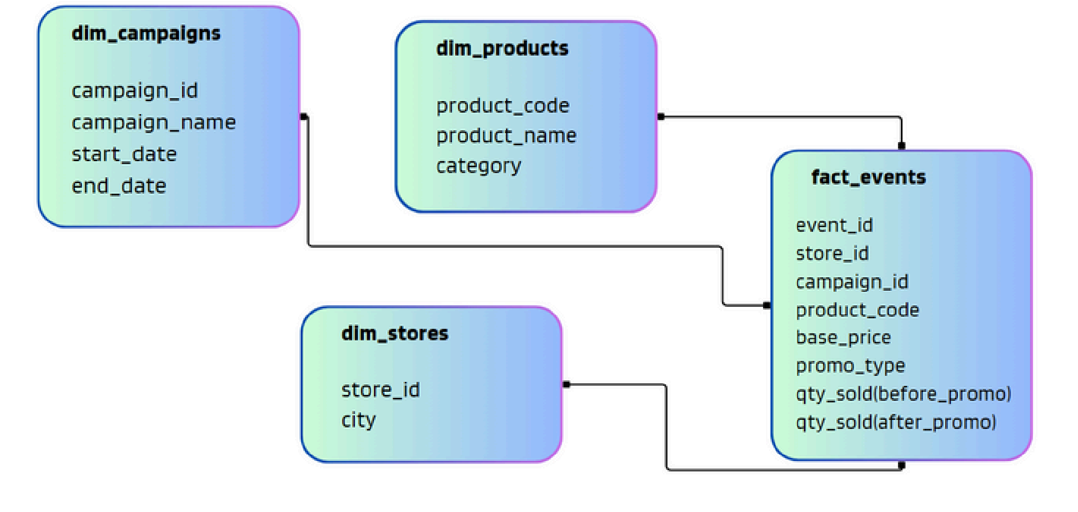
PROMOTIONAL SALES ANALYSIS

MYSQL PROJECT

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.

AtliQ Store Dataset



AD_HOC_INSIGHTS

BY KARAN CHASKAR

PROMOTIONAL SALES ANALYSIS

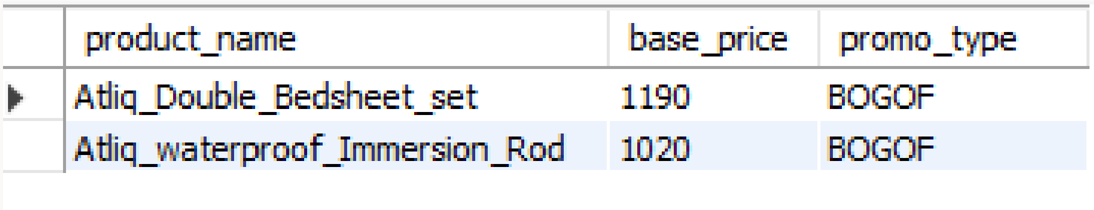
MYSQL PROJECT

Q1. 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.

- Input :

```
select distinct(dp.product_name), fe.base_price, fe.promo_type
from dim_products as dp
join fact_events as fe
on dp.product_code = fe.product_code
where fe.base_price > 500
and promo_type = 'BOGOF';
```

- Output :



product_name	base_price	promo_type
Atliq_Double_Bedsheet_set	1190	BOGOF
Atliq_waterproof_Immersion_Rod	1020	BOGOF

