

SQL PROJECT ON PIZZA SALES

BY VINOD BHAND





HELLO EVERYONE !

I AM VINOD BHAND

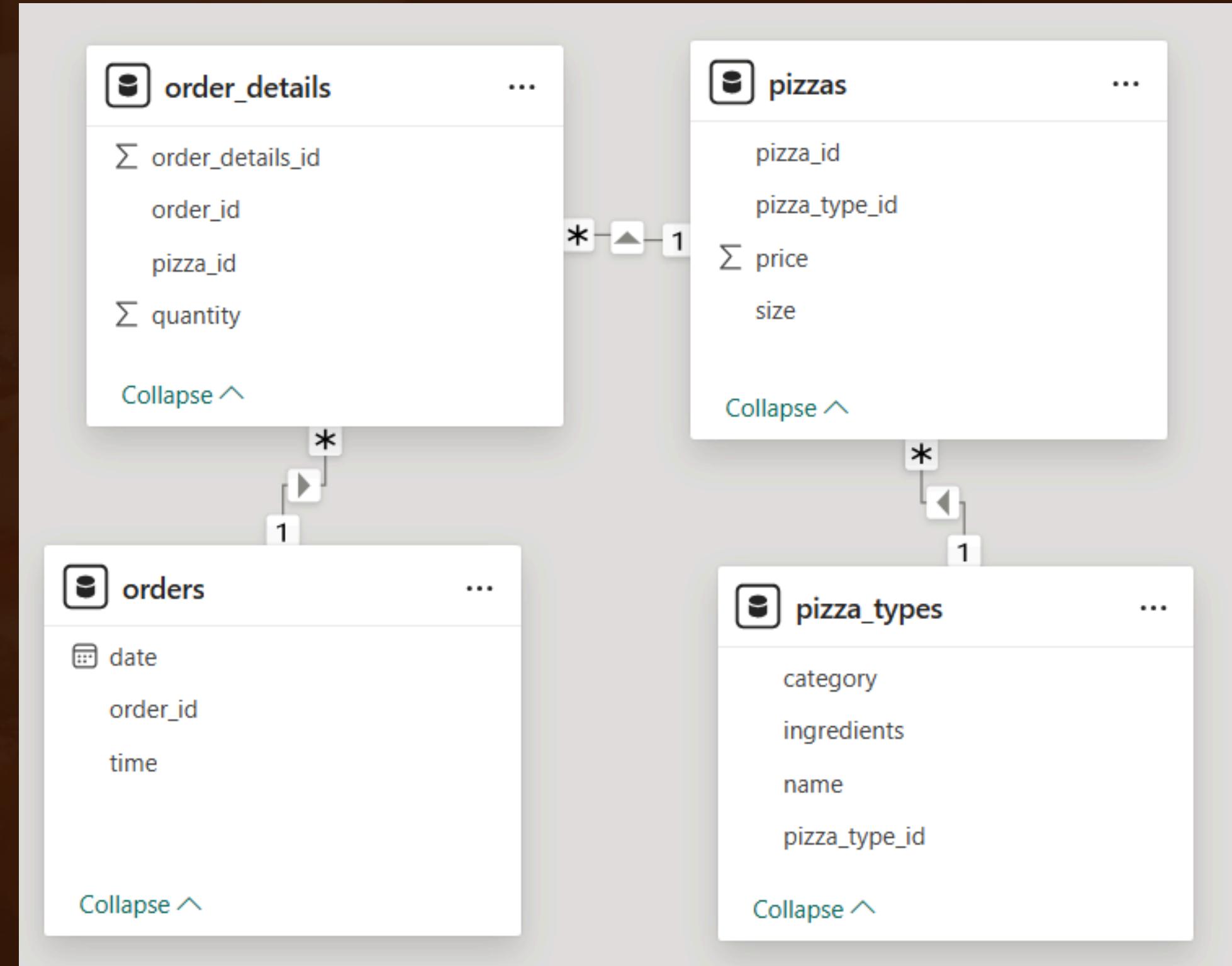
An aspiring Data Analyst. In this project, I utilized my SQL skills to write queries that analyzed pizza sales data, providing valuable business insights





SQL PROJECT

DATABASE SCHEMA



VINOD BHAND

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.



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

RESULT

	total_order
▶	21350

QUERY

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.



