Pizza Sales analyzing by MySQL

```
-- Basic:

-- 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

   quantity, COUNT(order_details_id)

FROM
```

```
    order_details
GROUP BY quantity;


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 4;



-- Intermediate:
-- 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) as hour,count(order_id) as order_count from orders
group by hour(order_time);

-- Join relevant tables to find the category-wise distribution of pizzas.
 select category, count(name) from pizza_types
 group by category;

-- Group the orders by date and calculate the average number of pizzas ordered per day.
SELECT
    ROUND(AVG(quantity), 0) as avg_pizza_orderd_per_day
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.name,
    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.name
ORDER BY revenue DESC
LIMIT 3;

-- Advanced:
-- Calculate the percentage contribution 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 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((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;
```