

# BrightLight Data Analytics

## SQL JOIN Practice

```
1  --1. Find all records where Size is missing and the purchase_amount is greater than 50.
2  SELECT Customer_ID,
3         Size,
4         purchase_amount,
5         Item_Purchased
6     FROM SHOPPINGSALES.SHOPPINGTRENDS.TRENDS
7    WHERE Size IS NULL AND purchase_amount > 50;
8
```

↳ 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
7	91	null	54.0	Shoes
8	97	null	56.0	Shoes
9	100	null	55.0	Sneakers
10	160	null	84.0	Coat
11	173	null	96.0	Sandals
12	219	null	78.0	Shoes
13	223	null	76.0	Handbag
14	224	null	77.0	Sneakers

```
8
9  --2. List the total number of purchases grouped by Season, treating NULL values as 'Unknown Season'.
10 SELECT
11     COALESCE(Season, 'Unknown Season') AS Season,
12     COUNT(*) AS Total_Purchases
13   FROM SHOPPINGSALES.SHOPPINGTRENDS.TRENDS
14  GROUP BY COALESCE(Season, 'Unknown Season');
15
16  --3.Count how many customers used each Payment Method, treating NULLs as 'Not Provided'.
```

↳ Results ↵ Chart

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

```

16  ---3.Count how many customers used each Payment Method, treating NULLs as 'Not Provided'.
17  SELECT
18      COALESCE(Payment_Method, 'Not Provided') AS Payment_Method,
19      COUNT(DISTINCT Customer_ID) AS Customer_Count
20  FROM SHOPPINGSALES.SHOPPINGTRENDS.TRENDS
21  GROUP BY COALESCE(Payment_Method, 'Not Provided');
22

```

↳ Results ↵ Chart

A PAYMENT_METHOD	# CUSTOMER_COUNT
PayPal	51
Bank Transfer	38
Debit Card	42
Venmo	53
Not Provided	30
Cash	42
Credit Card	44

```

23  ---4.Show customers where Promo Code Used is NULL and Review Rating is below 3.0.
24  SELECT Customer_ID,
25      Promo_Code_Used,
26      Review_Rating,
27      Item_Purchased
28  FROM SHOPPINGSALES.SHOPPINGTRENDS.TRENDS
29  WHERE Promo_Code_Used IS NULL AND Review_Rating < 3.0;
30

```

↳ Results ↵ 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

```

31  ---5.Group customers by Shipping Type, and return the average purchase_amount, treating missing values as 0.
32  SELECT Shipping_Type,
33      AVG(COALESCE(purchase_amount, 0)) AS Average_purchase_amount
34  FROM SHOPPINGSALES.SHOPPINGTRENDS.TRENDS
35  GROUP BY Shipping_Type;

```

↳ Results ↵ Chart

A SHIPPING_TYPE	# AVERAGE_PURCHASE_AMOUNT
Standard	47.6666667
Express	53.4545455
Store Pickup	55.3333333
null	52.7037037
Free Shipping	50.2142857
Next Day Air	54.8666667
2-Day Shipping	51.5576923

```

37 ---6.Display the number of purchases per Location only for those with more than 5 purchases and no NULL Payment Method.
38     SELECT LOCATION,
39             COUNT(*) AS Total_Purchases
40     FROM SHOPPINGSALES.SHOPPINGTRENDS.TRENDS
41     WHERE Payment_Method IS NOT NULL
42     GROUP BY Location
43     HAVING COUNT(*) > 5;
44

```

▶ Results ⚡ Chart

A SHIPPING_TYPE	# AVERAGE_PURCHASE_AMOUNT	Query Details
Standard	47.6666667	Query duration
Express	53.4545455	Rows
Store Pickup	55.3333333	Query ID 01
null	52.7037037	Show more ▾
Free Shipping	50.2142857	SHIPPING_TYPE
Next Day Air	54.8666667	86% filled
2-Day Shipping	51.5576923	

```

45 ---7.Create a column Spender Category that classifies customers using CASE: 'High' if amount > 80, 'Medium' if BETWEEN 50 AND 80, 'Low'
46 otherwise. Replace NULLs in purchase_amount with 0.
47     SELECT Customer_ID,
48             COALESCE(purchase_amount, 0) AS purchase_amount,
49             CASE
50                 WHEN COALESCE(purchase_amount, 0) > 80 THEN 'High'
51                 WHEN COALESCE(purchase_amount, 0) BETWEEN 50 AND 80 THEN 'Medium'
52                 ELSE 'Low'
53             END AS Spender_Category
54     FROM SHOPPINGSALES.SHOPPINGTRENDS.TRENDS;

