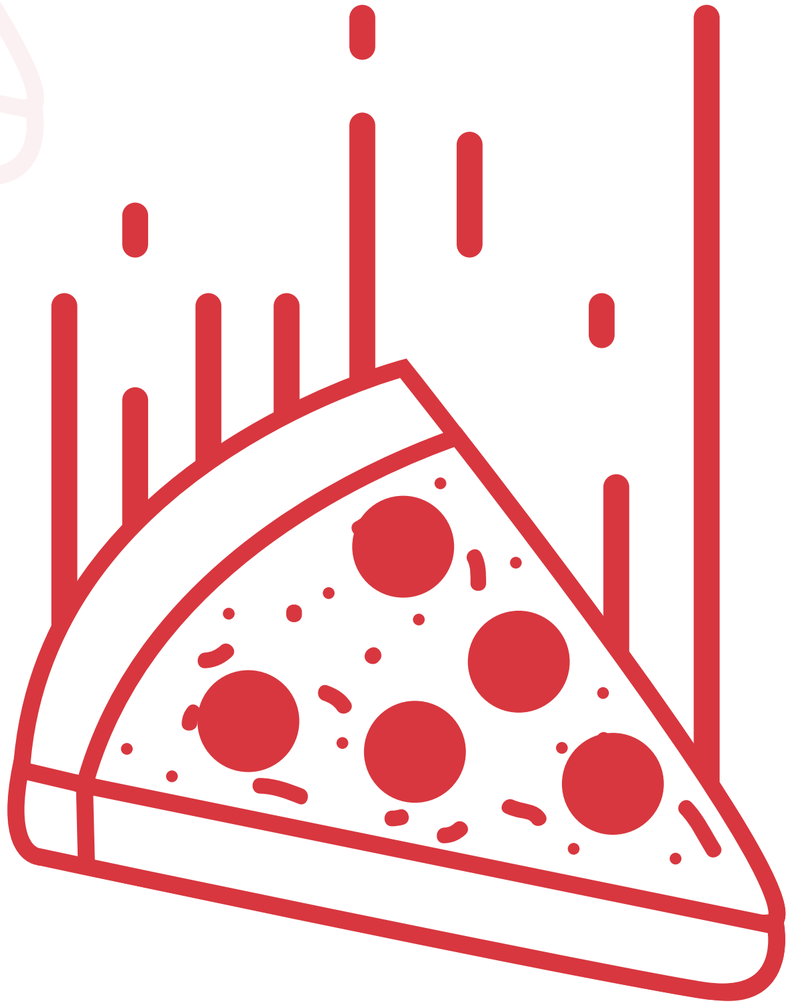
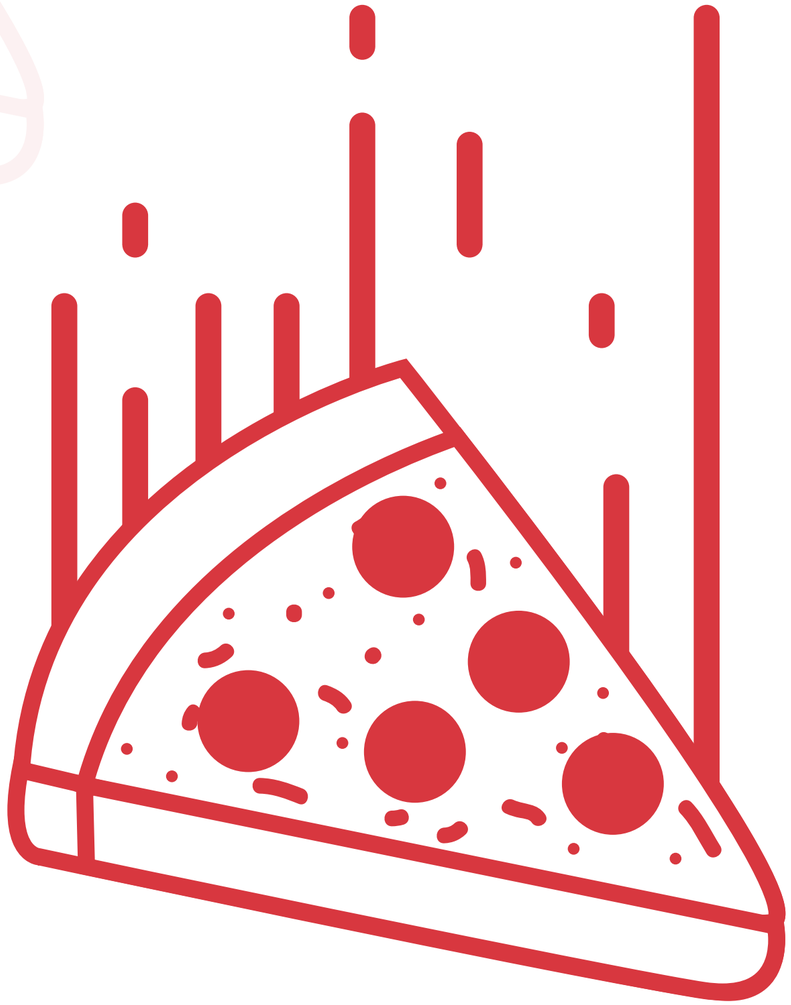
ANALYZING PIZZA SALES WITH MYSQL

