SQL BASED CASE STUDY-PIZZA SALES PROJECT

Domain : Sales

Tools : SQL, Canva

A Year's worth of sales from a fictitious pizza place, including the date and time of each order and the pizzas served, with additional details on the type, size, quantity price and ingredients.

Recommended Analysis:

1. Retrieve the total number of orders placed.
2. Calculate the total revenue generated from pizza sales.
3. Identify the highest-priced pizza.
4. Identify the most common pizza size ordered.
5. List the top 5 most ordered pizza types along with their quantities.
6. Determine the top 3 most ordered pizza types based on revenue for each pizza category.
7. Determine the distribution of orders by hour of the day.
8. Join the necessary tables to find the total quantity of each pizza category ordered.
9. Join relevant tables to find the category-wise distribution of pizzas.
10. Group the orders by date and calculate the average number of pizzas ordered per day.
11. Determine the top 3 most ordered pizza types based on revenue.
12. Calculate the percentage contribution of each pizza type to total revenue.
13. Analyze the cumulative revenue generated over time.

Created By : Sanket Savaliya
sanketsavaliya0111@gmail.com

#1 Retrieve the total number of orders placed.

```sql
SELECT COUNT(order_id) AS total_orders
FROM orders;
```

	total_orders
▶	21350

#2 Calculate the total revenue generated from pizza sales.

```sql
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
▶	817860.05

#3 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	price
▶	The Greek Pizza	35.95

#4 Identify the most common pizza size ordered.

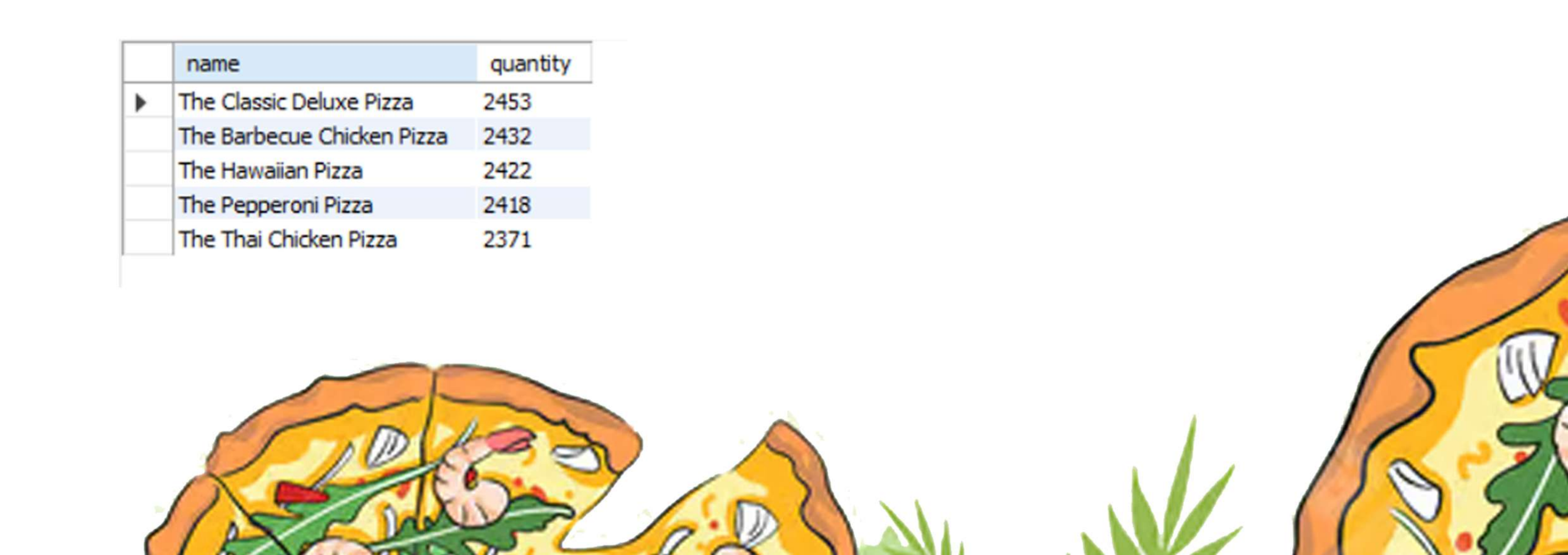
```sql
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC
LIMIT 1;
```

	size	order_count
▶	L	18526

#5 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	quantity
▶	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418
	The Thai Chicken Pizza	2371



#6 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 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;
```

