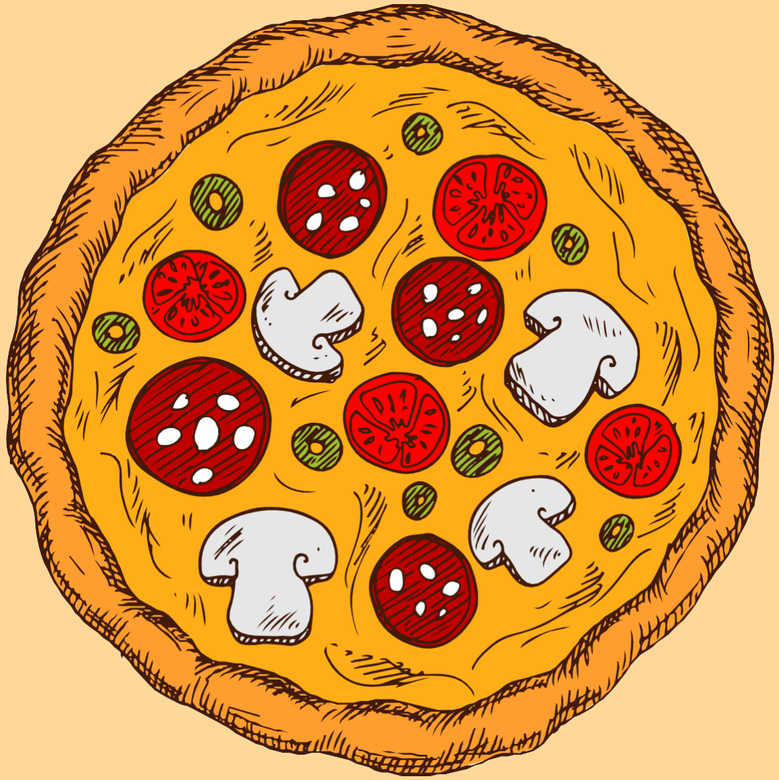
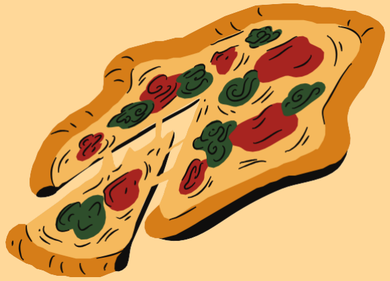


USING mySQL

PIZZA SALES

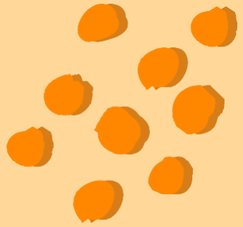
@DEEPAK PARASHAR



Hello!

"I,DEEPAK PARASHAR, leveraged a real-world dataset from a pizza store to meticulously design data models, execute advanced SQL queries, and master data manipulation techniques, enabling insightful analysis and informed decision-making within the realm of database management."

About Project



**"Pizza Sales"* is a MySQL-based project where advanced SQL querying techniques, including joins and subqueries, are employed to analyze sales data. This project aims to uncover valuable insights into customer preferences, optimize operational efficiencies, and enhance decision-making processes for a fictional pizza chain.

QUESTION

```
-- Retrieve the total number of orders placed.
```

QUERY

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

RESULT

Result Grid

	order_placed
▶	21350

QUESTION

```
-- Calculate the total revenue generated from pizza sales.
```

QUERY

```
select round(sum( order_details.quantity  * pizzas.price ),2) as total_revenue
from pizzas
join order_details
on pizzas.pizza_id =  order_details.pizza_id;
```

RESULT

Result Grid

	total_revenue
▶	817860.05

QUESTION

```
-- Identify the highest-priced pizza.
```

QUERY

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

RESULT

Result Grid | Filter Rows:

	name	price
▶	The Greek Pizza	35.95

QUESTION

```
-- Identify the most common pizza size ordered.
```

QUERY

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

RESULT

Result Grid | Filter

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

QUESTION

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

QUERY

```
SELECT pizza_types.name, SUM(order_details.quantity) AS total_quantity
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.name
ORDER BY total_quantity desc
LIMIT 5;
```

RESULT

Result Grid | Filter Rows:

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

QUESTION

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

QUERY

```sql
SELECT pizza_types.category, SUM(order_details.quantity) AS total_quantity
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
ORDER BY total_quantity desc;
```

RESULT

Result Grid | Filter Rows:

category	total_quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

QUESTION

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

QUERY

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

RESULT

Result Grid

hour	order_count
9	1
10	8
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

QUESTION

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

QUERY

```
select pizza_types.name,
sum(order_details.quantity * pizzas.price ) as revenue
FROM pizza_types
JOIN pizzas
ON pizzas.pizza_type_id = pizza_types.pizza_type_id
JOIN  order_details
ON order_details.pizza_id = pizzas.pizza_id
group by pizza_types.name order by revenue desc
limit 3;
```

RESULT

Result Grid | Filter Rows:

name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

QUESTION

```
-- Analyze the cumulative revenue generated over time.
```

QUERY

```
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 ;
```

Result Grid | Filter Rows:

order_date	cum_revenue
2015-01-01	2713.8500000000004
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	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.300000000003
2015-01-14	32358.700000000004
2015-01-15	34343.50000000001
2015-01-16	36937.65000000001

RESULT

TOTAL RESULTS OF CUMULATIVE REVENUE IS CANT”T BE TAKEN IN THIS SCREENSHOT …. SO THIS IS LIMITED RESULTS

QUESTION

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

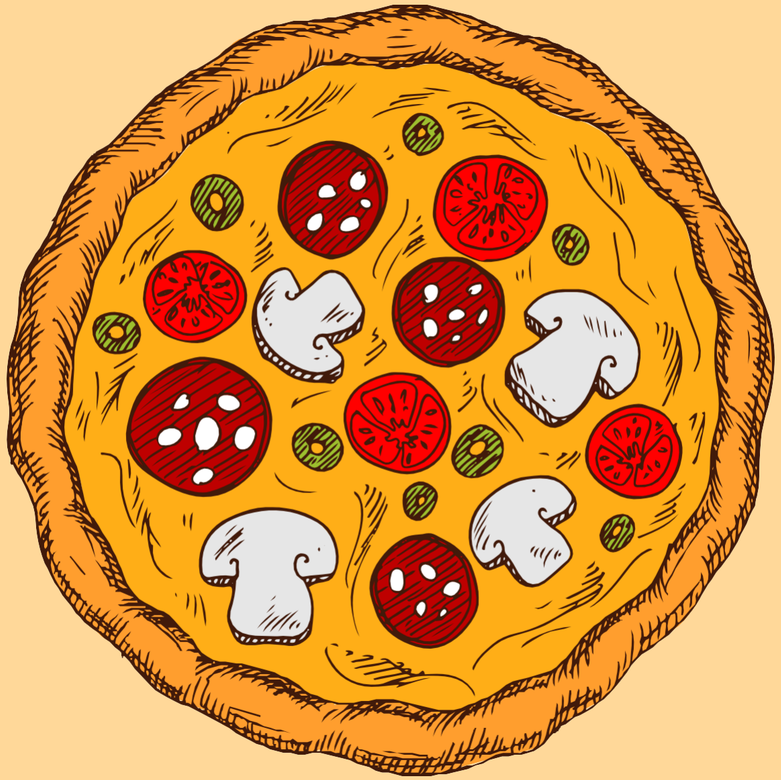
QUERY

```
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;
```

RESULT

Result Grid | Filter Rows: | Expo

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



Pizza Sales Project

THANK YOU

@DEEPAK PARASHAR