Icon

Description automatically generatedТЕХНИЧЕСКИ УНИВЕРСИТЕТ – СОФИЯ

**Катедра „Компютърни системи”**

**КУРСОВ ПРОЕКТ ПО БАЗИ ОТ ДАННИ**

# Студент: Мартин Георгиев Стоименов

**ФАК. № 121221049; Група: 46А**

A screenshot of a computer

Description automatically generated with medium confidence

**1. Да се проектира база от данни и да се представи ER диаграма със съответни CREATE TABLE заявки за средата MySQL.**

**2. Напишете заявка, в която демонстрирате SELECT с ограничаващо условие по избор.**

**3. Напишете заявка, в която използвате агрегатна функция и GROUP BY по ваш избор.**

**4. Напишете заявка, в която демонстрирате INNER JOIN по ваш избор.**

**5. Напишете заявка, в която демонстрирате OUTER JOIN по ваш избор.**

**6. Напишете заявка, в която демонстрирате вложен SELECT по ваш избор.**

**7. Напишете заявка, в която демонстрирате едновременно JOIN и агрегатна функция.**

**8. Създайте тригер по ваш избор.**

**9. Създайте процедура, в която демонстрирате използване на курсор.**

***Вашата работа трябва да включва: задание, ER-диаграма, CREATE TABLE заявки, всички останали заявки, решения на задачите от 2 до 9 и резултатите от тях.***

1. **Да се проектира база от данни и да се представи ER диаграма със съответни CREATE TABLE заявки за средата MySQL.**

A picture containing text, whiteboard

Description automatically generated

Обекта „ПОТРЕБИТЕЛ“ притежава id, name, password, и аз съм си добавил age, която може да е ограничена за деца над 12 години

Обект „Герои“ съдържа id, blood, attack, magicPoints, killedMonsters, frozenAtack\_protection, като това поле зависи от това дали героя е магьосник или боец, което се определя с допълнително поле: isWizard. Добавил съм и поле name, защото всеки герой може да притежава име в играта

Обект „чудовища“ има id, blood, attack, magicPoints, type ('zombie', 'troll', 'gnome') и отново съм добавил име

Обект „мисии“ има id, name, description, award

**Връзките между обектите са както следва:**

-Потребител (1) -> (М) Герои => един потребител може да притежава много герои, но един герой може да е само на един потребител. Връзката се осъществява с поле user\_id в таблица „герои“

-Герои (1) -> (М) Чудовища => един герой може да убива много чудовища, но едно чудовище умира само от един герой. Връзката се осъществява чрез чрез полето hero\_id таблица „чудовища“

-Герои (М) -> (М) Мисии => един герой участва в много мисии, и в една мисия участват много герои. Връзката се осъществява чрез помощна таблица с полета hero\_id, mission\_id

**Създаване на таблиците и вмъкване на информация в тях:**

CREATE DATABASE game;

USE game;

CREATE TABLE users (

id INT AUTO\_INCREMENT NOT NULL,

name VARCHAR(255) NOT NULL UNIQUE, #every username must be unique

password VARCHAR(255) NOT NULL,

age INT CHECK(age>=12),

PRIMARY KEY(id)

);

CREATE TABLE heroes(

id INT AUTO\_INCREMENT NOT NULL,

name VARCHAR(255), #every hero hs a name

blood INT NOT NULL,

atack INT NOT NULL,

magicPoints INT NOT NULL,

killedMonsters INT DEFAULT 0, #at first every hero isn't killed some monsters

isWizard BOOL DEFAULT TRUE, #if hero is wizard we have frozen\_atack, if hero is not wizard, it is fighter and we have protection

frozenAtack\_protection INT NOT NULL,

PRIMARY KEY (id),

user\_id INT NOT NULL, #user, to witch the hero belongs

CONSTRAINT FOREIGN KEY (user\_id) REFERENCES users (id)

ON DELETE CASCADE ON UPDATE CASCADE #Delete all heroes, if user delete his profile

