Pizza

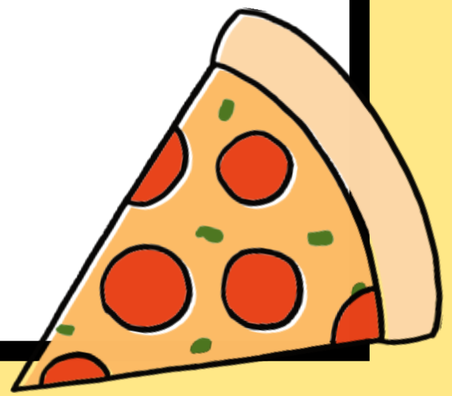
Sales Analysis

Analysis #1

Retrieve the total number of orders placed.

```sql
-- Retrieve the total number of orders placed.
SELECT COUNT(order_id) FROM orders;
```

	COUNT(order_id)
▶	21350

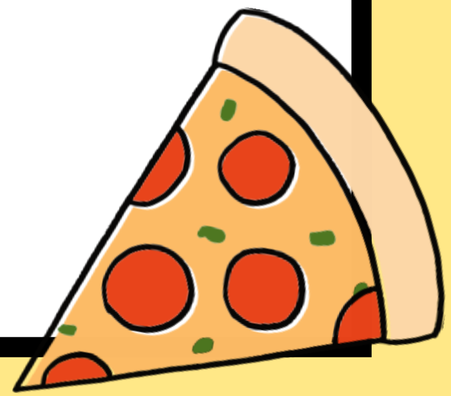


Analysis #2

Calculate the total revenue generated from pizza sales.

```sql
-- Calculate the total revenue generated from pizza sales.
SELECT
    SUM(pizzas.price * order_details.quantity) AS total_revenue
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id;
```

	total_revenue
▶	817860.05

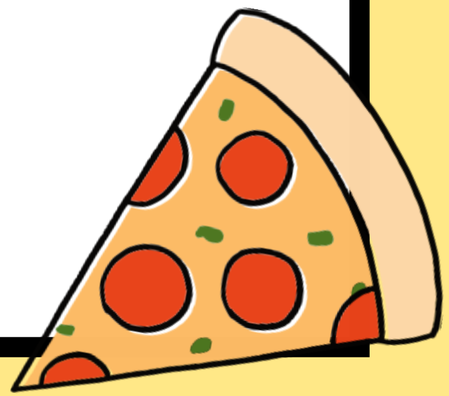


Analysis #3

Identify the highest-priced 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;
```

	name	price
▶	The Greek Pizza	35.95

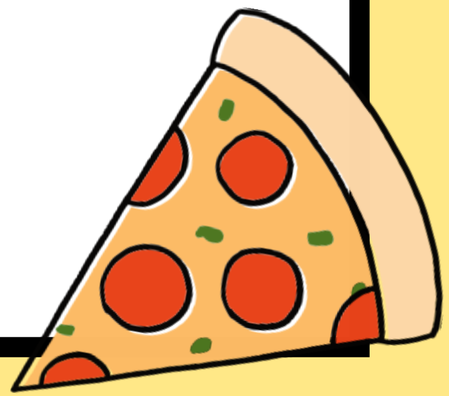


Analysis #4

Identify the most common pizza size ordered.

```
SELECT
    pizzas.size, COUNT(order_details.order_details_id)
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size;
```

size	COUNT(order_details.order_details_id)
M	15385
L	18526
S	14137
XL	544
XXL	28

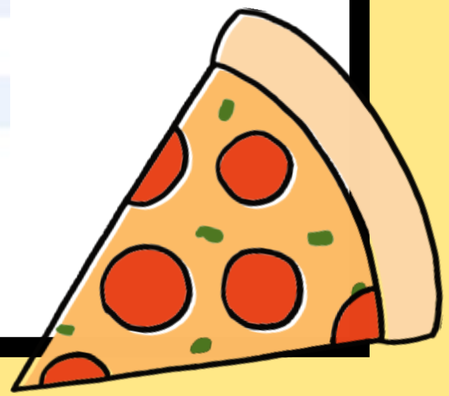


Analysis #5

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 pizzas.pizza_id = order_details.pizza_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

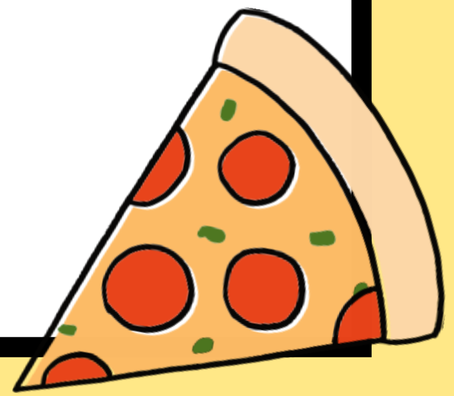


Analysis #7

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

