

SEMINARSKA NALOGA – PODATKOVNE BATE

Datum: 12. 05. 2024

Ime in priimek: Armen Hodža

Vpisna številka: 63230105

1. Naloga

Iz tabele x_world, naredite (CREATE TABLE) in napolnite tabele z naslednjimi relacijskimi shemami in pomeni. Pozorni bodite na primarne in tuje ključe ter pravilni vrsti red ustvarjanja tabel.

```
CREATE TABLE pleme(  
    tid integer default 0 NOT NULL,  
    tribe varchar(100) default '' NOT NULL,  
    PRIMARY KEY (tid)  
);  
  
CREATE TABLE aliansa(  
    aid integer default 0 NOT NULL,  
    alliance varchar(100) default '',  
    PRIMARY KEY (aid)  
);  
  
CREATE TABLE igralec(  
    pid integer default 0 NOT NULL,  
    player varchar(100) default '' NOT NULL,  
    tid integer default 0 NOT NULL,  
    aid integer default 0 NOT NULL,  
    PRIMARY KEY (pid),  
    FOREIGN KEY (tid) REFERENCES pleme(tid),  
    FOREIGN KEY (aid) REFERENCES aliansa(aid)  
);  
  
CREATE TABLE naselje(  
    id integer default 0 NOT NULL,  
    vid integer default 0 NOT NULL,  
    village varchar(100) default '' NOT NULL,  
    x integer default 0 NOT NULL,  
    y integer default 0 NOT NULL,  
    population integer default 0 NOT NULL,  
    pid integer default 0 NOT NULL,
```

PRIMARY KEY (id),

FOREIGN KEY (pid) REFERENCES igralec(pid)

);

INSERT INTO pleme VALUES(1,'Rimljani');

INSERT INTO pleme VALUES(2,'Tevtoni');

INSERT INTO pleme VALUES(3,'Galci');

INSERT INTO pleme VALUES(4,'Narava');

INSERT INTO pleme VALUES(5,'Natarji');

INSERT INTO pleme VALUES(6,'Egipčani');

INSERT INTO pleme VALUES(7,'Huni');

Tid	Tribe
1	Rimljani
2	Tevtoni
3	Galci
4	Narava
5	Natarji
6	Egipčani
7	Huni

INSERT INTO aliansa SELECT DISTINCT aid, alliance FROM x_world;

Aid	Alliance
0	
1	TG-TS
3	A
6	SJ
7	STARK
9	FIGHT-БГ
13	KT2

INSERT INTO igralec SELECT DISTINCT pid, player, tid, aid FROM x_world;

Pid	Player	Tid	Aid
1	Natars	5	0
2	Multihunter	1	1
6	Al ajz on mi	2	27
9	WaRoR	6	18
10	Тута Бугарин	7	95
14	Grey	7	7
15	Orcus	7	24

INSERT INTO naselje SELECT DISTINCT id, vid, village, x, y, population, pid FROM x_world;

Id	Vid	Village	X	Y	Population	Pid
212	47024	León	-39	250	57	5124
447	43603	*-C-*	196	250	246	5863
448	41943	*-C-*	197	250	352	5863
786	35141	Zemo2	34	249	518	19826
913	44837	m 03	161	249	141	9773
1002	27188	GAXXXXXXXXXXXXXXIT	250	249	852	4807
1192	32461	Lukzor	-61	248	787	16716

2. Naloga

a) Naredite pogled `x_view`, ki bo iz novih tabel naredil pogled ekvivalenten originalni tabeli `x_world` (`CREATE VIEW`).

```
CREATE VIEW x_view
```

```
AS
```

```
SELECT n.id AS id, n.x AS x, n.y AS y, p.tid AS tid, n.vid AS vid, n.village AS village, i.pid AS pid,
```

```
i.player AS player, a.aid AS aid, a.alliance AS alliance, n.population AS population
```

```
FROM naselje n
```

```
JOIN igralec i ON n.pid = i.pid
```

```
JOIN pleme p ON i.tid = p.tid
```

```
JOIN aliansa a ON i.aid = a.aid;
```

Id	X	Y	Tid	Vid	Village	Pid	Player	Aid	Alliance	Population
26733	-71	197	5	22796	Natars -71 197	1	Natars	0		597
32581	-235	185	5	26968	Natars -235 185	1	Natars	0		217
33141	-176	184	5	20726	Natars -176 184	1	Natars	0		473
35053	233	181	5	29089	Natars 233 181	1	Natars	0		628
40940	108	169	5	20621	Natars 108 169	1	Natars	0		401

b) S pomočjo SQL poizvedb smiselno preverite, ali sta vsebini `x_view` in `x_world` identični.

