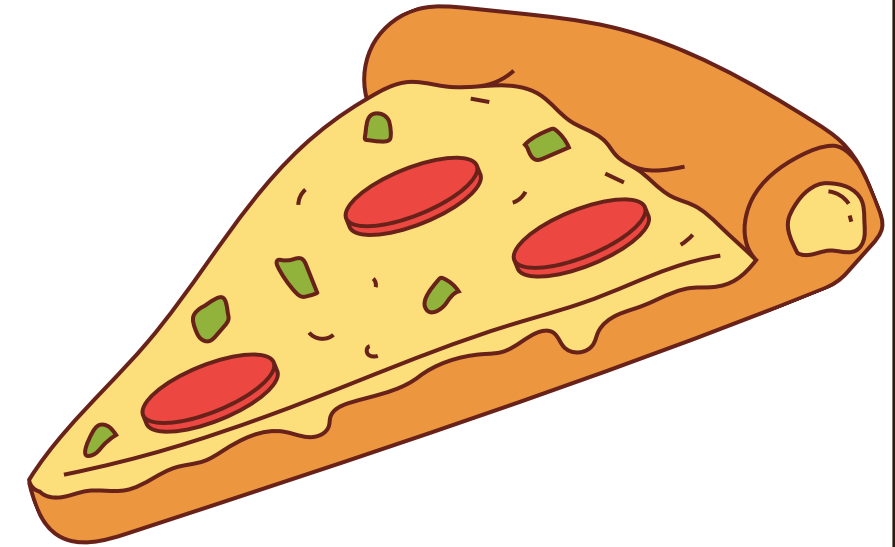
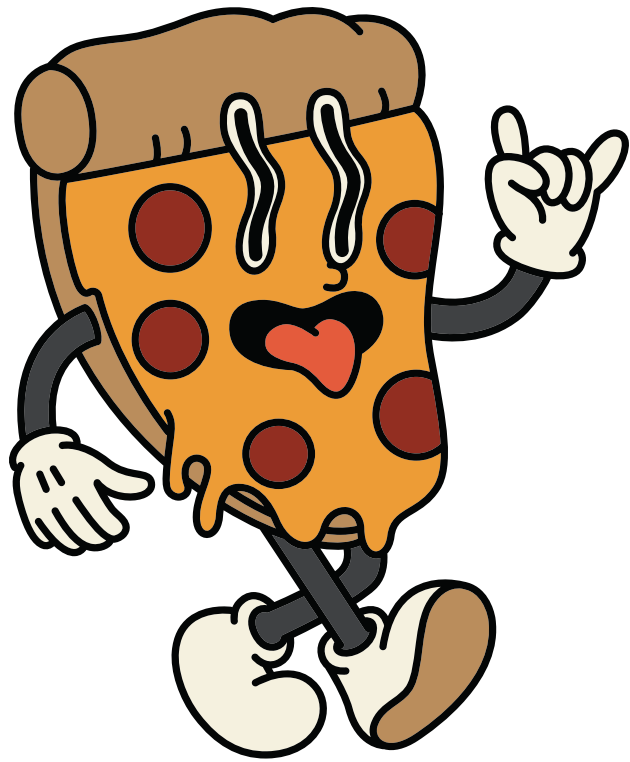


