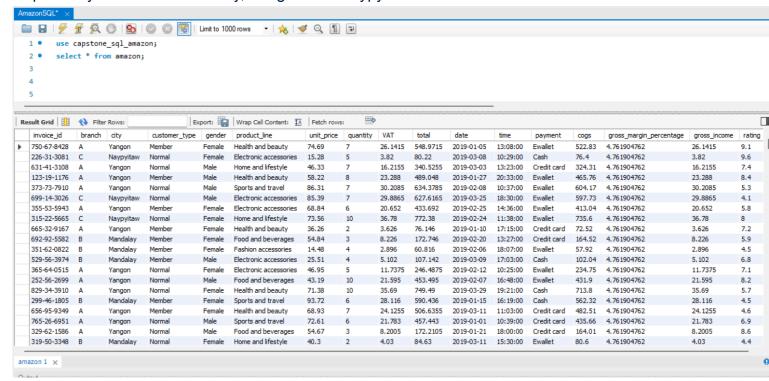
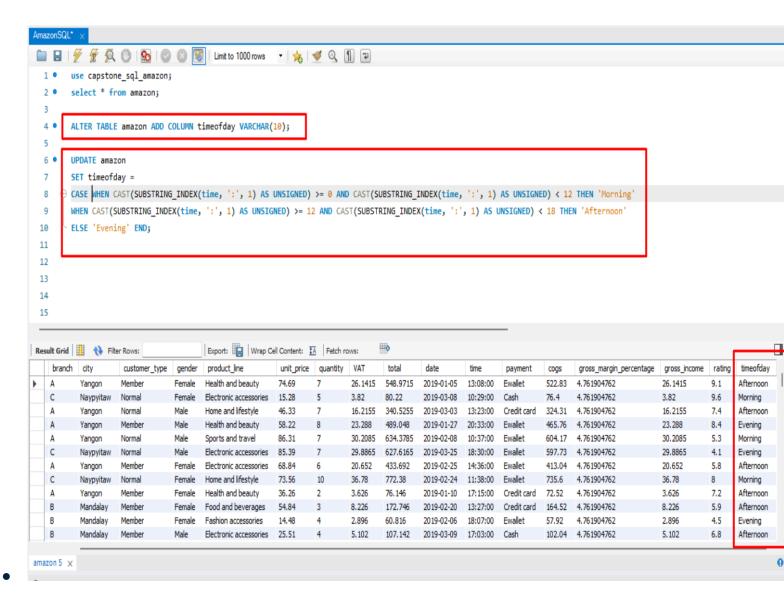
## **SQL - Capstone Project**

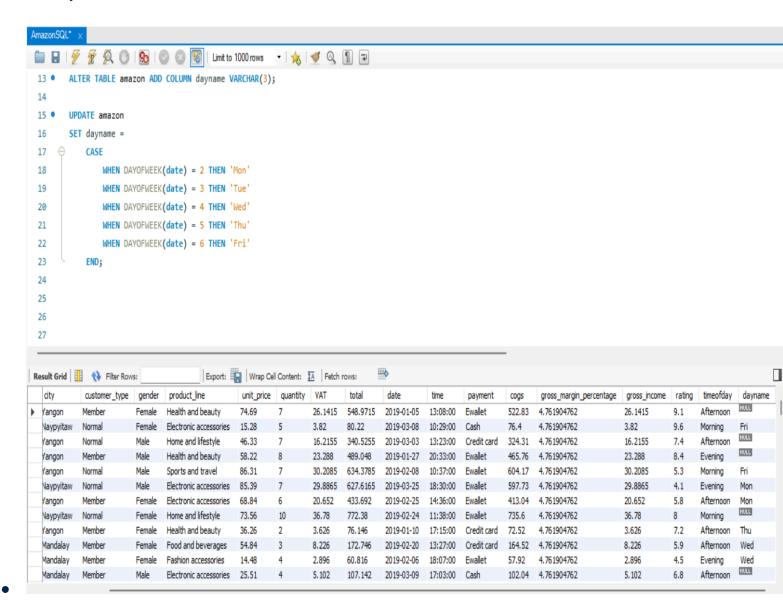
- AIM: The major aim of this project is to gain insight into the sales data of Amazon to understand the
  different factors that affect sales of the different branches.
- About Data: This dataset contains sales transactions from three different branches of Amazon, respectively located in Mandalay, Yangon and Naypyitaw. The data contains 17 columns and 1000 rows.



