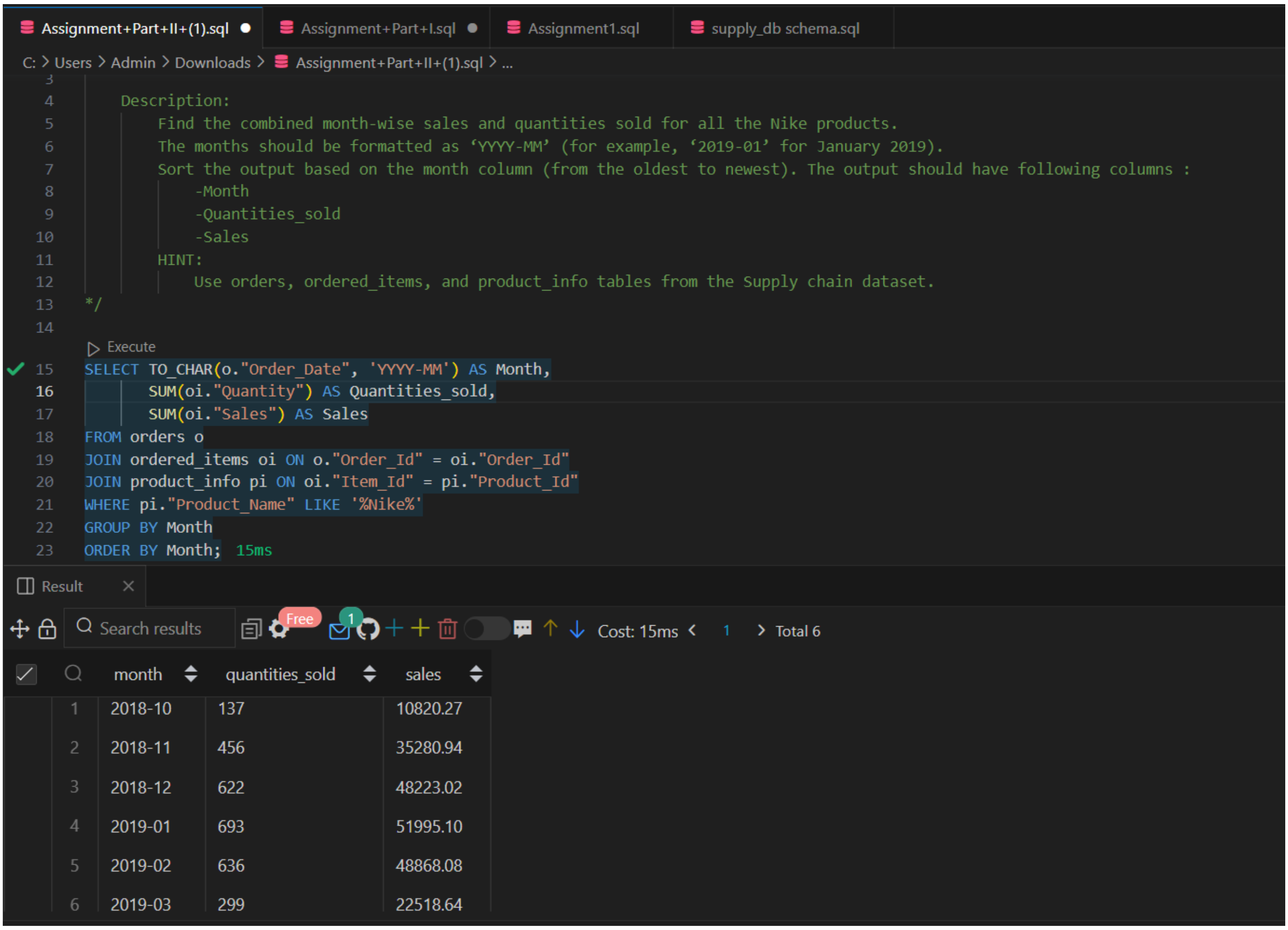
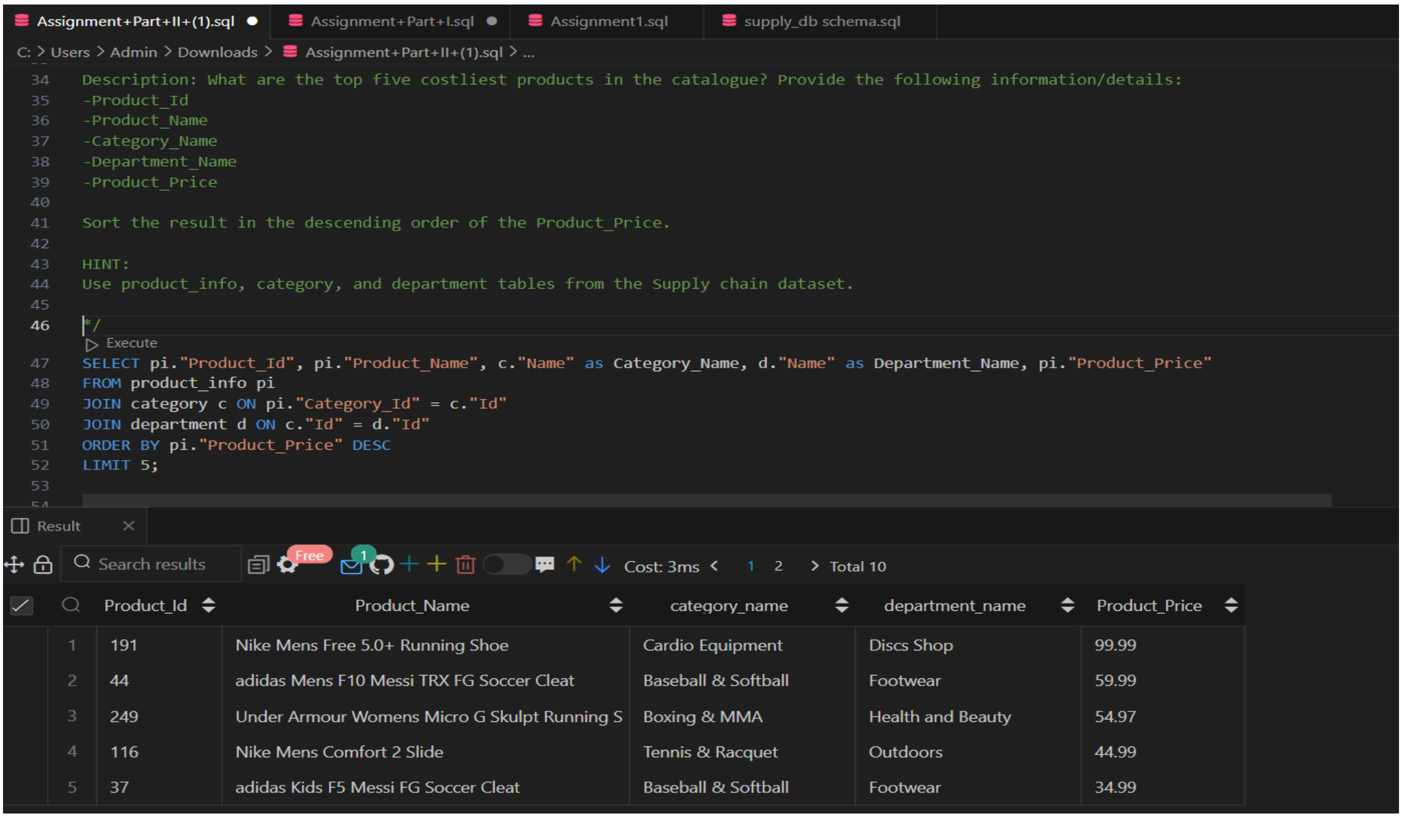
Solution 1:



```
    Description:
        Find the combined month-wise sales and quantities sold for all the Nike products.
        The months should be formatted as ‘YYYY-MM’ (for example, ‘2019-01’ for January 2019).
        Sort the output based on the month column (from the oldest to newest). The output should have following columns :
            -Month
            -Quantities_sold
            -Sales
        HINT:
            Use orders, ordered_items, and product_info tables from the Supply chain dataset.
*/

SELECT TO_CHAR(o."Order_Date", 'YYYY-MM') AS Month,
       SUM(oi."Quantity") AS Quantities_sold,
       SUM(oi."Sales") AS Sales
FROM orders o
JOIN ordered_items oi ON o."Order_Id" = oi."Order_Id"
JOIN product_info pi ON oi."Item_Id" = pi."Product_Id"
WHERE pi."Product_Name" LIKE '%Nike%'
GROUP BY Month
ORDER BY Month; 15ms
```

