EXECUTIVE MANAGEMENT PROJECT

USING MYSQL

```sql
-- Q1. 1. Provide the list of markets in which customer "Atliq Exclusive" operates its business in the APAC region.


select market from dim_customer
where customer = 'Atliq Exclusive'
and region = 'APAC' ;
```

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

market
India
Indonesia
Japan
Philiphines
South Korea
Australia
Newzealand
Bangladesh
India

```sql
-- Q2. 2. What is the percentage of unique product increase in 2021 vs. 2020? The final output contains these fields,


SELECT
    unique_products_2020,
    unique_products_2021,
    ((unique_products_2021 - unique_products_2020) / unique_products_2020) * 100 AS percentage_chg
FROM (
    SELECT
        (SELECT COUNT(DISTINCT product_code ) FROM  fact_gross_price WHERE fiscal_year = 2020) AS unique_products_2020,
        (SELECT COUNT(DISTINCT product_code) FROM fact_gross_price WHERE fiscal_year = 2021) AS unique_products_2021

) AS product_counts;

```

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

unique_products_2020	unique_products_2021	percentage_chg
245	334	36.3265

```sql
-- Q3. Provide a report with all the unique product counts for each segment and
--     sort them in descending order of product counts.


SELECT
    segment, COUNT(product_code) AS PRODUCT_COUNT
FROM
    dim_product
GROUP BY segment
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

```sql
-- Q4. 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

SELECT
    dp.segment,
    COUNT(DISTINCT CASE
            WHEN fg.fiscal_year = 2020 THEN fg.product_code
        END) AS product_count_2020,
    COUNT(DISTINCT CASE
            WHEN fg.fiscal_year = 2021 THEN fg.product_code
        END) AS product_count_2021
FROM
    dim_product dp
        JOIN
    fact_gross_price fg ON dp.product_code = fg.product_code
WHERE
    fg.fiscal_year IN (2020 , 2021)
GROUP BY dp.segment
ORDER BY (product_count_2021 - product_count_2020) DESC
LIMIT 1;
```

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

segment	product_count_2020	product_count_2021
Accessories	69	103

```sql
-- Q5. Get the products that have the highest and lowest manufacturing costs.

SELECT
  dim_product.product_code,
  dim_product.product,
  fact_manufacturing_cost.manufacturing_cost
FROM
  fact_manufacturing_cost
JOIN
  dim_product ON fact_manufacturing_cost.product_code = dim_product.product_code
WHERE
  fact_manufacturing_cost.manufacturing_cost = (SELECT MIN(manufacturing_cost) FROM fact_manufacturing_cost)
  OR fact_manufacturing_cost.manufacturing_cost = (SELECT MAX(manufacturing_cost) FROM fact_manufacturing_cost);
```

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

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

```sql
-- Q5. Get the products that have the highest and lowest manufacturing costs.

WITH ct AS (
  SELECT
    MIN(manufacturing_cost) AS MinCost,
    MAX(manufacturing_cost) AS MaxCost
  FROM
    fact_manufacturing_cost
)

SELECT
  dim_product.product_code,
  dim_product.product,
  fact_manufacturing_cost.manufacturing_cost
FROM
  fact_manufacturing_cost
JOIN
  dim_product  ON fact_manufacturing_cost.product_code = dim_product.product_code
JOIN
  Ct ON fact_manufacturing_cost.manufacturing_cost = ct.MinCost or fact_manufacturing_cost.manufacturing_cost = ct.MaxCost;
```

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

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

```sql
-- Q6. 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
    dim_customer.customer,
    dim_customer.customer_code,
    AVG(fact_pre_invoice_deductions.pre_invoice_discount_pct) AS average
FROM
    fact_pre_invoice_deductions
        JOIN
    dim_customer ON dim_customer.customer_code = fact_pre_invoice_deductions.customer_code
WHERE
    fact_pre_invoice_deductions.fiscal_year = 2021
        AND dim_customer.market = 'India'
GROUP BY dim_customer.customer , dim_customer.customer_code
ORDER BY average DESC
LIMIT 5;
```

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

customer	customer_code	average
Flipkart	90002009	0.30830000
Viveks	90002006	0.30380000
Ezone	90002003	0.30280000
Croma	90002002	0.30250000
Amazon	90002016	0.29330000

```sql
-- Q7.  Get the complete report of the Gross sales amount for the customer “Atliq Exclusive” for each month.

