**A case study on ZOMATO (DUMMY dataset) which includes tables like :**

1. **delivery\_partner**
2. **food**
3. **menu**
4. **order\_details**
5. **orders**
6. **restaurants**
7. **users**

**The Case study is done under the guidance of CampusX YouTuber.**

**-- Q1) select the database**

USE Zomato;

**-- Q2) count the total number of orders**

SELECT count(\*)

FROM orders;

**-- Q3) return n random records from the orders table (i.e 5 random samples)**

SELECT \* FROM orders

ORDER BY rand()

limit 5;

**-- Q4) find null values from the orders table (here null means empty)**

SELECT \*

FROM orders

WHERE restaurant\_rating ='';

**-- Q5) Replace empty fields with Null values**

-- UPDATE orders

SET restaurant\_rating ='NULL'

WHERE restaurant\_rating ='';

**-- Q6) Find the number of orders placed by each customer**

SELECT t2.name, count(\*) as order\_count

FROM orders t1

JOIN users t2

ON t1.user\_id = t2.user\_id

GROUP BY t2.user\_id,t2.name;

**-- Q7) Find the restaurants with highest number of menu**

SELECT r\_name,count(\*) as menu\_count

FROM menu t1

JOIN restaurants t2

ON t1.r\_id = t2.r\_id

GROUP by t2.r\_id,t2.r\_name

ORDER BY menu\_count DESC

LIMIT 1;

**-- Q8) Find the number of votes and average rating for all restaurants**

SELECT t2.r\_name, count(\*) AS num\_votes,ROUND(avg(restaurant\_rating),2) AS rating

FROM orders t1

JOIN restaurants t2

ON t1.r\_id = t2.r\_id

WHERE restaurant\_rating IS NOT NULL

GROUP BY t1.r\_id,t2.r\_name;

**-- Q9)Which food item is sold by the most of the restaurants**

SELECT t2.f\_name, COUNT(\*) AS cnt FROM menu t1

JOIN food t2

ON t1.f\_id=t2.f\_id

GROUP BY t1.f\_id,t2.f\_name

ORDER BY cnt DESC LIMIT 1;

**-- Q10) Which restaurant had maximum revenue in MAY?**

SELECT r\_name,SUM(amount) as revenue FROM orders t1

JOIN restaurants t2

ON t1.r\_id=t2.r\_id

WHERE MONTHNAME(DATE(date)) = "May"

GROUP BY t1.r\_id,t2.r\_name

ORDER BY revenue DESC LIMIT 1;

**-- Q11) FInd month wise revenue of a restaurants**

SELECT MONTHNAME(DATE(date)),SUM(amount) as revenue FROM orders t1

JOIN restaurants t2

ON t1.r\_id=t2.r\_id

WHERE r\_name= "box8"

GROUP BY MONTHNAME(DATE(date))

ORDER BY MONTH(DATE(date));

**-- Q12) Find restaurant of revenue more that 1500**

SELECT r\_name, SUM(amount) as revenue FROM orders t1

JOIN restaurants t2

ON t1.r\_id=t2.r\_id

GROUP BY t1.r\_id,t2.r\_name

HAVING revenue > 1500;

**-- Q13) Find Customer who never ordered**

SELECT user\_id,name FROM users

EXCEPT

SELECT t1.user\_id,name FROM orders t1

JOIN users t2

ON t1.user\_id = t2.user\_id;

***[“I see, the error you're encountering is due to the incorrect usage of the EXCEPT operator. In MySQL, the EXCEPT operator is not supported. Instead, you can achieve the desired result using a LEFT JOIN with a WHERE clause to filter out the rows where the join finds a match.”]***

SELECT t1.user\_id, name

FROM users AS t1

LEFT JOIN orders AS t2 ON t1.user\_id = t2.user\_id

WHERE t2.user\_id IS NULL;

**-- Q14) Show order details of a particular customer in a given date range**

**-- for eg. user\_id = 1 , from '2022-05-15' to '2022-06-15'**

SELECT t1.order\_id,user\_id,date,f\_name FROM orders t1

JOIN order\_details t2

ON t1.order\_id =t2.order\_id

LEFT JOIN food t3

ON t2.f\_id=t3.f\_id

WHERE user\_id = 1 AND date BETWEEN '2022-05-15' AND '2022-06-15'

ORDER BY date;

**-- Q15) Customer Favorite food (food ordered max no. of times)**

[Not Solved]

**-- Q16) Find most costly restaurants (Avg price/dish)** (by CampusX)

SELECT t2.r\_name,SUM(price)/COUNT(\*) AS AVg\_price FROM menu t1

JOIN restaurants t2

ON t1.r\_id=t2.r\_id

GROUP BY t1.r\_id

ORDER BY AVg\_price DESC LIMIT 1;

# ERROR : Code: 1055. Expression #1 of SELECT list is not in GROUP BY clause and contains nonaggregated column 'zomato.t2.r\_name' which is not functionally dependent on columns in GROUP BY clause; this is incompatible with sql\_mode=only\_full\_group\_by 0.000 sec

[The error you're encountering is due to MySQL's **ONLY\_FULL\_GROUP\_BY** mode, which enforces that all columns in the **SELECT** clause must either be aggregated (using functions like SUM(), AVG(), etc.) or included in the **GROUP BY** clause.

In your query, **t2.r\_name** is not included in the **GROUP BY** clause, nor is it aggregated. To fix this error, you can include **t2.r\_name** in the **GROUP BY** clause, like so:

In this modified query, **t2.r\_name** is included in the **GROUP BY** clause along with **t1.r\_id**, ensuring that it conforms to MySQL's **ONLY\_FULL\_GROUP\_BY** mode.] The same error was encountered in some of the above questions also.

# Correct

SELECT t2.r\_name,SUM(price)/COUNT(\*) AS Avg\_price FROM menu t1

JOIN restaurants t2

ON t1.r\_id=t2.r\_id

GROUP BY t1.r\_id, t2.r\_name

ORDER BY Avg\_price DESC LIMIT 1;

**-- Q17) Find delivery partner compensation using the formula (**#deliveries\*100 + 1000\*avg\_rating**)**

SELECT partner\_name,COUNT(\*)\*100 + AVG(delivery\_rating)\*1000 AS salaries

FROM orders t1

JOIN delivery\_partner t2

ON t1.partner\_id=t2.partner\_id

GROUP BY t1.partner\_id,t2.partner\_name

ORDER BY salaries DESC;

**-- Q18) Find all the Veg restaurants**

select r\_id, r\_name from restaurants

WHERE r\_id NOT IN

(SELECT DISTINCT(t1.r\_id) FROM menu t1

JOIN food t2

ON t1.f\_id=t2.f\_id

WHERE t2.type ='Non-veg')

**-- Q19) Find min and max order value foe all the customers**

SELECT name, MIN(amount), MAX(amount),AVG(amount) FROM orders t1

JOIN users t2

ON t1.user\_id=t2.user\_id

GROUP BY t1.user\_id,t2.name;