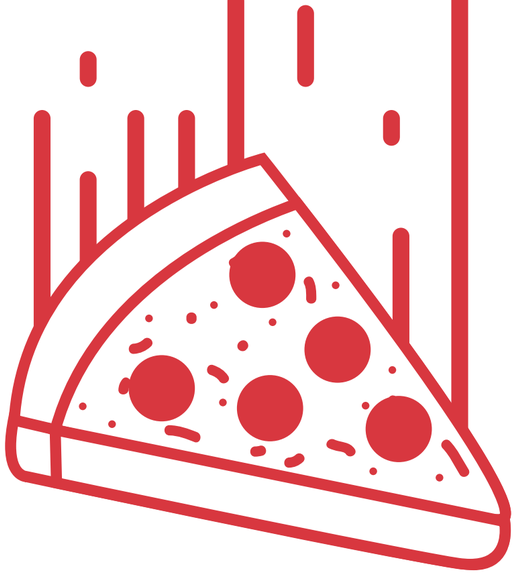


HELLO !

I am Muhammad Jahidul Islam In this project, I used MySQL to analyze some pizza sales datasets, running queries to uncover key trends and insights. This helped me identify sales patterns, popular products, and customer behavior, demonstrating my ability to leverage data for business insights.



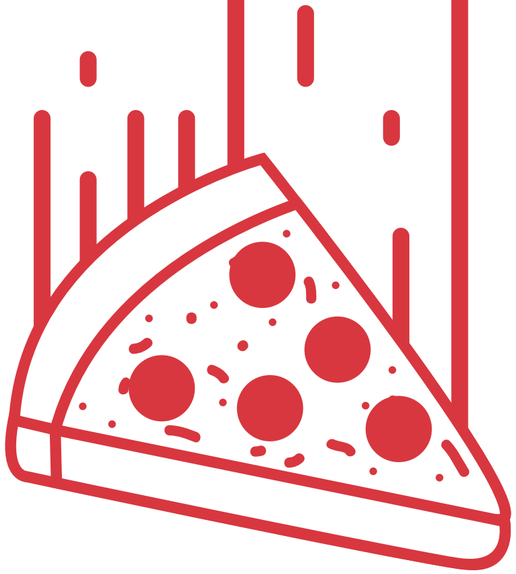
RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.



```sql
SELECT
    COUNT(order_id) AS Total_Order
FROM
    orders;
```

	Total_Order
▶	21350

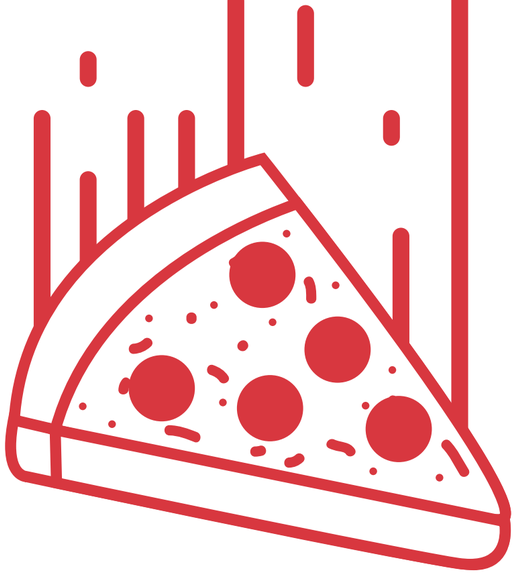
CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.



