

Zomato SQL Project Analysis



By :Chahat Gulat

INTRODUCTION

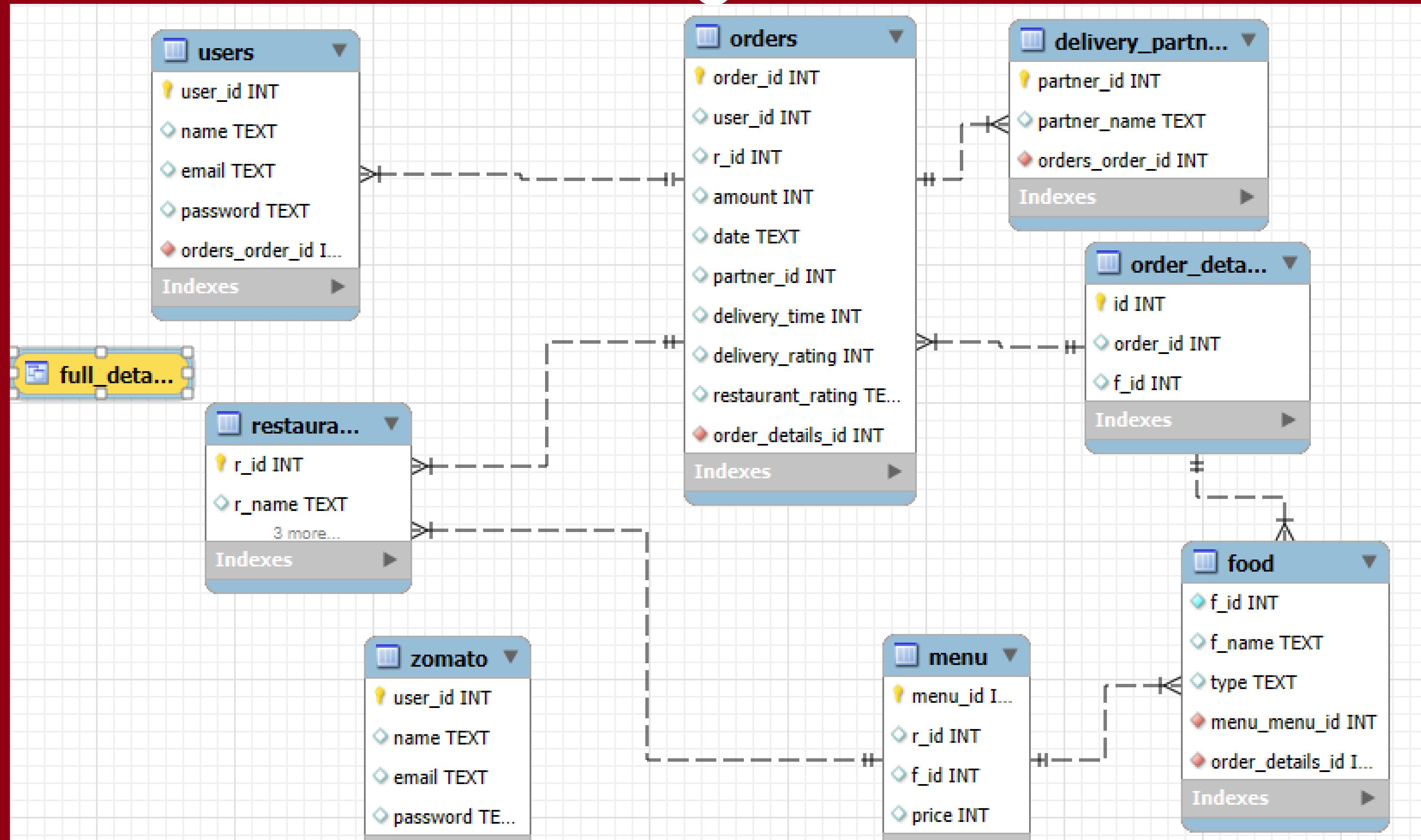
Zomato is one of India's leading food delivery platforms, connecting users with restaurants and delivery partners.

This project analyzes Zomato's dataset using SQL to uncover insights on customer behavior, restaurant performance, popular food trends, and delivery partner efficiency, with the goal of driving better business decisions and revenue growth.

