CONSUMER GOODS ANALYSIS

Unveiling Insights: Leveraging Data for Strategic Decision-Making

Problem Statement:

Atliq Hardware's, a leading computer hardware producer, faced a critical challenge. They needed quick and data-informed decisions to stay competitive in the ever-evolving market. The management noticed that they were missing crucial insights for strategic moves.

To tackle this, they hired a data analytics team, and I took on the SQL challenge. My aim? Answer 10 ad-hoc requests, translating data into valuable insights for strategic decision-making.

Key Insights:

•Significant increase in unique products, with 334 in 2021 compared to 245 in 2020.

•The "Notebook" segment boasts the highest product count, with 129 products.

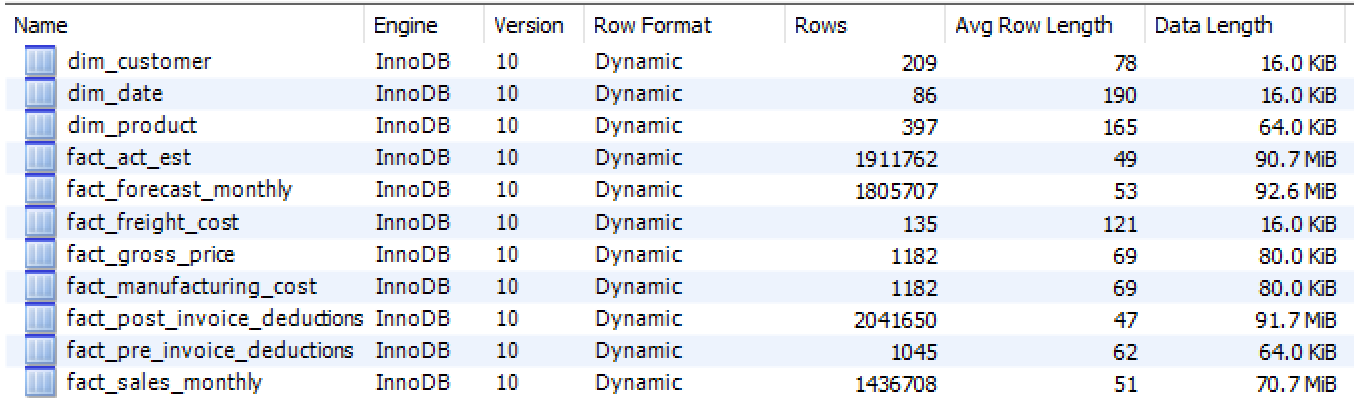
•The "Accessories" segment saw a notable increase in product counts, with 34 more products in 2021 compared to 2020.

•"Flipkart" leads with the highest average pre-invoice discount percentage at 30.83%.

•Fiscal year 2020's low sales in March and April improved in fiscal year 2021.

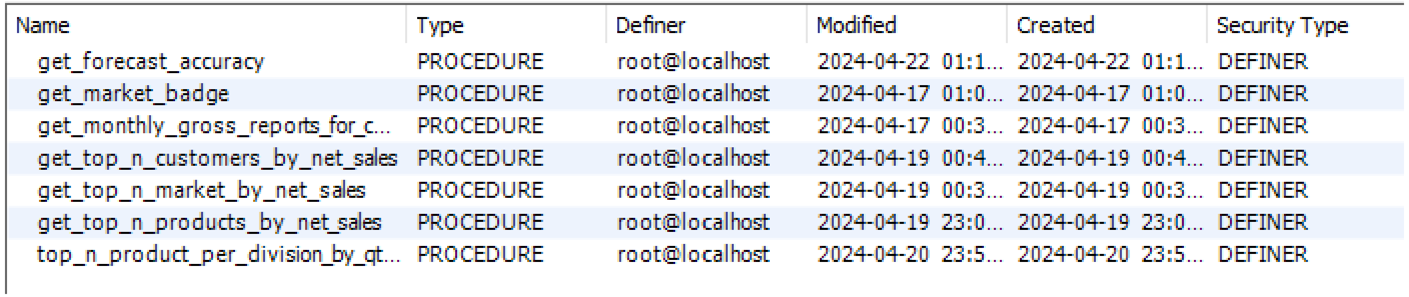
•The "Retailer" channel contributes to 73.22% of gross sales, making it the key driver.

Database Overview



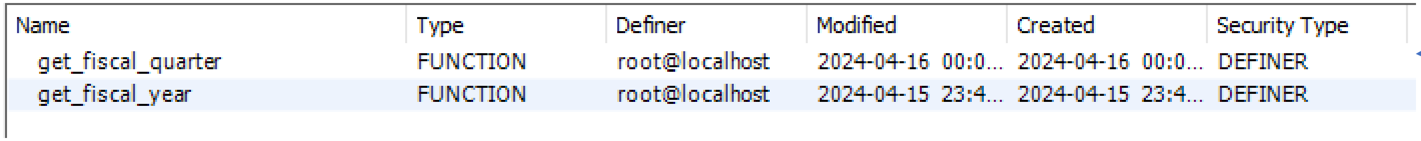
Name	Engine	Version	Row Format	Rows	Avg Row Length	Data Length
dim_customer	InnoDB	10	Dynamic	209	78	16.0 KiB
dim_date	InnoDB	10	Dynamic	86	190	16.0 KiB
dim_product	InnoDB	10	Dynamic	397	165	64.0 KiB
fact_act_est	InnoDB	10	Dynamic	1911762	49	90.7 MiB
fact_forecast_monthly	InnoDB	10	Dynamic	1805707	53	92.6 MiB
fact_freight_cost	InnoDB	10	Dynamic	135	121	16.0 KiB
fact_gross_price	InnoDB	10	Dynamic	1182	69	80.0 KiB
fact_manufacturing_cost	InnoDB	10	Dynamic	1182	69	80.0 KiB
fact_post_invoice_deductions	InnoDB	10	Dynamic	2041650	47	91.7 MiB
fact_pre_invoice_deductions	InnoDB	10	Dynamic	1045	62	64.0 KiB
fact_sales_monthly	InnoDB	10	Dynamic	1436708	51	70.7 MiB

Tables



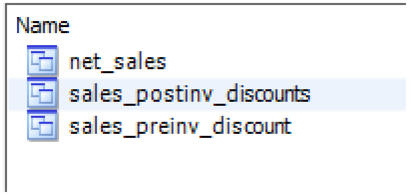
Name	Type	Definer	Modified	Created	Security Type
get_forecast_accuracy	PROCEDURE	root@localhost	2024-04-22 01:1...	2024-04-22 01:1...	DEFINER
get_market_badge	PROCEDURE	root@localhost	2024-04-17 01:0...	2024-04-17 01:0...	DEFINER
get_monthly_gross_reports_for_c...	PROCEDURE	root@localhost	2024-04-17 00:3...	2024-04-17 00:3...	DEFINER
get_top_n_customers_by_net_sales	PROCEDURE	root@localhost	2024-04-19 00:4...	2024-04-19 00:4...	DEFINER
get_top_n_market_by_net_sales	PROCEDURE	root@localhost	2024-04-19 00:3...	2024-04-19 00:3...	DEFINER
get_top_n_products_by_net_sales	PROCEDURE	root@localhost	2024-04-19 23:0...	2024-04-19 23:0...	DEFINER
top_n_product_per_division_by_qt...	PROCEDURE	root@localhost	2024-04-20 23:5...	2024-04-20 23:5...	DEFINER

Stored Procedures



Name	Type	Definer	Modified	Created	Security Type
get_fiscal_quarter	FUNCTION	root@localhost	2024-04-16 00:0...	2024-04-16 00:0...	DEFINER
get_fiscal_year	FUNCTION	root@localhost	2024-04-15 23:4...	2024-04-15 23:4...	DEFINER

