

DB Challenge 2014

General Instruction:

Read each problem carefully. Each problem specifies **task** and/or **assumption/s**. Input and output tables are provided per problem. Take note of the exact table names as well as field names and their respective types.

Answers should be encoded as stored procedures with the following naming format:

problem_<problem number>

e.g.

problem_a

```
DROP PROCEDURE IF EXISTS `problem_a`$$
```

```
CREATE PROCEDURE `problem_a`()
```

```
BEGIN
```

```
END$$
```

```
DELIMITER ;
```

Problem A
Service Net Income

Automoto is a company that offers car wash and detailing services. With each service rendered, the company collects sale then deducts the expenses incurred during each service to compute for the company's net income.

Task:

Given the input tables ***problem_a_input_service_sale*** and ***problem_a_input_service_expense*** below, determine the company's net income for each service transaction. The result will be stored into the output table ***problem_a_output***.

Assumption:

A service sale can be tagged with zero or many expense rows.

Input:

problem_a_input_service_sale

Field	Type	NULL	KEY	Extra
sale_id	bigint(30)	NO	PRI	auto_increment
sale_date	date	YES		
sale_amount	decimal(12,2)	YES		

problem_a_input_service_expense

Field	Type	NULL	KEY	Extra
expense_id	bigint(30)	NO	PRI	auto_increment
expense_amount	decimal(12,2)	YES		
sale_id	bigint(30)	YES	MUL	

Output:

problem_a_output

Field	Type	NULL	KEY	Extra
sale_id	bigint(30)	YES	MUL	
sale_amount	decimal(12,2)	YES		
total_expense	decimal(12,2)	YES		
net_income	decimal(12,2)	YES		

Problem B

Common Friends

ABC University is maintaining a database system for the students' social network. The first table holds the list of student names. Friendship is represented through a row in the second table.

Task:

Find common friends between two given student names: **Lebron James** and **Tim Duncan**. Use the output table to display the result ordered by name.

Assumptions:

1. Student names are unique.
2. A friendship is represented as a single row in the second table. So, if friendship link between A and B exists, A is friends with B and vice versa. No need for another link between B and A.

Input:

problem_b_input_person

Field	Type	NULL	KEY	Extra
person_id	bigint(30)	NO	PRI	auto_increment
name	varchar(45)	YES		

problem_b_input_friendship

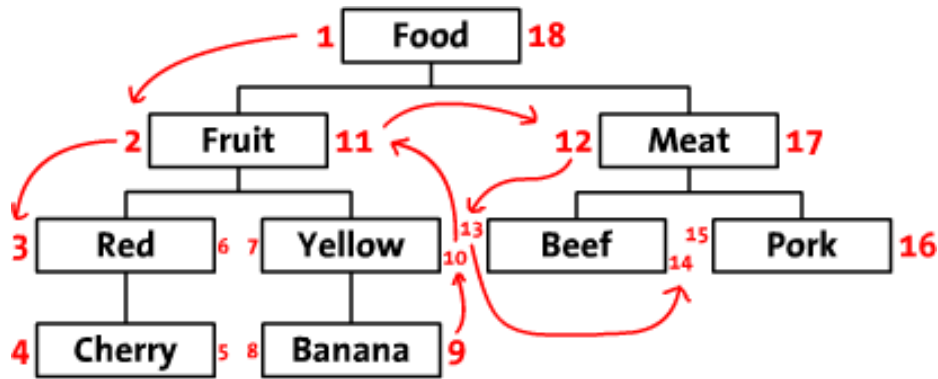
Field	Type	NULL	KEY	Extra
friendship_id	bigint(30)	NO	PRI	auto_increment
person_id	bigint(30)	YES	MUL	
friend_id	bigint(30)	YES	MUL	

Output:

problem_b_output

Field	Type	NULL	KEY	Extra
friend_name	varchar(30)	YES		

Problem C Organizational Chart



We'll start by laying out our tree in a horizontal way. Start at the root node ('Food'), and write a 1 to its left. Follow the tree to 'Fruit' and write a 2 next to it. In this way, you walk (traverse) along the edges of the tree while writing a number on the left and right side of each node. The last number is written at the right side of the 'Food' node. In this image, you can see the whole numbered tree, and a few arrows to indicate the numbering order.

Parent	Title	lft	ryt
	Food	1	18
Food	Fruit	2	11
Fruit	Red	3	6
Red	Cherry	4	5
Fruit	Yellow	7	10
Yellow	Banana	8	9
Food	Meat	12	17
Meat	Beef	13	14
Meat	Pork	15	16

We'll call these numbers left and right (e.g. the left value of 'Food' is 1, the right value is 18). As you can see, these numbers indicate the relationship between each node. Because 'Red' has the numbers 3 and 6, it is a descendant of the 1-18 'Food' node. In the same way, we can say that all nodes with left values greater than 2 and right values less than 11, are descendants of 2-11 'Fruit'. The tree structure is now stored in

the left and right values. This method of walking around the tree and counting nodes is called the '**modified preorder tree traversal**' algorithm. We now have the lft and ryt values to store the tree structure.

Hint: Right value of a parent is equals 2 x (number of child nodes plus itself).

Task:

Given the input table that holds the different employee designation, display the hierarchy of the employees using their left and right values. Display the output with indention indicating hierarchy using the table provided.

problem_c_input

Field	Type	NULL	KEY	Extra
emp_no	bigint(30)	NO	PRI	auto_increment
designation	varchar(45)	YES		
lft	int(11)	YES		
ryt	int(11)	YES		

Output:

problem_c_output

Field	Type	NULL	KEY	Extra
designation	varchar(30)	YES		

Sample:

Input:

<u>emp_no</u>	<u>designation</u>	<u>lft</u>	<u>ryt</u>
1	Manager	2	7
2	CEO	1	8
3	Clerk	3	4
4	Encoder	5	6

Output:

designation

```

CEO
  Manager
    Clerk
      Encoder
  
```

Problem D

Graduating Students

Your university would like to transfer student records. Students who will be graduating from High School will have their records copied to another table.

