

Pizza Sales Analysis Using SQL

By Aarjan

07/08/2024

A photograph showing two young women with long brown hair sitting at a table, looking down at their smartphones. They are smiling and appear to be engaged in a shared activity on their screens. In the foreground, a large pizza with various toppings like pepperoni, cheese, and vegetables is partially visible on a black plate. The background is slightly blurred, showing other people and a restaurant setting.

Introduction

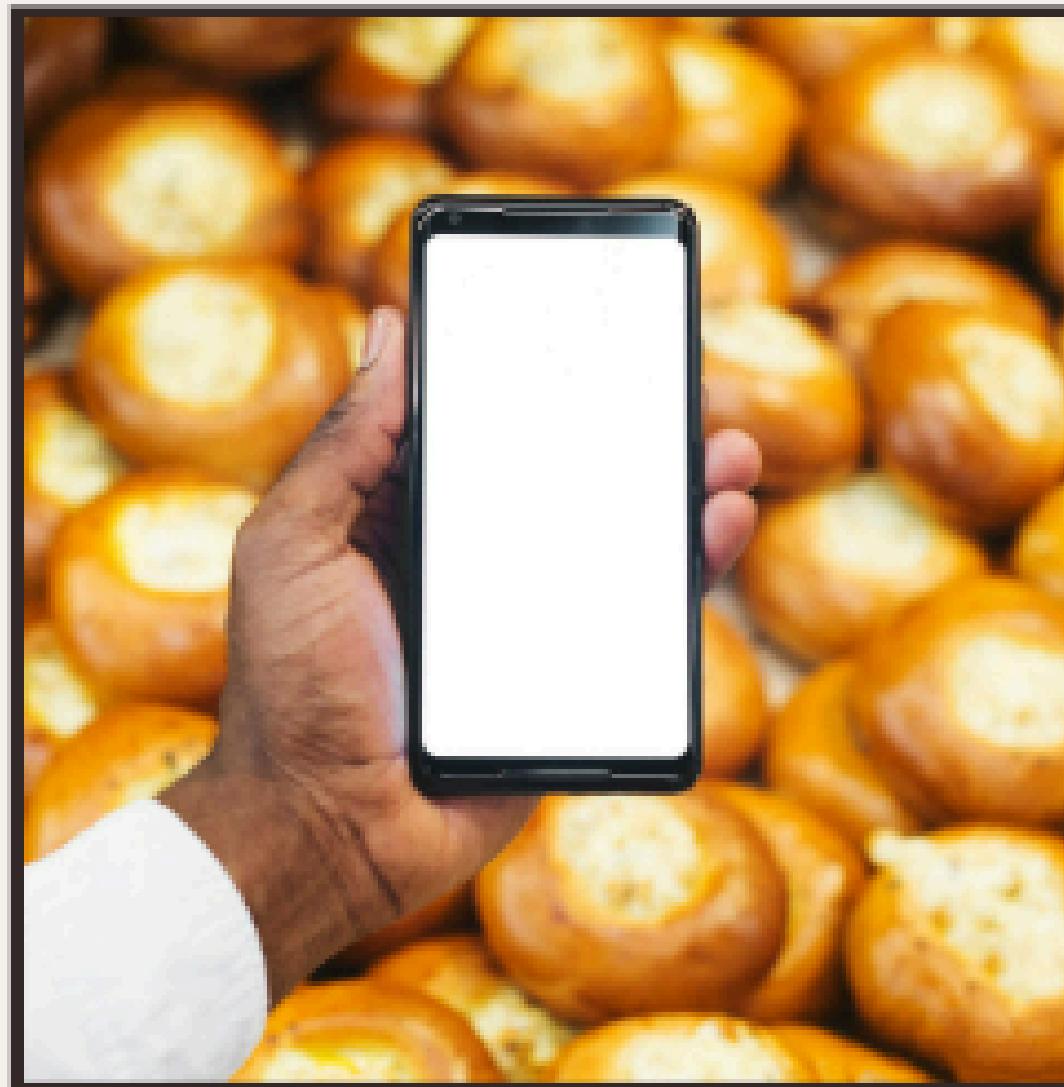
- Analyze pizza sales data to uncover business insights and opportunities
- Leverage SQL techniques to extract valuable insights from a comprehensive dataset
- Provide actionable recommendations to inform business decisions and drive growth



Database Structure

- orders table (21,351 rows): order_id, date, time
- order_details table (48,621 rows): order_details_id, order_id, pizza_id, quantity
- pizzas table (97 rows): pizza_id, pizza_type_id, size, price
- pizza_types table (33 rows): pizza_type_id, name, category, ingredients

total revenue generated from pizza sales



```
1  -- Calculate the total revenue generated from pizza sales.  
2 • SELECT  
3      round(sum(order_details.quantity * pizzas.price),2) as total_sales  
4  FROM  
5      order_details  
6      JOIN  
7      pizzas ON pizzas.pizza_id = order_details.pizza_id;
```

Result Grid | Filter Rows:

	total_sales
▶	817860.05



the total number of orders placed

-- Retrieve the total number of orders placed.

