



PIZZA HUTT





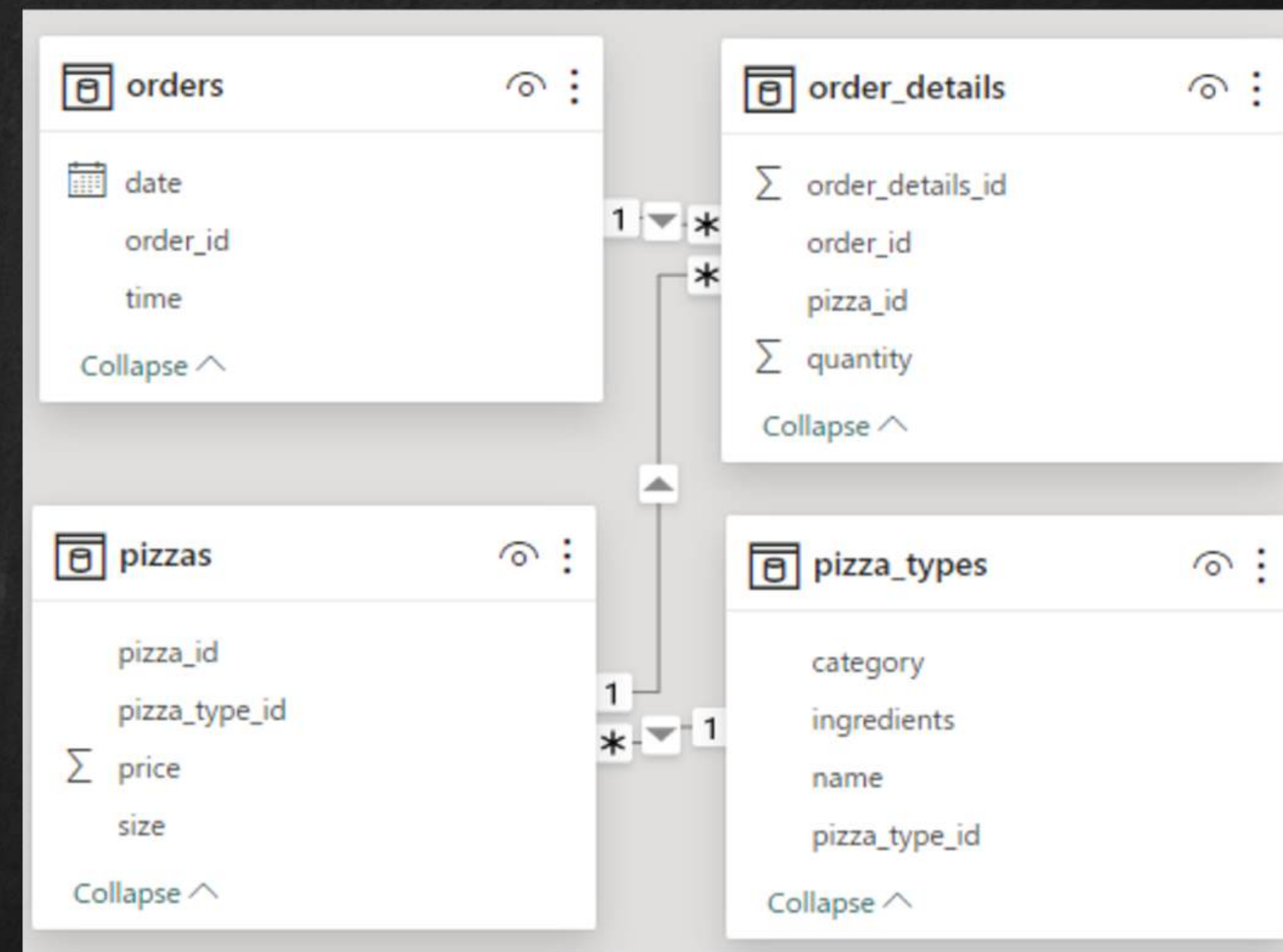
WELCOME TO PIZZA HUTT

Hi, I'm Deepika! In this project, I've used SQL queries to analyze pizza sales data. Let's uncover some tasty insights together!





ORDER DETAILS



RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

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

Output:

	total_orders
▶	21350

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
          2) AS total_pizza_sales
FROM
    order_details
    JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id;
```

Output:

	total_pizza_sales
▶	817860.05

IDENTIFY THE HIGHEST-PRICED PIZZA.

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

Output:

	name	price
▶	The Greek Pizza	35.95

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
SELECT
    pizzas.size, COUNT(order_details.order_details_id)
FROM
    pizzas
    JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size order by pizzas.size limit 1;
```

Output:

	size	COUNT(order_details.order_details_id)
▶	L	18526

LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THE 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 DESC
LIMIT 5;
```

Output:

	name	quantity
▶	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 TOTAL QUANTITY EACH PIZZA CATEGORY ORDERED.

```
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS quantity
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.category
ORDER BY quantity DESC;
```

Output:

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```
select hour(order_time) as hour, count(order_id) as order_count from orders  
group by hour;
```

Output:

	hour	order_count
▶	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(name) from pizza_types  
group by category;
```

Output:

	category	count(name)
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

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

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

Output:

	avg_pizzas_ordered_per_day
▶	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;
```

Output:

	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,
    CONCAT(ROUND((SUM(order_details.quantity * pizzas.price) / (SELECT
        ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_pizza_sales
        FROM
            order_details
        JOIN
            pizzas ON order_details.pizza_id = pizzas.pizza_id) * 100),
        2),
        '%') AS revenu
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 revenu DESC;
```

Output:

	name	revenu
▶	The Thai Chicken Pizza	5.31%
	The Barbecue Chicken Pizza	5.23%
	The California Chicken Pizza	5.06%
	The Classic Deluxe Pizza	4.67%
	The Spicy Italian Pizza	4.26%
	The Southwest Chicken Pizza	4.24%
	The Italian Supreme Pizza	4.09%
	The Hawaiian Pizza	3.95%
	The Four Cheese Pizza	3.95%
	The Sicilian Pizza	3.78%
	The Pepperoni Pizza	3.69%
	The Greek Pizza	3.48%
	The Mexicana Pizza	3.27%
	The Five Cheese Pizza	3.19%
	The Pepper Salami Pizza	3.12%
	The Italian Capocollo Pizza	3.07%
	The Vegetables + Vegetable...	2.98%
	The Prosciutto and Arugula ...	2.96%
	The Napolitana Pizza	2.95%
	The Spinach and Feta Pizza	2.85%
	The Big Meat Pizza	2.81%
	The Pepperoni, Mushroom, ...	2.3%
	The Chicken Alfredo Pizza	2.07%
	The Chicken Pesto Pizza	2.04%
	The Soppressata Pizza	2.01%
	The Italian Vegetables Pizza	1.96%
	The Calabrese Pizza	1.95%
	The Spinach Pesto Pizza	1.91%
	The Mediterranean Pizza	1.88%
	The Spinach Supreme Pizza	1.87%
	The Green Garden Pizza	1.71%
	The Brie Carre Pizza	1.42%

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
select order_date, round(sum(revenue) over(order by order_date),2) 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;
```

Output:

	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	21526.4
	2015-01-10	23990.35
	2015-01-11	25862.65
	2015-01-12	27781.7
	2015-01-13	29831.3
	2015-01-14	32358.7
	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
	2015-01-20	45763.65



THANK YOU!

