PIZZA HUT SQL PROJECT

Home About Contact

- WHERE EVERY SLICE TELLS A STORY

PROJECT INTRODUCTION

This SQL project analyzes Pizza Hut sales data to uncover insights on orders, revenue, best-selling pizzas, category trends, and cumulative sales performance using joins, aggregations, and window functions.

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```
select count(order_id) As total_order
  from orders;
```

Result Grid

	total_order
▶	2746

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.


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

Result Grid

	total_sales
▶	305379

IDENTIFY THE HIGHEST-PRICED PIZZA.

```sql
select pt. name,p.price
from pizza_types as pt
join pizzas as p
on pt.pizza_type_id=p.pizza_type_id
order by p.price desc
limit 1 ;
```

Result Grid | Filter Rows:

	name	price
▶	The Barbecue Chicken Pizza	20.75

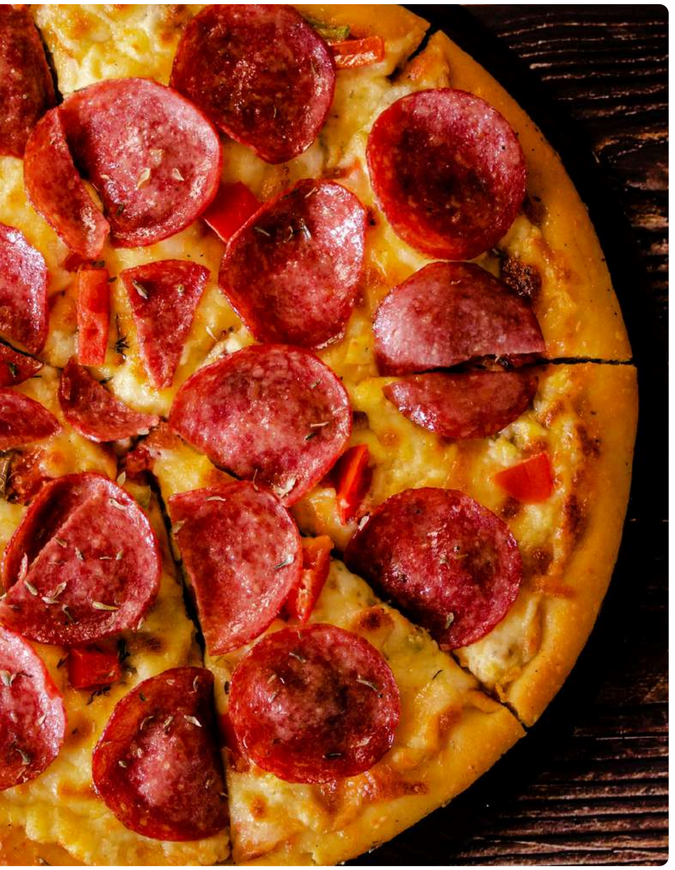
LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.


```sql
select pt.name,sum(od.quantity) as total_quantity
from pizza_types as pt
left join pizzas as p
   on pt.pizza_type_id=p.pizza_type_id
JOIN order_details AS od
  ON p.pizza_id = od.pizza_id
group by pt.name
order by total_quantity desc
limit 5;
```

name	total_quantity
The Barbecue Chicken Pizza	2169
The Classic Deluxe Pizza	2160
The Hawaiian Pizza	2125
The California Chicken Pizza	2093
The Thai Chicken Pizza	2079

SHODWE
Pizza Resto

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.



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

hour	order_count
11	160
12	305
13	294
14	242
15	190

hour	order_count
17	305
18	301
19	250
20	223
21	152

JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.


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

	category	distribution_of_pizzas
▶	Chicken	6
	Classic	8
	Supreme	2

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

```sql
SELECT
    pt.name, SUM(od.quantity * p.price) AS Revenue
FROM
    pizza_types AS pt
        JOIN
    pizzas AS p ON pt.pizza_type_id = p.pizza_type_id
        JOIN
    order_details AS od ON p.pizza_id = od.pizza_id
GROUP BY pt.name
LIMIT 3;
```

	name	Revenue
▶	The Hawaiian Pizza	28320.5
	The Classic Deluxe Pizza	33600
	The Thai Chicken Pizza	38051.25

SHODWE
Pizza Resto

Home About Contact

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```sql
SELECT
    o.date,
    ROUND(SUM(p.price * od.quantity), 2) AS daily_revenue,
    ROUND(SUM(SUM(p.price * od.quantity)) OVER (ORDER BY o.date), 2) AS cumulative_revenue
FROM orders AS o
JOIN order_details AS od
    ON o.order_id = od.order_id
JOIN pizzas AS p
    ON od.pizza_id = p.pizza_id
GROUP BY o.date
ORDER BY o.date;
```

date	daily_revenue	cumulative_revenue
2015-01-01	1150.25	1150.25
2015-01-02	1117.25	2267.5
2015-01-03	1216.5	3484
2015-01-04	777.75	4261.75
2015-01-05	917.25	5179

Home About Contact

THANK YOU