BRIGHTLEARN

PRACTICAL 3: NULL FUNCTIONS

LUBABALO SESONASIPHO SITAMA

QUESTION 1:

```
-- Find all records where Size is missing and the purchase_amount is greater than
-- 50.
-- Expected Columns: Customer ID, Size, purchase_amount, Item
-- Purchased
SELECT CUSTOMER_ID,
       SIZE,
       PURCHASE_AMOUNT,
       ITEM_PURCHASED
FROM PRACTICAL3.NULLFUNCTIONS.SHOPINGTRENDS
WHERE SIZE IS NULL AND PURCHASE_AMOUNT > 50;
```

Results | Chart

	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

QUESTION 2:

PRACTICAL3.NULLFUNCTIONS | Settings | Open in Workspaces

```
FROM PRACTICAL3.NULLFUNCTIONS.SHOPINGTRENDS
WHERE SIZE IS NULL AND PURCHASE_AMOUNT > 50;

-- List the total number of purchases grouped by Season, treating NULL values a
-- s 'Unknown Season'.
-- Expected Columns: Season, Total Purchases
SELECT IFNULL(SEASON,'UNKNOWN SEASON') AS SEASON,
       COUNT(*) AS TOTAL_PURCHASES
FROM PRACTICAL3.NULLFUNCTIONS.SHOPINGTRENDS
GROUP BY SEASON;
```

Results | Chart

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

QUESTION 3:

PRACTICAL3.NULLFUNCTIONS Settings Open in Workspaces

```
17  FROM PRACTICAL3.NULLFUNCTIONS.SHOPINGTRENDS
18  GROUP BY SEASON;
19
20  -- Count how many customers used each Payment Method, treating NULLs as
21  -- 'Not Provided'.
22  -- Expected Columns: Payment Method, Customer Count
23  SELECT IFNULL (PAYMENT_METHOD,'NOT PROVIDED') AS PAYMENT_METHOD,
24         COUNT(CUSTOMER_ID) AS CUSTOMER_COUNT
25  FROM PRACTICAL3.NULLFUNCTIONS.SHOPINGTRENDS
26  GROUP BY PAYMENT_METHOD;
27
```

Results Chart

	PAYMENT_METHOD	CUSTOMER_COUNT
1	Credit Card	44
2	PayPal	51
3	Debit Card	42
4	NOT PROVIDED	30
5	Cash	42
6	Bank Transfer	38

QUESTION 4:

PRACTICAL3.NULLFUNCTIONS Settings Open in Workspaces

```
29  -- 3.0.
30  -- Expected Columns: Customer ID, Promo Code Used, Review Rating, Item Purchased
31  SELECT CUSTOMER_ID,
32         PROMO_CODE_USED,
33         REVIEW_RATING,
34         ITEM_PURCHASED
35  FROM PRACTICAL3.NULLFUNCTIONS.SHOPINGTRENDS
36  WHERE PROMO_CODE_USED IS NULL AND REVIEW_RATING < 3.0;
37
38
39
```

Results Chart

	CUSTOMER_ID	PROMO_CODE_USED	REVIEW_RATING	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

QUESTION 5:

```
37
38  -- Group customers by Shipping
39  -- Type, and return the average purchase_amount, treating missing values as 0.
40  -- Expected Columns: Shipping Type, Average purchase_amount
41  SELECT SHIPPING_TYPE,
42         IFNULL(AVG(PURCHASE_AMOUNT),0) AS AVERAGE_PURCHASE_AMOUNT
43  FROM PRACTICAL3.NULLFUNCTIONS.SHOPINGTRENDS
44  GROUP BY SHIPPING_TYPE;
45
46
47
```

Results Chart

	SHIPPING_TYPE	AVERAGE_PURCHASE_AMOUNT
1	Standard	55.0000000
2	Express	60.3076923
3	Store Pickup	60.7317073
4	null	61.8695652
5	Free Shipping	60.2571429
6	Next Day Air	60.2195122

