

--NTSHEMBO MALUEKE

--Practical 3

Q1

The screenshot shows a SQL IDE interface with a dark theme. The top bar contains tabs for 'Practical 1', 'Practical 2', 'Practical 3', 'Bright_Coffee_Case_study 1', and 'BrightTV_Case_study 2'. The 'Practical 3' tab is active. Below the tabs, there's a toolbar with 'ACCOUNTADMIN', 'COMPUTE_WH (X-Small)', 'Share', and a play button. The main editor area shows a SQL query for Q1. The query is as follows:

```
1  --NTSHEMBO MALUEKE
2  --Practical 3
3
4  -----
5  --1.Find all records where Size is missing and the purchase_amount is greater than 50.
6  --Expected Columns: Customer ID, Size, purchase_amount, Item Purchased
7
8  select customer_id,
9         size,
10        purchase_amount,
11        item_purchased
12  from shopping_tread
13  where size is null
14        and purchase_amount > 50;
15
```

Below the query editor, there's a 'Results' tab selected. It shows a table with 5 rows and 4 columns: # CUSTOMER_ID, SIZE, # PURCHASE_AMOUNT, and ITEM_PURCHASED.

	# CUSTOMER_ID	SIZE	# PURCHASE_AMOUNT	ITEM_PURCHASED
1	11	null	74.0	Handbag
2	15	null	54.0	Jeans
3	22	null	88.0	Shirt
4	32	null	54.0	Blouse
5	62	null	57.0	Blouse

Q2

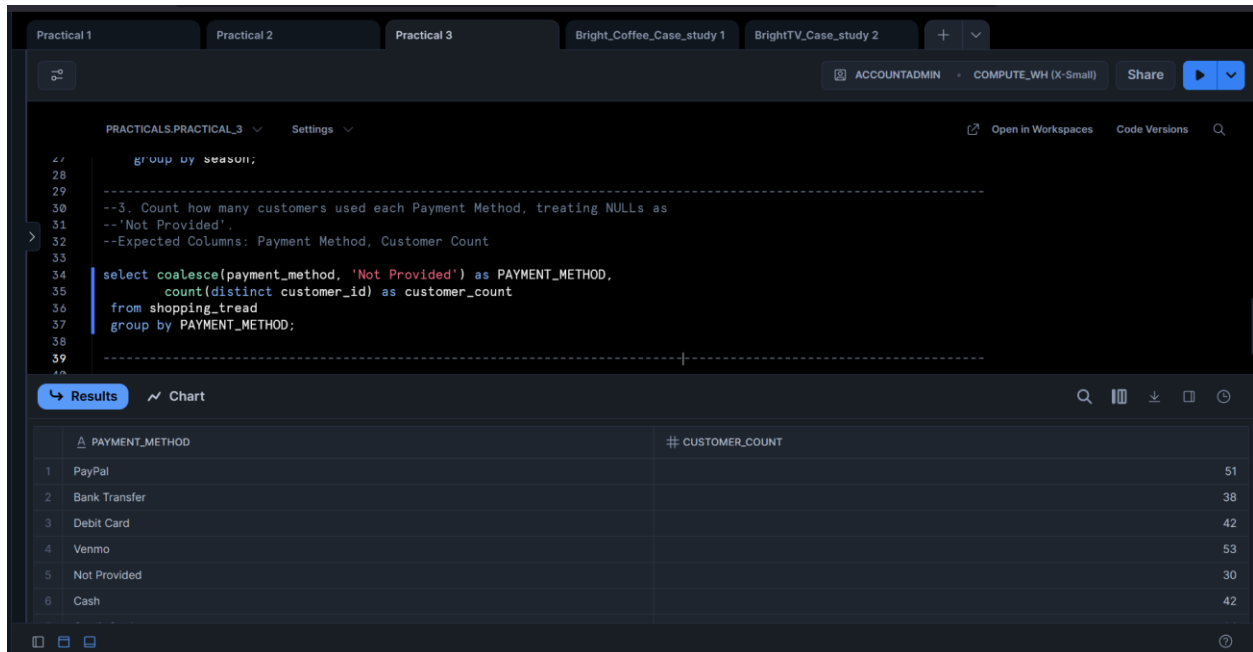
The screenshot shows the same SQL IDE interface as before, but with a different query for Q2. The query is as follows:

```
15  -----
16
17  --2. List the total number of purchases grouped by Season, treating NULL values as 'Unknown Season'.
18  --Expected Columns: Season, Total Purchases
19
20  select
21        coalesce(season, 'unknown Season') as season,
22        count(item_purchased) as total_purchase
23  from shopping_tread
24  group by season;
25
26  -----
```

