

NULL FUNCTIONS Exercise – Lecia Mochueneng

1.

```
8      -----
9      --1. Find all records where Size is missing and the purchase_amount is greater than 50. Expected Columns:
10     Customer_ID, Size, purchase_amount, Item Purchased
11
12     SELECT
13         CUSTOMER_ID,
14         SIZE,
15         PURCHASE_AMOUNT,
16         ITEM_PURCHASED
17     FROM SALESDB.SALES.SHOPPINGTRENDS
18     WHERE SIZE IS NULL
19     AND PURCHASE_AMOUNT > 50;
20
```

Results (just now)

#	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
6	73	null	65.0	Sandals
7	91	null	54.0	Shoes
8	97	null	56.0	Shoes

2.

```
17
18      -----
19      -- 2. List the total number of purchases grouped by Season, treating NULL values as 'Unknown Season'. Expected
20      Columns: Season, Total Purchases
21
22
23     SELECT
24         IFNULL(Season, 'Unknown Season') AS Season,
25         COUNT(*) AS Total_Purchases
26     FROM SALESDB.SALES.SHOPPINGTRENDS
27     GROUP BY IFNULL(Season, 'Unknown Season');
28
29      -----
30
31     Count how many customers used each Payment Method. treating NULLS as 'Not Provided'. Expected Columns: Payment
```

Results (just now)

#	SEASON	TOTAL_PURCHASES
1	Summer	65
2	Unknown Season	27
3	Winter	80
4	Spring	73
5	Fall	55

3.

```

29      -----
30      --3. Count how many customers used each Payment Method, treating NULLS as 'Not Provided'. Expected Columns:
31      | Payment Method, Customer Count
32      | Ctrl+I to generate
33      SELECT
34          IFNULL(Payment_Method, 'Not Provided') AS Payment_Method,
35          COUNT(*) AS Customer_Count
36      FROM SALESDB.SALES.SHOPPINGTRENDS
37      GROUP BY IFNULL(Payment_Method, 'Not Provided');

```

results (just now)

Table Chart

	A PAYMENT_METHOD	# CUSTOMER_COUNT
1	PayPal	51
2	Not Provided	30
3	Credit Card	44
4	Venmo	53
5	Debit Card	42
6	Bank Transfer	38
7	Cash	42

4.

```

37      -----
38      -- 4. Show customers where Promo Code Used is NULL and Review Rating is below 3.0. Expected Columns: Customer
39      ID, Promo Code Used, Review Rating, Item Purchased
40      SELECT
41          CUSTOMER_ID,
42          PROMO_CODE_USED,
43          REVIEW_RATING,
44          ITEM_PURCHASED
45      FROM SALESDB.SALES.SHOPPINGTRENDS
46      WHERE PROMO_CODE_USED IS NULL
47          AND REVIEW_RATING < 3.0;

```

results (2 minutes ago)

Table Chart

	# CUSTOMER_ID	0 1 PROMO_CODE_USED	# REVIEW_RATING	A ITEM_PURCHASED
1	21	null	2.5	Jeans
2	38	null	2.6	Jeans
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

Feedback

5.

```

48
49
50      -- 5. Group customers by Shipping Type, and return the average purchase_amount, treating missing values as 0.
      Expected Columns: Shipping Type, Average purchase_amount
51
52      | SELECT SHIPPING_TYPE, IFNULL(AVG(PURCHASE_AMOUNT),0)Average_Purchased_Amount
53      | FROM SALESDB.SALES.SHOPPINGTRENDS
54      | group by (SHIPPING_TYPE);
55
56

```

Results (just now)

Table Chart

	▲ SHIPPING_TYPE	# AVERAGE_PURCHASED_AMOUNT
1	Free Shipping	60.2571429
2	Store Pickup	60.7317073
3	null	61.8695652
4	Express	60.3076923
5	Standard	55.0000000
6	Next Day Air	60.2195122
7	2-Day Shipping	60.9318182

6.

```

--6. Display the number of purchases per Location only for those with more than 5 purchases and no NULL
Payment Method. Expected Columns: Location, Total Purchases

SELECT
    Location,
    COUNT(*) AS Total_Purchases
FROM SALESDB.SALES.SHOPPINGTRENDS
WHERE Payment_Method IS NOT NULL
GROUP BY Location
HAVING COUNT(*) > 5;

```

7.

