

SOL
PROJECT
FOR PIZZA
SALES

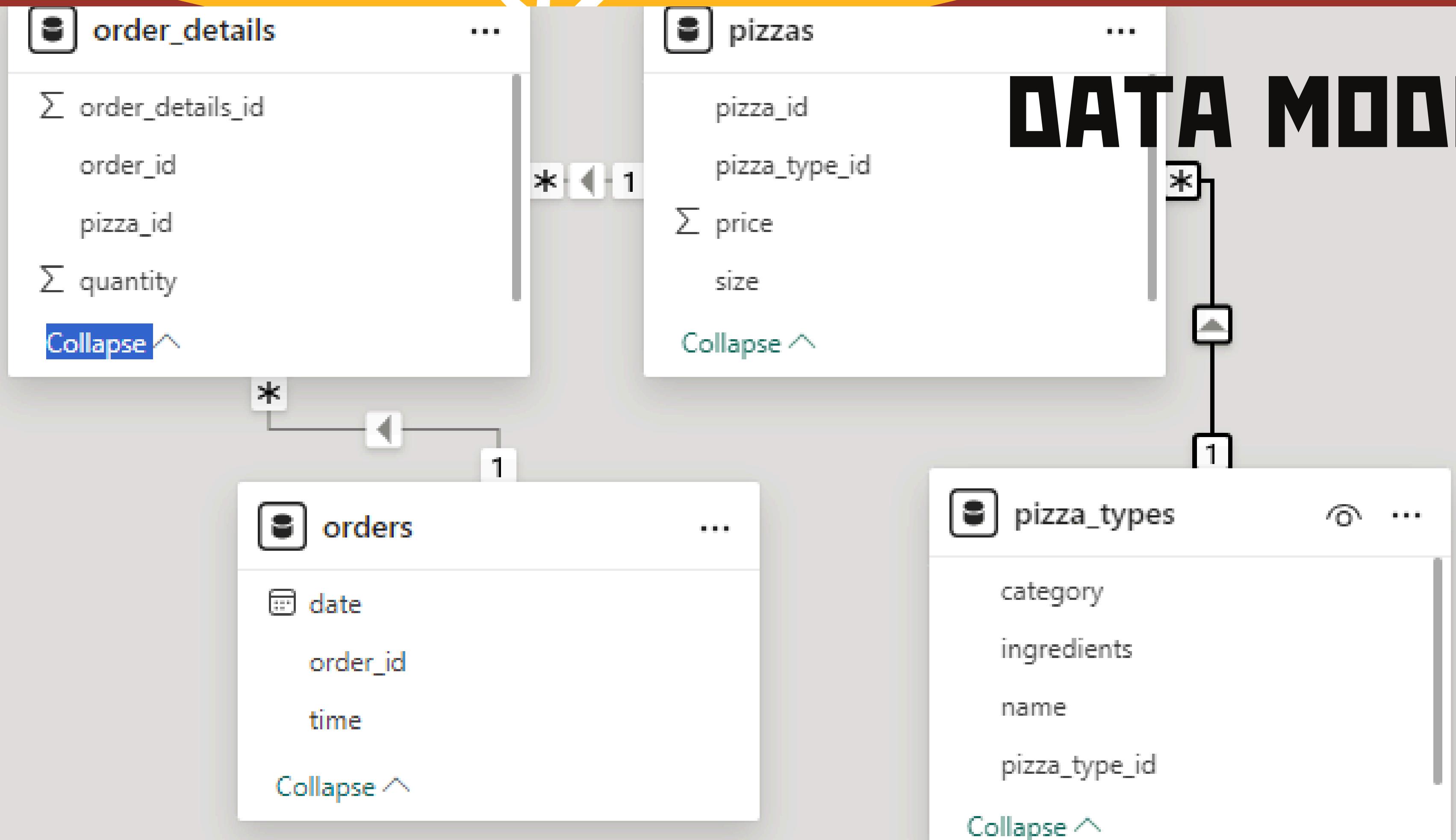


A cartoon illustration of two people eating pizza. On the left, a woman with dark hair and glasses, wearing a yellow shirt, holds a slice of pizza. On the right, a man with a mustache and a green baseball cap, wearing a yellow striped shirt, also holds a slice of pizza. They are both smiling. The background is a red grid with yellow starburst shapes.

