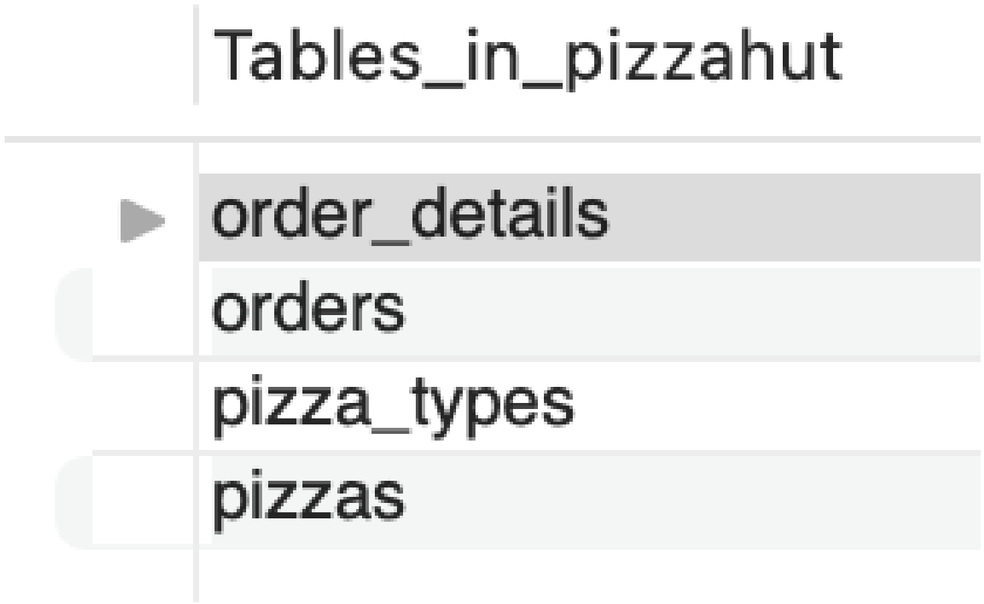
PIZZA STORE ANALYSIS

Pizza Store Database

- Our database named “**pizzahut**” consists of 4 tables containing information about the orders and pizzas.
- “**Order_details**” has details about the order with 4 columns of data.
- “**orders**’ has order information about date and time of the order.
- “**pizza_types**” has details about the type of pizza that was ordered.
- “**pizzas**” has information about size and price of the pizza.



Tables

Order_details

Field	Type	Null	Key	Default	Extra
order_details_id	int	NO	PRI	NULL	
order_id	int	NO		NULL	
pizza_id	text	NO		NULL	
quantity	int	NO		NULL	

Pizzas

Field	Type	Null	Key	Default	Extra
pizza_id	text	YES		NULL	
pizza_type_id	text	YES		NULL	
size	text	YES		NULL	
price	double	YES		NULL	

Orders

Field	Type	Null	Key	Default	Extra
order_id	int	NO	PRI	NULL	
order_date	datetime	NO		NULL	
order_time	time	NO		NULL	

pizza_types

Field	Type	Null	Key	Default	Extra
pizza_type_id	text	YES		NULL	
name	text	YES		NULL	
category	text	YES		NULL	
ingredients	text	YES		NULL	

Retrieve the total number of orders placed.

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

total_orders
21350

Calculate the total revenue generated from pizza sales

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

total_price
817860.05

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

Identify the most common pizza size ordered

```sql
SELECT
    pizzas.size AS 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
LIMIT 1;
```

size	order_count
L	18526

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

```sql
SELECT
    pizza_types.name,
    SUM(order_details.quantity) AS quantity_ordered
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_ordered DESC
LIMIT 5;
```

name	quantity_order...
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 pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY 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

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

hour	order_count
12	2520
13	2455
18	2399
17	2336
19	2009
16	1920
20	1642
14	1472
15	1468
11	1231
21	1198
22	663
23	28
10	8
9	1

Find the category-wise distribution of pizzas

```
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(sum_quantity)) AS avg_daily_orders
FROM
    (SELECT
        orders.order_date AS date,
            SUM(order_details.quantity) AS sum_quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY date) AS order_quantity;
```

avg_daily_orders
138

Determine the top 3 most ordered pizza types based on revenue

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

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Analyze the cumulative revenue generated over time

```sql
SELECT order_date,
       SUM(revenue) OVER (order by order_date) as cum_revenue
FROM
        (SELECT
                date(orders.order_date) as 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 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
2015-01-12	27781.7
2015-01-13	29831.300000000003
2015-01-14	32358.700000000004
2015-01-15	34343.50000000001
2015-01-16	36937.65000000001
2015-01-17	39001.75000000001
2015-01-18	40978.600000000006
2015-01-19	43365.75000000001
2015-01-20	45763.65000000001
2015-01-21	47804.20000000001
2015-01-22	50300.90000000001
2015-01-23	52724.600000000006
2015-01-24	55013.850000000006
2015-01-25	56631.40000000001
2015-01-26	58515.80000000001
2015-01-27	61043.85000000001
2015-01-28	63059.85000000001
2015-01-29	65105.150000000016
2015-01-30	67375.45000000001
2015-01-31	69793.30000000002

Determine the top 3 most ordered pizza types based on revenue for each pizza category

```sql
SELECT name, category, 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,
                        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 category, name) AS a) AS b
WHERE rn <=3;
```

name	category	revenue
The Thai Chicken Pizza	Chicken	43434.25
The Barbecue Chicken Pizza	Chicken	42768
The California Chicken Pizza	Chicken	41409.5
The Classic Deluxe Pizza	Classic	38180.5
The Hawaiian Pizza	Classic	32273.25
The Pepperoni Pizza	Classic	30161.75
The Spicy Italian Pizza	Supreme	34831.25
The Italian Supreme Pizza	Supreme	33476.75
The Sicilian Pizza	Supreme	30940.5
The Four Cheese Pizza	Veggie	32265.70000000065
The Mexicana Pizza	Veggie	26780.75
The Five Cheese Pizza	Veggie	26066.5