



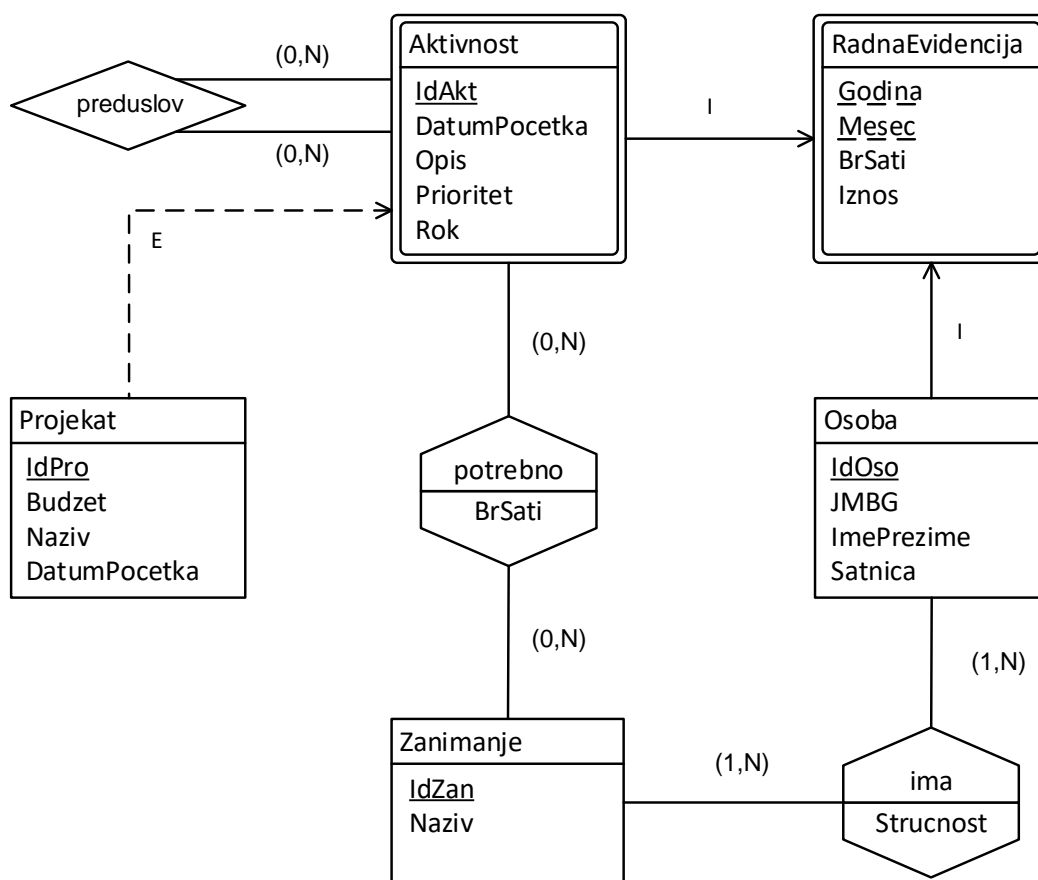
Базе података 1 (13С112БП1, 13Е113БП1)

- јануарски испитни рок -

Група А

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 чији је приоритет већи од 4. Резултат сортирати по IdAkt опадајуће.

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 prioritet>4)
ORDER BY IdAkt DESC
```

Задатак 2 [4 поена]

Потребно је направити SQL упит који приказује све пројекте на којима је "пробијен" буџет – укупан износ радних евиденција свих активности пројекта превазилази буџет тог пројекта. Резултат сортирати по Naziv растуће, а затим по IdPro опадајуће.

Резултат дати у форми: IdPro, Budzet, Naziv
У Cactus-у користити таб: Zadatak 2

