Retail Sales Analysis Using SQL

Demonstrating Fundamental SQL Skills

INTRODUCTION

- **Objective:** Analyze a fictional retail sales database using SQL.
- **Dataset Overview:** Transactional data: dates, times, demographics, categories, quantities, total sales.
- **Focus Areas:** Data cleaning, analysis, and deriving insights

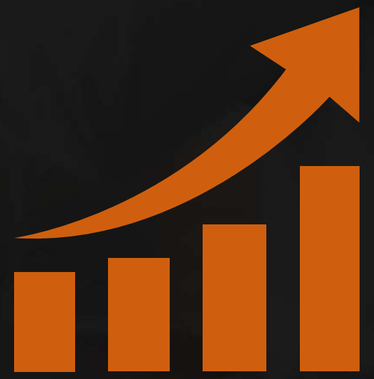

TABLE OF CONTENTS

List the main sections:

DATABASE AND TABLE CREATION

Highlight Key Points:

- Database name: SQL_Project
- Table name: Retail_Sales
- Include table structure with key columns and their data types.

SQL_PROJECT _1*

Limit to 1000 rows

```sql
-- Create Data Base
create database SQL_Project ;
USE SQL_Project;

-- create table
CREATE TABLE Retail_Sales (
    transactions_id INT PRIMARY KEY,
    sale_date DATE,
    sale_time TIME,
    customer_id INT,
    gender VARCHAR(20),
    age INT,
    category VARCHAR(25),
    quantiy FLOAT,
    price_per_unit FLOAT,
    cogs FLOAT,
    total_sale FLOAT
);
```

POLISHING THE DATA FOR ACTIONABLE INSIGHTS

Focus Areas:

1. Identifying NULL values.
2. Deleting incomplete rows.
3. Use a simple flowchart or table to explain the cleaning process.
4. Include a snippet of the SQL query used for cleaning.

```sql
SELECT *
FROM
    retail_sales
WHERE
    transactions_id IS NULL
        OR sale_date IS NULL
        OR sale_time IS NULL
        OR customer_id IS NULL
        OR gender IS NULL
        OR age IS NULL
        OR category IS NULL
        OR quantiy IS NULL
        OR price_per_unit IS NULL
        OR cogs IS NULL
        OR total_sale IS NULL;

        DELETE FROM retail_sales
WHERE
    transactions_id IS NULL
    OR sale_date IS NULL
    OR sale_time IS NULL
    OR customer_id IS NULL
    OR gender IS NULL
    OR age IS NULL
    OR category IS NULL
    OR quantiy IS NULL
    OR price_per_unit IS NULL
    OR cogs IS NULL
    OR total_sale IS NULL;
```

KEY SQL QUERIES FOR ANALYSIS

SALES SNAPSHOT: ACTIVITY ON NOVEMBER 5, 2022

```sql
-- SQL QUERIES
-- 1.Write a sql query to retrieve all columns for sales mode on '2022-11-05'

SELECT *
FROM retail_sales
WHERE sale_date = '2022-11-05';
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

transactions_id	sale_date	sale_time	customer_id	gender	age	category	quantiy	price_per_unit	cogs
180	2022-11-05	10:47:00	117	Male	41	Clothing	3	300	129
214	2022-11-05	16:31:00	53	Male	20	Beauty	2	30	8.1
240	2022-11-05	11:49:00	95	Female	23	Beauty	1	300	123
856	2022-11-05	17:43:00	102	Male	54	Electronics	4	30	9.3
943	2022-11-05	19:29:00	90	Female	57	Clothing	4	300	318
1137	2022-11-05	22:34:00	104	Male	46	Beauty	2	500	145
1256	2022-11-05	09:58:00	29	Male	23	Clothing	2	500	190
1265	2022-11-05	14:35:00	86	Male	55	Clothing	3	300	111
1587	2022-11-05	20:06:00	140	Female	40	Beauty	4	300	105
1819	2022-11-05	20:44:00	83	Female	35	Beauty	2	50	13.5
1896	2022-11-05	20:19:00	87	Female	30	Electronics	2	25	30.75
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

TRANSACTIONS FOR 'CLOTHING' CATEGORY WITH QUANTITY MORE 2 THAN IN NOVEMBER 2022

