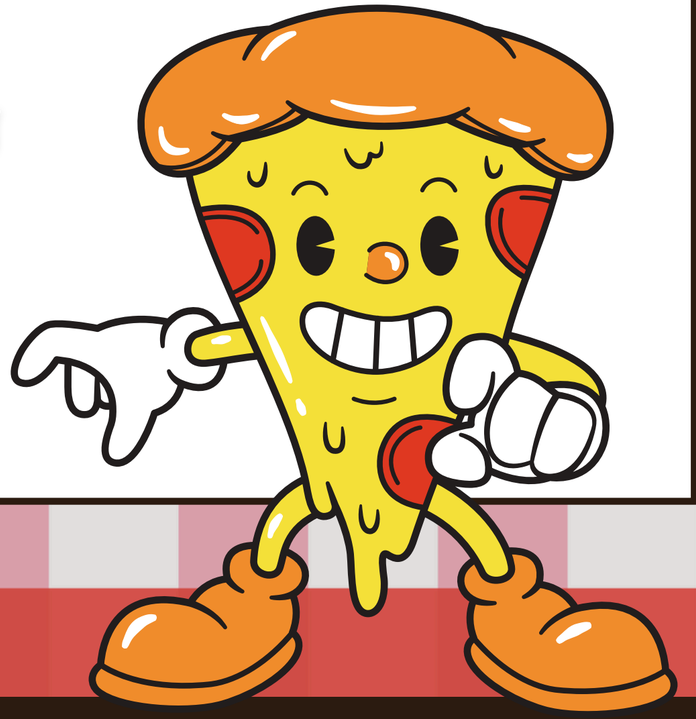
HELLO !!

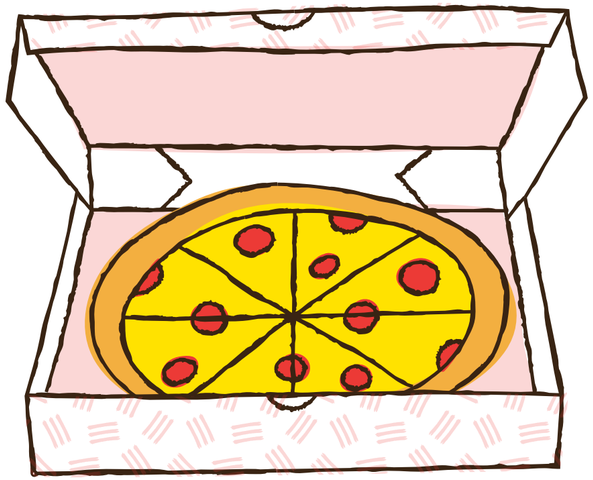
MY NAME IS SUYASH PATIDAR ..
I HAVE DONE ONE SMALL PROJECT ON SQL QUERIES
ON PIZZA SALES ....
IN WHICH I UTILIZE SQL QUERIES TO SOLVE QUESTION
THAT WAS RELATED TO PIZZA SALES.....

