# PIZZA SALES ANALYSIS

# HELLO!

MY NAME IS ASHISH KUMAR. THROUGHOUT THIS PROJECT, I HAVE UTILIZED A SQL QUERIES TO ADDRESS INQUIRIES PERTAINING TO PIZZA SALES.

# OBJECTIVE

ANALYZE PIZZA SALES DATA TO GAIN INSIGHTS INTO CUSTOMER BEHAVIOR AND REVENUE GENERATION.

# KEY PERFORMANCE INDICATORS (KPIS):

1. **TOTAL REVENUE:** CALCULATE THE OVERALL REVENUE GENERATED FROM PIZZA SALES.

2. **HIGHEST PRICED PIZZA:** IDENTIFY THE PIZZA WITH THE HIGHEST PRICE POINT.

3. **MOST COMMON PIZZA SIZE:** DETERMINE THE MOST FREQUENTLY ORDERED PIZZA SIZE.

4. **ORDER DISTRIBUTION BY HOUR:** ANALYZE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

**5. AVERAGE DAILY ORDERS:** CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.

**6. REVENUE CONTRIBUTION BY PIZZA TYPE:** DETERMINE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

**7. CUMULATIVE REVENUE OVER TIME:** TRACK THE CUMULATIVE REVENUE GENERATED OVER THE PROJECT DURATION.

**8. REVENUE BY PIZZA CATEGORY:** CATEGORIZE PIZZA TYPES BY REVENUE GENERATED FOR EACH CATEGORY.

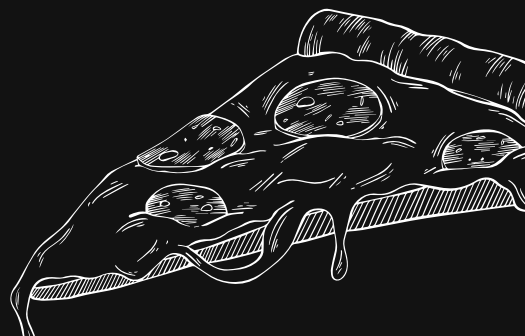
# QUESTION NO. 1

## RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED??

```sql
SELECT COUNT(order_id) AS total_orders
FROM orders;
```

