



THE SQL SLICE

# CRUNCHING PIZZA SALES DATA





# SAHIL SANDIP SHINDE: ASPIRING DATA ANALYST

SAHIL SANDIP SHINDE, A MECHANICAL ENGINEERING STUDENT, IS LEVERAGING HIS SKILLS IN PYTHON, SQL, AND EXCEL TO PURSUE A CAREER IN DATA ANALYTICS. HIS CURRENT PROJECT ON PIZZA SALES ANALYSIS DEMONSTRATES HIS GROWING EXPERTISE IN DATABASE MANAGEMENT AND INSIGHT EXTRACTION.

# RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```
SELECT count(order_id) FROM orders;
```

| Result Grid |                 |
|-------------|-----------------|
|             | count(order_id) |
| ▶           | 21350           |





# CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

SELECT

```
ROUND(SUM(orders_details.quantity * pizzas.price),  
2) AS total_sales
```

FROM

```
orders_details
```

JOIN

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

Result Grid



Filter Rows:

| total_sales |
|-------------|
| 817860.05   |

# 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;
```

|   | name            | price |
|---|-----------------|-------|
| ▶ | The Greek Pizza | 35.95 |

# 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 |      |             |
|-------------|------|-------------|
|             | size | order_count |
| ▶           | L    | 18526       |
|             | M    | 15385       |
|             | S    | 14137       |
|             | XL   | 544         |
|             | XXL  | 28          |



LIST THE TOP  
5 MOST  
ORDERED  
PIZZA TYPES  
ALONG WITH  
THEIR  
QUANTITIES.

```
SELECT pizza_types.name,  
       SUM(orders_details.quantity) AS Total_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 Total_Quantity DESC  
 LIMIT 5;
```

Result Grid | Filter Rows:

|   | name                       | Total_Quantity |
|---|----------------------------|----------------|
| ▶ | 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.

```
SELECT
    pizza_types.category,
    SUM(orders_details.quantity) AS Total_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;
```

|   | category | Total_Quantity |
|---|----------|----------------|
| ▶ | Classic  | 14888          |
|   | Veggie   | 11649          |
|   | Supreme  | 11987          |
|   | Chicken  | 11050          |



# 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;
```

| Result Grid |      |             |
|-------------|------|-------------|
|             | Hour | order_count |
| ▶           | 11   | 1231        |
|             | 12   | 2520        |
|             | 13   | 2455        |
|             | 14   | 1472        |
|             | 15   | 1468        |
|             | 16   | 1920        |
|             | 17   | 2336        |
|             | 18   | 2399        |
|             | 19   | 2009        |
|             | 20   | 1642        |

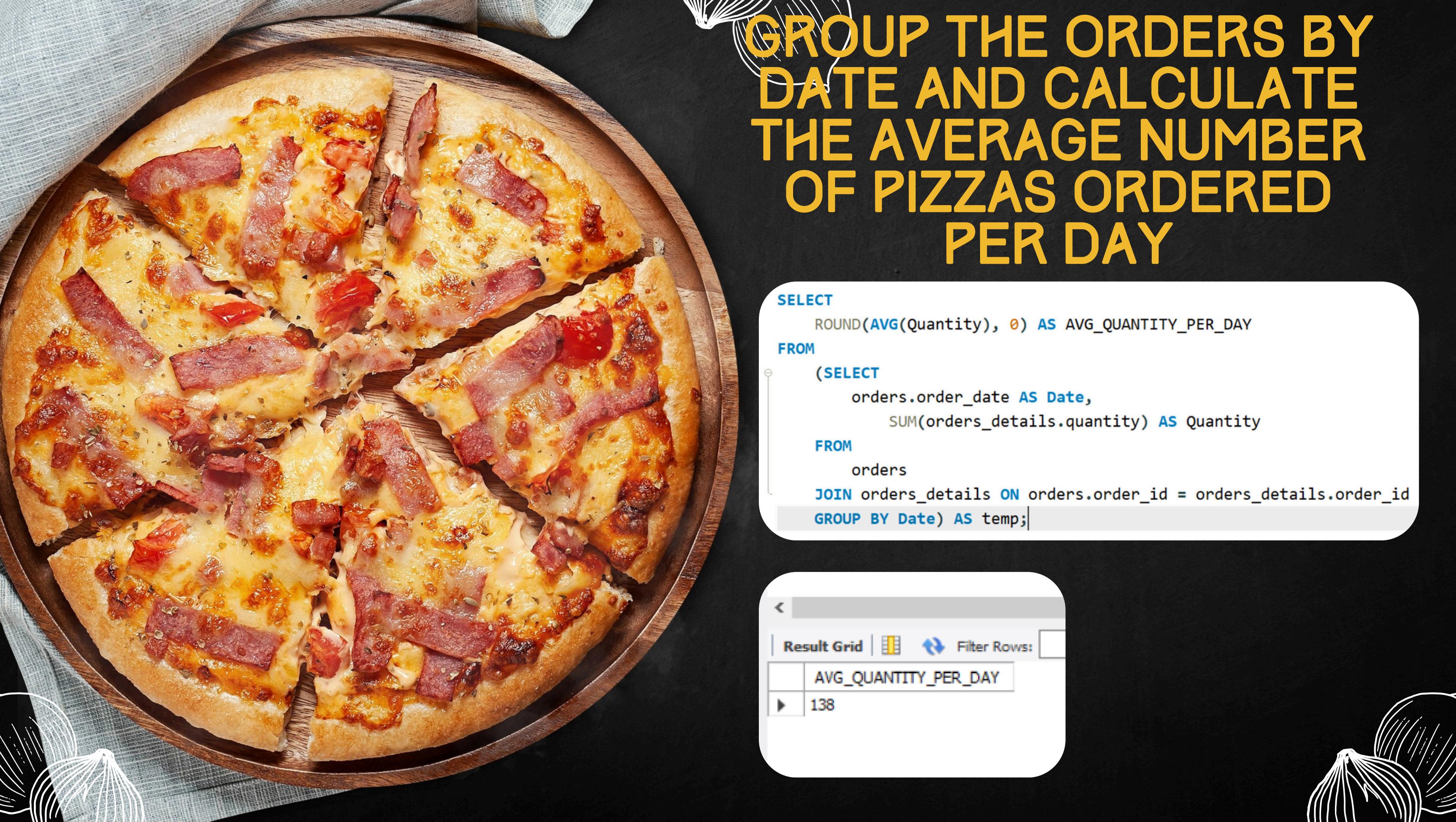
# JOIN RELEVANT TABLES TO FIND THE CATEGORY- WISE DISTRIBUTION OF PIZZAS

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

Result Grid | Filter Rows:

|   | 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), 0) AS AVG_QUANTITY_PER_DAY  
FROM  
    (SELECT  
        orders.order_date AS Date,  
        SUM(orders_details.quantity) AS Quantity  
    FROM  
        orders  
    JOIN orders_details ON orders.order_id = orders_details.order_id  
    GROUP BY Date) AS temp;
```

