



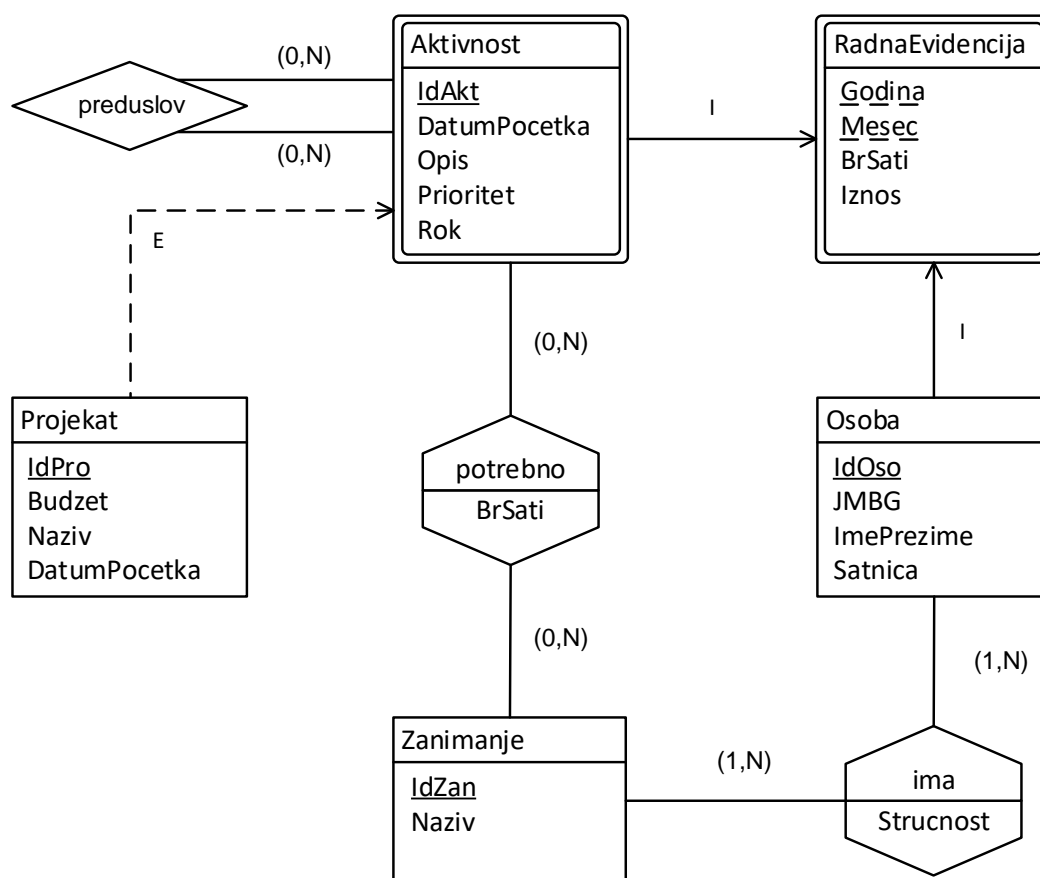
Базе података 1 (13Е113БП1)

- јануарски испитни рок -

Група Ц

Посматра се база софтвера за управљањем пројектима. За пројекат се дефинише буџет, назив, датум почетка, као и активности за његову реализацију. Активност карактеришу датум почетка рада активности, опис, приоритет (1-10, где је 1 највиши, а 10 најнижи приоритет), планирани рок за завршетак активности. Датум почетка активности може да буде непознат у тренутку креирања активности, а у тренутку почетка рада на активности овај податак се дефинише. За активност се прати колико је сати преостало ког занимања (прати се назив занимања који је јединствен) да би се завршила нека активност. Уколико не постоји ни један сат занимања за завршетак активности, онда се сматра да је та активност завршена. Да би нека активност уопште могла да крене у реализацију, потребно је да се оствари предуслов - да се све активности од које зависи активност буду завршене. У систему се прате подаци о особама, и то ЈМБГ, име и презиме, као и сатница. Особа може да има више занимања и том приликом се прати колика му је стручност. Ниво стручности од најнижег ка највишем је: јуниор, медиор, сениор, експерт. Особа у неком тренутку уноси у радну евиденцију колико је сати и за који износ радила у којем месецу у години на којој активности.

