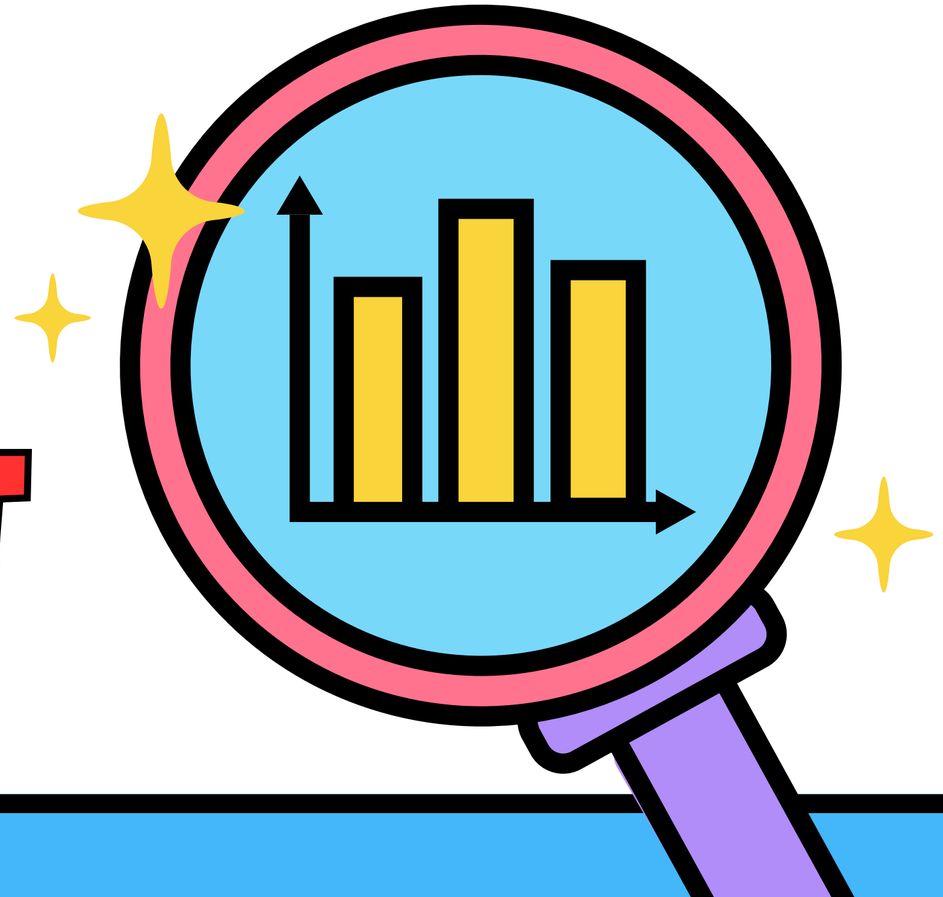
REPRESENTING DATA ANALYSIS OF # PIZZA HUT



DATA

Data is information we collect, like numbers, facts, or details. It helps us learn and make decisions. Data can be shown in tables. Data representation helps people understand the information quickly and easily.

PIZZA SALES DATA

order_details	06-03-2024 16:13	Microsoft Excel Com...	1,278 KB
orders	06-03-2024 16:13	Microsoft Excel Com...	553 KB
pizza_types	06-03-2024 16:13	Microsoft Excel Com...	4 KB
pizzas	06-03-2024 16:13	Microsoft Excel Com...	4 KB

```
SELECT * FROM pizza_hut.orders;
```

order_id	order_date	order_time
1	2015-01-01	11:38:36
2	2015-01-01	11:57:40
3	2015-01-01	12:12:28
4	2015-01-01	12:16:31
5	2015-01-01	12:21:30
6	2015-01-01	12:29:36
7	2015-01-01	12:50:37
8	2015-01-01	12:51:37
9	2015-01-01	12:52:01
10	2015-01-01	13:00:15
11	2015-01-01	13:02:59
12	2015-01-01	13:04:41
13	2015-01-01	13:11:55

```
SELECT * FROM pizza_hut.orders_details;
```

