

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.

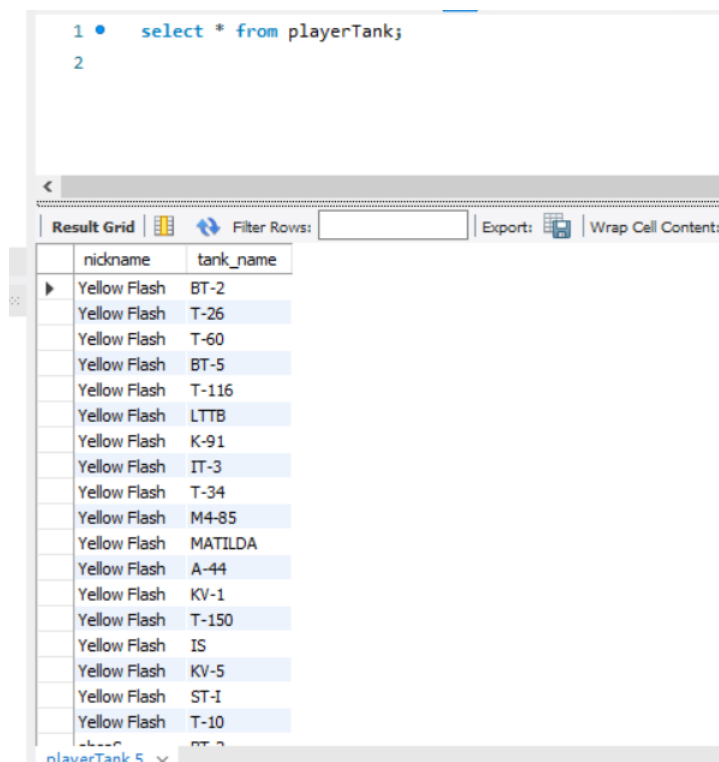
### **Задание:**

- Реализовать для своего варианта задания представление (с ограничением по строкам и столбцам)
- Для своего варианта задания сделать хранимую процедуру, использующую курсор

**Реализовать для своего варианта задания представление (с ограничением по строкам и столбцам)**

Сделаю представление на основе таблицы, которая получена на основе соединения таблиц *players*, *presence*, *tanks*, из которой возьму *nickname* и *name*. Получу таблицу игроков и названий танков, которые у него есть. Дам публичный доступ к полученному представлению с правом SELECT. Отменю все привилегии для публичного доступа к таблице *players*.

```
drop view if exists playerTank;  
create view playerTank as  
select nickname, name as tank_name from  
players join presence on players.player_id = presence.player_id  
join tanks on presence.tank_id = tanks.tank_id;  
grant select on playerTank to public;  
REVOKE ALL PRIVILEGES ON players FROM PUBLIC;
```



The screenshot shows a database client interface. At the top, a SQL query is entered: `select * from playerTank;`. Below the query editor, the results are displayed in a grid view. The grid has two columns: `nickname` and `tank_name`. The data shows 18 rows, all with the same `nickname` ('Yellow Flash') and various `tank_name` values.

nickname	tank_name
Yellow Flash	BT-2
Yellow Flash	T-26
Yellow Flash	T-60
Yellow Flash	BT-5
Yellow Flash	T-116
Yellow Flash	LTTB
Yellow Flash	K-91
Yellow Flash	IT-3
Yellow Flash	T-34
Yellow Flash	M4-85
Yellow Flash	MATILDA
Yellow Flash	A-44
Yellow Flash	KV-1
Yellow Flash	T-150
Yellow Flash	IS
Yellow Flash	KV-5
Yellow Flash	ST-1
Yellow Flash	T-10

Для своего варианта задания сделать хранимую процедуру, использующую курсор

Процедура будет менять нанесённый урон на *new\_damage\_done* в битве *battle1* для игрока с *nickname1*. Для удобства создам представление, в котором будут столбцы *battle\_id*, *player\_id*, *nickname*, *damage\_done*, *damage\_received*. Процедура будет работать именно с этим представлением.



До

	battle_id	player_id	nickname	damage_done	damage_received
►	1	1	Yellow Flash	150278	90132
	1	2	ahsaS	160321	101321
	1	3	hunterKiller228	98212	99121
	1	4	SaIyAjIn	127324	150237
	1	5	dmitriyBrekotkin	82112	92912
	1	6	your_master	103132	132136
	2	1	Yellow Flash	91372	83424
	2	3	hunterKiller228	94324	98432
	2	5	dmitriyBrekotkin	100000	132302
	3	1	Yellow Flash	107432	98424
	3	2	ahsaS	98433	92643

```
DELIMITER //
drop procedure if exists update_damage //
CREATE PROCEDURE update_damage(IN battle1 INT, IN nickname1 VARCHAR(45), IN new_damage_done int)
BEGIN
    DECLARE done INT DEFAULT FALSE;
    DECLARE player_id1 INT;
    DECLARE battle_id1 INT;
    DECLARE cur CURSOR FOR SELECT player_id, battle_id FROM playerDamage WHERE battle_id = battle1
    AND nickname = nickname1;
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;
    OPEN cur;
    update_loop: LOOP
        FETCH cur INTO player_id1, battle_id1;
        IF done THEN
            LEAVE update_loop;
        END IF;
        UPDATE playerDamage
        SET damage_done = new_damage_done
        WHERE player_id = player_id1
        AND battle_id = battle_id1;
    END LOOP;
    CLOSE cur;
END//
DELIMITER ;
```

После

```
38 • call update_damage(1, 'Yellow Flash', 1000000);
39 • select * from playerDamage;
```

<					
Result Grid					
Filter Rows: <input type="text"/>					
Export:  Wrap Cell Content: 					
	battle_id	player_id	nickname	damage_done	damage_received
▶	1	1	Yellow Flash	1000000	90132
	1	2	ahsaS	160321	101321
	1	3	hunterKiller228	98212	99121
	1	4	SaIyAjIn	127324	150237
	1	5	dmitriyBrekotkin	82112	92912
	1	6	your_master	103132	132136
	2	1	Yellow Flash	91372	83474