

SQL-Capstone Project

Analysis of Amazon sales to understand the different factors that affect sales of the branches.

Nikhil Karaka

Agenda



Introduction



Data Wrangling
And Feature
Engineering



Business questions



Analysis



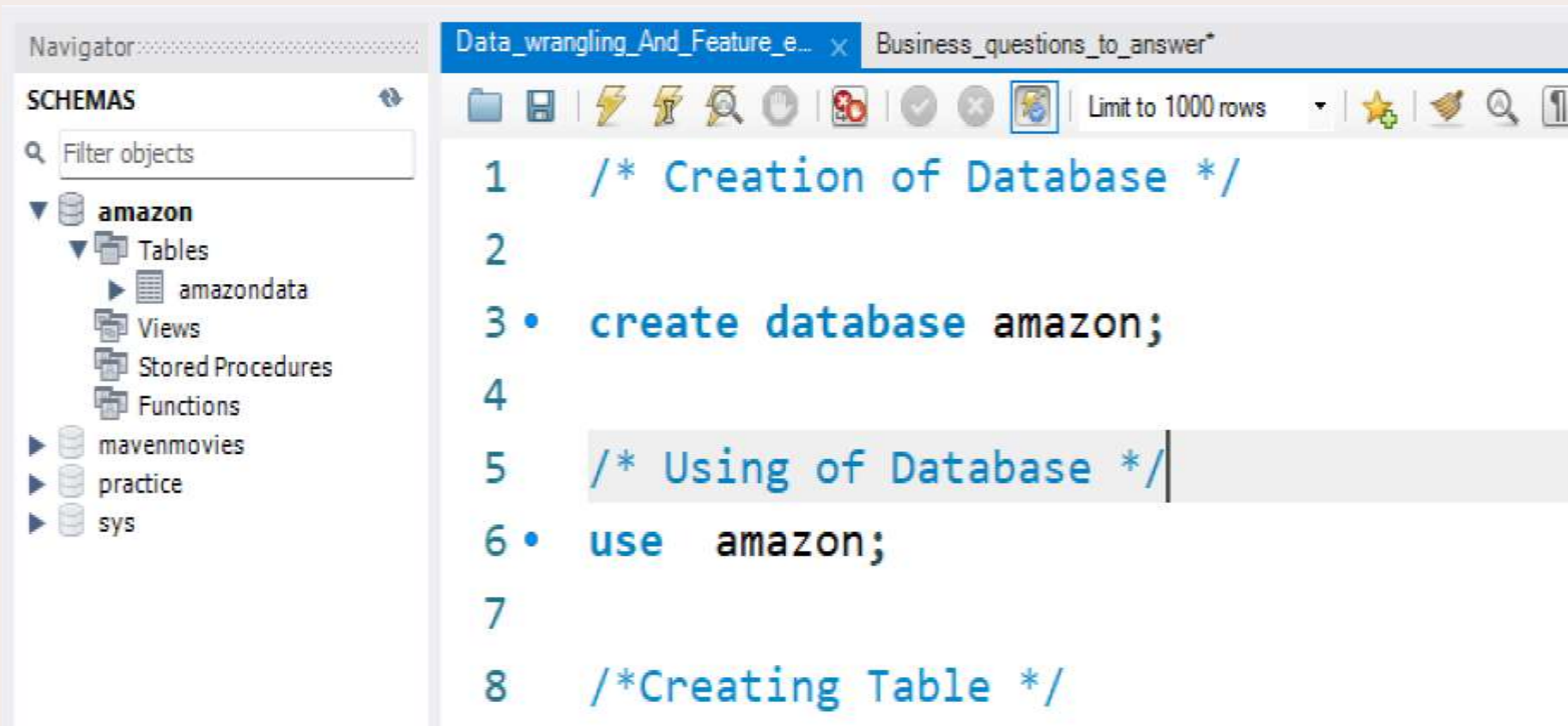
Insights and
Summary

Introduction

This dataset contains sales transactions from three different branches of Amazon, respectively located in Mandalay, Yangon and Naypyitaw. The data contains 20 columns and 1000 rows.

