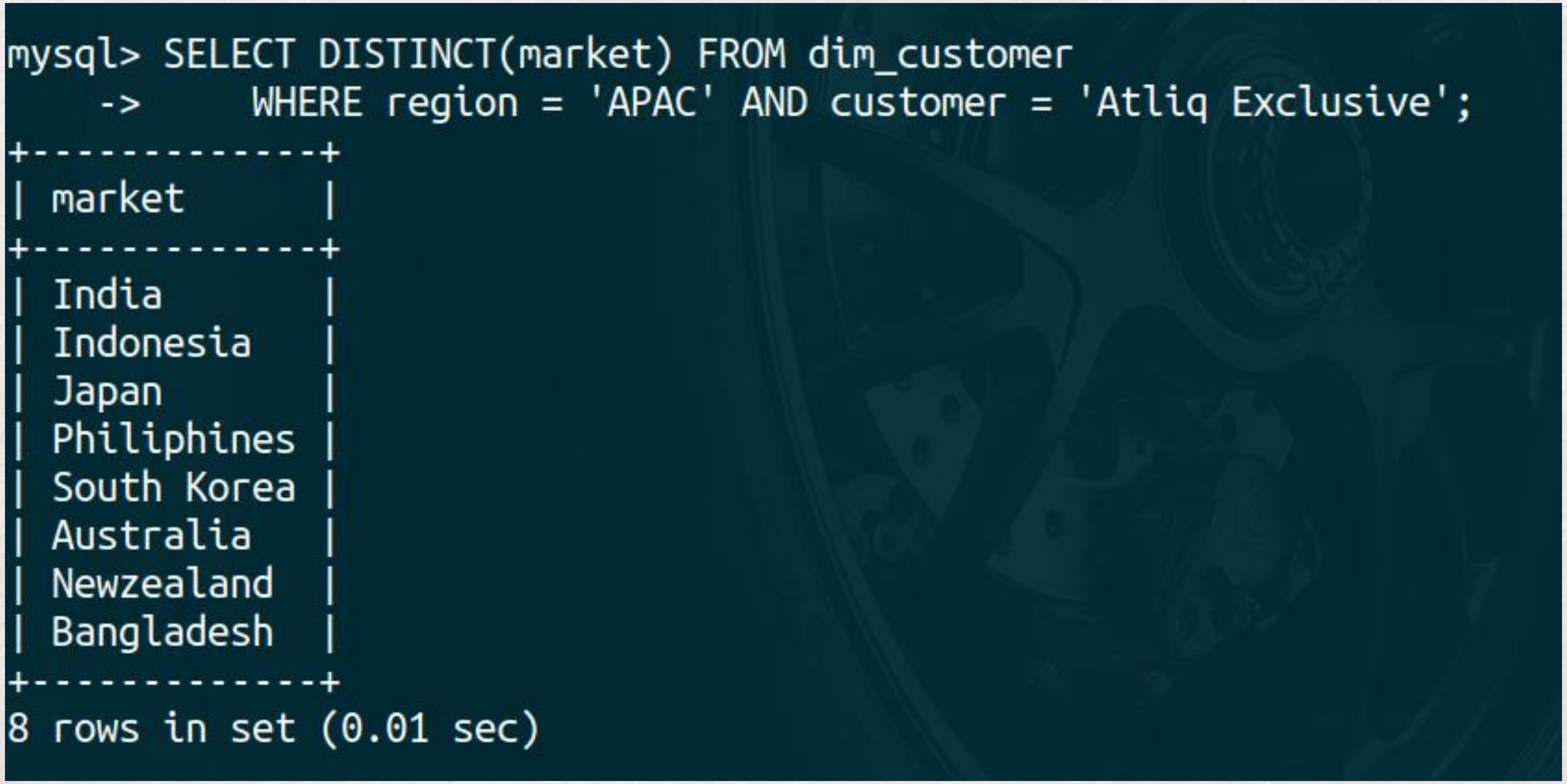
Question : 01

Provide the list of markets in which customer "**Atliq Exclusive**" operates its business in the **APAC** region.

