

Ad-Hoc Queried

1. Provide a list of products with a base price greater than 500 and that are featured in promo type of 'BOGOF' (Buy One Get One Free). This information will help us identify high-value products that are currently being heavily discounted, which can be useful for evaluating our pricing and promotion strategies.

```
SELECT DISTINCT p.product_name as List_of_products
FROM dim_products as p
LEFT JOIN fact_events as e
  ON p.product_code = e.product_code
WHERE e.base_price > 500 AND
      e.promo_type = 'BOGOF';
```

2. Generate a report that provides an overview of the number of stores in each city. The results will be sorted in descending order of store counts, allowing us to identify the cities with the highest store presence. The report includes two essential fields: city and store count, which will assist in optimizing our retail operations.

```
SELECT city,
       COUNT(store_id) as stores
FROM dim_stores
GROUP BY city
ORDER BY stores desc;
```

3. Generate a report that displays each campaign along with the total revenue generated before and after the campaign? The report includes three key fields: campaign_name, total_revenue(before_promotion), total_revenue(after_promotion). This report should help in evaluating the financial impact of our promotional campaigns. (Display the values in millions)

```
SELECT
  c.campaign_name,
  SUM(e.base_price * e.`quantity_sold(before_promo)`) / 1000000 AS total_revenue_before_promo,
  SUM(e.base_price * e.`quantity_sold(after_promo)`) / 1000000 AS total_revenue_after_promo
FROM
  dim_campaigns c
JOIN
  fact_events e ON c.campaign_id = e.campaign_id
GROUP BY
  c.campaign_name;
```

4. Produce a report that calculates the Incremental Sold Quantity (ISU%) for each category during the Diwali campaign. Additionally, provide rankings for the categories based on their ISU%. The report will include three key fields: category, isu%, and rank order. This information will assist in assessing the category-wise success and impact of the Diwali campaign on incremental sales.

```

WITH DiwaliCampaign AS (
    SELECT
        c.campaign_id,
        c.campaign_name,
        e.product_code,
        e.`quantity_sold(before_promo)`,
        e.`quantity_sold(after_promo)`,
        p.category
    FROM
        dim_campaigns c
    JOIN
        fact_events e ON c.campaign_id = e.campaign_id
    JOIN
        dim_products p ON e.product_code = p.product_code
    WHERE
        c.campaign_name = 'Diwali'
)

SELECT
    category,
    ROUND(SUM(ISU_percentage), 2) AS ISU_percentage,
    RANK() OVER (ORDER BY SUM(ISU_percentage) DESC) AS rank_order
FROM (
    SELECT
        category,
        100 * SUM(Diwali.`quantity_sold(after_promo)` - Diwali.`quantity_sold(before_promo)`) /
        SUM(Diwali.`quantity_sold(before_promo)`) AS ISU_percentage
    FROM
        DiwaliCampaign Diwali
    GROUP BY
        category
) AS ISU
GROUP BY
    category
ORDER BY
    rank_order;

```

5. Create a report featuring the Top 5 products, ranked by Incremental Revenue Percentage (IR%), across all campaigns. The report will provide essential information including product name, category, and ir%. This analysis helps identify the most successful products in terms of incremental revenue across our campaigns, assisting in product optimization.

```

WITH Campaign AS (
    SELECT
        p.product_name,
        c.campaign_name,
        e.product_code,
        e.`quantity_sold(before_promo)`,
        e.`quantity_sold(after_promo)`,
        p.category,
        e.base_price
    FROM
        dim_campaigns c
    JOIN
        fact_events e ON c.campaign_id = e.campaign_id
    JOIN
        dim_products p ON e.product_code = p.product_code
    WHERE
        c.campaign_name IN ('Diwali', 'Sankranti')
)

```

```
SELECT
    product_name,
    category,
    ROUND(100 * SUM(base_price * ('quantity_sold(after_promo)' - 'quantity_sold(before_promo)')) /
    SUM(base_price * 'quantity_sold(before_promo)'), 2) AS ir_percentage
FROM
    Campaign
GROUP BY
    product_name, category
ORDER BY
    ir_percentage DESC
```