

# PIZZA SALES REPORT

BY  
SHUBHAM KATKAR



# OBJECTIVE

Utilized SQL to analyze one year of pizza sales data, extracting valuable insights for informed decision-making.

The project aimed to identify key trends, customer preferences, and sales patterns, providing a foundation for optimizing inventory, and operational efficiency to drive future growth and profitability



# PROBLEM STATEMENT

## BASIC:

- 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.

## INTERMEDIATE:

- 6) JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.
- 7) DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.
- 8) JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.
- 9) GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.
- 10) DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

## ADVANCED:

- 11) CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.
- 12) ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.
- 13) DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.



# LET'S BEGIN ANALYSIS USING



RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.



SELECT

COUNT(order\_id) AS total\_orders

FROM

orders;

Result Grid |



Filter Rows:

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
```

Result Grid		Filter Rows:
total_sales		
▶	817860.05	



# 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;
```

Result Grid				Filter Rows:
	total_orders			
▶	21350			



# 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;
```

Result Grid | Filter Rows:

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



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

```
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;
```

Result Grid | Filter Rows:

	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

JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

```
SELECT pizza_types.category, 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.category
ORDER BY quantity DESC;
```

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050



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



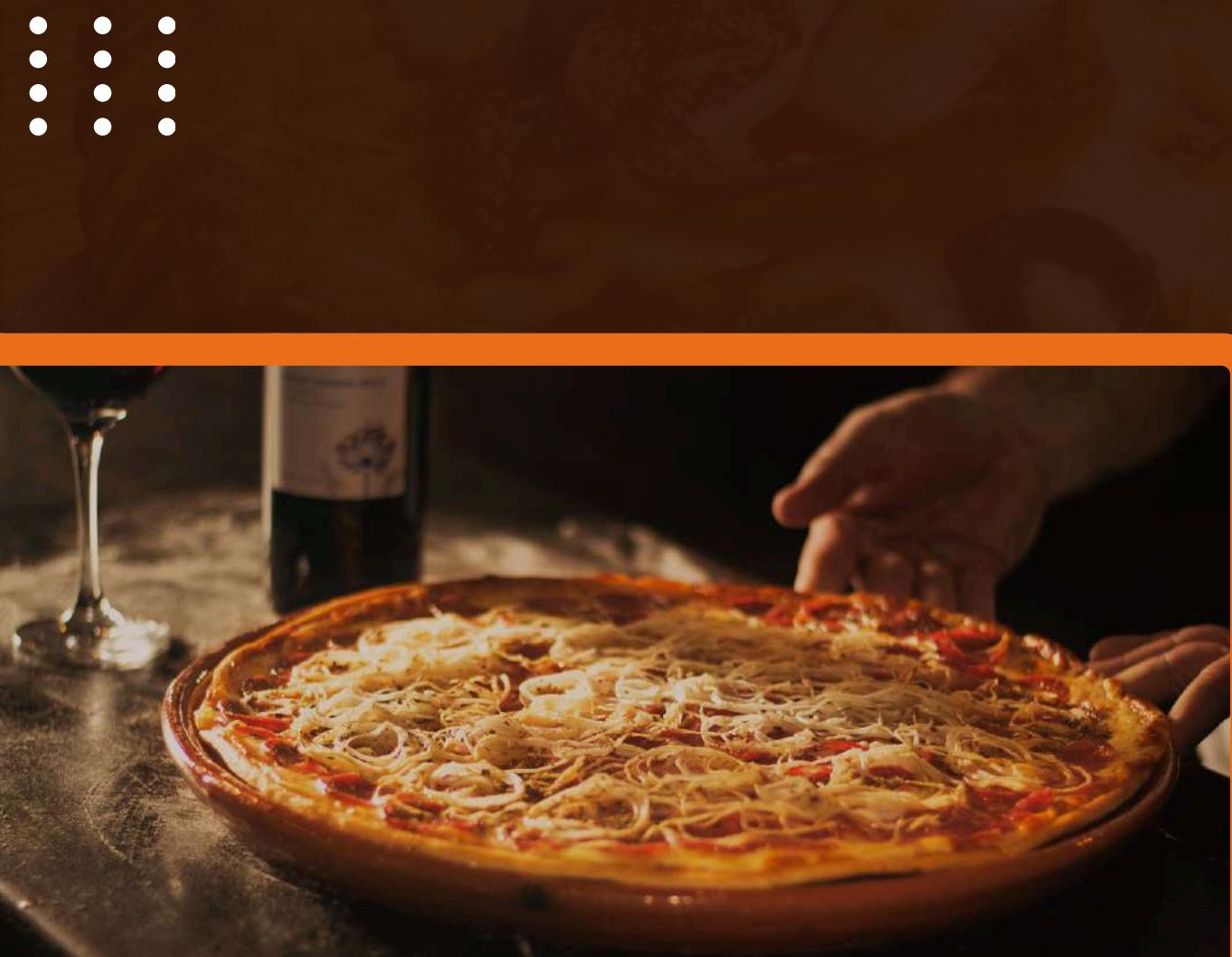
SELECT

