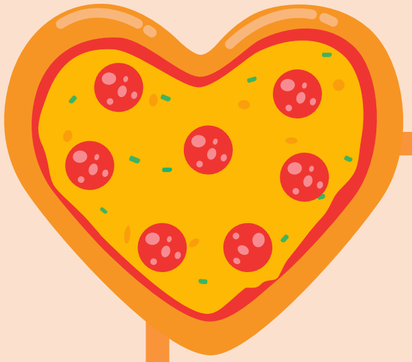
JULY 24, 2024

SQL PROJECT ON PIZZA SALES

Analyzing key metrics and trends of Pizza sales data

by Tanvi Sawant



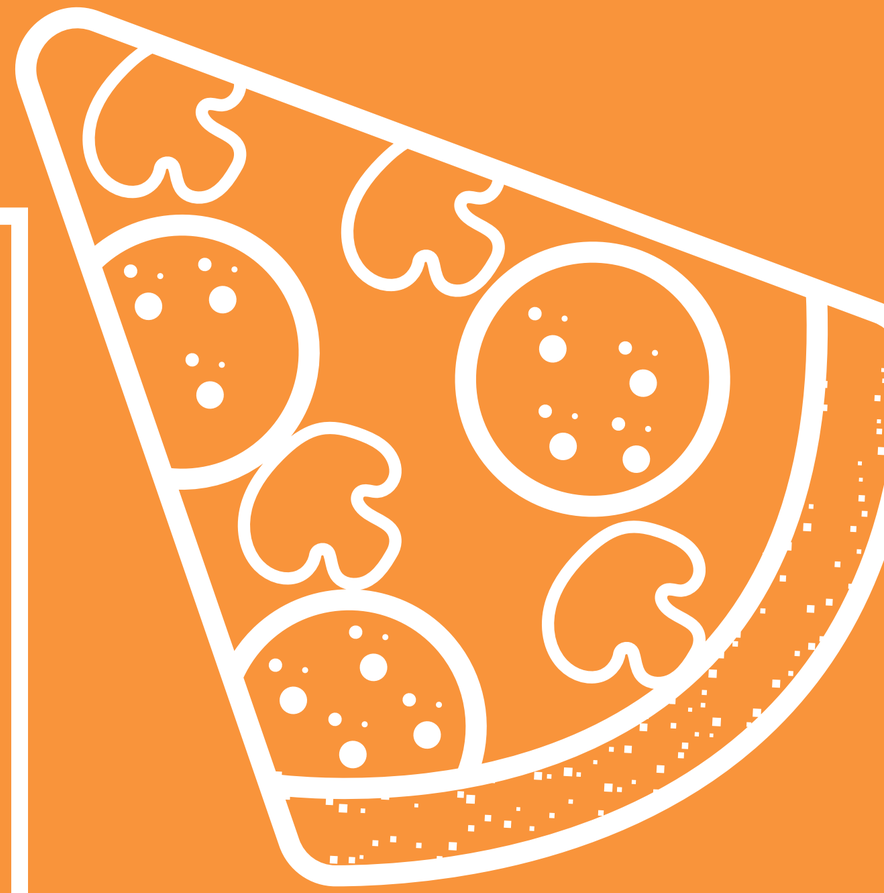
INTRODUCTION

- Welcome to the Pizza Sales SQL Analysis Project.
- This project delves into the sales data of a pizza business to uncover valuable insights.
- By examining orders, pizza types, sizes, and revenue, we aim to understand customer preferences and sales trends.

