

TRY OUR NEW

PIZZA

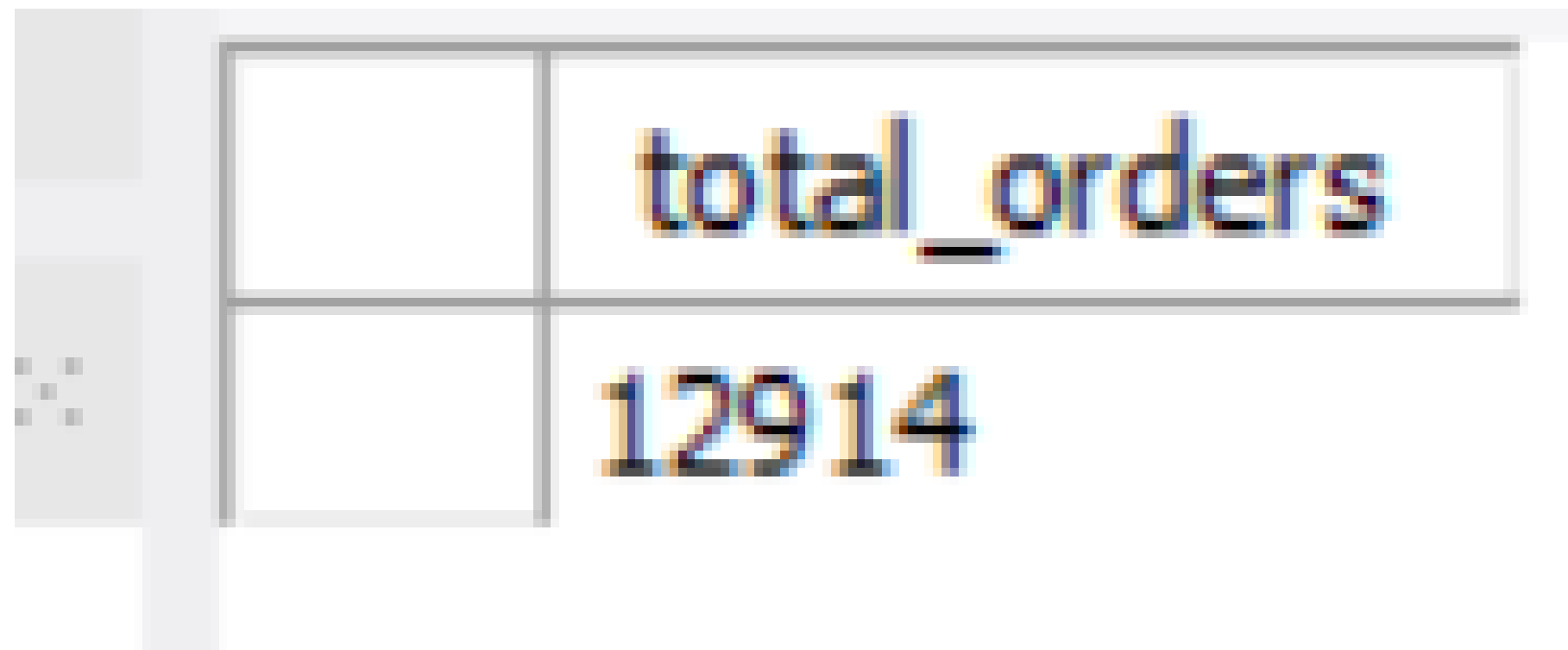
ORDER NOW

 +123-456-7890

 123 ANYWHERE ST., ANY CITY

1- RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```
select count(order_id) as total_orders  
from orders;
```



	total_orders
12914	

2-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 pizzas.pizza_id = order_details.pizza_id;
```

	total_sales
▶	49874.55

3-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;
```

	name	price
▶	The Greek Pizza	35.95

4-IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
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			FILE NAME	
	size	order_count		
▶	L	1127		

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

```
SELECT pizza_types.name,  
SUM(order_details.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 SUM(order_details.quantity) DESC  
LIMIT 5;
```

	name	SUM(order_details.quantity)
►	The Hawaiian Pizza	168
	The Thai Chicken Pizza	159
	The California Chicken Pizza	139
	The Classic Deluxe Pizza	137

6-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	925
	Veggie	733
	Supreme	715
	Meat	688

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

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

	hour	order_count
▶	11	730
	12	1512
	13	1461
	14	938
	15	897
	16	1156
	17	1450

Result 2

8-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

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

```
SELECT 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 pizzas.pizza_id = order_details.pizza_id  
GROUP BY pizza_types.name  
ORDER BY revenue DESC  
LIMIT 3;  
select* from pizza_types;
```

	pizza_type_id	name	category	ingredients
►	bbq_ckn	The Barbecue Chicken Pizza	Chicken	Barbecued Chicken, Red Peppers, Green Peppe...
	cali_ckn	The California Chicken Pizza	Chicken	Chicken, Artichoke, Spinach, Garlic, Jalapeno P...
	ckn_alfredo	The Chicken Alfredo Pizza	Chicken	Chicken, Red Onions, Red Peppers, Mushrooms...
	ckn_pesto	The Chicken Pesto Pizza	Chicken	Chicken, Tomatoes, Red Peppers, Spinach, Garl...
	southw_ckn	The Southwest Chicken Pizza	Chicken	Chicken, Tomatoes, Red Peppers, Red Onions, ...

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

```
(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:	E
	name	revenue		
	The Spicy Italian Pizza	34831.25		
	The Italian Supreme Pizza	33476.75		
	The Sicilian Pizza	30940.5		

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

```
SELECT AVG(total_quantity) AS average_pizzas_ordered_per_day
FROM (
    SELECT orders.date, SUM(order_details.quantity) AS total_quantity
    FROM orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.date
) AS order_quantity;
```

RESULT GRID		Filter Rows:
	average_pizzas_ordered_per_day	
▶	NULL	

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

```
SELECT pizza_types.category,  
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  
ORDER BY revenue DESC;
```

	category	revenue
▶	Classic	13835.9
	Supreme	12295.2499999999993
	Margherita	12295.2499999999993

13-ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME

```
SELECT orders.date,  
       SUM(order_details.quantity * pizzas.price) AS daily_revenue,  
       SUM(SUM(order_details.quantity * pizzas.price)) OVER (ORDER BY orders.date) AS cumulative_revenue  
FROM orders  
JOIN order_details ON orders.order_id = order_details.order_id  
JOIN pizzas ON order_details.pizza_id = pizzas.pizza_id  
JOIN pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id  
GROUP BY orders.date  
ORDER BY orders.date;
```

	date	daily_revenue	cumulative_revenue
--	------	---------------	--------------------