

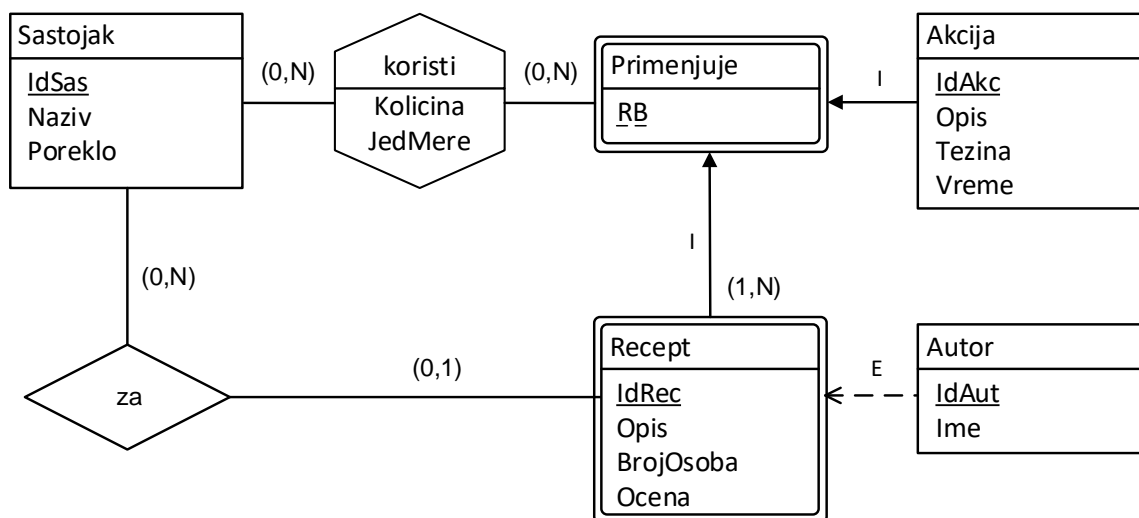


## Базе података 1 (13С112БП1, 13Е113БП1, 13Е113БП, 13Е112БП)

– фебруарски испитни рок –

Дат је део базе података за чување рецепата за кување. У бази се чувају аутори (име) и њихови рецепти. Рецепт има опис, за колико особа се припрема, као и реалну оцену која је од 0-5. Рецепти се састоје од низа акција које се примењују. Сматрати да сваки рецепт има бар једну акцију која користи неки састојак. За акције се прате опис, тежина (1-3), као и време дато у секундама. Редослед акција у рецепту се прати кроз поље RB (редни број). Сматрати да се адекватно попуњавају редни бројеви почевши од 1. Акције могу, али и не морају да користе састојке (може и више састојака да се користе за неку акцију). Том приликом се чувају информације о количини и јединици мере. Састојак има име и прати се његово порекло (животињско, биљно и остало). За прављење неког састојка може да постоји један или више рецепата.

У наставку је дата релациона шема посматране базе података.



Akcija (IdAkc, Opis, Tezina, Vreme)

|        |  |
|--------|--|
| IdAkc  | - цео број, идентификује акцију, аутоматско додељивање наредног идентификатора |
| Opis   | - низ до 300 знакова, обавезно, јединствено                                    |
| Tezina | - цео број, обавезно, већи или једнак 1 и мањи или једнак 3                    |
| Vreme  | - цео број, већи од 0  |

Autor (IdAut, Ime)

|       |  |
|-------|--|
| IdAut | - цео број, идентификује аутора, аутоматско додељивање наредног идентификатора |
| Ime   | - низ до 50 знакова, обавезно  |

Recept (IdRec, Opis, BrojOsoba, Ocena, IdAut)

|           |  |
|-----------|--|
| IdRec     | - цео број, идентификује рецепт, аутоматско додељивање наредног идентификатора |
| Opis      | - низ до 300 знакова, обавезно   |
| BrojOsoba | - цео број, већи од 0  |
| Ocena     | - реалан број, већи или једнак 0 и мањи или једнак 5                           |
| IdAut     | - страни кључ (табела Autor), обавезно   |

