PIZZA HUT

PROJECT

Using SQL

PIZZA HUT
PROJECT

ABOUT DATA

This project is created by Pinki Nigam just to demonstrate the skills of SQL. Data set used in this project is a public data set.

PIZZA HUT
PROJECT

WE ARE GOING TO SOLVE 11 QUESTION IN THIS PROJECT

Que 1 Calculate the total revenue generated from pizza sales.

```sql
-- Calculate the total revenue generated from pizza sales.

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

Result Grid | Filte

total_sales
817860.05

PIZZA HUT
PROJECT

NEXT QUESTION

Que 2 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;
```

Result Grid | Filter Rows

name	price
The Greek Pizza	35.95

PIZZA HUT
PROJECT

QUESTION 3

Identify the most common pizza size ordered.

```
-- Identify the most common pizza size ordered.
SELECT
    pizzas.size,
    SUM(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
```

Result Grid | Filter R

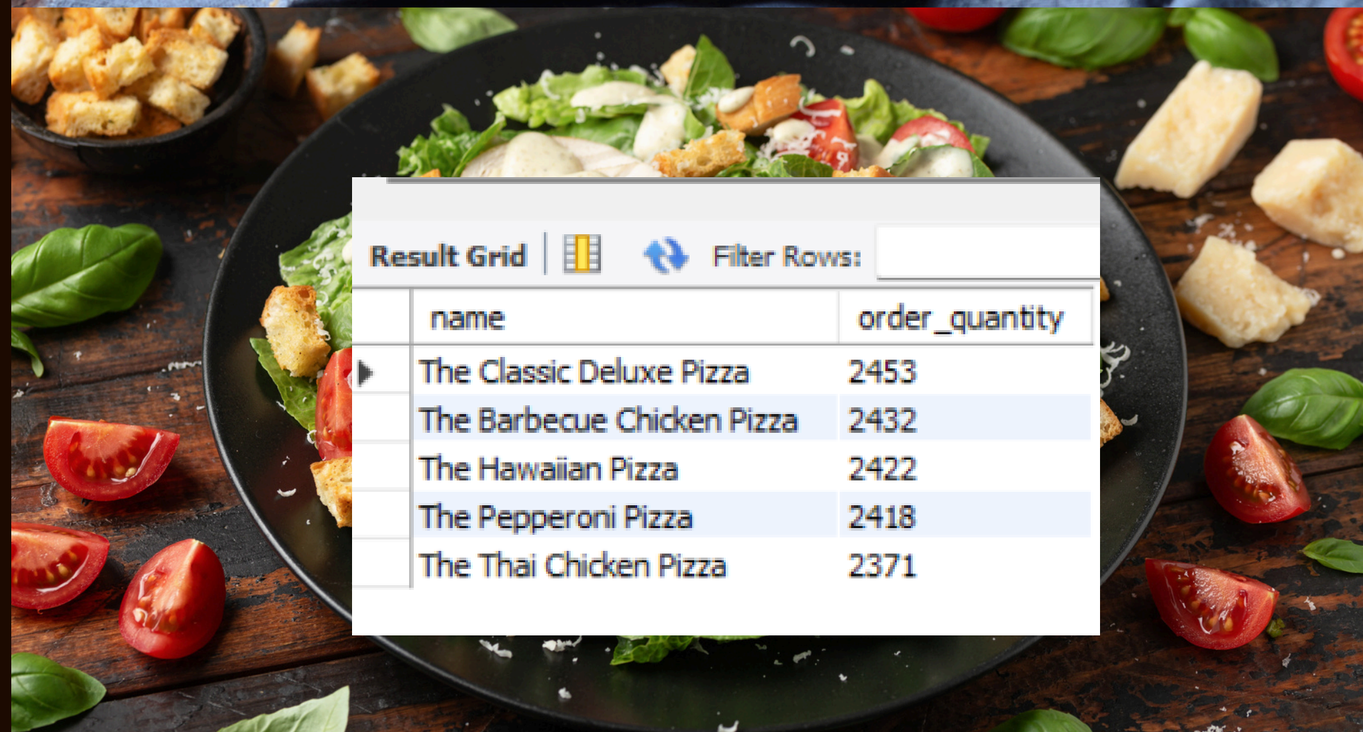
size	order_count
L	449383379
M	374739224
S	344360774
XL	12852430
XXL	640703

PIZZA HUT
PROJECT

Que-4

List the top 5 most ordered pizza types along with their quantities.

```
select pizza_types.name, sum(order_details.quantity) as order_quantity
from pizzas join pizza_types
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 order_quantity desc limit 5;
```



Result Grid | Filter Rows:

name	order_quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

```sql
-- Join the necessary tables to find the total quantity
-- of each pizza category ordered.

SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS total_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 total_quantity DESC
```

Result Grid | Filter Rov

category	total_quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

QUESTION NO.5

Join the necessary tables to find the total quantity of each pizza category ordered.