AD_HOC_INSIGHTS

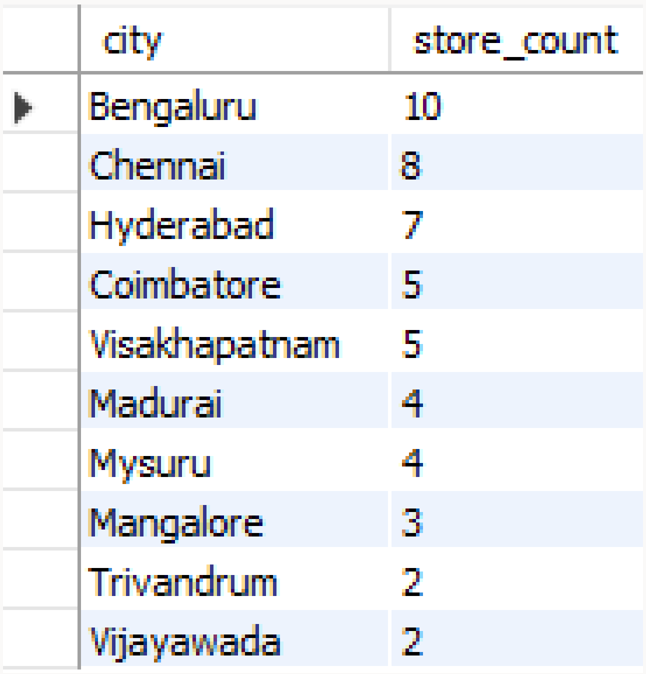
BY KARAN CHASKAR

Q2. 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

- Input :

```
select city, count(store_id) as store_count
from dim_stores group by city
order by store_count desc;
```

- Output :



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

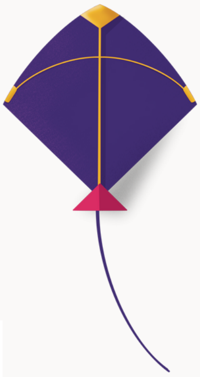
PROMOTIONAL SALES ANALYSIS

MYSQL PROJECT

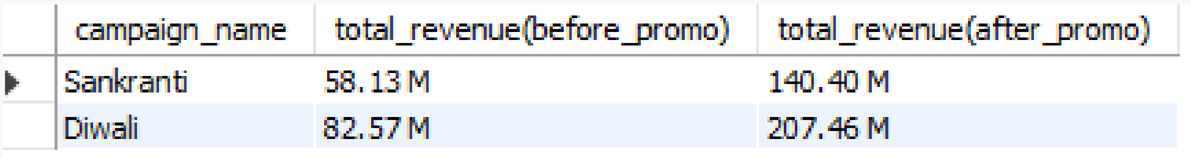
Q3. Generate a report that displays each campaign along with the total revenue generated before and after the campaign? The report includes three key fields: campaign_name, total_revenue (before_promotion), total_revenue(after_promotion). This report should help in evaluating the financial impact of our promotional campaigns. (Display the values in millions)

- Input :

```
select dc.campaign_name, concat(round(sum(base_price *
`quantity_sold(before_promo)`)/1000000,2),' M') as `total_revenue(before_promo)`,
concat(round(sum(base_price * `quantity_sold(after_promo)`)
/1000000,2),' M') as `total_revenue(after_promo)`
from dim_campaigns dc
join fact_events fe
on dc.campaign_id = fe.campaign_id
group by dc.campaign_name;
```



- Output :



campaign_name	total_revenue(before_promo)	total_revenue(after_promo)
Sankranti	58.13 M	140.40 M
Diwali	82.57 M	207.46 M

PROMOTIONAL SALES ANALYSIS

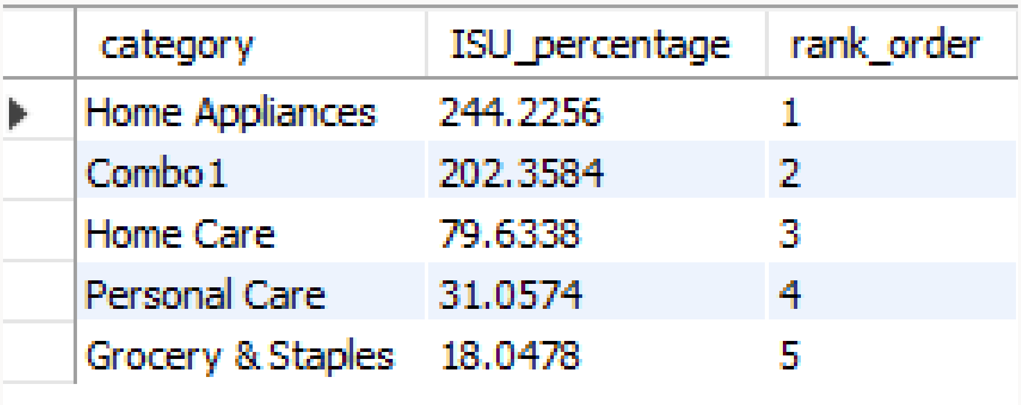
MYSQL PROJECT

Q4. 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.

