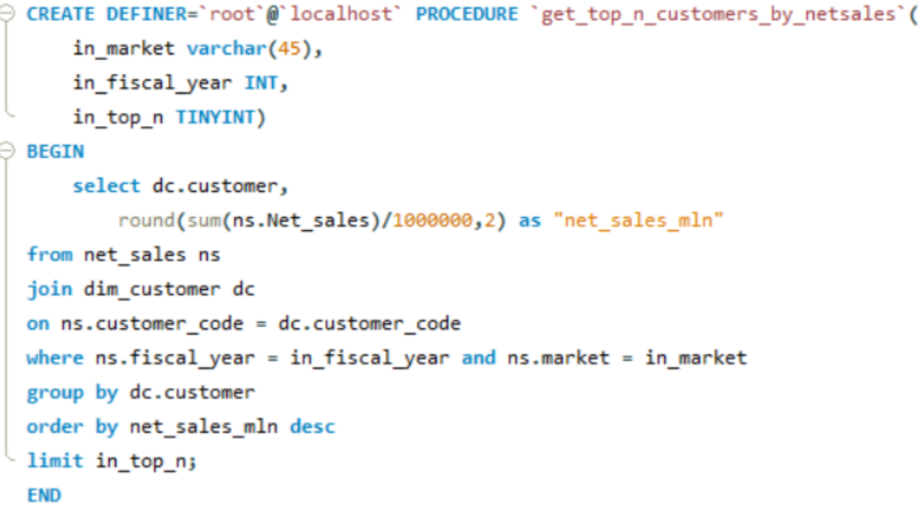
Consumer Domain
Ad-Hoc requests

AtliQ Hardware (imaginary company), a prominent computer hardware manufacturer based in India with international operations, faced challenges due to limited access to actionable insights for swift decision-making. To tackle this, the management highlighted several urgent, insight-driven requirements.

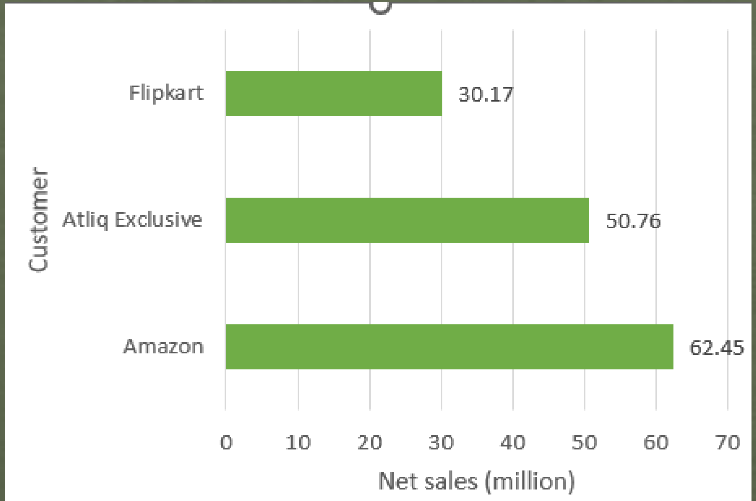
Presented by: Dhwani Thakkar

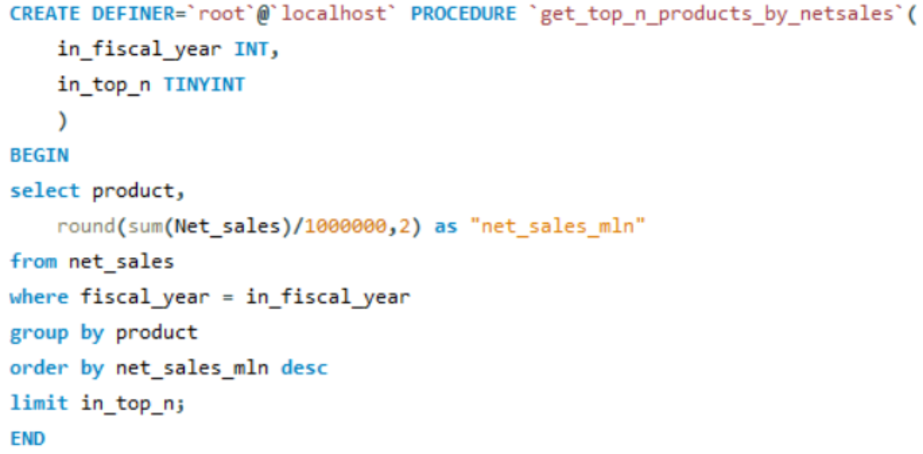
Ad-Hoc analysis

1) Prepare a report for top market, products, customers by net sales for given FY to have a holistic view of financial performance.

