

Where Every Slice is a Taste of Perfection

PIZZA SALES ANALYSIS

By - Siddhesh
Deshmukh

ORDER
NOW





THE TOTAL NUMBER OF ORDERS PLACED.

QUERY:-

```
select count(order_id) as order_placed from orders;
```



Result Grid	
	order_placed
▶	21350



TOTAL REVENUE GENERATED FROM PIZZA SALES.

QUERY:-

```
Select round(sum(order_details.quantity * pizzas.price),2) as Total_revenue  
from order_details join pizzas  
ON order_details.pizza_id=pizzas.pizza_id;
```

The screenshot shows a terminal window with a MySQL prompt. The user has run a query to calculate total revenue from pizza sales. The output displays the column name 'Total_revenue' and its corresponding value '817860.05'.

Total_revenue
817860.05

HIGHEST-PRICED PIZZA.



QUERY:-

```
select pizza_types.name as Pizza_name,pizzas.price as costly_pizza  
from pizza_types join pizzas  
on pizza_types.pizza_type_id=pizzas.pizza_type_id  
order by costly_pizza DESC limit 1;
```

	Pizza_name	costly_pizza
→	The Greek Pizza	35.95



MOST COMMON PIZZA SIZE ORDERED.



QUERY:-

```
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS count_of_pizza
FROM
    pizzas
    JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY count_of_pizza DESC
LIMIT 1;
```

	size	count_of_pizza
▶	L	18526

TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

QUERY:-

```
SELECT
    pizza_types.name, SUM(order_details.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 SUM(order_details.quantity) DESC
LIMIT 5;
```

name	sum(order_details.quantity)
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

QUERY:-

SELECT

```
pizza_types.category AS Category,  
SUM(order_details.quantity) AS quantity
```

FROM

pizzas

JOIN

```
pizza_types ON pizzas.pizza_type_id =
```

```
pizza_types.pizza_type_id
```

JOIN

```
order_details ON pizzas.pizza_id = order_details.pizza_id
```

```
GROUP BY Category;
```

Category	quantity
Classic	14888
Veggie	11649
Supreme	11987
Chicken	11050

TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.



QUERY:-

SELECT

```
    pizza_types.name AS pizza_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 pizzas.pizza_id = order_details.pizza_id
```

GROUP BY pizza_name

ORDER BY revenue DESC

LIMIT 3;

pizza_name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

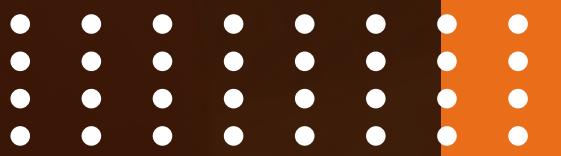


QUERY:-

```
SELECT
    HOUR(order_time), COUNT(order_id)
FROM
    orders
GROUP BY HOUR(order_time);
```

hour(order_time)	count(order_id)
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009

FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.



QUERY:-

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

category	sum(order_details.quantity)
Classic	14888
Veggie	11649
Supreme	11987
Chicken	11050

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

QUERY:-

```
SELECT
    ROUND(AVG(qunatity), 0) AS avg_pizza_perday
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) AS
qunatity
    FROM
        orders
    JOIN order_details ON orders.order_id =
order_details.order_id
    GROUP BY orders.order_date) AS order_quantity;
```

avg_pizza_perday
138



PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

QUERY:-

```
SELECT
    pizza_types.category AS Category,
    ROUND((SUM(pizzas.price * order_details.quantity) /
(SELECT
    ROUND(SUM(order_details.quantity *
pizzas.price),
        2) AS Total_revenue
FROM
    order_details
    JOIN
    pizzas ON order_details.pizza_id =
pizzas.pizza_id)),
        2) * 100 AS percentage_contribute
FROM
    pizza_types
    JOIN
    pizzas ON pizza_types.pizza_type_id =
pizzas.pizza_type_id
    JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY Category;
```

Category	percentage_contribute
Classic	27
Veggie	24
Supreme	25
Chicken	24

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.



QUERY:-

```
SELECT
    dates,
    ROUND(SUM(revenue) OVER (ORDER BY dates), 2) AS
cumulative_revenue
FROM (
    SELECT
        orders.order_date AS dates,
        ROUND(SUM(pizzas.price * order_details.quantity), 2)
    AS revenue
    FROM pizza_types
    JOIN pizzas
        ON pizza_types.pizza_type_id = pizzas.pizza_type_id
    JOIN order_details
        ON pizzas.pizza_id = order_details.pizza_id
    JOIN orders
        ON order_details.order_id = orders.order_id
    GROUP BY orders.order_date
) AS daily_revenue;
```

dates	cumulative_revenue
2015-01-01	2713.85
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



<https://www.linkedin.com/in/siddhesh-deshmukh-046455229/>

THANK YOU