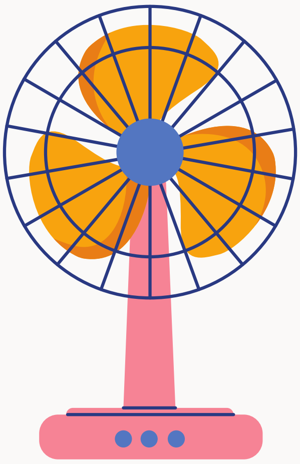
- Input :

```
select category,
((sum(`quantity_sold(after_promo)`) - sum(`quantity_sold(before_promo)`)) /
sum(`quantity_sold(before_promo)`)) * 100 as ISU_percentage,
rank() over (order by((sum(`quantity_sold(after_promo)`) - sum(`quantity_sold(before_promo)`)) /
sum(`quantity_sold(before_promo)`)) desc) as rank_order
from dim_products as dp
join fact_events as fe on dp.product_code = fe.product_code
where  campaign_id = 'CAMP_DIW_01'
group by category
order by ISU_percentage desc;
```

- Output :



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



PROMOTIONAL SALES ANALYSIS

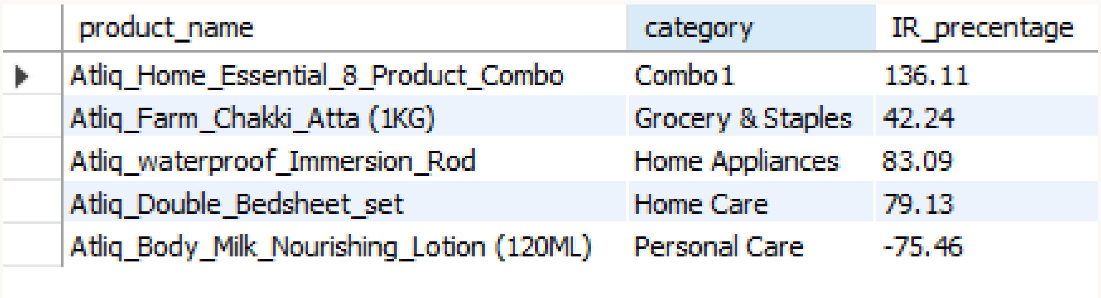
MYSQL PROJECT

Q5. 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.

- Input :

```
with promo_calulation as
(select dp.product_name, dp.category,
sum(case
when fe.promo_type = "BOGOF" then (fe.base_price-(fe.base_price*0.5))* `quantity_sold(after_promo)`
when fe.promo_type = "33% OFF" then (fe.base_price-(fe.base_price*0.33))* `quantity_sold(after_promo)`
when fe.promo_type = "25% OFF" then (fe.base_price-(fe.base_price*0.25)) * `quantity_sold(after_promo)`
when fe.promo_type = "500 Cashback" then (fe.base_price-500) * `quantity_sold(after_promo)`
when fe.promo_type = " 50% OFF" then (fe.base_price-(fe.base_price*0.5)) * `quantity_sold(after_promo)`
else 0 end ) as total_revenue_after_promo,
sum(fe.base_price* `quantity_sold(before_promo)`) as total_revenue_before_promo
from dim_products dp left join fact_events fe on dp.product_code = fe.product_code
group by dp.product_name,dp.category),
product_rank as (select product_name, category,
round(100*(total_revenue_after_promo-total_revenue_before_promo)/ total_revenue_before_promo,2)
as IR_precentage,
rank() over(partition by category order by round(100*(total_revenue_after_promo-total_revenue_before_promo)/
total_revenue_before_promo,2) desc) as "Ir_rank" from promo_calulation
order by IR_precentage desc, category asc)
select product_name,category,IR_precentage from product_rank
where Ir_rank<2
order by category ;
```

- Output :



product_name	category	IR_precentage
Atliq_Home_Essential_8_Product_Combo	Combo1	136.11
Atliq_Farm_Chakki_Atta (1KG)	Grocery & Staples	42.24
Atliq_waterproof_Immersion_Rod	Home Appliances	83.09
Atliq_Double_Bedsheet_set	Home Care	79.13
Atliq_Body_Milk_Nourishing_Lotion (120ML)	Personal Care	-75.46

