



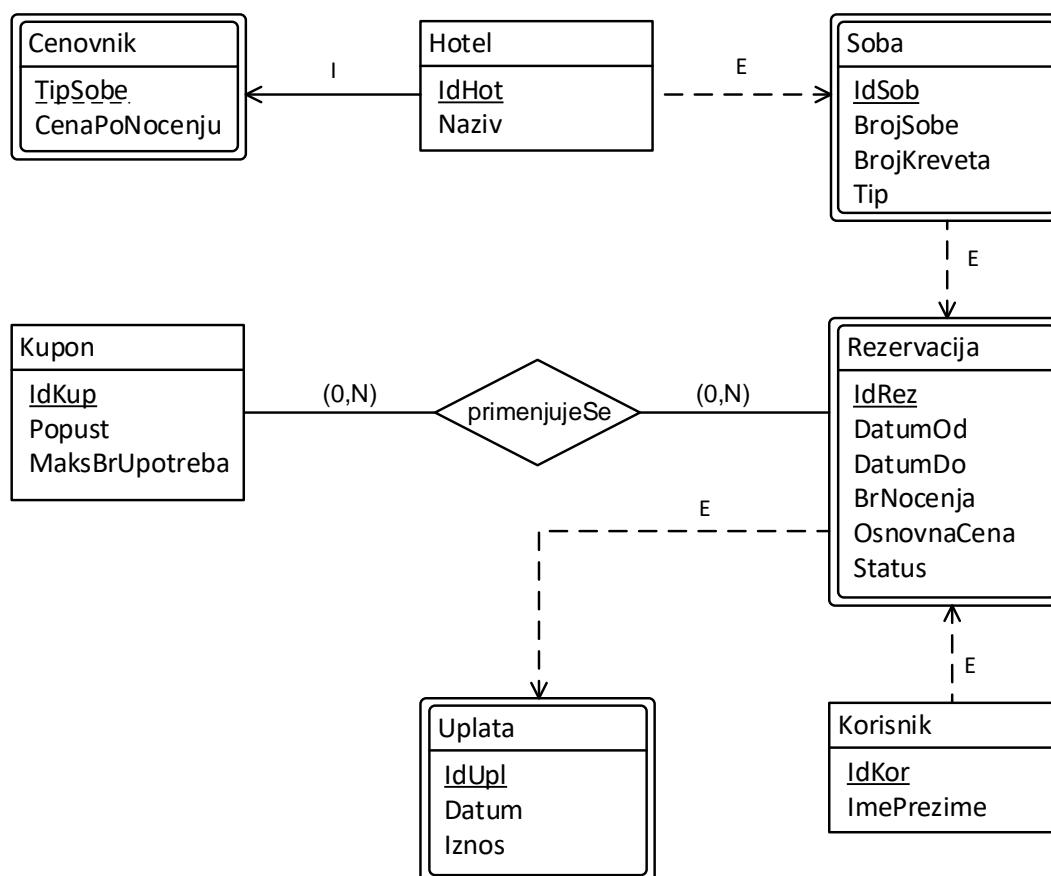
Базе података 1 (13С112БП1)

- јануарски испитни рок -

Група А

Посматра се база софтвера за резервације соба у хотелима. Хотелу се прати назив, а собама хотела број кревета, број собе и тип (луксузна, стандардна или економична). Хотел ценовником дефинише цену ноћења по особи за сваки тип собе. Корисници којим се памте име и презиме могу направити резервацију собе, за коју се прате датуми почетка и краја резервације, број ноћења, статус резервације (планирана, отказана, реализована) и основна цена боравка. Основна цена боравка формира се на основу броја ноћења, броја кревета у соби, типа собе и цене тог типа собе у том хотелу у тренутку креирања резервације. Корисници могу применити један или више купона са попустом на резервацију. Купони имају попуст и максималан број употреба. За уплате се прати резервација на коју се уплата односи, датум и износ уплате.

У наставку је дата релациона шема посматране базе података.



Hotel (IdHot, Naziv)

IdHot	- цео број, идентификује хотел, аутоматско додељивање наредног идентификатора
Naziv	- низ до 50 знакова, обавезно

Cenovnik (IdHot, TipSobe, CenaPoNocenju)

IdHot	- страни кључ (табела Hotel), обавезно, део примарног кључа
TipSobe	- тачно 1 знак, обавезно, део примарног кључа, могуће вредности су: 'L' (луксузна), 'S' (стандардна), 'E' (економична)
CenaPoNocenju	- реалан број, вредност већа од 0, обавезно

Soba (IdSob, BrojSobe, BrojKreveta, Tip, IdHot)

