Pizza Report

Using SQL



Retrieve the total number of order placed.

select count(order_id) as total_order from orders;



Calculate the total revenue generated from pizza sales.

```
SELECT

ROUND(SUM(orders_detail.quantity * pizzas.price),

2) AS total_sales

FROM

orders_detail

JOIN

pizzas ON pizzas.pizza_id = orders_detail.pizza_id;
```

Result Grid

total_sales 817860.05

Identify the highest-price pizza.

price

35.95

name

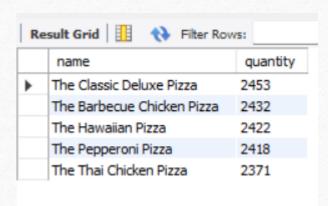
The Greek Pizza

Identify the most common pizza sized ordered.

Re	sult Grid	I │ Filter Ro
	size	order_count
•	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28

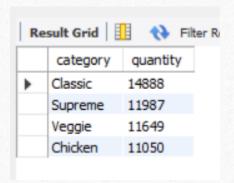
List the top 5 most ordered pizza type. Along with their quantities.

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



Join the necessary table to find the. Total quantity of each pizza category order

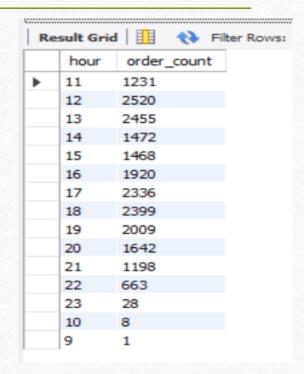
```
SELECT
    pizza_types.category,
    SUM(orders_detail.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_detail ON orders_detail.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity DESC;
```



Determine the distribution of orders by hour of the day.

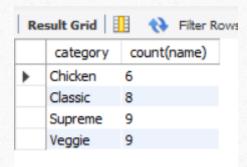
select

hour(order_time) as hour, count(order_id) as order_count from orders
group by hour(order_time);



Join relevant table to find the. Category-wise distribution of pizzas.

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



Group the orders by date and calculate the average alter. Number of pizza ordered per day.

```
SELECT

ROUND(AVG(quantity), 0)

FROM

(SELECT

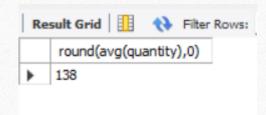
orders.order_date, SUM(orders_detail.quantity) AS quantity

FROM

orders

JOIN orders_detail ON orders.order_id = orders_detail.order_id

GROUP BY orders.order_date) AS order_quantity;
```



Determine the top 3 most ordered. Pizza type based on revenue.

```
select pizza_types.name,
sum(orders_detail.quantity * pizzas.price) as revenue
from pizza_types join pizzas
on pizzas.pizza_type_id =pizza_types.pizza_type_id
join orders_detail
on orders_detail.pizza_id = pizzas.pizza_id
group by pizza_types.name order by revenue desc limit 3;
```



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

```
SELECT
    pizza_types.category,
  round (SUM(orders_detail.quantity * pizzas.price) / (SELECT
            ROUND(SUM(orders_detail.quantity * pizzas.price),
                        2) AS total sales
        FROM
           orders_detail
                JOIN
            pizzas ON pizzas.pizza_id = orders_detail.pizza_id) * 100,2) AS revenue
FROM
    pizza types
        JOIN
    pizzas ON pizza types.pizza type id = pizzas.pizza type id
        JOIN
    orders_detail ON orders_detail.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY revenue DESC;
```



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(orders_detail.quantity * pizzas.price) as revenue
from orders_detail join pizzas
on orders_detail.pizza_id=pizzas.pizza_id
join orders
on orders.order_id = orders_detail.order_id
group by orders.order_date) as sales limit 5;
```

Re	esult 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

Determine the top 3 most ordered pizza type. Based on revenue for each pizza category.

```
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((orders_detail.quantity) * pizzas.price) as revenue
  from pizza_types join pizzas
  on pizza_types.pizza_type_id = pizzas.pizza_type_id
  join orders_detail
  on orders_detail.pizza_id = pizzas.pizza_id
  group by pizza_types.category,pizza_types.name) as a )as b
  where rn <= 3;</pre>
```

	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	