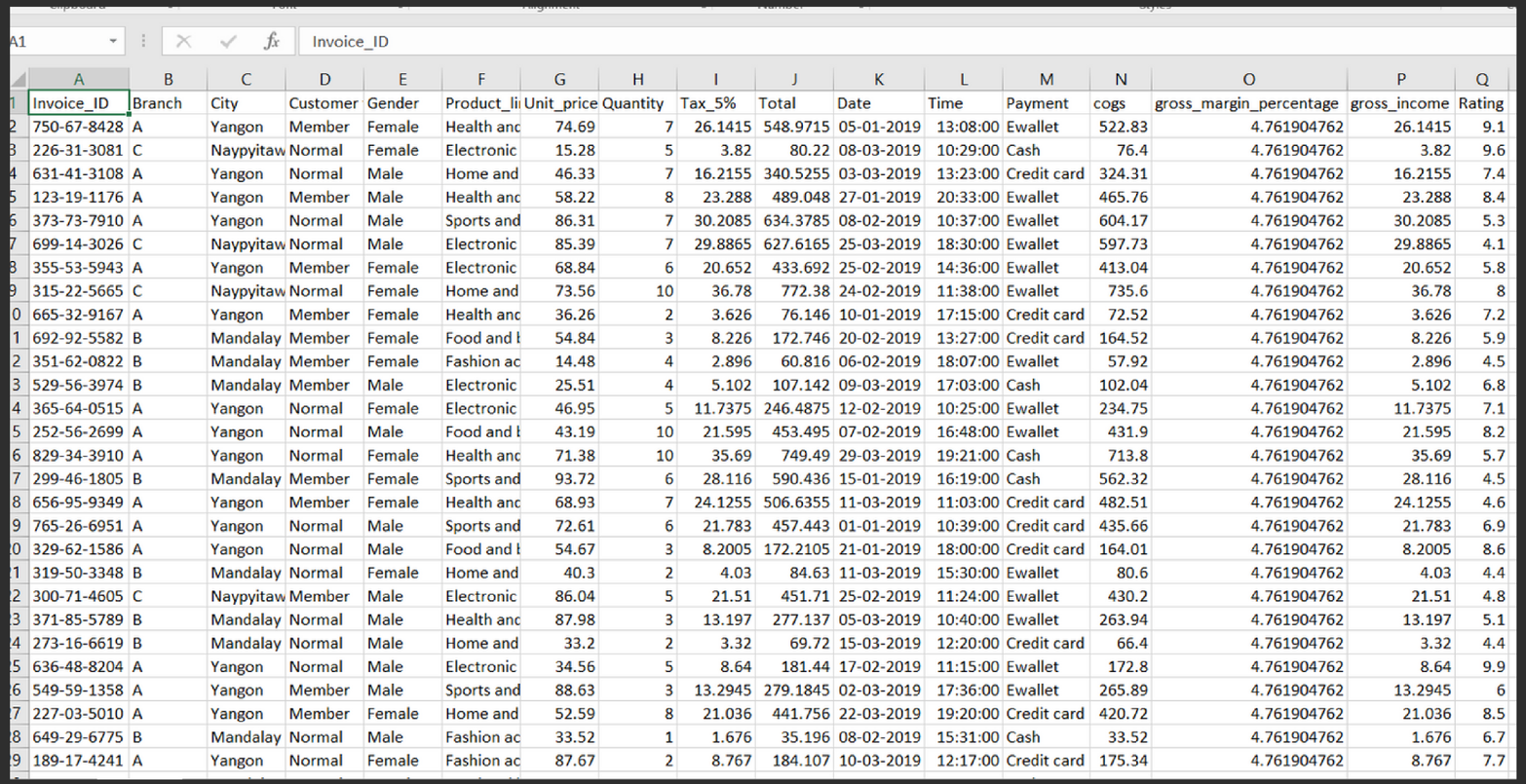
DATA ANALYTICS WITH SQL

Amazon sales Data Analysis

Based on the sales data from Amazon, address specific questions.