Sastojak (IdSas, Naziv, Poreklo)

|         |  |
|---------|--|
| IdSas   | - цео број, идентификује састојак, аутоматско додељивање наредног идентификатора |
| Naziv   | - низ до 300 знакова, обавезно   |
| Poreklo | - знак, обавезно, могуће вредности су 'Z', 'B' и 'O'                             |

**Напомена:**

Значење атрибута Tip: 'Z' – животињско порекло, 'B' – биљно порекло, 'O' – остало

Primenjuje (IdRec, RB, IdAkc)

|       |  |
|-------|--|
| IdRec | - страни кључ (табела Recept), обавезно, део примарног кључа |
| RB    | - цео број, обавезно, већи од 0, део примарног кључа         |
| IdAkc | - страни кључ (табела Akcija), обавезно, део примарног кључа |

korisiti (IdRec, RB, IdAkc, Kolicina, JedMere)

|          |   |
|----------|---|
| IdRec    | - страни кључ (табела primenjuje), обавезно |
| RB       | - страни кључ (табела primenjuje), обавезно |
| IdAkc    | - страни кључ (табела primenjuje), обавезно |
| IdSas    | - страни кључ (табела koristi), обавезно    |
| Kolicina | - реалан број, обавезно, већи од 0          |
| JedMere  | - низ до 50 знакова, обавезно               |

za (IdSas, Idrec)

|       |   |
|-------|---|
| IdSas | - страни кључ (табела Sastojak), обавезно |
| IdRec | - страни кључ (табела Recept), обавезно   |

Задатак 1 [4 поена]

Потребно је направити SQL упит који исписује све ауторе чији су сви рецепти са оценама изнад просека. У резултат укључити и ауторе којима су сви рецепти неоцењени. Сортирати резултат IdAut опадајуће.

Резултат дати у форми: IdAut,Ime.

У Sactus-у користити таб: Zadatak 1

**Није дозвољено коришћење погледа.**

---

```
SELECT *
FROM Autor
WHERE NOT EXISTS (
    SELECT *
    FROM Recept
    WHERE Recept.IdAut = Autor.IdAut
    AND Ocena < (SELECT AVG(Ocena) FROM Recept)
)
ORDER BY IdAut DESC
```

---

Задатак 2 [4 поена]

Потребно је направити SQL упит који проналази до три најбоље оцењена рецепта који могу да се спреме у року од једног сата и да им је притом оцена већа од 4. Уколико има више од три рецепта који би ушли у најбоља три рецепта (први критеријум је већа оцена), други критеријум за одабир је што мањи IdRec. Сортирати резултат по Ocena опадајуће, а затим по IdRec растуће.

Резултат дати у форми: Ocena, IdRec, Opis

У Sactus-у користити таб: Zadatak 2

---

```
SELECT Ocena, IdRec, Opis
FROM Recept
WHERE Ocena > 4 AND (
    SELECT SUM(Vreme)
    FROM primenjuje JOIN Akcija USING (IdAkc)
    WHERE IdRec = Recept.IdRec
) <= 3600
ORDER BY Ocena DESC, IdRec
LIMIT 3
```

---

### Задатак 3 [4 поена]

Потребно је направити SQL упит који за сваки рецепт приказује колико има састојака животињског, колико биљног и колико осталог порекла. Ако се више пута користи исти састојак у рецепту, бројати га једном. Сортирати резултат по IdRec растуће.

Резултат дати у форми: IdRec, Zivotinjsko poreklo, Biljno poreklo, Ostalo poreklo  
У Sactus-у користити таб: Zadatak 3

---

```
SELECT IdRec,(
    SELECT COUNT(DISTINCT IdSas) FROM koristi JOIN Sastojak USING(IdSas)
    WHERE koristi.IdRec = Recept.idRec AND Poreklo = 'Z'
) AS "Zivotinjsko poreklo",(
    SELECT COUNT(DISTINCT IdSas) FROM koristi JOIN Sastojak USING(IdSas)
    WHERE koristi.IdRec = Recept.idRec AND Poreklo = 'B'
) AS "Biljno poreklo",(
    SELECT COUNT(DISTINCT IdSas) FROM koristi JOIN Sastojak USING(IdSas)
    WHERE koristi.IdRec = Recept.idRec AND Poreklo = 'O'
) AS "Ostalo poreklo"
FROM Recept
```

