PIZZA SALES ANALYSIS

LARANA PIZZA

WELCOME TO
SQL PROJECT

This is Pizza Sales Analysis Project, where we dive deep into the world of data using the power of SQL. This project explores real-world pizza sales data to uncover insights into:

- Revenue trends
- Top-selling categories
- Customer purchase behavior
- Cumulative sales performance

By leveraging advanced SQL techniques — including joins, aggregations, window functions, and subqueries — we analyze how different pizza types and order patterns contribute to overall business performance.

LARANA PIZZA

1. RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```sql
select count(order_ID) as total_orders from orders;
```

Result Grid

total_orders
21350

LARANA PIZZA

2. THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```
SELECT
    ROUND(SUM(orders_details.quantity * pizzas.price),
            2) AS total_revenue
FROM
    orders_details
        JOIN
    pizzas ON pizzas.pizza_id = orders_details.pizza_id
```

Result Grid

	total_revenue
▶	817860.05

LARANA PIZZA

3. 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

LARANA PIZZA

4.IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
select pizzas.size,count(orders_details.order_details_id )as order_count
from pizzas join orders_details
on pizzas.pizza_id = orders_details.pizza_id
group by pizzas.size order by order_count desc;
```

Result Grid | Filter

size	order_count
L	18526
M	15385
S	14137
XL	544
XXL	28

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

LAKANA PIZZA

```
select pizza_types.name,
sum(orders_details.quantity) as quantity
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join orders_details
on orders_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

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

LARANA PIZZA

```sql
select pizza_types.category,
sum(orders_details.quantity) as quantity
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join orders_details
on orders_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

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

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

Result Grid | Filter R

hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468

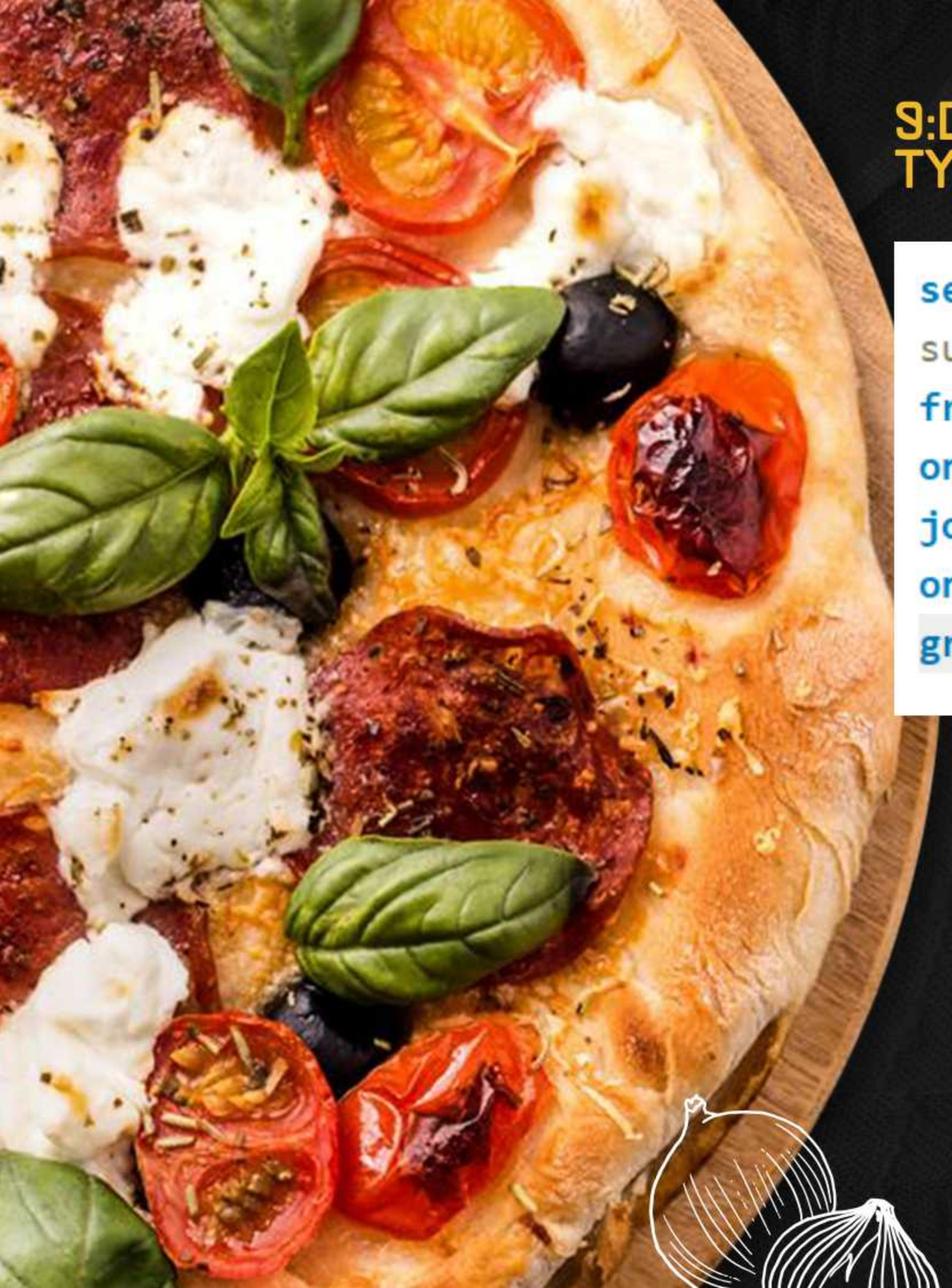
LARANA PIZZA

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

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

Result Grid | Filter Row

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



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

LARANA PIZZA

```sql
select pizza_types.name,
sum(orders_details.quantity * pizzas.price) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join orders_details
on orders_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

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

```sql
SELECT pizza_types.category,
round(SUM(orders_details.quantity * pizzas.price) / (select
    round(sum(orders_details.quantity * pizzas.price),
    2) as total_sales
FROM
    orders_details
        JOIN
    pizzas ON pizzas.pizza_id = orders_details.pizza_id) *100,2) as revenue
    from pizza_types join pizzas
    on pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY revenue DESC;
```

Result Grid | Filter Rov

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

11: ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

LARANA PIZZA

```
select order_date,
sum(revenue) over (order by order_date) as cum_revenue
from
(select orders.order_date,
sum(orders_details.quantity * pizzas.price) as revenue
from orders_details join pizzas
on orders_details.pizza_id = pizzas.pizza_id
join orders
on orders.order_id = orders_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

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

```sql
select round(avg(quantity),0) as average_pizza_order_per_day from
(select orders.order_date, sum(orders_details.quantity) as quantity
from orders join orders_details
on orders.order_ID = orders_details.order_ID
group by orders.order_date) as order_quantity;
```

Result Grid | Filter Rows:

average_pizza_order_per_day
138

13: DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED 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(orders_details.quantity * pizzas.price) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join orders_details
on orders_details.pizza_id = pizzas.pizza_id
group by pizza_types.category, pizza_types.name) as a) as b
where rn<= 3;
```

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

THANK YOU!