IdSob	- цео број, идентификује собу, аутоматско додељивање наредног идентификатора
BrojSobe	- цео број, обавезно
BrojKreveta	- цео број, вредност већа од 0, обавезно
Tip	- тачно 1 знак, обавезно, могуће вредности су: 'L' (луксузна), 'S' (стандардна), 'E' (економична)
IdHot	- страни кључ (табела Hotel), обавезно

Korisnik (IdKor, ImePrezime)

IdKor	- цео број, идентификује корисника, аутоматско додељивање наредног идентификатора
ImePrezime	- низ до 50 знакова, обавезно

Rezervacija (IdRez, DatumDo, DatumOd, BrNocenja, OsnovnaCena, Status, IdKor, IdSob)

IdRez	- цео број, идентификује резервацију, аутоматско додељивање наредног идентификатора
DatumOd	- низ од тачно 10 знакова, датум у формату: уууу-мм-дд, обавезно
DatumDo	- низ од тачно 10 знакова, датум у формату: уууу-мм-дд, обавезно
BrNocenja	- цео број, вредност већа или једнака 1, обавезно
OsnovnaCena	- реалан број, вредност већа од 0, обавезно
Status	- тачно 1 знак, обавезно, могуће вредности су: 'P' (планирана), 'R' (реализована), 'O' (отказана)
IdKor	- страни кључ (табела Korisnik), обавезно
IdSob	- страни кључ (табела Soba), обавезно

Uplata (IdUpl, Datum, Iznos, IdRez)

IdUpl	- цео број, идентификује уплату, аутоматско додељивање наредног идентификатора
Datum	- низ од тачно 10 знакова, датум у формату: уууу-мм-дд, обавезно
Iznos	- реалан број, вредност већа од 0, обавезно
IdRez	- страни кључ (табела Rezervacija), обавезно

Kupon (IdKup, Popust, MaksBrUpotreba)

IdKup	- цео број, идентификује купон, аутоматско додељивање наредног идентификатора
Popust	- реалан број, вредност већа од 0 и мања од 100, обавезно
MaksBrUpotreba	- цео број, вредност већа од 0

primenjujeSe (IdKup, IdRez)

IdKup	- страни кључ (табела Kupon), обавезно, део примарног кључа
IdRez	- страни кључ (табела Uplata), обавезно, део примарног кључа

Задатак 1 [4 поена]

Потребно је направити SQL упит који брише купоне који нису коришћени, а затим исписује све купоне уз информацију о величини попушта. Попуст се сматра малим ако је мањи од 10%, средњи ако је између 10% и 25% (укључујући 10% и 25% посто) и велики ако је преко 25%. Резултат сортирати по IdKup растуће.

Резултат дати у форми: IdKup, Popust, MaksBrUpotreba, Velicina kupona
У Сactus-у користити таб: Zadatak 1

```
DELETE FROM Kupon
WHERE IdKup NOT IN (SELECT IdKup FROM primenjujeSe);

SELECT IdKup, Popust, MaksBrUpotreba, CASE
    WHEN Popust < 10 THEN 'mali'
    WHEN Popust <=25 THEN 'srednji'
    ELSE 'veliki'
END AS "Velicina kupona"
FROM Kupon
ORDER BY IdKup
```

Задатак 2 [4 поена]

Потребно је направити SQL упит који приказује кориснике који су на сваку своју резервацију применили бар по један купон. Приказују се и корисници који нису имали резервације. Резултат сортирати по IdKor растуће.

Резултат дати у форми: IdKor, ImePrezime
У Сactus-у користити таб: Zadatak 2

```
SELECT IdKor, ImePrezime
FROM Korisnik
WHERE IdKor NOT IN(
    SELECT IdKor
    FROM Rezervacija
    WHERE IdRez NOT IN (SELECT IdRez FROM primenjujeSe)
)
ORDER BY IdKor
```

Задатак 3 [4 поена]

Потребно је направити SQL упит који приказује све кориснике који ни у једном тренутку нису имали резервације које се настављају једна на другу, не обавезно у истом хотелу. Резултат сортирати по IdKor опадајуће.

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

```
SELECT IdKor, ImePrezime
FROM Korisnik
WHERE IdKor NOT IN (
    SELECT IdKor
    FROM Rezervacija R1 JOIN Rezervacija R2 USING (IdKor)
    WHERE R1.DatumDo = R2.DatumOd
)
ORDER BY IdKor DESC
```

Задатак 4 [4 поена]

Потребно је направити SQL упит који за сваки хотел излистава колико је било резервација за сваки тип собе. Резултат сортирати по IdHot опадајуће.

