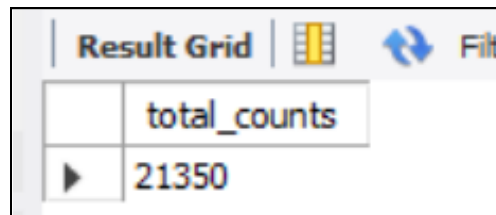


Pizza Sales & Orders Analysis

1. Retrieve the total number of orders placed.

```
SELECT
    COUNT(order_id) AS total_counts
FROM
    orders;
```

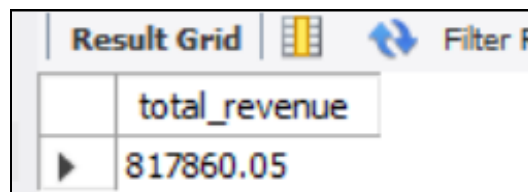


The screenshot shows a 'Result Grid' window with a single column header 'total_counts' and a single row of data containing the value '21350'.

	total_counts
▶	21350

2. Overall revenue earned from all pizza sales.

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



The screenshot shows a 'Result Grid' window with a single column header 'total_revenue' and a single row of data containing the value '817860.05'.

	total_revenue
▶	817860.05

3. Pizza with the highest price listed on the menu.

```
SELECT
    pt.name, p.price
FROM
    pizza_types AS pt
    JOIN
    pizzas AS p ON pt.pizza_type_id = p.pizza_type_id
ORDER BY p.price DESC LIMIT 1;
```

Result Grid			Filter Rows
	name	price	
▶	The Greek Pizza	35.95	

3. Which pizza size is ordered the most across all orders?

```
SELECT
    pizzas.size, COUNT(order_details_id) as order_count
FROM
    pizzas
    JOIN
        orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizzas.size order by order_count desc;
```

Result Grid			Filter Rows
	size	order_count	
▶	L	18526	
	M	15385	
	S	14137	
	XL	544	
	XXL	28	

3. Top 5 pizza types based on total quantity sold to customers.

```
SELECT
    pizza_types.name, SUM(orders_details.quantity) AS quantities
FROM
    pizza_types
    JOIN
        pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
    JOIN
        orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizza_types.name
ORDER BY quantities DESC LIMIT 5;
```

	name	quantities
▶	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418
	The Thai Chicken Pizza	2371

3. Total number of pizzas sold for each pizza category.

```
SELECT
    pizza_types.category,
    SUM(orders_details.quantity) AS quantity
FROM
    pizza_types
    JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
    JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizza_types.category;
```

	category	quantity
▶	Classic	14888
	Veggie	11649
	Supreme	11987
	Chicken	11050

3. Breakdown of orders by each hour of the day to spot peak times.

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

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

3. Count of different pizza types available under each category.

```

SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category;

```



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

3. Average number of pizzas sold per day over the recorded period.

```

SELECT
    ROUND(AVG(quantity), 0) AS avg_number_of_pizza_perday
FROM
    (SELECT
        orders.order_date AS date,
        SUM(orders_details.quantity) AS quantity
    FROM
        orders
    JOIN orders_details ON orders.order_id = orders_details.order_id
    GROUP BY date) AS order_quantity;

```

Result Grid			Filter Rows:	
	avg_number_of_pizza_perday			
▶	138			

3. Top 3 best-selling pizzas in terms of total revenue generated.

```

SELECT
    pizza_types.name,
    ROUND(SUM(pizzas.price * orders_details.quantity),2) AS revenue
FROM
    pizzas
    JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
    JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue DESC LIMIT 3;

```

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

3. Share of total revenue contributed by each pizza category in percentage.

```

SELECT
    pizza_types.category,
    round((SUM(pizzas.price * orders_details.quantity) / (SELECT
        ROUND(SUM(pizzas.price * orders_details.quantity),2)
    FROM
        pizzas
        JOIN
        orders_details ON pizzas.pizza_id = orders_details.pizza_id)) * 100,2) AS revenue
FROM
    pizzas
    JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
    JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizza_types.category
ORDER BY revenue;

```

	category	revenue
▶	Veggie	23.68
	Chicken	23.96
	Supreme	25.46
	Classic	26.91

3. Cumulative revenue trend shown over time (by day).

```

select date, round(sum(revenue) over (order by date), 2) as cumulative_revenue from
(SELECT
  orders.order_date AS date,
  ROUND(SUM(orders_details.quantity * pizzas.price),
    2) AS revenue
FROM
  orders_details
  JOIN
  orders ON orders.order_id = orders_details.order_id
  JOIN
  pizzas ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY date) as daily_data;

```

	date	cumulative_revenue
▶	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

3. Cumulative revenue growth tracked by hour across all days.

```

SELECT hour, round(sum(revenue) OVER (ORDER BY hour), 2) AS cumulative_revenue FROM
(SELECT
  hour(orders.order_time) AS hour,
  ROUND(SUM(orders_details.quantity * pizzas.price),
    2) AS revenue
FROM
  orders_details
  JOIN
  orders ON orders.order_id = orders_details.order_id
  JOIN
  pizzas ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY hour) AS hourly_data;

```

	hour	cumulative_revenue
▶	9	83
	10	386.65
	11	45322.45
	12	157200.35
	13	263266.05

3. Top 3 revenue-generating pizza types from every individual category.

```
SELECT
    category, name, revenue
FROM
    (SELECT
        category, name, revenue,
        RANK() OVER (PARTITION BY category ORDER BY revenue DESC) AS rev
    FROM
        (SELECT
            pizza_types.category AS category,
            pizza_types.name AS name,
            ROUND(SUM(orders_details.quantity * pizzas.price), 2) AS revenue
        FROM
            pizzas
        JOIN pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN orders_details ON pizzas.pizza_id = orders_details.pizza_id
        GROUP BY category, name
        ) AS a
    ) AS b
WHERE rev <= 3;
```

Result Grid				Filter Rows:	Exp
	category	name	revenue		
►	Chicken	The Thai Chicken Pizza	43434.25		
	Chicken	The Barbecue Chicken Pizza	42768		
	Chicken	The California Chicken Pizza	41409.5		
	Classic	The Classic Deluxe Pizza	38180.5		
	Classic	The Hawaiian Pizza	32273.25		