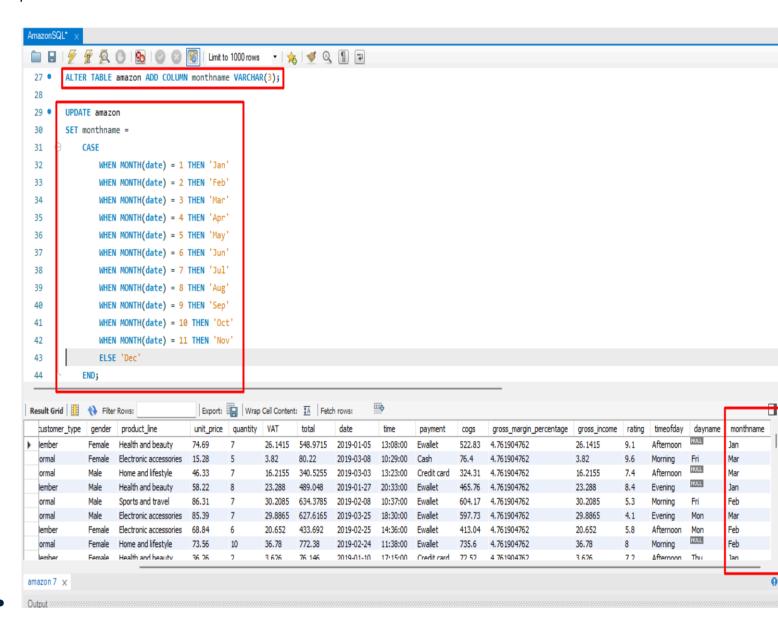
- Feature Engineering: This will help us generate some new columns from existing ones.
- Add a new column named timeofday to give insight of sales in the Morning, Afternoon and Evening. This
  will help answer the question on which part of the day most sales are made.



 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.

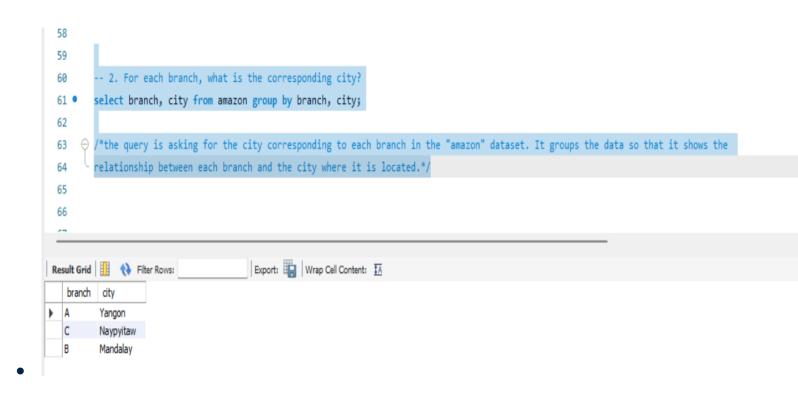


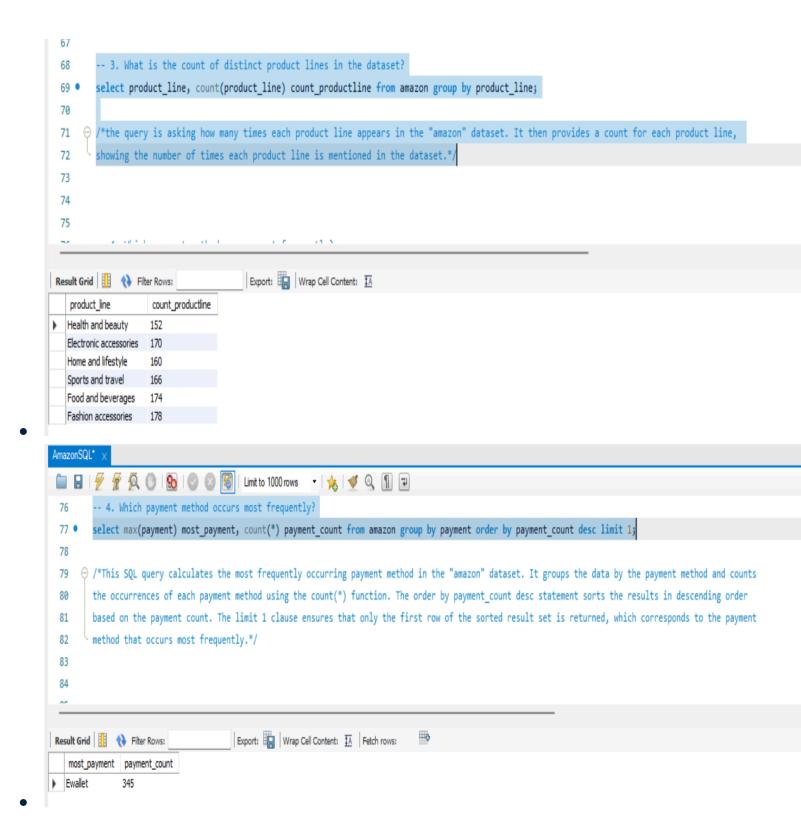
 Add a new column named monthname that contains the extracted months of the year on which the given transaction took place (Jan, Feb, Mar). Help determine which month of the year has the most sales and profit.