-- Če sta identični, vrne prazno množico

```
SELECT * FROM x_view
```

```
EXCEPT
```

```
SELECT * FROM x_world;
```

Result Grid										
Filter Rows: <input type="text"/>										
Export: <input type="button" value=""/>										
Wrap Cell Content: <input type="button" value=""/>										
id	x	y	tid	vid	village	pid	player	aid	alliance	population

Id	X	Y	Tid	Vid	Village	Pid	Player	Aid	Alliance	Population
----	---	---	-----	-----	---------	-----	--------	-----	----------	------------

c) S pomočjo ustreznih DDL ukazov ustvarite tabelo `top5(alliance, SteviloNaselij)`, ki hrani alianse z največ naselji. Dodajte bazne prožilce tako, da se bo lista posodabljala vsakič, ko nekdo zgradi novo naselje.

```
CREATE TABLE top5 (
```

```
    alliance VARCHAR(100),
```

```
    SteviloNaselij INTEGER DEFAULT 0 NOT NULL,
```

```
    PRIMARY KEY (alliance)
```

```
);
```

```

-- Napolnim tabelo

INSERT INTO top5 (alliance, SteviloNaselij)

SELECT a.alliance, COUNT(*) AS SteviloNaselij

FROM naselje n

JOIN igralec i ON n.pid = i.pid

JOIN aliansa a ON i.aid = a.aid

WHERE alliance != ''

GROUP BY a.alliance

ORDER BY COUNT(*) DESC

LIMIT 5;

```

Alliance	SteviloNaselij
RS-SN	555
DT RS #2	533
RS-H1N1™	525
DLZ SOS	459
DARK™	458

```

-- Bazni prožilec

DELIMITER //

CREATE TRIGGER prozilec_top5

AFTER INSERT ON naselje

FOR EACH ROW

BEGIN

    -- Izbrise vse iz tabele top5

    DELETE FROM top5;

    -- Ponovno vstavi podatke (vključno z novimi podatki) v tabelo top5

    INSERT INTO top5 (alliance, SteviloNaselij)

    SELECT a.alliance, COUNT(*) AS SteviloNaselij

    FROM naselje n

    JOIN igralec i ON n.pid = i.pid

    JOIN aliansa a ON i.aid = a.aid

    WHERE alliance != ''

    GROUP BY a.alliance

    ORDER BY COUNT(*) DESC

```

LIMIT 5;

END;

//

DELIMITER ;

3. Naloga

a) Kateri igralec ima največje naselje?

```
SELECT i.player, MAX(n.population) AS Najvecje_naselje
FROM igralec i
JOIN naselje n ON i.pid = n.pid
GROUP BY i.player
ORDER BY največje_naselje DESC
LIMIT 1;
```

Player	Najvecje_naselje
Bogatin	1243

b) Koliko igralcev ima nadpovprečno veliko naselje?

```
SELECT COUNT(*) AS Stevilo
FROM (
    SELECT i.player
    FROM igralec i
    JOIN naselje n ON i.pid = n.pid
    GROUP BY i.player
    HAVING AVG(n.population) > (SELECT AVG(population) FROM naselje)
) AS nadpovprečni_igralci;
```

Stevilo
1180

c) Izpišite podatke o vseh naseljih igralcev brez alianse, urejeno padajoče po x in nato y koordinati.

```
SELECT i.player, n.*
FROM naselje n
JOIN igralec i ON n.pid = i.pid
WHERE i.aid = ''
ORDER BY n.x DESC, n.y DESC;
```

Player	Id	Vid	Village	X	Y	Population	Pid
Norixus	40080	35105	04Odin	250	171	544	72
Norixus	39578	38845	05Thor	249	172	330	72
Norixus	40078	21410	02Slavs	248	171	787	72
Norixus	41580	43834	New village	247	168	74	72

d) Katero pleme je najštevilčnejše (glede na populacijo)?

```
SELECT p.tid, p.tribe, SUM(n.population) AS Skupna_populacija
FROM pleme p
JOIN igralec i ON p.tid = i.tid
JOIN naselje n ON i.pid = n.pid
GROUP BY p.tid
ORDER BY skupna_populacija DESC
LIMIT 1;
```

Tid	Tribe	Skupna_populacija
6	Egipčani	2532967

e) Izpišite število nadpovprečno močnih alians (povprečje populacije računajte glede na alianse, ne na vse igralce).

```
SELECT COUNT(*) AS Stevilo
FROM (
    SELECT a.alliance
    FROM aliansa a
    JOIN igralec i ON a.aid = i.aid
    JOIN naselje n ON i.pid = n.pid
    GROUP BY a.aid
    HAVING AVG(n.population) > (SELECT AVG(population) FROM naselje)
) AS nadpovprecne_alianse;
```

