

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

```
SELECT market FROM
gdb023.dim_customer
where customer='Atliq Exclusive' and region='APAC';
```



| | market |
|---|-------------|
| ▶ | India |
| | Indonesia |
| | Japan |
| | Philippines |
| | South Korea |
| | Australia |
| | Newzealand |
| | Bangladesh |
| | India |

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

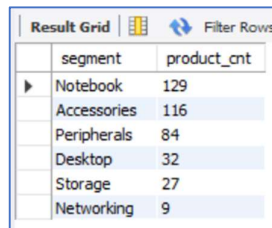
```
WITH unique_products as(
Select
count(distinct case when fiscal_year=2020 then product_code end) as
unique_products_2020,
count(distinct case when fiscal_year=2021 then product_code end) as
unique_products_2021
from fact_sales_monthly)
```

```
select *,
ROUND((unique_products_2021-unique_products_2020)*100 /
unique_products_2020,2) as per_diff
from unique_products;
```

| | unique_products_2020 | unique_products_2021 | per_diff |
|---|----------------------|----------------------|----------|
| ▶ | 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

```
SELECT segment, COUNT(DISTINCT product_code) as product_cnt
FROM gdb023.dim_product
group by segment
order by product_cnt DESC ;
```

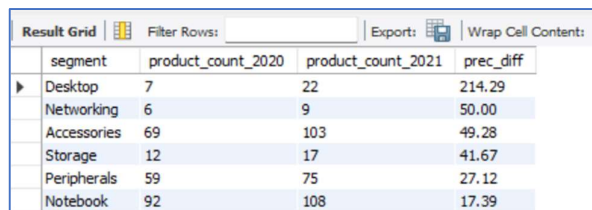


| segment | product_cnt |
|-------------|-------------|
| Notebook | 129 |
| Accessories | 116 |
| Peripherals | 84 |
| Desktop | 32 |
| Storage | 27 |
| Networking | 9 |

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 percentage_change as(
Select
d.segment,
count(distinct case when f.fiscal_year=2020 then d.product_code end) as
product_count_2020,
count(distinct case when f.fiscal_year=2021 then d.product_code end) as
product_count_2021
from fact_sales_monthly f
join dim_product d
on f.product_code=d.product_code
group by d.segment)

select *,
round(((product_count_2021-product_count_2020)*100/product_count_2020,2) as prec_diff
from percentage_change
order by prec_diff DESC;
```



| segment | product_count_2020 | product_count_2021 | prec_diff |
|-------------|--------------------|--------------------|-----------|
| Desktop | 7 | 22 | 214.29 |
| Networking | 6 | 9 | 50.00 |
| Accessories | 69 | 103 | 49.28 |
| Storage | 12 | 17 | 41.67 |
| Peripherals | 59 | 75 | 27.12 |
| Notebook | 92 | 108 | 17.39 |

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

SELECT

fm.product_code,product, manufacturing_cost

FROM fact_manufacturing_cost as fm

join dim_product as dm

on dm.product_code= fm.product_code

where manufacturing_cost=(select min(manufacturing_cost) from fact_manufacturing_cost)

or manufacturing_cost=(select max(manufacturing_cost) from fact_manufacturing_cost);

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

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.

SELECT

f.customer_code,

d.customer,

f.pre_invoice_discount_pct as average_discount_percentage

FROM

fact_pre_invoice_deductions AS f

JOIN

dim_customer AS d ON d.customer_code = f.customer_code

WHERE

f.fiscal_year = 2021

AND market = 'India'

AND f.pre_invoice_discount_pct > (SELECT

AVG(pre_invoice_discount_pct)

FROM fact_pre_invoice_deductions)

ORDER BY f.pre_invoice_discount_pct DESC

LIMIT 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 |

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

SELECT

dc.customer,

CONCAT(DATE_FORMAT(fs.date, '%b'), ' (', YEAR(fs.date), ')') AS 'Month',

fs.fiscal_year,

Round(SUM(fs.sold_quantity * fg.gross_price),2) as gross_sales_amount

FROM

fact_sales_monthly fs JOIN

dim_customer AS dc ON fs.customer_code = dc.customer_code

JOIN

fact_gross_price AS fg ON fg.product_code = fs.product_code

WHERE

dc.customer = 'Atliq Exclusive'

GROUP BY Month;

| | customer | Month | fiscal_year | gross_sales_amount |
|---|-----------------|------------|-------------|--------------------|
| ► | Atliq Exclusive | Sep (2019) | 2020 | 9092670.34 |
| | Atliq Exclusive | Nov (2019) | 2020 | 15231894.97 |
| | Atliq Exclusive | Dec (2019) | 2020 | 9755795.06 |
| | Atliq Exclusive | Jan (2020) | 2020 | 9584951.94 |
| | Atliq Exclusive | Mar (2020) | 2020 | 766976.45 |
| | Atliq Exclusive | Apr (2020) | 2020 | 800071.95 |
| | Atliq Exclusive | May (2020) | 2020 | 1586964.48 |
| | Atliq Exclusive | Jul (2020) | 2020 | 5151815.40 |
| | Atliq Exclusive | Aug (2020) | 2020 | 5638281.83 |
| | Atliq Exclusive | Sep (2020) | 2021 | 19530271.30 |

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

