AtliQ Hardware

Consumer Goods

AD-HOC INSIGHTS

By Shelly Nagar

Agenda

- About Company
- Problem Statement
- Background & Problem Statement
- Queries

About AtliQ Hardware

Domain : Consumer Goods

Function : Executive Management

01 Product Categories:

- Peripherals & Accessories (P & A)
- Network & Storage (N & S)
- Personal Computers (PC)

02 Sales Channels:

- Retail: Physical stores like Croma, Best Buy
- Direct: AtliQ's own platforms (e.g., AtliQ e-store, AtliQ Exclusive) and other E-commerce platform like Amazon
- Distributor: Third-party distributors like Neptune

03 Customer Types:

- Bricks & Mortar: Physical stores (Croma, Best Buy)
- Online Platforms: Amazon, Flipkart, AtliQ e-store
- Third-Party Distributors: Neptune, etc.

04 Global Reach:

Operations in India and Overseas

Background

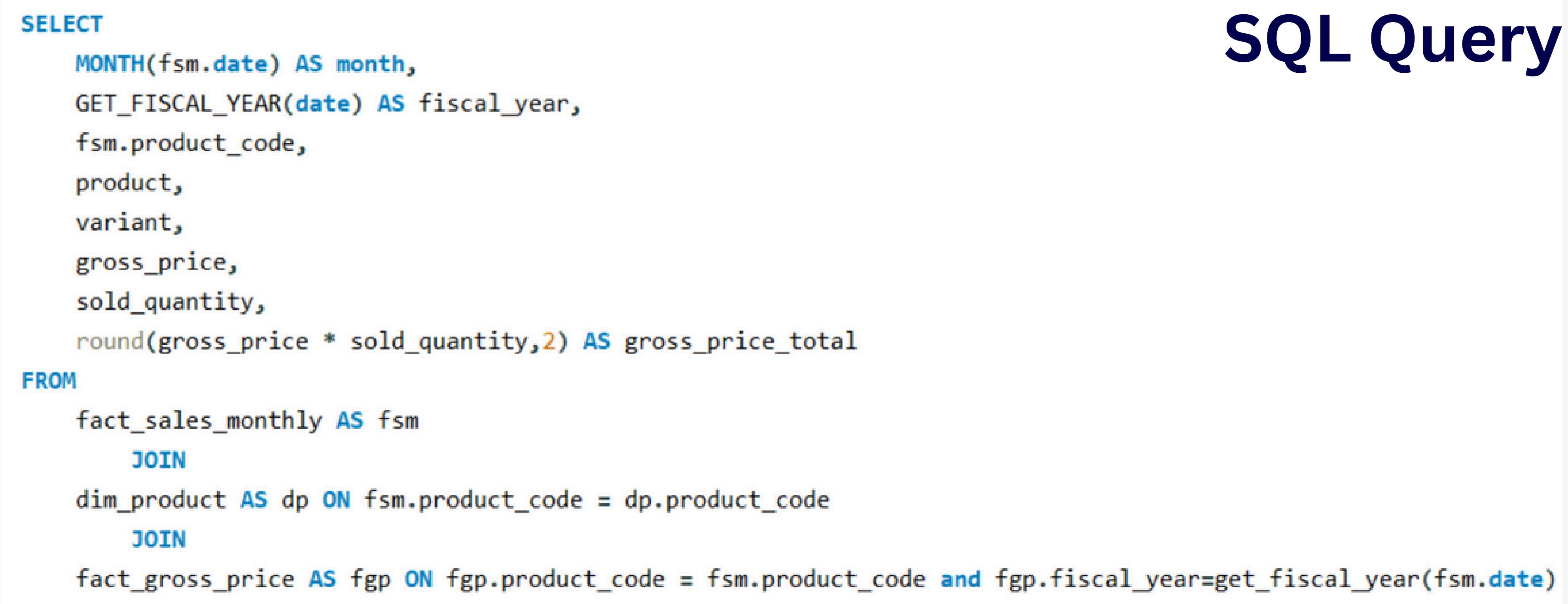
Management needs quick insights to make smart and data driven decision

Problem Statement

Write SQL Queries for Ad-Hoc requests from the management that require immediate insights

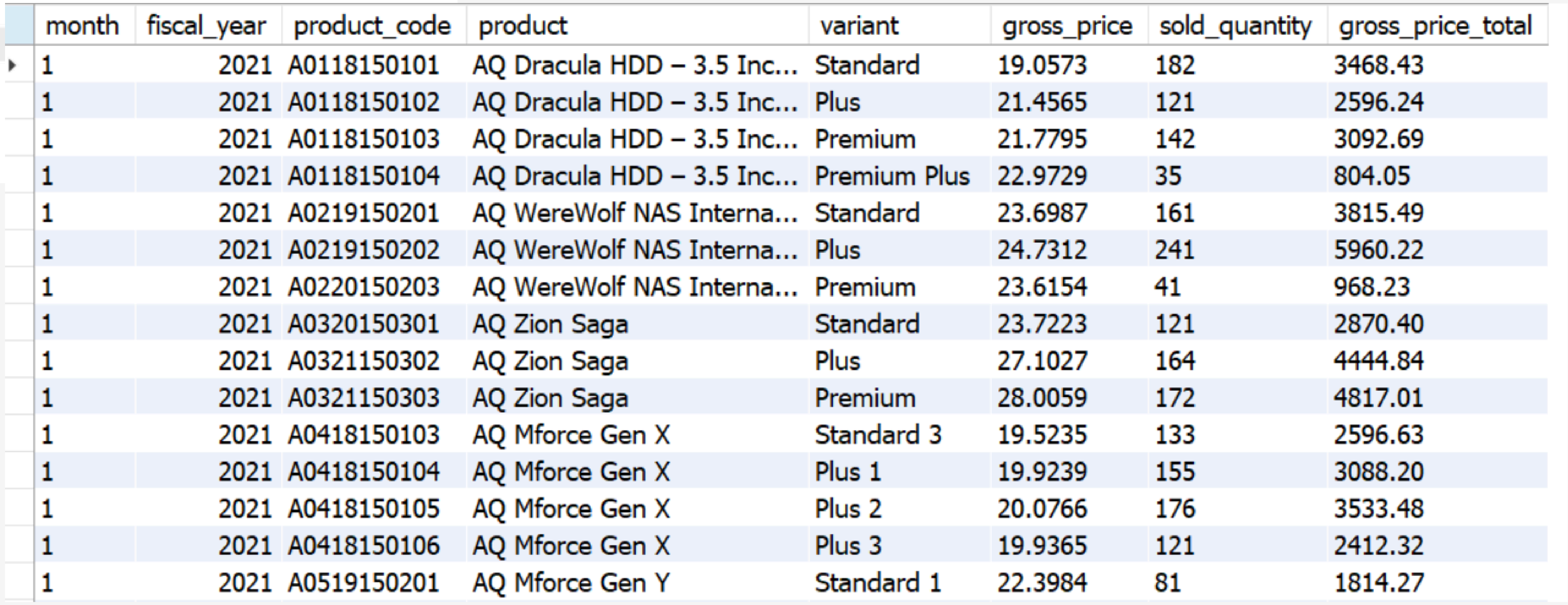
Generate a report of individual product sales (aggregated on a monthly basis at the product level) for Croma India customers for FY 2021

SQL Query



```
SELECT
    MONTH(fsm.date) AS month,
    GET_FISCAL_YEAR(date) AS fiscal_year,
    fsm.product_code,
    product,
    variant,
    gross_price,
    sold_quantity,
    round(gross_price * sold_quantity,2) AS gross_price_total
FROM
    fact_sales_monthly AS fsm
        JOIN
    dim_product AS dp ON fsm.product_code = dp.product_code
        JOIN
    fact_gross_price AS fgp ON fgp.product_code = fsm.product_code and fgp.fiscal_year=get_fiscal_year(fsm.date)
WHERE
    customer_code = 90002002 -- for croma india
        AND GET_FISCAL_YEAR(date) = 2021
ORDER BY month
LIMIT 10000000;
```

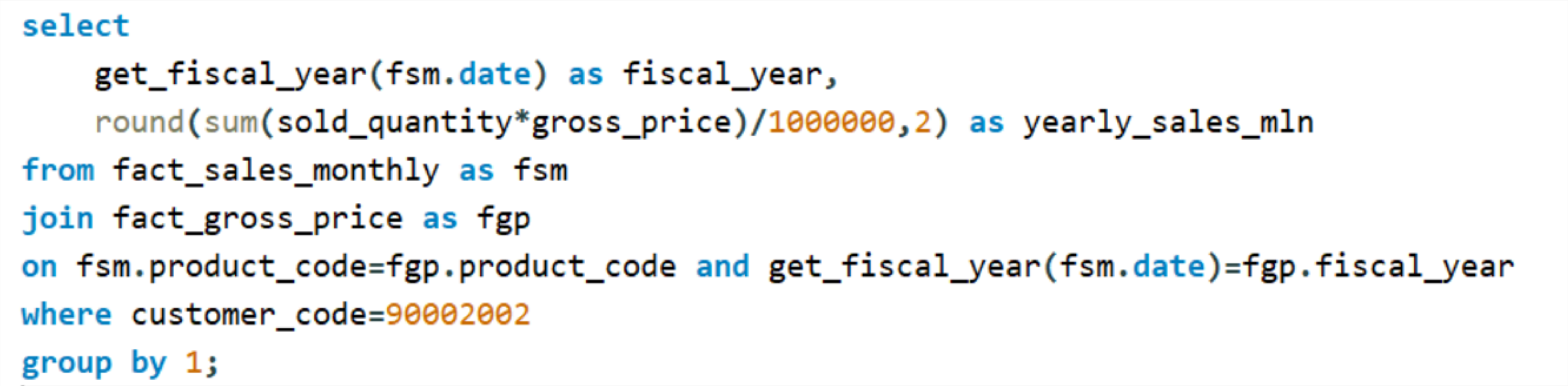
Result



month	fiscal_year	product_code	product	variant	gross_price	sold_quantity	gross_price_total
1	2021	A0118150101	AQ Dracula HDD – 3.5 Inc...	Standard	19.0573	182	3468.43
1	2021	A0118150102	AQ Dracula HDD – 3.5 Inc...	Plus	21.4565	121	2596.24
1	2021	A0118150103	AQ Dracula HDD – 3.5 Inc...	Premium	21.7795	142	3092.69
1	2021	A0118150104	AQ Dracula HDD – 3.5 Inc...	Premium Plus	22.9729	35	804.05
1	2021	A0219150201	AQ WereWolf NAS Interna...	Standard	23.6987	161	3815.49
1	2021	A0219150202	AQ WereWolf NAS Interna...	Plus	24.7312	241	5960.22
1	2021	A0220150203	AQ WereWolf NAS Interna...	Premium	23.6154	41	968.23
1	2021	A0320150301	AQ Zion Saga	Standard	23.7223	121	2870.40
1	2021	A0321150302	AQ Zion Saga	Plus	27.1027	164	4444.84
1	2021	A0321150303	AQ Zion Saga	Premium	28.0059	172	4817.01
1	2021	A0418150103	AQ Mforce Gen X	Standard 3	19.5235	133	2596.63
1	2021	A0418150104	AQ Mforce Gen X	Plus 1	19.9239	155	3088.20
1	2021	A0418150105	AQ Mforce Gen X	Plus 2	20.0766	176	3533.48
1	2021	A0418150106	AQ Mforce Gen X	Plus 3	19.9365	121	2412.32
1	2021	A0519150201	AQ Mforce Gen Y	Standard 1	22.3984	81	1814.27

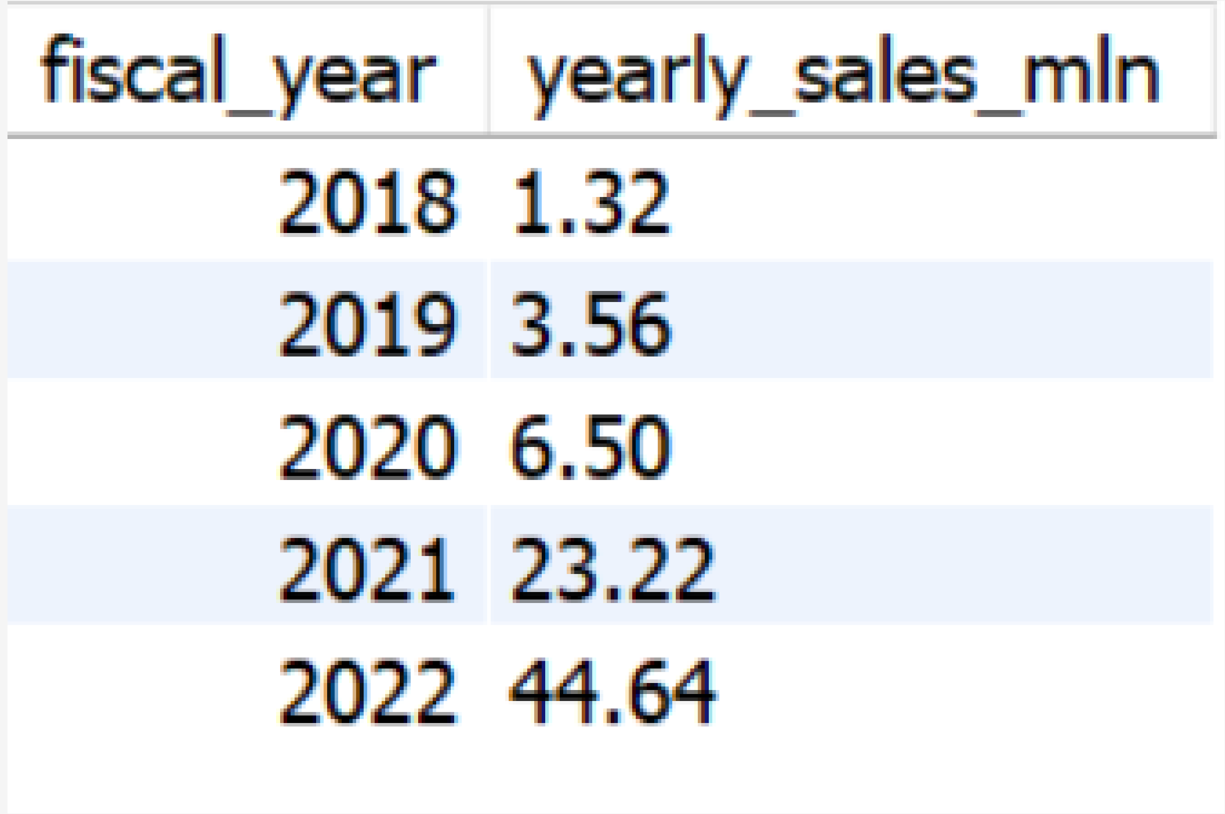
Generate Croma India Sales report on yearly basis

