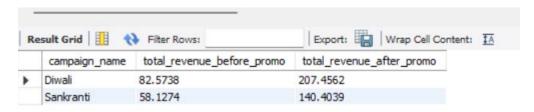
BUSINESS AD-HOC REQUEST

```
use retail events db;
select * from fact events;
select * from dim products;
select * from dim stores;
select * from dim_campaigns;
/*Question 1: Products with base price more than 500 and promotion type
BOGOF*/
select DISTINCT dp.product code, dp.product name
from dim products as dp
JOIN fact events as fte
ON dp.product code = fte.product code
where (fte.base price > 500) and (fte.promo type = "BOGOF");
 Export: Wrap Cell Content: IA
   product code product name
             Atliq_Double_Bedsheet_set
   P08
   P14
             Atliq_waterproof_Immersion_Rod
/*Question 2 - generate a report that provides number of stores in a
city. Arrange it in descending order*/
select city, count(*) as total_stores from dim stores
group by city
order by total stores DESC;
 Export: Wrap Cell Content: TA
    city
               total_stores
   Bengaluru
               10
   Chennai
   Hyderabad
               7
   Coimbatore
               5
   Visakhapatnam
   Madurai
   Mysuru
   Mangalore
               3
   Trivandrum
               2
               2
   Vijayawada
 Result 22 ×
```

/*Question 3 Campaign name and total revenue generation before promotion
and after promotion*/
SELECT dc.campaign name,

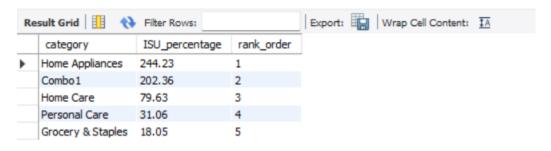
sum(round(fte.base_price*fte.`quantity_sold(before_promo)`,2))/1000000 as
total_revenue_before_promo,
sum(round(fte.base_price*fte.`quantity_sold(after_promo)`,2))/1000000 as
total_revenue_after_promo
FROM fact_events as fte
JOIN dim_campaigns as dc
ON fte.campaign_id = dc.campaign_id
GROUP BY dc.campaign_name
ORDER BY dc.campaign name ASC;



/*Question 4 -Calculation of ISU (Incremental Sold Quantity) for each category.
Ranking category based on ISU% */

Ranking category based on ISU% */

```
SELECT p.category,
  (round((SUM(fte.`quantity_sold(after_promo)`)-
SUM(fte.`quantity_sold(before_promo)`))/
        SUM(fte.`quantity_sold(before_promo)`)*100,2)) as ISU_percentage,
RANK() OVER (order by ((SUM(fte.`quantity_sold(after_promo)`)-
SUM(fte.`quantity_sold(before_promo)`))/
        SUM(fte.`quantity_sold(before_promo)`)*100) DESC) as rank_order
FROM dim_products as p
JOIN fact_events as fte
ON p.product_code = fte.product_code
WHERE fte.campaign_id = 'CAMP_DIW_01'
group by p.category;
```



/*Question 5 - Generate a query with top 5 products, ranked by IR %.
Report should have product name, category and ir%*/

```
SELECT p.product_name, p.category,
ROUND((SUM(fte.`quantity_sold(after_promo)`* fte.base_price) -
SUM(fte.`quantity_sold(before_promo)` * fte.base_price))/
SUM(fte.`quantity_sold(before_promo)` * fte.base_price)*100,2) AS
IR_Percentage
from dim_products as p
JOIN fact_events as fte
ON p.product_code = fte.product_code
group by p.product_name, p.category
Order by IR_percentage DESC
LIMIT 5;
```

