



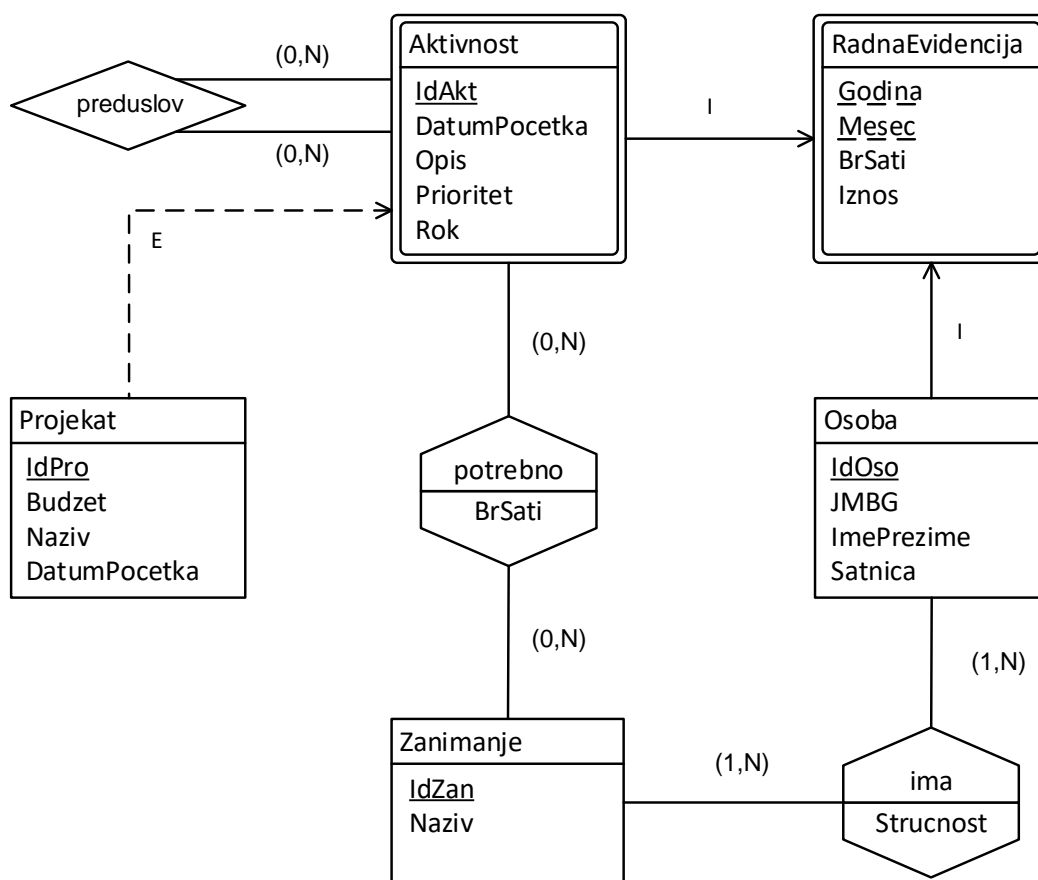
## Базе података 1 (13С112БП)

- јануарски испитни рок -

### Група Б

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) зависе од других активности чији је приоритет једнак 1. Резултат сортирати по IdAkt опадајуће.

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

---

```
SELECT IdAkt, Opis
FROM Aktivnost
WHERE IdAkt IN (
    SELECT p.IdAkt
    FROM preduslov p JOIN Aktivnost a ON(p.IdAktPred=a.IdAkt)
    WHERE Prioritet = 1)
ORDER BY IdAkt DESC
```

---

### Задатак 2 [4 поена]

Потребно је направити SQL упит који приказује информације о пројектима заједно са преосталим стањем буџета. Преостало стање буџета пројекта се рачуна као буџет умањен за укупан износ радних евиденција свих активности тог пројекта. Резултат сортирати по Naziv растуће, а затим по IdPro опадајуће.

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

---

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

---

### Задатак 3 [4 поена]

Потребно је направити SQL упит који за сваки пројекат исписује колико је особа било ангажовано на њему (особа је ангажована на пројекту ако има запис у радној евиденцији за било коју активност тог пројекта). Резултат сортирати по броју особа опадајуће, а затим по IdPro растуће.

Резултат дати у форми: IdPro, Naziv, BrojOsoba  
У Сactus-у користити таб: Zadatak 4

---

```
SELECT IdPro, Naziv, (
    SELECT COUNT( DISTINCT IdOso)
    FROM Aktivnost a NATURAL JOIN RadnaEvidencija
    WHERE a.IdPro = p.IdPro
) AS BrojOsoba
FROM Projekat p
ORDER BY BrojOsoba DESC, IdPro
```

---

#### Задатак 4 [4 поена]

Потребно је направити SQL скрипту која брише из система сва занимања за која ни једна активност нема потребу. Потребно је обрисати и све информације које се референцирају на таква занимања. На крају је потребно исписати све преостале особе, као и њихова занимања. Резултат сортирати по JMBG растуће, а затим по IdZan растуће.

