



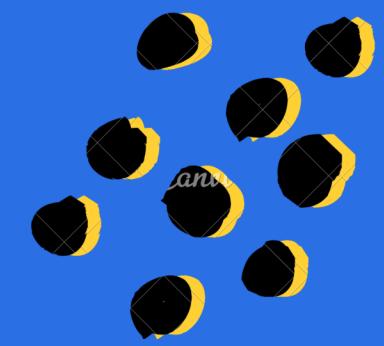
Delicious Pizza for Everyone!

# A SQL Project

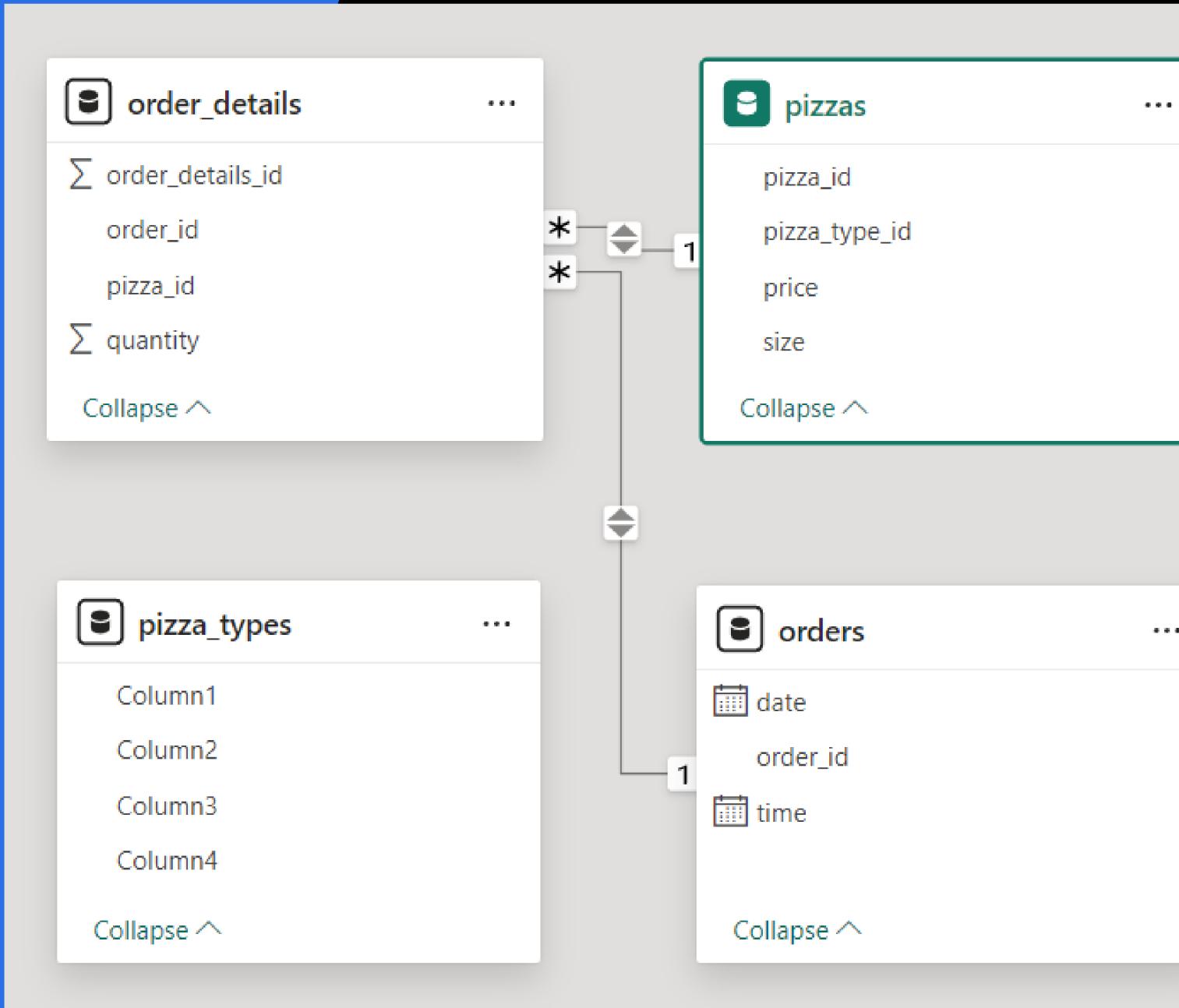
**“PIZZA HUT SALES ANALYSIS  
: UNVEILING INSIGHTS  
THROUGH SQL QUERIES”**

# Hello!

My name is kunal and in this project i have utilized SQL QUERIES to solve questions that were related to “pizzahut sales project”



AFTER ANALYZING THE COMPLETE DATA I IMPORTED IN POWER BI AND CREATE RELATION BETWEEN ALL TABLES HOW IT LOOKS IN POWER BI MODEL VIEW



### Basic:

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.