```
63      -- 2.Write a sql query to retrieve all transactions where the category is 'Clothing' and quantity sold is more than 2 in month of NOV 2022
64
65 •    SELECT *
66      FROM retail_sales
67      WHERE category = 'Clothing' AND quantiy > 2 AND DATE_FORMAT(sale_date, '%Y-%m') = '2022-11';
68
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

transactions_id	sale_date	sale_time	customer_id	gender	age	category	quantiy	price_per_unit	cogs	total_sale
64	2022-11-15	06:34:00	7	Male	49	Clothing	4	25	8.5	100
110	2022-11-18	17:28:00	149	Male	27	Clothing	3	300	99	900
126	2022-11-01	18:16:00	63	Female	28	Clothing	3	30	28.8	90
145	2022-11-06	19:21:00	64	Female	39	Clothing	3	25	27.5	75
146	2022-11-10	22:01:00	74	Male	38	Clothing	4	50	49	200
159	2022-11-10	21:30:00	42	Male	26	Clothing	4	50	23.5	200
180	2022-11-05	10:47:00	117	Male	41	Clothing	3	300	129	900
265	2022-11-24	12:12:00	80	Male	55	Clothing	3	300	132	900
284	2022-11-12	09:17:00	129	Male	43	Clothing	4	50	20.5	200
529	2022-11-29	17:43:00	46	Female	35	Clothing	3	50	20.5	150
547	2022-11-14	07:36:00	3	Male	63	Clothing	4	500	250	2000
580	2022-11-14	14:44:00	104	Female	31	Clothing	3	500	200	1500
699	2022-11-21	22:21:00	129	Female	37	Clothing	4	30	16.2	120
735	2022-11-26	21:38:00	153	Female	64	Clothing	4	500	515	2000
750	2022-11-13	07:33:00	69	Female	35	Clothing	3	25	8.75	75
943	2022-11-05	19:29:00	90	Female	57	Clothing	4	300	318	1200
965	2022-11-27	21:45:00	84	Male	22	Clothing	4	50	13	200

Category Performance: Total Sales Breakdown

```sql
-- 3.Write a SQL query to calculate the total sales (total_sale) for each category:

SELECT category, SUM(total_sale) AS TOTAL_Category_Sales
FROM Retail_Sales
GROUP BY category;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

category	TOTAL_Category_Sales
Beauty	286790
Clothing	309995
Electronics	311445

DEMOGRAPHIC INSIGHTS: BEAUTY CATEGORY CUSTOMERS

```
75      -- 4.Write a SQL query to find the average age of customers who purchased items from the 'Beauty' category
76
77  •   SELECT category, ROUND(AVG(age), 0) AS Avg_Age
78      FROM Retail_Sales
79      WHERE category = 'Beauty';
80
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	category	Avg_Age
▸	Beauty	40

HIGH-VALUE TRANSACTIONS: SALES ABOVE 1000

```sql
-- 5.Write a SQL query to find all transactions where the total_sale is greater than 1000:
SELECT *
FROM Retail_Sales
WHERE total_sale > 1000;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

transactions_id	sale_date	sale_time	customer_id	gender	age	category	quantiy	price_per_unit	cogs	total_sale
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
31	2023-12-31	17:47:00	3	Male	44	Electronics	4	300	129	1200
46	2022-11-08	17:50:00	54	Female	20	Electronics	4	300	84	1200
47	2022-10-22	17:22:00	96	Female	40	Beauty	3	500	600	1500
54	2022-10-20	10:17:00	142	Female	38	Electronics	3	500	200	1500
58	2023-09-16	19:18:00	53	Male	18	Clothing	4	300	75	1200
65	2022-12-11	20:03:00	84	Male	51	Electronics	4	500	160	2000
67	2023-08-19	20:19:00	119	Female	48	Beauty	4	300	129	1200
72	2023-12-06	19:19:00	5	Female	20	Electronics	4	500	195	2000
74	2023-10-05	19:50:00	56	Female	18	Beauty	4	500	205	2000
78	2023-02-17	21:08:00	68	Female	47	Clothing	3	500	265	1500
89	2023-12-30	21:15:00	117	Female	55	Electronics	4	500	590	2000
93	2022-01-25	20:52:00	148	Female	35	Beauty	4	500	140	2000
99	2023-11-19	15:12:00	71	Female	50	Electronics	4	300	132	1200

GENDER INSIGHTS: TRANSACTION TRENDS ACROSS CATEGORIES

