CODE BASICS

SQL

BUSINESS INSIGHTS WITH ADVANCED SQL

CODEBASICS DATA ANALYST BOOTCAMP PROJECT

By Mohammed Raza Govani

Mohammed Raza Govani

INTRODUCTION & BACKGROUND

AtliQ Hardware (Company)

- A global computer hardware company with presence across retail, e-commerce, and direct sales. Like most fast-growing businesses, AtliQ faces one big challenge: turning **huge amounts of raw data** into clear, actionable insights.

The Project

- As part of my Codebasics Data Analyst Bootcamp, I worked on solving **10 real business requests** using advanced SQL. Instead of just writing queries, the goal was to uncover insights that matter to decision-makers.

What Makes It Interesting?

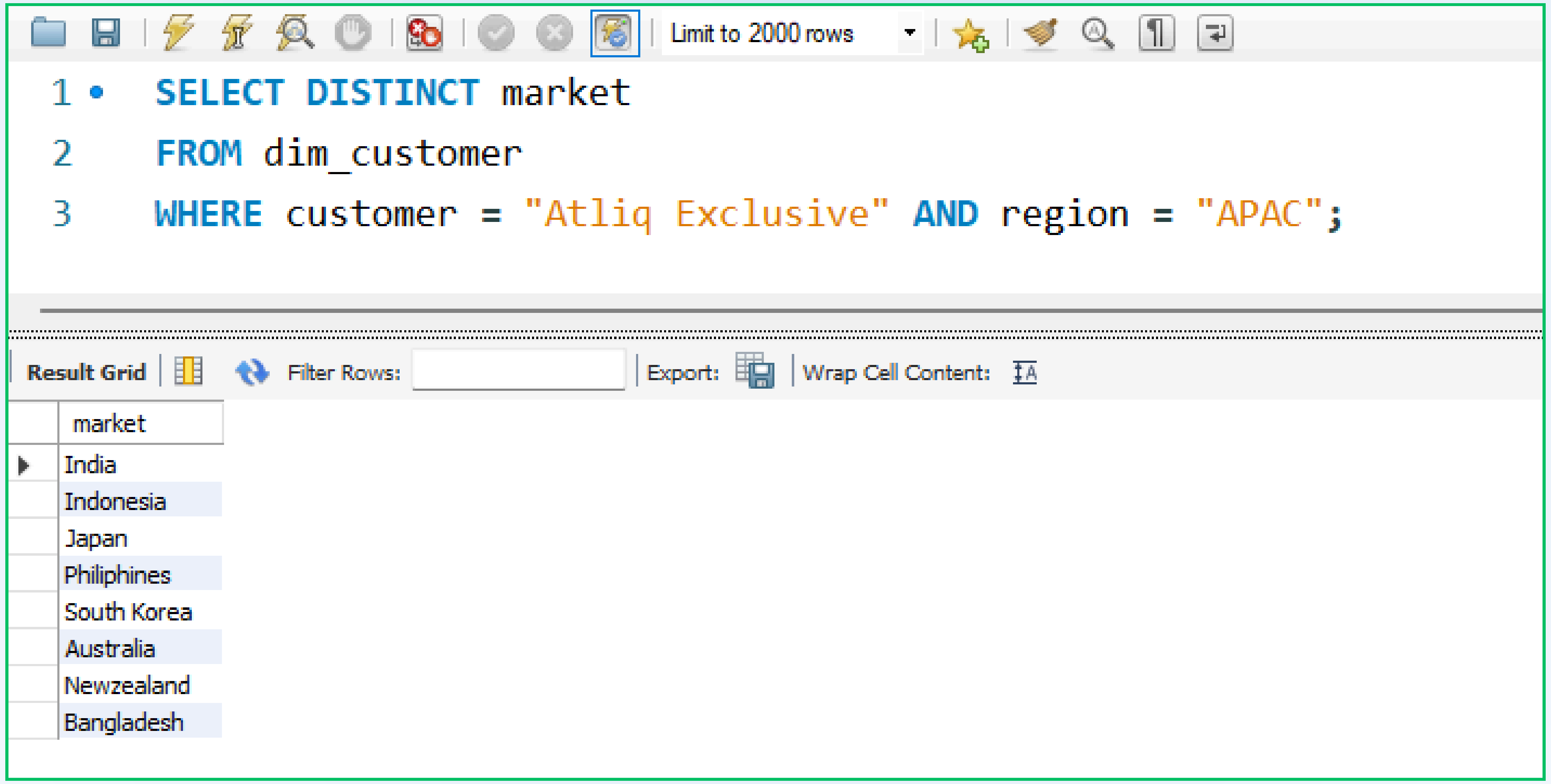
- The dataset is no toy example — it's **1.5M+ rows of sales, customers, and financial data** spread across multiple tables. That means dealing with the kind of messy, large-scale data challenges real analysts face every day.

Toolkit

- Basic & Advanced SQL (Joins, CTEs, Window Functions, Aggregations) in MYSQL

Request 1

- **Business Question:** Markets where customer “AtliQ Exclusive” operates in APAC
- **SQL Focus:** DISTINCT + Filtering