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

	Total_Revenue
▶	817860.05

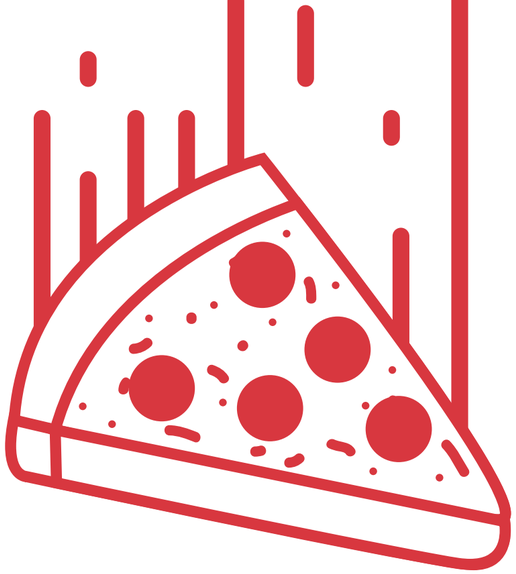
IDENTIFY THE HIGHEST-PRICED PIZZA.



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

	name	highest_price
▶	The Greek Pizza	35.95

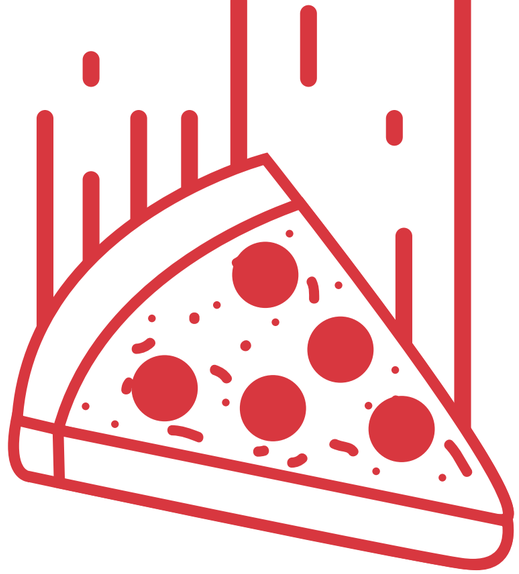
IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.



```
SELECT
   size, COUNT(*) AS highest_sale
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY size
ORDER BY highest_sale DESC
LIMIT 1;
```

	size	highest_sale
▶	L	18526

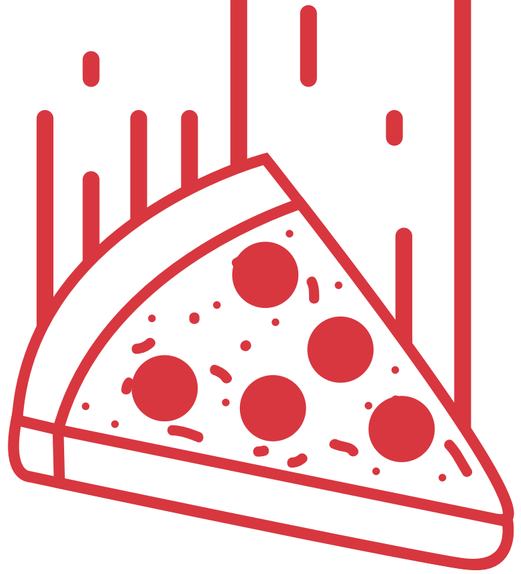
LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.



```sql
SELECT
    pizza_types.name AS name,
    sum(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY name
ORDER BY quantity DESC
LIMIT 5;
```

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

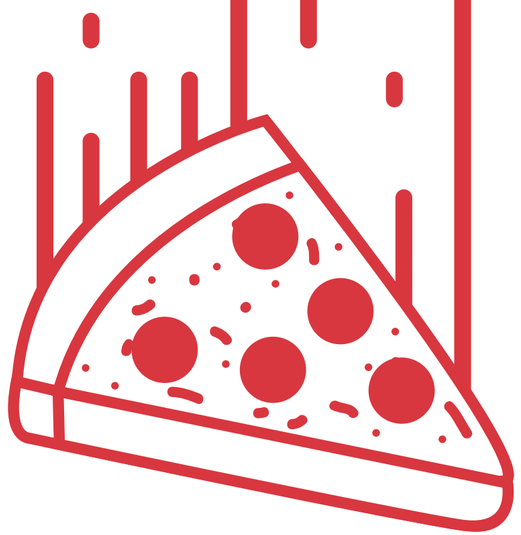
JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.



```sql
SELECT
    pizza_types.category AS category,
    sum(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas 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
LIMIT 3;
```

