

Where Every Slice is a Taste of Perfection

# PRESENTING PIZZA SALES INSIGHTS

VIEW  
INSIGHTS  
NOW



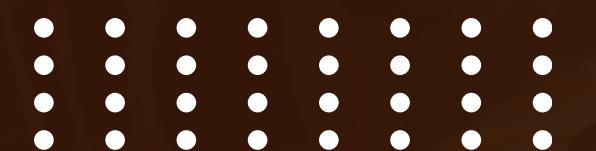
Start Slide

# ABOUT THE PROJECT



## Our Passion for Pizza

Hi this is Abhishek Shakya and I have completed and showcased a project that focuses on analyzing pizza sales data using SQL to gain insights into sales performance, customer preferences, and business trends. The dataset includes order details, pizza types, pizzas and orders. SQL queries are used to extract key insights such as total orders, revenue, Top-selling pizzas, average orders and customer purchasing behavior. The results will help optimize menu offerings, improve sales strategies, and enhance overall business efficiency.



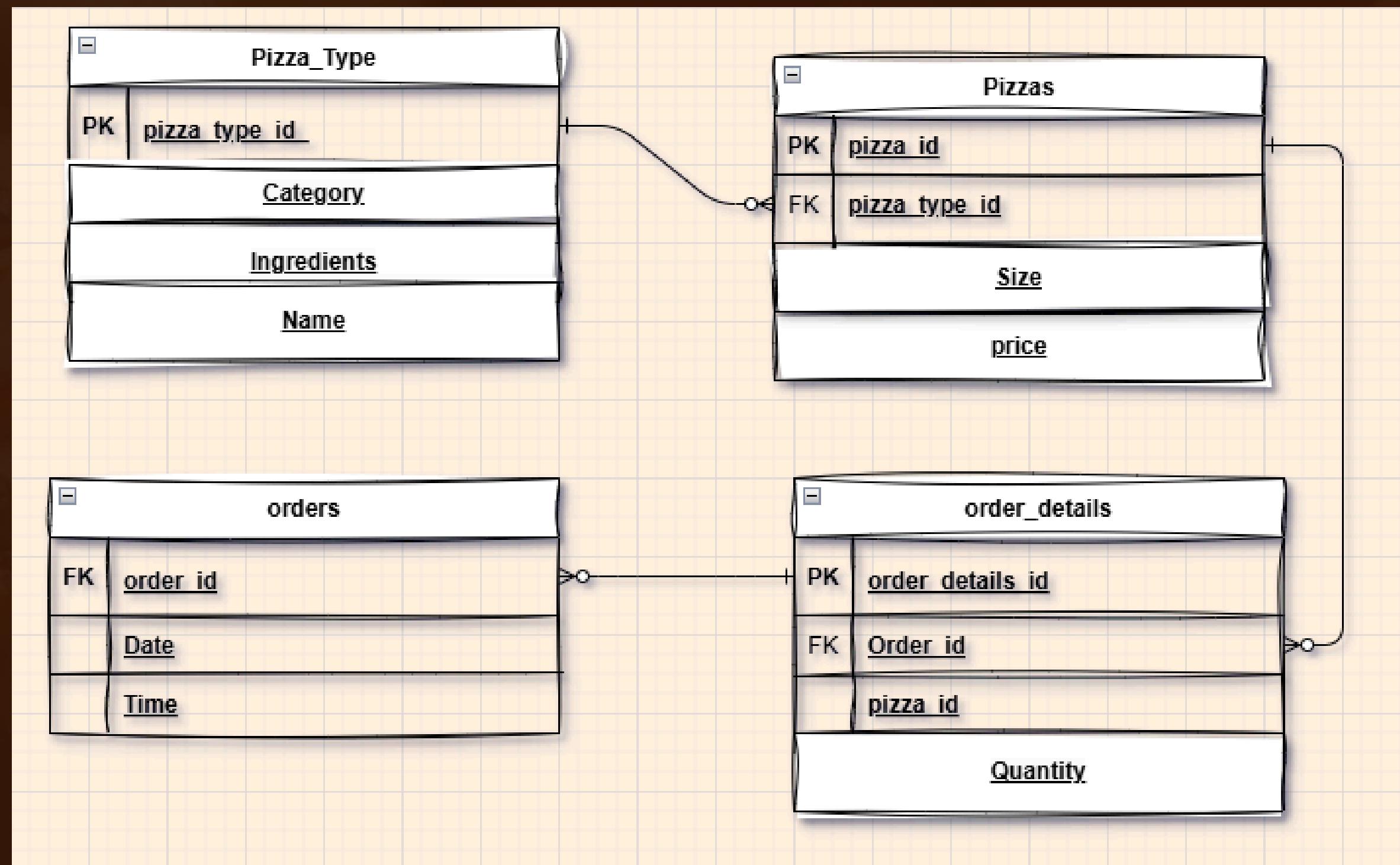
# SCHEMA

Tables used

- PIZZA\_TYPE
- PIZZAS
- ORDER\_DETAILS
- ORDERS



# ENTITY-RELATIONSHIP DIAGRAM



# Q1. RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.



```
17    -- Q1. Retrieve the total number of orders placed.  
18 •   select count(order_id)  
19     from pizza_sales.orders;
```

