



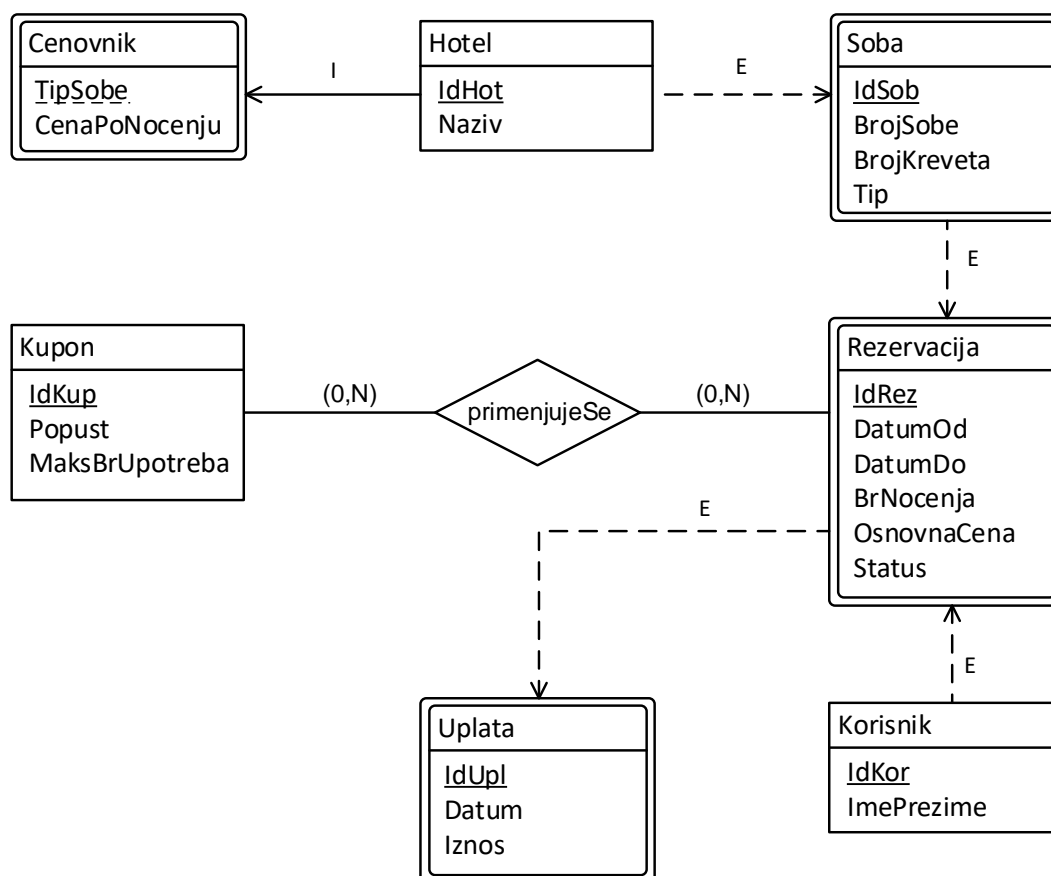
Базе података 1 (13Е113БП1)

- јануарски испитни рок -

Група Ц

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 кориснике без резервација, а затим исписује све кориснике. Резултат сортирати по IdKor растуће.

Резултат дати у форми: IdKor, Ime i prezime
У Сactus-у користити таб: Zadatak 1

```
DELETE FROM Korisnik
WHERE IdKor NOT IN (SELECT IdKor FROM Rezervacija);

SELECT IdKor, ImePrezime AS "Ime i prezime"
FROM Korisnik
ORDER BY IdKor
```

Задатак 2 [4 поена]

Потребно је направити SQL упит који приказује купоне које је могуће искористити барем још једном. 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 2

```
SELECT IdKup, Popust, MaksBrUpotreba, CASE
    WHEN Popust < 10 THEN 'mali'
    WHEN Popust <=25 THEN 'srednji'
    ELSE 'veliki'
END AS "Velicina kupona"
FROM Kupon
WHERE MaksBrUpotreba IS NULL OR MaksBrUpotreba > (
    SELECT COUNT(*)
    FROM primenjujeSe
    WHERE Kupon.IdKup = IdKup
)
ORDER BY IdKup
```

Задатак 3 [4 поена]

Потребно је направити SQL упит који за сваког корисника исписује да ли су барем у неком тренутку имали резервације које се настављају једна на другу, не обавезно у истом хотелу. Уколико су имали, потребно је исписати 'da'. У супротном, исписати 'ne'. Резултат сортирати по IdKor опадајуће.

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

```
SELECT IdKor, ImePrezime, CASE
    WHEN IdKor IN (
        SELECT IdKor
        FROM Rezervacija R1 JOIN Rezervacija R2 USING (IdKor)
        WHERE R1.DatumDo = R2.DatumOd) THEN 'da'
    ELSE 'ne'
END AS VezaneRezervacije
FROM Korisnik
ORDER BY IdKor DESC
```

Задатак 4 [4 поена]

Потребно је направити SQL упит који за сваки хотел излистава колико има соба којег типа. Резултат сортирати по IdHot растуће.