Task:

Given the student records, check if the student is about to graduate from high school. If graduating, add student records into the output table. Order the result by student name.

Assumption:

1. Grade 12 is the last level before going to college.
2. The student must not have a failing grade and average must not be less than 80.
3. Each student has multiple grade entries in the given grades table.

Input:

problem_d_input_student

Field	Type	NULL	KEY	Extra
stud_id	bigint(30)	NO	PRI	auto_increment
stud_name	varchar(45)	YES		
yearLevel	int(11)	YES		

problem_d_input_student_grades

Field	Type	NULL	KEY	Extra
avg_id	bigint(30)	NO	PRI	auto_increment
stud_id	bigint(30)	YES	MUL	
grade	decimal(2,0)	YES		

Output:

problem_d_output_college

Field	Type	NULL	KEY	Extra
stud_id	bigint(30)	NO	PRI	auto_increment
stud_name	varchar(45)	YES		
yearLevel	int(11)	YES		

Problem E BIOMETRIC

Mr John manages his employees' attendance using a biometric device which stores the time of in and out into a database. He would like to see the total number of hours of each employee per day. The first table contains the list of employee names. Another table holds the in and out details of each employee.

Task:

Given the attendance details, compute for the total number of hours of each employee. Store the result using the output table provided. Order the result by employee name.

Assumption:

The attendance contains no overlapping of time in and out for each employee (1 time in and 1 time out in the morning, and another time in and time out in the afternoon). The table contains a day's work attendance only.

problem_e_input_employee

Field	Type	Null	Key	Default	Extra
empName_ID	int(11)	NO	PRI		auto_increment
Employee	varchar(45)	YES			

problem_e_input_biometric

Field	Type	Null	Key	Default	Extra
problem_e_input_biometric_id	int(11)	NO	PRI		auto_increment
empName_ID	int(11)	YES	MUL		
TimeIn	'datetime'	YES			
'TimeOut	datetime	YES			

problem_e_input_biometric

Field	Type	Null	Key	Default	Extra
problem_e_output_id	int(11)	NO	PRI		auto_increment
Employee	VARCHAR(45)	YES	MUL		
DateWork	date	YES			
TotalWorkMorning	decimal(12,2)	YES			
TotalWorkAfternoon	decimal(12,2)				
TotalHours	decimal(12,2)				

Problem F

Savings Challenge

James thought of an interesting savings challenge to encourage his colleagues to save money. In this challenge, a person would decide on the starting amount for the challenge and the number of weeks that the challenge would last. Each week, the amount that a person would set aside for saving will increment by the decided starting amount.

Example:

Week	Amount
1	50
2	100
3	150

After 3 weeks the person would have P300 set aside as his savings with the starting amount of P50.

Task:

Given the number of weeks and the starting amount, get the resulting amount for each given colleague.

Input:

problem_f_input_savings

Field	Type	NULL	KEY	Extra
PK	int(10)	NO	PRI	auto_increment
Person	varchar(45)	YES		
NumberOfWeeks	int(10)	YES		
StartingAmount	decimal(12,2)	YES		

Output:

problem_f_output_savings

Field	Type	NULL	KEY	Extra
NumberOfWeeks	int(10)	YES		
StartingAmount	decimal(12,2)	YES		
Person	varchar(45)	YES		
TotalSavings	decimal(12,2)	YES		

Problem G

Number Permutation

You are given a set of numbers stored in the input table. Find the different ways to arrange these numbers. Display the output in the given table. The result should be in order similar to the given sample output.

Assumption:

1. Given numbers are single digit.
2. Each number in the given set is unique.

Hint: You may want to use temporary tables.

Input:

problem_g_input_numbers

Field	Type	NULL	KEY	Extra
num_id	int(11)	NO	PRI	Auto_increment
number_entry	int(11)	YES		

Output:

problem_g_output

Field	Type	NULL	KEY	Extra
permute_value	varchar(45)	YES		

Example:

Input:

num_id	number_entry
1	5
2	9
3	4

Output:

permute_value
4,5,9
4,9,5
5,4,9
5,9,4
9,4,5
9,5,4

Problem H

Who won?

SPEED-IT just finished conducting a quiz bee. The points were stored into database tables. The first table contains the contestant names. Another table stores the different levels with their corresponding points per question. Each row in the third table represents a correct answer for a particular contestant in a specific level.

Task:

Given the points, determine the ranking of the contestants. Store the ordered result into the given output table.

Input:

problem_h_input_contestant

Field	Type	NULL	KEY	Extra
contestant_id	int(11)	NO	PRI	Auto_increment
contestant_name	varchar(45)	YES		

problem_h_input_level

Field	Type	NULL	KEY	Extra
level_id	int(11)	NO	PRI	Auto_increment
description	varchar(45)	YES		
points_per_question	int(11)	YES		

problem_h_input_contestant_points

Field	Type	NULL	KEY	Extra
points_id	int(11)	NO	PRI	Auto_increment
contestant_id	int(11)	YES	MUL	
level_id	int(11)	YES	MUL	

Output:

problem_h_output_contest_result

Field	Type	NULL	KEY	Extra
result_id	int(11)	NO	PRI	Auto_increment
contestant	varchar(45)	YES		
total_points	int(11)	YES		