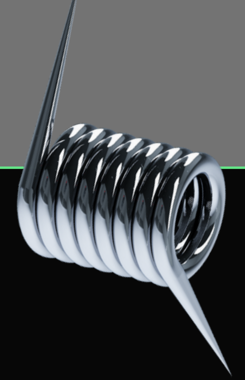
SQL-Driven Insights for Atliq Hardware's Global Sales



- Aditi Teli

Atliq Hardware: Business Overview

Atliq Hardware is a global hardware manufacturing company with a business model similar to brands like HP and Dell. They produce a wide range of hardware products, including PCs, mouse, printers, and other computer peripherals, which they sell to various types of customers, from large retail chains to online marketplaces.

Business Structure

Atliq’s supply chain is extensive, incorporating their own manufacturing facilities, warehouses, and distribution centers. This setup allows them to manufacture products, stock them in regional warehouses, and ship them worldwide to meet customer demand. The company's distribution strategy includes three distinct channels to meet the unique needs of different markets and customer types

Retailer Channel

This channel includes sales to third-party retailers who then sell Atliq’s products to end consumers.
The retailer channel is split into two main types:

Brick & Mortar Stores: Traditional physical stores such as Croma and Best Buy that stock Atliq products and sell them directly to walk-in customers.

E-commerce Platforms: Online marketplaces such as Amazon and Flipkart that serve as digital storefronts for Atliq’s products, making it convenient for customers to shop online.

Direct Channel

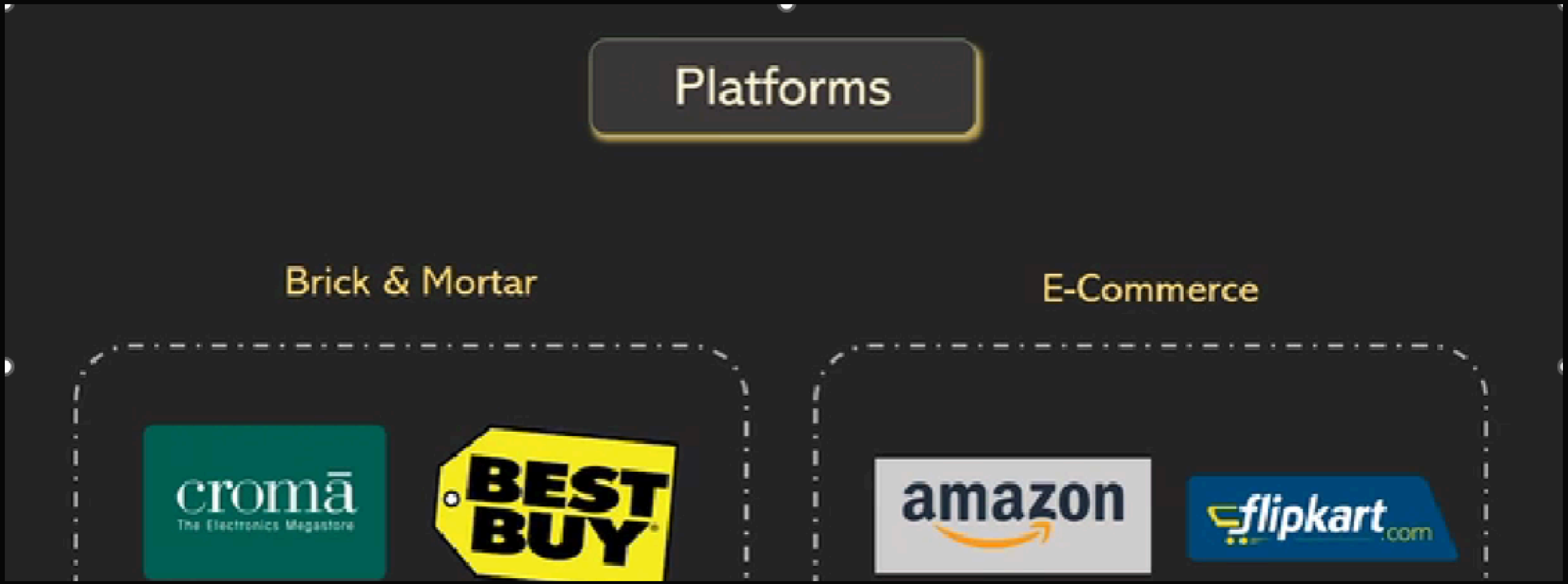
The direct channel includes Atliq’s own consumer-facing platforms:

Atliq Online: The company’s exclusive online store, where end consumers can browse and purchase products directly from Atliq without intermediaries.

Atliq Exclusive: Physical retail stores that are owned and operated by Atliq, offering a direct point of sale to customers who prefer an in-store shopping experience. This channel allows Atliq to build a strong brand presence and interact directly with customers, ensuring quality and customer satisfaction.

Distributor Channel

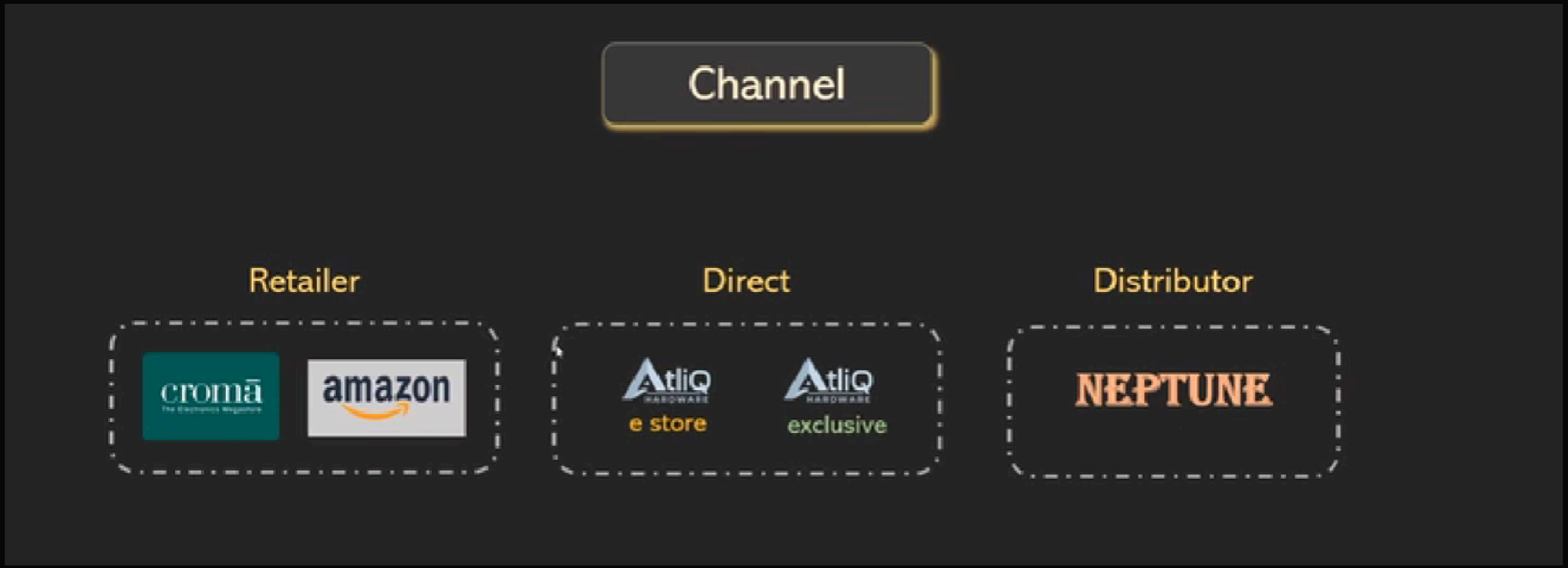
In certain international markets where direct sales are not feasible, Atliq operates through distributors. For instance, in countries like China, Atliq supplies products to authorized distributors, who then handle logistics and delivery to various retailers. This distributor channel allows Atliq to maintain a presence in regions with complex market entry regulations while reaching end consumers through authorized local sellers.



Platforms

Brick & Mortar

E-Commerce



Channel

Retailer

Direct

Distributor

Description of the tables in the database:

1. dim_customer

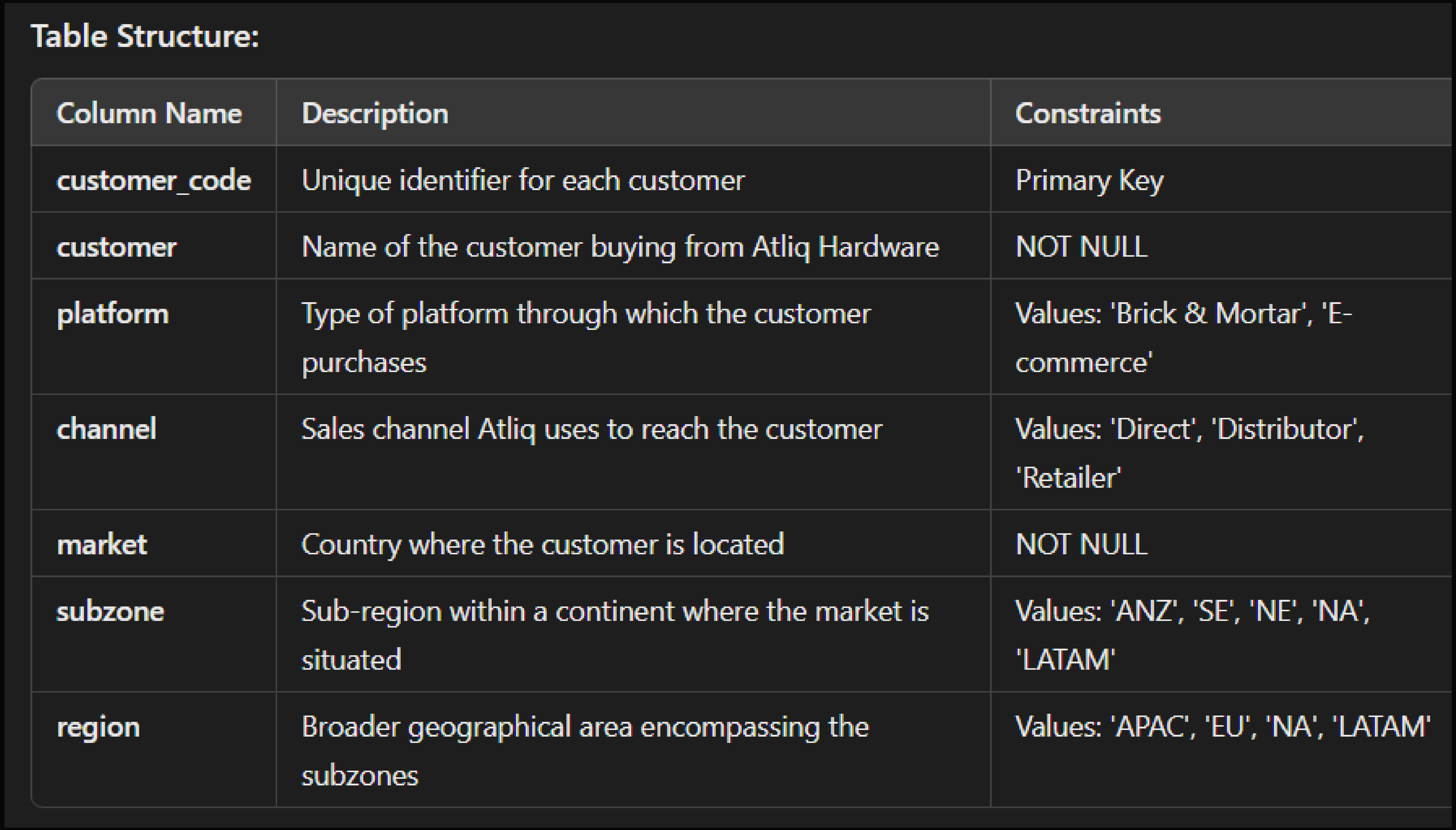
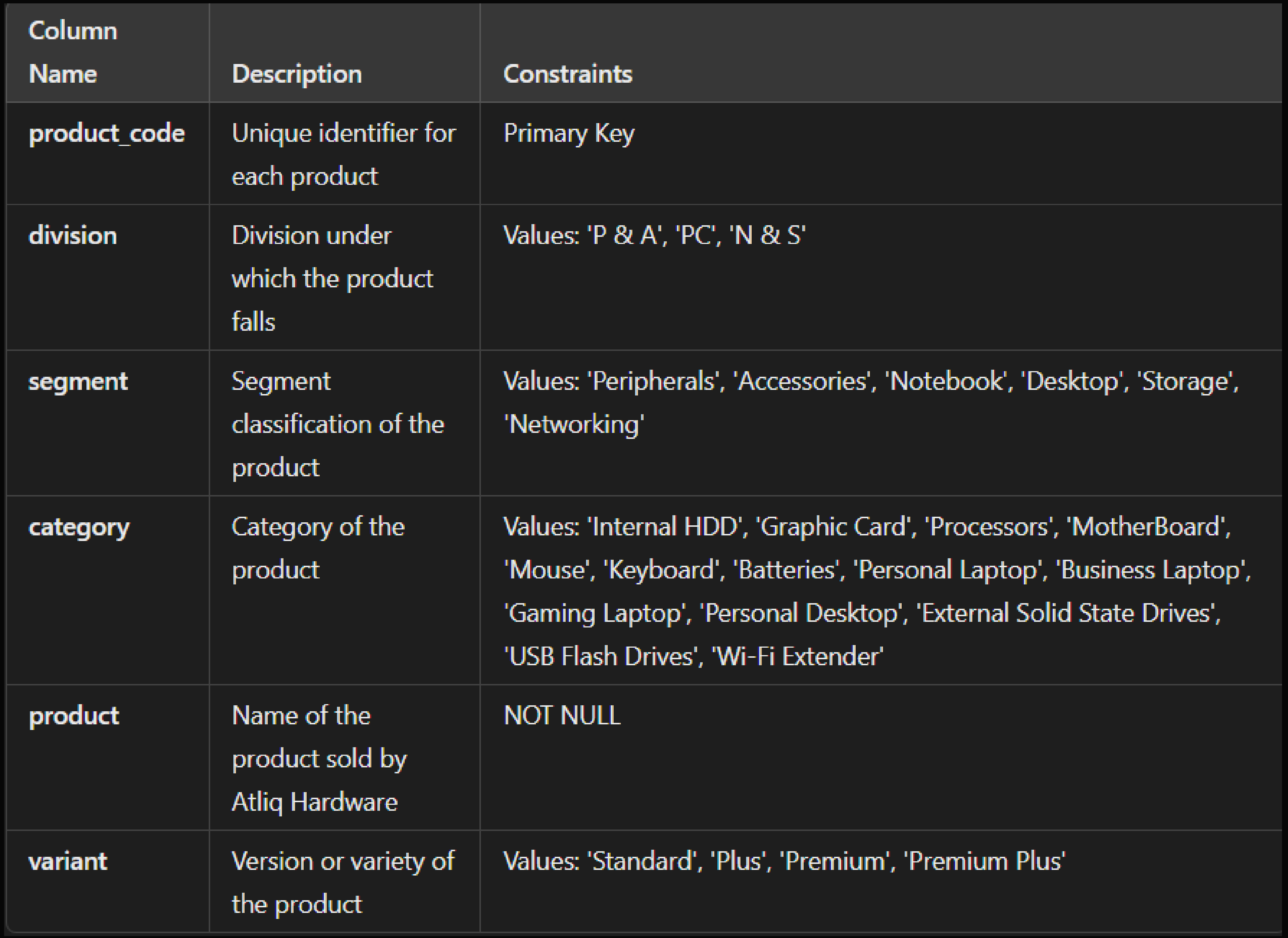


Table Structure:

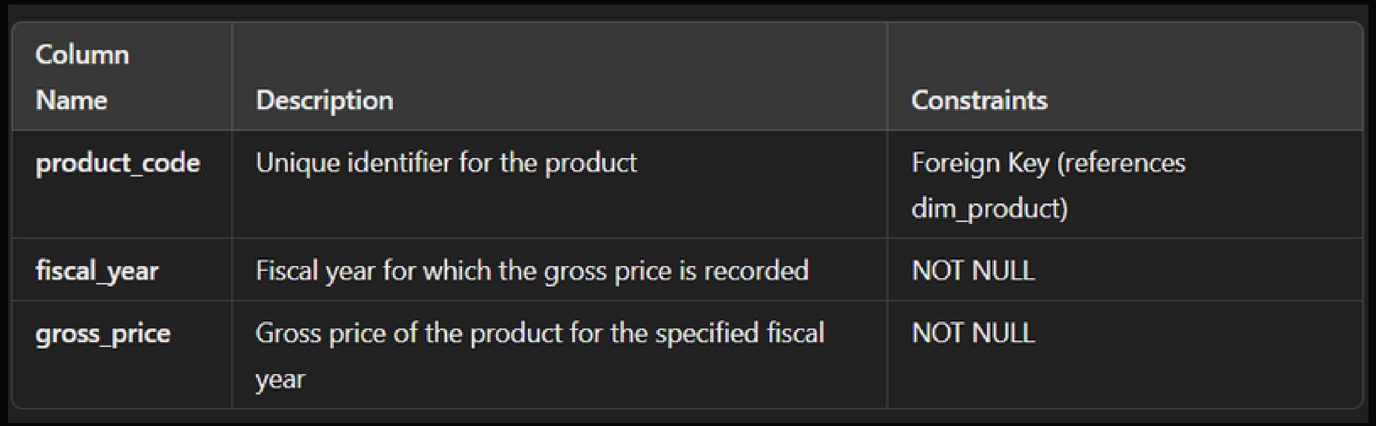
Column Name	Description	Constraints
customer_code	Unique identifier for each customer	Primary Key
customer	Name of the customer buying from Atliq Hardware	NOT NULL
platform	Type of platform through which the customer purchases	Values: 'Brick & Mortar', 'E-commerce'
channel	Sales channel Atliq uses to reach the customer	Values: 'Direct', 'Distributor', 'Retailer'
market	Country where the customer is located	NOT NULL
subzone	Sub-region within a continent where the market is situated	Values: 'ANZ', 'SE', 'NE', 'NA', 'LATAM'
region	Broader geographical area encompassing the subzones	Values: 'APAC', 'EU', 'NA', 'LATAM'

2. dim_product



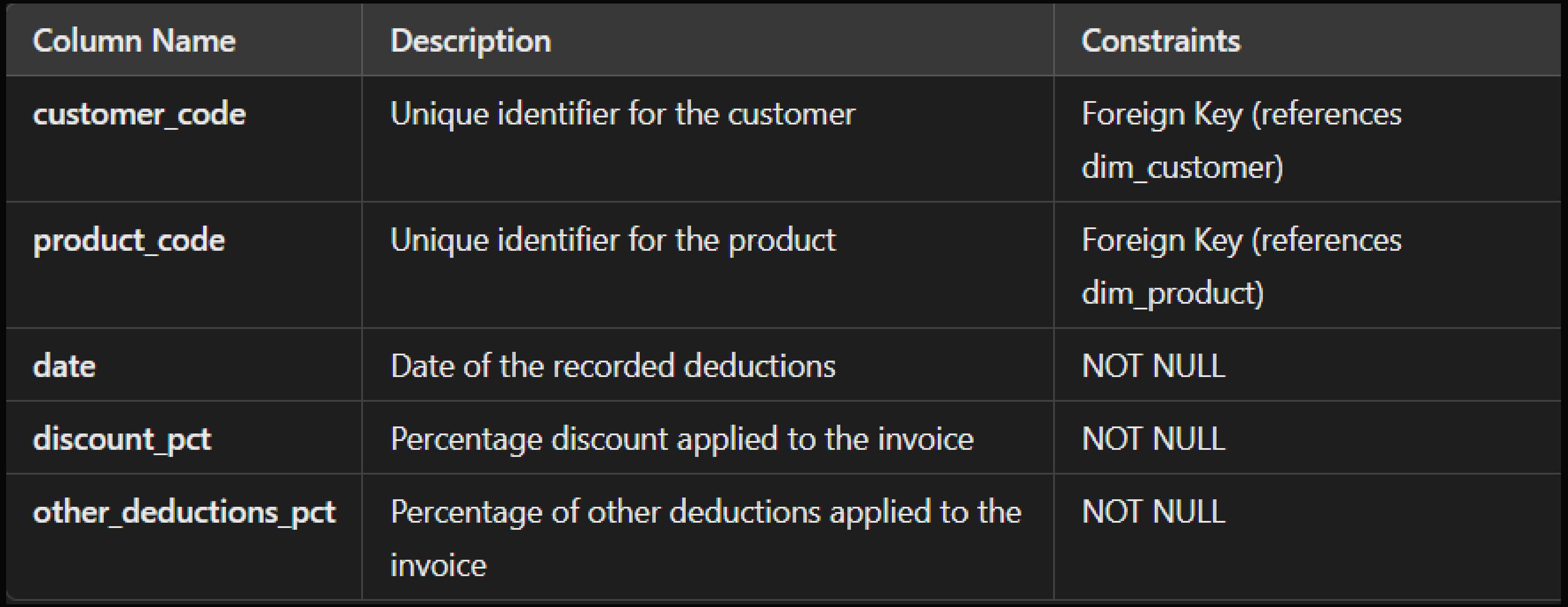
Column Name	Description	Constraints
product_code	Unique identifier for each product	Primary Key
division	Division under which the product falls	Values: 'P & A', 'PC', 'N & S'
segment	Segment classification of the product	Values: 'Peripherals', 'Accessories', 'Notebook', 'Desktop', 'Storage', 'Networking'
category	Category of the product	Values: 'Internal HDD', 'Graphic Card', 'Processors', 'MotherBoard', 'Mouse', 'Keyboard', 'Batteries', 'Personal Laptop', 'Business Laptop', 'Gaming Laptop', 'Personal Desktop', 'External Solid State Drives', 'USB Flash Drives', 'Wi-Fi Extender'
product	Name of the product sold by Atliq Hardware	NOT NULL
variant	Version or variety of the product	Values: 'Standard', 'Plus', 'Premium', 'Premium Plus'

