Q1 – What are the Top 3 best-selling pizzas each month by quantity?

```sql
-- What are the Top 3 best-selling pizzas each month by quantity?
with monthly_sales as
    (select
        month(order_date) as month_of_order,
        pizza_name,
        SUM(quantity) as total_quantity,
        rank() over (partition by month(order_date) order by SUM(quantity) desc) as rank_in_month
    from pizza_sales
    group by month_of_order, pizza_name)
select *
from monthly_sales
where rank_in_month <= 3
order by month_of_order, rank_in_month;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

month_of_order	pizza_name	total_quantity	rank_in_month
1	The Pepperoni Pizza	239	1
1	The Barbecue Chicken Pizza	211	2
1	The California Chicken Pizza	202	3
2	The Pepperoni Pizza	205	1
2	The California Chicken Pizza	204	2
2	The Hawaiian Pizza	198	3
3	The Barbecue Chicken Pizza	231	1
3	The Hawaiian Pizza	217	2
3	The Thai Chicken Pizza	213	3
4	The Hawaiian Pizza	219	1
4	The Classic Deluxe Pizza	216	2
4	The Barbecue Chicken Pizza	214	3
5	The Pepperoni Pizza	241	1
5	The Barbecue Chicken Pizza	226	2
5	The Classic Deluxe Pizza	218	3
6	The California Chicken Pizza	222	1
6	The Classic Deluxe Pizza	197	2
6	The Barbecue Chicken Pizza	194	3
7	The Classic Deluxe Pizza	227	1

Result 1

Q2 – What is the monthly revenue growth rate compared to the previous month?

```sql
-- What is the monthly revenue growth rate compared to the previous month?
WITH monthly_revenue as
    (select
        MONTH(order_date) as month_of_order,
        round(sum(total_price),0) as revenue
    from pizza_sales
    GROUP BY MONTH(order_date))
select
    month_of_order,
    revenue,
    lag(revenue) over(order by Month_of_order) as prev_month_revenue,
    round((revenue-(lag(revenue) over(order by Month_of_order)))/(lag(revenue) over(order by Month_of_order))*100,2) as
Monthly_growth
from monthly_revenue
order by month_of_order;
```

month_of_order	revenue	prev_month_revenue	Monthly_growth
1	69793	NULL	NULL
2	65160	69793	-6.64
3	70397	65160	8.04
4	68737	70397	-2.36
5	71403	68737	3.88
6	68230	71403	-4.44
7	72558	68230	6.34
8	68278	72558	-5.9
9	64180	68278	-6
10	64028	64180	-0.24
11	70395	64028	9.94
12	64701	70395	-8.09

Q3 – What is the cumulative revenue trend throughout the year?

```sql
-- What is the cumulative revenue trend throughout the year?
select
    distinct month(order_date) as Month_of_Order,
    round(sum(total_price) over(partition by month(order_date)),0) as Total_Monthly_Revenue,
    round(sum(total_price)over(order by month(order_date)),0) as Running_Total
from pizza_sales
```

Month_of_Order	Total_Monthly_Revenue	Running_Total
1	69793	69793
2	65160	134953
3	70397	205350
4	68737	274087
5	71403	345490
6	68230	413720
7	72558	486278
8	68278	554556
9	64180	618736
10	64028	682764
11	70395	753159
12	64701	817860

Q4 – Is there a difference in revenue on weekends vs weekdays?

```sql
-- Is there a difference in revenue on weekends vs weekdays?
select
    case
        when dayofweek(order_date) IN (1, 7) then 'Weekend' -- Sunday=1, Saturday=7
        else 'Weekday'
    end AS day_type,
    count(distinct order_id) AS total_orders,
    round(sum(total_price),0) AS total_revenue,
    round(avg(total_price), 2) AS avg_order_value
from pizza_sales
group by day_type
order by field(day_type, 'Weekday', 'Weekend');
```

day_type	total_orders	total_revenue	avg_order_value
Weekday	15568	595474	16.85
Weekend	5782	222386	16.76

Q5 – Which pizzas generate the most revenue per unit sold?

```sql
-- Which pizza name generates the most revenue per unit sold?
SELECT
    pizza_name,
    round(SUM(total_price),0) AS total_revenue,
    SUM(quantity) AS total_units_sold,
    ROUND(SUM(total_price) / SUM(quantity), 2) AS revenue_per_unit
FROM pizza_sales
GROUP BY pizza_name
ORDER BY revenue_per_unit DESC
LIMIT 1;
```

pizza_name	total_revenue	total_units_sold	revenue_per_unit
The Brie Carre Pizza	11588	490	23.65

Q6 – What are the most frequently ordered pizza sizes and types in each month?