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649

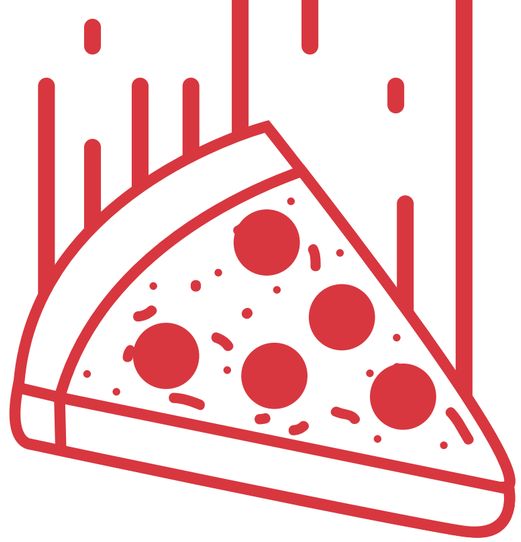
DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.



```
SELECT
    HOUR(order_time) as Orders_hour, count(order_id) AS Hour_Distri
FROM
    orders
GROUP BY Orders_hour
ORDER BY Orders_hour ;
```

	Orders_hour	Hour_Distri
▶	9	1
	10	8
	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336
	18	2399
	19	2009

FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.

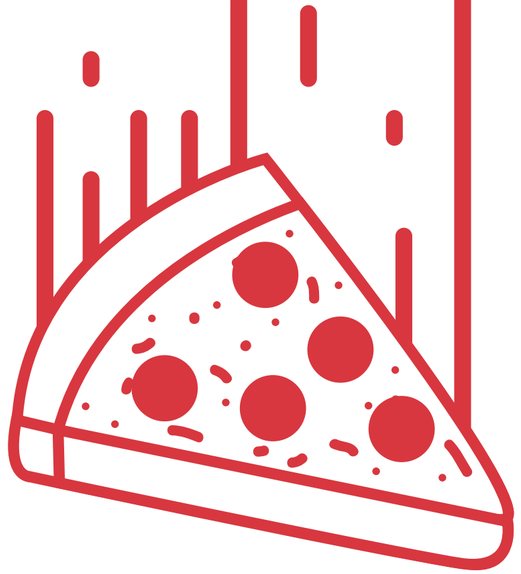


```sql
SELECT
    category,
    COUNT(category) AS quantity
FROM
     pizza_types

GROUP BY category
ORDER BY quantity;
```

category	quantity
Chicken	6
Classic	8
Supreme	9
Veggie	9

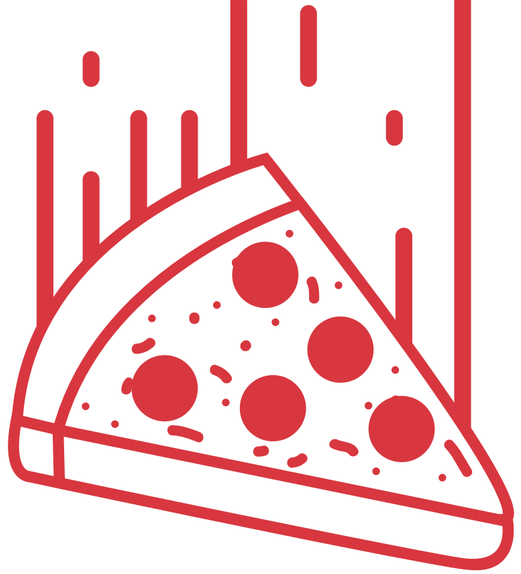
GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.



```sql
SELECT
    ROUND(AVG(quantity), 2) AS Avg_Sale_Per_Day
FROM
    (SELECT
        order_date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY order_date) AS order_quantity
;
```