```
With Quarterly_sold_qty as(  
SELECT  
    date,  
    fiscal_year,  
    CONCAT('Q',QUARTER(DATE_ADD(date, INTERVAL 4 MONTH))) AS fiscal_quarter,  
    sold_quantity  
FROM  
    gdb023.fact_sales_monthly)  
SELECT  
    fiscal_quarter AS 'Quarter',  
    SUM(sold_quantity) AS total_sold_quantity  
FROM  
    Quarterly_sold_qty  
WHERE  
    fiscal_year = 2020  
GROUP BY fiscal_quarter  
ORDER BY total_sold_quantity DESC;
```

| | 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

```
with gross_sales as(
SELECT
    dc.channel ,
    fs.fiscal_year,
    Round(SUM(fs.sold_quantity * fg.gross_price)/10000000,2) as gross_sales_amount_mln
FROM
    fact_sales_monthly fs JOIN
    dim_customer AS dc ON fs.customer_code = dc.customer_code
    JOIN
    fact_gross_price AS fg ON fg.product_code = fs.product_code
where fs.fiscal_year=2021
GROUP BY channel
ORDER BY fs.fiscal_year)
```

```
select channel, gross_sales_amount_mln,
(sum(gross_sales_amount_mln)/(select SUM(gross_sales_amount_mln) from
gross_sales))*100 as pct_contribution
from gross_sales
group by channel;
```

| | channel | gross_sales_amount_mln | pct_contribution |
|---|-------------|------------------------|------------------|
| ► | Direct | 40.67 | 15.475058 |
| | Distributor | 29.72 | 11.308550 |
| | Retailer | 192.42 | 73.216392 |

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

```
with top_products as (  
  select  
    d.division,  
    d.product_code,  
    d.product,  
    d.variant,  
    sum(f.sold_quantity) as sold_qty,  
    dense_rank() over(partition by division order by sum(f.sold_quantity) desc) as  
    product_ranking  
  from fact_sales_monthly as f  
  join dim_product as d  
  on f.product_code=d.product_code  
  where f.fiscal_year=2021  
  group by division,product,variant)
```

```
select *  
from top_products where product_ranking<=3;
```

| | division | product_code | product | variant | sold_qty | product_ranking |
|---|----------|--------------|---------------------|---------------------|----------|-----------------|
| ▶ | N & S | A6720160103 | AQ Pen Drive 2 IN 1 | Premium | 701373 | 1 |
| | N & S | A6818160202 | AQ Pen Drive DRC | Plus | 688003 | 2 |
| | N & S | A6819160203 | AQ Pen Drive DRC | Premium | 676245 | 3 |
| | P & A | A2319150302 | AQ Gamers Ms | Standard 2 | 428498 | 1 |
| | P & A | A2520150501 | AQ Maxima Ms | Standard 1 | 419865 | 2 |
| | P & A | A2520150504 | AQ Maxima Ms | Plus 2 | 419471 | 3 |
| | PC | A4218110202 | AQ Digit | Standard Blue | 17434 | 1 |
| | PC | A4319110306 | AQ Velocity | Plus Red | 17280 | 2 |
| | PC | A4218110208 | AQ Digit | Premium Misty Green | 17275 | 3 |

```

with top_products as (select
d.division,
d.product_code,
d.product,
sum(f.sold_quantity) as sold_qty,
dense_rank() over(partition by division order by sum(f.sold_quantity) desc) as
product_ranking
from fact_sales_monthly as f
join dim_product as d
on f.product_code=d.product_code
where f.fiscal_year=2021
group by product)

select *
from top_products where product_ranking<=5;

```

| | division | product_code | product | sold_qty | product_ranking |
|---|----------|--------------|--------------------------|----------|-----------------|
| ▶ | N & S | A6818160201 | AQ Pen Drive DRC | 2034569 | 1 |
| | N & S | A6218160101 | AQ Digit SSD | 1240149 | 2 |
| | N & S | A6419160301 | AQ Clx1 | 1238683 | 3 |
| | N & S | A6319160201 | AQ Neuer SSD | 1225985 | 4 |
| | N & S | A6519160401 | AQ Clx2 | 1201025 | 5 |
| | P & A | A2319150301 | AQ Gamers Ms | 2477098 | 1 |
| | P & A | A2520150501 | AQ Maxima Ms | 2461991 | 2 |
| | P & A | A2218150201 | AQ Master wireless x1 Ms | 2448784 | 3 |
| | P & A | A2118150101 | AQ Master wired x1 Ms | 2447468 | 4 |
| | P & A | A2419150401 | AQ Lite Ms | 2443425 | 5 |
| | PC | A4218110201 | AQ Digit | 135092 | 1 |
| | PC | A4620110601 | AQ Gen Y | 135031 | 2 |
| | PC | A4419110401 | AQ Elite | 134431 | 3 |
| | PC | A4519110501 | AQ Gen X | 134264 | 4 |
| | PC | A4318110301 | AQ Velocity | 101757 | 5 |