Retail Sales – SQL P1

Creating database

```
show databases;

use sql_project1;
```

Creating table

```
-- Create Table
create table retail_sales(
    transaction_id int primary key,
    sale_date date,
    sale_time time,
    customer_id int,
    gender varchar(10),
    age int,
    category varchar(50),
    quantity int,
    price_per_unit int,
    cogs float,
    total_sales int
);
```

Insert data to the table

-- To import data

Go to Tables -> right click -> Table data import wizard -> file path -> use existing table -> next and finish

Data Exploration

```
22
23 •  select * from retail_sales;
```

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

transaction_id	sale_date	sale_time	customer_id	gender	age	category	quantity	price_per_unit	cogs	total_sales
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

retail_sales 37 ×

How Many sales we have

```
24
25      -- How many sales we have?
26 •    select sum(total_sales) from retail_sales;
```

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

sum(total_sales)
908230

How many Customers we have?

```
28      -- How may customers we have?
29 •    select count(distinct customer_id) as no_of_cust from retail_sales;
30
```

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

no_of_cust
155

Check for null values

```sql
-- Check for null values
select * from retail_sales
where transaction_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 price_per_unit is null
    or total_sales is null;
```

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

transaction_id	sale_date	sale_time	customer_id	gender	age	category	quantity	price_per_unit	cogs	total_sales
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Data Analysis & Problems & Answers

1. Write a SQL query to retrieve all columns for sales made on '2022-11-05'.

```
45      -- 1. Write a SQL query to retrieve all columns for sales made on '2022-11-05'
46 •    select * from retail_sales
47      where sale_date = '2022-11-05';
```

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

transaction_id	sale_date	sale_time	customer_id	gender	age	category	quantity	price_per_unit	cogs	total_sales
180	2022-11-05	10:47:00	117	Male	41	Clothing	3	300	129	900
214	2022-11-05	16:31:00	53	Male	20	Beauty	2	30	8.1	60
240	2022-11-05	11:49:00	95	Female	23	Beauty	1	300	123	300
856	2022-11-05	17:43:00	102	Male	54	Electronics	4	30	9.3	120
943	2022-11-05	19:29:00	90	Female	57	Clothing	4	300	318	1200
1137	2022-11-05	22:34:00	104	Male	46	Beauty	2	500	145	1000
1256	2022-11-05	09:58:00	29	Male	23	Clothing	2	500	190	1000

retail_sales 42 ×

2. Write a SQL query to retrieve all transactions where the category is 'Clothing' and the quantity sold is more than 10 in the month of Nov-2022.

```
49      -- 2. Write a SQL query to retrieve all transactions where the category is 'Clothing' and the quantity sold is more
50      -- than or equal to 4 in the month of Nov-2022
51 •    select * from retail_sales
52      where category = 'Clothing'
53      and quantity >= 4
54      and year(sale_date) = 2022
55      and month(sale_date) = 11;
```

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

transaction_id	sale_date	sale_time	customer_id	gender	age	category	quantity	price_per_unit	cogs	total_sales
64	2022-11-15	06:34:00	7	Male	49	Clothing	4	25	8.5	100
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
284	2022-11-12	09:17:00	129	Male	43	Clothing	4	50	20.5	200
547	2022-11-14	07:36:00	3	Male	63	Clothing	4	500	250	2000
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

retail_sales 43 × Apply

3. Write a SQL query to calculate total sales for each category.

```sql
-- 3. Write a SQL query to calculate total sales for each category
select category, sum(total_sales) as sales_by_category
from retail_sales
group by category;
```

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

category	sales_by_category
Beauty	286790
Clothing	309995
Electronics	311445

4. Write a SQL query to find the average age of customers who purchase items from the 'Beauty' category.

```sql
-- 4. Write a SQL query to find the average age of customers who purchase items from the 'Beauty' category
select round(avg(age)) as avg_age
from retail_sales
where category = 'Beauty'
group by category;
```

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

avg_age
40

5. Write a SQL query to find all transactions where the total sales is > 1000.

```sql
-- 5. Write a SQL query to find all transactions where the total sales is > 1000
select * from retail_sales
where total_sales > 1000;
```

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

transaction_id	sale_date	sale_time	customer_id	gender	age	category	quantity	price_per_unit	cogs	total_sales
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

retail_sales 49 ×

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

```sql
-- 6. Write a SQL query to find the total number of transactions (transaction_id) made by each gender in each category
select category,gender, count(transaction_id) as total_no_of_transactions
from retail_sales
group by gender, category;
```

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

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

Result 50

7. Write a SQL query to calculate average sales for each month. Find out bestselling month in each year.

```sql
-- 7. Write a SQL query to calculate average sales for each month. Find out best selling month in each year
select * from
    (
    select
        year(sale_date) as year,
        month(sale_date) as month,
        avg(total_sales) as avg_sales,
        rank() over(partition by year(sale_date) order by avg(total_sales) desc) as rank1
    from retail_sales
    group by year(sale_date), month(sale_date)
    ) as t1
where rank1 = 1;
```

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

year	month	avg_sales	rank1
2022	7	541.3415	1
2023	2	535.5319	1

8. Write a SQL query to find the top 5 customers based on the highest total sales.

```sql
-- 8. Write a SQL query to find the top 5 customers based on the highest total sales
select customer_id, sum(total_sales) as total_sales
from retail_sales
group by customer_id
order by total_sales desc
limit 5;
```

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

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

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

```sql
-- 9. Write a SQL query to find the number of unique customers who purchased items from each category
select category, count(distinct customer_id) as cnt_distinct_cust
from retail_sales
group by category;
```

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

category	cnt_distinct_cust
Beauty	141
Clothing	149
Electronics	144

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

```sql
-- 10. Write a SQL query to create each shift and number of orders (Example - Morning<=12, Afternoon between 12 & 17, Evening > 17)
with hourly_sales as
(
    select
        case
            when sale_time >= '05-00-00' and sale_time < '12-00-00' then 'Morning'
            when sale_time >= '12-00-00' and sale_time < '17-00-00' then 'Afternoon'
            else 'Evening'
        end as shift
    from retail_sales
)
select shift, count(*) as no_of_orders
from hourly_sales
group by shift;
```

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

shift	no_of_orders
Evening	1275
Morning	548
Afternoon	164