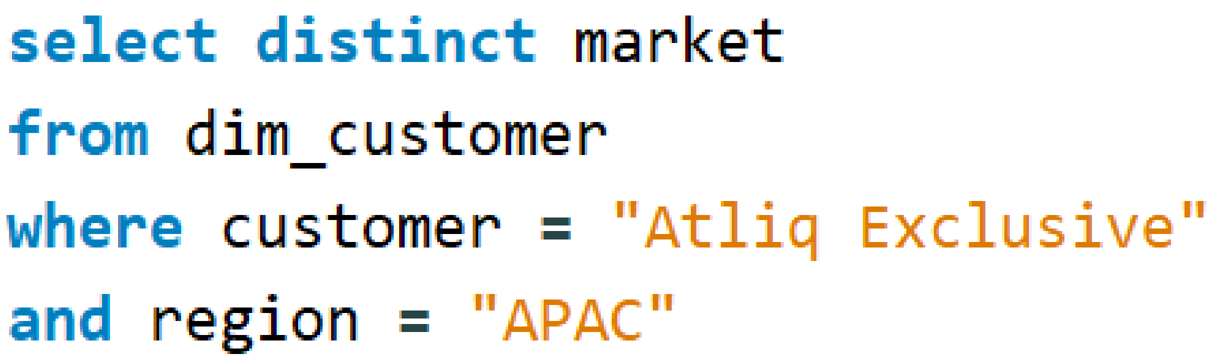
Functions



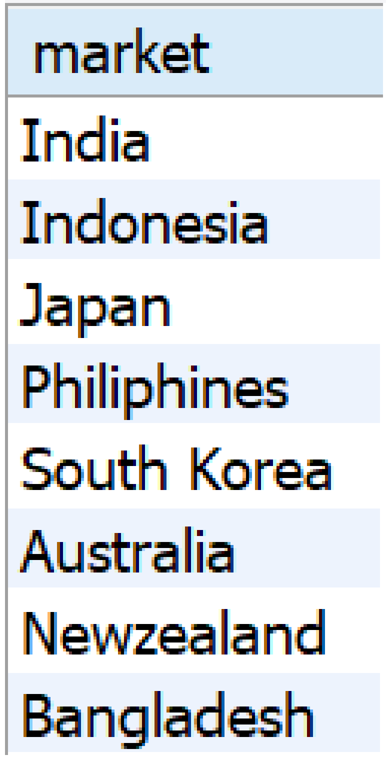
Name
net_sales
sales_postinv_discounts
sales_preinv_discount

Views

Provide the list of market in which customer “Atliq Exclusive” operates its business in APAC region

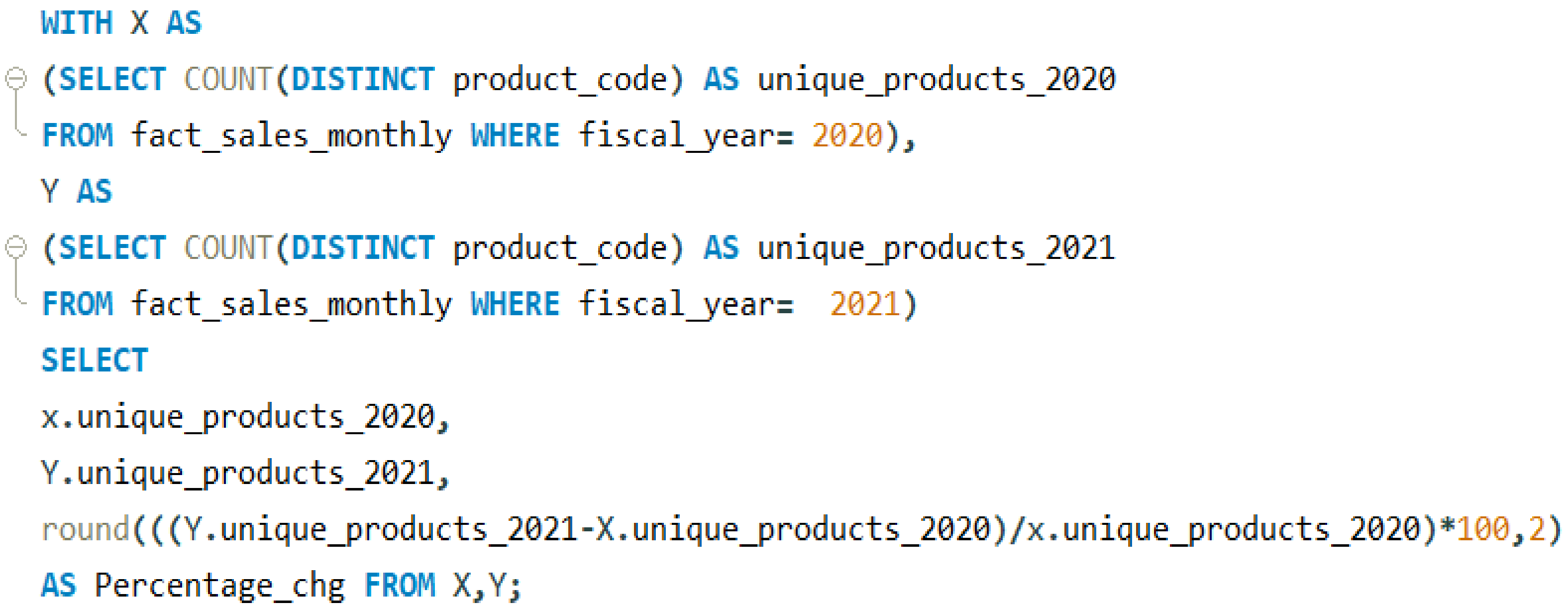


```
select distinct market
from dim_customer
where customer = "Atliq Exclusive"
and region = "APAC"
```

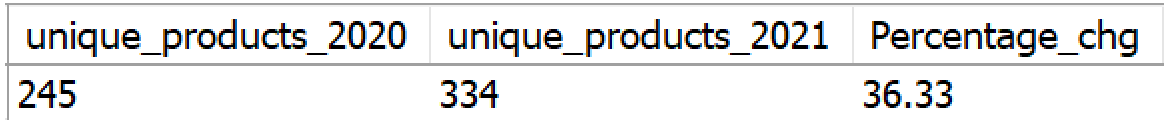


market
India
Indonesia
Japan
Philiphines
South Korea
Australia
Newzealand
Bangladesh

What is the percentage of unique product increase in 2021vs.2020?



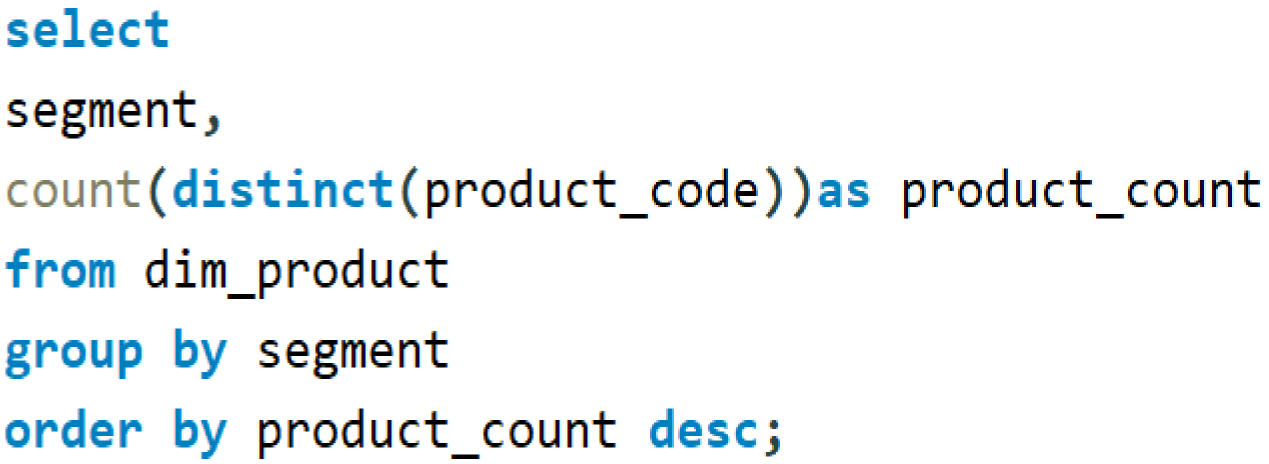
```
WITH X AS
(SELECT COUNT(DISTINCT product_code) AS unique_products_2020
FROM fact_sales_monthly WHERE fiscal_year= 2020),
Y AS
(SELECT COUNT(DISTINCT product_code) AS unique_products_2021
FROM fact_sales_monthly WHERE fiscal_year=  2021)
SELECT
x.unique_products_2020,
Y.unique_products_2021,
round(((Y.unique_products_2021-X.unique_products_2020)/x.unique_products_2020)*100,2)
AS Percentage_chg FROM X,Y;
```



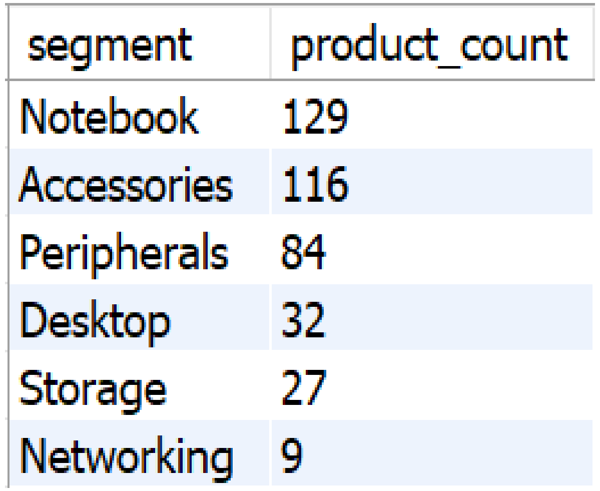
unique_products_2020	unique_products_2021	Percentage_chg
245	334	36.33

Provide a report with all the unique product counts for each segment and sort them in descending order of product counts.

The final output contains 2 fields, segment product count.



