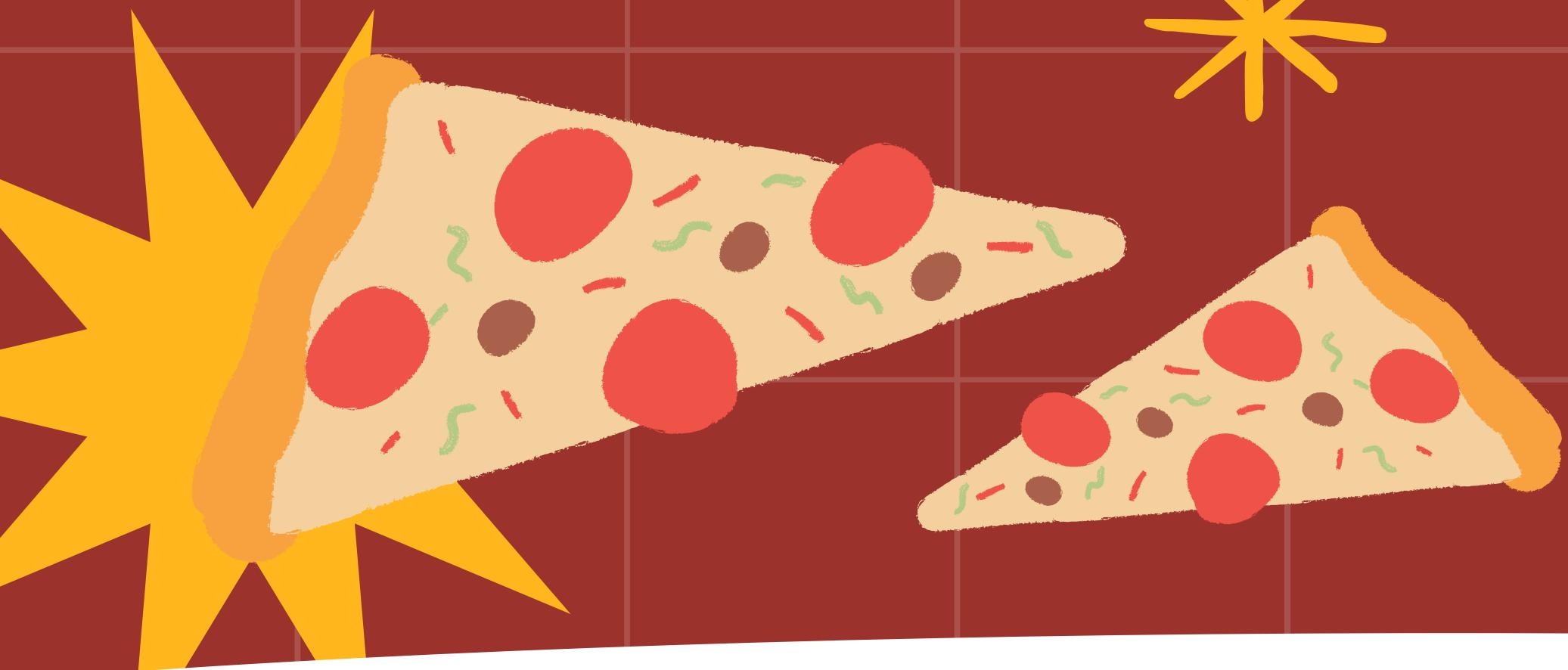


PIZZABITE

SQL PROJECT



INTRODUCTION

Hi, I'm Shubham Gupta. In this pizza project, I utilized SQL to analyze data and generate insights through various queries. This project showcases how SQL queries can be used to explore and understand pizza ordering patterns and trends.

Let's start our adventure in the world of pizza!

DATABASE

Brief:

This database analyzes pizza orders through four tables:

- **order_details**: Tracks pizza orders, quantities, and IDs.
- **orders**: Records order IDs, dates, and times.
- **pizza_types**: Details pizza types, categories, . and ingredients.
- **pizzas**: Provides pizza sizes and prices.



SQL QUERIES



- Retrieve the total number of orders placed.
- Calculate the total revenue generated from pizza sales.
- Identify the highest-priced pizza.
- Identify the most common pizza size ordered.
- List the top 5 most ordered pizza types along with their quantities.
- Join tables to find the total quantity of each pizza category ordered.
- Determine the distribution of orders by hour of the day.
- Find the category-wise distribution of pizzas.
- Calculate the average number of pizzas ordered per day.
- Determine the top 3 most ordered pizza types based on revenue.
- Calculate the percentage contribution of each pizza type to total revenue.
- Analyze the cumulative revenue generated over time.
- Determine the top 3 most ordered pizza types based on revenue for each pizza category.



