



SQL PROJECT ON PIZZA SALES

- WHERE EVERY SLICE TELLS A STORY

Home

About

Questions

Contact



ABOUT THIS PROJECT



In this project, I analyzed Pizza Hut's sales data by transferring it from Excel to SQL for structured querying and insights. The dataset includes tables like Order_Details, Pizzas, Pizza_Types, and Orders. I performed SQL queries at different levels:

- Basic Queries: Extracting and filtering data.
- Intermediate Queries: Joining tables and aggregating sales.
- Advanced Queries: Using subqueries and window functions for deeper analysis.

This analysis provided insights into sales trends, customer preferences, and top-selling pizzas



Q1. RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED



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

Result Grid	
	total_orders
▶	21350



Q2. CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.SELECT SUM(PRICE) AS TOTAL_REVENUE FROM PIZZAHUT.PIZZAS

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;
```



Result Grid	
	total_sales
	817860.05



Q3. 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;
```

Result Grid | Filter Rows:

	name	price
▶	The Greek Pizza	35.95



Q4. IDENTIFY THE MOST COMMON 2 PIZZA SIZES 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 limit 2;
```

Result Grid | Filter Rows:

	size	order_count
▶	L	18526
M		15385



Q5. 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 DESC
LIMIT 5;
```

Result Grid		Filter Rows:
	name	quantity
1	The Classic Deluxe Pizza	2453
2	The Barbecue Chicken Pizza	2432
3	The Hawaiian Pizza	2422
4	The Pepperoni Pizza	2418
5	The Thai Chicken Pizza	2371



Q6. 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 DESC;
```

Result Grid | Filter Rows:

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050



Q7. 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(order_time);****Result Grid** | Filter

	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



Q8. JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS



```
SELECT
    pizza_types.category, COUNT(pizza_types.name) AS numbers
FROM
    pizza_types
GROUP BY category;
```

Result Grid | Filter Rows:

	category	numbers
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9



Q9. GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.

SELECT

ROUND(AVG(quantity), 0) AS avg_order_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 order_date) AS order_quantity;

Result Grid

| Filter Rows:

avg_order_quantity

138



Q10. 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 pizzas.pizza_type_id = pizza_types.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 Grid | Filter Rows:

	name	revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5



Home

About

Questions

Contact

CONTACT

LINKEDIN:

<https://www.linkedin.com/in/diya-kukreja-8a662a332>

EMAIL:

diyakukreja33@gmail.com



[Home](#)[About](#)[Questions](#)[Contact](#)

THANK YOU FOR ATTENTION

• 2015 PIZZA SALES PROJECT