3. fact_gross_price



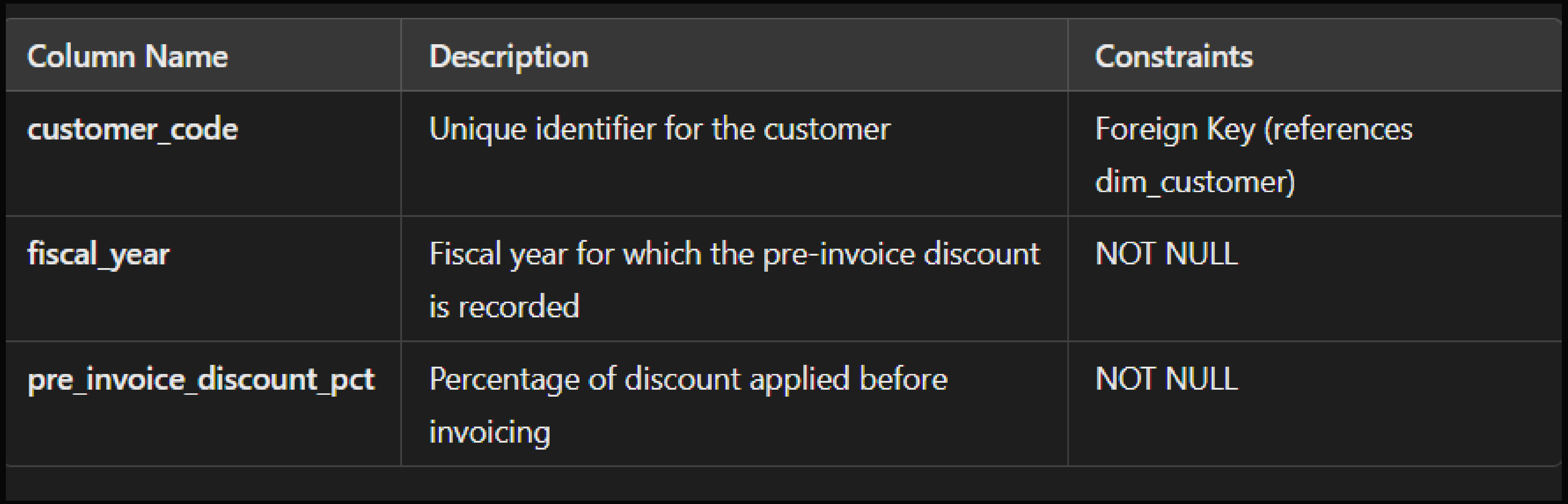
Column Name	Description	Constraints
product_code	Unique identifier for the product	Foreign Key (references dim_product)
fiscal_year	Fiscal year for which the gross price is recorded	NOT NULL
gross_price	Gross price of the product for the specified fiscal year	NOT NULL

4. fact_post_invoice_deductions



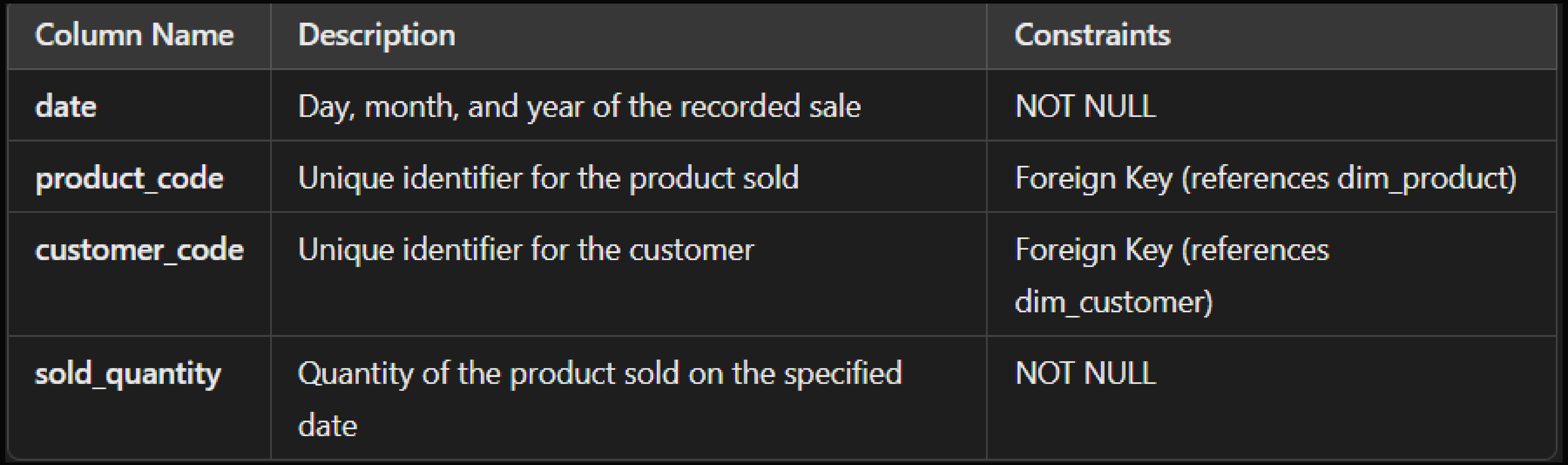
Column Name	Description	Constraints
customer_code	Unique identifier for the customer	Foreign Key (references dim_customer)
product_code	Unique identifier for the product	Foreign Key (references dim_product)
date	Date of the recorded deductions	NOT NULL
discount_pct	Percentage discount applied to the invoice	NOT NULL
other_deductions_pct	Percentage of other deductions applied to the invoice	NOT NULL

5. fact_pre_invoice_deductions



Column Name	Description	Constraints
customer_code	Unique identifier for the customer	Foreign Key (references dim_customer)
fiscal_year	Fiscal year for which the pre-invoice discount is recorded	NOT NULL
pre_invoice_discount_pct	Percentage of discount applied before invoicing	NOT NULL

6. fact_sales_monthly



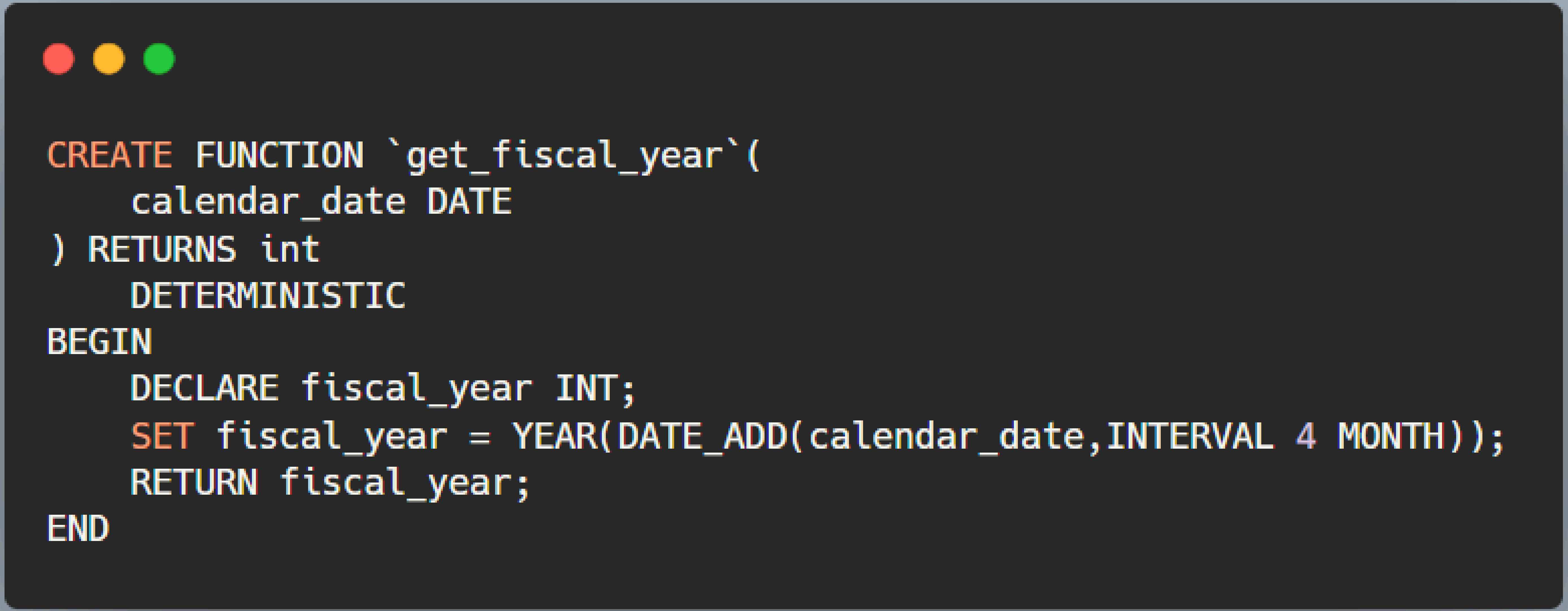
Column Name	Description	Constraints
date	Day, month, and year of the recorded sale	NOT NULL
product_code	Unique identifier for the product sold	Foreign Key (references dim_product)
customer_code	Unique identifier for the customer	Foreign Key (references dim_customer)
sold_quantity	Quantity of the product sold on the specified date	NOT NULL

Q1. As a product owner , I need an aggregate monthly gross sales report for Croma India customer so that I can track how much sales this particular customer is generating for Atliq and manage our relationships accordingly.

The report should have the following fields.

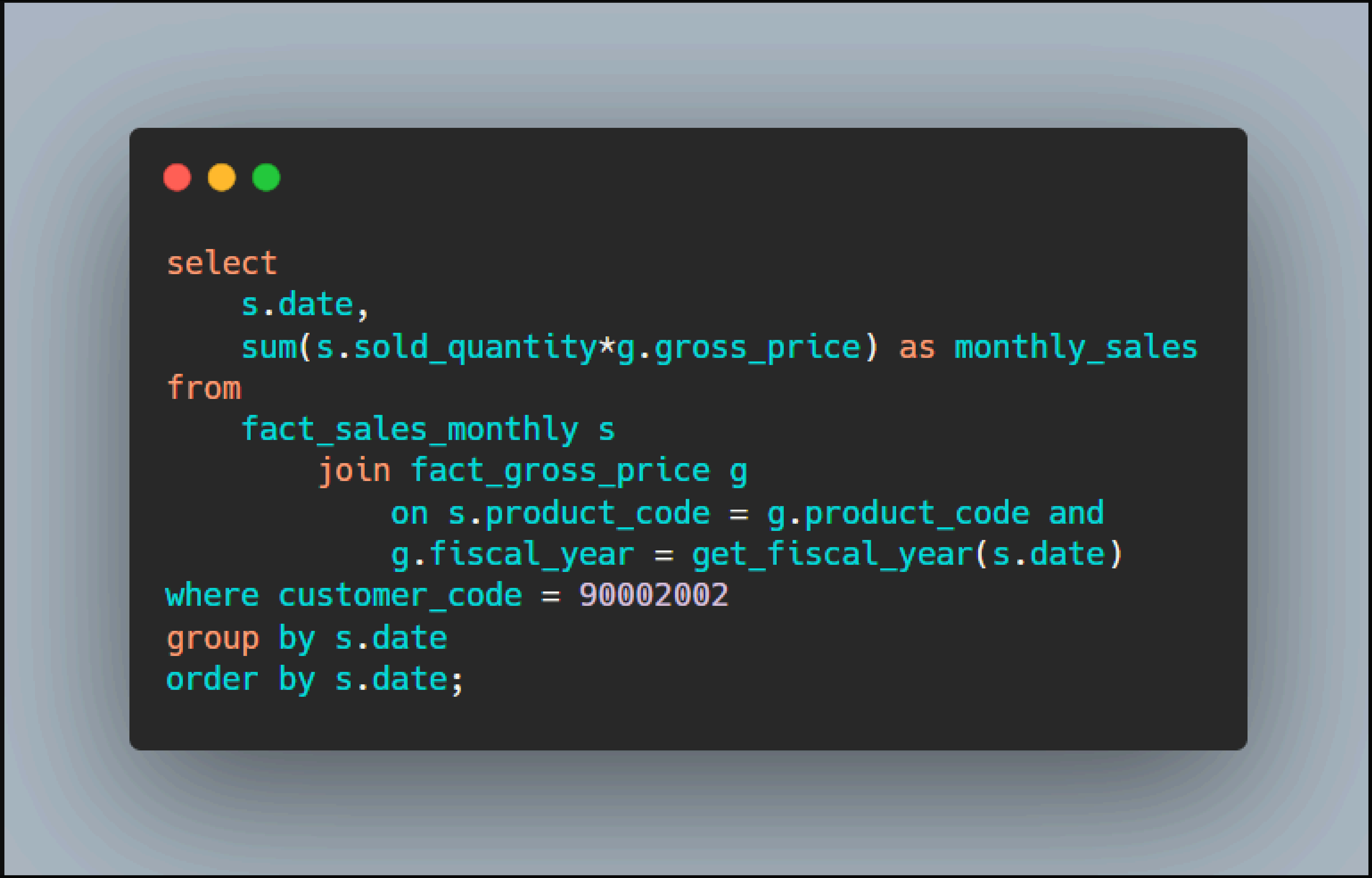
1. Month
2. Total gross sales amount to Croma India in this month