**SHARPEN YOUR SQL
SKILLS TO SLICE THROUGH
PIZZA SALES DATA AND
REVEAL KEY INSIGHTS.**

QUERHY

Retrieve the total number of orders placed.

```
select  
    count(*) as Total_Orders  
from    orders;
```

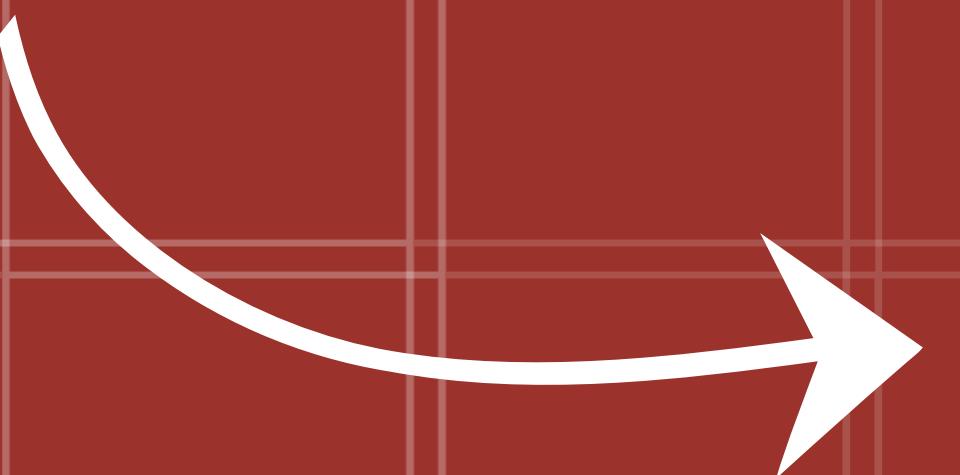
totalOrders
bigint

21350

QUERY 4

Calculate the total revenue generated from pizza sales.

```
select
    sum(order_details.quantity * pizzas.price::numeric) as total_revenue
from
order_details join pizzas
on
pizzas.pizza_id = order_details.pizza_id;
```

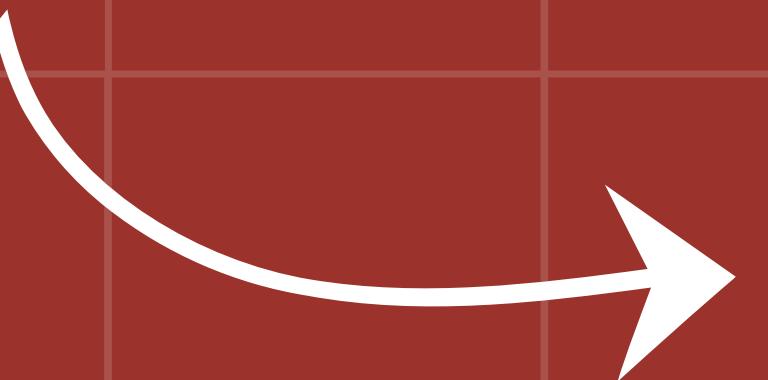


total_revenue	817860.05
numeric	

QUERY 3

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;
```



name	price
character varying (255)	double precision

The Greek Pizza

35.95

QUERY 4

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	order_count
1	L	18526
2	M	15385
3	S	14137
4	XL	544
5	XXL	28

QUERY 5

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

```
select pizza_types.name,  
sum(order_details.quantity) as quantity_order  
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_order desc limit 5;
```

	name	quantity_order
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

QUERY 6

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

```
select pizza_types.category,  
       sum(order_details.quantity) as quantity_order  
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_order desc ;
```

	category character varying (255)	quantity_order bigint
1	Classic	14888
2	Supreme	11987
3	Veggie	11649
4	Chicken	11050

QUERY 7

Determine the distribution of orders by hour of the day.

select