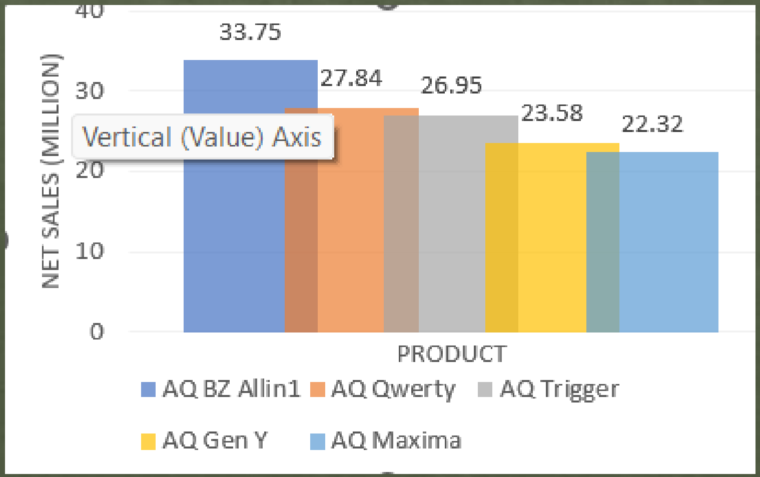


```
CREATE DEFINER=`root`@`localhost` PROCEDURE `get_top_n_customers_by_netsales`(
    in_market varchar(45),
    in_fiscal_year INT,
    in_top_n TINYINT)
BEGIN
    select dc.customer,
        round(sum(ns.Net_sales)/1000000,2) as "net_sales_mln"
from net_sales ns
join dim_customer dc
on ns.customer_code = dc.customer_code
where ns.fiscal_year = in_fiscal_year and ns.market = in_market
group by dc.customer
order by net_sales_mln desc
limit in_top_n;
END
```



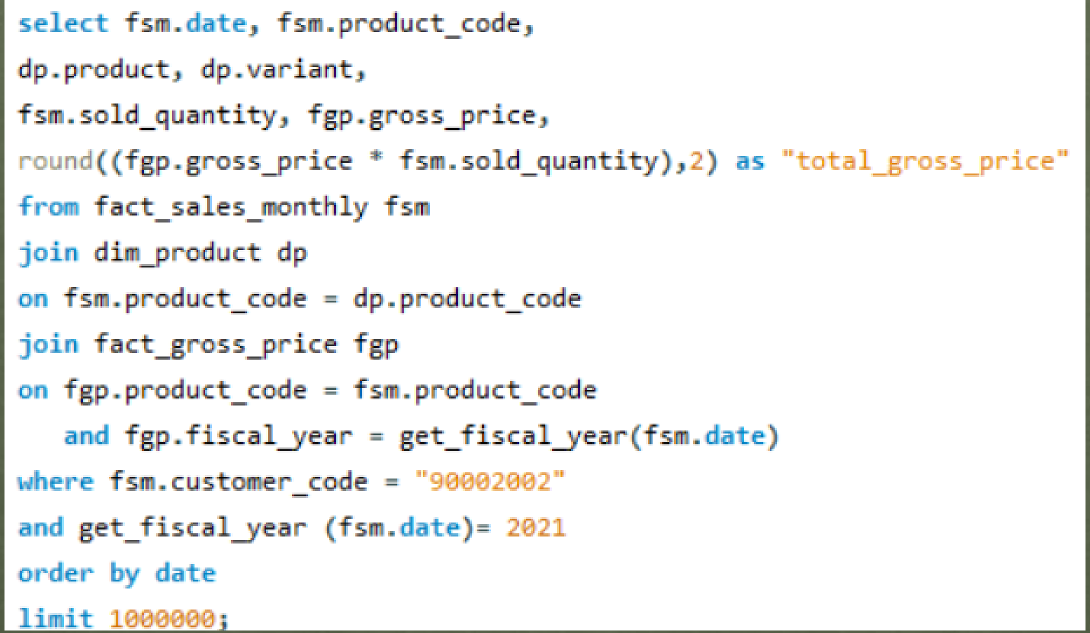


```
CREATE DEFINER=`root`@`localhost` PROCEDURE `get_top_n_products_by_netsales`(
    in_fiscal_year INT,
    in_top_n TINYINT
    )
BEGIN
select product,
    round(sum(Net_sales)/1000000,2) as "net_sales_mln"
from net_sales
where fiscal_year = in_fiscal_year
group by product
order by net_sales_mln desc
limit in_top_n;
END
```

