SQL PROJECT

Retail Sales Analysis

Creating Database sq_project_p1

Query :

```
CREATE  DATABASE sq_project_p1;
```

Using Database sq_project_p1

Query :

```
USE sq_project_p1;
```

Creating Table retail_sales

Query :

```
CREATE TABLE retail_sales (
    transactions_id INT PRIMARY KEY,
    sale_date DATE,
    sale_time TIME,
    customer_id INT,
    gender VARCHAR(15),
    age INT,
    category VARCHAR(20),
    quantity INT,
    price_per_unit FLOAT,
    cogs FLOAT,
    total_sale FLOAT
);
```

Imported retail_sales_csv data from Import wizard.

Data imported successfully.

Checking the imported data in Table form.

Query:

```
SELECT * FROM  retail_sales;
```

Output:

transactions_id	sale_date	sale_time	customer_id	gender	age	category	quantity	price_per_unit	cogs	total_sale
1	2022-12-16	19:10:00	50	Male	34	Beauty	3	50	16	150
2	2022-06-24	10:07:00	104	Female	26	Clothing	2	500	135	1000
3	2022-06-14	07:08:00	114	Male	50	Electronics	1	30	8.1	30
4	2023-08-27	18:12:00	3	Male	37	Clothing	1	500	200	500
5	2023-09-05	22:10:00	3	Male	30	Beauty	2	50	24	100
6	2023-11-15	22:16:00	2	Female	45	Beauty	1	30	15	30
7	2023-07-06	06:24:00	38	Male	46	Clothing	2	25	13.25	50
8	2022-12-27	11:19:00	148	Male	30	Electronics	4	25	11	100
9	2022-12-02	13:12:00	85	Male	63	Electronics	2	300	78	600
10	2022-10-24	22:55:00	81	Female	52	Clothing	4	50	62.5	200
11	2022-02-27	10:30:00	151	Male	23	Clothing	2	50	23.5	100
12	2022-12-09	22:09:00	114	Male	35	Beauty	3	25	25.25	75
13	2023-02-08	17:43:00	106	Male	22	Electronics	3	500	245	1500
14	2022-05-18	07:11:00	8	Male	64	Clothing	4	30	13.2	120
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
17	2023-02-25	21:08:00	3	Female	27	Clothing	4	25	13	100

Count of records in the table retail_sales

Query:

```
SELECT COUNT(*) AS _Total_records FROM  retail_sales;
```

Output:

_Total_records
1987

Data Cleaning checking for any null values in the data.

Query:

```
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  quantity  IS NULL
   OR  price_per_unit IS NULL
   OR  cogs IS NULL
   OR  total_sale IS NULL
   ;
```

Output:

transactions_id	sale_date	sale_time	customer_id	gender	age	category	quantity	price_per_unit	cogs	total_sale
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Data Exploration / Findings.

HOW MANY SALES WE HAVE ??

Query :

```
select COUNT(*) as Total_sales from retail_sales;
```

Output :

Total_sales
1987

HOW MANY CUSTOMER WE HAVE??

Query :

```
select COUNT(DISTINCT customer_id) as Total_customer from retail_sales;
```

Output :

Total_customer
155

HOW MANY CATEGORY WE HAVE??

Query :

```
select COUNT(DISTINCT category) as Total_category from retail_sales;
```

Output :

Total_category
3

ANALYSIS & KEY BUSINESS QUESTIONS.

1. SALES FOR A SPECIFIC DATA LIKE '2022-12-16'.

Query:

```
SELECT * FROM retail_sales
WHERE sale_date = '2022-12-16';
```

Output :

transactions_id	sale_date	sale_time	customer_id	gender	age	category	quantity	price_per_unit	cogs	total_sale
1	2022-12-16	19:10:00	50	Male	34	Beauty	3	50	16	150
786	2022-12-16	20:38:00	78	Male	48	Clothing	4	25	28.5	100
1077	2022-12-16	18:45:00	141	Female	47	Clothing	2	50	15	100
1284	2022-12-16	11:43:00	145	Male	43	Clothing	4	50	18	200
1544	2022-12-16	15:58:00	70	Female	27	Electronics	1	25	11.25	25
1935	2022-12-16	22:40:00	96	Female	34	Beauty	1	50	47.5	50

2. ALL THE TRANSACTIONS WHERE CATEGORY IS CLOTHING AND THE QUANTITY SOLD IS MORE THAN 3 FOR THE MONTH OF MAY-2022.

Query:

```
SELECT * FROM retail_sales
WHERE
    category = 'Clothing' AND quantity >= 3
        AND sale_date >= '2022-05-01'
        AND sale_date < '2022-06-01';
```

Output :

transactions_id	sale_date	sale_time	customer_id	gender	age	category	quantity	price_per_unit	cogs	total_sale
14	2022-05-18	07:11:00	8	Male	64	Clothing	4	30	13.2	120
288	2022-05-02	09:59:00	27	Male	28	Clothing	4	30	12	120
439	2022-05-06	21:46:00	42	Male	50	Clothing	3	25	11.75	75
474	2022-05-08	17:57:00	145	Female	26	Clothing	3	500	210	1500
1163	2022-05-04	10:52:00	120	Female	64	Clothing	3	50	27	150
1264	2022-05-27	09:23:00	82	Male	47	Clothing	3	300	123	900
1422	2022-05-07	20:41:00	35	Female	28	Clothing	3	30	13.5	90
1474	2022-05-15	20:49:00	84	Female	26	Clothing	3	500	255	1500

