AWS SaaS Revenue, Customer & Profitability Analysis using SQL

Dataset Source (Kaggle):

https://www.kaggle.com/datasets/nnthanh101/aws-saas-sales/data

1. What is the total revenue and total profit of the company?

```sql
-- 1 What is the total revenue and total profit of the company?
SELECT
    SUM(sales) AS total_revenue,
    SUM(profit) AS total_profit
FROM fact_sales;
```

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

total_revenue	total_profit
1165203.5362999933	130503.55640000013

2. How does revenue and profit trend over time (year-wise)?

```sql
-- 2 How does revenue and profit trend over time (year-wise)?
SELECT
    YEAR(order_date) AS order_year,
    SUM(sales) AS total_revenue,
    SUM(profit) AS total_profit
FROM fact_sales
GROUP BY YEAR(order_date)
ORDER BY order_year;
```

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

order_year	total_revenue	total_profit
2020	285130.12539999996	24427.429800000024
2021	238789.45759999953	29134.655500000008
2022	268275.0525000003	35351.397000000055
2023	373008.9008	41590.07409999995

3. Which products generate the highest revenue?

```sql
-- 3 Which products generate the highest revenue?
SELECT
    p.product_name,
    SUM(f.sales) AS total_revenue
FROM fact_sales f
JOIN dim_products p
    ON f.product_id = p.product_id
GROUP BY p.product_name
ORDER BY total_revenue DESC;
```

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

product_name	total_revenue
ContactMatcher	196355.0879999999
Site Analytics	174850.6160000001
FinanceHub	163369.9099999999
Marketing Suite - Gold	121768.67200000005
Big Ol Database	120455.71599999999
Data Smasher	83656.076
Support	68037.71600000003
Alchemy	62359.12399999998
Marketing Suite	53021.1683
OneView	50000.10599999997
SaaS Connector Pack	46585.28599999993
ChatBot Plugin	14186.821999999987
SaaS Connector Pack...	8838.405999999999
Storage	1718.829999999999

4. Which products generate the highest profit?

```sql
-- 4 Which products generate the highest profit?
SELECT
    p.product_name,
    SUM(f.profit) AS total_profit
FROM fact_sales f
JOIN dim_products p
    ON f.product_id = p.product_id
GROUP BY p.product_name
ORDER BY total_profit DESC;
```

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

product_name	total_profit
Site Analytics	23230.0128
Alchemy	22658.197699999993
Data Smasher	20457.36170000001
FinanceHub	16932.9897
Support	16796.267399999997
Marketing Suite - Gold	11665.258700000004
OneView	7448.081299999998
SaaS Connector Pack	6935.328200000002
SaaS Connector Pack - Gold	3724.659000000002
ChatBot Plugin	3313.1582000000003
Storage	571.3289000000001
ContactMatcher	566.733099999995
Big Ol Database	-970.4597999999999
Marketing Suite	-2825.3604999999993

5. Which products have the highest and lowest profit margins?

```sql
-- 5 Which products have the highest and lowest profit margins?
SELECT
    p.product_name,
    SUM(f.sales) AS total_revenue,
    SUM(f.profit) AS total_profit,
    ROUND(SUM(f.profit) / NULLIF(SUM(f.sales),0) * 100, 2) AS profit_margin_pct
FROM fact_sales f
JOIN dim_products p
    ON f.product_id = p.product_id
GROUP BY p.product_name
ORDER BY profit_margin_pct DESC;
```

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

product_name	total_revenue	total_profit	profit_margin_pct
SaaS Connector Pack - Gold	8838.405999999999	3724.659000000002	42.14
Alchemy	62359.12399999998	22658.197699999993	36.34
Storage	1718.829999999999	571.3289000000001	33.24
Support	68037.71600000003	16796.267399999997	24.69
Data Smasher	83656.076	20457.36170000001	24.45
ChatBot Plugin	14186.821999999987	3313.1582000000003	23.35
OneView	50000.10599999997	7448.081299999998	14.9
SaaS Connector Pack	46585.28599999993	6935.328200000002	14.89
Site Analytics	174850.6160000001	23230.0128	13.29
FinanceHub	163369.9099999999	16932.9897	10.36
Marketing Suite - Gold	121768.67200000005	11665.258700000004	9.58
ContactMatcher	196355.0879999999	566.733099999995	0.29
Big Ol Database	120455.71599999999	-970.4597999999999	-0.81
Marketing Suite	53021.1683	-2825.3604999999993	-5.33

6. Who are the top customers by revenue?

```sql
-- 6 Who are the top customers by revenue?
SELECT
    c.customer_name,
    SUM(f.sales) AS total_revenue
FROM fact_sales f
JOIN dim_customers c
    ON f.customer_id = c.customer_id
GROUP BY c.customer_name
ORDER BY total_revenue DESC;
```