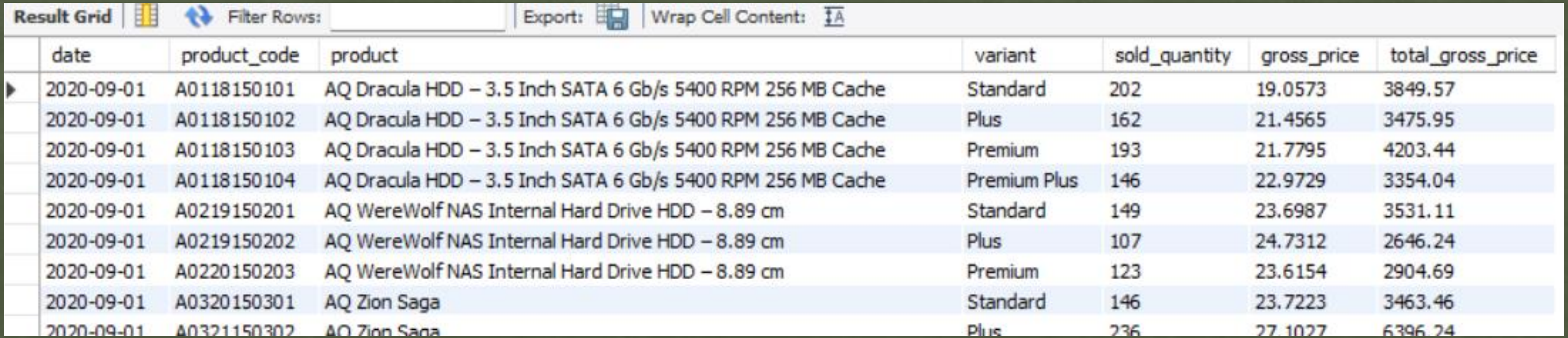


Ad-Hoc analysis

2) Generate a report with month, product name, variant, sold quantity, gross price per item, gross price total for Croma



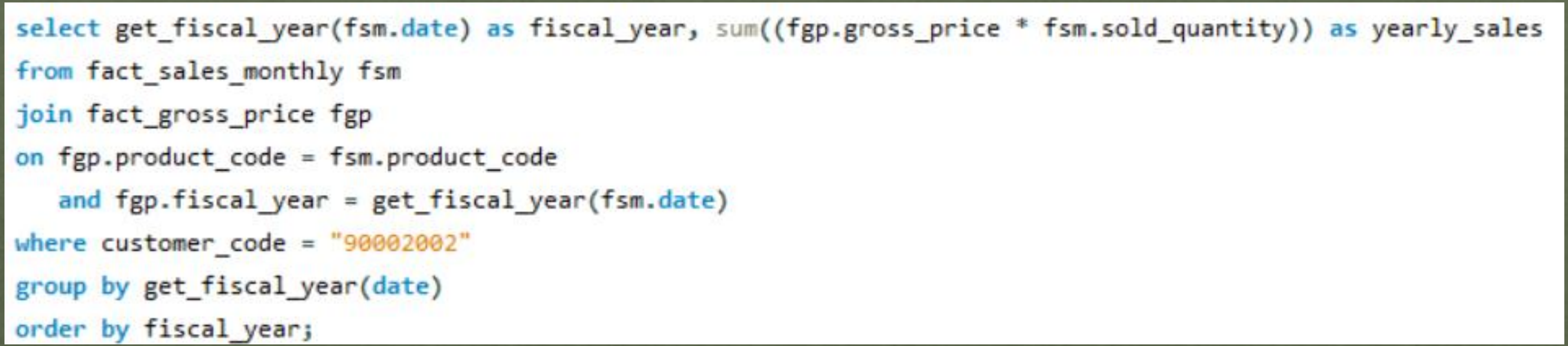
```
select fsm.date, fsm.product_code,
dp.product, dp.variant,
fsm.sold_quantity, fgp.gross_price,
round((fgp.gross_price * fsm.sold_quantity),2) as "total_gross_price"
from fact_sales_monthly fsm
join dim_product dp
on fsm.product_code = dp.product_code
join fact_gross_price fgp
on fgp.product_code = fsm.product_code
   and fgp.fiscal_year = get_fiscal_year(fsm.date)
where fsm.customer_code = "90002002"
and get_fiscal_year (fsm.date)= 2021
order by date
limit 1000000;
```