Additional Question

PROMOTIONAL SALES ANALYSIS

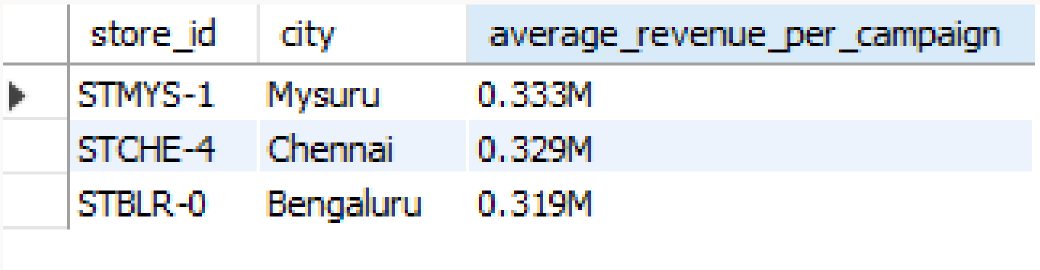
MYSQL PROJECT

Q6. Identify the top 3 stores with the highest average revenue per campaign. This report should include fields such as store_name, city, average_revenue_per_campaign, helping to identify the most profitable stores during campaigns.

- Input :

```
select ds.store_id, ds.city, concat(round(avg(base_price * `quantity_sold(after_promo)`)/1000000,3),'M')
as average_revenue_per_campaign
from dim_stores as ds
join fact_events as fe
on ds.store_id = fe.store_id
group by ds.store_id, ds.city
order by average_revenue_per_campaign desc
limit 3;
```

- Output :



store_id	city	average_revenue_per_campaign
STMYS-1	Mysuru	0.333M
STCHE-4	Chennai	0.329M
STBLR-0	Bengaluru	0.319M

Additional Question

PROMOTIONAL SALES ANALYSIS

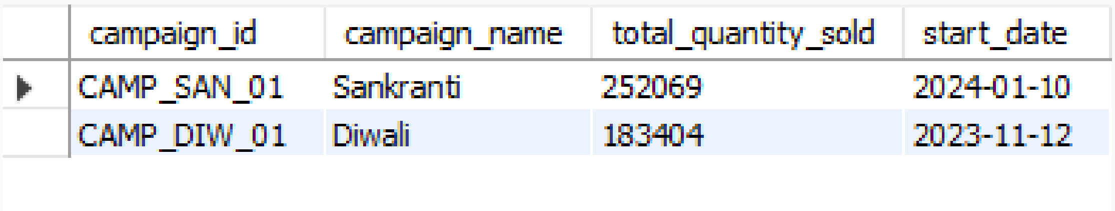
MYSQL PROJECT

Q7. Determine the campaign with the highest total quantity sold across all stores. This report will include campaign_name, total_quantity_sold, and campaign_start_date, providing insights into the most successful campaigns in terms of sales volume.

- Input :

```
select dc.campaign_id, dc.campaign_name, sum(`quantity_sold(after_promo)`)
as total_quantity_sold, dc.start_date
from dim_campaigns as dc
join fact_events as fe
on dc.campaign_id = fe.campaign_id
group by dc.campaign_id
order by total_quantity_sold desc;
```

- Output :



	campaign_id	campaign_name	total_quantity_sold	start_date
▶	CAMP_SAN_01	Sankranti	252069	2024-01-10
	CAMP_DIW_01	Diwali	183404	2023-11-12

Additional Question

PROMOTIONAL SALES ANALYSIS

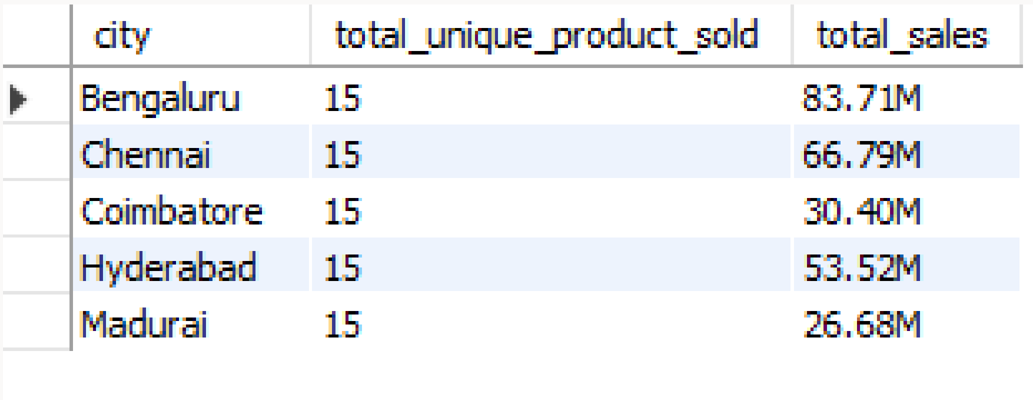
MYSQL PROJECT

Q8. List the top 5 cities by total number of unique products sold.
This report will include city, total_unique_products_sold, and total_sales, which will help identify cities with the most diverse product demand.

