1.Which days of the week generate the most orders and revenue

```
use pizzahut;
SELECT
    DAYNAME(orders.order_date) AS weekday,
    COUNT(DISTINCT orders.order_id) AS total_orders,
    SUM(order_details.quantity * pizzas.price) AS total_revenue
FROM
    orders
        JOIN
    order_details ON orders.order_id = order_details.order_id
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
GROUP BY DAYNAME(orders.order_date);
```

Output

	weekday	total_orders	total_revenue
▶	Friday	3538	136073.8999999994
	Monday	2794	107329.54999999965
	Saturday	3158	123182.39999999946
	Sunday	2624	99203.49999999964
	Thursday	3239	123528.49999999939
	Tuesday	2973	114133.79999999948
	Wednesday	3024	114408.39999999951

2. How does our sales performance vary month to month

```
SELECT
    MONTHNAME(orders.order_date) AS month,
    COUNT(DISTINCT orders.order_id) AS total_orders,
    SUM(order_details.quantity * pizzas.price) AS total_revenue
FROM
    orders
        JOIN
    order_details ON orders.order_id = order_details.order_id
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
GROUP BY MONTHNAME(orders.order_date) , MONTH(orders.order_date)
ORDER BY MONTH(orders.order_date);
```

Output

month	total_orders	total_revenue
January	1845	69793.29999999992
February	1685	65159.59999999991
March	1840	70397.09999999989
April	1799	68736.79999999992
May	1853	71402.7499999999
June	1773	68230.1999999999
July	1935	72557.89999999983
August	1841	68278.24999999994
September	1661	64180.04999999997
October	1646	64027.59999999989
November	1792	70395.34999999989
December	1680	64701.149999999885

3. Calculate how much customers spend on average per order

```
SELECT
    COUNT(DISTINCT orders.order_id) AS total_orders,
    SUM(order_details.quantity * pizzas.price) AS total_revenue,
    ROUND(SUM(order_details.quantity * pizzas.price) / COUNT(DISTINCT orders.order_id),
            2) AS average_order_value
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
        JOIN
    orders ON orders.order_id = order_details.order_id;
```

Output

total_orders	total_revenue	average_order_value
21350	817860.049999993	38.31

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

```
select name,revenue from
(select category,name,revenue,
rank() over(partition by category order by revenue desc) as rn from
(select pizza_types.name,pizza_types.category,sum(order_details.quantity*pizzas.price)
as revenue from order_details join pizzas on order_details.pizza_id=pizzas.pizza_id
join pizza_types on pizza_types.pizza_type_id=pizzas.pizza_type_id group by
pizza_types.name,pizza_types.category) as a)as b where rn<=3;
```

Output

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
The Italian Supreme Pizza	33476.75
The Sicilian Pizza	30940.5
The Four Cheese Pizza	32265.70000000065
The Mexicana Pizza	26780.75
The Five Cheese Pizza	26066.5

5.Analyse the cumulative revenue generated over time

```
select order_date,
sum(revenue) over( order by order_date) as cumulative_revenue from
(select 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 orders.order_date) as sales;
```

Output

order_date	cumulative_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

6. Calculate ulate 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
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
        JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
GROUP BY pizza_types.category;
```

Output

category	revenue
Classic	26.91
Veggie	23.68
Supreme	25.46
Chicken	23.96

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

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

Output

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

8.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 order_quantity;
```

Output

ROUND(AVG(quantity), 0)
138

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

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

Output

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

10. Retrieve the total no of orders placed

```
select count(order_id) as total_orders from orders;
```

Output

total_orders
21350

11.Calculate the total revenue generated from pizza sales

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

Output

total_revenue
817860.05

12.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;
```

Output

name	price
The Greek Pizza	35.95

13.Identify the most common pizza size ordered

Output

```
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
LIMIT 1;
```

size	order_count
L	18526

14.List the top 5 most ordered pizza types 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;
```

Output

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

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

```
SELECT
    COUNT(order_details.quantity), pizza_types.category
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
ORDER BY category;
```

Output

COUNT(order_details.quantity)	category
10815	Chicken
14579	Classic
11777	Supreme
11449	Veggie

16. Determine the distribution of orders by hour of the day

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

Output

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