Pizza Hut
Sales Report

Md Mujtaba

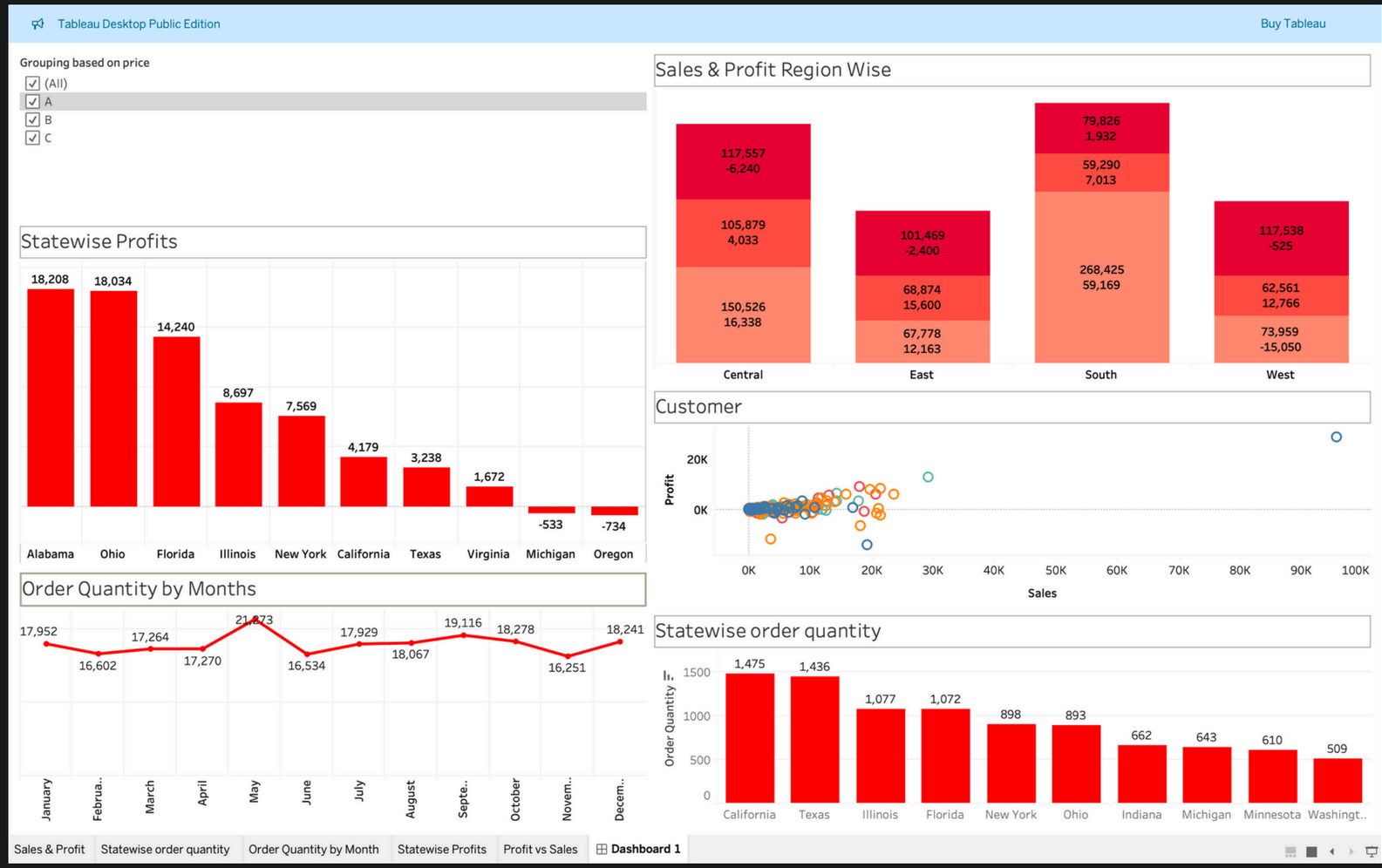
PROJECT OVERVIEW

This project integrates multiple SQL queries executed in MySQL Workbench with visual insights presented through a Tableau dashboard, combining data analysis, solutions, and outputs.