- Input :

```
select ds.city ,count(distinct fe.product_code) as total_unique_product_sold,
concat(round(sum(`quantity_sold(after_promo)` * base_price)/1000000,2),'M') as total_sales
from dim_stores as ds
join fact_events as fe
on ds.store_id = fe.store_id
join dim_products as dp
on fe.product_code = dp.product_code
group by ds.city
order by total_unique_product_sold desc
limit 5;
```

- Output :



city	total_unique_product_sold	total_sales
Bengaluru	15	83.71M
Chennai	15	66.79M
Coimbatore	15	30.40M
Hyderabad	15	53.52M
Madurai	15	26.68M

Additional Question

PROMOTIONAL SALES ANALYSIS

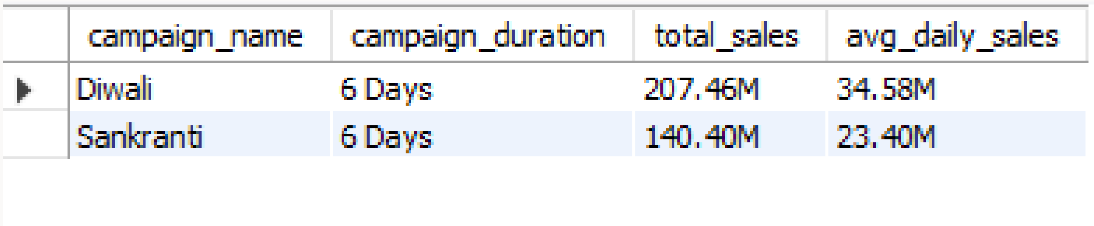
MYSQL PROJECT

Q9. Generate a report on the duration of campaigns and their total sales. This report will include campaign_name, campaign_duration_days, total_sales, and average_daily_sales, providing insights into the effectiveness of different campaign lengths.

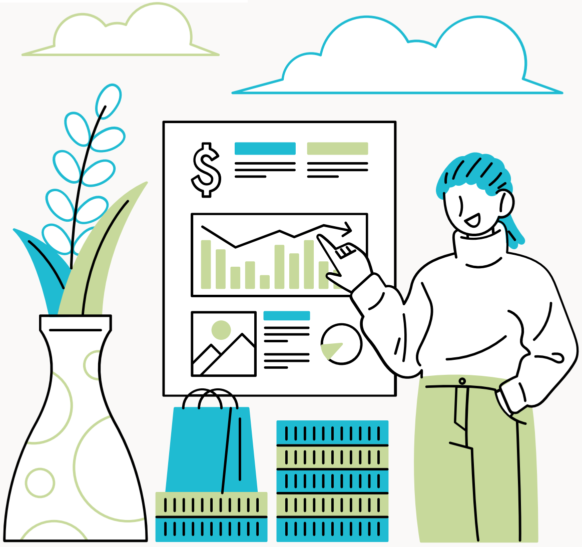
- Input :

```
select dc.campaign_name, concat(datediff(dc.end_date, dc.start_date),' Days') as campaign_duration,
concat(round(sum(`quantity_sold(after_promo)` * base_price)/1000000,2),'M') as total_sales,
concat(round(round(sum(`quantity_sold(after_promo)` * base_price) /
datediff(dc.end_date, dc.start_date),2)/1000000,2),'M') as avg_daily_sales
from dim_campaigns as dc
join fact_events as fe
on dc.campaign_id = fe.campaign_id
group by dc.campaign_name, dc.start_date, dc.end_date
order by total_sales desc;
```

- Output :



	campaign_name	campaign_duration	total_sales	avg_daily_sales
▶	Diwali	6 Days	207.46M	34.58M
	Sankranti	6 Days	140.40M	23.40M



PROMOTIONAL SALES ANALYSIS

MYSQL PROJECT

Project Resource

- Challenge Details : https://shorturl.at/sEIUV
- Dataset Link : https://bit.ly/3wYyp88
- GitHub Project Link : https://github.com/karna-chaskar/SQL-Projects/

Profile Links

- LinkedIn Profile : https://www.linkedin.com/in/karan-chaskar/
- GitHub Profile : https://github.com/karna-chaskar
- Mail Id : karanchaskar1009@gmail.com
- Contact : **+91 8623890809**
- Resume : https://drive.google.com/file/d/

THANK YOU