



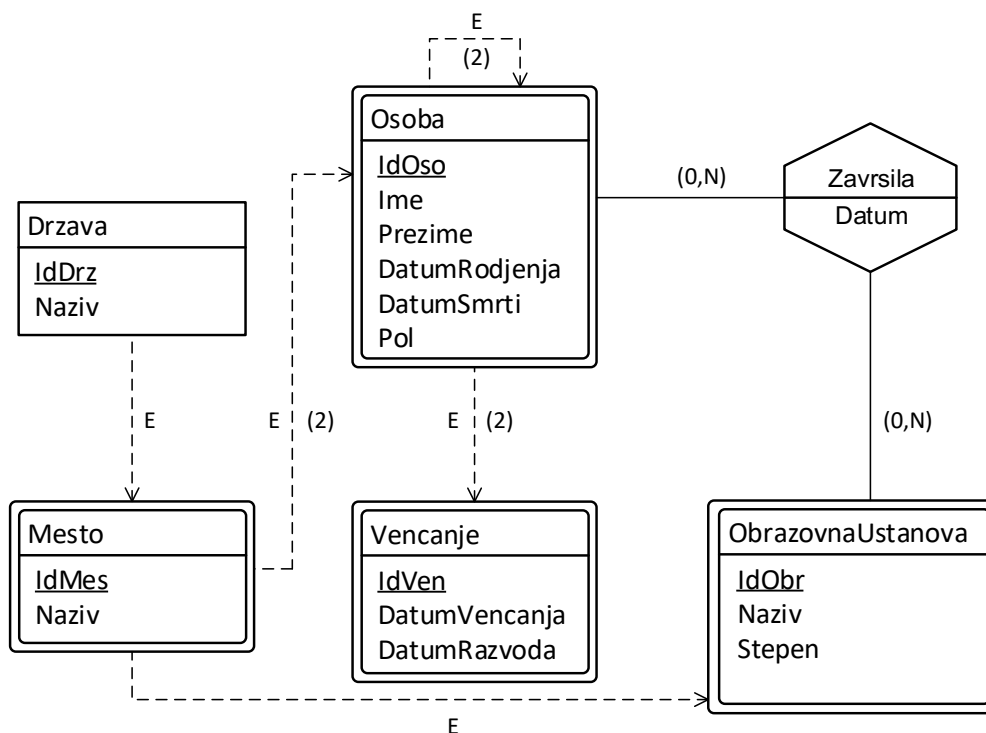
## Базе података 1

(13C112БП1)

- јануарски испитни рок -

### Група А

Посматра се база података за евидентирање родбинских односа и степена образовања. У овој бази се прате особе које поседују име, презиме, датум рођења и датум смрти, као и место рођења и место смрти. За особу се памте његови родитељи (отац и мајка) као и пол. Особе могу бити венчане и том приликом се прати датум венчања. Након венчања могу, а не морају, да се разведу, и том приликом се памти датум развода. Особа може одређеног датума завршити своје образовање у некој од образовних установа где се за сваку памти назив, степен и место у којем се налази. Степен образовне установе може бити: 1 - "основно образовање", 2 - "обучавање за рад", 3 - "средње стручно образовање у трајању од три године", 4 - "средње образовање у трајању од 4 године.", 5 - "мајсторско и специјалистичко образовање", 6 - "основне академске студије", 7 - "мастер академске студије", 8 - "докторске студије". Место има свој назив и државу у којој се налази. Сматрати да особа не може бити у више од једног брака у једном тренутку.



У наставку је дата релациона шема посматраног дела базе податка.

Drzava(IdDrz, Naziv)

- |       |   |
|-------|---|
| IdDrz | - цео број, идентификује венчање, аутоматско додељивање наредног идентификатора |
| Naziv | - низ до 50 знакова, обавезно   |

Mesto(IdMes, Naziv, IdDrz)

- |       |   |
|-------|---|
| IdMes | - цео број, идентификује венчање, аутоматско додељивање наредног идентификатора |
| Naziv | - низ до 50 знакова, обавезно   |
| IdDrz | - страни кључ (табела Drzava), обавезно   |

Osoba(IdOso, Ime, Prezime, IdOso1, IdOso2, DatumRodjenja, DatumSmrti, IdMesRodjenja, IdMesSmrti, Pol)