QUESTION 6:

PRACTICAL3.NULLFUNCTIONS Settings

```
46  -- Display the number of purchases per Location only for those with more than
47  -- 5 purchases and no NULL Payment Method.
48  -- Expected Columns: Location, Total Purchases
49  SELECT LOCATION,
50         PAYMENT_METHOD,
51         COUNT(*) AS TOTAL_PURCHASES,
52  FROM PRACTICAL3.NULLFUNCTIONS.SHOPINGTRENDS
53  GROUP BY ALL
54  HAVING TOTAL_PURCHASES > 5 AND PAYMENT_METHOD IS NOT NULL;
55
56
```

Results Chart

	LOCATION	PAYMENT_METHOD	TOTAL_PURCHASES
16	Kentucky	Credit Card	7
17	null	Credit Card	7
18	New York	Cash	7
19	Rhode Island	Venmo	7
20	Maine	Cash	8
21	null	PayPal	6

QUESTION 7:

PRACTICAL3.NULLFUNCTIONS Settings

```
56  -- Create a column Spender Category that classifies customers using CASE:
57  -- 'High' if amount > 80, 'Medium' if BETWEEN 50 AND 80,
58  -- 'Low' otherwise. Replace NULLs in purchase_amount with 0.
59  -- Expected Columns: Customer ID, purchase_amount, Spender Category
60  SELECT CUSTOMER_ID,
61         IFNULL(PURCHASE_AMOUNT,0) AS PURCHASE_AMOUNT,
62  CASE WHEN PURCHASE_AMOUNT > 80 THEN 'HIGH'
63       WHEN PURCHASE_AMOUNT BETWEEN 50 AND 80 THEN 'MEDIUM'
64       ELSE 'LOW'
65       END AS SPENDER_CATEGORY
66  FROM PRACTICAL3.NULLFUNCTIONS.SHOPINGTRENDS;
```

Results Chart

	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

QUESTION 8:

PRACTICAL3.NULLFUNCTIONS Settings

```
65      END AS SPENDER_CATEGORY
66  FROM PRACTICAL3.NULLFUNCTIONS.SHOPINGTRENDS;
67
68  -- Find customers who have no Previous
69  -- Purchases value but whose Color is not NULL.
70  -- Expected Columns: Customer ID, Color, Previous Purchases
71  SELECT CUSTOMER_ID,
72         COLOR,
73         PREVIOUS_PURCHASES
74  FROM PRACTICAL3.NULLFUNCTIONS.SHOPINGTRENDS
75  WHERE COLOR IS NOT NULL AND PREVIOUS_PURCHASES IS NULL;
```

Results Chart

	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

QUESTION 9:

PRACTICAL3.NULLFUNCTIONS Settings

```
76
77  -- Group records by Frequency of
78  -- Purchases and show the total amount spent per group, treating NULL frequenc
79  -- ies as 'Unknown'.
80  -- Expected Columns: Frequency of Purchases, Total purchase_amount
81  SELECT IFNULL(FREQUENCY_OF_PURCHASES, 'UNKNOWN') AS FREQUENCY_OF_PURCHASES,
82         SUM(PURCHASE_AMOUNT) AS TOTAL_PURCHASE_AMOUNT
83  FROM PRACTICAL3.NULLFUNCTIONS.SHOPINGTRENDS
84  GROUP BY FREQUENCY_OF_PURCHASES;
85
86
```

Results Chart

	FREQUENCY_OF_PURCHASES	TOTAL_PURCHASE_AMOUNT
1	Every 3 Months	1749
2	Weekly	2184
3	Bi-Weekly	2099
4	Monthly	1780
5	UNKNOWN	1518
6	Fortnightly	2033

QUESTION 10:

PRACTICAL3.NULLFUNCTIONS Settings

```
FROM PRACTICAL3.NULLFUNCTIONS.SHOPINGTRENDS
GROUP BY FREQUENCY_OF_PURCHASES;

-- Display a list of all Category values with the number of times each was purcha
-- sed, excluding rows where Categoryis NULL.
-- Expected Columns: Category, Total Purchases
SELECT CATEGORY,
       COUNT(*) AS TOTAL_PURCHASES
FROM PRACTICAL3.NULLFUNCTIONS.SHOPINGTRENDS
GROUP BY CATEGORY
HAVING CATEGORY IS NULL;
```

Results Chart

	CATEGORY	TOTAL_PURCHASES
1	null	33

QUESTION 11:

PRACTICAL3.NULLFUNCTIONS Settings

```

-- Return the top
-- 5 Locations with the highest total purchase_amount, replacing NULLs in amount
-- with 0.
-- Expected Columns: Location, Total purchase_amount
SELECT LOCATION,
       IFNULL(SUM(PURCHASE_AMOUNT),0) AS TOTAL_PURCHASE_AMOUNT
FROM PRACTICAL3.NULLFUNCTIONS.SHOPINGTRENDS
GROUP BY LOCATION
LIMIT 5;

```

Results Chart

	LOCATION	TOTAL_PURCHASE_AMOUNT
1	Rhode Island	1876.0
2	Kentucky	1798.0
3	Texas	1252.0
4	Massachusetts	1899.0
5	null	1470.0

QUESTION 12:

PRACTICAL3.NULLFUNCTIONS Settings

```
-- Expected Columns: Gender, Size, Null Color Count
SELECT GENDER,
       SIZE,
       SUM(CASE WHEN COLOR IS NULL THEN 1
       ELSE 0
       END) AS NULL_COLOR_COUNT
FROM PRACTICAL3.NULLFUNCTIONS.SHOPINGTRENDS
GROUP BY GENDER,SIZE
HAVING NULL_COLOR_COUNT > 0;
```

Results Chart

	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

QUESTION 13:

PRACTICAL3.NULLFUNCTIONS Settings

```
-- Type.
-- Expected Columns: Item Purchased, NULL Shipping Type Count
SELECT ITEM_PURCHASED,
       SUM(CASE WHEN SHIPPING_TYPE IS NULL THEN 1
       ELSE 1
       END) AS NULL_SHIPPING_TYPE
FROM PRACTICAL3.NULLFUNCTIONS.SHOPINGTRENDS
GROUP BY ITEM_PURCHASED
HAVING NULL_SHIPPING_TYPE > 3;
```

Results Chart

	ITEM_PURCHASED	NULL_SHIPPING_TYPE
1	Jeans	25
2	Sandals	32
3	Sweater	27
4	Shorts	30
5	null	29
6	Sneakers	39

QUESTION 14:

PRACTICAL3.NULLFUNCTIONS Settings

```
130  -- Expected Columns: Payment Method, Missing Review Rating Count
131  SELECT PAYMENT_METHOD,
132         SUM(CASE WHEN REVIEW_RATING IS NULL THEN 1
133         ELSE 0
134         END) AS MISSING_REVIEW_RATING
135  FROM PRACTICAL3.NULLFUNCTIONS.SHOPINGTRENDS
136  GROUP BY PAYMENT_METHOD
137  HAVING MISSING_REVIEW_RATING > 0;
138
139
140
```

Results Chart

	PAYMENT_METHOD	MISSING_REVIEW_RATING
1	Credit Card	8
2	PayPal	3
3	Debit Card	7
4	null	2
5	Cash	4
6	Bank Transfer	4

QUESTION 15:

PRACTICAL3.NULLFUNCTIONS Settings

```
139  -- Group by Category and return the average Review Rating, replacing NULLs with
140  -- 0, and filter only where average is greater than 3.5.
141  -- Expected Columns: Category, Average Review Rating
142  SELECT CATEGORY,
143         COALESCE(AVG(REVIEW_RATING),0) AS AVERAGE_REVIEW_RATING
144  FROM PRACTICAL3.NULLFUNCTIONS.SHOPINGTRENDS
145  GROUP BY CATEGORY
146  HAVING AVERAGE_REVIEW_RATING > 3.5;
147
148
149
```

