

# PIZZA HUT

## S A L E S   P R O J E C T

*on MySQL*



# ABOUT ME...



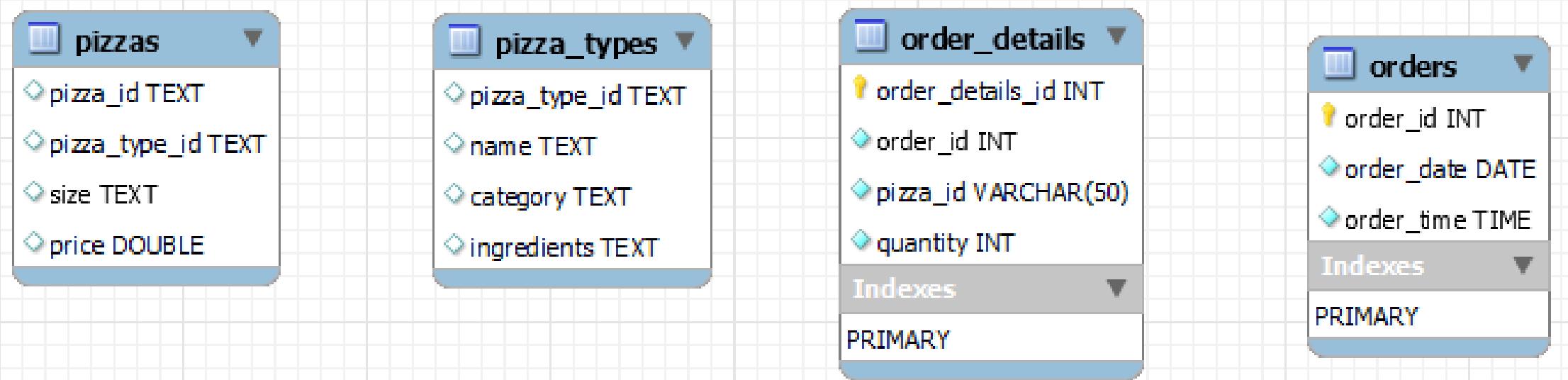
***Kuldeep Singh***, a dedicated aspiring data analyst, adept in ***SQL, Python, NumPy, Pandas, and PowerBI***. With a passion for unraveling insights from data, he leverages his analytical prowess to derive meaningful conclusions. Eager to contribute innovative solutions and drive data-driven decisions in a dynamic professional landscape.



# OBJECTIVE OF PROJECT

The project's objective on pizza sales using MySQL is to design and implement a relational database system that efficiently manages and analyzes data related to pizza sales. By utilizing SQL queries and data analysis techniques, the project aims to derive insights into sales trends, customer preferences, popular items, and operational efficiency, thereby facilitating informed decision-making for business improvement and growth within the pizza industry.

# ERR DIAGRAM



# BASIC LEVEL



- 1. Retrieve the total number of orders placed.*
- 2. Calculate the total revenue generated from pizza sales.*
- 3. Identify the highest-priced pizza.*
- 4. Identify the most common pizza size ordered.*
- 5. List the top 5 most ordered pizza types along with their quantities.*

*Retrieve the total number of orders placed.*

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

	total_orders
▶	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;
```

	total_sales
▶	817860.05



## *Identify the highest-priced pizza.*

-- 1st method

```
select pizza_id, price from pizzas  
order by price desc limit 1;
```

	pizza_id	price
▶	the_greek_xxl	35.95

-- 2nd method

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

	name	price
▶	The Greek Pizza	35.95



-- 3rd method

```
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
▶	The Greek Pizza	35.95