Data Wrangling

- Building a database



The screenshot displays a database management tool interface. On the left, a 'Navigator' pane shows a 'SCHEMAS' tree with a search bar 'Filter objects'. The tree includes a database named 'amazon' with sub-items 'Tables' (containing 'amazondata'), 'Views', 'Stored Procedures', and 'Functions'. Other databases listed are 'mavenmovies', 'practice', and 'sys'. The main editor area has a tab titled 'Data_wrangling_And_Feature_e...' and a sub-tab 'Business_questions_to_answer*'. The editor contains SQL code with line numbers 1 through 8. The code includes comments for database creation and usage, and a command to create a database named 'amazon'.

```
1  /* Creation of Database */  
2  
3  • create database amazon;  
4  
5  /* Using of Database */  
6  • use amazon;  
7  
8  /*Creating Table */
```

- Creating a table and Insert the data

The screenshot shows a database management interface. On the left, a 'Navigator' pane displays a tree of schemas: 'amazon' (containing 'Tables' with 'amazondata', 'Views', 'Stored Procedures', and 'Functions'), 'mavenmovies', 'practice', and 'sys'. The main editor window, titled 'Data_wrangling_And_Feature_e...' and 'Business_questions_to_answer*', contains a SQL script to create a table named 'amazondata'. The script is as follows:

```
10 • create table amazondata(  
11     invoice_id VARCHAR(30) not null ,  
12     branch VARCHAR(5) not null,  
13     city VARCHAR(30) not null,  
14     customer_type VARCHAR(30) not null,  
15     gender VARCHAR(10) not null,  
16     product_line VARCHAR(100) not null,  
17     unit_price DECIMAL(10,2) not null,  
18     quantity Int not null,  
19     VAT FLOAT(6,4) not null,  
20     total DECIMAL(10,2) not null,  
21     date Date not null,  
22     time time not null,  
23     payment_method varchar(30) not null,  
24     cogs DECIMAL(10,2) not null,  
25     gross_margin_percentage FLOAT(11,9) not null,  
26     gross_income DECIMAL(10,2) not null,  
27     rating FLOAT not null  
28 );
```

The screenshot shows a database IDE interface. On the left is a 'SCHEMAS' pane with a tree view containing 'amazon' (with sub-items 'Tables', 'Views', 'Stored Procedures', 'Functions'), 'mavenmovies', 'practice', and 'sys'. The main editor window has a title bar with 'Data_wrangling_And_Feature_e...' and 'Business_questions_to_answer'. The editor contains a SQL comment block starting at line 31 and ending at line 39. The comment text describes challenges faced and changes made during data wrangling. The IDE toolbar includes icons for file operations, a 'Limit to 1000 rows' dropdown, and other standard tools.

```
31  /* Challenges Faced and changes maded:  
32      -> time column in csv is not correctly formatted  
33      so we changed the Timestamp to time type of column to extract time  
34      -> Payment_method is in categorical format so we've used varchar  
35      instead of using the decimal type.  
36      ->In rating column of CSV the range exceeds than mentioned so I've  
37      removed the range in column type while creating the table  
38      ** AND I've imported data from csv with Table data import Wizard**  
39  */  
40
```


- There are no null values in our database as in creating the tables, we set NOT NULL for each field, hence null values are filtered out.

Feature Engineering

```
40
41  /* FEATURE ENGINEERING */
42
43  /* Add a new column named timeofday */
44 • alter table amazodata add column timeofday VARCHAR(15)
45 as (
46     case
47         when TIME(time) between '00:00:00' AND '11:59:59' then 'Morning'
48         when TIME(time) between '12:00:00' AND '17:59:59' then 'Afternoon'
49         when TIME(time) between '18:00:00' AND '23:59:59' then 'Evening'
50         else 'Unknown'
51     end
52 )stored;
53
```
















```
55  /* Add a new column named dayname */
56
57 • alter table amazodata add column dayname VARCHAR(3)
58 as (
59     case
60         when DAYOFWEEK(date) = 1 then 'Sun'
61         when DAYOFWEEK(date) = 2 then 'Mon'
62         when DAYOFWEEK(date) = 3 then 'Tue'
63         when DAYOFWEEK(date) = 4 then 'Wed'
64         when DAYOFWEEK(date) = 5 then 'Thu'
65         when DAYOFWEEK(date) = 6 then 'Fri'
66         when DAYOFWEEK(date) = 7 THEN 'Sat'
67         else 'Unknown'
68     end)
69 stored;
```