OBJECTIVE

- Analyze Zomato dataset using SQL queries
- Identify customer behavior, top restaurants, and food trends
- Evaluate delivery partner performance
- Provide business insights and actionable recommendations

ER Diagram



SQL Queries

List users who placed an order

```
select distinct users.user_id,  
users.name  
from users  
join orders on users.user_id =orders.user_id;
```

Result Grid

	user_id	name
	1	Nitish
	2	Khushboo
	3	Vartika
	4	Ankit
	5	Neha

Customer with max spend

```
select users.user_id, users.name,  
sum(orders.amount) as total_spent  
from users join orders on  
users.user_id = orders.user_id  
group by users.user_id, users.name  
order by total_spent desc ;
```

Result Grid | Filter Rows:

	user_id	name	total_spent
5	Neha	3035	
2	Khushboo	2670	
4	Ankit	1800	
1	Nitish	1665	
3	Wartika	1320	

Highest rated restaurant

```
select restaurants.r_name,  
round (avg(orders.restaurant_rating ))  
as avg_rating  
from restaurants join orders  
on restaurants.r_id = orders.r_id  
group by restaurants.r_name  
order by avg_rating desc;
```

Result Grid | Filter Rows:

	r_name	avg_rating
▶	box8	4
	China Town	4
	Dosa Plaza	2
	dominos	1
	kfc	1

Monthly revenue trend

```
select year(date) as year ,  
month(date) as month,  
sum(amount) as monthly_revenue  
from orders group by year(date),  
month(date) order by  
year(date),month(date);
```

Result Grid | Filter Rows:

	year	month	monthly_revenue
▶	2022	5	2425
	2022	6	3220
	2022	7	4845

Top 5 customers by orders

```
select
count(users.user_id) as total_orders,
users.name
from users
join orders on users.user_id =orders.user_id
join order_details on order_details.order_id =
orders.order_id
group by users.name
order by total_orders desc;
```

Result Grid | Filter Row

	total_orders	name
	13	Neha
	12	Khushboo
	10	Nitish
	10	Ankit
	5	Wartika

Top revenue restaurant

```
select
restaurants.r_name,
sum(orders.amount) as total_revenue
from restaurants
join orders
on restaurants.r_id = orders.r_id
group by restaurants.r_name
order by total_revenue desc
limit 1;
```

Result Grid | Filter Rows:

	r_name	total_revenue
	kfc	3570

Best delivery partner rating

```
select delivery_partner.partner_name,  
round(avg(orders.delivery_rating)) as  
avg_rating  
from delivery_partner join orders on  
delivery_partner.partner_id =  
orders.partner_id  
group by delivery_partner.partner_name  
order by avg_rating desc;
```

Result Grid | Filter Rows:

	partner_name	avg_rating
Gyandeep	4	Gyandeep
Lokesh	3	
Suresh	3	
Kartik	3	
Amit	3	

Orders view with details

```
create view full_details as
select users.name as user_name,
restaurants.r_id ,food.f_name as food_items,
delivery_partner.partner_name as partner_name,
count(*) as total_orders
from users join orders
on users.user_id =orders.user_id
join order_details on order_details.order_id = orders.order_id
join restaurants on restaurants.r_id =orders.r_id
join food on food.f_id =order_details.f_id
join delivery_partner on delivery_partner.partner_id =orders.partner_id
group by user_name,
food_items ,partner_name;
select order_id,user_name,food_items,r_id ,partner_name from full_details;
```

order_id	user_name	food_items	r_id	partner_name
1022	Neha	Non-veg Pizza	1	Suresh
1021	Neha	Non-veg Pizza	1	Gyandeep
1011	Vartika	Non-veg Pizza	1	Amit
1006	Khushboo	Non-veg Pizza	1	Amit
1001	Nitish	Non-veg Pizza	1	Suresh
...