Result Grid

| | total_orders |
|---|---|
| ▶ | 21350 |



# QUESTION NO. 2

## CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES??

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

Result Grid

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

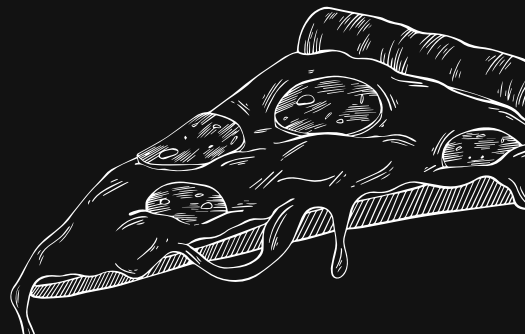
# QUESTION NO. 3

## 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;
```

Result Grid | Filter Ro

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



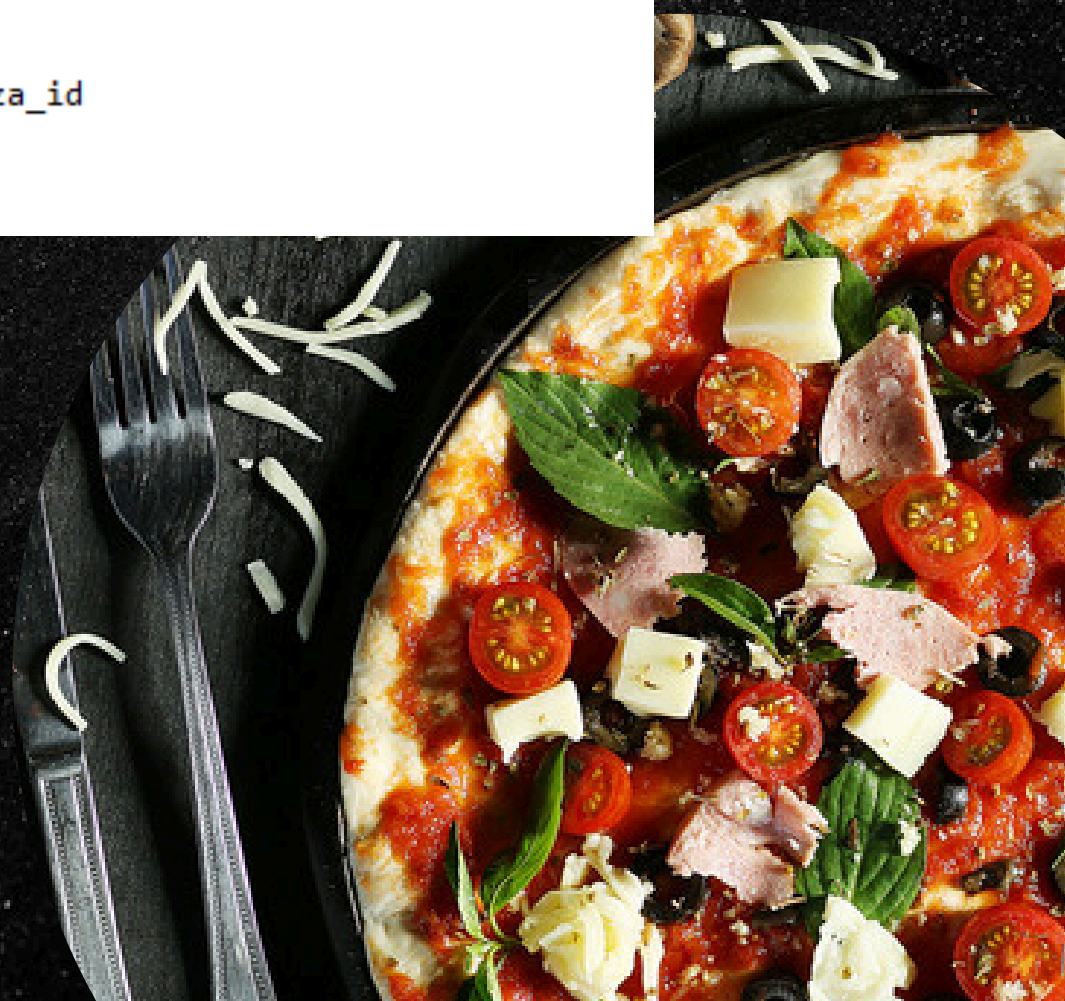
# QUESTION NO. 4

## IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED??

```sql
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;
```

Result Grid | Filte

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



# QUESTION NO. 5

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

```sql
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;
```

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 |

# QUESTION NO. 6

## JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED??

```
SELECT pizza_types.category, 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.category
ORDER BY quantity DESC;
```

Result Grid | Filte

| category | quantity |
|---|---|
| Classic | 14888 |
| Supreme | 11987 |
| Veggie | 11649 |
| Chicken | 11050 |

# QUESTION NO. 7

## 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 | Filt

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

Result 7

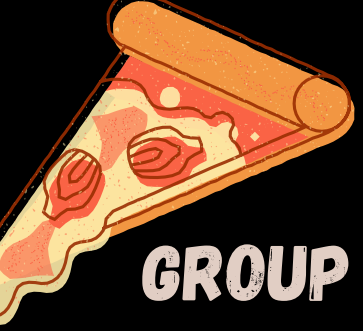
# QUESTION NO. 8

## 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 |



# QUESTION NO. 9

## GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY??

