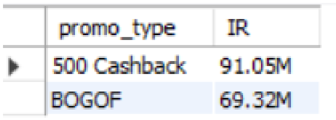
Promotion Type Analysis:

1.What are the top 2 promotion types that resulted in the highest Incremental Revenue?

CODE:

```
WITH ProductRevenue AS (
  SELECT
    promo_type,
  SUM(`quantity_sold(before_promo)` * base_price)
      AS Total_Revenue_Before_promotion,
  SUM(CASE
        WHEN promo_type = 'BOGOF' THEN base_price * 0.5 * (2 * `quantity_sold(after_promo)`)
        WHEN promo_type = '50% Off' THEN base_price * 0.50 * `quantity_sold(after_promo)`
        WHEN promo_type = '25% Off' THEN base_price * 0.75 * `quantity_sold(after_promo)`
        WHEN promo_type = '33% Off' THEN base_price * 0.67 * `quantity_sold(after_promo)`
        WHEN promo_type = '500 Cashback' THEN (base_price - 500) * `quantity_sold(after_promo)`
      END) AS Total_Revenue_After_promotion
    FROM
  retail_events_db.fact_events
  group by promo_type
),
IR_Calculation AS (
  SELECT
   promo_type, concat(round((Total_Revenue_After_promotion- Total_Revenue_Before_promotion)/1000000 ,2),"M") as IR
  FROM
    ProductRevenue
)
select promo_type, IR from IR_Calculation group by promo_type order by IR desc limit 2;
```

RESULT:



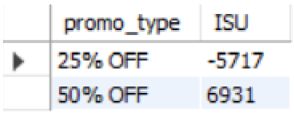
promo_type	IR
500 Cashback	91.05M
BOGOF	69.32M

2. What are the bottom 2 promotion types in terms of their impact on Incremental Sold Units?

CODE:

```
WITH CategorySales AS (
  SELECT
    promo_type,
    SUM(
      CASE
        WHEN fe.promo_type = 'BOGOF' THEN fe.`quantity_sold(after_promo)`* 2
        ELSE fe.`quantity_sold(after_promo)`
      END
    ) AS total_quantity_after_promo,
    SUM(`quantity_sold(before_promo)`) AS total_quantity_before_promo
  FROM
    retail_events_db.fact_events fe
  GROUP BY
   promo_type
),
ISU_Calculation AS (
  SELECT
    promo_type,
    (total_quantity_after_promo - total_quantity_before_promo) as ISU
          FROM
    CategorySales
)
select * from ISU_Calculation group by promo_type order by ISU asc limit 2;
```

RESULT:



promo_type	ISU
25% OFF	-5717
50% OFF	6931

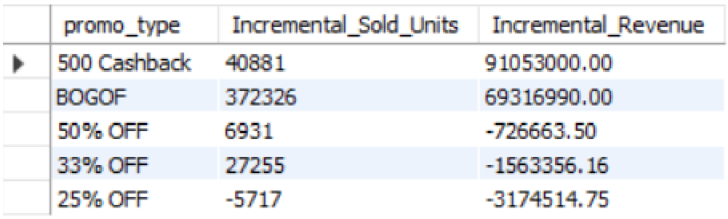
3. Is there a significant difference in the performance of discount-based promotions versus BOGOF (Buy One Get One Free) or cashback promotions?

CODE:

```
WITH ProductRevenue AS (
  SELECT
   promo_type,
  SUM(`quantity_sold(before_promo)` * base_price)
      AS Total_Revenue_Before_promotion,
  SUM(CASE
        WHEN promo_type = 'BOGOF' THEN base_price * 0.5 * (2 * `quantity_sold(after_promo)`)
```

```
            WHEN promo_type = '50% Off' THEN base_price * 0.50 * `quantity_sold(after_promo)`

            WHEN promo_type = '25% Off' THEN base_price * 0.75 * `quantity_sold(after_promo)`

            WHEN promo_type = '33% Off' THEN base_price * 0.67 * `quantity_sold(after_promo)`

            WHEN promo_type = '500 Cashback' THEN (base_price - 500) * `quantity_sold(after_promo)`

          END) AS Total_Revenue_After_promotion,

          SUM(

          CASE

            WHEN promo_type = 'BOGOF' THEN `quantity_sold(after_promo)`* 2

            ELSE `quantity_sold(after_promo)`

          END

        ) AS total_quantity_after_promo,

        SUM(`quantity_sold(before_promo)`) AS total_quantity_before_promo

        FROM

    retail_events_db.fact_events

    GROUP BY

      promo_type

),

IR_Calculation AS (

    SELECT promo_type, (total_quantity_after_promo - total_quantity_before_promo) as Incremental_Sold_Units,

    (Total_Revenue_After_promotion- Total_Revenue_Before_promotion) as Incremental_Revenue

FROM

        ProductRevenue

)

select * from IR_Calculation group by promo_type order by Incremental_Revenue desc ;
```

RESULT:



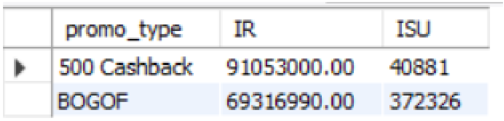
promo_type	Incremental_Sold_Units	Incremental_Revenue
500 Cashback	40881	91053000.00
BOGOF	372326	69316990.00
50% OFF	6931	-726663.50
33% OFF	27255	-1563356.16
25% OFF	-5717	-3174514.75

4. Which promotion types generated more incremental revenue or sold more units compared to the baseline period?

CODE:

```
WITH ProductRevenue AS (

    SELECT

      promo_type,
  SUM(`quantity_sold(before_promo)` * base_price)
      AS Total_Revenue_Before_promotion,
  SUM(CASE
        WHEN promo_type = 'BOGOF' THEN base_price * 0.5 * (2 * `quantity_sold(after_promo)`)
        WHEN promo_type = '50% Off' THEN base_price * 0.50 * `quantity_sold(after_promo)`
        WHEN promo_type = '25% Off' THEN base_price * 0.75 * `quantity_sold(after_promo)`
        WHEN promo_type = '33% Off' THEN base_price * 0.67 * `quantity_sold(after_promo)`
        WHEN promo_type = '500 Cashback' THEN (base_price - 500) * `quantity_sold(after_promo)`
      END) AS Total_Revenue_After_promotion,
      SUM(
      CASE
        WHEN promo_type = 'BOGOF' THEN `quantity_sold(after_promo)`* 2
        ELSE `quantity_sold(after_promo)`
      END
    ) AS total_quantity_after_promo,
    SUM(`quantity_sold(before_promo)`) AS total_quantity_before_promo
    FROM
  retail_events_db.fact_events
  GROUP BY
   promo_type
),
IR_Calculation AS (
  SELECT *, (total_quantity_after_promo - total_quantity_before_promo) as ISU,
  (Total_Revenue_After_promotion- Total_Revenue_Before_promotion) as IR
FROM
    ProductRevenue
)
select promo_type, IR ,ISU from IR_Calculation
group by promo_type
having sum(IR) > sum(Total_Revenue_Before_promotion) or sum(ISU) > sum(total_quantity_before_promo)
order by IR desc ;
```

RESULT:



	promo_type	IR	ISU
▶	500 Cashback	91053000.00	40881
	BOGOF	69316990.00	372326