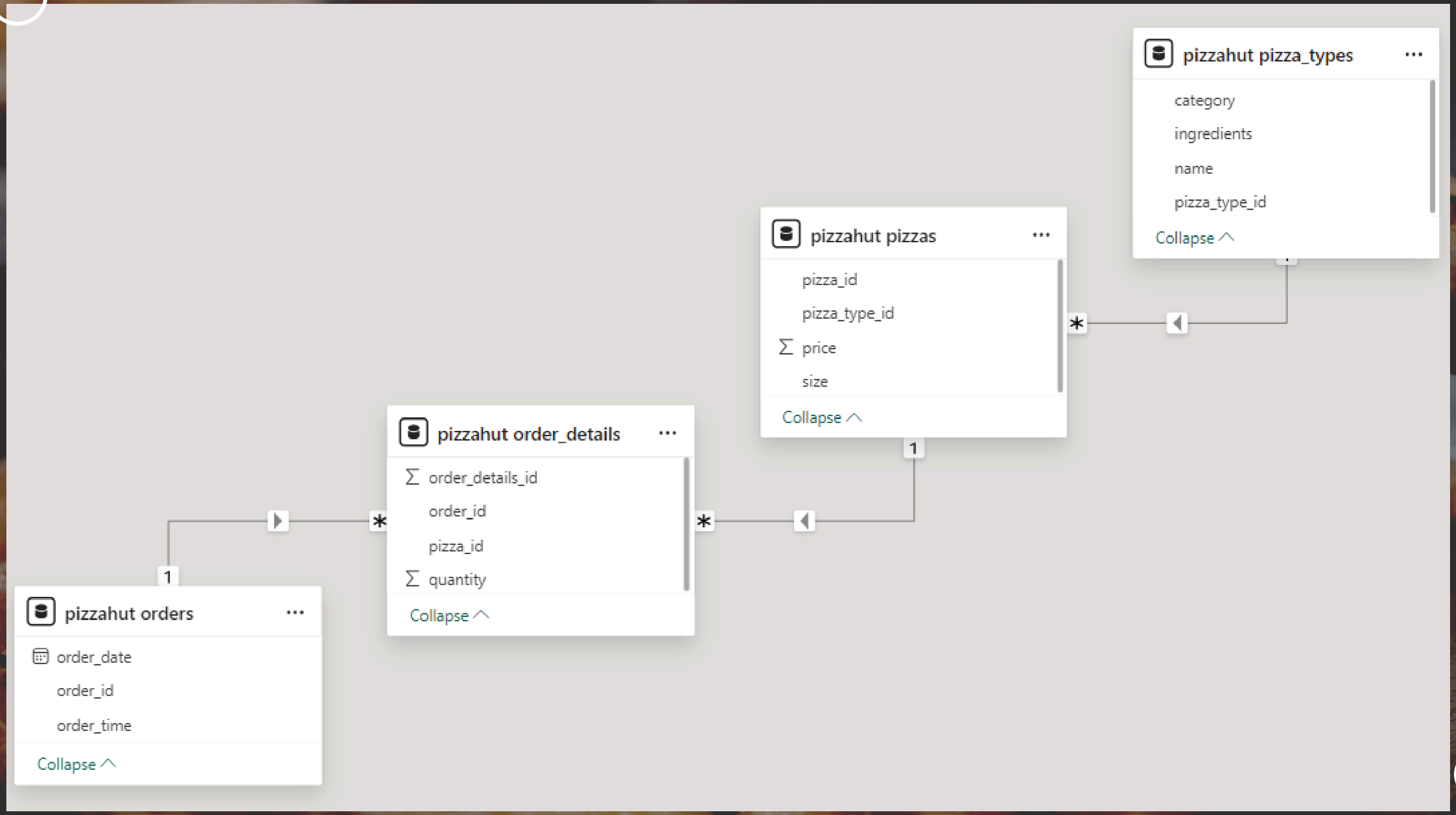
KEY TAKING POINTS

CONTENTS

- Project Overview: Analyzing pizza sales data for insights.
- Objectives: Identify trends, popular pizzas, and revenue distribution.
- Tools Used: MySQL for data analysis.

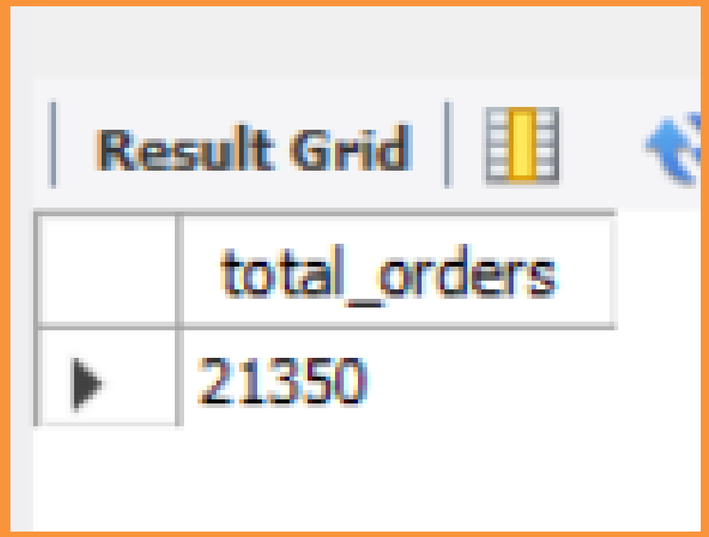


DATABASE SCHEMA



Retrieve the total number of orders placed

```sql
-- Q1. Retrieve the total number of orders placed

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



Calculate the total revenue generated from pizza sales

