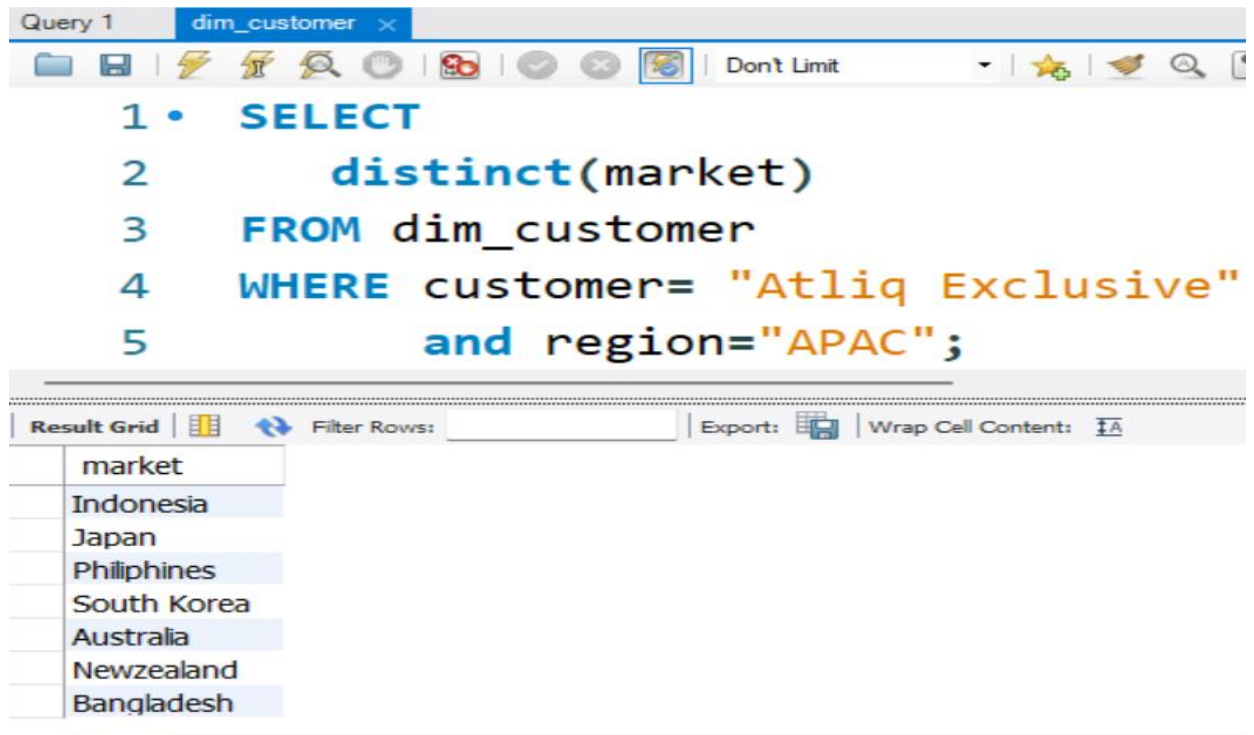


Consumer good Insights

Requests:

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



```
Query 1 dim_customer x
1 • SELECT
2     distinct(market)
3 FROM dim_customer
4 WHERE customer= "Atliq Exclusive"
5     and region="APAC";
```

market
Indonesia
Japan
Philippines
South Korea
Australia
Newzealand
Bangladesh

2. 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

dim_customer fact_gross_price

```

1 • WITH cte1 as
2   (SELECT
3     count(distinct product_code) as unique_product_2020
4   FROM fact_gross_price
5     where fiscal_year=2020),
6
7   cte2 as
8   (SELECT
9     count(distinct product_code) as unique_product_2021
10  FROM fact_gross_price
11    where fiscal_year=2021)
12
13  SELECT unique_product_2020,
14         unique_product_2021,
15         round(((unique_product_2021-unique_product_2020)/unique_product_2020) *100,2) as percentage_chg
16  FROM cte1,cte2;
17

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	unique_product_2020	unique_product_2021	percentage_chg
▶	245	334	36.33

3. 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

dim_customer fact_gross_price dim_product dim_product

```

1 • SELECT segment,
2         count(distinct product_code) as product_count
3 FROM dim_product
4 GROUP BY segment
5 ORDER BY product_count desc;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

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

4. 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

```
with cte1 as (  
SELECT  
    segment,  
    count(distinct p.product_code) as product_count_2020  
FROM dim_product p  
JOIN fact_gross_price g  
    on p.product_code=g.product_code  
    where fiscal_year=2020  
GROUP BY segment),  
  
cte2 as  
(SELECT  
    segment,  
    count(distinct p.product_code) as product_count_2021  
FROM dim_product p  
JOIN fact_gross_price g  
    on p.product_code=g.product_code  
    where fiscal_year=2021  
GROUP BY segment)  
  
SELECT  
    cte1.segment, product_count_2020,  
    product_count_2021,  
    (product_count_2021-product_count_2020) as Difference  
FROM cte1  
JOIN cte2  
    on cte1.segment=cte2.segment  
ORDER BY Difference desc;
```