У наставку је дата релациона шема посматране базе података.



Projekat (IdPro, Budzet, Naziv, DatumPocetka)

| | |
|--------------|--|
| IdPro | - цео број, идентификује пројекат, аутоматско додељивање наредног идентификатора |
| Budzet | - реалан број, вредност већа или једнака 0, обавезно |
| Naziv | - низ до 50 знакова, јединствено, обавезно |
| DatumPocetka | - низ од тачно 11 знакова, датум у формату: <i>dd.mm.уууу</i> . |

Aktivnost (IdAkt, DatumPocetka, Opis, Prioritet, Rok, IdPro)

| | |
|--------------|---|
| IdAkt | - цео број, идентификује активност, аутоматско додељивање наредног идентификатора |
| DatumPocetka | - низ од тачно 11 знакова, датум у формату: <i>dd.mm.уууу</i> . |
| Opis | - низ до 100 знакова, обавезно |
| Prioritet | - цео број, вредност већа или једнака 1 и мања или једнака 10 |
| Rok | - низ од тачно 11 знакова, датум у формату: <i>dd.mm.уууу.</i> , обавезно |
| IdPro | - страни кључ (табела Projekat), обавезно |

Zanimanje (IdZan, Naziv)

| | |
|-------|--|
| IdZan | - цео број, идентификује занимање, аутоматско додељивање наредног идентификатора |
| Naziv | - низ до 50 знакова, јединствено, обавезно |

Osoba (IdOso, JMBG, Ime, Satnica)

| | |
|------------|---|
| IdOso | - цео број, идентификује особу, аутоматско додељивање наредног идентификатора |
| JMBG | - низ од тачно 13 знакова, јединствено, обавезно |
| ImePrezime | - низ до 100 знакова, обавезно |
| Satnica | - реалан број, вредност већа или једнака 0, обавезно |

RadnaEvidencija (Godina, Mesec, IdOso, BrSati, Iznos, IdAkt)

| | |
|--------|--|
| Godina | - цео број, обавезно, део примарног кључа |
| Mesec | - цео број, обавезно, део примарног кључа, вредност већа или једнака 1 и мања или једнака 12 |
| IdOso | - страни кључ (табела Osoba), обавезно, део примарног кључа |
| IdAkt | - страни кључ (табела Aktivnost), обавезно, део примарног кључа |
| BrSati | - цео број, вредност већа од 0, обавезно |
| Iznos | - реалан број, вредност већа или једнака 0, обавезно |

preduslov (IdAkt, IdAktPred)

| | |
|-----------|---|
| IdAkt | - страни кључ (табела Aktivnost), обавезно, део примарног кључа |
| IdAktPred | - страни кључ (табела Aktivnost), обавезно, део примарног кључа |

Напомена:

IdAktPred је предуслов за почетак активности са идентификатором IdAkt.

potrebno (IdAkt, IdZan, BrSati)

| | |
|--------|---|
| IdAkt | - страни кључ (табела Aktivnost), обавезно, део примарног кључа |
| IdZan | - страни кључ (табела Zanimanje), обавезно, део примарног кључа |
| BrSati | - цео број, вредност већа од 0, обавезно |

ima (IdZan, IdOso, Strucnost)

| | |
|-----------|---|
| IdZan | - страни кључ (табела Zanimanje), обавезно, део примарног кључа |
| IdOso | - страни кључ (табела Osoba), обавезно, део примарног кључа |
| Strucnost | - тачно 1 знак, обавезно поље, могуће вредности су: 'j' (јуниор), 'm' (медиор), 's' (сениор), 'e' (експерт) |

Задатак 1 [4 поена]

Потребно је направити SQL упит који приказује све активности које директно (не транзитивно) не зависе од других активности које су почеле са радом. Резултат сортирати по IdAkt растуће.