## *Identify the most common pizza size ordered.*

```
-- 1st method  
  
select count(pizzas.size) a, size  
from pizzas join order_details  
on  
pizzas.pizza_id = order_details.pizza_id  
group by size  
order by a desc limit 1;
```

```
-- 2nd method
```

```
select pizzas.size,  
count(order_details.order_details_id) as a  
from pizzas join order_details  
on pizzas.pizza_id = order_details.pizza_id  
group by pizzas.size  
ORDER BY a desc limit 1;
```

	a	size
▶	18526	L



	size	a
▶	L	18526



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

```
SELECT
    pizza_types.name,
    SUM(order_details.quantity) AS quantity_ordered
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_ordered DESC
LIMIT 5;
```

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



# INTERMEDIATE LEVEL

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

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

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

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

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



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

**SELECT**

```
    pizza_types.category,  
    SUM(order_details.quantity) AS total_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;

	category	total_quantity
▶	Classic	14888
	Veggie	11649
	Supreme	11987
	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
	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

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

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

	category	count(name)
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9



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

**SELECT**

    ROUND(AVG(quantity), 0) **as** average\_pizza\_ordered

**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;

	<b>average_pizza_ordered</b>
▶	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 pizza_types.pizza_type_id = pizzas.pizza_type_id
```

**JOIN**

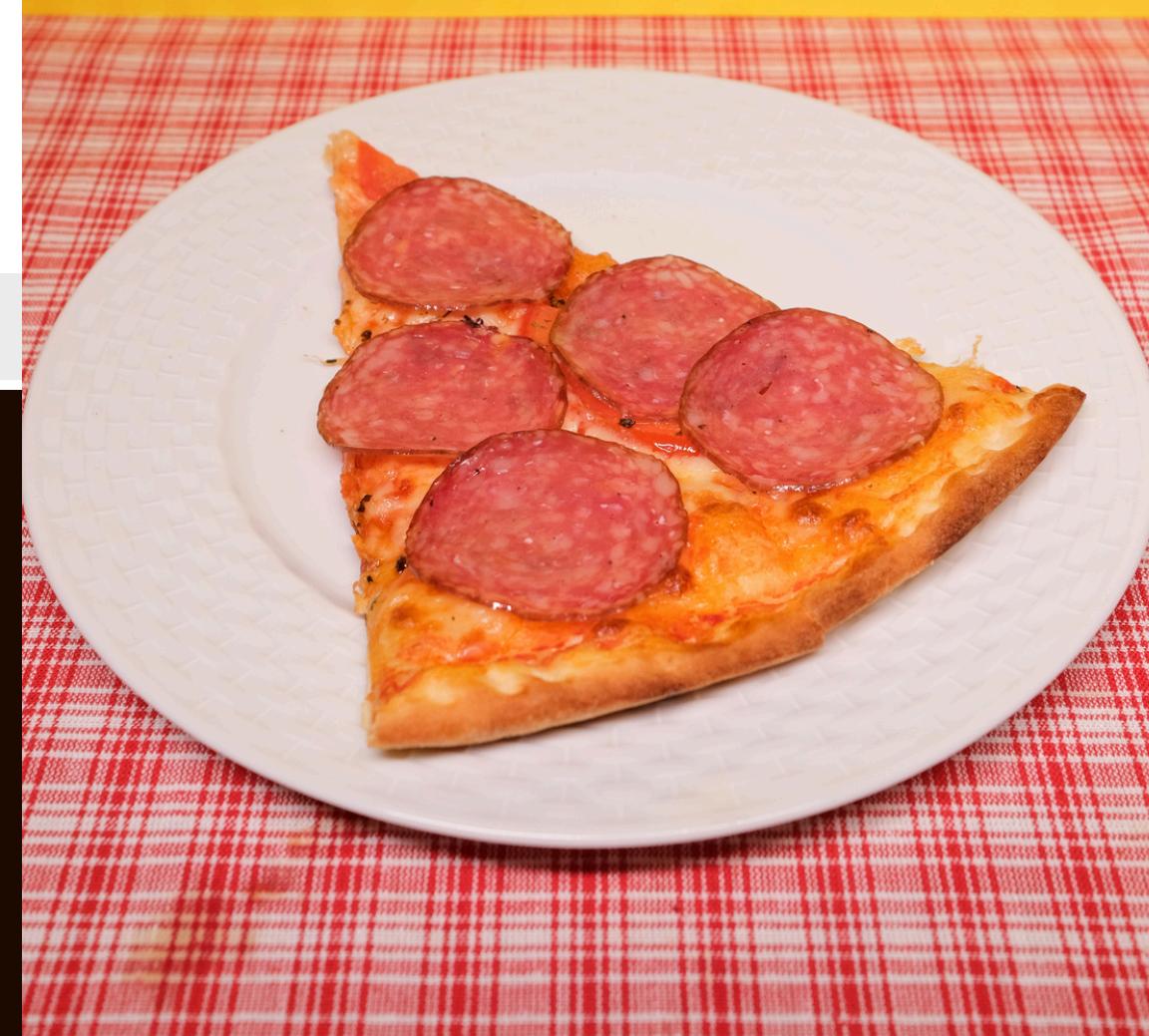
```
order_details ON order_details.pizza_id = pizzas.pizza_id
```

