Answers

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
    SUM(o.quantity * p.price) AS total_revenue
FROM
    order_details o
        JOIN
    pizzas p ON o.pizza_id = p.pizza_id;
```

- Identify the highest-priced pizza.

```
SELECT
    MAX(price)
FROM
    pizzas;
```

- Identify the most common pizza size ordered

```
SELECT
    p.size, SUM(o.quantity) AS total_times_ordered
FROM
    order_details o
        JOIN
    pizzas p ON o.pizza_id = p.pizza_id
GROUP BY p.size
ORDER BY total_times_ordered DESC
LIMIT 1
```

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

```
SELECT
    pizza_types.name, SUM(o.quantity) AS total_ordered
FROM
    order_details o
        JOIN
    pizzas p ON o.pizza_id = p.pizza_id
        JOIN
    pizza_types ON p.pizza_type_id = pizza_types.pizza_type_id
GROUP BY pizza_types.name
ORDER BY total_ordered DESC
LIMIT 5;
```

Intermediate

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

```sql
SELECT
    pt.category, SUM(o.quantity) AS total_ordered
FROM
    order_details o
        JOIN
    pizzas p ON o.pizza_id = p.pizza_id
        JOIN
    pizza_types pt ON p.pizza_type_id = pt.pizza_type_id
GROUP BY pt.category
ORDER BY total_ordered DESC
```

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

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

- Join relevant tables to find the average price of the pizza per category.

```sql
SELECT
    pt.category, AVG(p.price) AS avg_price
FROM
    pizzas p
        JOIN
    pizza_types pt ON p.pizza_type_id = pt.pizza_type_id
GROUP BY pt.category
```

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

```sql
SELECT
    date, AVG(total_pizzas) AS avg_pizzas_per_day
FROM
    (SELECT
        date, SUM(quantity) AS total_pizzas
    FROM
        orders o
    JOIN order_details od ON o.order_id = od.order_id
    GROUP BY date) AS daily_orders
GROUP BY date;
```

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

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

Advanced:

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

```
SELECT
    t.name,
    SUM(p.price * o.quantity) AS revenue,
    ROUND(SUM(p.price * o.quantity) / 817860.05 * 100,
            2) AS per
FROM
    order_details AS o
        JOIN
    pizzas AS p ON o.pizza_id = p.pizza_id
        JOIN
    pizza_types AS t ON p.pizza_type_id = t.pizza_type_id
GROUP BY t.name
ORDER BY per DESC;
```

- Analyze the cumulative revenue generated over time.

```
select monthname(o.date) as month,
round(sum(p.price*od.quantity),2) as revenue,
round(sum(sum(p.price*od.quantity)) over (order by min(o.date)),2) 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 month
```

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

```
WITH ranked_pizzas AS (
    SELECT *,
           ROW_NUMBER() OVER (PARTITION BY category ORDER BY total_revenue DESC) AS ranking
    FROM (
        SELECT p.pizza_id, pt.name, pt.category, p.price,
               SUM(od.quantity) AS total_quantity,
               SUM(od.quantity * p.price) AS total_revenue
        FROM pizzas p
        JOIN order_details od ON p.pizza_id = od.pizza_id
        join pizza_types pt on p.pizza_type_id = pt.pizza_type_id
        GROUP BY p.pizza_id, pt.name, pt.category, p.price
    ) AS pizza_revenue
)
SELECT category, name, total_quantity, total_revenue
FROM ranked_pizzas
WHERE ranking <= 3;
```