----> Inorder to solve this query we require two tables namely fact_sales_monthly and fact_gross_price. While solving this query the month was aggregated as the per calendar year and the Atliq has the fiscal year starting from 1st September - 31st Aug every year. We need to first convert the calendar date to fiscal date and perform the query accordingly. Here, we can make use of the “User defined function” in order to get the data in the fiscal year. We can write the function as : (see next slide)



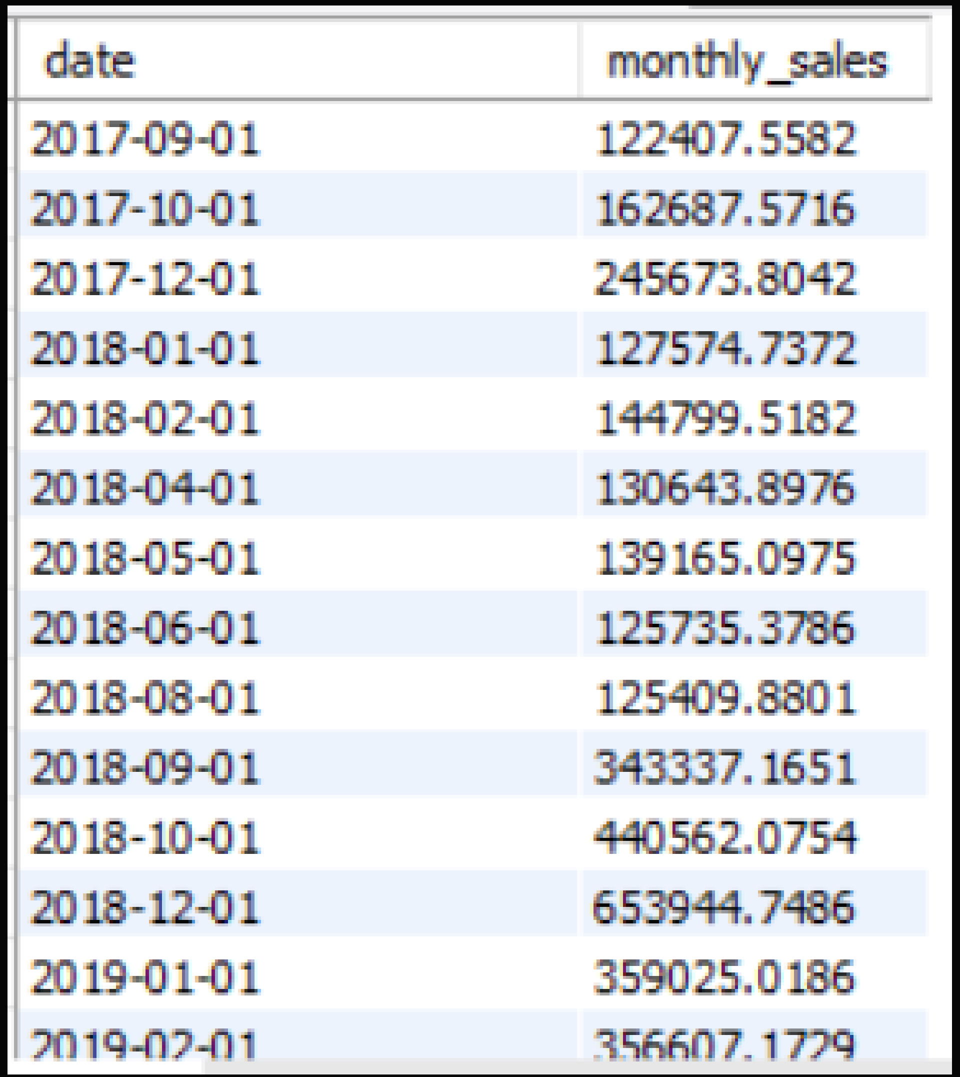
```sql
CREATE FUNCTION `get_fiscal_year`(
    calendar_date DATE
) RETURNS int
    DETERMINISTIC
BEGIN
    DECLARE fiscal_year INT;
    SET fiscal_year = YEAR(DATE_ADD(calendar_date,INTERVAL 4 MONTH));
    RETURN fiscal_year;
END
```

The query is now written as:



```
select
    s.date,
    sum(s.sold_quantity*g.gross_price) 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 = 90002002
group by s.date
order by s.date;
```

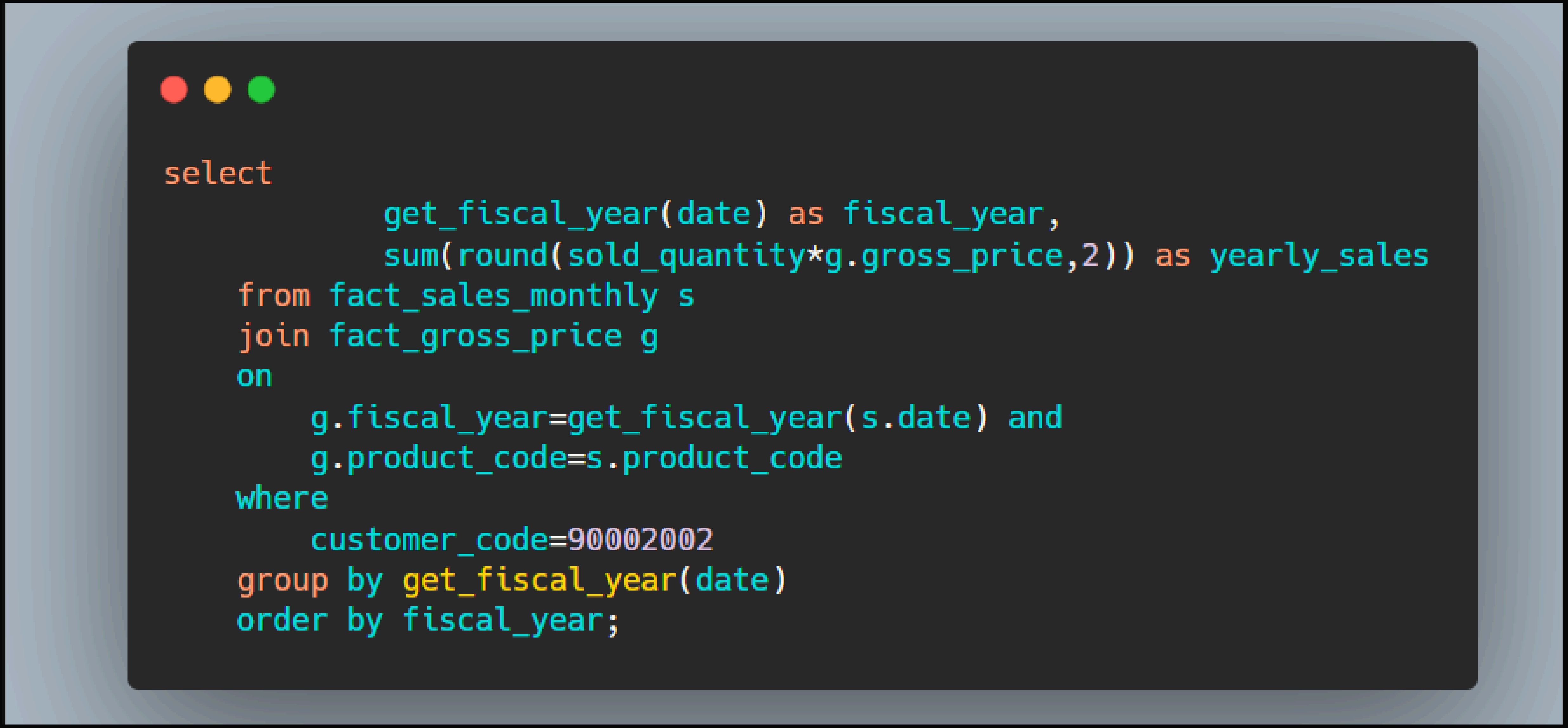
Output: :



date	monthly_sales
2017-09-01	122407.5582
2017-10-01	162687.5716
2017-12-01	245673.8042
2018-01-01	127574.7372
2018-02-01	144799.5182
2018-04-01	130643.8976
2018-05-01	139165.0975
2018-06-01	125735.3786
2018-08-01	125409.8801
2018-09-01	343337.1651
2018-10-01	440562.0754
2018-12-01	653944.7486
2019-01-01	359025.0186
2019-02-01	356607.1729

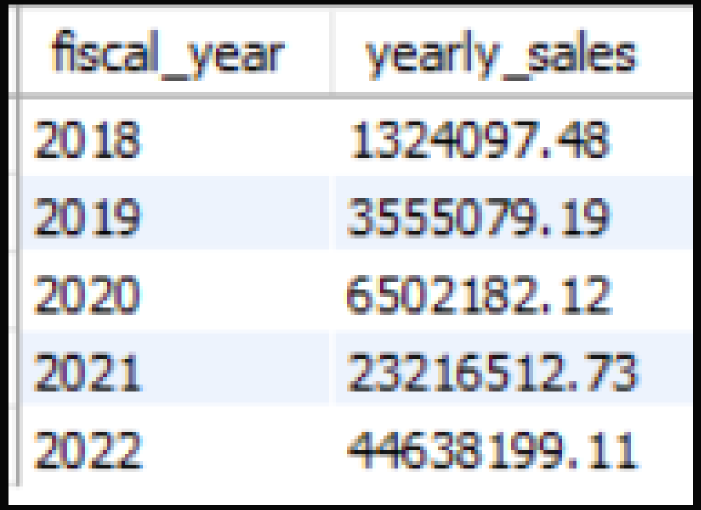
Generate a yearly report for Croma India where there are two columns

1. Fiscal Year
2. Total Gross Sales amount In that year from Croma



```
select
            get_fiscal_year(date) as fiscal_year,
            sum(round(sold_quantity*g.gross_price,2)) as yearly_sales
    from fact_sales_monthly s
    join fact_gross_price g
    on
        g.fiscal_year=get_fiscal_year(s.date) and
        g.product_code=s.product_code
    where
        customer_code=90002002
    group by get_fiscal_year(date)
    order by fiscal_year;
```

Output:



fiscal_year	yearly_sales
2018	1324097.48
2019	3555079.19
2020	6502182.12
2021	23216512.73
2022	44638199.11

