

Pizza Sales Report

Using MySQL WorkBench

Project Overview

This project is a combination of different queries along with their solutions and outputs from MySQL workBench.

What I Did

- Imported Sales Data as CSV files into the MySQL WorkBench.
- Cleaned and Verified data related to pizzas and order details
- Ran several queries to better understand the data using SQL.



Retrieve the total number of orders placed.

...

Query:

```
1 -- Retrieve the total number of orders placed.  
2  
3 • select count(order_id) from orders;
```

Output:

count(order_id)
21350

Calculate the total revenue generated from pizza sales.

...

Query:

```
1  -- Calculate the total revenue generated from pizza sales.  
2  
3 • SELECT  
4   ⌈   ROUND(SUM(order_details.quantity * pizzas.price),  
5   ⌋   2) AS total_sales  
6  FROM  
7    order_details  
8    JOIN  
9    pizzas ON pizzas.pizza_id = order_details.pizza_id;  
10  
11
```

Output:

total_sales
817860.05

Identify the highest-priced pizza.

...

Query:

```
1  -- Identify the highest-priced pizza.  
2  
3 • SELECT  
4      pizza_types.name, pizzas.price  
5  FROM  
6      pizza_types  
7      JOIN  
8      pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
9  ORDER BY pizzas.price DESC  
10 LIMIT 1;  
11  
12
```

Output:

name	price
The Greek Pizza	35.95

Identify the most common pizza size ordered.

•••

Query:

```
1 -- Identify the most common pizza size ordered.  
2  
3 • SELECT  
4     pizzas.size,  
5     COUNT(order_details.order_details_id) AS order_count  
6 FROM  
7     pizzas  
8     JOIN  
9     order_details ON pizzas.pizza_id = order_details.pizza_id  
10 GROUP BY pizzas.size  
11 ORDER BY order_count DESC  
12 LIMIT 1;  
13  
14
```

Output:

	size	order_count
	L	18526

List the top 5 most ordered pizza types along with their quantities.

...

Query:

```
1 -- List the top 5 most ordered pizza types
2 -- along with their quantities.
3
4 • SELECT
5     pizza_types.name, SUM(order_details.quantity) AS qty
6 FROM
7     pizza_types
8     JOIN
9     pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
10    JOIN
11       order_details ON order_details.pizza_id = pizzas.pizza_id
12 GROUP BY pizza_types.name
13 ORDER BY qty DESC
14 LIMIT 5;
15
```

Output:

name	count
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

Join the necessary tables to find the total quantity of each pizza category ordered.

...

Query:

```
1  -- Join the necessary tables to find the
2  -- total quantity of each pizza category ordered.
3
4 • SELECT
5      pizza_types.category, SUM(order_details.quantity) AS qty
6  FROM
7      pizza_types
8          JOIN
9      pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
10         JOIN
11     order_details ON pizzas.pizza_id = order_details.pizza_id
12  GROUP BY pizza_types.category
13  ORDER BY qty DESC;
14
```

Output:

	category	qty
	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

Determine the distribution of orders by hour of the day.

...

Query:

```
1  -- Determine the distribution of orders by hour of the day.  
2  
3 • SELECT  
4      HOUR(order_time) AS hour, COUNT(order_id) AS order_count  
5  FROM  
6      orders  
7  GROUP BY HOUR(order_time)  
8  ORDER BY HOUR(order_time);  
9  
10
```

Output:

hour	order_count
9	1
10	8
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28

Join relevant tables to find the category-wise distribution of pizzas.

...

Query:

```
1 -- Join relevant tables to find the category-wise distribution of pizzas.  
2  
3 • SELECT  
4     category, COUNT(pizza_type_id)  
5 FROM  
6     pizza_types  
7 GROUP BY category;  
8
```

Output:

category	count(pizza_type_id)
Chicken	6
Classic	8
Supreme	9
Veggie	9

Group the orders by date and calculate the average number of pizzas ordered per day.

...

Query:

```
1 -- Group the orders by date and
2 -- calculate the average number of pizzas ordered per day.
3
4 • SELECT
5     ROUND(AVG(qty), 0) as avg_pizza_order_per_day
6 FROM
7     (SELECT
8         orders.order_date AS date,
9         SUM(order_details.quantity) AS qty
10    FROM
11        orders
12    JOIN order_details ON orders.order_id = order_details.order_id
13    GROUP BY date) AS order_qty;
```

Output:

avg_pizza_order_per_day
138

Determine the top 3 most ordered pizza types based on revenue.

...