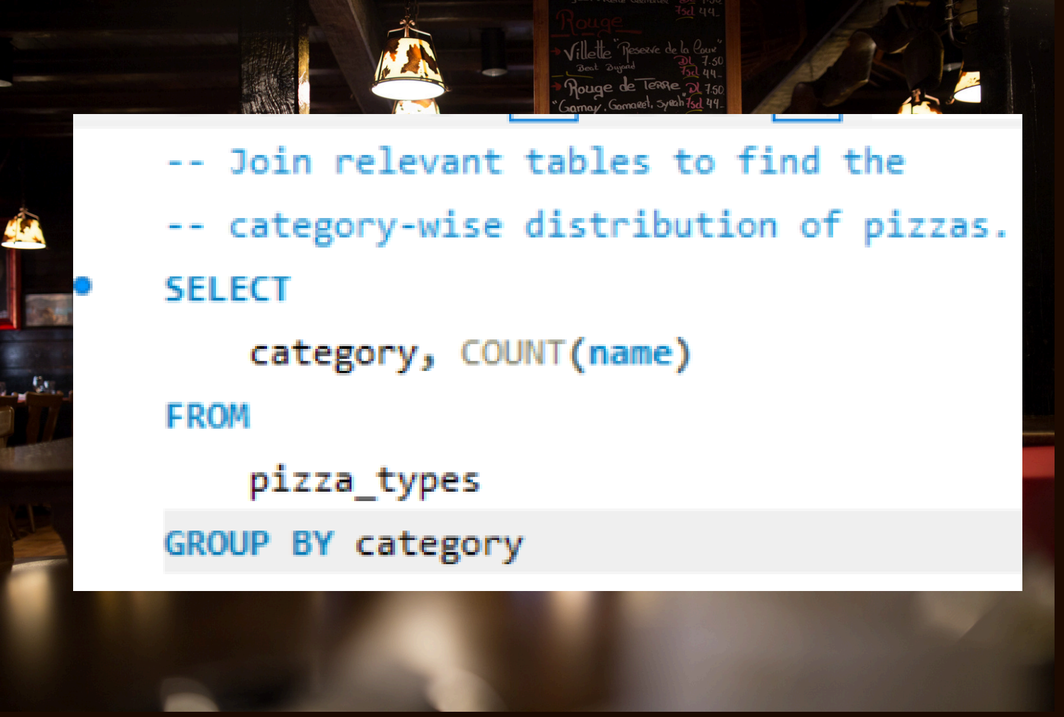
QUESTION NO. 6

Determine the distribution of orders by hour of the day.

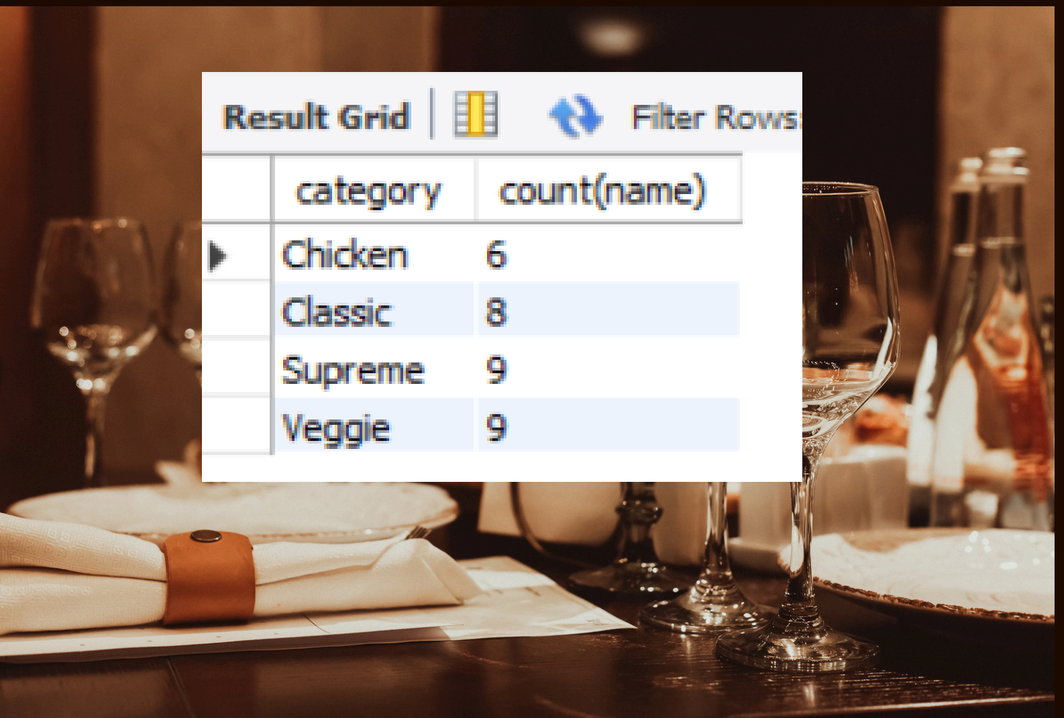
```sql
-- Determine the distribution of orders by hour of the day.

SELECT
    HOUR(time), COUNT(order_id) AS total_orders
FROM
    orders
GROUP BY HOUR(time);
```