WHAT I DID

- ImImported sales data from CSV files into MySQL Workbench and structured it for analysis.
- Cleaned and validated datasets related to pizzas and order details for consistency and accuracy.
- Executed multiple SQL queries to explore sales trends, order patterns, and category performance.
- Visualised key insights and revenue breakdowns through interactive Tableau dashboards.

Tableau Dashboard



Sql Questions and Queries

```sql
-- Question: Retrieve the total number of orders placed.

-- Query
SELECT COUNT(*) as total_orders FROM ORDERS;
```

Output

total_orders
21350

```sql
-- Question: Calculate the total revenue generated from pizza sales.

-- Query
select
round (sum(order_details.quantity * pizzas.price),2) as total_sales
from order_details
join pizzas on pizzas.pizza_id = order_details.pizza_id;
```

Output

total_sales
817860.05

```sql
-- Question: Identify the highest-priced pizza.

-- Query
SELECT * FROM pizzas
WHERE price = (SELECT MAX(price) FROM pizzas);
```

Output

pizza_id	pizza_type_id	size	price
the_greek_xxl	the_greek	XXL	35.95

```sql
-- Question: Identify the most common pizza size ordered.

-- Query
SELECT
    pz.size AS Pizza_size, COUNT(pz.size) AS size_sold
FROM
    order_details od
        JOIN
    pizzas pz ON pz.pizza_id = od.pizza_id
GROUP BY pz.size
ORDER BY size_sold DESC
LIMIT 1;
```

Output

Pizza_size	size_sold
L	18526

```sql
-- Question: List the top 5 most ordered pizza types along with their quantities.

-- Query:
SELECT
    pizza_id, SUM(quantity) AS Quantities
FROM
    order_details
GROUP BY pizza_id
ORDER BY Quantities DESC
LIMIT 5;
```

Output

pizza_id	Quantities
big_meat_s	1914
thai_ckn_l	1410
five_cheese_l	1409
four_cheese_l	1316
classic_dlx_m	1181

```sql
-- Question: Join the necessary tables to find the total quantity of each pizza category ordered.

-- Query
select
pzts.category,
sum(od.quantity)
from order_details od
join pizzas pzs on od.pizza_id = pzs.pizza_id
join pizza_type_sheet1 pzts on pzs.pizza_type_id = pzts.pizza_type_id
group by pzts.category;
```

Output

category	sum(od.quanti...
Classic	14888
Veggie	11649
Supreme	11987
Chicken	11050

```sql
-- Question: Determine the distribution of orders by hour of the day.

-- Query
select hour(order_time) as hour, count(order_id) from orders
group by hour(order_time);

SELECT
    CASE
        WHEN order_time BETWEEN '11:00:00' AND '15:00:00' THEN 'Noon'
        WHEN order_time BETWEEN '15:00:00' AND '18:00:00' THEN 'Afternoon'
        WHEN order_time BETWEEN '18:00:00' AND '21:00:00' THEN 'Evening'
        WHEN order_time BETWEEN '21:00:00' AND '23:59:59' THEN 'Late Night'
        WHEN order_time BETWEEN '00:00:00' AND '04:00:00' THEN 'Midnight'
        ELSE 'Morning'
    END AS time_period,
    count(*) as Order_distribution
FROM orders
group by time_period;
```

Output

hour	count(order_i...
9	1
10	8
11	1231
12	2520
13	2455
14	1472
15	1468

```sql
-- Question: Join relevant tables to find the category-wise distribution of pizzas.

-- Query
select
category,
sum(quantity) as Pizza_category
from order_details od
join pizzas pz on pz.pizza_id = od.pizza_id
join pizza_type_sheet1 pzts on pzts.pizza_type_id = pz.pizza_type_id
group by category
order by pizza_category desc;
```

OUTPUT

category	Pizza_category
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

```sql
-- Question: Group the orders by date and calculate the average number of pizzas ordered per day.

-- Query
SELECT
    AVG(daily_pizzas) AS avg_pizzas_per_day
FROM (
    SELECT
        o.order_date,
        SUM(od.quantity) AS daily_pizzas
    FROM orders o
    JOIN order_details od ON o.order_id = od.order_id
    GROUP BY o.order_date
) AS daily_summary;
```

