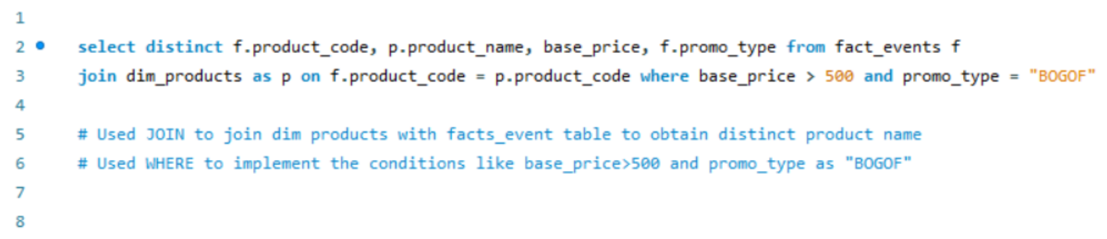
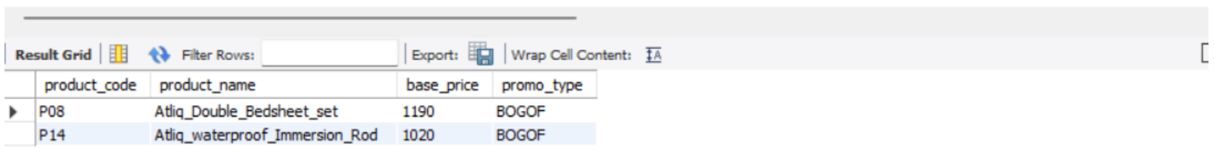
AD-HOC REQUEST 1

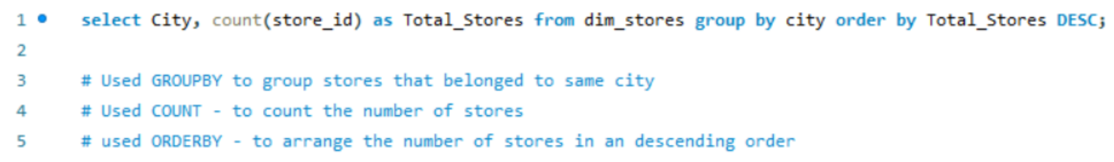


```sql
select distinct f.product_code, p.product_name, base_price, f.promo_type from fact_events f
join dim_products as p on f.product_code = p.product_code where base_price > 500 and promo_type = "BOGOF"

# Used JOIN to join dim products with facts_event table to obtain distinct product name
# Used WHERE to implement the conditions like base_price>500 and promo_type as "BOGOF"
```

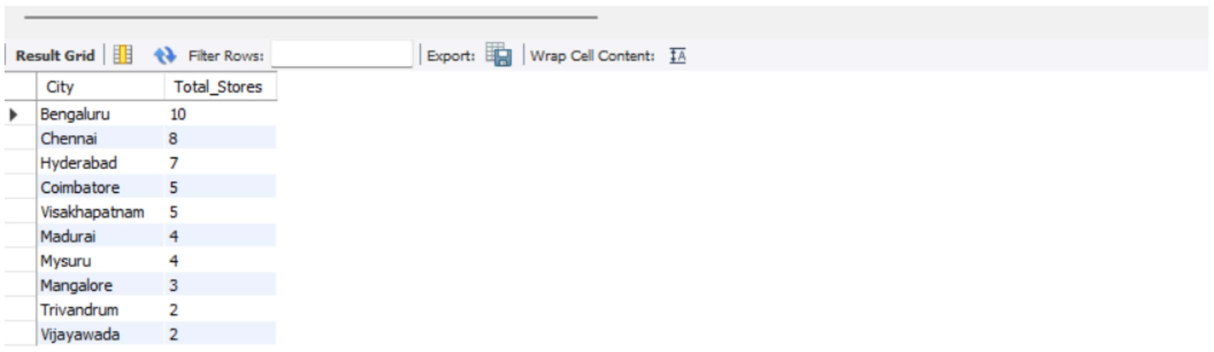


product_code	product_name	base_price	promo_type
P08	Atliq_Double_Bedsheet_set	1190	BOGOF
P14	Atliq_waterproof_Immersion_Rod	1020	BOGOF