**GROUP BY** pizza\_types.name

**ORDER BY** revenue DESC

**LIMIT** 3;

	name	revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5





# ADVANCE LEVEL

1. *Calculate the percentage contribution of each pizza type to total revenue.*
  2. *Analyze the cumulative revenue generated over time.*
  3. *Determine the top 3 most ordered pizza types based on revenue for each pizza category.*
- 

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

```
SELECT
    pizza_types.category,
    concat(ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
        SUM(order_details.quantity * pizzas.price)
    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;
-- 2nd method
SELECT
    SUM(order_details.quantity * pizzas.price) as total_revenue
    FROM
        order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id
;
SELECT
    pizza_types.category,
    concat(ROUND(SUM(order_details.quantity * pizzas.price) / 817860.049999993* 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;
```

	category	revenue
▶	Classic	26.91%
	Veggie	23.68%
	Supreme	25.46%
	Chicken	23.96%

## Analyze the cumulative revenue generated over time.

```
select order_date, round(sum(revenue) over(order by order_date),2) as cumulative_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	cumulative_revenue
2015-01-01	2713.85
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	2015-01-10
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.3
2015-01-14	32358.7
2015-01-15	34343.5
2015-01-16	36937.65
2015-01-17	39001.75
2015-01-18	40978.6

order_date	cumulative_revenue
2015-01-19	43365.75
2015-01-20	45763.65
2015-01-21	47804.2
2015-01-22	50300.9
2015-01-23	52724.6
2015-01-24	55013.85
2015-01-25	56631.4
2015-01-26	58515.8
2015-01-27	61043.85
2015-01-28	63059.85
2015-01-29	65105.15
2015-01-30	67375.45
2015-01-31	69793.3
2015-02-01	72982.5
2015-02-02	75311.1
2015-02-03	77925.9
2015-02-04	80159.8
2015-02-05	82375.6

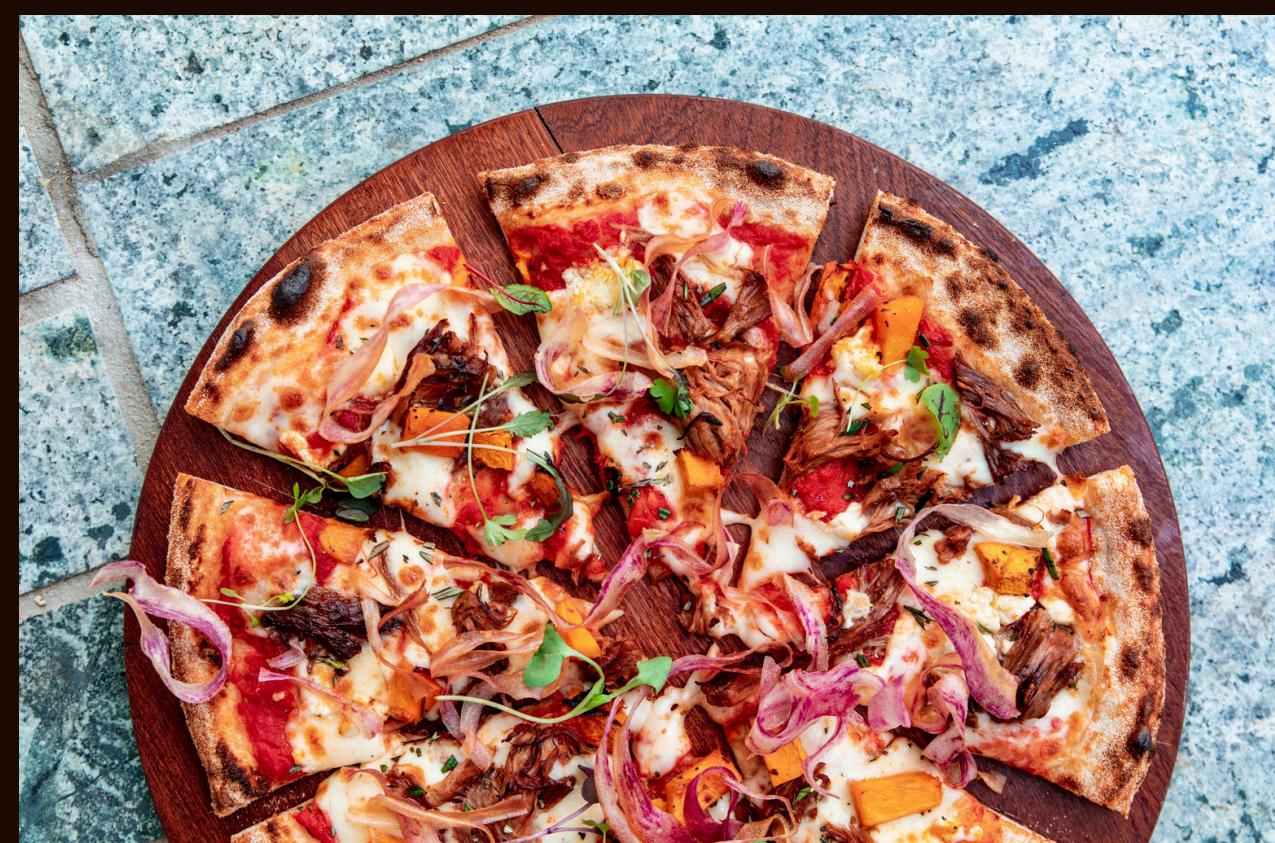
order_date	cumulative_revenue
2015-02-06	84885.55
2015-02-07	87123.2
2015-02-08	89158.2
2015-02-09	91353.55
2015-02-10	93410.05
2015-02-11	95870.05
2015-02-12	98028.85
2015-02-13	100783.35
2015-02-14	103102.5
2015-02-15	105243.75
2015-02-16	107212.55
2015-02-17	109334.45
2015-02-18	111977.3
2015-02-19	114007.55
2015-02-20	116898.7
2015-02-21	119009.7
2015-02-22	120589.65
2015-02-23	122758.2

2015-12-14	785389.55
2015-12-15	787777
2015-12-16	790011.8
2015-12-17	791892.55
2015-12-18	794778.85
2015-12-19	797083.05
2015-12-20	799187.95
2015-12-21	801288.65
2015-12-22	803171.6
2015-12-23	805415.9
2015-12-24	807553.75
2015-12-26	809196.8
2015-12-27	810615.8
2015-12-28	812253
2015-12-29	813606.25
2015-12-30	814944.05
2015-12-31	817860.05

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

```
select category ,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;
```

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



# THANK YOU