```
extract(hour from time) as hour,  
count(order_id) as orders  
from orders  
group by hour;
```

	hour numeric	orders bigint
1	11	1231
2	23	28
3	18	2399
4	19	2009
5	15	1468
6	9	1
7	21	1198
8	17	2336
9	20	1642
10	13	2455
11	10	8
12	16	1920
13	22	663
14	12	2520
15	14	1472

QUERY 8

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

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

	category character varying (255)	count bigint
1	Veggie	9
2	Chicken	6
3	Supreme	9
4	Classic	8

QUERY 9

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

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

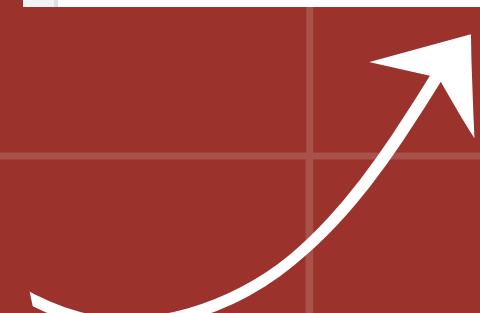
	avg_quantity
1	138

QUERHYO

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	43434.25
2	The Barbecue Chicken Pizza	42768
3	The California Chicken Pizza	41409.5

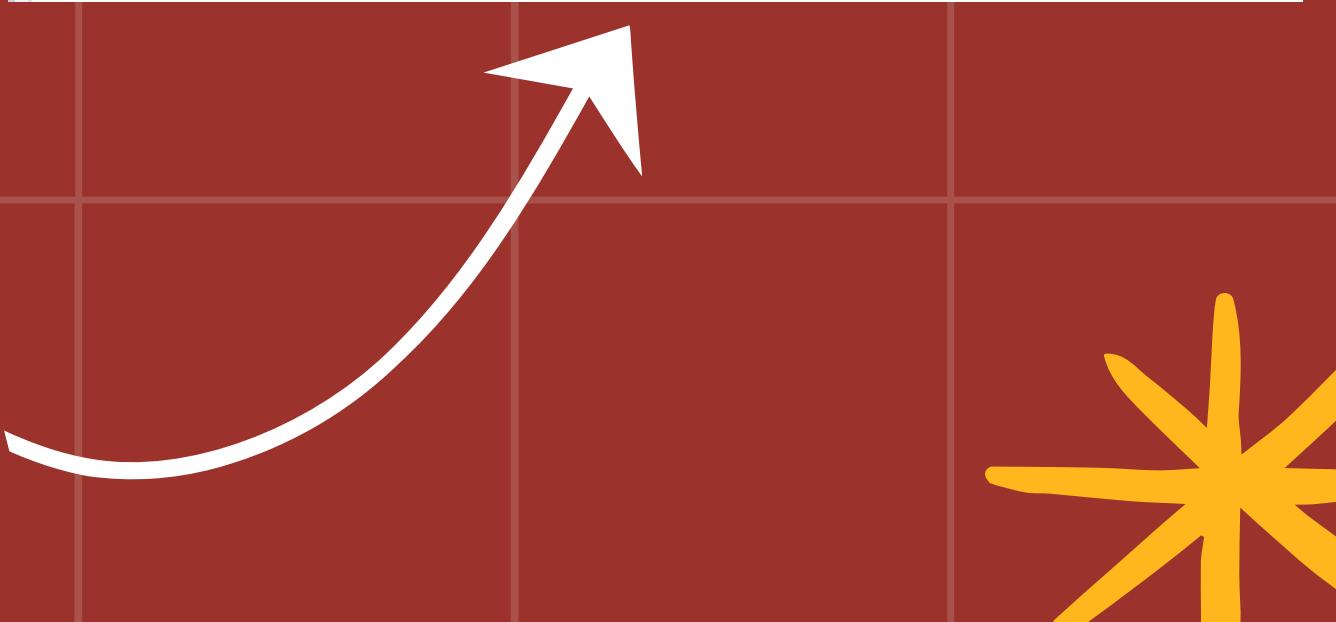


QUERY 11

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

```
select
    pizza_types.category,
    round(sum(order_details.quantity * pizzas.price :: numeric) / (select
        sum(order_details.quantity * pizzas.price::numeric) as total_sales
    from
        order_details join pizzas
    on
        pizzas.pizza_id = order_details.pizza_id)* 100,2) || '%' as percentage_contribution
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
order by percentage_contribution desc;
```

	category character varying (255)	percentage_contribution text
1	Classic	26.91%
2	Supreme	25.46%
3	Chicken	23.96%
4	Veggie	23.68%



QUERY 12

Analyze the cumulative revenue generated over time.