Results Chart

	CATEGORY	AVERAGE_REVIEW_RATING
1	Outerwear	3.8173077
2	Footwear	3.6573770
3	null	3.7258065
4	Accessories	3.7338235

QUESTION 16:

PRACTICAL3.NULLFUNCTIONS Settings

```
145  GROUP BY CATEGORY
146  HAVING AVERAGE_REVIEW_RATING > 3.5;
147
148  -- List all Colors that are missing (NULL) in at least
149  -- 2 rows and the average Age of customers for those rows.
150  -- Expected Columns: Color, Average Age
151  SELECT COLOR,
152         AVG(AGE) AS AVERAGE_AGE
153  FROM PRACTICAL3.NULLFUNCTIONS.SHOPINGTRENDS
154  GROUP BY COLOR
155  HAVING COLOR IS NULL AND COUNT (*) >= 2;
```

Results Chart

	COLOR	AVERAGE_AGE
1	null	47.8461538

QUESTION 17:

PRACTICAL3.NULLFUNCTIONS Settings

```
156
157  -- Use CASE to create a column Delivery Speed: 'Fast' if Shipping Type is 'Express' or
158  -- 'Next Day Air', 'Slow' if 'Standard',
159  -- 'Other' for all else including NULL. Then count how many customers fall into
160  -- each category.
161  -- Expected Columns: Delivery Speed, Customer Count
162  SELECT
163  CASE WHEN SHIPPING_TYPE = 'EXPRESS' OR SHIPPING_TYPE = 'NEXT DAY AIR' THEN 'FAST'
164       WHEN SHIPPING_TYPE = 'STANDARD' THEN 'SLOW'
165       ELSE 'OTHER'
166       END AS DELIVERY_SPEED,
167       COUNT(*) AS CUSTOMER_COUNT
168  FROM PRACTICAL3.NULLFUNCTIONS.SHOPINGTRENDS
169  GROUP BY DELIVERY_SPEED;
170
```

Results Chart

	DELIVERY_SPEED	CUSTOMER_COUNT
1	OTHER	300

QUESTION 18:

PRACTICAL3.NULLFUNCTIONS Settings

```
169  GROUP BY DELIVERY_SPEED;
170
171  -- Find customers whose purchase_amount is NULL and whose Promo Code Used is
172  -- 'Yes'.
173  -- Expected Columns: Customer ID, purchase_amount, Promo Code Used
174  SELECT CUSTOMER_ID,
175         PURCHASE_AMOUNT,
176         PROMO_CODE_USED
177  FROM PRACTICAL3.NULLFUNCTIONS.SHOPINGTRENDS
178  WHERE PURCHASE_AMOUNT IS NULL AND PROMO_CODE_USED = 'YES';
179
180
181
182
183
```

Results Chart

	CUSTOMER_ID	PURCHASE_AMOUNT	0|1 PROMO_CODE_USED
1	13	null	TRUE
2	30	null	TRUE
3	78	null	TRUE
4	95	null	TRUE

QUESTION 19:

PRACTICAL3.NULLFUNCTIONS Settings

```
178  WHERE PURCHASE_AMOUNT IS NULL AND PROMO_CODE_USED = 'YES';
179
180  -- Group by Location and show the maximum Previous
181  -- Purchases, replacing NULLs with 0, only where the average rating is above 4.0.
182  -- Expected Columns: Location, Max Previous Purchases, Average
183  -- Review Rating
184  SELECT LOCATION,
185         COALESCE(MAX(PREVIOUS_PURCHASES), 0) AS MAX_PREVIOUS_PURCHASES,
186         AVG(REVIEW_RATING) AS AVERAGE_RATING
187  FROM PRACTICAL3.NULLFUNCTIONS.SHOPINGTRENDS
188  GROUP BY LOCATION
189  HAVING AVERAGE_RATING > 4.0;
190
191
192
```