Create a 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

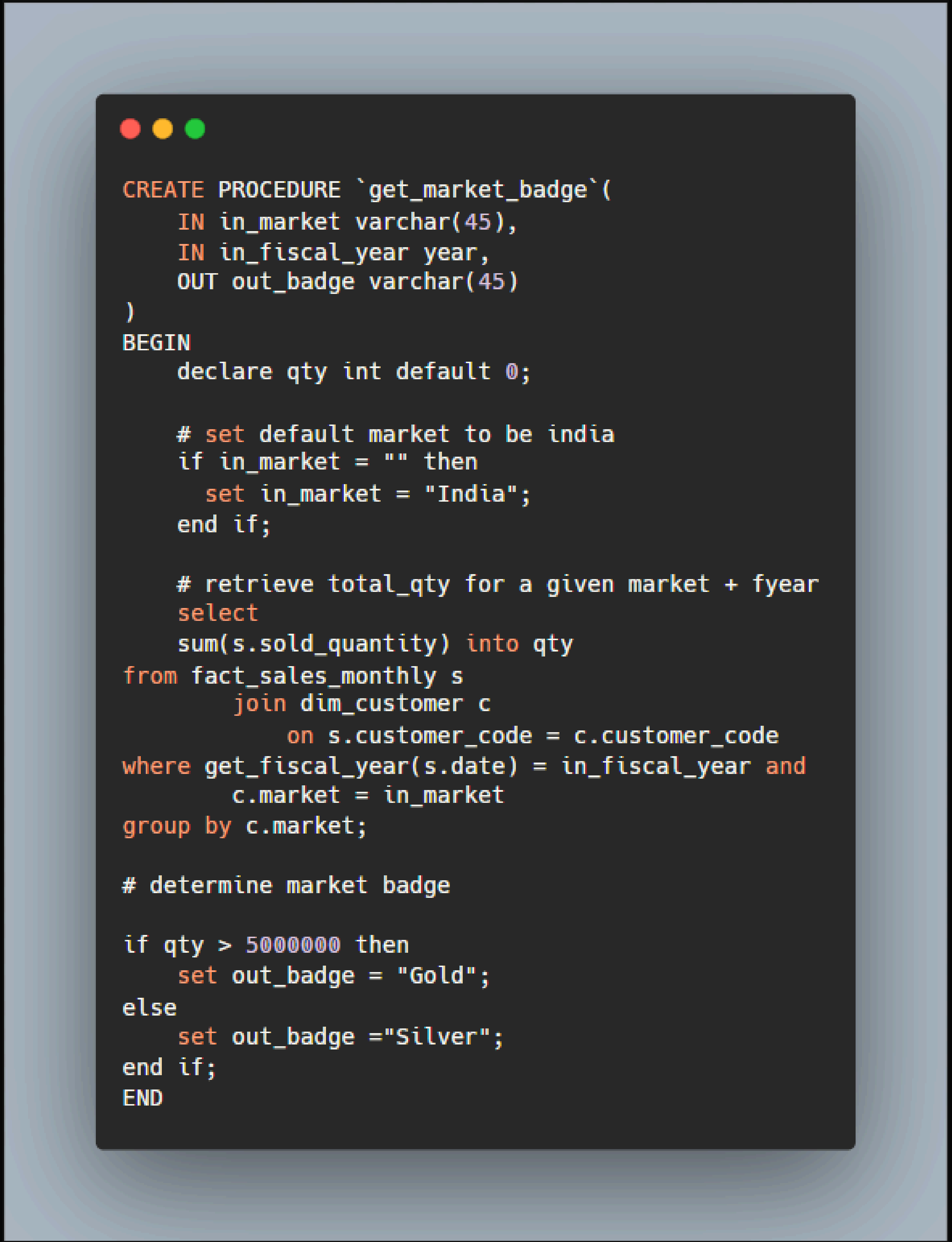
My input will be
Market
fiscal year

Output:
Market Badge

------>

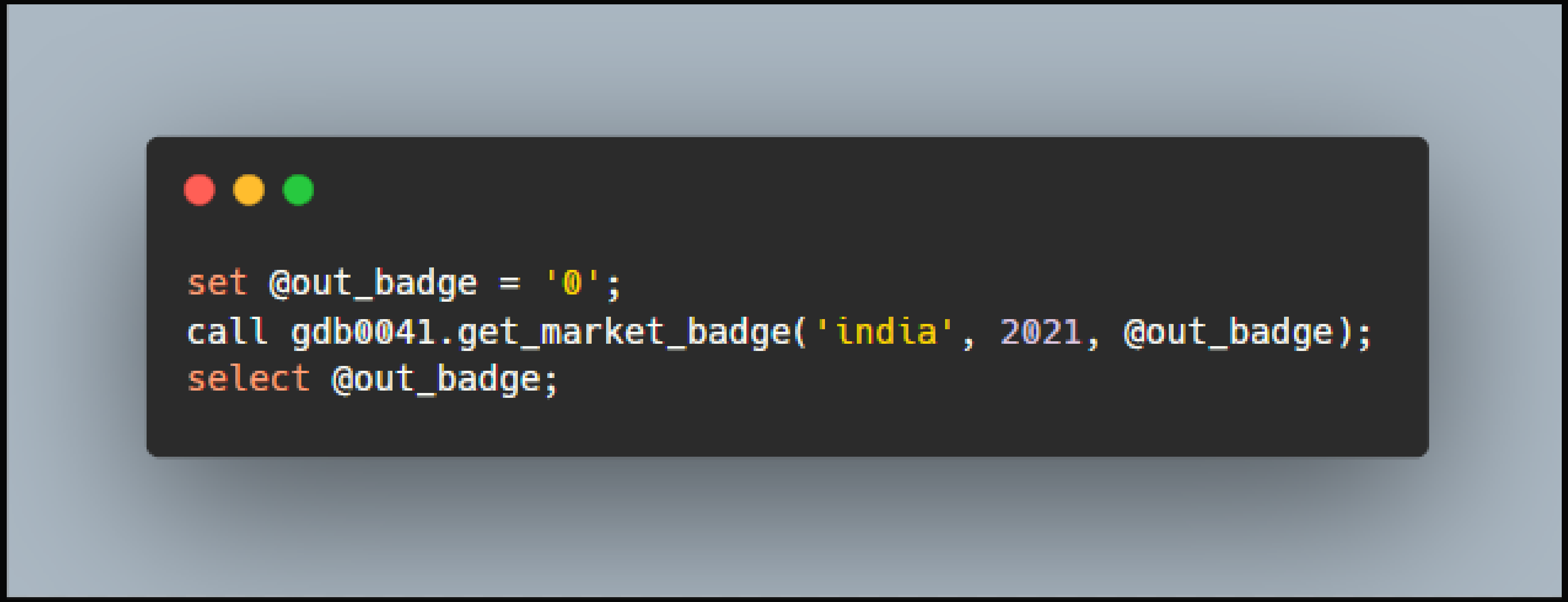
Summary of Components

- Inputs: in_market, in_fiscal_year
- Calculations: Sum of sold_quantity for specified market and fiscal_year
- Badge Assignment: Based on whether qty is greater than 5,000,000
- Output: out_badge set to either "Gold" or "Silver"



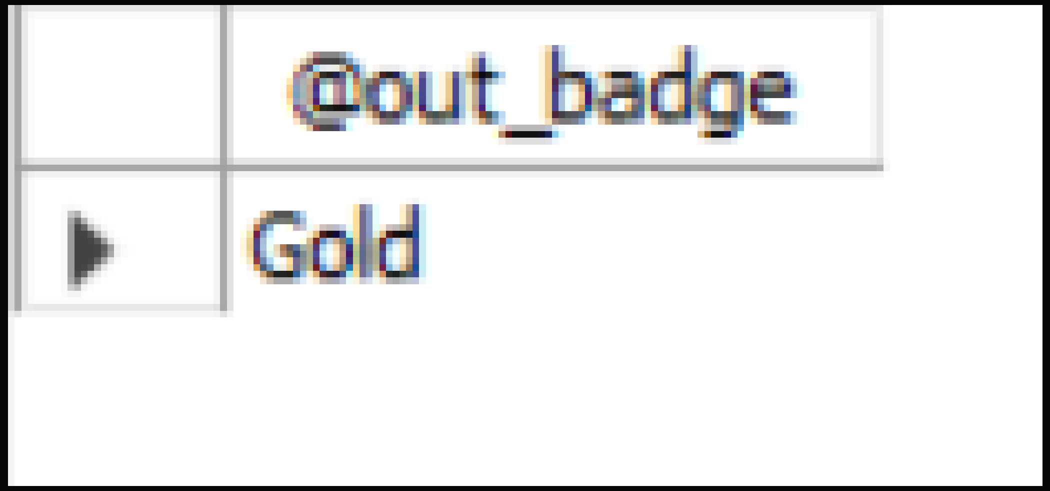
```
CREATE PROCEDURE `get_market_badge`(
    IN in_market varchar(45),
    IN in_fiscal_year year,
    OUT out_badge varchar(45)
)
BEGIN
    declare qty int default 0;

    # set default market to be india
    if in_market = "" then
      set in_market = "India";
    end if;

    # retrieve total_qty for a given market + fyear
    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 market badge

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

Example stored procedure:



```
set @out_badge = '0';
call gdb0041.get_market_badge('india', 2021, @out_badge);
select @out_badge;
```

Output:



```
   @out_badge
▶  Gold
```

Top customers, products and markets by net sales:

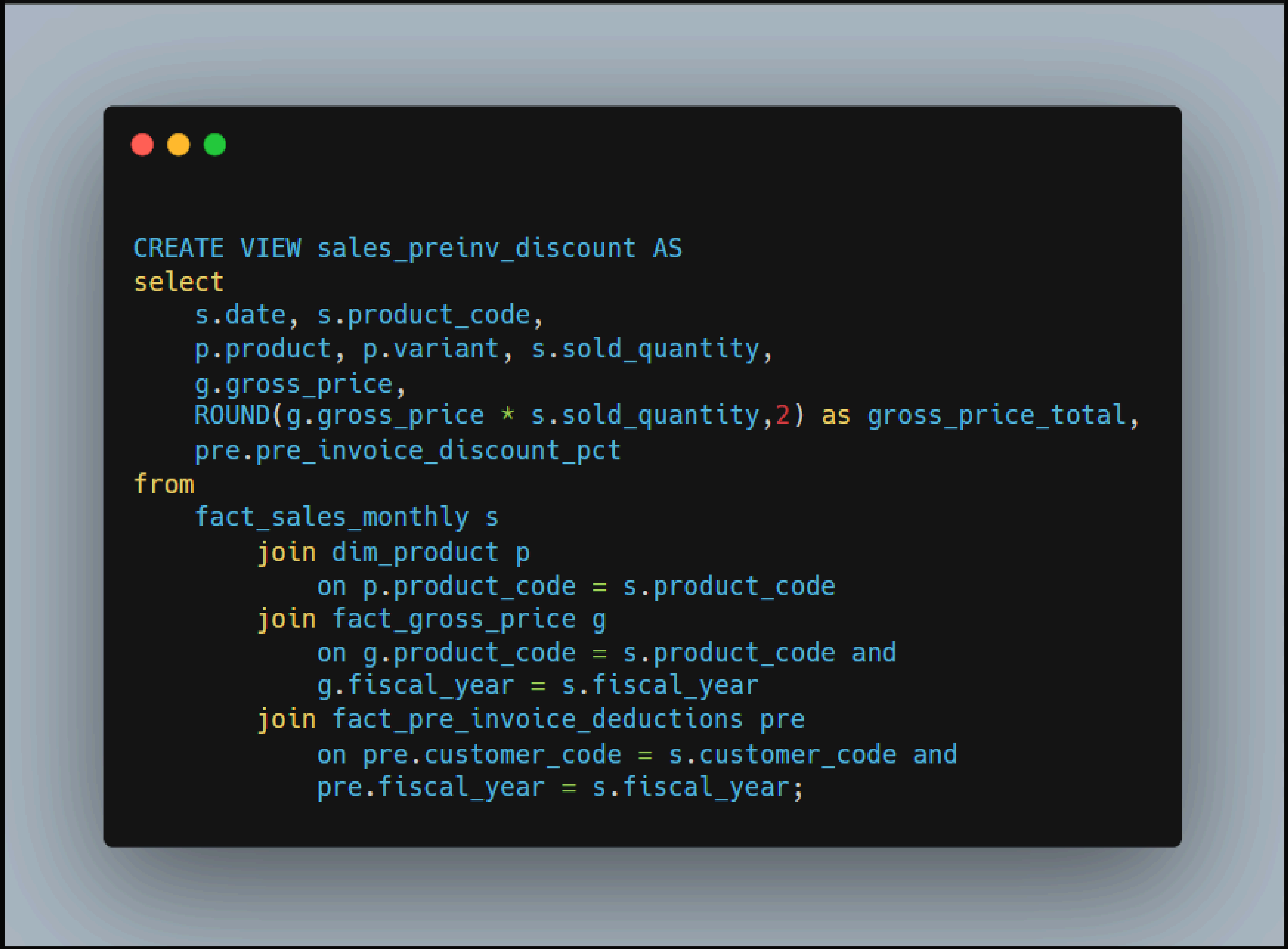
In order to get the net sales column we need to perform various joins and arrive at the result. This will make the query inefficient and difficult to read the query even if we arrive at the result. Hence, we need to create Views so that we can store the query in the form of table and run it anytime we require.

Here, in this case first we will create the view “sale_preinv_discount” and generate the column pre_invoice_discount_pct. Once, we get the column “pre_invoice_discount_pct.” then we need the column “post_invoice_discount_pct” which we can arrive by creating another view “sale_post_invoice_discount”. In this way we get the “Net invoice sale” by joining the two views. With the help of sale_post_invoice_discount we get the net sales finally.

In the next slide we will see the views created :

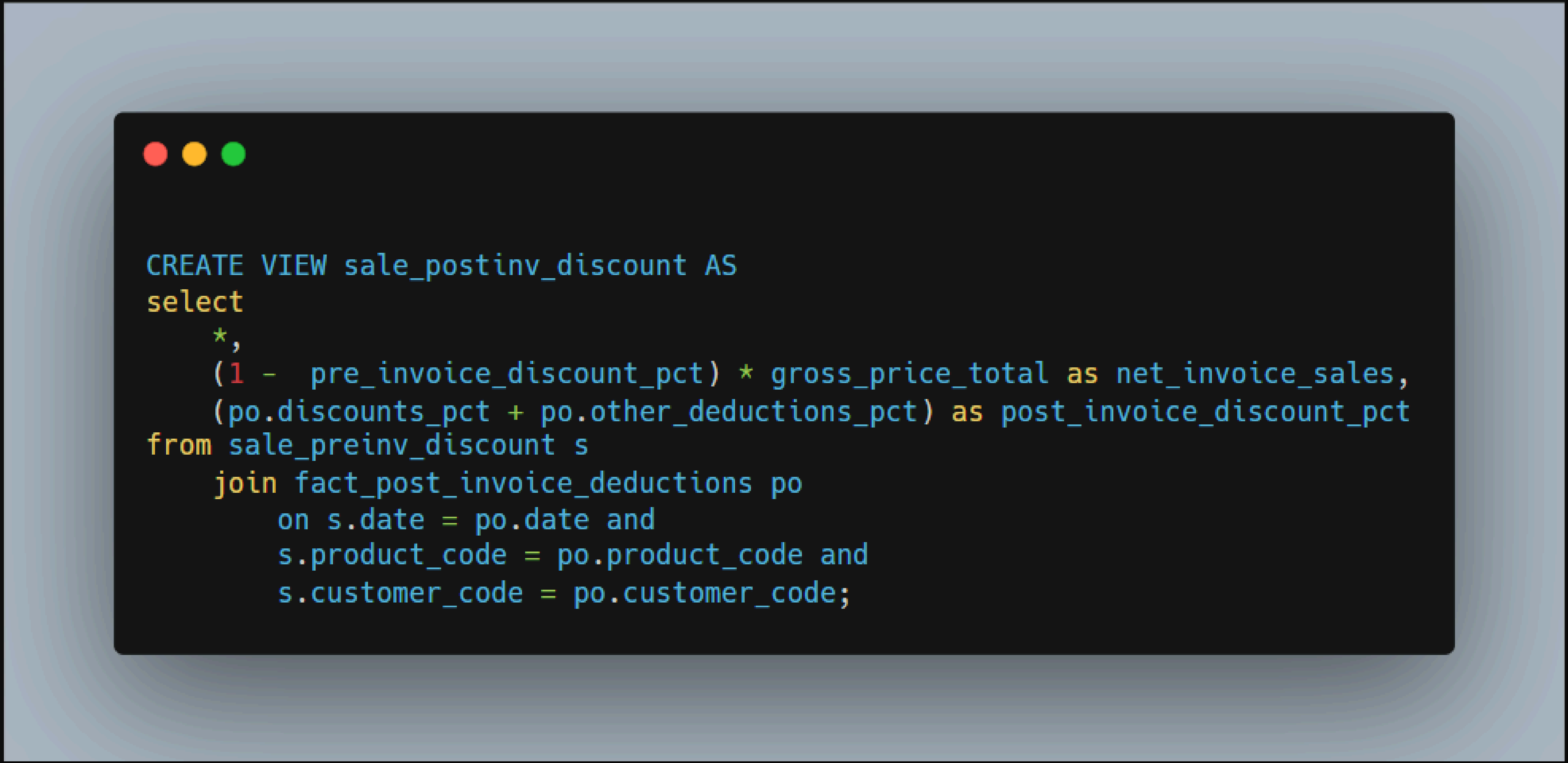
1. “sale_preinv_discount”
2. “sale_post_invoice_discount”

Sale_preinv_discount:



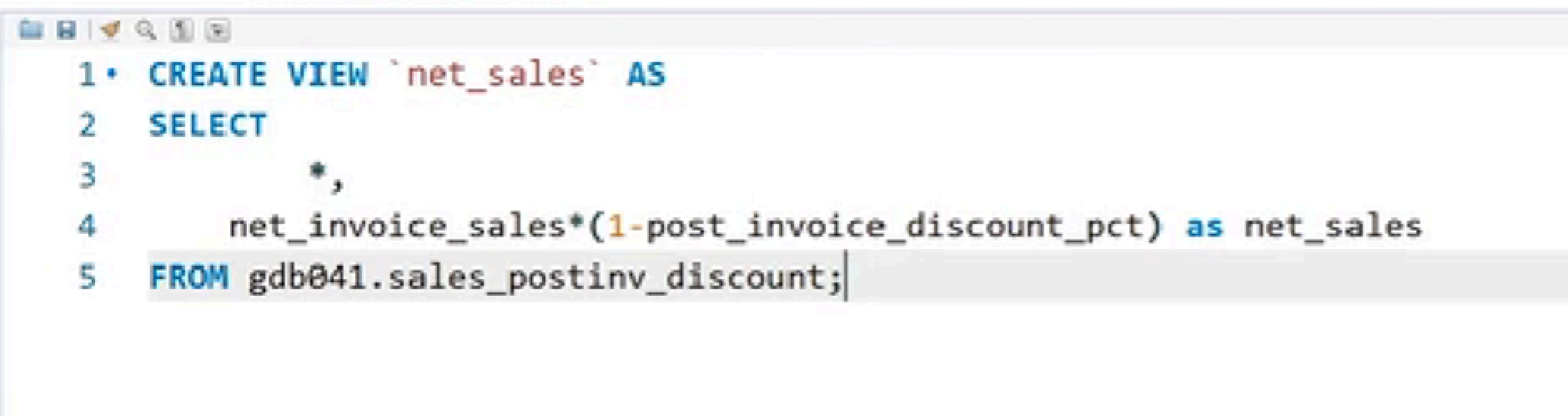
```sql
CREATE VIEW sales_preinv_discount AS
select
    s.date, s.product_code,
    p.product, p.variant, s.sold_quantity,
    g.gross_price,
    ROUND(g.gross_price * s.sold_quantity,2) as gross_price_total,
    pre.pre_invoice_discount_pct
from
    fact_sales_monthly s
        join dim_product p
            on p.product_code = s.product_code
        join fact_gross_price g
            on g.product_code = s.product_code and
            g.fiscal_year = s.fiscal_year
