Codebasics Resume Project Challenge 9

Analyze Promotions and Provide Tangible Insights to Sales Director

Problem Statement:

AtliQ Mart is a retail giant with over 50 supermarkets in the southern region of India. All their 50 stores ran a massive promotion during the **Diwali 2023** and **Sankranti 2024** (festive time in India) on their AtliQ branded products.

Now the sales director wants to understand which promotions did well and which did not so that they can make informed decisions for their next promotional period.

Business Request:

Start by importing the **'retail_events_db'** database into **MySQL Workbench**. Craft SQL queries to address the specified business questions.

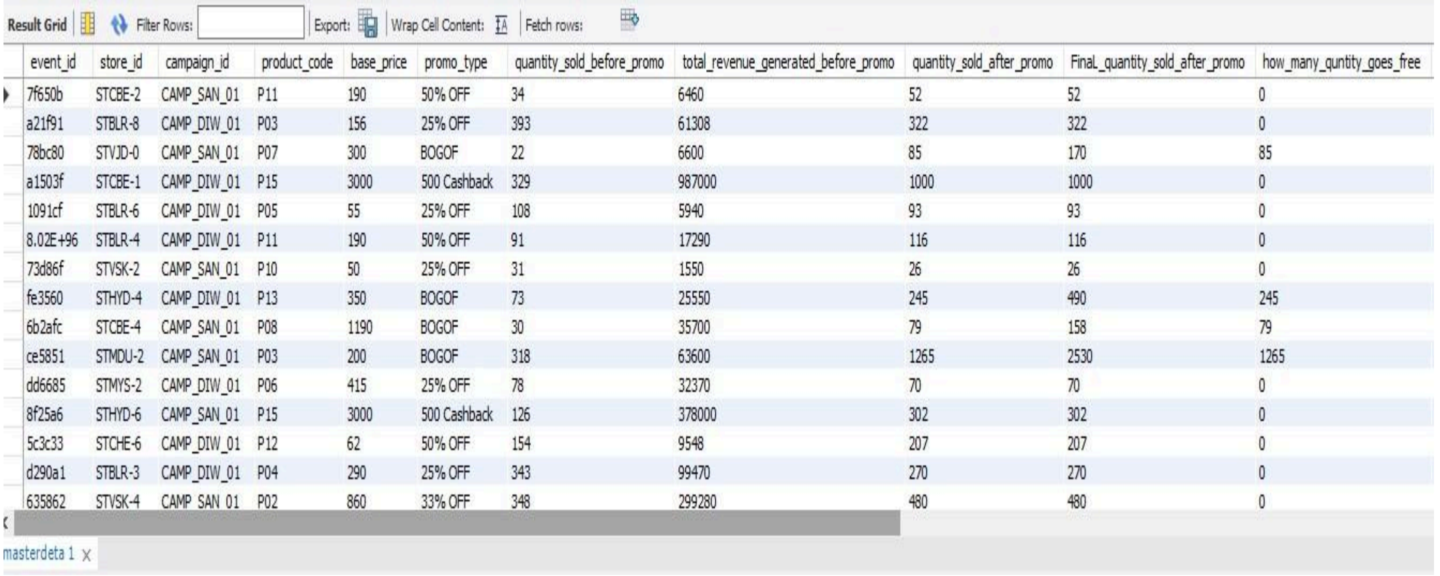
Note:-

I'm not as proficient in SQL as I aspire to be. To mitigate potential challenges in data retrieval and analysis, I've enriched our primary database, retail_event_db, by incorporating supplementary datasets using Excel. This strategic measure aims to facilitate smoother navigation and analysis for upcoming tasks and requests.

