Student: COZMA DENIS

**Laborator 2**

1. ***Completați tabelele cu următoarele date.***

**Tabelul Persoane**

|  |  |  |
| --- | --- | --- |
| **idPersoana** | **Numele** | **Vîtsta** |
| 1 | Elvi | 19 |
| 2 | Farouk | 19 |
| 3 | Sam | 19 |
| 4 | Tiany | 19 |
| 5 | Nadia | 14 |
| 6 | Chris | 12 |
| 7 | Kris | 10 |
| 8 | Bethany | 16 |
| 9 | Louis | 17 |
| 10 | Austin | 22 |
| 11 | Gabriel | 21 |
| 12 | Jessica | 20 |
| 13 | John | 16 |
| 14 | Alfred | 19 |
| 15 | Samantha | 17 |
| 16 | Craig | 17 |

**Tabelul Amici**

|  |  |
| --- | --- |
| **Idpersoana1** | **Idpersoana2** |
| 1 | 2 |
| 1 | 3 |
| 2 | 4 |
| 2 | 6 |
| 3 | 9 |
| 4 | 9 |
| 7 | 5 |
| 5 | 8 |
| 6 | 10 |
| 13 | 6 |
| 7 | 6 |
| 8 | 7 |
| 9 | 11 |
| 12 | 9 |
| 10 | 15 |
| 12 | 11 |
| 12 | 15 |
| 13 | 16 |
| 15 | 13 |
| 16 | 14 |

**Tabelul Rude**

|  |  |
| --- | --- |
| **Idpersoana1** | **Idpersoana2** |
| 4 | 6 |
| 2 | 4 |
| 9 | 7 |
| 7 | 8 |
| 11 | 9 |
| 13 | 10 |
| 14 | 5 |
| 12 | 13 |

Crearea bazei de date si a tabelelor

**CREATE** **DATABASE** laborator2

**create** **table** persoane

(

idPersoane **INT**,

Numele **VARCHAR**(20),

Virsta **INT**

)

**insert** **into** persoane

(idPersoane,Numele,Virsta)

**VALUES**

(1, 'Elvi', 19),

(2, 'Farouk', 19),

(3, 'Sam', 19),

(4, 'Tiany' , 19),

(5, 'Nadia' , 14),

(6, 'Chris' , 12),

(7, 'Kris' , 10),

(8, 'Bethany' , 16),

(9, 'Louis' , 17),

(10, 'Austin' , 22),

(11, 'Gabriel' , 21),

(12, 'Jessica' , 20),

(13, 'John' , 16),

(14, 'Alfred' , 19),

(15, 'Samantha' , 17),

(16, 'Craig' , 17);

**CREATE** **TABLE** `amici` (

`Idpersoana1` **INT**(10) **NOT** **NULL**,

`Idpersoana2` **INT**(10) **NOT** **NULL**

)

INSERT INTO `amici` (`Idpersoana1`, `Idpersoana2`) VALUES

(1, 2),

(1, 3),

(2, 4),

(2, 6),

(3, 9),

(4, 9),

(5, 8),

(6, 10),

(7, 5),

(7, 6),

(8, 7),

(9, 11),

(10, 15),

(12, 9),

(13, 6);

**CREATE** **TABLE** `rude` (

`Idpersoana1` **INT**(10) **NOT** **NULL**,

`Idpersoana2` **INT**(10) **NOT** **NULL**

)

1. ***Formulați interogări pentru fiecare sarcină.***
2. Lista prietenelor Elvirei.

**SELECT** persoane.\*

**FROM** amici

**INNER** **JOIN** persoane **ON** amici.Idpersoana2 = persoane.idPersoane

**WHERE** Idpersoana1 = 1

1. Lista persoanelor cu prietenii mai mici ca ele.

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

**FROM** amici **INNER** **JOIN** persoane **AS** p1

**ON** amici.idpersoana1 = p1.idPersoana

**INNER** **JOIN** persoane **AS** p2

**ON** amici.idpersoana2 = p2.idPersoana

**WHERE** p1.Virsta <> p2.Virsta;

1. Lista numelor prietenilor de aceeași vârstă. Rezultatele vor fi ordonate după două criterii:

* După ani;
* După nume pentru fiecare vârstă.

**SELECT** persoane.idPersoane **AS** idFriend1, persoane.Numele **AS** Friend1, persoane.Virsta **AS** age1,

friend.idPersoane **AS** idFriend2, friend.Numele **AS** Friend2, friend.Virsta **AS** age2