- |               |   |
|---------------|---|
| IdOso         | - цео број, идентификује особу, аутоматско додељивање наредног идентификатора |
| Ime           | - низ до 50 знакова, обавезно   |
| Prezime       | - низ до 50 знакова, обавезно   |
| IdOso1        | - страни кључ (табела Osoba), први предак                                     |
| IdOso2        | - страни кључ (табела Osoba), други предак                                    |
| DatumRodjenja | - цео број, обавезно, датум у формату (ууууммдд)                              |
| DatumSmrti    | - цео број, датум у формату (ууууммдд)  |
| IdMesRodjenja | - страни кључ (табела Mesto), обавезно  |
| IdMesSmrti    | - страни кључ (табела Mesto)  |
| Pol           | - карактер (Z – ”женски”, M – ”мушки”, O – ”остало”)                          |

Vencanje(IdVen, IdOso1, IdOso2, DatumVencanja, DatumRazvoda)

- |               |   |
|---------------|---|
| IdVen         | - цео број, идентификује венчање, аутоматско додељивање наредног идентификатора |
| IdOso1        | - страни кључ (табела Osoba), обавезно  |
| IdOso2        | - страни кључ (табела Osoba), обавезно  |
| DatumVencanja | - цео број, обавезно, датум у формату (ууууммдд)                                |
| DatumRazvoda  | - цео број, датум у формату (ууууммдд)  |

ObrazovnaUstanova(IdObr, Stepen, Naziv, IdMes)

- |        |   |
|--------|---|
| IdObr  | - цео број, идентификује венчање, аутоматско додељивање наредног идентификатора |
| Stepen | - цео број, обавезно  |
| Naziv  | - низ до 50 знакова, обавезно   |
| IdMes  | - страни кључ (табела Mesto), обавезно  |

Напомена: Знањење атрибута Stepen: 1 - ”основно образовање”, 2 – ”обучавање за рад”, 3 – ”средње стручно образовање у трајању од три године”, 4 – ”средње образовање у трајању од 4 године”, 5 – ”мајсторско и специјалистичко образовање”, 6 – ”основне академске студије”, 7 – ”мастер академске студије”, 8 – ”докторске студије”.

Zavrsila(IdObr, Datum, IdOso)

- |       |  |
|-------|--|
| IdObr | - страни кључ (табела ObrazovnaUstanova), обавезно, део идентификације |
| Datum | - број, обавезно поље, датум у формату (ууууммдд)                      |
| IdOso | - страни кључ (табела Osoba), обавезно, део идентификације             |

Задатак 1 [4 поена]

Потребно је направити SQL упит који исписује све парове који су се венчали и нису се развели, при чему су особе имале различит број бракова пре тога. За ове парове је потребно исписати и број њихове заједничке деце. Сортирати по IdOso1 растуће, а затим по IdOso2 растуће.

Резултат дати у форми: IdOso1, Ime1, IdOso2, Ime2, BrDece

У Cactus-у користити таб: Zadatak 1

**Није дозвољено коришћење погледа.**

---

```
SELECT V.IdOso1, O1.Ime AS Ime1, V.IdOso2, O2.Ime AS Ime2,
       COALESCE(( SELECT COUNT(IdOso)
                 FROM Osoba O3
                 WHERE (V.IdOso1=O3.IdOso1 AND V.IdOso2=O3.IdOso2)
                      OR (V.IdOso1=O3.IdOso2 AND V.IdOso2=O3.IdOso1)),0) AS BrDece
FROM Vencanje V JOIN Osoba O1 ON (V.IdOso1=O1.IdOso)
JOIN Osoba O2 ON (V.IdOso2=O2.IdOso)
AND ( SELECT COUNT(*)
      FROM Vencanje V2
      WHERE (V2.IdOso1=V.IdOso1 OR V2.IdOso2=V.IdOso1)
      AND V2.DatumVencanja<V.DatumVencanja )
!= ( SELECT COUNT(*)
     FROM Vencanje V2
     WHERE (V2.IdOso1=V.IdOso2 OR V2.IdOso2=V.IdOso2)
     AND V2.DatumVencanja<V.DatumVencanja )
AND v.Datumrazvoda IS NULL
ORDER BY V.IdOso1, V.IdOso2
```

---

Задатак 2 [4 поена]

Потребно је направити SQL скрипту која ако постоји табела **Osoba** избацује табелу **Osoba** из шеме, а затим формира нову табелу **Osoba** која треба да има одговарајућу структуру и ограничења.