	month	quantities_sold	sales
1	2018-10	137	10820.27
2	2018-11	456	35280.94
3	2018-12	622	48223.02
4	2019-01	693	51995.10
5	2019-02	636	48868.08
6	2019-03	299	22518.64

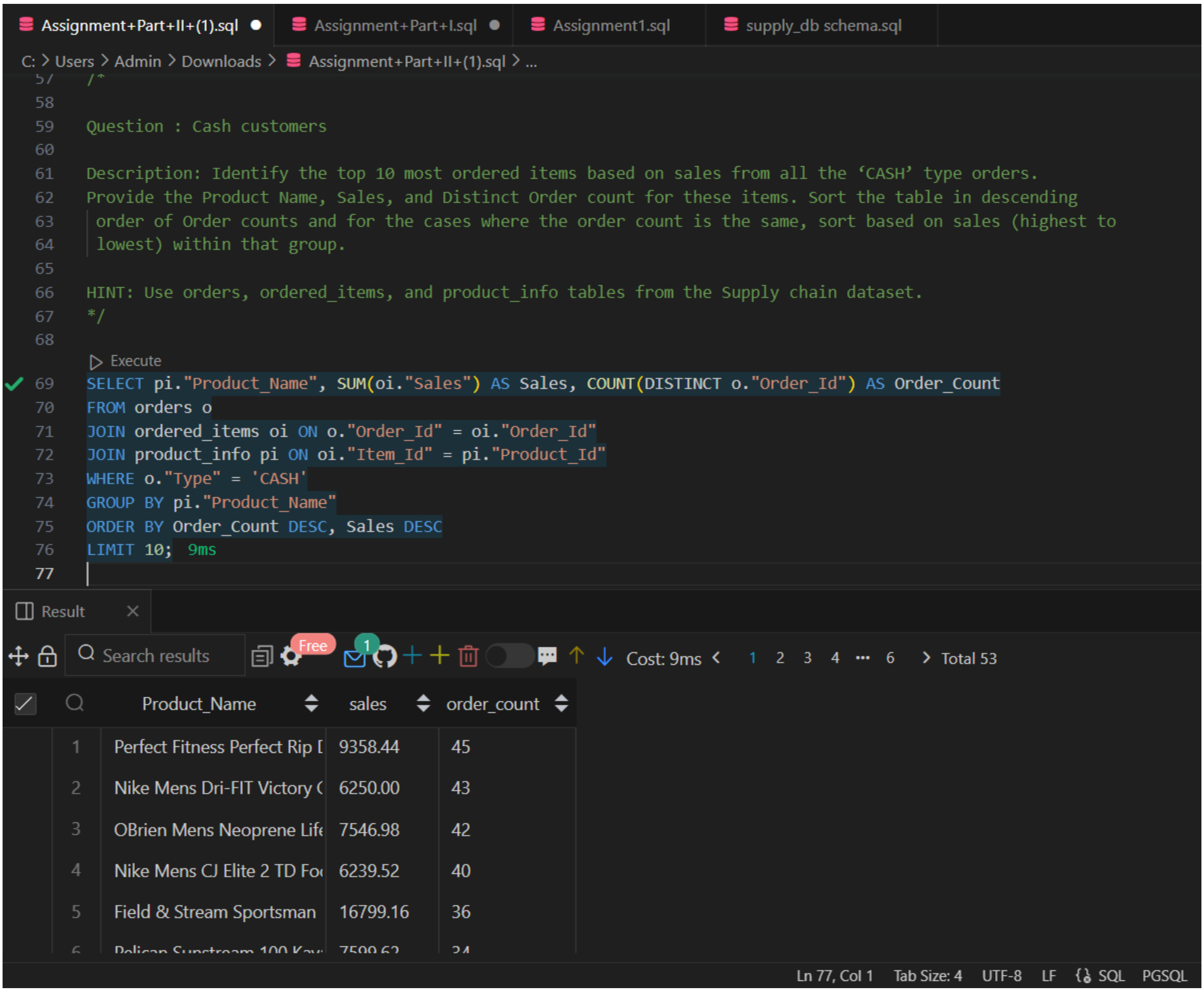
Solution 2 :



```
Description: What are the top five costliest products in the catalogue? Provide the following information/details:
-Product_Id
-Product_Name
-Category_Name
-Department_Name
-Product_Price

Sort the result in the descending order of the Product_Price.

HINT:
Use product_info, category, and department tables from the Supply chain dataset.

*/
SELECT pi."Product_Id", pi."Product_Name", c."Name" as Category_Name, d."Name" as Department_Name, pi."Product_Price"
FROM product_info pi
JOIN category c ON pi."Category_Id" = c."Id"
JOIN department d ON c."Id" = d."Id"
ORDER BY pi."Product_Price" DESC
LIMIT 5;
```

