Home About Contact

WHERE EVERY SLICE TELLS A STORY

Home About Contact

HELLOO !!

I am Ayush Suryawanshi, and in this project, I used PostgreSQL to analyze a pizza sales dataset. Through SQL queries, I derived insights on sales, revenue, and customer preferences, showcasing the power of database management for business analysis.

The project utilizes three main datasets:

1. **pizza_types**: Contains information about various pizza types, including their unique IDs, names, categories (e.g., Chicken, Classic, Veggie), and ingredients.
2. **orders**: Includes data on individual customer orders, with order IDs, dates, and times.
3. **order_details**: Provides details of each order, including the order ID, pizza IDs, and quantities of pizzas ordered.

Retrieve the total number of orders placed.

```
SELECT
    COUNT(ORDER_ID) AS TOTAL_ORDERS
FROM
    ORDERS;
```

	total_orders bigint
1	21350

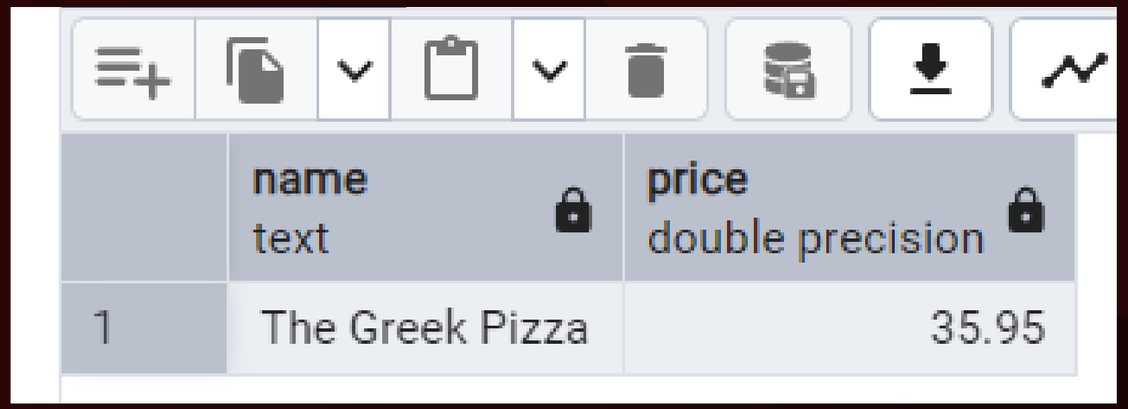
Calculate the total revenue generated from pizza sales.

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

	total_sales numeric
1	817860.05

Identify the highest-priced pizza.

```sql
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 text	price double precision
1	The Greek Pizza	35.95

Identify the most common pizza size ordered.

```sql
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;
```

	size text	order_count bigint
1	L	18526
2	M	15385
3	S	14137
4	XL	544
5	XXL	28

Home About Contact

List the top 5 most ordered pizza types along with their quantities

```sql
SELECT
    pizza_types.name,
    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.name
ORDER BY
    quantity DESC
LIMIT 5;
```

	name text	quantity numeric
1	The Classic Deluxe Pizza	2453
2	The Barbecue Chicken Pizza	2432
3	The Hawaiian Pizza	2422
4	The Pepperoni Pizza	2418
5	The Thai Chicken Pizza	2371

Join the necessary tables to find the total quantity of each pizza category ordered.

```sql
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 text	quantity numeric
1	Classic	14888
2	Supreme	11987
3	Veggie	11649
4	Chicken	11050

Determine the distribution of orders by hour of the day.

```sql
SELECT
    EXTRACT(HOUR FROM time::time) AS order_hour,
    COUNT(order_id) AS order_count
FROM
    orders
GROUP BY
    order_hour
ORDER BY
    order_hour;
```

	order_hour numeric	order_count bigint
1	9	1
2	10	8
3	11	1231
4	12	2520
5	13	2455
6	14	1472
7	15	1468
8	16	1920
9	17	2336
10	18	2399
11	19	2009
12	20	1642
13	21	1198
14	22	663

