

Базе података 1

– надокнада другог колоквијума –

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 о летовима и картама за летове. У табелама *Let* и *Karta* се прате подаци о прошлим, садашњим и будућим летовима као и подаци о картама за те летове. У табели *Karta* атрибут статус има значење: К – карта је купљена, N – карта није купљена.

Let					
IdLet	MestoOd	MestoDo	Poletanje	Status	PotrGoriva
1	Beograd	Rim	20190125	T	0
2	Rim	Pariz	20190127	K	0
3	Beograd	Berlin	20180305	O	0
4	Beograd	NYC	NULL	K	0
5	Rim	Istanbul	20190128	K	0
6	Istanbul	NYC	20180201	Z	5000

Karta			
IdKar	IdLet	Cena	Status
1	1	200	K
2	1	300	N
3	1	100	K
4	2	400	K
5	2	500	K
6	3	700	K

а) Написати *SQL* исказ за прављење табеле *Let*. *IdLet* је целобројна величина која идентификује лет, *MestoOd* и *MestoDo* су низови до 45 карактера и обавезни су, *Poletanje* је целобројна величина, *Status* је карактер ('T' – у току; 'K' - креиран; 'Z' - завршен; 'O' - отказан) и обавезан је, *PotrGoriva* је ненегативан цео број, обавезан је и подразумевана вредност је 0. Остала ограничења треба игнорисати.

б) Одлучено је да се укину сви летови из места „Rim“. Написати *SQL* упит који реализује брисање описаних летова.

в) Дефинисана је максимална цена 450 за карту. Написати *SQL* упит који регулише цену карата које нису купљене тако да цене веће од максималне поставља на максималну.

г) Одлучено је да се у продају пусте још две карте за лет 3. Прва карта има *IdKar 7*, а друга 8. Цена прве је 450, друге 400. Написати један *SQL* исказ који ће креирати обе карте.

д) Написати *SQL* упит за приказ летова којима је позната информација о полетању (није *NULL* вредност). Приказ резултата треба да буде у формату: *IdLet, Mesto poletanja, Mesto sletanja* (називи колоне треба да имају више речи).

ђ) Написати *SQL* упит за приказ укупне и просечне количине потрошеног горива на завршеним летовима из места *Istanbul*. Приказ треба да буде у формату *Ukupno, Prosek* и сортиран опадајуће по *Ukupno* и опадајуће по *Prosek*.

е) Потребно је написати *SQL* упит који за сваки лет који није отказан исписује суму зарађеног новца од продаје карата. Приказ резултата треба да буде у формату: *IdLet, MestoOd, MestoDo, Zaradjeno*.

ж) Потребно је написати *SQL* скрипту која прави поглед (*VIEW*) *KupljeneKarteZaJanuar* који као приказ даје оне карте које су купљене за летове чије је полетање у јануару 2019. Искористити поглед *KupljeneKarteZaJanuar* за приказ просечне цене скуних карата у јануару 2019. Карта је скупа ако јој је цена већа од 500. Приказ резултата треба да буде у формату: *Prosek*.

з) Потребно је написати *SQL* упит који за сваки град посебно, који је од карата на долазним неотказаним летовима у првом кварталу 2018. године зарадио више од 50000 од продаје карата и продао више од 100 карата, исписује назив тог града, минималну и максималну цену продате карте у формату *MestoDo, MinCena, MaxCena*.

2.(10) Дати су шема релације $R(S, T, E, F, A, N)$ и скуп функцијских зависности $F=\{SN\rightarrow A, FT\rightarrow N, TS\rightarrow EA, EF\rightarrow ST\}$. Потребно је:

а) Одредити скуп кандидат кључева КК дате шеме.

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

	$SN\rightarrow A$	$FT\rightarrow N$	$TS\rightarrow EA$	$EF\rightarrow ST$
BCNF				
3NF				
2NF				

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

г) Спровести нормализацију дате шеме у BC нормалну форму, издвајајући зависности редоследом слева на десно.

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