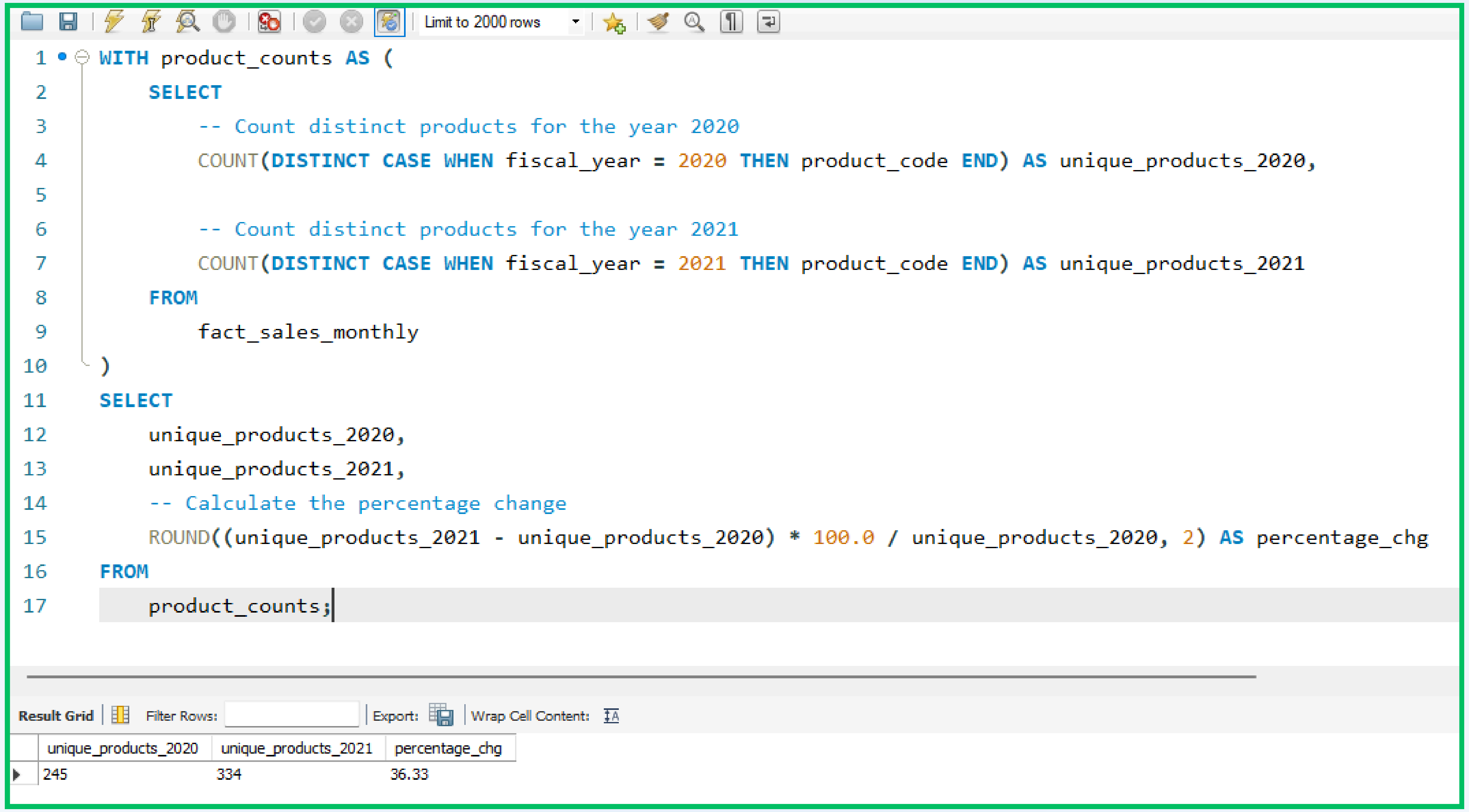
```sql
SELECT DISTINCT market
FROM dim_customer
WHERE customer = "Atliq Exclusive" AND region = "APAC";
```

market
India
Indonesia
Japan
Philiphines
South Korea
Australia
Newzealand
Bangladesh

Mohammed Raza Govani

Request 2

- **Business Question:** % increase in unique products in 2021 vs 2020
- **SQL Focus:** CTE, Aggregation Functions

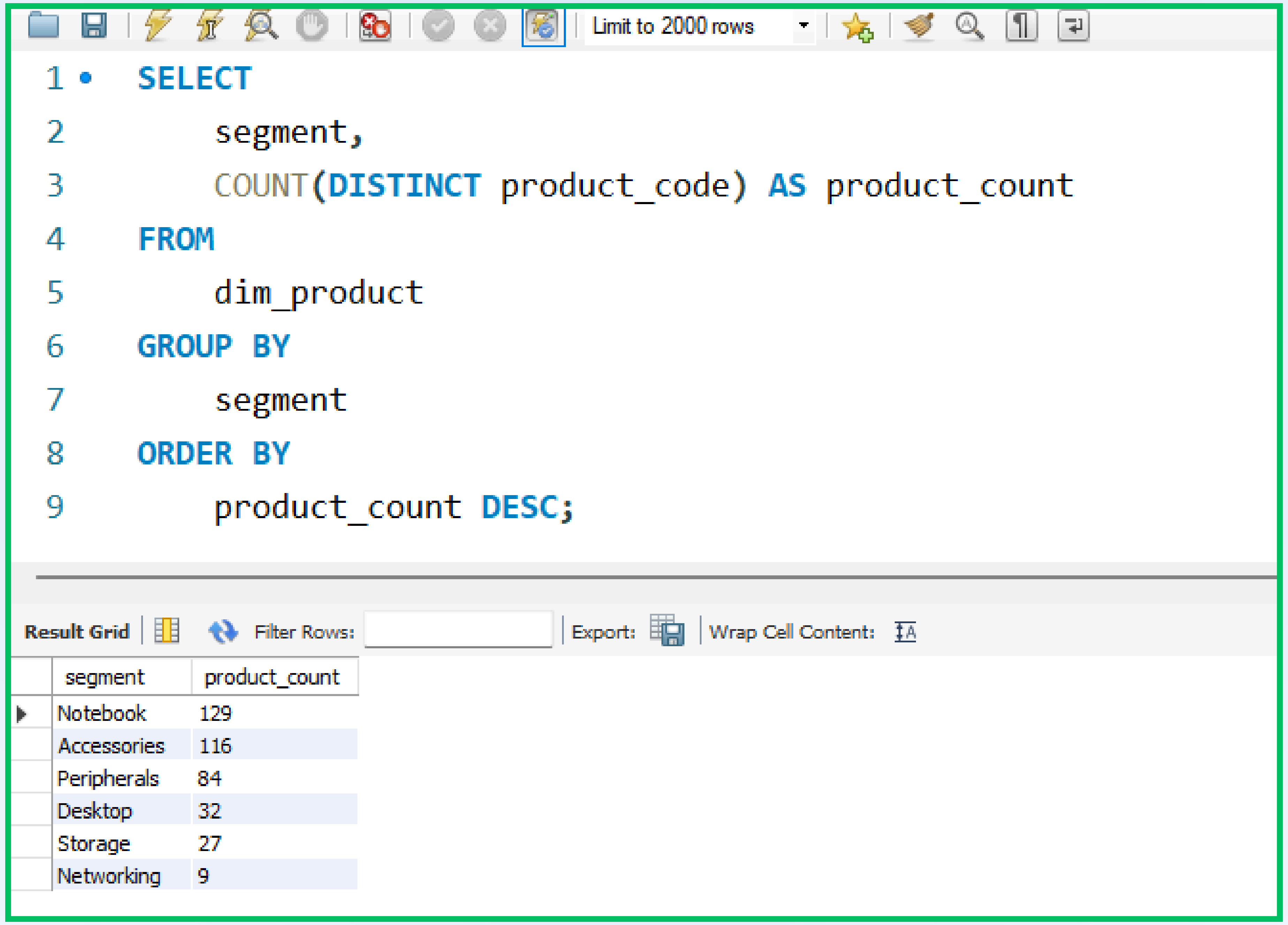


```sql
WITH product_counts AS (
    SELECT
        -- Count distinct products for the year 2020
        COUNT(DISTINCT CASE WHEN fiscal_year = 2020 THEN product_code END) AS unique_products_2020,

        -- Count distinct products for the year 2021
        COUNT(DISTINCT CASE WHEN fiscal_year = 2021 THEN product_code END) AS unique_products_2021
    FROM
        fact_sales_monthly
)
SELECT
    unique_products_2020,
    unique_products_2021,
    -- Calculate the percentage change
    ROUND((unique_products_2021 - unique_products_2020) * 100.0 / unique_products_2020, 2) AS percentage_chg
FROM
    product_counts;
```