SELECT
    MONTHNAME(fact_sales_monthly.date) AS Month,
    YEAR(fact_sales_monthly.date) AS Year,
    ROUND(SUM(fact_gross_price.gross_price * fact_sales_monthly.sold_quantity),2) AS Gross_Sales_Amount
FROM
    fact_sales_monthly
JOIN
    dim_customer ON dim_customer.customer_code = fact_sales_monthly.customer_code
JOIN
    fact_gross_price ON fact_gross_price.product_code = fact_sales_monthly.product_code
WHERE
    dim_customer.customer = 'Atliq Exclusive'
GROUP BY
    Year, Month;
```

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

Month	Year	Gross_Sales_Amount
September	2019	9092670.34
October	2019	10378637.60
November	2019	15231894.97
December	2019	9755795.06
January	2020	9584951.94
February	2020	8083995.55
March	2020	766976.45
April	2020	800071.95
May	2020	1586964.48
June	2020	3429736.57
July	2020	5151815.40
August	2020	5638281.83

Result 3

Output

```sql
-- Q8. In which quarter of 2020, got the maximum total_sold_quantity?
-- The final output contains these fields sorted by the total_sold_quantity and Quarter


SELECT
    CASE
        WHEN MONTH(fact_sales_monthly.date) IN (9, 10, 11) THEN 'Q1'
        WHEN MONTH(fact_sales_monthly.date) IN (12, 1, 2) THEN 'Q2'
        WHEN MONTH(fact_sales_monthly.date) IN (3, 4, 5) THEN 'Q3'
        WHEN MONTH(fact_sales_monthly.date) IN (6, 7, 8) THEN 'Q4'
    END AS Quarter,
    SUM(fact_sales_monthly.sold_quantity) AS Total_Sold_Quantity
FROM
    fact_sales_monthly
WHERE
    fact_sales_monthly.date BETWEEN '2019-09-01' AND '2020-08-31'
GROUP BY
    Quarter
ORDER BY
    Total_Sold_Quantity DESC;

```

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

Quarter	Total_Sold_Quantity
Q1	7005619
Q2	6649642
Q4	5042541
Q3	2075087

```sql
-- Q9.  Which channel helped to bring more gross sales in the fiscal year 2021 and the percentage of contribution?
SELECT dim_customer.channel,
    SUM(fact_gross_price.gross_price * fact_sales_monthly.sold_quantity) AS Gross_Sales,
    ROUND(
        (SUM(fact_gross_price.gross_price * fact_sales_monthly.sold_quantity) /
         (SELECT SUM(fact_gross_price.gross_price * fact_sales_monthly.sold_quantity)
          FROM fact_sales_monthly
          JOIN fact_gross_price
          ON fact_gross_price.product_code = fact_sales_monthly.product_code
          WHERE fact_sales_monthly.date BETWEEN '2020-09-01' AND '2021-08-31')
        ) * 100, 2) AS Percentage_Contribution
FROM
    fact_sales_monthly
JOIN
    dim_customer ON dim_customer.customer_code = fact_sales_monthly.customer_code
JOIN
    fact_gross_price ON fact_gross_price.product_code = fact_sales_monthly.product_code
WHERE
    fact_sales_monthly.date BETWEEN '2020-09-01' AND '2021-08-31'
GROUP BY dim_customer.channel
ORDER BY Gross_Sales DESC;
```

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

channel	Gross_Sales	Percentage_Contribution
Retailer	1924170397.9096	73.22
Direct	406686873.9033	15.47
Distributor	297175879.7188	11.31

```sql
-- Q10. 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 ranked_products AS (
    SELECT
        dim_product.division AS division,
        dim_product.product_code AS product_code,
        dim_product.product AS product,
        SUM(fact_sales_monthly.sold_quantity) AS total_sold_quantity,
        ROW_NUMBER() OVER (PARTITION BY dim_product.division ORDER BY SUM(fact_sales_monthly.sold_quantity) DESC) AS rank_order
    FROM fact_sales_monthly
    JOIN dim_product
        ON dim_product.product_code = fact_sales_monthly.product_code
    WHERE fact_sales_monthly.fiscal_year = 2021
    GROUP BY dim_product.division, dim_product.product_code, dim_product.product
)
SELECT
    division, product_code,product, total_sold_quantity, rank_order
FROM ranked_products
WHERE rank_order <= 3
ORDER BY division, rank_order;
```

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

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

Result 2