---

### Задатак 4 [4 поена]

Потребно је направити SQL упит који за сваки састојак рецепта исписује колико је састојака потребно уколико би се правила храна за две особе. Не треба обраћати пажњу да ли је физички могуће или не направити такав састојак. Уколико се неки састојак користи више пута у истој јединици мере онда треба гледати укупно коришћење таквог састојка у тој јединици мере. Приликом рачунања, уколико је потребно дељење, крајњи резултат треба заокружити на више на две децимале. Узети у обзир само рецепте код којих је познат број особа. Срачунату количину састојка треба приказати у формату састојка: <naziv sastojka><zarez><razmak><kolicina><razmak><jedinica mere>. Сортирати резултат по IdRec растуће, а затим по IdSas растуће, по формату састојка растуће.

Препоручена документација: [Built-In Scalar SQL Functions \(sqlite.org\)](https://www.sqlite.org/lang_aggfunc.html)

Резултат дати у форми: IdRec, Opis, Sastojak  
У Sactus-у користити таб: Zadatak 4

---

```
SELECT IdRec, Opis, Naziv || ' ' || round(SUM(Kolicina)*2.0/BrojOsoba, 2) || ' ' || JedMere AS Sastojak
FROM Recept JOIN koristi USING(IdRec) JOIN Sastojak USING(IdSas)
WHERE BrojOsoba IS NOT NULL
GROUP BY IdRec, IdSas, JedMere
ORDER BY IdRec, IdSas, Sastojak
```

---

### Задатак 5 [5 поена]

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

У Cactus-у користити таб: Zadatak 5

---

```
DROP TABLE IF EXISTS Recept;
```

```
CREATE TABLE Recept(  
    IdRec  INTEGER PRIMARY KEY AUTOINCREMENT,  
    Opis   VARCHAR(300) NOT NULL,  
    BrojOsoba INTEGER CHECK(BrojOsoba>0),  
    Ocena  REAL CHECK(Ocena>=0 AND Ocena<=5),  
    IdAut  INTEGER REFERENCES Autor(IdAut) NOT NULL  
);
```

---

### Задатак 6 [5 поена]

Потребно је направити SQL упит који исписује за сваки рецепт којем типу исхране припада. Ако се у рецепту налази било који састојак животињског порекла, онда је тип "Mesni", уколико се у рецепту не налази ниједан састојак животињског порекла, а налази се барем један састојак биљног порекла, онда је тип исхране "Biljni", у супротном тип исхране је "Ostalo". Сортирати резултат по типу исхране тако да прво буду рецепти типа "Biljni", па "Ostalo" и на крају "Mesni", а у оквиру типа сортирати рецепте IdRec опадајуће.

Резултат дати у форми: IdRec, Tip ishrane

У Cactus-у користити таб: Zadatak 6

---

```
SELECT IdRec, CASE  
    WHEN EXISTS (  
        SELECT *  
        FROM Sastojak JOIN koristi USING(IdSas)  
        WHERE koristi.IdRec = Recept.IdRec AND Poreklo = 'Z'  
    ) THEN 'Mesni'  
    WHEN EXISTS (  
        SELECT *  
        FROM Sastojak JOIN koristi USING(IdSas)  
        WHERE koristi.IdRec = Recept.IdRec AND Poreklo = 'B'  
    ) THEN 'Biljni'  
    ELSE 'Ostalo'  
    END AS "Tip ishrane"  
FROM Recept  
ORDER BY CASE "Tip ishrane"  
    WHEN 'Biljni' THEN 1  
    WHEN 'Ostalo' THEN 2  
    ELSE 3  
END, IdRec DESC
```

---

Задатак 7 [5 поена]

Потребно је направити SQL упит који за сваки састојак исписује која акција се најчешће примењује над тим састојком. Ако се једна акција примењује над истим састојком више пута у оквиру једног рецепта, такве примене акција бројати једном. Ако има више акција које се најчешће примењују над састојком исписати ону акцију са мањим IdАкc. Уколико над неким састојком нема ниједна акција, онда у резултату за акцију (идентификатор и опис) ставити '/'. Резултат сортирати по IdSas растуће.

