1.

```sql
--- Find all records where size is missing and purchase_amount is >50
SELECT customer_id,
size,
purchase_amount,
item_purchased,
FROM shopping_trends
WHERE size IS NULL
AND purchase_amount > 50;
```

Results | Chart

	CUSTOMER_ID	SIZE	PURCHASE_AMOUNT	ITEM_PURCHASED
1	11	null	74	Handbag
2	15	null	54	Jeans
3	22	null	88	Shirt
4	32	null	54	Blouse
5	62	null	57	Blouse
6	73	null	65	Sandals
7	91	null	54	Shoes
8	97	null	56	Shoes
9	100	null	55	Sneakers
10	160	null	84	Coat
11	173	null	96	Sandals
12	219	null	78	Shoes

2.

```sql
--- List the total number of purchase grouped by Season, treating null values as unknown season
SELECT
COALESCE(season, 'Unknown_Season') AS season,
COUNT(item_purchased) AS total_purchase
FROM shopping_trends
GROUP BY (season, 'Unknown_Season');
```

Results | Chart

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

3.

```sql
---- Count how many customers used each PaymentMethod, treating NULLs as 'Not Provided'
SELECT
COALESCE (PAYMENT_METHOD, 'Not Provided') AS payment_method,
COUNT (customer_id) as customer_count
FROM SHOPPING_TRENDS
GROUP BY (PAYMENT_METHOD, 'Not Provided')
```

Results | Chart

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

4.

```
--- Show customers where promo code is NULL and review rating is below 3.0
SELECT customer_id,
promo_code_used,
review_rating,
item_purchased,
FROM shopping_trends
WHERE promo_code_used IS NULL
AND review_rating < 3.0;
```

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
7	180	null	2.5	Shorts
8	285	null	2.9	Blouse

5.

```
---- Group customers by shipping, type and return the average purchase_amount, treating missing valuse as 0
SELECT
shipping_type,
AVG(COALESCE(purchase_amount, 0)) AS avg_purchase_amount
FROM shopping_trends
GROUP BY shipping_type;
```

Results | Chart

	SHIPPING_TYPE	AVG_PURCHASE_AMOUNT
1	Standard	47.666667
2	Store Pickup	55.333333
3	null	52.703704
4	2-Day Shipping	51.557692
5	Free Shipping	50.214286
6	Next Day Air	54.866667
7	Express	53.454545

6.

```sql
---- Display the number of purchases per location only for those with more than 5 purchases and no NULL payment method
SELECT
location,
COUNT(*) AS total_purchases
FROM shopping_trends
WHERE payment_method IS NOT NULL
GROUP BY location
HAVING COUNT(*) > 5;
```

Results | Chart

	LOCATION	TOTAL_PURCHASES
1	Maine	41
2	Rhode Island	29
3	null	24
4	Oregon	30
5	Florida	32
6	Kentucky	30
7	New York	31
8	Massachusetts	31
9	Texas	22

7.

```sql
---- Create a column 'spender category' that classifies customers using CASE: 'High' if amount
SELECT
customer_id,
COALESCE(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 shopping_trends;
```

Results | Chart

	CUSTOMER_ID	PURCHASE_AMOUNT	SPENDER_CATEGORY
10	10	97	High
11	11	74	Medium
12	12	59	Medium
13	13	0	Low
14	14	91	High
15	15	54	Medium
16	16	52	Medium
17	17	39	Low
18	18	94	High
19	19	23	Low
20	20	79	Medium

8.

```sql
---- Find customers who have no previous purchases value but whose color is not null
SELECT
customer_id,
color,
previous_purchases
FROM shopping_trends
WHERE previous_purchases IS NULL
AND color IS NOT 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
7	44	Green	null
8	70	White	null
9	73	Maroon	null
10	75	Pink	null
11	83	Black	null
12	85	Yellow	null

9.

```
---- Group records by frequency purchases and show the total amount spent per group, treating null frequencies as 'unknown'
SELECT
COALESCE(frequency_of_purchases, 'unknown') AS frequency_of_purchases,
SUM (purchase_amount) AS total_amount_spent
FROM shopping_trends
GROUP BY frequency_of_purchases;
```

Results | Chart

	FREQUENCY_OF_PURCHASES	TOTAL_AMOUNT_SPENT
1	Every 3 Months	1749
2	Weekly	2184
3	Bi-Weekly	2099
4	Monthly	1780
5	unknown	1518
6	Fortnightly	2033
7	Quarterly	2541
8	Annually	1765

10.

```
---- Display a list of all category values with the number of times each was purchased, excluding rows where category is null
SELECT
category,
COUNT(*) AS total_purchase
FROM shopping_trends
WHERE category IS NOT NULL
GROUP BY category
ORDER BY total_purchase DESC;
```

Results | Chart

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

11.