Limit to 1000 rows

```
71  /* Add a new column named monthname */
72 • alter table amazondata add column monthname VARCHAR(3)
73 as (
74     case
75         when MONTH(date) = 1 then 'Jan'
76         when MONTH(date) = 2 then 'Feb'
77         when MONTH(date) = 3 then 'Mar'
78         when MONTH(date) = 4 then 'Apr'
79         when MONTH(date) = 5 then 'May'
80         when MONTH(date) = 6 then 'Jun'
81         when MONTH(date) = 7 then 'Jul'
82         when MONTH(date) = 8 then 'Aug'
83         when MONTH(date) = 9 then 'Sep'
84         when MONTH(date) = 10 then 'Oct'
85         when MONTH(date) = 11 then 'Nov'
86         when MONTH(date) = 12 then 'Dec'
87         else 'Unknown'
88     end)stored;
```


Data_wrangling_And_Feature_e... x Business_questions_to_answer*





Limit to 1000 rows


```
88     end)stored;
89
90  /*  determining which month of the year has the most sales and profit. */
91
92  • select MONTH(date) as month_number,
93      MONTHNAME(MIN(date)) as month_name,
94      SUM(total) as total_sales,
95      SUM(gross_income) as total_profit
96  from amazondata
97  group by MONTH(date)
98  order by total_sales DESC, total_profit DESC
99  limit 1;
100
101
```

Result Grid

 Filter Rows:

Export: 

Wrap Cell Content: 

Fetch rows: 

	month_number	month_name	total_sales	total_profit
▶	1	January	116292.11	5537.95

Business Questions to answer:

```
Data_wrangling_And_Feature_e_ Business_questions_to_answer
Limit to 1000 rows

1 • use amazon;
2
3 /*1. What is the count of distinct cities in the dataset? */
4
5 • select count(distinct city) as count_of_distinct_cities
6   from amazondata;
7
8 /* 2.For each branch, what is the corresponding city? */

Result Grid | Filter Rows: | Exports: | Wrap Cell Contents:
count_of_distinct_cities
3
```

```
9
10 • select distinct branch, city from amazondata;
11
12 /*3. What is the count of distinct product lines in the d
13
14 • select count(distinct product_line) as count_of_distinct_

Result Grid | Filter Rows: | Exports: | Wrap Cell Contents:
branch city
A Yangon
C Naypyitaw
B Mandalay
```



```

12  /*3. What is the count of distinct product lines in the dataset? */
13
14 • select count(distinct product_line) as count_of_distinct_product_line
15 from amazondata;
16
17 /* 4.Which payment method occurs most frequently? */
18
19 • select payment method, count(payment method) AS No of times method u

```

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

	count_of_distinct_product_lines
▶	6

```

17 /* 4.Which payment method occurs most frequently? */
18 • select payment_method, count(payment_method) AS No_of_times_method_used
19 FROM amazondata
20 GROUP BY payment_method
21 ORDER BY No_of_times_method_used DESC
22 LIMIT 1;






```

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


	payment_method	No_of_times_method_used
▶	Ewallet	345



```
24  /* 5.Which product line has the highest sales? */
25  • select product_line, SUM(total) as total_sales
26  from amazondata
27  group by product_line
28  order by total_sales desc
29  limit 1;
```

Result Grid		 Filter Rows: <input type="text"/>	Export: 	Wrap Cell Content: 	Fetch rows: 
	product_line	total_sales			
▶	Food and beverages	56144.96			



```
31  /* 6.How much revenue is generated each month? */
32  • select MONTH(date) as month_number,
33          MONTHNAME(Min(date)) as month_name,
34          SUM(total) as monthly_revenue
35  from amazondata
36  group by month(date)
37  order by month_number;
```

Result Grid		 Filter Rows: <input type="text"/>	Export: 	Wrap Cell Content: 
	month_number	month_name	monthly_revenue	
▶	1	January	116292.11	
	2	February	97219.58	
	3	March	109455.74	

```

39  /* 7.In which month did the cost of goods sold reach its peak? */
40 • select MONTH(date) as month_number,
41       MONTHNAME(min(date)) as month_name,
42       SUM(cogs) as total_cogs
43 from amazodata
44 group by MONTH(date)
45 order by total_cogs desc
46 limit 1;

```






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

	month_number	month_name	total_cogs
▶	1	January	110754.16

```

48  /* 8.Which product line generated the highest revenue? */
49 • select product_line, SUM(total) as total_revenue
50 from amazodata group by product_line
51 order by total_revenue DESC
52 LIMIT 1;

```

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

	product_line	total_revenue
▶	Food and beverages	56144.96


```

54  /* 9.In which city was the highest revenue recorded? */
55  • select city,SUM(total) as total_revenue
56  from amazondata group by city
57  order by total_revenue DESC
58  LIMIT 1;
59
60  /* 10.Which product line incurred the highest Value Added

