



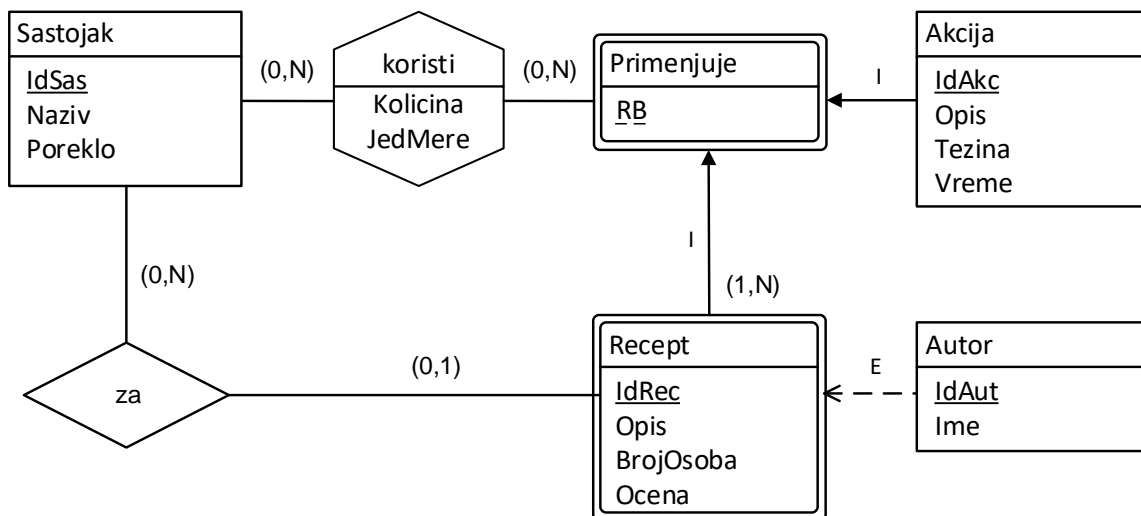
Базе података 1

(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 Акција), 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  
WHERE NOT EXISTS  
SELECT  
FROM  
WHERE Recipe.IdAut = Author.IdAut  
AND Ocena < (SELECT AVG(Ocena) FROM Recipe)  
  
|  
|
```

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 Осена опадајуће, а затим по IdRec растуће.

Резултат дати у форми: Осена, IdRec, Opis

У Sactus-у користити таб: Zadatak 2

```
SELECT Osena, IdRec, Opis  
FROM Recipe  
WHERE  
SELECT SUM(Ocena)  
FROM Recipe  
WHERE IdRec = Recipe.IdRec  
  
|  
ORDER BY Osena 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, Zivotinjsko Poreklo, Biljno Poreklo, Ostalo Poreklo
FROM Sactus.Zadatak3
ORDER BY IdRec ASC
```

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 sastoјka><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_built_in.html)

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

```
SELECT IdRec, Opis, Sastoјak
FROM Sactus.Zadatak4
ORDER BY IdRec ASC, IdSas ASC
```

Задатак 5 [5 поена]

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

У Sactus-у користити таб: Zadatak 5

```
CREATE TABLE Recipe
(
  IdRecipe INT PRIMARY KEY AUTOINCREMENT,
  Name VARCHAR(255) NOT NULL,
  Amount INT CHECK (Amount > 0),
  Price REAL CHECK (Price > 0 AND Price <= 1000),
  Type VARCHAR(255)
);
```

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 Sactus-у користити таб: Zadatak 6

```
SELECT IdRecipe, Tip ishrane
FROM Recipe
ORDER BY Tip ishrane, IdRecipe 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 IdАкс. 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, IdАкс, OpisАкције

У Сactus-у користити таб: Zadatak 7

Није дозвољено коришћење погледа.

```
WITH SASTOJAKS AS
SELECT IdSas, IdАкс, COUNT(*) AS BROJ
FROM SASTOJAKS
GROUP BY IdSas, IdАкс
ORDER BY IdSas ASC, BROJ DESC
LIMIT 1
UNION ALL
SELECT IdSas, Naziv, COALESCE(MAX(IdАкс), '/'), COALESCE(MAX(OpisАкције), '/')
FROM SASTOJAKS
WHERE NOT EXISTS (
SELECT 1
FROM SASTOJAKS SA
WHERE SA.IdSas = SASTOJAKS.IdSas AND SA.IdАкс < SASTOJAKS.IdАкс
OR (SA.IdАкс = SASTOJAKS.IdАкс AND SA.BROJ < SASTOJAKS.BROJ)
)
ORDER BY IdSas ASC
```

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, IdАкс, Opis, Tezina, Vreme

У Sactus-у користити таб: Zadatak 8

Није дозвољено коришћење погледа.

```
SELECT DISTINCT R.RECIPID AS IDREC,
                R.RECIPID AS RB,
                A.IDAKS AS IDAKS,
                A.OPIS AS OPIS,
                A.TEZINA AS TEZINA,
                A.VREME AS VREME
FROM RECIPI R
LEFT JOIN AKCIJE A ON R.RECIPID = A.RECIPID
ORDER BY R.RECIPID, A.IDAKS
```

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, IdАкц, Tezina

У Сactus-у користити таб: Zadatak 9

Није дозвољено коришћење погледа.

```
WITH Serija AS
SELECT IdRec, RB, IdAc, Tezina
FROM pomenje P JOIN Akcija A USING (IdAc)
WHERE NOT EXISTS
    (SELECT
        FROM pomenje JOIN Akcija USING (IdAc)
        WHERE pomenje.IdRec = P.IdRec AND pomenje.RB > P.RB
        AND Akcija.Tezina < A.Tezina
    )
AND
    (SELECT COUNT(*)
     FROM pomenje JOIN Akcija USING (IdAc)
     WHERE pomenje.IdRec = P.IdRec AND pomenje.RB > P.RB
     AND pomenje.IdRec = P.IdRec AND Akcija.Tezina < A.Tezina
    ) < 3
ORDER BY IdRec, RB, IdAc, Tezina
AS RedBroj
FROM Serija
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, BrАкција

У Сactus-у користити таб: Zadatak 10

Није дозвољено коришћење погледа.

```
[REDACTED SQL CODE]
```