MY SQL

## Provide a list of products with base price greater than 500 and that are featured in promo type of 'BOGOF' (Buy one get one free).

SELECT \* FROM dim\_products;

SELECT \* FROM fact\_events;

SELECT

p.product\_code,

product\_name,

base\_price

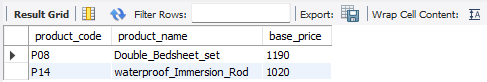
FROM dim\_products AS p

JOIN fact\_events AS e USING(product\_code)

WHERE promo\_type = "BOGOF" AND base\_price > 500

GROUP BY p.product\_code,product\_name

order by base\_price desc;



## Generate a report that provides an overview of the number of stores in each city. The results should be sorted in descending order of store count, allowing us to identify the cities with the highest store presence.

SELECT \* FROM dim\_stores;

SELECT

city,

count(\*) AS store\_count

FROM dim\_stores

GROUP BY city

ORDER BY store\_count DESC;

A screenshot of a computer

Description automatically generated

## Generate a report that displays each campaign along with the total revenue generated before or after campaign?

SELECT \* FROM dim\_campaigns;

SELECT \* FROM fact\_events;

SELECT

campaign\_name,

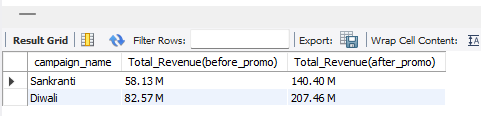
CONCAT(ROUND(SUM(base\_price\*`quantity\_sold(before\_promo)`)/1000000,2),' ',"M") AS "Total\_Revenue(before\_promo)",

CONCAT(ROUND(SUM(base\_price\*`quantity\_sold(after\_promo)`)/1000000,2),' ',"M") AS "Total\_Revenue(after\_promo)"

FROM dim\_campaigns AS c

JOIN fact\_events AS e USING(campaign\_id)

GROUP BY campaign\_name;



## Produce a report that calculates the incremental sold Quanitity(ISU%) for each category for Diwali Campaign. Additionally, provide randkings for the category based on ISU%.

SELECT \* FROM dim\_campaigns;

SELECT \* FROM dim\_products;

SELECT \* FROM fact\_events;

WITH

Diwali\_Campaign AS

(SELECT

category,

SUM(`quantity\_sold(after\_promo)`) AS "Diwali\_Sold\_After\_Promo",

SUM(`quantity\_sold(before\_promo)`) AS "Diwali\_Sold\_Before\_Promo"

FROM dim\_campaigns AS c

JOIN fact\_events AS e USING(campaign\_id)

JOIN dim\_products AS p USING(product\_code)

WHERE campaign\_name = "Diwali"

GROUP BY category

ORDER BY Diwali\_Sold\_After\_Promo DESC),

Total\_No\_Campaign AS

(SELECT

category,

SUM(`quantity\_sold(after\_promo)`) AS "Total\_No\_Campaign\_After\_Promo",

SUM(`quantity\_sold(before\_promo)`) AS "Total\_No\_Campaign\_Before\_Promo"

FROM dim\_campaigns AS c

JOIN fact\_events AS e USING(campaign\_id)

JOIN dim\_products AS p USING(product\_code)

GROUP BY category

ORDER BY Total\_No\_Campaign\_After\_Promo DESC)

SELECT

d.category,

Diwali\_Sold\_After\_Promo,

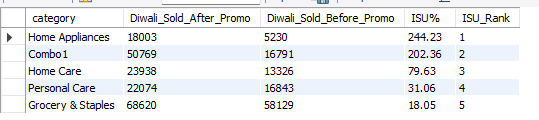
Diwali\_Sold\_Before\_Promo,

ROUND((Diwali\_Sold\_After\_Promo-Diwali\_Sold\_Before\_Promo)/Diwali\_Sold\_Before\_Promo\*100,2) AS "ISU%",

RANK() OVER(ORDER BY ROUND((Diwali\_Sold\_After\_Promo-Diwali\_Sold\_Before\_Promo)/Diwali\_Sold\_Before\_Promo\*100,2) DESC) AS ISU\_Rank

FROM Diwali\_Campaign AS d

JOIN Total\_No\_Campaign AS t USING(category);



## Create a report featuring top 5 products, ranked by incremental Reavenue percentage (IR%) across all the campaings.

SELECT \* FROM dim\_products;

SELECT \* FROM fact\_events;

SELECT

dp.product\_name,

dp.category,

ROUND(SUM(base\_price\*`quantity\_sold(after\_promo)`)/(select SUM(base\_price\*`quantity\_sold(after\_promo)`) FROM fact\_events)\*100,2) AS "IR%"

FROM dim\_products AS dp

JOIN fact\_events AS fe USING(product\_code)

GROUP BY product\_name,category

ORDER BY `IR%` DESC

LIMIT 5;