Below the query editor, there's a 'Results' tab selected. It shows a table with 5 rows and 2 columns: SEASON and # TOTAL_PURCHASE.

	SEASON	# TOTAL_PURCHASE
1	Summer	58
2	Winter	71
3	unknown Season	26
4	Fall	50
5	Spring	66

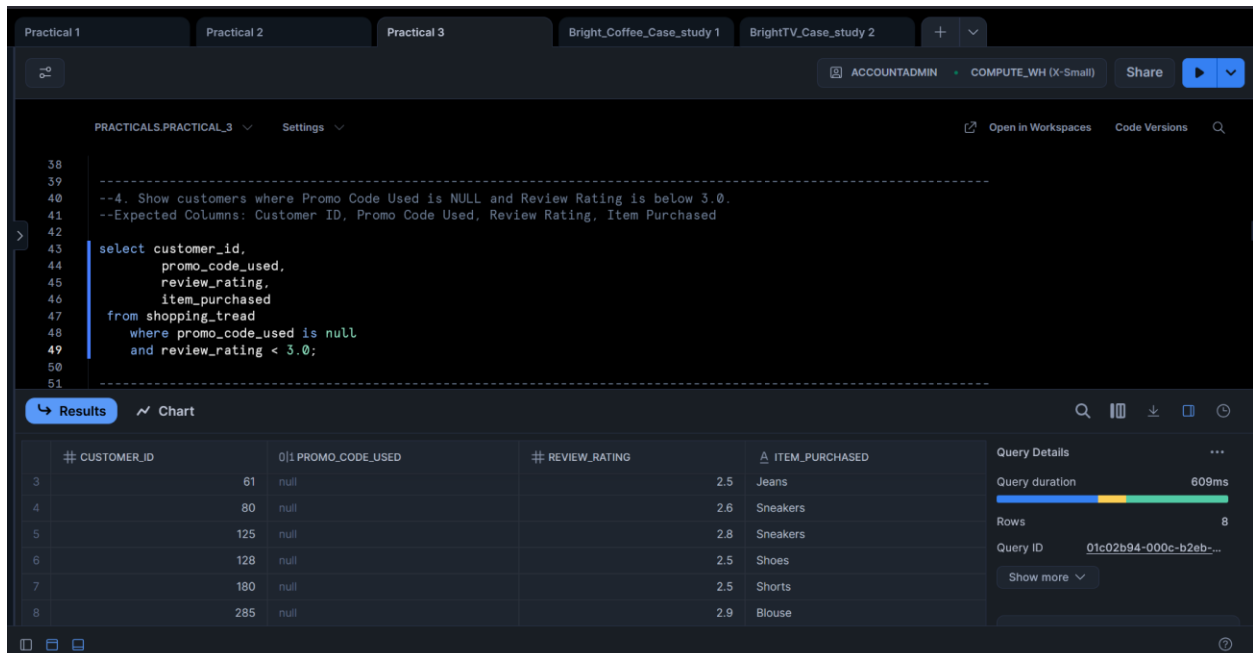
Q3



```
group by season;
-----
--3. Count how many customers used each Payment Method, treating NULLs as
--'Not Provided'
--Expected Columns: Payment Method, Customer Count
select coalesce(payment_method, 'Not Provided') as PAYMENT_METHOD,
       count(distinct customer_id) as customer_count
from shopping_tread
group by PAYMENT_METHOD;
-----
```