INTRODUCTION

Pizza, as a universally beloved dish, serves as a unique lens through which we can examine consumer behavior, operational challenges, and opportunities for growth. By harnessing the power of SQL, we have been able to transform raw transactional data into actionable insights that drive informed decision-making and enhance our overall business strategy.

DATA MODEL



VARIATIONS



Pizza Margherita



Pepperoni Pizza



BBQ Chicken Pizza



Hawaiian Pizza



QUESTIONS WITH SQL QUERIES AND OUTPUT

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```
3 •     SELECT  
4         COUNT(order_id)  
5     FROM  
6     order_details;
```



Result Grid	
	COUNT(order_id)
↓	48620

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES

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

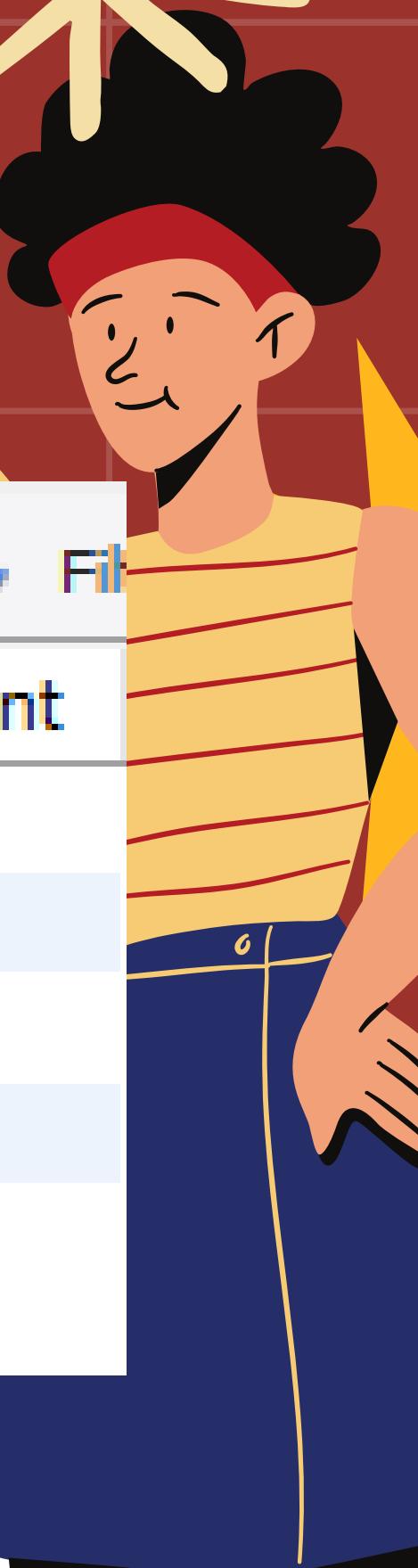
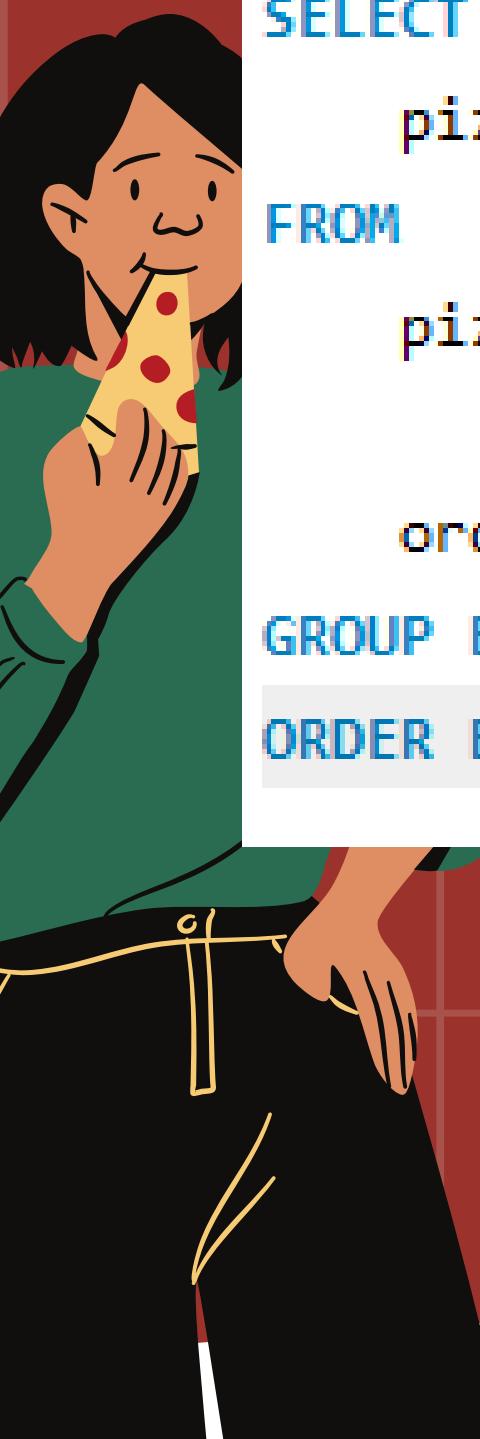
Result Grid | Filter Rows

	name	price
▶	The Greek Pizza	35.95

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
SELECT
    pizzas.size, COUNT(order_details.quantity) 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 total_quantity  
FROM  
    pizza_types  
        JOIN  
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id  
        JOIN  
    order_details ON pizzas.pizza_id = order_details.pizza_id  
GROUP BY pizza_types.name  
ORDER BY total_quantity DESC  
LIMIT 5;
```