order_details_id	order_id	pizza_id	quantiy
1	1	hawaiian_m	1
2	2	classic_dlx_m	1
3	2	five_cheese_l	1
4	2	ital_supr_l	1
5	2	mexicana_m	1
6	2	thai_ckn_l	1
7	3	ital_supr_m	1
8	3	prsc_argla_l	1
9	4	ital_supr_m	1
10	5	ital_supr_m	1
11	6	bbq_ckn_s	1
12	6	the_greek_s	1
13	7	spinach_supr_s	1
14	8	spinach_supr_s	1
15	9	classic_dlx_s	1
16	9	green_garde...	1
17	9	ital_cpdllo_l	1
18	9	ital_supr_l	1

```
SELECT * FROM pizza_hut.pizza_types;
```

pizza_type_id	name	category	ingredients
bbq_ckn	The Barbecue Chicken Pizza	Chicken	Barbecued Chicken, R
cali_ckn	The California Chicken Pizza	Chicken	Chicken, Artichoke, S
ckn_alfredo	The Chicken Alfredo Pizza	Chicken	Chicken, Red Onions,
ckn_pesto	The Chicken Pesto Pizza	Chicken	Chicken, Tomatoes, R
southw_ckn	The Southwest Chicken Pizza	Chicken	Chicken, Tomatoes, R
thai_ckn	The Thai Chicken Pizza	Chicken	Chicken, Pineapple, T
big_meat	The Big Meat Pizza	Classic	Bacon, Pepperoni, Ita
classic_dlx	The Classic Deluxe Pizza	Classic	Pepperoni, Mushroom
hawaiian	The Hawaiian Pizza	Classic	Sliced Ham, Pineapple
ital_cpdllo	The Italian Capocollo Pizza	Classic	Capocollo, Red Peppe
napolitana	The Napolitana Pizza	Classic	Tomatoes, Anchovies
pep_msh_pep	The Pepperoni, Mushroom, ...	Classic	Pepperoni, Mushroom
pepperoni	The Pepperoni Pizza	Classic	Mozzarella Cheese, P
the_greek	The Greek Pizza	Classic	Kalamata Olives, Feta
brie_carre	The Brie Carre Pizza	Supreme	Brie Carre Cheese, Pr
calabrese	The Calabrese Pizza	Supreme	'Nduja Salami, Pancet
ital_supr	The Italian Supreme Pizza	Supreme	Calabrese Salami, Ca
peppr_salami	The Pepper Salami Pizza	Supreme	Genoa Salami, Capoc
prsc_argla	The Prosciutto and Arugula ...	Supreme	Prosciutto di San Dani
sicilian	The Sicilian Pizza	Supreme	Coarse Sicilian Salami,

```
SELECT * FROM pizza_hut.pizzas;
```

pizza_id	pizza_type_id	size	price
bbq_ckn_s	bbq_ckn	S	12.75
bbq_ckn_m	bbq_ckn	M	16.75
bbq_ckn_l	bbq_ckn	L	20.75
cali_ckn_s	cali_ckn	S	12.75
cali_ckn_m	cali_ckn	M	16.75
cali_ckn_l	cali_ckn	L	20.75
ckn_alfredo_s	ckn_alfredo	S	12.75
ckn_alfredo_m	ckn_alfredo	M	16.75
ckn_alfredo_l	ckn_alfredo	L	20.75
ckn_pesto_s	ckn_pesto	S	12.75
ckn_pesto_m	ckn_pesto	M	16.75
ckn_pesto_l	ckn_pesto	L	20.75
southw_ckn_s	southw_ckn	S	12.75
southw_ckn_m	southw_ckn	M	16.75
southw_ckn_l	southw_ckn	L	20.75
thai_ckn_s	thai_ckn	S	12.75

KPI'S REQUIREMENTS

- Retrieve the total number of orders placed.
- Calculate the total revenue generated from pizza sales.
- Identify the highest-priced pizza.
- Identify the most common pizza size ordered.
- List the top 5 most ordered pizza types along with their quantities.
- Join the necessary tables to find the total quantity of each pizza category ordered.
- Determine the distribution of orders by hour of the day.
- Join relevant tables to find the category-wise distribution of pizzas.
- Group the orders by date and calculate the average number of pizzas ordered per day.
- Determine the top 3 most ordered pizza types based on revenue.
- Calculate the percentage contribution of each pizza type to total revenue.
- Analyze the cumulative revenue generated over time.
- Determine the top 3 most ordered pizza types based on revenue for each pizza category.