Results Chart

LOCATION	MAX_PREVIOUS_PURCHASES	AVERAGE_RATING
Query produced no results		

QUESTION 20:

PRACTICAL3.NULLFUNCTIONS Settings

```
189  HAVING AVERAGE_RATING > 4.0;
190
191  -- Show customers who have a NULL Shipping
192  -- Type but made a purchase in the range of 30 to 70 USD.
193  -- Expected Columns: Customer ID, Shipping
194  -- Type, purchase_amount, Item Purchased
195  SELECT CUSTOMER_ID,
196         SHIPPING_TYPE,
197         PURCHASE_AMOUNT,
198         ITEM_PURCHASED
199  FROM PRACTICAL3.NULLFUNCTIONS.SHOPINGTRENDS
200  WHERE SHIPPING_TYPE IS NULL AND PURCHASE_AMOUNT BETWEEN 30 AND 70;
201
202
203
```

Results Chart

	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

PRACTICAL 3 SQL CODE:

-- Find all records where Size is missing and the purchase_amount is greater than

-- 50.

-- Expected Columns: Customer ID, Size, purchase_amount, Item

-- Purchased

```
SELECT CUSTOMER_ID,
    SIZE,
    PURCHASE_AMOUNT,
    ITEM_PURCHASED
FROM PRACTICAL3.NULLFUNCTIONS.SHOPINGTRENDS
WHERE SIZE IS NULL AND PURCHASE_AMOUNT > 50;
```

-- List the total number of purchases grouped by Season, treating NULL values a

-- s 'Unknown Season'.