unique_products_2020	unique_products_2021	percentage_chg
245	334	36.33

Request 3

- **Business Question:** Unique product counts by segment (descending order)
- **SQL Focus:** Aggregation + GROUP BY + ORDER BY



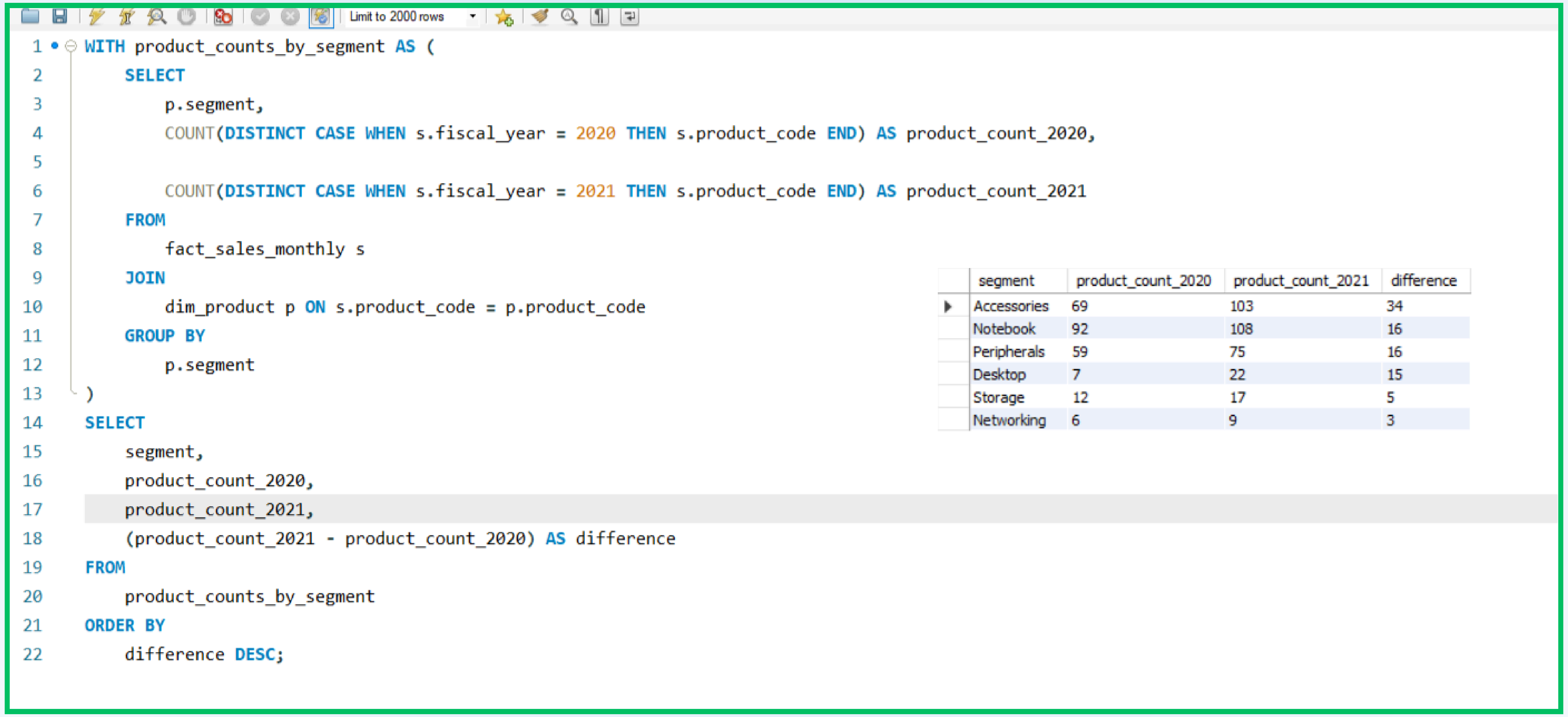
```sql
SELECT
    segment,
    COUNT(DISTINCT product_code) AS product_count
FROM
    dim_product
GROUP BY
    segment
ORDER BY
    product_count DESC;
```

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

Mohammed Raza Govani

Request 4

- **Business Question:** Segment with highest growth in unique products from 2020 to 2021
- **SQL Focus:** CTE + JOINS + GROUP BY



