PROMOTION BASED TANGILE INSIGHTS IN MYSQL

select \* from fact\_events;

Update fact\_events /\* updating revenuee values in fact events table on base price and quantity sold before discount \*/

set Revenuee = base\_price\*`quantity\_sold(before\_promo)`;

alter table fact\_events /\* add new discount column which holds a discount price for promotype\*/

add column discount int;

update fact\_events

set discount = base\_price - 500 where promo\_type = "500 Cashback" ;

update fact\_events

set discount = base\_price \* 0.33 where promo\_type = "33% OFF" ;

update fact\_events

set discount = base\_price \* 0.50 where promo\_type = "50% OFF" ;

alter table fact\_events /\*Adding new column revenue ADP,which multiply the discount prices and quantity sold after promo\*/

add column Revenue\_ADP int;

update fact\_events /\*calculating revenue after promo by discount prices \*/

set Revenue\_ADP = discount \* `quantity\_sold(after\_promo)`;

select sum(Revenue\_ADP),sum(Revenue\_BDP) from fact\_events; /\*display revenue ADP ,BDP\*/

alter table fact\_events /\*Adding new column IR, which doing Revenue\_ADP - BDP\*/

add column IR int;

update fact\_events

set IR = Revenue\_ADP - Revenue\_BDP ;

Select\* from fact\_events;

alter table fact\_events /\*Adding new column IR, which doing `quantity\_sold(after\_promo)` - `quantity\_sold(before\_promo)`\*/

add column ISU int;

Update fact\_events

Set ISU =`quantity\_sold(after\_promo)` - `quantity\_sold(before\_promo)` ;

# Fact\_events(Table) Screenshot 2024-02-05 220614

1. List of products which has a base price >+500 and promotype as BOGOF Which helps to identify the high value product that are heavily discounted\*/

**OUTPUT:**

SELECT product\_code, product\_name

FROM dim\_products

WHERE product\_code IN (

SELECT product\_code

FROM fact\_events

WHERE base\_price >= 500 AND promo\_type = 'BOGOF' );

|  |  |
| --- | --- |
| product\_code | product\_name |
| P08 | Atliq\_Double\_Bedsheet\_set |
| P14 | Atliq\_waterproof\_Immersion\_Rod |

1. In this syntax , I retrieved a number of stores are located in every city.

**OUTPUT:**

select city , count(store\_id) as" Number\_of\_stores" /\*2. display a total number of stores in each city in descorder\*/

from dim\_stores

group by city

order by number\_of\_stores desc ;

|  |  |
| --- | --- |
| city | Number\_of\_stores |
| Bengaluru | 10 |
| Chennai | 8 |
| Hyderabad | 7 |
| Coimbatore | 5 |
| Visakhapatnam | 5 |
| Madurai | 4 |
| Mysuru | 4 |
| Mangalore | 3 |
| Trivandrum | 2 |
| Vijayawada | 2 |

1. Returns a revenue before and after promotion in the field of campaign name.

Syntax:

select dc.campaign\_name,

SUM(fe.Revenuee)/1000000 AS Revenue\_BDP\_Million,

SUM(fe.Revenue\_ADP)/1000000 AS Revenue\_ADP\_million

FROM

dim\_campaigns dc

JOIN

fact\_events fe ON dc.campaign\_id = fe.campaign\_id

GROUP BY

dc.campaign\_name;

|  |  |  |
| --- | --- | --- |
| campaign\_name | Revenue\_BDP\_Million | Revenue\_ADP\_million |
| Sankranti | 58.1274 | 77.5425 |
| Diwali | 82.5738 | 148.7262 |

**OUTPUT:**

1. Returns a revenue before and after promotion in the field of campaign name.

**SYNTAX:**

SELECT

dc.category,

SUM(fe.IR)/1000000 AS Incremental\_Revenue,

RANK()OVER (ORDER BY sum(fe.IR) DESC)as ranks

FROM

dim\_products dc

JOIN

fact\_events fe ON dc.product\_code = fe.product\_code

where fe.campaign\_id ="CAMP\_DIW\_01"

GROUP BY

dc.category;

|  |  |  |
| --- | --- | --- |
| category | Incremental\_Revenue | ranks |
| Combo1 | 76.5495 | 1 |
| Home Appliances | 2.2982 | 2 |
| Home Care | 0.9236 | 3 |
| Personal Care | -0.5626 | 4 |
| Grocery & Staples | -13.0563 | 5 |

**OUTPUT:**

1. Returns a revenue before and after promotion in the field of campaign name.

**SYNTAX:**

SELECT

dc.product\_name,dc.category,

SUM(fe.IR)/1000000 AS Incremental\_Revenues

FROM

dim\_products dc

JOIN

fact\_events fe ON fe.product\_code = dc.product\_code

GROUP BY

dc.product\_code;

**OUTPUT:**

|  |  |  |
| --- | --- | --- |
| product\_name | category | Incremental\_Revenues |
| Atliq\_Doodh\_Kesar\_Body\_Lotion (200ML) | Personal Care | -0.3317 |
| Atliq\_Suflower\_Oil (1L) | Grocery & Staples | 1.0374 |
| Atliq\_Curtains | Home Care | 1.07 |
| Atliq\_Home\_Essential\_8\_Product\_Combo | Combo1 | 91.053 |
| Atliq\_Scrub\_Sponge\_For\_Dishwash | Home Care | -0.2471 |
| Atliq\_Cream\_Beauty\_Bathing\_Soap (125GM) | Personal Care | -0.1625 |
| Atliq\_High\_Glo\_15W\_LED\_Bulb | Home Appliances | 2.3517 |
| Atliq\_Double\_Bedsheet\_set | Home Care | 3.9579 |
| Atliq\_Fusion\_Container\_Set\_of\_3 | Home Care | -1.6958 |
| Atliq\_Lime\_Cool\_Bathing\_Bar (125GM) | Personal Care | -0.1598 |
| Atliq\_Farm\_Chakki\_Atta (1KG) | Grocery & Staples | 2.8193 |
| Atliq\_Sonamasuri\_Rice (10KG) | Grocery & Staples | -16.9429 |
| Atliq\_Masoor\_Dal (1KG) | Grocery & Staples | -2.3504 |
| Atliq\_waterproof\_Immersion\_Rod | Home Appliances | 5.482 |
| Atliq\_Body\_Milk\_Nourishing\_Lotion (120ML) | Personal Care | -0.3134 |

**Note:**

In this above SQL query I performed some basic operations like adding a new column, calculating new measures for in-depth analysis.