	PAYMENT_METHOD	# CUSTOMER_COUNT
1	PayPal	51
2	Bank Transfer	38
3	Debit Card	42
4	Venmo	53
5	Not Provided	30
6	Cash	42

Q4



```
-----
--4. Show customers where Promo Code Used is NULL and Review Rating is below 3.0.
--Expected Columns: Customer ID, Promo Code Used, Review Rating, Item Purchased
select customer_id,
       promo_code_used,
       review_rating,
       item_purchased
from shopping_tread
where promo_code_used is null
and review_rating < 3.0;
-----
```

	# CUSTOMER_ID	PROMO_CODE_USED	# REVIEW_RATING	ITEM_PURCHASED
3	61	null	2.5	Jeans
4	80	null	2.6	Sneakers
5	125	null	2.8	Sneakers
6	128	null	2.5	Shoes
7	180	null	2.5	Shorts
8	285	null	2.9	Blouse

Query Details
Query duration: 609ms
Rows: 8
Query ID: 01c02b94-000c-b2eb-...
[Show more](#)

Q5

The screenshot shows a SQL IDE interface with a dark theme. At the top, there are tabs for 'Practical 1', 'Practical 2', 'Practical 3', 'Bright_Coffee_Case_study 1', and 'BrightTV_Case_study 2'. The 'Practical 3' tab is active. Below the tabs, there is a toolbar with a search icon, 'ACCOUNTADMIN', 'COMPUTE_WH (X-Small)', 'Share', and a play button. The main area displays a SQL query in a code editor. The query is as follows:

```
--
50
51
52 -----
53 --5. Group customers by Shipping Type, and return the average purchase_amount, treating missing values as 0.
54 --Expected Columns: Shipping Type, Average purchase_amount
55
56 select shipping_type,
57        avg(coalesce(purchase_amount, 0)) as Average_purchase_amount
58 from shopping_tread
59 group by shipping_type;
60
61 -----
```

Below the code editor, there is a 'Results' tab selected, showing a table with 7 rows and 2 columns. The columns are 'SHIPPING_TYPE' and 'AVERAGE_PURCHASE_AMOUNT'. The data is as follows:

SHIPPING_TYPE	AVERAGE_PURCHASE_AMOUNT
1 Standard	47.666667
2 Express	53.454545
3 Store Pickup	55.333333
4 null	52.703703
5 Free Shipping	50.214285
6 Next Day Air	54.866667
7 2-Day Shipping	51.557692

Q6

The screenshot shows a SQL IDE interface with a dark theme. At the top, there are tabs for 'Practical 1', 'Practical 2', 'Practical 3', 'Bright_Coffee_Case_study 1', and 'BrightTV_Case_study 2'. The 'Practical 3' tab is active. Below the tabs, there is a toolbar with a search icon, 'ACCOUNTADMIN', 'COMPUTE_WH (X-Small)', 'Share', and a play button. The main area displays a SQL query in a code editor. The query is as follows:

```
64
65
66 select location,
67        count(purchase_amount) as total_purchases
68 from shopping_tread
69 where payment_method is not null
70 group by location
71 having count(purchase_amount) > 5;
72
73 -----
74 --7. Create a column Spender Category that classifies customers using CASE:
75 --'High' if amount > 80, 'Medium' if BETWEEN 50 AND 80.
```

Below the code editor, there is a 'Results' tab selected, showing a table with 7 rows and 2 columns. The columns are 'LOCATION' and 'TOTAL_PURCHASES'. The data is as follows:

LOCATION	TOTAL_PURCHASES
1 Maine	34
2 Kentucky	27
3 null	21
4 New York	27
5 Oregon	24
6 Rhode Island	26
7 Florida	29