```
HOUR(order_time), COUNT(order_id)as order_count
```

FROM

```
orders
```

```
GROUP BY HOUR(order_time);
```



Result Grid | Filter Rows:

	HOUR(order_time)	order_count
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336
	18	2399
	19	2266

# JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.

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

Result Grid		Filter Rows:
	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)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;
```

round(avg(quantity),0)
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 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 ;
```



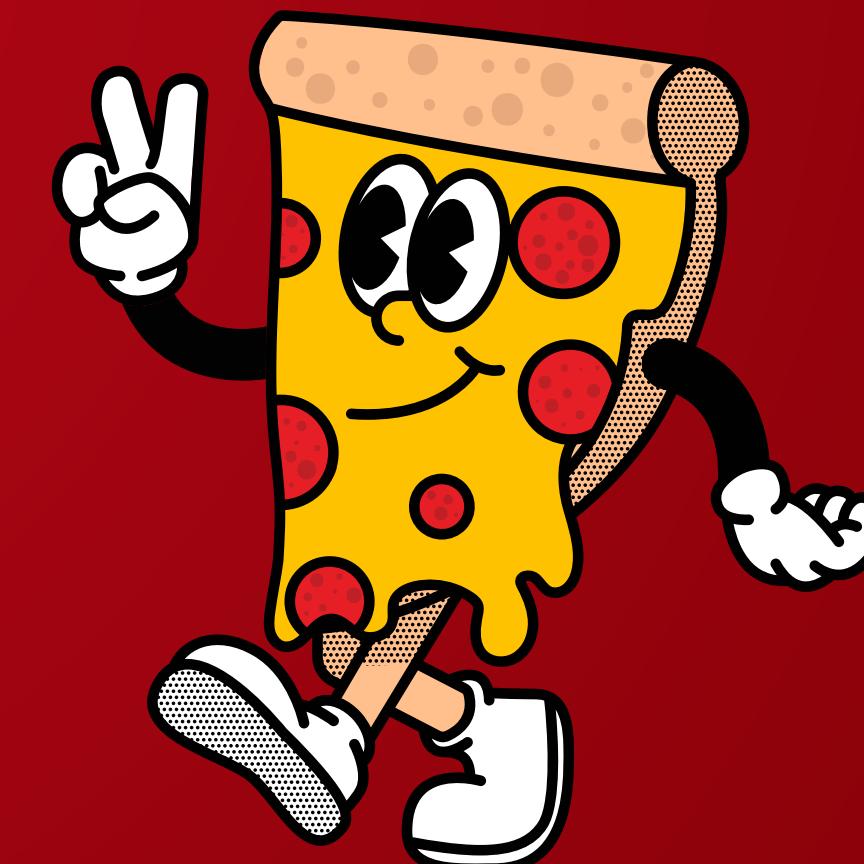
Result Grid |  Filter Rows: 

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