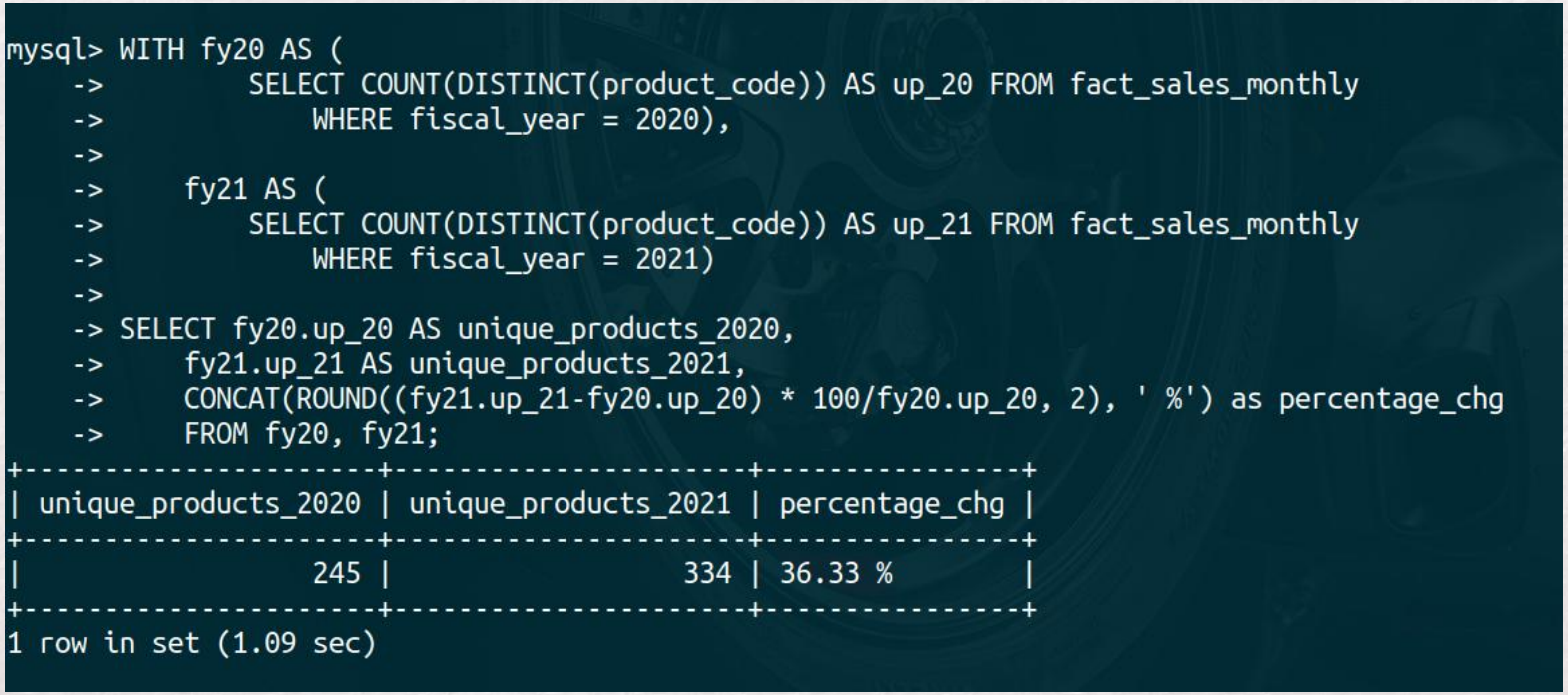


```
mysql> SELECT DISTINCT(market) FROM dim_customer
    ->     WHERE region = 'APAC' AND customer = 'Atliq Exclusive';
+-------------+
| market      |
+-------------+
| India       |
| Indonesia   |
| Japan       |
| Philiphines |
| South Korea |
| Australia   |
| Newzealand  |
| Bangladesh  |
+-------------+
8 rows in set (0.01 sec)
```

Question : 02

What is the percentage of unique product increase in 2021 vs. 2020? The final output contains these fields,

- unique_products_2020
- unique_products_2021
- percentage_chg

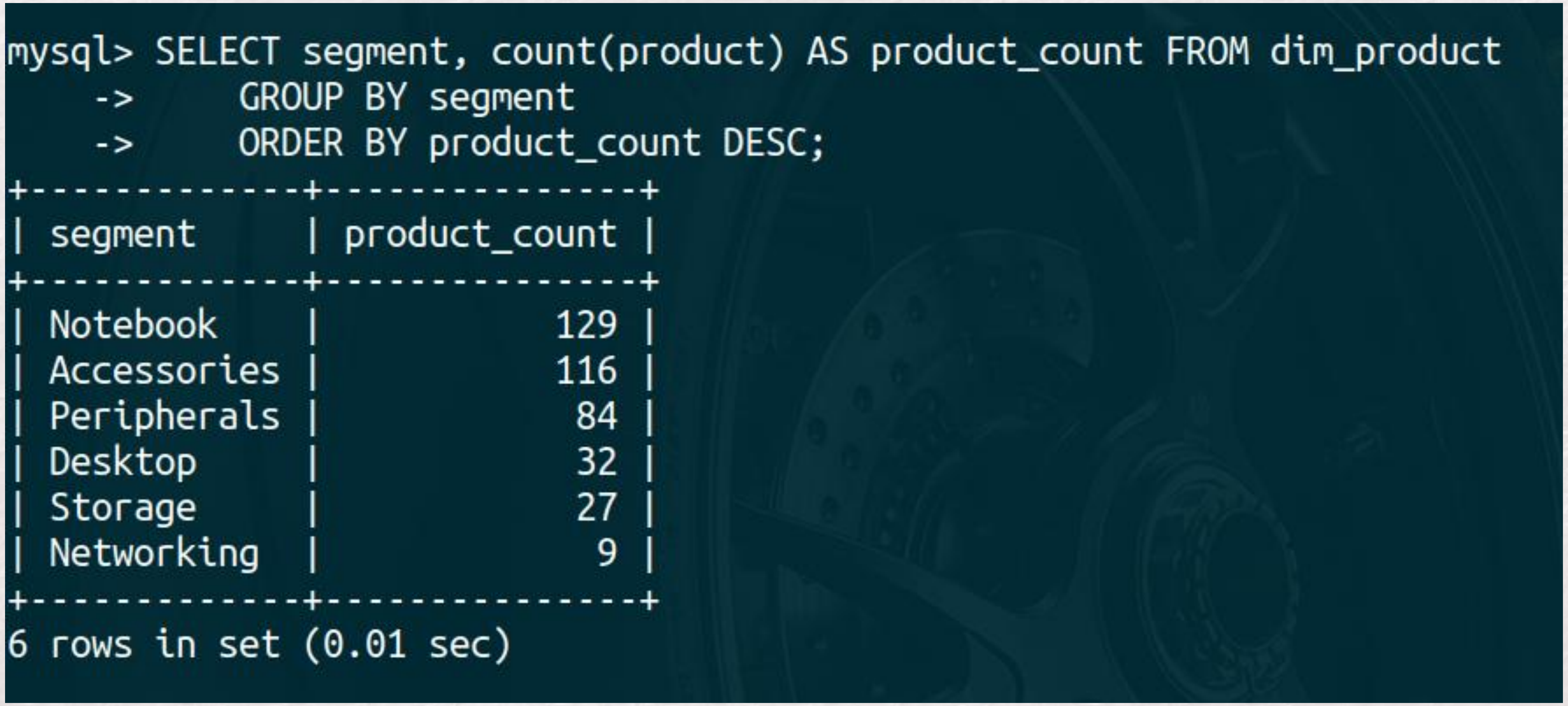


```
mysql> WITH fy20 AS (
    ->         SELECT COUNT(DISTINCT(product_code)) AS up_20 FROM fact_sales_monthly
    ->             WHERE fiscal_year = 2020),
    ->
    ->     fy21 AS (
    ->         SELECT COUNT(DISTINCT(product_code)) AS up_21 FROM fact_sales_monthly
    ->             WHERE fiscal_year = 2021)
    ->
    -> SELECT fy20.up_20 AS unique_products_2020,
    ->     fy21.up_21 AS unique_products_2021,
    ->     CONCAT(ROUND((fy21.up_21-fy20.up_20) * 100/fy20.up_20, 2), ' %') as percentage_chg
    ->     FROM fy20, fy21;
+----------------------+----------------------+----------------+
| unique_products_2020 | unique_products_2021 | percentage_chg |
+----------------------+----------------------+----------------+
|                  245 |                  334 | 36.33 %        |
+----------------------+----------------------+----------------+
1 row in set (1.09 sec)
```