SQL Query



```
select
    get_fiscal_year(fsm.date) as fiscal_year,
    round(sum(sold_quantity*gross_price)/1000000,2) as yearly_sales_mln
from fact_sales_monthly as fsm
join fact_gross_price as fgp
on fsm.product_code=fgp.product_code and get_fiscal_year(fsm.date)=fgp.fiscal_year
where customer_code=90002002
group by 1;
```

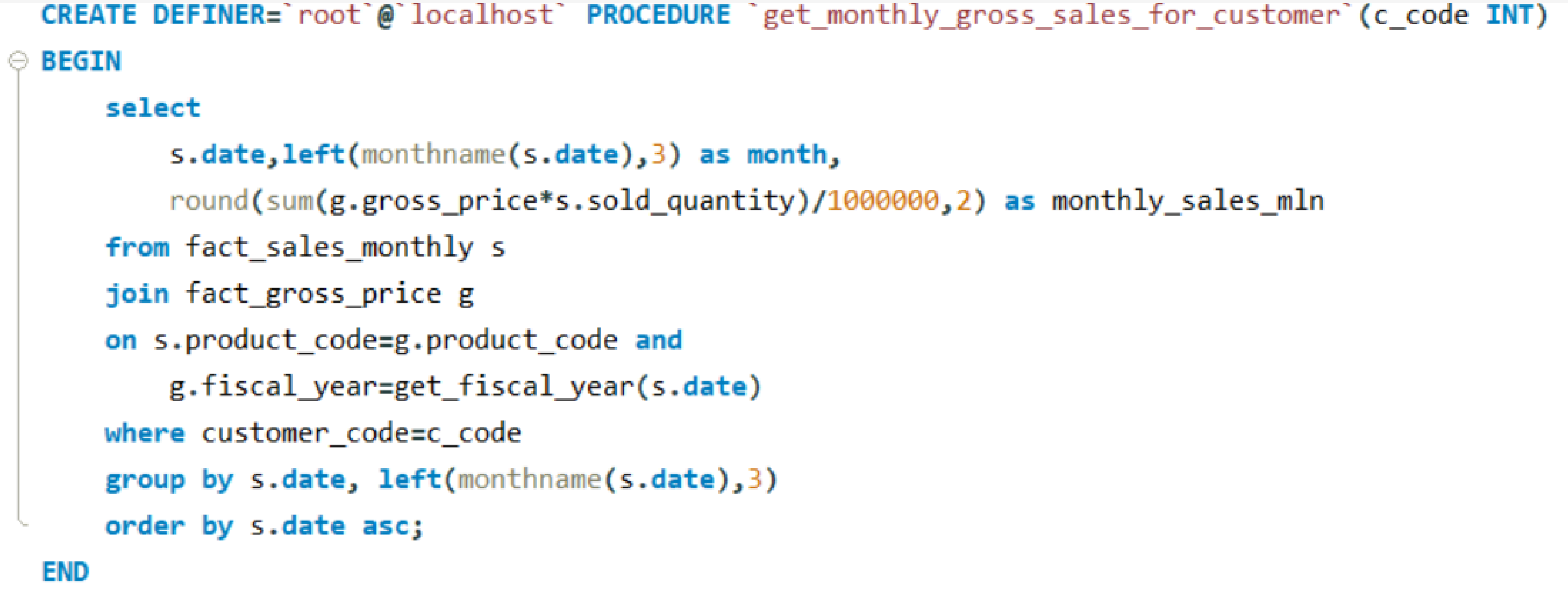
Result



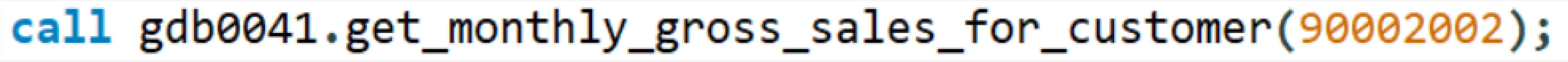
fiscal_year	yearly_sales_mln
2018	1.32
2019	3.56
2020	6.50
2021	23.22
2022	44.64

Create Stored Procedure to get gross monthly total sales report by Customer

SQL Query



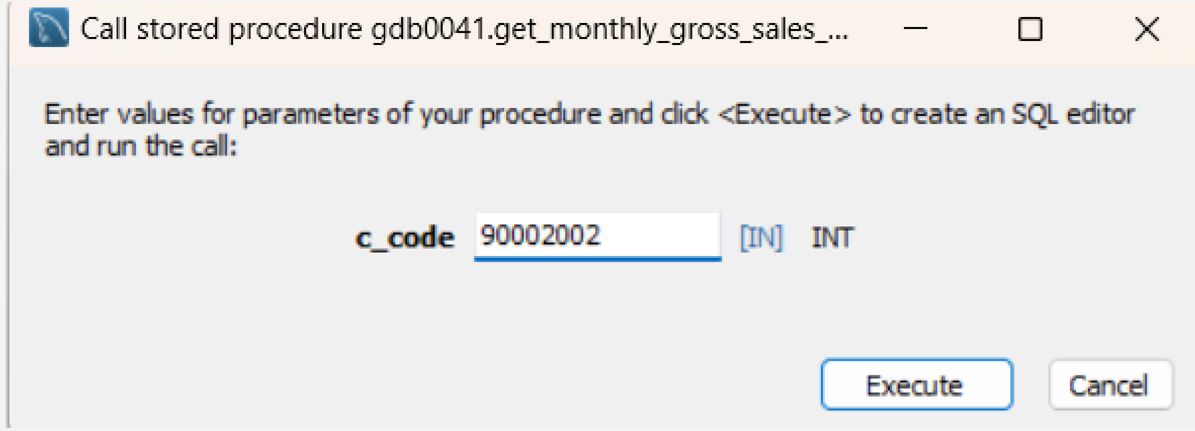
```
CREATE DEFINER=`root`@`localhost` PROCEDURE `get_monthly_gross_sales_for_customer`(c_code INT)
BEGIN
    select
        s.date,left(monthname(s.date),3) as month,
        round(sum(g.gross_price*s.sold_quantity)/1000000,2) as monthly_sales_mln
    from fact_sales_monthly s
    join fact_gross_price g
    on s.product_code=g.product_code and
        g.fiscal_year=get_fiscal_year(s.date)
    where customer_code=c_code
    group by s.date, left(monthname(s.date),3)
    order by s.date asc;
END
```



```
call gdb0041.get_monthly_gross_sales_for_customer(90002002);
```

OR

Calling



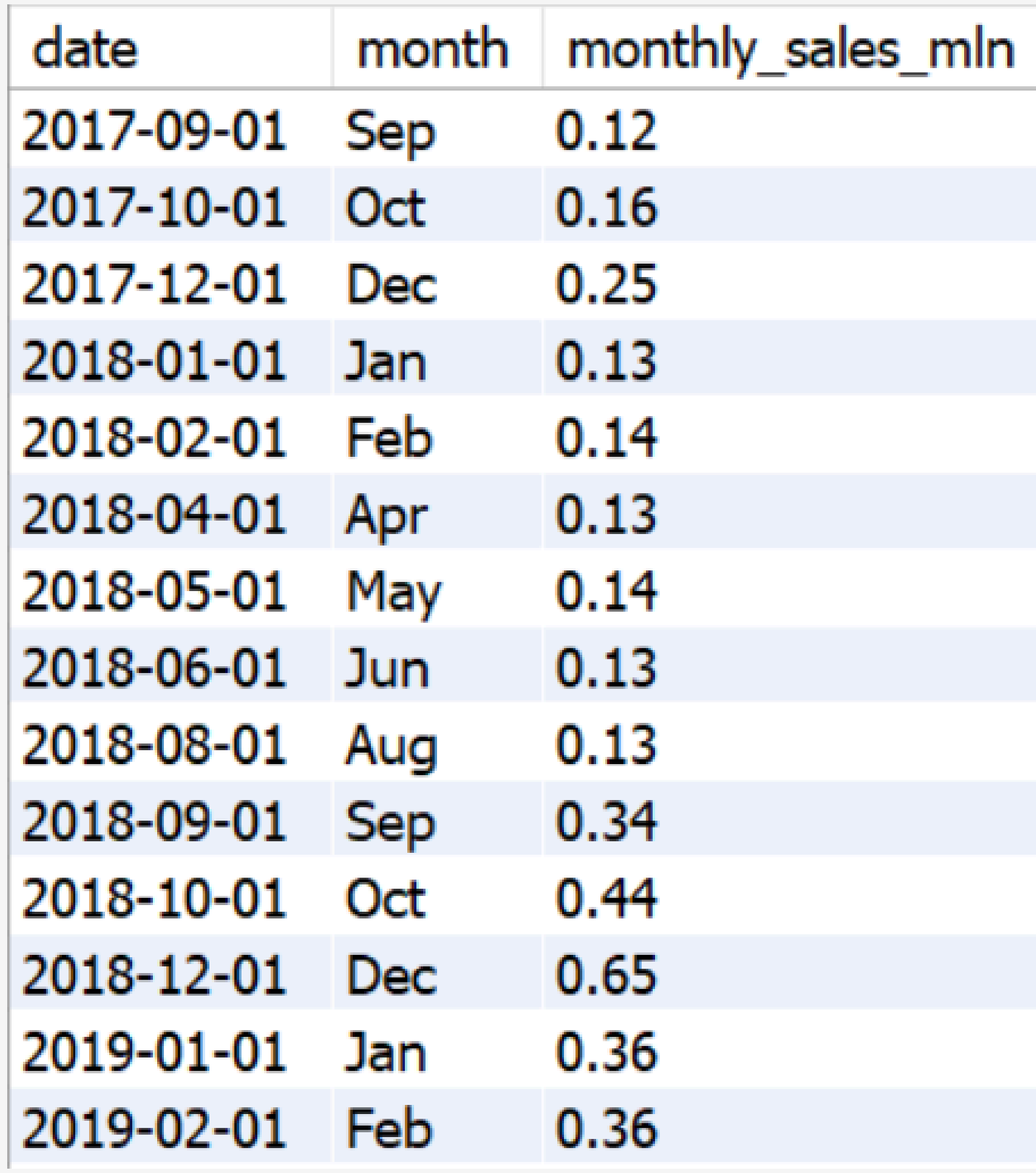
Call stored procedure gdb0041.get_monthly_gross_sales_...

Enter values for parameters of your procedure and click <Execute> to create an SQL editor and run the call:

c_code 90002002 [IN] INT

For Croma

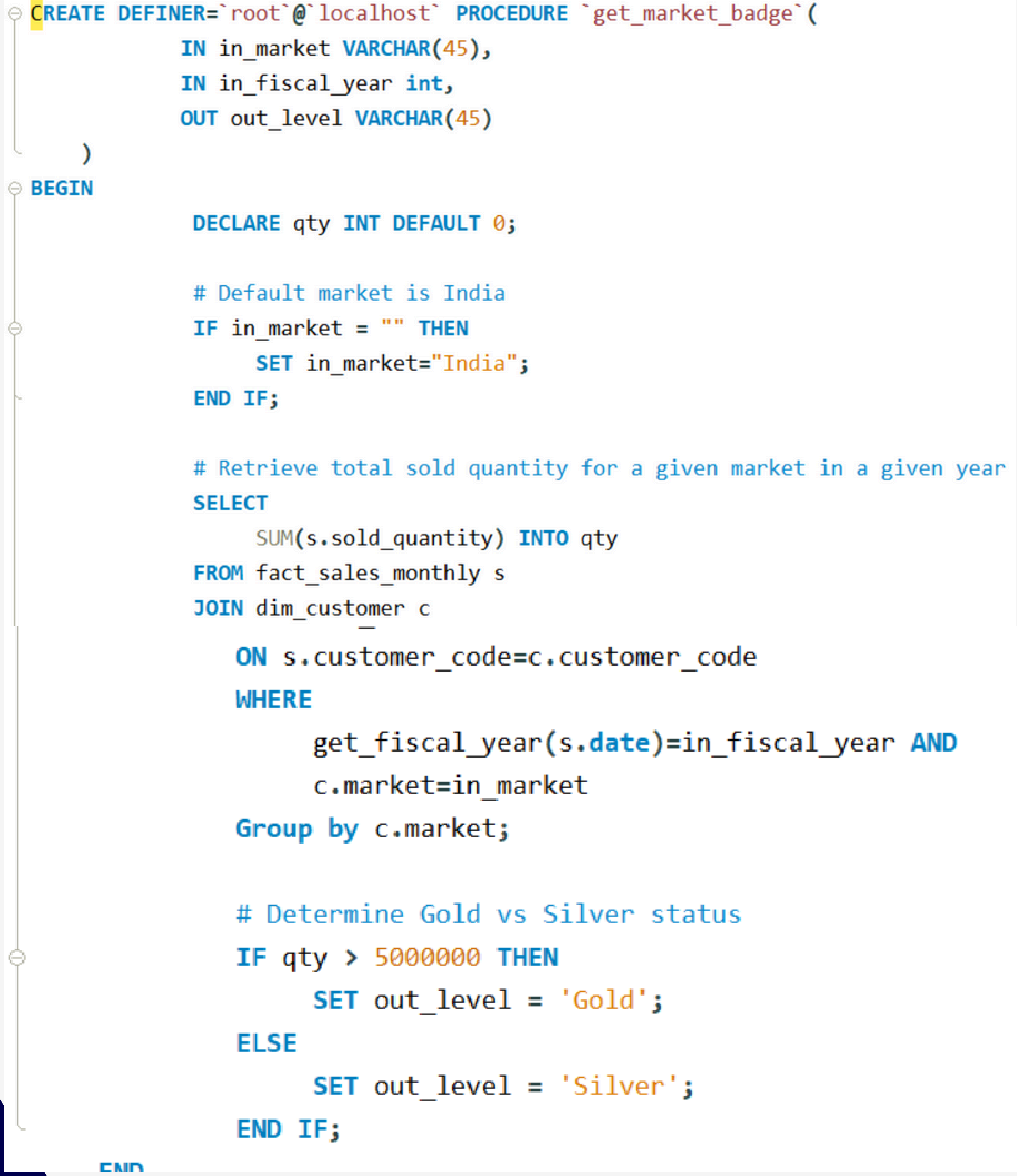
Result



date	month	monthly_sales_mln
2017-09-01	Sep	0.12
2017-10-01	Oct	0.16
2017-12-01	Dec	0.25
2018-01-01	Jan	0.13
2018-02-01	Feb	0.14
2018-04-01	Apr	0.13
2018-05-01	May	0.14
2018-06-01	Jun	0.13
2018-08-01	Aug	0.13
2018-09-01	Sep	0.34
2018-10-01	Oct	0.44
2018-12-01	Dec	0.65
2019-01-01	Jan	0.36
2019-02-01	Feb	0.36

