

SPECIAL PIZZA

Life is short, eat the pizza!

ORDER NOW



Hello, my self Ritik Raushan, i utilize the concept of
MySQL query to develop this project by solving some
questions on pizza sales.

```
create database pizzahut;
```

```
use pizzahut;
```

```
create table orders (
    order_id INT NOT NULL,
    order_date date NOT NULL,
    order_time time not null,
    primary key(order_id)
);
```

```
create table order_details (
    order_details_id INT NOT NULL,
    order_id int not null,
    pizza_id text NOT NULL,
    quantity int not null,
    primary key(order_details_id)
);
```

- RETRIEVE THE TOTAL NUMBER OF ORDER PLACED

- `select count(order_id) from orders;`

-- CALCULATE REVENUE GENERATED FROM PIZZA SALES

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

-- IDENTIFY THE HIGHEST PRICE PIZZA

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

-- IDENTIFY THE MOST COMMON SIZE PIZZA ORDERED

```
select * from pizzahut.order_details;  
SELECT  
    pizzas.size,  
    COUNT(order_deatils.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;
```

-- LIST THE MOST ORDER TYPE PIZZA ALONG THEIR QUANTITY

```
SELECT  
    pizza_types.name, SUM(order_details.quantity) AS quantity  
FROM  
    pizza_types  
        JOIN  
    pizzas ON pizza_types.pizza_types_id = pizzas.pizza_type_id  
        JOIN  
    order_details ON order_details.pizza_id = pizzas.pizza_id  
GROUP BY pizza_types.name  
ORDER BY quantity;
```

-- 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;
```

-- DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY

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

-- JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF
PIZZA

```
select category, count(name) from pizza_types  
group by category;
```

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

```
SELECT
    AVG(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 orders.order_date) AS order_quantity;
```

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

```
SELECT  
    pizza_types.names,  
    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.names order by revenue desc limit 3;
```

-- CALCULATE THE PERCENTAGE DISTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE

```
select pizza_types.category,  
    round (sum(order_details.quantity * pizzas.price) / (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) *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;
```

- ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME

```
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 order_details join pizzas
on order_details.pizza_id = pizzas.pizza_id
join orders
on orders.order_id = order_details.order_id
group by orders.order_date) as sales ;
```

-- DETERMINE THE TOP 3 MOST ORDERED PIZZA BASE 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(order_details.quantity * pizzas.price) 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,pizza_types.name) as a) as b
where rn <= 3;
```

<THANK YOU>