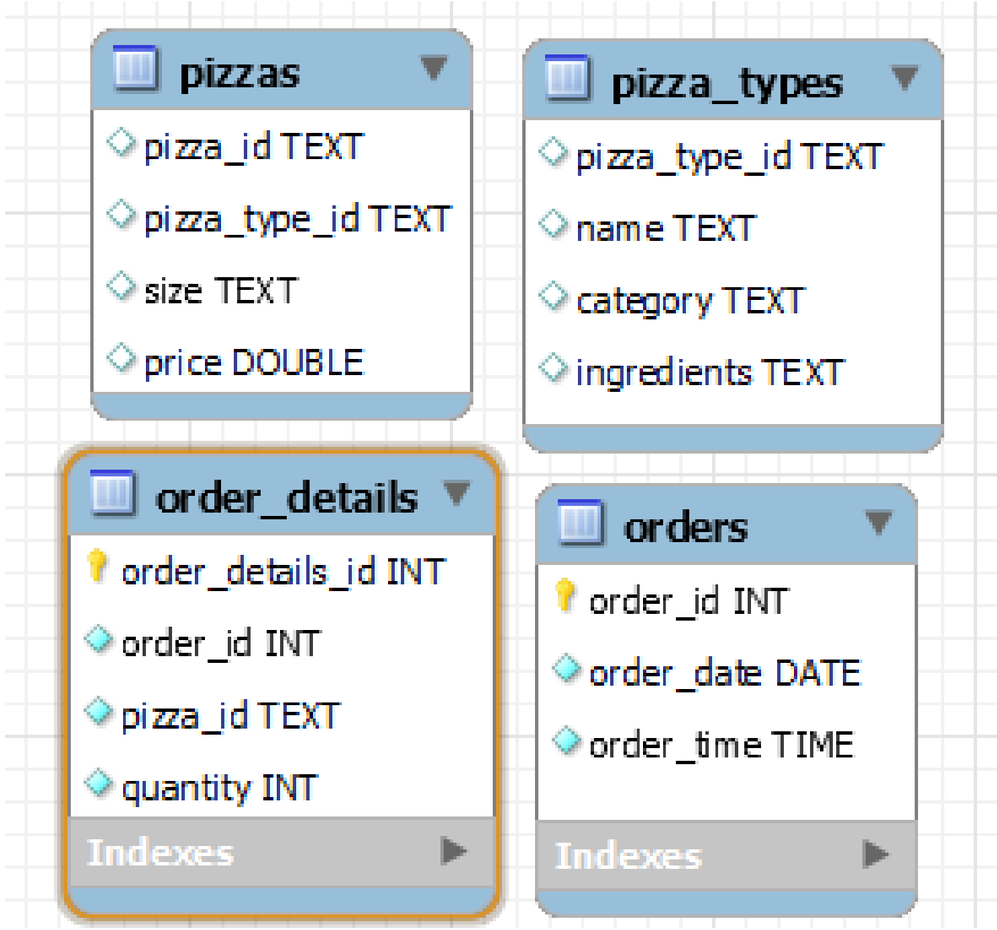


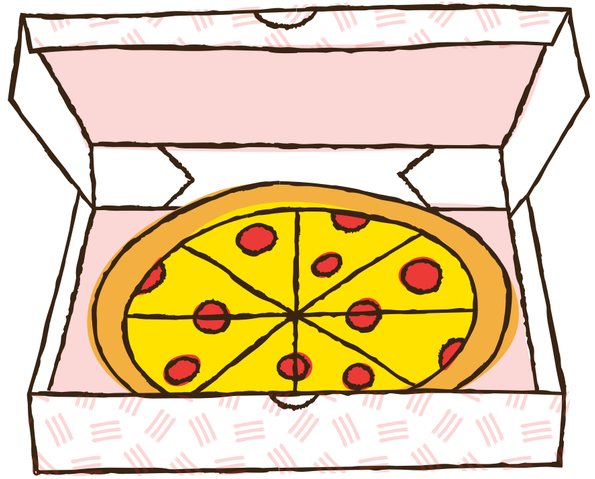
Its yummy !!

This pizza is amazing!