Query:

```
1 -- Determine the top 3 most ordered pizza types based on revenue.  
2  
3 • SELECT  
4     pizza_types.name AS name,  
5     SUM((order_details.quantity * pizzas.price)) AS revenue  
6 FROM  
7     order_details  
8         JOIN  
9     pizzas ON order_details.pizza_id = pizzas.pizza_id  
10        JOIN  
11     pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id  
12 GROUP BY name  
13 ORDER BY revenue DESC  
14 LIMIT 3;
```

Output:

	name	revenue	
	The Thai Chicken Pizza	43434.25	
	The Barbecue Chicken Pizza	42768	
	The California Chicken Pizza	41409.5	

Calculate the percentage contribution of each pizza type to total revenue.

...

Query:

```
1  -- Calculate the percentage contribution of each pizza type to total revenue.
2
3 • SELECT
4      pizza_types.category AS category,
5      ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
6          ROUND(SUM(order_details.quantity * pizzas.price),
7          2)
8      FROM
9          order_details
10     JOIN
11         pizzas ON order_details.pizza_id = pizzas.pizza_id) * 100,
12  2) AS revenue
13 FROM
14     order_details
15     JOIN
16         pizzas ON order_details.pizza_id = pizzas.pizza_id
17     JOIN
18         pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
19 GROUP BY category
20 ORDER BY revenue;
```

Output:

category	revenue
Veggie	23.68
Chicken	23.96
Supreme	25.46
Classic	26.91

Analyze the cumulative revenue generated over time.

...

Query:

```
1 -- Analyze the cumulative revenue generated over time.  
2  
3 • select date,  
4 sum(revenue) over (order by date) as cum_revenue  
5 from  
6 (select date(orders.order_date) as date,  
7 sum(order_details.quantity * pizzas.price) as revenue  
8 from order_details join pizzas  
9 on order_details.pizza_id = pizzas.pizza_id  
10 join orders  
11 on orders.order_id = order_details.order_id  
12 group by date) as sales;  
13
```

Output:

date	cum_revenue
2015-01-01	2713.850000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399.05
2015-01-09	21526.4
2015-01-10	23990.35000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.30000000003
2015-01-14	32358.70000000004
2015-01-15	34343.5000000001
2015-01-16	36937.6500000001
2015-01-17	39001.7500000001
2015-01-18	40978.6000000006
2015-01-19	43365.7500000001
2015-01-20	45763.6500000001
2015-01-21	47804.2000000001
2015-01-22	50300.9000000001
2015-01-23	52724.6000000006
2015-01-24	55013.8500000006
2015-01-25	56631.4000000001
2015-01-26	58515.8000000001
2015-01-27	61043.8500000001
2015-01-28	63059.8500000001
2015-01-29	65105.15000000016
2015-01-30	67375.4500000001
2015-01-31	69793.3000000002
2015-02-01	72982.5000000001
2015-02-02	75311.1000000002
2015-02-03	77925.9000000002
2015-02-04	80159.8000000002
2015-02-05	82375.6000000002
2015-02-06	84885.5500000002
2015-02-07	87123.2000000001
2015-02-08	89158.2000000001
2015-02-09	91353.5500000002
2015-02-10	93410.0500000002
2015-02-11	95870.0500000002
2015-02-12	98028.8500000002
2015-02-13	100783.3500000002
2015-02-14	103102.5000000001
2015-02-15	105243.7500000001
2015-02-16	107212.5500000002
2015-02-17	109334.4500000001
2015-02-18	111977.3000000002
2015-02-19	114007.5500000002
2015-02-20	116898.7000000001
2015-02-21	119009.7000000001
2015-02-22	120589.6500000001
2015-02-23	122758.2000000001

•••

Determine the top 3 most ordered pizza types based on revenue for each pizza category.

Query:

```

1  -- Determine the top 3 most ordered pizza types based on revenue for each pizza category.
2
3 • select name,category,revenue
4   from
5   (select category, name, revenue,
6    rank() over(partition by category order by revenue desc) as rn
7   from
8   (select pizza_types.category, pizza_types.name,
9    sum(order_details.quantity * pizzas.price) as revenue
10  from order_details join pizzas
11  on order_details.pizza_id = pizzas.pizza_id
12  join pizza_types
13  on pizzas.pizza_type_id = pizza_types.pizza_type_id
14  group by pizza_types.category, pizza_types.name) as T) as b
15  where rn <= 3;
16

```

Output:

name	category	revenue
The Thai Chicken Pizza	Chicken	43434.25
The Barbecue Chicken Pizza	Chicken	42768
The California Chicken Pizza	Chicken	41409.5
The Classic Deluxe Pizza	Classic	38180.5
The Hawaiian Pizza	Classic	32273.25
The Pepperoni Pizza	Classic	30161.75
The Spicy Italian Pizza	Supreme	34831.25
The Italian Supreme Pizza	Supreme	33476.75
The Sicilian Pizza	Supreme	30940.5
The Four Cheese Pizza	Veggie	32265.70000000065
The Mexicana Pizza	Veggie	26780.75
The Five Cheese Pizza	Veggie	26066.5

...



Thank You