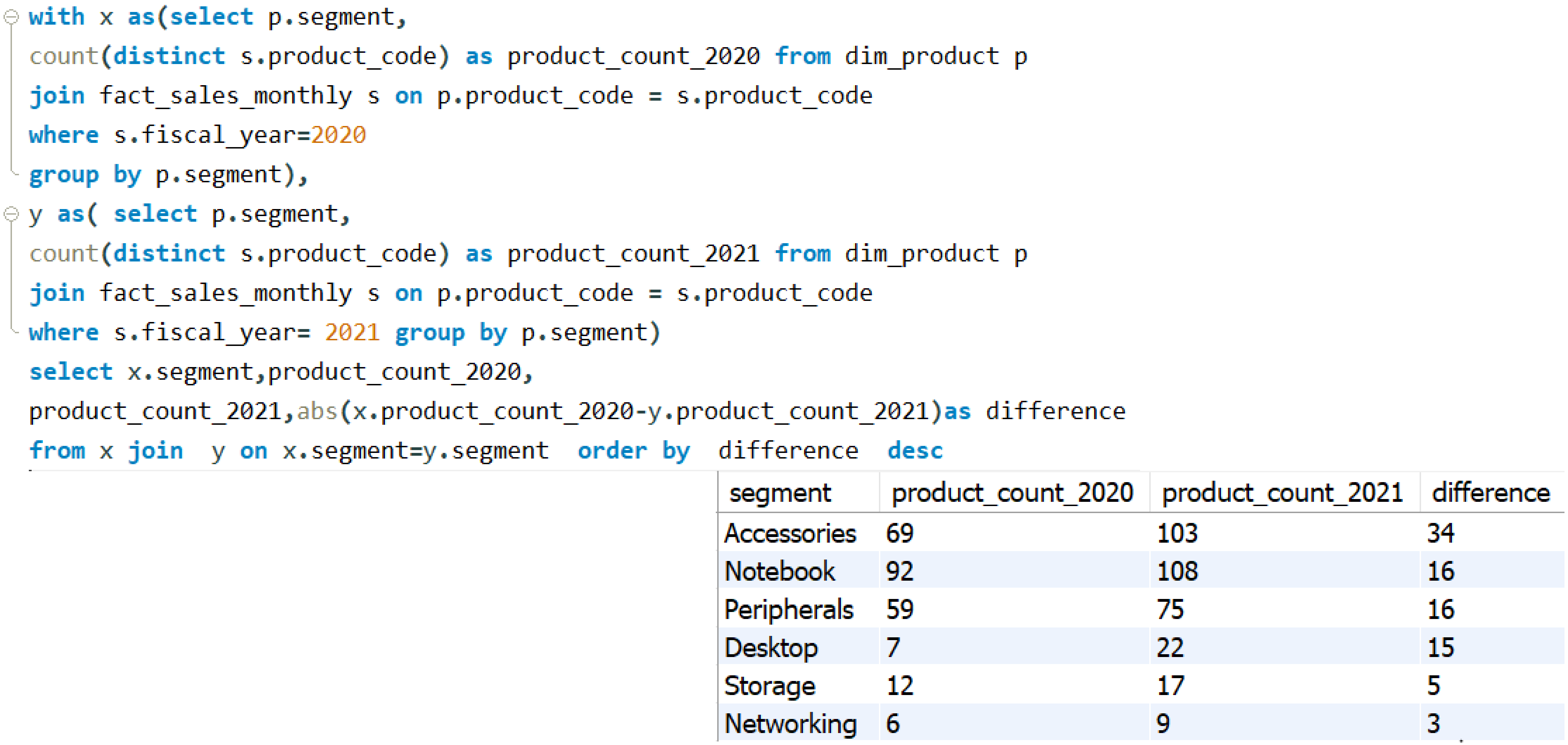
```
select
segment,
count(distinct(product_code))as product_count
from dim_product
group by segment
order by product_count desc;
```



segment	product_count
Notebook	129
Accessories	116
Peripherals	84
Desktop	32
Storage	27
Networking	9

Follow-up: Which segment had the most increase in unique products in 2021 vs 2020?

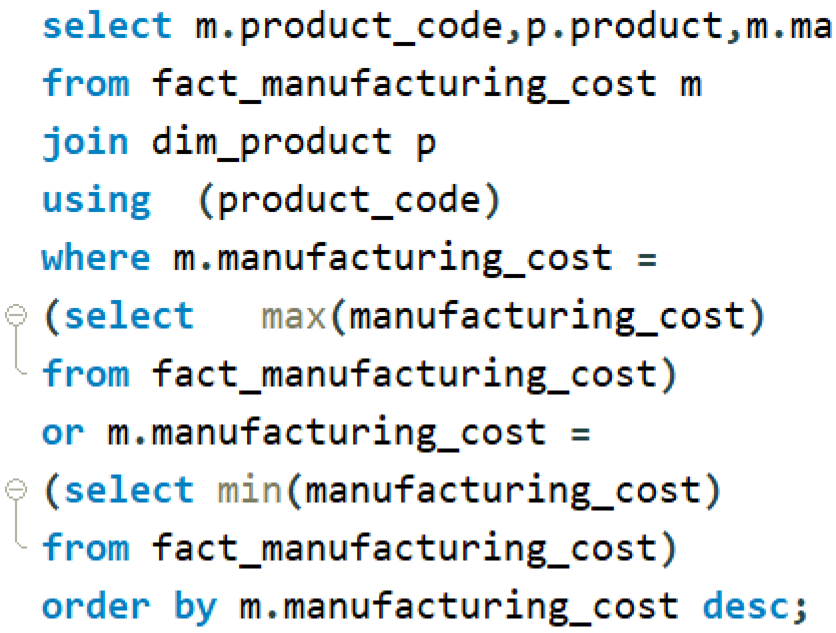
The final output contains these fields, segment, product count 2020 product_count_202I difference.



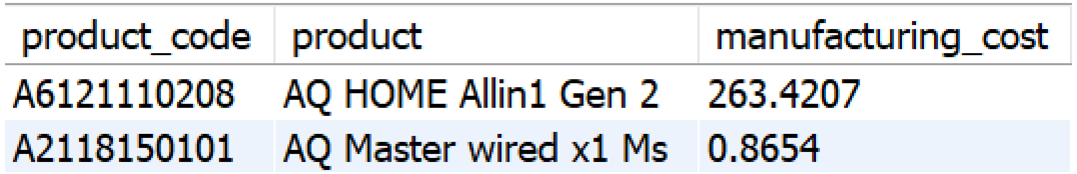
```
with x as(select p.segment,
count(distinct s.product_code) as product_count_2020 from dim_product p
join fact_sales_monthly s on p.product_code = s.product_code
where s.fiscal_year=2020
group by p.segment),
y as( select p.segment,
count(distinct s.product_code) as product_count_2021 from dim_product p
join fact_sales_monthly s on p.product_code = s.product_code
where s.fiscal_year= 2021 group by p.segment)
select x.segment,product_count_2020,
product_count_2021,abs(x.product_count_2020-y.product_count_2021)as difference
from x join  y on x.segment=y.segment  order by  difference  desc
```

segment	product_count_2020	product_count_2021	difference
Accessories	69	103	34
Notebook	92	108	16
Peripherals	59	75	16
Desktop	7	22	15
Storage	12	17	5
Networking	6	9	3

Get the products The final output that have the highest and lowest manufacturing costs should contain these fields, product_code, product