Avg_Sale_Per_Day
138.47

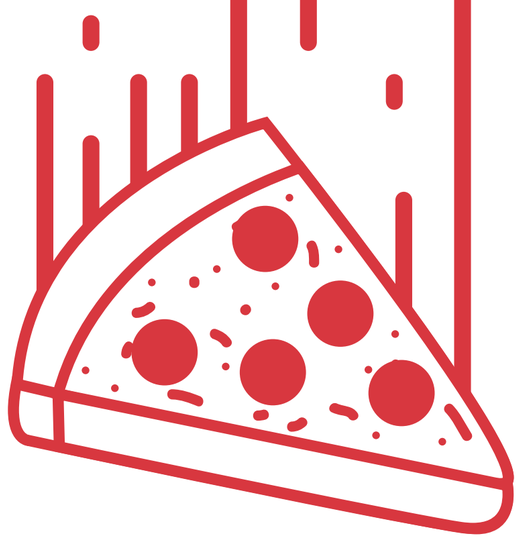
DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.



```sql
SELECT
    pizza_types.name AS name,
    SUM(pizzas.price * order_details.quantity) AS revenue
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
        JOIN
    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
GROUP BY  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

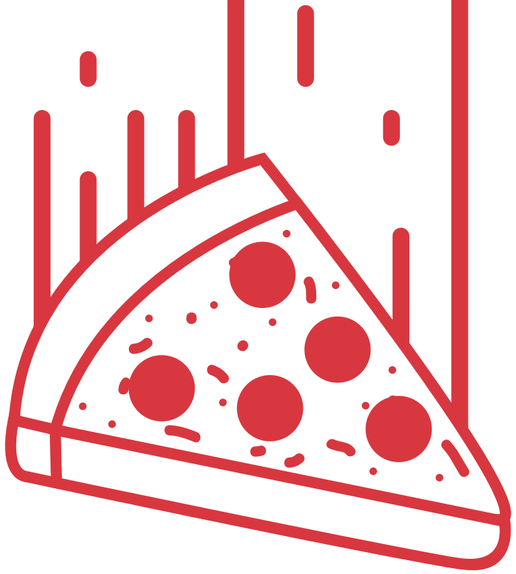
CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.



```sql
SELECT
    pizza_types.category AS name,ROUND(SUM(pizzas.price * order_details.quantity),2) as revenue,
    ROUND(SUM((pizzas.price * order_details.quantity) / total_revenue) * 100,2) AS revenue_percentage
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
        JOIN
    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    (SELECT
        ROUND(SUM(pizzas.price * order_details.quantity), 2) AS Total_Revenue
    FROM
        pizzas
    JOIN order_details ON pizzas.pizza_id = order_details.pizza_id) AS total_revenue
GROUP BY pizza_types.category
ORDER BY revenue DESC
;
```

	name	revenue	revenue_percentage
▶	Classic	220053.1	26.91
	Supreme	208197	25.46
	Chicken	195919.5	23.96
	Veggie	193690.45	23.68

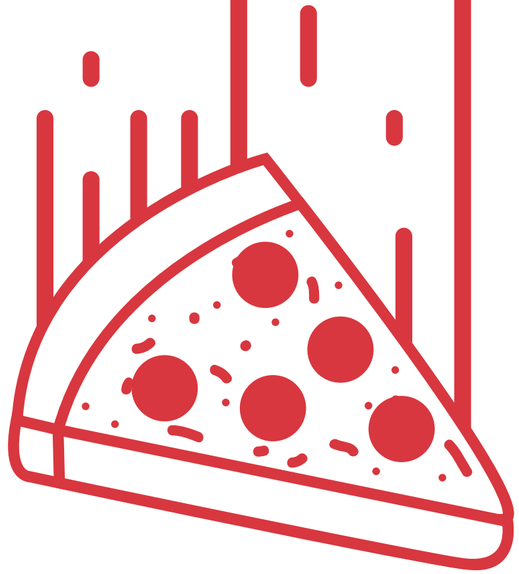
ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.



```
SELECT dates,SUM(Revenue) OVER(order by dates) as Cum_Revenue
FROM
(SELECT
    orders.order_date AS dates,
    ROUND(SUM(pizzas.price * order_details.quantity),2) AS Revenue
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
        JOIN
    orders ON orders.order_id = order_details.order_id
GROUP BY dates) as Total
;
```

	dates	Cum_Revenue
▶	2015-01-01 00:00:00	2713.85
	2015-01-02 00:00:00	5445.75
	2015-01-03 00:00:00	8108.15
	2015-01-04 00:00:00	9863.6
	2015-01-05 00:00:00	11929.55
	2015-01-06 00:00:00	14358.5
	2015-01-07 00:00:00	16560.7
	2015-01-08 00:00:00	19399.05
	2015-01-09 00:00:00	21526.4
	2015-01-10 00:00:00	23990.35

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



```sql
SELECT name,revenue FROM
(SELECT category,name,revenue,
rank() over(partition by category order by revenue desc) as rn
FROM
(SELECT
    pizza_types.category AS category,
    pizza_types.name AS name,
    ROUND(SUM(pizzas.price * order_details.quantity),2) AS revenue
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
        JOIN
    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
GROUP BY pizza_types.category,pizza_types.name
ORDER BY revenue DESC
LIMIT 3) AS Rank_Table) as a
;
```

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

THANK YOU