AD-HOC REQUEST -2

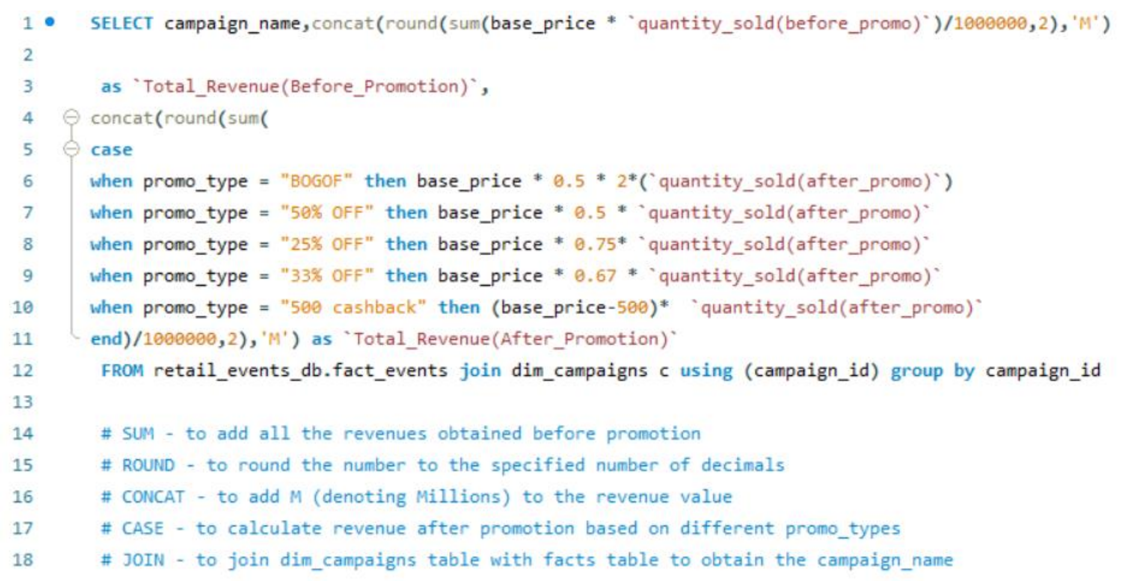


```sql
select City, count(store_id) as Total_Stores from dim_stores group by city order by Total_Stores DESC;

# Used GROUPBY to group stores that belonged to same city
# Used COUNT - to count the number of stores
# used ORDERBY - to arrange the number of stores in an descending order
```

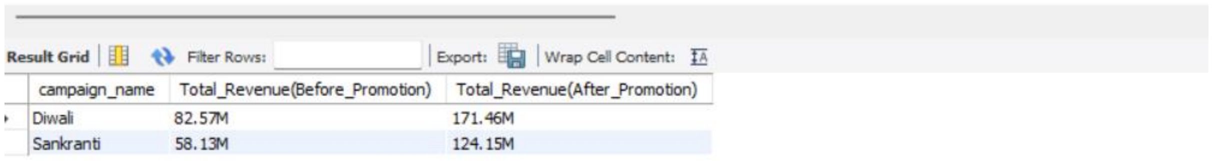


City	Total_Stores
Bengaluru	10
Chennai	8
Hyderabad	7
Coimbatore	5
Visakhapatnam	5
Madurai	4
Mysuru	4
Mangalore	3
Trivandrum	2
Vijayawada	2

AD-HOC REQUEST -3



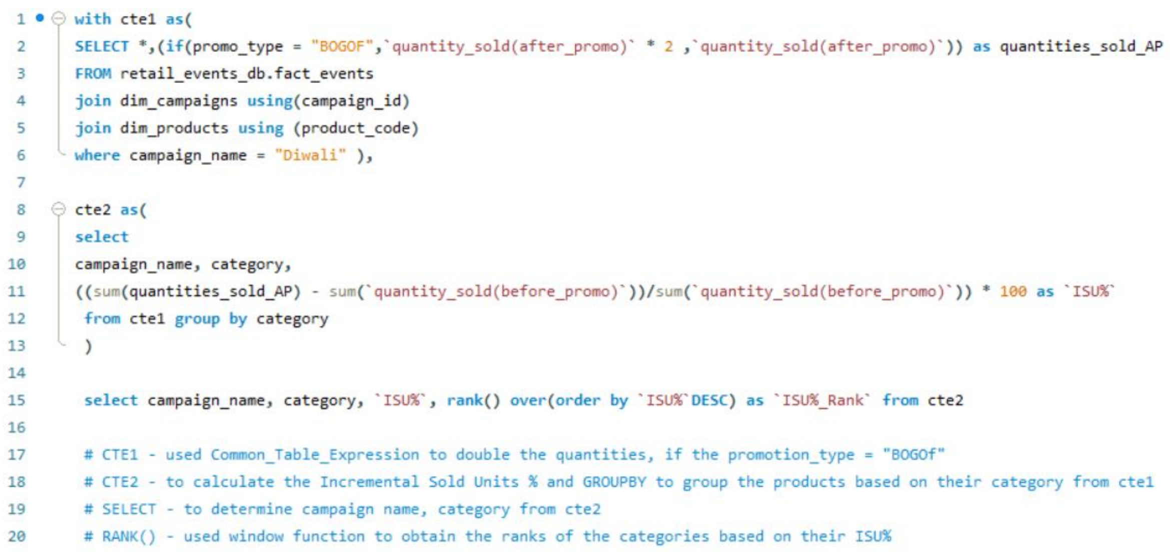
```sql
SELECT campaign_name,concat(round(sum(base_price * `quantity_sold(before_promo)`)/1000000,2),'M')

 as `Total_Revenue(Before_Promotion)`,
concat(round(sum(
case
when promo_type = "BOGOF" then base_price * 0.5 * 2*(`quantity_sold(after_promo)`)
when promo_type = "50% OFF" then base_price * 0.5 * `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)/1000000,2),'M') as `Total_Revenue(After_Promotion)`
 FROM retail_events_db.fact_events join dim_campaigns c using (campaign_id) group by campaign_id

 # SUM - to add all the revenues obtained before promotion
 # ROUND - to round the number to the specified number of decimals
 # CONCAT - to add M (denoting Millions) to the revenue value
 # CASE - to calculate revenue after promotion based on different promo_types
 # JOIN - to join dim_campaigns table with facts table to obtain the campaign_name
```