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```sql
-- RETRIEVE THE TOTAL NO.OF ORDERS PLACED.

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

	total_orders
▶	21350

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```sql
-- CALCUALTED THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

SELECT
    ROUND(SUM(orders_details.quantiy * 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
-- 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;
```

	name	price
▶	The Greek Pizza	35.95

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```sql
-- IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

SELECT
    pizzas.size,
    COUNT(orders_details.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.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
-- LIST THE TOP 5 MOST ORDERED PIZZA TYPE ALONG WITH THEIR QUANTITIES

SELECT
    pizza_types.name, SUM(orders_details.quantiy)
    as quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.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
-- JOIN THE NECESSARY TABLES TO FIND THE  QUANTITY OF EACH PIZZA CATEGORY ORDERED.

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

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```sql
-- DETERMINE THE DISTRIBUTION OF ORDERS BY HOURS OF THE DAY.

SELECT
    HOUR(order_time) AS hour, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY HOUR(order_time);
```

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.

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

SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category
```

	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.

```sql
--  GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZA ORDERED PER DAY.

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

	avg_pizza_ordered_per_day
▶	138

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

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

select pizza_types.name,
sum(orders_details.quantiy * pizzas.price) as revenue
from pizza_types join pizzas
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
-- CALCULATE THE % CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

select pizza_types.category,
round(sum(orders_details.quantiy * pizzas.price) / (SELECT
    ROUND(SUM(orders_details.quantiy * pizzas.price),
            2) AS total_sales
FROM
    orders_details
        JOIN
    pizzas ON pizzas.pizza_id = orders_details.pizza_id) *
    100,2)
    as revenue
from pizza_types
join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join orders_details
on orders_details.pizza_id = pizzas.pizza_id
group by pizza_types.category
order by revenue desc
```

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```sql
-- ANALYZE THE CUMULATIVE REVENUEE GENERATED OVER TIME.

select order_date,round(
sum(revenue) over(order by order_date),2) as cum_revenue
from
(select orders.order_date,
sum(orders_details.quantiy * pizzas.price) as revenue
from orders_details join pizzas
on orders_details.pizza_id = pizzas.pizza_id
join orders
on orders.order_id = orders_details.order_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
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
2015-01-21	47804.2
2015-01-22	50300.9
2015-01-23	52724.6
2015-01-24	55013.85
2015-01-25	56631.4
2015-01-26	58515.8
2015-01-27	61043.85

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

```sql
-- 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((orders_details.quantiy) * pizzas.price) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join orders_details
on orders_details.pizza_id = pizzas.pizza_id
group by pizza_types.category,pizza_types.name) as a) as b
where rn <= 3;
```

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

CONCLUSION

"OUR PIZZA SALES PROJECT HAS BEEN A VALUABLE LEARNING EXPERIENCE, DEMONSTRATING THE POWER OF SQL IN ANALYZING LARGE DATASETS AND EXTRACTING INSIGHTS. WE HAVE GAINED A DEEPER UNDERSTANDING OF THE PIZZA INDUSTRY AND HAVE IDENTIFIED OPPORTUNITIES FOR GROWTH AND IMPROVEMENT. THIS PROJECT HAS ALSO HIGHLIGHTED THE IMPORTANCE OF DATA VISUALIZATION AND COMMUNICATION IN PRESENTING COMPLEX DATA INSIGHTS TO STAKEHOLDERS. WE HOPE THAT THIS PROJECT WILL SERVE AS A MODEL FOR FUTURE DATA ANALYSIS PROJECTS AND WILL INSPIRE OTHERS TO EXPLORE THE POSSIBILITIES OF DATA-DRIVEN DECISION MAKING."

THANK YOU