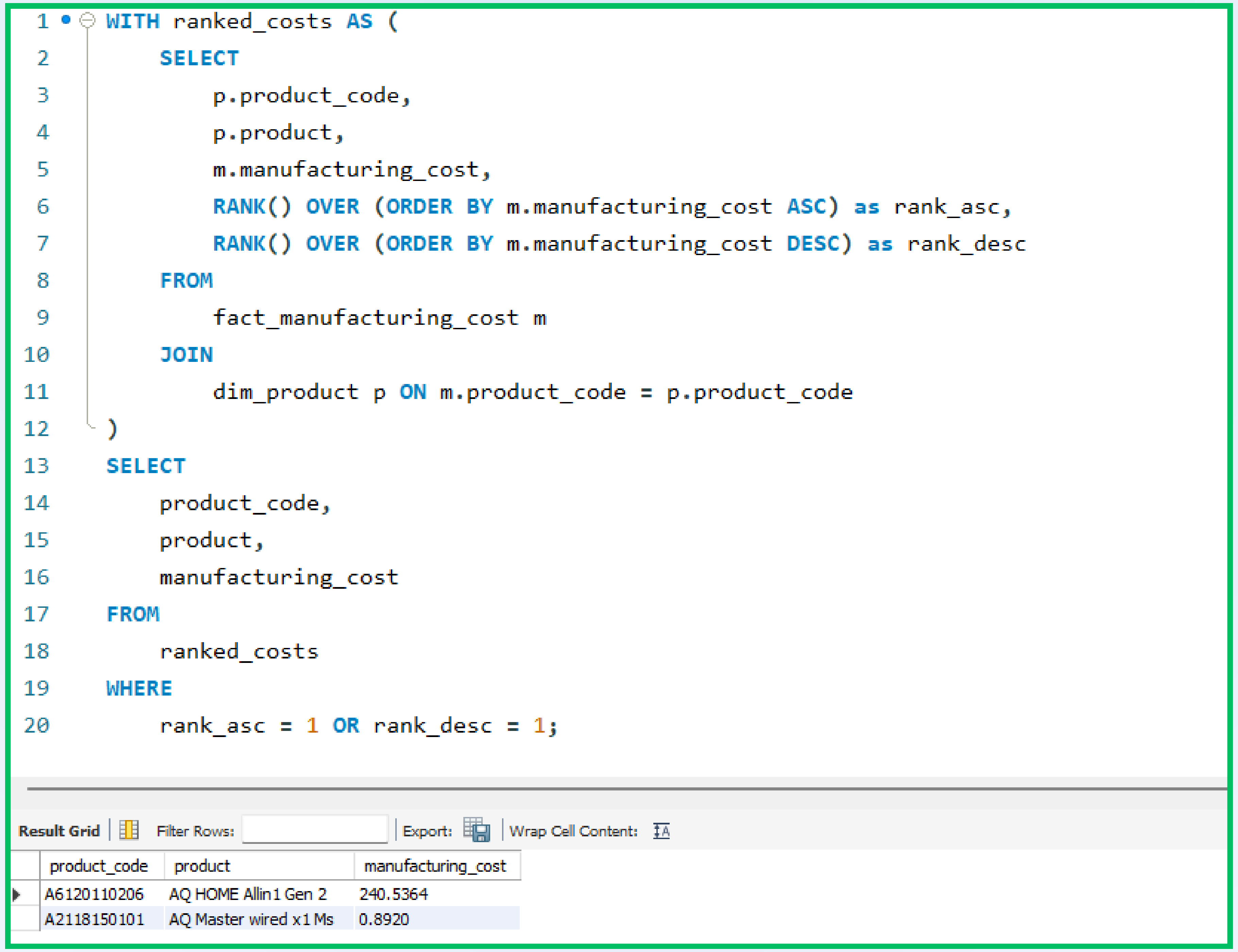
```sql
WITH product_counts_by_segment AS (
    SELECT
        p.segment,
        COUNT(DISTINCT CASE WHEN s.fiscal_year = 2020 THEN s.product_code END) AS product_count_2020,

        COUNT(DISTINCT CASE WHEN s.fiscal_year = 2021 THEN s.product_code END) AS product_count_2021
    FROM
        fact_sales_monthly s
    JOIN
        dim_product p ON s.product_code = p.product_code
    GROUP BY
        p.segment
)
SELECT
    segment,
    product_count_2020,
    product_count_2021,
    (product_count_2021 - product_count_2020) AS difference
FROM
    product_counts_by_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

Mohammed Raza Govani

Request 5

- **Business Question:** Products with highest & lowest manufacturing cost
- **SQL Focus:** WINDOW Functions + CTE + JOINS + Filtering



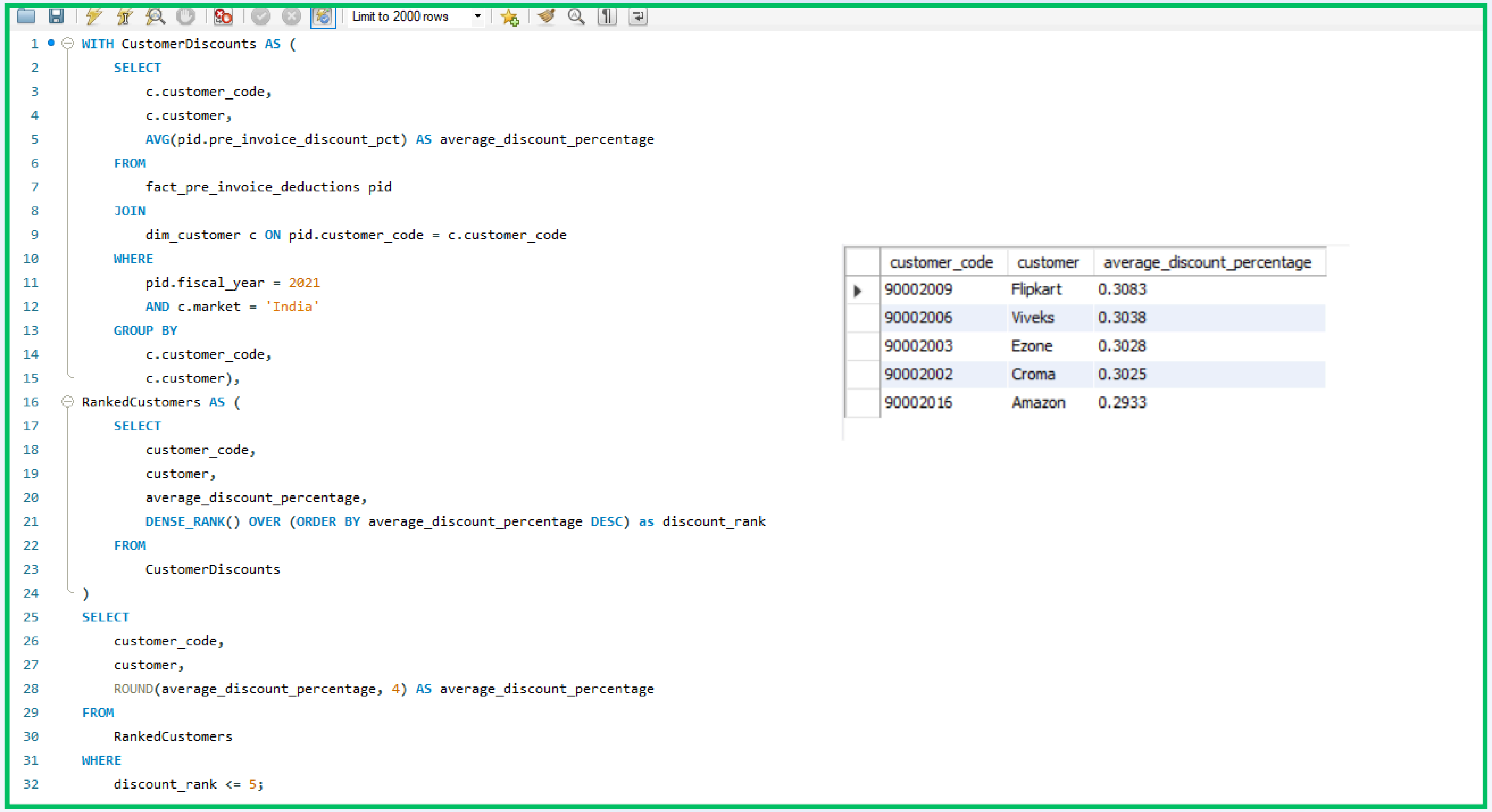
```sql
WITH ranked_costs AS (
    SELECT
        p.product_code,
        p.product,
        m.manufacturing_cost,
        RANK() OVER (ORDER BY m.manufacturing_cost ASC) as rank_asc,
        RANK() OVER (ORDER BY m.manufacturing_cost DESC) as rank_desc
    FROM
        fact_manufacturing_cost m
    JOIN
        dim_product p ON m.product_code = p.product_code
)
SELECT
    product_code,
    product,
    manufacturing_cost
FROM
    ranked_costs
WHERE
    rank_asc = 1 OR rank_desc = 1;
```

product_code	product	manufacturing_cost
A6120110206	AQ HOME Allin1 Gen 2	240.5364
A2118150101	AQ Master wired x1 Ms	0.8920

Mohammed Raza Govani

Request 6

