

# 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 ски аранжманима једне агенције. У табелама *Skijaliste* и *Termin* се прате подаци о скијалиштима која се налазе у понуди и организованим терминима .

Skijaliste			
IdSki	Naziv	BrStaza	Osnezenje
1	Kopaonik	25	D
2	Zlatibor	4	N
3	Les 2 Alpes	96	D
4	Paradiski	253	D
5	Chatel	49	NULL
6	Bansko	18	D

Termin				
IdTer	Od	Do	IdSki	Cena
1	20181229	20190105	1	350
2	20190105	20190112	1	230
3	20190126	20190202	6	165
4	20190125	20190203	3	396
5	20190201	20190210	3	410
6	20190308	20190317	3	379

а) Написати *SQL* упит за прављење табеле *Skijaliste*. *IdSki* је целобројна величина која идентификује скијалиште, *Naziv* представља низ до 45 карактера и обавезно је, *BrStaza* је целобројна величина, *Osnezenje* је карактер ('D' - има оснежење; 'N' - нема оснежење) . Остала ограничења треба игнорисати.

```
CREATE TABLE Skijaliste (IdSki INTEGER PRIMARY KEY,
Naziv VARCHAR(45) NOT NULL,
BrStaza INTEGER,
Osnezenje CHAR(1))
```

б) Одлучено је да се укину сви термини који су скупљи од 400. Написати *SQL* упит које реализују брисање описаних термина.

```
DELETE FROM Termin
WHERE Cena >400
```

в) Уведено је вештачко оснежење на Златибору. Написати *SQL* упит за промену информације о оснежењу за *IdSki* једнако 2.

```
UPDATE Skijaliste  
SET Osnezenje = 'D'  
WHERE IdSki = 2
```

г) Одлучено је да се уведе нови термин за Срећење на Копаоник. Датум поласка је предвиђен за 13. фебруар 2019., датум повратка за 17. фебруар 2019., *idTer* једнако 7 и *Cenom* од 170. Написати *SQL* упит који додаје наведени термин у одговарајућу табелу.

```
INSERT INTO Termin  
VALUES ( 7, 20190213, 20190217, 1, 170)
```

д) Написати *SQL* упит за приказ скијалишта којима није позната информација о оснежењу (*NULL* вредност). Приказ резултата треба да буде у формату: *IdSki, Naziv Skijalista* (називи колоне треба да имају више речи).

```
SELECT IdSki, Naziv AS "Naziv Skijalista"  
FROM Skijaliste  
WHERE Osnezenje IS NULL
```

ђ) Написати *SQL* упит за приказ просечног броја стаза на свим скијалиштима која поседују вештачко оснежење и имају више од 10 стаза. Приказ резултата треба да буде у формату: *Prosek*.

```
SELECT AVG(BrStaza) AS Prosek  
FROM Skijaliste  
WHERE Osnezenje = 'D' AND BrStaza > 10
```

е) Потребно је написати *SQL* упит која дохвати све термине који се односе на скијалишта са више од 15 стаза. Резултат треба сортирати прво растуће по датуму поласка, а после опадајући по броју стаза. Приказ резултата треба да буде у формату: *Naziv, BrStaza, Od, Do, Cena*

```
SELECT Naziv, BrStaza, Od, Do, Cena  
FROM Termin T, Skijaliste S  
WHERE T.IdSki = S.IdSki AND BrStaza > 15  
ORDER BY Od, BrStaza DESC
```

ж) Потребно је написати *SQL* скрипту која прави поглед (*VIEW*) *TerminUMartu* који као приказ даје оне термине који су у потпуности у марту 2019. Искористи поглед *TerminUMartu* за приказ свих јефтиних термина у марту. Термин је јефтин ако је његова цена нижа од 200. Приказ резултата треба да буде у формату: *IdTer, Od, Do, Cena*

```
CREATE VIEW TerminUMartu AS
SELECT *
FROM Termin
WHERE Od >= 20190301 AND Do < 20190401;
```

```
SELECT IdTer, Od, Do, Cena
FROM TerminUMartu
WHERE Cena < 200;
```

з) Потребно је написати *SQL* упит која дохвата све датуме када је било 2 или више поласка. Приказ резултата треба да буде у формату: *Datum*

```
SELECT Od AS Datum
FROM Termin
GROUP BY Od
HAVING COUNT(*) >= 2
```

2.(10) Дати су шема релације  $R(A, B, C, D, E, F, G, H)$  и скуп функцијских зависности  $F = \{ AFG \rightarrow H, E \rightarrow DF, FG \rightarrow AG, C \rightarrow AF, CD \rightarrow EFG \}$ . Потребно је:

а) Одредити скуп кандидат кључева  $KK$  дате шеме.

Одговор:

$KK = \{ BCD, BCE \}$

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.

	$AFG \rightarrow H$	$E \rightarrow DF$	$FG \rightarrow AG$	$C \rightarrow AF$	$CD \rightarrow EFG$
BCNF	x	x	x	x	x
3NF	x	x	x	x	x
2NF	√	x	√	x	x

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

Одговор:

$R_1(G, F, A, H) R_2(E, D, F) R_3(C, A, F) R_4(C, D, E, G) R_5(B, C, D)$   
iii

$R_1(G, F, A, H) R_2(E, D, F) R_3(C, A, F) R_4(C, D, E, G) R_5(B, C, E)$

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

Одговор:

$R_1(C, F) R_2(E, D) R_3(C, E, G) R_4(C, A) R_5(C, D, H) R_6(C, D, B)$

iii

$R_1(E, F, D) R_2(C, E, G) R_3(C, A) R_4(C, D, H) R_5(C, D, B)$

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

Одговор:

Дошло је до губитка функцијских зависности:

$AFG \rightarrow H, E \rightarrow F, FG \rightarrow A, CD \rightarrow EG$

iii

$AFG \rightarrow H, FG \rightarrow A, C \rightarrow F, CD \rightarrow EFG$