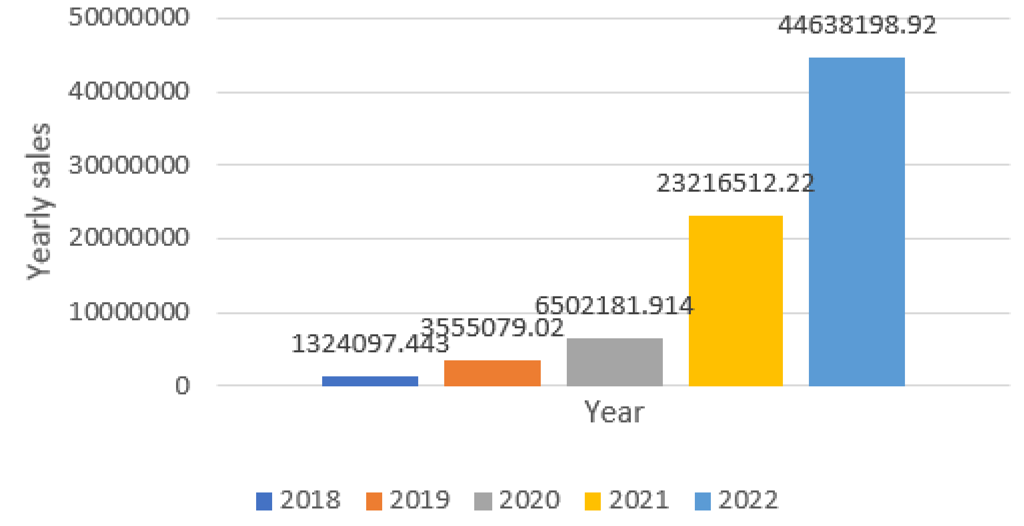
date	product_code	product	variant	sold_quantity	gross_price	total_gross_price
2020-09-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 RPM 256 MB Cache	Standard	202	19.0573	3849.57
2020-09-01	A0118150102	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 RPM 256 MB Cache	Plus	162	21.4565	3475.95
2020-09-01	A0118150103	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 RPM 256 MB Cache	Premium	193	21.7795	4203.44
2020-09-01	A0118150104	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 RPM 256 MB Cache	Premium Plus	146	22.9729	3354.04
2020-09-01	A0219150201	AQ WereWolf NAS Internal Hard Drive HDD – 8.89 cm	Standard	149	23.6987	3531.11
2020-09-01	A0219150202	AQ WereWolf NAS Internal Hard Drive HDD – 8.89 cm	Plus	107	24.7312	2646.24
2020-09-01	A0220150203	AQ WereWolf NAS Internal Hard Drive HDD – 8.89 cm	Premium	123	23.6154	2904.69
2020-09-01	A0320150301	AQ Zion Saga	Standard	146	23.7223	3463.46
2020-09-01	A0321150302	AQ Zion Saga	Plus	236	27.1027	6396.24

Ad-Hoc analysis

3) Generate a yearly report for Croma India where there are two columns a. Fiscal Year b. Total Gross Sales amount In that year from Croma

