Student: Bazaochi Dumitru

Evaluare periodica repetata

Ex 1

Să se proiecteze o interogare care ar determina numele și vârsta persoanelor cu cel mai mare număr de rude.

**SELECT** persoane.Numele, persoane.Virsta, **COUNT**(persoane.idPersoana) **AS** rude

**FROM** persoane

**INNER** **JOIN** rude r **ON** persoane.idPersoana = r.idpersoana1

**OR** persoane.idPersoana = r.idpersoana2

**GROUP** **BY** persoane.idPersoana

**HAVING** rude = (**SELECT** **COUNT**(persoane.idPersoana) **AS** rude

**FROM** persoane

**INNER** **JOIN** rude r **ON** persoane.idPersoana = r.idpersoana1

**OR** persoane.idPersoana = r.idpersoana2

**GROUP** **BY** persoane.idPersoana

**LIMIT** 1);

Ex 2

Să se proiecteze o interogare care ar returna numele persoanelor care sunt amici. Perechile de prieteni nu trebuie să se repete.

**SELECT** p1.Numele, p2.Numele

**FROM** amici

**INNER** **JOIN** persoane p1 **ON** p1.idPersoana = amici.idpersoana1

**INNER** **JOIN** persoane p2 **ON** p2.idPersoana = amici.idpersoana2;

Ex 3

Să se proiecteze o interogare care ar determina rudele persoanelor care sunt mai mici ca ei. Să se afișeze perechi de persoane fără repetări.

**SELECT** P1.NUMELE, P1.Virsta, P2.NUMELE, P2.Virsta

**FROM** PERSOANE **AS** P1

**INNER** **JOIN** RUDE **ON** P1.IDPERSOANA = RUDE.IDPERSOANA1

**INNER** **JOIN** PERSOANE **AS** P2 **ON** P2.IDPERSOANA = RUDE.IDPERSOANA2

**WHERE** P1.VIRSTA<>P2.VIRSTA;

Ex 4

Să se elaboreze o procedură care pentru o persoană ar determina lista membrilor familiei sale. Dacă A este membru al familiei X și B este membru al familiei X, atunci B trebuie să apară în lista membrilor familiei lui A. Ca parametru este dat numele persoanei.

delimiter $$

**CREATE** **PROCEDURE** proc4\_ev2(**IN** numpers **VARCHAR**(20))

**BEGIN**

**SELECT** **DISTINCT** rude.idpersoana1 **AS** A,

rude.idpersoana2 **AS** **X**,

**if** ( r.idpersoana1 <> rude.idpersoana1 **AND** r.idpersoana1 <> rude.idpersoana2, r.idpersoana1,

**if** ( r.idpersoana2 <> rude.idpersoana1 **AND** r.idpersoana2 <> rude.idpersoana2, r.idpersoana2,

**NULL** )) **AS** B

**FROM** rude

**INNER** **JOIN** rude r **ON** r.idpersoana1 = rude.idpersoana2

**INNER** **JOIN** persoane **ON** persoane.idpersoana= rude.idpersoana1 **OR** persoane.idPersoana=rude.idpersoana2

**WHERE** Persoane.Numele=numpers

**HAVING** B **IS** **NOT** **NULL**;

**END** $$

delimiter ;

**CALL** proc4\_ev2("Farouk");

Ex 5

Să se elaboreze o procedură care pentru o persoană ar determina numărul de prieteni. Ca parametru este dat numele persoanei.

delimiter $$

**CREATE** **PROCEDURE** proc5\_ev(nume\_pers **VARCHAR**(20))

**BEGIN**

**SELECT** **COUNT**(persoane.idPersoana)

**FROM** persoane

**INNER** **JOIN** amici **ON** persoane.idPersoana = amici.idpersoana1

**OR** persoane.idPersoana = amici.idpersoana2

**WHERE** persoane.Numele = nume\_pers

**GROUP** **BY** persoane.idPersoana;

**END**; $$

delimiter ;

**CALL** proc5\_ev("Elvi");

Ex 6

Să se elaboreze o funcție care returnează True sau False dacă două persoane sunt prieteni. Numele persoanelor sunt date ca parametru.

delimiter $$

**CREATE** **FUNCTION** func6\_ev(nume\_pers1 **VARCHAR**(40), nume\_pers2 **VARCHAR**(40))

**RETURNS** **VARCHAR**(5)

**BEGIN**

**if** **EXISTS** (**SELECT** \*

**FROM** persoane

**INNER** **JOIN** amici **ON** persoane.idPersoana = amici.idpersoana1

**OR** persoane.idPersoana =amici.idpersoana2

**WHERE** persoane.Numele = nume\_pers1 **OR** persoane.Numele = nume\_pers2)

**then**

**RETURN** "TRUE";

**ELSE**

**RETURN** "FALSE";

**END** **if**;

**END** $$

delimiter ;

**SELECT** func6\_ev("Farouk", "Chris");

Ex 7

Să se creeze o vedere care ar conține lista persoanelor ce au un membru de familie, dar nu au nici un prieten. Este posibil de a adăuga date în tabele prin intermediul vederii? Este posibil de a elimina date din tabele prin intermediul vederii? Este posibil de a modifica date în tabele prin intermediul vederii? Dați exemple de operații de actualizare prin intermediul vederilor, dacă este posibil.

**CREATE** **VIEW** VIEW7 **AS**

**SELECT** p1.idPersoana, p1.Numele

**FROM** persoane p1

**INNER** **JOIN** rude **ON** p1.idPersoana = rude.idpersoana1

**OR** p1.idPersoana = rude.idpersoana2

**WHERE** p1.idPersoana **NOT** **IN** (

**SELECT** p2.idPersoana

**FROM** persoane p2

**INNER** **JOIN** amici **ON** p2.idPersoana = amici.idpersoana1

**OR** p2.idPersoana = amici.idpersoana2 )

**GROUP** **BY** p1.idPersoana

**HAVING** **COUNT**(rude.idpersoana1) = 1;

**SELECT** \* **FROM** VIEW7;

Ex 8

Dați câte un exemplu de vedere care ar permite de a adăuga date în tabelul PERSOANE, a doua vedere ar permite modificarea, și a treia ar permite eliminarea datelor din tabelul menționat. Toate trei vederi trebuie să conțină interogări diferite.

Insert:

**CREATE** **VIEW** VIEW8 **AS**

**SELECT** \*

**FROM** persoane

**SELECT** \* **FROM** VIEW8

**INSERT** **INTO** VIEW8 **VALUES** (18, 'Dumitru', 21);

Update:

**CREATE** **VIEW** VIEW8 **AS**

**SELECT** \*

**FROM** persoane

**SELECT** \* **FROM** VIEW8

**UPDATE** VIEW8

**SET** Numele = "Samantha2" **WHERE** Numele = "Samantha";

Delete:

**CREATE** **VIEW** VIEW8 **AS**

**SELECT** \*

**FROM** persoane

**SELECT** \* **FROM** VIEW8

**DELETE** **FROM** VIEW8 **WHERE** idPersoana = 18;