Eg-

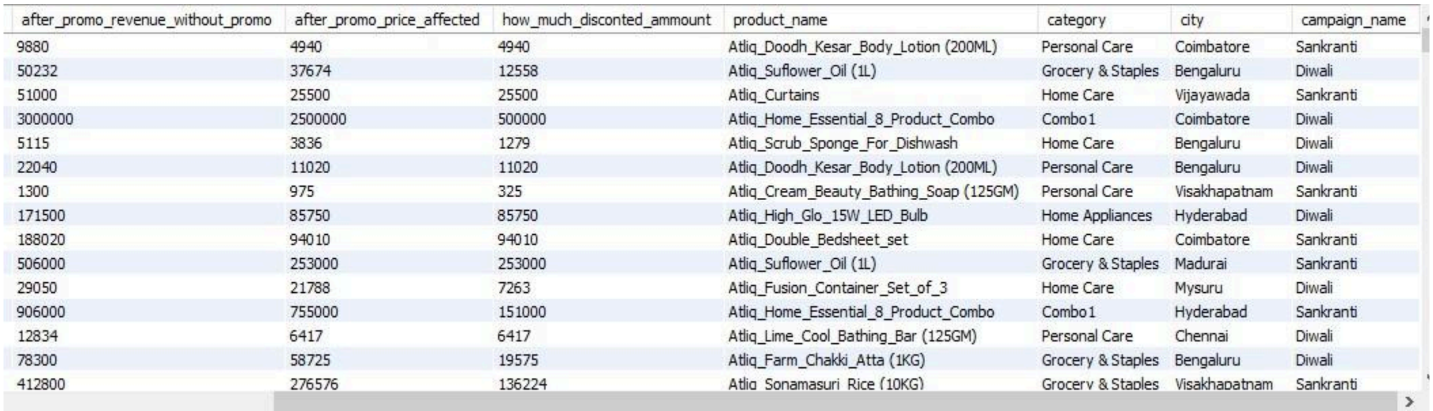
```
SELECT * FROM masterdeta;
```

Output-



event_id	store_id	campaign_id	product_code	base_price	promo_type	quantity_sold_before_promo	total_revenue_generated_before_promo	quantity_sold_after_promo	FinaL_quantity_sold_after_promo	how_many_quntity_goes_free
7f650b	STCBE-2	CAMP_SAN_01	P11	190	50% OFF	34	6460	52	52	0
a21f91	STBLR-8	CAMP_DIW_01	P03	156	25% OFF	393	61308	322	322	0
78bc80	STVJD-0	CAMP_SAN_01	P07	300	BOGOF	22	6600	85	170	85
a1503f	STCBE-1	CAMP_DIW_01	P15	3000	500 Cashback	329	987000	1000	1000	0
1091cf	STBLR-6	CAMP_DIW_01	P05	55	25% OFF	108	5940	93	93	0
8.02E+96	STBLR-4	CAMP_DIW_01	P11	190	50% OFF	91	17290	116	116	0
73d86f	STVSK-2	CAMP_SAN_01	P10	50	25% OFF	31	1550	26	26	0
fe3560	STHYD-4	CAMP_DIW_01	P13	350	BOGOF	73	25550	245	490	245
6b2afc	STCBE-4	CAMP_SAN_01	P08	1190	BOGOF	30	35700	79	158	79
ce5851	STMDU-2	CAMP_SAN_01	P03	200	BOGOF	318	63600	1265	2530	1265
dd6685	STMYS-2	CAMP_DIW_01	P06	415	25% OFF	78	32370	70	70	0
8f25a6	STHYD-6	CAMP_SAN_01	P15	3000	500 Cashback	126	378000	302	302	0
5c3c33	STCHE-6	CAMP_DIW_01	P12	62	50% OFF	154	9548	207	207	0
d290a1	STBLR-3	CAMP_DIW_01	P04	290	25% OFF	343	99470	270	270	0
635862	STVSK-4	CAMP_SAN_01	P02	860	33% OFF	348	299280	480	480	0

and



after_promo_revenue_without_promo	after_promo_price_affected	how_much_disconted_ammount	product_name	category	city	campaign_name
9880	4940	4940	Atliq_Doodh_Kesar_Body_Lotion (200ML)	Personal Care	Coimbatore	Sankranti
50232	37674	12558	Atliq_Suflower_Oil (1L)	Grocery & Staples	Bengaluru	Diwali
51000	25500	25500	Atliq_Curtains	Home Care	Vijayawada	Sankranti
3000000	2500000	500000	Atliq_Home_Essential_8_Product_Combo	Combo1	Coimbatore	Diwali
5115	3836	1279	Atliq_Scrub_Sponge_For_Dishwash	Home Care	Bengaluru	Diwali
22040	11020	11020	Atliq_Doodh_Kesar_Body_Lotion (200ML)	Personal Care	Bengaluru	Diwali
1300	975	325	Atliq_Cream_Beauty_Bathing_Soap (125GM)	Personal Care	Visakhapatnam	Sankranti
171500	85750	85750	Atliq_High_Glo_15W_LED_Bulb	Home Appliances	Hyderabad	Diwali
188020	94010	94010	Atliq_Double_Bedsheet_set	Home Care	Coimbatore	Sankranti
506000	253000	253000	Atliq_Suflower_Oil (1L)	Grocery & Staples	Madurai	Sankranti
29050	21788	7263	Atliq_Fusion_Container_Set_of_3	Home Care	Mysuru	Diwali
906000	755000	151000	Atliq_Home_Essential_8_Product_Combo	Combo1	Hyderabad	Sankranti
12834	6417	6417	Atliq_Lime_Cool_Bathing_Bar (125GM)	Personal Care	Chennai	Diwali
78300	58725	19575	Atliq_Farm_Chakki_Atta (1KG)	Grocery & Staples	Bengaluru	Diwali
412800	276576	136224	Atliq_Sonamasuri_Rice (10KG)	Grocery & Staples	Visakhapatnam	Sankranti