Question : 03

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

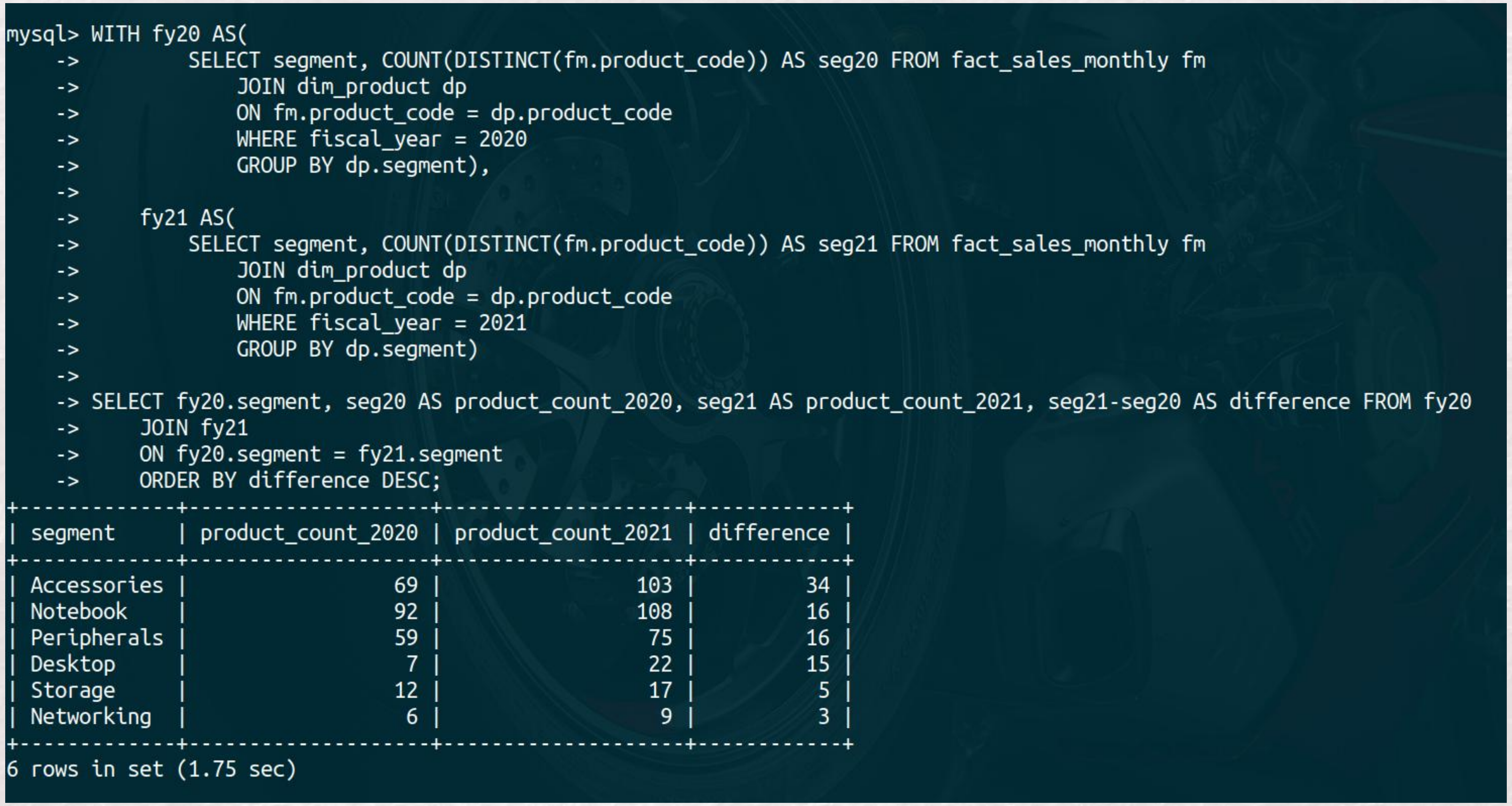


```
mysql> SELECT segment, count(product) AS product_count FROM dim_product
    ->     GROUP BY segment
    ->     ORDER BY product_count DESC;
+-------------+---------------+
| segment     | product_count |
+-------------+---------------+
| Notebook    |           129 |
| Accessories |           116 |
| Peripherals |            84 |
| Desktop     |            32 |
| Storage     |            27 |
| Networking  |             9 |
+-------------+---------------+
6 rows in set (0.01 sec)
```