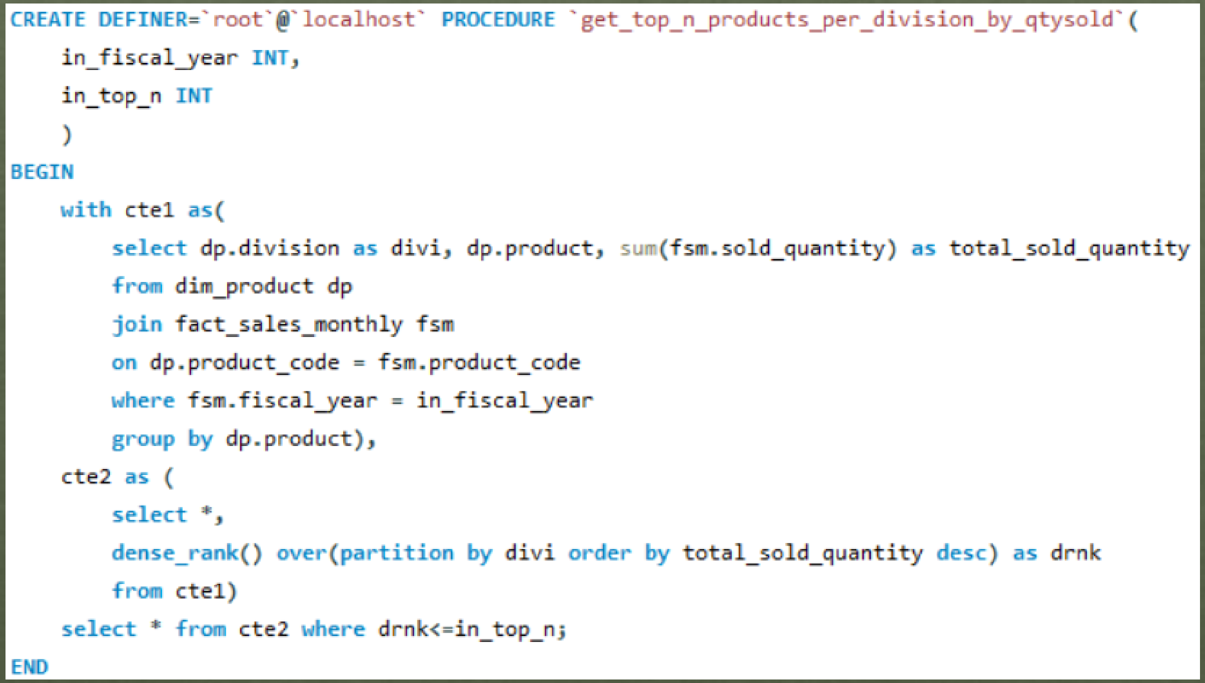


```
select get_fiscal_year(fsm.date) as fiscal_year, sum((fgp.gross_price * fsm.sold_quantity)) as yearly_sales
from fact_sales_monthly fsm
join fact_gross_price fgp
on fgp.product_code = fsm.product_code
   and fgp.fiscal_year = get_fiscal_year(fsm.date)
where customer_code = "90002002"
group by get_fiscal_year(date)
order by fiscal_year;
```