Result Grid | Filter Rows:

| AVG_QUANTITY_PER_DAY |
|----------------------|
| 138                  |

# DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE

```
SELECT
    pizza_types.name,
    SUM(orders_details.quantity * pizzas.price) AS revenue
FROM
    orders_details
        JOIN
    pizzas ON orders_details.pizza_id = pizzas.pizza_id
        JOIN
    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
GROUP BY pizza_types.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

```
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
    JOIN
        pizzas ON orders_details.pizza_id = pizzas.pizza_id) * 100,
    2) AS percentage_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;
```

|   | category | percentage_revenue |
|---|----------|--------------------|
| ▶ | Classic  | 26.91              |
|   | Veggie   | 23.68              |
|   | Supreme  | 25.46              |
|   | Chicken  | 23.96              |

# 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,  
round(sum(orders_details.quantity * pizzas.price),2) as revenue  
from orders_details join pizzas  
on orders_details.pizza_id = pizzas.pizza_id  
join orders  
on orders.order_id = orders_details.order_id  
group by orders.order_date) as sales;
```

| order_date | cum_revenue        |
|------------|--------------------|
| 2015-01-28 | 63059.850000000006 |
| 2015-01-29 | 65105.15000000001  |
| 2015-01-30 | 67375.45000000001  |
| 2015-01-31 | 69793.30000000002  |
| 2015-02-01 | 72982.50000000001  |
| 2015-02-02 | 75311.10000000002  |
| 2015-02-03 | 77925.90000000002  |



# DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY

```
select Name, Revenue, Category
from
(select Category, Name, Revenue,
rank() over(partition by Category order by revenue desc) as RNK
from
(select pizza_types.category, pizza_types.name,
round(sum(orders_details.quantity * pizzas.price),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, pizza_types.name) as a) as b
where RNK <= 3;
```

|   | Name                         | Revenue  | Category |
|---|------------------------------|----------|----------|
| ▶ | The Thai Chicken Pizza       | 43434.25 | Chicken  |
|   | The Barbecue Chicken Pizza   | 42768    | Chicken  |
|   | The California Chicken Pizza | 41409.5  | Chicken  |
|   | The Classic Deluxe Pizza     | 38180.5  | Classic  |
|   | The Hawaiian Pizza           | 32273.25 | Classic  |
|   | The Pepperoni Pizza          | 30161.75 | Classic  |
|   | The Spicy Italian Pizza      | 34831.25 | Supreme  |

# DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY

```
select Name, Revenue, Category
from
  (select Category, Name, Revenue,
  rank() over(partition by Category order by revenue desc) as RNK
  from
    (select pizza_types.category, pizza_types.name,
    round(sum(orders_details.quantity * pizzas.price),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, pizza_types.name) as a) as b
where RNK <= 3;
```

|   | Name                         | Revenue  | Category |
|---|------------------------------|----------|----------|
| ▶ | The Thai Chicken Pizza       | 43434.25 | Chicken  |
|   | The Barbecue Chicken Pizza   | 42768    | Chicken  |
|   | The California Chicken Pizza | 41409.5  | Chicken  |
|   | The Classic Deluxe Pizza     | 38180.5  | Classic  |
|   | The Hawaiian Pizza           | 32273.25 | Classic  |
|   | The Pepperoni Pizza          | 30161.75 | Classic  |
|   | The Spicy Italian Pizza      | 34831.25 | Supreme  |

# THANK YOU!