Create stored procedure that can determine the market badge based on the following logic {If total sold quantity > 5 million that market is considered Gold else it is Silver}

SQL Query



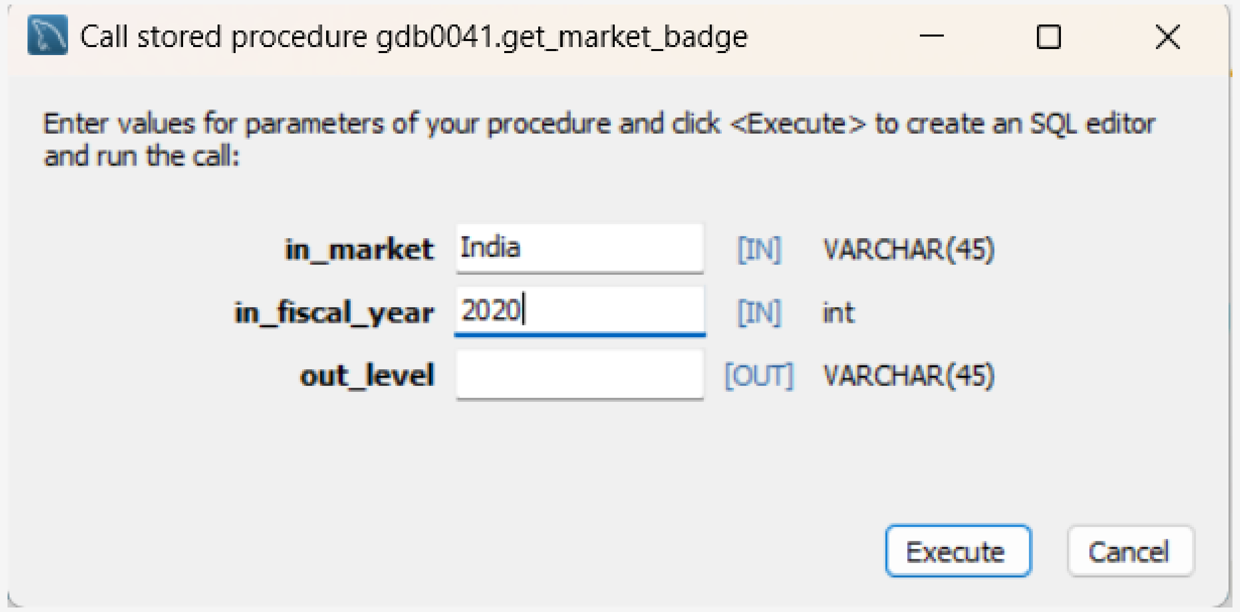
```
CREATE DEFINER=`root`@`localhost` PROCEDURE `get_market_badge`(
            IN in_market VARCHAR(45),
            IN in_fiscal_year int,
            OUT out_level VARCHAR(45)
    )
BEGIN
             DECLARE qty INT DEFAULT 0;

             # Default market is India
             IF in_market = "" THEN
                  SET in_market="India";
             END IF;

             # Retrieve total sold quantity for a given market in a given year
             SELECT
                  SUM(s.sold_quantity) INTO qty
             FROM fact_sales_monthly s
             JOIN dim_customer c
             ON s.customer_code=c.customer_code
             WHERE
                  get_fiscal_year(s.date)=in_fiscal_year AND
                  c.market=in_market
             Group by c.market;

             # Determine Gold vs Silver status
             IF qty > 5000000 THEN
                  SET out_level = 'Gold';
             ELSE
                  SET out_level = 'Silver';
             END IF;
     END
```

Calling

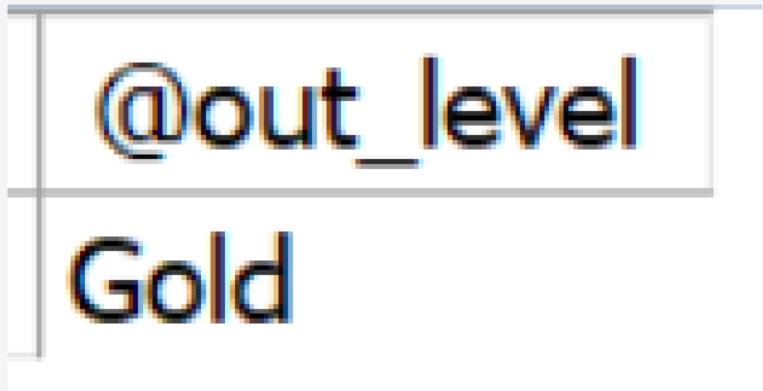


Call stored procedure gdb0041.get_market_badge

Enter values for parameters of your procedure and click <Execute> to create an SQL editor and run the call:

in_market	India	[IN]	VARCHAR(45)
in_fiscal_year	2020	[IN]	int
out_level		[OUT]	VARCHAR(45)

Result



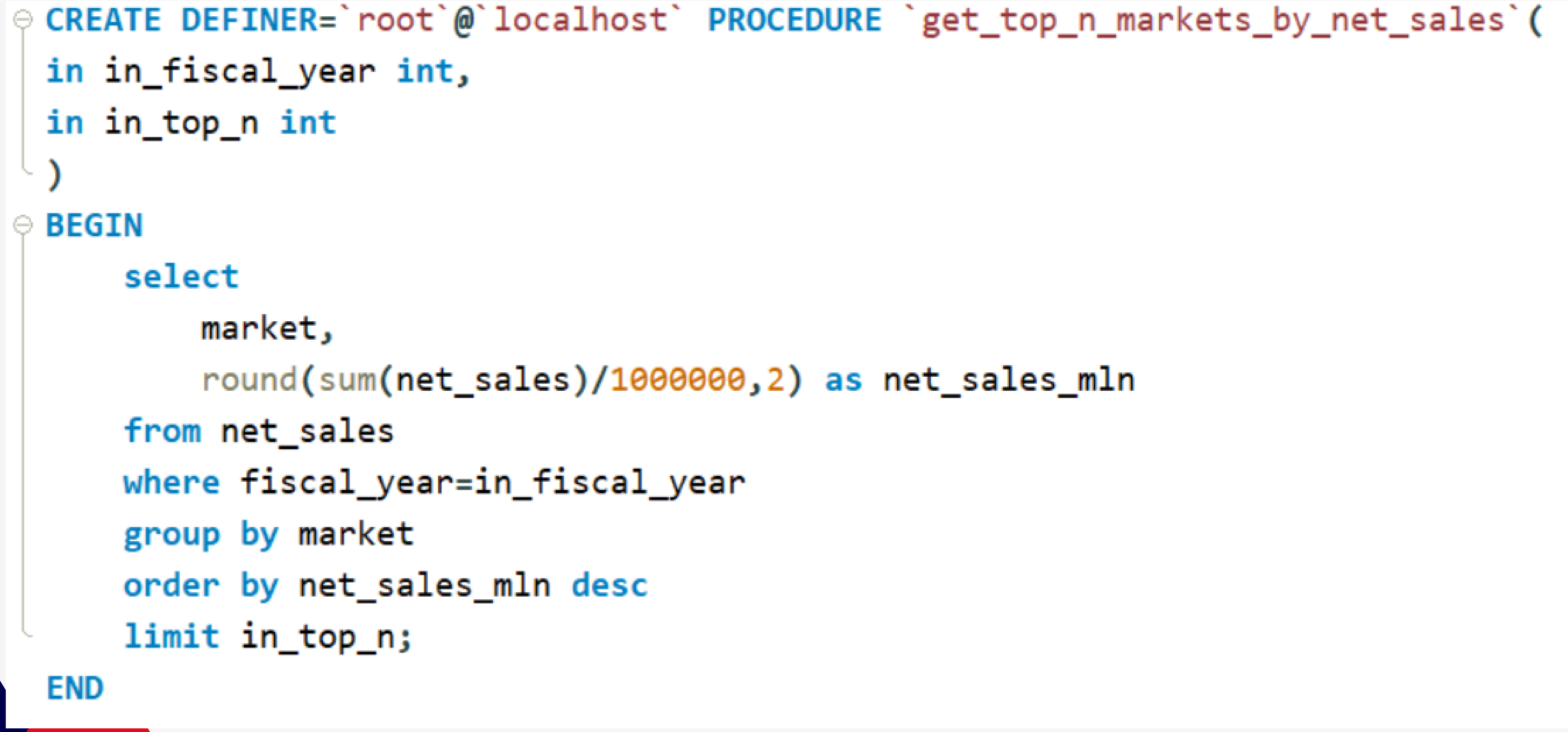
@out_level
Gold

Write Stored Procedure for generating report for top markets, products, customers by net sales for a given financial year to get an holistic view of financial performance and to take appropriate actions to address any potential issues.

Used views for gross_sales, sales_preinv_discount, sales_postinv_discount and net_sales for query

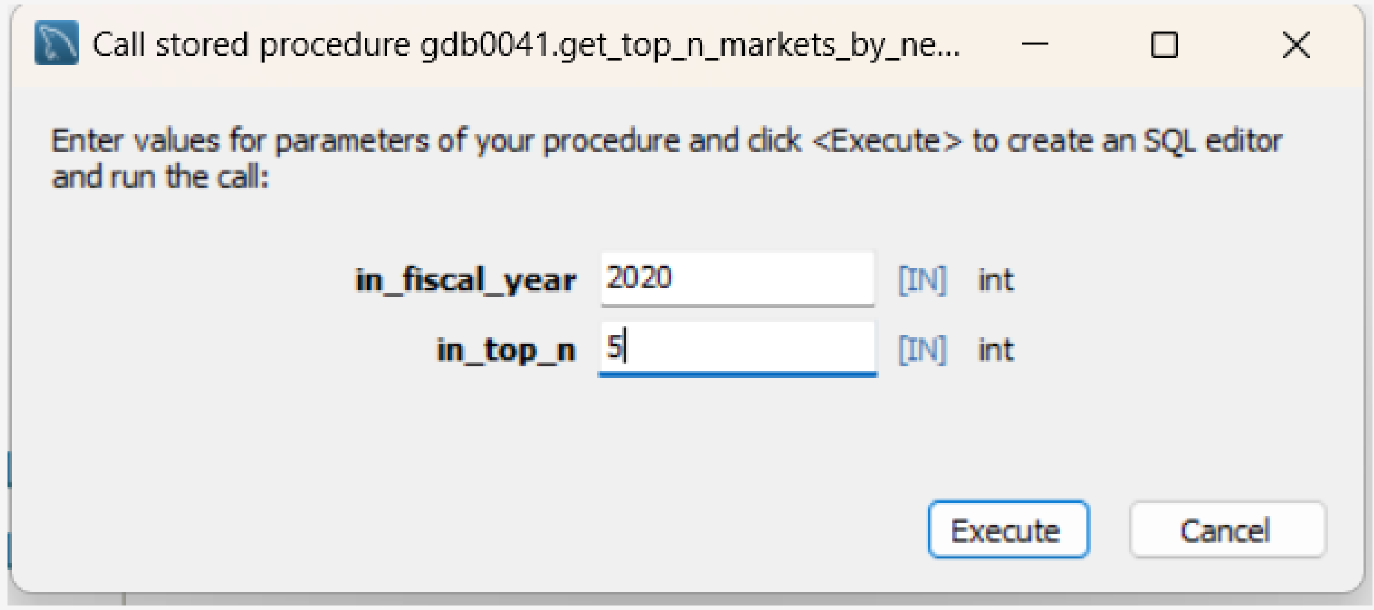
Top 5 Markets by Net Sales for FY20

SQL Query



```
CREATE DEFINER=`root`@`localhost` PROCEDURE `get_top_n_markets_by_net_sales`(
in in_fiscal_year int,
in 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
```