Резултат дати у форми: IdAkt, Opis
У Cactus-у користити таб: Zadatak 1

```
SELECT IdAkt, Opis
FROM Aktivnost
WHERE IdAkt NOT IN (
    SELECT p.IdAkt
    FROM preduslov p JOIN Aktivnost a ON(p.IdAktPred=a.IdAkt)
    WHERE DatumPocetka IS NOT NULL)
ORDER BY IdAkt
```

Задатак 2 [4 поена]

Потребно је направити SQL упит који приказује за сваку особу колика му је просечна зарада. Уколико особа нема ни један запис у радног евиденцији, треба исписати NULL. Резултат сортирати по просечној заради растуће, а затим по IdOso опадајуће.

Резултат дати у форми: IdOso, ImePrezime, ProsecnaZarada
У Cactus-у користити таб: Zadatak 2

```
SELECT IdOso, ImePrezime, (
    SELECT SUM(Iznos)/SUM(BrSati)
    FROM RadnaEvidencija re
    WHERE re.IdOso=o.IdOso
) AS ProsecnaZarada
FROM Osoba o
ORDER BY ProsecnaZarada, IdOso DESC
```

Задатак 3 [4 поена]

Потребно је направити SQL упит који за сваки пројекат исписује колико различитих занимања је потребно да би пројекат завршио реализацију. Резултат сортирати по броју занимања опадајуће, а затим по IdPro растуће.

Резултат дати у форми: IdPro, Naziv, BrojZanimanja
У Cactus-у користити таб: Zadatak 3

```
SELECT IdPro, Naziv, (
    SELECT COUNT( DISTINCT IdZan)
    FROM Aktivnost a NATURAL JOIN potrebno
    WHERE a.IdPro = p.IdPro
) AS BrojZanimanja
FROM Projekat p
ORDER BY BrojZanimanja DESC, IdPro
```

Задатак 4 [4 поена]

Потребно је направити SQL скрипту која брише из система све активности за које нема ни једна радна евиденција и да при томе да та активност не зависи ни од једне друге активности, као ни да ни једна друга активност не зависи од ње. Потребно је обрисати и све информације које се референцирају на такве активности. На крају је потребно исписати све преостале активности, као и потребна занимања за израду тих активности. Резултат сортирати по IdAkt опадајуће, а затим по IdZan растуће.

Резултат дати у форми: IdAkt, DatumPocetka, Opis, Prioritet, Rok, IdPro, IdZan, BrSati, Naziv
У Cactus-у користити таб: Zadatak 4

```
DELETE FROM potrebno
WHERE IdAkt NOT IN (SELECT IdAkt FROM RadnaEvidencija)
AND IdAkt NOT IN (SELECT IdAkt FROM preduslov)
AND IdAkt NOT IN (SELECT IdAktPred FROM preduslov);
```

```
DELETE FROM Aktivnost
WHERE IdAkt NOT IN (SELECT IdAkt FROM RadnaEvidencija)
AND IdAkt NOT IN (SELECT IdAkt FROM preduslov)
AND IdAkt NOT IN (SELECT IdAktPred FROM preduslov);
```

```
SELECT *
FROM Aktivnost NATURAL JOIN potrebno NATURAL JOIN Zanimanje
ORDER BY IdAkt DESC, IdZan
```

Задатак 5 [5 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 5

```
DROP TABLE IF EXISTS Osoba;
```

```
CREATE TABLE Osoba (
    IdOso          INTEGER PRIMARY KEY AUTOINCREMENT,
    JMBG           CHAR(11) UNIQUE NOT NULL,
    ImePrezime     VARCHAR(100) NOT NULL,
    Satnica        INTEGER CHECK (Satnica >= 0) NOT NULL
)
```

Задатак 6 [5 поена]

Потребно је написати SQL упит који исписује описно за сваки ниво стручности колико постоји особа са тим нивоом стручности:

- 'junior' – за ниво стручности 'j'
- 'medior' – за ниво стручности 'm'
- 'senior' – за ниво стручности 's'
- 'ekspert' – за ниво стручности 'e'

Ако особа има више занимања, та особа ће се рачунати онолико пута колико има занимања. Резултат сортирати тако да се прво исписује колико има особа стручност 'junior', затим колико има особа стручност 'medior', затим колико има особа стручност 'senior', и на колико има особа стручност 'ekspert'.

