Pizza Sales Insights

The tastieyf pizza ever!

Best Pizza in Town !

Markkest
Parm ductien

Project Overview: PizzaHut Sales Data Analysis

Hi, I am Yashaswini ,data enthusiast

 I undertook this project to extract meaningful insights from pizza sales data. I utilized SQL queries to analyze the data, answering critical sales-related questions and uncovering actionable trends to inform business decisions.

Database Schema

PIZZAS
Contains pizza details including IDs, prices, and size information

ORDER_DETAILS
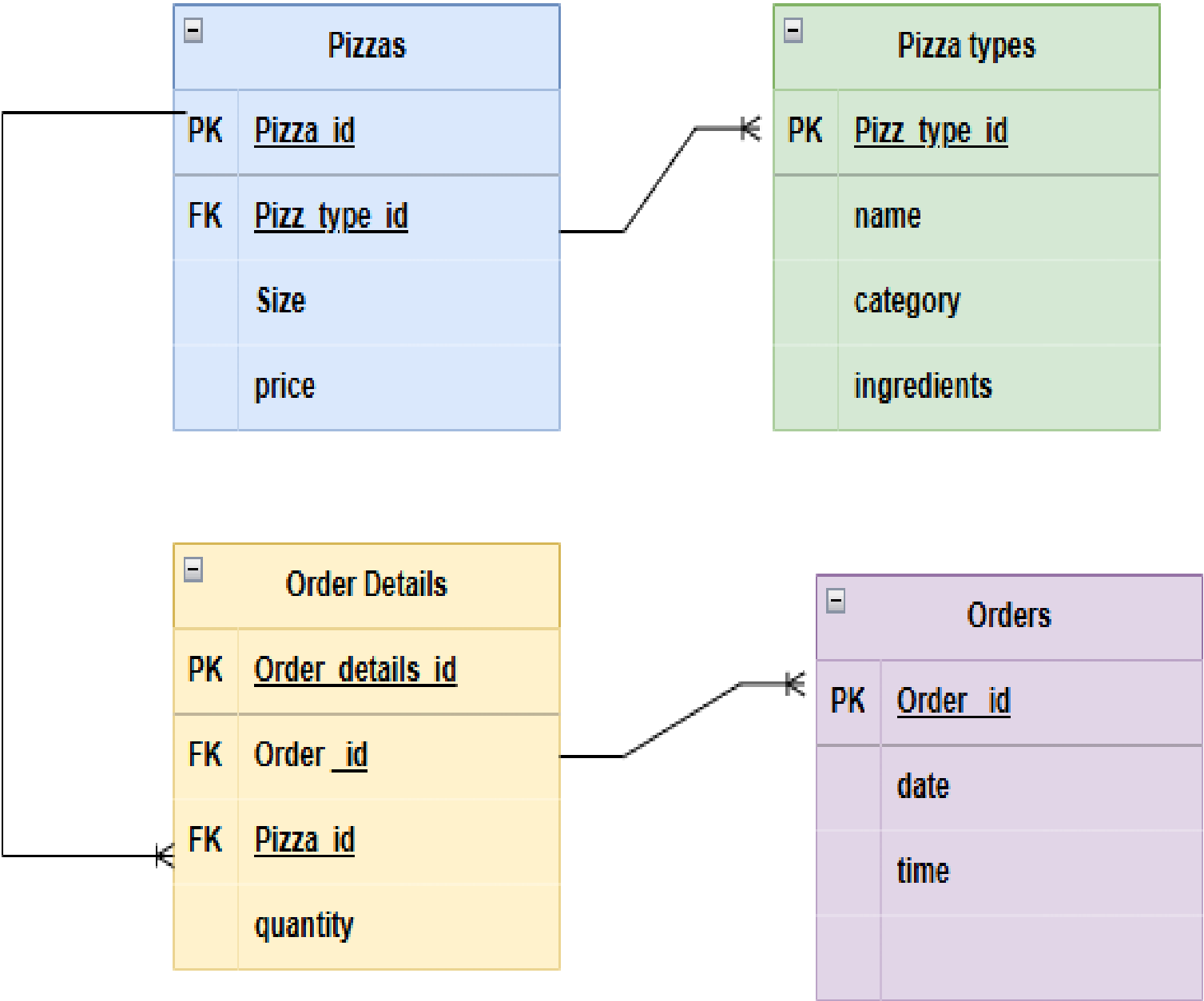Links orders to specific pizzas with quantity information

PIZZA_TYPE
Stores categories and names of different pizza varieties

ORDERS
Contains order dates, times, and unique identifiers

ER DIAGRAM FOR PIZZA HUT

	Pizzas
PK	Pizza_id
FK	Pizz_type_id
	Size
	price

	Pizza types
PK	Pizz_type_id
	name
	category
	ingredients

	Order Details
PK	Order_details_id
FK	Order _id
FK	Pizza_id
	quantity

	Orders
PK	Order _id
	date
	time

Q1. RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

SQL Query used:

```
Select count(order_id) Orders_placed
From Orders;
```

Output:

Orders_placed
21350

Q2. CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

SQL Query used:

```sql
SELECT SUM(o.quantity * p.price) AS TotalRevenue
FROM orders_detail o
JOIN pizzas p ON o.pizza_id = p.pizza_id;
```

Output:

TotalRevenue
817860.0500000101

Q3. IDENTIFY THE HIGHEST-PRICED PIZZA

SQL Query used:

```sql
select p. price, pt.name
from pizzas p
join pizza_type pt on p.pizza_type_id = pt.pizza_type_id
order by price desc
limit 5;
```

Output

price	name
9.75	The Pepperoni Pizza
35.95	The Greek Pizza
25.5	The Greek Pizza
23.65	The Brie Carre Pizza
21	The Italian Vegetables Pizza