**FROM** amici

**INNER** **JOIN** persoane **AS** friend **ON** amici.Idpersoana2 = friend.idPersoane

**INNER** **JOIN** persoane **ON** amici.Idpersoana1 = persoane.idPersoane

**WHERE** persoane.Virsta = friend.Virsta

**ORDER** **BY** persoane.Virsta **desc**, persoane.Numele **desc**

*-- dupa id*

*-- SELECT persoane.idPersoane AS idFriend1, friend.idPersoane AS idFriend2*

*-- FROM amici*

*-- INNER JOIN persoane AS friend ON amici.Idpersoana2 = friend.idPersoane*

*-- INNER JOIN persoane ON amici.Idpersoana1 = persoane.idPersoane*

1. Lista persoanelor ce au cel puțin doi membri ai familiei.

**CREATE VIEW persoane\_rude AS**

**-- SELECT p1.numele, COUNT(idpersoana1) AS SUM1**

**-- FROM rude**

**-- INNER JOIN persoane AS p1 ON rude.idpersoana1 = p1.idPersoane**

**-- INNER JOIN persoane AS p2 ON rude.idpersoana2 = p2.idPersoane**

**-- GROUP BY p1.numele**

**-- UNION**

**-- SELECT p2.numele, COUNT(idpersoana2) AS SUM2**

**-- FROM rude**

**-- INNER JOIN persoane AS p1 ON rude.idpersoana1 = p1.idPersoane**

**-- INNER JOIN persoane AS p2 ON rude.idpersoana2 = p2.idPersoane**

**-- GROUP BY p2.numele**

**--**

**-- Lista persoanelor ce au cel puțin doi prieteni 2 membri ai familiei**

**-- SELECT \***

**-- from persoane\_rude**

**-- SELECT numele, SUM(SUM1) AS nr**

**-- FROM persoane\_rude**

**-- GROUP BY numele**

**-- HAVING nr>2**

1. Lista persoanelor ce nu au familie.

**SELECT** \*

**FROM** persoane **AS** p1

**LEFT** **JOIN** rude **AS** r1 **ON** p1.idPersoane = r1.Idpersoana1

**LEFT** **JOIN** rude **AS** r2 **ON** p1.idPersoane = r2.Idpersoana2

**WHERE** r1.Idpersoana1 **IS** **NULL** **AND** r1.Idpersoana2 **IS** **NULL** **AND** r2.Idpersoana1 **IS** **NULL** **AND** r2.Idpersoana2 **IS** **NULL**

1. Lista prietenilor împreună cu membrii familiei ce nu sunt prieteni.

**SELECT** a.\*, p.Numele, p2.Numele, r.\*

**FROM** amici **AS** a

**INNER** **JOIN** persoane **AS** p

**ON** a.Idpersoana1 = p.idPersoane

**INNER** **JOIN** persoane **AS** p2

**ON** a.Idpersoana2 = p2.idPersoane

**left** **JOIN** rude **AS** r

**ON** r.Idpersoana1 = a.Idpersoana1 **OR** r.Idpersoana2 = a.Idpersoana2

**WHERE** (r.Idpersoana1, r.Idpersoana2) != **all** (**SELECT** \* **FROM** amici)

1. Diferența între numărul persoanelor în total și numărul persoanelor ce au atins majoratul.

**select** **count**(idPersoane) **as** Total, (**select** **count**(idPersoane) **from** persoane **where** Virsta >= 18) **as** Majoratul,

**count**(idPersoane) -(**select** **count**(idPersoane) **from** persoane **where** Virsta >= 18) **as** Diferența

**from** persoane

1. Lista persoanelor care au cel puțin un prieten și cel puțin un membru al familiei.

**select** persoane.Numele

**from** laborator2.persoane

**inner** **join** laborator2.amici **on** persoane.idPersoane = amici.Idpersoana1

**left** **join** laborator2.rude **on** persoane.idPersoane = rude.Idpersoana1

**where** rude.Idpersoana1 **is** **not** **null**

**union**

**select** persoane.Numele

**from** laborator2.persoane

**inner** **join** laborator2.amici **on** persoane.idPersoane = amici.Idpersoana2

**left** **join** laborator2.rude **on** persoane.idPersoane = rude.Idpersoana2

**where** rude.Idpersoana2 **is** **not** **null**

1. Cifra medie de prieteni per persoană.

**SELECT** **COUNT**(idPersoana) / **COUNT**(**DISTINCT** persoane.idPersoana) **AS** Prieteni\_per\_persoana

**FROM** persoane

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

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

1. Numărul persoanelor ce sunt prieteni sau prietenii prietenilor lui Tiffany.

**SELECT** **COUNT**(p.idPersoana) **AS** Nr\_persoane

**FROM** persoane

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

**OR** persoane.idPersoana = a.idpersoana2

**INNER** **JOIN** persoane p **ON** (p.idPersoana = a.idpersoana1

**OR** p.idPersoana = a.idpersoana2)

**AND** p.idPersoana <> persoane.idPersoana

**INNER** **JOIN** amici a2 **ON** p.idPersoana = a2.idpersoana1

**OR** p.idPersoana = a2.idpersoana2

**WHERE** persoane.Numele = 'Tiany';

1. Numele și vârsta persoanelor cu cel mai mare număr de prieteni.

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

**FROM** persoane

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

**OR** persoane.idPersoana = a.idpersoana2

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

**HAVING** amici = (

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

**FROM** persoane

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

**OR** persoane.idPersoana = a.idpersoana2

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

**ORDER** **BY** amici **DESC**

**LIMIT** 1);

1. Pentru fiecare persoană, lista membrilor familiei sale. Dacă A este membru al familiei B și C este membru al familiei B, atunci C trebuie să apară în lista membrilor familiei A.

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

rude.idpersoana2 **AS** B,

**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** C

**FROM** rude

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

**HAVING** C **IS** **NOT** **NULL**;

1. Lista a trei persoane diferite, de vârstă diferită, astfel în cât prima e cea mai în vârstă, a doua e mai mică și a treia e cea mai tânără. Lista interogărilor va conține numele persoanelor și ID – ul.

*-- SELECT idpersoane , numele FROM persoane WHERE virsta = (SELECT MAX(virsta )FROM persoane )*

*-- OR virsta = (SELECT MIN(virsta) FROM persoane )*

*-- ORDER BY virsta DESC*

1. Lista persoanelor ce au un membru de familie, dar nu au nici un prieten.

*-- SELECT persoane .\* FROM persoane INNER JOIN amici AS p1*

*-- ON persoane.idpersoane = p1.idPersoana1*

*-- INNER JOIN rude AS p2 ON (persoane.idpersoane =p1.idPersoana1 ||(persoane.idPersoane = p1.idPersoana2) )*

*-- WHERE p1.idPersoana2 IS NULL*

*-- UNION*

*-- SELECT persoane.\**

*-- FROM persoane INNER JOIN rude AS p1*

*--*