	Product_Id	Product_Name	category_name	department_name	Product_Price
1	191	Nike Mens Free 5.0+ Running Shoe	Cardio Equipment	Discs Shop	99.99
2	44	adidas Mens F10 Messi TRX FG Soccer Cleat	Baseball & Softball	Footwear	59.99
3	249	Under Armour Womens Micro G Skulpt Running S	Boxing & MMA	Health and Beauty	54.97
4	116	Nike Mens Comfort 2 Slide	Tennis & Racquet	Outdoors	44.99
5	37	adidas Kids F5 Messi FG Soccer Cleat	Baseball & Softball	Footwear	34.99

Solution 3 :

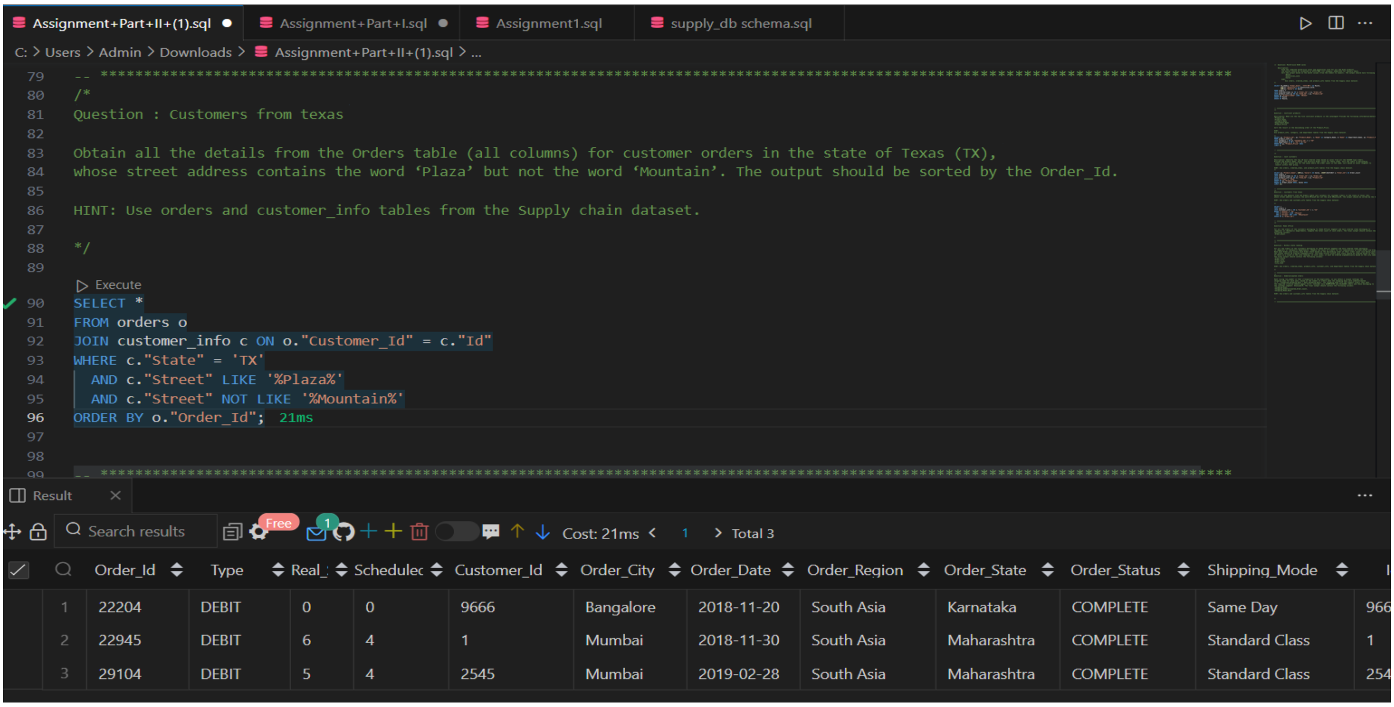


Assignment+Part+II+(1).sql
Assignment+Part+I.sql
Assignment1.sql
supply_db schema.sql
C: > Users > Admin > Downloads > Assignment+Part+II+(1).sql > ...
Execute

```sql
/*

Question : Cash customers

Description: Identify the top 10 most ordered items based on sales from all the ‘CASH’ type orders.
Provide the Product Name, Sales, and Distinct Order count for these items. Sort the table in descending
 order of Order counts and for the cases where the order count is the same, sort based on sales (highest to
 lowest) within that group.

HINT: Use orders, ordered_items, and product_info tables from the Supply chain dataset.
*/

SELECT pi."Product_Name", SUM(oi."Sales") AS Sales, COUNT(DISTINCT o."Order_Id") AS Order_Count
FROM orders o
JOIN ordered_items oi ON o."Order_Id" = oi."Order_Id"
JOIN product_info pi ON oi."Item_Id" = pi."Product_Id"
WHERE o."Type" = 'CASH'
GROUP BY pi."Product_Name"
ORDER BY Order_Count DESC, Sales DESC
LIMIT 10;  9ms
```

	Product_Name	sales	order_count
1	Perfect Fitness Perfect Rip [	9358.44	45
2	Nike Mens Dri-FIT Victory (	6250.00	43
3	OBrien Mens Neoprene Life	7546.98	42
4	Nike Mens CJ Elite 2 TD Foo	6239.52	40
5	Field & Stream Sportsman	16799.16	36
6	Pelican Sunstream 100 Kay	7599.62	34

Solution 4:

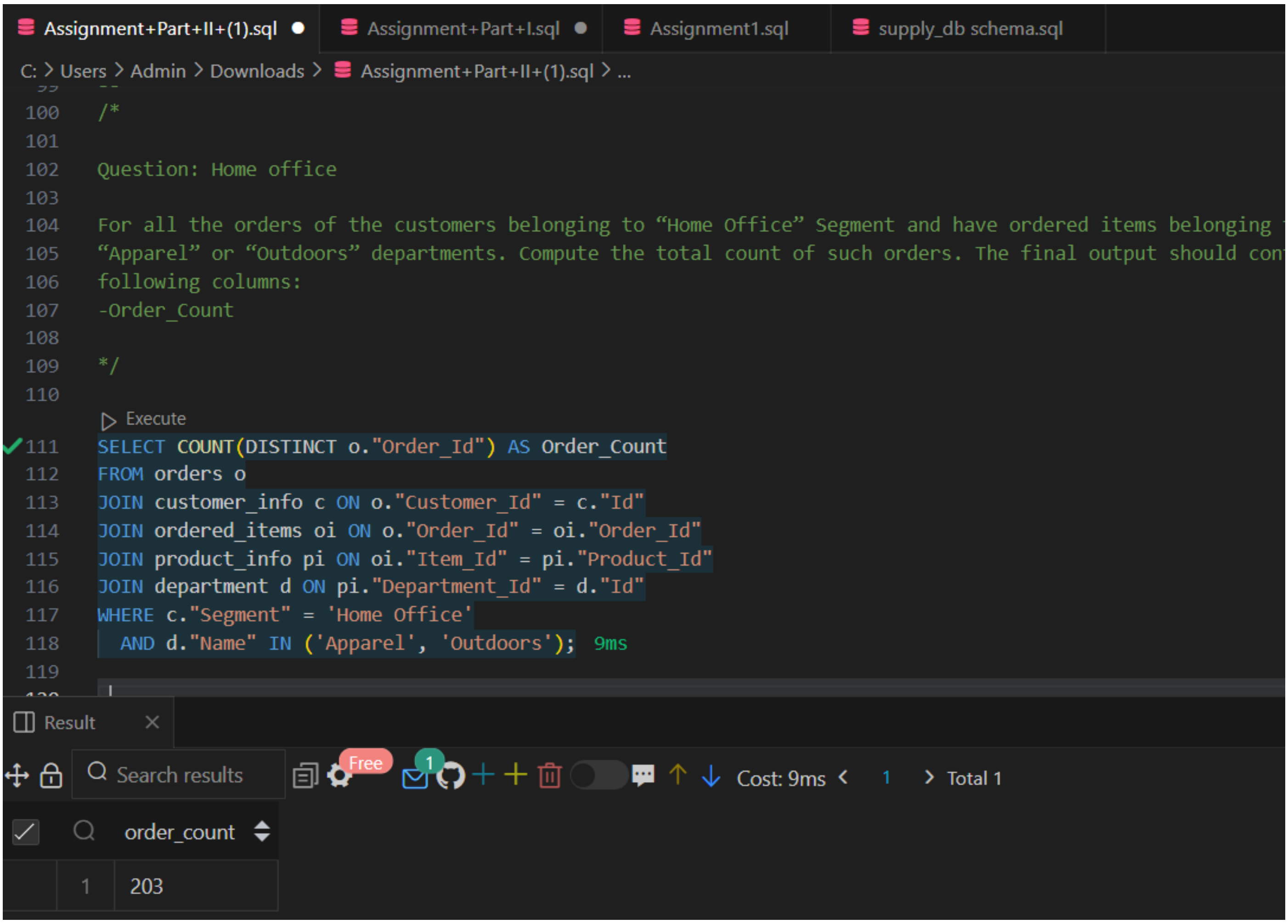


Assignment+Part+II+(1).sql
Assignment+Part+I.sql
Assignment1.sql
supply_db schema.sql
C: > Users > Admin > Downloads > Assignment+Part+II+(1).sql > ...
Execute

```sql
-- ********************************************************************************************************
/*
Question : Customers from texas

Obtain all the details from the Orders table (all columns) for customer orders in the state of Texas (TX),
whose street address contains the word ‘Plaza’ but not the word ‘Mountain’. The output should be sorted by the Order_Id.

HINT: Use orders and customer_info tables from the Supply chain dataset.

*/

SELECT *
FROM orders o
JOIN customer_info c ON o."Customer_Id" = c."Id"
WHERE c."State" = 'TX'
  AND c."Street" LIKE '%Plaza%'
  AND c."Street" NOT LIKE '%Mountain%'
ORDER BY o."Order_Id";  21ms


-- ********************************************************************************************************
```