```sql
---- Return the top 5 locations with the highest total purchase_amount, replacing NULL in amount with 0
SELECT
location,
SUM(COALESCE(purchase_amount, 0)) AS total_purchase_amount
FROM shopping_trends
GROUP BY location
ORDER BY total_purchase_amount DESC
LIMIT 5;
```

Results | Chart

	LOCATION	TOTAL_PURCHASE_AMOUNT
1	Maine	2294
2	Florida	1980
3	Massachusetts	1899
4	Rhode Island	1876
5	Kentucky	1798

12.

```sql
--- Group customers by gender and size and count how many entries has a null color
SELECT
GENDER,
SIZE,
COUNT (*) AS null_color_count
FROM shopping_trends
WHERE color IS NULL
GROUP BY gender,size;
```

Results | Chart

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

13.

```sql
--- Identify all items purchased where more than 3 purchases had null shipping type
SELECT
ITEM_PURCHASED,
COUNT(*) AS null_shipping_count
FROM shopping_trends
WHERE shipping_type IS NULL
GROUP BY item_purchased
HAVING COUNT(*) > 3;
```

Results | Chart

	ITEM_PURCHASED	NULL_SHIPPING_COUNT
1	null	4
2	Shoes	4
3	Shirt	5

14.

```sql
---- Show a count of how many customers per payment Method have null review rating
SELECT
payment_method,
COUNT(*) AS null_review_count
FROM shopping_trends
WHERE review_rating IS NULL
GROUP BY payment_method;
```

Results | Chart

	PAYMENT_METHOD	NULL_REVIEW_COUNT
1	Credit Card	8
2	Cash	4
3	Bank Transfer	4
4	Debit Card	7
5	Venmo	9
6	PayPal	3
7	null	2

15.

```sql
---Group by category and return review_rating, replacing NULL with 0 and filter any where average is greater than 3.5
SELECT
category,
AVG(COALESCE(review_rating, 0)) AS avg_review_rating
FROM shopping_trends
GROUP BY category
HAVING avg_review_rating > 3.5;
```

Results | Chart

CATEGORY	AVG_REVIEW_RATING

Query produced no results

16.

```sql
--- List all colors that are missing (NULL) in at least 2 rows and the average Age of customers for those rows
SELECT
color,
avg (age) AS Average_age
FROM shopping_trends
WHERE color IS NULL
GROUP BY color
HAVING COUNT(*) >= 2;
```

Results | Chart

	COLOR	AVERAGE_AGE
1	null	47.846154

17.

```sql
--- Use CASE to create a column Delivery_Speed: 'fast' is 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
SELECT
count (customer_id) AS customer_count,
case WHEN shipping_type IN ('express', 'next day air') THEN 'fast'
WHEN shipping_type = 'standard' THEN 'slow'
ELSE 'other'
END AS Delivery_speed
FROM shopping_trends
GROUP BY Delivery_Speed;
```

Results | Chart

	CUSTOMER_COUNT	DELIVERY_SPEED
1	300	other

18.

```
130  --- Find customers whose purchase_amount is NULL and whose promo_code_used is 'yes'
131  SELECT
132  customer_id,
133  purchase_amount,
134  promo_code_used
135  FROM shopping_trends
136  WHERE purchase_amount IS NULL
137  AND promo_code_used = 'yes';
```

Results | Chart

	CUSTOMER_ID	PURCHASE_AMOUNT	PROMO_CODE_USED
1	13	null	TRUE
2	30	null	TRUE
3	78	null	TRUE
4	95	null	TRUE
5	124	null	TRUE
6	129	null	TRUE
7	130	null	TRUE
8	138	null	TRUE
9	153	null	TRUE
10	168	null	TRUE
11	177	null	TRUE
12	202	null	TRUE

19.

```
138  --- Group by location and show maximum previous purchase, replacing nulls with 0, only where the average rating is above 4.0
139  SELECT
140  LOCATION,
141  MAX (previous_purchases) AS Maximum_Previous_Purchases,
142  AVG(COALESCE(Review_rating,0)) Average_Review_Rating
143  FROM shopping_trends
144  Group by LOCATION
145  HAVING Average_Review_Rating >4.0;
```

Results | Chart

LOCATION	MAXIMUM_PREVIOUS_PURCHASES	AVERAGE_REVIEW_RATING
Query produced no results		

20.

```sql
----
SELECT
CUSTOMER_ID, SHIPPING_TYPE,
PURCHASE_AMOUNT, ITEM_PURCHASED
FROM shopping_trends
WHERE SHIPPING_TYPE is NULL
AND PURCHASE_AMOUNT BETWEEN 30 AND 70;
```

Results | Chart

	CUSTOMER_ID	SHIPPING_TYPE	PURCHASE_AMOUNT	ITEM_PURCHASED
1	15	null	54	Jeans
2	105	null	43	Shirt
3	141	null	37	Shorts
4	196	null	66	Coat
5	213	null	36	Shirt
6	235	null	38	Sandals
7	293	null	35	null