Q7

Practical 1 Practical 2 Practical 3 Bright_Coffee_Case_study 1 BrightTV_Case_study 2 +

ACCOUNTADMIN COMPUTE_WH (X-Small) Share

PRACTICALS.PRACTICAL_3 Settings

Open in Workspaces Code Versions

```
72 -----
73 --7. Create a column Spender Category that classifies customers using CASE:
74 --'High' if amount > 80, 'Medium' if BETWEEN 50 AND 80,
75 --'Low' otherwise. Replace NULLs in purchase_amount with 0.
76 --Expected Columns: Customer ID, purchase_amount, Spender Category
77 select customer_id,
78        coalesce(purchase_amount,0) as purchase_amount,
79
80        case
81          when coalesce(purchase_amount,0) > 80 then 'High Spender'
82          when coalesce(purchase_amount,0) between 50 and 80 then 'Medium Spender'
83          else 'Low Spender'
84        end as Spender_Category
85 from shopping_tread;
```

Results Chart

#	CUSTOMER_ID	#	PURCHASE_AMOUNT	#	SPENDER_CATEGORY
120		120		99.0	High Spender
121		121		60.0	Medium Spender
122		122		37.0	Low Spender
123		123		0.0	Low Spender
124		124		0.0	Low Spender
125		125		28.0	Low Spender

Query Details

Query duration 57ms

Rows 300

Query ID 01c02bc8-000c-b2eb-0...

Show more

Q8

Practical 1 Practical 2 Practical 3 Bright_Coffee_Case_study 1 BrightTV_Case_study 2 +

ACCOUNTADMIN COMPUTE_WH (X-Small) Share

PRACTICALS.PRACTICAL_3 Settings

Open in Workspaces Code Versions

```
87 --8. Find customers who have no Previous Purchases value but whose Color is not NULL.
88 --Expected Columns: Customer ID, Color, Previous Purchases
89 select customer_id,
90        color,
91        previous_purchases
92 from shopping_tread
93 where previous_purchases is null
94 and color is not null;
95 -----
```

Results Chart

#	CUSTOMER_ID	#	COLOR	#	PREVIOUS_PURCHASES
23		226	Black		null
24		230	Green		null
25		251	Gray		null
26		253	Green		null
27		260	Pink		null
28		271	Yellow		null
29		282	Pink		null
30		283	Green		null
31		290	White		null

Q9

The screenshot shows a SQL IDE interface with a dark theme. At the top, there are tabs for 'Practical 1', 'Practical 2', 'Practical 3', 'Bright_Coffee_Case_study 1', and 'BrightTV_Case_study 2'. The 'Practical 3' tab is active. Below the tabs, there is a toolbar with a search icon, a dropdown menu showing 'ACCOUNTADMIN' and 'COMPUTE_WH (X-Small)', a 'Share' button, and a play button. The main editor area shows a SQL query for Q9. The query is as follows:

```
--9. Group records by Frequency of Purchases and show the total amount spent per group, treating NULL frequencies as 'Unknown'.
--Expected Columns: Frequency of Purchases, Total purchase_amount
96 select
97     coalesce(frequency_of_purchases, 'Unknown') as frequency_of_purchases,
98     sum(purchase_amount) as total_amount_spent_per_group
99
100 from shopping_tread
101
102 group by frequency_of_purchases;
103
104
```

Below the query editor, there is a 'Results' tab and a 'Chart' tab. The 'Results' tab is active, showing a table with two columns: 'FREQUENCY_OF_PURCHASES' and 'TOTAL_AMOUNT_SPENT_PER_GROUP'. The table contains 8 rows of data:

	FREQUENCY_OF_PURCHASES	TOTAL_AMOUNT_SPENT_PER_GROUP
1	Every 3 Months	1749.0
2	Weekly	2184.0
3	Bi-Weekly	2099.0
4	Monthly	1780.0
5	Unknown	1518.0
6	Fortnightly	2033.0
7	Annually	1765.0
8	Quarterly	2541.0

