RETAIL SALES SQL PROJECT

PRESENTED BY: MOHAMMAD AMIL KHAN

1. Write a SQL query to find the total sale made by gender category?:

```sql
SELECT gender, SUM(total_sale) AS total_sales_by_gender
FROM retail_sales_analysis
GROUP BY gender;
```

gender	total_sales_by_gender
Male	445120
Female	463110

Total sales by gender: Sales are fairly distributed between genders, with slight variations indicating differing purchasing behaviors.

2. Write a SQL query to find all transactions where the total_sale is greater than 700?

```sql
SELECT * FROM retail_sales_analysis
WHERE total_sale > 700;
```

transactions_id	sale_date	sale_time	customer_id	gender	age	category	quantity	price_per_unit	cogs	total_sale
2	2022-06-24	10:07:00	104	Female	26	Clothing	2	500	135	1000
13	2023-02-08	17:43:00	106	Male	22	Electronics	3	500	245	1500
15	2022-07-01	11:50:00	75	Female	42	Electronics	4	500	210	2000
16	2022-06-25	10:33:00	82	Male	19	Clothing	3	500	180	1500
20	2023-12-12	21:35:00	85	Male	22	Clothing	3	300	345	900
26	2023-10-14	17:54:00	51	Female	28	Electronics	2	500	600	1000
30	2023-10-10	20:05:00	59	Female	39	Beauty	3	300	285	900
31	2023-12-31	17:47:00	3	Male	44	Electronics	4	300	129	1200
35	2022-10-29	17:30:00	122	Female	58	Beauty	3	300	126	900

Transactions with total_sale > 700: These high-value transactions indicate premium purchases or bulk buying trends.

3. Write a SQL query to retrieve all columns for sales made on '2022-05-10'?

```sql
SELECT *
FROM retail_sales_analysis
WHERE sale_date = '2022-05-10';
```

transactions_id	sale_date	sale_time	customer_id	gender	age	category	quantity	price_per_unit	cogs	total_sale
238	2022-05-10	11:52:00	20	Female	39	Beauty	1	500	150	500
1041	2022-05-10	07:27:00	74	Male	34	Clothing	2	25	11	50
1257	2022-05-10	11:55:00	29	Male	19	Beauty	4	500	255	2000

Sales on '2022-05-10': This query retrieves all transactions from the specified date for daily trend analysis.

4. Write a SQL query to calculate the total sales (total_sale) for each category?

```sql
SELECT
    category,
    SUM(total_sale) as net_sale,
    COUNT(*) as total_orders
FROM retail_sales_analysis
GROUP BY 1;
```

category	net_sale	total_orders
Beauty	286790	611
Clothing	309995	698
Electronics	311445	678

Total sales per category: Some categories outperform others, highlighting top-selling product segments.

5. Write a SQL query to find the average age of customers who purchased items from the 'Beauty' category?

```sql
SELECT
    ROUND(AVG(age), 2) as avg_age
FROM retail_sales_analysis
WHERE category = 'Beauty';
```

avg_age
40.42

Average age of customers buying 'Beauty' products: The average age of Beauty category buyers indicates target customer demographics.

6. Write a SQL query to find the total number of transactions (transaction_id) made by each gender in each category?

```sql
SELECT
    category,
    gender,
    COUNT(*) as total_trans
FROM retail_sales_analysis
GROUP BY category,gender
ORDER BY 1;
```

category	gender	total_trans
Beauty	Female	330
Beauty	Male	281
Clothing	Female	347
Clothing	Male	351
Electronics	Female	335
Electronics	Male	343

Total transactions by gender in each category: Some categories have gender-dominant buyers, which can inform targeted marketing.

7. Write a SQL query to find the number of unique customers who purchased items from each category?

```sql
SELECT
    category,
    COUNT(DISTINCT customer_id) as cnt_unique_cs
FROM retail_sales_analysis
GROUP BY category;
```

category	cnt_unique_cs
Beauty	141
Clothing	149
Electronics	144

Unique customers per category: Some categories attract a wider customer base, indicating broad product appeal.