```sql
-- Q2. Calculate the total revenue generated from pizza sales

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

Result Grid

total_revenue
817860.05

Identify the highest-priced pizza

```sql
-- Q3. Identify the highest-priced pizza --

SELECT DISTINCT pt.*, p.price AS highest_price_pizza
FROM pizza_types AS pt
JOIN pizzas AS p
ON pt.pizza_type_id = p.pizza_type_id
ORDER BY p.price DESC
LIMIT 10;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

pizza_type_id	name	category	ingredients	highest_price_pizza
the_greek	The Greek Pizza	Classic	Kalamata Olives, Feta Cheese, Tomatoes, Garli...	35.95
the_greek	The Greek Pizza	Classic	Kalamata Olives, Feta Cheese, Tomatoes, Garli...	25.5
brie_carre	The Brie Carre Pizza	Supreme	Brie Carre Cheese, Prosciutto, Caramelized Oni...	23.65
ital_veggie	The Italian Vegetables Pizza	Veggie	Eggplant, Artichokes, Tomatoes, Zucchini, Red ...	21
bbq_ckn	The Barbecue Chicken Pizza	Chicken	Barbecued Chicken, Red Peppers, Green Peppe...	20.75
spinach_supr	The Spinach Supreme Pizza	Supreme	Spinach, Red Onions, Pepperoni, Tomatoes, Art...	20.75
ital_supr	The Italian Supreme Pizza	Supreme	Calabrese Salami, Capocollo, Tomatoes, Red O...	20.75
cali_ckn	The California Chicken Pizza	Chicken	Chicken, Artichoke, Spinach, Garlic, Jalapeno P...	20.75
thai_ckn	The Thai Chicken Pizza	Chicken	Chicken, Pineapple, Tomatoes, Red Peppers, T...	20.75
spin_pesto	The Spinach Pesto Pizza	Veggie	Spinach, Artichokes, Tomatoes, Sun-dried Toma...	20.75

Identify the most common pizza size ordered

```sql
-- Q4. Identify the most common pizza size ordered --

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

Result Grid

	size	orders
▶	L	18526

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

```sql
-- Q5. List the top 5 most ordered pizza types along with their quantities

SELECT pizza_types.name, SUM(order_details.quantity) AS quantity
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 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

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

```sql
-- Q6. Join the necessary tables to find the total quantity of each pizza category ordered

SELECT pizza_types.category, SUM(order_details.quantity) AS total_quantity
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.category
ORDER BY total_quantity DESC;
```

Result Grid | Filter Rows:

category	total_quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Determine the distribution of orders by hour of the day

```sql
-- Q7. Determine the distribution of orders by hour of the day

SELECT HOUR(order_time) AS hour_of_day, COUNT(order_id)
FROM orders
GROUP BY hour_of_day
ORDER BY hour_of_day ASC;
```

Result Grid | Filter Rows:

hour_of_day	COUNT(order_id)
9	1
10	8
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28

Retrieve the total number of orders placed

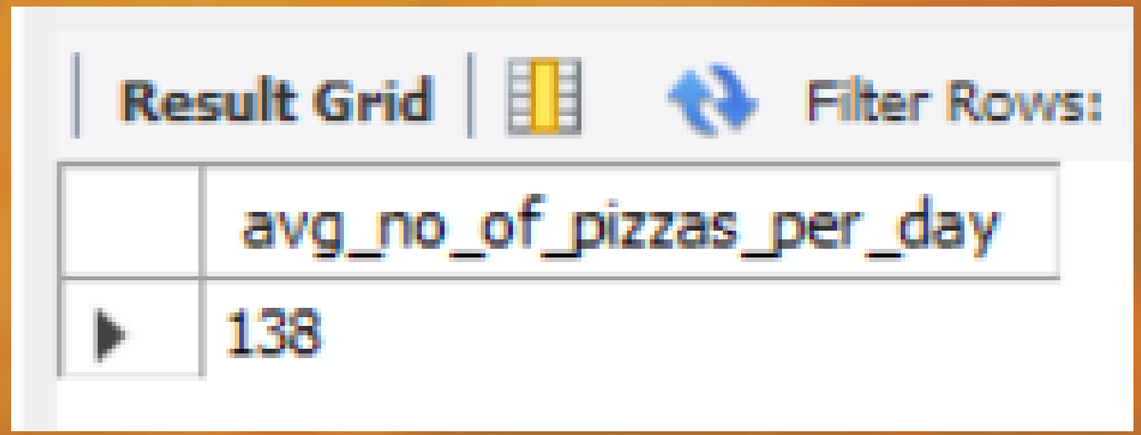
```sql
-- Q8. Join relevant tables to find the category-wise distribution of pizzas

