

"Data Doughnut: Unraveling Pizza Sales Trends with SQL Magic!"

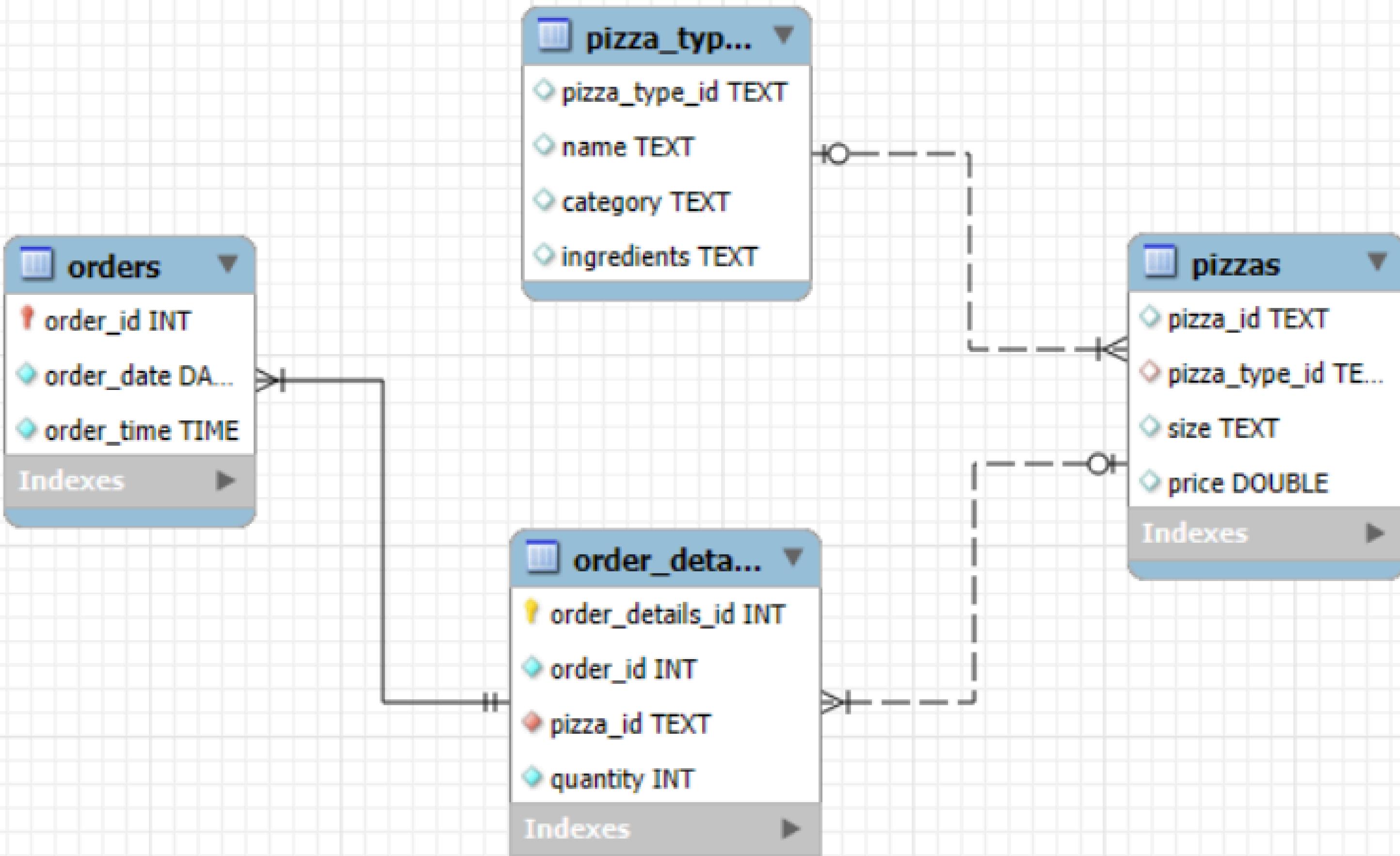
BY YASH GUPTA



Project Title: Pizza Sales Analysis Using SQL

EXPLORE OUR COMPREHENSIVE ANALYSIS OF PIZZA SALES, WHERE SQL QUERIES REVEAL DEEP INSIGHTS INTO CUSTOMER PREFERENCES, POPULAR TOPPINGS, AND PEAK ORDERING TIMES. THROUGH METICULOUS DATA ANALYSIS, ENHANCE OPERATIONAL EFFICIENCY, OPTIMIZE INVENTORY MANAGEMENT, AND BOOST CUSTOMER SATISFACTION. JOIN US IN LEVERAGING DATA-DRIVEN STRATEGIES TO ACHIEVE SUCCESS IN THE PIZZA INDUSTRY.

DATA MODEL



RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```
# Retrieve the total number of orders placed
```

```
SELECT  
    COUNT(order_id) AS Total_orders_placed  
FROM  
    orders;
```

Result Grid		Filter
Total_orders_placed		
21350		

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

SELECT

```
ROUND(SUM(quantity * price), 2) AS Total_Revenue
```

FROM

```
order_details
```

JOIN

```
pizzas ON pizzas.pizza_id = order_details.pizza_id;
```



Result Grid	
	Total_Revenue
▶	817860.05

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
SELECT  
    size, COUNT(order_details_id) as order_count  
FROM  
    pizzas  
    JOIN  
        order_details ON pizzas.pizza_id = order_details.pizza_id  
GROUP BY size  
ORDER BY order_count DESC  
LIMIT 1;
```



Result Grid |  Filter 

size	order_count
L	18526



IDENTIFY THE HIGHEST PRICED PIZZA

```
SELECT  
    name, price  
FROM  
    pizza_types  
        JOIN  
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
ORDER BY price DESC  
LIMIT 1;
```

Result Grid | Filter Row

	name	price
▶	The Greek Pizza	35.95



LIST THE TOP 5 MOST ORDERED PIZZA ALONG WITH THEIR QUANTITIES

SELECT

```
name, SUM(quantity) AS Total_quantity_per_pizza
```

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 name

ORDER BY Total_quantity_per_pizza DESC

LIMIT 5;



	name	Total_quantity_per_pizza
▶	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 CATEGORY WISE DISTRIBUTION OF PIZZAS.

