

SQL Data Analysis Report on Customer Shopping Behavior

--Q1. Write a query to display customer_id, gender, category, and purchase amount.

```
SELECT customer_id, gender, category, purchase_amount_usd  
FROM customer;
```

customer_id	gender	category	purchase_amount_usd
1	1	Clothing	53
2	2	Clothing	64
3	2	Clothing	73
4	4	Footwear	90
5	5	Clothing	49
6	6	Footwear	20
7	7	Clothing	85
8	8	Clothing	34
9	9	Outerwear	97
10	10	Accessories	31
11	11	Footwear	34
12	12	Clothing	68
13	13	Outerwear	72
14	14	Clothing	81
15	15	Outerwear	83
16	16	Clothing	81
17	17	Accessories	36
18	18	Clothing	38
19	19	Clothing	48
20	20	Clothing	90
21	21	Clothing	51

--Q2. Retrieve all customers who spent more than 50 USD.

```
SELECT *  
FROM customer  
WHERE purchase_amount_usd > 50;
```

customer_id	age	gender	item_purchased	category	purchase_amount_usd	location	size	color
1	1	55	Blouse	Clothing	53	Kentucky	L	Gray
2	2	19	Sweater	Clothing	64	Maine	L	Maroon
3	3	50	Jeans	Clothing	73	Massachusetts	S	Maroon
4	4	21	Sandals	Footwear	90	Rhode Island	M	Maroon
5	7	63	Shirt	Clothing	49	Montana	M	Gray
6	9	26	Coat	Outerwear	97	West Virginia	L	Silver
7	12	50	Shirts	Clothing	68	Hawaii	S	Olive
8	13	61	Coat	Outerwear	72	Delaware	M	Gold
9	14	65	Dress	Clothing	51	New Hampshire	M	Purple
10	15	64	Coat	Outerwear	63	New York	L	Teal
11	16	64	Skirt	Clothing	81	Rhode Island	M	Teal
12	20	66	Pants	Clothing	90	Rhode Island	M	Green
13	21	21	Pants	Clothing	51	Louisiana	M	Black
14	22	31	Pants	Clothing	62	North Carolina	M	Charcoal
15	24	31	Pants	Clothing	88	Oklahoma	XL	White
16	28	56	Shorts	Clothing	56	Kentucky	L	Cyan
17	39	54	Hairbrush	Accessories	94	North Carolina	XL	Grey

--Q3. List the top 10 highest spending customers.

```
SELECT customer_id, purchase_amount_usd FROM customer ORDER BY
purchase_amount_usd DESC LIMIT 10;
```

	customer_id bigint	purchase_amount_usd bigint
1	519	100
2	582	100
3	456	100
4	96	100
5	249	100
6	205	100
7	194	100
8	244	100
9	43	100
10	616	100

--Q4. Find total revenue generated from each product category.

```
SELECT
category,
SUM(purchase_amount_usd) AS total_revenue
FROM customer
GROUP BY category
ORDER BY total_revenue DESC;
```

	category text	total_revenue numeric
1	Clothing	104264
2	Accessories	74200
3	Footwear	36093
4	Outerwear	18524

--Q5. Perform an INNER JOIN between customer table & category_info table.

```
SELECT c.customer_id, c.item_purchased, c.category, ci.department FROM customer c  
INNER JOIN category_info ci ON c.category = ci.category;
```

	customer_id bigint	item_purchased text	category text	department character varying (50)
1		1 Blouse	Clothing	Fashion
2		2 Sweater	Clothing	Fashion
3		3 Jeans	Clothing	Fashion
4		4 Sandals	Footwear	Fashion
5		5 Blouse	Clothing	Fashion
6		6 Sneakers	Footwear	Fashion
7		7 Shirt	Clothing	Fashion
8		8 Shorts	Clothing	Fashion
9		9 Coat	Outerwear	Lifestyle
10		10 Handbag	Accessories	Fashion
11		11 Shoes	Footwear	Fashion
12		12 Shorts	Clothing	Fashion
13		13 Coat	Outerwear	Lifestyle
14		14 Dress	Clothing	Fashion

--Q6. LEFT JOIN to show all customers even if no department exists.

```
SELECT c.customer_id, c.item_purchased, c.category, ci.department  
FROM customer c  
LEFT JOIN category_info ci  
ON c.category = ci.category;
```

	customer_id bigint	item_purchased text	category text	department character varying (50)
1		1 Blouse	Clothing	Fashion
2		2 Sweater	Clothing	Fashion
3		3 Jeans	Clothing	Fashion
4		4 Sandals	Footwear	Fashion
5		5 Blouse	Clothing	Fashion
6		6 Sneakers	Footwear	Fashion
7		7 Shirt	Clothing	Fashion
8		8 Shorts	Clothing	Fashion
9		9 Coat	Outerwear	Lifestyle
10		10 Handbag	Accessories	Fashion
11		11 Shoes	Footwear	Fashion
12		12 Shorts	Clothing	Fashion
13		13 Coat	Outerwear	Lifestyle
14		14 Dress	Clothing	Fashion

Total rows: 1000 of 3900 Query complete 00:00:00.197

--Q7. RIGHT JOIN to show all departments even if no purchases happened.