	Order_Id	Type	Real_	Schedulec	Customer_Id	Order_City	Order_Date	Order_Region	Order_State	Order_Status	Shipping_Mode	I
1	22204	DEBIT	0	0	9666	Bangalore	2018-11-20	South Asia	Karnataka	COMPLETE	Same Day	966
2	22945	DEBIT	6	4	1	Mumbai	2018-11-30	South Asia	Maharashtra	COMPLETE	Standard Class	1
3	29104	DEBIT	5	4	2545	Mumbai	2019-02-28	South Asia	Maharashtra	COMPLETE	Standard Class	254

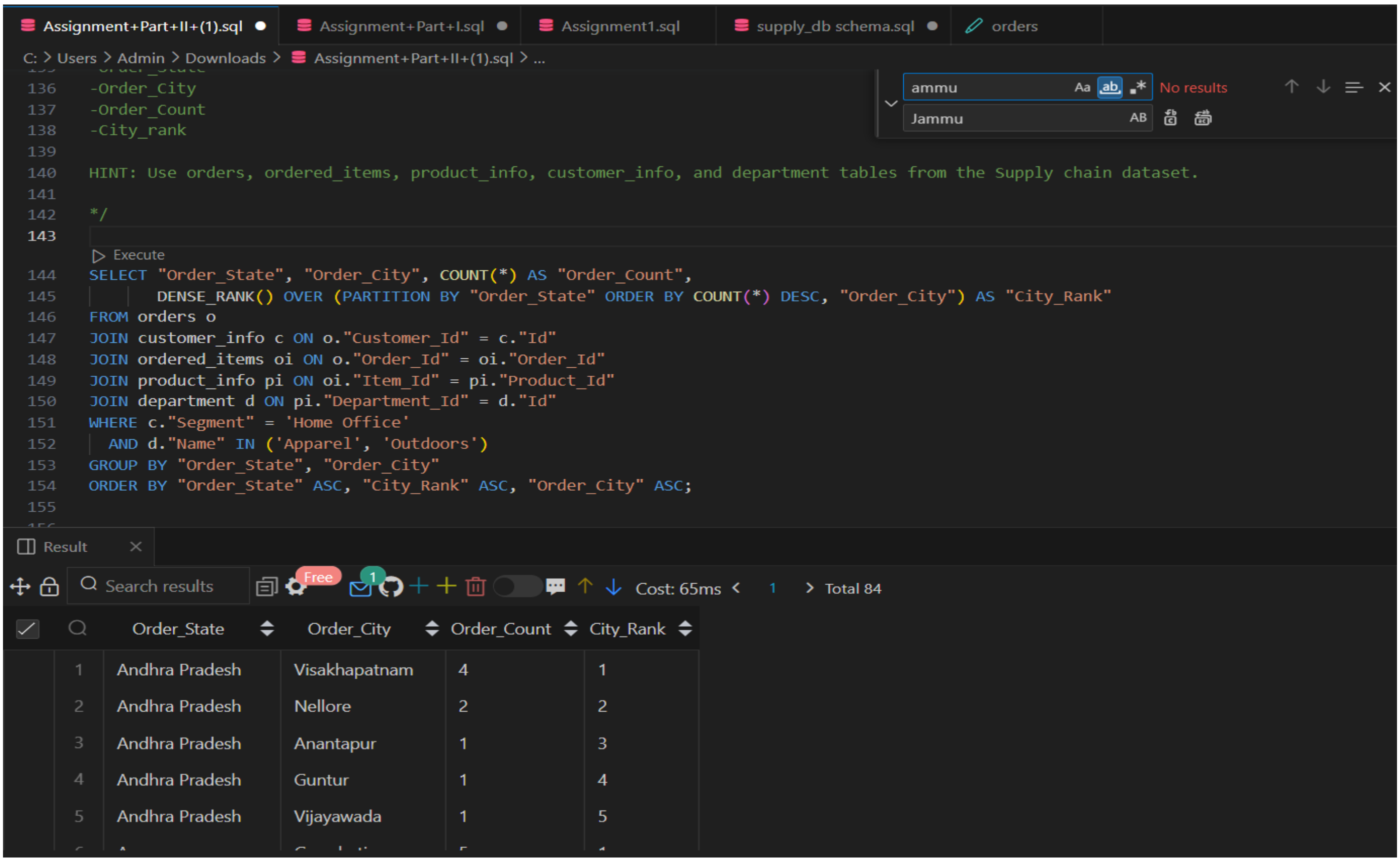
Solution 5:



```sql
/*

Question: Home office

For all the orders of the customers belonging to “Home Office” Segment and have ordered items belonging
“Apparel” or “Outdoors” departments. Compute the total count of such orders. The final output should con
following columns:
-Order_Count

*/

SELECT COUNT(DISTINCT o."Order_Id") AS Order_Count
FROM orders o
JOIN customer_info c ON o."Customer_Id" = c."Id"
JOIN ordered_items oi ON o."Order_Id" = oi."Order_Id"
JOIN product_info pi ON oi."Item_Id" = pi."Product_Id"
JOIN department d ON pi."Department_Id" = d."Id"
WHERE c."Segment" = 'Home Office'
  AND d."Name" IN ('Apparel', 'Outdoors'); 9ms
```