Result Grid	
	count(order_id)
▶	21350

# Q2. CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```
21    -- Q2 Calculate the total revenue generated from pizza sales.  
22 •   select sum(p1.price*od1.quantity) as total_Revenue  
23     from pizza_sales.order_details as od1  
24     inner join pizza_sales.pizzas as p1  
25     on od1.pizza_id=p1.pizza_id;  
26  
27
```

Result Grid	
	total_Revenue
▶	817860.049999993

## Q3. IDENTIFY THE HIGHEST-PRICED PIZZA.

```
27    -- Q3 Identify the highest-priced pizza.  
28 •  select pt1.pizza_type_id, pt1.name, p1.price  
29    from pizza_sales.pizzas as p1  
30    inner join pizza_sales.pizza_types as pt1  
31    on p1.pizza_type_id=pt1.pizza_type_id  
32    order by price desc  
33    limit 1;
```

The screenshot shows a MySQL command-line interface with the following output:

	pizza_type_id	name	price
▶	the_greek	The Greek Pizza	35.95

## Q4. IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
35    -- Q4 Identify the most common pizza size ordered.  
36  
37 •  select p1.size, count(od1.quantity) as total_quantity  
38    from pizza_sales.order_details as od1  
39    inner join pizza_sales.pizzas as p1  
40    on od1.pizza_id=p1.pizza_id  
41    group by p1.size  
42    order by total_quantity desc  
43    limit 1;
```

The screenshot shows a MySQL command-line interface with the following output:

	size	total_quantity
▶	L	18526

## Q5. LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

```
43    -- Q5. List the top 5 most ordered pizza types along with their quantities.  
44 •  select p1.pizza_type_id, count(od1.quantity) as total_quantity  
45    from pizza_sales.order_details as od1  
46    inner join pizza_sales.pizzas as p1  
47    on od1.pizza_id=p1.pizza_id  
48    group by p1.pizza_type_id  
49    order by total_quantity desc  
50    limit 5;  
51
```

	pizza_type_id	total_quantity
▶	classic_dlx	2416
	bbq_ckn	2372
	hawaiian	2370
	pepperoni	2369
	thai_ckn	2315

```
54    -- Q6 Determine the distribution of orders by hour of the day.  
55 •  with h1 as (  
56    SELECT *, hour(o1.time) as hours  
57    FROM pizza_sales.orders as o1  
58  )  
59  
60    select h1.hours, count(h1.order_id) as hourly_order_count  
61    from h1  
62    group by h1.hours;  
63
```

	hours	hourly_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

## Q6. DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

# Q7. LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

```
59      -- Q7 calculate the average number of pizzas ordered per day.  
60 •  SELECT o1.date, avg(o1.order_id) as Average_orders_per_day  
61   FROM pizza_sales.orders as o1  
62   group by o1.date  
63
```

Result Grid | Filter Rows:  Export: Wrap Cell Content:

	date	Average_orders_per_day
▶	1/1/2015	35.0000
	1/2/2015	103.0000
	1/3/2015	169.5000
	1/4/2015	228.5000
	1/5/2015	281.5000
	1/6/2015	340.5000
	1/7/2015	401.5000
	1/8/2015	466.5000
	1/9/2015	533.5000
	1/10/2015	597.0000
	1/11/2015	655.5000
	1/12/2015	709.0000
	1/13/2015	760.5000
	1/14/2015	815.5000
	1/15/2015	877.5000
	1/16/2015	942.0000
	1/17/2015	1003.0000
	1/18/2015	1056.0000
	1/19/2015	1109.5000
	1/20/2015	1169.5000
	1/21/2015	1227.0000

Result 45 ×

# Q8. DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

```
66      -- Q8 Determine the top 3 most ordered pizza types based on revenue.  
67  
68 •  select p1.pizza_type_id, pt1.name, sum(p1.price*od1.quantity) as revenue_from_each_type  
69   from pizza_sales.order_details as od1  
70   left join pizza_sales.pizzas as p1  
71   on od1.pizza_id=p1.pizza_id  
72   left join pizza_sales.pizza_types as pt1  
73   on p1.pizza_type_id=pt1.pizza_type_id  
74   group by p1.pizza_type_id, pt1.name  
75   order by revenue_from_each_type desc  
76   limit 3;  
77
```

Result Grid | Filter Rows:  Export: Wrap Cell Content: Fetch rows:

	pizza_type_id	name	revenue_from_each_type
▶	thai_ckn	The Thai Chicken Pizza	43434.25
	bbq_ckn	The Barbecue Chicken Pizza	42768
	cali_ckn	The California Chicken Pizza	41409.5