(just now)

Chart

	▲ LOCATION	# TOTAL_PURCHASES
	null	24
	Maine	41
	Oregon	30
	Kentucky	30
	Florida	32
	Massachusetts	31
	Texas	22
	Rhode Island	29

7.

```

66      --7. Create a column Spender Category that classifies customers using CASE: 'High' if amount > 80, 'Medium' if
67      BETWEEN 50 AND 80, 'Low' otherwise. Replace NULLs in purchase_amount with 0. Expected Columns: Customer ID,
68      purchase_amount, Spender Category
69
70      SELECT
71          CUSTOMER_ID,
72          IFNULL(purchase_amount, 0) AS purchase_amount,
73          CASE
74              WHEN IFNULL(purchase_amount, 0) > 80 THEN 'High'
75              WHEN IFNULL(purchase_amount, 0) BETWEEN 50 AND 80 THEN 'Medium'
76              ELSE 'Low'
77          END AS Spender_Category
78      FROM SALESDB.SALES.SHOPPINGTRENDS;

```

results (just now)

Table Chart

300 rows 55ms

#	CUSTOMER_ID	PURCHASE_AMOUNT	SPENDER_CATEGORY
1	1	20.0	Low
2	2	21.0	Low
3	3	27.0	Low
4	4	45.0	Low
5	5	80.0	Medium
6	6	82.0	High
7	7	50.0	Medium
8	8	29.0	Low

8.

```

// -----
78      --8. Find customers who have no Previous Purchases value but whose Color is not NULL. Expected Columns:
79      Customer ID, Color, Previous Purchases
80
81      SELECT
82          CUSTOMER_ID,
83          COLOR,
84          PREVIOUS_PURCHASES
85      FROM SALESDB.SALES.SHOPPINGTRENDS
86      WHERE PREVIOUS_PURCHASES IS NULL
87          AND Color IS NOT NULL;

```

results (just now)

Table Chart

36 rows 59ms

#	CUSTOMER_ID	COLOR	PREVIOUS_PURCHASES
1	8	Green	null
2	21	Yellow	null
3	25	White	null
4	37	Maroon	null
5	40	Gray	null
6	43	Black	null
7	44	Green	null
8	70	White	null

9.

```

87      -----
88      -- 9. Group records by Frequency of Purchases and show the total amount spent per group, treating NULL
89      frequencies as 'Unknown'. Expected Columns: Frequency of Purchases, Total purchase_amount
90
91      SELECT
92          IFNULL(Frequency_of_Purchases, 'Unknown') AS Frequency_of_Purchases,
93          SUM(IFNULL(purchase_amount, 0)) AS Total_purchase_amount
94      FROM SALESDB.SALES.SHOPPINGTRENDS
95      GROUP BY IFNULL(Frequency_of_Purchases, 'Unknown');
96
97      10.

```

Results (just now)

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

10.

```

95      -----
96      -- 10. Display a list of all Category values with the number of times each was purchased, excluding rows where
97      Category is NULL. Expected Columns: Category, Total Purchases
98
99      SELECT
100         Category,
101         COUNT(*) AS Total_Purchases
102     FROM SALESDB.SALES.SHOPPINGTRENDS
103     WHERE Category IS NOT NULL
104     GROUP BY Category;
105
106      11.