SELECT

```
ci.category,  
ci.department,  
c.item_purchased  
  
FROM customer c  
RIGHT JOIN category_info ci  
ON c.category = ci.category;
```

	category character varying (50) 	department character varying (50) 	item_purchased text 
1	Clothing	Fashion	Blouse
2	Clothing	Fashion	Sweater
3	Clothing	Fashion	Jeans
4	Footwear	Fashion	Sandals
5	Clothing	Fashion	Blouse
6	Footwear	Fashion	Sneakers
7	Clothing	Fashion	Shirt
8	Clothing	Fashion	Shorts
9	Outerwear	Lifestyle	Coat
10	Accessories	Fashion	Handbag
11	Footwear	Fashion	Shoes

Total rows: 1000 of 3900 Query complete 00:00:00.126

Q8. Find customers who spent above the average purchase amount.

```
SELECT customer_id, purchase_amount_usd  
FROM customer  
WHERE purchase_amount_usd > (  
    SELECT AVG(purchase_amount_usd)  
    FROM customer
```

	customer_id bigint	purchase_amount_usd bigint
1	2	64
2	3	73
3	4	90
4	7	85
5	9	97
6	12	68
7	13	72
8	16	81
9	20	90
10	22	62
11	24	88
12	29	94
13	32	79
14	33	67
15	35	91
16	37	69
17	40	60
18	41	76
19	43	100
20	44	69
21	55	94

Total rows: 1000 of 1963 Query complete 00:00:00.187

--Q9. Find the total number of customers from each gender.

```
SELECT gender, COUNT(*) AS total_customers  
FROM customer  
GROUP BY gender;
```

	gender text	total_customers bigint
1	Female	1248
2	Male	2652

--Q10. Find the average age of customers for each product category.

```
SELECT category, AVG(age) AS avg_age  
FROM customer  
GROUP BY category  
ORDER BY avg_age;
```

	category text	avg_age numeric
1	Clothing	43.7829591249280368
2	Accessories	44.2241935483870968
3	Outerwear	44.3117283950617284
4	Footwear	44.4424040066777963

--Q11. Count how many customers purchased each item.

```
SELECT item_purchased, COUNT(*) AS total_orders  
FROM customer  
GROUP BY item_purchased  
ORDER BY total_orders DESC;
```

	item_purchased text	total_orders bigint
1	Jewelry	171
2	Pants	171
3	Blouse	171
4	Shirt	169
5	Dress	166
6	Sweater	164
7	Jacket	163
8	Sunglasses	161
9	Belt	161
10	Coat	161
11	Sandals	160

--Q12. Find the total number of online subscription customers.

```
SELECT subscription_status, COUNT(*) AS total_count  
FROM customer  
GROUP BY subscription_status;
```

	subscription_status	total_count
1	No	2847
2	Yes	1053

--Q13. Get the average review rating for each season

```
SELECT season, AVG(review_rating) AS avg_rating  
FROM customer  
GROUP BY season  
ORDER BY avg_rating DESC;
```

	season	avg_rating
1	Spring	3.7889564336372867
2	Winter	3.750571131879544
3	Fall	3.73261093911249
4	Summer	3.726800847457627

--Q14. Find the location with the highest total revenue.

```
SELECT location, SUM(purchase_amount_usd) AS total_revenue  
FROM customer  
GROUP BY location  
ORDER BY total_revenue DESC LIMIT 1;
```

	location	total_revenue
1	Montana	5784

--Q15. Find the total revenue generated during each season.

```
SELECT season, SUM(purchase_amount_usd) AS total_revenue  
FROM customer  
GROUP BY season  
ORDER BY total_revenue DESC;
```

	season text	total_revenue numeric
1	Fall	60018
2	Spring	58679
3	Winter	58607
4	Summer	55777

--Q16. List customers who used a promo code.

```
SELECT customer_id, item_purchased, promo_code_used  
FROM customer  
WHERE promo_code_used = 'Yes';
```

	customer_id bigint	item_purchased text	promo_code_used text
1	1	Blouse	Yes
2	2	Sweater	Yes
3	3	Jeans	Yes
4	4	Sandals	Yes
5	5	Blouse	Yes
6	6	Sneakers	Yes

Total rows: 1000 of 1677 Query complete 00:00:00.116

--Q17. Show the count of purchases based on payment method.

```
SELECT payment_method, COUNT(*) AS total_purchases  
FROM customer  
GROUP BY payment_method  
ORDER BY total_purchases DESC;
```

	payment_method	total_purchases
1	PayPal	677
2	Credit Card	671
3	Cash	670
4	Debit Card	636
5	Venmo	634
6	Bank Transfer	612

Total rows: 6 of 6 Query complete 00:00:00.072

--Q18. Find customers who made more than 5 previous purchases.

```
SELECT customer_id, previous_purchases  
FROM customer  
WHERE previous_purchases > 5;
```

	customer_id	previous_purchases
1	1	14
2	3	23
3	4	49
4	5	31
5	6	14
6	7	49

Total rows: 1000 of 3476 Query complete 00:00:00.128

--Q19. List items purchased by female customers only.

```
SELECT customer_id, item_purchased, category  
FROM customer  
WHERE gender = 'Female';
```

	customer_id bigint	item_purchased text	category text
1	2653	Shorts	Clothing
2	2654	Blouse	Clothing
3	2655	Coat	Outerwear
4	2656	Sunglasses	Accessories
5	2657	Shorts	Clothing
6	2658	Dress	Clothing

Total rows: 1000 of 1248 Query complete 00:00:00.264

--Q20. Fetch the top 5 highest-rated products.

```
SELECT item_purchased, review_rating  
FROM customer  
ORDER BY review_rating DESC  
LIMIT 5;
```

	item_purchased text	review_rating double precision
1	Gloves	[null]
2	Dress	[null]
3	Blouse	[null]
4	Sandals	[null]
5	Pants	[null]