

Now we need to design a nickname database for a gun game. The specific requirements are as follows:

First, we need to create two tables.

The first table is called "player" and is used to store player information.

The "player" table contains the following fields:

- id: The ID that uniquely identifies the player.
- nickname: The player's nickname.
- country: The country where the player is located.
- createdAt: The creation time of the player's information.
- changeCount: The number of times the player has changed his nickname.

```
create table if not exists player
(
  id bigint primary key,
  nickname text not null,
  country varchar(20),
  createdAt date,
  changeCount INT DEFAULT 0
);
```

The second table is called "restricted_words", which is used to store sensitive vocabulary in different countries to avoid being involved in nicknames.

The "restricted_words" table contains the following fields:

- id: an auto-incrementing primary key.
- word: restricted vocabulary.
- country: corresponding country.

```
create table if not exists restricted_words
(
  id serial primary key,
  word varchar(255) not null,
  country varchar(20) not null
);
```

When a user logs into the game for the first time, he needs to set a nickname. The setting of nickname needs to comply with the following rules.

1. The length of ID needs to be 8. The first number of the id implies the country he is in, we need to check if they correspond. It should be noted that the first number of id cannot be 0.

First digit of ID	country
1	China

First digit of ID	country
2	America
3	Other

- Nickname length must be between 4 and 20 characters.
- Only letters (case insensitive), numbers, underscores (_), asterisks (*) and pound signs (#) are allowed.
- It must contain at least one uppercase letter, one lowercase letter, and at least one number.
- The nickname needs to be unique in the database and is not allowed to be consistent with others.
- It is forbidden to include sensitive words in nicknames. We have set up different sensitive word lists for different countries. For example, Chinese user nicknames cannot contain "woc".
- Nickname cannot contain ID.

After the user is initialized, if he wants to change his nickname, we need to check the following rules.

- The new nickname must comply with the naming rules mentioned above.
- We need to limit player permissions to avoid frequent modifications. Therefore, each user can only modify their nickname 3 times.
- The changed nickname cannot be the same as the old nickname to avoid abuse.

If the modification of the new nickname meets the above requirements, we will update the nickname in the player table.

There is a limit to the number and name of functions you can submit.

Name	Description
nickname_trigger	Before trigger, check nickname before inserting and updating
nickname_check	Function used by nickname_trigger

Please submit the code at the format below:

```
create or replace function nickname_check()
    returns trigger
as
$$
    declare
        -- declare variables here
begin
    -- write your code here
end
$$ language plpgsql;

create trigger nickname_trigger
    before insert or update
```

```
on player
for each row
execute procedure nickname_check();
```

Here are some examples may help you understand the rules:

```
insert into restricted_words(word,country) values('cnm','China');
insert into restricted_words(word,country) values('fuck','America');
insert into restricted_words(word,country) values('restricted','Other');

insert into player(id, nickname,country) values (11234567,'ZhangSan123','China');--
成功
insert into player(id, nickname,country) values (1111223,'ZhangSang123','China');--
id不合法
insert into player(id, nickname,country) values (11133223,'ZhangSang123aaaaa
aaaaaaa','USA');-- 过长
insert into player(id, nickname,country) values (31112333,'ZhangSan123!','Other');--
含有非法字符
insert into player(id, nickname,country) values (21113333,'ZhangSan','America');-- 无
数字
insert into player(id, nickname,country) values (11123333,'zhangsan123','China');--
无大写
insert into player(id, nickname,country) values (11123344,'ABC1234567','China');-- 无
小写
insert into player(id, nickname,country) values
(10131415,'WangfuckZi1999','China');-- 成功
insert into player(id, nickname,country) values (12323435,'WW2cnmd11111','China');--
- 包含敏感词汇
insert into player(id, nickname,country) values (22223145,'fuck2Shit','America');--
包含敏感词汇
insert into player(id, nickname,country) values
(11234567,'LiSi0fuck11234567','China');-- 包含id

insert into player(id, nickname,country) values
(27184222,'Bright*Night0','America');-- 成功
update player set nickname = 'Bright*Night0' where id = 27184222; -- 与旧昵称一致
update player set nickname = 'WangfuckZi1999' where id = 27184222; -- 重名
update player set nickname = 'BrightNight0' where id = 27184222; -- 成功
update player set nickname = 'BrightNight0123' where id = 27184222; -- 成功
update player set nickname = 'Bright0Night*' where id = 27184222; -- 成功
update player set nickname = 'Brigt0Night*' where id = 27184222; -- 更改名字超过三次
```

Some notice:

1. The oj is running on `postgresql 13.7`, please note the type conversion when you use some function, it may not work well on oj.
2. You can use `raise notice` to get the information you needed in trigger when debugging.
3. If you decide to drop the insertion, please **don't** use `raise exception`, exceptions thrown by sql may cause different return value in oj.

4. Enjoy it~