Result Grid Filter Rows: Export: Wrap Cell Content:				
	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

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

product_code

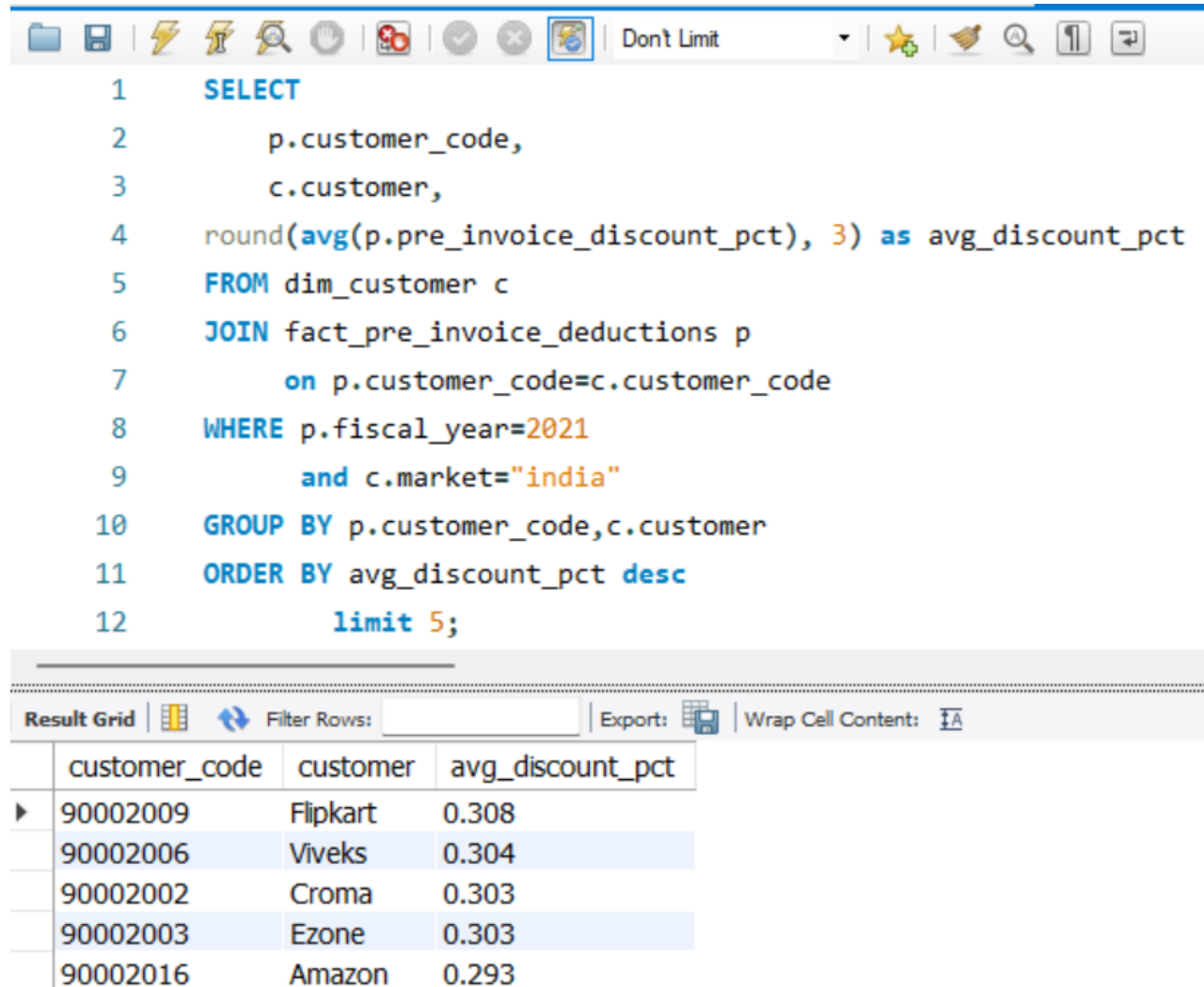
product

manufacturing_cost

dim_customer fact_gross_price dim_product fact_gross_price dim_product ×				
Don't Limit				
<pre> 1 • SELECT 2 p.product, 3 p.product_code, 4 m.manufacturing_cost AS max_min_manufacturing_cost 5 FROM dim_product p 6 JOIN fact_manufacturing_cost m 7 ON m.product_code = p.product_code 8 WHERE m.manufacturing_cost = (SELECT MAX(manufacturing_cost) 9 FROM fact_manufacturing_cost) 10 UNION 11 SELECT 12 p.product, 13 p.product_code, 14 m.manufacturing_cost AS max_min_manufacturing_cost 15 FROM dim_product p 16 JOIN fact_manufacturing_cost m ON m.product_code = p.product_code 17 WHERE m.manufacturing_cost = (SELECT MIN(manufacturing_cost) 18 FROM fact_manufacturing_cost); </pre>				
Result Grid Filter Rows: Export: Wrap Cell Content:				
	product	product_code	max_min_manufacturing_cost	
▶	AQ HOME Allin1 Gen 2	A6121110208	263.4207	
	AQ Master wired x1 Ms	A2118150101	0.8654	