- **Business Question:** Top 5 customers with highest average pre-invoice discount in FY21 (India)
- **SQL Focus:** Multiple CTE's + Window + JOINs



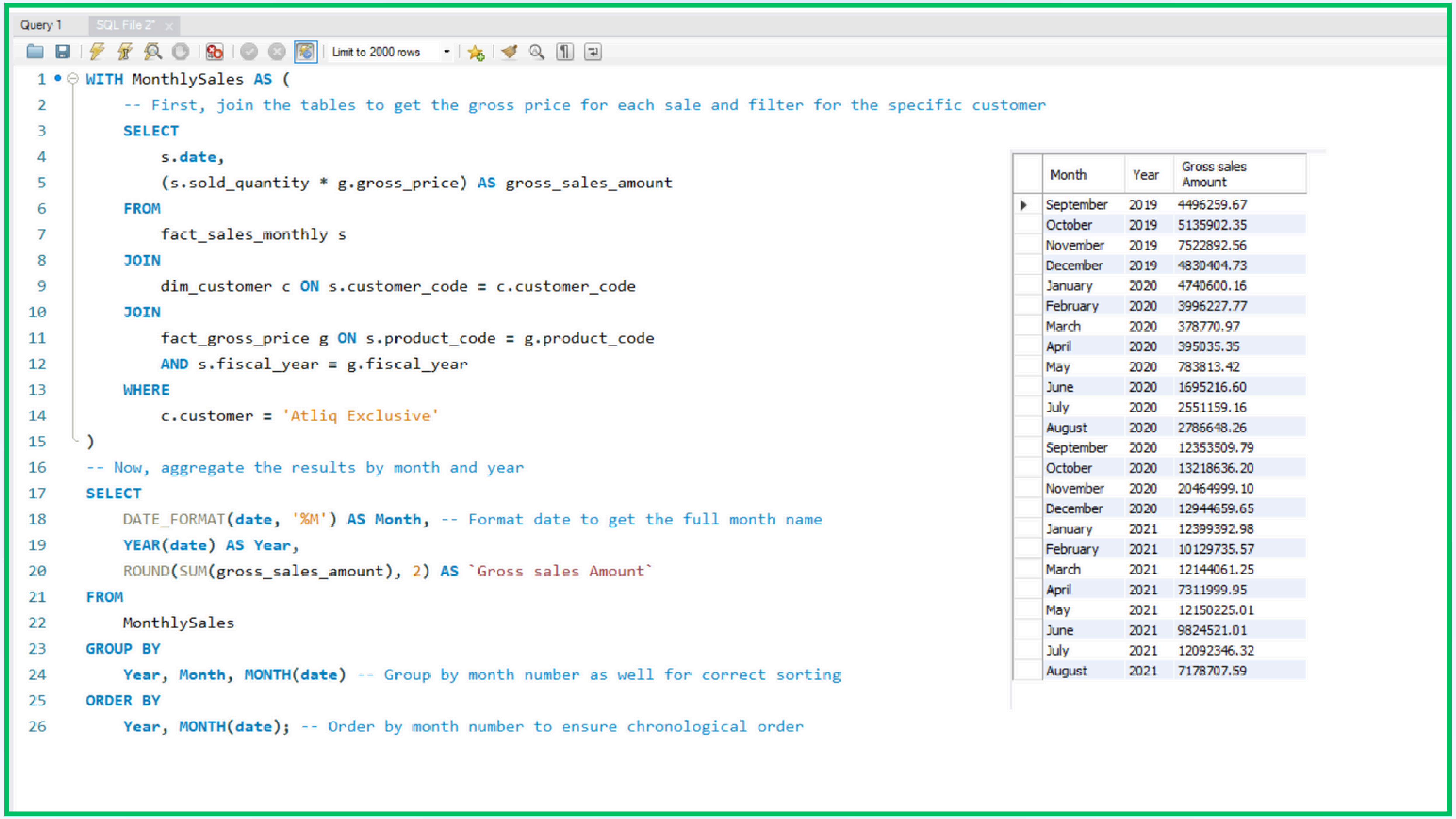
```sql
WITH CustomerDiscounts AS (
    SELECT
        c.customer_code,
        c.customer,
        AVG(pid.pre_invoice_discount_pct) AS average_discount_percentage
    FROM
        fact_pre_invoice_deductions pid
    JOIN
        dim_customer c ON pid.customer_code = c.customer_code
    WHERE
        pid.fiscal_year = 2021
        AND c.market = 'India'
    GROUP BY
        c.customer_code,
        c.customer),
RankedCustomers AS (
    SELECT
        customer_code,
        customer,
        average_discount_percentage,
        DENSE_RANK() OVER (ORDER BY average_discount_percentage DESC) as discount_rank
    FROM
        CustomerDiscounts
)
SELECT
    customer_code,
    customer,
    ROUND(average_discount_percentage, 4) AS average_discount_percentage
FROM
    RankedCustomers
WHERE
    discount_rank <= 5;
```

customer_code	customer	average_discount_percentage
90002009	Flipkart	0.3083
90002006	Viveks	0.3038
90002003	Ezone	0.3028
90002002	Croma	0.3025
90002016	Amazon	0.2933

Mohammed Raza Govani

Request 7

- **Business Question:** Monthly gross sales for “AtliQ Exclusive”
- **SQL Focus:** CTE’s + Advanced JOIN’s + Date Functions



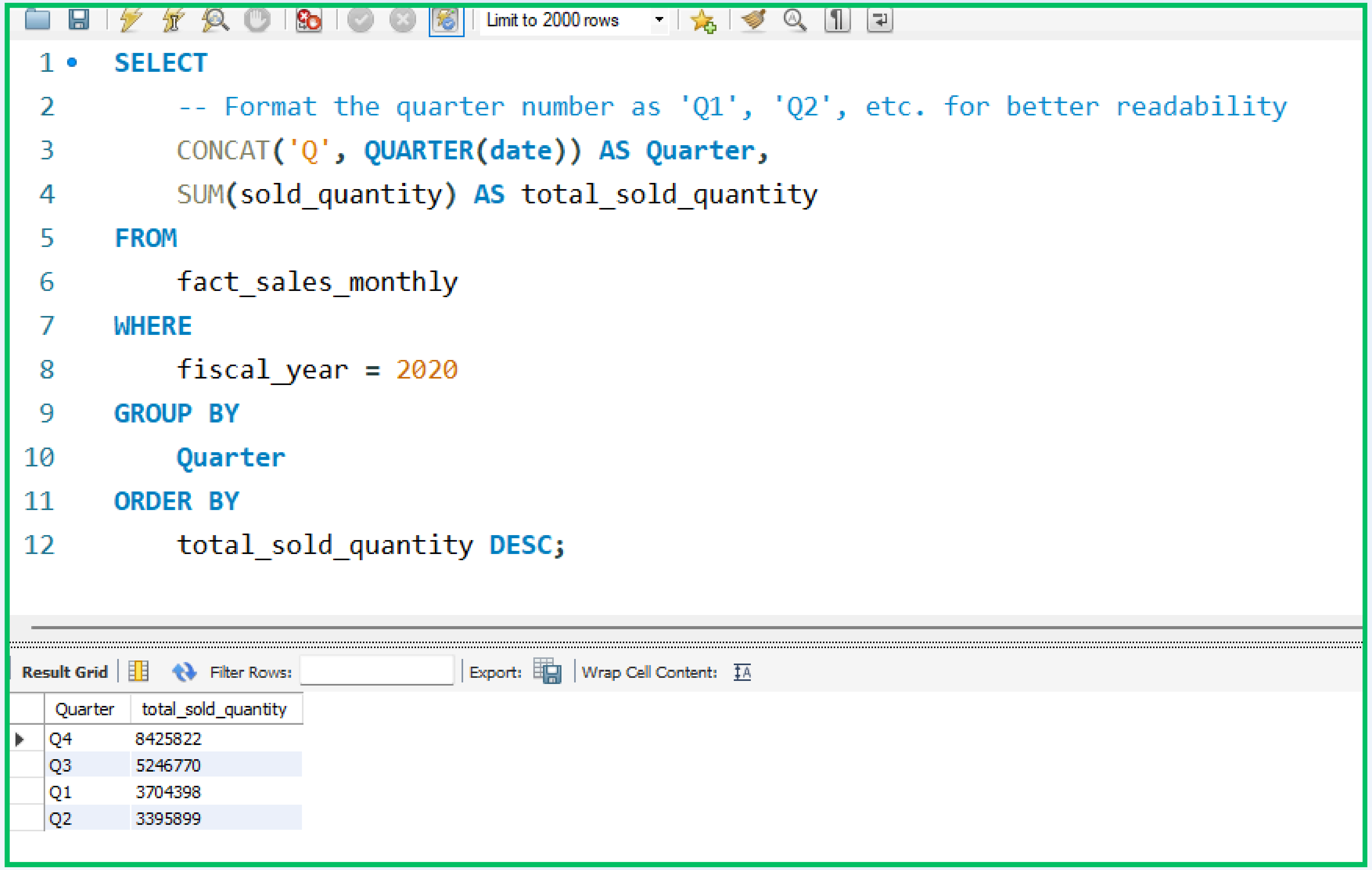
```sql
WITH MonthlySales AS (
    -- First, join the tables to get the gross price for each sale and filter for the specific customer
    SELECT
        s.date,