```

▶ Results ⚡ Chart

# CUSTOMER_ID	# PURCHASE_AMOUNT	▲ SPENDER_CATEGORY	Query Details
1	20.0	Low	Query duration 61ms
2	21.0	Low	Rows 300
3	27.0	Low	Query ID 01bfece0-000c-b23e-0...
4	45.0	Low	Show more ▾
5	80.0	Medium	CUSTOMER_ID
6	82.0	High	1 300
7	50.0	Medium	
8	29.0	Low	
9	100.0	High	
10	97.0	High	

```

55 ---8.Find customers who have no Previous Purchases value but whose Color is not NULL.
56     SELECT Customer_ID,
57             Color,
58             Previous_Purchases
59     FROM SHOPPINGSALES.SHOPPINGTRENDS.TRENDS
60     WHERE Previous_Purchases IS NULL AND Color IS NOT NULL;
61

```

▶ Results ⚡ Chart

# CUSTOMER_ID	▲ COLOR	# PREVIOUS_PURCHASES
8	Green	null
21	Yellow	null
25	White	null
37	Maroon	null
40	Gray	null
43	Black	null
44	Green	null
70	White	null
73	Maroon	null
75	Pink	null
83	Black	null
85	Yellow	null

```

62 | ---9.Group records by Frequency of Purchases and show the total amount spent per group, treating NULL frequencies as 'Unknown'.
63 | SELECT
64 |     COALESCE(Frequency_of_Purchases, 'Unknown') AS Frequency_of_Purchases,
65 |     SUM(purchase_amount) AS Total_purchase_amount
66 | FROM SHOPPINGSALES.SHOPPINGTRENDS.TRENDS
67 | GROUP BY COALESCE(Frequency_of_Purchases, 'Unknown');
68 |

```

→ Results ↵ Chart

FREQUENCY_OF_PURCHASES	TOTAL_PURCHASE_AMOUNT
Every 3 Months	1749.0
Weekly	2184.0
Bi-Weekly	2099.0
Monthly	1780.0
Fortnightly	2033.0
Annually	1765.0
Unknown	1518.0
Quarterly	2541.0

```

69 | ---10.Display a list of all Category values with the number of times each was purchased, excluding rows where Category is NULL.
70 | SELECT Category,
71 |         COUNT(*) AS Total_Purchases
72 |     FROM SHOPPINGSALES.SHOPPINGTRENDS.TRENDS
73 |     WHERE Category IS NOT NULL
74 |     GROUP BY Category;
75 |

```

→ Results ↵ Chart

CATEGORY	TOTAL_PURCHASES
Outerwear	60
Footwear	70
Clothing	59
Accessories	78

```

76 | ---11.Return the top 5 Locations with the highest total purchase_amount, replacing NULLs in amount with 0.
77 | SELECT Location,
78 |         SUM(COALESCE(purchase_amount, 0)) AS Total_purchase_amount
79 |     FROM SHOPPINGSALES.SHOPPINGTRENDS.TRENDS
80 |     GROUP BY Location
81 |     ORDER BY Total_purchase_amount DESC
82 |     LIMIT 5;
83 |

```

→ Results ↵ Chart

LOCATION	TOTAL_PURCHASE_AMOUNT
Maine	2294.0
Florida	1980.0
Massachusetts	1899.0
Rhode Island	1876.0
Kentucky	1798.0

```

84 | ---12.Group customers by Gender and Size, and count how many entries have a NULL Color.
85 | SELECT Gender,
86 |         Size,
87 |         COUNT(*) AS Null_Color_Count
88 |     FROM SHOPPINGSALES.SHOPPINGTRENDS.TRENDS
89 |     WHERE Color IS NULL
90 |     GROUP BY Gender, Size;
91 |

```

→ Results ↵ Chart

GENDER	SIZE	NULL_COLOR_COUNT
Male	null	6
Male	M	7
Male	L	6
Male	S	5
Male	XL	5

```

92  ---13.Identify all Item Purchased where more than 3 purchases had NULL Shipping Type.
93  | SELECT Item_Purchased,
94  |       COUNT(*) AS NULL_Shipping_Type_Count
95  | FROM SHOPPINGSALES.SHOPPINGTRENDS.TRENDS
96  | WHERE Shipping_Type IS NULL
97  | GROUP BY Item_Purchased
98  | HAVING COUNT(*) > 3;
99