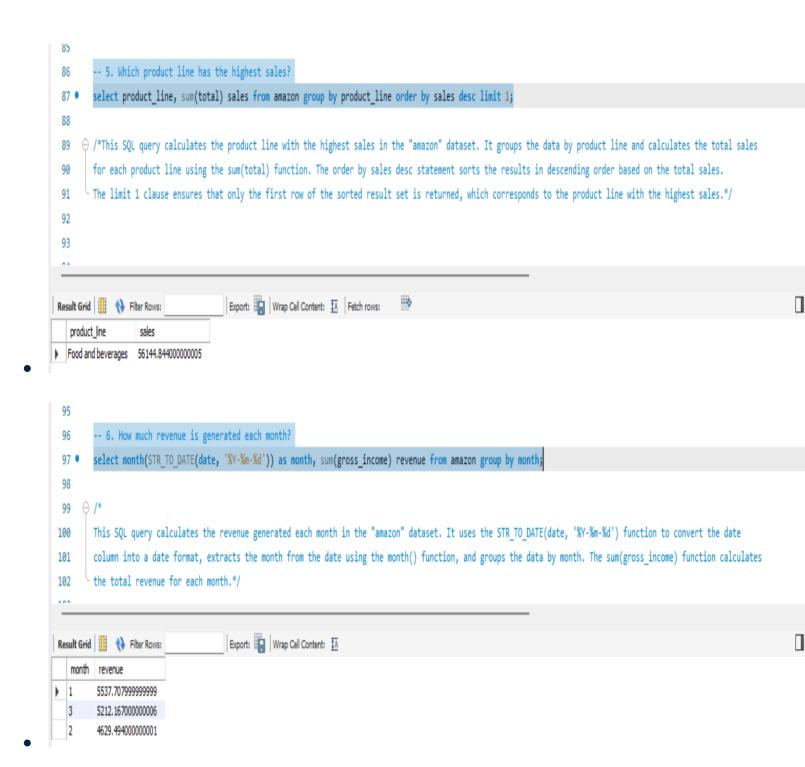


## Business Questions To Answer:

```
-- 1. What is the count of distinct cities in the dataset?
52
      select city, count(city) count_cities from amazon group by city;
53 •
54
    55
      The count(city) function is used to count the occurrences of each city, and the result is stored in a column named count_cities.
56