Request: 1-

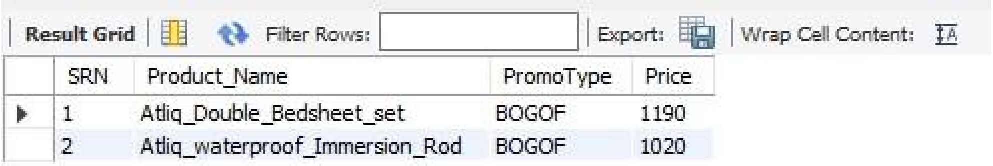
Provide a list of products with a base price greater than 500 and that are featured in promo type of 'BOGOF' (Buy One Get One Free).

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.

# query-

```
SELECT
        ROW_NUMBER () OVER  ( ORDER BY  product_name ) AS  SRN,
        Product_name  AS  Product_Name,
        promo_type  AS  PromoType,
        base_price  AS  Price

FROM
  (SELECT  DISTINCT  product_name , promo_type , base_price  FROM  masterdeta  WHERE
base_price  >  500  AND  promo_type  =  'BOGOF' )  AS  Unique_Products;
```

Output-



SRN	Product_Name	PromoType	Price
1	Atliq_Double_Bedsheet_set	BOGOF	1190
2	Atliq_waterproof_Immersion_Rod	BOGOF	1020

Query work function- This SQL query retrieves unique products with their corresponding promotional types and base prices, where the base price is greater than 500 and the promotional type is 'BOGOF' (Buy One Get One Free). The query then assigns a serial number to each result row based on the product name's alphabetical order.

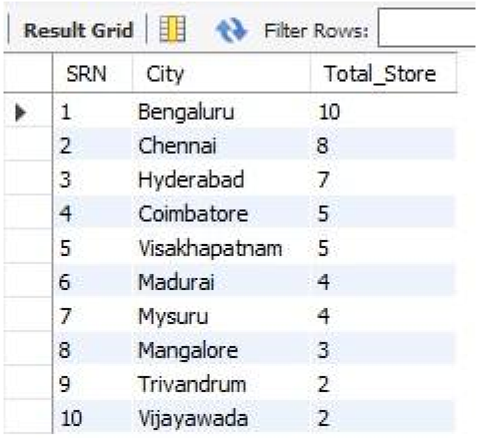
Request: 2-

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.

# query-

```
SELECT
 ROW_NUMBER () OVER (ORDER BY COUNT (store_id) DESC) AS SRN,
 city AS City,
 COUNT (DISTINCT store_id ) AS Total_Store
 FROM masterdeta
 GROUP BY city;
```

Output-



SRN	City	Total_Store
1	Bengaluru	10
2	Chennai	8
3	Hyderabad	7
4	Coimbatore	5
5	Visakhapatnam	5
6	Madurai	4
7	Mysuru	4
8	Mangalore	3
9	Trivandrum	2
10	Vijayawada	2

Query work function- This query utilizes the ROW_NUMBER() function alongside the OVER() clause to assign a sequential row number to each row in the result set. The rows are ordered based on the descending count of store_id within each city group.

Additionally, it selects the city column as "City" and calculates the count of distinct store_id values for each city, labeling it as "Total_Store". This calculation is performed within the masterdeta table.

Finally, the results are grouped by the city column.

