



ome Abou

Contact

# SQL PROJECT ON PIZZA SALES

KANCHAN CHOUHAN





#### ABOUT PROJECT

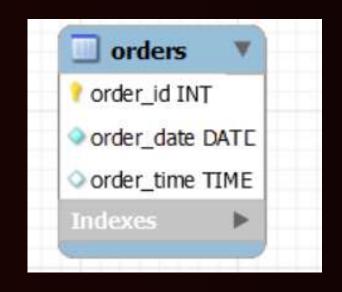
In this project, I utilized SQL queries to solve business problems related to pizza sales



# DATABASE

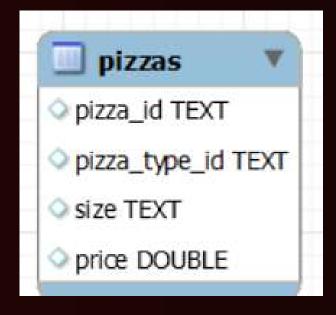
order\_details ▼
order\_details\_id INT
order\_id INT
order\_id INT
pizza\_id TEXT
quantity INT

ORDER DETAILS
ORDERS
PIZZA TYPES
PIZZAS



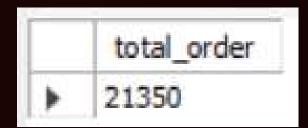








#### RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED







# CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

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

	total_sales
<b>&gt;</b>	188333.25





#### IDENTIFY THE HIGHEST-PRICED PIZZA.

	name	price	
•	The Greek Pizza	35.95	





#### IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
SELECT

COUNT(order_details.pizza_id) AS order_count, pizzas.size

FROM

order_details

JOIN

pizzas ON order_details.pizza_id = pizzas.pizza_id

GROUP BY pizzas.size

ORDER BY order_count DESC;
```

	order_count	size
<b>•</b>	4326	L
	3463	M
	3298	S
	118	XL
	8	XXL





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

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

	name	Order_Quantity
>	The Barbecue Chicken Pizza	583
	The Pepperoni Pizza	579
	The Hawaiian Pizza	555
	The California Chicken Pizza	549
	The Thai Chicken Pizza	532



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

```
SELECT
    pizza_types.category AS category,
    SUM(order_details.quantity) AS Ordered_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	Ordered_quantity
<b>&gt;</b>	Classic	3382
	Veggie	2752
	Supreme	2745
	Chicken	2544





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

```
SELECT

orders.order_time, COUNT(orders.order_id)

FROM

orders

GROUP BY order_time;
```

	order_time	count(orders.order_id)
١	11:38:36	2
	11:57:40	1
	12:12:28	1
	12:16:31	3
	12:21:30	1
	12:29:36	1
	12:50:37	1
	12:51:37	1
	12:52:01	1
	13:00:15	2
	13:02:59	1
	13:04:41	2
	13:11:55	1
	13:14:19	2
	13:33:00	2
	13:34:07	3
	13:53:00	1
	13:57:08	1
	13:59:09	1
	14:03:08	3
Re	sult 10 ×	



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

```
SELECT

pizza_types.category, COUNT(pizza_types.name) AS pizzas

FROM

pizza_types

GROUP BY pizza_types.category
```

category	pizzas
Chicken	6
Classic	8
Supreme	9
Veggie	9





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

100	Suit Oilo   D.D
	AVG(qty)
<b>b</b>	137 6265





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

	name	revenue
<b>&gt;</b>	The Barbecue Chicken Pizza	10297.25
	The Thai Chicken Pizza	9659
	The California Chicken Pizza	9563.75



# CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

```
SELECT
    pizza_types.category,
    SUM(order_details.quantity * pizzas.price) /
    (
        SELECT SUM(order_details.quantity * pizzas.price)
        FROM order_details
        JOIN pizzas ON pizzas.pizza_id = order_details.pizza_id
    ) * 100 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_types.category
ORDER BY revenue DESC;
```

	category	revenue
>	Classic	26.458206397436406
	Supreme	25.23943594665308
	Veggie	24.378807247259715
	Chicken	23.92355040865058





