



PIZZA SALE



BUY ONE GET
ONE FREE

lahore pakistan

ORDER NOW

www.stillblack.com

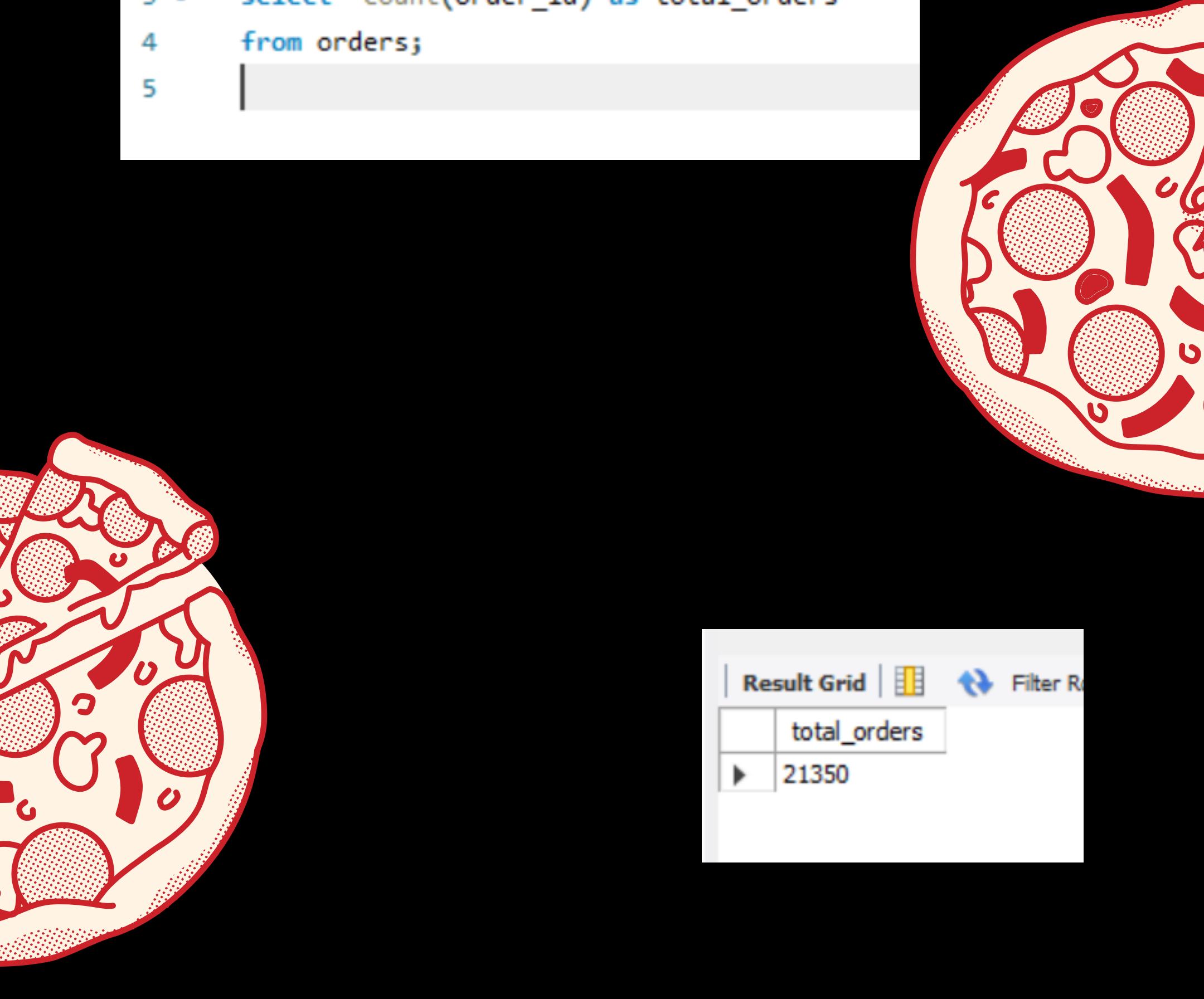
PROJECT INTRODUCTION

This project focuses on analyzing pizza sales data using SQL to uncover key business insights. The dataset contains detailed information about customer orders, pizza types, quantities, prices, and order times. Using SQL queries, I performed data cleaning, aggregation, and exploration to identify sales trends, peak ordering times, most popular pizzas, and revenue performance. The goal was to provide actionable insights that can help improve decision-making in areas such as inventory management, marketing, and menu optimization.



TOTAL NUMBER OF ORDERS PLACED

```
3 •  select count(order_id) as total_orders  
4   from orders;  
5   |
```



A decorative illustration of a pizza slice on the left and a whole pizza on the right, both rendered in red and white colors.

Result Grid	
	total_orders
▶	21350

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES

```
select  
ROUND(sum(pizzas.price * orders_details.quantity ),2 )as total_sales  
from pizzas join orders_details  
on pizzas.pizza_id = orders_details.pizza_id
```

Result Grid	
	total_sales
▶	817860.05

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 ;
```



Result Grid | Filter Rows:

	name	price
▶	The Greek Pizza	35.95

IDENTIFY THE MOST COMMON SIZE OF PIZZA

```
select size ,count(*) as count_per_size  
from pizzas  
group by size  
order by count_per_size desc  
limit 1;
```

Result Grid | Filter Rows:

	size	count_per_size
▶	S	32



JOIN THE NECESSARY TABLE TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED

```
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;
```



	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

DISTRIBUTION OF ORDERS BY HOURS 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
	16	1920
	17	2336
	18	2399



JOIN RELEVANT TABLES TO FIND 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(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	
	avg_pizza_ordered_per_day
▶	138



TOP 3 PIZZA ORDERED BASED ON REVENUE

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

CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE

```
select pizza_types.category,
       (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;
```



	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

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(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
	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



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(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;
```

	name	revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5
	The Southwest Chicken Pizza	34705.75
	The Chicken Alfredo Pizza	16900.25
	The Chicken Pesto Pizza	16701.75
	The Classic Deluxe Pizza	38180.5
	The Hawaiian Pizza	32273.25
	The Pepperoni Pizza	30161.75
	The Greek Pizza	28454.100000000013
	The Italian Capriccio Pizza	25094