Pizza Sales –sql

1) Join relavant tables to 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

2) Identify the highest priced pizza?

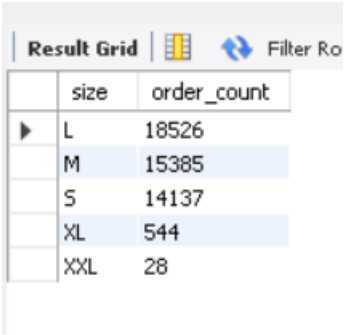
```
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

Result Grid | Filter Rows:

name	price
The Greek Pizza	35.95

3) Identify the most common pizza size ordered

```
SELECT
    pizzas.size,
    COUNT(orders_details.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC;
```



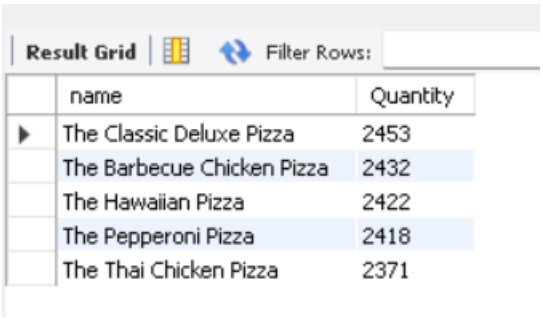
Result Grid | Filter Ro

size	order_count
L	18526
M	15385
S	14137
XL	544
XXL	28

4)List the top 5 most ordered pizzas types among with their quantities

```
SELECT
    pizza_types.name, SUM(orders_details.quantity) AS Quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
```

```
GROUP BY pizza_types.name
ORDER BY Quantity DESC
LIMIT 5;
```

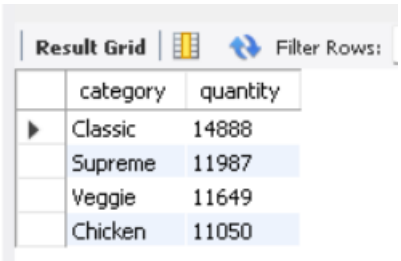


Result Grid | Filter Rows:

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

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

```
select pizza_types.category,
sum(orders_details.quantity) as quantity
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join orders_details
on orders_details.pizza_id = pizzas.pizza_id
group  by pizza_types.category order by quantity DESC ;
```

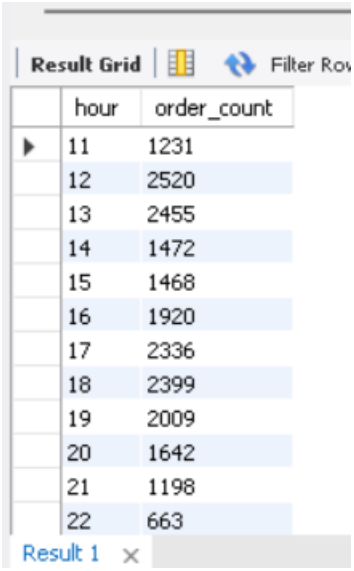


Result Grid | Filter Rows:

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

6)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);
```



Result Grid | Filter Rov

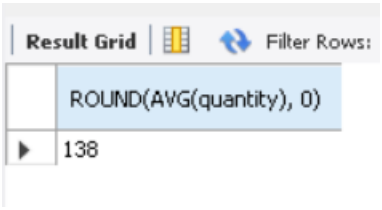
hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663

Result 1

<u>7) Group the orders by date and calculate the average number of pizzas ordered per day</u>

```
SELECT
    ROUND(AVG(quantity), 0)
FROM
    (SELECT
        orders.order_date, SUM(orders_details.quantity) AS quantity
    FROM
        orders
    JOIN orders_details ON orders.order_id = orders_details.order_id
```

GROUP BY orders.order_date) AS order_quantity;

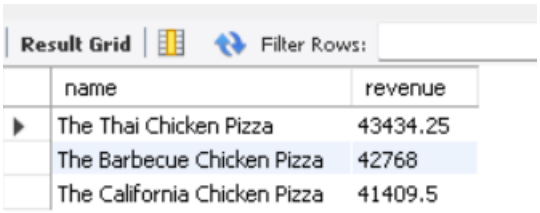


Result Grid | Filter Rows:

	ROUND(AVG(quantity), 0)
▶	138

8) Determine the top 3 most ordered pizzas type based on revenue

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



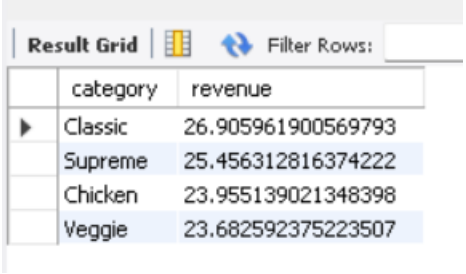
Result Grid | Filter Rows:

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

9)Calculate the percentage contribution of each pizza type to total revenue

```
SELECT
    pizza_types.category,
    (SUM(pizzas.price * orders_details.quantity) / (SELECT
        ROUND(SUM(orders_details.quantity * pizzas.price),
                0) AS total_sales
```

```
        FROM
            orders_details
                JOIN
            pizzas ON pizzas.pizza_id = orders_details.pizza_id)) * 100 AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY revenue DESC;
```



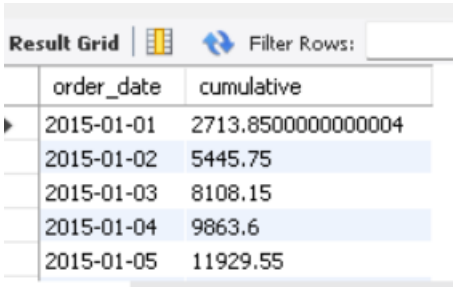
Result Grid | Filter Rows:

category	revenue
Classic	26.905961900569793
Supreme	25.456312816374222
Chicken	23.955139021348398
Veggie	23.682592375223507

10) analyze the cumulative revenue generated over time

```
select order_date,
sum(revenue) over (order by order_date) as cumulative
from
(select orders.order_date,
sum(pizzas.price * orders_details.quantity) as revenue
from orders_details join pizzas
on pizzas.pizza_id = orders_details.pizza_id
join orders
on orders.order_id = orders_details.order_id
```

group by orders.order_date)as sales ;



Result Grid | Filter Rows:

order_date	cumulative
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