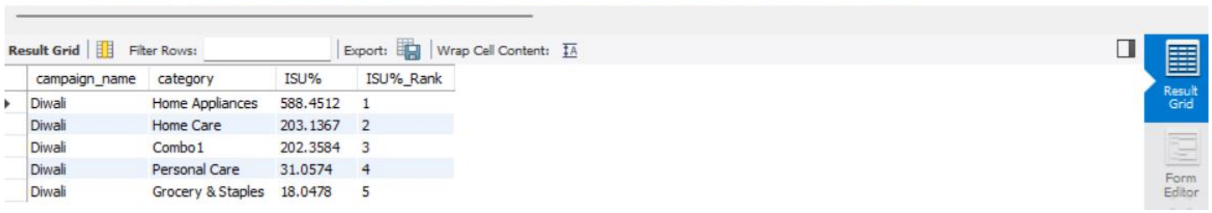
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

campaign_name	Total_Revenue(Before_Promotion)	Total_Revenue(After_Promotion)
Diwali	82.57M	171.46M
Sankranti	58.13M	124.15M

AD-HOC REQUEST – 4



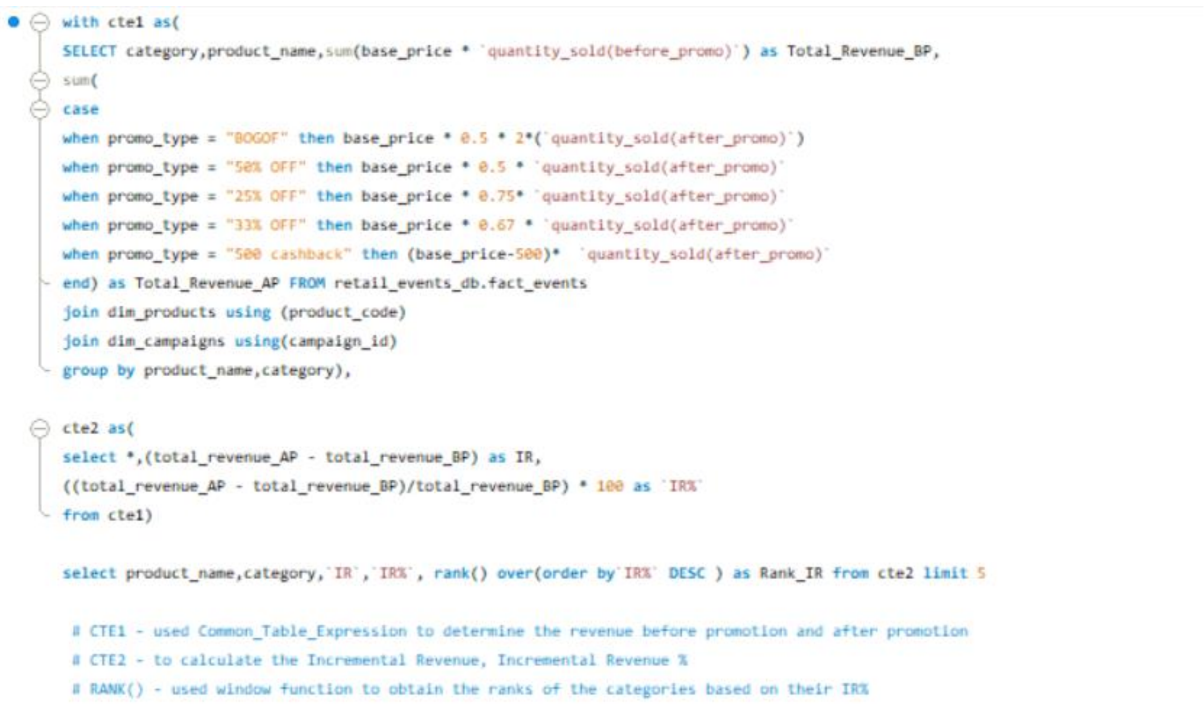
```sql
with cte1 as(
SELECT *,(if(promo_type = "BOGOF",`quantity_sold(after_promo)` * 2 ,`quantity_sold(after_promo)`)) as quantities_sold_AP
FROM retail_events_db.fact_events
join dim_campaigns using(campaign_id)
join dim_products using (product_code)
where campaign_name = "Diwali" ),

cte2 as(
select
campaign_name, category,
((sum(quantities_sold_AP) - sum(`quantity_sold(before_promo)`))/sum(`quantity_sold(before_promo)`)) * 100 as `ISU%`
 from cte1 group by category
 )

 select campaign_name, category, `ISU%`, rank() over(order by `ISU%`DESC) as `ISU%_Rank` from cte2

 # CTE1 - used Common_Table_Expression to double the quantities, if the promotion_type = "BOGOf"
 # CTE2 - to calculate the Incremental Sold Units % and GROUPBY to group the products based on their category from cte1
 # SELECT - to determine campaign name, category from cte2
 # RANK() - used window function to obtain the ranks of the categories based on their ISU%
```



