LARANA PIZZA

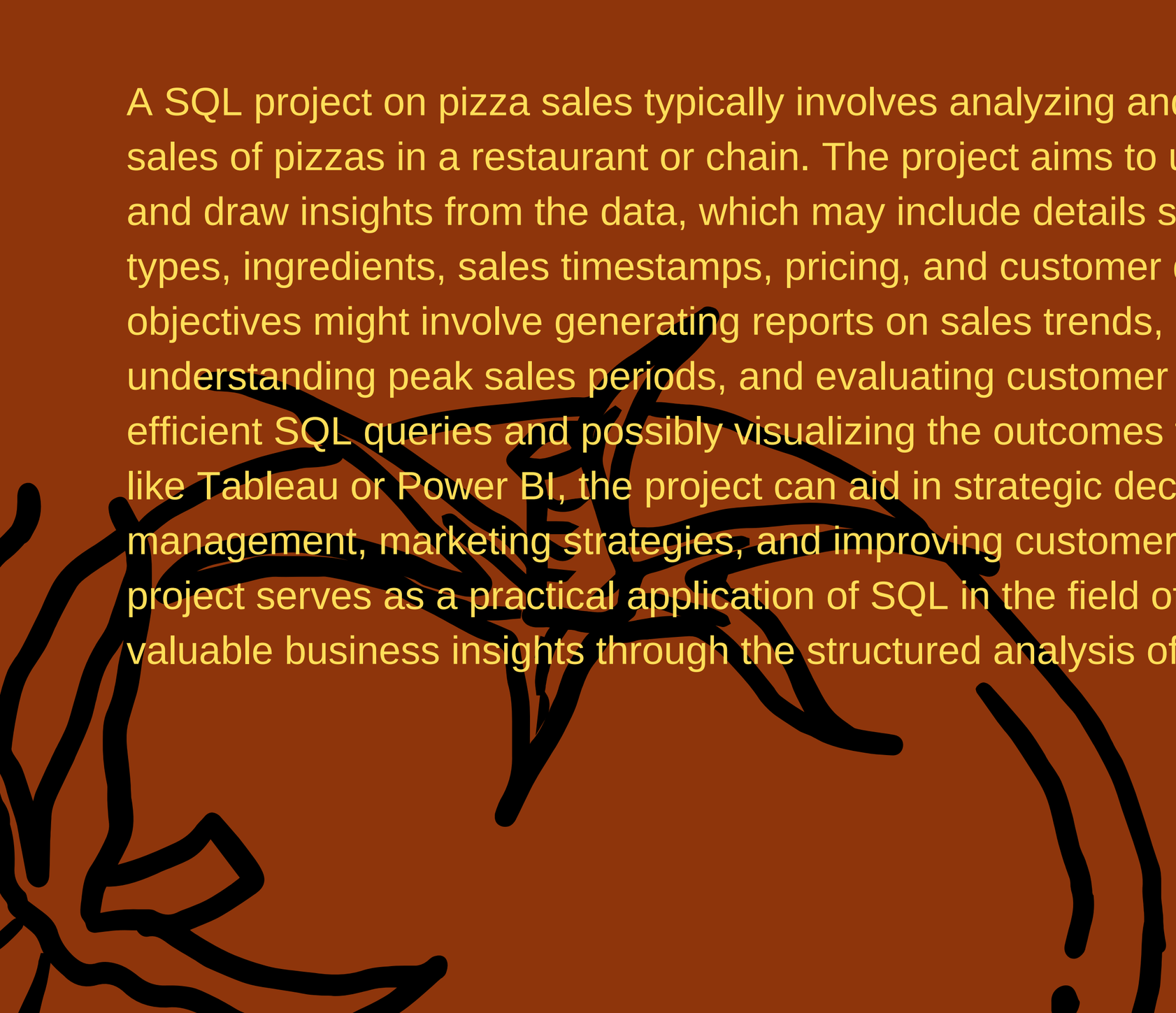
LARANA PIZZA

LARANA PIZZA

WELCOME TO
LARANA PIZZA

Welcome to our Pizza Café, where we blend tradition and innovation to serve the most delicious pizzas in town! Nestled in the heart of the city, our café is a cozy spot perfect for casual lunches, family dinners, or a quick bite with friends. We pride ourselves on using only the finest, freshest ingredients, from our hand-tossed dough to our rich, flavorful sauces and premium toppings. Our menu offers a variety of classic and gourmet pizzas, ensuring there's something to satisfy every palate. We also cater to dietary preferences with gluten-free and vegan options. Beyond pizza, we offer a selection of appetizers, salads, and desserts, all crafted with the same dedication to quality. Join us for a warm meal, friendly service, and an inviting atmosphere that feels like home. Whether you're a pizza purist or an adventurous foodie, our Pizza Café is your new favorite destination.

A SQL project on pizza sales typically involves analyzing and managing data related to the sales of pizzas in a restaurant or chain. The project aims to utilize SQL to organize, query, and draw insights from the data, which may include details such as customer orders, pizza types, ingredients, sales timestamps, pricing, and customer demographics. The primary objectives might involve generating reports on sales trends, identifying best-selling pizzas, understanding peak sales periods, and evaluating customer preferences. By creating efficient SQL queries and possibly visualizing the outcomes through integration with tools like Tableau or Power BI, the project can aid in strategic decision-making, such as inventory management, marketing strategies, and improving customer satisfaction. Overall, such a project serves as a practical application of SQL in the field of data analytics, providing valuable business insights through the structured analysis of sales data.



-- RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED .

```sql
SELECT
    COUNT(order_id) AS total_order
FROM
    orders;
```

Result Grid

	total_order
▶	21350

-- CALCULATE THE TOTAL REVENUE GENERATED FROM PLACED.

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

Result Grid | Filter R

	total_sales
▶	817860.05

LIST THE TOP 5 MOST ORDERED PIZZA TYPE -- ALONG WITH THEIR QUANTITIES.

```sql
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 | Filter Rows:

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

-- ANALYZE THE CULMATIVE REVENUE GENERATED OVER TIME .

```sql
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	2713.8500000000004
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	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7

IDENTIFY THE HIGHEST -PRICED PIZZA.

```sql
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

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

```sql
SELECT
    pizza_types.category,
    ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
                ROUND(SUM(order_details.quantity * pizzas.price),
                            2) AS total_sales
            FROM
                order_details
                    JOIN
                pizzas ON (pizzas.pizza_id = order_details.pizza_id)) * 100,
        2) 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
ORDER BY revenue DESC;
```

Result Grid | Filte

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

- DETERMINE THE TOP3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY .

```sql
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 | Filter Rows:

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
The Four Cheese Pizza	32265.70000000065
The Mexicana Pizza	26780.75
The Five Cheese Pizza	26066.5

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

```sql
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 | Filter Rows:

avg_pizza_ordered_per_day
138

-- IDENTITY THE MOST COMMON PIZZA SIZE ORDERED.

```sql
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
LIMIT 1;
```

Result Grid | Filter

	size	order_count
▶	L	18526

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

```sql
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;
```

Result Grid | Filter Rows:

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

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

```sql
SELECT
    HOUR(order_time) AS hour, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY HOUR(order_time);
```

Result Grid | Filt

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
10	8
9	1

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

```sql
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;
```

Result Grid | Filte

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

LARANA PIZZA

THANK YOU!