Join relevant tables to find the category-wise distribution of pizzas.

```sql
SELECT
    pizza_types.category,
    COUNT(DISTINCT pizzas.pizza_id) AS pizza_count
FROM
    pizzas
JOIN
    pizza_types
ON
    pizzas.pizza_type_id = pizza_types.pizza_type_id
GROUP BY
    pizza_types.category
ORDER BY
    pizza_count DESC;
```

	category text	pizza_count bigint
1	Veggie	27
2	Classic	26
3	Supreme	25
4	Chicken	18

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

```sql
SELECT
    orders.date,
    COUNT(order_details.order_id) AS total_pizzas_ordered,
    COUNT(DISTINCT orders.date) AS total_days,
    ROUND(COUNT(order_details.order_id) / COUNT(DISTINCT orders.date)::numeric, 2) AS avg_pizzas_per_day
FROM
    orders
JOIN
    order_details
ON
    orders.order_id = order_details.order_id
GROUP BY
    orders.date
ORDER BY
    orders.date;
```

date date	total_pizzas_ordered bigint	total_days bigint	avg_pizzas_per_day numeric
2015-01-01	161	1	161.00
2015-01-02	160	1	160.00
2015-01-03	154	1	154.00
2015-01-04	106	1	106.00
2015-01-05	121	1	121.00
2015-01-06	144	1	144.00
2015-01-07	133	1	133.00
2015-01-08	171	1	171.00
2015-01-09	123	1	123.00
2015-01-10	145	1	145.00
2015-01-11	114	1	114.00
2015-01-12	118	1	118.00
2015-01-13	117	1	117.00
2015-01-14	144	1	144.00
2015-01-15	123	1	123.00
2015-01-16	155	1	155.00
2015-01-17	122	1	122.00
2015-01-18	119	1	119.00
2015-01-19	139	1	139.00
2015-01-20	139	1	139.00
2015-01-21	127	1	127.00

Home About Contact

Determine the top 3 most ordered pizza types based on revenue.

```
    SELECT
    pizza_types.name AS pizza_type,
    SUM(order_details.quantity * pizzas.price) AS revenue
FROM
    order_details
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
    pizza_types.name
ORDER BY
    revenue DESC
LIMIT 3;
```

	pizza_type text	revenue double precision
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768
3	The California Chicken Pizza	41409.5

Analyze the cumulative revenue generated over time.

```
ITH order_revenue AS (
    SELECT
        orders.order_id,
        -- Combine date and time to form a full timestamp
        CONCAT(orders.date, ' ', orders.time) AS order_time,
        SUM(order_details.quantity * pizzas.price) AS revenue
    FROM
        order_details
    JOIN
        orders ON order_details.order_id = orders.order_id
    JOIN
        pizzas ON order_details.pizza_id = pizzas.pizza_id
    GROUP BY
        orders.order_id, orders.date, orders.time
),
umulative_revenue AS (
    SELECT
        order_time,
        revenue,
        SUM(revenue) OVER (ORDER BY order_time) AS cumulative_revenue
    FROM
        order_revenue

ELECT
    order_time,
    revenue,
    cumulative_revenue
ROM
    cumulative_revenue
RDER BY
    order_time;
```

	order_time text	revenue double precision	cumulative_revenue double precision
1	2015-01-01 11:38:36+05:30	13.25	13.25
2	2015-01-01 11:57:40+05:30	92	105.25
3	2015-01-01 12:12:28+05:30	37.25	142.5
4	2015-01-01 12:16:31+05:30	16.5	159
5	2015-01-01 12:21:30+05:30	16.5	175.5
6	2015-01-01 12:29:36+05:30	24.75	200.25
7	2015-01-01 12:50:37+05:30	12.5	212.75
8	2015-01-01 12:51:37+05:30	12.5	225.25
9	2015-01-01 12:52:01+05:30	143.25	368.5
10	2015-01-01 13:00:15+05:30	41	409.5
11	2015-01-01 13:02:59+05:30	73.5	483
12	2015-01-01 13:04:41+05:30	70.75	553.75
13	2015-01-01 13:11:55+05:30	20.25	574
14	2015-01-01 13:14:19+05:30	12	586
15	2015-01-01 13:33:00+05:30	63.25	649.25
16	2015-01-01 13:34:07+05:30	50.7	699.95
17	2015-01-01 13:53:00+05:30	184.5	884.45
18	2015-01-01 13:57:08+05:30	20.5	904.95
19	2015-01-01 13:59:09+05:30	40.75	945.7
20	2015-01-01 14:03:08+05:30	30.5	976.2
21	2015-01-01 14:14:29+05:30	20.5	996.7