Min/Max order value per user

```
select users.user_id,users.name ,  
min(orders.amount),  
max(orders.amount)  
from users join orders on  
users.user_id = orders.user_id  
group by users.user_id,users.name;
```

user_id	name	min(orders.amount)	max(orders.amount)
1	Nitish	220	550
2	Khushboo	240	950
3	Vartika	180	450
4	Ankit	300	400
5	Neha	550	645

Restaurant with most menu items

```
SELECT restaurants.r_name,  
       count(menu.f_id) AS total_menu_items  
  from restaurants  
  join menu  
    on restaurants.r_id = menu.r_id  
 group by restaurants.r_name  
order by total_menu_items desc;
```

Result Grid | Filter Rows:

	r_name	total_menu_items
▶	dominos	3
▶	kfc	3
▶	box8	3
▶	Dosa Plaza	3
▶	China Town	3

Food sold across most restaurants

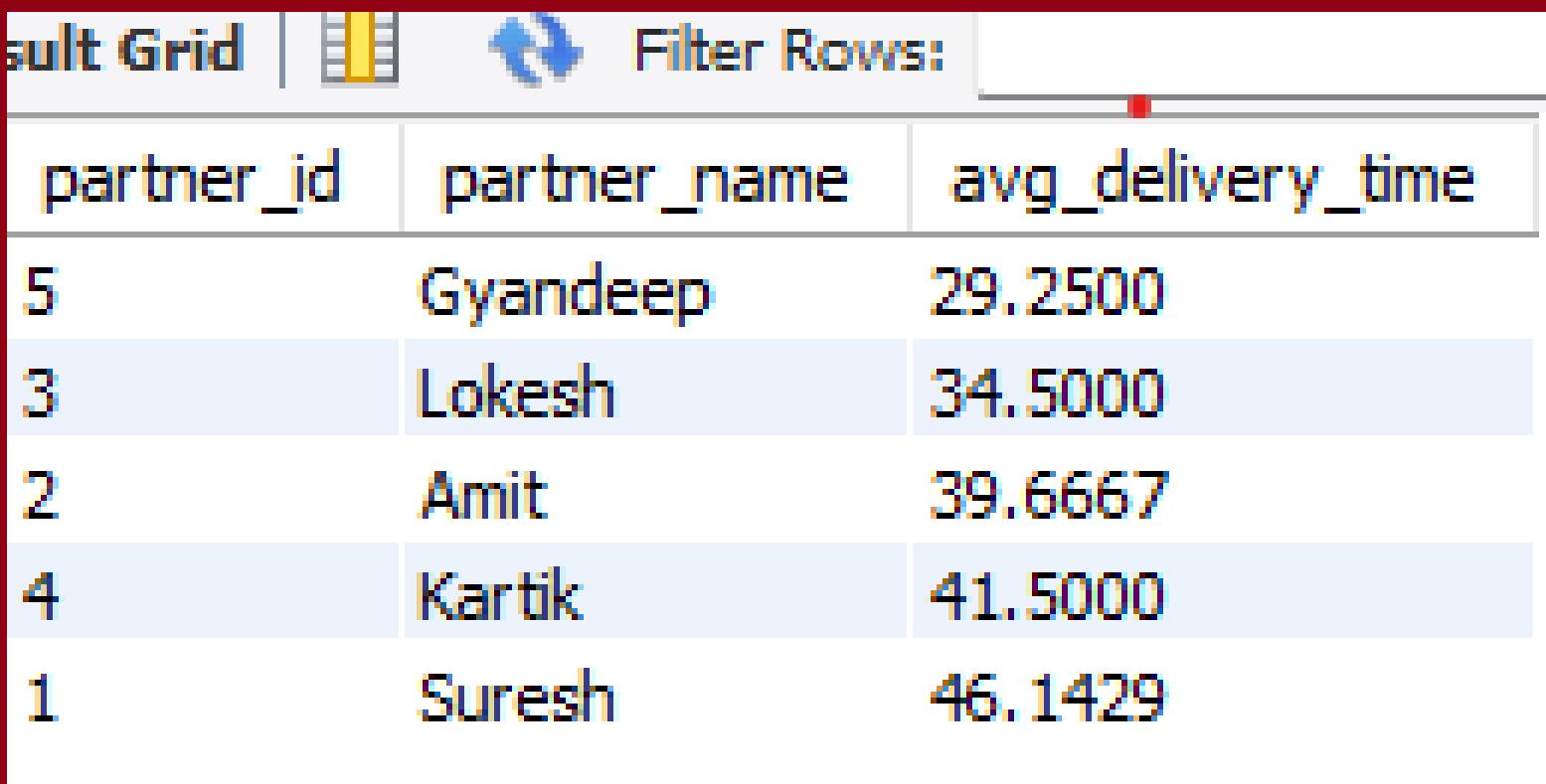
```
select food.f_name, count(*) as item_sold  
from food join order_details on  
food.f_id = order_details.f_id  
join orders on orders.order_id =  
order_details.order_id  
join restaurants on restaurants.r_id =  
orders.r_id  
group by food.f_name order by item_sold  
desc;
```