Резултат дати у форми: Nivo, Ukupno osoba

У Сactus-у користити таб: Zadatak 6

Није дозвољено коришћење погледа.

```
SELECT 'junior' AS Nivo, (SELECT COUNT(*) FROM ima WHERE Strucnost='j') AS "Ukupno osoba"
UNION ALL
SELECT 'medior', (SELECT COUNT(*) FROM ima WHERE Strucnost='m')
UNION ALL
SELECT 'senior', (SELECT COUNT(*) FROM ima WHERE Strucnost='s')
UNION ALL
SELECT 'ekspert', (SELECT COUNT(*) FROM ima WHERE Strucnost='e')
```

Задатак 7 [5 поена]

Потребно је написати SQL упит који исписује за сваку особу занимање (или занимања) у којима је та особа најстручнија. Резултат сортирати по стручности опадајуће (прво експерти, па сениори, па медиори, па јуниори), па по IdOso растуће, а затим по IdZan опадајуће.

Резултат дати у форми: IdOso, ImePrezime, IdZan, Naziv

У Сactus-у користити таб: Zadatak 7

Није дозвољено коришћење погледа.

```
WITH OsobaZanimanje (IdOso, ImePrezime, IdZan, Naziv, StrucnostNivo) AS
(
    SELECT IdOso, ImePrezime, IdZan, Naziv, CASE Strucnost
        WHEN 'e' THEN 4
        WHEN 's' THEN 3
        WHEN 'm' THEN 2
        ELSE 1
    END
    FROM Zanimanje NATURAL JOIN ima NATURAL JOIN Osoba
)
SELECT IdOso, ImePrezime, IdZan, Naziv
FROM OsobaZanimanje oz
WHERE StrucnostNivo = (SELECT MAX(StrucnostNivo) FROM OsobaZanimanje WHERE
IdOso=oz.IdOso)
ORDER BY StrucnostNivo DESC, IdOso, IdZan DESC
```

Задатак 8 [6 поена]

За пројекат са идентификатором 5 појавила се потреба за додатном активности са описом 'Testiranje', роком до 25.01.2022. са приоритетом 1 и датумом почетка који одговара последњем року од свих активности тог пројекта. Уколико не постоји ни једна друга активност пројекта, онда датум почетка треба да буде непознат. Потребно је написати SQL скрипту која додаје нов запис или ажурира тренутни запис (ако већ постоји активност са описом 'Testiranje') активности за наведени пројекат, а затим исписује све активности свих пројеката (које имају активност). Резултат сортирати по IdPro опадајуће, а затим по IdAkt опадајуће.