);

CREATE TABLE monsters(

id INT AUTO\_INCREMENT NOT NULL,

name VARCHAR(255) NOT NULL,

blood INT NOT NULL,

atack INT NOT NULL,

magicPoints INT NOT NULL,

types ENUM('zombie', 'troll', 'gnome') NOT NULL,

PRIMARY KEY (id),

hero\_id INT, #hero, who kill the monster

CONSTRAINT FOREIGN KEY(hero\_id) REFERENCES heroes(id) ON DELETE SET NULL

);

CREATE TABLE missions (

id INT AUTO\_INCREMENT NOT NULL,

name VARCHAR(255) NOT NULL,

descriptions VARCHAR(255),

award INT,

PRIMARY KEY(id)

);

CREATE TABLE heroes\_missions(

id\_hero INT NOT NULL,

id\_mission INT, #не може да е NOT NULL защото при изтриване на мисия сме го направили да става NULL

CONSTRAINT FOREIGN KEY (id\_hero) REFERENCES heroes(id) ON DELETE CASCADE, #ако изтрием герой, не е необходимо вече да пазим, в кои мисии е участвал

CONSTRAINT FOREIGN KEY (id\_mission) REFERENCES missions(id) ON DELETE SET NULL #ако изтрием мисия, героя трябва да се запази че е участвал в дадена мисия, дори вече тя да не същшествува

);

INSERT INTO users (name, password, age) VALUES

('Alice', 'password1', 14),

('Bob', 'password2', 12),

('Charlie', 'password3', 19),

('David', 'password4', 21),

('Eve', 'password5', 13),

('Frank', 'password6', 14),

('Grace', 'password7', 39),

('Hannah', 'password8', 64),

('Isabella', 'password9', 15),

('Jack', 'password10', 16);

INSERT INTO heroes (name, blood, atack, magicPoints, killedMonsters, isWizard, frozenAtack\_protection, user\_id)