```

Result Grid		Filter Rows: <input type="text"/>	Export:	Wrap Cell Content:	Fetch rows:
	city	total_revenue			
▶	Naypyitaw	110568.86			

```

60  /* 10.Which product line incurred the highest Value Added Tax? */
61  • select product_line,SUM(vat) as total_vat
62  from amazondata
63  group by product_line
64  ORDER BY total_vat DESC
65  LIMIT 1;
66

```

Result Grid		Filter Rows: <input type="text"/>	Export:	Wrap Cell Content:	Fetch rows:
	product_line	total_vat			
▶	Food and beverages	2673.5640			

```

67  /* 11.For each product line, add a column indicating "Good" if its sales are above average,
68  otherwise "Bad." */
69  • select product_line,SUM(total) AS total_sales,
70  if(SUM(total) > (select AVG(total) from amazondata), 'Good', 'Bad') as sales_position
71  from amazondata
72  group by product_line;
73

```

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

	product_line	total_sales	sales_position
▶	Health and beauty	49193.84	Good
	Electronic accessories	54337.64	Good
	Home and lifestyle	53861.96	Good
	Sports and travel	55123.00	Good
	Food and beverages	56144.96	Good
	Fashion accessories	54306.03	Good

```

74  /* 12.Identify the branch that exceeded the average number of products sold. */
75  • select branch, SUM(quantity) as total_quantity_sold
76  from amazondata
77  group by branch
78  having total_quantity_sold > (SELECT AVG(quantity) FROM amazondata);
79

```

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

	branch	total_quantity_sold
▶	A	1859
	C	1831
	B	1820

```

80  /* 13.Which product line is most frequently associated with each gender? */
81 • select gender,product_line,COUNT(*) as frequency
82   from amazondata
83  group by gender, product_line
84  order by gender, frequency desc
85  limit 1;
86
87  /* 14.Calculate the average rating for each product line. */

```

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

	gender	product_line	frequency
▶	Female	Fashion accessories	96

```

87  /* 14.Calculate the average rating for each product line. */
88 • select product_line, avg(rating) as average_rating
89   from amazondata group by product line;

```

Result Grid   Filter Rows: Export:  Wrap Cell Content: 

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

```
91  /* 15.Count the sales occurrences for each time of day on every weekday. */
92  • select DAYNAME(date) as weekday,time(time) as time_of_day,
93     COUNT(*) as sales_occurrences
94  from amazondata
95  group by weekday, time_of_day
96  order by weekday, time_of_day;
```

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

	weekday	time_of_day	sales_occurrences
►	Friday	10:08:00	1
	Friday	10:11:00	2
	Friday	10:12:00	1
	Friday	10:22:00	1
	Friday	10:25:00	1
	Friday	10:26:00	1
	Friday	10:29:00	1
	Friday	10:36:00	2
	Friday	10:37:00	2
	Friday	10:43:00	1
	Friday	10:49:00	1
	Friday	10:50:00	1
	Friday	10:53:00	2
	Friday	10:58:00	1
	Friday	11:05:00	1
	Friday	11:15:00	1
	Friday	11:20:00	1
	Friday	11:21:00	1
	Friday	11:26:00	1
	Friday	11:27:00	1
	Friday	11:30:00	1
	Friday	11:30:00	1


```

98      /* 16. Identify the customer type contributing the highest revenue. */
99 •    select customer_type, SUM(total) as total_revenue
100    from amazondata
101   group by customer_type
102  order by total_revenue desc
103  limit 1;
104
105      /* 17. Determine the city with the highest VAT percentage. */

```

Result Grid		Filter Rows: <input type="text"/>	Export:	Wrap Cell Content:	Fetch rows:
	customer_type	total_revenue			
▶	Member	164223.81			

```

105      /* 17. Determine the city with the highest VAT percentage. */
106 •    select city, AVG(VAT) as average_VAT
107    from amazondata
108   group by city
109  order by average_VAT desc
110  limit 1;
111

```

Result Grid		Filter Rows: <input type="text"/>	Export:	Wrap Cell Content:	Fetch rows:
	city	average_VAT			
▶	Naypyitaw	16.05236732			

```

112  /* 18. Identify the customer type with the highest VAT payments. */
113  • select customer_type, SUM(VAT) as VAT_payments
114    from amazondata
115   group by customer_type
116  order by VAT_payments desc
117  LIMIT 1;