Calling

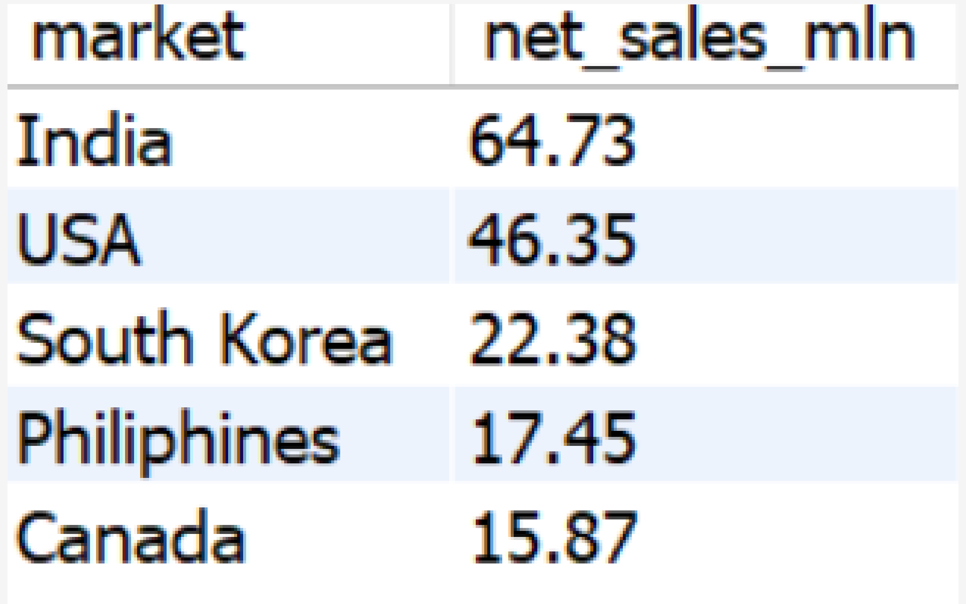


Enter values for parameters of your procedure and click <Execute> to create an SQL editor and run the call:

in_fiscal_year 2020 [IN] int

in_top_n 5 [IN] int

Result

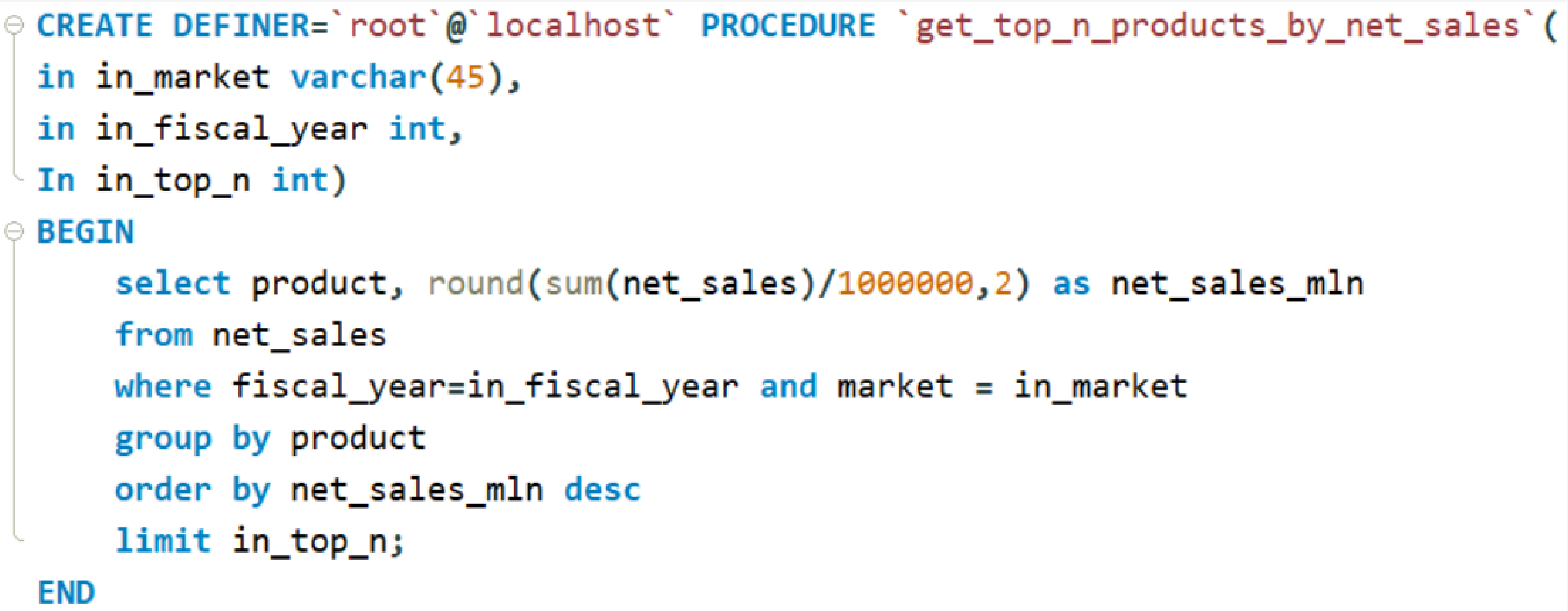


market	net_sales_mln
India	64.73
USA	46.35
South Korea	22.38
Philiphines	17.45
Canada	15.87

Write Stored Procedure for generating report for top markets, products, customers by net sales for a given financial year to get an holistic view of financial performance and to take appropriate actions to address any potential issues.

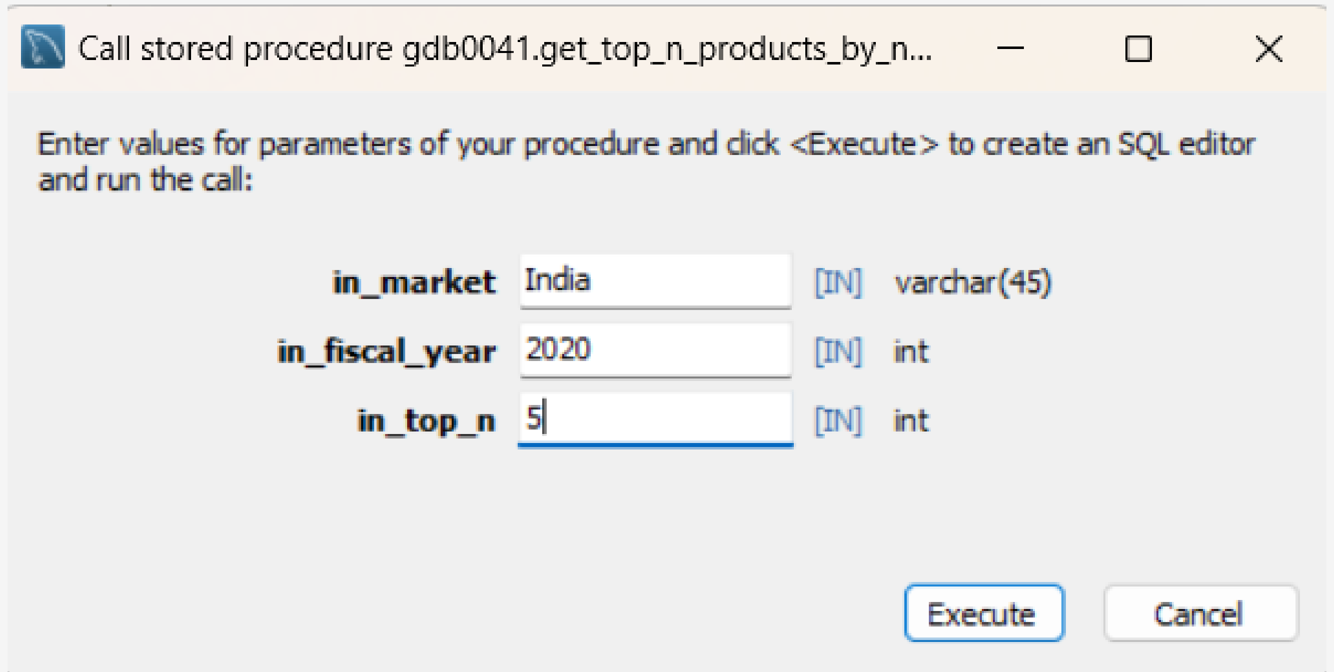
Top 5 Products by Net Sales for FY20 in India

SQL Query



```
CREATE DEFINER=`root`@`localhost` PROCEDURE `get_top_n_products_by_net_sales`(
in in_market varchar(45),
in in_fiscal_year int,
In in_top_n int)
BEGIN
    select product, round(sum(net_sales)/1000000,2) as net_sales_mln
    from net_sales
    where fiscal_year=in_fiscal_year and market = in_market
    group by product
    order by net_sales_mln desc
    limit in_top_n;
END
```

Calling



Call stored procedure gdb0041.get_top_n_products_by_n...

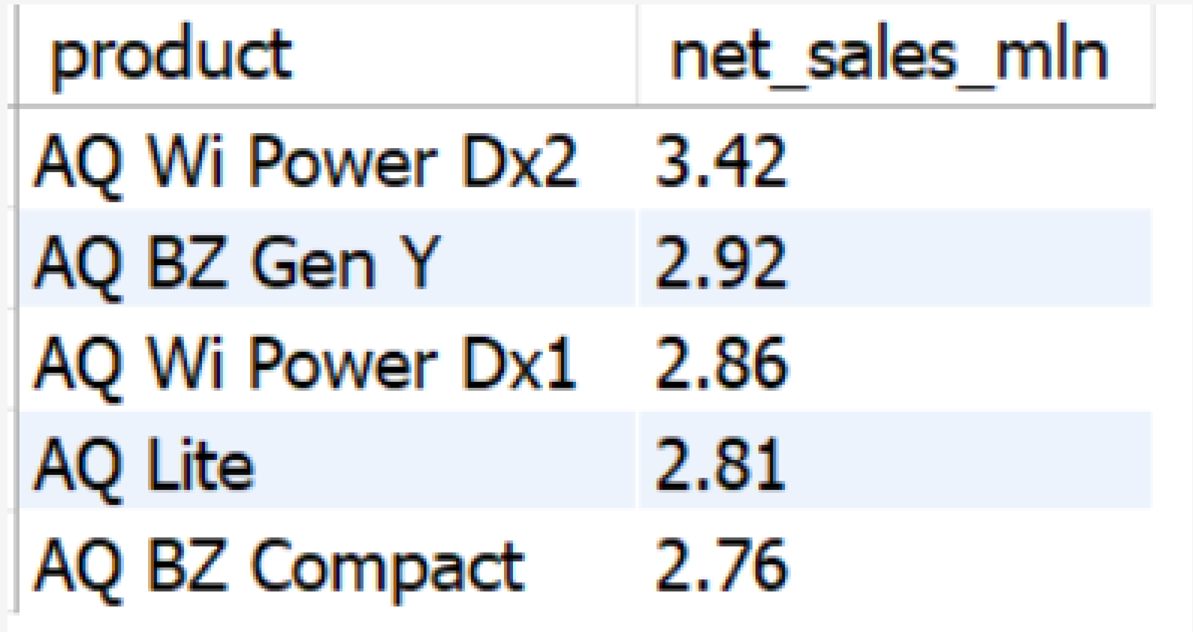
Enter values for parameters of your procedure and click <Execute> to create an SQL editor and run the call:

in_market: India [IN] varchar(45)

in_fiscal_year: 2020 [IN] int

in_top_n: 5 [IN] int

Result

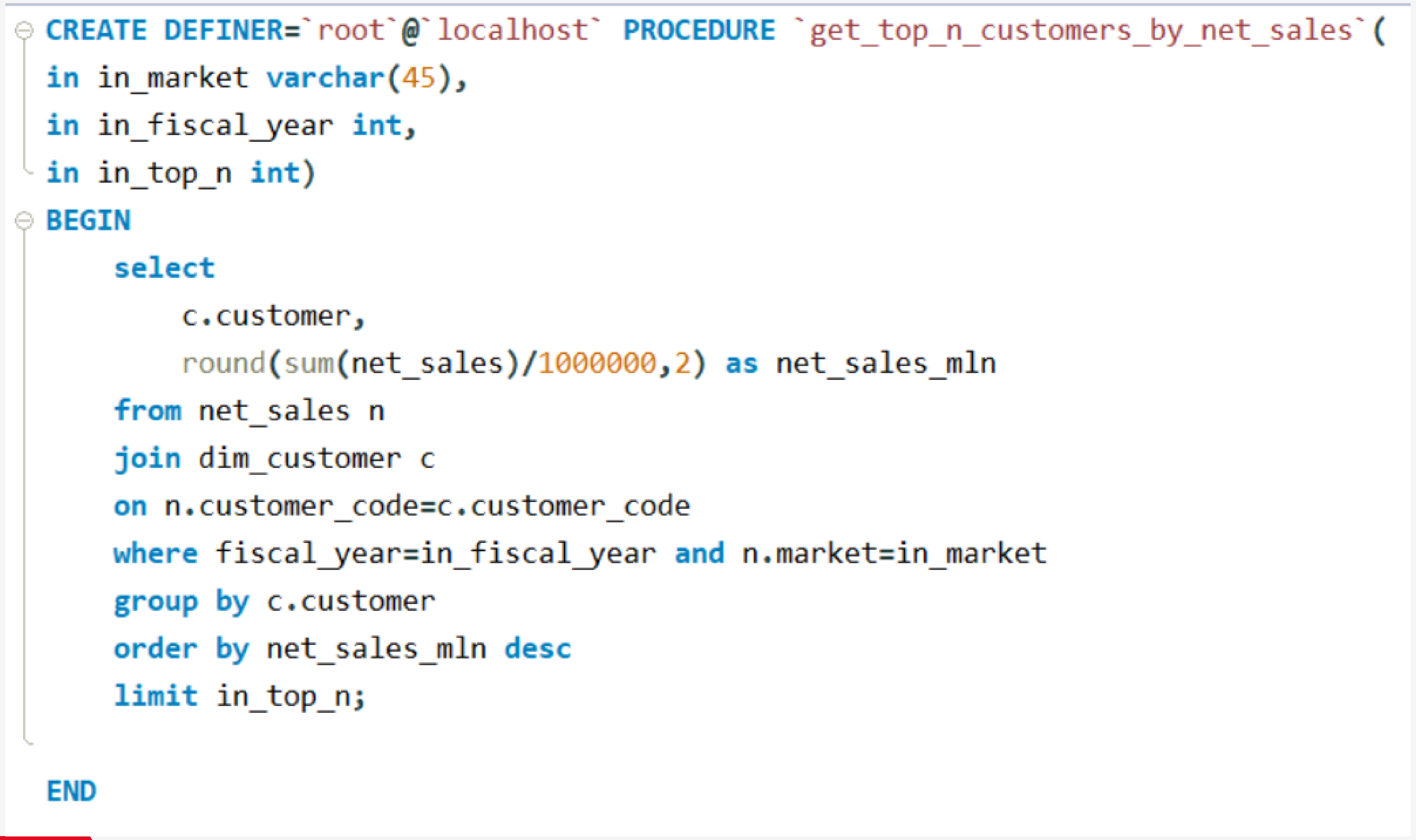


product	net_sales_mln
AQ Wi Power Dx2	3.42
AQ BZ Gen Y	2.92
AQ Wi Power Dx1	2.86
AQ Lite	2.81
AQ BZ Compact	2.76

Write Stored Procedure for generating report for top markets, products, customers by net sales for a given financial year to get an holistic view of financial performance and to take appropriate actions to address any potential issues.

