

Информационная система поддерживает сайт любителей игры «Танки». Она обладает информацией о игроках, танках, которые у них есть и типах танков.

Типичными для информационной системы являются вопросы:

- Танки каких типов есть у игрока X?
- Танков каких типов больше всего у игроков?
- У кого больше всего тяжёлых танков.

### **Расширенное задание**

Информационная система также обладает информацией о сыгранных битвах и урону который нанесли и получили танки в них.

Реализуйте запросы:

- Топ 10 танков по нанесённому урону.
- Титул «Убойная гусеница » получает пользователь, который сыграл больше 5 битв и лидирует по показателю нанесённый урон/ полученный урон.
- Титул «Бешеный» получает пользователь, который атаковал больше всего других игроков.

### **Реализовать:**

- Две функции (для строковых и числовых аргументов и/или результатов).
- Две хранимые процедуры (одну для выборки данных, одну для модификации данных).
- Два триггера (один, модифицирующий изменяемые данные, а второй запрещающий изменение при определенном условии).

Две функции (для строковых и числовых аргументов и/или результатов).

1. Функция получает на вход строку, а возвращает эту же строку, но перевёрнутую.

```
1 • drop function if exists rev;
2 • create function rev (nickname varchar(45))
3   returns varchar(45)
4   deterministic
5   return reverse(nickname);
6
7 • select rev(nickname) from players;
8
9
```

Result Grid

rev(nickname)
hsalf wolley
Sasha
822relliKretnuh
nijayiaS
niktokerByirtimD
retsam_ruoy

2. Функция получает на вход показатель урона, который нанёс игрок в определённой игре и определяет в какую из трёх групп по этому показателю он попадает.

- III - урон меньше 100000
- II - урон больше 100000, но меньше 150000
- I - урон больше 150000

```
1 • drop function if exists grade;
2 • create function grade(damage_done int) returns varchar(3)
3   deterministic
4   return(case when damage_done < 100000 then "III"
5   when damage_done between 100000 and 150000 then "II"
6   when damage_done > 150000 then "I" end);
7 • select battle_id, nickname, grade(damage_done) as grade
8   from battle_info join presence on battle_info.presence_id = presence.presence_id
9   join players on presence.player_id = players.player_id where battle_id = 1;
```

Result Grid

battle_id	nickname	grade
1	yellow Flash	I
1	ahsaS	I
1	hunterKiller228	III
1	Saiyajin	II
1	DmitriyBrekotkin	III
1	your_master	II

Две хранимые процедуры (одну для выборки данных, одну для модификации данных).

1. Процедура получает на вход число и выводит игроков, у которых показатель *damage\_done-damage\_received* больше этого числа.

```
1 delimiter //
2 • drop procedure if exists DR //
3 • create procedure DR (val int)
4 • begin select battle_id, nickname, damage_done-damage_received as 'd-r' from
5   battle_info join presence on battle_info.presence_id = presence.presence_id
6   join players on presence.player_id = players.player_id
7   where damage_done-damage_received >= val; end//
8 delimiter ;
9 • call dr(10000);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	battle_id	nickname	d-r
▶	1	yellow Flash	60146
	1	ahsaS	59000
	4	yellow Flash	10047
	6	DmitriyBrekotkin	15209

2. Процедура обновляет таблицу *players* следующим образом: считает количество букв в *nickname* и добавляет к *nickname* через нижнее подчёркивание.

До:

```
1 • select * from players;
2
```

Result Grid | Filter Rows: |

	player_id	nickname
▶	1	yellow Flash
	2	ahsaS
	3	hunterKiller228
	4	SaIyAjIn
	5	dmitriyBrekotkin
	6	your_master
*	NULL	NULL

После:

```
1 DELIMITER //
```

```
2 • DROP PROCEDURE IF EXISTS UpdateNickname //
```

```
3 • CREATE PROCEDURE UpdateNickname()
```

```
4 BEGIN
```

```
5     UPDATE players
```

```
6     SET nickname = CONCAT(nickname, '_', LENGTH(nickname))
```

```
7     WHERE nickname REGEXP '[A-Z]';
```

```
8 END//
```

```
9 DELIMITER ;
```

```
10 • call UpdateNickname();
```

```
11 • select * from players;
```

```
12
```

Result Grid

	player_id	nickname
▶	1	yellow Flash_12
	2	ahsaS_5
	3	hunterKiller228_15
	4	SaIyAjIn_8
	5	dmitriyBrekotkin_16
	6	your_master_11
*	NULL	NULL

Два триггера (один, модифицирующий изменяемые данные, а второй запрещающий изменение при определенном условии).

1. Триггер присваивает новому игроку один из танков.

```
1  delimiter //
2  • drop trigger if exists addTank //
3  • create trigger addTank
4    after insert on players
5    for each row
6  begin
7    insert into presence (player_id, tank_id)
8    value (new.player_id, 18);
9  end;
10 //
11 delimiter ;
12 • insert into players (nickname)
13   value ('dark_master');
14 • select * from presence;
```

< Result Grid Filter Rows: Edit: Export/In

	presence_id	player_id	tank_id
	60	5	18
	61	6	3
	62	6	6
	63	6	9
	64	6	12
	65	6	15
	66	6	18
	67	29	18
•	NULL	NULL	NULL

	player_id	nickname
▶	1	yellow Flash
	2	ahsaS
	3	hunterKiller228
	4	SaIyAjIn
	5	dmitriyBrekotkin
	6	your_master
	29	dark_master
•	NULL	NULL

2. Триггер запрещает обновление таблицы, если длина *nickname* больше 20.  
*nickname* по длине меньше 20.

```

3 • create trigger check_nickname
4   after update on players
5   for each row
6   begin
7     if length(new.nickname) >= 20 then signal sqlstate '45000'
8       set message_text = 'Длина ника должна быть меньше 20 символов';
9     end if;
10  end;
11  //
12  delimiter ;
13
14 • update players set nickname = '1' limit 1;
15 • select * from players;

```

player_id	nickname
1	1
2	ahsaS
3	hunterKiller228
4	SaIyAjIn
5	dmitryBrekotkin
6	your_master
29	dark_master
* HULL	HULL

*nickname* по длине больше 20.

```

14 • update players set nickname = 'aaaaaaaaaaaaaaaaaaaaa' limit 1;
15 • select * from players;

```

Output

Action Output

#	Time	Action	Message
✓ 317	22:34:59	drop trigger if exists check_nickname	0 row(s) affected
✓ 318	22:34:59	create trigger check_nickname after update on players for each row begin if length(new.nickname) >= 20 then ...	0 row(s) affected
✓ 319	22:34:59	update players set nickname = '1' limit 1	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0
✓ 320	22:34:59	select * from players LIMIT 0, 1000	7 row(s) returned
✓ 321	22:35:18	drop trigger if exists check_nickname	0 row(s) affected
✓ 322	22:35:18	create trigger check_nickname after update on players for each row begin if length(new.nickname) >= 20 then ...	0 row(s) affected
✓ 323	22:35:18	update players set nickname = 'aaaaaaaaaaaaaaaaaaaaa' limit 1	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0
✓ 324	22:35:18	select * from players LIMIT 0, 1000	7 row(s) returned
✓ 325	22:35:56	drop trigger if exists check_nickname	0 row(s) affected
✓ 326	22:35:56	create trigger check_nickname after update on players for each row begin if length(new.nickname) >= 20 then ...	0 row(s) affected
✗ 327	22:35:56	update players set nickname = 'aaaaaaaaaaaaaaaaaaaaa' limit 1	Error Code: 1644. Длина ника должна быть меньше 20 символов