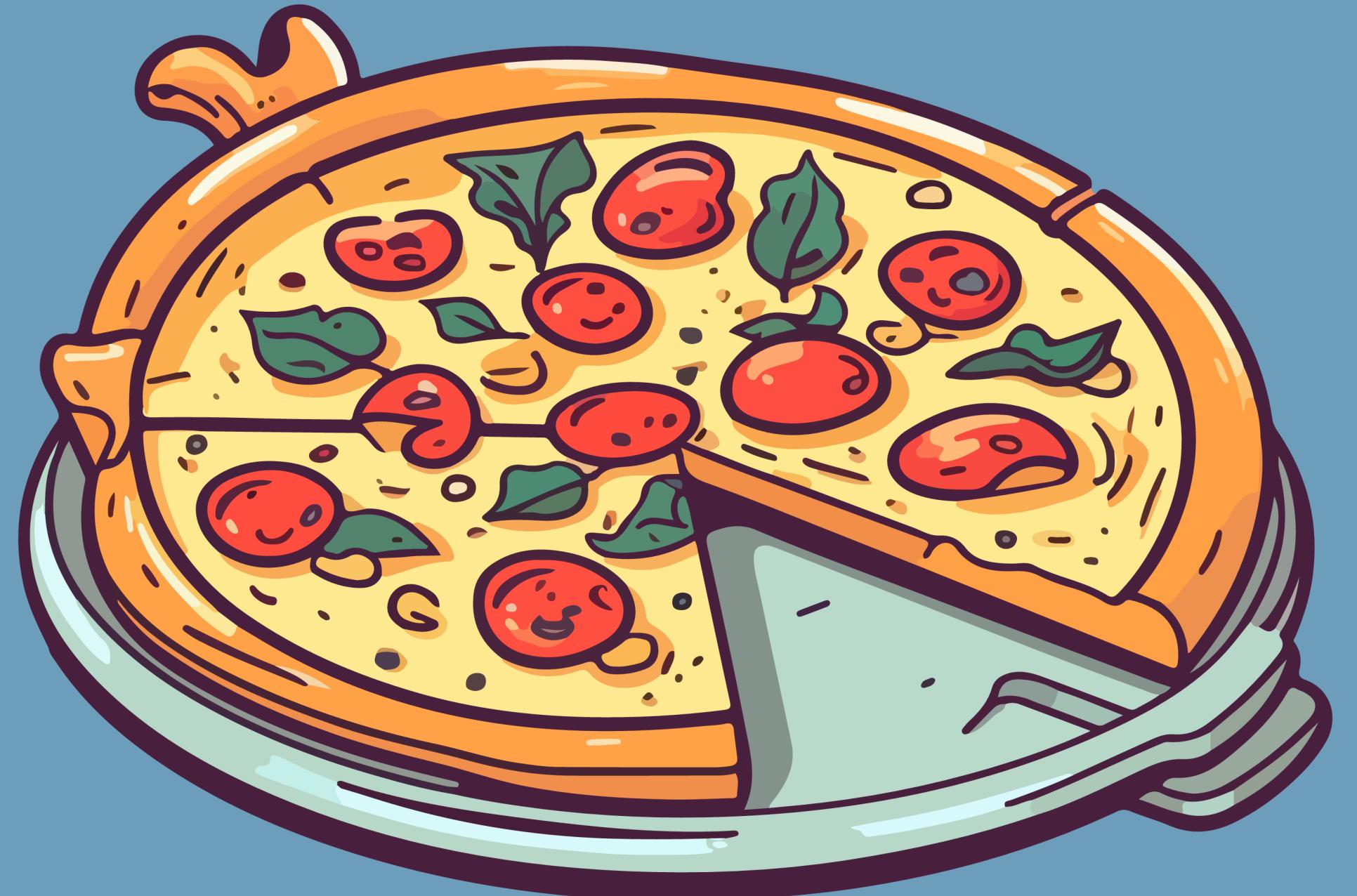


# PIZZA\_HUT

By SQL



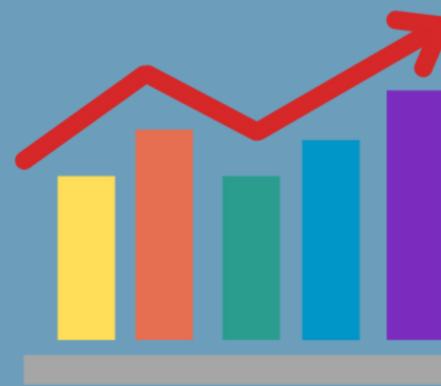
# ANALYZING SALES TRENDS FOR BUSINESS GROWTH



## PURPOSE

- ★ Show how SQL-driven data analysis supports decision-making and growth for Pizza Hut.
- ★ To analyze Pizza Hut's sales data to uncover trends and patterns using SQL.
- ★ Identify best-selling products and peak sales periods.