8. Write a SQL query to retrieve all transactions where the category is 'Beauty' and the quantity sold is more than 3 in the month of Oct-2022?

```sql
SELECT *
FROM retail_sales_analysis
WHERE category = 'Beauty'
  AND DATE_FORMAT(sale_date, '%Y-%m') = '2022-10'
  AND quantity >= 3;
```

transactions_id	sale_date	sale_time	customer_id	gender	age	category	quantity	price_per_unit	cogs	total_sale
35	2022-10-29	17:30:00	122	Female	58	Beauty	3	300	126	900
47	2022-10-22	17:22:00	96	Female	40	Beauty	3	500	600	1500
51	2022-10-12	22:09:00	78	Male	27	Beauty	3	25	24.75	75
200	2022-10-09	22:29:00	130	Male	27	Beauty	3	50	60	150
292	2022-10-10	06:33:00	111	Male	20	Beauty	4	300	141	1200
330	2022-10-11	20:18:00	116	Female	25	Beauty	4	50	55	200

Beauty category sales with quantity > 3 in Oct-2022: Identifies high-quantity Beauty product purchases in a specific month.

9. Write a SQL query to find what is the average total sale amount for transactions where the quantity sold is above 3 units?

```sql
SELECT AVG(total_sale) AS avg_sale_above_03_units
FROM retail_sales_analysis
WHERE quantity > 3;
```

avg_sale_above_03_units
710.095602294455

Average total sale amount for quantity > 3: Larger quantities generally correlate with higher total sales per transaction.

10. Write a SQL query to create each shift and number of orders (Example Morning <=12, Afternoon Between 12 & 17, Evening >17)?

```sql
WITH hourly_sale AS (
    SELECT *,
        CASE
            WHEN HOUR(sale_time) < 12 THEN 'Morning'
            WHEN HOUR(sale_time) BETWEEN 12 AND 17 THEN 'Afternoon'
            ELSE 'Evening'
        END AS shift
    FROM retail_sales_analysis
)
SELECT
    shift,
    COUNT(*) AS total_orders
FROM hourly_sale
GROUP BY shift;
```

shift	total_orders
Evening	1062
Morning	548
Afternoon	377

Sales shifts and order count: Most sales occur in the Afternoon and Evening, indicating peak shopping hours.

11. Write a SQL query to find the average price per unit for products sold in each category?

```sql
SELECT category, AVG(price_per_unit) AS avg_price_per_unit
FROM retail_sales_analysis
GROUP BY category;
```

category	avg_price_per_unit
Beauty	184.82815057283142
Clothing	173.77507163323781
Electronics	181.8362831858407

Average price per unit per category: Electronics and premium categories have higher unit prices compared to essentials.

12. Write a SQL query to calculate the average sale for each month. Find out best selling month in each year?

```sql
SELECT year, month, avg_sale
FROM
(
    SELECT
        YEAR(sale_date) as year,
        MONTH(sale_date) as month,
        AVG(total_sale) as avg_sale,
        RANK() OVER(PARTITION BY YEAR(sale_date) ORDER BY AVG(total_sale) DESC) as ranked
    FROM retail_sales_analysis
    GROUP BY YEAR(sale_date), MONTH(sale_date)
) AS t1
WHERE ranked = 1;
```

year	month	avg_sale
2022	7	541.3414634146342
2023	2	535.531914893617

Best-selling month per year: Identifies peak sales months, useful for planning promotions and inventory management.

13. Write a SQL query to find which month had the highest number of products sold in 2023?

```sql
SELECT MONTH(sale_date) AS sale_month, SUM(quantity) AS total_quantity_sold
FROM retail_sales_analysis
WHERE YEAR(sale_date) = 2023
GROUP BY sale_month
ORDER BY total_quantity_sold DESC
```

sale_month	total_quantity_sold
9	371

Month with highest quantity sold in 2023: Determines when demand peaked, useful for supply chain optimization.