Резултат дати у форми: IdPro, DatumPocetka, Opis, Prioritet, Rok

У Сactus-у користити таб: Zadatak 8

Није дозвољено коришћење додатних структура – погледа или помоћних табела.

```
UPDATE Aktivnost
SET   Rok='25.01.2022.',
      Prioritet = 1,
      DatumPocetka = (
          SELECT Rok
          FROM Aktivnost
          WHERE (substr(Rok,7,4)|| substr(Rok,4,2) || substr(Rok,1,2))=
                (SELECT MAX(substr(Rok,7,4)|| substr(Rok,4,2) || substr(Rok,1,2)) FROM Aktivnost
WHERE IdPro=5) )
WHERE IdPro=5 AND Opis='Testiranje';

INSERT INTO Aktivnost (IdPro, Opis, Prioritet, Rok, DatumPocetka)
SELECT 5, 'Testiranje', 1, '25.01.2022.', (
    SELECT Rok
    FROM Aktivnost
    WHERE (substr(Rok,7,4)|| substr(Rok,4,2) || substr(Rok,1,2))=
          (SELECT MAX(substr(Rok,7,4)|| substr(Rok,4,2) || substr(Rok,1,2)) FROM Aktivnost WHERE
IdPro=5))
WHERE NOT EXISTS (
    SELECT *
    FROM Aktivnost
    WHERE IdPro=5 AND Opis='Testiranje'
);

SELECT IdPro, DatumPocetka, Opis, Prioritet, Rok
FROM Aktivnost
ORDER BY IdPro DESC, IdAkt DESC
```

Задатак 9 [6 поена]

Потребно је направити SQL упит који за сваку активност приказује статус те активности. Статус се рачуна на следећи начин (редом се гледају услови):

- Ако је датум почетка непознат, онда је статус 'Nije pocela'
- У супротном, ако не постоји више ни један потребан сат било ког занимања, онда је статус 'Zavrsena'
- У супротном, ако максимални број сати неког потребног занимања дода на датум почетка превазилази рок активности (сматрати да је у дану када активност почиње као и у дану када рок истиче дозвољен рад на тој активности), онда је статус 'Kasnjenje'.
- У супротном, статус је 'U izradi'

Резултат сортирати по IdAkt опадајуће.

Препоручена документација:

- [Date And Time Functions \(sqlite.org\)](https://www.sqlite.org/lang_datefunc.html)
- [Built-In Scalar SQL Functions \(sqlite.org\)](https://www.sqlite.org/lang_aggfunc.html)

Резултат дати у форми: IdAkt, Status

У Sactus-у користити таб: Zadatak 9

Није дозвољено коришћење погледа.

```
SELECT IdAkt, CASE
  WHEN DatumPocetka IS NULL THEN 'Nije pocela'
  WHEN NOT EXISTS (SELECT * FROM potrebno WHERE potrebno.IdAkt=a.IdAkt)
  THEN 'Zavrsena'
  WHEN (
    SELECT MAX(
      DATE(substr(a.DatumPocetka,7,4) || '-' || substr(a.DatumPocetka,4,2)
        || '-' || substr(a.DatumPocetka,1,2), '+' || (BrSati-1) || ' hours'))
    FROM potrebno
    WHERE potrebno.IdAkt=a.IdAkt
  ) >
    DATE(substr(a.Rok,7,4) || '-' || substr(a.Rok,4,2) || '-' || substr(a.Rok,1,2))
  THEN 'Kasnjenje'
  ELSE 'U izradi' END AS Status
FROM Aktivnost a
ORDER BY IdAkt
```

Задатак 10 [7 поена]

Потребно је направити SQL упит који за сваку активност на пројекту са називом 'Тajни пројекат' исписује колико минимално активности је потребно да буде реализовано како би та активност могла бити реализована. Резултат сортирати по IdAkt растуће.

Резултат дати у форми: IdAkt, Broj

У Сactus-у користити таб: Zadatak 10

Није дозвољено коришћење погледа.

```
WITH RECURSIVE AktivnostiUslovi (IdAktTajnogProjekta, IdAkt) AS (  
    SELECT IdAkt, IdAkt  
    FROM Aktivnost  
    WHERE IdPro = (SELECT MAX(IdPro) FROM Projekat WHERE Naziv='Tajni projekat')  
    UNION  
    SELECT IdAktTajnogProjekta, IdAktPred  
    FROM preduslov, AktivnostiUslovi  
    WHERE preduslov.IdAkt = AktivnostiUslovi.IdAkt  
)  
SELECT IdAktTajnogProjekta AS IdAkt, COUNT(IdAkt)-1 AS Broj  
FROM AktivnostiUslovi  
GROUP BY IdAktTajnogProjekta  
ORDER BY IdAktTajnogProjekta DESC
```
