

## Queries For Analysis

-- Retrieve the total number of orders placed.

```
select count(order_id) as total_orders from orders;
```

-- 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 pizzas.pizza_id = order_details.pizza_id
```

-- 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;
```

-- 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 pizzas.pizza_id = order_details.pizza_id GROUP BY pizzas.size  
ORDER BY order_count DESC;
```

-- 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;
```

-- 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
order_details JOIN pizzas ON order_details.pizza_id = pizzas.pizza_id JOIN pizza_types ON
pizzas.pizza_type_id = pizza_types.pizza_type_id GROUP BY pizza_types.category;
```

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

```
SELECT HOUR(order_time) AS order_hour, COUNT(*) AS total_orders FROM orders GROUP
BY HOUR(order_time) ORDER BY order_hour;
```

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

```
SELECT pizza_types.category, COUNT(pizzas.pizza_id) AS total_pizzas FROM pizzas JOIN
pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id GROUP BY
pizza_types.category ORDER BY total_pizzas DESC;
```

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

```
SELECT pizza_types.category, COUNT(pizzas.pizza_id) AS total_pizzas FROM pizzas JOIN
pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id GROUP BY
pizza_types.category ORDER BY total_pizzas DESC;
```

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

```
SELECT DATE(o.order_date) AS order_day, SUM(od.quantity) AS total_pizzas FROM orders
o JOIN order_details od ON o.order_id = od.order_id GROUP BY DATE(o.order_date);
```

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

```
SELECT pt.name AS pizza_type, SUM(od.quantity * p.price) AS total_revenue FROM
order_details od JOIN pizzas p ON od.pizza_id = p.pizza_id JOIN pizza_types pt ON
p.pizza_type_id = pt.pizza_type_id GROUP BY pt.name ORDER BY total_revenue DESC
LIMIT 3;
```

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

```
SELECT pt.name AS pizza_type, SUM(od.quantity * p.price) AS pizza_revenue,  
(SUM(od.quantity * p.price) / (SELECT SUM(od2.quantity * p2.price) FROM order_details  
od2 JOIN pizzas p2 ON od2.pizza_id = p2.pizza_id)) * 100 AS percentage_contribution  
FROM order_details od JOIN pizzas p ON od.pizza_id = p.pizza_id JOIN pizza_types pt ON  
p.pizza_type_id = pt.pizza_type_id GROUP BY pt.name ORDER BY pizza_revenue DESC;
```

-- Analyze the cumulative revenue generated over time.

```
SELECT DATE(o.order_date) AS order_day, SUM(od.quantity * p.price) AS daily_revenue,  
SUM(SUM(od.quantity * p.price)) OVER (ORDER BY DATE(o.order_date)) AS  
cumulative_revenue FROM orders o JOIN order_details od ON o.order_id = od.order_id  
JOIN pizzas p ON od.pizza_id = p.pizza_id GROUP BY DATE(o.order_date) ORDER BY  
order_day;
```

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

```
WITH Revenue_Per_Category AS ( SELECT pt.category, pt.name AS pizza_type,  
SUM(od.quantity * p.price) AS pizza_revenue FROM order_details od JOIN pizzas p ON  
od.pizza_id = p.pizza_id JOIN pizza_types pt ON p.pizza_type_id = pt.pizza_type_id GROUP  
BY pt.category, pt.name ), Ranked_Pizzas AS ( SELECT category, pizza_type,  
pizza_revenue, RANK() OVER (PARTITION BY category ORDER BY pizza_revenue DESC) AS
```

```
pizza_rank FROM Revenue_Per_Category ) SELECT category, pizza_type, pizza_revenue  
FROM Ranked_Pizzas WHERE pizza_rank <= 3 ORDER BY category, pizza_rank;
```