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

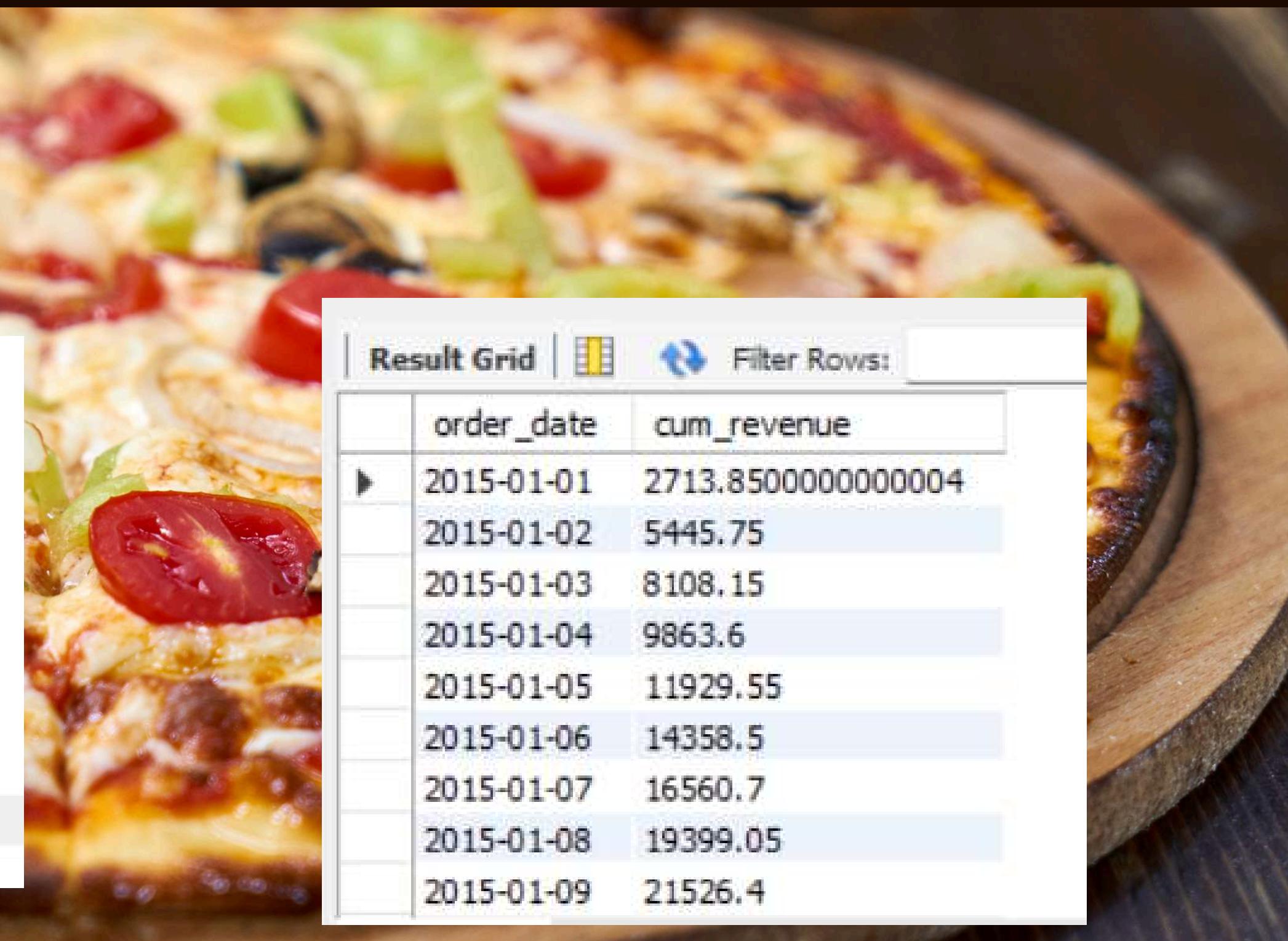
```
select pizza_types.category,  
       (sum(order_details.quantity * pizzas.price) / (SELECT  
              ROUND(SUM(order_details.quantity * pizzas.price),  
                    2) AS total_sales  
        FROM  
              order_details  
        JOIN  
              pizzas ON pizzas.pizza_id = order_details.pizza_id))*100 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;
```

	category	revenue
▶	Classic	26.90596025566967
	Supreme	25.45631126009862
	Chicken	23.955137556847287
	Veggie	23.682590927384577



# ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
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;
```

