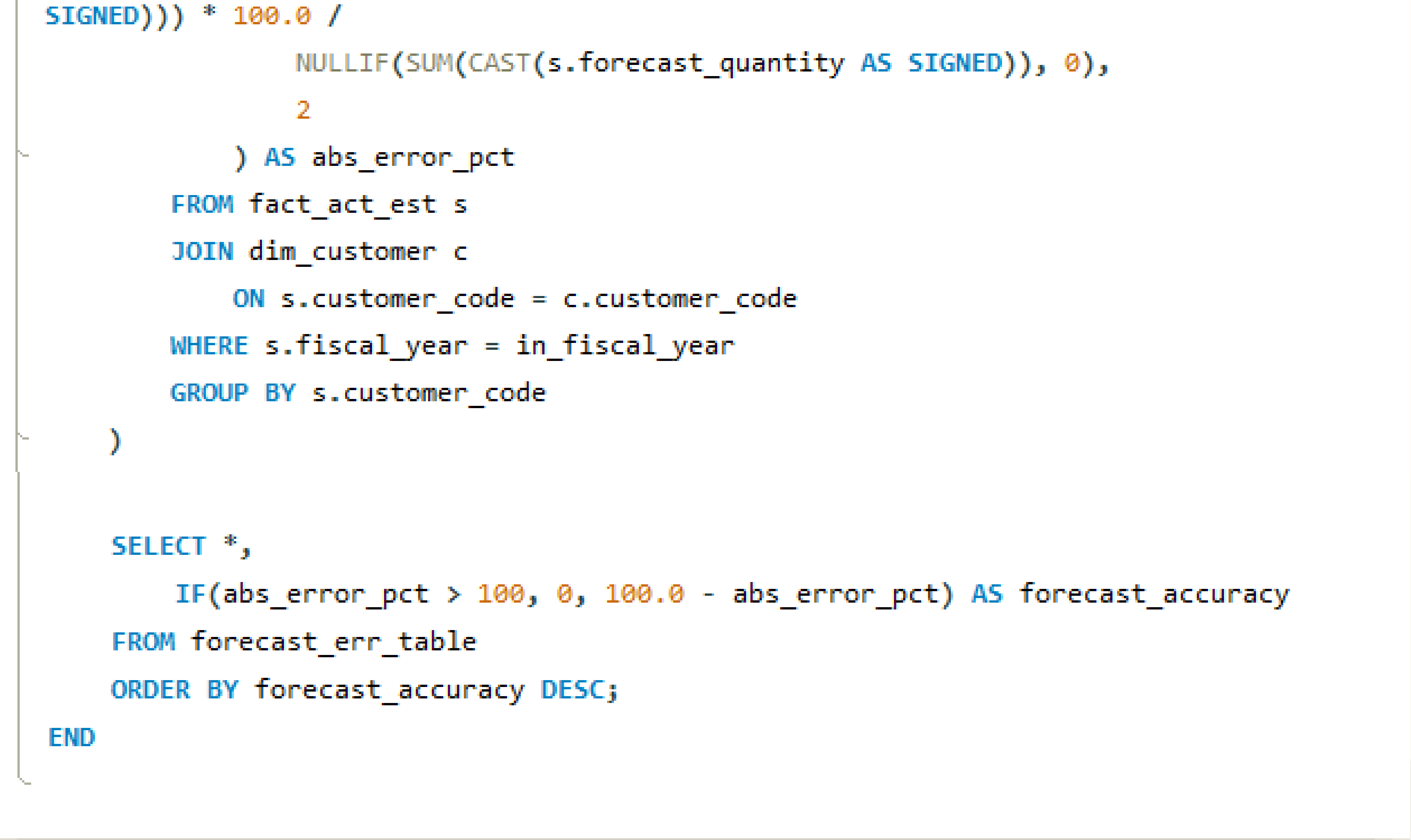
PROBLEM-9. Write a stored procedure for the Forecast accuracy report



```
CREATE PROCEDURE `get_forecast_accuracy`(
    IN 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(CAST(s.sold_quantity AS SIGNED)) AS total_sold_qty,
            SUM(CAST(s.forecast_quantity AS SIGNED)) AS total_forecast_qty,
            SUM(CAST(s.forecast_quantity AS SIGNED) - CAST(s.sold_quantity AS SIGNED))
```



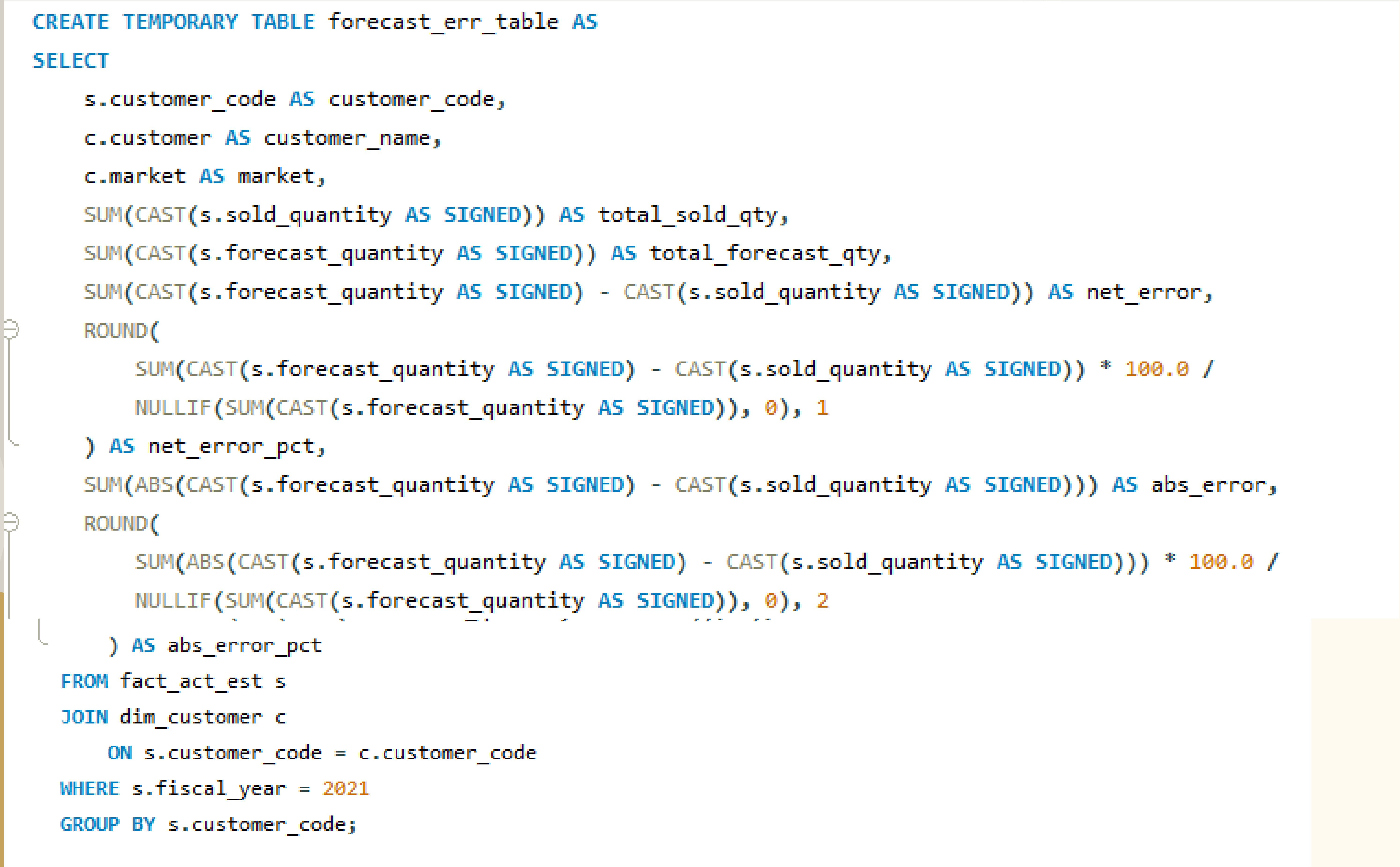
```
 AS net_error,
            ROUND(
                SUM(CAST(s.forecast_quantity AS SIGNED) - CAST(s.sold_quantity AS
SIGNED)) * 100.0 /
                NULLIF(SUM(CAST(s.forecast_quantity AS SIGNED)), 0),
                1
            ) AS net_error_pct,
            SUM(ABS(CAST(s.forecast_quantity AS SIGNED) - CAST(s.sold_quantity AS
