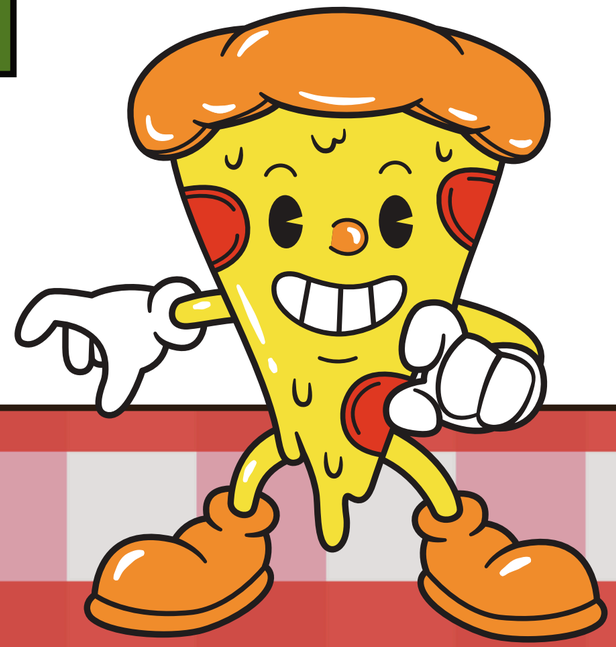
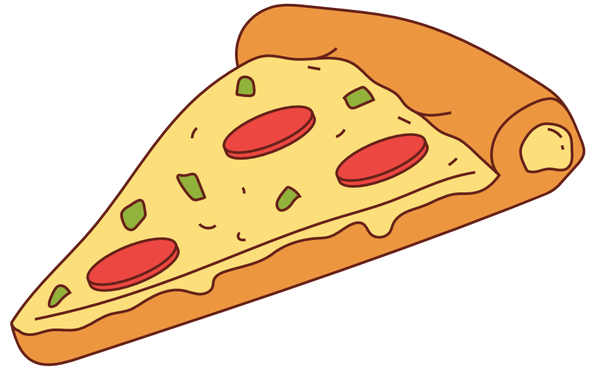
PIZZA HUT DATA ANALYSIS PROJECT THROUGH SQL



QUESTIONS



Basic:

- 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.

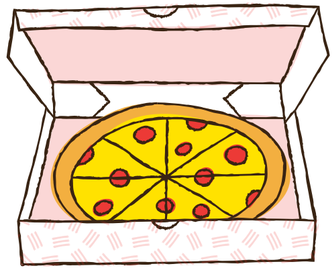
Intermediate:

- 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.

Advanced:

- 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.

Q.Retrieve the total number of orders placed.



```sql
SELECT
    COUNT(order_id) AS total_orders
FROM
    pizzahut.orders;
```

Result Grid

total_orders
21350

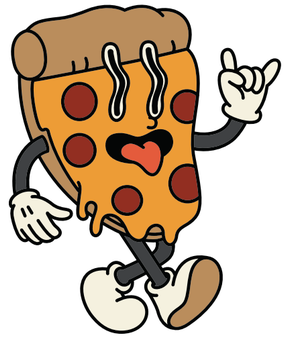
Q.Calculate the total revenue generated from pizza sales.

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

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

total_revenue
817860.05

Q.Identify the highest-priced pizza.



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

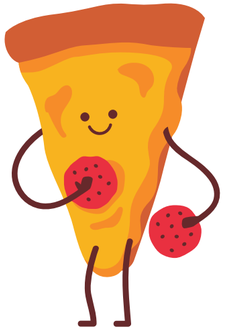
	price	name
▶	35.95	The Greek Pizza

Q. Identify the most common pizza size ordered.

```
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    order_details ON
    order_details.pizza_id = pizzas.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC
LIMIT 1;
```

	size	order_count
▶	L	18526

Q. 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 pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_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

Q.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 pizzas join order_details
on order_details.pizza_id=pizzas.pizza_id
join pizza_types
on pizza_types.pizza_type_id=pizzas.pizza_type_id
group by pizza_types.category order by quantity desc;
```

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

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

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

hour(order_time)	count(order_id)
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

Q. 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

Q.Group the orders by date and calculate the average number of pizzas ordered per day.

```sql
SELECT
    ROUND(AVG(quantity), 0) as avg_pizza_orders_perDay
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;
```

avg_pizza_orders_perDay
138

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

```sql
select pizza_types.name,
sum(pizzas.price*order_details.quantity) as revenue
from pizza_types join pizzas on
pizza_types.pizza_type_id=pizzas.pizza_type_id
join order_details
on pizzas.pizza_id=order_details.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

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

```
select pizza_types.category, round(sum(pizzas.price*order_details.quantity)/
( select sum(pizzas.price*order_details.quantity) from pizzas
join order_details
on pizzas.pizza_id=order_details.pizza_id) *100,2)
 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.category order by revenue desc;
```

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Q.Analyze the cumulative revenue generated over time.

```sql
select order_date,
sum(revenue) over(order by order_date) as cum_revenue
from
(select orders.order_date, sum(order_details.quantity*pizzas.price) as revenue
from orders join order_details
on orders.order_id=order_details.order_id
join pizzas
on pizzas.pizza_id=order_details.pizza_id
group by orders.order_date ) as sales;
```

order_date	cum_revenue
2015-01-01	2713.8500000000004
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	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.300000000003
2015-01-14	32358.700000000004

and more...

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

```
select name, revenue from
(select name, category, revenue,
 rank()over(partition by category order by revenue desc)as rn
 from
(select pizza_types.name, pizza_types.category,
SUM(pizzas.price*(order_details.quantity)) 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, pizza_types.category) as a) as b
where rn<=3;
```

name	revenue
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5
The Classic Deluxe Pizza	38180.5
The Five Cheese Pizza	26066.5
The Four Cheese Pizza	32265.70000000065
The Hawaiian Pizza	32273.25
The Italian Supreme Pizza	33476.75
The Mexicana Pizza	26780.75
The Pepperoni Pizza	30161.75
The Sicilian Pizza	30940.5
The Spicy Italian Pizza	34831.25
The Thai Chicken Pizza	43434.25

THANK YOU !