      The group by city statement groups the data by city so that the count is calculated for each unique city in the dataset.*/
57
58
59
                               Export: Wrap Cell Content: IA
count_cities
  city
 Yangon
          340
 Naypyitaw
         328
 Mandalay
         332
```

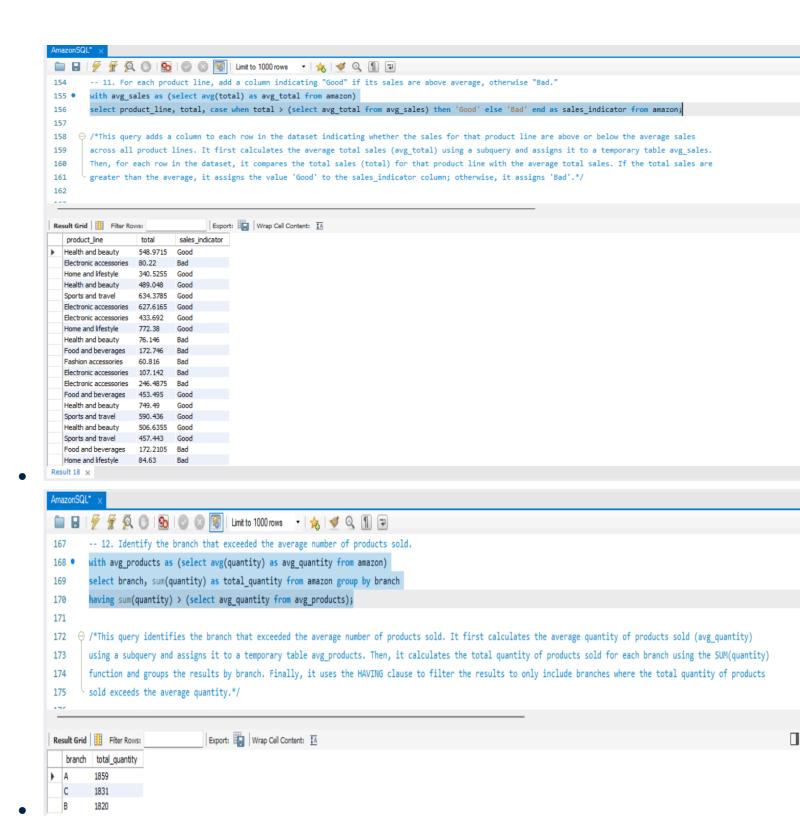


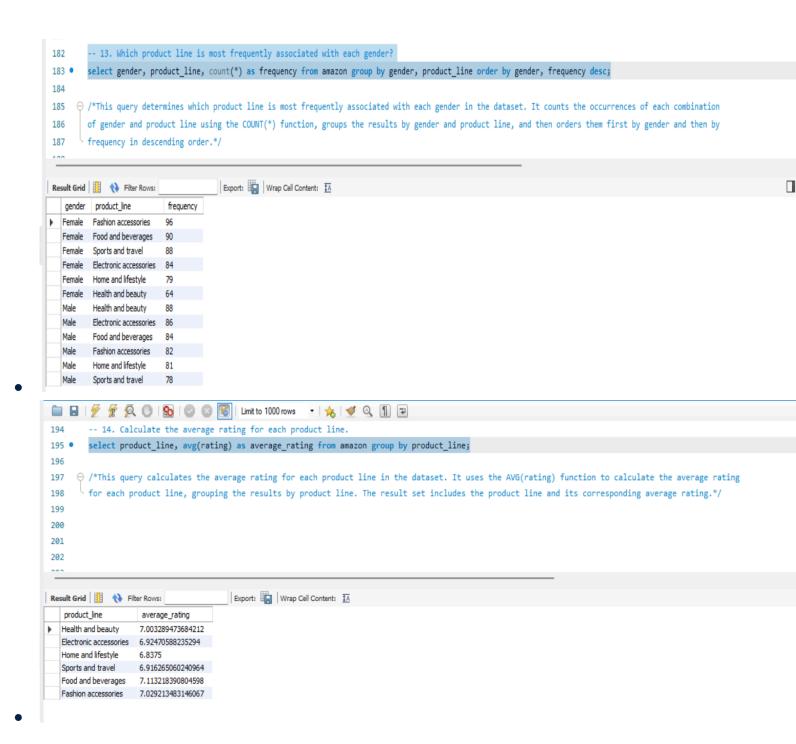


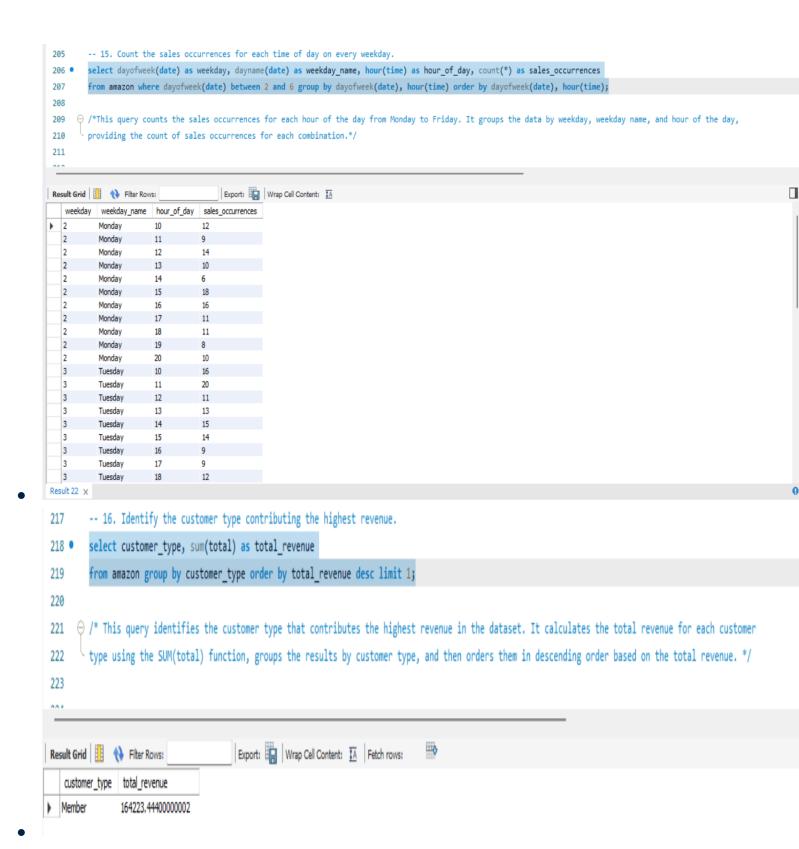


```
-- 7. In which month did the cost of goods sold reach its peak?
