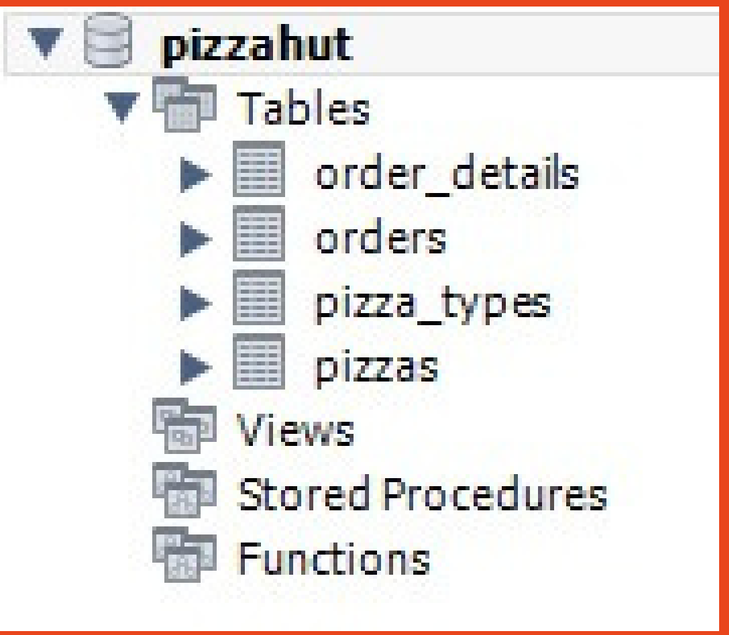
SQL PROJECT ON PIZZA SALES

Hello!

SQL project, I created a database to study pizza sales.The database had tables for customers, pizzas, sales.

This project taught me how to use SQL to get useful information from data and helped me understand pizza sales better.

Schema



SQL queries

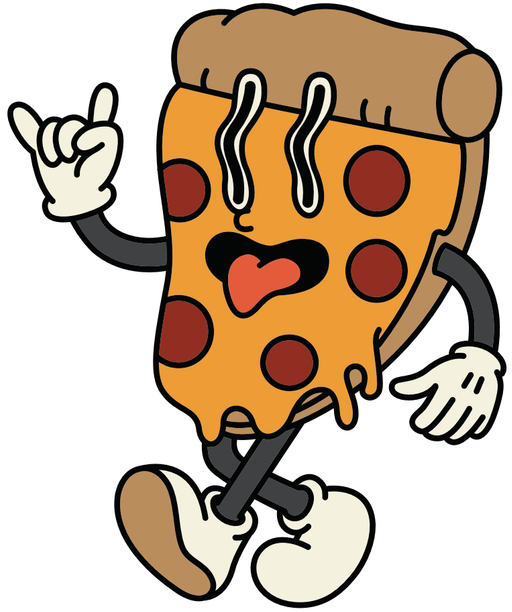
```
Basic:
Retrieve the total number of orders placed.
Calculate the total revenue generated from pizza sales.
Identify the highest-priced pizza.
Identify the most common pizza size ordered.
List the top 5 most ordered pizza types along with their quantities.


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

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

Rerive the total number of orders placed

```sql
SELECT
    COUNT(ORDER_ID) AS total_orders
FROM
    orders;
```

Result Grid

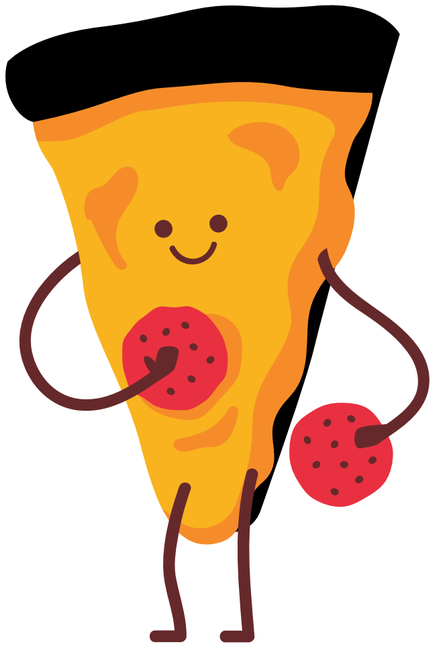
total_orders
18511



CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES

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

total_sale
99219



IDENTIFY THE HIGHEST-PRICEED 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 Rows:

name	price
The Greek Pizza	35.95

Identify the most common pizza ordered

```sql
SELECT
    pizzas.size, COUNT(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	2265
M	1837
S	1748
XL	63
XXL	3