-- Expected Columns: Season, Total Purchases

```
SELECT IFNULL(SEASON,'UNKNOWN SEASON') AS SEASON,
    COUNT(*) AS TOTAL_PURCHASES
FROM PRACTICAL3.NULLFUNCTIONS.SHOPINGTRENDS
GROUP BY SEASON;
-----------------------------------------------------------------------------

-- Count how many customers used each Payment Method, treating NULLs as
-- 'Not Provided'.
-- Expected Columns: Payment Method, Customer Count
SELECT IFNULL (PAYMENT_METHOD,'NOT PROVIDED') AS PAYMENT_METHOD,
    COUNT(CUSTOMER_ID) AS CUSTOMER_COUNT
FROM PRACTICAL3.NULLFUNCTIONS.SHOPINGTRENDS
GROUP BY PAYMENT_METHOD;
-----------------------------------------------------------------------------

-- 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 PRACTICAL3.NULLFUNCTIONS.SHOPINGTRENDS
WHERE PROMO_CODE_USED IS NULL AND REVIEW_RATING < 3.0;
-----------------------------------------------------------------------------

-- Group customers by Shipping
-- Type, and return the average purchase_amount, treating missing values as 0.
-- Expected Columns: Shipping Type, Average purchase_amount
SELECT SHIPPING_TYPE,
    IFNULL(AVG(PURCHASE_AMOUNT),0) AS AVERAGE_PURCHASE_AMOUNT
```

```
FROM PRACTICAL3.NULLFUNCTIONS.SHOPINGTRENDS
GROUP BY SHIPPING_TYPE;
-------------------------------------------------------------------------------

-- 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,
    PAYMENT_METHOD,
    COUNT(*) AS TOTAL_PURCHASES,
FROM PRACTICAL3.NULLFUNCTIONS.SHOPINGTRENDS
GROUP BY ALL
HAVING TOTAL_PURCHASES > 5 AND PAYMENT_METHOD IS NOT NULL;
-------------------------------------------------------------------------------

-- Create a column Spender Category that classifies customers using CASE:
-- 'High' if amount > 80, 'Medium' if BETWEEN 50 AND 80,
-- 'Low' otherwise. Replace NULLs in purchase_amount with 0.
-- Expected Columns: Customer ID, purchase_amount, Spender Category
SELECT CUSTOMER_ID,
    IFNULL(PURCHASE_AMOUNT,0) AS PURCHASE_AMOUNT,
CASE WHEN PURCHASE_AMOUNT > 80 THEN 'HIGH'
   WHEN PURCHASE_AMOUNT BETWEEN 50 AND 80 THEN 'MEDIUM'
   ELSE 'LOW'
   END AS SPENDER_CATEGORY
FROM PRACTICAL3.NULLFUNCTIONS.SHOPINGTRENDS;
-------------------------------------------------------------------------------

-- Find customers who have no Previous
-- Purchases value but whose Color is not NULL.
-- Expected Columns: Customer ID, Color, Previous Purchases
```

```
SELECT CUSTOMER_ID,
    COLOR,
    PREVIOUS_PURCHASES
FROM PRACTICAL3.NULLFUNCTIONS.SHOPINGTRENDS
WHERE COLOR IS NOT NULL AND PREVIOUS_PURCHASES IS NULL;
--------------------------------------------------------------------------------

-- Group records by Frequency of
-- Purchases and show the total amount spent per group, treating NULL frequenc
-- ies as 'Unknown'.
-- Expected Columns: Frequency of Purchases, Total purchase_amount
SELECT IFNULL(FREQUENCY_OF_PURCHASES, 'UNKNOWN') AS
FREQUENCY_OF_PURCHASES,
    SUM(PURCHASE_AMOUNT) AS TOTAL_PURCHASE_AMOUNT
FROM PRACTICAL3.NULLFUNCTIONS.SHOPINGTRENDS
GROUP BY FREQUENCY_OF_PURCHASES;
--------------------------------------------------------------------------------

-- Display a list of all Category values with the number of times each was purcha
-- sed, excluding rows where Categoryis NULL.
-- Expected Columns: Category, Total Purchases
SELECT CATEGORY,
    COUNT(*) AS TOTAL_PURCHASES
FROM PRACTICAL3.NULLFUNCTIONS.SHOPINGTRENDS
GROUP BY CATEGORY
HAVING CATEGORY IS NULL;
--------------------------------------------------------------------------------

-- Return the top
-- 5 Locations with the highest total purchase_amount, replacing NULLs in amount
-- with 0.
```

```
-- Expected Columns: Location, Total purchase_amount
SELECT LOCATION,
    IFNULL(SUM(PURCHASE_AMOUNT),0) AS TOTAL_PURCHASE_AMOUNT
FROM PRACTICAL3.NULLFUNCTIONS.SHOPINGTRENDS
GROUP BY LOCATION
LIMIT 5;
--------------------------------------------------------------------------------

-- Group customers by Gender and Size, and count how many entries have a NUL
-- L Color.
