



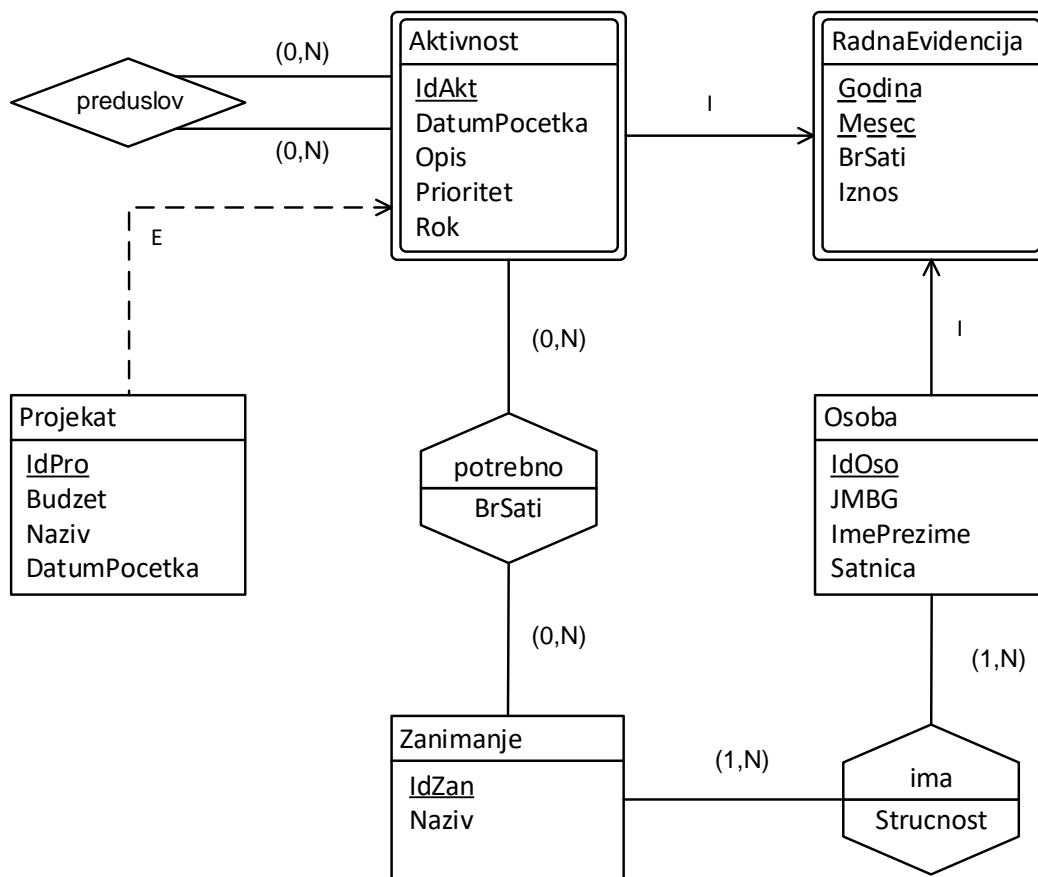
Базе података 1 (13С112БП1, 13Е113БП1)

- 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: *dd.mm.yyy*.

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: *dd.mm.yyy*.
- 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: *dd.mm.yyy*., 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 Сactus-у користити таб: Zadatak 1

```
SELECT IdAkt, Opis
FROM Aktivnosti
WHERE IdAkt < 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 Сactus-у користити таб: Zadatak 2

```
SELECT IdPro, Budzet, Naziv
FROM Projekti
WHERE (SELECT SUM(Price) FROM Aktivnosti
WHERE IdPro = Projekti.IdPro) > Budzet
ORDER BY Naziv ASC, 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 ЈМВГ растуће.

Резултат дати у форми: ЈМВГ, ИмеPrezime, БројProjekata
У Сactus-у користити таб: Zadatak 3

```
SELECT JMBG, ИмеPrezime, БројProjekata
FROM Osobe
WHERE (SELECT COUNT(DISTINCT IdPro) FROM Aktivnosti
WHERE IdOsoba = Osobe.IdOsoba) > 0
ORDER BY БројProjekata DESC, JMBG ASC
```

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 ЈМБГ опадајуће, а затим по IdZan растуће.

Резултат дати у форми: IdOso, ЈМБГ, ИмеPrezime, Satnica, IdZan, Strucnost
У Сactus-у користити таб: Zadatak 4

```
DELETE FROM Osoba
WHERE NOT EXISTS (SELECT * FROM RadnaEvidencija
WHERE Osoba.IdOso = RadnaEvidencija.IdOso)

DELETE FROM Informacije
WHERE Osoba.IdOso = Informacije.IdOso

DELETE FROM Strucnost
WHERE Osoba.IdOso = Strucnost.IdOso
```

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 DatumRosetka и Rok проверавати само да ли се на одговарајућим позицијама налази карактер '!'.
У Сactus-у користити таб: Zadatak 5

```
DROP TABLE IF EXISTS Aktivnost

CREATE TABLE Aktivnost (
    IdAktivnost INT(11) UNSIGNED NOT NULL AUTO_INCREMENT,
    DatumRosetka CHAR(10) CHECK (DatumRosetka LIKE '!!-!!-!!'),
    Rok VARCHAR(4) NOT NULL,
    Mesto VARCHAR(255) CHECK (Mesto != '' AND Mesto != '!'),
    Rok VARCHAR(4) CHECK (Rok != '' AND Rok != '!'),
    IdOso INT(11) UNSIGNED NOT NULL REFERENCES Osoba (IdOso)
)
```