Result Grid | Filter Rows:

hour(time)	total_orders
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920



```sql
-- Join relevant tables to find the
-- category-wise distribution of pizzas.
SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category
```



Result Grid | Filter Rows:

category	count(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

QUESTION NO. 7

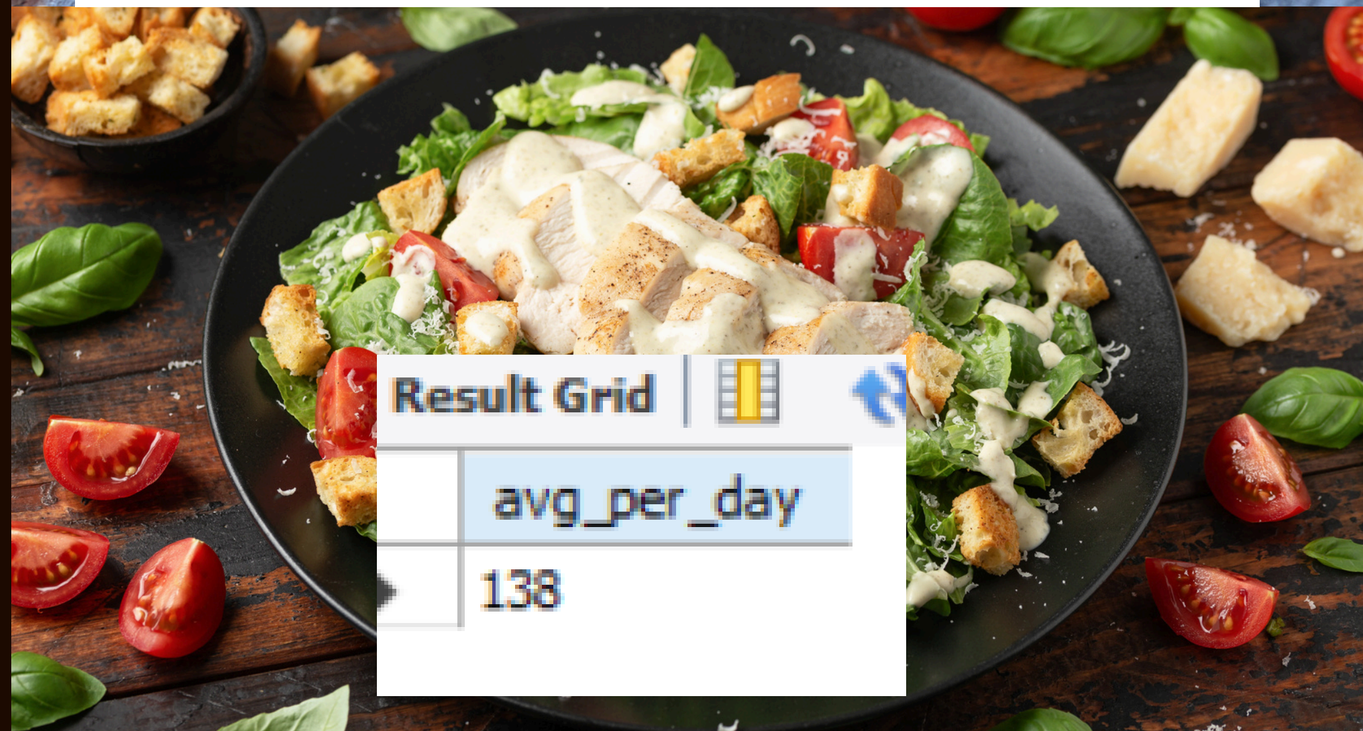
Join relevant tables to find the category-wise distribution of pizzas.

PIZZA HUT
PROJECT

Que-8

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 pizzas ordered per day
SELECT
    ROUND(AVG(order_quantity), 0) AS avg_per_day
FROM
    (SELECT
        orders.date, SUM(order_details.quantity) AS order_quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.date) AS total_orders
```



Result Grid

avg_per_day
138

PIZZA HUT
PROJECT

Que-9

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(pizzas.price * order_details.quantity) AS revenue
FROM
    pizzas
        JOIN
    pizza_types 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

PIZZA HUT
PROJECT

Que-10

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

```sql
-- 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) AS total_revenue
                FROM
                    pizzas
                        JOIN
                    order_details ON pizzas.pizza_id = order_details.pizza_id) * 100,
            2) AS revenue_contribution
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.category
ORDER BY revenue_contribution DESC
```

Result Grid | Filter Rows:

category	revenue_contribution
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

PIZZA HUT
PROJECT

Que-11

Analyze the cumulative revenue generated over time.

```sql
-- Analyze the cumulative revenue generated over time.
select date,sum(revenue) over (order by date) as cum_revenue
from
(select orders.date, round(sum(order_details.quantity * pizzas.price),2) as revenue
from order_details join orders
on order_details.order_id = orders.order_id
join pizzas
on pizzas.pizza_id = order_details.pizza_id
group by orders.date) as sales
```

Result Grid | Filter Rows:

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

PIZZAHUT
PROJECT

THANK YOU