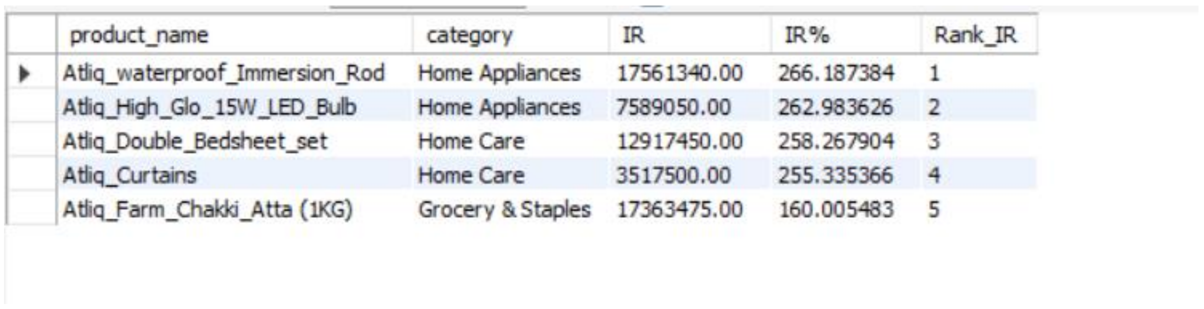
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

campaign_name	category	ISU%	ISU%_Rank
Diwali	Home Appliances	588.4512	1
Diwali	Home Care	203.1367	2
Diwali	Combo1	202.3584	3
Diwali	Personal Care	31.0574	4
Diwali	Grocery & Staples	18.0478	5

AD-HOC REQUEST – 5



```sql
with cte1 as(
SELECT category,product_name,sum(base_price * `quantity_sold(before_promo)`) as Total_Revenue_BP,
sum(
case
when promo_type = "BOGOF" then base_price * 0.5 * 2*(`quantity_sold(after_promo)`)
when promo_type = "50% OFF" then base_price * 0.5 * `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_AP FROM retail_events_db.fact_events
join dim_products using (product_code)
join dim_campaigns using(campaign_id)
group by product_name,category),

cte2 as(
select *,(total_revenue_AP - total_revenue_BP) as IR,
((total_revenue_AP - total_revenue_BP)/total_revenue_BP) * 100 as `IR%`
from cte1)

select product_name,category,`IR`,`IR%`, rank() over(order by`IR%` DESC ) as Rank_IR from cte2 limit 5

 # CTE1 - used Common_Table_Expression to determine the revenue before promotion and after promotion
 # CTE2 - to calculate the Incremental Revenue, Incremental Revenue %
 # RANK() - used window function to obtain the ranks of the categories based on their IR%
```



	product_name	category	IR	IR%	Rank_IR
▶	Atliq_waterproof_Immersion_Rod	Home Appliances	17561340.00	266.187384	1
	Atliq_High_Glo_15W_LED_Bulb	Home Appliances	7589050.00	262.983626	2
	Atliq_Double_Bedsheet_set	Home Care	12917450.00	258.267904	3
	Atliq_Curtains	Home Care	3517500.00	255.335366	4
	Atliq_Farm_Chakki_Atta (1KG)	Grocery & Staples	17363475.00	160.005483	5