Ad-hoc Requests

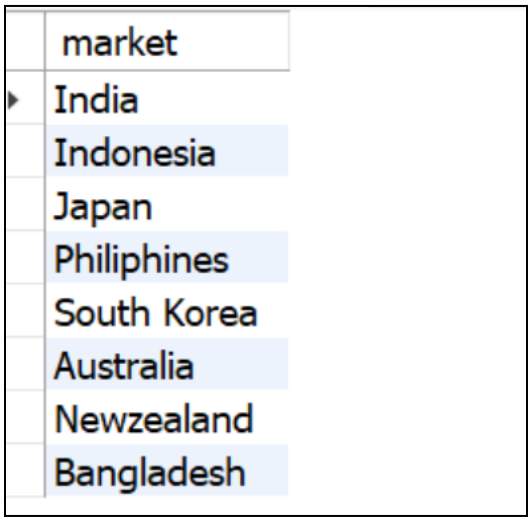
Request 1 -

Provide the list of markets in which customer "Atliq Exclusive" operates its business in the APAC region. 2.

SQL Code -

```
SELECT DISTINCT(market)
FROM gdb023.dim_customer
WHERE customer = "Atliq Exclusive" AND region="APAC"
```

Output 1 -



market
India
Indonesia
Japan
Philiphines
South Korea
Australia
Newzealand
Bangladesh

P.T.O

Request 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

SQL Code -

```
with cte1 as (
SELECT
        (SELECT COUNT(DISTINCT product_code) FROM fact_sales_monthly where
fiscal_year="2020") AS unique_products_2020,
   (SELECT COUNT(DISTINCT product_code) FROM fact_sales_monthly where
fiscal_year="2021") AS unique_products_2021
        FROM gdb023.fact_sales_monthly
   LIMIT 1)

select
unique_products_2020,unique_products_2021,
ROUND((unique_products_2021/unique_products_2020-1)*100,2) as percentage_chg
from cte1
```

Output 2 -

	unique_products_2020	unique_products_2021	percentage_chg
▸	245	334	36.33

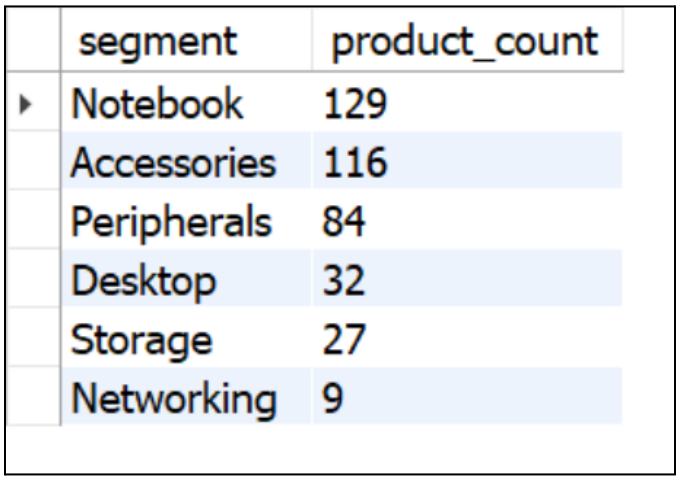
Request 3 -

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

SQL Code -

```
SELECT
   segment, COUNT(DISTINCT product_code) AS product_count
FROM
   gdb023.dim_product
GROUP BY segment
ORDER BY product_count DESC
```

Output 3 -



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

Request 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

SQL Code -

```
with unique_product as (
SELECT
        segment,
        count(distinct case when fiscal_year = 2020 then product_code END) as
product_count_2020,
        count(distinct case when fiscal_year = 2021 then product_code END) as
product_count_2021
FROM
        (select
        d.*,f.fiscal_year
   from dim_product d
   Join fact_sales_monthly f
   on d.product_code=f.product_code) as new_table
GROUP by segment)

select
segment, product_count_2020, product_count_2021,
(product_count_2021- product_count_2020) as difference
from unique_product
ORDER by difference DESC
```