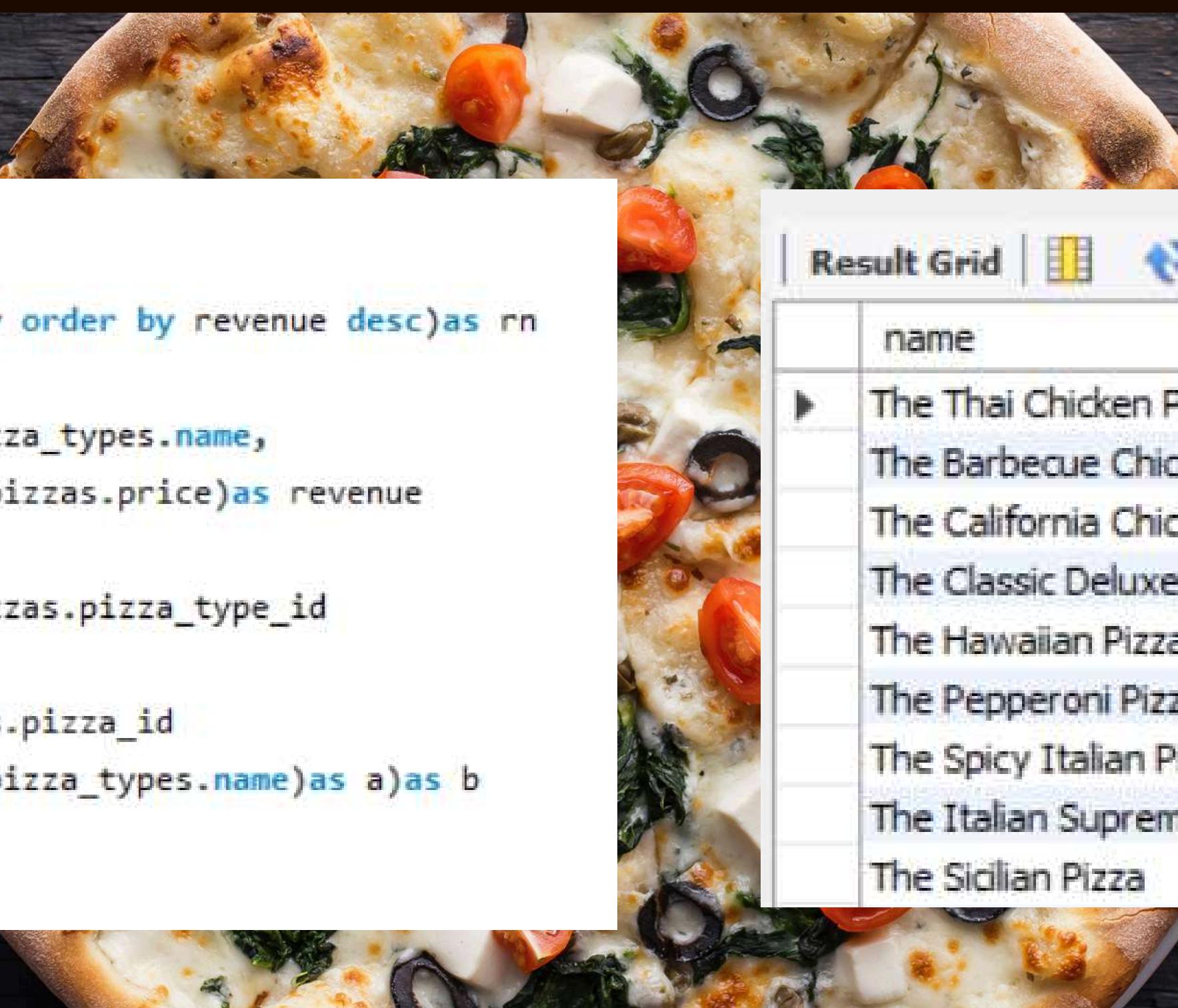


A close-up photograph of a pizza with various toppings like pepperoni, cheese, and vegetables.

Result Grid		Filter Rows:
	order_date	cum_revenue
▶	2015-01-01	2713.8500000000004
	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

# 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)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 b
where rn<=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

THANK YOU FOR  
YOUR ATTENTION