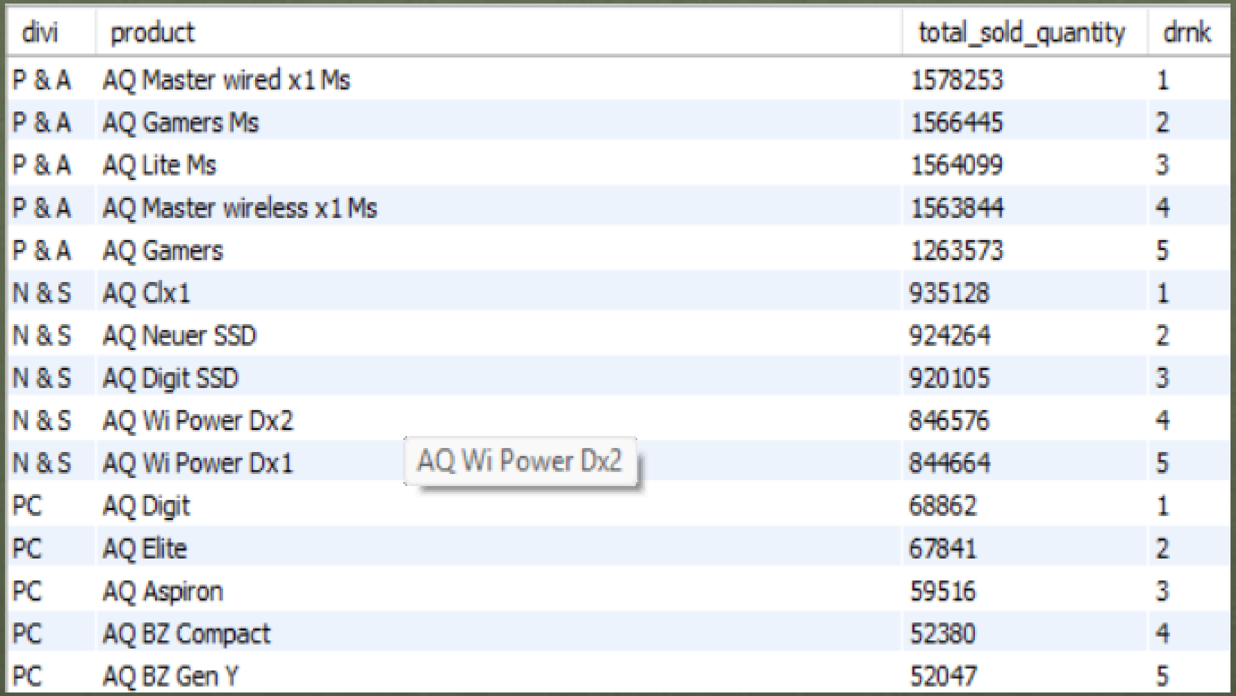


Ad-Hoc analysis

4) Get top n products per division by quantity sold



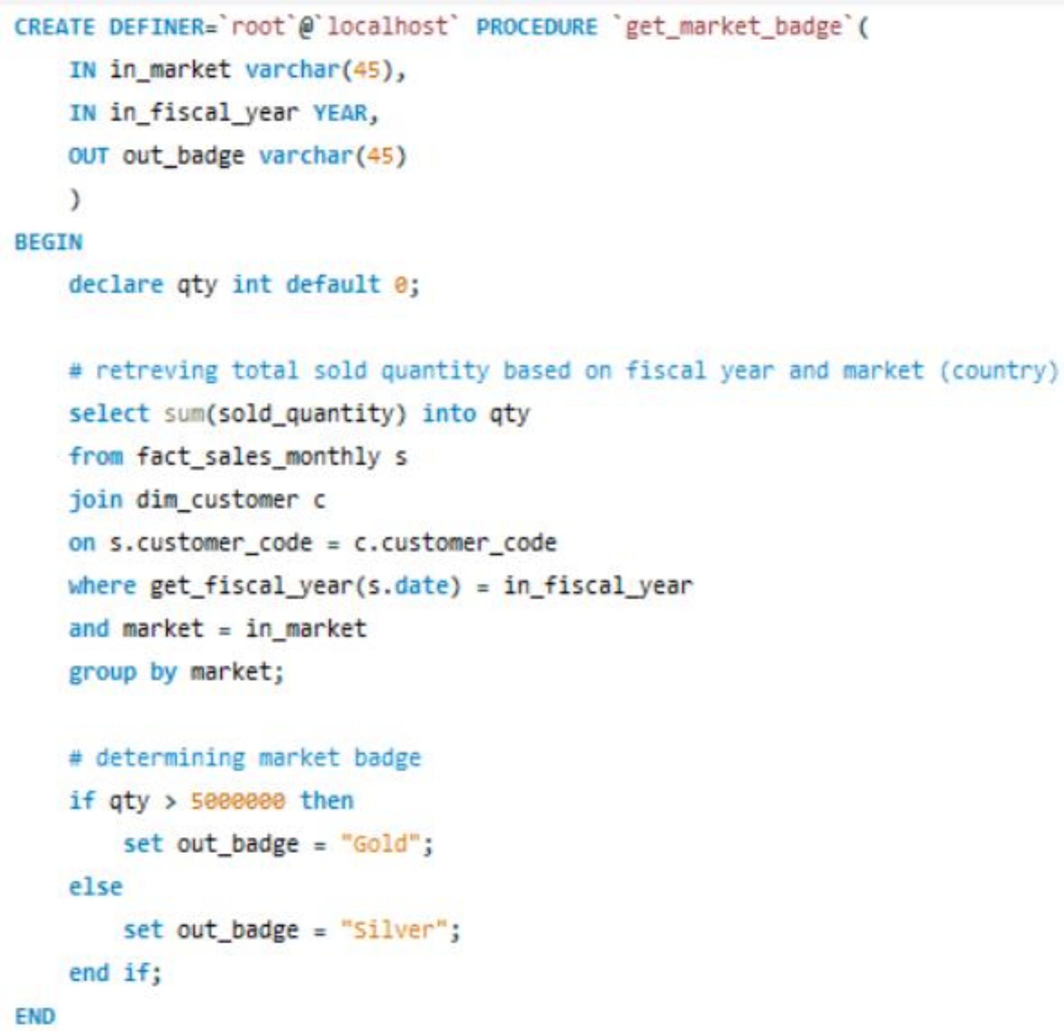
```
CREATE DEFINER=`root`@`localhost` PROCEDURE `get_top_n_products_per_division_by_qtysold`(
    in_fiscal_year INT,
    in_top_n INT
    )
BEGIN
    with cte1 as(
        select dp.division as divi, dp.product, sum(fsm.sold_quantity) as total_sold_quantity
        from dim_product dp
        join fact_sales_monthly fsm
        on dp.product_code = fsm.product_code
        where fsm.fiscal_year = in_fiscal_year
        group by dp.product),
    cte2 as (
        select *,
        dense_rank() over(partition by divi order by total_sold_quantity desc) as drnk
        from cte1)
    select * from cte2 where drnk<=in_top_n;
END
```



