

Базе података 1

– други колоквијум –

Број индекса (гггг/бббб), име и презиме	Потпис дежурног	Број поена	

Напомена: Није дозвољена употреба литературе. Колоквијум траје **90** минута.

1. (5) Дат је садржај дела базе података са смештајима и њиховим резервацијама. У табелама *Smestaj* и *Rezervacija* се прате подаци за сваки смештај и резервацију за сваки од њих.

Smestaj				
IdS	Parking	Cena	Osoba	Naziv
1	N	600	2	Oasis
2	B	500	2	Panorama
3	D	100	3	Avenue
4	B	200	4	7Seasons

Rezervacija				
IdR	IdS	Prijava	Odjava	Ocena
1	1	20190302	20190312	3.5
2	2	20190110	20190115	4.5
3	1	20190312	20190315	5
4	2	20190204	20190207	4
5	3	20190204	20190208	NULL

а) Написати *SQL* упит за прављење табеле *Smestaj*. *IdS* је целобројна величина која идентификује смештај, *Parking* представља карактер и обавезно је поље (вредности су "N"-нема паркинга, "B" – бесплатан паркинг, "D" – доплаћује се паркинг), *Cena* је целобројна величина и обавезна је, *Osoba* је цео број, представља број особа који може да стане у смештај и обавезно је поље, *Naziv* је низ до 50 карактера и није обавезан. Остала ограничења треба игнорисати.

```
CREATE TABLE Smestaj (IdS INTEGER PRIMARY KEY,
Parking CHAR NOT NULL,
Cena INTEGER NOT NULL,
Osoba INTEGER NOT NULL,
Naziv VARCHAR(50))
```

б) Одлучено је да се обришу сви смештаји код којих се паркинг доплаћује. Написати *SQL* упит које реализује брисање описаних смештаја. Сматрати да су резервације за њих већ обрисане.

```
DELETE FROM Smestaj
WHERE Parking = 'D'
```

в) Смештај Oasis је купио паркинг место и решио је да га понуди без доплате својим гостима. Написати *SQL* упит за промену информације о паркингу за *IdS* једнако 1.

```
UPDATE Parking
SET Parking = 'B'
WHERE IdS = 1
```

г) Гост је одлучио да са својом женом и дететом преспава пет ноћи у смештају ”7Seasons” почевши од 1. фебруара 2019. године. па је резервисао смештај *idS* једнако 4. Датум одјаве је дан након последње ноћи, датум пријаве је дан првог ноћења. Написати *SQL* упит који додаје наведену резервацију у одговарајућу табелу.

```
INSERT INTO Rezervacija (IdR, IdS, Prijava, Odjava) VALUES (6, 4, 20190201, 20190206)
```

д) Написати *SQL* упит за приказ оцењених резервација (оцена није NULL) из 2019. године. Приказ резултата треба да буде у формату: *IdR, IdS, Prijava, Odjava, Ocena*.

```
SELECT *
FROM Rezervacija
WHERE Ocena IS NOT NULL AND Prijava LIKE '2019 ____' AND Odjava LIKE '2019 ____'
```

ђ) Написати *SQL* упит за приказ просечне цене смештаја који може угостити 2 особе и има паркинг. Приказ резултата треба да буде у формату: *Prosek*.

```
SELECT AVG(Cena) AS Prosek
FROM Smestaj
WHERE Parking <> 'N' AND Osoba > 2
```

е) Потребно је написати *SQL* упит која дохвата све резервације. У резултату прво треба исписати резервације за смештаје који немају паркинг, затим за смештаје којима се доплаћује паркинг и на крају за смештаје којима је бесплатан паркинг. У оквиру те три групе резервације треба сортирати неопдајуће по идентификатору смештаја, па неопдајуће по времену пријаве, па неопдајуће по времену одјаве и на крају неопдајуће по идентификатору резервације. Приказ резултата треба да буде у формату: *Parking, IdS, Prijava, Odjava, IdR*.

```
SELECT S.Parking AS "Parking", S.IdS AS "IdS", R.Prijava AS "Prijava", R.Odjava AS
"Odjava", R.IdR AS "IdR"
FROM Rezervacija R, Smestaj S
WHERE R.IdS = S.IdS
ORDER BY S.Parking DESC, S.IdS ASC, R.Prijava ASC, R.Odjava ASC, R.IdR
```

ж) Потребно је написати *SQL* скрипту која прави поглед (*VIEW*) *Praznici* који као приказ даје оне резервације које су у почеле и завршиле се у периоду од 1. децембра до 31. јануара. Сматрати да резервација не може да буде дужа од 30 дана. Искористи поглед *Praznici* како би се за смештаје који су имали ресервације у том периоду приказао број резервација у том периоду. Приказ резултата треба да буде у формату: *IdS, Broj*.

```
CREATE VIEW Praznici AS
SELECT *
FROM Rezervacija
WHERE (Prijava LIKE '__12__' OR Prijava LIKE '__01__') AND (Odjava LIKE '__12__' OR
Odjava LIKE '__01__');

SELECT IdS, COUNT(*) AS Broj
FROM Praznici P
GROUP BY IdS;
```

з) Потребно је написати *SQL* упит који за сваки смештај који има више од 10 резервација исписује његову просечну оцену, као и број добијених оцена и број резервација. Приказ резултата треба да буде у формату: *IdS, Naziv smestaja, Ocena, Broj ocena, Broj rezervacija*. Одређене колоне морају имати размаке у називу.

```
SELECT S.IdS AS IdS, S.Naziv AS "Naziv smestaja", AVG(Ocena) AS Ocena,
COUNT(Ocena) AS "Broj ocena", COUNT(*) AS "Broj rezervacija"
FROM Smestaj S, Rezervacija R
WHERE S.IdS = R.IdS
GROUP BY S.IdS
HAVING COUNT(*) > 10
```

2.(10) Дати су шема релације $R(A, B, C, D, E, F, G, H)$ и скуп функцијских зависности $F = \{CE \rightarrow GA, G \rightarrow FH, HD \rightarrow B, F \rightarrow CGA\}$. Потребно је:

а) Одредити скуп кандидат кључева КК дате шеме.

Одговор:

$KK = \{CED, FED, GED\}$

б) Испитати редом да ли је дата шема у BC, 3. и 2. нормалној форми и сваки пут у табели назначити да ли посматрана зависност нарушава посматрану нормалну форму.

	$CE \rightarrow GA$	$G \rightarrow FH$	$HD \rightarrow B$	$F \rightarrow CGA$
BCNF	x	x	x	x
3NF	x	x	x	x
2NF	x	x	√	x

в) Спровести нормализацију дате шеме у 3. нормалну форму алгоритмом који гарантује очување функцијских зависности.

Одговор:

$R_1(\underline{C}, \underline{E}, G) \quad R_2(\underline{G}, F, H) \quad R_3(\underline{H}, \underline{D}, B) \quad R_4(\underline{E}, C, G, A) \quad R_5(\underline{C}, \underline{E}, \underline{D})$

г) Спровести нормализацију дате шеме у BC нормалну форму, издвајајући зависности редоследом са десна на лево.

Одговор:

$R_1(\underline{E}, C, G, A) \quad R_2(\underline{H}, \underline{D}, B) \quad R_3(\underline{E}, H) \quad R_4(\underline{D}, \underline{E}, F)$

д) Испитати да ли је при поступку у оквиру тачке г) дошло до суштинских губитка функцијских зависности и којих?

Одговор:

$CE \rightarrow GA$