```
SELECT IdPro, Budzet, Naziv
FROM Projekat p
WHERE Budzet < (
    SELECT COALESCE(SUM(Iznos),0)
    FROM RadnaEvidencija NATURAL JOIN Aktivnost a
    WHERE a.IdPro=p.IdPro
)
ORDER BY Naziv, IdPro DESC
```

Задатак 3 [4 поена]

Потребно је направити SQL упит који за сваку особу исписује на колико пројеката је била ангажована (особа је ангажована на пројекту ако има запис у радној евиденцији за било коју активност тог пројекта). Резултат сортирати по броју пројеката опадајуће, а затим по JMBG растуће.

Резултат дати у форми: JMBG, ImePrezime, BrojProjekata
У Cactus-у користити таб: Zadatak 3

```
SELECT JMBG, ImePrezime, (
    SELECT COUNT( DISTINCT IdPro)
    FROM Aktivnost NATURAL JOIN RadnaEvidencija re
    WHERE re.IdOso = o.IdOso
) AS BrojProjekata
FROM Osoba o
ORDER BY BrojProjekata DESC, JMBG
```

Задатак 4 [4 поена]

Потребно је направити SQL скрипту која брише из система све особе које немају ни један запис у радној евиденцији. Потребно је обрисати и све информације које се референцирају на такве особе. На крају је потребно исписати све преостале особе, као и њихову стручност. Резултат сортирати по JMBG опадајуће, а затим по IdZan растуће.

Резултат дати у форми: IdOso, JMBG, ImePrezime, Satnica, IdZan, Strucnost
У Cactus-у користити таб: Zadatak 4

```
DELETE FROM ima
WHERE IdOso NOT IN (SELECT IdOso FROM RadnaEvidencija);
```

```
DELETE FROM Osoba
WHERE IdOso NOT IN (SELECT IdOso FROM RadnaEvidencija);
```

```
SELECT *
FROM Osoba NATURAL JOIN ima
ORDER BY JMBG DESC, IdZan;
```

Задатак 5 [5 поена]

Потребно је направити SQL скрипту која ако постоји табела **Aktivnost** избацује табелу **Aktivnost** из шеме, а затим формира нову табелу **Aktivnost** која треба да има одговарајућу структуру и ограничења. За формат поља DatumPocetka и Rok проверавати само да ли се на одговарајућим позицијама налази карактер '.'.

У Cactus-у користити таб: Zadatak 5

```
DROP TABLE IF EXISTS Aktivnost;
```

```
CREATE TABLE Aktivnost (
    IdAkt      INTEGER PRIMARY KEY AUTOINCREMENT,
    DatumPocetka CHAR(11) CHECK (DatumPocetka LIKE '__.__.____.'),
    Opis       VARCHAR(100) NOT NULL,
    Prioritet  INTEGER CHECK (Prioritet >= 1 AND Prioritet <= 10),
    Rok        CHAR(11) CHECK (Rok LIKE '__.__.____.') NOT NULL,
    IdPro      INTEGER NOT NULL REFERENCES Projekat (IdPro)
)
```

Задатак 6 [5 поена]

Потребно је написати SQL упит који исписује описно по приоритетима колико има активности тог приоритета. Опис приоритета активности може да има вредност:

- 'Urgentno' – за активности чији је приоритет једнак 1
- 'Hitno' – за активности чији је приоритет између 2 и 4
- 'Redovno' – за активности чији је приоритет између 5 и 7
- 'Nizak prioritet' – за активности чији је приоритет између 8 и 10
- 'Nepoznato' – за активности чији је приоритет непознат

Резултат сортирати тако да се прво исписује колико има активности са статусом 'Urgentno', затим колико има активности са статусом 'Hitno', затим колико има активности са статусом 'Redovno', затим колико има активности са статусом 'Nizak prioritet' и на крају колико има активности са статусом 'Nepoznato'.

Резултат дати у форми: Prioritet, Broj aktivnosti

У Сцус-у користити таб: Zadatak 6

Није дозвољено коришћење погледа.