```sql
SELECT ROUND(AVG(quantity), 0) AS avg_pizza_ordered_per_day
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;
```

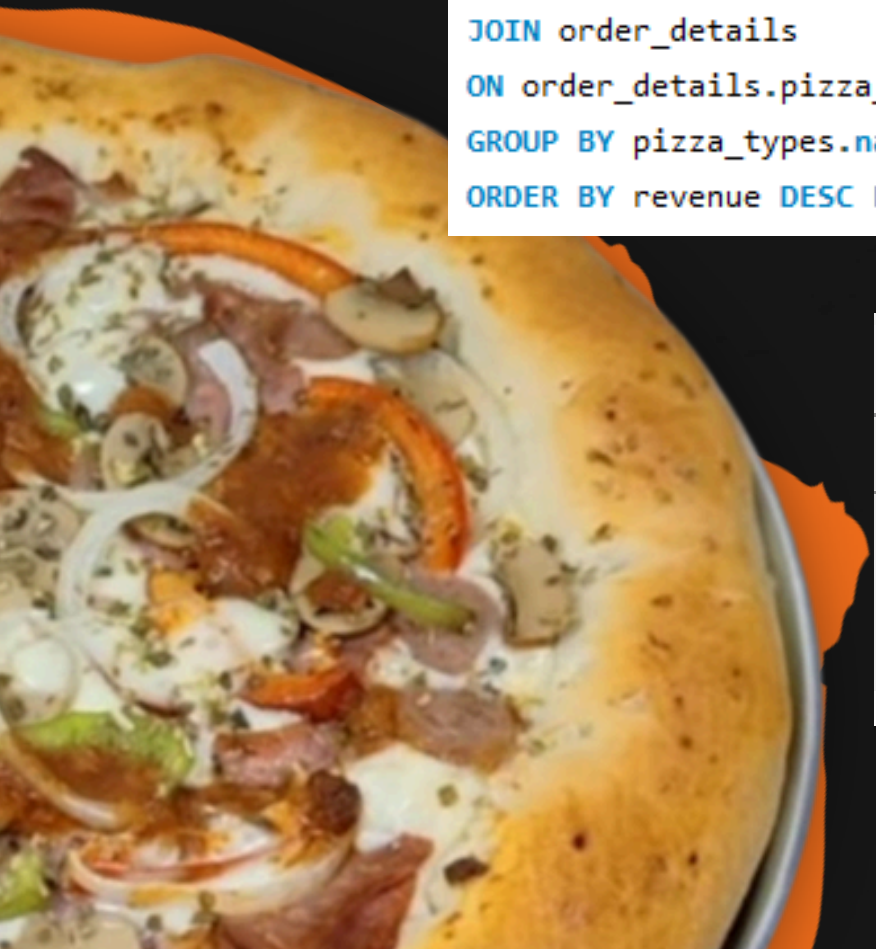
Result Grid | Filter Rows:

| | avg_pizza_ordered_per_day |
|---|---|
| ▶ | 138 |

# QUESTION NO. 10

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

```
SELECT pizza_types.name, SUM(order_details.quantity*pizzas.price) AS revenue
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 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 |

# QUESTION NO. 11

## CALCULATE 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 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 revenue DESC;
```

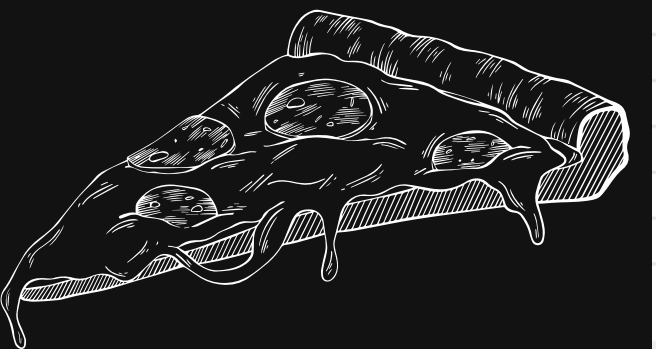
Result Grid | Filte

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

# QUESTION NO. 12

## ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME??

```
SELECT order_date, SUM(revenue) OVER(ORDER BY order_date) AS cumulative_revenue
FROM
(SELECT orders.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;
```



Result Grid  Filter Rows:

| 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 |

Result 2

# QUESTION NO. 13

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

```sql
SELECT name, revenue
FROM
(SELECT category, name, revenue,
RANK() OVER(PARTITION BY category ORDER BY revenue DESC) AS rnk
FROM
(SELECT pizza_types.category, pizza_types.name,
SUM((order_details.quantity) * pizzas.price) AS revenue
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, pizza_types.name) AS a) AS b
WHERE rnk <=3;
```

Result Grid | Filter Rows:

| 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 |

Result 3

# THANK YOU SO MUCH!

*thank you so much*