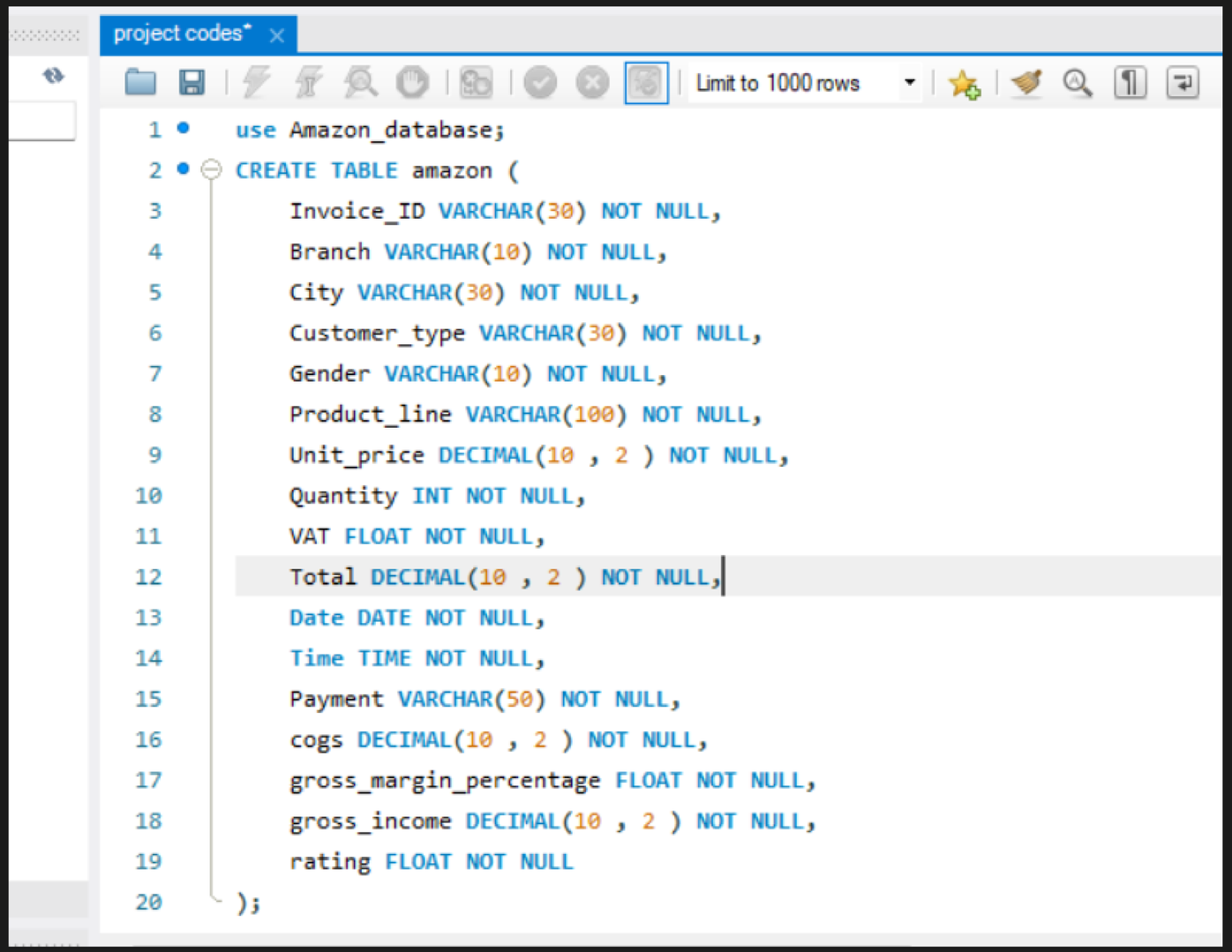
The source of the data is Kaggle.



