```
-- ============================================
-- BASIC QUESTIONS
-- ============================================

-- 1. Total number of orders placed
SELECT COUNT(order_id) AS total_orders
FROM orders;

-- 2. Total revenue generated (Rounded to 2 decimal places)
SELECT ROUND(SUM(od.quantity * p.price), 2) AS total_sales
FROM order_details od
JOIN pizzas p ON p.pizza_id = od.pizza_id;

-- 3. The highest-priced pizza
SELECT TOP 1 pt.name, ROUND(p.price, 2) AS highest_price
FROM pizza_types pt
JOIN pizzas p ON pt.pizza_type_id = p.pizza_type_id
ORDER BY p.price DESC;

-- 4. Most common pizza size ordered
SELECT p.size, COUNT(od.order_details_id) AS order_count
FROM pizzas p
JOIN order_details od ON p.pizza_id = od.pizza_id
GROUP BY p.size
ORDER BY order_count DESC;

-- 5. Top 5 most ordered pizza types (by quantity)
SELECT TOP 5 pt.name, SUM(od.quantity) AS total_quantity
FROM pizza_types pt
JOIN pizzas p ON pt.pizza_type_id = p.pizza_type_id
JOIN order_details od ON od.pizza_id = p.pizza_id
GROUP BY pt.name
ORDER BY total_quantity DESC;


-- ============================================
-- INTERMEDIATE QUESTIONS
-- ============================================

-- 1. Total quantity of each pizza category
SELECT pt.category, SUM(od.quantity) AS total_quantity
FROM pizza_types pt
JOIN pizzas p ON pt.pizza_type_id = p.pizza_type_id
JOIN order_details od ON od.pizza_id = p.pizza_id
GROUP BY pt.category
ORDER BY total_quantity DESC;

-- 2. Distribution of orders by hour
SELECT DATEPART(HOUR, order_time) AS order_hour, COUNT(order_id) AS order_count
FROM orders
GROUP BY DATEPART(HOUR, order_time)
ORDER BY order_hour;

-- 3. Category-wise distribution of pizzas (Inventory variety)
SELECT category, COUNT(name) AS variant_count
FROM pizza_types
```

```
GROUP BY category;

-- 4. Average number of pizzas ordered per day
SELECT ROUND(AVG(daily_total), 0) AS avg_pizzas_per_day
FROM (
SELECT o.order_date, SUM(od.quantity) AS daily_total
FROM orders o
JOIN order_details od ON o.order_id = od.order_id
GROUP BY o.order_date
) AS daily_sales;

-- 5. Top 3 pizza types based on revenue
SELECT TOP 3 pt.name, ROUND(SUM(od.quantity * p.price), 2) AS total_revenue
FROM pizza_types pt
JOIN pizzas p ON p.pizza_type_id = pt.pizza_type_id
JOIN order_details od ON od.pizza_id = p.pizza_id
GROUP BY pt.name
ORDER BY total_revenue DESC;


-- ===========================================
-- ADVANCED QUESTIONS
-- ===========================================

-- 1. Percentage contribution of each category to total revenue
SELECT pt.category,
ROUND((SUM(od.quantity * p.price) / (
SELECT SUM(od2.quantity * p2.price)
FROM order_details od2
JOIN pizzas p2 ON p2.pizza_id = od2.pizza_id
)) * 100, 2) AS revenue_percentage
FROM pizza_types pt
JOIN pizzas p ON pt.pizza_type_id = p.pizza_type_id
JOIN order_details od ON od.pizza_id = p.pizza_id
GROUP BY pt.category
ORDER BY revenue_percentage DESC;

-- 2. Cumulative revenue generated over time
SELECT order_date,
ROUND(SUM(daily_revenue) OVER (ORDER BY order_date), 2) AS cumulative_revenue
FROM (
SELECT o.order_date, SUM(od.quantity * p.price) AS daily_revenue
FROM order_details od
JOIN pizzas p ON od.pizza_id = p.pizza_id
JOIN orders o ON o.order_id = od.order_id
GROUP BY o.order_date
) AS sales_summary;

-- 3. Top 3 pizza types by revenue within each category
SELECT category, name, revenue
FROM (
SELECT pt.category, pt.name,
ROUND(SUM(od.quantity * p.price), 2) AS revenue,
RANK() OVER (PARTITION BY pt.category ORDER BY SUM(od.quantity * p.price) DESC) AS
rank_num
FROM pizza_types pt
```

```
JOIN pizzas p ON pt.pizza_type_id = p.pizza_type_id
JOIN order_details od ON od.pizza_id = p.pizza_id
GROUP BY pt.category, pt.name
) AS ranked_sales
WHERE rank_num <= 3;
```