DBMS – Exercises – Stored Procedures & Triggers

1. Write a SQL procedure that takes username and password and returns 1 if they match or 0 if they do not.

Note, you can use either of the below statements to return 0 or 1.

```
select "1";
select 1;
```

user(userID, username, password, userType)

2. Create user_audit table and before a username or password is updated in user table, insert the OLD details in the user_audit including datetime when it was updated. You are required to implement a trigger that will get executed automatically before an update query is executed.
Use NOW() to insert datetime and make sure the attribute data type is datetime as well.

3. Implement a trigger that will ensure the integrity of data such that, it will throw a SIGNAL '45000' error with an appropriate message for an update query that updates the balance with a balance that is less than the current (old) balance.
savings_account(account_number, **balance**, clientID)

Example

X878712	7600	870081

```
Update savings_account set balance = 5000 where ClientID = 870081
```

Error message/Exception:
You are not allowed to update the balance of this savings account with a balance that is less than the current balance which is 7600

Time for Tricks

1. You returned from vacation with a huge suitcase full of presents. Unfortunately, you forgot the correct combination for the combination lock on the bag, and now you have to try all of them until you find the correct one. You're curious about how many possible combinations the lock has.

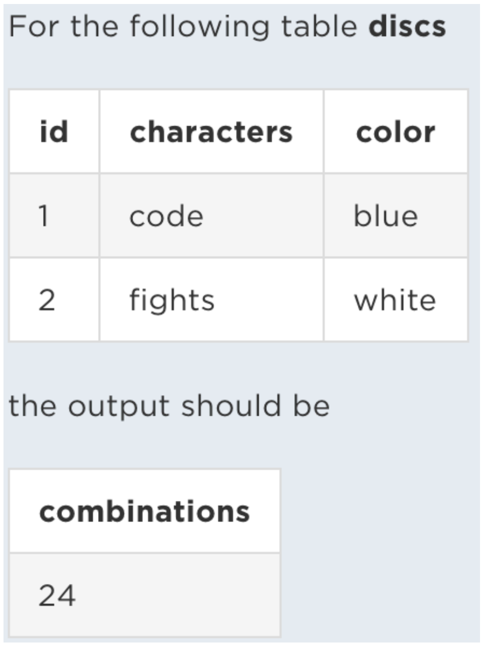
The combination lock consists of several rotating discs, where each disc has its own set of possible characters. You have a table **discs** which stores the information about these discs and has the following structure:

- id: the unique ID of a disc;
- characters: the list of characters the disc has (the characters are guaranteed to be unique);
- color: the color of the disc.

Calculate the total number of all possible combinations that the lock has, and return it as a table that has only one column combinations and one row.

Note, you can use length(attribute) to get number of characters

Example:



For the following table **discs**

id	characters	color
1	code	blue
2	fights	white

the output should be

combinations
24

The set of possible characters for the first disc is equal to 4, and the set for the second disc is 6, so the total number of combinations is 4 * 6 = 24

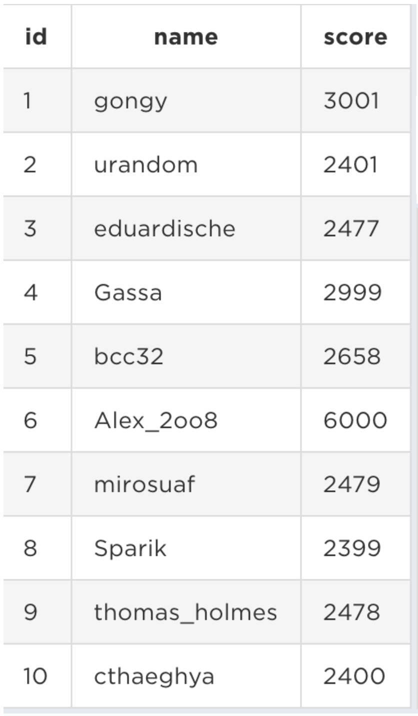
2. You are working as a recruiter at a big IT company, and you're actively looking for candidates who take the top places in major programming contests. Since the grand finale of the annual City Competition, you've been reaching out to the top participants from the leaderboard, and successfully so.

You have already interviewed all the prize winners (the top 3 participants), but that's not enough right now. Your company needs more specialists, so now you would like to connect with the participants who took the next 5 places.
The contest leaderboard is stored in a table **leaderboard** with the following columns:

- id: unique id of the participant;
- name: the name of the participant;
- score: the score the participant achieved in the competition.

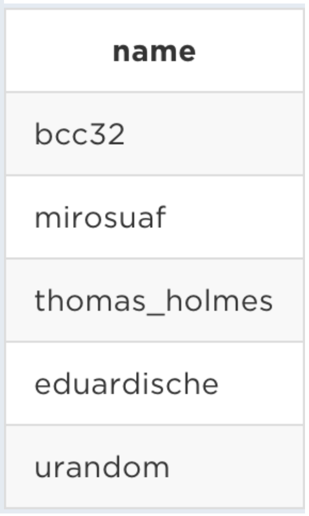
The resulting table should contain the names of the participants who took the 4th to 8thplaces inclusive, sorted in descending order of their places. If there are fewer than 8 participants, the results should contain those who ranked lower than 3rd place.
It is guaranteed that there are at least 3 prize winners in the leaderboard and that all participants have different scores.

Examples:



id	name	score
1	gongy	3001
2	urandom	2401
3	eduardische	2477
4	Gassa	2999
5	bcc32	2658
6	Alex_2oo8	6000
7	mirosuaf	2479
8	Sparik	2399
9	thomas_holmes	2478
10	cthaeghya	2400

The output should be:



name
bcc32
mirosuaf
thomas_holmes
eduardische
urandom