A1 | Invoice_ID

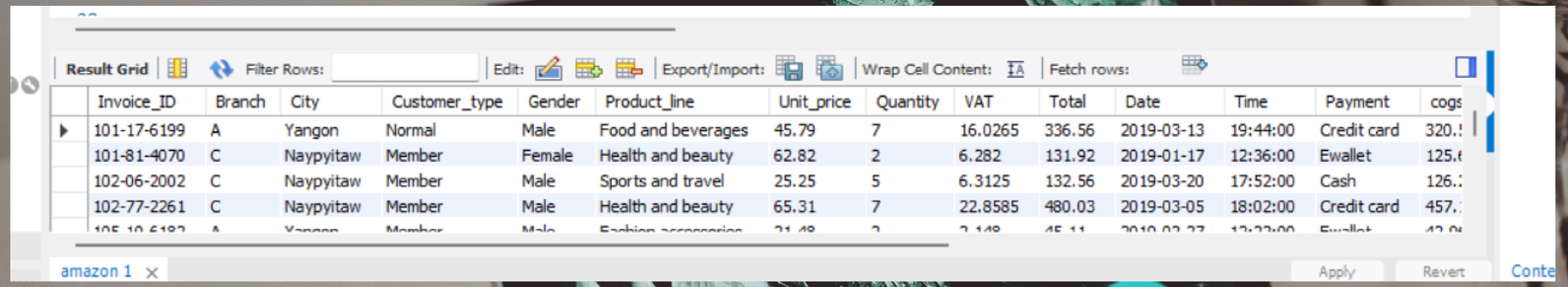
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Invoice_ID	Branch	City	Customer	Gender	Product_li	Unit_price	Quantity	Tax_5%	Total	Date	Time	Payment	cogs	gross_margin_percentage	gross_income	Rating
2	750-67-8428	A	Yangon	Member	Female	Health and	74.69	7	26.1415	548.9715	05-01-2019	13:08:00	Ewallet	522.83	4.761904762	26.1415	9.1
3	226-31-3081	C	Naypyitaw	Normal	Female	Electronic	15.28	5	3.82	80.22	08-03-2019	10:29:00	Cash	76.4	4.761904762	3.82	9.6
4	631-41-3108	A	Yangon	Normal	Male	Home and	46.33	7	16.2155	340.5255	03-03-2019	13:23:00	Credit card	324.31	4.761904762	16.2155	7.4
5	123-19-1176	A	Yangon	Member	Male	Health and	58.22	8	23.288	489.048	27-01-2019	20:33:00	Ewallet	465.76	4.761904762	23.288	8.4
6	373-73-7910	A	Yangon	Normal	Male	Sports and	86.31	7	30.2085	634.3785	08-02-2019	10:37:00	Ewallet	604.17	4.761904762	30.2085	5.3
7	699-14-3026	C	Naypyitaw	Normal	Male	Electronic	85.39	7	29.8865	627.6165	25-03-2019	18:30:00	Ewallet	597.73	4.761904762	29.8865	4.1
8	355-53-5943	A	Yangon	Member	Female	Electronic	68.84	6	20.652	433.692	25-02-2019	14:36:00	Ewallet	413.04	4.761904762	20.652	5.8
9	315-22-5665	C	Naypyitaw	Normal	Female	Home and	73.56	10	36.78	772.38	24-02-2019	11:38:00	Ewallet	735.6	4.761904762	36.78	8
10	665-32-9167	A	Yangon	Member	Female	Health and	36.26	2	3.626	76.146	10-01-2019	17:15:00	Credit card	72.52	4.761904762	3.626	7.2
11	692-92-5582	B	Mandalay	Member	Female	Food and b	54.84	3	8.226	172.746	20-02-2019	13:27:00	Credit card	164.52	4.761904762	8.226	5.9
12	351-62-0822	B	Mandalay	Member	Female	Fashion ac	14.48	4	2.896	60.816	06-02-2019	18:07:00	Ewallet	57.92	4.761904762	2.896	4.5
13	529-56-3974	B	Mandalay	Member	Male	Electronic	25.51	4	5.102	107.142	09-03-2019	17:03:00	Cash	102.04	4.761904762	5.102	6.8
14	365-64-0515	A	Yangon	Normal	Female	Electronic	46.95	5	11.7375	246.4875	12-02-2019	10:25:00	Ewallet	234.75	4.761904762	11.7375	7.1
15	252-56-2699	A	Yangon	Normal	Male	Food and b	43.19	10	21.595	453.495	07-02-2019	16:48:00	Ewallet	431.9	4.761904762	21.595	8.2
16	829-34-3910	A	Yangon	Normal	Female	Health and	71.38	10	35.69	749.49	29-03-2019	19:21:00	Cash	713.8	4.761904762	35.69	5.7
17	299-46-1805	B	Mandalay	Member	Female	Sports and	93.72	6	28.116	590.436	15-01-2019	16:19:00	Cash	562.32	4.761904762	28.116	4.5
18	656-95-9349	A	Yangon	Member	Female	Health and	68.93	7	24.1255	506.6355	11-03-2019	11:03:00	Credit card	482.51	4.761904762	24.1255	4.6
19	765-26-6951	A	Yangon	Normal	Male	Sports and	72.61	6	21.783	457.443	01-01-2019	10:39:00	Credit card	435.66	4.761904762	21.783	6.9
20	329-62-1586	A	Yangon	Normal	Male	Food and b	54.67	3	8.2005	172.2105	21-01-2019	18:00:00	Credit card	164.01	4.761904762	8.2005	8.6
21	319-50-3348	B	Mandalay	Normal	Female	Home and	40.3	2	4.03	84.63	11-03-2019	15:30:00	Ewallet	80.6	4.761904762	4.03	4.4
22	300-71-4605	C	Naypyitaw	Member	Male	Electronic	86.04	5	21.51	451.71	25-02-2019	11:24:00	Ewallet	430.2	4.761904762	21.51	4.8
23	371-85-5789	B	Mandalay	Normal	Male	Health and	87.98	3	13.197	277.137	05-03-2019	10:40:00	Ewallet	263.94	4.761904762	13.197	5.1
24	273-16-6619	B	Mandalay	Normal	Male	Home and	33.2	2	3.32	69.72	15-03-2019	12:20:00	Credit card	66.4	4.761904762	3.32	4.4
25	636-48-8204	A	Yangon	Normal	Male	Electronic	34.56	5	8.64	181.44	17-02-2019	11:15:00	Ewallet	172.8	4.761904762	8.64	9.9
26	549-59-1358	A	Yangon	Member	Male	Sports and	88.63	3	13.2945	279.1845	02-03-2019	17:36:00	Ewallet	265.89	4.761904762	13.2945	6
27	227-03-5010	A	Yangon	Member	Female	Home and	52.59	8	21.036	441.756	22-03-2019	19:20:00	Credit card	420.72	4.761904762	21.036	8.5
28	649-29-6775	B	Mandalay	Normal	Male	Fashion ac	33.52	1	1.676	35.196	08-02-2019	15:31:00	Cash	33.52	4.761904762	1.676	6.7
29	189-17-4241	A	Yangon	Normal	Female	Fashion ac	87.67	2	8.767	184.107	10-03-2019	12:17:00	Credit card	175.34	4.761904762	8.767	7.7