        join fact_pre_invoice_deductions pre
            on pre.customer_code = s.customer_code and
            pre.fiscal_year = s.fiscal_year;
```

Sale_postinv_discount:



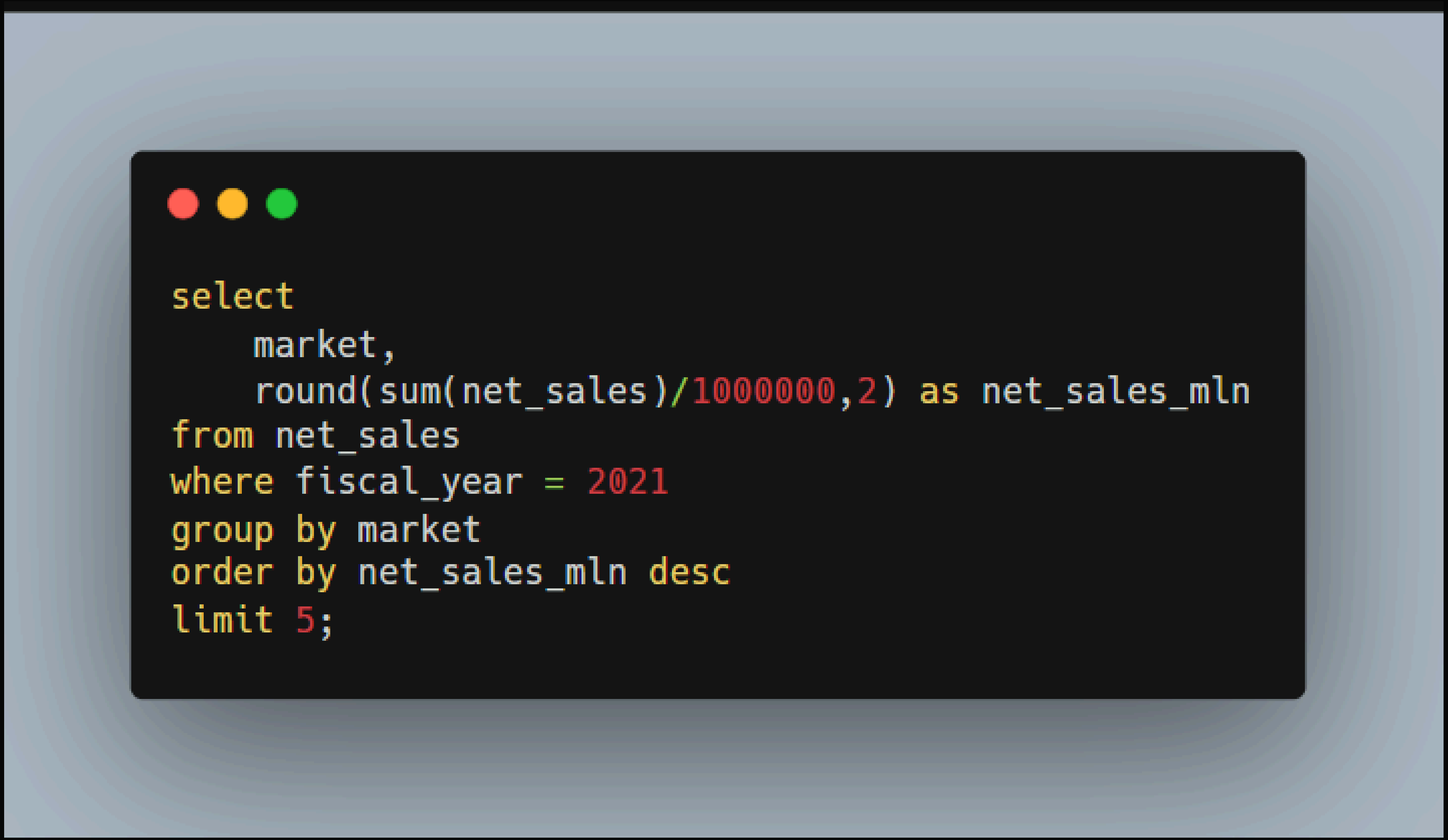
```sql
CREATE VIEW sale_postinv_discount AS
select
    *,
    (1 -  pre_invoice_discount_pct) * gross_price_total as net_invoice_sales,
    (po.discounts_pct + po.other_deductions_pct) as post_invoice_discount_pct
from sale_preinv_discount s
    join fact_post_invoice_deductions po
        on s.date = po.date and
        s.product_code = po.product_code and
        s.customer_code = po.customer_code;
```

Net sales View:



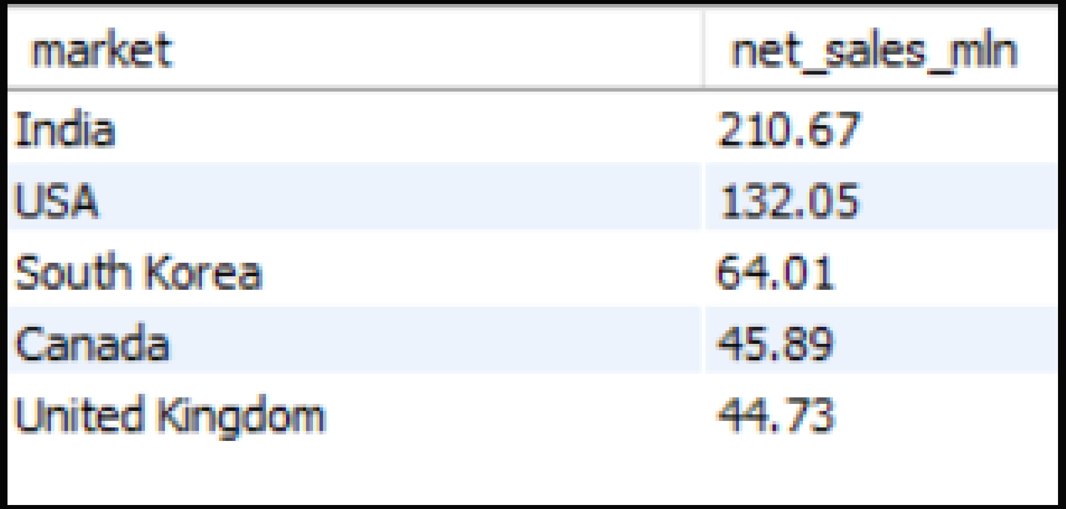
```
CREATE VIEW `net_sales` AS
SELECT
        *,
    net_invoice_sales*(1-post_invoice_discount_pct) as net_sales
FROM gdb041.sales_postinv_discount;
```

Top 5 markets for the fiscal year 2021 by Net sales



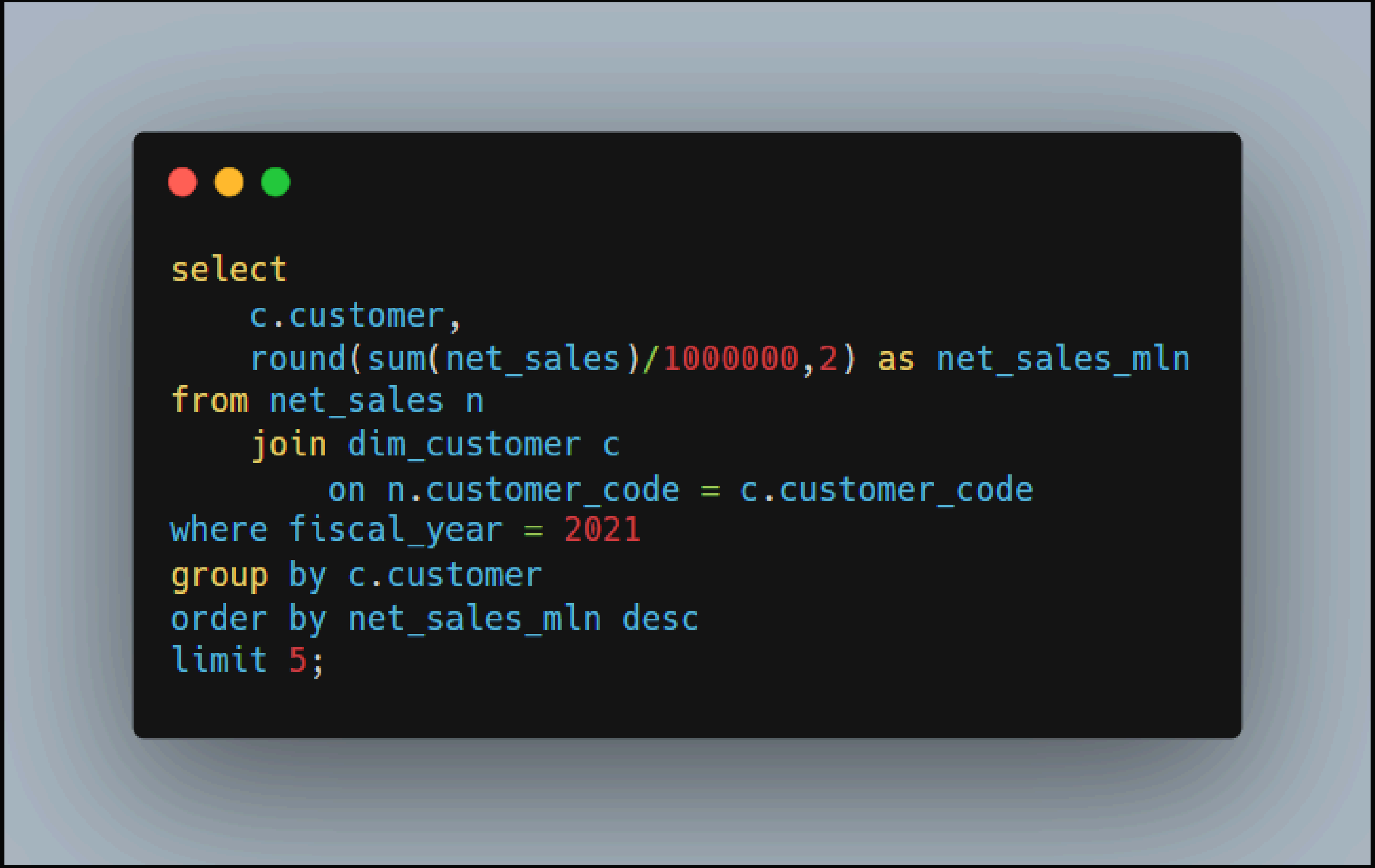
```
select
    market,
    round(sum(net_sales)/1000000,2) as net_sales_mln
from net_sales
where fiscal_year = 2021
group by market
order by net_sales_mln desc
limit 5;
```

Output:



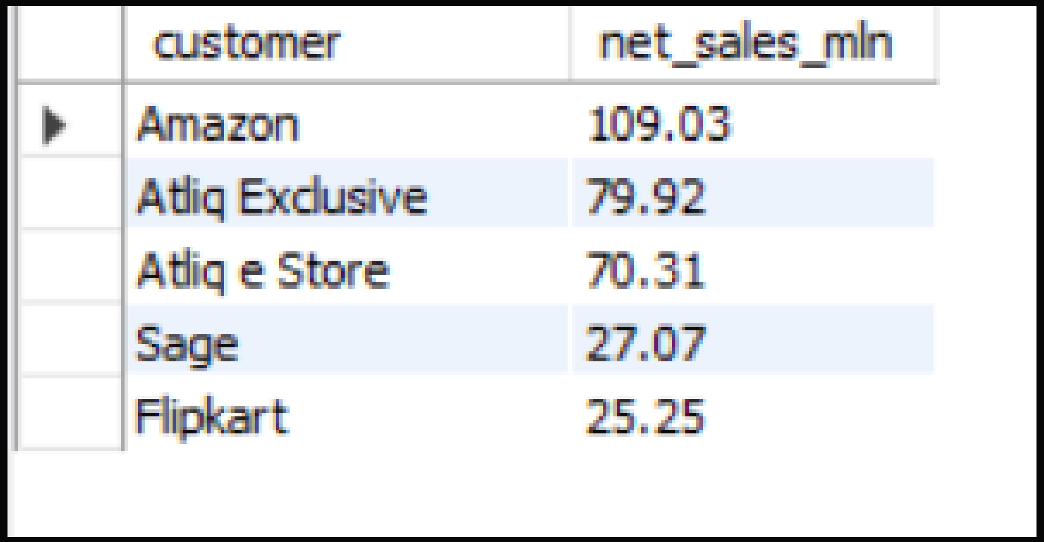
market	net_sales_mln
India	210.67
USA	132.05
South Korea	64.01
Canada	45.89
United Kingdom	44.73

Top 5 customers for the fiscal year 2021 by Net sales:



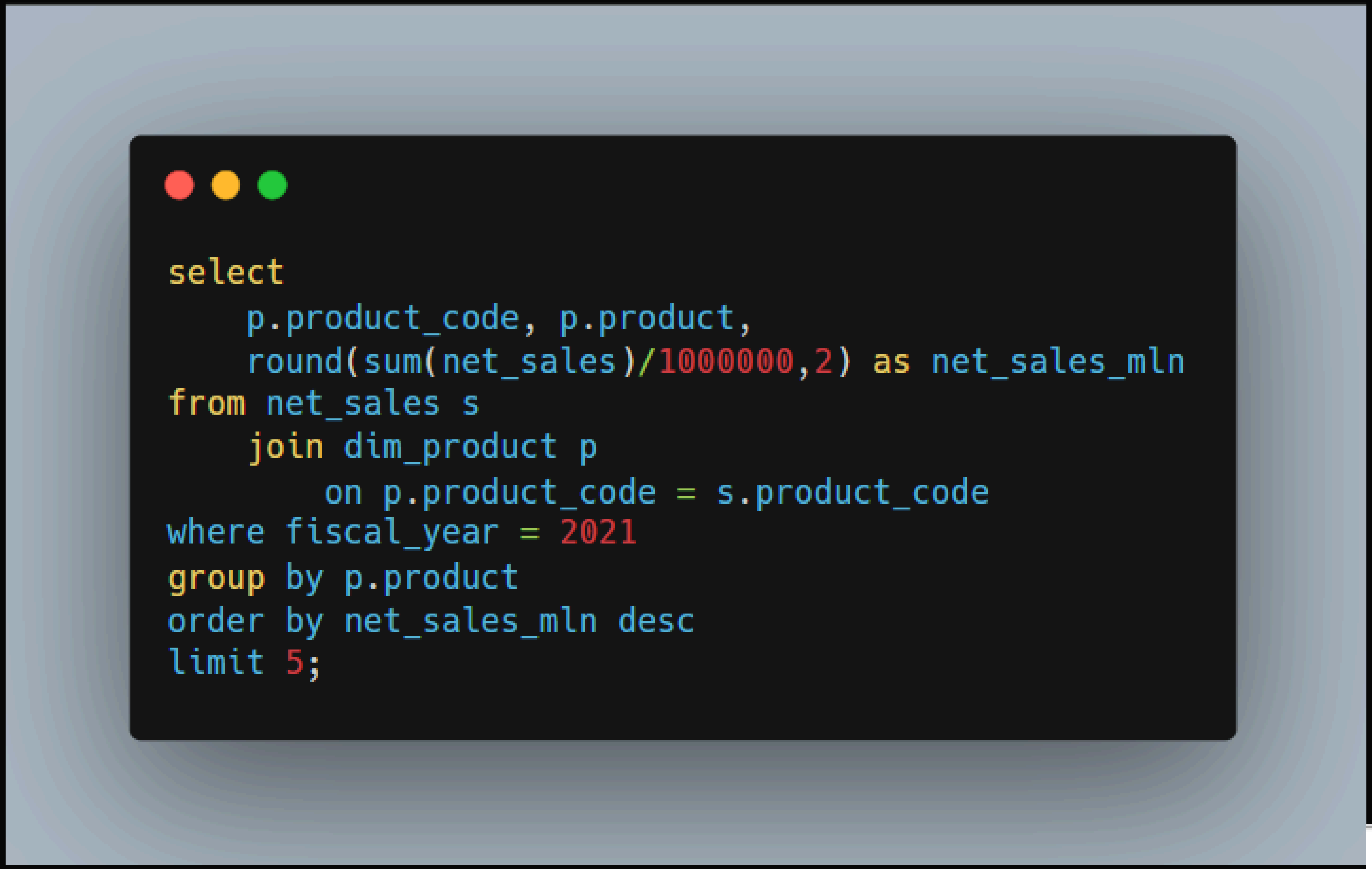
```
select
    c.customer,
    round(sum(net_sales)/1000000,2) as net_sales_mln
from net_sales n
    join dim_customer c
        on n.customer_code = c.customer_code
