

Pizza Sales Analysis Project

Presentation

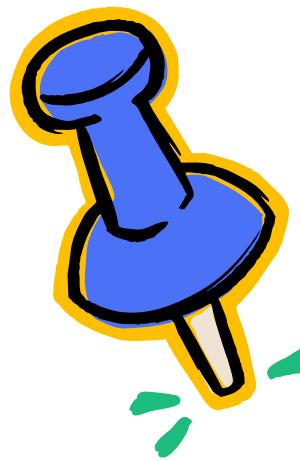


Pizza Pleasures: A Delectable Dive into Sales Insights

Delve into the essence of pizza data, from crust to toppings. Uncover tantalizing trends like total orders and top-sellers. Discover the highest-priced masterpiece and common sizes. Explore the top 5 customer favorites. From basic stats to advanced insights, savor the flavors of success, enriching both senses and bottom line.

Questions :

- Retrieve the total number of orders placed.
- Calculate the total revenue generated from pizza sales.
- Identify the highest-priced pizza.
- Identify the most common pizza size ordered.
- List the top 5 most ordered pizza types along with their quantities
- Join the necessary tables to find the total quantity of each pizza category ordered.
- Determine the distribution of orders by hour of the day.
- Join relevant tables to find the category-wise distribution of pizzas.
- Group the orders by date and calculate the average number of pizzas ordered per day.
- Determine the top 3 most ordered pizza types based on revenue.
- Analyze the cumulative revenue generated over time.

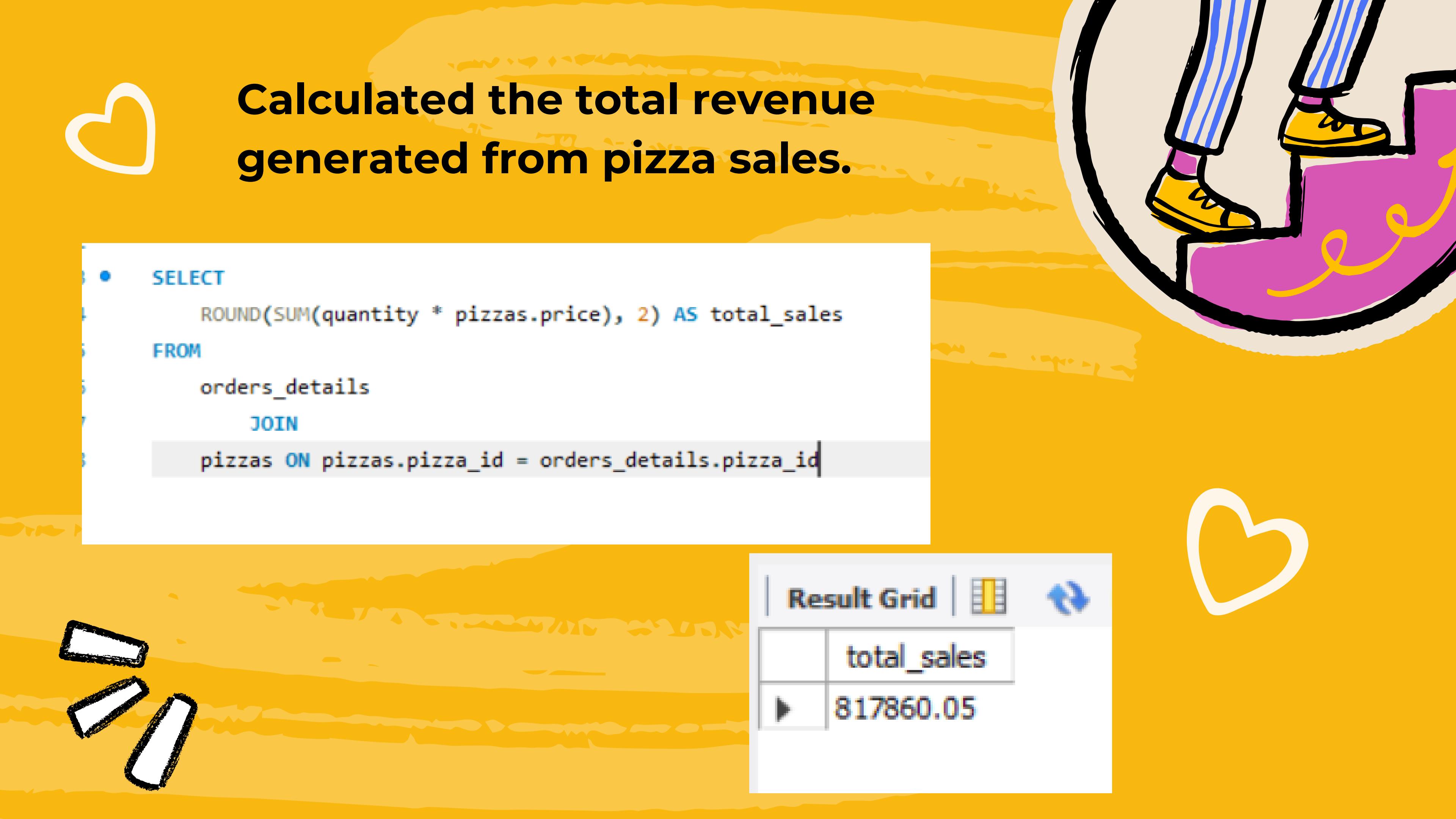


Retrieve the total number of orders placed.

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

Result Grid | Filter Rows:

total_orders
21350



Calculated the total revenue generated from pizza sales.

```
• SELECT  
  ROUND(SUM(quantity * pizzas.price), 2) AS total_sales  
FROM  
  orders_details  
  JOIN  
    pizzas 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 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

Top 5 most ordered pizza types along with their quantities.



```
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

Joined the necessary tables 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;
```

Result Grid | Filter |

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

Determine the distribution of orders by hour of the day.



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

	HOUR(order_time)	order_count
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	

Joined 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

Grouped the orders by date and calculate the average number of pizzas ordered per day.

```
• SELECT  
    ROUND(AVG(quantity), 0)  
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;
```



	round(AVG(quantity),0)
▶	138

Determine the top 3 most ordered pizza types 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

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 | 23990.35000000002  |
|   | 2015-01-11 | 25862.65           |
|   | 2015-01-12 | 27781.7            |
|   | 2015-01-13 | 29831.30000000003  |
|   | 2015-01-14 | 32358.70000000004  |
|   | 2015-01-15 | 34343.50000000001  |

Await your feedback and any additional requirements