VALUES

('Gandalf', 100, 50, 200, 0, TRUE, 80, 1),

('Aragorn', 120, 70, 50, 2, FALSE, 10, 1),

('Legolas', 80, 90, 20, 0, FALSE, 0, 2),

('Gimli', 150, 60, 0, 5, FALSE, 10, 2),

('Frodo', 50, 20, 100, 0, TRUE, 50, 3),

('Sam', 70, 30, 50, 0, FALSE, 20, 3),

('Boromir', 100, 80, 0, 2, FALSE, 30, 4),

('Gollum', 40, 10, 70, 0, TRUE, 30, 4),

('Gandalf the White', 200, 100, 500, 9, TRUE, 100, 5),

('Sauron', 1000, 200, 1000, 0, TRUE, 500, 6),

('Saruman', 150, 80, 200, 0, TRUE, 70, 6),

('Faramir', 100, 60, 30, 0, FALSE, 40, 7),

('Eowyn', 90, 70, 40, 13, FALSE, 50, 7),

('Treebeard', 500, 100, 100, 5, TRUE, 150, 8),

('Bilbo', 60, 30, 70, 3, TRUE, 40, 9),

('Thorin Oakenshield', 200, 100, 17, 17, FALSE, 60, 10),

('Bard', 80, 50, 20, 5, FALSE, 70, 10),

('Gandalf the Grey', 150, 70, 300, 10, TRUE, 90, 1),

('Gimli', 130, 80, 0, 0, FALSE, 80, 3),

('Frodo', 60, 20, 120, 50, TRUE, 40, 5),

('Legolas', 100, 100, 50, 0, FALSE, 90, 8),

('Eomer', 120, 70, 20, 0, FALSE, 100, 10),

('Elrond', 150, 90, 200, 0, TRUE, 70, 2),

('Galadriel', 120, 80, 300, 20, TRUE, 90, 3),

('Glorfindel', 110, 90, 40, 0, FALSE, 10, 4),

('Boromir', 100, 80, 0, 11, FALSE, 20, 5),

('Frodo', 50, 20, 100, 0, TRUE, 50, 6),

('Gandalf the White', 200, 100, 500, 0, TRUE, 100, 7),

('Sauron', 1000, 200, 1000, 0, TRUE, 500, 8),

('Saruman', 150, 80, 200, 2, TRUE, 70, 9),

('Faramir', 100, 60, 30, 3, FALSE, 30, 10),

('Eowyn', 90, 70, 40, 3, FALSE, 40, 1),

('Treebeard', 500, 100, 100, 3, TRUE, 150, 2),

('Bilbo', 60, 30, 70, 3, TRUE, 40, 3),

('Thorin Oakenshield', 200, 100, 0, 2, FALSE, 50, 4),

('Bard', 80, 50, 20, 1, FALSE, 60, 5),

('Gandalf the Grey', 150, 70, 300, 0, TRUE, 90, 6),

('Gimli', 130, 80, 0, 1, FALSE, 70, 7),

('Frodo', 60, 20, 120, 0, TRUE, 40, 8),

('Legolas', 100, 100, 50, 7, FALSE, 80, 9),

('Eomer', 120, 70, 20, 6, FALSE, 90, 10),

('Elrond', 150, 90, 200, 8, TRUE, 70, 1),

('Galadriel', 120, 80, 300, 0, TRUE, 90, 2),

('Glorfindel', 110, 90, 40, 0, FALSE, 0, 3),

('Boromir', 100, 80, 0, 6, FALSE, 10, 4),

('Frodo', 50, 20, 100, 8, TRUE, 50, 5),

('Gandalf the White', 200, 100, 500, 0, TRUE, 100, 6),

('Sauron', 1000, 200, 1000, 12, TRUE, 500, 7),

('Saruman', 150, 80, 200, 0, TRUE, 70, 8),

('Faramir', 100, 60, 30, 0, FALSE, 110, 9),

('Eowyn', 90, 70, 40, 0, 56, FALSE, 10),

('Treebeard', 500, 100, 100, 1, TRUE, 150, 1),

('Bilbo', 60, 30, 70, 1, TRUE, 40, 2);

INSERT INTO monsters (name, blood, atack, magicPoints, types, hero\_id)