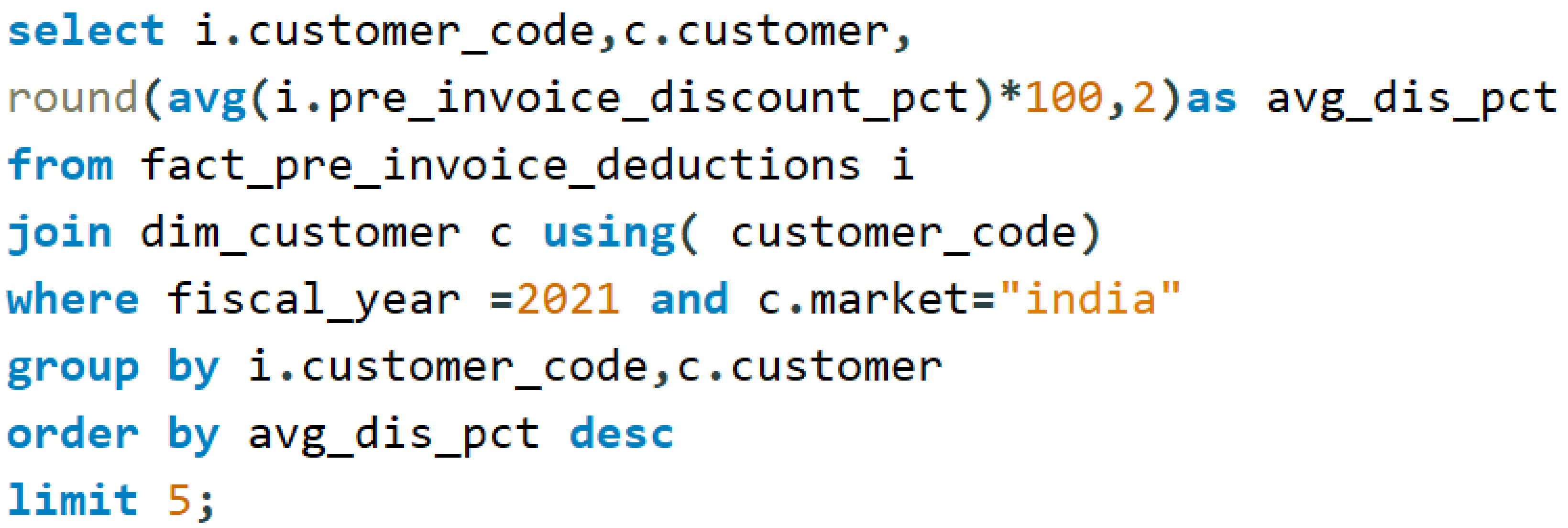


```sql
select m.product_code,p.product,m.manufacturing_cost
from fact_manufacturing_cost m
join dim_product p
using  (product_code)
where m.manufacturing_cost =
(select   max(manufacturing_cost)
from fact_manufacturing_cost)
or m.manufacturing_cost =
(select min(manufacturing_cost)
from fact_manufacturing_cost)
order by m.manufacturing_cost desc;
```

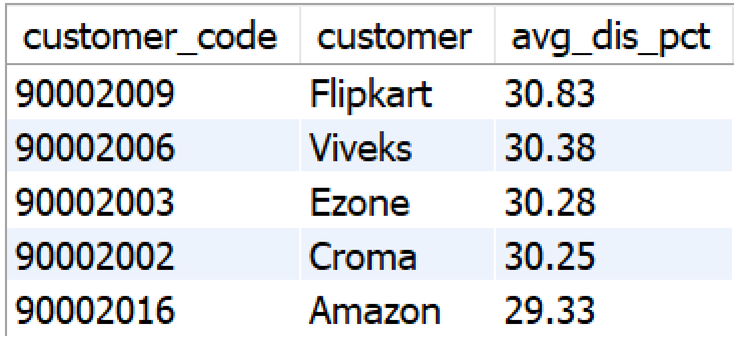


product_code	product	manufacturing_cost
A6121110208	AQ HOME Allin1 Gen 2	263.4207
A2118150101	AQ Master wired x1 Ms	0.8654

Generate a report which contains the top 5 customers who received an average high pre invoice discount_pct for the fiscal year 202I and in the Indian market. The final output contains these fields customer_code customer average_discount_percentage.

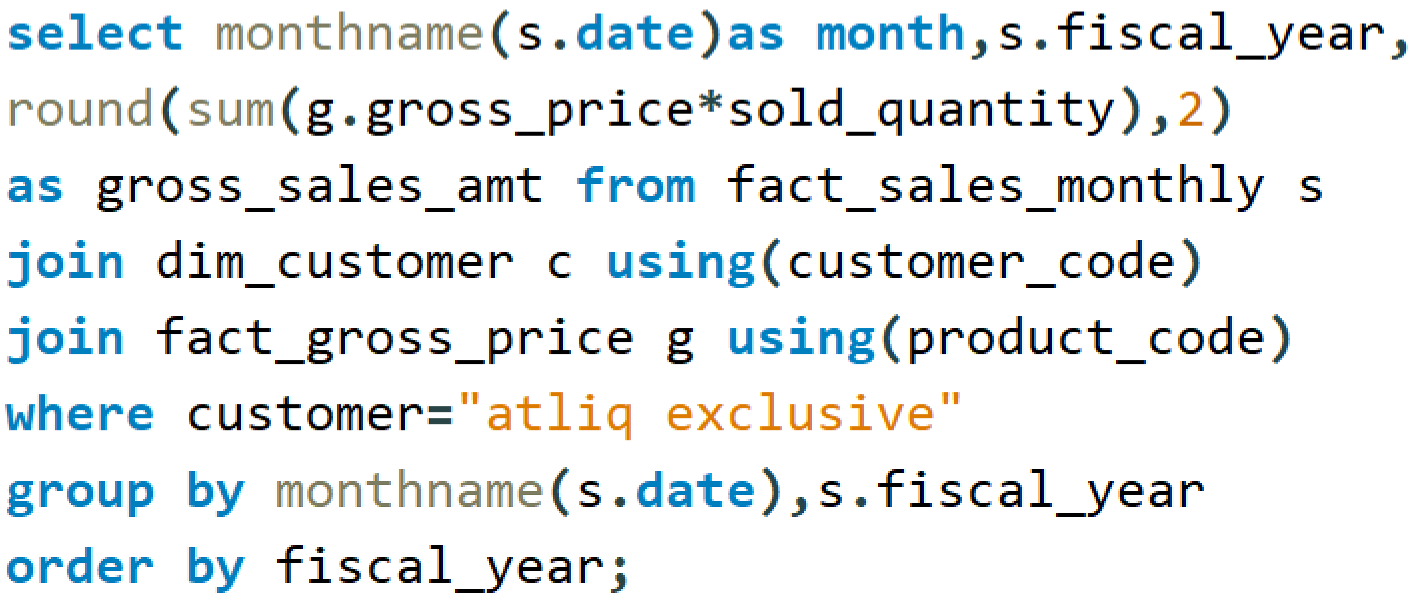


```sql
select i.customer_code,c.customer,
round(avg(i.pre_invoice_discount_pct)*100,2)as avg_dis_pct
from fact_pre_invoice_deductions i
join dim_customer c using( customer_code)
where fiscal_year =2021 and c.market="india"
group by i.customer_code,c.customer
order by avg_dis_pct desc
limit 5;
```

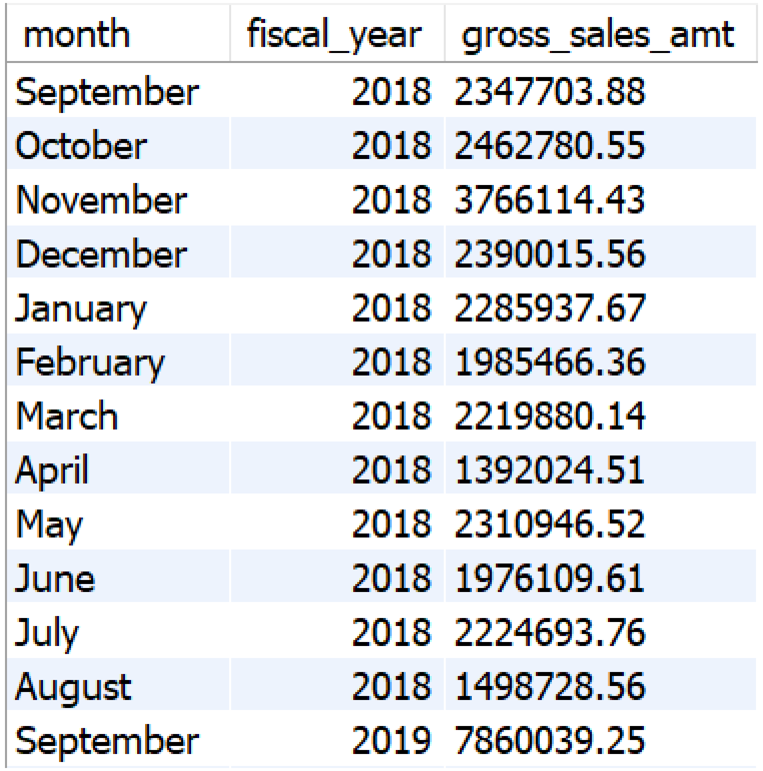


customer_code	customer	avg_dis_pct
90002009	Flipkart	30.83
90002006	Viveks	30.38
90002003	Ezone	30.28
90002002	Croma	30.25
90002016	Amazon	29.33

Get the complete report of the Gross sales amount for the customer "Atliq Exclusive "for each month. This analysis helps to get an idea of low and high-performing months and take strategic decisions. The final report contains these columns: Month Year Gross sales Amount.