Top 5 Customers by Net Sales for FY20 in India

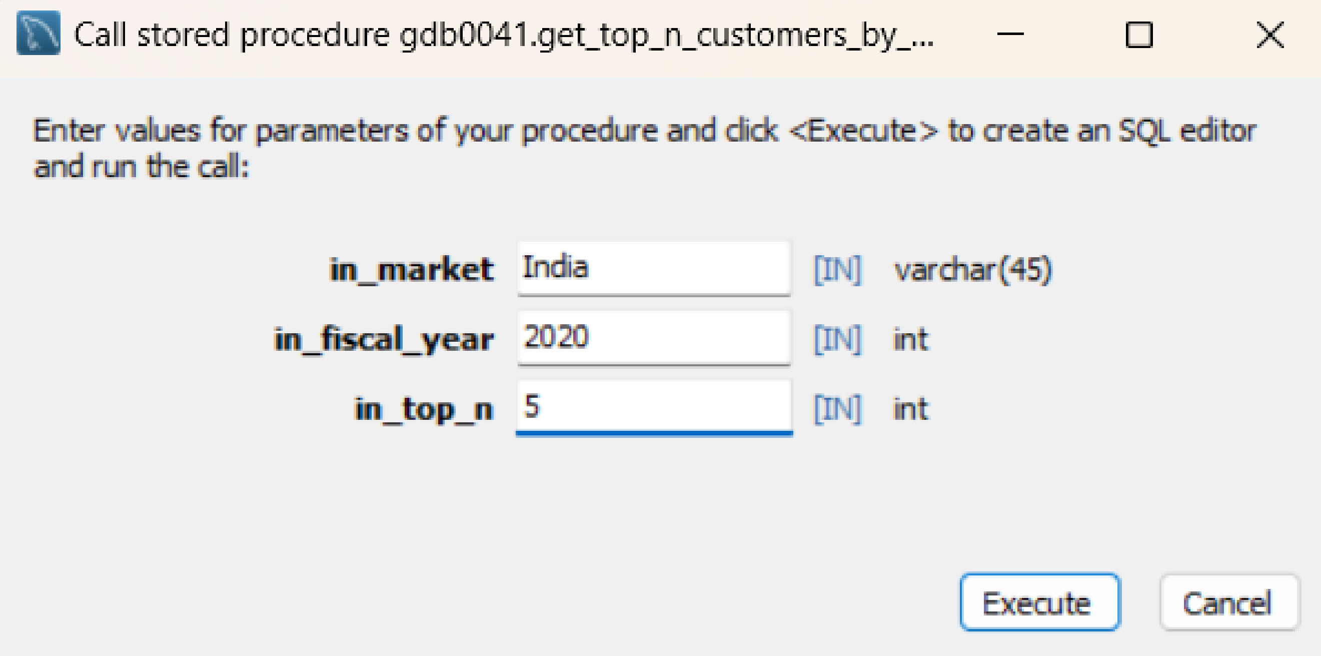
SQL Query



```
CREATE DEFINER=`root`@`localhost` PROCEDURE `get_top_n_customers_by_net_sales`(
in in_market varchar(45),
in in_fiscal_year int,
in in_top_n int)
BEGIN
    select
        c.customer,
        round(sum(net_sales)/1000000,2) as net_sales_mln
    from net_sales n
    join dim_customer c
    on n.customer_code=c.customer_code
    where fiscal_year=in_fiscal_year and n.market=in_market
    group by c.customer
    order by net_sales_mln desc
    limit in_top_n;

END
```

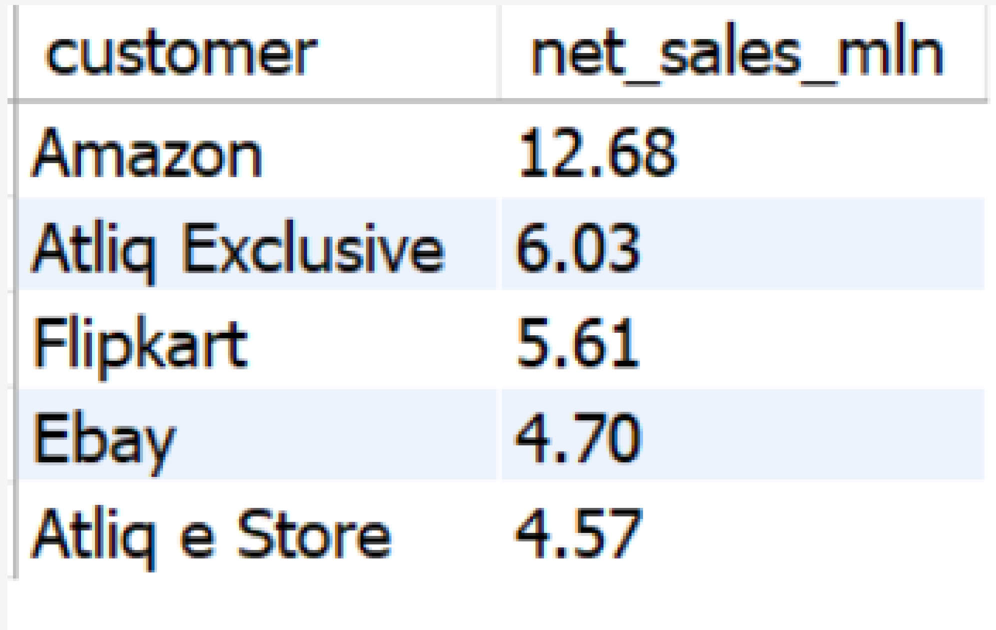
Calling



Enter values for parameters of your procedure and click <Execute> to create an SQL editor and run the call:

in_market	India	[IN]	varchar(45)
in_fiscal_year	2020	[IN]	int
in_top_n	5	[IN]	int

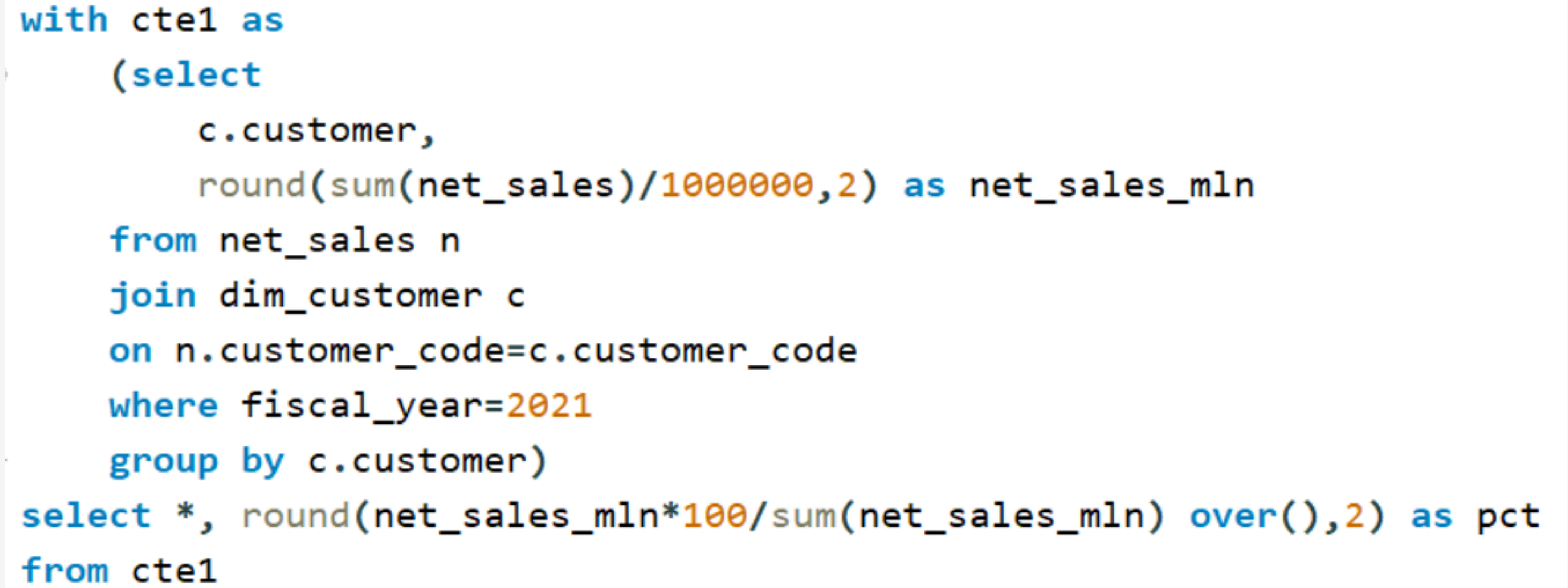
Result



customer	net_sales_mln
Amazon	12.68
Atliq Exclusive	6.03
Flipkart	5.61
Ebay	4.70
Atliq e Store	4.57

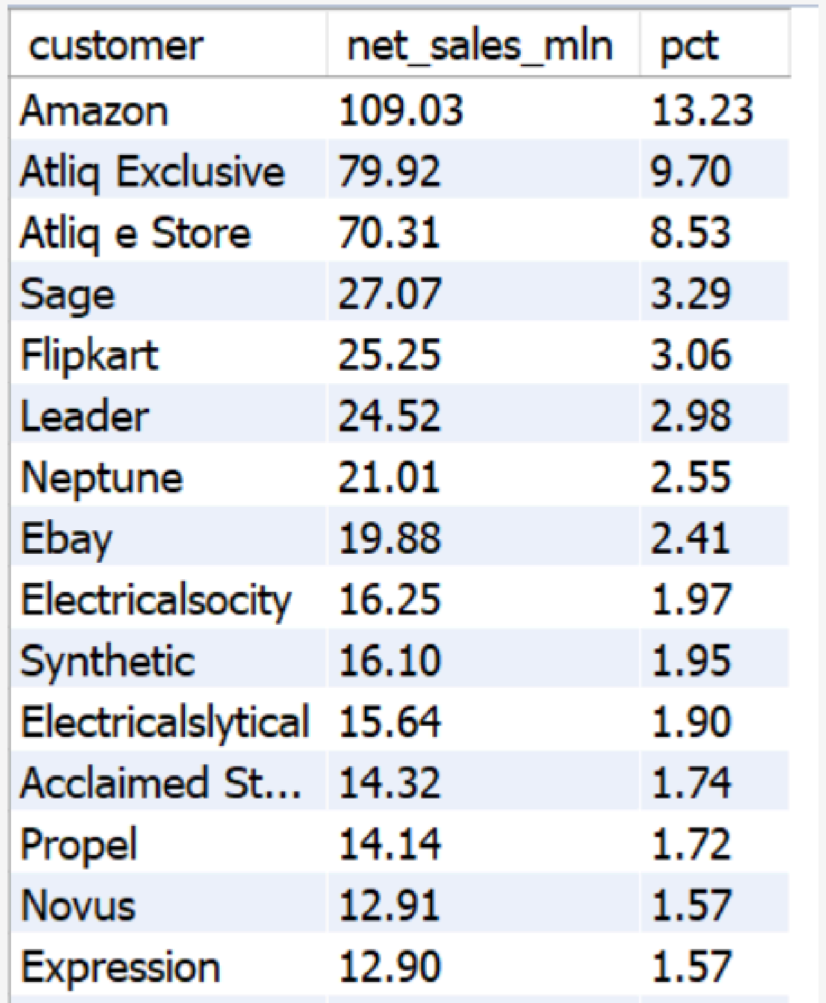
Generate Customers net sales % share report for FY 2021

SQL Query

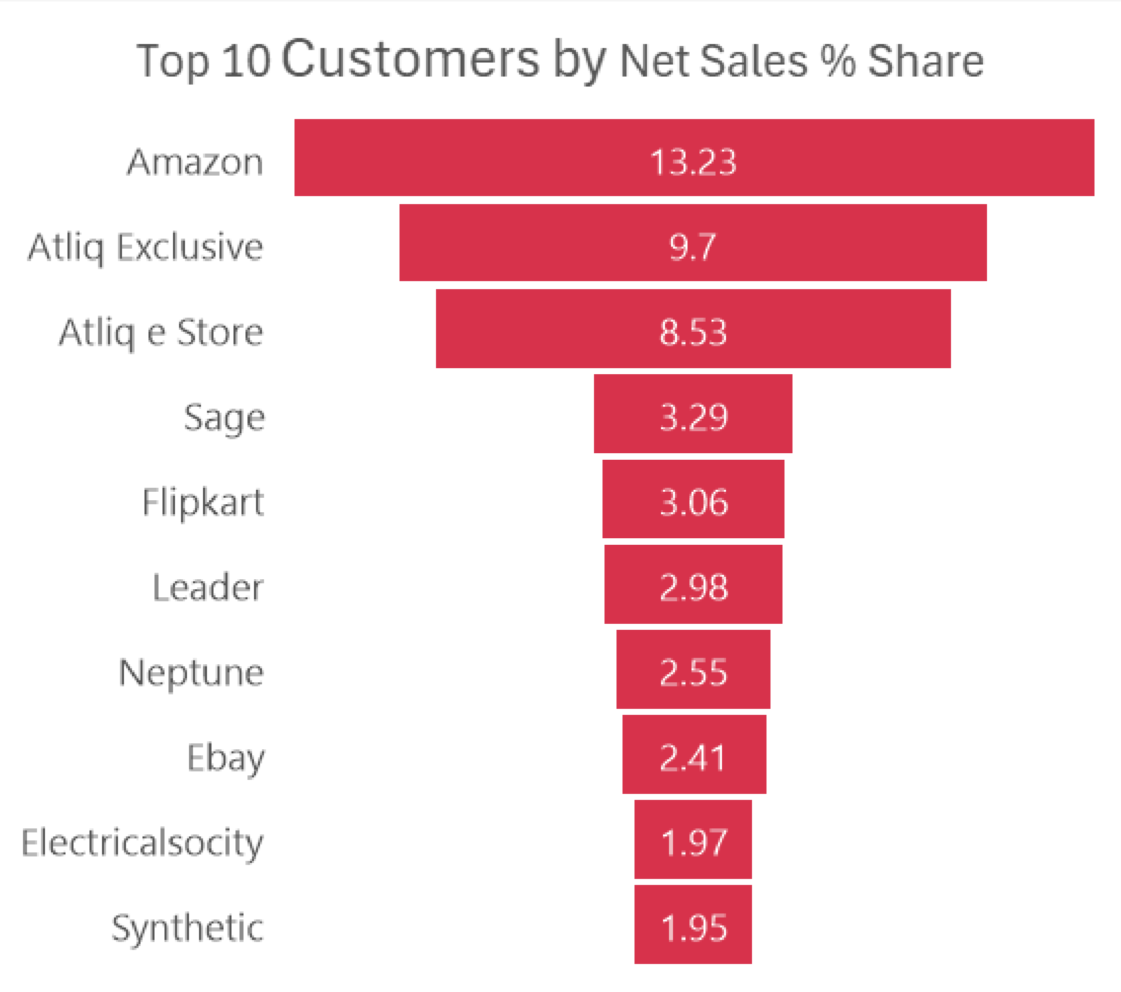


```
with cte1 as
    (select
        c.customer,
        round(sum(net_sales)/1000000,2) as net_sales_mln
    from net_sales n
    join dim_customer c
    on n.customer_code=c.customer_code
    where fiscal_year=2021
    group by c.customer)
select *, round(net_sales_mln*100/sum(net_sales_mln) over(),2) as pct
from cte1
```