### Intermediate:

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.

### Advanced:

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

Analyze the cumulative revenue generated over time.

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

# Retrieve the total number of orders placed.

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

Result Grid	
	TOTAL_ORDERS
▶	21350



# Calculate the total revenue generated from pizza sales.

```
• SELECT  
    ROUND(SUM(ORDERS_DETAILS.QUANTITY * PIZZAS.PRICE),  
          2) AS TOTAL_REVENUE  
FROM  
    ORDERS_DETAILS  
JOIN  
    PIZZAS ON ORDERS_DETAILS.PIZZA_ID = PIZZAS.PIZZA_ID;
```



Result Grid	
	TOTAL_REVENUE
▶	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  
LIMIT 1;
```



	Result Grid			Filter Rows:
	SIZE	ORDER_COUNT		
▶	L	18526		

# List the 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;
```

	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.

- **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 Rows:

	CATEGORY	QUANTITY
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

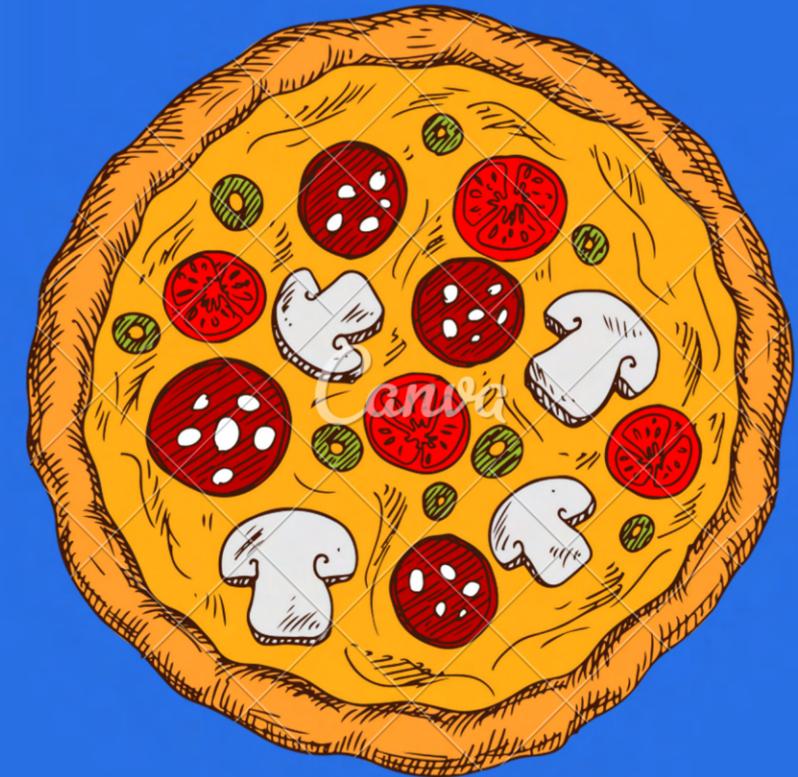
# DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

- SELECT

```
HOUR(ORDER_TIME) AS HOURS, COUNT(ORDER_ID) AS ORDER_COUNT  
FROM  
ORDERS  
GROUP BY HOUR(ORDER_TIME);
```

Result Grid | Filter Rows:

	HOURS	ORDER_COUNT
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920



# GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.

```
SELECT  
    ROUND(AVG(QUANTITY),0) AS AVERAGE_PIZZAS_PERDAY  
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;
```

	AVERAGE_PIZZAS_PERDAY
▶	138



# DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

- **SELECT**

```
PIZZA_TYPES.NAME,  
ROUND(SUM(ORDERS_DETAILS.QUANTITY * PIZZAS.PRICE),0) AS REVANUE  
FROM  
ORDERS_DETAILS  
JOIN  
PIZZAS ON ORDERS_DETAILS.PIZZA_ID = PIZZAS.PIZZA_ID  
JOIN  
PIZZA_TYPES ON PIZZA_TYPES.PIZZA_TYPE_ID = PIZZAS.PIZZA_TYPE_ID  
GROUP BY PIZZA_TYPES.NAME  
ORDER BY REVANUE DESC  
LIMIT 3;
```



	NAME	REVANUE
▶	The Thai Chicken Pizza	43434
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41410

# CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

- **SELECT**

```
PIZZA_TYPES.CATEGORY,  
ROUND(SUM(ORDERS_DETAILS.QUANTITY * PIZZAS.PRICE) / (SELECT  
ROUND(SUM(ORDERS_DETAILS.QUANTITY * PIZZAS.PRICE),  
2) AS TOTAL_SALES  
FROM  
ORDERS_DETAILS  
JOIN  
PIZZAS ON ORDERS_DETAILS.PIZZA_ID = PIZZAS.PIZZA_ID) * 100,  
2) AS REVENUE  
FROM  
ORDERS_DETAILS  
JOIN  
PIZZAS ON ORDERS_DETAILS.PIZZA_ID = PIZZAS.PIZZA_ID  
JOIN  
PIZZA_TYPES ON PIZZA_TYPES.PIZZA_TYPE_ID = PIZZAS.PIZZA_TYPE_ID  
GROUP BY PIZZA_TYPES.CATEGORY  
ORDER BY REVENUE DESC;
```



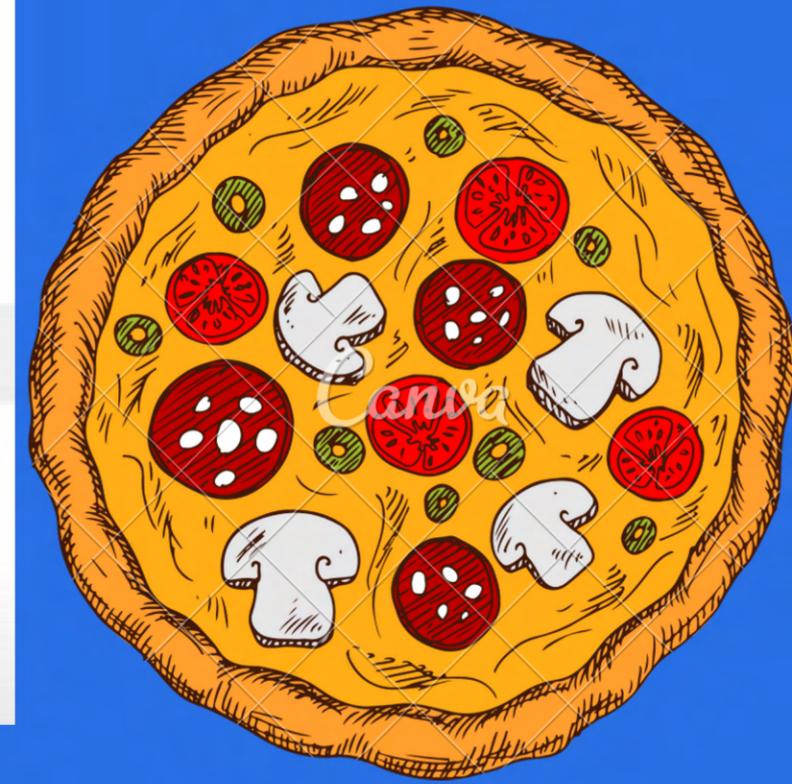
Result Grid | Filter Rows:

	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 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;
```

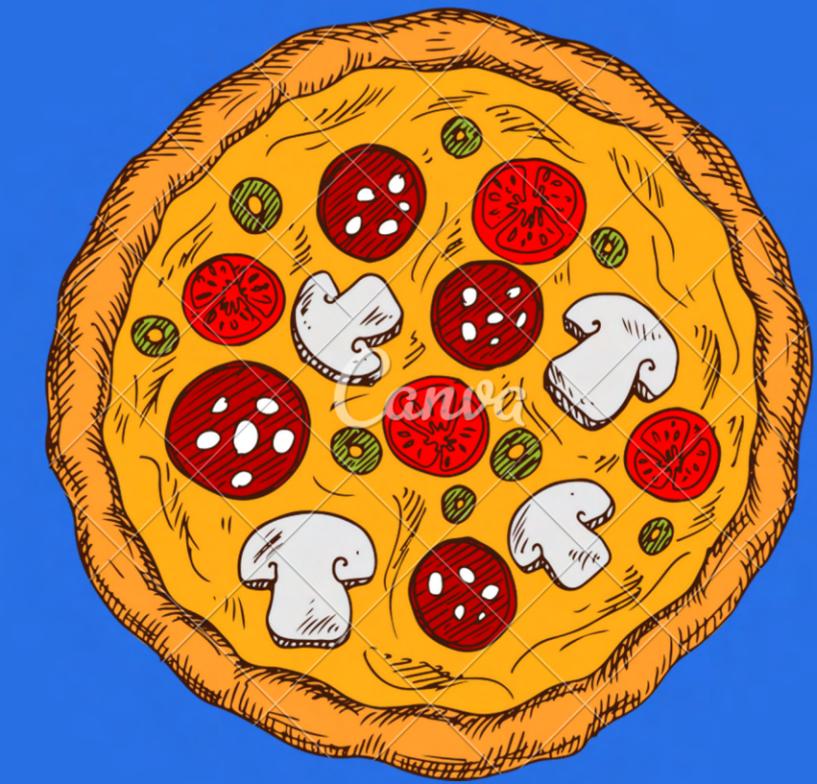
	order_date	cum_revenue
▶	2015-01-01	2713.850000000004
	2015-01-02	5445.75
	2015-01-03	8108.15
	2015-01-04	9863.6
	2015-01-05	11929.55
	2015-01-06	14358.5

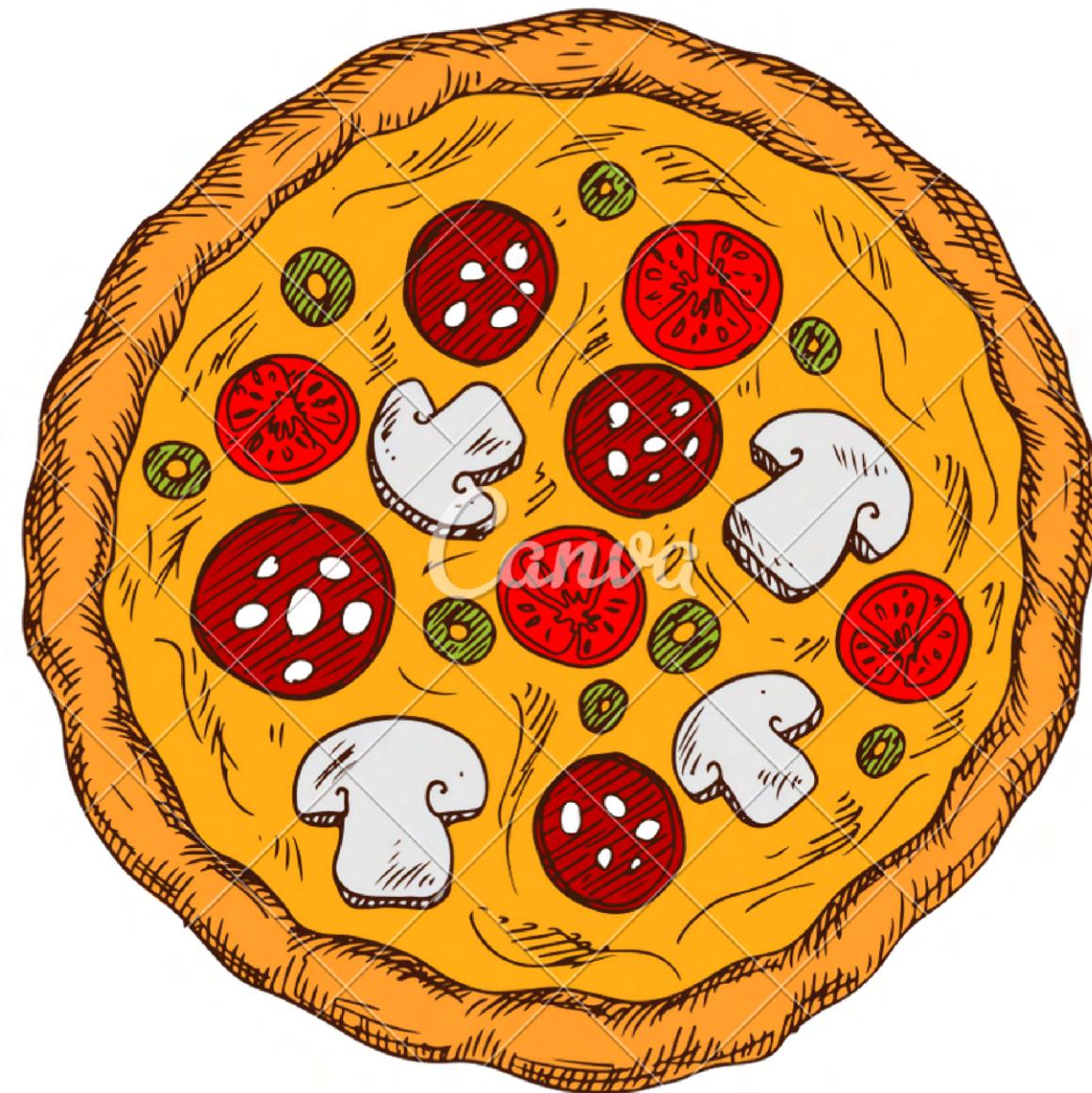


# DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

```
• 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





Pizza Boxcar Present

**thank  
you**

@reallygreatsite