6. 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



The screenshot shows a SQL query editor with a toolbar at the top. The query is as follows:

```
1  SELECT
2      p.customer_code,
3      c.customer,
4      round(avg(p.pre_invoice_discount_pct), 3) as avg_discount_pct
5  FROM dim_customer c
6  JOIN fact_pre_invoice_deductions p
7      on p.customer_code=c.customer_code
8  WHERE p.fiscal_year=2021
9         and c.market="india"
10 GROUP BY p.customer_code,c.customer
11 ORDER BY avg_discount_pct desc
12      limit 5;
```

Below the query editor is a 'Result Grid' section. It includes a 'Filter Rows' input field, an 'Export' button, and a 'Wrap Cell Content' checkbox. The results are displayed in a table with the following data:

	customer_code	customer	avg_discount_pct
▶	90002009	Flipkart	0.308
	90002006	Viveks	0.304
	90002002	Croma	0.303
	90002003	Ezone	0.303
	90002016	Amazon	0.293

7. 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

dim_customer	fact_gross_price	dim_product	fact_gross_price	dim_product	fact_manufacturing_cost
Don't Limit					
<pre>1 • SELECT 2 s.date, 3 monthname(s.date) as month_name, 4 year(s.date) as year, 5 round(sum(s.sold_quantity*g.gross_price), 2) as gross_sales_amt 6 FROM fact_sales_monthly s 7 JOIN dim_customer c 8 on c.customer_code=s.customer_code 9 JOIN fact_gross_price g 10 on g.product_code=s.product_code 11 WHERE customer="Atliq Exclusive" 12 GROUP BY month_name, year, date</pre>					
Result Grid Filter Rows: Export: Wrap Cell Content:					
date	month_name	year	gross_sales_amt		
2017-12-01	December	2017	2390015.56		
2017-11-01	November	2017	3766114.43		
2017-10-01	October	2017	2462780.55		
2017-09-01	September	2017	2347703.88		
2018-04-01	April	2018	1392024.51		
2018-08-01	August	2018	1498728.56		

8. 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

dim_customer fact_gross_price dim_product fact_gross_price dim_product fact_manufacturing_cost SQL File 4* SQL File 5* x

1 • SELECT

2 CASE when date between "2019-09-01" and "2019-11-30" then "Q1"

3 when date between "2019-12-01" and "2020-02-29" then "Q2"

4 when date between "2020-03-01" and "2020-5-30" then "Q3"

5 when date between "2020-6-01" and "2020-08-30" then "Q4"

6 end as quarter,

7 sum(sold_quantity) as total_sold_quantity

8 FROM fact_sales_monthly

9 WHERE fiscal_year=2020

10 GROUP BY quarter

11 ORDER BY total_sold_quantity desc;

Result Grid Filter Rows: Export: Wrap Cell Content: FA

	quarter	total_sold_quantity
▶	Q1	7005619
	Q2	6649642
	Q4	5042541
	Q3	2075087

9. 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

dim_customer fact_gross_price dim_product fact_gross_price dim_product fact_manu

1 ● WITH channel_sales AS (
 2 SELECT
 3 c.channel,
 4 SUM(s.sold_quantity * g.gross_price) AS total_sales
 5 FROM fact_sales_monthly s
 6 JOIN dim_customer c
 7 ON c.customer_code = s.customer_code
 8 JOIN fact_gross_price g
 9 ON g.product_code = s.product_code
 10 WHERE s.fiscal_year = 2021
 11 GROUP BY c.channel)
 12
 13 SELECT
 14 channel,
 15 concat(ROUND(total_sales / 1000000, 2), " ", "M") AS gross_sales_mln,
 16 ROUND((total_sales / SUM(total_sales) OVER ()) * 100, 2) AS percentage
 17 FROM channel_sales
 18 ORDER BY gross_sales_mln ASC;

Result Grid Filter Rows: Export: Wrap Cell Content: [IA](#)

	channel	gross_sales_mln	percentage
▶	Retailer	3708.46 M	73.21
	Distributor	572.86 M	11.31
	Direct	784.14 M	15.48

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

The screenshot shows a SQL IDE interface with a query editor and a result grid. The query editor contains a SQL query that uses a Common Table Expression (CTE) to calculate the total sold quantity and rank order for products. The result grid displays the output of the query, showing columns for division, product_code, product, total_sold_quantity, and rank_order. The results are sorted by rank_order, with the top three products for each division being displayed.

```
1 with CTE1 as (  
2     select  
3         p.division,  
4         p.product_code,  
5         p.product,  
6         sum(s.sold_quantity) as total_sold_quantity,  
7         RANK() OVER (PARTITION BY division ORDER BY SUM(sold_quantity) DESC) AS rank_order  
8     from dim_product p  
9     join fact_sales_monthly s  
10        on s.product_code=p.product_code  
11    where s.fiscal_year=2021  
12    group by p.division, p.product_code,p.product)  
13  
14     SELECT  
15         *  
16     FROM CTE1  
17     WHERE rank_order <= 3  
18     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