

FAUGET PIZZA



+123-456-7890



www.reallygreatsite.com

50%
DISCOUNT



HELLO!

I AM RAHUL KUMAR, AND I HAVE DEVELOPED THE PIZZA SALES MANAGEMENT SYSTEM TO ADDRESS VARIOUS QUESTIONS RELATED TO PIZZA SALES USING SQL QUERIES. THIS PROJECT INVOLVES CREATING A ROBUST BACK-END SYSTEM TO MANAGE CUSTOMER INFORMATION, ORDERS, PIZZAS, TOPPINGS, AND OVERALL SALES DATA. THE SYSTEM DEMONSTRATES MY ABILITY TO DESIGN EFFICIENT DATABASES, OPTIMIZE QUERY PERFORMANCE, AND AUTOMATE DATA HANDLING PROCESSES.

-- RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```
SELECT  
    COUNT(order_id) AS totel_oders  
FROM  
    orders;
```

Result Grid	
	totel_oders
▶	21350

-- CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

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

Result Grid	
	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;
```

Result Grid		Filter Row
	name	price
▶	The Greek Pizza	35.95

-- IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
• SELECT
    pizzas.size,
    COUNT(oder_details.oder_details_id) AS order_count
FROM
    pizzas
    JOIN
    oder_details ON pizzas.pizza_id = oder_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

-- IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

- **SELECT**

```
    pizza_types.name, SUM(oder_details.quantity) AS quantity
  FROM
    pizza_types
    JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
    JOIN
    oder_details ON oder_details.pizza_id = pizzas.pizza_id
 GROUP BY pizza_types.name
 ORDER BY quantity
 LIMIT 5;
```

Result Grid		Filter Rows:
	name	quantity
▶	The Brie Carre Pizza	490
	The Mediterranean Pizza	934
	The Calabrese Pizza	937
	The Spinach Supreme Pizza	950
	The Soppressata Pizza	961

-- JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

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

Result Grid		
	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

-- DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

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

Result Grid		Filter Rows:
	hour (order_time)	count(order_id)
12	2520	
13	2455	
14	1472	
15	1468	
16	1920	
17	2336	
18	2399	
19	2009	
20	1642	
21	1198	
22	663	
23	28	
10	8	

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

```
SELECT  
    category  
FROM  
    pizza_types  
GROUP BY category;
```

Result Grid	
	category
▶	Chicken
	Classic
	Supreme
	Veggie

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

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

Result Grid	
	round(avg(quantity),0)
▶	138

-- DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

```
SELECT
    pizza_types.name,
    SUM(oder_details.quantity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    oder_details ON oder_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue DESC
LIMIT 3;
```

Result Grid		
	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,  
    SUM(oder_details.quantity * pizzas.price) AS revenue  
FROM  
    pizza_types  
        JOIN  
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
        JOIN  
    oder_details ON oder_details.pizza_id = pizzas.pizza_id  
GROUP BY pizza_types.category  
ORDER BY revenue DESC;
```

Result Grid		Filter Rows:
	category	revenue
▶	Classic	220053.1000000001
	Supreme	208196.99999999822
	Chicken	195919.5
	Veggie	193690.45000000298

-- ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
SELECT
    orders.order_date,
    SUM(oder_details.quantity * pizzas.price) AS revenue
FROM
    oder_details
    JOIN
    pizzas ON oder_details.pizza_id = pizzas.pizza_id
    JOIN
    orders ON orders.order_id = oder_details.order_id
GROUP BY orders.order_date;
```

Result Grid	
order_date	revenue
2015-02-08	2035
2015-02-09	2195.35
2015-02-10	2056.5000000000005
2015-02-11	2459.999999999999
2015-02-12	2158.8
2015-02-13	2754.5
2015-02-14	2319.1500000000005
2015-02-15	2141.25
2015-02-16	1968.800000000002
2015-02-17	2121.9
2015-02-18	2642.85
2015-02-19	2030.2500000000002
2015-02-20	2891.149999999996
2015-02-21	2111
2015-02-22	1579.95

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

```
select name, revenue from
(select category,name,revenue,
rank()over(partition by category order by revenue desc)as rn
from
(select pizza_types.category,pizza_types.name,
sum((oder_details.quantity) * pizzas.price)as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join oder_details
on oder_details.pizza_id = pizzas.pizza_id
group by pizza_types.category,pizza_types.name)as a)as b
where rn=3 ;
```

Result Grid		Filter Rows:
	name	revenue
▶	The California Chicken Pizza	41409.5
	The Pepperoni Pizza	30161.75
	The Sicilian Pizza	30940.5
	The Five Cheese Pizza	26066.5