# Q9. CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

```
79    -- Q9. Calculate the percentage contribution of each pizza type to total revenue.
80 • with TR as (
81     select sum(p1.price*od1.quantity) as total_Revenue
82     from pizza_sales.order_details as od1
83     inner join pizza_sales.pizzas as p1
84     on od1.pizza_id=p1.pizza_id
85   ),
86 • ER as (
87     select p1.pizza_type_id,
88            pt1.name,
89            sum(p1.price*od1.quantity) as revenue_from_each_type
90     from pizza_sales.order_details as od1
91     left join pizza_sales.pizzas as p1
92       on od1.pizza_id=p1.pizza_id
93     left join pizza_sales.pizza_types as pt1
94       on p1.pizza_type_id=pt1.pizza_type_id
95     group by p1.pizza_type_id, pt1.name
96     order by revenue_from_each_type desc
97   )
98
99 select
100    ER.pizza_type_id as pizza_id,
101    ER.name as pizza_name,
102    ER.revenue_from_each_type ,
103    concat(round((ER.revenue_from_each_type*100)/ TR.total_Revenue,1),('%')) as percentage_contribution
104  from ER, TR
105  order by ER.revenue_from_each_type desc;
106
```

Result Grid				
	pizza_id	pizza_name	revenue_from_each_type	percentage_contribution
▶	thai_ckn	The Thai Chicken Pizza	43434.25	5.3%
	bbq_ckn	The Barbecue Chicken Pizza	42768	5.2%
	cali_ckn	The California Chicken Pizza	41409.5	5.1%
	classic_dlx	The Classic Deluxe Pizza	38180.5	4.7%
	spicy_ital	The Spicy Italian Pizza	34831.25	4.3%
	southw_ckn	The Southwest Chicken Pizza	34705.75	4.2%
	ital_supr	The Italian Supreme Pizza	33476.75	4.1%
	hawaiian	The Hawaiian Pizza	32273.25	3.9%
	four_ckn	The Four Cheese Pizza	22755.700000000005	2.8%

```
107    -- Q10. Analyze the cumulative revenue generated over time.
108
109 • with daily_revenue as (
110     select
111            o1.date,
112            round(sum(quantity*price),1) as daily_total_revenue
113     from pizza_sales.order_details as od1
114     inner join pizza_sales.orders as o1
115       on od1.order_id=o1.order_id
116     inner join pizza_sales.pizzas as p1
117       on od1.pizza_id=p1.pizza_id
118     group by o1.date
119   ),
120 • lead_revenue as (
121     select daily_revenue.*,
122     lead(daily_total_revenue) over (order by daily_revenue.date asc) as next_daily_revenue
123     from daily_revenue
124   )
125
126   select lead_revenue.date, lead_revenue.daily_total_revenue,
127         sum(daily_total_revenue) over (order by date asc) as cumulitative_revenue
128     from lead_revenue
```

Result Grid			
	date	daily_total_revenue	cumulative_revenue
▶	1/1/2015	2713.9	2713.9
	1/10/2015	2464	5177.9
	1/11/2015	1872.3	7050.2
	1/12/2015	1919	8969.2
	1/13/2015	2049.6	11018.800000000001
	1/14/2015	2527.4	13546.2
	1/15/2015	1984.8	15531
	1/16/2015	2594.2	18125.2
	1/17/2015	2064.1	20189.3
	1/18/2015	1976.8	22166.1
	1/19/2015	2387.1	24553.199999999997
	1/20/2015	2731.9	27285.1
	1/21/2015	2397.9	29683
	1/22/2015	2040.6	31723.6

# Q10. ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

# Q11. DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

```
130    -- Q11. Determine the top 3 most ordered pizza types based on revenue for each pizza category.  
131  
132 • select p1.pizza_type_id, p1.pizza_id, pt1.name, sum(p1.price*od1.quantity) as revenue_from_each_type  
133   from pizza_sales.order_details as od1  
134   left join pizza_sales.pizzas as p1  
135   on od1.pizza_id=p1.pizza_id  
136   left join pizza_sales.pizza_types as pt1  
137   on p1.pizza_type_id=pt1.pizza_type_id  
138   group by p1.pizza_type_id, p1.pizza_id, pt1.name  
139   order by revenue_from_each_type desc  
140   limit 3;  
141
```

Result Grid				
	pizza_type_id	pizza_id	name	revenue_from_each_type
▶	thai_ckn	thai_ckn_1	The Thai Chicken Pizza	29257.5
	five_cheese	five_cheese_1	The Five Cheese Pizza	26066.5
	four_cheese	four_cheese_1	The Four Cheese Pizza	23622.200000000554

Pizza Restro Sales Insights

THANK YOU  
FOR ATTENTION

Find me on Linkedin  
Name: Abhishek Shakya