



# DOMINO'S SQL PROJECT



# HELLO !!!11



R E S T A U R A N T

**Project Brief:** This project involves creating an SQL-based analysis to identify Domino's best-selling and most frequently ordered items. The analysis will utilize sales data to generate insights into customer preferences and popular menu items. The final output will include visual representations and a detailed report summarizing key findings.



R E S T A U R A N T

- RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

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

total\_orders  
21350



-- CALCULATE THE TOTAL REVENUE GENERATED  
FROM PIZZA SALES.

```
select round (sum(order_details.quantity*pizzas.price),2)  
as total_revenue  
from order_details  
join  
pizzas on pizzas.pizza_id= order_details.pizza_id;
```

total\_revenue  
817860.05



RESTAURANT

## -- 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(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;
```

	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(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;
```



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

-- 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;
```

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050





-- 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(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;order_details
```

round(avg(quantity),0)

138

## -- 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 pizzas.pizza_type_id = pizza_types.pizza_types_id  
join order_details  
on order_details.pizza_id = pizzas.pizza_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(order_details.quantity*pizzas.price) / ( select round (sum(order_details.quantity*pizzas.price))  
        as total_revenue  
    from order_details  
    join  
    pizzas on pizzas.pizza_id= order_details.pizza_id) * 100,2) as revenue  
    from pizza_types join pizzas  
    on pizzas.pizza_type_id = pizza_types.pizza_type_id  
    join order_details  
    on order_details.pizza_id = pizzas.pizza_id  
    group by pizza_types.category
```

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

# -- 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(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;
```

order_date	cum_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
2015-01-10	23990.35000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29631.30000000003
2015-01-14	32358.70000000004
2015-01-15	34343.50000000001
2015-01-16	36937.65000000001
2015-01-17	39001.75000000001
2015-01-18	40978.60000000005

# THANK YOU