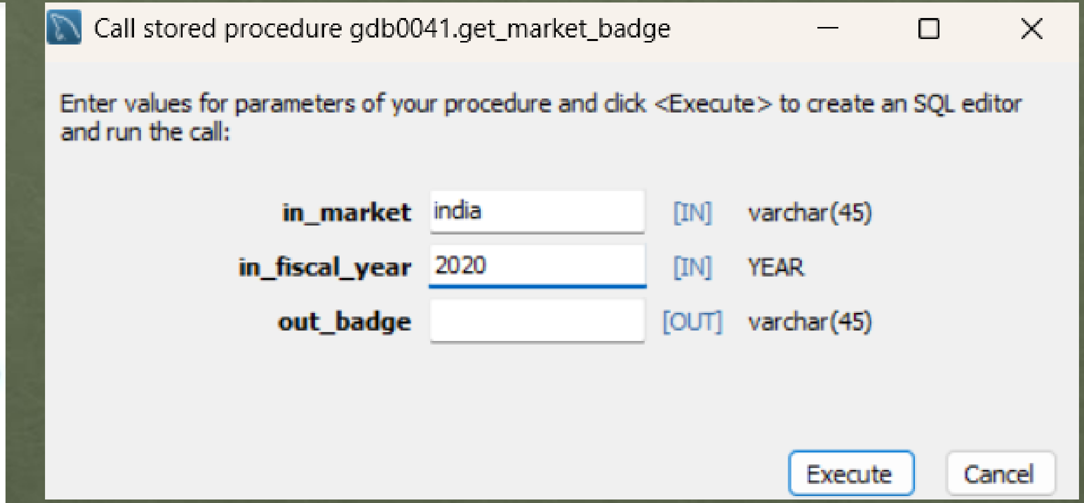
divi	product	total_sold_quantity	drnk
P & A	AQ Master wired x1 Ms	1578253	1
P & A	AQ Gamers Ms	1566445	2
P & A	AQ Lite Ms	1564099	3
P & A	AQ Master wireless x1 Ms	1563844	4
P & A	AQ Gamers	1263573	5
N & S	AQ Clx1	935128	1
N & S	AQ Neuer SSD	924264	2
N & S	AQ Digit SSD	920105	3
N & S	AQ Wi Power Dx2	846576	4
N & S	AQ Wi Power Dx1	844664	5
PC	AQ Digit	68862	1
PC	AQ Elite	67841	2
PC	AQ Aspiron	59516	3
PC	AQ BZ Compact	52380	4
PC	AQ BZ Gen Y	52047	5

Ad-Hoc analysis

5) Determine market badge based on the total sold quantity based on fiscal year and country. If sold quantity > 5million → gold, Else Silver



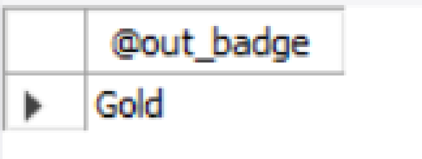
```
CREATE DEFINER=`root`@`localhost` PROCEDURE `get_market_badge`(
    IN in_market varchar(45),
    IN in_fiscal_year YEAR,
    OUT out_badge varchar(45)
    )
BEGIN
    declare qty int default 0;

    # retreving total sold quantity based on fiscal year and market (country)
    select sum(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 market = in_market
    group by market;

    # determining market badge
    if qty > 5000000 then
        set out_badge = "Gold";
    else
        set out_badge = "Silver";
    end if;
END
```



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]	YEAR
out_badge		[OUT]	varchar(45)



	@out_badge
▶	Gold