        (s.sold_quantity * g.gross_price) AS gross_sales_amount
    FROM
        fact_sales_monthly s
    JOIN
        dim_customer c ON s.customer_code = c.customer_code
    JOIN
        fact_gross_price g ON s.product_code = g.product_code
        AND s.fiscal_year = g.fiscal_year
    WHERE
        c.customer = 'Atliq Exclusive'
)
-- Now, aggregate the results by month and year
SELECT
    DATE_FORMAT(date, '%M') AS Month, -- Format date to get the full month name
    YEAR(date) AS Year,
    ROUND(SUM(gross_sales_amount), 2) AS `Gross sales Amount`
FROM
    MonthlySales
GROUP BY
    Year, Month, MONTH(date) -- Group by month number as well for correct sorting
ORDER BY
    Year, MONTH(date); -- Order by month number to ensure chronological order
```

Month	Year	Gross sales Amount
September	2019	4496259.67
October	2019	5135902.35
November	2019	7522892.56
December	2019	4830404.73
January	2020	4740600.16
February	2020	3996227.77
March	2020	378770.97
April	2020	395035.35
May	2020	783813.42
June	2020	1695216.60
July	2020	2551159.16
August	2020	2786648.26
September	2020	12353509.79
October	2020	13218636.20
November	2020	20464999.10
December	2020	12944659.65
January	2021	12399392.98
February	2021	10129735.57
March	2021	12144061.25
April	2021	7311999.95
May	2021	12150225.01
June	2021	9824521.01
July	2021	12092346.32
August	2021	7178707.59

Mohammed Raza Govani

Request 8

- **Business Question:** Quarter of 2020 with maximum sold quantity
- **SQL Focus:** String + Date + Aggregate Functions



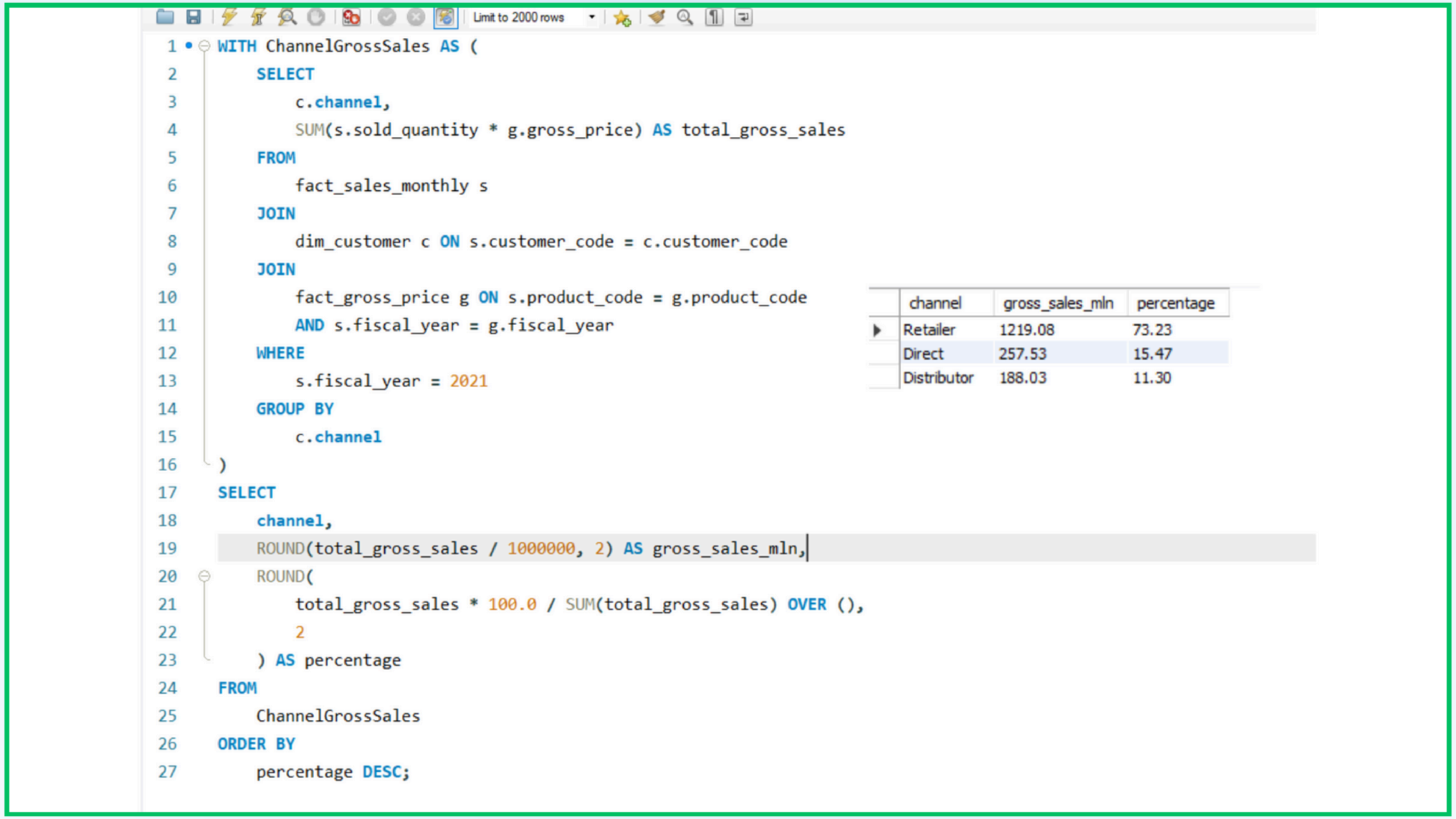