У Cactus-у користити таб: Zadatak 2

---

```
DROP TABLE IF EXISTS Osoba;
CREATE TABLE Osoba (
  IdOso          INTEGER PRIMARY KEY AUTOINCREMENT,
  Ime            VARCHAR(50) NOT NULL,
  Prezime       VARCHAR(50) NOT NULL,
  IdOso1         INTEGER REFERENCES Osoba(IdOso),
  IdOso2         INTEGER REFERENCES Osoba(IdOso),
  DatumRodjenja  INTEGER NOT NULL,
  DatumSmrti     INTEGER,
  IdMesRodjenja  INTEGER NOT NULL REFERENCES Mesto(IdMes),
  IdMesSmrti     INTEGER REFERENCES Mesto(IdMes),
  Pol            INTEGER NOT NULL CHECK (Pol IN ('Z','M','O'))
);
```

---

Задатак 3 [4 поена]

Потребно је направити SQL упит којим се одузима стечена диплома свима који су завршили неку образовну установу мање од годину дана након завршетка претходно завршене образовне установе.

Након ажурираних података, исписати све податке из табеле **Zavrsila**.

Резултат дати у форми: IdObr, Datum, IdOso

У Cactus-у користити таб: Zadatak 3

---

```
DELETE FROM Zavrsila
WHERE EXISTS (SELECT *
              FROM Zavrsila Z
              WHERE Z.Datum+10000>Zavrsila.Datum AND Z.IdOso=Zavrsila.IdOso
              AND Z.Datum<Zavrsila.Datum);
```

```
SELECT * FROM Zavrsila
```

---

Задатак 4 [4 поена]

Потребно је направити SQL упит који исписује све живе особе женског пола чије име има између 5 и 6 слова од којих је бар једно слово а. Сортирати по IdOso опадајуће.

Резултат дати у форми: IdOso, Ime, Prezime, DatumRodjenja, IdMesRodjenja

У Cactus-у користити таб: Zadatak 4

---

```
SELECT IdOso, Ime, Prezime, Datumrodjenja, IdMesRodjenja
FROM Osoba
WHERE Pol='Z' AND DatumSmrti IS NULL
AND Ime LIKE '_____%' AND Ime NOT LIKE '_____%' AND Ime LIKE '%a%'
ORDER BY IdOso DESC
```

---

Задатак 5 [5 поена]

Потребно је направити SQL упит који исписује особе које имају бар једног полубрата или полусестру. За те особе је потребно исписати и информације о броју рођених браћа, рођених сестара, полубраће и полусестара. Сматрати да су браћа мушког, а сестре женског пола. Сортирати по IdOso опадајуће.

Резултат дати у форми: IdOso, Ime, Prezime, BrBraca, BrSestara, BrPolubraca, BrPolusestara

У Сactus-у користити таб: Zadatak 5

**Није дозвољено коришћење погледа.**

---

```
SELECT IdOso, Ime, Prezime,
COALESCE ((SELECT COUNT(*)
            FROM Osoba O2
            WHERE ((O2.IdOso1=O1.IdOso1 AND O2.IdOso2=O1.IdOso2) OR (O2.IdOso1=O1.IdOso2
AND O2.IdOso2=O1.IdOso1))
            AND O2.Pol='M' AND O1.IdOso!=O2.IdOso),0) AS BrBraca,
COALESCE ((SELECT COUNT(*)
            FROM Osoba O2
            WHERE ((O2.IdOso1=O1.IdOso1 AND O2.IdOso2=O1.IdOso2) OR (O2.IdOso1=O1.IdOso2
AND O2.IdOso2=O1.IdOso1))
            AND O2.Pol='Z' AND O1.IdOso!=O2.IdOso),0) AS BrSestara,
COALESCE ((SELECT COUNT(*)
            FROM Osoba O2
            WHERE ((O2.IdOso1=O1.IdOso1 AND O2.IdOso2!=O1.IdOso2)
                OR (O2.IdOso1=O1.IdOso2 AND O2.IdOso2!=O1.IdOso1)
                OR (O2.IdOso1!=O1.IdOso1 AND O2.IdOso2=O1.IdOso2)
                OR (O2.IdOso1!=O1.IdOso2 AND O2.IdOso2=O1.IdOso1))
            AND O2.Pol='M' AND O1.IdOso!=O2.IdOso),0) AS BrPolubraca,
COALESCE ((SELECT COUNT(*)
            FROM Osoba O2
            WHERE ((O2.IdOso1=O1.IdOso1 AND O2.IdOso2!=O1.IdOso2)
                OR (O2.IdOso1=O1.IdOso2 AND O2.IdOso2!=O1.IdOso1)
                OR (O2.IdOso1!=O1.IdOso1 AND O2.IdOso2=O1.IdOso2)
                OR (O2.IdOso1!=O1.IdOso2 AND O2.IdOso2=O1.IdOso1))
            AND O2.Pol='Z' AND O1.IdOso!=O2.IdOso),0) AS BrPolusestara
FROM Osoba O1
WHERE BrPolusestara + BrPolubraca >0
ORDER BY IdOso DESC
```