creates a table named "amazon" within the "Amazon_database" database, with the specified columns and their data types.



Retrieve all records from the Amazon dataset.

```
24 •    SELECT * FROM amazon_database.amazon;
25
```



Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Fetch rows:

Invoice_ID	Branch	City	Customer_type	Gender	Product_line	Unit_price	Quantity	VAT	Total	Date	Time	Payment	cogs
101-17-6199	A	Yangon	Normal	Male	Food and beverages	45.79	7	16.0265	336.56	2019-03-13	19:44:00	Credit card	320.5
101-81-4070	C	Naypyitaw	Member	Female	Health and beauty	62.82	2	6.282	131.92	2019-01-17	12:36:00	Ewallet	125.6
102-06-2002	C	Naypyitaw	Member	Male	Sports and travel	25.25	5	6.3125	132.56	2019-03-20	17:52:00	Cash	126.2
102-77-2261	C	Naypyitaw	Member	Male	Health and beauty	65.31	7	22.8585	480.03	2019-03-05	18:02:00	Credit card	457.2
105-10-6182	A	Yangon	Member	Male	Fashion accessories	21.48	2	2.148	45.11	2019-02-27	12:22:00	Ewallet	42.96

amazon 1 | Apply | Revert | Conte

Verify whether there are any duplicate values within the dataset.

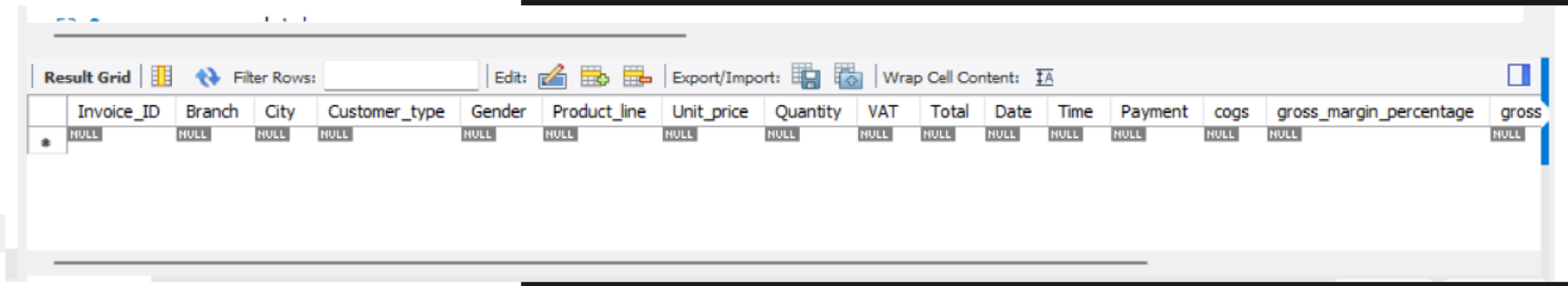
```
-- there is no duplicate value in this data.
select Invoice_ID, count(*) from amazon group by Invoice_ID having count(*)>1;
```

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

Invoice_ID	count(*)

Verify the presence of any null values within the dataset.

```sql
-- There is no null value in this data.
SELECT
    *
FROM
    amazon
WHERE
    Invoice_ID IS NULL OR Branch IS NULL
        OR City IS NULL
        OR 'Customer type' IS NULL
        OR Gender IS NULL
        OR Product_line IS NULL
        OR Unit_price IS NULL
        OR Quantity IS NULL
        OR 'Tax_5%' IS NULL
        OR Total IS NULL
        OR 'Date' IS NULL
           OR 'Time' IS NULL
           OR Payment IS NULL
           OR cogs IS NULL
           OR gross_margin_percentage IS NULL
           OR gross_income IS NULL
 or Rating is null;
```