Request: 3-

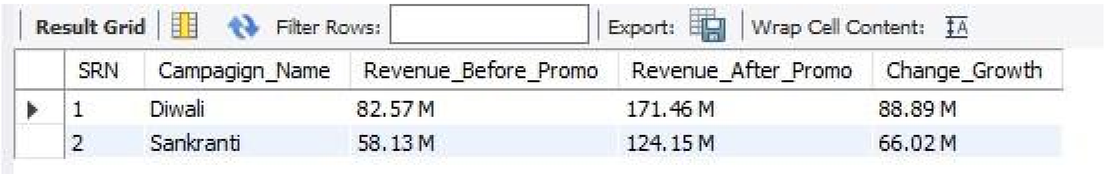
Generate a report that displays each campaign along with the total revenue generated before and after the campaign? The report includes three key fields: campaign_name, total_revenue(before_promotion), total_revenue(after_promotion).

This report should help in evaluating the financial impact of our promotional campaigns. (Display the values in millions)

# query-

```
SELECT
 ROW_NUMBER () OVER ( ORDER BY campaign_name ) AS SRN,
 campaign_name AS Campagign_Name,
 CONCAT ( FORMAT ( SUM (total_revenue_generated_before_promo) / 1000000, 2 ) , ' M ') AS
Revenue_Before_Promo,
 CONCAT (FORMAT (SUM ( after_promo_price_affected ) /1000000, 2 ) , ' M' ) AS
Revenue_After_Promo,
 CONCAT ( FORMAT (( SUM (after_promo_price_affected ) - SUM
(total_revenue_generated_before_promo )) / 1000000 , 2 ) , ' M' ) AS Change_Growth
 FROM masterdeta
 GROUP BY campaign_name;
```

Output-



SRN	Campagign_Name	Revenue_Before_Promo	Revenue_After_Promo	Change_Growth
1	Diwali	82.57 M	171.46 M	88.89 M
2	Sankranti	58.13 M	124.15 M	66.02 M

Query work function- This SQL query utilizes the **`ROW_NUMBER()`** function to assign a sequential **number to each row** based on the order of the **'campaign_name'**. It then selects the 'campaign_name' column as 'Campagign_Name'.

Next, it calculates the total revenue generated before the promotion **('total_revenue_generated_before_promo')** and after the promotion price is affected **('after_promo_price_affected')** for each campaign. These values are formatted to display in **million units ('M').**

Finally, it calculates the **change in growth by subtracting the revenue after the promotion from the revenue before the promotion**. All calculated metrics are formatted to display in million units ('M'). The results are grouped by 'campaign_name'.

Request: 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%.

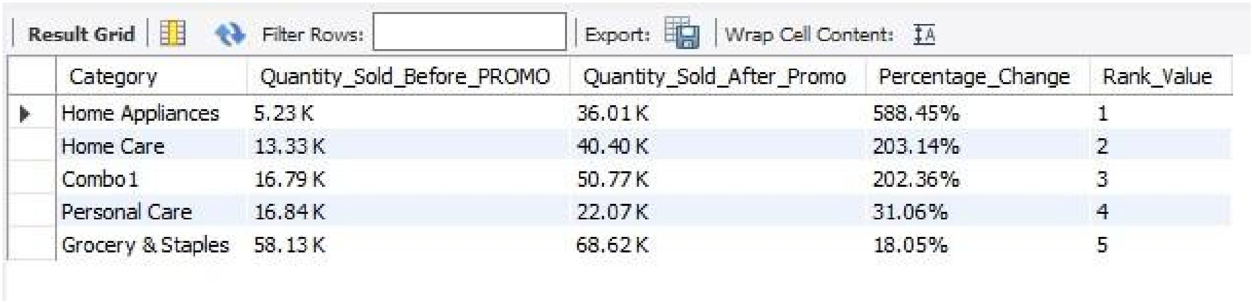
The report will include three key fields: category, isu%, and rank order. This information will assist in assessing the category-wise success and impact of the Diwali campaign on incremental sales.

Note: ISU% (Incremental Sold Quantity Percentage) is calculated as the percentage increase/decrease in quantity sold (after promo) compared to quantity sold (before promo)

# query-

```
SELECT
category  AS  Category,
CONCAT ( FORMAT ( SUM ( quantity_sold_before_promo ) / 1000,2 ) , ' K' )  AS
Quantity_Sold_Before_PROMO,

CONCAT ( FORMAT ( SUM ( FinaL_quantity_sold_after_promo  / 1000,2 ), ' K' ) AS
Quantity_Sold_After_Promo,

CONCAT ( FORMAT (((  SUM ( FinaL_quantity_sold_after_promo) - SUM ( quantity_sold_
before_promo )) / SUM( quantity_sold_before_promo )) * 100, 2), '%') AS Percentage_Change,

ROW_NUMBER () OVER ( ORDER BY (( SUM ( FinaL_quantity_sold_after_promo) - SUM (quantity
_sold_before_promo )) / SUM (quantity_sold_before_promo )) DESC) AS Rank_Value

FROM masterdeta
WHERE  campaign_name = 'Diwali'
GROUP BY category;
```

