



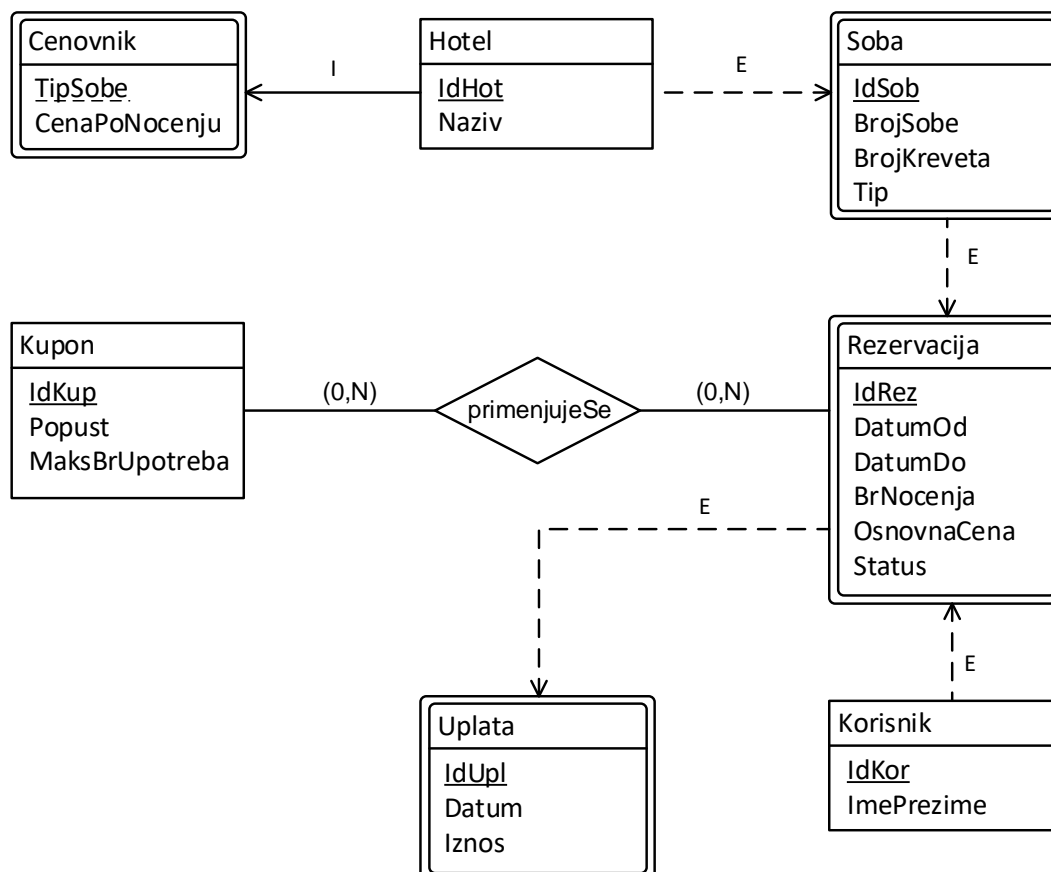
Базе података 1 (13С112БП1)

- 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	- 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: <i>уууу-мм-дд</i> , обавезно
DatumDo	- низ од тачно 10 знакова, датум у формату: <i>уууу-мм-дд</i> , 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: <i>уууу-мм-дд</i> , 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 купоне који нису коришћени, а затим испишује све купоне 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 1

```
DELETE FROM Kuponi
WHERE IdKup NOT IN (SELECT IdKup FROM primenjene);

UPDATE Kuponi SET Popust = 10 WHEN Popust < 10 THEN 10;
UPDATE Kuponi SET Popust = 25 WHEN Popust >= 10 AND Popust < 25 THEN 25;
UPDATE Kuponi SET Popust = 25 WHEN Popust >= 25 THEN 25;

ORDER BY IdKup ASC;
FROM Kuponi;
WHERE IdKup NOT IN (SELECT IdKup FROM primenjene);
```

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 Korisnici
WHERE IdKor NOT IN (
    SELECT IdKor
    FROM Rezervacije
    WHERE IdKor NOT IN (
        SELECT IdKor
        FROM primenjene
    )
);
```

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 Сactus-у користити таб: Zadatak 3

