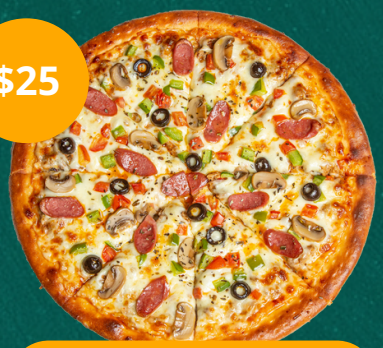
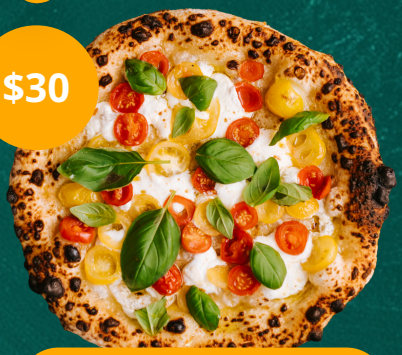
PIZZA Menu



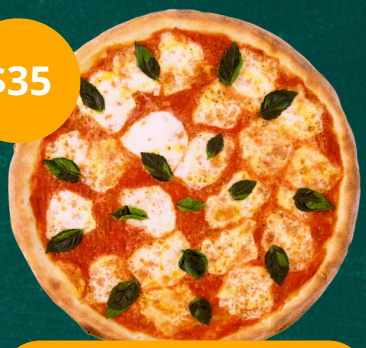
$25

Classic



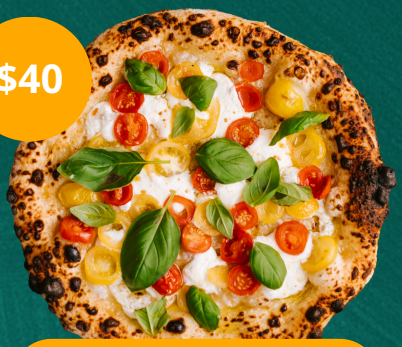
$30

Chicken



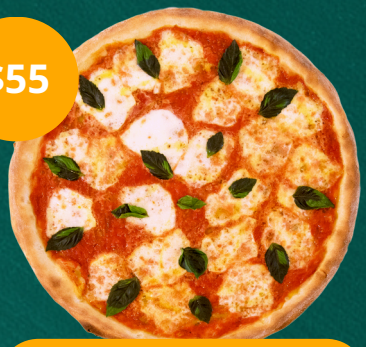
$35

Supreme



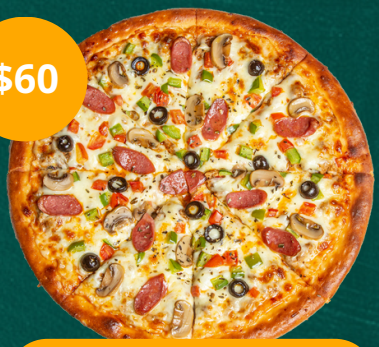
$40

Veggie



$55

Paneer



$60

Cheese

Pizza Store Database Project

Made By

Adarsh Raghav

GitHub Profile Link

https://github.com/adarshraghav26

(Project is upload to GitHub Repositories)

Q1. Retrieve the total number of order placed.

Solution :-

```
SELECT
    COUNT(order_details_id) AS total_orders
FROM
    order_details;
```

Output

	total_orders
▶	48620

Q2. Calculate total revenue generated from pizza sales.

Solution :-

```
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS Total_Sales
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id;
```

Output

	Total_Sales
▶	817860.05

Q3. Price of which pizza is highest.

Solution :-

```
SELECT
    pizza_types.name AS Name, pizzas.price as Price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

Output

	Name	Price
▶	The Greek Pizza	35.95

Q4. Identify the most common orderred pizza.

Solution :-

```
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;
```

Output

	size	order_count
▶	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28

Q5. List the top 5 most common ordered pizza types along with their quantities.

Solution :-

```
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;
```

Output

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

Q6. Join the necessary tables to find the quantity of each pizza category ordered.

Solution :-

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

Output

	Category	Quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

Q7. Determine the distribution of orders by hour of the day.

Solution :-

```
SELECT
    HOUR(order_time) AS Hour, COUNT(order_id) AS Orders
FROM
    orders
GROUP BY Hour;
```

Output

Hour	Orders
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

Q8. Find the category-wise distribution of pizzas.

Solution :-

```
SELECT
    COUNT(name) as Category_Count, category as Category
FROM
    pizza_types
GROUP BY Category;
```

Output

	Category_Count	Category
▶	6	Chicken
	8	Classic
	9	Supreme
	9	Veggie

Q9. Group the orders by date and calculate the average number of pizza ordered by day.

Solution :-

```sql
SELECT
    ROUND(AVG(quantity), 0) as Avg_Pizza_Ordered_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;
```

Output

	Avg_Pizza_Ordered_Per_Day
▶	138

Q10. Determine the top 3 most ordered pizza types based on revenue.

Solution :-

```
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;
```

Output

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

Q11. Calculate the percentage contribution of each pizza type to total revenue.

Solution :-

```
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,
            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;
```

Output

	category	Revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

Q12. Analyze the cummulative revenue generated over time.

Solution :-

```
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;
```

Output

order_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	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.300000000003
2015-01-14	32358.700000000004
2015-01-15	34343.50000000001
2015-01-16	36937.65000000001

Q13. Determine the top 3 most ordered pizza types based on revenue for each pizza category.

Solution :-

```
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;
```

Output

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