Output-



Category	Quantity_Sold_Before_PROMO	Quantity_Sold_After_Promo	Percentage_Change	Rank_Value
Home Appliances	5.23 K	36.01 K	588.45%	1
Home Care	13.33 K	40.40 K	203.14%	2
Combo1	16.79 K	50.77 K	202.36%	3
Personal Care	16.84 K	22.07 K	31.06%	4
Grocery & Staples	58.13 K	68.62 K	18.05%	5

This SQL query retrieves data related to sales performance before and after a campaign named 'Diwali' for different categories. It calculates the quantity sold before and after the promotion along with the percentage change in sales .Additionally, it assigns a rank to each category based on the percentage change, ordering them in descending order of improvement.

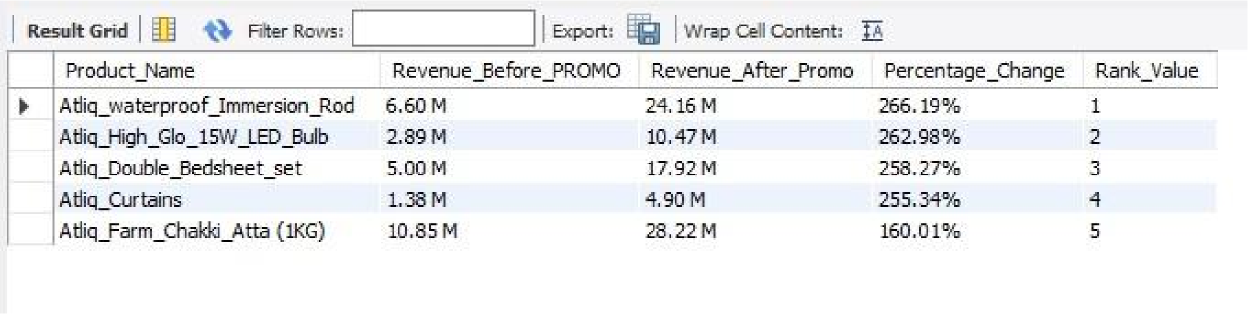
Request: 5-

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.

# query-

```
SELECT
  Product_Name,
  Revenue_Before_PROMO,
  Revenue_After_Promo,
  Percentage_Change,
  Rank_Value

FROM (
  SELECT
    product_name AS Product_Name,

    CONCAT ( FORMAT( SUM ( total_revenue_generated_before_promo ) / 1000000, 2 ), ' M' ) AS
Revenue_Before_PROMO,

    CONCAT ( FORMAT ( SUM ( after_promo_price_affected ) / 1000000, 2 ), ' M' ) AS
Revenue_After_Promo,

    CONCAT ( FORMAT ((( SUM (after_promo_price_affected ) - SUM(total_revenu e
_generated_before_promo )) / SUM ( total_revenue_generated_before_promo )) * 100, 2 ), '%' ) AS
Percentage_Change,

    ROW_NUMBER () OVER ( ORDER BY (( SUM (after_promo_price_affected) - SUM ( total_
revenue_generated_before_promo )) / SUM  (total_revenue_generated_before_promo )) DESC ) AS
Rank_Value

    FROM masterdeta
    GROUP BY product_name
) AS RankedResults
WHERE Rank_Value <= 5;
```

Output-



Product_Name	Revenue_Before_PROMO	Revenue_After_Promo	Percentage_Change	Rank_Value
Atliq_waterproof_Immersion_Rod	6.60 M	24.16 M	266.19%	1
Atliq_High_Glo_15W_LED_Bulb	2.89 M	10.47 M	262.98%	2
Atliq_Double_Bedsheet_set	5.00 M	17.92 M	258.27%	3
Atliq_Curtains	1.38 M	4.90 M	255.34%	4
Atliq_Farm_Chakki_Atta (1KG)	10.85 M	28.22 M	160.01%	5

This SQL query retrieves data on product revenue before and after promotions, calculates the percentage change, and ranks the products based on the percentage change in descending order.

Analyzed by-
Saddam Ansari @Aspiring Data Analyst