```sql
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 pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity desc;
```

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

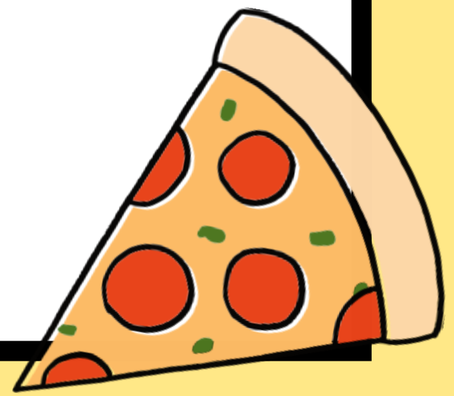


Analysis #8

Determine the distribution of orders by hour of the day.

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

	hour	COUNT(order_id)
	14	1472
	15	1468
	16	1920
	17	2336
	18	2399
	19	2009
	20	1642
	21	1198
	22	663
	23	28
	10	8
	9	1

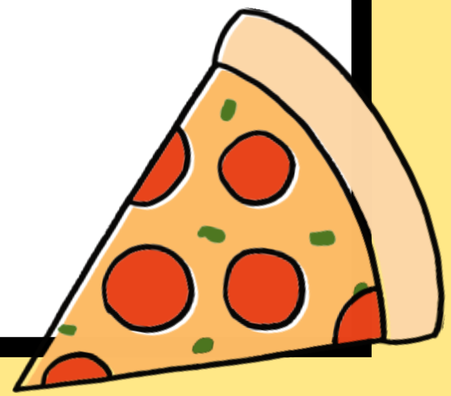


Analysis #9

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

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

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

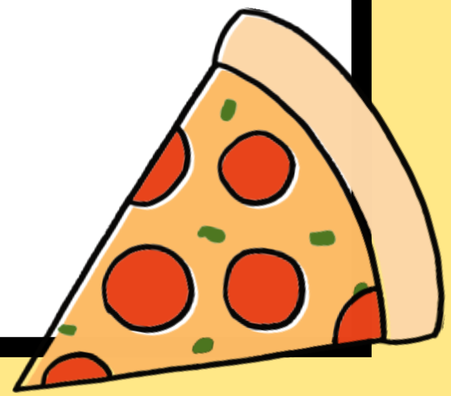


Analysis #10

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 ordee_quantity;
```

	round(AVG(quantity),0)
▶	138

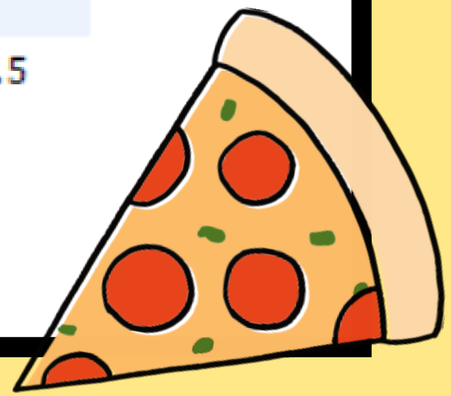


Analysis #11

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

```sql
SELECT
    pizza_types.name,
    SUM(pizzas.price * order_details.quantity) 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
        JOIN
    orders ON order_details.order_id = orders.order_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

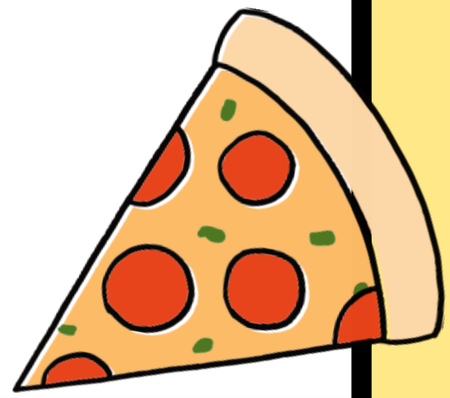


Analysis #12

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

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

	category	total_revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

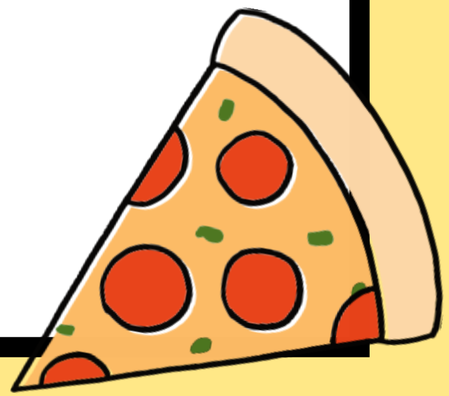


Analysis #13

Analyze the cumulative revenue generated over time.

```
select order_date,
round(sum(revenue) over(order by order_date),2) 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;
```

order_date	cum_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	23990.35
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



Thank You!