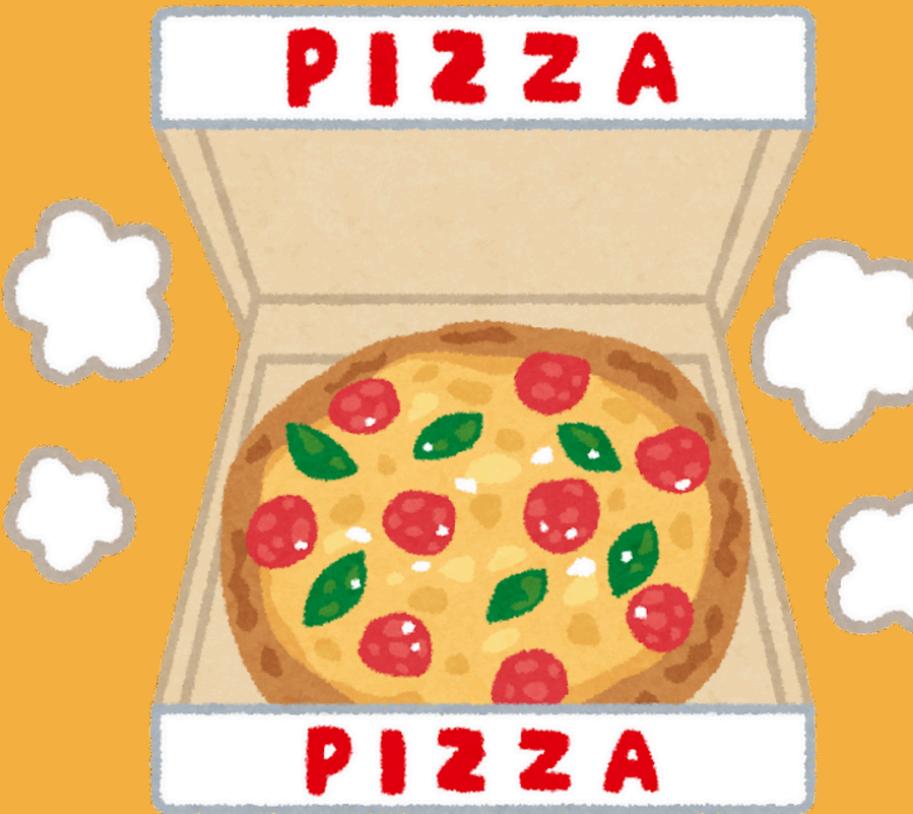
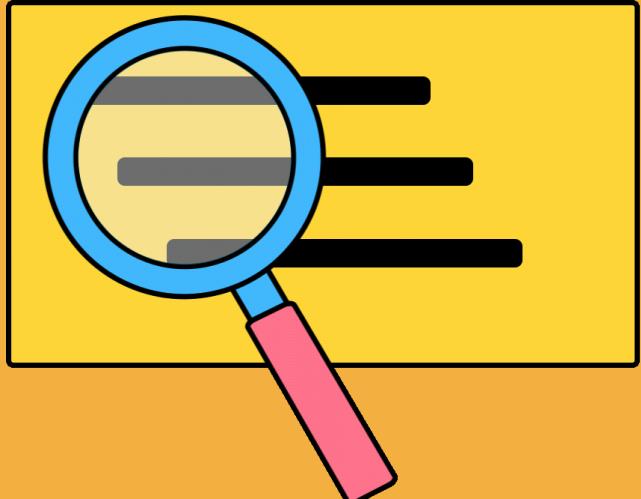