SELECT

```
ROUND(SUM(order_details.quantity * pizzas.price),  
2) AS Total_Sales
```

FROM

```
order_details
```

JOIN

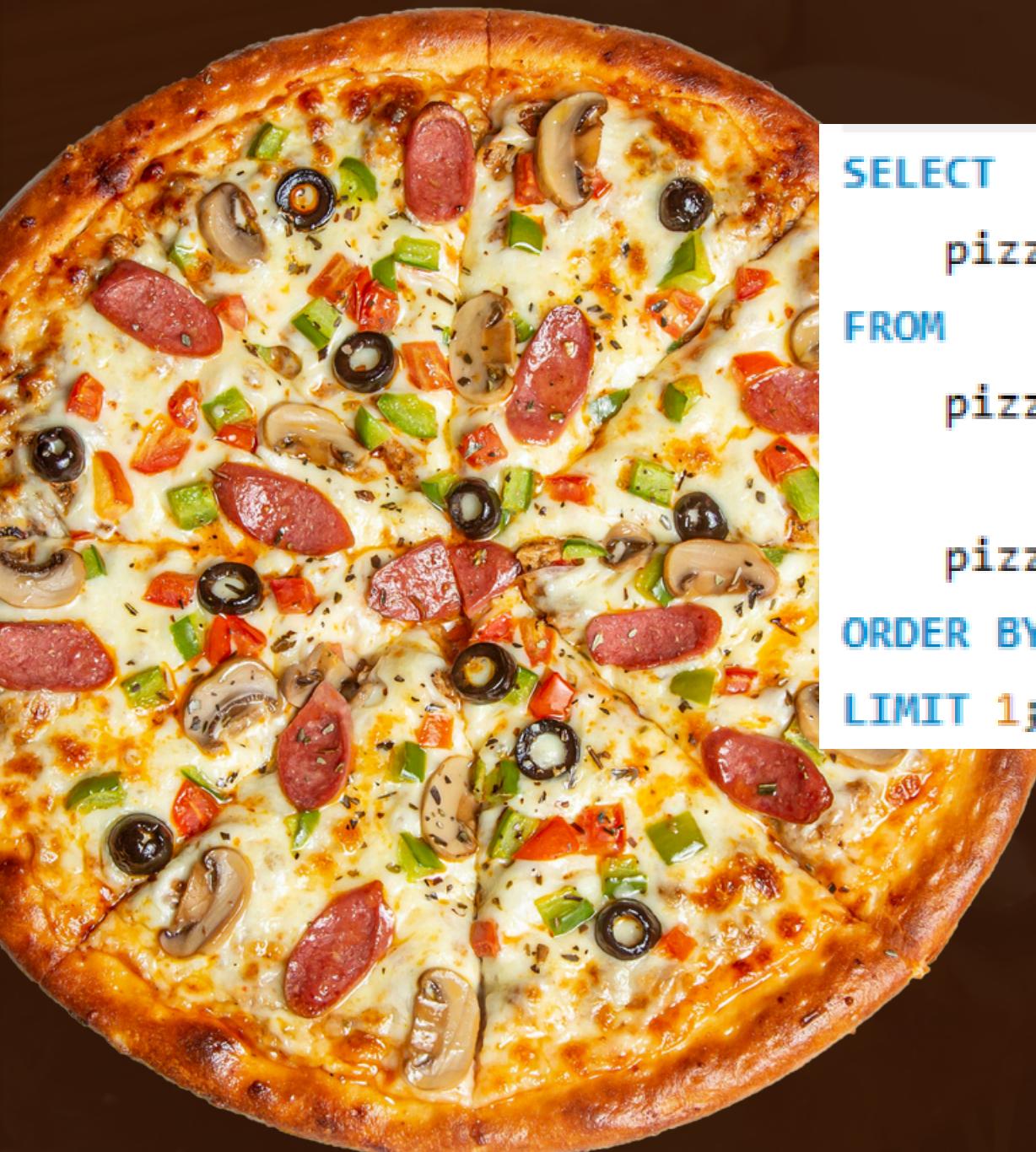
```
pizzas ON pizzas.pizza_id = order_details.pizza_id
```

QUERY

RESULT

	Total_Sales
▶	817860.05

IDENTIFY THE HIGHEST-PRICED PIZZA.



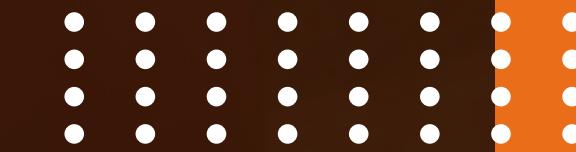
```
SELECT
    pizza_types.name, pizzas.price
FROM
    pizzas
        JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
ORDER BY price DESC
LIMIT 1;
```

QUERY

RESULT

	name	price
▶	The Greek Pizza	35.95

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.