Резултат дати у форми: IdHot, Naziv, Broj ekonomичних rezervacija, Broj standardnih rezervacija, Broj luksuznih rezervacija
У Cactus-у користити таб: Zadatak 4

```
SELECT IdHot, Naziv,
    (SELECT COUNT(*) FROM Soba JOIN Rezervacija USING(IdSob)
     WHERE Tip = 'E' AND Soba.IdHot=Hotel.IdHot) AS "Broj ekonomичних rezervacija",
    (SELECT COUNT(*) FROM Soba JOIN Rezervacija USING(IdSob)
     WHERE Tip = 'S' AND Soba.IdHot=Hotel.IdHot) AS "Broj standardnih rezervacija",
    (SELECT COUNT(*) FROM Soba JOIN Rezervacija USING(IdSob)
     WHERE Tip = 'L' AND Soba.IdHot=Hotel.IdHot) AS "Broj luksuznih rezervacija"
FROM Hotel
ORDER BY IdHot DESC
```

Задатак 5 [5 поена]

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

У Cactus-у користити таб: Zadatak 5

DROP TABLE IF EXISTS Soba;

CREATE TABLE Soba (
 IdSob **INTEGER PRIMARY KEY AUTOINCREMENT,**
 BrojSobe **INTEGER NOT NULL,**
 BrojKreveta **INTEGER NOT NULL CHECK(BrojKreveta > 0),**
 Tip **CHAR NOT NULL CHECK(Tip IN ('L', 'S', 'E')),**
 IdHot **INTEGER NOT NULL REFERENCES Hotel (IdHot)**
);

Задатак 6 [5 поена]

Потребно је написати SQL упит који исписује кориснике који су боравили у највећем броју различитих хотела. Резултат сортирати по IdKor опадајуће.

Резултат дати у форми: IdKor, ImePrezime, BrPosecenihHotela

У Cactus-у користити таб: Zadatak 6

Није дозвољено коришћење погледа.

WITH KorisnikInfo(IdKor, BrPosecenihHotela) AS (
 SELECT IdKor, COUNT(DISTINCT IdHot)
 FROM Rezervacija JOIN Soba USING(IdSob)
 WHERE Status = 'R'
 GROUP BY IdKor
)
SELECT IdKor, ImePrezime, BrPosecenihHotela AS BrPosecenihHotela
FROM Korisnik JOIN KorisnikInfo USING(IdKor)
WHERE BrPosecenihHotela = (SELECT MAX(BrPosecenihHotela) FROM KorisnikInfo)
ORDER BY IdKor DESC

Задатак 7 [5 поена]

Потребно је написати SQL скрипту која исписује хотеле у којима је просечна цена ноћења за реализоване резервације плаћена више од просека, сортирано прво по ProsecnaCenaPoNoci растуће, па по IdHot растуће.

Резултат дати у форми: IdHot, Naziv, ProsecnaCenaPoNoci

У Сactus-у користити таб: Zadatak 7

Није дозвољено коришћење погледа.

```
SELECT IdHot, Naziv, SUM(OsnovnaCena)/SUM(BrNocenja) AS ProsecnaCenaPoNoci
FROM Hotel JOIN Soba USING(IdHot) JOIN Rezervacija USING(IdSob)
WHERE Status = 'R'
GROUP BY IdHot
HAVING SUM(OsnovnaCena)/SUM(BrNocenja) > (
    SELECT SUM(OsnovnaCena)/SUM(BrNocenja)
    FROM Rezervacija
    WHERE Status = 'R'
)
ORDER BY ProsecnaCenaPoNoci, IdHot
```

Задатак 8 [6 поена]

Потребно је написати SQL скрипту која за сваку резервацију проверава да ли је уплаћена свота новца већа од потребне (обрачунавајући и евентуалне попусте остварене купонима), па у случају да јесте брише најскорије забележене уплате које су вишак и смањује износ последње преостале забележене уплате везане за ту резервацију, тако да резервација буде тачно покривена уплатама. Скрипта након тога треба да испише све уплате, сортирано по IdUpl растуће.