```

Results Chart

ITEM_PURCHASED	# NULL_SHIPPING_TYPE_COUNT
null	4
Shirt	5
Shoes	4

```

100 ---14.Show a count of how many customers per Payment Method have NULL Review Rating.
101 | SELECT Payment_Method,
102 |       COUNT(*) AS Missing_Review_Rating_Count
103 | FROM SHOPPINGSALES.SHOPPINGTRENDS.TRENDS
104 | WHERE Review_Rating IS NULL
105 | GROUP BY Payment_Method;
106

```

Results Chart

PAYMENT_METHOD	# MISSING REVIEW_RATING COUNT
Credit Card	8
Cash	4
null	2
Debit Card	7
Venmo	9
PayPal	3
Bank Transfer	4

```

107 ---15.Group by Category and return the average Review Rating, replacing NULLs with 0, and filter only where average is greater than 3.5.
108 | SELECT Category,
109 |       AVG(Review_Rating) AS Average_Review_Rating
110 | FROM SHOPPINGSALES.SHOPPINGTRENDS.TRENDS
111 | GROUP BY Category
112 | HAVING AVG(Review_Rating) > 3.5;
113

```

Results Chart

CATEGORY	# AVERAGE REVIEW RATING
null	3.7258065
Outerwear	3.8173077
Footwear	3.6573770
Accessories	3.7338235

```

114 ---16.List all Colors that are missing (NULL) in at least 2 rows and the average Age of customers for those rows.
115 | SELECT Color,
116 |       AVG(Age) AS Average_Age
117 | FROM SHOPPINGSALES.SHOPPINGTRENDS.TRENDS
118 | WHERE Color IS NULL
119 | GROUP BY Color
120 | HAVING COUNT(*) >= 2;
121

```

Results Chart

COLOR	# AVERAGE AGE
null	47.8461538

```

22 ---17.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.
23
24     CASE
25         WHEN Shipping_Type IN ('Express', 'Next Day Air') THEN 'Fast'
26         WHEN Shipping_Type = 'Standard' THEN 'Slow'
27         ELSE 'Other'
28     END AS Delivery_Speed,
29     COUNT(DISTINCT Customer_ID) AS Customer_Count
30 FROM SHOPPINGSALES.SHOPPINGTRENDS.TRENDS
31 GROUP BY
32     CASE
33         WHEN Shipping_Type IN ('Express', 'Next Day Air') THEN 'Fast'
34         WHEN Shipping_Type = 'Standard' THEN 'Slow'
35         ELSE 'Other'
36     END;
37

```

Results

A DELIVERY_SPEED	# CUSTOMER_COUNT
Fast	89
Slow	45
Other	166

```

138 ---18.Find customers whose purchase_amount is NULL and whose Promo Code Used is 'Yes'.
139     SELECT Customer_ID, purchase_amount, Promo_Code_Used
140     FROM SHOPPINGSALES.SHOPPINGTRENDS.TRENDS
141     WHERE purchase_amount IS NULL AND Promo_Code_Used = 'Yes';
142

```

Results

# CUSTOMER_ID	# PURCHASE_AMOUNT	01 PROMO_CODE_USED
	13	null TRUE
	30	null TRUE
	78	null TRUE
	95	null TRUE
	124	null TRUE
	129	null TRUE
	130	null TRUE
	138	null TRUE
	153	null TRUE
	168	null TRUE
	177	null TRUE

```

143 ---19.Group by Location and show the maximum Previous Purchases, replacing NULLs with 0, only where the average rating is above 4.0.
144     SELECT LOCATION,
145         MAX(PREVIOUS_PURCHASES) AS Max_Previous_Purchases,
146         AVG(REVIEW_RATING) AS Average_Review_Rating
147     FROM SHOPPINGSALES.SHOPPINGTRENDS.TRENDS
148     GROUP BY LOCATION
149     HAVING AVG(REVIEW_RATING) > 4.0;
150

```

Results

LOCATION	MAX_PREVIOUS_PURCHASES	AVERAGE REVIEW RATING
		Query produced no results

```

151 ---20.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
152     SELECT Customer_ID, Shipping_Type, purchase_amount, Item_Purchased
153     FROM SHOPPINGSALES.SHOPPINGTRENDS.TRENDS
154     WHERE Shipping_Type IS NULL AND purchase_amount BETWEEN 30 AND 70;
155

```

Results

# CUSTOMER_ID	A SHIPPING_TYPE	# PURCHASE_AMOUNT	A ITEM_PURCHASED
1	15	54.0	Jeans
2	105	43.0	Shirt
3	141	37.0	Shorts
4	196	66.0	Coat
5	213	36.0	Shirt
6	235	38.0	Sandals
7	293	35.0	null