```sql
SELECT
    -- Format the quarter number as 'Q1', 'Q2', etc. for better readability
    CONCAT('Q', QUARTER(date)) AS Quarter,
    SUM(sold_quantity) AS total_sold_quantity
FROM
    fact_sales_monthly
WHERE
    fiscal_year = 2020
GROUP BY
    Quarter
ORDER BY
    total_sold_quantity DESC;
```

Quarter	total_sold_quantity
Q4	8425822
Q3	5246770
Q1	3704398
Q2	3395899

Mohammed Raza Govani

Request 9

- **Business Question:** Channel with highest FY21 gross sales & its % contribution
- **SQL Focus:** CTE + Window Functions + JOINs



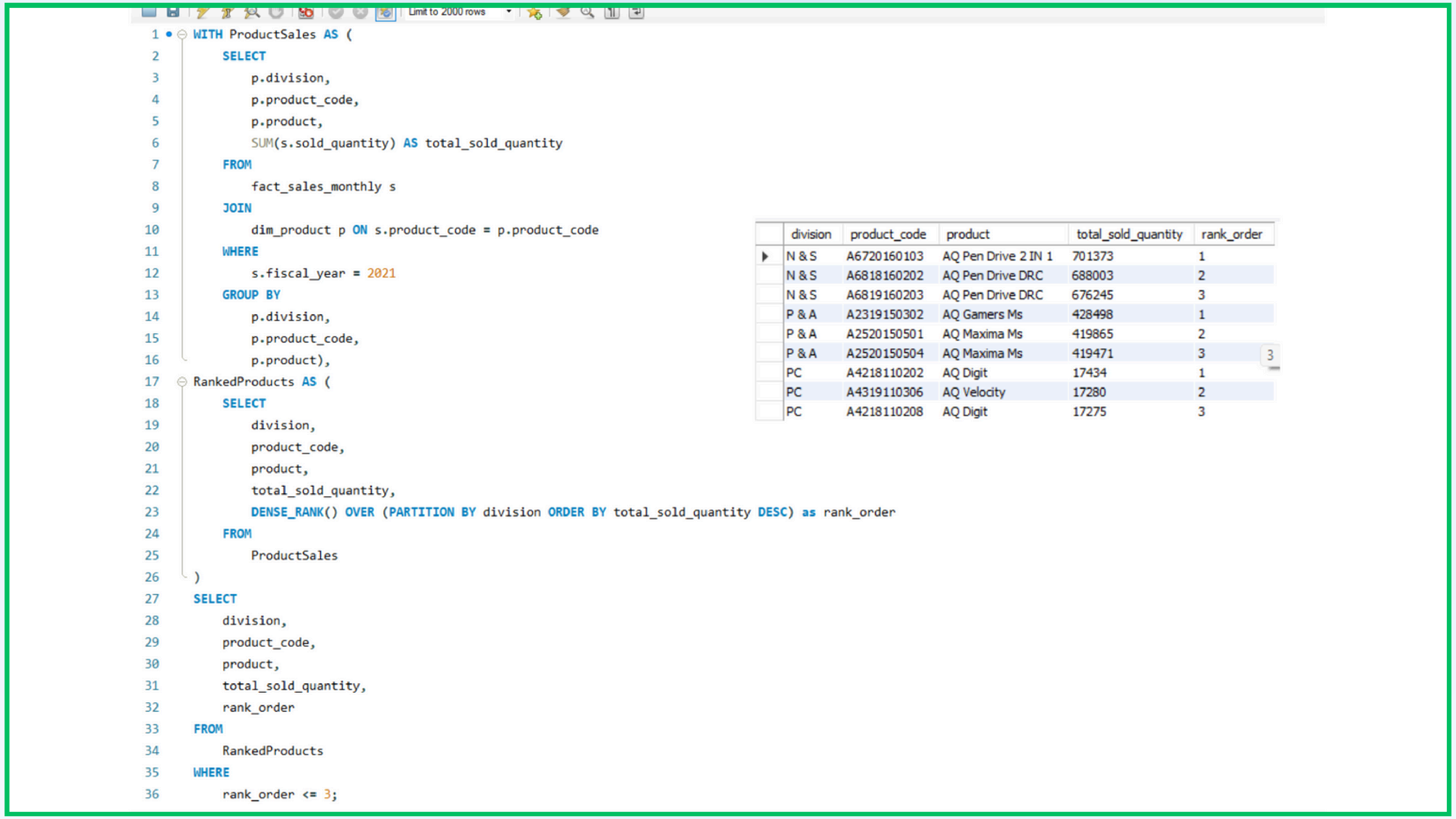
```sql
WITH ChannelGrossSales AS (
    SELECT
        c.channel,
        SUM(s.sold_quantity * g.gross_price) AS total_gross_sales
    FROM
        fact_sales_monthly s
    JOIN
        dim_customer c ON s.customer_code = c.customer_code
    JOIN
        fact_gross_price g ON s.product_code = g.product_code
        AND s.fiscal_year = g.fiscal_year
    WHERE
        s.fiscal_year = 2021
    GROUP BY
        c.channel
)
SELECT
    channel,
    ROUND(total_gross_sales / 1000000, 2) AS gross_sales_mln,
    ROUND(
        total_gross_sales * 100.0 / SUM(total_gross_sales) OVER (),
        2
    ) AS percentage
FROM
    ChannelGrossSales
ORDER BY
    percentage DESC;
```

channel	gross_sales_mln	percentage
Retailer	1219.08	73.23
Direct	257.53	15.47
Distributor	188.03	11.30

Mohammed Raza Govani

Request 10

- **Business Question:** Top 3 products by sold quantity in each division (FY21)
- **SQL Focus:** Advanced CTE's + Window Functions