3. CALCULATING THE TOTAL SALES (total_sale) & TOTAL ORDERS FOR EACH CATEGORY.

Query:

```
SELECT
    category,
    sum(total_sale) AS Total_sales,
    count(*)  AS Total_orders
FROM retail_sales
GROUP BY category;
```

Output :

category	Total_sales	Total_orders
Beauty	286790	611
Clothing	309995	698
Electronics	311445	678

4. FIND THE TOTAL SALE FOR BOTH THE YEAR.

Query:

```
SELECT
    YEAR(sale_date) AS YEAR,
    SUM(total_sale) AS SALE
FROM
    retail_sales
GROUP BY YEAR;
```

Output :

YEAR	SALE
2022	449335
2023	458895

5. AVERAGE AGE OF THE CUSTOMERS WHO HAVE PURCHASED THE ITEMS FROM CATEGORY 'beauty'.

Query:

```
SELECT
    category,
    round(AVG(age),2) As average_age
FROM
    retail_sales
WHERE
    category = 'beauty';
```

Output :

category	average_age
Beauty	40.42

6. FIND ALL THE TRANSACTIONS WHERE THE TOTAL SALES IS GREATER THAN 1500.

Query:

```
SELECT * FROM  retail_sales
WHERE total_sale > 1500;
```

Output :

transactions_id	sale_date	sale_time	customer_id	gender	age	category	quantity	price_per_unit	cogs	total_sale
15	2022-07-01	11:50:00	75	Female	42	Electronics	4	500	210	2000
65	2022-12-11	20:03:00	84	Male	51	Electronics	4	500	160	2000
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
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
109	2023-09-06	19:57:00	94	Female	34	Electronics	4	500	560	2000
118	2023-03-13	20:07:00	3	Female	30	Electronics	4	500	270	2000
124	2022-12-24	21:17:00	83	Male	33	Clothing	4	500	515	2000
139	2023-09-15	14:03:00	113	Male	36	Beauty	4	500	230	2000
152	2022-06-16	11:58:00	120	Male	43	Electronics	4	500	210	2000
155	2023-07-18	18:05:00	3	Male	31	Electronics	4	500	150	2000
157	2022-05-15	21:59:00	98	Male	62	Electronics	4	500	170	2000
166	2023-01-28	11:42:00	32	Male	34	Clothing	4	500	225	2000
253	2022-09-30	21:26:00	66	Female	53	Clothing	4	500	525	2000
257	2022-12-10	08:49:00	130	Male	19	Beauty	4	500	165	2000
269	2022-09-19	11:31:00	87	Male	25	Clothing	4	500	250	2000

7. FIND THE NUMBER OF TRANSACTION MADE BY EACH GENDER IN EACH CATEGORY.

Query:

```
SELECT
    category,
    gender ,count(*) AS Total_transactoins
FROM
    retail_sales
GROUP BY category, gender
ORDER BY category ;
```

Output :

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

8. FIND THE AVERAGE SALE FOR EACH MONTH AND ALSO FIND THE BEST SELLING MONTH OF THE YEAR.

Query:

```
SELECT * FROM
    (SELECT
        YEAR(sale_date) AS YEAR,
        MONTHNAME(sale_date) AS MONTH,
        round(AVG(total_sale),2) AS avg_sale,
        RANK()
        OVER (PARTITION BY YEAR(sale_date)
        ORDER BY AVG(total_sale) DESC)
        AS RANK_NO
    FROM
        retail_sales
    GROUP BY
        YEAR,MONTH) AS T1
WHERE
    RANK_NO = 1 ;
```

Output :

YEAR	MONTH	avg_sale	RANK_NO
2022	July	541.34	1
2023	February	535.53	1

9. FIND THE TOP 5 CUSTOMERS BASED ON THE HIGHEST TOTAL SALES.

Query:

```
SELECT
    customer_id,
    sum(total_Sale) AS Total_Sale
FROM  retail_sales
GROUP BY customer_id
ORDER BY total_sale DESC
LIMIT 5 ;
```

Output :

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

10. FIND THE NO. OF UNIQUE CUSTOMER WHO HAVE PURCHASED FROM EACH CATEGORY.

Query:

```
SELECT
    category,
    count(distinct customer_id)
    AS unique_customer
FROM retail_sales
GROUP BY category;
```

Output :

category	unique_customer
Beauty	141
Clothing	149
Electronics	144

11. FIND THE SHIFT WISE SALES COUNT (FOR EXAMPLE : MORNING <12 , AFTERNOON BETWEEN 12 AND 17 , EVENING >17).

Query:

```sql
WITH hourly_sales AS (
    SELECT *,
           CASE
               WHEN HOUR(sale_time) < 12 THEN 'MORNING SHIFT'
               WHEN HOUR(sale_time) BETWEEN 12 AND 17 THEN 'AFTERNOON SHIFT'
               ELSE 'EVENING SHIFT'
           END AS shift
    FROM retail_sales
)
SELECT shift,
       COUNT(*) AS sales_count
FROM hourly_sales
GROUP BY shift;
```

Output :

shift	sales_count
EVENING SHIFT	1062
MORNING SHIFT	548
AFTERNOON SHIFT	377