SELECT

```
category,  
COUNT(name) AS 'Total no of pizzas under each category'
```

FROM

```
pizza_types
```

```
GROUP BY category;
```



Result Grid | Filter Rows:

	category	Total no of pizzas under each category
▶	Chicken	6
▶	Classic	8
▶	Supreme	9
▶	Veggie	9

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

```
SELECT
    category,
    SUM(quantity) AS 'total quantity of each pizza category'
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 category
ORDER BY 'total quantity of each pizza category' DESC;
```

	category	total quantity of each pizza category
▶	Classic	14888
	Veggie	11649
	Supreme	11987
	Chicken	11050

DETERMINE THE DISTRIBUTION OF ORDERS BY THE HOURS OF THE DAY

SELECT

```
HOUR(order_time) AS hours, COUNT(order_id) AS order_count
```

FROM

```
orders
```

GROUP BY hours

ORDER BY hours;

hours	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

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

SELECT

```
ROUND(AVG(Sum_Pizzas), 0) AS 'average pizzas ordered per day'
```

FROM

```
(SELECT
```

```
order_date, SUM(quantity) AS Sum_Pizzas
```

FROM

```
orders
```

```
JOIN order_details ON orders.order_id = order_details.order_id
```

```
GROUP BY order_date) AS order_quantity;
```

	Result Grid	Filter Rows:
	average pizzas ordered per day	
▶	138	

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

```
SELECT  
    name, ROUND(SUM(quantity * price), 2) AS Total_Revenue  
FROM  
    pizza_types  
        JOIN  
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
        JOIN  
    order_details ON pizzas.pizza_id = order_details.pizza_id  
GROUP BY name  
ORDER BY Total_Revenue DESC  
LIMIT 3;
```

Result Grid | Filter Rows:

	name	Total_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

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

	category	Revenue(%)
▶	Classic	26.91%
	Veggie	23.68%
	Supreme	25.46%
	Chicken	23.96%

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

```
select category ,name , Total_Revenue from
(SELECT
    category,name , Total_Revenue ,
    rank() over(partition by category order by Total_Revenue desc) as rn
  from
  (SELECT
      category,name , ROUND(SUM(quantity * price), 2) AS Total_Revenue
  FROM
      pizza_types
      JOIN
      pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
      JOIN
      order_details ON pizzas.pizza_id = order_details.pizza_id
  GROUP BY category , name
) as Sales) as Sales2
where rn <=3 ;
```

	category	name	Total_Revenue
▶	Chicken	The Thai Chicken Pizza	43434.25
	Chicken	The Barbecue Chicken Pizza	42768
	Chicken	The California Chicken Pizza	41409.5
	Classic	The Classic Deluxe Pizza	38180.5
	Classic	The Hawaiian Pizza	32273.25
	Classic	The Pepperoni Pizza	30161.75
	Supreme	The Spicy Italian Pizza	34831.25
	Supreme	The Italian Supreme Pizza	33476.75
	Supreme	The Sicilian Pizza	30940.5
	Veggie	The Four Cheese Pizza	32265.7
	Veggie	The Mexicana Pizza	26780.75
	Veggie	The Five Cheese Pizza	26066.5

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME

```
select order_date ,  
Round(revenue,2) as REVENUE,  
Round(sum(revenue) over( order by order_date ) ,2 ) as cum_revenue  
from  
(select order_date , sum(quantity*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 order_date) as Sales;
```

	order_date	REVENUE	cum_revenue
▶	2015-01-01	2713.85	2713.85
	2015-01-02	2731.9	5445.75
	2015-01-03	2662.4	8108.15
	2015-01-04	1755.45	9863.6
	2015-01-05	2065.95	11929.55
	2015-01-06	2428.95	14358.5
	2015-01-07	2202.2	16560.7
	2015-01-08	2838.35	19399.05
	2015-01-09	2127.35	21526.4
	2015-01-10	2463.95	23990.35
	2015-01-11	1872.3	25862.65
	2015-01-12	1919.05	27781.7
	2015-01-13	2049.6	29831.3
	2015-01-14	2527.4	32358.7
	2015-01-15	1984.8	34343.5
	2015-01-16	2594.15	36937.65
	2015-01-17	2064.1	39001.75
	2015-01-18	1976.85	40978.6
	2015-01-19	2387.15	43365.75
	2015-01-20	2397.9	45763.65
	2015-01-21	2040.55	47804.2
	2015-01-22	2496.7	50300.9
	2015-01-23	2423.7	52724.6

As we conclude our pizza sales project, SQL queries have proven indispensable in uncovering customer preferences. Armed with these insights, we're poised to enhance operations, boost profits, and ensure our pizza aficionados remain thoroughly satisfied. Thank you for joining us on this flavorful journey!

THANK YOU