where fiscal_year = 2021
group by c.customer
order by net_sales_mln desc
limit 5;
```

Output:



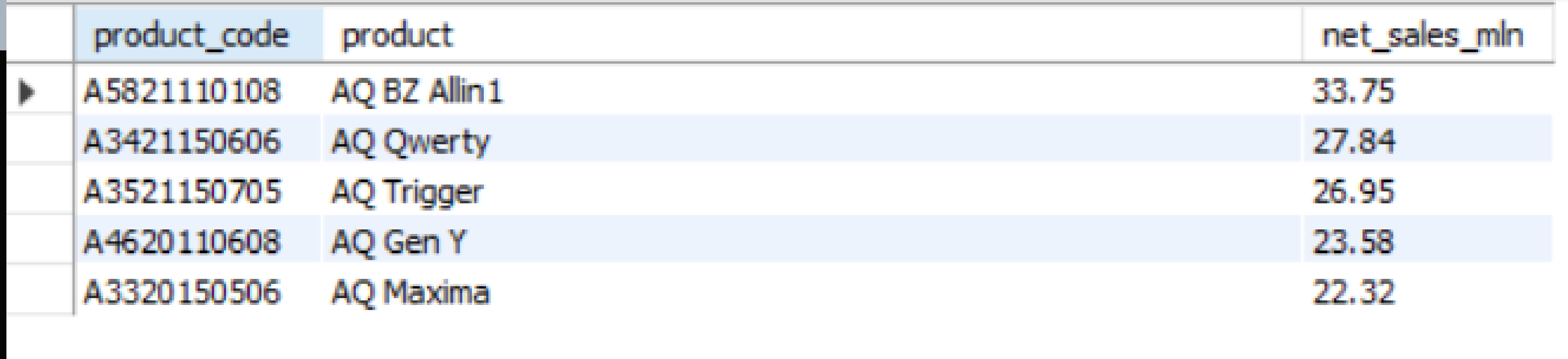
customer	net_sales_mln
Amazon	109.03
Atliq Exclusive	79.92
Atliq e Store	70.31
Sage	27.07
Flipkart	25.25

Top 5 products for the fiscal year 2021 by Net sales:



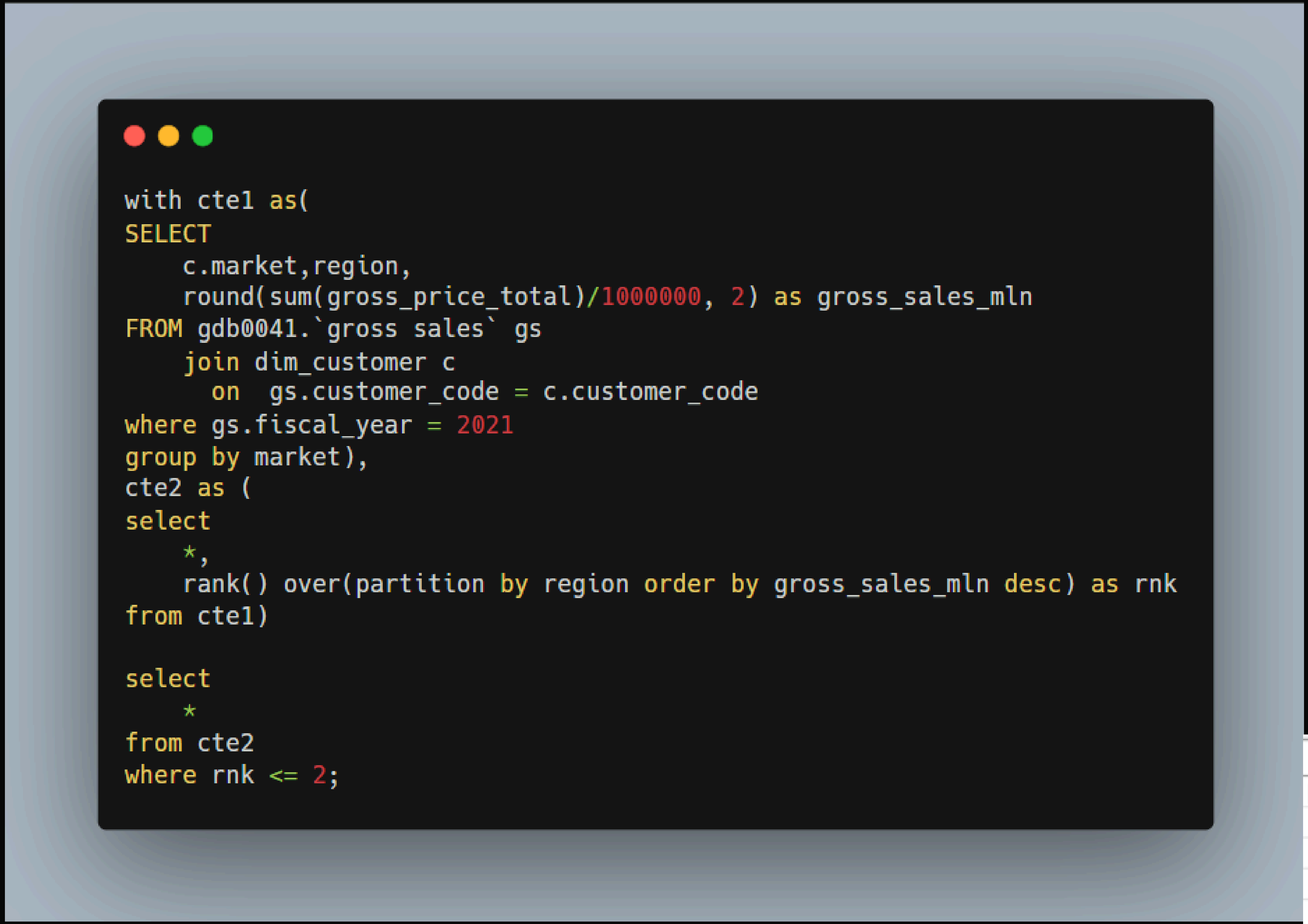
```sql
select
    p.product_code, p.product,
    round(sum(net_sales)/1000000,2) as net_sales_mln
from net_sales s
    join dim_product p
        on p.product_code = s.product_code
where fiscal_year = 2021
group by p.product
order by net_sales_mln desc
limit 5;
```

Output:



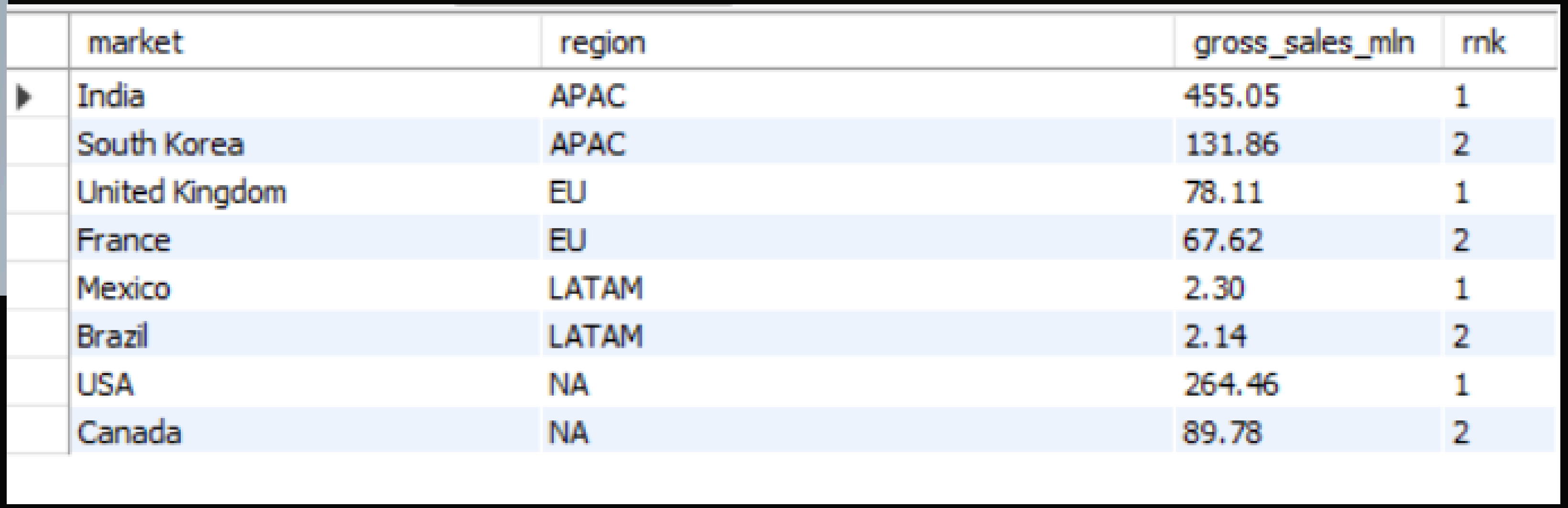
product_code	product	net_sales_mln
A5821110108	AQ BZ Allin1	33.75
A3421150606	AQ Qwerty	27.84
A3521150705	AQ Trigger	26.95
A4620110608	AQ Gen Y	23.58
A3320150506	AQ Maxima	22.32

Retrieve the top 2 markets in every region by their gross sales amount in FY=2021



```
with cte1 as(
SELECT
    c.market,region,
    round(sum(gross_price_total)/1000000, 2) as gross_sales_mln
FROM gdb0041.`gross sales` gs
    join dim_customer c
      on  gs.customer_code = c.customer_code
where gs.fiscal_year = 2021
group by market),
cte2 as (
select
    *,
    rank() over(partition by region order by gross_sales_mln desc) as rnk
from cte1)

select
    *
from cte2
where rnk <= 2;
```

Output:



market	region	gross_sales_mln	rnk
India	APAC	455.05	1
South Korea	APAC	131.86	2
United Kingdom	EU	78.11	1
France	EU	67.62	2
Mexico	LATAM	2.30	1
Brazil	LATAM	2.14	2
USA	NA	264.46	1
Canada	NA	89.78	2

Thank you!