Q10

The screenshot shows a SQL IDE interface with a dark theme. At the top, there are tabs for 'Practical 1', 'Practical 2', 'Practical 3', 'Bright_Coffee_Case_study 1', and 'BrightTV_Case_study 2'. The 'Practical 3' tab is active. Below the tabs, there is a toolbar with a search icon, a dropdown menu showing 'ACCOUNTADMIN' and 'COMPUTE_WH (X-Small)', a 'Share' button, and a play button. The main editor area shows a SQL query for Q10. The query is as follows:

```
--11. Return the top 5 Locations with the highest total purchase_amount, replacing NULLs in amount with 0.
--Expected Columns: Location, Total purchase_amount
112 select location,
113     sum(coalesce(purchase_amount, 0)) as total_purchase_amount
114
115 from shopping_tread
116
117 group by location
118
119 order by total_purchase_amount desc
120
121 limit 5;
122
```

Below the query editor, there is a 'Results' tab and a 'Chart' tab. The 'Results' tab is active, showing a table with two columns: 'LOCATION' and 'TOTAL_PURCHASE_AMOUNT'. The table contains 5 rows of data:

	LOCATION	TOTAL_PURCHASE_AMOUNT
1	Maine	2294.0
2	Florida	1980.0
3	Massachusetts	1899.0
4	Rhode Island	1876.0
5	Kentucky	1798.0

Q11

```
108     count(item_purchased) as total_purchases
109   from shopping_tread
110   where category is not null
111   group by category;
-----
112 --11. Return the top 5 Locations with the highest total purchase_amount, replacing NULLs in amount with 0.
113 --Expected Columns: Location, Total purchase_amount
114
115   select location,
116          sum(coalesce(purchase_amount, 0)) as total_purchase_amount
117   from shopping_tread
118   group by location
119   order by total_purchase_amount desc
120   limit 5;
```

	LOCATION	# TOTAL_PURCHASE_AMOUNT
1	Maine	2294.0
2	Florida	1980.0
3	Massachusetts	1899.0
4	Rhode Island	1876.0
5	Kentucky	1798.0

Query Details

- Query duration: 29ms
- Rows: 5
- Query ID: 01c031c1-000c-b2ea-0...

Q12

```
122 --12. Group customers by Gender and Size, and count how many entries have a NULL Color.
123 --Expected Columns: Gender, Size, Null Color Count
124
125   select gender,
126          size,
127          count_if(color is null) as Null_Color_Count
128   from shopping_tread
129   group by gender,
130          size;
```

	GENDER	SIZE	# NULL_COLOR_COUNT
1	Male	null	6
2	Male	M	7
3	Male	L	6
4	Male	XL	5
5	Male	S	5

Query Details

- Query duration: 25ms
- Rows: 5
- Query ID: 01c031c2-000c-b2ea-0...

GENDER

- Male: 5

Q13

The screenshot shows a SQL IDE interface with a dark theme. At the top, there are tabs for 'Practical 1', 'Practical 2', 'Practical 3', 'Bright_Coffee_Case_study 1', and 'BrightTV_Case_study 2'. The 'Practical 3' tab is active. Below the tabs, there is a toolbar with 'ACCOUNTADMIN', 'COMPUTE_WH (X-Small)', 'Share', and a play button. The main editor area shows a SQL query for Q13. The query is as follows:

```
132 -----
133 --13. Identify all Item Purchased where more than 3 purchases had NULL Shipping Type.
134 --Expected Columns: Item Purchased, NULL Shipping Type Count
135 select item_purchased,
136        count_if(shipping_type is null) as NULL_Shipping_Type_Count
137 from shopping_tread
138 group by item_purchased
139 having count_if(shipping_type is null) > 3;
140 -----
```

Below the query editor, there is a 'Results' tab selected. It shows a table with two columns: 'ITEM_PURCHASED' and 'NULL_SHIPPING_TYPE_COUNT'. The table has three rows of data:

	ITEM_PURCHASED	NULL_SHIPPING_TYPE_COUNT
1	null	4
2	Shirt	5
3	Shoes	4

Q14

The screenshot shows the same SQL IDE interface as above, but with a different query for Q14. The query is as follows:

```
141 -----
142 --14. Show a count of how many customers per Payment Method have NULL Review Rating.
143 --Expected Columns: Payment Method, Missing Review Rating Count
144 select payment_method,
145        count_if(review_rating is null) as Missing_Review_rating_count
146 from shopping_tread
147 group by payment_method;
148 -----
149 -----
```

Below the query editor, there is a 'Results' tab selected. It shows a table with two columns: 'PAYMENT_METHOD' and 'MISSING_REVIEW_RATING_COUNT'. The table has five rows of data:

	PAYMENT_METHOD	MISSING_REVIEW_RATING_COUNT
1	Credit Card	8
2	PayPal	3
3	Debit Card	7
4	null	2
5	Cash	4

Q15

Practical 1 Practical 2 Practical 3 Bright_Coffee_Case_study 1 BrightTV_Case_study 2 +

ACCOUNTADMIN COMPUTE_WH (X-Small) Share

PRACTICALS.PRACTICAL_3 Settings

Open in Workspaces Code Versions

```
149 -----
150 --15. Group by Category and return the average Review Rating, replacing NULLs with 0, and filter only where average is greater than 3.5.
151 --Expected Columns: Category, Average Review Rating
152 select category,
153        avg(coalesce(review_rating, 0)) as Average_Review_Rating
154 from shopping_tread
155 group by category
156 having avg(coalesce(review_rating, 0)) > 3.5;
```

Results Chart

CATEGORY AVERAGE_REVIEW_RATING

Query produced no results

Q16

Practical 1 Practical 2 Practical 3 Bright_Coffee_Case_study 1 BrightTV_Case_study 2 +

ACCOUNTADMIN COMPUTE_WH (X-Small) Share

PRACTICALS.PRACTICAL_3 Settings

Open in Workspaces Code Versions

```
154 -----
155 from shopping_tread
156 group by category
157 having avg(coalesce(review_rating, 0)) > 3.5;
158 --16. List all Colors that are missing (NULL) in at least 2 rows and the average Age of customers for those rows.
159 --Expected Columns: Color, Average Age
160 select color,
161        avg(age) as Average_Age
162 from shopping_tread
163 group by color
164 having count_if(color is null) >= 2;
```

Results Chart

	COLOR	AVERAGE_AGE
1	null	47.8461538

Q17

The screenshot shows a SQL IDE interface with a dark theme. At the top, there are tabs for 'Practical 1', 'Practical 2', 'Practical 3', 'Bright_Coffee_Case_study 1', and 'BrightTV_Case_study 2'. The 'Practical 3' tab is active. Below the tabs, there is a toolbar with 'ACCOUNTADMIN', 'COMPUTE_WH (X-Small)', 'Share', and a play button. The main editor area shows a SQL query for Q17. The query is as follows:

```
164 having count_if(color is null) >= 2;
165 -----
166 --17. Use CASE to create a column Delivery Speed: 'Fast' if Shipping Type is 'Express' or
167 --'Next Day Air', 'Slow' if 'Standard', 'Other' for all else including NULL. Then count how many customers fall into each category.
168 --Expected Columns: Delivery Speed, Customer Count
169 select
170     case
171         when shipping_type in ('Express','Next Day Air') then 'Fast'
172         when shipping_type = 'Standard' then 'Slow'
173         else 'Other'
174     end as Delivery_Speed,
175     count(*) as customer_count
176 from shopping_tread
177 group by Delivery_Speed;
```

Below the query editor, there is a 'Results' tab. The results are displayed in a table with two columns: 'DELIVERY_SPEED' and '# CUSTOMER_COUNT'.

