Business Requests using SQL

1 Provide a list of products with base price greater than 500 which feature in promo type BOGOF. This can help us in analyzing high value products that are currently being discounted, which can be useful in evaluating our pricing and promotion strategies

Solution:

```
SELECT DISTINCT(p.product_name),f.base_price,f.promo_type

FROM retail_events_db.fact_events  f

join dim_products p

on f.product_code=p.product_code

where f.base_price>500 and f.promo_type='BOGOF'

order by f.base_price desc;
```

Result:

product_name	base_price	promo_type
Atliq_Double_Bedsheet_set	1190	BOGOF
Atliq_waterproof_Immersion_Rod	1020	BOGOF

2. Generate a report that finds the number of stores in each city, sorted in descending order of store counts.This will give us an idea of city with highest store count.The report includes two essential fields city and store count which will assist in optimizing our retail operations

Solution:

```
SELECT s.city,count(store_id) as store_count

FROM retail_events_db.fact_events f

join  dim_stores s

using (store_id)

GROUP BY s.city

order by store_count desc;
```

Result:

city	store_count
Bengaluru	300
Chennai	240
Hyderabad	210
Coimbatore	150
Visakhapatnam	150
Madurai	120
Mysuru	120
Mangalore	90
Vijayawada	60

Trivandrum	60

3. Generate a report that displays each campaign with total revenue before and after the campaign. The report includes 3 key fields: Campaign name, Total revenue before promotion and after promotion. This report should help in evaluating the financial impact of our promotions(Display values in millions)

Solution:

```
with cte1(campaign_id,revenue_after) as
(select campaign_id,
case
   when promo_type='50% OFF' then base_price*50/100 *fe.`quantity_sold(after_promo)`
   when promo_type='25% OFF' then base_price *75/100*fe.`quantity_sold(after_promo)`
   when promo_type='33% OFF' then base_price*67/100*fe.`quantity_sold(after_promo)`
   when promo_type='BOGOF' then base_price*fe.`quantity_sold(after_promo)`*50/100
   when promo_type= '500 Cashback' then (base_price-500)*fe.`quantity_sold(after_promo)`
 end
 as revenue_after
 from fact_events fe)
select dc.campaign_name,
round(sum(f.`quantity_sold(before_promo)`*f.base_price)/1000000,2) as total_revenue_bf_Mill,
round(sum(c.revenue_after)/1000000,2) as total_revenue_af_Mill
from fact_events f
join cte1 c
using(campaign_id)
join dim_campaigns dc
USING(campaign_id)
group by campaign_id
```

Result:

campaign_name	total_revenue_bf_Mill	total_revenue_af_Mill
Sankranti	43595.57	65771.86
Diwali	61930.32	120216.5

4. Create a report that calculates Incremental Sold Quantity %(ISU%) for each category during the Diwali campaign . Additionaly provide rankings for the categories based on their ISU%. The report will include 3 fields: category, ISU%, rank. This will assist in evaluating the success of category wise success and impact of the Diwali campaign on incremental sales.

Solution:

```
with cte1 as
(select p.category,
(sum(`quantity_sold(after_promo)`)-sum(`quantity_sold(before_promo)`))/sum(`quantity_sold(before_promo)`)*100 as ISUperc
from fact_events f
join dim_products p
on f.product_code=p.product_code
where f.campaign_id="CAMP_DIW_01"
GROUP BY p.category
)
select category,ISUperc,DENSE_RANK() over(order by ISUperc desc)
as ranking
from cte1;
```

Result:

category	ISUperc	ranking
Home Appliances	244.2256	1
Combo1	202.3584	2
Home Care	79.6338	3
Personal Care	31.0574	4
Grocery & Staples	18.0478	5