Резултат дати у форми: IdSas, Naziv, IdАкc, OpisАкције

У Сactus-у користити таб: Zadatak 7

**Није дозвољено коришћење погледа.**

---

```
WITH SastojakАкција AS (  
    SELECT IdSas, IdАкc, Opis, COUNT(DISTINCT IdRec) AS BrPrimena  
    FROM koristi JOIN Акција USING(IdАкc)  
    GROUP BY IdSas, IdАкc  
)  
SELECT IdSas, Naziv, COALESCE( IdАкc, '/') AS IdАкc, COALESCE( Opis, '/') AS OpisАкције  
FROM Sastojak LEFT JOIN SastojakАкција SA USING(IdSas)  
WHERE NOT EXISTS (  
    SELECT *  
    FROM SastojakАкција SA2  
    WHERE SA.IdSas = SA2.IdSas AND (SA2.BrPrimena > SA.BrPrimena  
        OR (SA2.BrPrimena = SA.BrPrimena AND SA2.IdАкc < SA.IdАкc))  
)  
ORDER BY IdSas
```

---

### Задатак 8 [6 поена]

Потребно је направити SQL скрипту која по потреби формира нове акције и додаје их на крај сваког рецепта који има дефинисан број особа и то акцију са тежином 1, временом по формули  $10\text{sec} * \text{BrojOsoba}$  и са описом:

- "Postaviti pribor za jelo za 1 osobu", уколико је BrojOsoba на рецепту једнак 1
- "Postaviti pribor za jelo za <BrojOsoba> osobe", уколико је BrojOsoba на рецепту једнак између 2 и 4
- "Postaviti pribor za jelo za <BrojOsoba> osoba", уколико је BrojOsoba на рецепту већи од 4

Уколико већ постоје акције са претходно наведеним описима, онда није потребно формирати нове акције, нити већ постојећим мењати тежину и време. Приликом додавања нових акција, за идентификатор узети први већи број у односу на тренутно највећи и новим акцијама редом делити наредне идентификаторе тако да онај са већим <BrojOsoba> има и већи идентификатор.

Након додавања акција, потребно их је додати на крај листе примењених акција одговарајућих рецепата.

Потребно је исписати за сваки рецепт редослед свих акција које се примењују. Сортирати резултат по IdRec растуће, а затим по RB растуће.

Резултат дати у форми: IdRec, RB, IdAkc, Opis, Tezina, Vreme

У Sactus-у користити таб: Zadatak 8

**Није дозвољено коришћење погледа.**

---

```
WITH NoveAkcije AS (  
    SELECT DISTINCT 'Postaviti pribor za jelo za ' || BrojOsoba || CASE  
        WHEN BrojOsoba = 1 THEN ' osobu'  
        WHEN BrojOsoba <= 4 THEN ' osobe'  
        ELSE ' osoba'  
    END AS OpisAkcije, BrojOsoba  
    FROM Recept  
    WHERE BrojOsoba IS NOT NULL AND OpisAkcije NOT IN (SELECT Opis FROM Akcija)  
)  
INSERT INTO Akcija  
SELECT (SELECT MAX(IdAkc) FROM Akcija) + (SELECT COUNT(*) FROM NoveAkcije WHERE  
    BrojOsoba<=np.BrojOsoba), OpisAkcije, 1, BrojOsoba*10  
FROM NoveAkcije np;  
  
INSERT INTO Primenuje (RB, IdAkc, IdRec)  
SELECT      (SELECT MAX(RB)+1 FROM Primenuje WHERE Primenuje.IdRec=r.IdRec),  
    (  
        SELECT MAX(IdAkc)  
        FROM Akcija  
        WHERE Opis = 'Postaviti pribor za jelo za ' || BrojOsoba || CASE  
            WHEN BrojOsoba = 1 THEN ' osobu'  
            WHEN BrojOsoba <= 4 THEN ' osobe'  
            ELSE ' osoba'  
        END  
    ),  
    IdRec  
FROM Recept r;  
  
SELECT * FROM primenuje JOIN Akcija USING (IdAkc) ORDER BY IdRec, RB
```