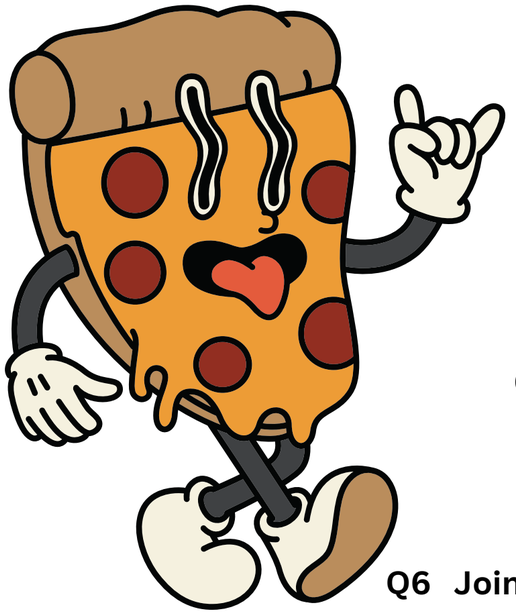
pizza’s table





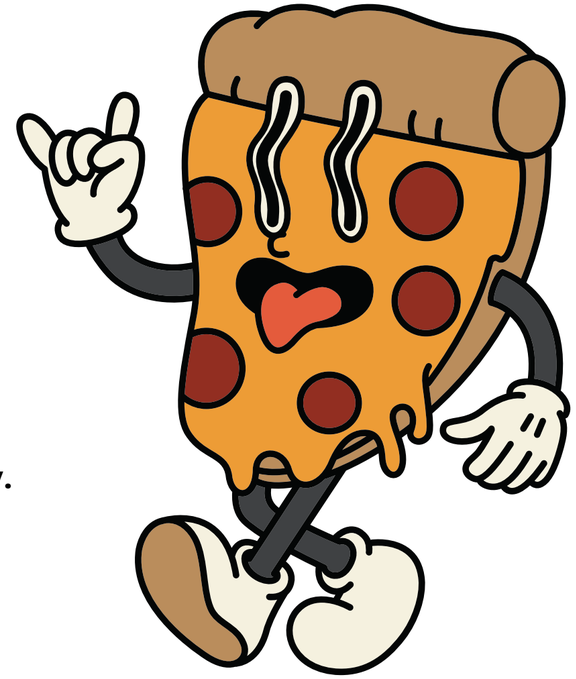


The Questions which i have solved here



Basic:

Q1 Retrieve the total number of orders placed.
Q2 Calculate the total revenue generated from pizza sales.
Q3 Identify the highest-priced pizza.
Q4 Identify the most common pizza size ordered.
Q5 List the top 5 most ordered pizza types along with their quantities.



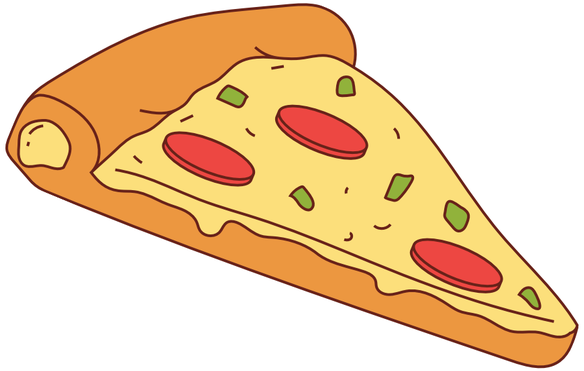
Intermediate:

Q6 Join the necessary tables to find the total quantity of each pizza category ordered.
Q7 Determine the distribution of orders by hour of the day.
Q8 Join relevant tables to find the category-wise distribution of pizzas.
Q9 Group the orders by date and calculate the average number of pizzas ordered per day.
Q10 Determine the top 3 most ordered pizza types based on revenue.

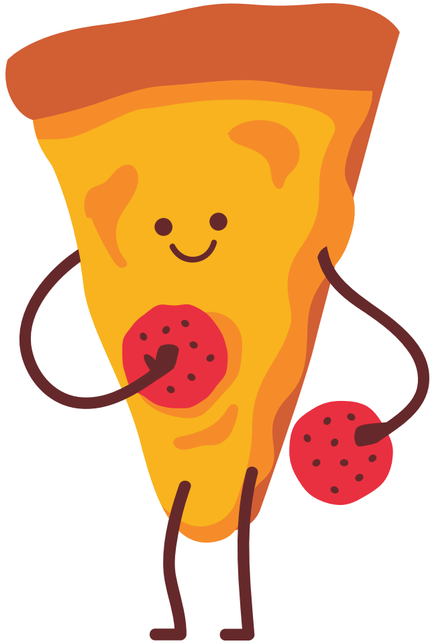
Advanced:

Q11 Calculate the percentage contribution of each pizza type to total revenue.
Q12 Analyze the cumulative revenue generated over time.
Q13 Determine the top 3 most ordered pizza types based on revenue for each pizza category.

Q1 retrieve the total number of order placed,



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



Result Grid

	total_order
▶	21350

Q2 calculated total revenue generated from pizza sales.

```sql
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

	total_sales
▶	817860.05

Q3 Identify the highest price pizza

```sql
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 Rov

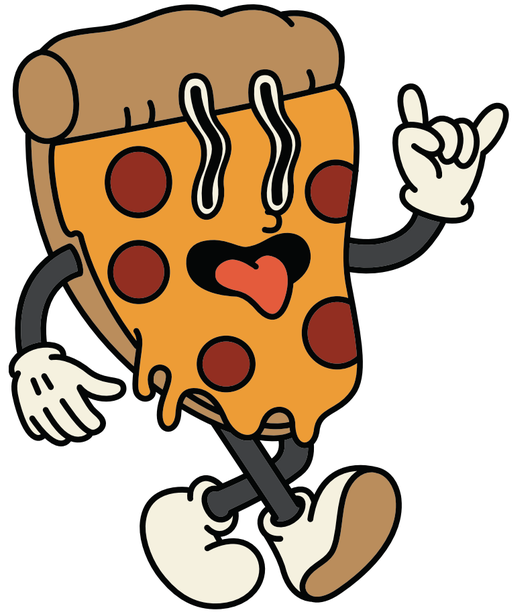
	name	price
▶	The Greek Pizza	35.95

Q4 identify the most common pizza size ordered.

```sql
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 ;
```