Result



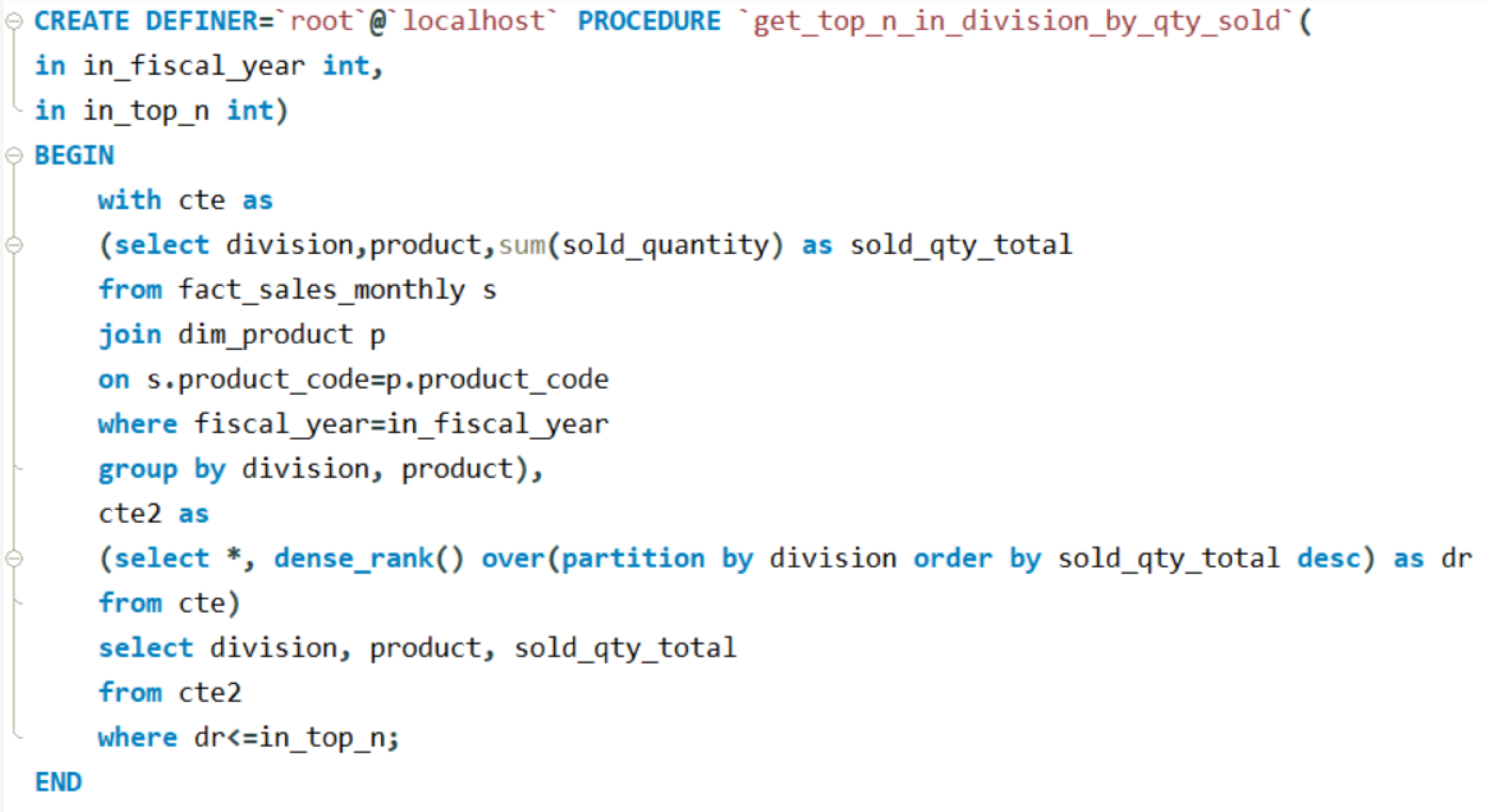
customer	net_sales_mln	pct
Amazon	109.03	13.23
Atliq Exclusive	79.92	9.70
Atliq e Store	70.31	8.53
Sage	27.07	3.29
Flipkart	25.25	3.06
Leader	24.52	2.98
Neptune	21.01	2.55
Ebay	19.88	2.41
Electricalsocity	16.25	1.97
Synthetic	16.10	1.95
Electricalslytical	15.64	1.90
Acclaimed St...	14.32	1.74
Propel	14.14	1.72
Novus	12.91	1.57
Expression	12.90	1.57



Top 10 Customers by Net Sales % Share

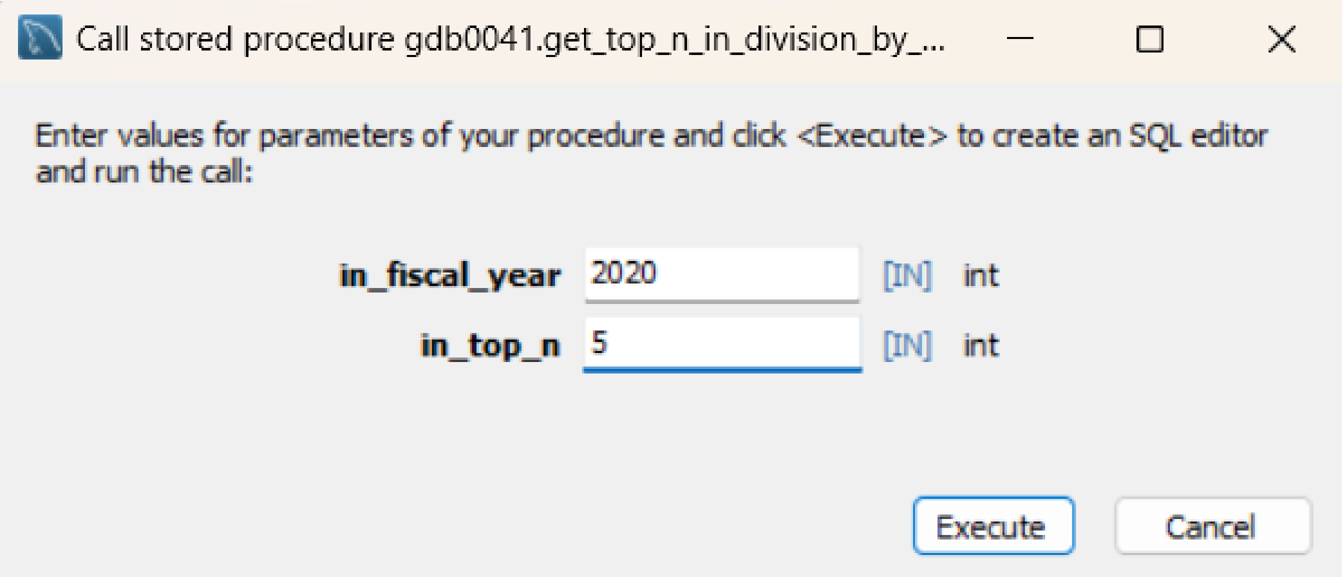
Create Stored Procedure to get top n products in each division by their quantity sold as per Fiscal Year

SQL Query



```
CREATE DEFINER=`root`@`localhost` PROCEDURE `get_top_n_in_division_by_qty_sold`(
in in_fiscal_year int,
in in_top_n int)
BEGIN
    with cte as
    (select division,product,sum(sold_quantity) as sold_qty_total
    from fact_sales_monthly s
    join dim_product p
    on s.product_code=p.product_code
    where fiscal_year=in_fiscal_year
    group by division, product),
    cte2 as
    (select *, dense_rank() over(partition by division order by sold_qty_total desc) as dr
    from cte)
    select division, product, sold_qty_total
    from cte2
    where dr<=in_top_n;
END
```

Calling



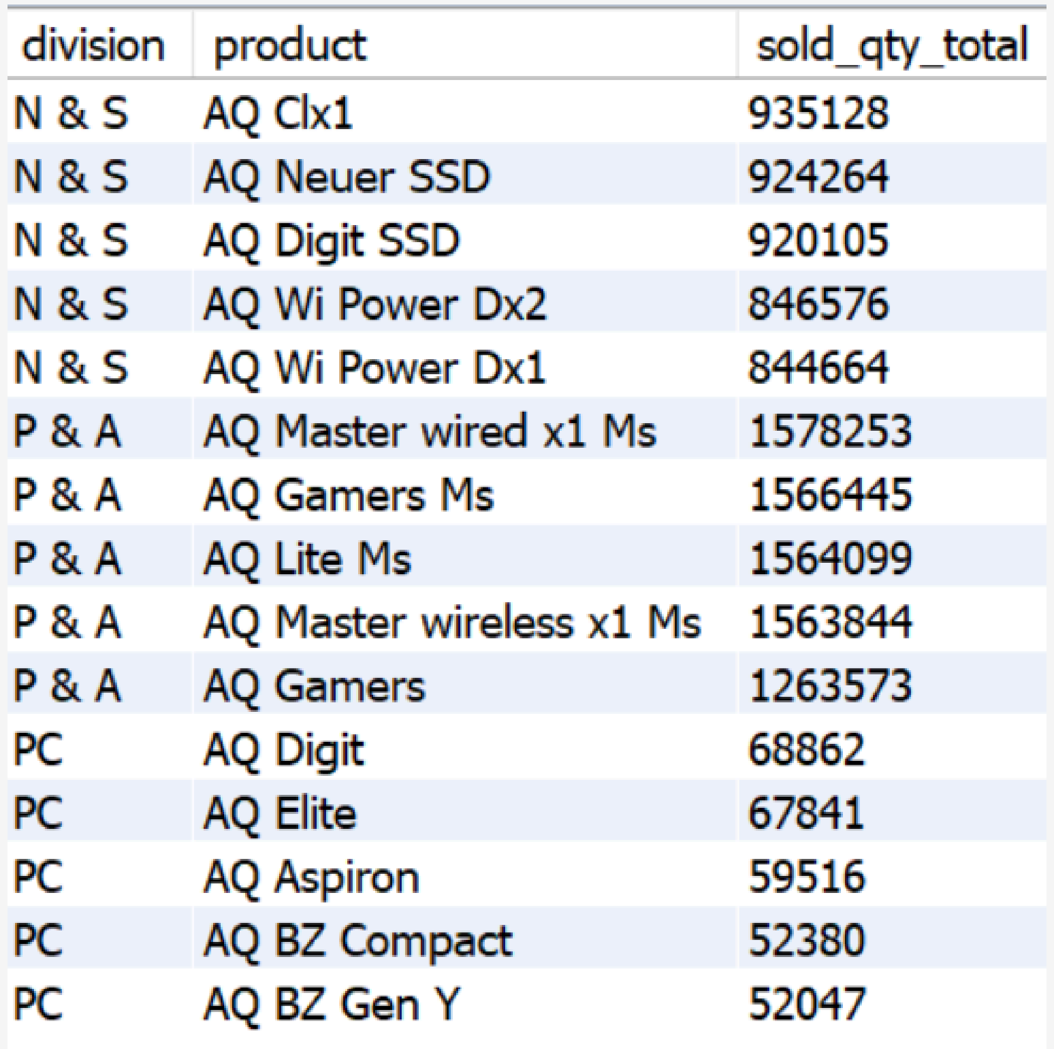
Call stored procedure gdb0041.get_top_n_in_division_by_...