Q4. IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

SQL Query used:

```
SELECT p.size,count(od.orders_deatil_id)
FROM orders_detail od
JOIN pizzas p ON od.pizza_id = p.pizza_id
Group by p.size
order by p.size
limit 5;
```

Output

size	count(od.orders_deatil_id)
L	18526
M	15385
S	14137
XL	544
XXL	28

Q5. LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

SQL Query used:

```sql
SELECT pt.name, COUNT(od.orders_id) AS pizza_count
FROM pizzas p
 JOIN pizza_type pt ON p.pizza_type_id = pt.pizza_type_id
 JOIN orders_detail od ON p.pizza_id = od.pizza_id
 GROUP BY pt.name
 ORDER BY pizza_count DESC
 LIMIT 5;
```

Output

name	pizza_count
The Classic Deluxe Pizza	2416
The Barbecue Chicken Pizza	2372
The Hawaiian Pizza	2370
The Pepperoni Pizza	2369
The Thai Chicken Pizza	2315

Q6.Determine the distribution of orders by hour of the day.

SQL Query used:

```sql
SELECT hour(order_time) AS hour_of_day, COUNT(orders.order_id) AS number_of_orders
FROM orders
GROUP BY hour_of_day
ORDER BY number_of_orders DESC
limit 5;
```

Output

hour_of_day	number_of_orders
12	2520
13	2455
18	2399
17	2336
19	2009

Q7. Calculate the average number of pizzas ordered per day.

SQL Query used:

```sql
select date(o.order_date) , avg(od.quantity)
FROM orders o
join orders_detail od on o.order_id=od.orders_id
group by order date
```

Output:

date(o.order_date)	avg(od.quantity)
2015-01-01	1.0062111801242235
2015-01-02	1.03125
2015-01-17	1.0245901639344261
2015-06-17	1.0161290322580645
2015-06-18	1.0175438596491229
2015-06-19	1
2015-06-20	1.032258064516129
2015-06-21	1.0087719298245614
2015-06-22	1.0218978102189782
2015-01-18	1.0252100840336134
2015-06-23	1.008130081300813
2015-06-24	1.0073529411764706
2015-06-25	1.0277777777777777

Q8. Determine the top 3 most ordered pizza types based on revenue.

SQL Query used:

```sql
SELECT SUM(o.quantity * p.price) AS TotalRevenue,pt.name as type
FROM orders_detail o
JOIN pizzas p ON o.pizza_id = p.pizza_id
join pizza_type pt on p.pizza_type_id=pt.pizza_type_id
group by o.quantity,p.price,pt.category,pt.name
order by totalRevenue desc
limit 3;
```

Output

TotalRevenue	type
27410.75	The Thai Chicken Pizza
24235	The Five Cheese Pizza
22161	The Spicy Italian Pizza

Q9. Calculate the percentage contribution of each pizza type to total revenue.

SQL Query used:

```
SELECT pt.category AS pizza_type_name, SUM(od.quantity * p.price) AS total_revenue, (SUM(od.quantity * p.price) * 100.0) /
(SELECT SUM(od.quantity * p.price)
FROM orders_detail od JOIN pizzas p ON od.pizza_id = p.pizza_id) AS percentage
FROM orders_detail od JOIN pizzas p ON od.pizza_id = p.pizza_id
JOIN pizza_type pt ON p.pizza_type_id = pt.pizza_type_id
GROUP BY pt.category ORDER BY percentage  DESC;
```

Output

pizza_type_name	total_revenue	percentage
Classic	220053.10000000033	26.90596025566937
Supreme	208196.99999999988	25.456311260098513
Chicken	195919.5	23.955137556846992
Veggie	193690.44999999777	23.68259092738365

Q10 . Analyze the cumulative revenue generated over time.

SQL Query used:

```sql
select DATE(o.order_date) AS order_date,
SUM(od.quantity * p.price) AS daily_revenue,
SUM(SUM(od.quantity * p.price))
 OVER (ORDER BY DATE(o.order_date)) AS cumulative_revenue
 FROM orders_detail od
 JOIN orders o ON od.orders_id = o.order_id
 JOIN pizzas p ON od.pizza_id = p.pizza_id
 GROUP BY DATE(o.order_date)
 ORDER BY DATE(o.order_date)
 limit 5;
```

Output

order_date	daily_revenue	cumulative_revenue
2015-01-01	2713.8500000000004	2713.8500000000004
2015-01-02	2731.9	5445.75
2015-01-03	2662.3999999999996	8108.15
2015-01-04	1755.4500000000003	9863.6
2015-01-05	2065.95	11929.55

Q 11 Determine the top 3 most ordered pizza types based on revenue for each pizza category.

SQL Query used:

```sql
SELECT SUM(o.quantity * p.price) AS TotalRevenue,pt.category,pt.name
FROM orders_detail o
JOIN pizzas p ON o.pizza_id = p.pizza_id
join pizza_type pt on p.pizza_type_id=pt.pizza_type_id
group by o.quantity,p.price,pt.category,pt.name
order by totalRevenue desc
limit 3;
```

Output

	TotalRevenue	category	name
▶	27410.75	Chicken	The Thai Chicken Pizza
	24235	Veggie	The Five Cheese Pizza
	22161	Supreme	The Spicy Italian Pizza

Conclusion: Using Data for Business Growth

This analysis of pizza sales data shows how SQL queries can turn raw information into useful business insights, highlighting key trends and growth opportunities.

Key Insights

We found top-selling pizzas, popular sizes, major revenue drivers, and busiest order times.

Smart Data Use

The project used the database well to combine and group data, showing how structured data helps deep analysis.

Future Business Plans

These insights will help us make better decisions, like managing inventory and improving marketing.