```
select date,  
sum(revenue) over(order by date) as cum_revenue  
from  
(select orders.date,  
       sum(order_details.quantity*pizzas.price::numeric) 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.date) as sales;
```

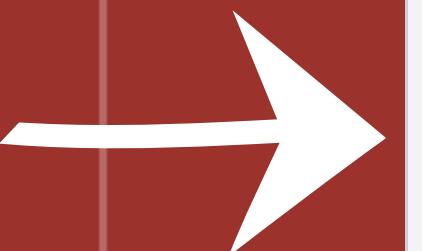


	date	date	cum_revenue
1	2015-01-01		2713.85
2	2015-01-02		5445.75
3	2015-01-03		8108.15
4	2015-01-04		9863.60
5	2015-01-05		11929.55
6	2015-01-06		14358.50
7	2015-01-07		16560.70
8	2015-01-08		19399.05

QUERY 13

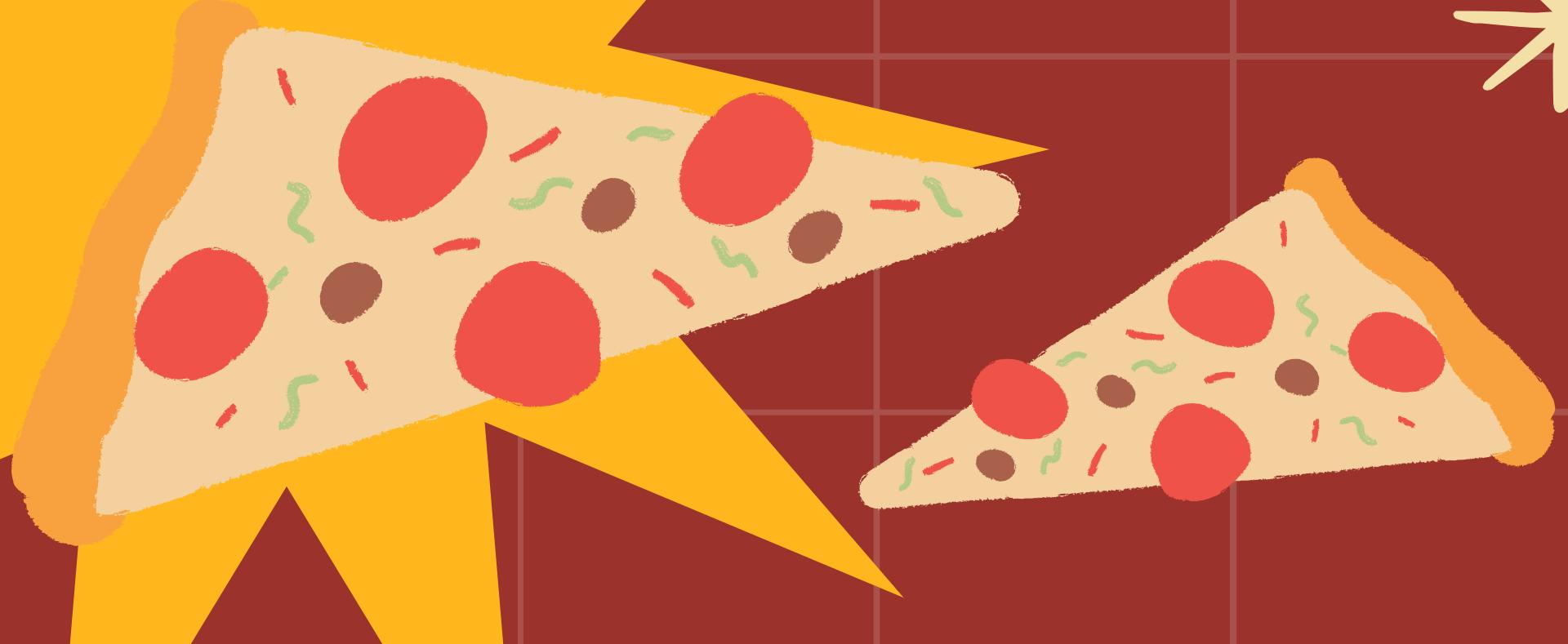
Determine the top 3 most ordered pizza types based on revenue for each pizza category.

```
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::numeric) 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 character varying (255)	revenue numeric
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768.00
3	The California Chicken Pizza	41409.50
4	The Classic Deluxe Pizza	38180.5
5	The Hawaiian Pizza	32273.25
6	The Pepperoni Pizza	30161.75
7	The Spicy Italian Pizza	34831.25
8	The Italian Supreme Pizza	33476.75
9	The Sicilian Pizza	30940.50
10	The Four Cheese Pizza	32265.70
11	The Mexicana Pizza	26780.75
12	The Five Cheese Pizza	26066.5

KEY FINDINGS



- **Total Orders:** 21,350
- **Revenue:** ₹817,860.05
- **Highest-Priced Pizza:** The Greek Pizza,
Price: ₹35.95
- **Most Common Size:** Large.
- **Top Revenue Pizzas:** Thai Chicken,
Barbecue Chicken,
California Chicken

TOP PREFERRED PIZZA



Classic Deluxe



Pepperoni Pizza



BBQ Chicken Pizza



Hawaiian Pizza



THANK YOU

Thank you for your time and attention. Please like and comment to share your feedback.