Question : 04

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_2021
- difference

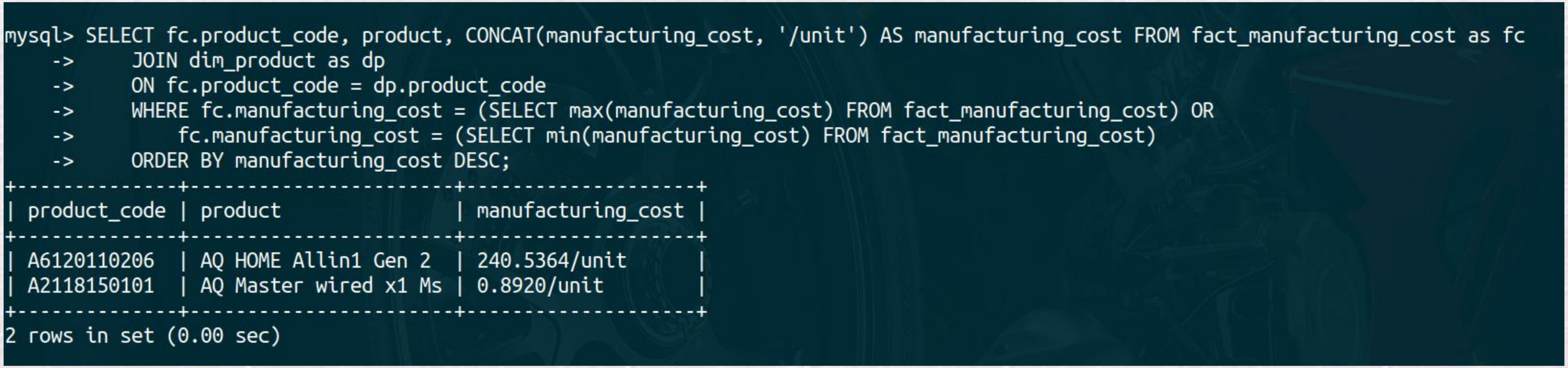


```
mysql> WITH fy20 AS(
    ->         SELECT segment, COUNT(DISTINCT(fm.product_code)) AS seg20 FROM fact_sales_monthly fm
    ->             JOIN dim_product dp
    ->             ON fm.product_code = dp.product_code
    ->             WHERE fiscal_year = 2020
    ->             GROUP BY dp.segment),
    ->
    ->     fy21 AS(
    ->         SELECT segment, COUNT(DISTINCT(fm.product_code)) AS seg21 FROM fact_sales_monthly fm
    ->             JOIN dim_product dp
    ->             ON fm.product_code = dp.product_code
    ->             WHERE fiscal_year = 2021
    ->             GROUP BY dp.segment)
    ->
    -> SELECT fy20.segment, seg20 AS product_count_2020, seg21 AS product_count_2021, seg21-seg20 AS difference FROM fy20
    ->     JOIN fy21
    ->     ON fy20.segment = fy21.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 |
+-------------+--------------------+--------------------+------------+
6 rows in set (1.75 sec)
```

Question : 05

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

- product_code
- product
- manufacturing_cost

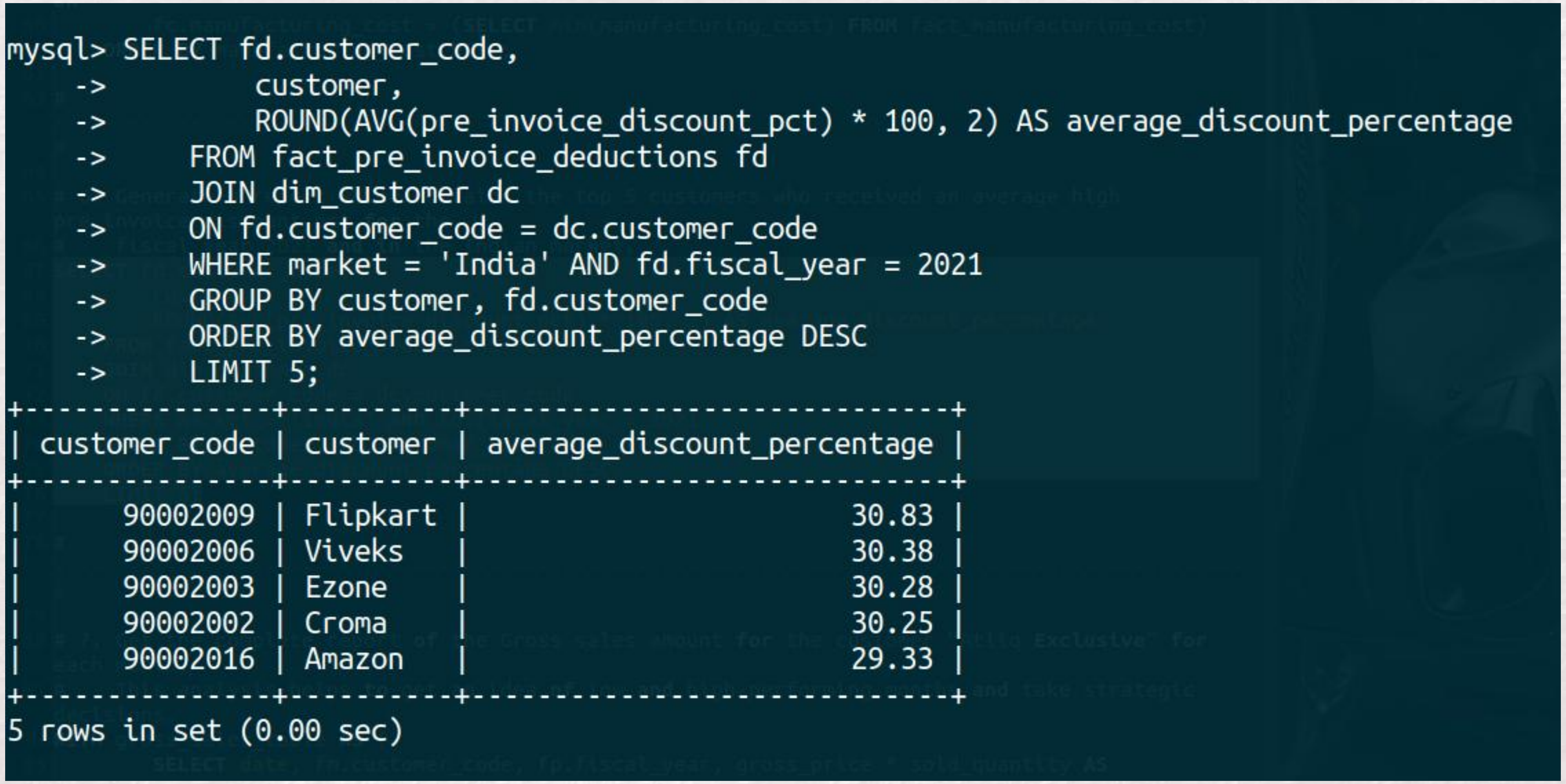


```
mysql> SELECT fc.product_code, product, CONCAT(manufacturing_cost, '/unit') AS manufacturing_cost FROM fact_manufacturing_cost as fc
    ->     JOIN dim_product as dp
    ->     ON fc.product_code = dp.product_code
    ->     WHERE fc.manufacturing_cost = (SELECT max(manufacturing_cost) FROM fact_manufacturing_cost) OR
    ->         fc.manufacturing_cost = (SELECT min(manufacturing_cost) FROM fact_manufacturing_cost)
    ->     ORDER BY manufacturing_cost DESC;
+--------------+-----------------------+--------------------+
| product_code | product               | manufacturing_cost |
+--------------+-----------------------+--------------------+
| A6120110206  | AQ HOME Allin1 Gen 2  | 240.5364/unit      |
| A2118150101  | AQ Master wired x1 Ms | 0.8920/unit        |
+--------------+-----------------------+--------------------+
2 rows in set (0.00 sec)
```