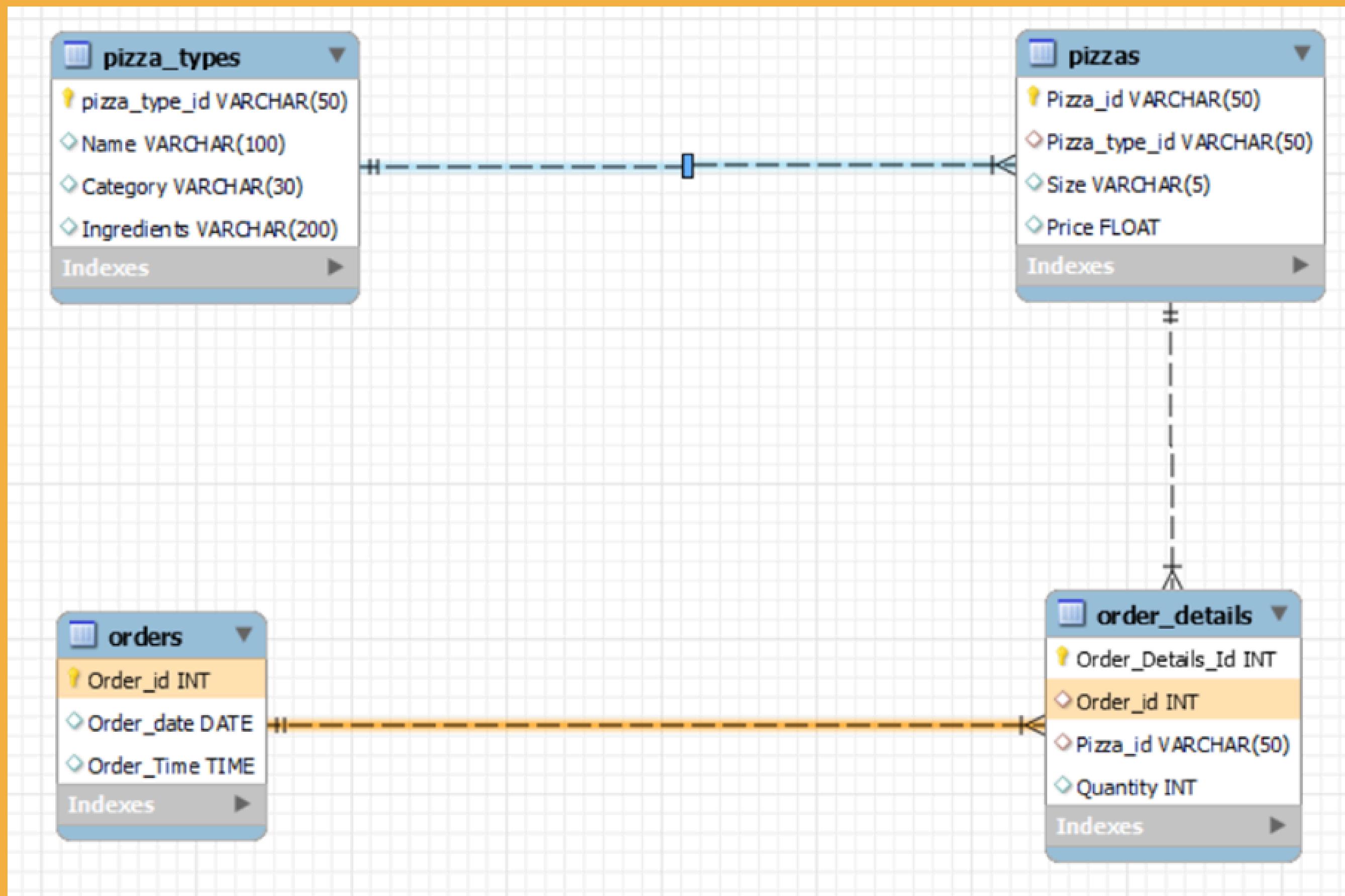
# PIZZAHUT SALES DATA ANAYSIS USING SQL





In this project I have utilized  
SQL Queries to solve the  
Business Problems of PizzaHut  
Sales

