

PRODUCT PERFORMANCE ANALYSIS BUSINESS INTELLIGENCE REPORT

A Sales analysis by Ayush Singh

PROJECT OBJECTIVE

TO ANALYZE SALES DATA OF
A PIZZA COMPANY USING SQL
AND GENERATE ACTIONABLE
BUSINESS INSIGHTS TO
SUPPORT DATA-DRIVEN
DECISIONS.

KEY BUSINESS QUESTIONS

- HOW MANY TOTAL ORDERS WERE PLACED?
- WHAT IS THE HIGHEST-PRICED PIZZA?
- WHAT IS THE TOTAL REVENUE GENERATED?
- WHICH PIZZA SIZE IS MOST POPULAR?
- WHAT ARE THE TOP 5 MOST ORDERED PIZZA TYPES?

TOTAL NUMBER OF ORDERS PLACED

```
SELECT  
count(order_id) as total_orders  
from  
orders
```

Result Grid	
	total_orders
▶	21350

TOTAL ORDERS:21.350





HIGHEST-PRICED PIZZA

```
SELECT
    pt.name, p.price
FROM
    pizzas p
    JOIN
    pizza_types pt ON p.pizza_type_id = pt.pizza_type_id
ORDER BY price DESC limit 1
```

Result Grid | Filter Row

	name	price
▶	The Greek Pizza	35.95

POPULAR PIZZA SIZES

```
SELECT  
    p.size, COUNT(size) AS pizza_order  
FROM  
    PIZZAS p  
    JOIN  
        order_details o ON o.pizza_id = p.pizza_id  
GROUP BY size  
ORDER BY pizza_order DESC  
LIMIT 1
```

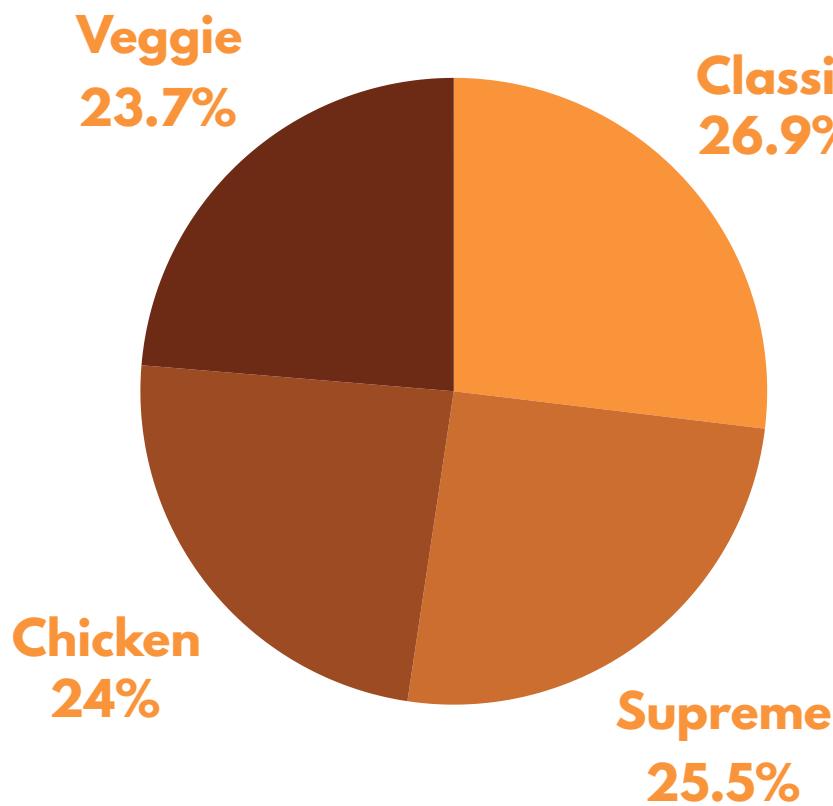
	Result Grid	Filter
	size	pizza_order
▶	L	18526



TOP 5 MOST ORDERED PIZZA ALONG WITH THEIR QUANTITIES

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

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



Result Grid | Filter Rows:

	category	revenue_percent
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

```

SELECT
    category,
    round(SUM(price * quantity) / (SELECT
                                        SUM(price * quantity) AS total_revenue
                                    )
        FROM
            order_details od
        JOIN
            pizzas p ON p.pizza_id = od.pizza_id) * 100,2) AS revenue_percent
FROM
    pizzas p
JOIN
    pizza_types pt ON pt.pizza_type_id = p.pizza_type_id
JOIN
    order_details od ON od.pizza_id = p.pizza_id
GROUP BY category
ORDER BY revenue_percent DESC
  
```

THE MIX OF PRODUCT OFFERINGS IS WELL-BALANCED ACROSS CATEGORIES.