Question : 06

Generate a report which contains the top 5 customers who received an average high pre_invoice_discount_pct for the **fiscal year 2021** and in the **Indian** market. The final output contains these fields,

- customer_code
- customer
- average_discount_percentage

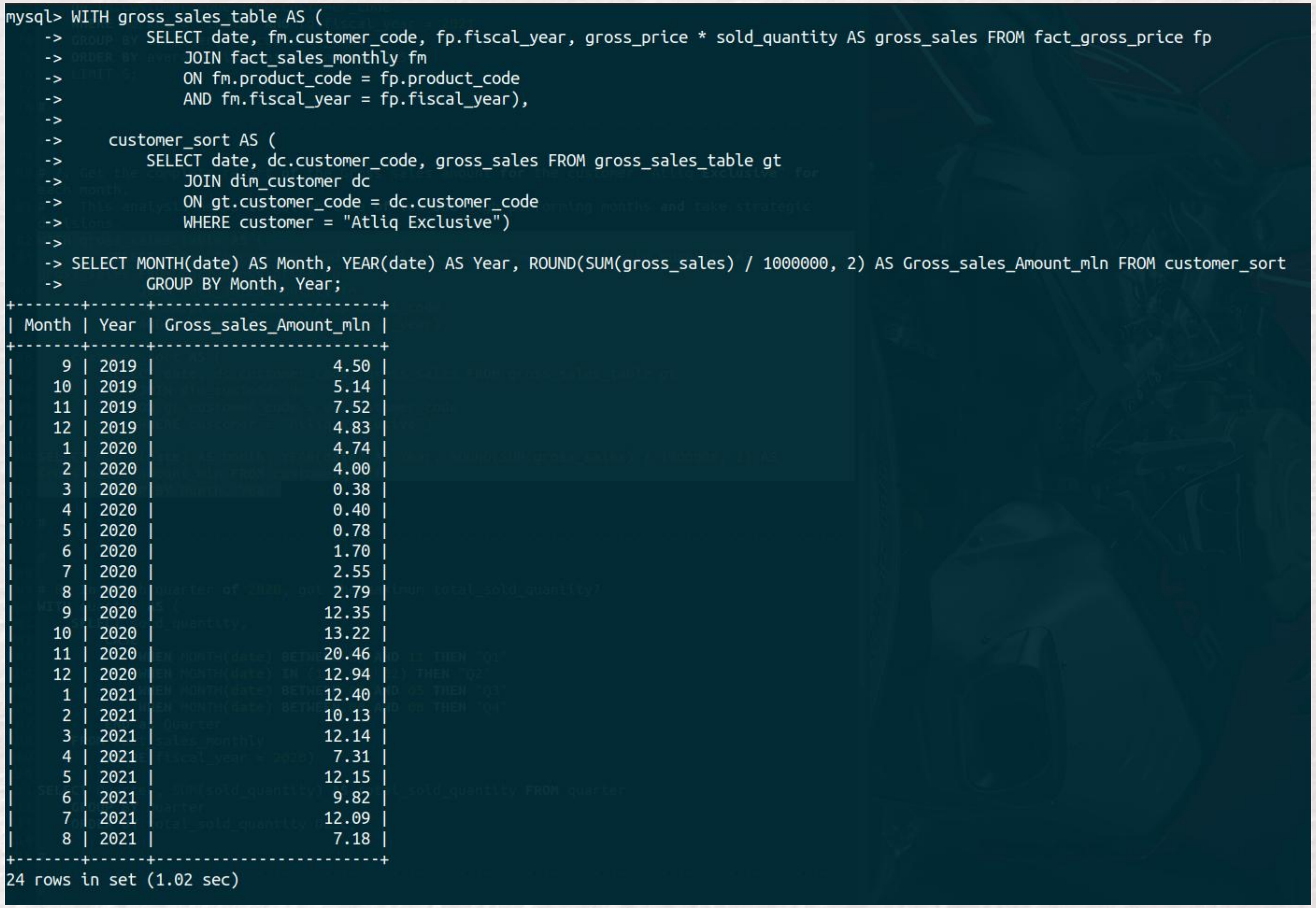


```
mysql> SELECT fd.customer_code,
    ->         customer,
    ->         ROUND(AVG(pre_invoice_discount_pct) * 100, 2) AS average_discount_percentage
    ->     FROM fact_pre_invoice_deductions fd
    ->     JOIN dim_customer dc
    ->     ON fd.customer_code = dc.customer_code
    ->     WHERE market = 'India' AND fd.fiscal_year = 2021
    ->     GROUP BY customer, fd.customer_code
    ->     ORDER BY average_discount_percentage DESC
    ->     LIMIT 5;
+---------------+----------+-----------------------------+
| customer_code | customer | average_discount_percentage |
+---------------+----------+-----------------------------+
|      90002009 | Flipkart |                       30.83 |
|      90002006 | Viveks   |                       30.38 |
|      90002003 | Ezone    |                       30.28 |
|      90002002 | Croma    |                       30.25 |
|      90002016 | Amazon   |                       29.33 |
+---------------+----------+-----------------------------+
5 rows in set (0.00 sec)
```

