PIZZA SALES ANALYSIS PROJECT

BUSINESS INSIGHTS FROM REAL SALES DATA USING SQL & POWER BI & CANVA

PREPARED BY: **FENIL PATEL**

PROJECT PURPOSE

THIS PROJECT WAS CREATED TO ANALYZE PIZZA SALES DATA AND ANSWER IMPORTANT BUSINESS QUESTIONS, SUCH AS:

BASIC INSIGHTS:

- TOTAL NUMBER OF ORDERS PLACED
- TOTAL REVENUE GENERATED
- MOST COMMON PIZZA SIZE ORDERED
- TOP 5 MOST ORDERED PIZZA TYPES

INTERMEDIATE INSIGHTS:

- CATEGORY-WISE PIZZA SALES
- ORDERS DISTRIBUTION BY HOUR
- AVERAGE PIZZAS ORDERED PER DAY
- TOP PIZZAS BY REVENUE

ADVANCED INSIGHTS:

- PERCENTAGE REVENUE CONTRIBUTION OF EACH PIZZA
- CUMULATIVE REVENUE OVER TIME
- TOP PIZZAS BY REVENUE WITHIN EACH CATEGORY

THE PURPOSE OF THIS PROJECT IS TO TRANSFORM RAW DATA INTO CLEAR BUSINESS INSIGHTS THAT CAN HELP IN DECISION-MAKING.

BUSINESS GROWTH POWERED BY INSIGHTS

Knowing My Customers Better

- While working on this project, I realized how much data can tell about customer choices. For example, I was able to see which pizzas are ordered the most and even at what time of the day people love to order them. This makes the analysis feel more real and connected to customer behavior.

Helping the Business Take Smarter Steps

- After answering these questions, it became clear how a company can take smarter decisions. If they know which pizzas generate the highest revenue, they can promote those items more. If they understand peak hours, they can plan better staff and inventory. This not only saves cost but also improves overall profit.

Growth with Happy Customers

- For me, the best part of this project is seeing how data can directly improve customer satisfaction. When a business serves customers what they like the most and avoids waste, it grows in the right direction. That's the kind of value I believe data analysis can bring to any company.
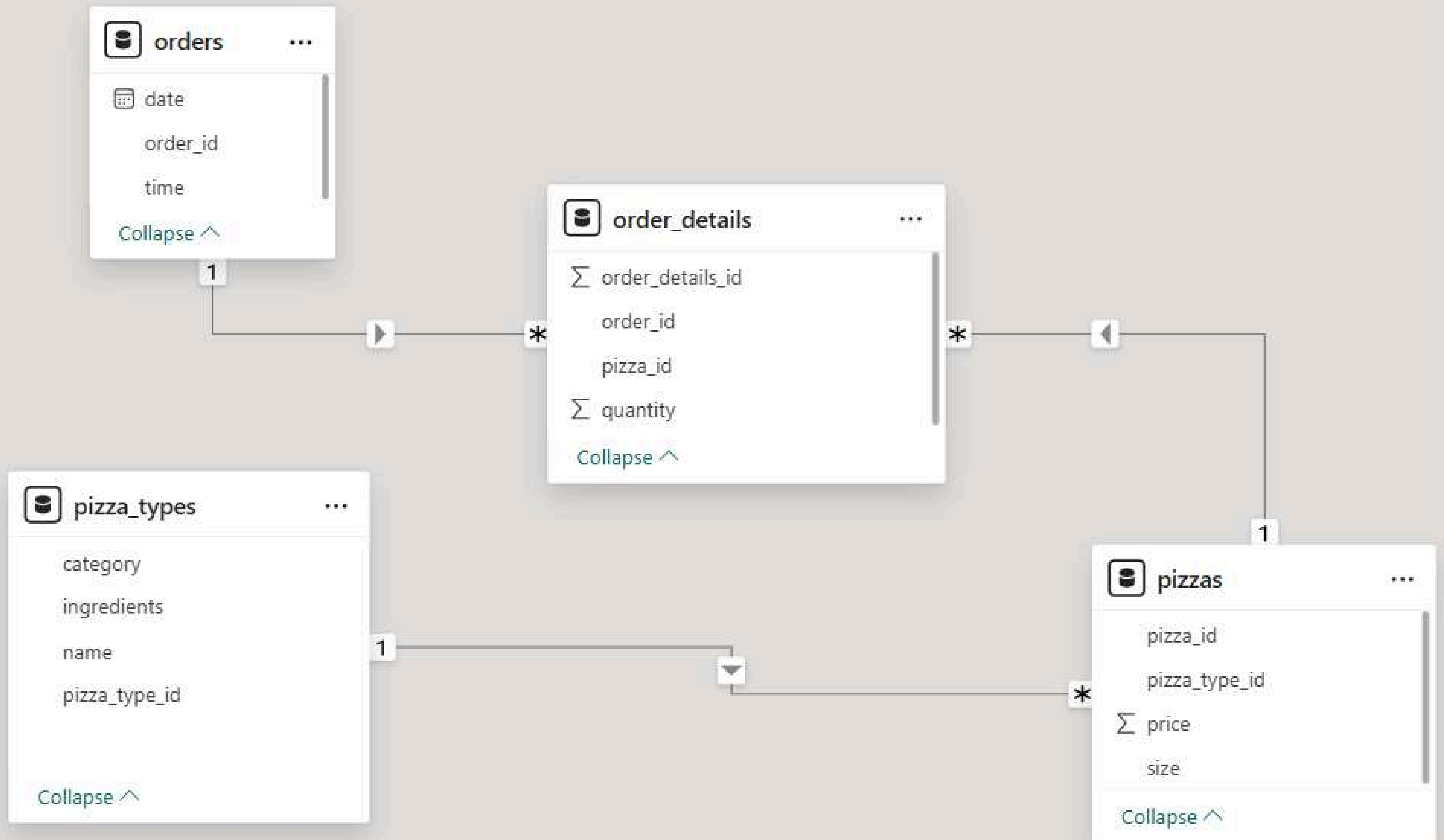
-- RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED

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

total_order
▶ 21350

-- CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES

```sql
SELECT
    ROUND(SUM(orders_details.quantity * pizzas.price),
          2) AS total_sales
FROM
    orders_details
        JOIN
    pizzas ON pizzas.pizza_id = orders_details.pizza_id
```

total_sales
▶ 817860.05

-- IDENTIFY THE HIGHEST-PRICED PIZZA

```sql
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;
```

name	price
▶ The Greek Pizza	35.95

-- IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED

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

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.

```sql
SELECT
    pizza_types.name, SUM(orders_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5;
```

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 OF EACH PIZZA CATEGORY ORDERED.

```sql
SELECT
    pizza_types.category,
    SUM(orders_details.quantity) AS total_quantity
FROM
    pizzas
        JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY total_quantity DESC;
```

category	total_quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

-- DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

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

hours	count_order
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.

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

category	pizza_type_category_wise
Chicken	6
Classic	8
Supreme	9
Veggie	9

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

```sql
SELECT
    ROUND(AVG(quantity), 0) AS avrage_pizza_per_day
FROM
    (SELECT
        orders.order_date, SUM(orders_details.quantity) AS quantity
    FROM
        orders
    JOIN orders_details ON orders.order_id = orders_details.order_id
    GROUP BY orders.order_date) AS orders_quantity;
```

avrage_pizza_per_day
138

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

```sql
SELECT
    pizza_types.name,
    SUM(orders_details.quantity * pizzas.price) AS revenue
FROM
    pizzas
        JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue DESC
LIMIT 3;
```

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.

```sql
SELECT
    pt.category,
    ROUND(SUM(od.quantity * p.price), 2) AS revenue,
    ROUND(100.0 * SUM(od.quantity * p.price)
        / SUM(SUM(od.quantity * p.price)) OVER (),
        2
    ) AS percentage_contribution
FROM orders_details od
JOIN pizzas p
    ON p.pizza_id = od.pizza_id
JOIN pizza_types pt
    ON pt.pizza_type_id = p.pizza_type_id
GROUP BY pt.category
ORDER BY percentage_contribution DESC;
```

category	revenue	percentage_contribution
Classic	220053.1	26.91
Supreme	208197	25.46
Chicken	195919.5	23.96
Veggie	193690.45	23.68

-- ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```sql
select order_date,
round(sum(REVENUE) over (order by order_date),2) as cum_revenue
from
(select orders.order_date,
sum(orders_details.quantity*pizzas.price) AS REVENUE
FROM orders_details JOIN orders
ON orders_details.order_id = orders.order_id
JOIN pizzas
ON orders_details.pizza_id = pizzas.pizza_id
group by orders.order_date) AS SALES ;
```

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

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

```sql
SELECT
    pizza_types.category,
    pizza_types.name AS pizza_name,    -- full name
    SUM(orders_details.quantity * pizzas.price) AS revenue,
    RANK() OVER (
        PARTITION BY pizza_types.category
        ORDER BY SUM(orders_details.quantity * pizzas.price) DESC
    ) AS rnk
FROM pizzas
JOIN pizza_types
    ON pizzas.pizza_type_id = pizza_types.pizza_type_id
JOIN orders_details
    ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category, pizza_types.name
)
SELECT
    pizza_revenue.category,
    pizza_revenue.pizza_name,
    ROUND(pizza_revenue.revenue, 2) AS revenue
FROM pizza_revenue
WHERE pizza_revenue.rnk <= 3
ORDER BY pizza_revenue.category, pizza_revenue.revenue DESC;
```

category	pizza_name	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