



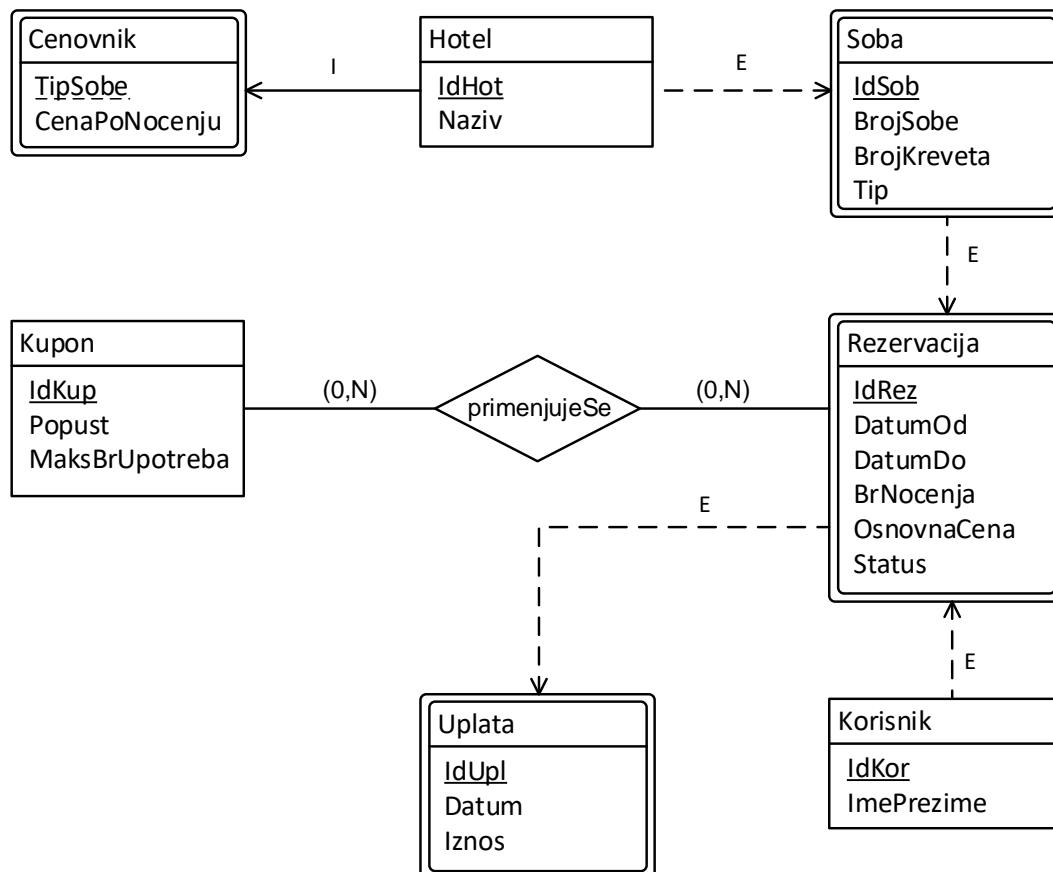
Базе података 1 (13С112БП1)

- 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 - ceo 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 - ceo број, идентификује собу, аутоматско додељивање наредног идентификатора
- BrojSobe - ceo број, обавезно
- BrojKreveta - ceo 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 - ceo број, идентификује корисника, аутоматско додељивање наредног идентификатора
- ImePrezime - низ до 50 знакова, обавезно

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

- IdRez - ceo 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 - ceo 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 - ceo 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 - ceo број, идентификује купон, аутоматско додељивање наредног идентификатора
- Popust - реалан број, вредност већа од 0 и мања од 100, обавезно
- MaksBrUpotreba - ceo 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 Сactus-у користити таб: Zadatak 1

```
DELETE FROM CENOVNIK WHERE NOT EXISTS (SELECT * FROM HOTEL WHERE HOTEL.IDHOT = CENOVNIK.IDHOT);
SELECT * FROM CENOVNIK ORDER BY IdHOT ASC, TipSobe ASC;
```

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, ИмеPrezime
У Сactus-у користити таб: Zadatak 2

```
SELECT * FROM KORISNICI ORDER BY IdKor ASC;
```

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
    WHERE IdSob = Soba.IdSob
)
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 = Kornik.IdKor) AS 'Broj ekonomичnih rezervacija',
       (SELECT COUNT(*) FROM Soba JOIN Rezervacija USING (IdSob)
        WHERE Tip = 'S' AND Rezervacija.IdKor = Kornik.IdKor) AS 'Broj standardnih rezervacija',
       (SELECT COUNT(*) FROM Soba JOIN Rezervacija USING (IdSob)
        WHERE Tip = 'L' AND Rezervacija.IdKor = Kornik.IdKor) AS 'Broj luksuznih rezervacija'
FROM Kornik
ORDER BY IdKor ASC
```

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
(
    Datum DATE,
    IdHot INT,
    Naziv VARCHAR(255),
    BrGostiju INT,
    PRIMARY KEY (Datum, IdHot),
    FOREIGN KEY (IdHot) REFERENCES Rezervacije (IdHot)
);
```