PIZZA PARTY WITH SQL



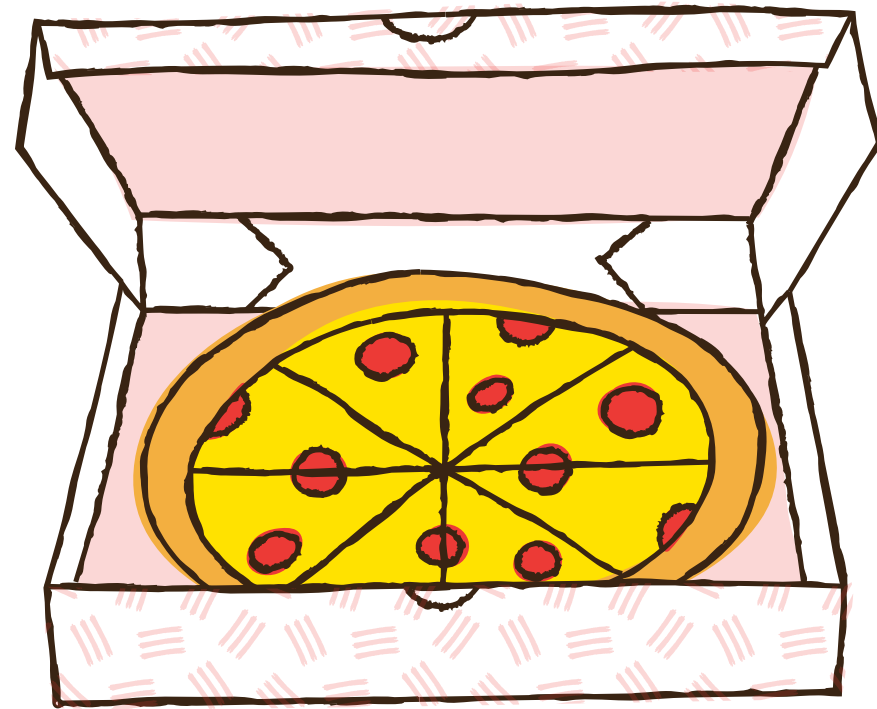
HELLO!



Greetings, I'm Md Ashraf, and I'm excited to share how SQL queries provided invaluable business insights into Pizza Hut's sales.

Through SQL analysis, we gained deep insights into Pizza Hut's sales data, enabling us to make informed business decisions and drive growth.

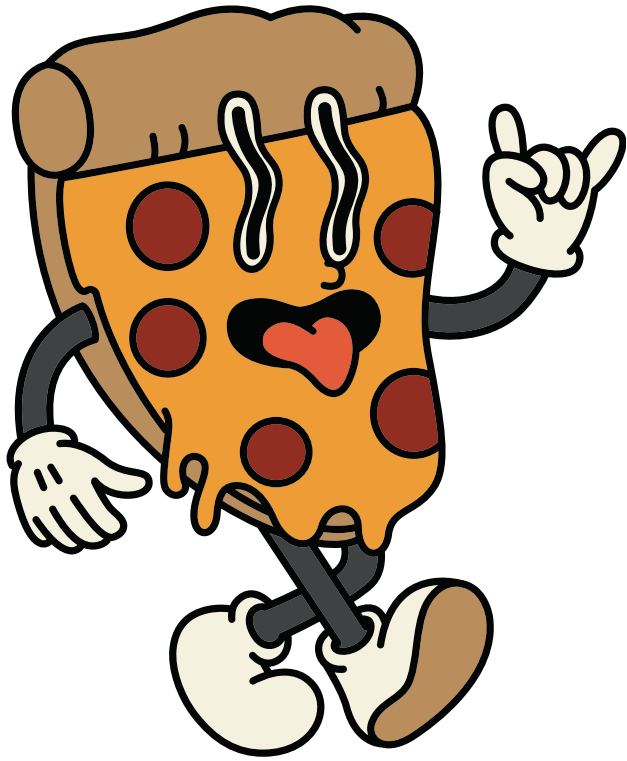
-- Retrieve the total number of orders placed.



```
select count(order_id)  
as Total_Orders from  
orders;
```

