-- USE zomato

-- SELECT COUNT(\*) FROM order\_details

-- replicated sample function from pandas

-- SELECT \* FROM users ORDER BY rand() LIMIT 5

-- To find the NULL values

-- 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

-- Q5

-- SELECT t2.name,COUNT(\*) AS '#orders' FROM orders t1

-- JOIN users t2

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

-- GROUP BY t2.user\_id

-- Q6

-- 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

-- Q7

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;

-- Q8

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;

-- Q9 -> 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 restautant = 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 = 'box8'

GROUP BY MONTHNAME(DATE(date))

ORDER BY MONTH(DATE(date));

-- Q10

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;

-- Q11

SELECT user\_id,name FROM users

EXCEPT

SELECT t1.user\_id,name FROM orders t1;

-- Q12

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';

-- Q13

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;

-- Q14

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;

-- Q15

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;

-- Q17

-- SELECT CORR(delivery\_time,delivery\_rating) AS 'corr'

-- FROM orders;

-- Q19

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';

-- Q 20

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