---

Задатак 6 [5 поена]

Потребно је направити SQL упит који дохвата све особе којима су сви преци из претходне две генерације рођени у истом месту у коме је рођена и та особа. Није потребно исписивати особе којима је непознат барем један предак у претходне две генерације. Сортирати по IdMes опадајуће, а затим по IdOso растуће.

Резултат дати у форми: IdOso, Ime i Prezime, IdMes, Naziv Mesta

У Сactus-у користити таб: Zadatak 6

**Није дозвољено коришћење погледа.**

---

```
SELECT IdOso, Ime || ' ' || Prezime AS "Ime i Prezime", IdMes, Naziv AS "Naziv Mesta"
FROM Osoba O1 JOIN Mesto M ON(IdMesRodjenja=IdMes)
WHERE 4= (SELECT COUNT(*)
          FROM Osoba O2 JOIN Osoba O3 ON (O3.IdOso=O2.IdOso1 OR O3.IdOso=O2.IdOso2)
          WHERE (O2.IdOso=O1.IdOso1 OR O2.IdOso=O1.IdOso2)
          AND O1.IdMesRodjenja=O2.IdMesRodjenja AND O2.IdMesRodjenja=O3.IdMesRodjenja)
ORDER BY IdMes DESC, IdOso
```

---

Задатак 7 [5 поена]

Потребно је направити SQL упит који дохвата све особе уз информације о установи у којој су завршили свој највиши степен образовања. Уколико су завршили исти степен образовања у више установа, потребно је исписати информације о свим таквим установама. Особе које ништа нису завршиле је потребно исписати и доделити им степен 0 (НЕПОЗНАТО). Степен обрзовања је потребно исписати у текстуалном облику. Сортирати по Stepен опадајуће, а затим по IdOso опадајуће.

Резултат дати у форми: Stepен obrazovanja, IdObr, Naziv, IdOso, Ime, Prezime

У Сactus-у користити таб: Zadatak 7

**Није дозвољено коришћење погледа.**

---

```
SELECT CASE Stepен
        WHEN 1 THEN 'osnovno obrazovanje'
        WHEN 2 THEN 'obucavanje za rad'
        WHEN 3 THEN 'srednje strucno obrazovanje u trajanju od tri godine'
        WHEN 4 THEN 'srednje obrazovanje u trajanju od 4 godine'
        WHEN 5 THEN 'majstorsko i specijalisticko obrazovanje'
        WHEN 6 THEN 'osnovne akademske studije'
        WHEN 7 THEN 'master akademske studije'
        WHEN 8 THEN 'doktorske studije'
        ELSE 'NEPOZNATO'
END AS "Stepен obrazovanja", IdObr, Naziv, IdOso, Ime, Prezime
FROM Osoba O LEFT JOIN Zavrsila Z USING (IdOso) LEFT JOIN ObrazovnaUstanova OU
USING(IdObr)
WHERE (Stepен = ( SELECT MAX (OU1.Stepен)
                  FROM Zavrsila Z1 JOIN ObrazovnaUstanova OU1 USING(IdObr)
                  WHERE Z1.IdOso=O.IdOso ) OR Stepен IS NULL)
ORDER BY Stepен DESC, IdOso DESC
```

---

Задатак 8 [6 поена]

Потребно је направити SQL упит који дохвата све сурдепачкиње најстарије особе рођене у 21. веку. Сматрати да постоји само једна најстарија особа рођена у 21. веку. Сортирати по IdOso растуће.