VALUES

("Monster1", 100, 20, 30, 'zombie', 1),

("Monster2", 120, 15, 25, 'troll', 2),

("Monster3", 80, 25, 35, 'gnome', NULL),

("Monster4", 110, 18, 20, 'zombie', 3),

("Monster5", 90, 30, 40, 'troll', NULL),

("Monster6", 95, 22, 27, 'gnome', 4),

("Monster7", 130, 12, 18, 'zombie', NULL),

("Monster8", 70, 35, 50, 'troll', 5),

("Monster9", 115, 20, 30, 'gnome', NULL),

("Monster10", 100, 15, 25, 'zombie', 6),

("Monster11", 85, 28, 35, 'troll', 7),

("Monster12", 75, 18, 20, 'gnome', NULL),

("Monster13", 120, 22, 27, 'zombie', 8),

("Monster14", 110, 30, 40, 'troll', NULL),

("Monster15", 95, 12, 18, 'gnome', 9),

("Monster16", 80, 35, 50, 'zombie', 10),

("Monster17", 90, 20, 30, 'troll', NULL),

("Monster18", 105, 28, 35, 'gnome', 11),

("Monster19", 125, 18, 20, 'zombie', NULL),

("Monster20", 100, 22, 27, 'troll', 12),

("Monster21", 110, 30, 40, 'gnome', 13),

("Monster22", 95, 12, 18, 'zombie', NULL),

("Monster23", 80, 35, 50, 'troll', 14),

("Monster24", 85, 20, 30, 'gnome', NULL),

("Monster25", 120, 28, 35, 'zombie', 15),

("Monster26", 100, 18, 20, 'troll', 16),

("Monster27", 110, 22, 27, 'gnome', NULL),

("Monster28", 90, 30, 40, 'zombie', 16),

("Monster29", 75, 15, 25, 'troll', NULL),

("Monster30", 105, 20, 30, 'gnome', 17),

("Monster31", 125, 12, 18, 'zombie', NULL),

("Monster32", 95, 35, 50, 'troll', 18),

("Monster33", 80, 20, 30, 'gnome', NULL),

("Monster34", 110, 28, 35, 'zombie', 19),

("Monster35", 100, 18, 20, 'troll', 20),

("Monster36", 120, 22, 27, 'gnome', NULL),

("Monster37", 90, 30, 40, 'zombie', 21),

("Monster38", 85, 15, 25, 'troll', NULL),

("Monster39", 100, 20, 30, 'gnome', 22),

("Monster40", 110, 12, 18, 'zombie', NULL),

("Monster41", 120, 35, 50, 'troll', 23),

("Monster42", 75, 20, 30, 'gnome', NULL),

("Monster43", 95, 28, 35, 'zombie', 24),

("Monster44", 85, 18, 20, 'troll', 25),

("Monster45", 105, 22, 27, 'gnome', NULL),

("Monster46", 125, 30, 40, 'zombie', 26),

("Monster47", 90, 15, 25, 'troll', NULL),

("Monster48", 110, 20, 30, 'gnome', 27),

("Monster49", 120, 12, 18, 'zombie', NULL),

("Monster50", 100, 35, 50, 'troll', 28),

("Monster51", 85, 18, 20, 'troll', 1),

("Monster52", 105, 22, 27, 'gnome', NULL),

("Monster53", 125, 30, 40, 'zombie', 2),

("Monster54", 90, 15, 25, 'troll', NULL),

("Monster55", 110, 20, 30, 'gnome', 3),

("Monster56", 120, 12, 18, 'zombie', NULL),

("Monster57", 100, 35, 50, 'troll', 4),

("Monster58", 75, 20, 30, 'gnome', NULL),

("Monster59", 95, 28, 35, 'zombie', 5),

("Monster60", 85, 18, 20, 'troll', 6),

("Monster61", 105, 22, 27, 'gnome', NULL),

("Monster62", 125, 30, 40, 'zombie', 7),

("Monster63", 90, 15, 25, 'troll', NULL),

("Monster64", 110, 20, 30, 'gnome', 8),

("Monster65", 120, 12, 18, 'zombie', NULL),

("Monster66", 100, 35, 50, 'troll', 9),

("Monster67", 75, 20, 30, 'gnome', NULL),

("Monster68", 95, 28, 35, 'zombie', 10),

("Monster69", 85, 18, 20, 'troll', 11),

("Monster70", 105, 22, 27, 'gnome', NULL),

("Monster71", 125, 30, 40, 'zombie', 12),

("Monster72", 90, 15, 25, 'troll', NULL),

("Monster73", 110, 20, 30, 'gnome', 13),

("Monster74", 120, 12, 18, 'zombie', NULL),

("Monster75", 100, 35, 50, 'troll', 14),

("Monster76", 75, 20, 30, 'gnome', NULL),

("Monster77", 95, 28, 35, 'zombie', 15),

("Monster78", 85, 18, 20, 'troll', 16),

("Monster79", 105, 22, 27, 'gnome', NULL),

("Monster80", 125, 30, 40, 'zombie', 17),

("Monster81", 90, 15, 25, 'troll', NULL),

("Monster82", 110, 20, 30, 'gnome', 18),

("Monster83", 120, 12, 18, 'zombie', NULL),

("Monster84", 100, 35, 50, 'troll', 19),

("Monster85", 75, 20, 30, 'gnome', NULL),

("Monster86", 95, 28, 35, 'zombie', 20);

INSERT INTO missions (name, descriptions, award) VALUES

('Collecting herbs', 'Gather rare herbs from the forest', 2),

('Clearing a dungeon', 'Clear out a dangerous dungeon and secure its treasures', 5),