-- Add a new column named timeofday to give insight of sales in the Morning, Afternoon and Evening.

```
56 •    alter table amazon
57      add column time_of_day varchar(30);
```

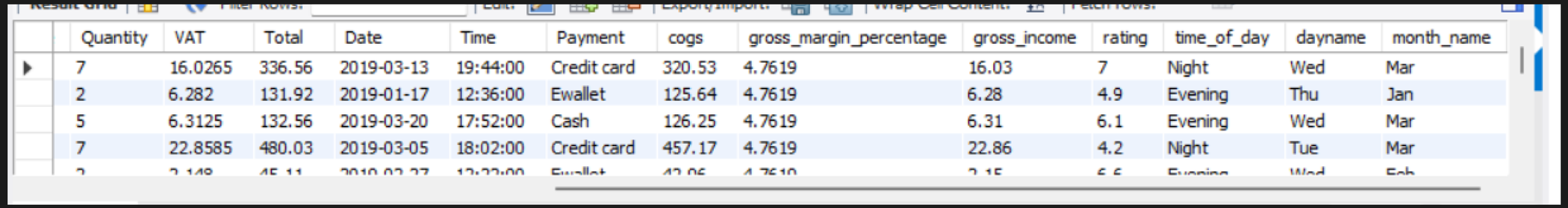
```
60 •    update amazon
61      set time_of_day =
62          case when time(time) between '00:00:00' and '11:59:59' then 'Morning'
63               when time(Time) between '12:00:00' and '17:59:59' then 'Evening'
64               else 'Night'
65          end;
```

-- Add a new column named dayname that contains the extracted days of the week on which the given transaction
-- took place (Mon, Tue, Wed, Thur, Fri). This will help answer the question on which week of the day each branch is busiest.

```
71 •    alter table amazon
72      add column dayname varchar(3);
73
74 •    update amazon
75      set dayname = date_format(Date,'%a');
76
```

-- Add a new column named monthname that contains the extracted months of the year on which the given transaction took place (Jan, Feb, Mar). Help determine which month of the year has the most sales and profit.

```
82 •  alter table amazon
83    add column month_name varchar(3);
84 •  alter table amazon
85    drop month_name;
86 •  update amazon
87    set month_name = substring(monthname(Date),1,3);
```



Quantity	VAT	Total	Date	Time	Payment	cogs	gross_margin_percentage	gross_income	rating	time_of_day	dayname	month_name
7	16.0265	336.56	2019-03-13	19:44:00	Credit card	320.53	4.7619	16.03	7	Night	Wed	Mar
2	6.282	131.92	2019-01-17	12:36:00	Ewallet	125.64	4.7619	6.28	4.9	Evening	Thu	Jan
5	6.3125	132.56	2019-03-20	17:52:00	Cash	126.25	4.7619	6.31	6.1	Evening	Wed	Mar
7	22.8585	480.03	2019-03-05	18:02:00	Credit card	457.17	4.7619	22.86	4.2	Night	Tue	Mar

-- 1. What is the count of distinct cities in the dataset?

```sql
SELECT
    city, COUNT(DISTINCT city) AS count_distinct_city
FROM
    amazon
GROUP BY city
ORDER BY count_distinct_city DESC;
```

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

city	count_distinct_city
Mandalay	1
Naypyitaw	1
Yangon	1

-- 2. For each branch, what is the corresponding city?

```sql
-- 2. For each branch, what is the corresponding city?
select branch, max(city) as coressponding_city from amazon group by branch;
```

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

branch	coressponding_city
A	Yangon
C	Naypyitaw
B	Mandalay

-- 3.What is the count of distinct product lines in the dataset?

```
19      -- 3.What is the count of distinct product lines in the dataset?
20  •   select count(distinct product_line) as distinct_product_line from amazon;
```

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

distinct_product_line
6

```
19      -- 3.What is the count of distinct product lines in the dataset?
20  •   select count(distinct product_line) as distinct_product_line from amazon;
21
22  •   select product_line, count(distinct product_line) as count_of_distinct_product_line from amazon group by product_line;
23
```

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

product_line	count_of_distinct_product_line
Electronic accessories	1
Fashion accessories	1
Food and beverages	1
Health and beauty	1
Home and lifestyle	1
Sports and travel	1

-- 4. Which payment method occurs most frequently?

```sql
-- 4. Which payment method occurs most frequently?
SELECT
    payment,
    COUNT(payment) AS most__frequently_used_payment_method
FROM
    amazon
GROUP BY payment
ORDER BY most__frequently_used_payment_method DESC
LIMIT 1;
```