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, Број активности

У Сactus-у користити таб: Zadatak 6

Није дозвољено коришћење погледа.

```
SELECT Urgentno AS Prioritet,
       COUNT(*) AS Broj_aktivnosti
FROM Aktivnosti WHERE Prioritet = 1
UNION ALL
SELECT Hitno AS Prioritet,
       COUNT(*) AS Broj_aktivnosti
FROM Aktivnosti WHERE Prioritet BETWEEN 2 AND 4
UNION ALL
SELECT Redovno AS Prioritet,
       COUNT(*) AS Broj_aktivnosti
FROM Aktivnosti WHERE Prioritet BETWEEN 5 AND 7
UNION ALL
SELECT Nizak AS Prioritet,
       COUNT(*) AS Broj_aktivnosti
FROM Aktivnosti WHERE Prioritet BETWEEN 8 AND 10
UNION ALL
SELECT Nepoznato AS Prioritet,
       COUNT(*) AS Broj_aktivnosti
FROM Aktivnosti 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 Oso AS (SELECT Naziv, IdOso, ImePrezime FROM Oso),
Zan AS (SELECT Naziv, IdZan, Strucnost FROM Zadatak_7)
SELECT Zan.Naziv, Zan.IdZan, Oso.IdOso, Oso.ImePrezime
FROM Zan NATURAL JOIN Oso
ORDER BY Strucnost DESC, IdZan ASC, 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: Година, Mesec, IdOso, IdAkt, BrSati, Iznos

У Сactus-у користити таб: Zadatak 8

Није дозвољено коришћење додатних структура – погледа или помоћних табела.

```
INSERT INTO radna_evidencija (Godina, Mesec, IdOso, IdAkt, BrSati, Iznos)
VALUES (2022, 1, 5, 15, 20, 0)
ON DUPLICATE KEY UPDATE Iznos = Iznos + 20 * (SELECT MAX(Satnica) FROM osoba WHERE IdOso = 5)

INSERT INTO radna_evidencija (Godina, Mesec, IdOso, IdAkt, BrSati, Iznos)
SELECT 2022, 1, IdOso, IdAkt, BrSati, Iznos
FROM radna_evidencija
WHERE IdOso = 5 AND NOT EXISTS
(
    SELECT 1
    FROM radna_evidencija
    WHERE Godina = 2022 AND Mesec = 1
    AND IdOso = IdOso AND IdAkt = IdAkt
)

SELECT *
FROM radna_evidencija
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 'Zavrшена'
- 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 \(sqlite.org\)](http://sqlite.org/date)
- [Built-In Scalar SQL Functions \(sqlite.org\)](http://sqlite.org/built-in-scalar-sql-functions)

Резултат дати у форми: IdAkt, Status

У Sactus-у користити таб: Zadatak 9

Није дозвољено коришћење погледа.

```
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 1;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 2;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 3;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 4;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 5;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 6;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 7;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 8;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 9;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 10;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 11;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 12;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 13;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 14;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 15;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 16;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 17;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 18;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 19;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 20;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 21;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 22;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 23;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 24;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 25;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 26;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 27;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 28;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 29;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 30;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 31;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 32;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 33;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 34;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 35;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 36;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 37;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 38;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 39;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 40;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 41;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 42;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 43;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 44;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 45;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 46;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 47;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 48;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 49;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 50;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 51;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 52;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 53;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 54;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 55;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 56;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 57;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 58;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 59;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 60;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 61;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 62;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 63;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 64;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 65;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 66;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 67;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 68;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 69;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 70;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 71;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 72;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 73;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 74;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 75;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 76;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 77;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 78;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 79;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 80;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 81;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 82;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 83;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 84;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 85;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 86;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 87;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 88;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 89;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 90;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 91;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 92;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 93;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 94;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 95;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 96;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 97;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 98;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 99;
SELECT IdAkt, Status FROM Zadatak_9 WHERE IdAkt = 100;
```

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 AktivnostiOsnovniIDAKI AS  
  SELECT IDAKI  
  FROM AktivnostiOsnovniIDAKI  
  WHERE IDPro = (SELECT IDPro FROM Projekat WHERE Naziv = 'Tajni projekat')  
  UNION ALL  
  SELECT IDAKIPro  
  FROM AktivnostiOsnovniIDAKI  
  WHERE IDPro = (SELECT IDPro FROM Projekat WHERE Naziv = 'Tajni projekat')  
  UNION ALL  
  SELECT IDAKIPro  
  FROM AktivnostiOsnovniIDAKI  
  WHERE IDPro = (SELECT IDPro FROM Projekat WHERE Naziv = 'Tajni projekat')  
  UNION ALL  
  SELECT IDAKIPro  
  FROM AktivnostiOsnovniIDAKI  
  WHERE IDPro = (SELECT IDPro FROM Projekat WHERE Naziv = 'Tajni projekat')  
  UNION ALL  
  SELECT IDAKIPro  
  FROM AktivnostiOsnovniIDAKI  
  WHERE IDPro = (SELECT IDPro FROM Projekat WHERE Naziv = 'Tajni projekat')  
  UNION ALL  
  SELECT IDAKIPro  
  FROM AktivnostiOsnovniIDAKI  
  WHERE IDPro = (SELECT IDPro FROM Projekat WHERE Naziv = 'Tajni projekat')  
  UNION ALL  
  SELECT IDAKIPro  
  FROM AktivnostiOsnovniIDAKI  
  WHERE IDPro = (SELECT IDPro FROM Projekat WHERE Naziv = 'Tajni projekat')  
  UNION ALL  
  SELECT IDAKIPro  
  FROM AktivnostiOsnovniIDAKI  
  WHERE IDPro = (SELECT IDPro FROM Projekat WHERE Naziv = 'Tajni projekat')
```