SQL QUERIES

BASIC QUERIES:

```
select * from walmart;
select count(*) from walmart;
select payment_method , count(*)
from walmart group by payment_method order by count(*) desc;

select count(Branch) from walmart;
select count(distinct Branch) from walmart;
select max(quantity) from walmart;
select max(total) from walmart;
```

Business problems

-- Q1: Find different payment methods, number of transactions, and quantity sold by payment method

```
select  payment_method,
count(*) as no_of_trans,
sum(quantity) as no_of_quantity
from walmart group by payment_method;
```

-- Question 2: Identify the highest-rated category in each branch
-- Display the branch, category, and avg rating

```
WITH category_avg AS (
    SELECT
        branch,
        category,
        AVG(rating) AS avg_rating
    FROM walmart
    GROUP BY branch, category
)
SELECT
    branch,
    category,
avg_rating,
    rnk
FROM (
    SELECT
        branch,
        category,
        avg_rating,
        RANK() OVER (PARTITION BY branch ORDER BY avg_rating DESC) AS rnk
    FROM category_avg
) AS ranked
WHERE rnk = 1;
```

-- Q3: Identify the busiest day for each branch based on the number of transactions

```
Select branch,day_name,no_of_transactions from(
Select branch ,
  dayname(STR_TO_DATE(date, '%d/%m/%y')) AS day_name,
count(*) as no_of_transactions,
Rank()OVER (Partition by branch order by count(*) Desc) as rnk
from walmart
Group by branch,day_name
 ) as ranked
 where rnk = 1;
```

-- Q4: Calculate the total quantity of items sold per payment method

```
select payment_method ,
(sum(quantity)) as total_quantity
from walmart group by payment_method;
```

-- Q5: Determine the average, minimum, and maximum rating of categories for each city

```
Select city ,category,
avg(rating) as average_rating ,
min(rating) as mininum_rating ,
max(rating) as maximum_rating
from walmart group by city,category;
```

-- Q6: Calculate the total profit for each category

```
select category , sum(unit_price*quantity*profit_margin) as total_profit
from walmart
group by category;
```

-- Q7: Determine the most common payment method for each branch

```
select branch , payment_method , total_trans
from(
select branch ,payment_method ,
count(*) as total_trans,
Rank()over(partition by branch order by count(*) desc) as rnk
from walmart
group by branch,payment_method
) as ranked
where rnk = 1;
```

-- Q8: Categorize sales into Morning, Afternoon, and Evening shifts

```
select
branch,
CASE
        WHEN HOUR(TIME(time)) < 12 THEN 'Morning'
        WHEN HOUR(TIME(time)) BETWEEN 12 AND 17 THEN 'Afternoon'
        ELSE 'Evening'
END as shift,
count(*) as num_invoices
from walmart
group by branch, shift
order by branch,num_invoices desc;
```

-- Q9: Identify the 5 branches with the highest revenue decrease ratio from last year to current year (e.g., 2022 to 2023)

```
WITH revenue_2022 AS (
    SELECT
        branch,
        SUM(total) AS revenue
    FROM walmart
    WHERE YEAR(STR_TO_DATE(date, '%d/%m/%Y')) = 2022
    GROUP BY branch
),
revenue_2023 AS (
    SELECT
        branch,
        SUM(total) AS revenue
    FROM walmart
```

```
   WHERE YEAR(STR_TO_DATE(date, '%d/%m/%Y')) = 2023
```

```
GROUP BY branch
)
SELECT
   r2022.branch,
   r2022.revenue AS last_year_revenue,
   r2023.revenue AS current_year_revenue,
   ROUND(((r2022.revenue - r2023.revenue) / r2022.revenue) * 100, 2) AS
revenue_decrease_ratio
FROM revenue_2022 AS r2022
JOIN revenue_2023 AS r2023 ON r2022.branch = r2023.branch
WHERE r2022.revenue > r2023.revenue
ORDER BY revenue_decrease_ratio DESC
LIMIT 5;
```

-- Q10: Determine category wise total_profit upto 2 decimal places

```
select category ,
round(sum(unit_price*quantity*profit_margin),2) as total_profit
from walmart
group by category  order by category asc ,total_profit desc;
```