Maven Pizza Challenge

Domain: Sales

Tools: MySQL, MS-Word

A year's worth of sales from a fictitious pizza place, including the date and time of each order and the pizzas served, with additional details on the type, size, quantity, price, and ingredients.

Recommended Analysis

1. How many customers do we have each day? Are there any peak hours?
2. How many pizzas are typically in an order? Do we have any bestsellers?
3. How much money did we make this year? Can we identify any seasonality in the sales?
4. Are there any pizzas we should take of the menu, or any promotions we could leverage?

Pizza Place Sales

```sql
#1 Retrieve the total number of orders placed.
SELECT COUNT(*) AS total_orders
FROM orders;
```

	total_orders
▶	21350

```sql
#2 Calculate the total revenue generated from pizza sales.
SELECT ROUND(SUM(price * quantity), 2) AS rev
FROM order_details o
JOIN pizzas p
ON o.pizza_id = p.pizza_id;
```

	rev
▶	817860.05

```sql
#3 Identify the highest-priced pizza.
SELECT name, price
FROM pizzas p
JOIN pizza_types t
ON p.pizza_type_id=t.pizza_type_id
ORDER BY price DESC
LIMIT 1;
```

	name	price
▶	The Greek Pizza	35.95

```sql
#4 Identify the most common pizza size ordered.
SELECT size, COUNT(*) AS cnt
FROM order_details o
JOIN pizzas p
ON o.pizza_id=p.pizza_id
GROUP BY size
ORDER BY cnt DESC;
```

size	cnt
L	18526
M	15385
S	14137
XL	544
XXL	28

```sql
#5 List the top 5 most ordered pizza types along with their quantities.
WITH cte1 AS(
SELECT pt.name, SUM(o.quantity) AS qty
FROM pizza_types pt
JOIN pizzas p
ON pt.pizza_type_id=p.pizza_type_id
JOIN order_details o
ON p.pizza_id=o.pizza_id
GROUP BY pt.name)
SELECT * FROM cte1
ORDER BY qty DESC
LIMIT 5;
```

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

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

```
SELECT pt.category, SUM(o.quantity) AS qty
FROM pizza_types pt
JOIN pizzas p
ON pt.pizza_type_id=p.pizza_type_id
JOIN order_details o
ON p.pizza_id=o.pizza_id
GROUP BY category
ORDER BY qty DESC;
```

category	qty
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

#7 Determine the distribution of orders by hour of the day.

```
SELECT HOUR(order_time) AS hrs, COUNT(order_id) AS order_cnt
FROM orders
GROUP BY hrs
ORDER BY order_cnt DESC;
```

hrs	order_cnt
12	2520
13	2455
18	2399
17	2336
19	2009
16	1920
20	1642
14	1472
15	1468
11	1231
21	1198
22	663
23	28

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

```
SELECT category, COUNT(*) AS pizza_type
FROM pizza_types
GROUP BY category;
```

	category	pizza_type
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

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

```
WITH cte1 AS(
SELECT DATE(order_date) AS order_date, SUM(od.quantity) AS
pizza
FROM orders o
JOIN order_details od
ON o.order_id=od.order_id
GROUP BY order_date)
SELECT ROUND(AVG(pizza)) AS avg_pizza_order_per_day
FROM cte1;
```

	avg_pizza_order_per_day
▶	138

```
#10 Determine the top 3 most ordered pizza types
based on revenue.
SELECT pt.name, SUM((p.price*od.quantity)) AS rev
FROM pizza_types pt
JOIN pizzas p
ON pt.pizza_type_id=p.pizza_type_id
JOIN order_details od
ON p.pizza_id=od.pizza_id
GROUP BY pt.name
ORDER BY rev DESC
LIMIT 3;
```

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

```
#11 Calculate the percentage contribution of each pizza type
to total revenue.
WITH cte1 AS(
SELECT pt.category, ROUND(SUM(p.price*o.quantity)) AS rev
FROM pizza_types pt
JOIN pizzas p
ON pt.pizza_type_id=p.pizza_type_id
JOIN order_details o
ON o.pizza_id=p.pizza_id
GROUP BY pt.category),
cte2 AS(
SELECT category, rev*100/SUM(rev) OVER() AS pct_decimal
FROM cte1)
SELECT category, ROUND(pct_decimal,2) AS pct
FROM cte2;
```

category	pct
Classic	26.91
Veggie	23.68
Supreme	25.46
Chicken	23.96

```
#12 Analyze the cumulative revenue generated over
time.
WITH cte1 AS(
SELECT DATE(o.order_date) AS o_date,
ROUND(SUM(od.quantity*p.price),2) AS rev
FROM orders o
JOIN order_details od
ON o.order_id=od.order_id
JOIN pizzas p
ON od.pizza_id=p.pizza_id
GROUP BY o_date)
SELECT o_date AS order_date, SUM(rev) OVER(ORDER BY o_date)
AS cummulative_rev
FROM cte1;
```

order_date	cummulative_rev
2015-01-01	2713.85
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	21526.399999999998
2015-01-10	23990.35
2015-01-11	25862.649999999998
2015-01-12	27781.699999999997
2015-01-13	29831.299999999996

```
#13 Determine the top 3 most ordered pizza types
based on revenue for each pizza category.
WITH cte1 AS(
SELECT pt.category, pt.name,
SUM(p.price*o.quantity) AS rev
FROM pizza_types pt
JOIN pizzas p
ON pt.pizza_type_id=p.pizza_type_id
JOIN order_details o
ON o.pizza_id=p.pizza_id
GROUP BY pt.category, pt.name
),
cte2 AS(
SELECT *, RANK() OVER(PARTITION BY category ORDER BY rev
DESC) AS rn
FROM cte1)
SELECT * FROM cte2
WHERE rn<=3;
```

	category	name	rev	rn
▶	Chicken	The Thai Chicken Pizza	43434.25	1
	Chicken	The Barbecue Chicken Pizza	42768	2
	Chicken	The California Chicken Pizza	41409.5	3
	Classic	The Classic Deluxe Pizza	38180.5	1
	Classic	The Hawaiian Pizza	32273.25	2
	Classic	The Pepperoni Pizza	30161.75	3
	Supreme	The Spicy Italian Pizza	34831.25	1
	Supreme	The Italian Supreme Pizza	33476.75	2
	Supreme	The Sicilian Pizza	30940.5	3
	Veggie	The Four Cheese Pizza	32265.70000000065	1
	Veggie	The Mexicana Pizza	26780.75	2
	Veggie	The Five Cheese Pizza	26066.5	3

Resources

Icons Used in Dashboard.
https://www.flaticon.com/

Dataset Source.
https://mavenanalytics.io/blog/maven-pizza-challenge

Report by

Ankit Negi @ Data Analyst

Do you have any question related to report/dashboard?
Please email ankitnegi996@rocketmail.com