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

customer_name	total_revenue
Anthem	45360.113999999994
Lowes	34883.829000000005
Ford Motor	27240.488
Trafigura Group	25542.954000000005
AmerisourceBergen	24802.624
Allstate	20682.602000000006
Volkswagen	20185.635999999995
Chevron	20002.787000000008
UnitedHealth Group	19984.193000000003
Bank of America Corp.	19945.619599999998
ConocoPhillips	19313.658599999995
Tyson Foods	18865.1218
Nissan Motor	18843.663600000003
Airbus	18172.965
Lukoil	17995.150999999994
FedEx	17224.541
Walt Disney	17065.762999999995

7. Who are the top customers by profit?

```sql
-- 7 Who are the top customers by profit?
SELECT
    c.customer_name,
    SUM(f.profit) AS total_profit
FROM fact_sales f
JOIN dim_customers c
    ON f.customer_id = c.customer_id
GROUP BY c.customer_name
ORDER BY total_profit DESC;
```

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

customer_name	total_profit
Lowes	8331.595099999999
Trafigura Group	7428.653200000001
Lukoil	5915.941499999995
Anthem	4685.9443999999985
UnitedHealth Group	4142.8912
Airbus	4091.4761
Chevron	3877.270100000001
Walt Disney	3431.0807000000004
Phillips 66	3277.9552000000003
Bank of America Corp.	3157.4713
Itochu	2997.0699999999993
Allstate	2597.4298000000003
Kroger	2545.86
Allianz	2289.3663
Caterpillar	2288.3577
ConocoPhillips	2246.4165
Bosch	2228.0466

8. How many unique customers does the company have?

```sql
-- 8 How many unique customers do we have?
SELECT
    COUNT(DISTINCT customer_id) AS total_customers
FROM dim_customers;
```

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

total_customers
99

9. Which customers place the most orders?

```sql
-- 9 Which customers place the most orders?
SELECT
    c.customer_name,
    COUNT(DISTINCT f.order_id) AS total_orders
FROM fact_sales f
JOIN dim_customers c
    ON f.customer_id = c.customer_id
GROUP BY c.customer_name
ORDER BY total_orders DESC;
```

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

customer_name	total_orders
Ford Motor	51
Siemens	46
Allianz	43
Tyson Foods	41
Chevron	40
Aetna	39
Comcast	39
AmerisourceBergen	38
American Express	38
Airbus	37
Volkswagen	37
Royal Dutch Shell	37
Johnson & Johnson	37
CVS Health	36
BNP Paribas	36
Allstate	35

10. Which industries generate the highest revenue?

```sql
-- 10 Which industries generate the highest revenue?
SELECT
    c.industry,
    SUM(f.sales) AS total_revenue
FROM fact_sales f
JOIN dim_customers c
    ON f.customer_id = c.customer_id
GROUP BY c.industry
ORDER BY total_revenue DESC;
```

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

industry	total_revenue
Finance	234231.41630000004
Healthcare	161941.37600000002
Energy	152084.04939999976
Manufacturing	145720.47859999983
Tech	130460.0048
Retail	130239.59750000003
Consumer Products	85060.31360000002
Communications	61794.72050000001
Transportation	51223.0436
Misc	12448.536000000002

11. Which industries are the most profitable?

```sql
-- 11 Which industries are the most profitable?
SELECT
    c.industry,
    SUM(f.profit) AS total_profit
FROM fact_sales f
JOIN dim_customers c
    ON f.customer_id = c.customer_id
GROUP BY c.industry
ORDER BY total_profit DESC;
```

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

industry	total_profit
Finance	31932.07110000003
Energy	22844.914199999985
Healthcare	18180.446399999986
Retail	14283.923799999995
Manufacturing	12474.307599999987
Consumer Products	10230.8573
Tech	9299.36760000001
Communications	7544.679499999999
Misc	2691.1202000000003
Transportation	1021.8687000000003

12. Is the SMB or Enterprise segment more profitable?

```sql
-- 12 Is SMB or Enterprise more profitable?
SELECT
    c.segment,
    SUM(f.sales) AS total_revenue,
    SUM(f.profit) AS total_profit
FROM fact_sales f
JOIN dim_customers c
    ON f.customer_id = c.customer_id
GROUP BY c.segment
ORDER BY total_profit DESC;
```

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

segment	total_revenue	total_profit
Strategic	1104626.3832999943	124991.26110000005
SMB	60577.15299999997	5512.295300000001

13. Which countries generate the highest sales?

```sql
-- 13 Which countries generate the highest sales?
SELECT
    c.country,
    SUM(f.sales) AS total_revenue
FROM fact_sales f
JOIN dim_customers c
    ON f.customer_id = c.customer_id
GROUP BY c.country
ORDER BY total_revenue DESC;
```

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

country	total_revenue
United States	1141377.301699992
United Kingdom	23826.234600000003