Output 4 -

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

P.T.O

Request 5 -

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

SQL Code -

```
SELECT
    *
FROM
    (SELECT
        p.product_code, p.product, manufacturing_cost
    FROM
        fact_manufacturing_cost m
    JOIN dim_product p ON m.product_code = p.product_code) AS manu_product
WHERE
    manufacturing_cost = (SELECT
            MAX(manufacturing_cost)
        FROM
            fact_manufacturing_cost)
        OR manufacturing_cost = (SELECT
            MIN(manufacturing_cost)
        FROM
            fact_manufacturing_cost)
```

Output 5 -

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

Request 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

SQL Code -

```
SELECT
    c.customer_code,
    c.customer,
    ROUND(AVG(f.pre_invoice_discount_pct) * 100, 2) AS average_discount_percentage
FROM
    gdb023.dim_customer c
        JOIN
    fact_pre_invoice_deductions f ON c.customer_code = f.customer_code
WHERE
    market = 'India' AND fiscal_year = 2021
GROUP BY customer , customer_code
ORDER BY AVG(pre_invoice_discount_pct) DESC
LIMIT 5
```

Output 6 -

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

Request 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

SQL Code -

```
SELECT
    MONTHNAME(date) AS month_name,
    YEAR(date) AS year_,
    CONCAT('$',
            ROUND(SUM((f.sold_quantity) * (g.gross_price)) / 1000000,
                    2)) AS gross_sales_amount_mlns
FROM
    gdb023.fact_sales_monthly f
        JOIN
    fact_gross_price g ON f.product_code = g.product_code
        AND f.fiscal_year = g.fiscal_year
        JOIN
    dim_customer c ON c.customer_code = f.customer_code
WHERE
    c.customer = 'Atliq Exclusive'
GROUP BY month_name , year_
ORDER BY year_
```

Output 7 -

month_name	year_	gross_sales_amount_mlns
September	2019	$4.50
October	2019	$5.14
November	2019	$7.52
December	2019	$4.83
January	2020	$4.74
February	2020	$4.00
March	2020	$0.38
April	2020	$0.40
May	2020	$0.78
June	2020	$1.70
July	2020	$2.55
August	2020	$2.79
September	2020	$12.35
October	2020	$13.22
November	2020	$20.46
December	2020	$12.94
January	2021	$12.40
February	2021	$10.13
March	2021	$12.14

P.T.O

Request 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

SQL Code -

```
SELECT
   CASE
      WHEN MONTH(date) IN (9 , 10, 11) THEN 'Q1'
      WHEN MONTH(date) IN (12 , 1, 2) THEN 'Q2'
      WHEN MONTH(date) IN (3 , 4, 5) THEN 'Q3'
      ELSE 'Q4'
   END AS quarters,
   SUM(sold_quantity) AS total_sold_quantity
FROM
   gdb023.fact_sales_monthly
WHERE
   fiscal_year = 2020
GROUP BY quarters
ORDER BY total_sold_quantity DESC
```

Output 8 -

quarters	total_sold_quantity
Q1	7005619
Q2	6649642
Q4	5042541
Q3	2075087

Request 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

SQL Code -

```
with cte1 as (
SELECT
   c.channel ,
   ROUND(SUM((f.sold_quantity) * (g.gross_price)) / 1000000,
            2) AS gross_sales_amount_mlns
FROM gdb023.fact_sales_monthly f
JOIN fact_gross_price g
ON f.product_code = g.product_code AND f.fiscal_year = g.fiscal_year
JOIN dim_customer c
ON c.customer_code = f.customer_code
WHERE f.fiscal_year = 2021
GROUP by c.channel)

select
channel,
gross_sales_amount_mlns,
ROUND(gross_sales_amount_mlns/Sum(gross_sales_amount_mlns) OVER()*100,2) as
percentage
from cte1
GROUP by channel
```

Output 9 -

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

Request 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

SQL Code -

```
with sold_quantity_table as (
SELECT
        d.division as division,
   d.product_code as product_code,
   d.product as product,
   SUM(f.sold_quantity) AS total_sold_quantity
FROM gdb023.fact_sales_monthly f
JOIN dim_product d
ON d.product_code = f.product_code
WHERE f.fiscal_year = 2021
GROUP by d.product, d.product_code, d.division
ORDER by total_sold_quantity),

top_sold_per_division 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 sold_quantity_table)

select * from top_sold_per_division
where rank_order <=3
```

Output 10 -

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

P.T.O