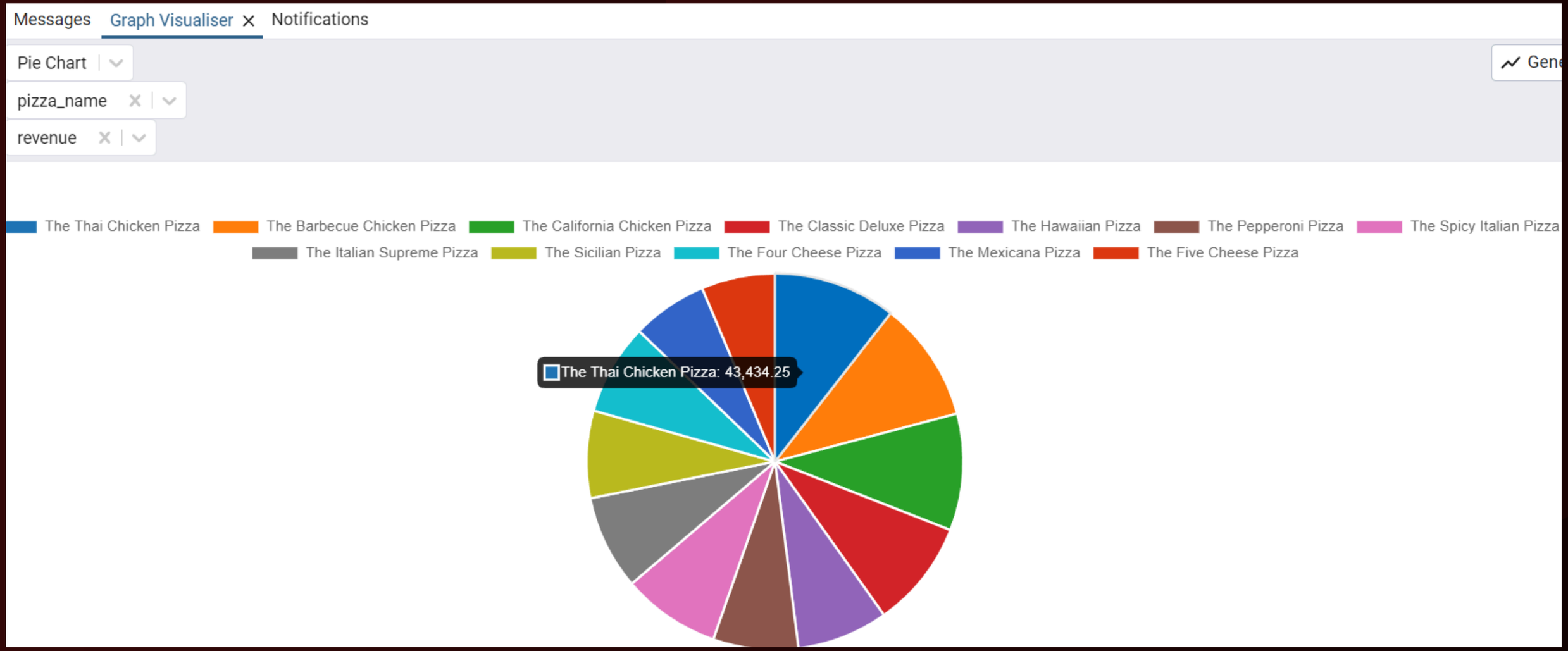
Determine the top 3 most ordered pizza types based on revenue for each pizza category.

```sql
WITH pizza_revenue AS (
    SELECT
        pizza_types.category,
        pizza_types.name AS pizza_name,  -- Correcting this to use pizza_types.name
        SUM(order_details.quantity * pizzas.price) AS revenue
    FROM
        order_details
    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
        pizza_types.category, pizza_types.name  -- Adjusted to use pizza_types.name
),
ranked_pizzas AS (
    SELECT
        category,
        pizza_name,
        revenue,
        ROW_NUMBER() OVER (PARTITION BY category ORDER BY revenue DESC) AS rank
    FROM
        pizza_revenue
)
SELECT
    category,
    pizza_name,
    revenue
FROM
    ranked_pizzas
WHERE
    rank <= 3
```

