- Business Problems

1. For each payment method find the number of transactions and quantity sold.
   Query:
   select payment_method, count(invoice_id) as Transactions, sum(quantity) as Quantity_Sold from walmart
   group by payment_method;

```
+----------------+--------------+---------------+
| payment_method | Transactions | Quantity_Sold |
+----------------+--------------+---------------+
| Ewallet        |         3881 |          8932 |
| Cash           |         1832 |          4984 |
| Credit card    |         4256 |          9567 |
+----------------+--------------+---------------+
```

   3 rows in set (0.03 sec)

2. Identify the highest rated category in each brand, displaying the branch, category and average rating.
   Query:
   select * from (select branch, category, avg(rating) as Rating, dense_RANK() OVER(PARTITION BY branch order by avg(rating) desc) as ranking from walmart group by branch, category) as r(must give alias to subquery in mysql) where ranking=1;

```
+---------+----------------------+------------------+---------+
| branch  | category             | Rating           | ranking |
+---------+----------------------+------------------+---------+
| WALM001 | Electronic accessories |             7.45 |       1 |
| WALM002 | Food and beverages     |             8.25 |       1 |
| WALM003 | Sports and travel      |              7.5 |       1 |
| WALM004 | Food and beverages     |              9.3 |       1 |
```

3. Select busiest day for each day depending upon number of transactions.
   Query:
   select * from (select branch, DAYNAME(STR_TO_DATE(date,'%d/%m/%y')) as day_name, count(*) as no_transactions, dense_rank() over(partition by branch order by count(*) desc) as ranking from walmart group by branch, day_name) as r where ranking=1;

```
+---------+-----------+-----------------+---------+
| branch  | day_name  | no_transactions | ranking |
+---------+-----------+-----------------+---------+
| WALM001 | Thursday  |              16 |       1 |
| WALM002 | Thursday  |              15 |       1 |
| WALM003 | Tuesday   |              33 |       1 |
```

4. Calculate the total quantity of item sold per payment method.
   Query:
   select payment_method, sum(quantity) from walmart group by payment_method;

```
+----------------+---------------+
| payment_method | sum(quantity) |
+----------------+---------------+
| Ewallet        |          8932 |
| Cash           |          4984 |
| Credit card    |          9567 |
+----------------+---------------+
```

   3 rows in set (0.03 sec)

5. Determine the average, minimum, and maximum rating of products for each city. List the city, average_rating, min_rating, and max_rating.
   Query:
   select city, category, avg(rating) as Avg_Rating, min(rating) as Min_Rating, max(rating) as Max_Rating from walmart group by city, category;

city	category	Avg_Rating	Min_Rating	Max_Rating
San Antonio	Health and beauty	7.05	5	9.1
Harlingen	Electronic accessories	9.6	9.6	9.6
Haltom City	Home and lifestyle	6.227777777777778	3	9.5

6. Calculate total profit for each category by considering total profit as(unit_price * quantity * profit_margin). List category, total_price, ordered from highest to lowest profit.
   Query:
   select category, sum(unit_price * quantity * profit_margin) as total_profit from walmart group by category;

category	total_profit
Health and beauty	18671.7345
Electronic accessories	30772.489499999978

7. Determine the most common payment method for each branch. Display branch and preferred payment_method.
   Query:
   select * from (select branch, payment_method, count(*), dense_rank() over(partition by branch order by count(*) desc) as ranking from walmart group by branch, payment_method) as r where ranking=1;

branch	payment_method	count(*)	ranking
WALM001	Ewallet	45	1
WALM002	Ewallet	37	1
WALM003	Credit card	115	1

   Using CTE:
   with cte as (select branch, payment_method, count(*), dense_rank() over(partition by branch order by count(*) desc) as ranking from walmart group by branch, payment_method) select * from cte where ranking=1;

branch	payment_method	count(*)	ranking
WALM001	Ewallet	45	1
WALM002	Ewallet	37	1
WALM003	Credit card	115	1
WALM004	Ewallet	44	1
WALM005	Ewallet	56	1

8. Categorize sales into 3 groups Morning, Afternoon and Evening. Find out shift according to number of invoices.
   Query:

```
SELECT *, CASE WHEN HOUR(`time`) < 12 THEN 'Morning'
WHEN HOUR(`time`) BETWEEN 12 AND 17 THEN 'Afternoon'
ELSE 'Evening'
```

END AS day_time FROM walmart;

invoice_id	branch	city	category	unit_price	quantity	date	time	payment_method	rating	profit_margin	total	day_time
1	WALM003	San Antonio	Health and beauty	74.69	7	05/01/19	13:08:00	Ewallet	9.1	0.48	522.83	Afternoon
2	WALM048	Harlingen	Electronic accessories	15.28	5	08/03/19	10:29:00	Cash	9.6	0.48	76.40	Morning
3	WALM067	Haltom City	Home and lifestyle	46.33	7	03/03/19	13:23:00	Credit card	7.4	0.33	324.31	Afternoon
4	WALM064	Bedford	Health and beauty	58.22	8	27/01/19	20:33:00	Ewallet	8.4	0.33	465.76	Evening
5	WALM013	Irving	Sports and travel	86.31	7	08/02/19	10:37:00	Ewallet	5.3	0.48	604.17	Morning

9. Identify 5 branches with the highest decrease ratio in revenue compared to last year(2022 to 2023).
   Query:
   ```
   WITH revenue_2022 AS (SELECT branch, SUM(total) AS total_2022 FROM walmart WHERE YEAR(STR_TO_DATE(date, '%d/%m/%y')) = 2022 GROUP BY branch), revenue_2023 AS (SELECT branch, SUM(total) AS total_2023 FROM walmart WHERE YEAR(STR_TO_DATE(date, '%d/%m/%y')) = 2023 GROUP BY branch) SELECT r22.branch, r22.total_2022, r23.total_2023, ROUND(((r22.total_2022 - r23.total_2023) / r22.total_2022) * 100, 2) AS percentage_drop FROM revenue_2022 r22 JOIN revenue_2023 r23 ON r22.branch = r23.branch WHERE r22.total_2022 > r23.total_2023 ORDER BY percentage_drop DESC limit 5;
   ```

branch	total_2022	total_2023	percentage_drop
WALM045	1731.00	647.00	62.62
WALM047	2581.00	1069.00	58.58
WALM098	2446.00	1030.00	57.89