107
108
         select month(STR_TO_DATE(date, '%Y-%m-%d')) as month, sum(total) revenue from amazon group by month order by revenue desc limit 1;
109
      😔 /*This SQL query calculates the month when the cost of goods sold (COGS) reached its peak in the "amazon" dataset. It converts the date column into a date format,
110
         extracts the month from the date, and groups the data by month. The sum(total) function calculates the total revenue for each month.
111
        The order by revenue desc statement sorts the results in descending order based on the total revenue. */
112
113
114
                                        Export: Wrap Cell Content: A Fetch rows:
Result Grid Filter Rows:
   month
         revenue
         116291.86800000005
1
          -- 8. Which product line generated the highest revenue?
118
         select product_line, sum(unit_price * quantity) revenue from amazon group by product_line order by revenue desc limit 1;
119 •
120
         /*the query is asking which product line generated the highest total revenue in the "amazon" dataset, and it provides the total revenue for that product line.*/
121
 122
123
                                                                                                                                                                               Export: Wrap Cell Content: A Fetch rows:
product_line
▶ Food and beverages 53471.28000000006
128
          -- 9. In which city was the highest revenue recorded?
         select city, sum(gross_income) as total_revenue from amazon group by city order by total_revenue desc limit 1;
129
 130
 131
      ⊖ /*
 132
         This query finds the city where the highest total revenue was recorded in the dataset. It calculates the total revenue for each
 133
         city by summing the gross_income column, groups the results by city, and then orders them in descending order based on the total revenue.
         The LIMIT 1 clause ensures that only the city with the highest total revenue is returned.*/
 134
 135
                                        Export: Wrap Cell Content: 🔼 Fetch rows:
                                                                                                                                                                               П
Result Grid ## Filter Rows:
Naypyitaw
            5265, 176500000002
 140
          -- 10. Which product line incurred the highest Value Added Tax?
 141
         select product_line, sum(vat) as total_vat from amazon group by product_line order by total_vat desc limit 1;
 142 •
 143
 144
         This query identifies the product line that incurred the highest Value Added Tax (VAT). It calculates the total VAT for each product line by
 145
         summing the vat column, groups the results by product line, and then orders them in descending order based on the total VAT. The LIMIT 1 clause ensures
 146
         that only the product line with the highest total VAT is returned.*/
 147
Export: Wrap Cell Content: 🔼 Fetch rows:
    product_line

    Food and beverages 2673,5639999999994
```







```
228
229
          -- 17. Determine the city with the highest VAT percentage.
         select city, sum(vat) as total_vat, sum(total) as total_sales, (sum(vat) / sum(total)) * 100 as vat percentage
230
         from amazon group by city order by vat_percentage desc limit 1;
231
232
233
         This query determines the city with the highest Value Added Tax (VAT) percentage in the dataset. It calculates the total VAT and total sales for
234
         each city using the SUM(vat) and SUM(total) functions, respectively. Then, it calculates the VAT percentage for each city by dividing the total VAT
235
         by the total sales and multiplying by 100. The results are then ordered in descending order based on the VAT percentage*/
236
                                                                                     Result Grid Filter Rows:
                                          Export: Wrap Cell Content: TA Fetch rows:
             total vat
                              total sales
                                                vat percentage
Mandalay
            5057.032000000003
                              106197.67199999996 4.761904761904766
  241
            Execute the selected portion of the script or everything, if there is no selection
  242
          -- 18. Identify the customer type with the highest VAT payments.