-- Expected Columns: Gender, Size, Null Color Count
SELECT GENDER,
    SIZE,
    SUM(CASE WHEN COLOR IS NULL THEN 1
    ELSE 0
    END) AS NULL_COLOR_COUNT
FROM PRACTICAL3.NULLFUNCTIONS.SHOPINGTRENDS
GROUP BY GENDER,SIZE
HAVING NULL_COLOR_COUNT > 0;
--------------------------------------------------------------------------------

-- Identify all Item Purchased where more than 3 purchases had NULL Shipping
-- Type.
-- Expected Columns: Item Purchased, NULL Shipping Type Count
SELECT ITEM_PURCHASED,
    SUM(CASE WHEN SHIPPING_TYPE IS NULL THEN 1
    ELSE 1
    END) AS NULL_SHIPPING_TYPE
FROM PRACTICAL3.NULLFUNCTIONS.SHOPINGTRENDS
GROUP BY ITEM_PURCHASED
HAVING NULL_SHIPPING_TYPE > 3;
```

```
--------------------------------------------------------------------------------

-- Show a count of how many customers per Payment Method have NULL Review
-- Rating.
-- Expected Columns: Payment Method, Missing Review Rating Count
SELECT PAYMENT_METHOD,
    SUM(CASE WHEN REVIEW_RATING IS NULL THEN 1
    ELSE 0
    END) AS MISSING_REVIEW_RATING
FROM PRACTICAL3.NULLFUNCTIONS.SHOPINGTRENDS
GROUP BY PAYMENT_METHOD
HAVING MISSING_REVIEW_RATING > 0;
--------------------------------------------------------------------------------

-- Group by Category and return the average Review Rating, replacing NULLs with
-- 0, and filter only where average is greater than 3.5.
-- Expected Columns: Category, Average Review Rating
SELECT CATEGORY,
    COALESCE(AVG(REVIEW_RATING),0) AS AVERAGE_REVIEW_RATING
FROM PRACTICAL3.NULLFUNCTIONS.SHOPINGTRENDS
GROUP BY CATEGORY
HAVING AVERAGE_REVIEW_RATING > 3.5;
--------------------------------------------------------------------------------

-- List all Colors that are missing (NULL) in at least
-- 2 rows and the average Age of customers for those rows.
-- Expected Columns: Color, Average Age
SELECT COLOR,
    AVG(AGE) AS AVERAGE_AGE
FROM PRACTICAL3.NULLFUNCTIONS.SHOPINGTRENDS
GROUP BY COLOR
```

```
HAVING COLOR IS NULL AND COUNT (*) >= 2;
-------------------------------------------------------------------------------


-- Use CASE to create a column Delivery Speed: 'Fast' if Shipping Type is 'Express' or
-- 'Next Day Air', 'Slow' if 'Standard',
-- 'Other' for all else including NULL. Then count how many customers fall into
-- each category.
-- Expected Columns: Delivery Speed, Customer Count
SELECT
CASE WHEN SHIPPING_TYPE = 'EXPRESS' OR SHIPPING_TYPE = 'NEXT DAY AIR' THEN 'FAST'
    WHEN SHIPPING_TYPE = 'STANDARD' THEN 'SLOW'
    ELSE 'OTHER'
    END AS DELIVERY_SPEED,
    COUNT(*) AS CUSTOMER_COUNT
FROM PRACTICAL3.NULLFUNCTIONS.SHOPINGTRENDS
GROUP BY DELIVERY_SPEED;
-------------------------------------------------------------------------------


-- Find customers whose purchase_amount is NULL and whose Promo Code Used is
-- 'Yes'.
-- Expected Columns: Customer ID, purchase_amount, Promo Code Used
SELECT CUSTOMER_ID,
     PURCHASE_AMOUNT,
     PROMO_CODE_USED
FROM PRACTICAL3.NULLFUNCTIONS.SHOPINGTRENDS
WHERE PURCHASE_AMOUNT IS NULL AND PROMO_CODE_USED = 'YES';
-------------------------------------------------------------------------------


-- Group by Location and show the maximum Previous
-- Purchases, replacing NULLs with 0, only where the average rating is above 4.0.
-- Expected Columns: Location, Max Previous Purchases, Average
```

```
-- Review Rating
SELECT LOCATION,
    COALESCE(MAX(PREVIOUS_PURCHASES), 0) AS MAX_PREVIOUS_PURCHASES,
    AVG(REVIEW_RATING) AS AVERAGE_RATING
FROM PRACTICAL3.NULLFUNCTIONS.SHOPINGTRENDS
GROUP BY LOCATION
HAVING AVERAGE_RATING > 4.0;
--------------------------------------------------------------------------------


-- Show customers who have a NULL Shipping
-- Type but made a purchase in the range of 30 to 70 USD.
-- Expected Columns: Customer ID, Shipping
-- Type, purchase_amount, Item Purchased
SELECT CUSTOMER_ID,
    SHIPPING_TYPE,
    PURCHASE_AMOUNT,
    ITEM_PURCHASED
FROM PRACTICAL3.NULLFUNCTIONS.SHOPINGTRENDS
WHERE SHIPPING_TYPE IS NULL AND PURCHASE_AMOUNT BETWEEN 30 AND 70;
```