Zomato SQL Practice Questions (Answer Key)

1. **Select a particular database.**

```
-- USE zomato
```

2. **Count the number of rows in a table.**

```
-- SELECT COUNT(*) FROM order_details
```

3. **Return n random records from a table.**

```
-- replicated sample function from pandas
-- SELECT * FROM users ORDER BY rand() LIMIT 5
```

4. **Find null values in a column.**

```
-- SELECT * FROM orders WHERE restaurant_rating IS NULL

-- To replace NULL values with 0

-- UPDATE orders SET restaurant_rating = 0
-- WHERE restaurant_rating IS NULL
```

5. **Find the number of orders placed by each customer.**

```
-- SELECT t2.name,COUNT(*) AS '#orders' FROM orders t1
-- JOIN users t2
-- ON t1.user_id = t2.user_id
-- GROUP BY t2.user_id
```

6. **Find the restaurant with the most number of menu items.**

```
-- SELECT r_name,COUNT(*) AS 'menu_items' FROM restaurants t1
-- JOIN menu t2
-- ON t1.r_id = t2.r_id
-- GROUP BY t2.r_id
```

7. **Find the number of votes and average rating for all restaurants.**

```
SELECT 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;
```

8. **Find the food item being sold at the most number of restaurants.**

```
SELECT f_name,COUNT(*) FROM menu t1
JOIN food t2
ON t1.f_id = t2.f_id
GROUP BY t1.f_id
ORDER BY COUNT(*) DESC LIMIT 1;
```

9. **Find the restaurant with the maximum revenue in a given month. (eg: may)**

May

```
-- SELECT MONTHNAME(DATE(date)),date FROM orders
SELECT r_name,SUM(amount) AS 'revenue' FROM orders t1
JOIN restaurants t2
ON t1.r_id = t2.r_id
WHERE MONTHNAME(DATE(date)) = 'July'
GROUP BY t1.r_id
ORDER BY revenue DESC LIMIT 1;

-- month by month revenue for a particular restaurant = box8
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));
```

10. **Find restaurants with sales greater than x. (eg: 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
HAVING revenue > 1500;
```

11. **Find customers who have never ordered.**

```
SELECT user_id,name FROM users
EXCEPT
SELECT t1.user_id,name FROM orders t1;
```

OR

```
select t1.user_id,t1.name from users t1
 except
select t2.user_id,t3.name from orders t2
 JOIN users t3 on t2.user_id =t3.user_id
```

12. **Show order details of a particular customer in a given date range.**

```
SELECT t1.order_id,f_name,date FROM orders t1
JOIN order_details t2
ON t1.order_id = t2.order_id
JOIN food t3
ON t2.f_id = t3.f_id
WHERE user_id = 5 AND date BETWEEN '2022-05-15' AND '2022-07-15';
```

13. **Find customers' favorite food.**

```
SELECT t1.user_id,t3.f_id,COUNT(*) FROM users t1
JOIN orders t2
ON t1.user_id = t2.user_id
JOIN order_details t3
ON t2.order_id = t3.order_id
GROUP BY t1.user_id,t3.f_id
ORDER BY COUNT(*) DESC;
```

14. **Find the most costly restaurants (average price per dish).**

```
SELECT 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 ASC LIMIT 1;
```

15. **Find delivery partner compensation using the formula (#deliveries * 100 + 1000 * avg_rating).**

```
SELECT partner_name,COUNT(*) * 100  + AVG(delivery_rating)*1000 AS 'salary'
FROM orders t1
```

```
JOIN delivery_partner t2
ON t1.partner_id = t2.partner_id
GROUP BY t1.partner_id
ORDER BY salary DESC;
```

16. Find revenue per month for a restaurant.

```
-- month by month revenue for a particular restaurant = kfc
SELECT MONTHNAME(DATE(date)),SUM(amount) AS 'revenue' FROM orders t1
JOIN restaurants t2
ON t1.r_id = t2.r_id
WHERE r_name = 'kfc'
GROUP BY MONTHNAME(DATE(date))
ORDER BY MONTH(DATE(date));
```

17. Find all vegetarian restaurants.

```
SELECT r_name FROM menu t1
JOIN food t2
ON t1.f_id = t2.f_id
JOIN restaurants t3
ON t1.r_id = t3.r_id
GROUP BY t1.r_id
HAVING MIN(type) = 'Veg' AND MAX(type) = 'Veg';
```

18. Find the minimum and maximum order value for all 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
```