14. Which regions generate the highest profit?

```sql
-- 14 Which regions generate the highest profit?
SELECT
    c.region,
    SUM(f.profit) AS total_profit
FROM fact_sales f
JOIN dim_customers c
    ON f.customer_id = c.customer_id
GROUP BY c.region
ORDER BY total_profit DESC;
```

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

region	total_profit
EMEA	130503.5563999999

15. Do higher discounts lead to lower profitability?

```sql
-- 15 Do higher discounts lead to lower profit?
SELECT
    f.discount,
    COUNT(*) AS total_orders,
    SUM(f.sales) AS total_revenue,
    SUM(f.profit) AS total_profit,
    AVG(f.profit) AS avg_profit
FROM fact_sales f
GROUP BY f.discount
ORDER BY f.discount;
```

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

discount	total_orders	total_revenue	total_profit	avg_profit
0	2441	540577.6800000006	154793.40329999986	63.413930069643534
0.1	41	22486.328999999994	4124.6362	100.60088292682927
0.15	28	13203.8745	650.3429000000001	23.226532142857145
0.2	1880	387029.22400000034	43909.04090000003	23.355872819148953
0.3	105	47770.863	-4669.2807999999995	-44.469340952380946
0.32	15	8787.476799999999	-1735.6602000000003	-115.71068000000001
0.4	109	60119.78400000002	-11751.283300000001	-107.8099385321101
0.45	6	2745.0060000000003	-1168.4172	-194.73620000000003
0.5	36	49748.35499999998	-15249.9726	-423.61035
0.6	60	3190.692	-2895.6785999999997	-48.261309999999995
0.7	221	20921.322000000004	-19944.36419999999	-90.24599185520357
0.8	157	8622.929999999998	-15559.210000000006	-99.10324840764335

16. Which products are most frequently sold at a loss?

```sql
-- 16 Which products are most frequently sold at a loss?
SELECT
    p.product_name,
    COUNT(*) AS loss_orders,
    SUM(f.profit) AS total_loss
FROM fact_sales f
JOIN dim_products p
    ON f.product_id = p.product_id
WHERE f.profit < 0
GROUP BY p.product_name
ORDER BY loss_orders DESC;
```

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

product_name	loss_orders	total_loss
ContactMatcher	437	-36610.636899999976
FinanceHub	110	-4311.892699999998
Marketing Suite - Gold	80	-3410.660800000001
SaaS Connector Pack	74	-3204.7609999999995
Site Analytics	67	-4057.9222
Marketing Suite	52	-6600.906499999998
Data Smasher	38	-389.0391999999998
OneView	32	-4725.2019
Big O! Database	23	-18777.421400000003
Support	18	-1849.4710999999998
Storage	6	-10.4133

17. Which customers generate the highest losses?

```sql
-- 17 Which customers generate the highest losses?
SELECT
    c.customer_name,
    SUM(f.profit) AS total_profit
FROM fact_sales f
JOIN dim_customers c
    ON f.customer_id = c.customer_id
GROUP BY c.customer_name
HAVING total_profit < 0
ORDER BY total_profit;
```

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

customer_name	total_profit
Costco Wholesale	-2246.5576000000005
Nissan Motor	-2036.5983000000006
HonHai Precision Industry	-1704.9733000000006
Walgreens	-1580.7978
Honda Motor	-581.6806999999998
Coca-Cola	-385.52500000000003
Gazprom	-178.3722
Boeing	-161.31649999999996
Exxon Mobil	-139.99020000000002
Intel	-128.11049999999966
HSBC Holdings	-93.36119999999995
Mitsubishi	-31.042700000000067

18. Which products and industries show the strongest growth over time?

```sql
-- 18 Which products and industries show the strongest growth over time?
SELECT
    YEAR(f.order_date) AS order_year,
    p.product_name,
    c.industry,
    SUM(f.sales) AS total_revenue
FROM fact_sales f
JOIN dim_products p
    ON f.product_id = p.product_id
JOIN dim_customers c
    ON f.customer_id = c.customer_id
GROUP BY YEAR(f.order_date), p.product_name, c.industry
ORDER BY order_year, total_revenue DESC;
```

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

order_year	product_name	industry	total_revenue
2020	Big Ol Database	Healthcare	26372.733
2020	ContactMatcher	Finance	16480.539
2020	ContactMatcher	Healthcare	12193.140000000001
2020	Site Analytics	Finance	10340.964000000002
2020	Site Analytics	Energy	10079.704000000002
2020	FinanceHub	Consumer Products	8977.158000000001
2020	Support	Retail	8782.006
2020	Big Ol Database	Transportation	8159.952
2020	Site Analytics	Tech	7923.902
2020	FinanceHub	Retail	7117.832
2020	Big Ol Database	Tech	5979.4800000000005
2020	ContactMatcher	Retail	5868.2025
2020	FinanceHub	Healthcare	5851.514
2020	FinanceHub	Finance	5699.308
2020	Site Analytics	Healthcare	5594.245999999999