SELECT

COUNT(order_id) **AS** total_orders

FROM

orders;

| Result Grid |

	total_orders
▶	21350



Identify the highest-priced pizza.

```
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 =  
9         pizzas.pizza_type_id  
10    ORDER BY pizzas.price DESC  
11    LIMIT 1;
```

	name	price
▶	The Greek Pizza	35.95

Identify the most common pizza size ordered.



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

	size	order_count
▶	L	18526

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

```
SELECT
    pizza_types.name, SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5;
```

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



A close-up photograph of a person's hands holding a large pizza. The pizza has visible toppings such as pepperoni and mushrooms. The person is wearing a light-colored shirt.

find the total quantity of each pizza category ordered

SELECT

```
    pizza_types.category,  
    SUM(order_details.quantity) AS quantity
```

FROM

```
    pizza_types
```

JOIN

```
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
```

JOIN

```
    order_details ON order_details.pizza_id = pizzas.pizza_id
```

GROUP BY pizza_types.category

ORDER BY quantity DESC;

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050



Determine the distribution of orders by hour of the day.

```
SELECT  
    HOUR(order_time) AS hour,  
    COUNT(order_id) AS order_count  
FROM  
    orders  
GROUP BY HOUR(order_time);
```

	hour	order_count
▶	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
	10	8
	9	1



find the category-wise distribution of pizzas

```
select category, count(name) as count  
from pizza_types  
group by category;
```

Result Grid

	category	count
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

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

SELECT

```
    round(AVG(quantity), 0) as avg_pizza_order_per_day
```

FROM

```
(SELECT
```

```
    orders.order_date, SUM(order_details.quantity) AS quantity
```

FROM

```
orders
```

```
JOIN order_details ON orders.order_id = order_details.order_id
```

```
GROUP BY orders.order_date) AS order_quantity;
```

	avg_pizza_order_per_day
▶	138



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

```
SELECT pizza_types.category,  
    round(SUM(order_details.quantity * pizzas.price) / (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)*100)  
    AS Contribution_in_revenue  
  
FROM  
    pizza_types  
    JOIN  
        pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
    JOIN  
        order_details ON order_details.pizza_id = pizzas.pizza_id  
GROUP BY pizza_types.category  
order by Contribution_in_revenue desc;
```

	category	Contribution_in_revenue
▶	Classic	27
	Supreme	25
	Veggie	24
	Chicken	24

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

```
select pizza_types.name,  
sum(order_details.quantity * pizzas.price)as revenue  
from pizza_types join pizzas  
on pizzas.pizza_type_id = pizza_types.pizza_type_id  
join order_details  
on order_details.pizza_id = pizzas.pizza_id  
group by pizza_types.name  
order by revenue desc limit 3;
```

Result Grid | Filter Rows: _____

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



Analyze the cumulative revenue generated over time.

```
select order_date,  
round(sum(revenue) over(order by order_date),2) as cum_revenue  
from  
(select orders.order_date,  
round(sum(order_details.quantity* pizzas.price),2) as revenue  
from order_details join pizzas  
on order_details.pizza_id= pizzas.pizza_id  
join orders  
on orders.order_id = order_details.order_id  
group by orders.order_date) as sales ;
```

	order_date	cum_revenue
▶	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	21526.4
	2015-01-10	23990.35
	2015-01-11	25862.65
	2015-01-12	27781.7
	2015-01-13	29831.3
	2015-01-14	32358.7
	2015-01-15	34343.5
	2015-01-16	36937.65
	2015-01-17	39001.75
	2015-01-18	40978.6
	2015-01-19	43365.75
	2015-01-20	45763.65
	2015-01-21	47804.2

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

```
select category, name , revenue from  
(select category,name, revenue,  
rank() over (partition by category order by revenue desc ) as rn  
from  
(select pizza_types.category, pizza_types.name,  
sum(order_details.quantity*pizzas.price) as revenue  
from pizza_types join pizzas  
on pizzas.pizza_type_id= pizza_types.pizza_type_id  
join order_details  
on order_details.pizza_id=pizzas.pizza_id  
group by pizza_types.category, pizza_types.name) as a) as b  
where rn <=3 ;
```

category	name	revenue
Chicken	The Thai Chicken Pizza	43434.25
Chicken	The Barbecue Chicken Pizza	42768
Chicken	The California Chicken Pizza	41409.5
Classic	The Classic Deluxe Pizza	38180.5
Classic	The Hawaiian Pizza	32273.25
Classic	The Pepperoni Pizza	30161.75
Supreme	The Spicy Italian Pizza	34831.25
Supreme	The Italian Supreme Pizza	33476.75
Supreme	The Sicilian Pizza	30940.5
Veggie	The Four Cheese Pizza	32265.7000000065
Veggie	The Mexicana Pizza	26780.75
Veggie	The Five Cheese Pizza	26066.5

Thanks!

Do you have any questions?

aarjan.tiwarii@gmail.com

visit my website

<https://aarjan-tiwari.github.io/>

<https://www.linkedin.com/in/aarjan-tiwari-4a1232183>