Result Grid | Filter Rows:

f_name	item_sold
Choco Lava cake	13
Chicken Wings	8
Non-veg Pizza	5
Rice Meal	4
Masala Dosa	4

Avg delivery time per partner

```
SELECT
    delivery_partner.partner_id,
    delivery_partner.partner_name,
    AVG(orders.delivery_time) AS avg_delivery_time
FROM orders JOIN
    delivery_partner
ON orders.partner_id = delivery_partner.partner_id
WHERE
    orders.delivery_time IS NOT NULL
GROUP BY
    delivery_partner.partner_id, delivery_partner.partner_name
ORDER BY
    avg_delivery_time;
```



partner_id	partner_name	avg_delivery_time
5	Gyandeep	29.2500
3	Lokesh	34.5000
2	Amit	39.6667
4	Kartik	41.5000
1	Suresh	46.1429

Monthly restaurant revenue

```
select restaurants.r_name,  
sum(orders.amount) as revenue ,  
month(orders.date) as month,  
year(orders.date ) as year  
from restaurants join orders on  
restaurants.r_id =orders.r_id  
group by restaurants.r_name ,month,  
year order by year ,month;
```

r_name	revenue	month	year
dominos	1000	5	2022
kfc	645	5	2022
Dosa Plaza	1000	5	2022
box8	480	6	2022
dominos	950	6	2022

Most costly restaurants

```
select restaurants.r_name as  
restaurant_name,  
round(avg(menu.price)) as  
avg_menu_price,  
max(menu.price) as max_menu_price  
from menu join restaurants on  
restaurants.r_id = menu.r_id  
group by restaurant_name;
```

restaurant_name	avg_menu_price	max_menu_p
dominos	317	450
kfc	215	300
box8	127	160
Dosa Plaza	177	230
China Town	217	250

Delivery partner compensation

```
delivery_partner.partner_name,  
count(orders.order_id) as total_deliveries,  
round(avg(orders.delivery_rating)) as  
avg_rating,  
(count(orders.order_id) * 100 + 1000 *  
round(avg(orders.delivery_rating))) as  
compensation  
from delivery_partner  
join orders  
on delivery_partner.partner_id =  
orders.partner_id  
group by delivery_partner.partner_name  
order by compensation desc;
```

partner_name	total_deliveries	avg_rating	compensation
Gyandeep	4	4	4400
Lokesh	4	4	4400
Suresh	7	3	3700
Amit	6	3	3600
Kartik	4	3	3400

Key Insights

- kfc generated the highest revenue (3570), followed by dominos (3050)
- Choco Lava Cake was the most popular item (13 orders)
- Neha spent the most overall (3035), followed by Khushboo (2670)
- Revenue increased steadily from May to July 2022
- dominos had the highest average menu price (317)
- Gyandep & Lokesh were top delivery partners (compensation 4400 each)

Recommendations

- Promote top-selling items like Choco Lava Cake through bundles
- Partner strategically with kfc & dominos for marketing campaigns
- Launch loyalty rewards for top spenders (Neha, Khushboo)
- Replicate July's promotional strategies to sustain revenue growth
- Train delivery partners like Suresh,Amit and Kartik to improve ratings
- Create price-segmented offerings (affordable vs premium items)

Conclusion

SQL-driven analysis of Zomato data revealed:

- Clear leaders in revenue, food trends, and delivery performance
- Strong opportunities to boost customer loyalty and restaurant partnerships
- Improvements needed in delivery quality for long-term growth

THANKYOU