MySQL runner: https://extendsclass.com/mysql-online.html#

1. **"Colors" table is as follows:**

ID	C1	C2	C3
1	Red	Yellow	Blue
2	NULL	Red	Green
3	Yellow	NULL	Violet

Print the rows which have 'Yellow' in one of the columns C1, C2, or C3, but without using OR.
Output:

ID	C1	C2	C3
1	Red	Yellow	Blue
3	Yellow		Violet

2. **Write a SQL query to find the 10th highest employee salary from an Employee table. Assume that there are at least 10 records in the Employee table.**

salary
10000
20000
25000
30000
40000
50000
22000
50000
70000
76000
77000
11000

Output:

salary
20000

3. **Imagine a single column in a table that is populated with either a single digit (0-9) or a single character (a-z, A-Z). Write a SQL query to print 'Fizz' for a numeric value or 'Buzz' for alphabetical value for all values in that column.**

Example: ['d', 'x', 'T', 8, 'a', 9, 6, 2, 'V']

Output: ['Buzz', 'Buzz', 'Buzz', 'Fizz', 'Buzz','Fizz', 'Fizz', 'Fizz', 'Buzz']

4. **Given two tables created as follows**

create table test_a(id numeric);
create table test_b(id numeric);
insert into test_a(id) values
(10),
(20),
(30),
(40),
(50);

insert into test_b(id) values
(10),
(30),
(50);

Write a query to fetch values in table test_a that are and not in test_b without using the NOT keyword.

Output:

id
20
40

5. **Given the table mass_table:**

weight
5.67
34.567
365.253
34

Write a query that produces the output:

weight	kg	gms
5.67	5	67
34.567	34	567
365.253	365	253
34	34	0

6. **Consider the Employee table below.**

Emp_Id	Emp_name	Salary	Manager_Id
10	Anil	50000	18
11	Vikas	75000	16
12	Nisha	40000	18
13	Nidhi	60000	17
14	Priya	80000	18
15	Mohit	45000	18
17	Raman	55000	16
18	Santosh	65000	17
16	Rajesh	90000	-

Write a query to generate below output:

Manager_Id	Manager	Average_Salary_Under_Manager
16	Rajesh	65000
17	Raman	62500
18	Santosh	53750

7. **Given the following table named A:**

2
-2
4
-4
-3
0
2

Write a single query to calculate the sum of all positive values and negative values of A.

8. **Given the following tables:**

SELECT * FROM users;

user_id	username
1	John Doe
2	Jane Don
3	Alice Jones
4	Lisa Romero

SELECT * FROM training_details;

user_training_id	user_id	training_id	training_date
1	1	1	"2015-08-02"
2	2	1	"2015-08-03"
3	3	2	"2015-08-02"
4	4	2	"2015-08-04"
5	2	2	"2015-08-03"
6	1	1	"2015-08-02"
7	3	2	"2015-08-04"
8	4	3	"2015-08-03"
9	1	4	"2015-08-03"
10	3	1	"2015-08-02"
11	4	2	"2015-08-04"
12	3	2	"2015-08-02"
13	1	1	"2015-08-02"
14	4	3	"2015-08-03"

Write a query to get the list of users who took the training lesson more than once in the same day, grouped by user and training lesson, each ordered from the most recent lesson date to oldest date.

Output:

user_id	username	training_id	training_date	count
4	Lisa Romero	2	August, 04 2015 00:00:00	2
4	Lisa Romero	3	August, 03 2015 00:00:00	2
1	John Doe	1	August, 02 2015 00:00:00	3
3	Alice Jones	2	August, 02 2015 00:00:00	2

9. **You are given the following table containing historical employee salaries for company MyCompany:**

Table: EmployeeSalaries

employee_ID	salary	year

1	80000	2020
1	70000	2019
1	60000	2018
2	65000	2020
2	65000	2019
2	60000	2018
3	65000	2019
3	60000	2018
3	70000	2020

Given the above table, can you write a SQL query to return the employees who have received at least 3 year over year raises based on the table's data

Output:

employee_ID
1
3

10. **Suppose you're given the following table that shows spend by keyword, advertiser, and unique ad ID:**

Table: keyword_summary

keyword	ad_id	advertiser_id	total_spend
bicycle	1243213	1234	95
bike	1243213	1234	71
bike tires	1243213	1234	22
bike breaks	1243213	1234	95
bike accessories	1243213	1234	28
...	...	...	...

Using the table above, write a SQL query that returns each keyword along with the advertiser_id of the top spending advertiser, and total spend on the keyword (agnostic of advertiser). Sort the results in descending order by the total spend on the keyword.

Output:

keyword	advertiser_id	spend_keyword
bike	21781	3599.040069580078
bicycle	1234	2518.320000767708
bike breaks	1234	266.1999969482422
bike accessories	3829	195.3400001525879
bike tires	3829	171.63999938964844
mountain bike	3829	59.16999816894531