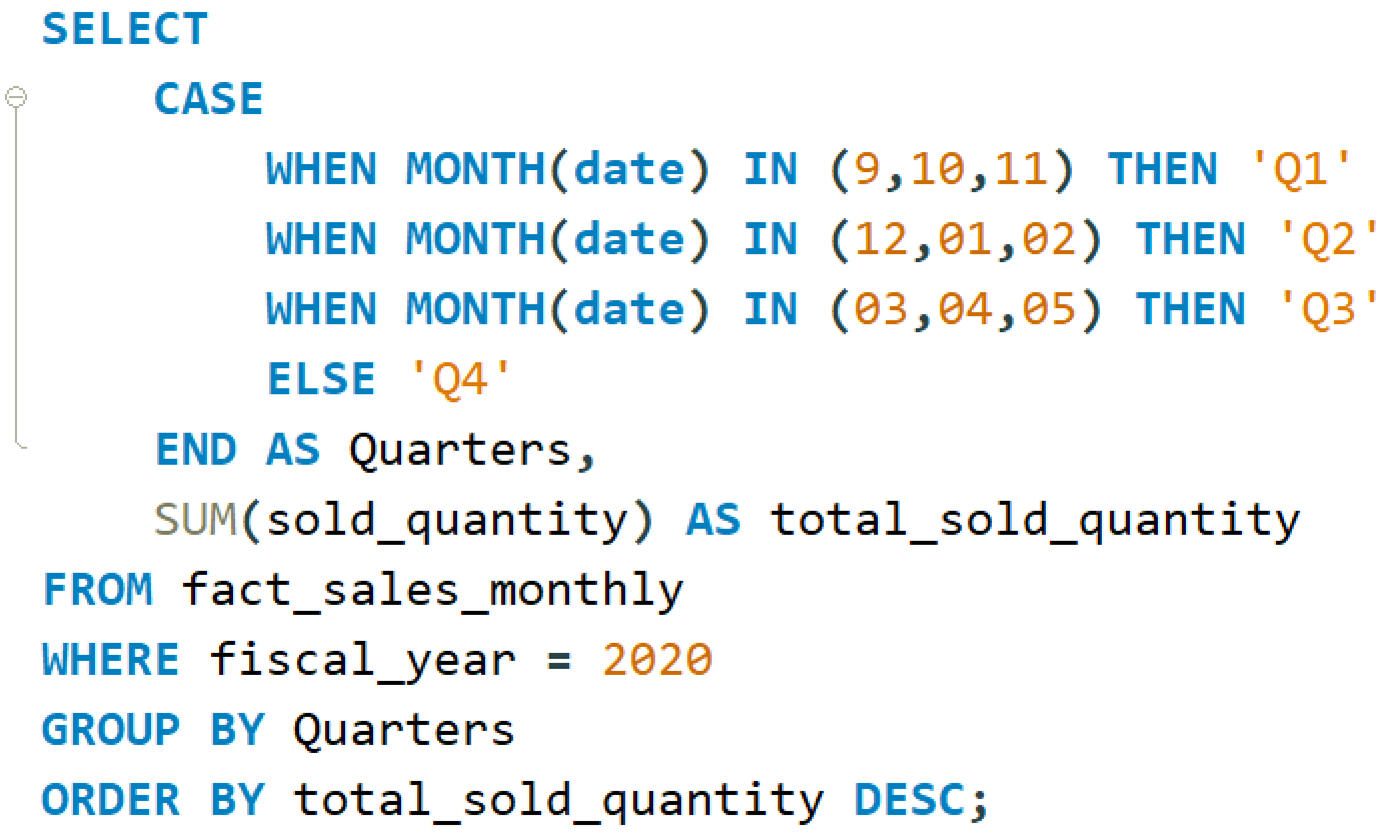


```sql
select monthname(s.date)as month,s.fiscal_year,
round(sum(g.gross_price*sold_quantity),2)
as gross_sales_amt from fact_sales_monthly s
join dim_customer c using(customer_code)
join fact_gross_price g using(product_code)
where customer="atliq exclusive"
group by monthname(s.date),s.fiscal_year
order by fiscal_year;
```

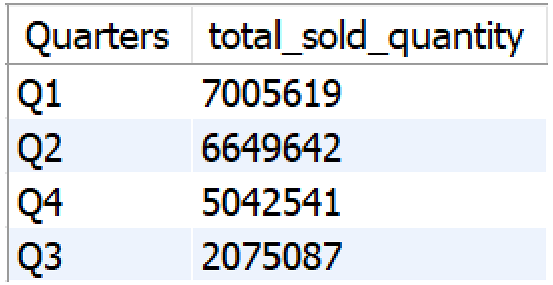


month	fiscal_year	gross_sales_amt
September	2018	2347703.88
October	2018	2462780.55
November	2018	3766114.43
December	2018	2390015.56
January	2018	2285937.67
February	2018	1985466.36
March	2018	2219880.14
April	2018	1392024.51
May	2018	2310946.52
June	2018	1976109.61
July	2018	2224693.76
August	2018	1498728.56
September	2019	7860039.25

Which quarter of 2020, got the maximum total_sold_quantity? The final output contains these fields sorted by the total_sold_quantiy, Quarter, total_sold_quantity.

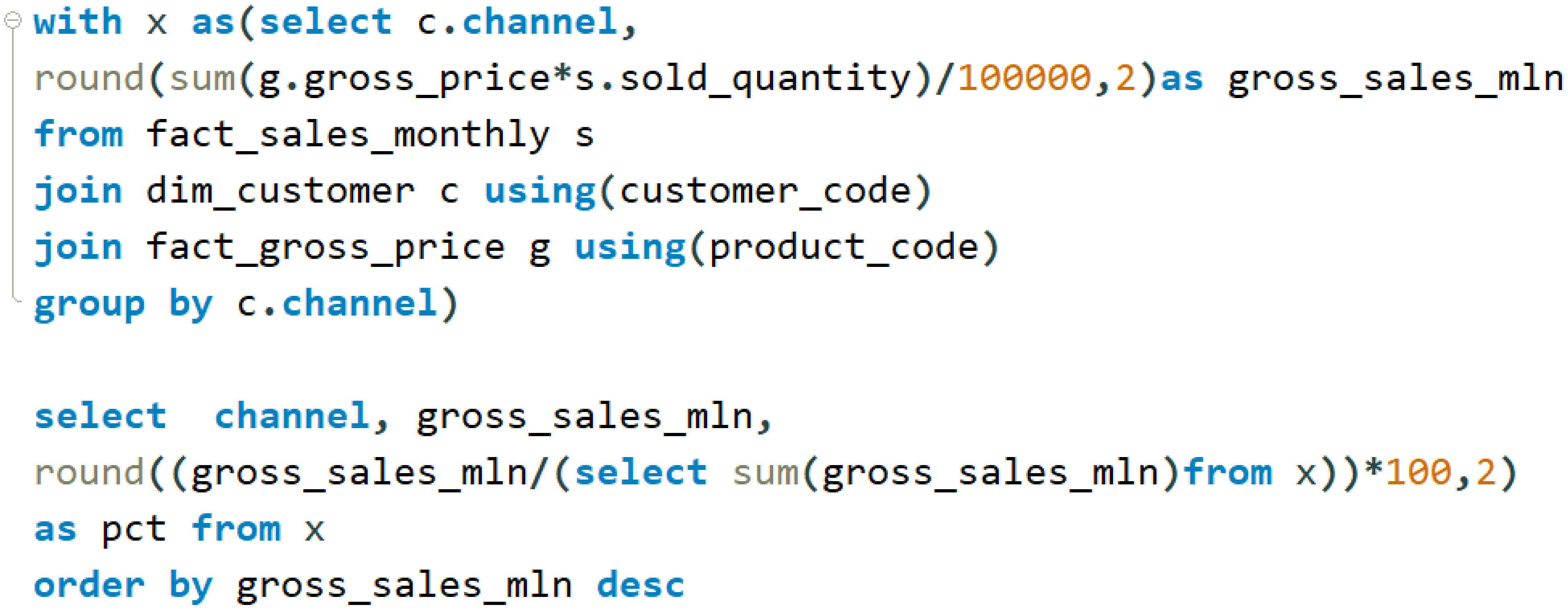


```
SELECT
    CASE
        WHEN MONTH(date) IN (9,10,11) THEN 'Q1'
        WHEN MONTH(date) IN (12,01,02) THEN 'Q2'
        WHEN MONTH(date) IN (03,04,05) THEN 'Q3'
        ELSE 'Q4'
    END AS Quarters,
    SUM(sold_quantity) AS total_sold_quantity
FROM fact_sales_monthly
WHERE fiscal_year = 2020
GROUP BY Quarters
ORDER BY total_sold_quantity DESC;
```



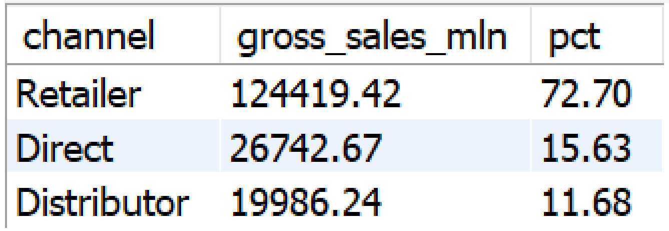
Quarters	total_sold_quantity
Q1	7005619
Q2	6649642
Q4	5042541
Q3	2075087

Which channel helped to bring more gross sales in the fiscal year 2021 and the percentage of contribution? The final output contains these fields, channel gross sales min percentage.