  243 •
          select customer_type, sum(vat) as total_vat_payments
  244
          from amazon group by customer_type order by total_vat_payments desc limit 1;
  245
  246

⊕ /*This query identifies the customer type that has the highest VAT payments in the dataset. It calculates the total VAT payments for each customer type

         using the SUM(vat) function, groups the results by customer type, and then orders them in descending order based on the total VAT payments.*/
  247
  248
  Export: Wrap Cell Content: TA Fetch rows:
     customer_type total_vat_payments
 Member
                 7820.164000000002
          -- 19. What is the count of distinct customer types in the dataset?
 254
          select count(distinct customer_type) as distinct_customer_types from amazon;
 255
 256
      ⊖ /*
 257
          This query calculates the count of distinct customer types in the dataset. It uses the COUNT(DISTINCT customer type) function to count the number of
 258
         unique customer types in the "amazon" dataset and assigns the result to a column named distinct_customer_types.*/
 259
 260
Result Grid Filter Rows:
                                          Export: Wrap Cell Content: $\overline{1}{4}$
    distinct_customer_types
) 2
```

```
-- 20. What is the count of distinct payment methods in the dataset?
 266
         select count(distinct payment) as distinct_payment_methods from amazon;
 267
 268

→ /*This query calculates the count of distinct payment methods in the dataset. It uses the COUNT(DISTINCT payment) function to count the number of

 269
        ^{\circ} unique payment methods in the "amazon" dataset and assigns the result to a column named distinct payment methods.^*/
 270
 271
 272
                                       Export: Wrap Cell Content: IA
distinct_payment_methods
) 3
         -- 21. Which customer type occurs most frequently?
277
         select customer type, count(*) as frequency
278
         from amazon group by customer_type order by frequency desc limit 1;
279
280
 281
 282
        This query identifies the customer type that occurs most frequently in the dataset. It counts the occurrences of each customer type using the
        COUNT(*) function, groups the results by customer type, and then orders them in descending order based on the frequency.*/
 283
 284
                                      Export: Wrap Cell Content: 🔣 Fetch rows:
customer_type
               frequency
Member
               501
       288
289
        -- 22. Identify the customer type with the highest purchase frequency.
        select customer_type, count(*) as purchase_frequency
290 •
291
        from amazon group by customer_type order by purchase_frequency desc limit 1;
292
     ⊕ /*This query identifies the customer type with the highest purchase frequency in the dataset. It counts the number of purchases made by each customer type using
293
      the COUNT(*) function, groups the results by customer type, and then orders them in descending order based on the purchase frequency.*/
294
295
296
                                                                                                                                                             П
Export: Wrap Cell Content: A Fetch rows:
  customer_type purchase_frequency
Member
              501
```

```
300
         -- 23. Determine the predominant gender among customers.
301
         select gender, count(*) as customer_count
         from amazon group by gender order by customer_count desc limit 1;
302
303
304
305
         This query determines the predominant gender among customers in the dataset. It counts the occurrences of each gender
306
         using the COUNT(*) function, groups the results by gender, and then orders them in descending order based on the customer count.*/
307
308
                                                                                     4
                                         Export: Wrap Cell Content: A Fetch rows:
gender customer_count
Female 501
          -- 24. Examine the distribution of genders within each branch.
 312
         select branch, gender, count(*) as gender_count
 313 •
 314
         from amazon group by branch, gender order by branch, gender;
 315
 316
         This query examines the distribution of genders within each branch in the dataset. It counts the occurrences of each gender within each branch
 317
 318
         using the COUNT(*) function, groups the results by branch and gender, and then orders them first by branch and then by gender.*/
 319
 320
                                        Export: Wrap Cell Content: IA
branch gender gender_count
   Α
          Male
                 179
   В
                162
          Female
   В
          Male 170
   c
          Female 178
   С
          Male 150
        -- 25. Identify the time of day when customers provide the most ratings.
325
        select hour(time) as hour_of_day,
326 •
327
        count(*) as rating_count from amazon where rating is not null group by hour(time) order by rating_count desc limit 1;
328
      /*This query identifies the time of day when customers provide the most ratings in the dataset. It extracts the hour of the day from the time column
329
        using the HOUR(time) function, counts the occurrences of ratings (where the rating is not null), groups the results by the hour of the day, and then orders
330
        them in descending order based on the rating count.*/
331
332
Result Grid Filter Rows:
                                     Export: Wrap Cell Content: TA Fetch rows:
   hour_of_day rating_count
19
             113
```