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

#7 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
11	1231
12	2520
13	2455
14	1472

#8 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	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

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

```
select category, count(*) as pizza_types
from pizza_types
group by category
```

	category	pizza_types
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

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

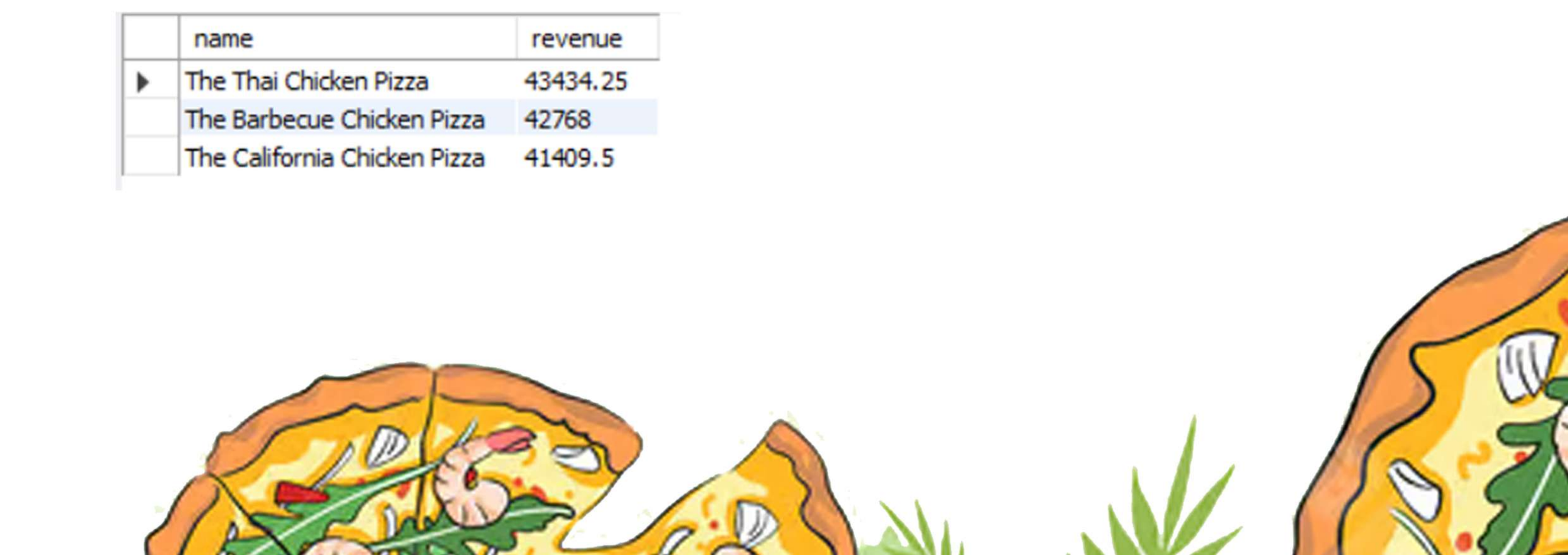
```
SELECT round(avg(quantity),0)as Avg_Pizza_Per_Day from
    (select orders.order_Date, SUM(order_details.quantity) as quantity
FROM
    orders
        JOIN
    order_details ON orders.order_id = order_details.order_id
GROUP BY orders.order_date) as order_quantity;
```

	Avg_Pizza_Per_Day
▶	138

#11 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 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
```

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



#12 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 pizzas.pizza_id = order_details.pizza_id) *100,2) 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 category order by revenue desc;
```

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

#13 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(order_details.quantity * pizzas.price)as revenue
from order_Details join pizzas
on order_details.pizza_id = pizzas.pizza_id
join orders
on orders.order_id = order_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