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

payment	most__frequently_used_payment_method
Ewallet	345

-- 5. Which product line has the highest sales?

```sql
-- in terms of quantity

SELECT
    product_line,
    SUM(quantity) AS highest_product_line_quantity_sales
FROM
    amazon
GROUP BY product_line
ORDER BY highest_product_line_quantity_sales DESC
LIMIT 1;
```

product_line	highest_product_line_quantity_sales
Electronic accessories	971

```sql
-- In terms amaount
SELECT
    Product_line, SUM(total) AS total_sales
FROM
    amazon
GROUP BY Product_line
ORDER BY total_sales DESC
LIMIT 1;
```

Product_line	total_sales
Food and beverages	56144.96

-- 6. How much revenue is generated each month?

```
56    -- 6. How much revenue is generated each month?
57 •  select count(distinct month_name) from amazon;
58
```

Result Grid | Filter Rows: | Export: | Wrap Cel

	count(distinct month_name)
▶	3

```
58
59 •  SELECT
60        month_name, SUM(total) AS total_revenue
61    FROM
62        amazon
63    GROUP BY month_name
64    ORDER BY total_revenue DESC;
65
```

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

	month_name	total_revenue
▶	Jan	116292.11
	Mar	109455.74
	Feb	97219.58

-- 7. In which month did the cost of goods sold reach its peak?

```sql
66      -- 7. In which month did the cost of goods sold reach its peak?
67
68      SELECT
69          month_name, SUM(cogs) AS highest_cogs
70      FROM
71          amazon
72      GROUP BY month_name
73      ORDER BY highest_cogs DESC
74      LIMIT 1;
75
```

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

month_name	highest_cogs
Jan	110754.16

-- 8. Which product line generated the highest revenue?

```sql
-- 8. Which product line generated the highest revenue?

SELECT
    product_line, SUM(total) AS highest_revenue
FROM
    amazon
GROUP BY product_line
ORDER BY highest_revenue DESC
LIMIT 1;
```

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

product_line	highest_revenue
Food and beverages	56144.96

-- 9. In which city was the highest revenue recorded?

```sql
-- 9. In which city was the highest revenue recorded?

SELECT
    city, SUM(total) AS highest_revenue
FROM
    amazon
GROUP BY city
ORDER BY highest_revenue DESC
LIMIT 1;
```

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

city	highest_revenue
Naypyitaw	110568.86

-- 10. Which product line incurred the highest Value Added Tax?

```sql
-- 10. Which product line incurred the highest Value Added Tax?

SELECT
    product_line, SUM(VAT) AS highest_VAT
FROM
    amazon
GROUP BY product_line
ORDER BY highest_VAT DESC
LIMIT 1;
```

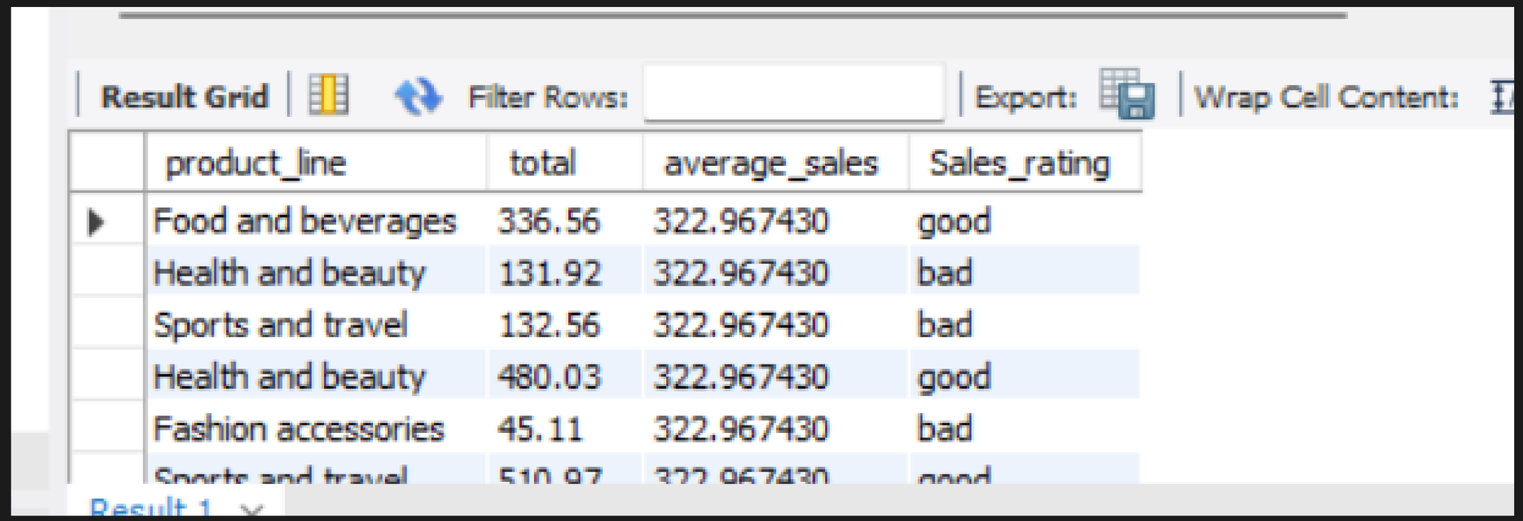
Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