```
SELECT 'Urgentno' AS Prioritet,  
       (SELECT COUNT(*) FROM Aktivnost WHERE Prioritet=1) AS "Broj aktivnosti"  
UNION ALL  
SELECT 'Hitno', (SELECT COUNT(*) FROM Aktivnost WHERE Prioritet BETWEEN 2 AND 4)  
UNION ALL  
SELECT 'Redovno', (SELECT COUNT(*) FROM Aktivnost WHERE Prioritet BETWEEN 5 AND 7)  
UNION ALL  
SELECT 'Nizak prioritet',  
       (SELECT COUNT(*) FROM Aktivnost WHERE Prioritet BETWEEN 8 AND 10)  
UNION ALL  
SELECT 'Nepoznato', (SELECT COUNT(*) FROM Aktivnost WHERE Prioritet IS NULL)
```

Задатак 7 [5 поена]

Потребно је написати SQL упит који исписује за свако занимање особу (или особе) која је најстручнија за то занимање. Резултат сортирати по стручности опадајуће (прво експерти, па сениори, па медиори, па јуниори), па по IdZan растуће, а затим по IdOso опадајуће.

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

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 IdZan, Naziv, IdOso, ImePrezime
FROM OsobaZanimanje oz
WHERE StrucnostNivo =
    (SELECT MAX(StrucnostNivo) FROM OsobaZanimanje WHERE IdZan=oz.IdZan)
ORDER BY StrucnostNivo DESC, IdZan, IdOso DESC
```

Задатак 8 [6 поена]

Особа са идентификатором 5 треба да пријави на постојеће стање још 20 сати рада за јануар 2022. за активност са идентификатором 15. Потребно је написати SQL скрипту која додаје запис у радну евиденцију или ажурира (ако већ постоји) радну евиденцију, а затим исписује комплетну радну евиденцију за јануар 2022. Уколико запис већ постоји потребно је на постојећи износ додати нов износ срачунат на основу тренутне сатнице особе. Резултат сортирати по IdOso растуће, а затим IdAkt растуће.

Резултат дати у форми: Godina, Mesec, IdOso, IdAkt, BrSati, Iznos

У Сactus-у користити таб: Zadatak 8

Није дозвољено коришћење додатних структура – погледа или помоћних табела.

```
UPDATE RadnaEvidencija
SET    BrSati = BrSati + 20,
       Iznos = Iznos + 20 * ( SELECT MAX(Satnica) FROM Osoba WHERE IdOso = 5 )
WHERE Godina=2022 AND Mesec=1 AND IdOso=5 AND IdAkt=15;
```

```
INSERT INTO RadnaEvidencija (Godina, Mesec, IdOso, IdAkt, BrSati, Iznos)
SELECT 2022, 1, 5, 15, 20, 20 * Satnica
FROM Osoba
WHERE IdOso = 5 AND NOT EXISTS (
    SELECT *
    FROM RadnaEvidencija
    WHERE Godina=2022 AND Mesec=1 AND IdOso=5 AND IdAkt=15
);
```

```
SELECT *
FROM RadnaEvidencija
WHERE Godina=2022 AND Mesec=1
ORDER BY IdOso, IdAkt
```

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 растуће.

Препоручена документација:

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

Резултат дати у форми: IdAkt, Status

У Cactus-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 исписује који све пројекти су потребни да буду барем делимично реализовани како би се могао у потпуности реализовао пројекат са називом 'Тajни пројекат'. Резултат сортирати по IdPro опадајуће.

Резултат дати у форми: IdPro, Naziv

У Сactus-у користити таб: Zadatak 10

Није дозвољено коришћење погледа.

```
WITH RECURSIVE AktivnostiUslovi (IdAkt) AS (  
    SELECT IdAkt  
    FROM Aktivnost  
    WHERE IdPro = (SELECT MAX(IdPro) FROM Projekat WHERE Naziv='Tajni projekat')  
    UNION  
    SELECT IdAktPred  
    FROM preduslov, AktivnostiUslovi  
    WHERE preduslov.IdAkt = AktivnostiUslovi.IdAkt  
)  
SELECT DISTINCT IdPro, Naziv  
FROM Projekat JOIN Aktivnost USING (IdPro)  
WHERE IdAkt IN (SELECT IdAkt FROM AktivnostiUslovi)  
    AND IdPro != (SELECT MAX(IdPro) FROM Projekat WHERE Naziv='Tajni projekat')  
ORDER BY IdPro DESC
```
