



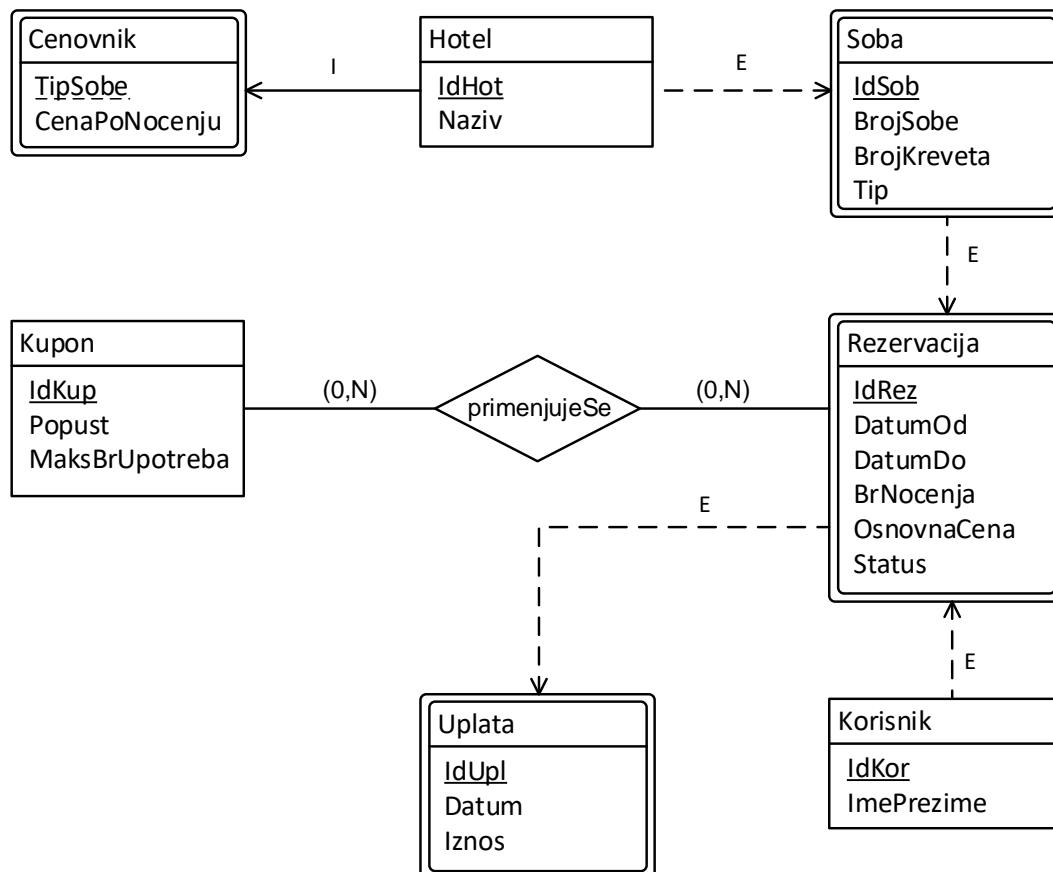
Базе података 1 (13Е113БП1)

- 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 - 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 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
  (korisnik_idkor NOT IN (SELECT korisnik_idkor
    FROM rezervacija))
SELECT IdKor, Ime, Prezime 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 попуста. 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, Velicina Kupona
FROM kupon
WHERE (MaksBrUpotreba IS NULL OR MaksBrUpotreba > 1)
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

```
[REDACTED SQL CODE]
```

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

```
[REDACTED SQL CODE]
```

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

```
CREATE TABLE Rezervacija
(
    IdKor INT NOT NULL CHECK (IDKOR > 0) PRIMARY KEY,
    IdKup INT NOT NULL CHECK (IDKUP > 0),
    DatumOd DATETIME NOT NULL CHECK (DATEDIFF(DAY, DatumOd, DatumDo) >= 0),
    DatumDo DATETIME NOT NULL CHECK (DATEDIFF(DAY, DatumOd, DatumDo) >= 0),
    BrojMesta INT NOT NULL CHECK (BROJMESTA > 0),
    Cena DECIMAL(10, 2) NOT NULL CHECK (CENA > 0),
    IdKorisnik INT NOT NULL CHECK (IDKORISNIK > 0) REFERENCES Korisnik (IdKor),
    IdSala INT NOT NULL CHECK (IDSALA > 0) REFERENCES Sala (IdSala)
);
```

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 Korisnik AS
(
    SELECT IdKor, IdKup, COUNT(*) AS BrojKoriscenja
    FROM Rezervacija
    GROUP BY IdKor, IdKup
)
SELECT IdKor, ImePrezime
FROM Korisnik JOIN KorisnikInfo ON IdKor = IdKor
ORDER BY BrojKoriscenja 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 Sactus-у користити таб: Zadatak 7

Није дозвољено коришћење погледа.

[Redacted content]

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, COALESCE(SUM(Iznos) OVER (PARTITION BY IdRez ORDER BY Datum ASC), 0) AS SumIznos
FROM Rezervacija LEFT JOIN PomoćniPse USING (IdRez) LEFT JOIN Kupon USING (IdKup)
ORDER BY IdRez
;
WITH RECURSIVE U AS
SELECT IdRez, Datum, Iznos
FROM Uplata
WHERE IdRez = RC.IdRez
ORDER BY IdRez, Datum ASC
;
WITH RECURSIVE U AS
SELECT IdRez, Datum, Iznos
FROM Uplata
WHERE IdRez = RC.IdRez AND
(Datum < Uplata.Datum OR (Datum = Uplata.Datum AND OtIdUpl < Uplata.OtIdUpl))
;
WITH RECURSIVE C AS
SELECT IdRez, COALESCE(SUM(Iznos) OVER (PARTITION BY IdRez ORDER BY Datum ASC), 0) AS SumIznos
FROM Rezervacija LEFT JOIN PomoćniPse USING (IdRez) LEFT JOIN Kupon USING (IdKup)
ORDER BY IdRez
;
WITH RECURSIVE U AS
SELECT IdRez, Datum, Iznos
FROM Uplata
WHERE IdRez = RC.IdRez AND
(Datum < Uplata.Datum OR (Datum = Uplata.Datum AND OtIdUpl < Uplata.OtIdUpl))
;
WITH RECURSIVE U AS
SELECT IdRez, Datum, Iznos
FROM Uplata
WHERE IdRez = RC.IdRez AND
(Datum < Uplata.Datum OR (Datum = Uplata.Datum AND OtIdUpl < Uplata.OtIdUpl))
;
WITH RECURSIVE U AS
SELECT IdRez, Datum, Iznos
FROM Uplata
ORDER BY IdRez, Datum ASC
```

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 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 (DatumOn = Datum OR Datum = DatumOn))
        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 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.

```
[REDACTED SQL CODE]
```