product_line	highest_VAT
Food and beverages	2673.563990712166

-- 11. For each product line, add a column indicating "Good" if its sales are above average, otherwise "Bad."

```sql
-- 111. For each product line, add a column indicating "Good" if its sales are above average, otherwise "Bad."
Open a script file in this editor
        product_line, total, average_sales,
        CASE WHEN total > average_sales THEN 'good' ELSE 'bad' END AS Sales_rating
        FROM (SELECT  product_line, total, (SELECT  AVG(total) FROM amazon) AS average_sales FROM amazon) AS subquery;

```



product_line	total	average_sales	Sales_rating
Food and beverages	336.56	322.967430	good
Health and beauty	131.92	322.967430	bad
Sports and travel	132.56	322.967430	bad
Health and beauty	480.03	322.967430	good
Fashion accessories	45.11	322.967430	bad
Sports and travel	510.97	322.967430	good

Result 1

-- 12. Identify the branch that exceeded the average number of products sold.

```sql
-- 12. Identify the branch that exceeded the average number of products sold.
SELECT
    branch, SUM(quantity) AS total_quantity_sold
FROM
    amazon
GROUP BY branch
HAVING total_quantity_sold > (SELECT
        AVG(quantity)
    FROM
        amazon);
```

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

branch	total_quantity_sold
A	1859
C	1831
B	1820

-- 13. Calculate the average rating for each product line.

```sql
-- 13. Calculate the average rating for each product line.

SELECT
    product_line, AVG(rating) AS average_rating
FROM
    amazon
GROUP BY Product_line;
```

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

product_line	average_rating
Food and beverages	7.11321838970842
Health and beauty	7.003289457998778
Sports and travel	6.916265062538974
Fashion accessories	7.0292134660013605
Home and lifestyle	6.8375
Electronic accessories	6.924705881230971

-- 14. Which product line is most frequently associated with each gender?

```sql
-- 14. Which product line is most frequently associated with each gender?

SELECT
    product_line, gender, COUNT(*) AS frequency
FROM
    amazon
GROUP BY Product_line , gender
HAVING frequency = (SELECT
        MAX(frequency)
    FROM
        (SELECT
            product_line, gender, COUNT(*) AS frequency
        FROM
            amazon
        GROUP BY Product_line , gender) AS subquery
    WHERE
        subquery.gender = amazon.gender);
```

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

product_line	gender	frequency
Health and beauty	Male	88
Fashion accessories	Female	96

-- 15. Count the sales occurrences for each time of day on every weekday.

```sql
-- In terms of quantity
SELECT
    dayname, time_of_day, SUM(quantity)
FROM
    amazon
GROUP BY dayname , time_of_day
ORDER BY dayname;
```

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

dayname	time_of_day	SUM(quantity)
Fri	Evening	400
Fri	Morning	140
Fri	Night	218
Mon	Evening	387
Mon	Morning	116
Mon	Night	135

```sql
-- in terms of revenue
SELECT
    dayname, time_of_day, SUM(total)
FROM
    amazon
GROUP BY dayname , time_of_day
ORDER BY dayname;
```

Result Grid | Filter Rows: | Export: | Wrap

dayname	time_of_day	sum(total)
Fri	Evening	22831.66
Fri	Morning	8321.02
Fri	Night	12773.75
Mon	Evening	23500.73
Mon	Morning	6611.08
Mon	Night	7787.34

Result 9

-- 16. Identify the customer type contributing the highest revenue.

```sql
-- 16. Identify the customer type contributing the highest revenue
SELECT
    customer_type, SUM(total) AS total_revenue
FROM
    amazon
GROUP BY Customer_type
ORDER BY total_revenue DESC
LIMIT 1;
```

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

customer_type	total_revenue
Member	164223.81

-- 17. Determine the city with the highest VAT percentage.

```sql
-- 17. Determine the city with the highest VAT percentage.

SELECT
    city,
    (SUM(VAT) / SUM(total)) * 100 AS Highest_VAT_percentage
FROM
    amazon
GROUP BY city
ORDER BY Highest_VAT_percentage DESC
LIMIT 1;
```