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

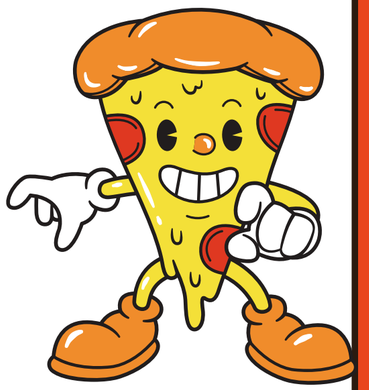
```sql
SELECT
    pizza_types.name,
    SUM(order_details.quantity) AS total_quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.name
ORDER BY total_quantity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

name	total_quantity
The Pepperoni Pizza	334
The Barbecue Chicken Pizza	307
The California Chicken Pizza	288
The Hawaiian Pizza	272
The Classic Deluxe Pizza	269

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

```sql
SELECT
    pizza_types.name,
    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 pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.name
ORDER BY total_quantity DESC
LIMIT 5;
```

category	quantity
Classic	1789
Supreme	1495
Veggie	1448
Chicken	1292



Determine the distribution of orders by hour of the day

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

	HOUR(order_time)	COUNT(order_id)
▶	11	1062
	12	2169
	13	2122
	14	1297
	15	1283
	16	1664
	17	2041
	18	2083
	19	1716
	20	1411

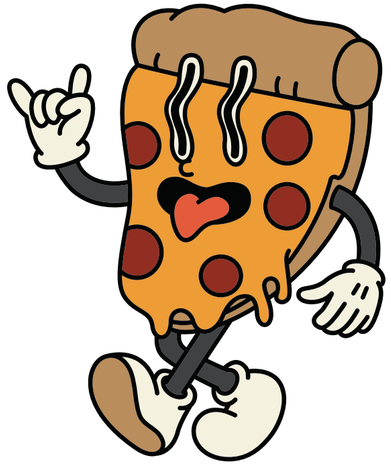
Join relevant tables to find the category -wise distribution of pizzas.

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

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

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;
```

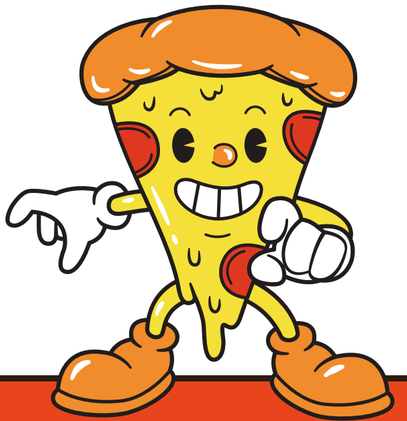


avg_pizza_ordered_per_day
137

Determine the top 3 most ordered pizza types based on revenue

```sql
SELECT
    pizza_types.name,
    ROUND(SUM(order_details.quantity * pizzas.price),
            0) 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;
```

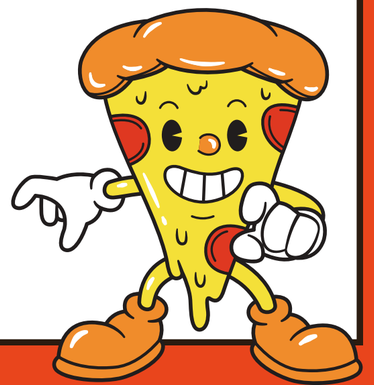
name	revenue
The Barbecue Chicken Pizza	5470
The California Chicken Pizza	4960
The Thai Chicken Pizza	4776



Calculate the percentage contribution of each pizza type to total revenue

```sql
SELECT
    pizza_types.category,
    ROUND((SUM(order_details.quantity * pizzas.price) / (SELECT
                    ROUND(SUM(order_details.quantity * pizzas.price),
                                0) AS total_sale
                FROM
                    order_details
                        JOIN
                    pizzas ON order_details.pizza_id = pizzas.pizza_id)) * 100,
            0) 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	27
	Veggie	24
	Supreme	26
	Chicken	23



Analyze the cumulative revenue generated over time

```sql
select order_date, sum(total_revenue) over (order by order_date) as cum_revenue
from
 (select orders.order_date, sum(order_details.quantity*pizzas.price)
as total_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	2713.8500000000004
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
2015-01-10	23990.350000000002
2015-01-11	25862.65

Determine the top 3 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 ranking
from
(select pizza_types.category,pizza_types.name , round(sum(order_details.quantity*pizzas.price),0) 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 ranking<= 3 ;
```

name	revenu
The Barbecue Chicken Pizza	5470
The California Chicken Pizza	4960
The Thai Chicken Pizza	4776
The Pepperoni Pizza	4189
The Classic Deluxe Pizza	4153
The Hawaiian Pizza	3574
The Sicilian Pizza	4223
The Italian Supreme Pizza	4222
The Spicy Italian Pizza	4135
The Four Cheese Pizza	3933
The Five Cheese Pizza	3570