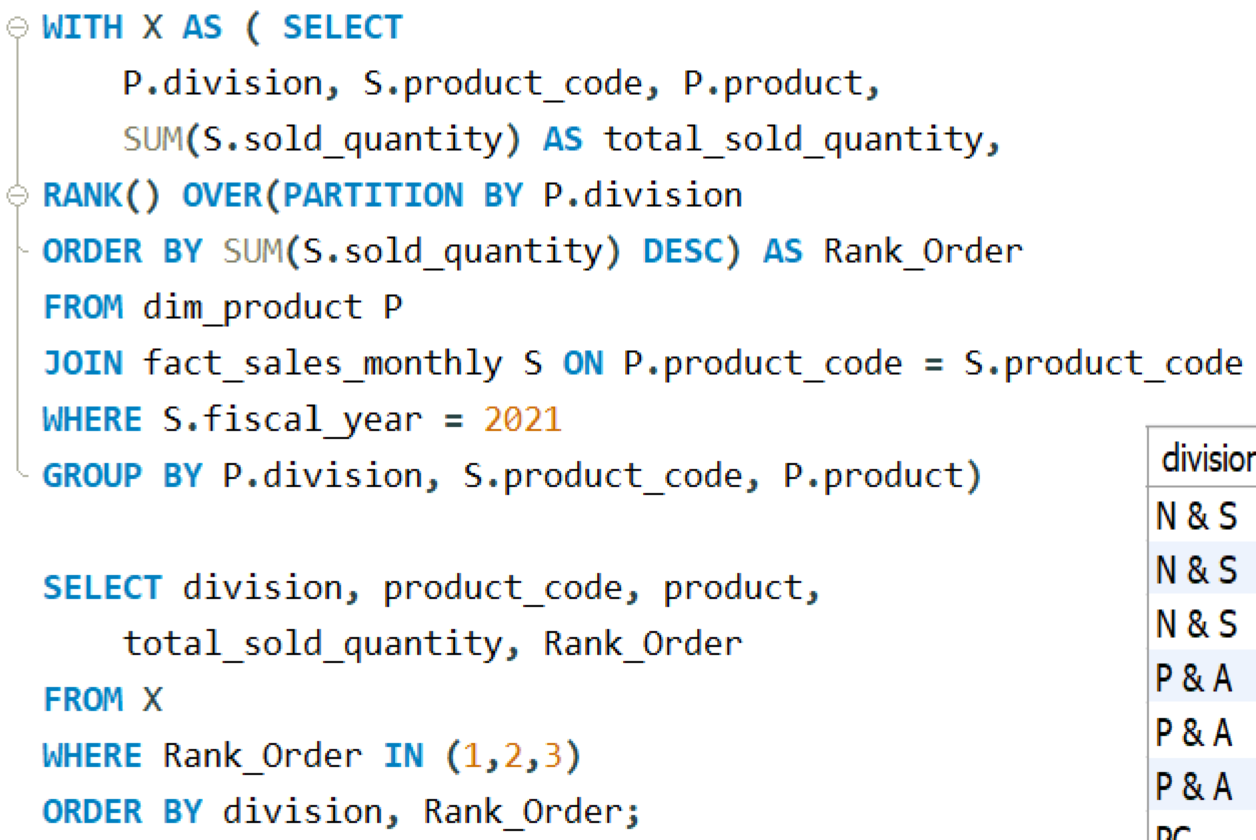
```
with x as(select c.channel,
round(sum(g.gross_price*s.sold_quantity)/100000,2)as gross_sales_mln
from fact_sales_monthly s
join dim_customer c using(customer_code)
join fact_gross_price g using(product_code)
group by c.channel)

select  channel, gross_sales_mln,
round((gross_sales_mln/(select sum(gross_sales_mln)from x))*100,2)
as pct from x
order by gross_sales_mln desc
```



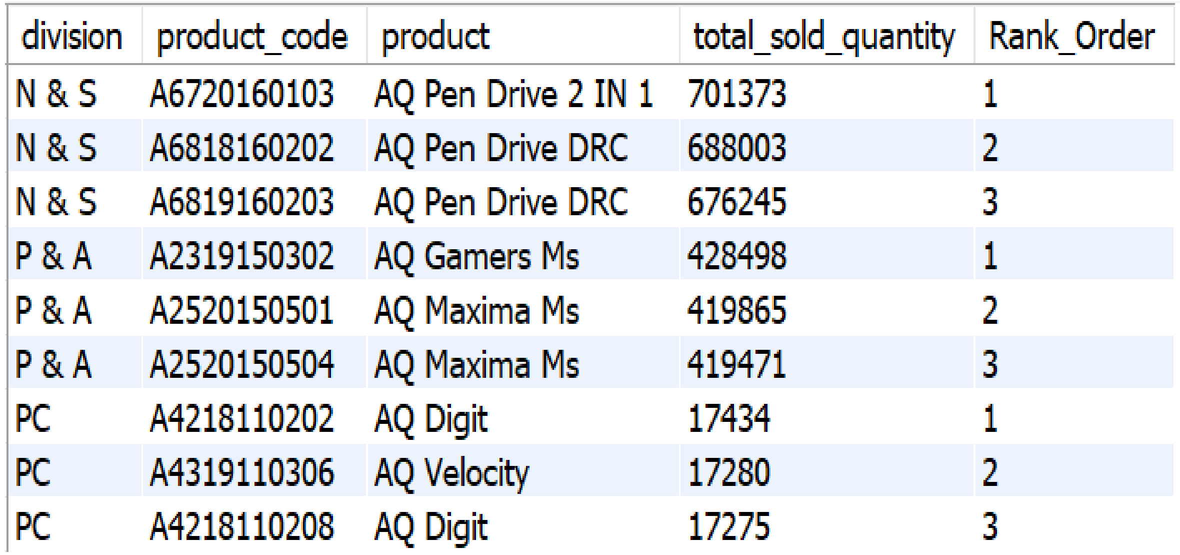
channel	gross_sales_mln	pct
Retailer	124419.42	72.70
Direct	26742.67	15.63
Distributor	19986.24	11.68

Get the Top 3 products in each division that have a high total_sold_quantity in the fiscal_year 2021? The final output contains these fields, division product_code.



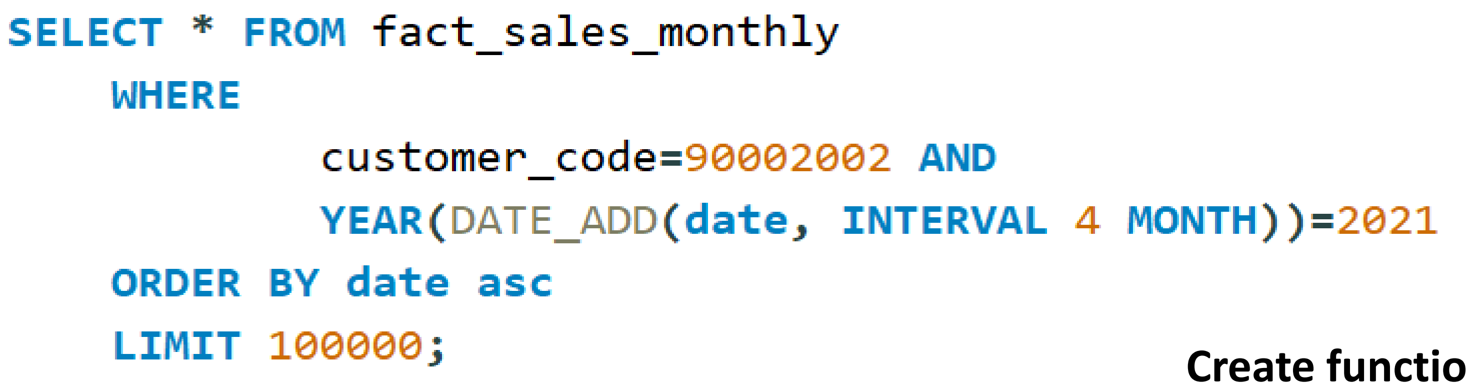
```
WITH X AS ( SELECT
    P.division, S.product_code, P.product,
    SUM(S.sold_quantity) AS total_sold_quantity,
RANK() OVER(PARTITION BY P.division
ORDER BY SUM(S.sold_quantity) DESC) AS Rank_Order
FROM dim_product P
JOIN fact_sales_monthly S ON P.product_code = S.product_code
WHERE S.fiscal_year = 2021
GROUP BY P.division, S.product_code, P.product)

SELECT division, product_code, product,
    total_sold_quantity, Rank_Order
FROM X
WHERE Rank_Order IN (1,2,3)
ORDER BY division, Rank_Order;
```