Резултат дати у форми: IdUpl, Datum, Iznos, IdRez

У Sactus-у користити таб: Zadatak 8

Није дозвољено коришћење додатних структура – погледа или помоћних табела.

```
WITH RC(IdRez, Cena) AS (  
    SELECT IdRez, OsnovnaCena*(100.0- COALESCE(SUM(Popust),0))/100.0  
    FROM Rezervacija LEFT JOIN PrimenujeSe USING(IdRez) LEFT JOIN Kupon USING(IdKup)  
    GROUP BY IdRez  
)  
DELETE FROM Uplata  
WHERE (  
    SELECT MAX(Cena)  
    FROM RC  
    WHERE Uplata.IdRez= RC.IdRez  
) <= (  
    SELECT SUM(Iznos)  
    FROM Uplata U1  
    WHERE U1.IdRez = Uplata.IdRez AND  
    (U1.Datum < Uplata.Datum OR (U1.Datum = Uplata.Datum AND U1.IdUpl < Uplata.IdUpl))  
);  
  
WITH RC(IdRez, Cena) AS (  
    SELECT IdRez, OsnovnaCena*(100.0- COALESCE(SUM(Popust),0))/100.0  
    FROM Rezervacija LEFT JOIN PrimenujeSe USING(IdRez) LEFT JOIN Kupon USING(IdKup)  
    GROUP BY IdRez  
)  
UPDATE Uplata  
SET Iznos = (  
    SELECT MAX(Cena) FROM RC WHERE Uplata.IdRez = RC.IdRez) -(  
    SELECT COALESCE(SUM(Iznos),0)  
    FROM Uplata U1  
    WHERE U1.IdRez= Uplata.IdRez AND  
    (U1.Datum < Uplata.Datum OR (U1.Datum = Uplata.Datum AND U1.IdUpl < Uplata.IdUpl))  
)  
WHERE Datum = (SELECT MAX(Datum) FROM Uplata U1 WHERE U1.IdRez = Uplata.IdRez)  
AND (SELECT MAX(Cena) FROM RC WHERE Uplata.IdRez = RC.IdRez) < (  
    SELECT COALESCE(SUM(Iznos),0)  
    FROM Uplata U1  
    WHERE U1.IdRez= Uplata.IdRez AND  
    (U1.Datum < Uplata.Datum OR (U1.Datum = Uplata.Datum AND U1.IdUpl <= Uplata.IdUpl))  
);  
  
SELECT * FROM Uplata ORDER BY IdUpl
```

Задатак 9 [6 поена]

Потребно је направити SQL упит који за сваки датум у 2022. години исписује колико резервација је било актуелно тога датума. Резервација је актуелна неког датума ако је реализована, а тај датум се налази између почетног и крајњег датума резервације, не укључујући их. Резултат сортирати по датуму растуће.

Препоручена документација:

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

Резултат дати у форми: Datum, BrojAktuelnihRezervacija

У Sactus-у користити таб: Zadatak 9

Није дозвољено коришћење погледа.

```
WITH RECURSIVE Datumi(Datum) AS (  
    VALUES('2022-01-01')  
    UNION ALL  
    SELECT DATE(Datum, '+1 day')  
    FROM Datumi  
    WHERE Datum < '2022-12-31'  
)  
SELECT Datum, (  
    SELECT COUNT(*)  
    FROM Rezervacija  
    WHERE Status='R' AND DatumOd < Datum AND Datum < DatumDo  
) AS BrojAktuelnihRezervacija  
FROM Datumi  
ORDER BY Datum
```

Задатак 10 [7 поена]

Потребно је направити SQL упит који за сваку собу проналази најдужи интервал дана у којим је она узастопно била резервисана и исписује датуме почетка и краја тог интервала и број реализованих резервација у том периоду. У случају да је за једну собу пронађено више интервала континуалног издавања максималне дужине, исписати их све. За собе које никад нису имале резервацију, за датум од, датум до и број реализованих исписати NULL. Резултат сортирати по IdSob растуће, па по датуму почетка интервала растуће.

Резултат дати у форми: IdSob, DatumOd, DatumDo, BrojRealizovanih

У Сactus-у користити таб: Zadatak 10

Није дозвољено коришћење погледа.

```
WITH RECURSIVE IntervaliRezervacijaSoba AS
(
    SELECT IdSob, DatumOd, DatumDo, BrNocenja,
           CASE WHEN Status='R' THEN 1 ELSE 0 END AS BrojRealizovanih
    FROM Rezervacija
    UNION ALL
    SELECT IdSob, I.DatumOd, R.DatumDo, I.BrNocenja + R.BrNocenja,
           I.BrojRealizovanih + CASE WHEN Status='R' THEN 1 ELSE 0 END
    FROM Rezervacija R JOIN IntervaliRezervacijaSoba I USING(IdSob)
    WHERE I.DatumDo = R.DatumOd
),
MaxDuzinaIntervalaPoSobi AS (
    SELECT IdSob, DatumOd, DatumDo, BrNocenja, BrojRealizovanih
    FROM IntervaliRezervacijaSoba I1
    WHERE BrNocenja = (
        SELECT MAX(BrNocenja)
        FROM IntervaliRezervacijaSoba I2
        WHERE I1.IdSob = I2.IdSob
    )
)
SELECT IdSob, DatumOd, DatumDo, BrojRealizovanih
FROM Soba LEFT JOIN MaxDuzinaIntervalaPoSobi USING (IdSob)
ORDER BY IdSob, DatumOd
```