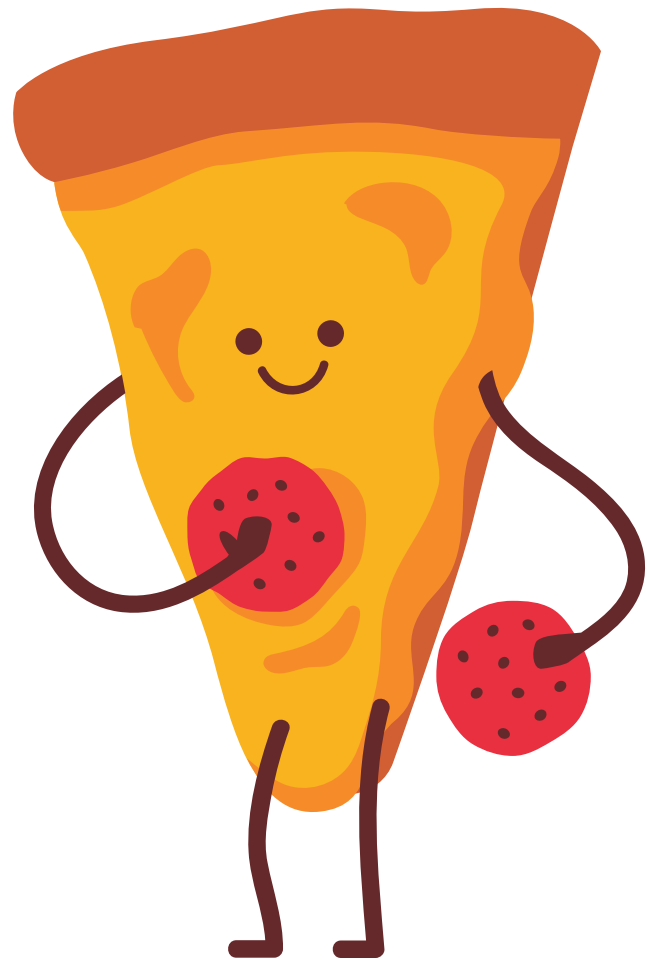
Result Grid		Filter Rows:	
	Total_Orders		
▶	21350		

-- Calculate the total revenue generated from
pizza sales.



```
SELECT  
ROUND(SUM(order_details.quantity *  
pizzasp.price),  
2) AS Total_Revenue  
FROM  
order_details  
JOIN  
pizzasp ON pizzasp.pizza_id =  
order_details.pizza_id;
```

Result Grid	
Total_Revenue	
▶	817860.05



-- Identify the highest-priced pizza.

```
SELECT
pizza_types.name, pizzasp.price
  AS Highest_Price
FROM
  pizza_types
  INNER JOIN
    pizzasp ON
      pizza_types.pizza_type_id =
        pizzasp.pizza_type_id
ORDER BY pizzasp.price DESC
LIMIT 1
;
```

Result Grid			Filter Rows:
	name	Highest_Price	
▶	The Greek Pizza	35.95	

-- Identify the most common pizza size ordered.





```
SELECT
pizzasp.size AS Most_common_pizza_size,
COUNT(order_details.order_details_id) AS
order_count
FROM
pizzasp
INNER JOIN
order_details ON pizzasp.pizza_id =
order_details.pizza_id
GROUP BY pizzasp.size
ORDER BY order_count DESC;
```

Result Grid			Filter Rows:
	Most_common_pizza_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.





Result Grid   Filter Rows: <input type="text"/>	
Pizza_name	total_quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

```
SELECT
    pizza_types.name AS Pizza_name,
    SUM(order_details.quantity) AS total_quantity
FROM
    pizza_types
INNER JOIN
    pizzasp ON pizza_types.pizza_type_id =
              pizzasp.pizza_type_id
INNER JOIN
    order_details ON pizzasp.pizza_id =
                    order_details.pizza_id
GROUP BY Pizza_name
ORDER BY total_quantity DESC
LIMIT 5;
```



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



Result Grid   Filter Rows:	
Pizza_category	total_quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

```
SELECT
    pizza_types.category AS Pizza_category,
    SUM(order_details.quantity) AS total_quantity
FROM
    pizza_types
    INNER JOIN
    pizzasp ON pizza_types.pizza_type_id = pizzasp.pizza_type_id
    INNER JOIN
    order_details ON pizzasp.pizza_id = order_details.pizza_id
GROUP BY Pizza_category
ORDER BY total_quantity DESC;
```