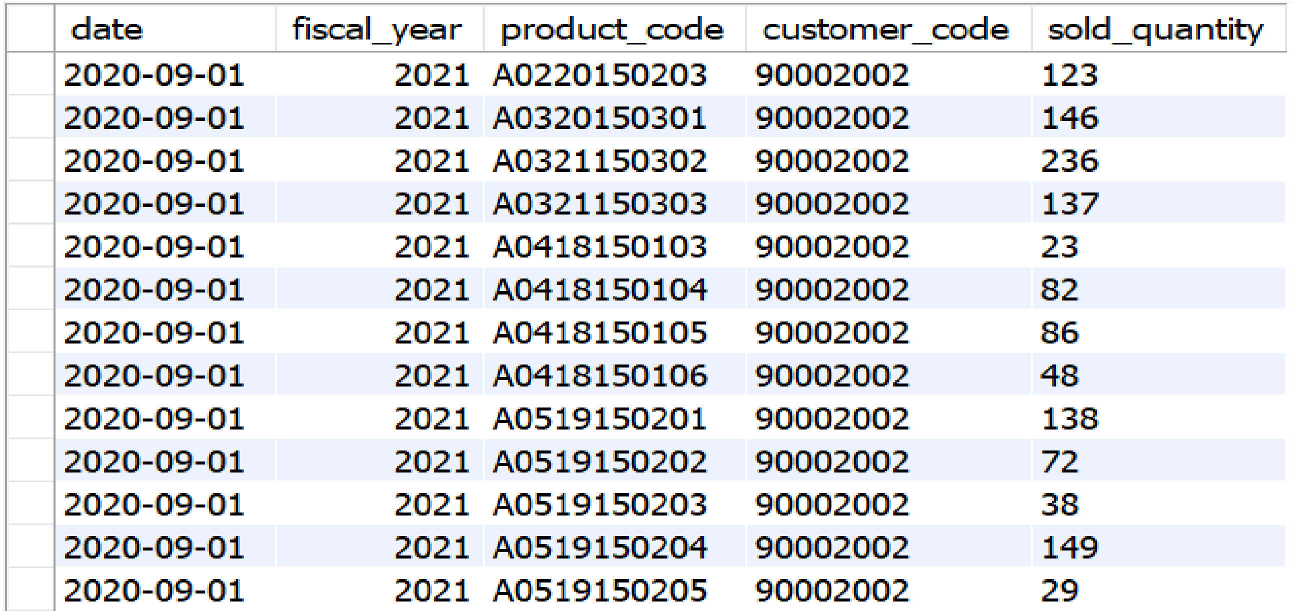


division	product_code	product	total_sold_quantity	Rank_Order
N & S	A6720160103	AQ Pen Drive 2 IN 1	701373	1
N & S	A6818160202	AQ Pen Drive DRC	688003	2
N & S	A6819160203	AQ Pen Drive DRC	676245	3
P & A	A2319150302	AQ Gamers Ms	428498	1
P & A	A2520150501	AQ Maxima Ms	419865	2
P & A	A2520150504	AQ Maxima Ms	419471	3
PC	A4218110202	AQ Digit	17434	1
PC	A4319110306	AQ Velocity	17280	2
PC	A4218110208	AQ Digit	17275	3

Get all the sales transaction data from fact_sales_monthly table for that customer(croma: 90002002) in the fiscal_year 2021

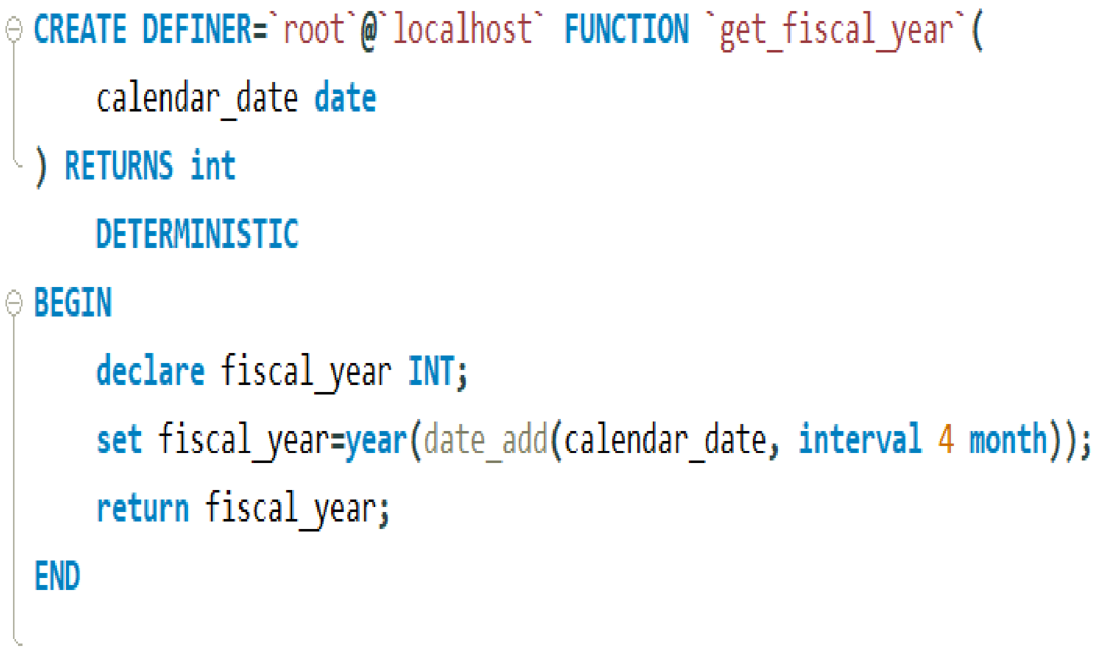


```
SELECT * FROM fact_sales_monthly
    WHERE
            customer_code=90002002 AND
            YEAR(DATE_ADD(date, INTERVAL 4 MONTH))=2021
    ORDER BY date asc
    LIMIT 100000;
```



date	fiscal_year	product_code	customer_code	sold_quantity
2020-09-01	2021	A0220150203	90002002	123
2020-09-01	2021	A0320150301	90002002	146
2020-09-01	2021	A0321150302	90002002	236
2020-09-01	2021	A0321150303	90002002	137
2020-09-01	2021	A0418150103	90002002	23
2020-09-01	2021	A0418150104	90002002	82
2020-09-01	2021	A0418150105	90002002	86
2020-09-01	2021	A0418150106	90002002	48
2020-09-01	2021	A0519150201	90002002	138
2020-09-01	2021	A0519150202	90002002	72
2020-09-01	2021	A0519150203	90002002	38
2020-09-01	2021	A0519150204	90002002	149
2020-09-01	2021	A0519150205	90002002	29