Question : 07

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

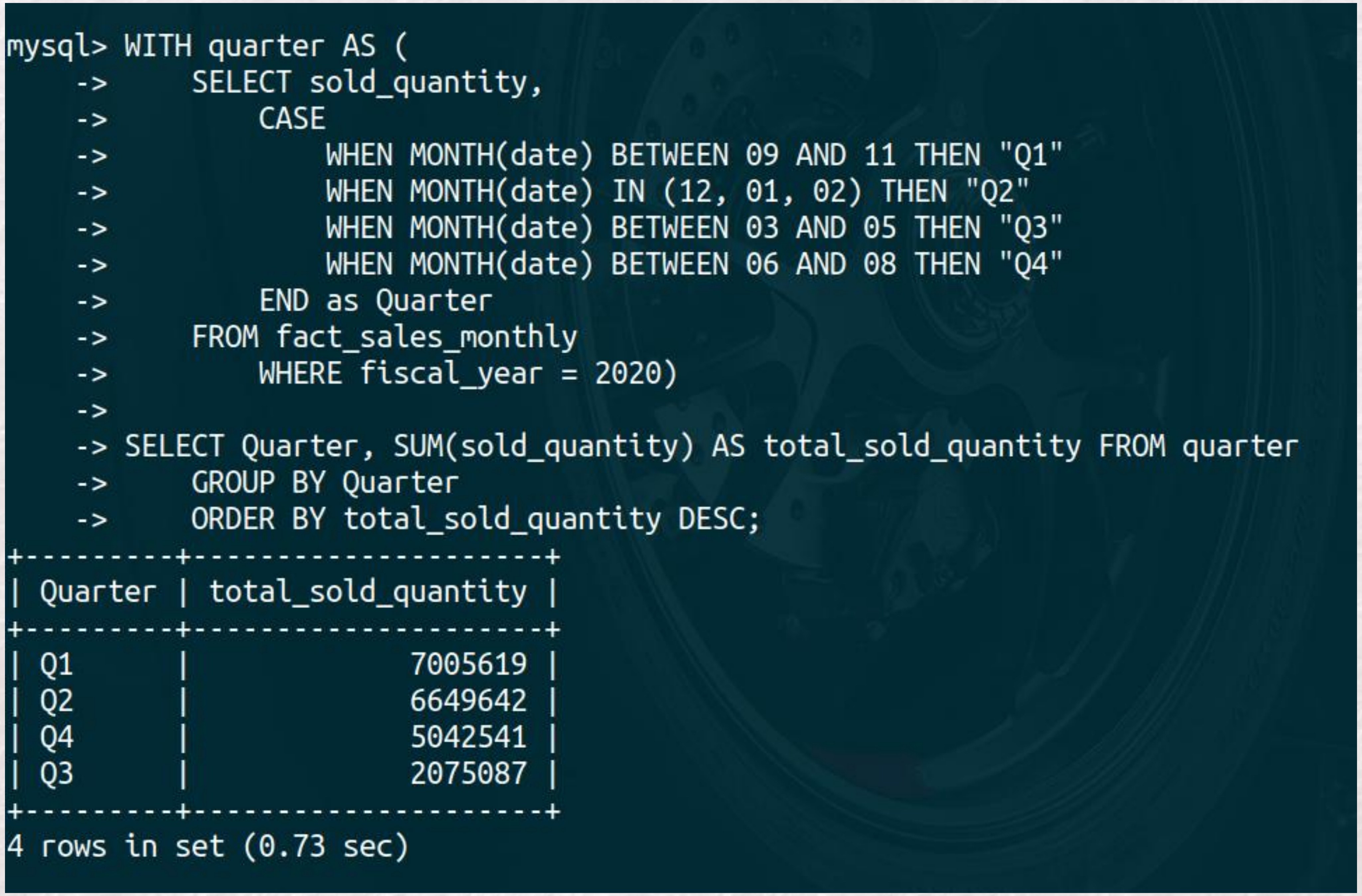


```
mysql> WITH gross_sales_table AS (
    ->         SELECT date, fm.customer_code, fp.fiscal_year, gross_price * sold_quantity AS gross_sales FROM fact_gross_price fp
    ->             JOIN fact_sales_monthly fm
    ->             ON fm.product_code = fp.product_code
    ->             AND fm.fiscal_year = fp.fiscal_year),
    ->
    ->     customer_sort AS (
    ->         SELECT date, dc.customer_code, gross_sales FROM gross_sales_table gt
    ->             JOIN dim_customer dc
    ->             ON gt.customer_code = dc.customer_code
    ->             WHERE customer = "Atliq Exclusive")
    ->
    -> SELECT MONTH(date) AS Month, YEAR(date) AS Year, ROUND(SUM(gross_sales) / 1000000, 2) AS Gross_sales_Amount_mln FROM customer_sort
    ->         GROUP BY Month, Year;
+-------+------+------------------------+
| Month | Year | Gross_sales_Amount_mln |
+-------+------+------------------------+
|     9 | 2019 |                   4.50 |
|    10 | 2019 |                   5.14 |
|    11 | 2019 |                   7.52 |
|    12 | 2019 |                   4.83 |
|     1 | 2020 |                   4.74 |
|     2 | 2020 |                   4.00 |
|     3 | 2020 |                   0.38 |
|     4 | 2020 |                   0.40 |
|     5 | 2020 |                   0.78 |
|     6 | 2020 |                   1.70 |
|     7 | 2020 |                   2.55 |
|     8 | 2020 |                   2.79 |
|     9 | 2020 |                  12.35 |
|    10 | 2020 |                  13.22 |
|    11 | 2020 |                  20.46 |
|    12 | 2020 |                  12.94 |
|     1 | 2021 |                  12.40 |
|     2 | 2021 |                  10.13 |
|     3 | 2021 |                  12.14 |
|     4 | 2021 |                   7.31 |
|     5 | 2021 |                  12.15 |
|     6 | 2021 |                   9.82 |
|     7 | 2021 |                  12.09 |
|     8 | 2021 |                   7.18 |
+-------+------+------------------------+
24 rows in set (1.02 sec)
```