	order_count
1	203

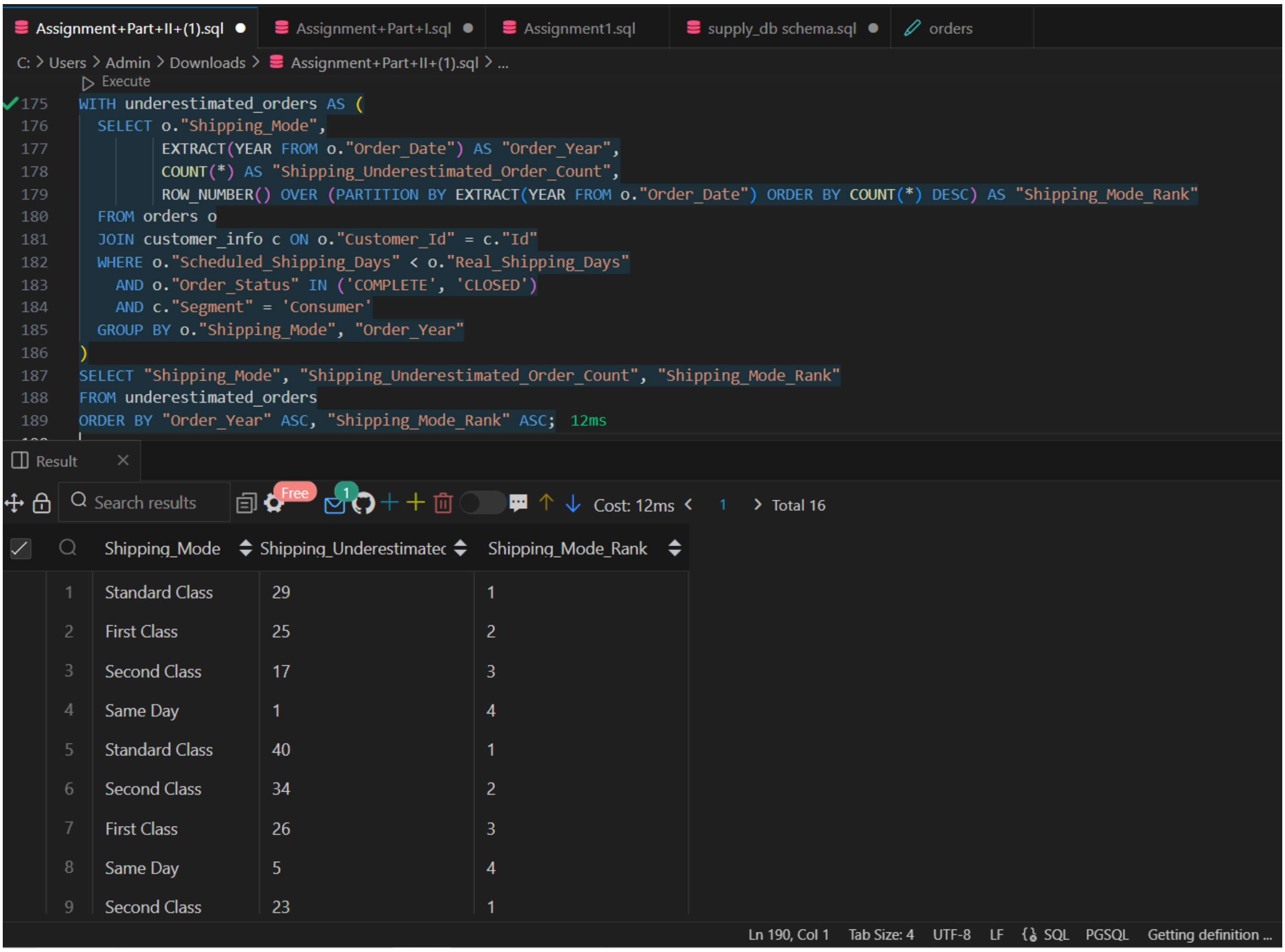
Solution 6:



```sql
-Order_City
-Order_Count
-City_rank

HINT: Use orders, ordered_items, product_info, customer_info, and department tables from the Supply chain dataset.

*/

SELECT "Order_State", "Order_City", COUNT(*) AS "Order_Count",
       DENSE_RANK() OVER (PARTITION BY "Order_State" ORDER BY COUNT(*) DESC, "Order_City") AS "City_Rank"
FROM orders o
JOIN customer_info c ON o."Customer_Id" = c."Id"
JOIN ordered_items oi ON o."Order_Id" = oi."Order_Id"
JOIN product_info pi ON oi."Item_Id" = pi."Product_Id"
JOIN department d ON pi."Department_Id" = d."Id"
WHERE c."Segment" = 'Home Office'
  AND d."Name" IN ('Apparel', 'Outdoors')
GROUP BY "Order_State", "Order_City"
ORDER BY "Order_State" ASC, "City_Rank" ASC, "Order_City" ASC;
```

	Order_State	Order_City	Order_Count	City_Rank
1	Andhra Pradesh	Visakhapatnam	4	1
2	Andhra Pradesh	Nellore	2	2
3	Andhra Pradesh	Anantapur	1	3
4	Andhra Pradesh	Guntur	1	4
5	Andhra Pradesh	Vijayawada	1	5