('Rescuing hostages', 'Free hostages held by a group of bandits', 3),

('Hunting for food', 'Hunt wild game to feed a starving village', 1),

('Stopping a thief', 'Catch a notorious thief and recover stolen goods', 4),

('Recovering an artifact', 'Retrieve a powerful artifact from a guarded tomb', 7),

('Tracking a monster', 'Hunt down a dangerous monster terrorizing a nearby town', 6),

('Escorting a caravan', 'Safely escort a caravan through bandit-infested territory', 3),

('Breaking a siege', 'Break a siege and lift the blockade of a fortified city', 8),

('Exploring ruins', 'Explore mysterious ruins and uncover their secrets', 5),

('Assassinating a target', 'Eliminate a high-value target for a wealthy client', 9),

('Sabotaging an operation', 'Sabotage the operations of a rival guild or organization', 6),

('Resolving a conflict', 'Resolve a conflict between two warring factions and restore peace', 4),

('Spying on an enemy', 'Infiltrate enemy territory and gather crucial intelligence', 3),

('Training new recruits', 'Train and mentor a group of new recruits for a guild or organization', 2);

INSERT INTO heroes\_missions (id\_hero, id\_mission)

SELECT

FLOOR(RAND() \* 53) + 1, # generate a random number between 1 and 53 for id\_hero

FLOOR(RAND() \* 15) + 1 # generate a random number between 1 and 15 for id\_mission

FROM

(SELECT a.n + b.n \* 10 + c.n \* 100 + 1 AS n

FROM

(SELECT 0 AS n UNION SELECT 1 UNION SELECT 2 UNION SELECT 3 UNION SELECT 4

UNION SELECT 5 UNION SELECT 6 UNION SELECT 7 UNION SELECT 8 UNION SELECT 9) AS a

CROSS JOIN

(SELECT 0 AS n UNION SELECT 1 UNION SELECT 2 UNION SELECT 3 UNION SELECT 4

UNION SELECT 5 UNION SELECT 6 UNION SELECT 7 UNION SELECT 8 UNION SELECT 9) AS b

CROSS JOIN

(SELECT 0 AS n UNION SELECT 1 UNION SELECT 2 UNION SELECT 3 UNION SELECT 4

UNION SELECT 5 UNION SELECT 6 UNION SELECT 7 UNION SELECT 8 UNION SELECT 9) AS c

) AS numbers

LIMIT 300;

**Да бъде възможно всеки различен потребител да се автентикира с име и парола и да се извлича информация за всички негови герои:**

SELECT \* FROM heroes

JOIN users ON heroes.user\_id = users.id

WHERE users.name = "Bob" AND users.password = "password2";

Table

Description automatically generated

1. **Напишете заявка, в която демонстрирате SELECT с ограничаващо условие по избор.**

CREATE VIEW usl2 AS SELECT heroes.name AS heroName, missions.name AS missionName

FROM heroes JOIN heroes\_missions ON heroes\_missions.id\_hero = heroes.id

JOIN missions ON heroes\_missions.id\_mission = missions.id

WHERE missions.name LIKE "%food%" AND heroes.name NOT LIKE "%Frodo%"; #тази заявка връща имената на героите и мисиите в които са участвали свързани с "храна", но без героя Фродо

SELECT \* FROM usl2;

Table

Description automatically generated

1. **Напишете заявка, в която използвате агрегатна функция и GROUP BY по ваш избор.**

CREATE VIEW usl3 AS SELECT COUNT(heroes.id) AS countOfHeroes, heroes.blood

FROM heroes

WHERE heroes.atack BETWEEN 0 AND 100

GROUP BY heroes.blood; #тази заявка връща броя герои, които имат определено количество кръв

SELECT \* FROM usl3;