Stevilo
87

f) *Igralec bananamen želi preimenovati vsa svoja naselja na naslednji način. Uredil jih bo po populaciji, najmočnejše bo »Banana-00«, naslednje »Banana-01« in tako dalje. Nalogo lahko rešite v več korakih (zaporedju poizvedb).

```
SET @st := -1;          -- spremenljivka

UPDATE naselje n
JOIN igralec i ON n.pid = i.pid

SET n.village = CONCAT('Banana-', LPAD(@st := @st + 1, 2, '0'))

WHERE i.player = 'bananamen'

ORDER BY n.population DESC;
```

Player	Village	Population
bananamen	Banana-00	971
bananamen	Banana-01	901
bananamen	Banana-02	882
bananamen	Banana-03	875
bananamen	Banana-04	836

g) Napišite shranjen podprogram, ki za poljubne koordinate (parametra x in y) vrne populacijo na največ podani razdalji (parameter razdalja). Npr. razdalja 10 pomeni vse koordinate od vključno (x-10, y-10) do (x+10, y+10). Za preverjanje robnih pogojev (koordinate izven [-400,400] po potrebi uporabite IF stavek.

DELIMITER //

CREATE PROCEDURE PopulacijaNaPodaniRazdalji(x INTEGER, y INTEGER, razdalja INTEGER)

BEGIN

 DECLARE x_min INT;

 DECLARE x_max INT;

 DECLARE y_min INT;

 DECLARE y_max INT;

 IF x < -400 THEN

 SET x = -400;

 ELSEIF x > 400 THEN

 SET x = 400;

 END IF;

 IF y < -400 THEN

 SET y = -400;

 ELSEIF y > 400 THEN

 SET y = 400;

 END IF;

 SET x_min = GREATEST(-400, x - razdalja);

 SET x_max = LEAST(400, x + razdalja);

 SET y_min = GREATEST(-400, y - razdalja);

 SET y_max = LEAST(400, y + razdalja);

 SELECT SUM(n.population)

 FROM naselje n

```

WHERE n.x BETWEEN x_min AND x_max AND n.y BETWEEN y_min AND y_max;

END //

DELIMITER ;

```

h) Izpišite imena igralcev, ki imajo vsa svoja naselja na območju x, ki je med 100 in 200 in y, ki je med 0 in 100.

```

SELECT DISTINCT player
FROM igralec
WHERE pid NOT IN (
    SELECT DISTINCT i.pid
    FROM naselje n
    JOIN igralec i ON n.pid = i.pid
    WHERE n.x < 100 OR n.x > 200 OR n.y < 0 OR n.y > 100
);

```

Player
TotalnoPoludeli
grand_danko
mataba
Сирак скитник
slavi

i) Napišite poizvedbo, ki bo poiskala vsa naselja s populacijo točno 1000. Napišite še stavek, ki bi to poizvedbo pohitрил, če bi bila tabela naselje dovolj velika.

```

SELECT *
FROM naselje
WHERE population = 1000;

```

Id	Vid	Village	X	Y	Population	Pid
78978	20619	Batica	70	93	1000	18206

-- Kreiram indeks na atribut population, in s tem pohitrim poizvedbo

```

CREATE INDEX index_population ON naselje(population);

```

j) Poiščite igralce, ki imajo umirajoče naselje. Za umirajoče naselje vzemite tista naselja, ki imajo manj kot 3% povprečne populacije igralca (povprečna populacija igralca je populacija igralca ulomljeno s številom njegovih naselij).

```
SELECT i.player
FROM igralec i
JOIN (
    SELECT pid, AVG(population) AS povp
    FROM naselje
    GROUP BY pid
) AS povprecje ON i.pid = povprecje.pid
JOIN naselje n ON i.pid = n.pid
WHERE n.population < 0.03 * povprecje.povp;
```

Player
Norixus
Lemi chl
japy
verbanko
getlers

4. Naloga

V programskem jeziku Python napišite program, ki se priključi na podatkovno bazo in za celotno igralno polje izračuna gostoto populacije in gostoto populacije določene alianse.

Najprej importam modul pyodbc.

Nato se program poveže z bazo podatkov, ter izbriše tabeli gostotaPopulacije in gostotaAlianse, če obstajata, ter jih ponovno ustvari.

Program nato iterira skozi zanki, premikajoč se po polju po 10 enot, shrani skupno populacijo na tem območju, shrani najmočnejšo alianso na tem območju ter te podatke vstavi v tabelah po določenih formulah.

Na koncu se spremembe potrdijo in povezava z bazo se zapre.