Enter values for parameters of your procedure and click <Execute> to create an SQL editor and run the call:

in_fiscal_year 2020 [IN] int

in_top_n 5 [IN] int

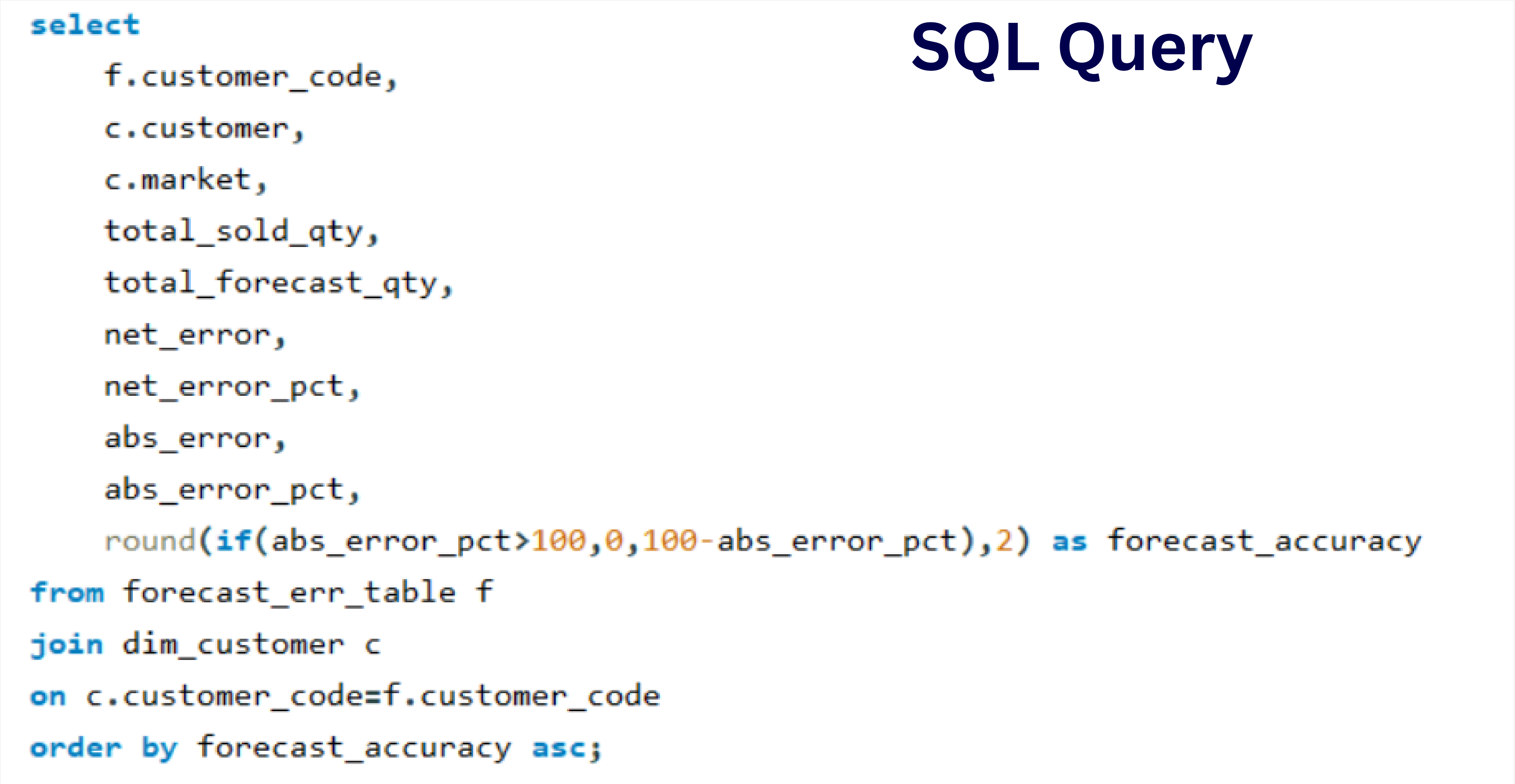
Result



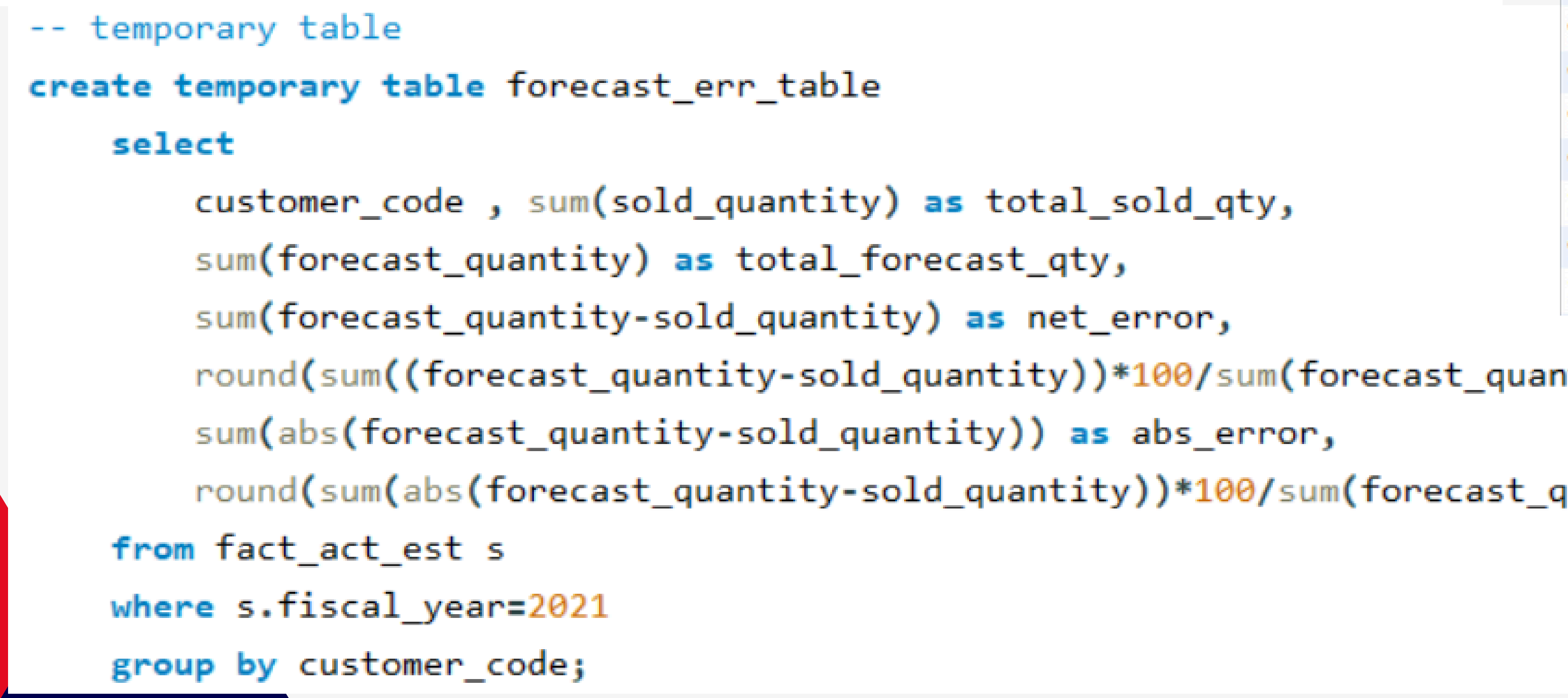
division	product	sold_qty_total
N & S	AQ Clx1	935128
N & S	AQ Neuer SSD	924264
N & S	AQ Digit SSD	920105
N & S	AQ Wi Power Dx2	846576
N & S	AQ Wi Power Dx1	844664
P & A	AQ Master wired x1 Ms	1578253
P & A	AQ Gamers Ms	1566445
P & A	AQ Lite Ms	1564099
P & A	AQ Master wireless x1 Ms	1563844
P & A	AQ Gamers	1263573
PC	AQ Digit	68862
PC	AQ Elite	67841
PC	AQ Aspiron	59516
PC	AQ BZ Compact	52380
PC	AQ BZ Gen Y	52047

Create an aggregate forecast accuracy report for all the customers for a given fiscal year to track forecast accuracy by customer

SQL Query

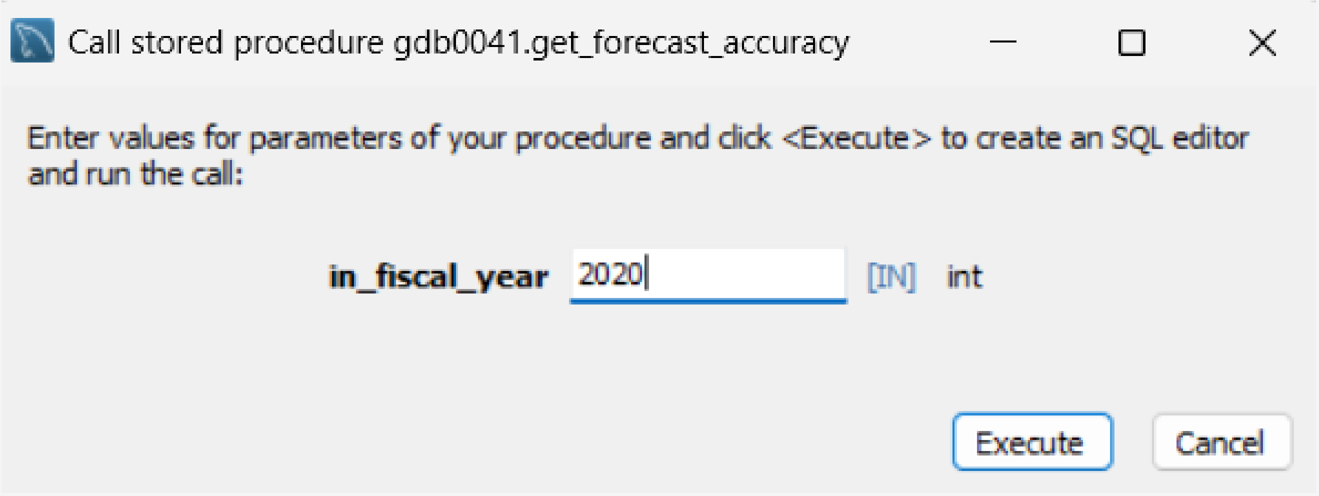


```
select
    f.customer_code,
    c.customer,
    c.market,
    total_sold_qty,
    total_forecast_qty,
    net_error,
    net_error_pct,
    abs_error,
    abs_error_pct,
    round(if(abs_error_pct>100,0,100-abs_error_pct),2) as forecast_accuracy
from forecast_err_table f
join dim_customer c
on c.customer_code=f.customer_code
order by forecast_accuracy asc;
```



```
-- temporary table
create temporary table forecast_err_table
    select
        customer_code , sum(sold_quantity) as total_sold_qty,
        sum(forecast_quantity) as total_forecast_qty,
        sum(forecast_quantity-sold_quantity) as net_error,
        round(sum((forecast_quantity-sold_quantity))*100/sum(forecast_quantity),2) as net_error_pct,
        sum(abs(forecast_quantity-sold_quantity)) as abs_error,
        round(sum(abs(forecast_quantity-sold_quantity))*100/sum(forecast_quantity),2) as abs_error_pct
    from fact_act_est s
    where s.fiscal_year=2021
    group by customer_code;
```

Using Temporary Table Method

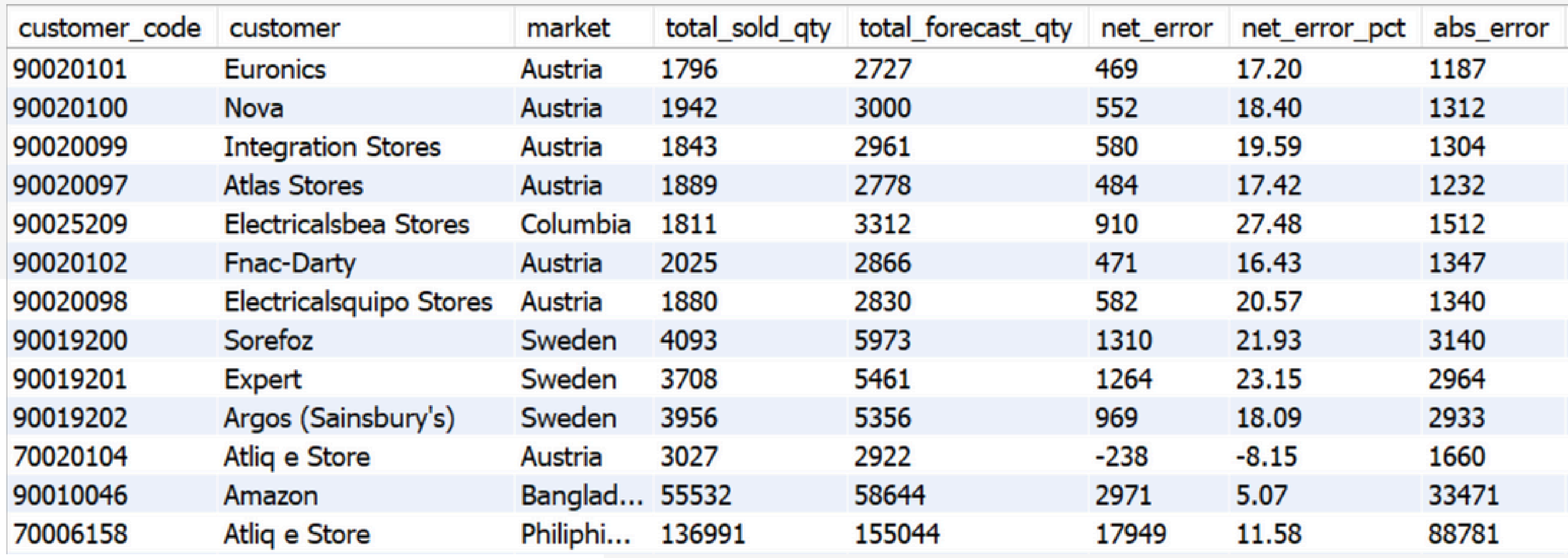


Call stored procedure gdb0041.get_forecast_accuracy

Enter values for parameters of your procedure and click <Execute> to create an SQL editor and run the call:

in_fiscal_year 2020 [IN] int

Result

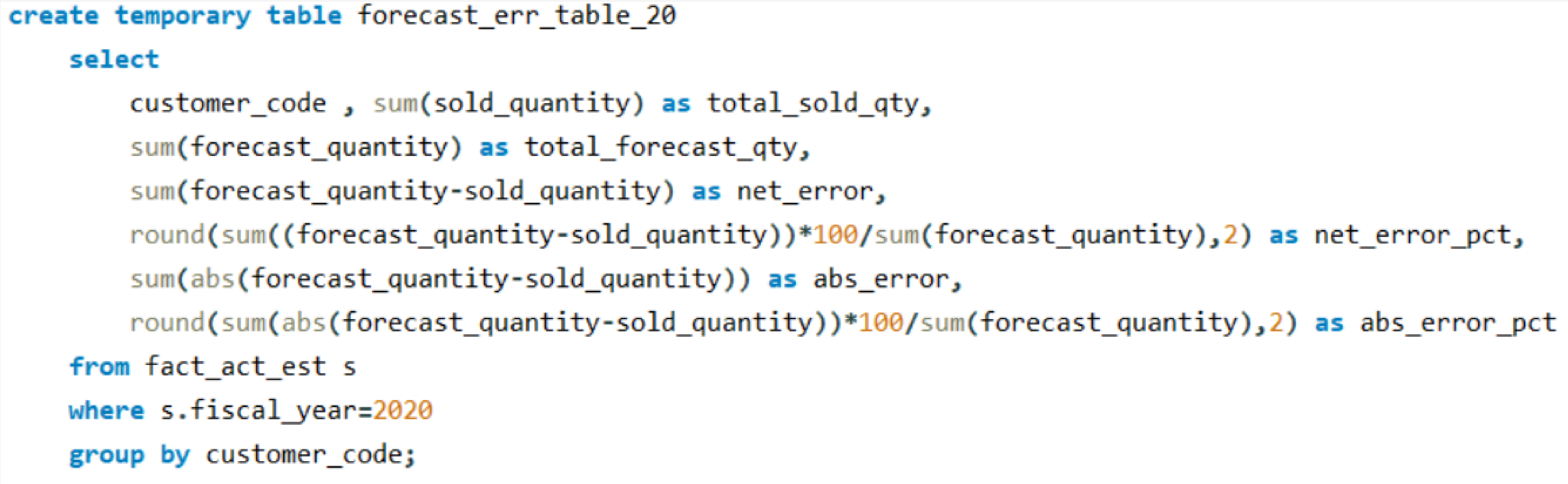


customer_code	customer	market	total_sold_qty	total_forecast_qty	net_error	net_error_pct	abs_error
90020101	Euronics	Austria	1796	2727	469	17.20	1187
90020100	Nova	Austria	1942	3000	552	18.40	1312
90020099	Integration Stores	Austria	1843	2961	580	19.59	1304
90020097	Atlas Stores	Austria	1889	2778	484	17.42	1232
90025209	Electricalsbea Stores	Columbia	1811	3312	910	27.48	1512
90020102	Fnac-Darty	Austria	2025	2866	471	16.43	1347
90020098	Electricalsquipo Stores	Austria	1880	2830	582	20.57	1340
90019200	Sorefoz	Sweden	4093	5973	1310	21.93	3140
90019201	Expert	Sweden	3708	5461	1264	23.15	2964
90019202	Argos (Sainsbury's)	Sweden	3956	5356	969	18.09	2933
70020104	Atliq e Store	Austria	3027	2922	-238	-8.15	1660
90010046	Amazon	Banglad...	55532	58644	2971	5.07	33471
70006158	Atliq e Store	Philiphi...	136991	155044	17949	11.58	88781

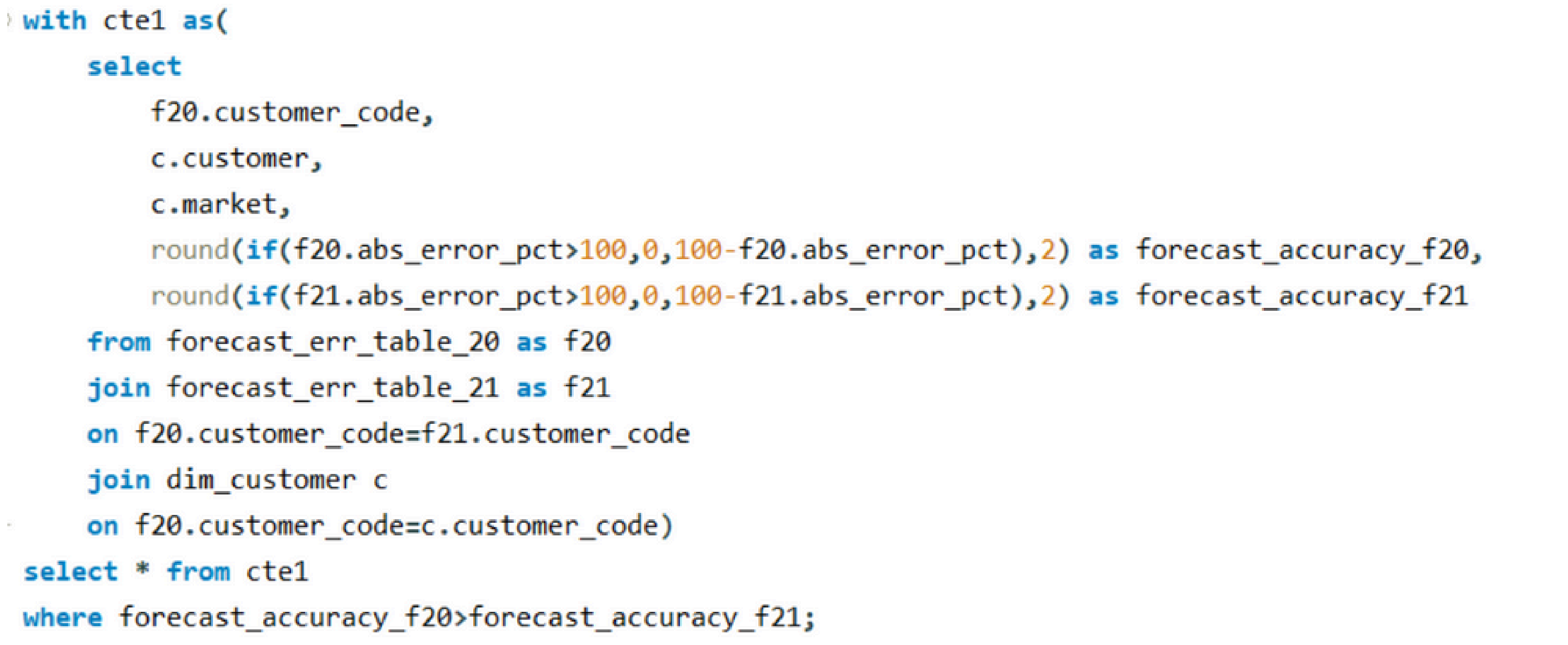
Note: Same can be done with help of Stored Procedure

Generate a report for Supply Chain Manager for identifying customers whose forecast accuracy has dropped from FY 2020 to 2021

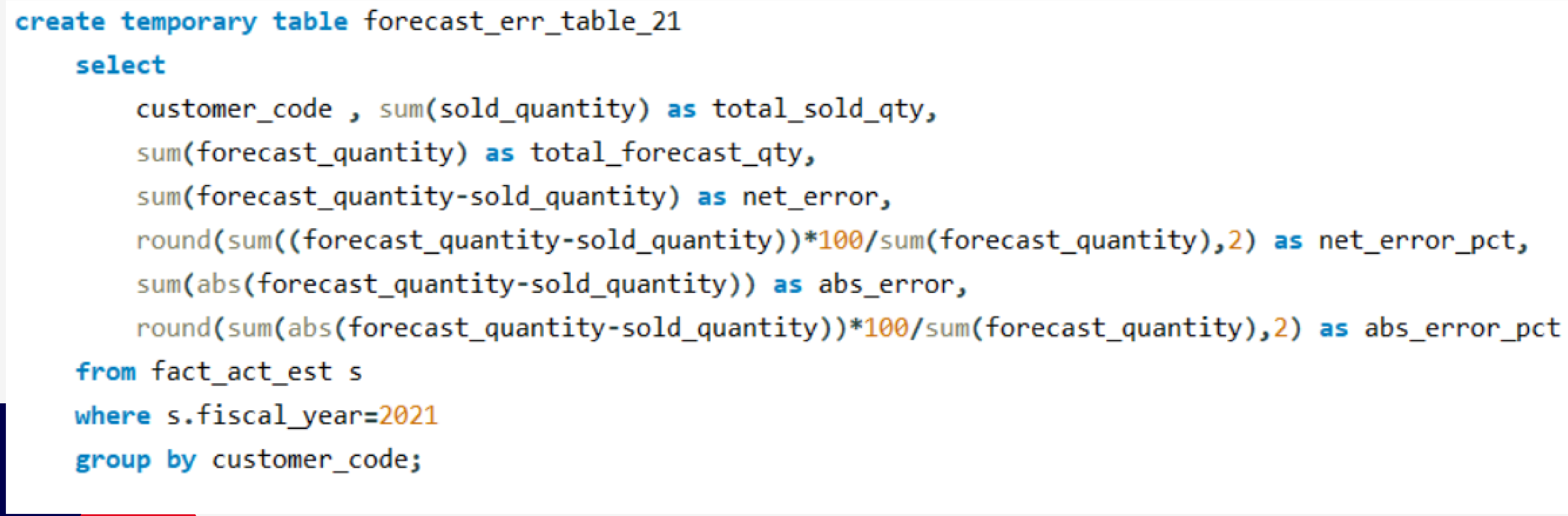
SQL Query



```sql
create temporary table forecast_err_table_20
    select
        customer_code , sum(sold_quantity) as total_sold_qty,
        sum(forecast_quantity) as total_forecast_qty,
        sum(forecast_quantity-sold_quantity) as net_error,
        round(sum((forecast_quantity-sold_quantity))*100/sum(forecast_quantity),2) as net_error_pct,
        sum(abs(forecast_quantity-sold_quantity)) as abs_error,
        round(sum(abs(forecast_quantity-sold_quantity))*100/sum(forecast_quantity),2) as abs_error_pct
    from fact_act_est s
    where s.fiscal_year=2020
    group by customer_code;
```

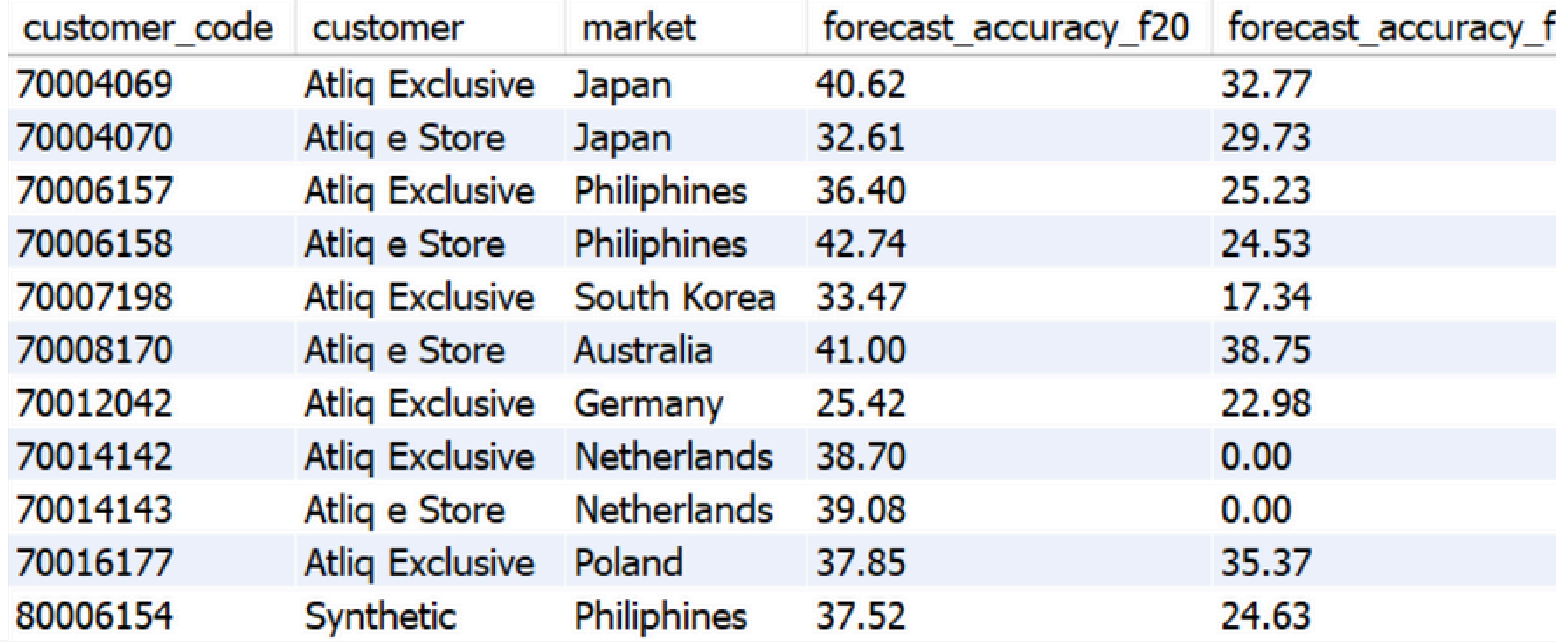


```sql
with cte1 as(
    select
        f20.customer_code,
        c.customer,
        c.market,
        round(if(f20.abs_error_pct>100,0,100-f20.abs_error_pct),2) as forecast_accuracy_f20,
        round(if(f21.abs_error_pct>100,0,100-f21.abs_error_pct),2) as forecast_accuracy_f21
    from forecast_err_table_20 as f20
    join forecast_err_table_21 as f21
    on f20.customer_code=f21.customer_code
    join dim_customer c
    on f20.customer_code=c.customer_code)
select * from cte1
where forecast_accuracy_f20>forecast_accuracy_f21;
```



```sql
create temporary table forecast_err_table_21
    select
        customer_code , sum(sold_quantity) as total_sold_qty,
        sum(forecast_quantity) as total_forecast_qty,
        sum(forecast_quantity-sold_quantity) as net_error,
        round(sum((forecast_quantity-sold_quantity))*100/sum(forecast_quantity),2) as net_error_pct,
        sum(abs(forecast_quantity-sold_quantity)) as abs_error,
        round(sum(abs(forecast_quantity-sold_quantity))*100/sum(forecast_quantity),2) as abs_error_pct
    from fact_act_est s
    where s.fiscal_year=2021
    group by customer_code;
```

Result



customer_code	customer	market	forecast_accuracy_f20	forecast_accuracy_f
70004069	Atliq Exclusive	Japan	40.62	32.77
70004070	Atliq e Store	Japan	32.61	29.73
70006157	Atliq Exclusive	Philiphines	36.40	25.23
70006158	Atliq e Store	Philiphines	42.74	24.53
70007198	Atliq Exclusive	South Korea	33.47	17.34
70008170	Atliq e Store	Australia	41.00	38.75
70012042	Atliq Exclusive	Germany	25.42	22.98
70014142	Atliq Exclusive	Netherlands	38.70	0.00
70014143	Atliq e Store	Netherlands	39.08	0.00
70016177	Atliq Exclusive	Poland	37.85	35.37
80006154	Synthetic	Philiphines	37.52	24.63

Thank you!