	DELIVERY_SPEED	# CUSTOMER_COUNT
1	Other	166
2	Fast	89
3	Slow	45

Q18

The screenshot shows a SQL IDE interface with a dark theme. At the top, there are tabs for 'Practical 1', 'Practical 2', 'Practical 3', 'Bright_Coffee_Case_study 1', and 'BrightTV_Case_study 2'. The 'Practical 3' tab is active. Below the tabs, there is a toolbar with 'ACCOUNTADMIN', 'COMPUTE_WH (X-Small)', 'Share', and a play button. The main editor area shows a SQL query for Q18. The query is as follows:

```
177
178 from shopping_tread
179 group by Delivery_Speed;
180 -----
181
182 --18. Find customers whose purchase_amount is NULL and whose Promo Code Used is 'Yes'.
183 --Expected Columns: Customer ID, purchase_amount, Promo Code Used
184 select customer_id,
185     purchase_amount,
186     promo_code_used
187 from shopping_tread
188 where purchase_amount is null
189 and promo_code_used = 'Yes';
190 -----
```

Below the query editor, there is a 'Results' tab. The results are displayed in a table with four columns: '# CUSTOMER_ID', '# PURCHASE_AMOUNT', 'O|I PROMOTE_CODE_USED', and 'PROMO_CODE_USED'.

	# CUSTOMER_ID	# PURCHASE_AMOUNT	O I PROMOTE_CODE_USED	PROMO_CODE_USED
1	13	null	TRUE	TRUE
2	30	null	TRUE	TRUE
3	78	null	TRUE	TRUE
4	95	null	TRUE	TRUE
5	124	null	TRUE	TRUE
6	129	null	TRUE	TRUE

Q19

The screenshot shows a SQL IDE interface with a query editor and a results table. The query is for Q19, which involves grouping by location and calculating the maximum previous purchases and average review rating. The results table shows 7 rows of data.

```
186         promo_code_used
187     from shopping_tread
188     where purchase_amount is null
189         and promo_code_used = 'Yes';
190
191     -----
192     --19. Group by Location and show the maximum Previous Purchases, replacing NULLs with 0, only where the average rating is above 4.0.
193     --Expected Columns: Location, Max Previous Purchases, Average Review Rating
194     select location,
195            max(coalesce(previous_purchases, 0)) as Max_Previous_Purchases,
196            avg(review_rating) as Average_Review_Rating
197     from shopping_tread
198     group by location;
```

	LOCATION	MAX_PREVIOUS_PURCHASES	AVERAGE_REVIEW_RATING
1	Rhode Island	50.0	3.7428571
2	Kentucky	46.0	3.7107143
3	Texas	47.0	3.5523810
4	Massachusetts	47.0	3.6580645
5	null	50.0	3.5956522
6	Oregon	50.0	3.6133333
7	New York	49.0	3.9280000

Q20

The screenshot shows a SQL IDE interface with a query editor and a results table. The query is for Q20, which involves selecting customers with a NULL shipping type and a purchase amount between 30 and 70 USD. The results table shows 5 rows of data.

```
195         avg(review_rating) as Average_Review_Rating
196     from shopping_tread
197     group by location;
198
199     -----
200     --20. Show customers who have a NULL Shipping Type but made a purchase in the range of 30 to 70 USD.
201     --Expected Columns: Customer ID, Shipping Type, purchase_amount, Item Purchased
202     select customer_id,
203            shipping_type,
204            purchase_amount,
205            item_purchased
206     from shopping_tread
207     where shipping_type is null
208         and purchase_amount between 30 and 70;
```

	CUSTOMER_ID	SHIPPING_TYPE	PURCHASE_AMOUNT	ITEM_PURCHASED
1	15	null	54.0	Jeans
2	105	null	43.0	Shirt
3	141	null	37.0	Shorts
4	196	null	66.0	Coat
5	213	null	36.0	Shirt