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.

```
SELECT IdHot, Naziv, BrGostiju AS BrGostiju
FROM Rezervacije JOIN Hoteli USING (IdHot)
GROUP BY IdHot, Naziv
ORDER BY BrGostiju DESC;

SELECT IdHot, Naziv, BrGostiju AS BrGostiju
FROM Rezervacije JOIN Hoteli USING (IdHot)
GROUP BY IdHot, Naziv
ORDER BY BrGostiju 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 ProsečnaCenaPoNoci растуће, па по IdKor растуће.

Резултат дати у форми: IdKor, ИмеPrezime, ProsečnaCenaPoNoci

У Сactus-у користити таб: Zadatak 7

Није дозвољено коришћење погледа.

```
SELECT IdKor, ИмеPrezime, SUM(OSNOVNA_CENA)/SUM(BROJNOCI) AS ProsečnaCenaPoNoci
FROM Korisnik JOIN Rezervacija USING (IdKor)
GROUP BY IdKor, ИмеPrezime
HAVING SUM(OSNOVNA_CENA)/SUM(BROJNOCI) >
    (SELECT SUM(OSNOVNA_CENA)/SUM(BROJNOCI)
     FROM Korisnik
     GROUP BY IdKor, ИмеPrezime)
ORDER BY ProsečnaCenaPoNoci, 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 Cactus-у користити таб: Zadatak 8

Није дозвољено коришћење додатних структура – погледа или помоћних табела.

```
WITH RECURSIVE C AS
SELECT IdRez, CnovoIznos, Datum, COALESCE(SUM(Popust),0) AS Popust
FROM Rezervacija LEFT JOIN PrimljenjeP using(IdRez) LEFT JOIN Kupon USING(KuponId)
ORDER BY IdRez;
|
|
WITH RECURSIVE R AS
WITH RECURSIVE R AS
SELECT MAX(Iznos) AS Iznos
FROM R
WHERE Datum <= Uplata.Datum OR (Uplata.Datum = Uplata.Datum AND Uplata.Iznos < Uplata.Iznos)
|
|
WITH RECURSIVE C AS
SELECT IdRez, CnovoIznos, Datum, COALESCE(SUM(Popust),0) AS Popust
FROM Rezervacija LEFT JOIN PrimljenjeP using(IdRez) LEFT JOIN Kupon USING(KuponId)
ORDER BY IdRez;
|
|
WITH RECURSIVE R AS
WITH RECURSIVE R AS
SELECT MAX(Iznos) FROM R WHERE Uplata.Iznos = R.Iznos
SELECT COALESCE(SUM(Iznos),0)
FROM R
WHERE Uplata.Iznos = Uplata.Iznos AND
Uplata.Datum <= Uplata.Datum OR (Uplata.Datum = Uplata.Datum AND Uplata.Iznos < Uplata.Iznos)
|
|
WITH RECURSIVE R AS
WITH RECURSIVE R AS
SELECT MAX(Iznos) FROM R WHERE Uplata.Iznos = R.Iznos
SELECT COALESCE(SUM(Iznos),0)
FROM R
WHERE Uplata.Iznos = Uplata.Iznos AND
Uplata.Datum <= Uplata.Datum OR (Uplata.Datum = Uplata.Datum AND Uplata.Iznos < Uplata.Iznos)
|
|
WITH RECURSIVE R AS
WITH RECURSIVE R AS
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/dropfunctions.html)
- [Built-In Scalar SQL Functions \(sqlite.org\)](https://www.sqlite.org/dropfunctions.html)

Резултат дати у форми: Datum, BrojSmena

У Sactus-у користити таб: Zadatak 9

Није дозвољено коришћење погледа.

```
WITH RECURSIVE Datum AS
(
    SELECT DATE('2022-01-01') AS Datum
    UNION ALL
    SELECT DATE(Datum, '+1 day') AS Datum
    FROM Datum
    WHERE Datum < DATE('2022-12-31')
)
SELECT Datum,
COUNT(*) AS BrojSmena
FROM (
    SELECT Datum,
    CASE
        WHEN (Sobid = R AND Ddatum = Datum OR Ddatum = Datum)
        THEN 1
        ELSE 0
    END AS BrojSmena
    FROM Sactus.Zadatak_9 AS S
    JOIN Datum AS D
    ON S.Datum = D.Datum
)
GROUP BY Datum
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 исписује 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.

```
[REDACTED SQL CODE]
```