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;
```

QUERY

RESULT

	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.



SELECT

```
    pizza_types.name, SUM(order_details.quantity) AS 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 quantity
LIMIT 5;
```

QUERY

RESULT

	name	quantity
▶	The Brie Carre Pizza	490
▶	The Mediterranean Pizza	934
▶	The Calabrese Pizza	937
▶	The Spinach Supreme Pizza	950
▶	The Soppressata Pizza	961

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



SELECT

```
pizza_types.category,  
SUM(order_details.quantity) AS 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 quantity;
```

RESULT

QUERY

	category	quantity
▶	Chicken	11050
	Veggie	11649
	Supreme	11987
	Classic	14888

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.



SELECT

```
HOUR(order_time) AS hours, COUNT(order_id) AS orders  
FROM  
orders  
GROUP BY HOUR(order_time);
```

QUERY

RESULT

	hours	orders
▶	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

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



SELECT

```
category, COUNT(pizza_types.name) AS count  
FROM  
pizza_types  
GROUP BY category  
ORDER BY category DESC;
```

QUERY

RESULT

	category	count
▶	Veggie	9
	Supreme	9
	Classic	8
	Chicken	6

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



SELECT

```
ROUND(AVG(quantity), 0) AS quantity  
FROM  
(SELECT  
    orders.order_date, SUM(order_details.quantity) AS quantity  
FROM  
    orders  
JOIN order_details ON orders.order_id = order_details.order_id  
GROUP BY orders.order_date) AS order_quantity;
```

QUERY

RESULT

	quantity
▶	138

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



SELECT

```
    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.name  
ORDER BY revenue DESC  
LIMIT 3;
```

RESULT

QUERY

	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.



```
SELECT
    pizza_types.category,
    ROUND((SUM(order_details.quantity * pizzas.price) / (SELECT
        ROUND(SUM(order_details.quantity * pizzas.price),
        2) AS Total_Sales
    )
    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 order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY revenue DESC;
```

RESULT

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

QUERY

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.



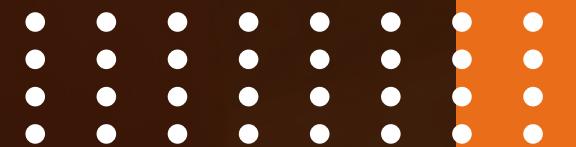
```
SELECT order_date, revenue,sum( revenue) OVER(ORDER BY order_date) AS cumulative_revenue
FROM
(SELECT orders.order_date, SUM(order_details.quantity * pizzas.price) AS revenue
FROM order_details
JOIN
pizzas ON order_details.pizza_id = pizzas.pizza_id
JOIN
orders ON orders.order_id = order_details.order_id
GROUP BY orders.order_date) AS sales ;
```

QUERY

RESULT

	order_date	revenue	cumulative_revenue
▶	2015-01-01 00:00:00	2713.8500000000004	2713.8500000000004
	2015-01-02 00:00:00	2731.899999999996	5445.75
	2015-01-03 00:00:00	2662.399999999996	8108.15
	2015-01-04 00:00:00	1755.450000000003	9863.6
	2015-01-05 00:00:00	2065.95	11929.55
	2015-01-06 00:00:00	2428.95	14358.5
	2015-01-07 00:00:00	2202.200000000003	16560.7
	2015-01-08 00:00:00	2838.349999999995	19399.05
	2015-01-09 00:00:00	2127.350000000004	21526.4
	2015-01-10 00:00:00	2463.95	23990.350000000002
	2015-01-11 00:00:00	1872.300000000002	25862.65
	2015-01-12 00:00:00	1919.050000000002	27781.7
	2015-01-13 00:00:00	2049.600000000004	29831.300000000003
	2015-01-14 00:00:00	2527.399999999996	32358.700000000004
	2015-01-15 00:00:00	1984.800000000002	34343.500000000001
	2015-01-16 00:00:00	2594.15	36937.650000000001
	2015-01-17 00:00:00	2064.100000000004	39001.750000000001
	2015-01-18 00:00:00	1976.850000000001	40978.600000000006
	2015-01-19 00:00:00	2387.149999999996	43365.750000000001
	2015-01-20 00:00:00	2397.900000000005	45763.650000000001

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



```
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;
```

QUERY

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 FOR ATTENTION



[WWW.LINKEDIN.COM/IN/VINOD-BHAND2074](https://www.linkedin.com/in/vinod-bhand2074)



[GITHUB.COM/VINOD2074](https://github.com/vinod2074)