```

Result Grid		Filter Rows: <input type="text"/>	Export:	Wrap Cell Content:	Fetch rows:
	customer_type	VAT_payments			
▶	Member	7820.1640			

```

119  /* 19. What is the count of distinct customer types in the dataset? */
120  • select COUNT(distinct customer_type) as count_of_distinct_customer_types
121    from amazondata;
122
123  /* 20. What is the count of distinct payment methods in the dataset? */

```






Result Grid		Filter Rows: <input type="text"/>	Export:	Wrap Cell Content:
	count_of_distinct_customer_types			
▶	2			


```
123  /* 20.What is the count of distinct payment methods in the dataset? */
124 • select COUNT(distinct payment_method) as count_of_distinct_payment_methods
125 from amazondata;
126
```

Result Grid   Filter Rows: Export:  Wrap Cell Content: 






	count_of_distinct_payment_methods
▶	3

```
127  /* 21.Which customer type occurs most frequently? */
128 • select customer_type, COUNT(customer_type) as frequency
129 from amazondata
130 group by customer_type
131 order by frequency desc limit 1;
132
```

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





	customer_type	frequency
▶	Member	501

```
133  /*22.Identify the customer type with the highest purchase frequency.*/
134 • select customer_type, COUNT(customer_type) as purchase_frequency
135 from amazodata
136 group by customer_type
137 order by purchase_frequency desc limit 1;
138
```

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

	customer_type	purchase_frequency
▶	Member	501

```
139  /* 23.Determine the predominant gender among customers. */
140 • select gender, COUNT(*) as gender_count
141 from amazodata group by gender
142 order by gender_count desc
143 limit 1;
144
```


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

	gender	gender_count
▶	Female	501

```

145  /* 24.Examine the distribution of genders within each branch.*/
146 • select branch, gender, COUNT(*) as gender_count
147 from amazondata
148 group by branch, gender
149 order by branch, gender;
150

```


Result Grid   Filter Rows: Export:  Wrap Cell Content: 

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

```

151  /* 25.Identify the time of day when customers provide the most ratings.*/
152 • select TIME(time) as time_of_day, COUNT(*) AS rating_count
153 from amazondata
154 group by time_of_day
155 order by rating_count desc
156 limit 1;

```

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

	time_of_day	rating_count
▶	19:48:00	7

```




158  /* 26.Determine the time of day with the highest customer ratings for each branch.*/
159 • select branch, TIME(time) as time_of_day, COUNT(*) as rating_count
160 from amazondata
161 GROUP BY branch, time_of_day
162 ORDER BY branch, rating_count DESC;
163

```

Result Grid   Filter Rows: Export:  Wrap Cell Content: 

	branch	time_of_day	rating_count
▶	A	13:34:00	3
	A	19:44:00	3
	A	12:43:00	3
	A	15:48:00	3
	A	11:51:00	3
	A	11:00:00	2
	A	11:40:00	2
	A	10:37:00	2
	A	14:36:00	2
	A	11:32:00	2
	A	17:15:00	2
	A	17:43:00	2
	A	12:52:00	2
	A	16:10:00	2
	A	11:44:00	2
	A	14:33:00	2
	A	17:01:00	2
	A	16:06:00	2
	A	10:39:00	2
	A	17:59:00	2
	A	10:55:00	2
	A	19:30:00	2
	A	18:14:00	2
	A	12:55:00	2
	A	16:23:00	2
	A	17:24:00	2

```
166  /* 27.Identify the day of the week with the highest average ratings.*/
167 • select DAYNAME(date) as day_of_week, AVG(rating) as average_rating
168    from amazodata
169   group by day_of_week
170  order by average_rating desc limit 1;
--
```



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

	day_of_week	average_rating
▶	Monday	7.153599990844727


```

172  /*28.Determine the day of the week with the highest average ratings for each branch.*/
173  • select branch, DAYNAME(date) as day_of_week, AVG(rating) as average_rating
174  from amazondata
175  group by branch, day_of_week
176  order by branch, average_rating desc;
177