Table

Description automatically generated

1. **Напишете заявка, в която демонстрирате INNER JOIN по ваш избор.**

CREATE VIEW usl4 AS SELECT users.name AS userName, heroes.name AS heroName, heroes.blood, heroes.atack,

heroes.magicPoints, heroes.killedMonsters, missions.name AS missionName

FROM users JOIN heroes ON heroes.user\_id = users.id

JOIN heroes\_missions ON heroes\_missions.id\_hero = heroes.id

JOIN missions ON heroes\_missions.id\_mission = missions.id

WHERE heroes.killedMonsters!=0

ORDER BY users.name; #тази заявка връща имената на играчите, както и техните герои и кой герой в коя мисия е участвал

SELECT \* FROM usl4;

Table

Description automatically generated

SELECT monsters.name as MonsterName, heroes.name as HeroName

FROM monsters JOIN heroes ON monsters.hero\_id = heroes.id; #тази заявка връща като резултат само убитите чудовища, и от кой са убити

Graphical user interface, application, table

Description automatically generated

1. **Напишете заявка, в която демонстрирате OUTER JOIN по ваш избор.**

SELECT monsters.name as MonsterName, heroes.name as HeroName

FROM monsters LEFT JOIN heroes ON monsters.hero\_id = heroes.id; #тази заявка връща всички чудовища (живи и убити) и героите, от които са убити

Table

Description automatically generated

SELECT monsters.name as MonsterName, heroes.name as HeroName

FROM monsters RIGHT JOIN heroes ON monsters.hero\_id = heroes.id; #тази заявка връща всички герои и убитите от тях чудовища

Table

Description automatically generated

**SELF JOIN, CONCAT**

SELECT heroes.name as HeroName, GROUP\_CONCAT(monsters.name ORDER BY monsters.id SEPARATOR ', ') as MonsterName

FROM heroes LEFT JOIN monsters ON monsters.hero\_id = heroes.id

GROUP BY heroes.id; #тази заявка връща всички герои и убитите от тях чудовища но в една колона

Graphical user interface, table

Description automatically generated

CREATE VIEW usl5 AS (SELECT monsters.name as MonsterName, heroes.name as HeroName

FROM monsters LEFT JOIN heroes ON monsters.hero\_id = heroes.id)

UNION

(SELECT monsters.name as MonsterName, heroes.name as HeroName

FROM monsters RIGHT JOIN heroes ON monsters.hero\_id = heroes.id); #тази заявка връща всички герои и всички чудовища, без да повтаря записи

SELECT \* FROM usl5;

Table

Description automatically generated

1. **Напишете заявка, в която демонстрирате вложен SELECT по ваш избор.**

CREATE VIEW usl6 AS SELECT heroes.name AS heroName, missions.name AS missionName

FROM heroes JOIN missions

ON heroes.id IN(

SELECT id\_hero

FROM heroes\_missions

WHERE heroes\_missions.id\_mission = missions.id

)

WHERE missions.name LIKE "%food%" AND heroes.name NOT LIKE "%Frodo%"; #тази заявка връща имената на героите и мисиите в които са участвали свързани с "храна", но без героя Фродо

SELECT \* FROM usl6;

Table

Description automatically generated

1. **Напишете заявка, в която демонстрирате едновременно JOIN и агрегатна функция.**

SELECT users.name, COUNT(heroes.id)

FROM users JOIN heroes

ON heroes.user\_id = users.id

GROUP BY users.name

ORDER BY COUNT(heroes.id) DESC; #тази заявка имената на играчите и броя герои, които имат

Graphical user interface

Description automatically generated with low confidence

SELECT heroes.name, SUM(missions.award)

FROM heroes

JOIN heroes\_missions ON heroes\_missions.id\_hero = heroes.id

JOIN missions ON heroes\_missions.id\_mission = missions.id

WHERE heroes.isWizard = TRUE

GROUP BY heroes.name

HAVING SUM(missions.award)>10

ORDER BY SUM(missions.award) DESC

LIMIT 5; #тази заявка връща имената на героите и точките, които са взели от мисии в ТОП 5