---

### Задатак 9 [6 поена]

Потребно је направити SQL упит који проналази почетак сваке серије од бар 3 акције које су исте тежине унутар рецепта (обратити пажњу да може да их има више). Резултат треба да садржи почетну акцију такве серије, као и редни број овакве серије у целој бази. Редни број серије се додељује редом од рецепата са мањим идентификатором, а унутар једног рецепте по мањем RB. Сортирати резултат по RedBroj серије растуће.

Резултат дати у форми: RedBroj, IdRec, RB, IdAkc, Tezina

У Сactus-у користити таб: Zadatak 9

**Није дозвољено коришћење погледа.**

---

```
WITH Serija AS (
SELECT IdRec, RB, IdAkc, Tezina
FROM primenjuje P JOIN Akcija A USING (IdAkc)
WHERE NOT EXISTS (
    SELECT *
    FROM primenjuje JOIN Akcija USING (IdAkc)
    WHERE primenjuje.RB = P.RB - 1 AND primenjuje.IdRec = P.IdRec
    AND Akcija.Tezina=A.Tezina
)
AND (
    SELECT MAX(RB)
    FROM primenjuje P1 JOIN Akcija USING (IdAkc)
    WHERE P1.RB > P.RB AND P1.IdRec = P.IdRec
    AND Akcija.Tezina=A.Tezina
    AND NOT EXISTS (
        SELECT *
        FROM primenjuje JOIN Akcija USING (IdAkc)
        WHERE primenjuje.RB > P.RB AND primenjuje.RB < P1.RB
        AND primenjuje.IdRec = P.IdRec AND Akcija.Tezina<>A.Tezina
    )
) >= P.RB + 2
)
SELECT (
    SELECT COUNT(*)
    FROM Serija
    WHERE IdRec<s.IdRec OR(IdRec=s.IdRec AND RB<=s.RB)
) AS RedBroj, *
FROM Serija s
ORDER BY RedBroj
```

---



Задатак 10 [7 поена]

Потребно је направити SQL упит који за сваки рецепт који је поставио аутор са идентификатором 10000 испишује колико је акција потребно за његову реализацију. Уколико се у рецепту налази састојак за који постоји један или више рецепата, сматрати да је потребно и тај састојак реализовати и користи се рецепт са најмањим идентификатором. Уколико се неки састојак за који постоји рецепт налази у више рецепата који се користе као састојци за главни рецепте, потребно га је направити само једном. Сматрати да се не може десити кружна зависност између рецепата. Сортирати резултат по IdRec рестуће.

Резултат дати у форми: IdRec, Opis, BrAkcija

У Sactus-у користити таб: Zadatak 10

**Није дозвољено коришћење погледа.**

---

```
WITH RECURSIVE ReceptPodrecept( IdRecOsn, IdRec) AS(
    SELECT IdRec, IdRec
    FROM Recept
    WHERE IdAut = 10000
    UNION
    SELECT RP.IdRec, R.IdRec
    FROM ReceptPodrecept RP JOIN koristi USING(IdRec)
        JOIN za Z USING(IdSas) JOIN recept R ON(Z.IdRec=R.IdRec)
    WHERE NOT EXISTS
        (SELECT * FROM za WHERE za.IdRec<Z.IdRec AND  za.IdSas=Z.IdSas)
)
SELECT IdRec, Opis, COUNT(IdAkc) AS BrAkcija
FROM ReceptPodrecept JOIN Recept USING(IdRec) JOIN Primenjuje USING(IdRec)
GROUP BY IdRec, Opis
ORDER BY IdRec
```

---