```
88      -- 6.Write a SQL query to find the total number of transactions (transaction_id) made by each gender in each category.:
89
90 •    SELECT category, gender, COUNT(transactions_id) AS total_transactions
91      FROM Retail_Sales
92      GROUP BY category, gender
93      ORDER BY category, gender;
94
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

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

SEASONAL HIGHLIGHTS: BEST-SELLING MONTHS UNVEILED BASED ON AVERAGE SALES

```sql
-- 7.Write a SQL query to calculate the average sale for each month. Find out best selling month in each year
SELECT *
FROM (
    SELECT
        YEAR(sale_date) AS 'Year',
        MONTH(sale_date) AS 'Month',
        ROUND(AVG(total_sale), 2) AS 'avg sale for month',
        RANK() OVER (PARTITION BY YEAR(sale_date) ORDER BY ROUND(AVG(total_sale), 2) DESC) AS ranks
    FROM retail_sales
    GROUP BY YEAR(sale_date), MONTH(sale_date)
) AS T1
WHERE ranks = 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Year	Month	avg sale for month	ranks
2022	7	541.34	1
2023	2	535.53	1

TOP PERFORMERS: LEADING CUSTOMERS BY REVENUE

```sql
109     -- 8.Write a SQL query to find the top 5 customers based on the highest total sales
110 •   SELECT customer_id, SUM(total_sale) AS 'sales'
111     FROM retail_sales
112     GROUP BY customer_id
113     ORDER BY SUM(total_sale) DESC
114     LIMIT 5;
115
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

customer_id	sales
3	38440
1	30750
5	30405
2	25295
4	23580

CATEGORY REACH: UNIQUE CUSTOMER COUNTS

```sql
-- 9.Write a SQL query to find the number of unique customers who purchased items from each category.
SELECT DISTINCT
    COUNT(customer_id) AS 'unique customers', category
FROM
    retail_sales
GROUP BY category;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

unique customers	category
611	Beauty
698	Clothing
678	Electronics

TIME MATTERS: ORDER PATTERNS BY SHIFT

```sql
-- 10.Write a SQL query to create each shift and number of orders (Example Morning less than 12, Afternoon Between 12 & 17, Evening more than 17)
SELECT
    CASE
        WHEN HOUR(sale_time) < 12 THEN 'Morning'
        WHEN HOUR(sale_time) BETWEEN 12 AND 17 THEN 'Afternoon'
        ELSE 'Evening'
    END AS shift, COUNT(*) AS 'num_orders'
FROM retail_sales
GROUP BY shift
ORDER BY FIELD(shift, 'Morning', 'Aftertoon', 'Evening');
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

shift	num_orders
Afternoon	377
Morning	548
Evening	1062

Key Takeaways

1. Category Performance: "Clothing" emerged as a top-performing category with significant sales volume."Beauty" category customers had a higher average age compared to others.

2. Time-Shift Analysis: Morning and afternoon shifts saw the highest number of transactions, revealing peak shopping times.

3. Customer Trends: A small percentage of customers accounted for a large portion of total sales, highlighting the importance of loyal customers.

4. Sales Peaks: November recorded higher sales due to increased shopping activity, potentially influenced by seasonal demand.


THANK YOU