Table

Description automatically generated

CREATE VIEW usl7 AS SELECT users.name, SUM(missions.award) AS Points

FROM heroes JOIN users ON heroes.user\_id = users.id

JOIN heroes\_missions ON heroes\_missions.id\_hero = heroes.id

JOIN missions ON heroes\_missions.id\_mission = missions.id

GROUP BY users.name

HAVING SUM(missions.award)>20

ORDER BY SUM(missions.award) DESC

LIMIT 10; #тази заявка връща играчите и общия брой точки от мисии на всеки от тях, в ТОП 10

SELECT \* FROM usl7;

Table

Description automatically generated

1. **Създайте тригер по ваш избор**

delimiter |

create trigger not\_negative\_value\_of\_blood BEFORE UPDATE ON heroes #ако някой герой има по малко от 0 кръв, то той е умрял

FOR EACH ROW

BEGIN

IF (NEW.blood < 0)

THEN SET NEW.blood = 0;

END IF;

END;

|

delimiter ;

SELECT \* FROM heroes;

UPDATE heroes SET blood = blood - 50 WHERE heroes.id=4;

SELECT \* FROM heroes;



След 1 update:



След още n-брой update:



**Създайте тригер по ваш избор - 2**

create table killing\_monsters(

id int auto\_increment primary key,

operation ENUM('INSERT','UPDATE','DELETE') not null,

id\_monster INT NOT NULL,

oldBlood INT,

newBlood INT,

oldAtack INT,

newAtack INT,

oldMagicPoints INT,

newMagicPoints INT,

dateOfChange datetime

)Engine = Innodb;

delimiter |

CREATE TRIGGER change\_blood\_of\_monster BEFORE UPDATE ON monsters

FOR EACH ROW

BEGIN

IF (NEW.blood < 0)

THEN SET NEW.blood = 0;

END IF;

INSERT INTO killing\_monsters(operation,

id\_monster,

oldBlood,

newBlood,

oldAtack,

newAtack,

oldMagicPoints,

newMagicPoints,

dateOfChange)

VALUES ('UPDATE',

OLD.id,

OLD.blood,

CASE NEW.blood WHEN OLD.blood THEN NULL ELSE NEW.blood END,

OLD.atack,

CASE NEW.atack WHEN OLD.atack THEN NULL ELSE NEW.atack END,

OLD.magicPoints,

CASE NEW.magicPoints WHEN OLD.magicPoints THEN NULL ELSE NEW.magicPoints END,

NOW());

END;

|

Delimiter ;

SELECT \* FROM monsters;

UPDATE monsters SET blood = blood - 50 WHERE monsters.id=4;

SELECT \* FROM monsters;

SELECT \* FROM killing\_monsters;

update:







След като стане 0, повече не се намалява кръвта, защото чудовището вече е убито

Table

Description automatically generated

1. **Създайте процедура, в която демонстрирате използване на курсор - променяме кръвта на всеки герой, участвал в определена мисия с 50 нагоре**

DELIMITER //

CREATE PROCEDURE update\_heroes\_blood(IN id\_mission\_out INT)

BEGIN

DECLARE done INT DEFAULT 0;

DECLARE hero\_id INT;

DECLARE hero\_blood INT;

DECLARE hero\_cursor CURSOR FOR

SELECT heroes.id, heroes.blood

FROM heroes

INNER JOIN heroes\_missions ON heroes.id = heroes\_missions.id\_hero

WHERE heroes\_missions.id\_mission = id\_mission\_out;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

OPEN hero\_cursor;

hero\_loop: WHILE(done = 0)

DO

FETCH hero\_cursor INTO hero\_id, hero\_blood;

IF done=1 THEN

LEAVE hero\_loop;

END IF;

-- update hero blood

UPDATE heroes SET blood = hero\_blood + 50 WHERE id = hero\_id;

END WHILE;

CLOSE hero\_cursor;

END//

DELIMITER ;

CALL update\_heroes\_blood(4);

SELECT heroes.id, heroes.name, heroes.blood FROM heroes;

Table

Description automatically generatedGraphical user interface, table

Description automatically generatedTable

Description automatically generated