Question : 08

In which quarter of 2020, got the maximum total_sold_quantity? The final output contains these fields sorted by the total_sold_quantity,

- Quarter
- total_sold_quantity

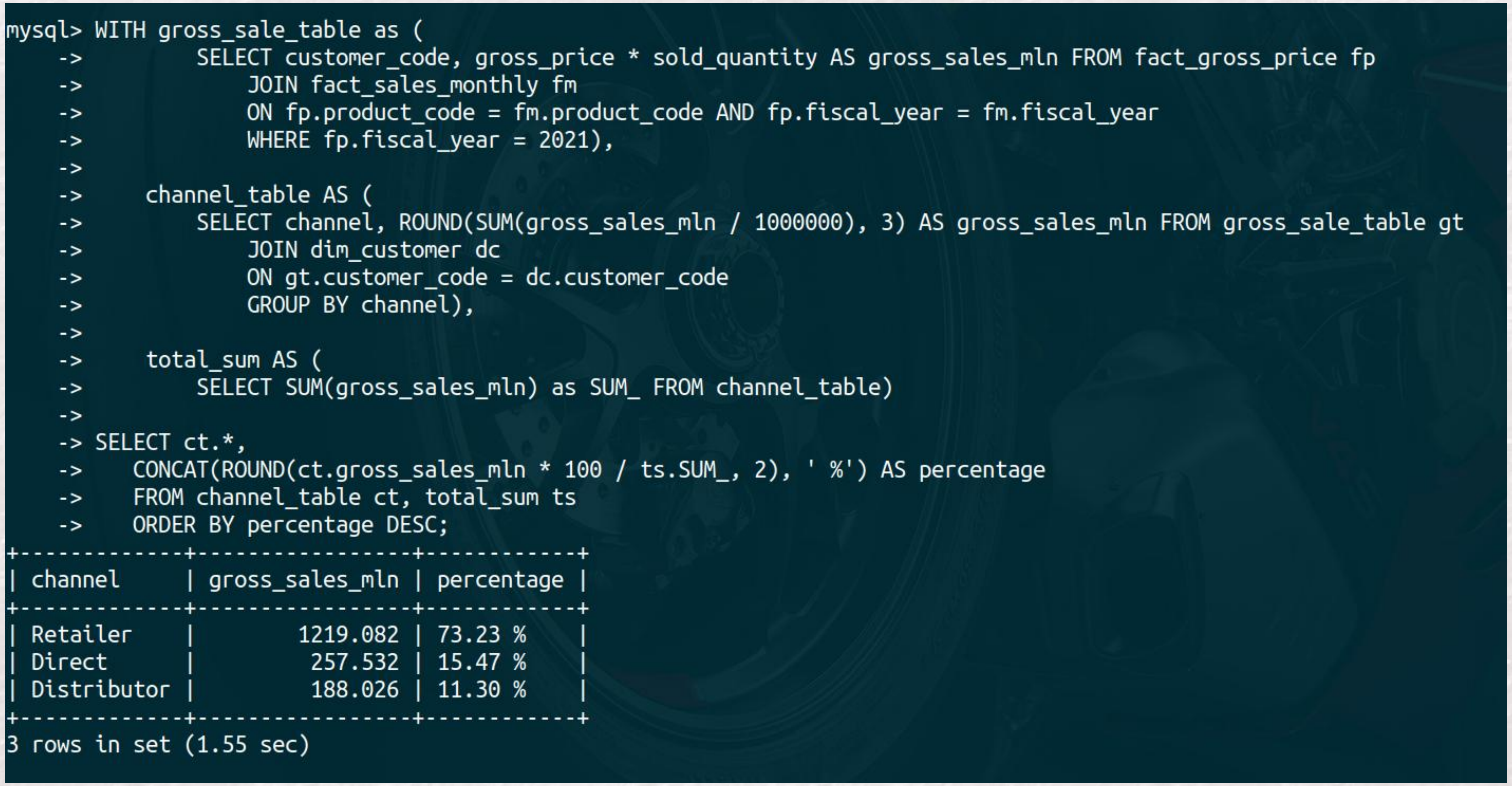


```
mysql> WITH quarter AS (
    ->     SELECT sold_quantity,
    ->         CASE
    ->             WHEN MONTH(date) BETWEEN 09 AND 11 THEN "Q1"
    ->             WHEN MONTH(date) IN (12, 01, 02) THEN "Q2"
    ->             WHEN MONTH(date) BETWEEN 03 AND 05 THEN "Q3"
    ->             WHEN MONTH(date) BETWEEN 06 AND 08 THEN "Q4"
    ->         END as Quarter
    ->     FROM fact_sales_monthly
    ->         WHERE fiscal_year = 2020)
    ->
    -> SELECT Quarter, SUM(sold_quantity) AS total_sold_quantity FROM quarter
    ->     GROUP BY Quarter
    ->     ORDER BY total_sold_quantity DESC;
+---------+---------------------+
| Quarter | total_sold_quantity |
+---------+---------------------+
| Q1      |             7005619 |
| Q2      |             6649642 |
| Q4      |             5042541 |
| Q3      |             2075087 |
+---------+---------------------+
4 rows in set (0.73 sec)
```