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

Result Grid

category	pizzas
Chicken	6
Classic	8
Supreme	9
Veggie	9

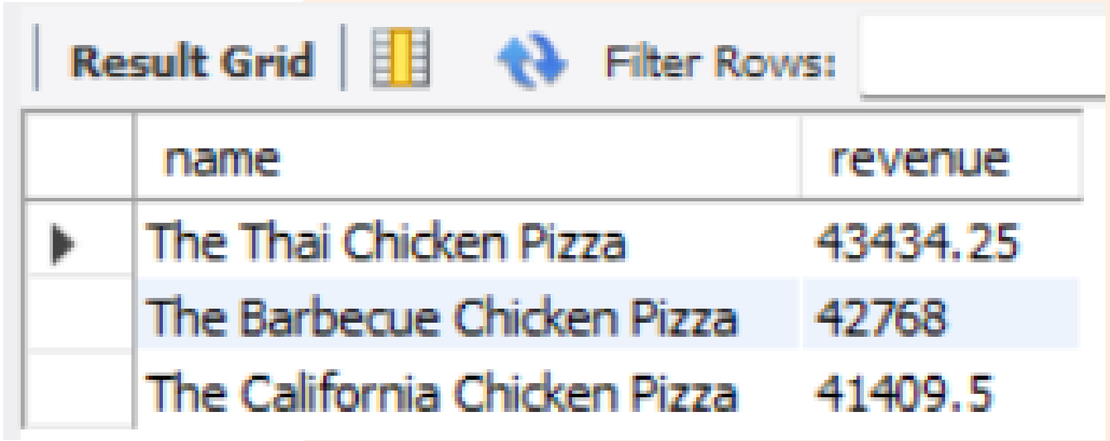
Group the orders by date and calculate the average number of pizzas ordered per day

```sql
-- Q9. Group the orders by date and calculate the average number of pizzas ordered per day

SELECT ROUND(AVG(total_quantity),0) AS avg_no_of_pizzas_per_day FROM
(SELECT orders.order_date, SUM(order_details.quantity) AS total_quantity
FROM orders
JOIN order_details
ON order_details.order_id = orders.order_id
GROUP BY orders.order_date
ORDER BY total_quantity DESC) AS order_quantity;
```



Determine the top 3 most ordered pizza types based on revenue

```sql
-- Q10. Determine the top 3 most ordered pizza types based on revenue --

SELECT pizza_types.name, SUM(pizzas.price * order_details.quantity) 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;
```



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

Calculate the percentage contribution of each pizza type to total revenue

```sql
-- Q.11 Calculate the percentage contribution of each pizza type to total revenue
SELECT pizza_types.category,
CONCAT(ROUND(SUM(order_details.quantity * pizzas.price) /
  (SELECT SUM(order_details.quantity * pizzas.price) AS total_sales
   FROM order_details
   JOIN pizzas
   ON pizzas.pizza_id = order_details.pizza_id
  )* 100, 2),"%") AS revenue_distribution
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.category;
```

Result Grid | Filter Rows:

category	revenue_distribution
Classic	26.91%
Veggie	23.68%
Supreme	25.46%
Chicken	23.96%

Analyze the cumulative revenue generated over time

```sql
-- Q.12 Analyze the cumulative revenue generated over time.

SELECT order_date, SUM(revenue) OVER(ORDER BY order_date) AS cumulative_revenue
FROM
(SELECT orders.order_date,
SUM(order_details.quantity * pizzas.price) AS revenue
FROM orders
JOIN order_details
ON order_details.order_id = orders.order_id
JOIN pizzas
ON order_details.pizza_id = pizzas.pizza_id
GROUP BY orders.order_date) AS sales;
```

Result Grid | Filter Rows:

order_date	cumulative_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	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.300000000003
2015-01-14	32358.700000000004
2015-01-15	34343.50000000001
2015-01-16	36937.65000000001
2015-01-17	39001.75000000001
2015-01-18	40978.600000000006
2015-01-19	43365.75000000001
2015-01-20	45763.65000000001
2015-01-21	47804.20000000001
2015-01-22	50300.90000000001
2015-01-23	52724.600000000006
2015-01-24	55013.850000000006

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

```sql
-- Q.13 Determine the top 3 most ordered pizza types based on revenue for each pizza category.
SELECT category, name, revenue
FROM
(SELECT category, name, revenue,
RANK() OVER(PARTITION BY category ORDER BY revenue DESC) AS top_ranks
FROM
(SELECT pizza_types.category, 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.category, pizza_types.name) as a)as b
WHERE top_ranks <= 3;
```

Result Grid | Filter Rows: | Export:

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
Classic	The Pepperoni Pizza	30161.75
Supreme	The Spicy Italian Pizza	34831.25
Supreme	The Italian Supreme Pizza	33476.75
Supreme	The Sicilian Pizza	30940.5
Veggie	The Four Cheese Pizza	32265.70000000065
Veggie	The Mexicana Pizza	26780.75
Veggie	The Five Cheese Pizza	26066.5


Thank you!