	name	total_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 total_quantity  
  
FROM  
    pizza_types  
        JOIN  
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id  
        JOIN  
    order_details ON pizzas.pizza_id = order_details.pizza_id  
GROUP BY pizza_types.category  
ORDER BY total_quantity DESC;
```

	category	total_quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

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

	hour	count_of_orders
▶	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 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

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

SELECT

```
ROUND(AVG(qty), 0) as avg_pizza_per_day
```

FROM

```
(SELECT
```

```
    orders.order_date, SUM(order_details.quantity) AS qty
```

FROM

```
    orders
```

```
JOIN order_details ON order_details.order_id = orders.order_id
```

```
GROUP BY orders.order_date) AS order_qty;
```

	Result Grid	Filter
	avg_pizza_per_day	
▶	138	

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

```
select pizza_types.name, sum(order_details.quantity * pizzas.price) as revenue
from pizzas join pizza_types
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

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

```
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 order_details.pizza_id = pizzas.pizza_id)*100,2) as revenue  
from pizzas join pizza_types  
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.category  
order by revenue desc;
```

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
select order_date, round(sum(revenue) over (order by order_date),2) as cummulative_rev
from
(select orders.order_date, sum(order_details.quantity * pizzas.price) as revenue
from order_details join orders
on order_details.order_id = orders.order_id
join pizzas
on pizzas.pizza_id = order_details.pizza_id
group by orders.order_date) as sales;
```

Result Grid | Filter Rows:

	order_date	cummulative_rev
▶	2015-01-01	2713.85
	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	22929.25

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 pizzas join pizza_types
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.category, pizza_types.name) as a) as b
where rn<=3;
```

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

OUTPUT

	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	32265.700
	The Mexicana Pizza	26780.75
	The Five Cheese Pizza	26066.5

A festive illustration set against a red background with a white grid. In the center, the words "THANK YOU" are written in large, bold, white capital letters. To the left, a person with dark curly hair and a green shirt holds a slice of pizza. To the right, another person with glasses and a green shirt holds a piece of pie. Above them, a reindeer with a yellow and orange patterned collar and a bell hangs from its neck looks down. The reindeer's collar features a dollar sign (\$) symbol. The scene is decorated with yellow stars and a large yellow starburst at the bottom.

THANK YOU