Резултат дати у форми: IdOso, Ime, Prezime, DatumRodjenja

У Сactus-у користити таб: Zadatak 8

**Није дозвољено коришћење погледа.**

---

```
WITH RECURSIVE PreciDoPardjupana AS
(
SELECT 0 AS Koleno, Osoba.* FROM Osoba
WHERE DatumRodjenja = (
    SELECT MIN(DatumRodjenja)
    FROM Osoba
    WHERE DatumRodjenja >= 20010101
)
UNION
SELECT PreciDoPardjupana.Koleno+1, Osoba.*
FROM PreciDoPardjupana, Osoba
WHERE Osoba.IdOso = PreciDoPardjupana.IdOso1
    OR Osoba.IdOso = PreciDoPardjupana.IdOso2
    AND PreciDoPardjupana.Koleno < 11
)
SELECT IdOso, Ime, Prezime, DatumRodjenja
FROM PreciDoPardjupana
WHERE Koleno=11 AND Pol='Z'
ORDER BY IdOso
```

---

Задатак 9 [6 поена]

Потребно је направити SQL упит који дохвата све особе које су током образовног периода тачно два пута промениле место привременог боравка. Сматра се да су у току школовања живели у месту у којем се налазили образовна установа коју су у том тренутку похађали и завршили. Такође, сматрати да особа није могла да заврши више образовних установа истог датума. Сортирати по IdOso растуће.

Резултат дати у форми: IdOso, Ime, Prezime, DatumRodjenja, DatumSmrti

У Sactus-у користити таб: Zadatak 9

**Није дозвољено коришћење погледа.**

---

```
WITH T AS (  
    SELECT IdOso, IdMes, Datum  
    FROM Zavrsila NATURAL JOIN ObrazovnaUstanova  
)  
T1 AS (  
    SELECT IdOso, IdMes, ( SELECT COUNT(*)  
        FROM T t1  
        WHERE t1.Datum<=t.Datum AND t1.IdOso=t.IdOso) AS RB  
    FROM T t  
)  
SELECT IdOso, Ime, Prezime, DatumRodjenja, DatumSmrti  
FROM Osoba  
WHERE IdOso IN (  
    SELECT IdOso  
    FROM T1 y  
    GROUP BY IdOso  
    HAVING SUM (CASE WHEN EXISTS (  
        SELECT * FROM T1 x WHERE x.RB=y.RB+1 AND x.IdMes!=y.IdMes AND  
x.IdOso=y.IdOso  
    ) THEN 1 ELSE 0 END )=2  
)  
ORDER BY IdOso
```

---



Задатак 10 [7 поена]

Потребно је направити SQL упит који дохвата све рођаке професора База података 1, којима су сви потомци живи. Рођаци су сви они који имају бар једног заједничког претка. Сматрати да у систему постоји само једна особа која се зове као и професор. Сортирати по IdOso растуће.

Резултат дати у форми: IdOso, Ime, Prezime

У Сactus-у користити таб: Zadatak 10

**Није дозвољено коришћење погледа.**

---

WITH RECURSIVE

PreciProf AS (

SELECT IdOso, IdOso1, IdOso2 FROM Osoba WHERE Ime='Miroslav' AND Prezime='Bojovic'

UNION

SELECT Osoba.IdOso, Osoba.IdOso1, Osoba.IdOso2

FROM Osoba, PreciProf

WHERE PreciProf.IdOso1 = Osoba.IdOso OR PreciProf.IdOso2 = Osoba.IdOso

),

Rodjaci AS (

SELECT IdOso FROM PreciProf

UNION

SELECT Osoba.IdOso

FROM Osoba, Rodjaci

WHERE Osoba.IdOso1 = Rodjaci.IdOso OR Osoba.IdOso2=Rodjaci.IdOso

),

PreciMrtvihRodjaka AS (

SELECT Rodjaci.IdOso, Osoba.IdOso1, Osoba.IdOso2 FROM Rodjaci, Osoba

WHERE Rodjaci.IdOso=Osoba.IdOso AND Osoba.DatumSmrti IS NOT NULL

UNION

SELECT Osoba.IdOso, Osoba.IdOso1, Osoba.IdOso2

FROM Osoba, PreciMrtvihRodjaka

WHERE PreciMrtvihRodjaka.IdOso1 = Osoba.IdOso OR PreciMrtvihRodjaka.IdOso2 =

Osoba.IdOso

)

SELECT IdOso, Ime, Prezime

FROM Osoba

WHERE IdOso IN (

SELECT IdOso FROM Rodjaci

EXCEPT

SELECT IdOso FROM PreciMrtvihRodjaka

)

ORDER BY IdOso

---