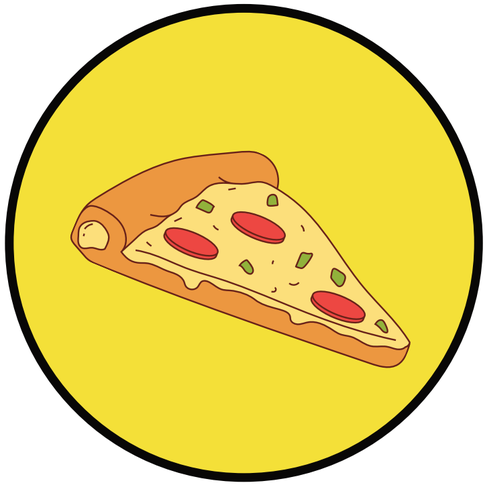
Result Grid

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



Q5.. List the top 5 most ordered pizza type 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

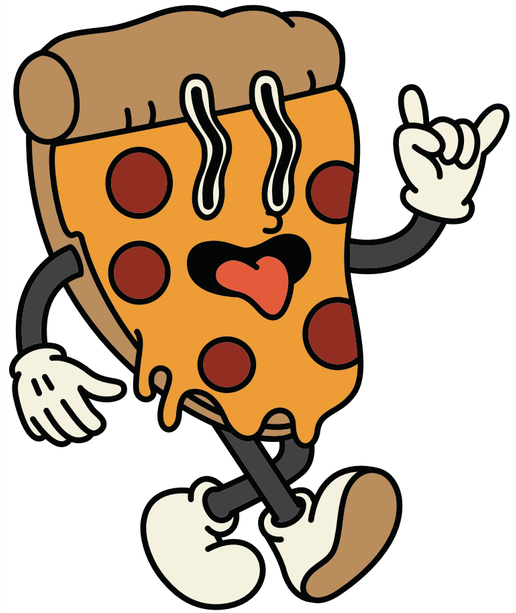
Q6 List the top 5 most ordered pizza types along with their quantities.

```sql
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_type_id
GROUP BY pizza_types.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

Q 7 determine the distribution of order by hour of the day.

```
SELECT
    HOUR(order_time) AS hour, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY HOUR(order_time);
```

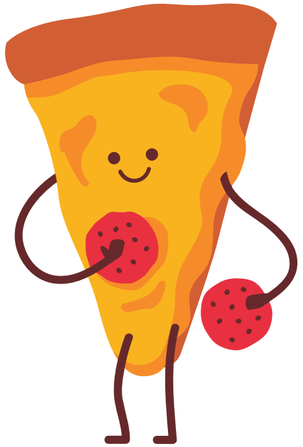


hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2390

Result 1

Q 8 join relevent tables to find the catogory wise distribution of pizzas.

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

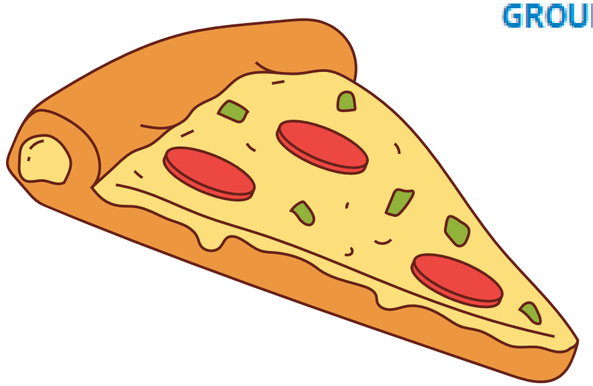


Result Grid | Filter Rows:

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

Q9 group the orders by date and calculate the average number of pizzas ordered per day.

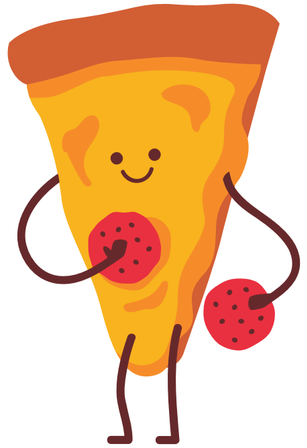
```sql
SELECT
    ROUND(AVG(quantity), 0) as avg_pizza_ordered_per_day
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS order_quantity;
```



	avg_pizza_ordered_per_day
▶	138

Q10 deteremine the top 3 most ordered pizza types based on revenue.

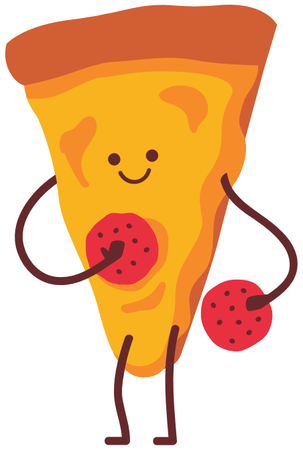
```sql
SELECT
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue DESC
LIMIT 3;
```



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

Q11 Calculate the percentage contribution of each pizza type to total revenue.