# SCHEMA



Info	Tables	Columns	Indexes	Triggers	Views	Stored Procedures	Functions	Grants	Events		
Table		Column		Type		Default Value		Nullable	Character Set	Collation	Privileges
order_details		◆ order_detaial_id		int				NO			select,insert,update,references
order_details		◆ order_id		int				NO			select,insert,update,references
order_details		◆ pizza_id		varchar(255)				YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references
order_details		◆ Quantity		int				NO			select,insert,update,references
orders		◆ order_id		int				NO			select,insert,update,references
orders		◆ Order_date		date				NO			select,insert,update,references
orders		◆ order_time		time				NO			select,insert,update,references
pizza_types		◆ pizza_type_id		text				YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references
pizza_types		◆ name		text				YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references
pizza_types		◆ category		text				YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references
pizza_types		◆ ingredients		text				YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references
pizzas		◆ pizza_id		text				YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references
pizzas		◆ pizza_type_id		text				YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references
pizzas		◆ size		text				YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references
pizzas		◆ price		double				YES			select,insert,update,references

# START ANALYSIS QUESTIONS



**TO EXTRACT VALUABLE INSIGHTS,  
WE FOCUS ON KEY QUESTIONS THAT ADDRESS  
PIZZA HUT'S SALES TRENDS AND BUSINESS PERFORMANCE**

1

-- CALCULATE THE TOTAL REVENUE  
GENERATED FROM PIZZA SALES.

## Query

```
32
33 • SELECT
34     ROUND(SUM(order_details.Quantity * pizzas.price),
35             2) AS total_sales
36
37     FROM
38         order_details
39     JOIN
40         pizzas ON order_details.pizza_id = pizzas.pizza_id;
```

## Result

Result Grid		Filter Row
	total_sales	
▶	817860.05	

# 2 -- LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

## Query

```
66 •   SELECT
67       pizza_types.name, SUM(order_details.Quantity) AS count_order
68   FROM
69       pizza_types
70   JOIN
71       pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
72   JOIN
73       order_details ON order_details.pizza_id = pizzas.pizza_id
74   GROUP BY pizza_types.name
75   ORDER BY count_order DESC
76   LIMIT 5;
```

## Result

	name	count_order
▶	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418
	The Thai Chicken Pizza	2371

# 3 -- JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

## Query

```
80
81 •   SELECT
82     pizza_types.category,
83     SUM(order_details.Quantity) AS Category_count
84   FROM
85     pizza_types
86       JOIN
87     pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
88       JOIN
89     order_details ON order_details.pizza_id = pizzas.pizza_id
90   GROUP BY pizza_types.category
91   ORDER BY Category_count DESC;
92
```

## Result

	category	Category_count
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050
...		

## 4

-- ANALYZE THE CUMULATIVE REVENUE  
GENERATED OVER TIME.

## Query

```
151 •   SELECT order_date,  
152     SUM(revenue) OVER(Order BY order_date) AS cum_revenue  
153   FROM  
154   ( SELECT orders.order_date,  
155     Round(sum(order_details.Quantity * pizzas.price), 2) As revenue  
156   From order_details JOIN pizzas  
157     ON order_details.pizza_id = pizzas.pizza_id  
158   JOIN orders  
159     ON order_details.order_id = orders.order_id  
160   Group BY orders.order_date)  
161   AS Sales;  
162
```

## Result

Result Grid		Filter Rows:
	order_date	cum_revenue
▶	2015-01-01	2713.85
	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	21526.39999999998
	2015-01-10	23990.35
	2015-01-11	25862.64999999998
	2015-01-12	27781.69999999997
	2015-01-13	29831.29999999996
	2015-01-14	32358.69999999997
	2015-01-15	34343.5
	2015-01-16	36937.65
	2015-01-17	39001.75
	2015-01-18	40978.6
	2015-01-19	43365.75

## 5

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

## Query

```
165
166
167 •   select name, revenue from
168   (select category, name, revenue,
169    rank() over(partition by category order by revenue desc ) as revenue_pizza
170   from
171   (SELECT pizza_types.category, pizza_types.name ,
172    sum( order_details.Quantity * pizzas.price ) As revenue
173    FROM order_details Join pizzas
174    ON order_details.pizza_id = pizzas.pizza_id
175    JOIN pizza_types
176    ON
177    pizza_types.pizza_type_id = pizzas.pizza_type_id
178    Group By pizza_types.category, pizza_types.name) as a) as b
179    Where revenue_pizza <=3 ;
180  |
```

## Result

	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	32265.70000000065
	The Mexicana Pizza	26780.75
	The Five Cheese Pizza	26066.5

# THANK YOU!!