SIGNED))) AS abs_error,
            ROUND(
                SUM(ABS(CAST(s.forecast_quantity AS SIGNED) - CAST(s.sold_quantity AS
```



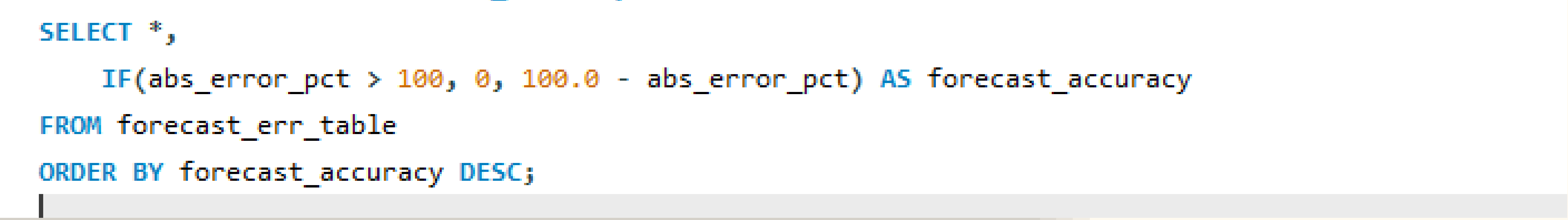
```
SIGNED))) * 100.0 /
                NULLIF(SUM(CAST(s.forecast_quantity AS SIGNED)), 0),
                2
            ) AS abs_error_pct
        FROM fact_act_est s
        JOIN dim_customer c
            ON s.customer_code = c.customer_code
        WHERE s.fiscal_year = in_fiscal_year
        GROUP BY s.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
```

PROBLEM-10. Forecast accuracy report using Temporary table



```sql
CREATE TEMPORARY TABLE forecast_err_table AS
SELECT
    s.customer_code AS customer_code,
    c.customer AS customer_name,
    c.market AS market,
    SUM(CAST(s.sold_quantity AS SIGNED)) AS total_sold_qty,
    SUM(CAST(s.forecast_quantity AS SIGNED)) AS total_forecast_qty,
    SUM(CAST(s.forecast_quantity AS SIGNED) - CAST(s.sold_quantity AS SIGNED)) AS net_error,
    ROUND(
        SUM(CAST(s.forecast_quantity AS SIGNED) - CAST(s.sold_quantity AS SIGNED)) * 100.0 /
        NULLIF(SUM(CAST(s.forecast_quantity AS SIGNED)), 0), 1
    ) AS net_error_pct,
    SUM(ABS(CAST(s.forecast_quantity AS SIGNED) - CAST(s.sold_quantity AS SIGNED))) AS abs_error,
    ROUND(
        SUM(ABS(CAST(s.forecast_quantity AS SIGNED) - CAST(s.sold_quantity AS SIGNED))) * 100.0 /
        NULLIF(SUM(CAST(s.forecast_quantity AS SIGNED)), 0), 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 s.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;
```

Thank You!