



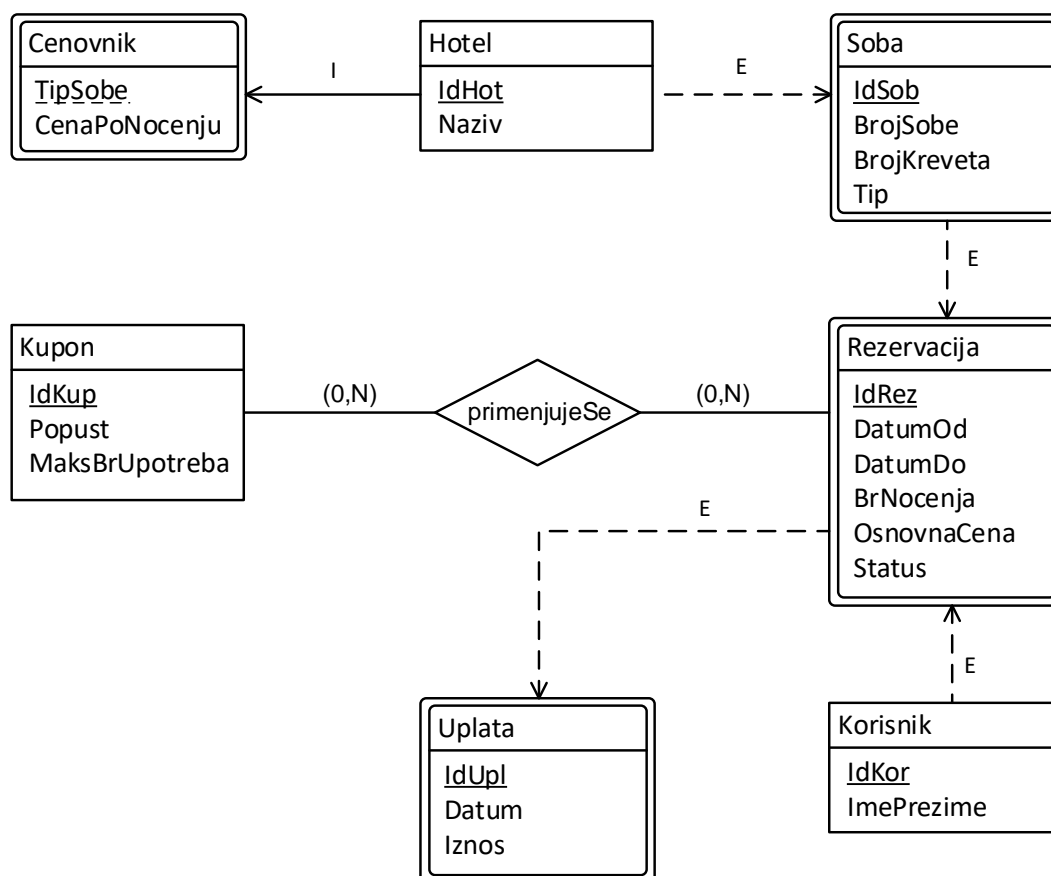
Базе података 1 (13С112БП1)

- јануарски испитни рок -

Група Б

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 из ценовника брише цене за оне типове соба које се не појављују у хотелу на који се ценовник односи, а затим исписује ценовник заједно са пуним типом собе. Резултат сортирати по IdHot растуће, па по типу собе растуће.

Резултат дати у форми: IdHot, Tip Sobe, CenaPoNocenju

У Cactus-у користити таб: Zadatak 1

```
DELETE FROM Cenovnik
WHERE NOT EXISTS (
    SELECT IdSob
    FROM Soba
    WHERE Cenovnik.IdHot=Soba.IdHot AND Cenovnik.TipSobe=Soba.Tip);

SELECT IdHot, CASE TipSobe
    WHEN 'L' THEN 'luksuzna'
    WHEN 'S' THEN 'standardna'
    WHEN 'E' THEN 'ekonomicna'
END AS "Tip Sobe", CenaPoNocenju
FROM Cenovnik
ORDER BY IdHot, TipSobe
```

Задатак 2 [4 поена]

Потребно је направити SQL упит који приказује кориснике који су на сваку своју резервацију применили максимално један купон, не обавезно исти на сваку од њих. Приказују се и корисници који нису имали резервације. Резултат сортирати по IdKor растуће.