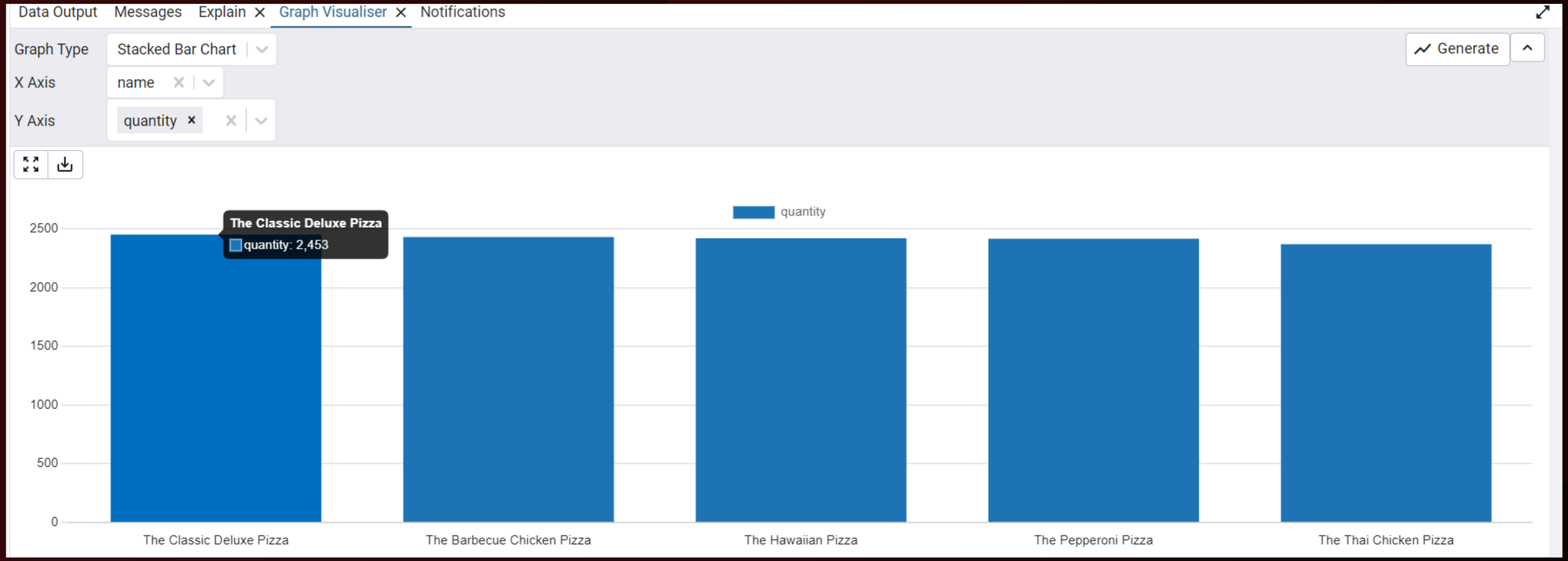
Data Output | Messages | Notifications

	category text	pizza_name text	revenue double precision
1	Chicken	The Thai Chicken Pizza	43434.25
2	Chicken	The Barbecue Chicken Pizza	42768
3	Chicken	The California Chicken Pizza	41409.5
4	Classic	The Classic Deluxe Pizza	38180.5
5	Classic	The Hawaiian Pizza	32273.25
6	Classic	The Pepperoni Pizza	30161.75
7	Supreme	The Spicy Italian Pizza	34831.25
8	Supreme	The Italian Supreme Pizza	33476.75
9	Supreme	The Sicilian Pizza	30940.5
10	Veggie	The Four Cheese Pizza	32265.70000000065
11	Veggie	The Mexicana Pizza	26780.75
12	Veggie	The Five Cheese Pizza	26066.5

Analyze the cumulative revenue generated over time.



List the top 5 most ordered pizza types along with their quantities



THANK YOU