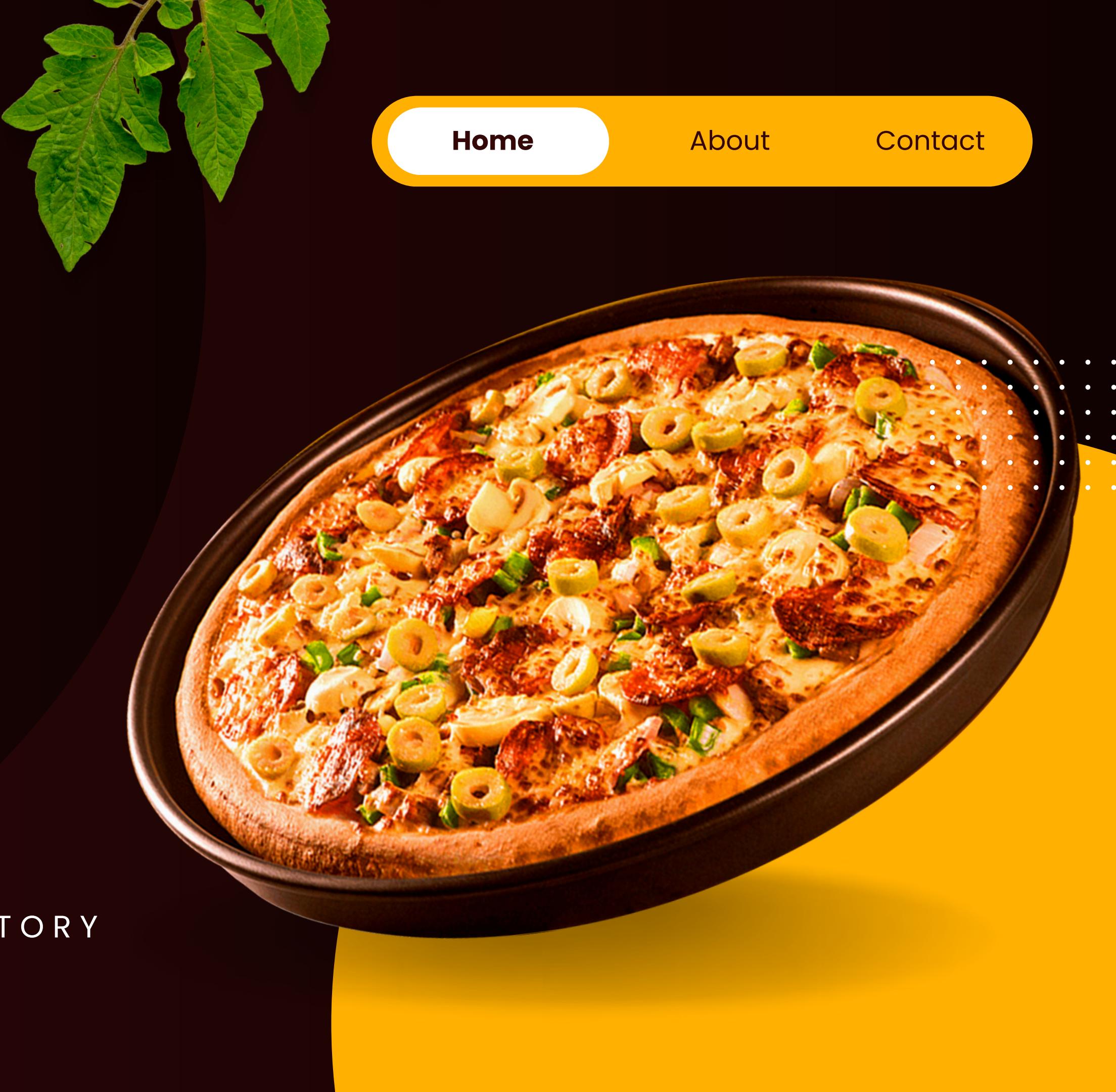


# SQL PROJECT ON PIZZA SALES

● WHERE EVERY SLICE TELLS A STORY



Home

About

Contact

# HELLO!!

Home

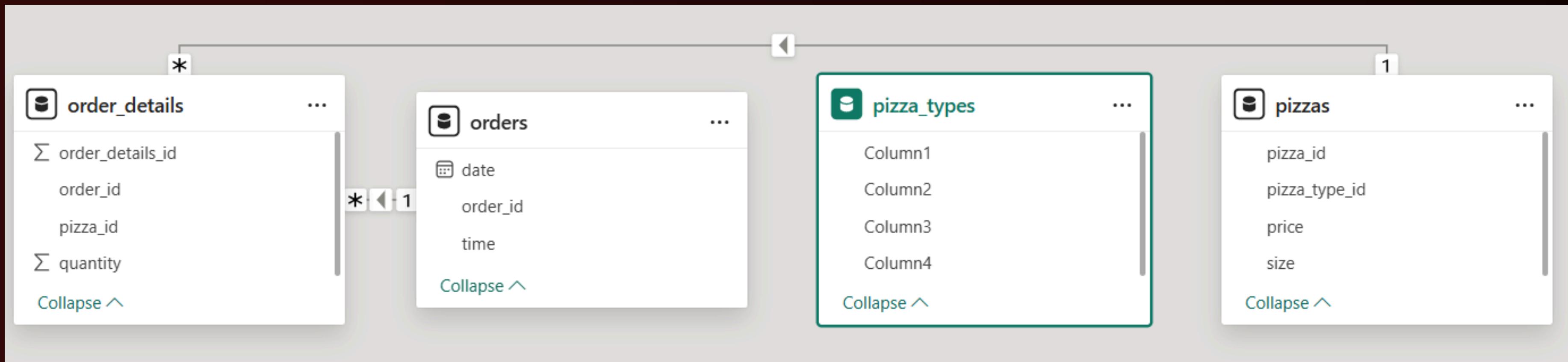
About

Contact



This SQL-based project involves analyzing pizza sales data to extract meaningful business insights. Using SQL queries, key metrics such as total orders, total revenue, and the most popular pizza types, sizes, and categories are identified. Advanced SQL techniques like joins, grouping, and aggregate functions are applied to explore order distribution by time, calculate daily averages, and analyze cumulative revenue trends.

# SCHEMA



# Retrieve the total number of orders placed.

[Home](#)

[About](#)

[Contact](#)

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

Result Grid	
	total_orders
▶	21350

# Calculate the total revenue generated from pizza sales.

[Home](#)[About](#)[Contact](#)

```
SELECT  
    SUM(order_details.quantity * pizzas.price) AS total_sales  
FROM  
    order_details  
        JOIN  
    pizzas ON pizzas.pizza_id = order_details.pizza_id
```

	total_sales
▶	817860.049999993



# Identify the highest-priced pizza.

[Home](#)[About](#)[Contact](#)

```
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 Rows

	name	price
▶	The Greek Pizza	35.95



# Identify the most common pizza size ordered.

[Home](#)[About](#)[Contact](#)

```
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;
```

Result Grid		
	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

# Determine the distribution of orders by hour of the day.

[Home](#)[About](#)[Contact](#)

```
select hour(order_time) as hour, count(order_id) as order_count from orders  
group by hour(order_time);
```

Result Grid		
	hour	order_count
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336
	18	2399
	19	2009
	20	1642
	21	1198
	22	663
	23	28



# Join relevant tables to find the category-wise distribution of pizzas.

```
select category, count(name) from pizza_types  
group by category;
```

	category	count(name)
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

# Group the orders by date and calculate the average number of pizzas ordered per day.

```
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	
	avg_pizza_ordered_per_day
▶	138

# Determine the top 3 most ordered pizza types based on revenue.

[Home](#)[About](#)[Contact](#)

```
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_type_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

# Analyze the cumulative revenue generated over time.

[Home](#)[About](#)[Contact](#)

```
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;
```

Result Grid | Filter Rows:

order_date	cum_revenue
2015-01-01	2713.8500000000
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

# 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((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 rn <=3;
```

Result Grid		
	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
	The Sicilian Pizza	30940.5



Home

About

Contact

**THANK YOU  
FOR ATTENTION.**