```
[REDACTED SQL CODE]
```

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 economicnih rezervacija, Broj standardnih rezervacija, Broj luksuznih rezervacija
У Сactus-у користити таб: Zadatak 4

```
[REDACTED SQL CODE]
```

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 (
    IdSoba INT(11) PRIMARY KEY AUTOINCREMENT,
    ImeSoba VARCHAR(255) NOT NULL,
    BrojSoba INT(11) NOT NULL CHECK (BrojSoba > 0),
    BrojKor INT(11) NOT NULL REFERENCES Korisnik,
    BrojHotela INT(11) NOT NULL REFERENCES Hoteli
);
```

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, ИмеPrezime, BrPosecenihHotela

У Cactus-у користити таб: Zadatak 6

Није дозвољено коришћење погледа.

```
WITH Korisnik AS (
    SELECT IdKor, ИмеPrezime, BrPosecenihHotela AS BrPoseceniHoteli
    FROM Korisnik JOIN Rezervacija ON Korisnik.IdKor = Rezervacija.IdKor
)
SELECT IdKor, ИмеPrezime, BrPoseceniHoteli AS BrPoseceniHoteli
FROM Korisnik JOIN Korisnik ON Korisnik.IdKor = Korisnik.IdKor
ORDER BY BrPoseceniHoteli DESC;
```

Задатак 7 [5 поена]

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

Резултат дати у форми: IdHot, Naziv, ProsečnaCenaPoNoci

У Sactus-у користити таб: Zadatak 7

Није дозвољено коришћење погледа.

```
SELECT IdHot, Naziv, SUM(OsobnaCena)/SUM(BrojNoci) AS ProsečnaCenaPoNoci
FROM Hotel JOIN Sactus.USING(IdHot) JOIN Rezervacija USING(Sactus)
WHERE ProsečnaCenaPoNoci > (SELECT AVG(ProsečnaCenaPoNoci)
FROM Hotel JOIN Sactus.USING(IdHot) JOIN Rezervacija USING(Sactus))
ORDER BY ProsečnaCenaPoNoci ASC, IdHot ASC
```

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 Сactus-у користити таб: Zadatak 8

Није дозвољено коришћење додатних структура – погледа или помоћних табела.

```
[REDACTED SQL CODE]
```

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, БројАктуелнихRezervacija

У Sactus-у користити таб: Zadatak 9

Није дозвољено коришћење погледа.

```
WITH RECURSIVE DATES(Datum) AS
(
    SELECT DATE('2022-01-01')
    UNION ALL
    SELECT DATE(Datum + 1)
    FROM DATES
    WHERE Datum < DATE('2022-12-31')
)
SELECT D.Datum, COUNT(R.ID) AS BrojAktuelnihRezervacija
FROM DATES D
LEFT JOIN RESERVACIJE R
ON D.Datum BETWEEN R.DatumPočetno AND R.DatumKrajnje
GROUP BY D.Datum
ORDER BY D.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.

```
SELECT IdSob, DatumOd, DatumDo, BrojRealizovanih
```

```
FROM
```

```
  (SELECT IdSob, DatumOd, DatumDo, BrojRealizovanih
```

```
    FROM IntervalRezervacija
```

```
   WHERE IdSob = 1
```

```
   AND DatumOd < DatumDo
```

```
   ORDER BY IdSob, DatumOd, DatumDo, BrojRealizovanih
```

```
    LIMIT 1000000
```

```
   GROUP BY IdSob, DatumOd, DatumDo
```

```
   HAVING COUNT(DISTINCT DatumDo) > 1
```

```
UNION ALL
```

```
SELECT IdSob, DatumOd, DatumDo, BrojRealizovanih
```

```
  FROM IntervalRezervacija
```

```
 WHERE IdSob = 1
```

```
   AND DatumOd < DatumDo
```

```
   ORDER BY IdSob, DatumOd, DatumDo, BrojRealizovanih
```

```
    LIMIT 1000000
```

```
   GROUP BY IdSob, DatumOd, DatumDo
```

```
UNION ALL
```

```
SELECT IdSob, DatumOd, DatumDo, BrojRealizovanih
```

```
  FROM IntervalRezervacija
```

```
 WHERE IdSob = 1
```

```
   AND DatumOd < DatumDo
```