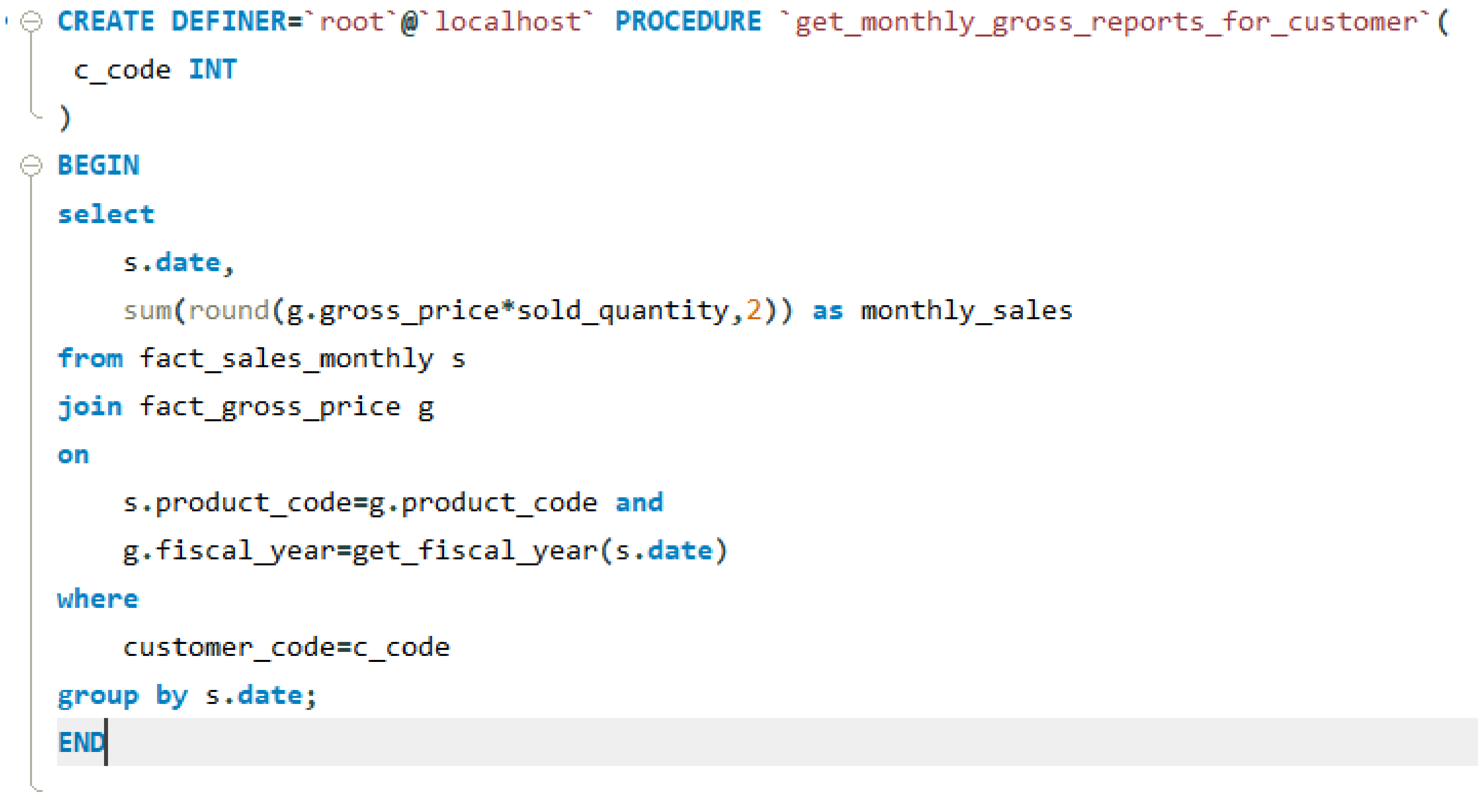
Create function for fiscal_year



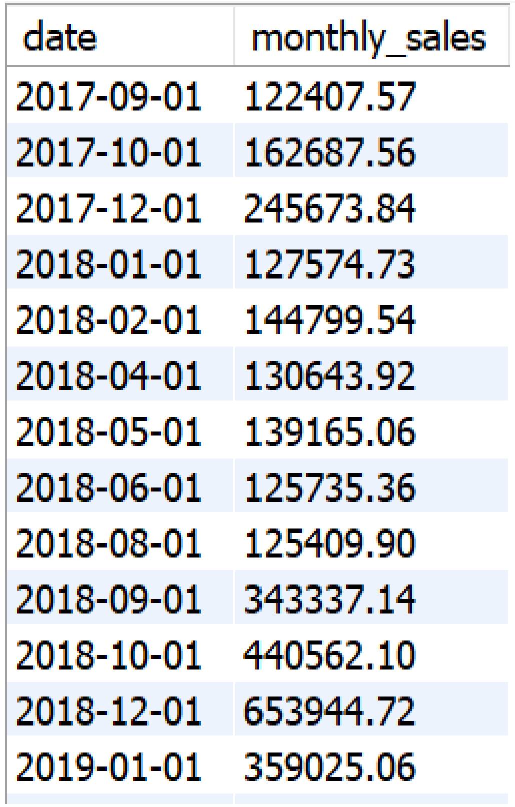
```
CREATE DEFINER=`root`@`localhost` FUNCTION `get_fiscal_year`(
    calendar_date date
) RETURNS int
    DETERMINISTIC
BEGIN
    declare fiscal_year INT;
    set fiscal_year=year(date_add(calendar_date, interval 4 month));
    return fiscal_year;
END
```

AtliQ Hardwares

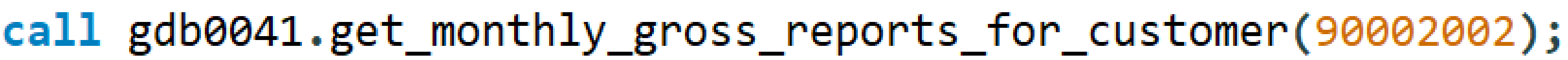
Generate monthly gross sales report for any customer using stored procedure



```
CREATE DEFINER=`root`@`localhost` PROCEDURE `get_monthly_gross_reports_for_customer`(
 c_code INT
)
BEGIN
select
    s.date,
    sum(round(g.gross_price*sold_quantity,2)) as monthly_sales
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;
END
```



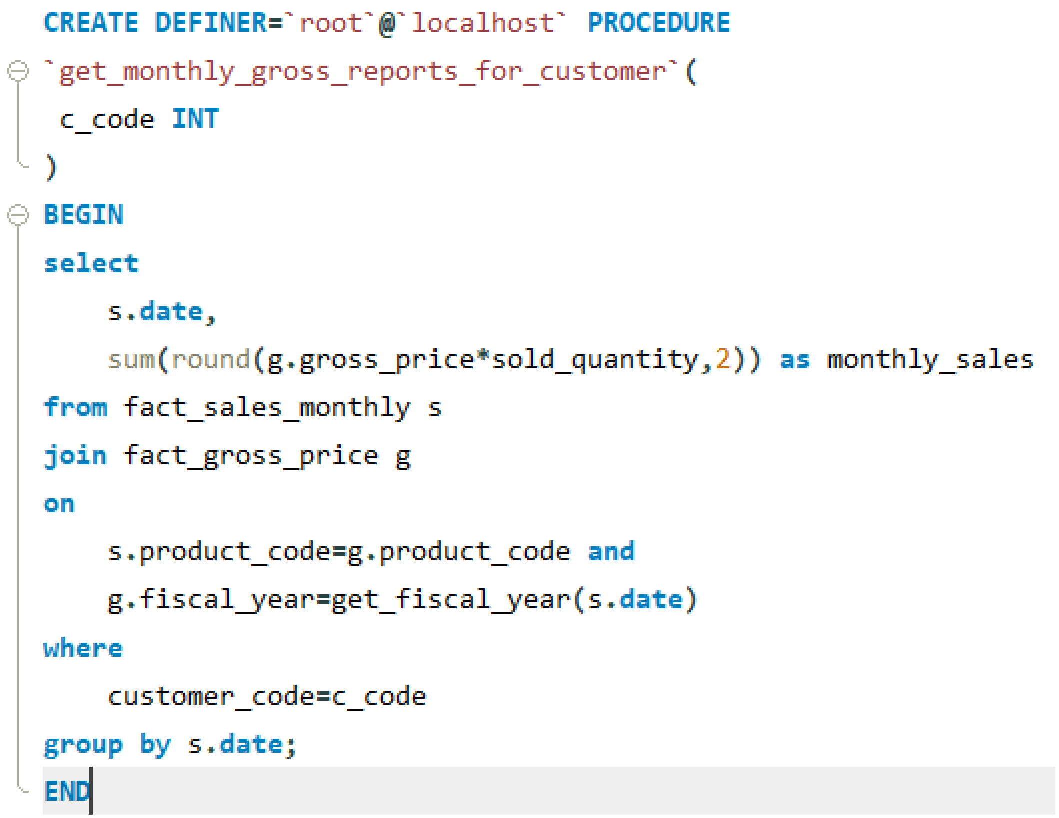
date	monthly_sales
2017-09-01	122407.57
2017-10-01	162687.56
2017-12-01	245673.84
2018-01-01	127574.73
2018-02-01	144799.54
2018-04-01	130643.92
2018-05-01	139165.06
2018-06-01	125735.36
2018-08-01	125409.90
2018-09-01	343337.14
2018-10-01	440562.10
2018-12-01	653944.72
2019-01-01	359025.06



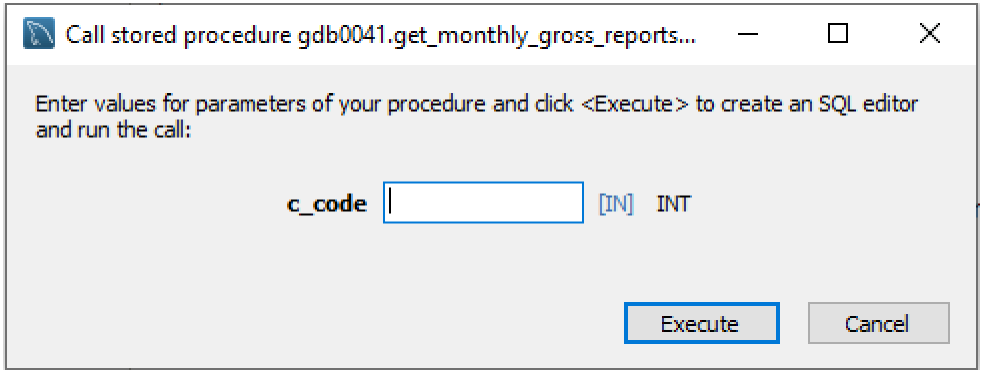
```
call gdb0041.get_monthly_gross_reports_for_customer(90002002);
```

AtliQ Hardwares

Generate monthly gross sales report for any customer using stored procedure



```
CREATE DEFINER=`root`@`localhost` PROCEDURE
`get_monthly_gross_reports_for_customer`(
 c_code INT
)
BEGIN
select
    s.date,
    sum(round(g.gross_price*sold_quantity,2)) as monthly_sales
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;
END
```



Call stored procedure gdb0041.get_monthly_gross_reports...

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

c_code [IN] INT

AtliQ Hardwares

Thank You

AtliQ Hardwares