```sql
WITH ProductSales AS (
    SELECT
        p.division,
        p.product_code,
        p.product,
        SUM(s.sold_quantity) AS total_sold_quantity
    FROM
        fact_sales_monthly s
    JOIN
        dim_product p ON s.product_code = p.product_code
    WHERE
        s.fiscal_year = 2021
    GROUP BY
        p.division,
        p.product_code,
        p.product),
RankedProducts AS (
    SELECT
        division,
        product_code,
        product,
        total_sold_quantity,
        DENSE_RANK() OVER (PARTITION BY division ORDER BY total_sold_quantity DESC) as rank_order
    FROM
        ProductSales
)
SELECT
    division,
    product_code,
    product,
    total_sold_quantity,
    rank_order
FROM
    RankedProducts
WHERE
    rank_order <= 3;
```

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

Mohammed Raza Govani

Key Business Insights

- **Retailer Channel Leads** – Over **73% of FY21 gross sales** came from the retailer channel, making it the company's primary revenue driver.
- **Notebook Segment Expansion** – The **"Notebook" category recorded the largest jump in unique products** between 2020 and 2021.
- **Peak Sales Quarter** – **Q2 of 2020 posted the highest sales volume,** marking the year's strongest performance period.
- **High-Discount Customers** – In India, the **top five customers enjoyed notably higher average discounts,** indicating room to optimize margins.

Mohammed Raza Govani

THANK YOU

Project Takeaway

Advanced SQL can uncover business-critical insights from millions of records, guiding real-world decision making.

Let’s Connect

LinkedIn: https://www.linkedin.com/razagovani/
GitHub: https://github.com/gRaza25
Email: mohammedgovani18@gmail.com

Mohammed Raza Govani