REVENUE CONTRIBUTION BY PIZZA TYPES

QUANTITY ORDERED BY CATEGORY

SELECT

pt.category, SUM(od.quantity) AS total_quantity

FROM

pizza_types pt

JOIN

pizzas p ON pt.pizza_type_id = p.pizza_type_id

JOIN

order_details od ON od.pizza_id = p.pizza_id

GROUP BY pt.category

ORDER BY total_quantity DESC

	category	total_quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

DAILY SALES TREND

```
SELECT  
    ROUND(AVG(SUM_quantity), 0) AS AVG_PIZZAS_ORDERED  
FROM  
    (SELECT  
        DATE, SUM(quantity) AS SUM_QUANTITY  
    FROM  
        ORDERS o  
    JOIN order_details OD ON OD.order_id = o.order_id  
    GROUP BY date) AS ORDER_QUANTITY
```

	AVG_PIZZAS_ORDERED
▶	138

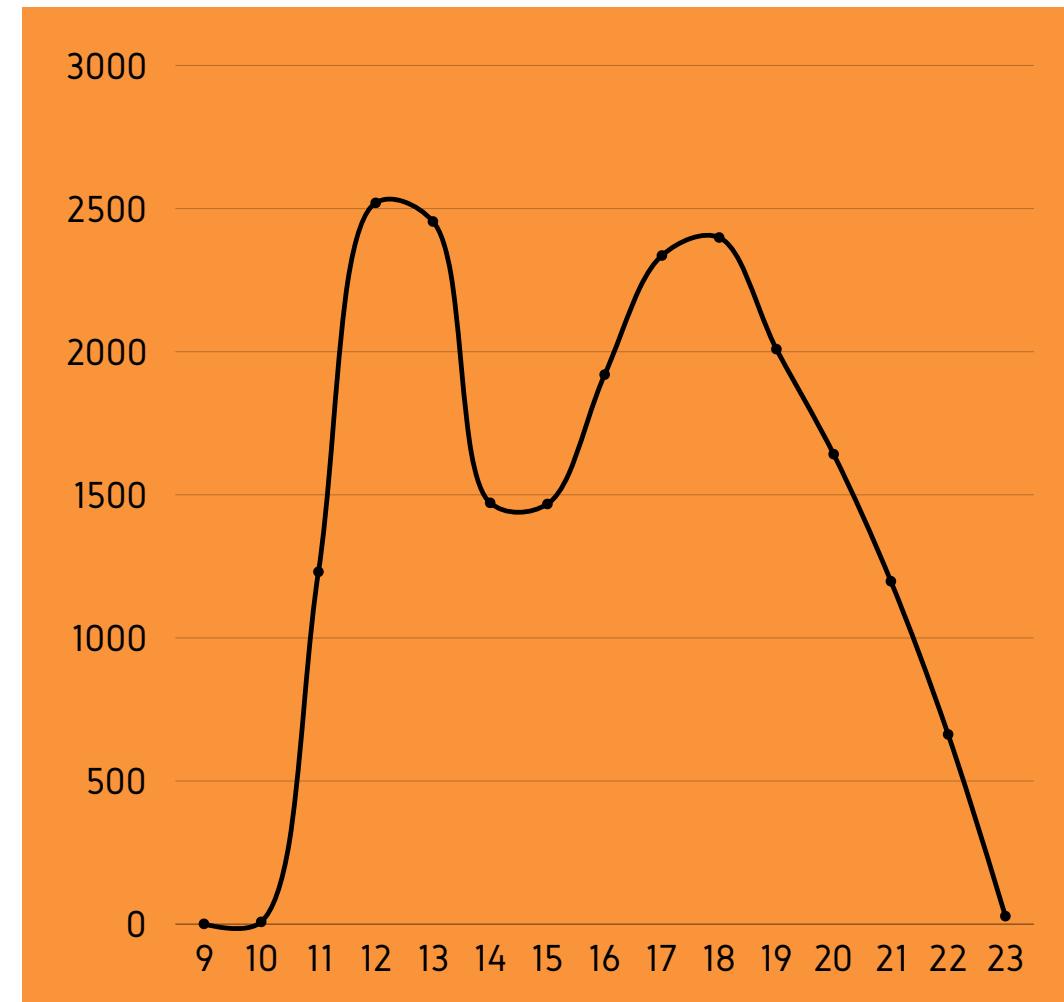
Order distribution by Hour



Valuable for optimizing staffing and inventory.



```
SELECT  
    HOUR(time) AS hour, COUNT(order_id) AS orders_placed  
FROM  
    orders  
GROUP BY HOUR(time)
```



	hour	orders_placed
▶	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

TOP 3 MOST ORDERED PIZZA BASED ON REVENUE

These products drive the highest revenue.

```
SELECT
    pt.name, SUM((PRICE * QUANTITY)) AS REVENUE
FROM
    PIZZAS P
        JOIN
    order_details OD ON OD.pizza_id = P.PIZZA_ID
        JOIN
    pizza_types pt ON pt.pizza_type_id = p.pizza_type_id
GROUP BY pt.name
ORDER BY revenue DESC
LIMIT 3
```

Result Grid		
	name	REVENUE
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5

```
select  
    date,  
    revenue,  
    round(sum(revenue) over(order by date),2) as cumulative_revenue  
from (select  
    date,round(sum(quantity*price),2) as revenue  
    from pizzas p  
    join order_details od  
    on od.pizza_id=p.pizza_id  
    join pizza_types pt  