Question : 09

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_mln
- percentage

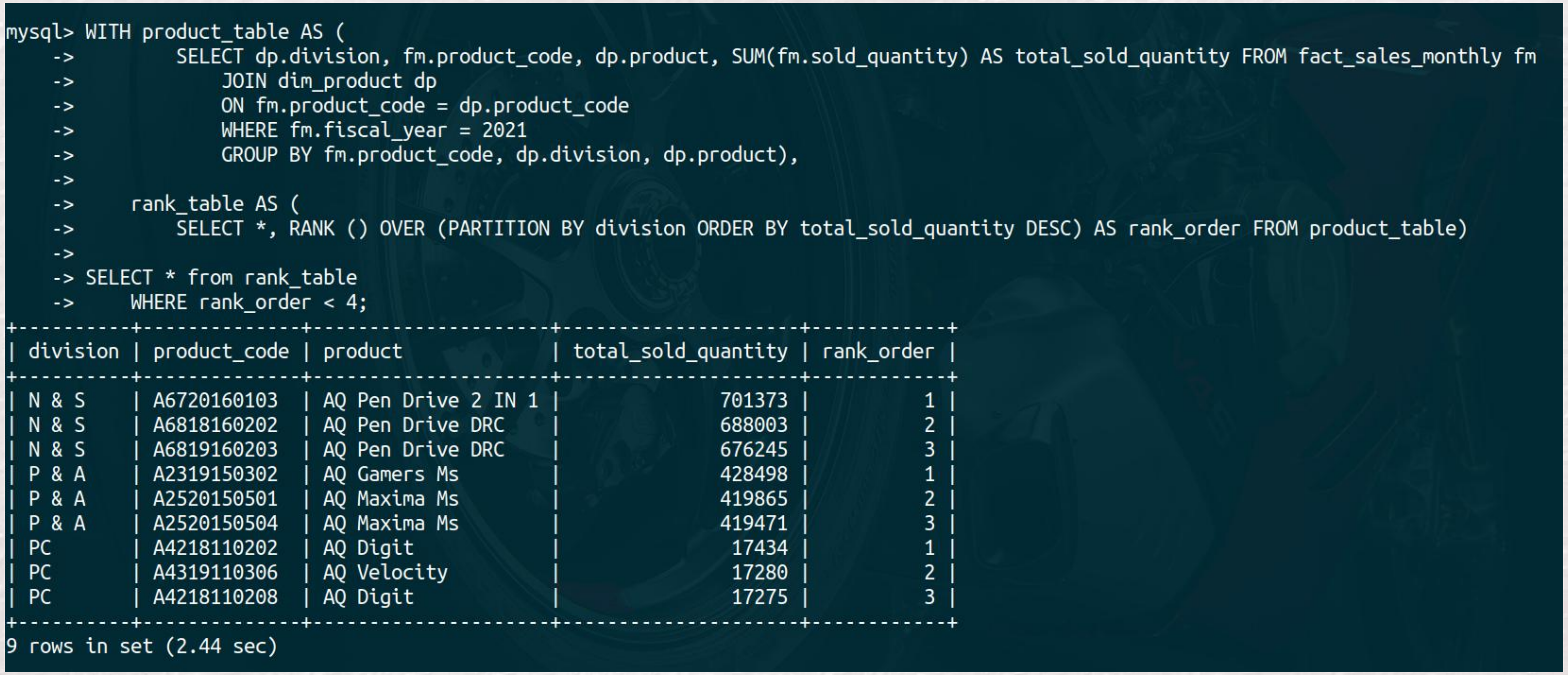


```
mysql> WITH gross_sale_table as (
    ->         SELECT customer_code, gross_price * sold_quantity AS gross_sales_mln FROM fact_gross_price fp
    ->             JOIN fact_sales_monthly fm
    ->             ON fp.product_code = fm.product_code AND fp.fiscal_year = fm.fiscal_year
    ->             WHERE fp.fiscal_year = 2021),
    ->
    ->     channel_table AS (
    ->         SELECT channel, ROUND(SUM(gross_sales_mln / 1000000), 3) AS gross_sales_mln FROM gross_sale_table gt
    ->             JOIN dim_customer dc
    ->             ON gt.customer_code = dc.customer_code
    ->             GROUP BY channel),
    ->
    ->     total_sum AS (
    ->         SELECT SUM(gross_sales_mln) as SUM_ FROM channel_table)
    ->
    -> SELECT ct.*,
    ->    CONCAT(ROUND(ct.gross_sales_mln * 100 / ts.SUM_, 2), ' %') AS percentage
    ->    FROM channel_table ct, total_sum ts
    ->    ORDER BY percentage DESC;
+-------------+-----------------+------------+
| channel     | gross_sales_mln | percentage |
+-------------+-----------------+------------+
| Retailer    |        1219.082 | 73.23 %    |
| Direct      |         257.532 | 15.47 %    |
| Distributor |         188.026 | 11.30 %    |
+-------------+-----------------+------------+
3 rows in set (1.55 sec)
```

Question : 10

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
- product
- total_sold_quantity
- rank_order



```
mysql> WITH product_table AS (
    ->         SELECT dp.division, fm.product_code, dp.product, SUM(fm.sold_quantity) AS total_sold_quantity FROM fact_sales_monthly fm
    ->             JOIN dim_product dp
    ->             ON fm.product_code = dp.product_code
    ->             WHERE fm.fiscal_year = 2021
    ->             GROUP BY fm.product_code, dp.division, dp.product),
    ->
    ->     rank_table AS (
    ->         SELECT *, RANK () OVER (PARTITION BY division ORDER BY total_sold_quantity DESC) AS rank_order FROM product_table)
    ->
    -> SELECT * from rank_table
    ->     WHERE rank_order < 4;
+----------+--------------+---------------------+---------------------+------------+
| 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 |
+----------+--------------+---------------------+---------------------+------------+
9 rows in set (2.44 sec)
```