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

---

```
DELETE FROM ima  
WHERE IdZan NOT IN (SELECT IdZan FROM potrebno);
```

```
DELETE FROM Zanimanje  
WHERE IdZan NOT IN (SELECT IdZan FROM potrebno);
```

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

---

#### Задатак 5 [5 поена]

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

У Сactus-у користити таб: Zadatak 5

---

```
DROP TABLE IF EXISTS RadnaEvidencija;
```

```
CREATE TABLE RadnaEvidencija (  
    Godina        INTEGER NOT NULL,  
    Mesec         INTEGER NOT NULL CHECK (Mesec >= 1 AND Mesec <= 12),  
    IdOso         INTEGER NOT NULL REFERENCES Osoba (IdOso),  
    IdAkt         INTEGER NOT NULL REFERENCES Aktivnost (IdAkt),  
    BrSati        INTEGER CHECK (BrSati > 0) NOT NULL,  
    Iznos         REAL CHECK (Iznos >= 0) NOT NULL,  
    PRIMARY KEY (Godina, Mesec, IdOso, IdAkt)  
)
```

---

Задатак 6 [5 поена]

Потребно је написати SQL упит који исписује по кварталима колики је укупан износ евидентиран у радним евиденцијама за годину 2021. Радне евиденције могу да буду заведене у кварталима:

- 'Q1' – за радне евиденције које се односе за јануар, фебруар и март
- 'Q2' – за радне евиденције које се односе за април, мај и јун
- 'Q3' – за радне евиденције које се односе за јул, август и септембар
- 'Q4' – за радне евиденције које се односе за октобар, новембар и децембар

Резултат сортирати тако да се прво исписује колики је укупан износ радних евиденција у првом кварталу (Q1), затим колики је укупан износ радних евиденција у другом кварталу (Q2), затим колики је укупан износ радних евиденција у трећем кварталу (Q3), и на крају колики је укупан износ радних евиденција у четвртом кварталу (Q4).

Резултат дати у форми: Kvartal, Ukupan iznos

У Сactus-у користити таб: Zadatak 6

**Није дозвољено коришћење погледа.**

---

```
SELECT 'Q1' AS Kvartal, (SELECT COALESCE(SUM(Iznos), 0) FROM RadnaEvidencija
    WHERE Godina=2021 AND Mesec BETWEEN 1 AND 3) AS "Ukupan iznos"
UNION ALL
SELECT 'Q2', (SELECT COALESCE(SUM(Iznos), 0) FROM RadnaEvidencija
    WHERE Godina=2021 AND Mesec BETWEEN 4 AND 6)
UNION ALL
SELECT 'Q3', (SELECT COALESCE(SUM(Iznos), 0) FROM RadnaEvidencija
    WHERE Godina=2021 AND Mesec BETWEEN 7 AND 9)
UNION ALL
SELECT 'Q4', (SELECT COALESCE(SUM(Iznos), 0) FROM RadnaEvidencija
    WHERE Godina=2021 AND Mesec BETWEEN 10 AND 12)
```

---

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 активност са идентификатором 15 појавила се потреба за додатна 24 сата за занимање са називом 'Tester'. Потребно је написати SQL скрипту која додаје нов запис или ажурира тренутни запис (ако већ постоји) за потребе наведене активности, а затим исписује комплетне потребе свих активности. Резултат сортирати по IdAkt опадајуће, а затим по IdZan растуће.

Резултат дати у форми: IdAkt, IdZan, BrSati

У Cactus-у користити таб: Zadatak 8

**Није дозвољено коришћење додатних структура – погледа или помоћних табела.**

---

UPDATE potrebno

SET BrSati = BrSati + 24

WHERE IdAkt=15 AND IdZan = (SELECT MAX(IdZan) FROM Zanimanje WHERE Naziv='Tester');

INSERT INTO potrebno (IdAkt, IdZan, BrSati)

SELECT 15, (SELECT MAX(IdZan) FROM Zanimanje WHERE Naziv='Tester'), 24

WHERE NOT EXISTS (

SELECT \*

FROM potrebno

WHERE IdAkt=15

AND IdZan = (SELECT MAX(IdZan) FROM Zanimanje WHERE Naziv='Tester')

);

SELECT \*

FROM potrebno

ORDER BY IdAkt DESC, IdZan

---

### 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/funcscalar.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
```

---