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

city	Highest_VAT_percentage
Naypyitaw	4.761898133409432

18. Identify the customer type with the highest VAT payments.

```sql
-- 18. Identify the customer type with the highest VAT payments.

SELECT
    customer_type, SUM(VAT) AS total_VAT
FROM
    amazon
GROUP BY Customer_type
ORDER BY total_VAT DESC
LIMIT 1;
```

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

customer_type	total_VAT
Member	7820.163996100426

-- 19. What is the count of distinct customer types in the dataset?

```
95      -- 19. What is the count of distinct customer types in the dataset?
96
97  •   SELECT
98          COUNT(DISTINCT customer_type) AS distinct_customer_type
99      FROM
100         amazon;
101
```

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

distinct_customer_type
2

-- 20. What is the count of distinct payment methods in the dataset?

```
102     -- 20. What is the count of distinct payment methods in the dataset?
103
104 •   SELECT
105         COUNT(DISTINCT payment) AS total_payment_methods
106     FROM
107         amazon;
108
```

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

total_payment_methods
3

-- 21. Which customer type occurs most frequently?

```sql
-- 21. Which customer type occurs most frequently?

SELECT
    customer_type, COUNT(customer_type) AS frequency
FROM
    amazon
GROUP BY customer_type
ORDER BY frequency DESC
LIMIT 1;

-- 22. Identify the customer type with the highest purchase
```

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

customer_type	frequency
Member	501

-- 22. Identify the customer type with the highest purchase frequency.
-- in terms of quantity

```sql
-- 22. Identify the customer type with the highest purchase frequency.
-- in terms of quantity
SELECT
    customer_type, SUM(quantity) AS Total_quantity_purchase
FROM
    amazon
GROUP BY Customer_type
ORDER BY Total_quantity_purchase DESC
LIMIT 1;
```

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

customer_type	Total_quantity_purchase
Member	2785

-- 23. Determine the predominant gender among customers. -- 24. Examine the distribution of genders within each branch.

```sql
-- 23. Determine the predominant gender among customers.

SELECT
    customer_type, gender, COUNT(gender) AS frequency
FROM
    amazon
GROUP BY customer_type , gender
ORDER BY frequency DESC
LIMIT 2;
```

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

customer_type	gender	frequency
Member	Female	261
Normal	Male	259

```sql
-- 24. Examine the distribution of genders within each branch.

SELECT
    branch, gender, COUNT(gender) AS frequncy
FROM
    amazon
GROUP BY branch , gender
ORDER BY branch;
```

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

branch	gender	frequncy
A	Female	161
A	Male	179
B	Female	162
B	Male	170
C	Female	178
C	Male	150

Result 4

-- 25. Identify the time of day when customers provide the most ratings.

```
43      -- 25. Identify the time of day when customers provide the most ratings.
44
45      SELECT
46          time_of_day, COUNT(rating) total_number_of_rating
47      FROM
48          amazon
49      GROUP BY Time_of_day
50      ORDER BY total_number_of_rating DESC
51      LIMIT 1;
52
```

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

time_of_day	total_number_of_rating
Evening	528

-- 26. Determine the day of the week with the highest average ratings for each branch.

```
63
64 •  SELECT
65        dayname, branch, AVG(rating) AS average_rating
66    FROM
67        amazon
68    GROUP BY dayname , branch
69    ORDER BY average_rating DESC
70    LIMIT 1;
```

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

dayname	branch	average_rating
Mon	B	7.335897384545742

Result 6

The project involves the creation of a database named "Amazon_database" and the design of a table called "amazon" to store Amazon sales data. This table comprises various fields such as Invoice_ID, Branch, City, Customer_type, Gender, Product_line, Unit_price, Quantity, VAT, Total, Date, Time, Payment, cogs, gross_margin_percentage, gross_income, and rating.

Following the creation of the table, several SQL queries are executed to analyze the dataset and extract meaningful insights. These queries include identifying duplicate values, checking for null values, and adding new columns for further analysis. Additionally, various queries are conducted to answer specific questions related to sales trends, revenue generation, product performance, customer behavior, and more.

Furthermore, the project involves querying the dataset to uncover insights such as the count of distinct cities and product lines, the most frequently used payment method, the highest revenue-generating product line, and the customer type contributing the highest revenue, among others. Moreover, the analysis includes examining the distribution of genders within each branch, determining the predominant gender among customers, and identifying the time of day when customers provide the most ratings.

Overall, the project aims to provide comprehensive insights into Amazon sales data using SQL queries and database operations, enabling stakeholders to make informed decisions and optimize business strategies.

Thank you