    on pt.pizza_type_id=p.pizza_type_id  
    join orders o  
    on od.order_id=o.order_id  
    group by date  
) as x
```

Result Grid			
	date	revenue	cumulative_revenue
1	2015-12-05	2627.95	765199.2
2	2015-12-06	2350.25	767549.45
3	2015-12-07	2414.8	769964.25
4	2015-12-08	1856.25	771820.5
5	2015-12-09	2571.55	774392.05
6	2015-12-10	1985.6	776377.65
7	2015-12-11	2634	779011.65
8	2015-12-12	1960.15	780971.8
9	2015-12-13	2245.15	783216.95
10	2015-12-14	2172.6	785389.55
11	2015-12-15	2387.45	787777
12	2015-12-16	2234.8	790011.8
13	2015-12-17	1880.75	791892.55
14	2015-12-18	2886.3	794778.85
15	2015-12-19	2304.2	797083.05
16	2015-12-20	2104.9	799187.95
17	2015-12-21	2100.7	801288.65
18	2015-12-22	1882.95	803171.6
19	2015-12-23	2244.3	805415.9
20	2015-12-24	2137.85	807553.75

CUMULATIVE REVENUE GENERATED OVER TIME

```
select
name,revenue
from
(select
name,category,revenue,rank() over(partition by category order by revenue desc) as rank_based
from (select
name,category,sum(price*quantity) as revenue
from pizzas p
join pizza_types pt
on pt.pizza_type_id=p.pizza_type_id
join order_details od
on od.pizza_id=p.pizza_id
group by category,name
order by revenue) as s) as r
where rank based<=3
```

	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

TOP REVENUE GENERATING PIZZAS BY CATEGORY

TOTAL REVENUE GENERATED FROM PIZZA SALES

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

Result Grid	
	total_revenue
▶	817860.05



CONCLUSION & RECOMMENDATIONS

- FOCUS PROMOTIONS ON TOP-SELLING AND HIGH-REVENUE PIZZAS.
- ALIGN INVENTORY AND STAFFING WITH PEAK ORDER TIMES.
- CONSIDER BUNDLING OR UPSELLING DURING OFF-PEAK HOURS.
- USE REVENUE TRENDS TO FORECAST DEMAND AND OPTIMIZE LOGISTICS.