SQL Project - Pizza Sales

This SQL project analyzes pizza sales data to uncover key business insights. It includes calculating total orders, revenue, and identifying popular pizzas by size, category, and time. Advanced queries highlight revenue trends and top-performing items, using data aggregation and table joins for in-depth analysis.

Retrieve the total number of orders placed.

```
1  -- Retrieve the total number order placed
2
3 • Select count(order_id) as total_orders
4  from orders;
```



Calculate the total revenue generated from pizza sales.alter

```
SELECT

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

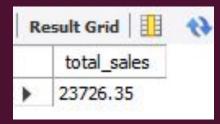
2) AS total_sales

FROM

orders_details

JOIN

pizzas ON pizzas.pizza_id = orders_details.pizza_id
```



Identify the highest priced pizza.



Identify the most common pizza size

ordered.



List the top 5 most ordered pizza types along 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;
```

| R | esult Grid 🔠 🙌 Filter | Rows: | |
|---|---|----------|--|
| | name | quantity | |
| • | The Pepperoni Pizza | 82 | |
| | The Classic Deluxe Piz The Pepperoni Pizz | | |
| | The Thai Chicken Pizza | 70 | |
| | The Hawaiian Pizza | 66 | |
| | The Italian Supreme Pizza | 65 | |

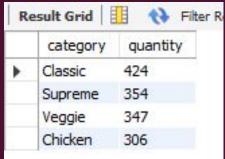
Join the necessary tables to find the total quantity of each pizza category ordered.

```
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;
```



Determine the distribution of orders by hour of the day.

```
HOUR(order_time) AS hour,

COUNT(order_id) AS number_of_orders

FROM

orders

GROUP BY HOUR(order_time);
```

| Result Grid (1) Filter Row | | | |
|--------------------------------|------|------------------|--|
| | hour | number_of_orders | |
| • | 11 | 1231 | |
| | 12 | 2520 | |
| | 13 | 2455 | |
| | 14 | 1472 | |
| | 15 | 1468 | |
| | 16 | 1920 | |
| | 17 | 2226 | |

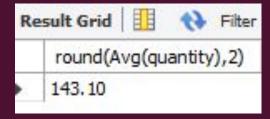
Join relevant tables to find the category wise distribution of pizzas.

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

| R | esult Grid | No Filter Row |
|---|------------|---------------|
| | 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.

```
Select round(Avg(quantity),2) 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;
```



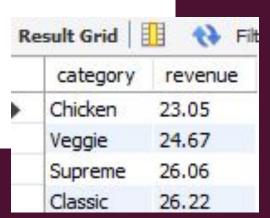
Determine the top 3 most ordered pizza types based on revenue.

```
SELECT
    pizza types.name,
    SUM(orders_details.quantity * pizzas.price) AS revenue
FROM
    pizza types
        DOIN
    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 revenue DESC
LIMIT 3;
```

| R | esult Grid 🔢 🙌 Filter Ro | ws: | |
|---|----------------------------|---------|--|
| | name | revenue | |
| • | The Thai Chicken Pizza | 1296.5 | |
| | The Barbecue Chicken Pizza | 1160.75 | |
| | The Italian Supreme Pizza | 1158.75 | |
| | The Italian Supreme Fizza | 1136.73 | |

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

```
SELECT
    pizza types.category,
    ROUND(SUM(orders_details.quantity * pizzas.price) / (SELECT
                    ROUND(SUM(orders details.quantity * pizzas.price),
                                2) AS total sales
                FROM
                    orders details
                        TOTAL
                    pizzas ON orders details.pizza id = pizzas.pizza id) * 100, 2) AS revenue
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 revenue;
```



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 details.quantity*pizzas.price) as revenue
from orders details join pizzas
                                                        tesult Grid
                                                                   Filter Rows:
on orders details.pizza id = pizzas.pizza id
                                                          order date
                                                                   cum revenue
                                                          2015-01-01
                                                                   2713.85000000000004
join orders
                                                          2015-01-02 5445.75
on orders.order id = orders details.order id
                                                          2015-01-03
                                                                  8108, 15
                                                          2015-01-04 9863.6
group by orders.order_date) as sales;
                                                          2015-01-05 11929.55
                                                          2015-01-06 14358.5
                                                          2015-01-07
                                                                   16560.7
                                                          2015-01-08 19399.05
```

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

```
Select name, revenue, rank pizza types
from(
Select category, name, revenue,
rank() over(partition by category order by revenue desc) as rank pizza types
from(
Select pizza types.category, pizza types.name,
sum(orders details.quantity*pizzas.price) as revenue
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 , pizza_types.name) as a) as b
where rank pizza types<=3;
```

| R | esult Grid 🎚 🙌 Filter Ro | ws: | |
|---|------------------------------|---------|-----------|
| | name | revenue | rank_pizz |
| | The Thai Chicken Pizza | 1296.5 | 1 |
| | The Barbecue Chicken Pizza | 1160.75 | 2 |
| | The California Chicken Pizza | 1115.25 | 3 |
| | The Classic Deluxe Pizza | 1100 | 1 |
| | The Pepperoni Pizza | 1038.75 | 2 |
| | The Greek Pizza | 899 | 3 |
| | 71 11 1 0 0° | 4450 75 | 121 |