

## SQL Query and Output data:

```
-- 1. Total number of records in the dataset
SELECT COUNT(*) AS total_records FROM Retail_Sales;
```

Output data:

total_records
1997

```
-- 2. Total number of unique customers
SELECT COUNT(DISTINCT customer_id) AS total_unique_customers FROM Retail_Sales;
```

Output data:

total_unique_customers
155

- **Findings:** The dataset contains **1997 total records** and **155 unique customers**.
- **Business Impact:** Understanding the volume of transactions and customer reach helps assess overall business activity.
- **Recommendation:** If the number of unique customers is low compared to total records, Zenith should focus on customer acquisition and retention strategies.

```
-- 3. Number of distinct product categories
SELECT COUNT(DISTINCT category) AS distinct_categories FROM Retail_Sales;
```

Output data:

distinct_categories
3

```
-- 4. Total sales per category (to identify the highest revenue-generating categories)
SELECT category, SUM(total_sale) AS total_sales
FROM Retail_Sales
GROUP BY category
ORDER BY total_sales DESC;
```

Output data:

category	total_sales
Electronics	313810
Clothing	311070
Beauty	286840

```
-- 5. Checking for missing values in key fields (data quality check)
SELECT
    SUM(CASE WHEN category IS NULL THEN 1 ELSE 0 END) AS missing_categories,
    SUM(CASE WHEN transactions_id IS NULL THEN 1 ELSE 0 END) AS missing_transaction_ids,
    SUM(CASE WHEN customer_id IS NULL THEN 1 ELSE 0 END) AS missing_customer_ids,
    SUM(CASE WHEN category IS NULL THEN 1 ELSE 0 END) AS missing_categories,
    SUM(CASE WHEN quantity IS NULL THEN 1 ELSE 0 END) AS missing_quantity_sold,
    SUM(CASE WHEN total_sale IS NULL THEN 1 ELSE 0 END) AS missing_total_sales,
```

```

SUM(CASE WHEN age IS NULL THEN 1 ELSE 0 END) AS missing_customer_ages,
SUM(CASE WHEN gender IS NULL THEN 1 ELSE 0 END) AS missing_customer_genders,
SUM (CASE WHEN cogs IS NULL THEN 1 ELSE 0 END) AS missing_cogs

```

FROM Retail\_Sales;

Output data:

	missing_transaction_ids	missing_customer_ids	missing_categories	missing_quantity_sold	missing_total_sales	missing_customer_ages	missing_customer_genders
1	0	0	0	3	3	10	0

-- 6. Customer demographics (age and gender) per category

```

SELECT category, gender, ROUND(AVG(CAST(age AS FLOAT)), 2) AS avg_age, COUNT(*) AS
total_customers
FROM Retail_Sales
GROUP BY category, gender
ORDER BY category;

```

Output data:

category	gender	avg_age	total_customers
Beauty	Female	40.37	330
Beauty	Male	40.47	282
Clothing	Male	40.68	354
Clothing	Female	43.2	347
Electro...	Female	40.21	340
Electro...	Male	42.95	344

-- 7. Transactions where the total sale is greater than R1,000

```

SELECT * FROM Retail_Sales WHERE total_sale > 1000;

```

Output data:

transactions_id	sale_date	sale_time	customer_id	gender	age	category	quantity	price_per_unit	cogs	total_sale
522	2022-07-09	11:00:00.0000000	52	Male	46	Beauty	3	500	145	1500
559	2022-12-12	10:48:00.0000000	5	Female	40	Clothing	4	300	84	1200
1522	2022-11-14	08:35:00.0000000	48	Male	46	Beauty	3	500	235	1500
1559	2022-08-20	07:40:00.0000000	49	Female	40	Clothing	4	300	144	1200
421	2022-04-08	08:43:00.0000000	66	Female	37	Clothing	3	500	235	1500
1421	2022-01-17	07:07:00.0000000	59	Female	37	Clothing	3	500	185	1500
484	2022-03-13	07:52:00.0000000	135	Female	19	Clothing	4	300	75	1200
1484	2022-11-23	09:29:00.0000000	22	Female	19	Clothing	4	300	147	1200

-- 8. Top revenue-generating customers

```

SELECT TOP 10 customer_id, SUM(total_sale) AS total_spent
FROM Retail_Sales
GROUP BY customer_id
ORDER BY total_spent DESC;

```

Output data:

customer_id	total_spent
3	38440
1	30750
5	30405
2	25295
4	23580
87	15855
54	13475
71	12790
55	12080
84	11730

```
-- 9. Total number of transactions made by each gender in each category
SELECT category, gender, COUNT(*) AS total_trans
FROM Retail_Sales
GROUP BY category, gender
ORDER BY category;
```

Output data:

category	gender	total_trans
Beauty	Female	330
Beauty	Male	282
Clothing	Male	354
Clothing	Female	347
Electro...	Female	340
Electro...	Male	344

```
-- 10. Total sales contribution by gender
SELECT gender, SUM(total_sale) AS total_sales
FROM Retail_Sales
GROUP BY gender;
```