```
SELECT
    pizza_types.category,
    round( 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)*100,2) 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.category
ORDER BY revenue DESC;
```

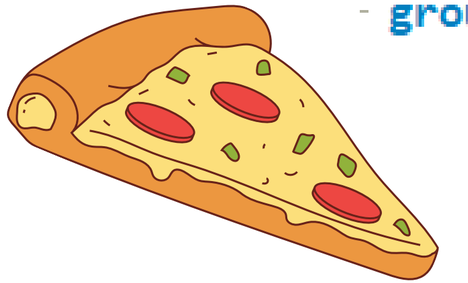


Result Grid | Filter R

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

Q 12 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;
```



Result Grid | Filter Rows:

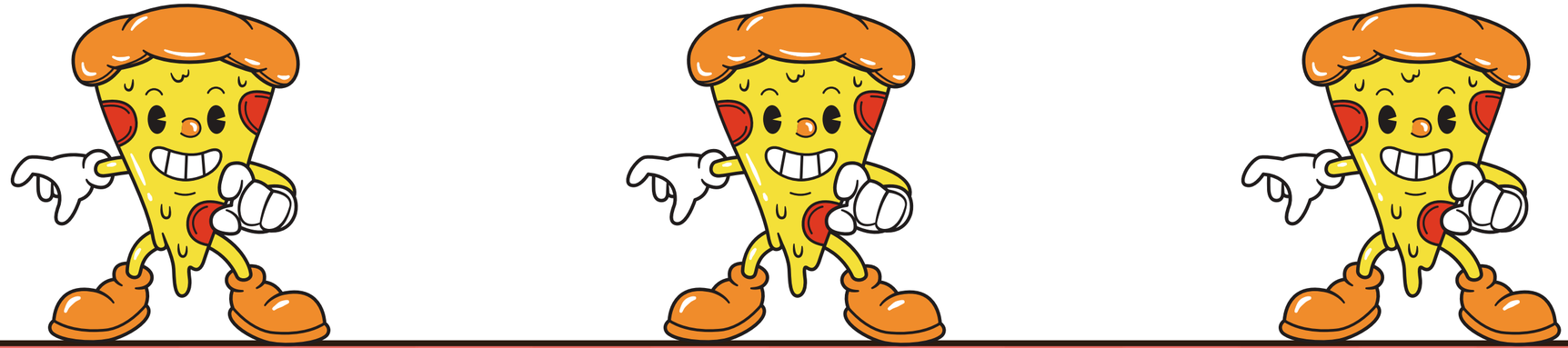
order_date	cum_revenue
2015-01-01	2713.8500000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399.05

Q13 determine the top 3 ordered pizza types based on revenue for each pizza category.

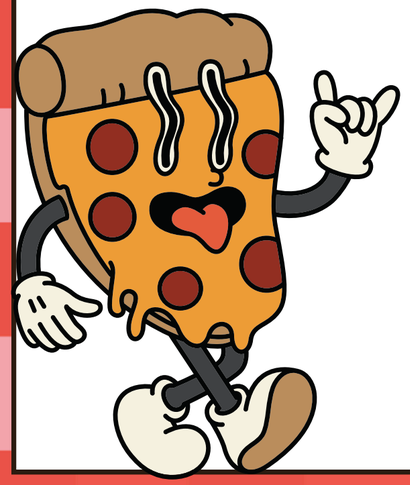
```sql
select name, revenue from
(select category, name, revenue,
rank() over(partition by category order by revenue desc) as rn
from
(select pizza_types.category, pizza_types.name,
sum((order_details.quantity) * pizzas.price) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
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;
```

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

PIZZA PARTY!



Thank you !!!



Pizza

TIME