```

Result Grid   Filter Rows: Export:  Wrap Cell Content: 

	branch	day_of_week	average_rating
▶	A	Friday	7.311999988555908
	A	Monday	7.097916682561238
	A	Sunday	7.078846133672274
	A	Tuesday	7.058823510712268
	A	Thursday	6.958695639734683
	A	Wednesday	6.9162790941637615
	A	Saturday	6.74600001335144
	B	Monday	7.335897384545742
	B	Tuesday	7.0018867726595895
	B	Sunday	6.888571412222726
	B	Thursday	6.75227270343087
	B	Saturday	6.7366667032241825
	B	Friday	6.694117648928773
	B	Wednesday	6.451999950408935
	C	Friday	7.278947328266344
	C	Saturday	7.229629649056329
	C	Wednesday	7.064000034332276
	C	Monday	7.036842107772827
	C	Sunday	7.028260852979577
	C	Tuesday	6.951851844787598
	C	Thursday	6.949999998013179

Analysis

- Product Analysis:
performance of different product lines, determine which product lines are performing well, and least.

```
3  /* Top-performing product lines based on total sales */
4  • select product_line, SUM(total) as total_sales
5  from amazodata
6  group by product_line
7  order by total_sales desc;
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	product_line	total_sales			
▶	Food and beverages	56144.96			
	Sports and travel	55123.00			
	Electronic accessories	54337.64			
	Fashion accessories	54306.03			
	Home and lifestyle	53861.96			
	Health and beauty	49193.84			

```

9  /* Least performing product lines with lowest sales to identify
10 areas of improvement */
11 • select product_line, SUM(total) as total_sales
12 from amazodata
13 group by product_line
14 order by total_sales asc;
15

```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	product_line	total_sales			
▶	Health and beauty	49193.84			
	Home and lifestyle	53861.96			
	Fashion accessories	54306.03			
	Electronic accessories	54337.64			
	Sports and travel	55123.00			
	Food and beverages	56144.96			


- Sales Analysis:

```
16  /* Total sales per month */
17 • select MONTH(date) as month, SUM(total) as total_sales
18   from amazondata group by month
19   order by month;
20
```

Result Grid   Filter Rows: Export:  Wrap Cell Content: 

	month	total_sales
▶	1	116292.11
	2	97219.58
	3	109455.74

```
21  /* Total sales per branch */
22 • select branch, SUM(total) as total_sales
23   from amazondata group by branch
24   order by total_sales desc;
```

Result Grid   Filter Rows: Export:  Wrap Cell Content: 

	branch	total_sales
▶	C	110568.86
	A	106200.57
	B	106198.00

```
26  /*Total sales by payment method*/
27  • select payment_method, SUM(total) as total_sales
28  from amazondata group by payment_method
29  order by total_sales desc;
```

Result Grid   Filter Rows: Export:  Wrap Cell Content: 

	payment_method	total_sales
▶	Cash	112206.76
	Ewallet	109993.38
	Credit card	100767.29

• Customer Analysis:

```
37  /*Total amount spent by customer type and How many times customer type purchased*/
38  •  select customer_type, SUM(total) as total_amount, COUNT(*) as purchase_count
39      from amazondata group by customer_type
40      order by total_amount desc;
41
```




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

	customer_type	total_amount	purchase_count
▶	Member	164223.81	501
	Normal	158743.62	499

Data_wrangling_And_Feature_e... Business_questions_to_answer Analysis_list x

         Limit to 1000 rows     

```
37
38  /*Top performing branches based on customer rating*/
39  •  select branch, AVG(rating) as avg_rating
40      from amazondata
41      group by branch order by avg_rating desc;
42
```

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

	branch	avg_rating
▶	C	7.072865853949291
	A	7.027058821565965
	B	6.81807227450681

Insights and Summary

Product analysis:

The top-performing product lines are dominated by the “Food and Beverages”, with total sales of 56,122.96, respectively On the other hand, the least performing product lines include Health and beauty, Home and lifestyle, and Fashion accessories, with total sales ranging from 49,193.84 to 56,144.96. These product lines may require attention and improvement strategies to boost sales.

Sales Analysis:

Monthly sales are relatively consistent, with the highest sales in January (116,292.11) and the lowest in February (97,219.58).- Branch C leads in terms of total sales (110,568.86), followed closely by Branch A and Branch B.- Cash payments account for the largest share of total sales (112,206.76), followed by E-wallet payments (109,993.38) and Credit Card payments (100,767.29).

Customer Analysis:

Members are the top-spending customer type, with a total spend of 164,223.81 and 501 purchases, closely followed by Normal customers with a total spend of 158,743.62 and 499 purchases.- Branch C is the top-performing branch based on customer ratings, with an average rating of 7.07, followed by Branch A (7.03) and Branch B (6.82).

Nikhil Karaka

THANK
YOU!