14. Write a SQL query to find which month had the highest number of transactions in 2023?

```sql
SELECT MONTH(sale_date) AS sale_month, COUNT(*) AS total_transactions
FROM retail_sales_analysis
WHERE YEAR(sale_date) = 2023
GROUP BY sale_month
ORDER BY total_transactions DESC
LIMIT 1;
```

sale_month	total_transactions
9	146

Month with highest transactions in 2023: Identifies the busiest month based on transaction count, which may align with festivals or sales.

15. Write a SQL query to find what is the average total sale amount for each age group (e.g., 18-25, 26-35, etc.)?

```sql
SELECT
    CASE
        WHEN age BETWEEN 18 AND 25 THEN '18-25'
        WHEN age BETWEEN 26 AND 35 THEN '26-35'
        WHEN age BETWEEN 36 AND 45 THEN '36-45'
        WHEN age BETWEEN 46 AND 55 THEN '46-55'
        WHEN age BETWEEN 56 AND 65 THEN '56-65'
        ELSE '65+'
    END AS age_group,
    AVG(total_sale) AS average_total_sale
FROM retail_sales_analysis
GROUP BY age_group
ORDER BY age_group;
```

age_group	average_total_sale
18-25	502.85714285714283
26-35	481.3447432762836
36-45	455.74441687344915
46-55	440.2307692307692
56-65	412.578125

Average total sale per age group: Middle-aged customers (26-45) tend to have higher purchase values.

16. Write a SQL query to find which customer made the largest purchase in terms of a single transaction in 2023?

```sql
SELECT customer_id, MAX(total_sale) AS largest_purchase
FROM retail_sales_analysis
WHERE YEAR(sale_date) = 2023
GROUP BY customer_id
ORDER BY largest_purchase DESC
LIMIT 1;
```

customer_id	largest_purchase
3	2000

Customer with the largest transaction in 2023: Highlights high-value customers for loyalty programs and personalized marketing.

17. Write a SQL query to find the percentage of total sales made in the 'Electronics' category compared to the overall sales in 2023?

```sql
SELECT
    (SUM(CASE WHEN category = 'Electronics' THEN total_sale ELSE 0 END) / SUM(total_sale)) * 100 AS electronics_sales_percentage
FROM retail_sales_analysis
WHERE YEAR(sale_date) = 2023;
```

electronics_sales_percentage
35.37846348293183

Percentage of sales from 'Electronics' in 2023: Electronics contribute a significant share to total sales, indicating a strong product category.

18. Write a SQL query to find the top 3 products (categories) based on the highest total sales in the first quarter of 2023?

```sql
SELECT category, SUM(total_sale) AS total_sales
FROM retail_sales_analysis
WHERE sale_date BETWEEN '2023-01-01' AND '2023-03-31'
GROUP BY category
ORDER BY total_sales DESC
LIMIT 3;
```

category	total_sales
Clothing	29750
Beauty	21790
Electronics	17950

Top 3 categories in Q1 2023: Reveals the best-performing product categories in the first quarter.

19. Write a SQL query to find the top 10 customers based on the highest total sales?

```sql
SELECT customer_id,
       SUM(total_sale) as total_sales
FROM retail_sales_analysis
GROUP BY 1
ORDER BY 2 DESC
LIMIT 10;
```

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

Top 10 customers by total sales: Identifies the most valuable customers based on total spending.

20. Write a SQL query to find how many transactions had a total sale amount greater than 1000 and were made by customers aged over 40?

```sql
SELECT COUNT(*) AS high_value_transactions
FROM retail_sales_analysis
WHERE total_sale > 1000 AND age > 40;
```

high_value_transactions
136

Transactions > 1000 by customers aged 40+: Older customers tend to make higher-value purchases, potentially buying premium products.

CONCLUSION

This project offers a comprehensive overview of SQL for data analysts, covering key areas such as database creation, data cleaning, exploratory data analysis, and business-oriented SQL queries. The insights gained from this analysis can assist in making informed business decisions by revealing trends in sales, customer preferences, and product performance.