Резултат дати у форми: IdHot, Naziv, Broj economicnih soba, Broj standardnih soba, Broj luksuznih soba
У Сactus-у користити таб: Zadatak 4

```
SELECT IdHot, Naziv,
    (SELECT COUNT(*) FROM Soba
     WHERE Tip = 'E' AND Soba.IdHot=Hotel.IdHot) AS "Broj economicnih soba",
    (SELECT COUNT(*) FROM Soba
     WHERE Tip = 'S' AND Soba.IdHot=Hotel.IdHot) AS "Broj standardnih soba",
    (SELECT COUNT(*) FROM Soba
     WHERE Tip = 'L' AND Soba.IdHot=Hotel.IdHot) AS "Broj luksuznih soba"
FROM Hotel
ORDER BY IdHot
```

Задатак 5 [5 поена]

Потребно је направити SQL скрипту која, ако постоји табела **Rezervacija**, избацује табелу **Rezervacija** из шеме, а затим формира нову табелу **Rezervacija** која треба да има одговарајућу структуру и ограничења. За формат поља DatumOd и DatumDo проверавати само да ли се на одговарајућим позицијама налази карактер '-!'.
У Cactus-у користити таб: Zadatak 5

DROP TABLE IF EXISTS Rezervacija;

CREATE TABLE Rezervacija (
 IdRez INTEGER PRIMARY KEY AUTOINCREMENT,
 DatumOd CHAR(10) NOT NULL CHECK(DatumOd LIKE '____-__-__'),
 DatumDo CHAR(10) NOT NULL CHECK(DatumDo LIKE '____-__-__'),
 BrNocenja INTEGER NOT NULL CHECK(BrNocenja > 0),
 OsnovnaCena REAL NOT NULL CHECK(OsnovnaCena > 0),
 Status CHAR NOT NULL CHECK(Status IN ('P', 'O', 'R')),
 IdKor INTEGER NOT NULL REFERENCES Korisnik (IdKor),
 IdSob INTEGER NOT NULL REFERENCES Soba (IdSob)
);

Задатак 6 [5 поена]

Потребно је написати SQL упит који исписује кориснике који су највећи број пута искористили исти купон. Резултат сортирати по IdKor опадајуће.

Резултат дати у форми: IdKor, ImePrezime

У Cactus-у користити таб: Zadatak 6

Није дозвољено коришћење погледа.

WITH KorisnikInfo(IdKor, IdKup, BrPuti) AS (
 SELECT IdKor, IdKup, COUNT(*)
 FROM Rezervacija JOIN primenjujeSe USING(IdRez)
 GROUP BY IdKor, IdKup
)
SELECT DISTINCT IdKor, ImePrezime
FROM Korisnik JOIN KorisnikInfo USING(IdKor)
WHERE BrPuti = (SELECT MAX(BrPuti) FROM KorisnikInfo)
ORDER BY IdKor DESC

Задатак 7 [5 поена]

Потребно је написати SQL скрипту која исписује кориснике и њихове просечне лојалности хотелима, само за кориснике чија је просечна лојалност већа од просека лојалности свих корисника свим хотелима. Лојалност корисника хотелу је укупан број ноћи које је корисник провео у хотелу. Резултат сортирати прво по ProsecnaLojalnostKorisnika растуће, па по IdKor растуће.

Резултат дати у форми: IdKor, ImePrezime, ProsecnaLojalnostKorisnika

У Сactus-у користити таб: Zadatak 7

Није дозвољено коришћење погледа.

```
WITH Lojalnost (IdKor, IdHotel, BrNocenja) AS (  
    SELECT IdKor, IdHot, SUM(BrNocenja)  
    FROM Rezervacija JOIN Soba USING(IdSob)  
    WHERE Status = 'R'  
    GROUP BY IdKor, IdHot  
)  
SELECT IdKor, ImePrezime, AVG(BrNocenja) AS ProsecnaLojalnostKorisnika  
FROM Korisnik JOIN Lojalnost USING(IdKor)  
GROUP BY IdKor  
HAVING AVG(BrNocenja) > (  
    SELECT AVG(BrNocenja)  
    FROM Lojalnost  
)  
ORDER BY ProsecnaLojalnostKorisnika, IdKor
```

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 смена се десило тога датума. Сменом се сматра промена корисника који користи собу, односно долазак корисника у некоришћену собу, одлазак корисника из собе или одлазак једног, а долазак другог корисника у собу. Резултат сортирати по датуму растуће.

Препоручена документација:

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

Резултат дати у форми: Datum, BrojSmena

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(DISTINCT IdSob)  
    FROM Rezervacija  
    WHERE Status='R' AND (DatumOd = Datum OR Datum = DatumDo)  
) AS BrojSmena  
FROM Datumi  
ORDER BY Datum
```

Задатак 10 [7 поена]

Потребно је направити SQL упит који проналази собе које су биле резервисане највећи број дана у континуитету и за њих исписује и период када је то било и број реализованих резервација у том периоду. У случају да је за једну собу пронађено више интервала континуалног издавања максималне дужине, исписати их све. Резултат сортирати по IdSob растуће, па по датуму почетка интервала растуће.

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

У Sactus-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
)
SELECT IdSob, DatumOd, DatumDo, BrojRealizovanih
FROM IntervaliRezervacijaSoba
WHERE BrNocenja = (SELECT MAX(BrNocenja) FROM IntervaliRezervacijaSoba)
ORDER BY IdSob, DatumOd
```