# Database Schema



# SQL Queries to Create DB, Tables(with Relations, Constraints) & Import Data



```
mysql> create database pizzaHut;
Query OK, 1 row affected (0.01 sec)
```

```
mysql> use pizzaHut;
Database changed
```

```
mysql> create table pizza_types(Pizza_type_id varchar(50) Primary Key,Name varchar(100),Category varchar(30),Ingredients varchar(200));
Query OK, 0 rows affected (0.10 sec)
```

```
mysql> create Table Order_Details(Order_Details_Id Int Primary Key, Order_id Int, Pizza_id varchar(50),Quantity Int, Foreign Key(Order_id) References Orders (Order_id), Foreign Key(Pizza_id) References Pizzas(Pizza_id));
Query OK, 0 rows affected (0.13 sec)
```

```
mysql> create Table Orders(Order_id Int Primary Key,Order_date Date, Order_Time Time);
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> create table pizzas(Pizza_id varchar(50) Primary Key,Pizza_type_id varchar(50),Size varchar(5),Price Float, Foreign Key(Pizza_type_id) References Pizza_types(Pizza_type_id));
Query OK, 0 rows affected (0.14 sec)
```

```
mysql> show tables;
+-----+
| Tables_in_pizzahut |
+-----+
| order_details      |
| orders             |
| pizza_types        |
| pizzas             |
+-----+
```

# Retrieve the total number of orders placed.



Query 1 x

1 -- Retrieve the total number of orders placed.  
2 • Select count(\*) as Total\_Orders from orders;  
3  
4

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

	Total_Orders
▶	21350

# Calculate the total revenue generated from pizza sales.



```
1 --- Calculate the total revenue generated from pizza sales.  
2  
3 • SELECT  
4     ROUND(SUM(Pizzas.Price * order_details.Quantity),  
5             2) AS Total_Revenue  
6 FROM  
7     pizzas  
8     INNER JOIN  
9     order_details ON pizzas.Pizza_id = order_details.Pizza_id;
```

Result Grid | Filter Rows: \_\_\_\_\_ | Export: | Wrap Cell Content: |

Total_Revenue
817860.05

# Calculate the total revenue generated from pizza sales.



```
2  
3 • SELECT  
4     ROUND(SUM(Pizzas.Price * order_details.Quantity),  
5             2) AS Total_Revenue  
6 FROM  
7 pizzas  
8 INNER JOIN  
9 order_details ON pizzas.Pizza_id = order_details.Pizza_id;
```

Result Grid | Filter Rows:

	Total_Revenue
▶	817860.05

# Identify the highest-priced pizza.



```
3 • SELECT
4      p.Pizza_id, pt.Name, p.Price
5 FROM
6 pizzas p
7     INNER JOIN
8 pizza_types pt ON p.Pizza_type_id = pt.pizza_type_id
9 ORDER BY Price DESC
10 LIMIT 1;
```

---

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

Pizza_id	Name	Price
the_greek_xxL	The Greek Pizza	35.95

# Identify the most common pizza size ordered.

```
3 • SELECT
4      p.size, COUNT(o.Order_Details_Id) AS Order_count
5  FROM
6      pizzas p
7      INNER JOIN
8          order_details o ON p.Pizza_id = o.Pizza_id
9  GROUP BY p.size
10 ORDER BY Order_count DESC;
```

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

size	Order_count
L	18526
M	15385
S	14137
XL	544
XXL	28



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



```
3 • SELECT
4     pizza_types.name, SUM(order_details.Quantity) AS Quantity
5 FROM
6     pizza_types
7         INNER JOIN
8     pizzas ON pizza_types.pizza_type_id = pizzas.Pizza_type_id
9         INNER JOIN
10    Order_details ON Pizzas.pizza_id = order_details.Pizza_id
11 GROUP BY pizza_types.name
12 ORDER BY Quantity DESC
13 LIMIT 5;
```

result Grid | Filter Rows:  Export: Wrap Cell Content: Fetch 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.

```
3 • SELECT
4     pt.Category, Sum(o.Quantity) as Total_Quantity
5 FROM
6     pizza_types pt
7         INNER JOIN
8     pizzas p ON pt.pizza_type_id = p.Pizza_type_id
9         INNER JOIN
10    order_details o ON p.Pizza_id = o.Pizza_id
11    Group by pt.Category
12    Order BY Total_Quantity Desc;
```

result Grid | Filter Rows:  Export: Wrap Cell Content:

Category	Total_Quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050



# Determine the distribution of orders by hour of the day.



- **SELECT**

```
HOUR(order_time) AS Hour,COUNT(Order_id) AS Total_orders  
FROM  
orders  
GROUP BY Hour  
Order By Total_orders ASC;
```

Result Grid | Filter Rows:

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

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



2 • SELECT

```
3     Category, count(Name) AS Pizza_count  
4 FROM pizza_types  
5 GROUP BY Category;
```

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

	Category	Pizza_count
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

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



```
2 • SELECT
3     Round(AVG(pizza_quantity),2) AS Average_pizza_Quantity
4 FROM
5     (SELECT
6         orders.order_date,
7             SUM(order_details.Quantity) AS pizza_quantity
8     FROM
9         orders
10    INNER JOIN order_details ON orders.Order_id = order_details.Order_
11    GROUP BY orders.order_date) as Order_quantity;
```

---

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

	Average_pizza_Quantity
▶	138.47

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



SELECT

```
pt.Name, SUM(p.price * o.quantity) AS Revenue  
FROM  
pizza_types pt  
    INNER JOIN  
pizzas p ON pt.pizza_type_id = p.Pizza_type_id  
    INNER JOIN  
order_details o ON p.Pizza_id = o.Pizza_id  
Group By pt.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

# Calculate the percentage contribution of each pizza type to total revenue.



SELECT

```
pt.category, Round(SUM(p.price * o.quantity) / (SELECT
ROUND(SUM(Pizzas.Price * order_details.Quantity),
2) AS Total_Revenue
FROM
pizzas
    INNER JOIN
order_details ON pizzas.Pizza_id = order_details.Pizza_id)*100,2) AS Revenue
FROM
pizza_types pt
    INNER JOIN
pizzas p ON pt.pizza_type_id = p.Pizza_type_id
    INNER JOIN
order_details o ON p.Pizza_id = o.Pizza_id
GROUP BY pt.category
ORDER BY Revenue DESC;
```

Result Grid | Filter

	category	Revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

# 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
    INNER JOIN
    pizzas ON order_details.Pizza_id = pizzas.Pizza_id
    INNER JOIN
    orders ON orders.Order_id = order_details.Order_id
  GROUP BY orders.Order_date) As Sales;
```

	order_date	Cum_Revenue
▶	2015-01-01	2713.8500022888184
	2015-01-02	5445.750003814697
	2015-01-03	8108.150007247925
	2015-01-04	9863.600008010864
	2015-01-05	11929.550008773804
	2015-01-06	14358.500011444092
	2015-01-07	16560.70001220703
	2015-01-08	19399.050018310547
	2015-01-09	21526.400022506714
	2015-01-10	23990.350025177002
	2015-01-11	25862.65002632141
	2015-01-12	27781.70002746582
	2015-01-13	29831.30002784729
	2015-01-14	32358.70002937317
	2015-01-15	34343.50003242493
	2015-01-16	36937.650033950806
	2015-01-17	39001.750034332275

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



```
Select Category, Name, Revenue, RN 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
    INNER JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.Pizza_type_id
    INNER 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;
```

	Category	Name	Revenue	RN
▶	Chicken	The Thai Chicken Pizza	43434.25	1
	Chicken	The Barbecue Chicken Pizza	42768	2
	Chicken	The California Chicken Pizza	41409.5	3
	Classic	The Classic Deluxe Pizza	38180.5	1
	Classic	The Hawaiian Pizza	32273.25	2
	Classic	The Pepperoni Pizza	30161.75	3
	Supreme	The Spicy Italian Pizza	34831.25	1
	Supreme	The Italian Supreme Pizza	33476.75	2
	Supreme	The Sicilian Pizza	30940.5	3
	Veggie	The Four Cheese Pizza	32265.70100402832	1
	Veggie	The Mexicana Pizza	26780.75	2
	Veggie	The Five Cheese Pizza	26066.5	3

# THANK YOU!