```sql
-- What are the most frequently ordered pizza sizes and types in each month?
with monthly_sales as
    (select
        date_format(order_date, '%Y-%m') as month_of_order,
        pizza_size,
        pizza_name,
        sum(quantity) as total_quantity
    from pizza_sales
    group by date_format(order_date, '%Y-%m'), pizza_size, pizza_name),
    ranked as
    (select
        month_of_order,
        pizza_size,
        pizza_name,
        total_quantity,
        rank() over(partition by month_of_order order by total_quantity desc) as size_type_rank
    from monthly_sales)
select *
from ranked
where size_type_rank <=5
order by month_of_order, size_type_rank;
```

month_of_order	pizza_size	pizza_name	total_quantity	size_type_rank
2015-01	S	The Big Meat Pizza	150	1
2015-01	L	The Five Cheese Pizza	138	2
2015-01	L	The Thai Chicken Pizza	119	3
2015-01	L	The Four Cheese Pizza	102	4
2015-01	L	The Barbecue Chicken Pizza	96	5
2015-02	S	The Big Meat Pizza	151	1
2015-02	L	The Four Cheese Pizza	117	2
2015-02	L	The Five Cheese Pizza	113	3
2015-02	L	The Thai Chicken Pizza	95	4
2015-02	L	The Spicy Italian Pizza	89	5
2015-03	S	The Big Meat Pizza	176	1
2015-03	L	The Five Cheese Pizza	125	2
2015-03	L	The Four Cheese Pizza	118	3
2015-03	L	The Thai Chicken Pizza	118	3
2015-03	L	The Spicy Italian Pizza	107	5
2015-04	S	The Big Meat Pizza	139	1
2015-04	L	The Thai Chicken Pizza	116	2
2015-04	L	The Four Cheese Pizza	111	3
2015-04	L	The Five Cheese Pizza	107	4
2015-04	S	The Hawaiian Pizza	101	5
2015-05	S	The Big Meat Pizza	190	1
2015-05	L	The Five Cheese Pizza	124	2
2015-05	L	The Thai Chicken Pizza	117	3

Q7 – What is the average order value segmented by time of the day?

```sql
-- What is the average order value segmented by time of day?
WITH order_totals AS
    (SELECT
            order_id,
            SUM(total_price) AS order_value,
            MIN(order_time) AS order_time -- earliest time for the order
        FROM pizza_sales
        GROUP BY order_id),
    time_segments AS
    (SELECT
            order_id,
            order_value,
            CASE
                WHEN HOUR(order_time) BETWEEN 5 AND 11 THEN 'Morning'
                WHEN HOUR(order_time) BETWEEN 12 AND 16 THEN 'Afternoon'
                WHEN HOUR(order_time) BETWEEN 17 AND 21 THEN 'Evening'
                ELSE 'Night'
            END AS time_of_day
        FROM order_totals)
SELECT
    time_of_day,
    ROUND(AVG(order_value), 2) AS avg_order_value,
    COUNT(*) AS total_orders
FROM time_segments
GROUP BY time_of_day
ORDER BY FIELD(time_of_day, 'Morning', 'Afternoon', 'Evening', 'Night');
```

	time_of_day	avg_order_value	total_orders
▶	Morning	36.55	1240
	Afternoon	40.69	9835
	Evening	36.35	9584
	Night	34.64	691

Q8 – Which are the best and the worst selling pizzas by quantity sold?

```sql
-- Best and worst selling pizzas by quantity sold
WITH sales_summary AS
    (SELECT
        pizza_name,
        SUM(quantity) AS total_units_sold
    FROM pizza_sales
    GROUP BY pizza_name)
    (SELECT
        'Best Selling' AS category,
        pizza_name,
        total_units_sold
    FROM sales_summary
    ORDER BY total_units_sold DESC
    LIMIT 5)
UNION ALL
    (SELECT
        'Worst Selling' AS category,
        pizza_name,
        total_units_sold
    FROM sales_summary
    ORDER BY total_units_sold ASC
    LIMIT 5);
```

category	pizza_name	total_units_sold
Best Selling	The Classic Deluxe Pizza	2453
Best Selling	The Barbecue Chicken Pizza	2432
Best Selling	The Hawaiian Pizza	2422
Best Selling	The Pepperoni Pizza	2418
Best Selling	The Thai Chicken Pizza	2371
Worst Selling	The Brie Carre Pizza	490
Worst Selling	The Mediterranean Pizza	934
Worst Selling	The Calabrese Pizza	937
Worst Selling	The Spinach Supreme Pizza	950
Worst Selling	The Soppressata Pizza	961