Резултат дати у форми: IdKor, ImePrezime

У Cactus-у користити таб: Zadatak 2

```
SELECT IdKor, ImePrezime
FROM Korisnik
WHERE IdKor NOT IN(
    SELECT IdKor
    FROM Rezervacija JOIN primenjujeSe USING(IdRez)
    GROUP BY IdRez, IdKor
    HAVING COUNT(*)>1
)
ORDER BY IdKor
```

Задатак 3 [4 поена]

Потребно је направити SQL упит који приказује све собе које ни у једном тренутку нису имале резервације које се настављају једна на другу. Резултат сортирати по IdSob опадајуће.

Резултат дати у форми: IdSob, BrojSobe, BrojKreveta, Tip, IdHot
У Сactus-у користити таб: Zadatak 3

```
SELECT IdSob, BrojSobe, BrojKreveta, Tip, IdHot
FROM Soba
WHERE IdSob NOT IN (
    SELECT IdSob
    FROM Rezervacija R1 JOIN Rezervacija R2 USING (IdSob)
    WHERE R1.DatumDo = R2.DatumOd
)
ORDER BY IdSob DESC
```

Задатак 4 [4 поена]

Потребно је направити SQL упит који за сваког корисника излистава колико пута је резервисао који тип собе. Резултат сортирати по IdKor растуће.

Резултат дати у форми: IdKor, ImePrezime, Broj ekonomичних rezervacija, Broj standardnih rezervacija, Broj luksuznih rezervacija
У Сactus-у користити таб: Zadatak 4

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

Задатак 5 [5 поена]

Потребно је направити SQL скрипту која, ако постоји табела **Uplata**, избацује табелу **Uplata** из шеме, а затим формира нову табелу **Uplata** која треба да има одговарајућу структуру и ограничења. За формат поља Datum проверавати само да ли се на одговарајућим позицијама налази карактер '-'.

У Cactus-у користити таб: Zadatak 5

DROP TABLE IF EXISTS Uplata;

CREATE TABLE Uplata (
 IdUpl **INTEGER PRIMARY KEY AUTOINCREMENT,**
 Datum **CHAR(10) NOT NULL CHECK(Datum LIKE '____-__-__'),**
 Iznos **REAL NOT NULL CHECK(Iznos > 0),**
 IdRez **INTEGER NOT NULL REFERENCES Rezervacija (IdRez)**
);

Задатак 6 [5 поена]

Потребно је написати SQL упит који исписује хотеле који су угостили највећи број различитих корисника. Резултат сортирати по IdHot опадајуће.

Резултат дати у форми: IdHot, Naziv, BrGostiju

У Cactus-у користити таб: Zadatak 6

Није дозвољено коришћење погледа.

WITH HotelInfo(IdHot, BrGostiju) AS (
 SELECT IdHot, COUNT(DISTINCT IdKor)
 FROM Rezervacija JOIN Soba USING(IdSob)
 WHERE Status = 'R'
 GROUP BY IdHot
)
SELECT IdHot, Naziv, BrGostiju AS BrGostiju
FROM Hotel JOIN HotelInfo USING(IdHot)
WHERE BrGostiju = (SELECT MAX(BrGostiju) FROM HotelInfo)
ORDER BY IdHot DESC

Задатак 7 [5 поена]

Потребно је написати SQL скрипту која исписује кориснике који су ноћ у хотелу за реализоване резервације у просеку платили више од просека, сортирано прво по ProsecnaCenaPoNoci растуће, па по IdKor растуће.

Резултат дати у форми: IdKor, ImePrezime, ProsecnaCenaPoNoci

У Сactus-у користити таб: Zadatak 7

Није дозвољено коришћење погледа.

```
SELECT IdKor, ImePrezime, SUM(OsnovnaCena)/SUM(BrNocenja) AS ProsecnaCenaPoNoci
FROM Korisnik JOIN Rezervacija USING(IdKor)
WHERE Status = 'R'
GROUP BY IdKor
HAVING SUM(OsnovnaCena)/SUM(BrNocenja) > (
    SELECT SUM(OsnovnaCena)/SUM(BrNocenja)
    FROM Rezervacija
    WHERE Status = 'R'
)
ORDER BY ProsecnaCenaPoNoci, 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/funcintro.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 кориснике који су резервисале собе у хотелу или хотелима највећи број дана у континуитету и за њих испишује и период када је то било и број реализованих резервација у том периоду. У случају да је за једног корисника пронађено више интервала континуалног боравка у хотелима максималне дужине, исписати их све. За кориснике које никад нису имале резервацију, за датум од, датум до и број реализованих исписати NULL. Резултат сортирати по IdKor растуће, па по датуму почетка интервала растуће.

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

У Сactus-у користити таб: Zadatak 10

Није дозвољено коришћење погледа.

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