

PIZZA SALES ANALYTICS



A cartoon illustration of a chef with dark hair, wearing a white chef's hat and a red apron over a green shirt. The chef is holding a pizza cutter and is about to slice a pizza that is covered in various toppings like pepperoni and cheese. The background is a solid reddish-orange color.

Hello everyone!

My name is Gavin Aranha,
In this Pizza Sales Analytics project, I've
leveraged SQL queries to address questions
related to pizza sales.
Through data analysis and insights, we'll
explore key trends and patterns in pizza
consumption.

Let's dive in!

DATA MODEL



RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```
select count(order_id) as total_orders from orders;
```

	TotalOrders
1	21350

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```
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
```

	TotalRevenue
1	817860.05

IDENTIFY THE HIGHEST-PRICED PIZZA

```
select pizza_types.name, pizzas.price  
from pizza_types join pizzas  
on pizza_types.pizza_type_id = pizzas.pizza_type_id  
order by pizzas.price desc limit 1;
```

	pizza_id	price
1	the_greek_xxL	35.95

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC;
```

	size	OrderCount
1	L	18956
2	M	15635
3	S	14403

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
1	The Classic Deluxe Pizza	2453
2	The Barbecue Chicken Pizza	2432
3	The Hawaiian Pizza	2422
4	The Pepperoni Pizza	2418
5	The Thai Chicken Pizza	2371

JOIN THE NECESSARY TABLES TO 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	total_quantity
1	Classic	14888
2	Supreme	11987
3	Veggie	11649
4	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
1	11	1231
2	12	2520
3	13	2455
4	14	1472
5	15	1468
6	16	1920
7	17	2336
8	18	2399
9	19	2009
10	20	1642
11	21	1198
12	22	663
13	23	28
14	24	8
15	25	1

JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS

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

	category	count
1	chicken	6
2	Classic	8
3	Supreme	9
4	Veggie	9

GROUP THE ORDERS BY DATE AND
CALCULATE THE AVERAGE NUMBER
OF PIZZAS ORDERED PER DAY.

```
select round(avg(quantity),0) from
(select orders.order_date, sum(order_details.quantity)
from orders join order_details
on orders.order_id = order_details.order_id
group by orders.order_date) as order_quantity ;
```

Average Pizzas	
1	138

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;
```

	name	revenue
1	The Thai Chicken Pizza	43434
2	The Barbecue Chicken Pizza	42768
3	The California Chicken Pizza	41409

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  
I  
FROM  
order_details  
JOIN  
pizzas ON pizzas.pizza_id = order_details.pizza_id) *100,2) as 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 revenue desc;
```

	category	revenue
1	Classic	26
2	Supreme	25
3	Chicken	23
4	Veggie	23

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
select order_date,  
sum(revenue) over(order by order_date) as cum_revenue  
from  
(select orders.order_date,  
sum(order_details.quantity * pizzas.price) 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
1	2015-01-01	2713
2	2015-01-02	5445
3	2015-01-03	8108
4	2015-01-04	9863
5	2015-01-05	11929
6	2015-01-06	14358

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

```
select 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 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, pizza_types.name) as a) as b
where rn <= 3;
```

	name	revenue
1	The Thai Chicken Pizza	43434
2	The Barbecue Chicken Pizza	42768
3	The California Chicken Pizza	41409
4	The Classic Deluxe Pizza	38180
5	The Mexican Pizza	32273
6	The Pepperoni Pizza	30161