```

Results (just now)

	CATEGORY	TOTAL PURCHASES
1	Footwear	70
2	Outerwear	60
3	Clothing	59
4	Accessories	78

11.

```

104      -----
105      -- 11.Return the top 5 Locations with the highest total purchase_amount, replacing NULLs in amount with 0.
106      Expected Columns: Location, Total_purchase_amount
107
108      | SELECT TOP 5
109      |     Location,
110      |     SUM(IFNULL(purchase_amount, 0)) AS Total_purchase_amount
111      | FROM SALESDB.SALES.SHOPPINGTRENDS
112      | GROUP BY Location
113      | ORDER BY SUM(IFNULL(purchase_amount, 0)) DESC;
114      -----

```

Results (just now)

	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

12.

```

113      -----
114      -- 12. Group customers by Gender and Size, and count how many entries have a NULL Color. Expected Columns:
115      |     Gender, Size, COUNT(*) AS Null_Color_Count
116      |     SELECT
117      |         Gender,
118      |         Size,
119      |         COUNT(*) AS Null_Color_Count
120      |     FROM SALESDB.SALES.SHOPPINGTRENDS
121      |     WHERE Color IS NULL
122      |     GROUP BY Gender, Size;
123      -----

```

Results (just now)

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

13.

```

123      --13. Identify all Item Purchased where more than 3 purchases had NULL Shipping Type. Expected Columns: Item
124          Purchased, NULL Shipping Type Count
125
126      SELECT
127          Item_Purchased,
128          COUNT(*) AS Null_Shipping_Type_Count
129      FROM SALESDB.SALES.SHOPPINGTRENDS
130      WHERE Shipping_Type IS NULL
131      GROUP BY Item_Purchased
132      HAVING COUNT(*) > 3;

```

Results (just now)

	ITEM_PURCHASED	# NULL_SHIPPING_TYPE_COUNT
1	Shirt	5
2	null	4
3	Shoes	4

14.

```

135      --
134          -- 14. Show a count of how many customers per Payment Method have NULL Review Rating. Expected Columns:
135              Payment Method, Missing Review Rating Count
136
137      SELECT
138          IFNULL(Payment_Method, 'Not Provided') AS Payment_Method,
139          COUNT(*) AS Missing_Review_Rating_Count
140      FROM SALESDB.SALES.SHOPPINGTRENDS
141      WHERE Review_Rating IS NULL
142      GROUP BY IFNULL(Payment_Method, 'Not Provided');

```

Results (just now)

	PAYMENT_METHOD	# MISSING REVIEW_RATING_COUNT
1	PayPal	3
2	Not Provided	2
3	Credit Card	8
4	Venmo	9
5	Cash	4
6	Bank Transfer	4
7	Debit Card	7

15.

```

143      --
144          -- 15. Group by Category and return the average Review Rating, replacing NULLs with 0, and filter only where
145          -- average is greater than 3.5. Expected Columns: Category, Average Review Rating
146
147      SELECT
148          Category,
149          AVG(IFNULL(Review_Rating, 0)) AS Average_Review_Rating
150      FROM SALESDB.SALES.SHOPPINGTRENDS
151      GROUP BY Category
152      HAVING AVG(IFNULL(Review_Rating, 0)) > 3.5;

```

Results (just now)

	CATEGORY	AVERAGE REVIEW RATING
Query produced no results		

16.

```
152      -----
153      -- 16. List all Colors that are missing (NULL) in at least 2 rows and the average Age of customers for those
154      rows. Expected Columns: Color, Average_Age
155
156      | SELECT
157      |   Color,
158      |   AVG(Age) AS Average_Age
159      | FROM SALESDB.SALES.SHOPPINGTRENDS
160      | WHERE Color IS NULL
161      | GROUP BY Color
162      | HAVING COUNT(*) >= 2;
163      -----
```

Results (just now)								
	Table	Chart						
		Q ⏷ 1 row ⓘ 84ms 🔍 ↴						
	<table><thead><tr><th></th><th>COLOR</th><th>AVERAGE_AGE</th></tr></thead><tbody><tr><td>1</td><td>null</td><td>47.8461538</td></tr></tbody></table>		COLOR	AVERAGE_AGE	1	null	47.8461538	
	COLOR	AVERAGE_AGE						
1	null	47.8461538						