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

Result Grid |   Filter Rows:

	hour	order_count
▶	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
	10	8
	9	1



```
SELECT
  HOUR(order_time) AS hour, COUNT(order_id) AS
    order_count
  FROM
    orders
  GROUP BY hour;
```

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

Result Grid			Filter Row
	category	count(name)	
▶	Chicken	6	
	Classic	8	
	Supreme	9	
	Veggie	9	

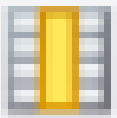

```
select category ,count(name)
from pizza_types
group by category;
```



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



```
SELECT
    ROUND(AVG(Quantity), 0) AS
    Average_no_of_pizza_ordered_per_day
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) AS
        Quantity
    FROM
        order_details
    INNER JOIN orders USING (order_id)
    GROUP BY orders.order_date) AS order_quantity;
```

Result Grid   Filter Rows: <input data-bbox="1239 1271 1539 1410" type="text"/>	
Average_no_of_pizza_ordered_per_day	
138	

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



```
select pizza_types.name as pizza_name,  
sum(order_details.quantity * pizzasp.price) as Revenue  
from pizza_types inner join pizzasp  
using(pizza_type_id) inner join  
order_details using(pizza_id) group by pizza_name  
order by Revenue desc limit 3;
```

pizza_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.





	category	Revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

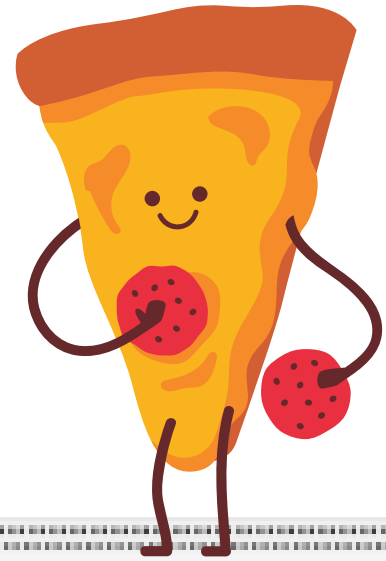
```
SELECT
    pizza_types.category,
    ROUND((SUM(order_details.quantity * pizzasp.price) / (SELECT
        ROUND(SUM(order_details.quantity * pizzasp.price),
            2) AS Total_Revenue
        FROM
            order_details
        JOIN
            pizzasp ON pizzasp.pizza_id = order_details.pizza_id)) * 100,
        2) AS Revenue
    FROM
        pizza_types
    INNER JOIN
        pizzasp USING (pizza_type_id)
    INNER JOIN
        order_details USING (pizza_id)
    GROUP BY pizza_types.category
    ORDER BY Revenue DESC;
```

-- Analyze the cumulative revenue generated over time.




Result Grid   Filter Rows		
	order_date	cum_revenue
▶	2015-01-01	2713.85000000
	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	23990.3500000
	2015-01-11	25862.65
	2015-01-12	27781.7
	2015-01-13	29831.3000000
	2015-01-14	32358.7000000
	2015-01-15	34343.5000000
	2015-01-16	36937.6500000


```
select order_date,sum(Revenue) over(order by order_date) as
        cum_revenue
        from
        (select orders.order_date, sum(order_details.quantity *
        pizzasp.price) as Revenue
        from orders inner join order_details using(order_id) inner
        join pizzasp using(pizza_id)
        group by orders.order_date) as Sales ;
```



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

result Grid





Filter Rows:

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
The Four Cheese Pizza	32265.7000
The Mexicana Pizza	26780.75
The Five Cheese Pizza	26066.5

```
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 * pizzasp.price) as revenue
from pizza_types inner join pizzasp using(pizza_type_id)
inner join
order_details using(pizza_id)
group by pizza_types.category,pizza_types.name) as a) as b
where rn <=3;
```

PIZZA PARTY END!

THANK YOU