Solution 7:



```sql
WITH underestimated_orders AS (
  SELECT o."Shipping_Mode",
         EXTRACT(YEAR FROM o."Order_Date") AS "Order_Year",
         COUNT(*) AS "Shipping_Underestimated_Order_Count",
         ROW_NUMBER() OVER (PARTITION BY EXTRACT(YEAR FROM o."Order_Date") ORDER BY COUNT(*) DESC) AS "Shipping_Mode_Rank"
  FROM orders o
  JOIN customer_info c ON o."Customer_Id" = c."Id"
  WHERE o."Scheduled_Shipping_Days" < o."Real_Shipping_Days"
    AND o."Order_Status" IN ('COMPLETE', 'CLOSED')
    AND c."Segment" = 'Consumer'
  GROUP BY o."Shipping_Mode", "Order_Year"
)
SELECT "Shipping_Mode", "Shipping_Underestimated_Order_Count", "Shipping_Mode_Rank"
FROM underestimated_orders
ORDER BY "Order_Year" ASC, "Shipping_Mode_Rank" ASC;  12ms
```

	Shipping_Mode	Shipping_Underestimatec	Shipping_Mode_Rank
1	Standard Class	29	1
2	First Class	25	2
3	Second Class	17	3
4	Same Day	1	4
5	Standard Class	40	1
6	Second Class	34	2
7	First Class	26	3
8	Same Day	5	4
9	Second Class	23	1