Output

avg_pizzas_per_d...
138.4749

```sql
-- Question: Determine the top 3 most ordered pizza types based on revenue.

-- Query
SELECT
    pi.pizza_id,
    pizza_sub_quantity.pizza_quantity * pi.price AS total_revenue
FROM (
    SELECT
        pizza_id,
        SUM(quantity) AS pizza_quantity
    FROM order_details
    GROUP BY pizza_id
) AS pizza_sub_quantity
JOIN pizzas pi ON pi.pizza_id = pizza_sub_quantity.pizza_id
order by total_revenue desc
limit 3;
```

OUTPUT

pizza_id	total_revenue
thai_ckn_l	29257.5
five_cheese_l	26066.5
four_cheese_l	23622.2

```sql
-- Question: Calculate the percentage contribution of each pizza type to total revenue.

-- Query
SELECT
    pi.pizza_id,
    (pizza_sub_quantity.pizza_quantity * pi.price) AS Pizza_revenue,
    (pizza_sub_quantity.pizza_quantity * pi.price) / round(sum(pizza_sub_quantity.pizza_quantity * pi.price),2) AS percent_revenue
FROM (
    SELECT
        pizza_id,
        SUM(quantity) AS pizza_quantity
    FROM order_details
    GROUP BY pizza_id
) AS pizza_sub_quantity
JOIN pizzas pi ON pi.pizza_id = pizza_sub_quantity.pizza_id
order by Pizza_revenue desc;
```

OUTPUT:

pizza_id	Pizza_revenue	Percent
thai_ckn_l	29257.5	3.58
five_cheese_l	26066.5	3.19
four_cheese_l	23622.2	2.89
spicy_ital_l	23011.75	2.81
big_meat_s	22968	2.81
southw_ckn_l	21082	2.58
bbq_ckn_l	20584	2.52
cali_ckn_l	19235.25	2.35

```sql
-- Question: Analyze the cumulative revenue generated over time.

-- Query
SELECT
    o.order_date,
    SUM(od.quantity) AS pizzas_sold,
    SUM(od.quantity * p.price) AS daily_revenue,
    SUM(SUM(od.quantity * p.price)) OVER (ORDER BY o.order_date) AS cumulative_revenue
FROM order_details od
JOIN pizzas p ON p.pizza_id = od.pizza_id
JOIN orders o ON o.order_id = od.order_id
GROUP BY o.order_date
ORDER BY o.order_date;
```

OUTPUT:

order_date	pizzas_sold	daily_revenue	cumulative_revenue
2015-01-01	162	2713.8500000…	2713.8500000000…
2015-01-02	165	2731.8999999…	5445.75
2015-01-03	158	2662.3999999…	8108.15
2015-01-04	106	1755.4500000…	9863.6
2015-01-05	125	2065.95	11929.55
2015-01-06	147	2428.95	14358.5
2015-01-07	138	2202.2000000…	16560.7

```sql
-- Question: Determine the top 3 most ordered pizza types based on revenue for each pizza category.

-- Query
SELECT
    category, pizza_id, total_revenue
FROM (
    SELECT
        pi.pizza_id,
        (pizza_sub_quantity.pizza_quantity * pi.price) AS total_revenue,
        pzts.category,
        ROW_NUMBER() OVER (PARTITION BY pzts.category ORDER BY (pizza_sub_quantity.pizza_quantity * pi.price) DESC) AS rank_in_category
    FROM (
        SELECT
            pizza_id,
            SUM(quantity) AS pizza_quantity
        FROM order_details
        GROUP BY pizza_id
    ) AS pizza_sub_quantity
    JOIN pizzas pi ON pi.pizza_id = pizza_sub_quantity.pizza_id
    JOIN pizza_type_sheet1 pzts ON pi.pizza_type_id = pzts.pizza_type_id
) AS ranked
WHERE rank_in_category <= 3;
```

Output:

category	pizza_id	total_revenue
Chicken	thai_ckn_l	29257.5
Chicken	southw_ckn_l	21082
Chicken	bbq_ckn_l	20584
Classic	big_meat_s	22968
Classic	classic_dlx_m	18896
Classic	hawaiian_l	15163.5
Supreme	spicy_ital_l	23011.75
Supreme	ital_supr_m	15526.5
Supreme	ital_supr_l	15500.25
Veggie	five_cheese_l	26066.5
Veggie	four_cheese_l	23622.2
Veggie	mexicana_l	17556.75

Thank You