17.

```
162      -----
163      -- 17. Use CASE to create a column Delivery Speed: 'Fast' if Shipping Type is 'Express' or 'Next Day Air',
164      -- 'Slow' if 'Standard', 'Other' for all else including NULL. Then count how many customers fall into each
165      -- category. Expected Columns: Delivery Speed, Customer Count
166
167      | SELECT
168      |   CASE
169      |     WHEN Shipping_Type IN ('Express', 'Next Day Air') THEN 'Fast'
170      |     WHEN Shipping_Type = 'Standard' THEN 'Slow'
171      |     ELSE 'Other'
172      |   END AS Delivery_Speed,
173      |   COUNT(*) AS Customer_Count
174      | FROM SALESDB.SALES.SHOPPINGTRENDS
175      | GROUP BY CASE
176      |     WHEN Shipping_Type IN ('Express', 'Next Day Air') THEN 'Fast'
177      |     WHEN Shipping_Type = 'Standard' THEN 'Slow'
178      |     ELSE 'Other'
179      |   END;
```

Results (just now)														
	Table	Chart												
		Q ⏷ 3 rows ⓘ 85ms 🔍 ↴												
	<table><thead><tr><th></th><th>DELIVERY_SPEED</th><th>CUSTOMER_COUNT</th></tr></thead><tbody><tr><td>1</td><td>Slow</td><td>45</td></tr><tr><td>2</td><td>Fast</td><td>89</td></tr><tr><td>3</td><td>Other</td><td>166</td></tr></tbody></table>		DELIVERY_SPEED	CUSTOMER_COUNT	1	Slow	45	2	Fast	89	3	Other	166	
	DELIVERY_SPEED	CUSTOMER_COUNT												
1	Slow	45												
2	Fast	89												
3	Other	166												

18.

```

178  -----
179  -- 18. Find customers whose purchase_amount is NULL and whose Promo Code Used is 'Yes'. Expected Columns:
180  Customer ID, purchase_amount, Promo Code Used
181
182  SELECT
183      CUSTOMER_ID,
184      purchase_amount,
185      Promo_Code_Used
186  FROM SALESDB.SALES.SHOPPINGTRENDS
187  WHERE purchase_amount IS NULL
188      AND Promo_Code_Used = 'Yes';
189  -----

```

Results (just now)

Table Chart

#	CUSTOMER_ID	PURCHASE_AMOUNT	PROMO_CODE_USED
1		13	null
2		30	null
3		78	null
4		95	null

19.

```

188  -----
189  -- 19. Group by Location and show the maximum Previous Purchases, replacing NULLs with 0, only where the
190  average rating is above 4.0. Expected Columns: Location, Max Previous Purchases, Average Review Rating
191
192  SELECT
193      Location,
194      MAX(IFNULL(Previous_Purchases, 0)) AS Max_Previous_Purchases,
195      AVG(IFNULL(Review_Rating, 0)) AS Average_Review_Rating
196  FROM SALESDB.SALES.SHOPPINGTRENDS
197  GROUP BY Location
198  HAVING AVG(IFNULL(Review_Rating, 0)) > 4.0;
199  -----

```

Results (just now)

Table Chart

LOCATION	MAX_PREVIOUS_PURCHASES	AVERAGE REVIEW RATING
Query produced no results		

20.

```
199      -----
200      -- 20. Show customers who have a NULL Shipping Type but made a purchase in the range of 30 to 70 USD. Expected
201      Columns: Customer ID, Shipping Type, purchase_amount, Item Purchased
202
203      SELECT
204          CUSTOMER_ID,
205          Shipping_Type,
206          purchase_amount,
207          Item_Purchased
208      FROM SALESDB.SALES.SHOPPINGTRENDS
209      WHERE Shipping_Type IS NULL
210      AND purchase_amount BETWEEN 30 AND 70;
-----
```

Results (just now)

Table

Chart

Q ⏷ 7 rows ⓘ 74ms 🔍 ↴

#	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
6	235	null	38.0	Sandals
7	293	null	35.0	null