

# Базе података 1

– SQL Увод –

## Задатак 1

Дат је садржај дела базе података телевизијске станице. У табелама *Program* и *Voditelj* се прате подаци о програмској шеми и водитељима који су ангажовани у датим емисијама.

Program					
Naziv	Datum	Od	SifV	Gledanost	Tip
Vesti	20180122	1600	NULL	10	Info
Žikina šarenica	20180122	800	1	65	Mix
Munje	20180121	2000	NULL	42	Film
SAT	20180122	1500	2	24	Auto
Lavirint	20180120	2100	3	22	Kviz
Visok napon	20180125	2045	3	17	Kviz

Voditelj		
SifV	Ime	Staz
1	Žika Nikolić	34
2	Mirko Alvirović	32
3	Dejan Pantelić	18
4	Jovan Memedović	22
5	Jelena Obućina	10
6	Mladen Alvirović	11

а) Написати *SQL* упит за прављење табеле *Voditelj*. *SifV* је целобројна величина која идентификује водитеља, *Ime* представља низ до 40 карактера, *Staz* је целобројна величина. Сва поља су обавезна. Остала ограничења треба игнорисати.

```
CREATE TABLE Voditelj (SifV INTEGER PRIMARY KEY,
Ime VARCHAR(40) NOT NULL,
Staz INTEGER NOT NULL)
```

б) Потребно је написати *SQL* скрипту која прави поглед (*VIEW*) *InfoJutro* који као приказ даје оне емисије које су информативног типа (*Info*), а при том почињу између 6:00 и 10:00. Искористи поглед *InfoJutro* за приказ само оних водитеља (без понављања) који имају стаж мањи од 30 и воде јутарње информативне емисије. Приказ резултата треба да буде у формату: *SifV*, *Ime*.

```
CREATE VIEW InfoJutro AS
SELECT *
FROM Program
WHERE Od BETWEEN 600 AND 1000 AND Tip='Info';
```

```
SELECT DISTINCT SifV, Ime
FROM InfoJutro, Voditelj
WHERE InfoJutro.SifV = Voditelj.SifV AND Staz < 30;
```

в) Написати *SQL* упит за приказ емисија који немају водитеља (*NULL* вредност). Користити само табелу *Program*. Приказ резултата треба да буде у формату: *Naziv, Datum, Tip*.

```
SELECT Naziv, Datum, Tip FROM Program WHERE SifV IS NULL
```

г) Одлучено је да се укину квизови који имају мању гледаност од 20. Написати *SQL* упит (или упите) које реализују брисање описаних емисија.

```
DELETE FROM Program  
WHERE Gledanost < 20 AND Tip='Kviz'
```

д) Водитељ Мирко Алвировић препушта своју емисију (SAT) да води његов син Младен Алвировић (*SifV=6*). Написати *SQL* упит за промену водитеља емисије и при том за гледаност емисије поставити вредност 0.

```
UPDATE Program  
SET SifV=6, Gledanost=0  
WHERE Naziv = 'SAT'
```

ђ) Директор телевизијске станице захтева информацију о просечној гледаности филмова који су били у јануару у 2018 години. Написати *SQL* упит који враћа као резултат претходни захтев. Приказ резултата треба да буде у формату: *FilmJan2018*.

```
SELECT AVG (Gledanost) AS FilmJan  
FROM Program  
WHERE Tip = 'Film' AND Datum/100 = 201801
```

е) У програмску шему потребно је додати нову емисију *Око* која је информативног типа и на којој ће бити ангажован водитељ са шифром 6. Датум, време и гледаност су непознати. Потребно је написати *SQL* упит који додаје нову емисију.

```
INSERT INTO Program (Naziv, SifV, Tip) VALUES ('Oko', 6, 'Info')
```

ж) Потребно је написати SQL упит који за све филмове даје скраћен приказ који треба да буде у формату : Naziv filma, Datum, Vreme pocetka. (Називи колона треба да имају више речи).

```
SELECT Naziv AS "Naziv filma", Datum, Od AS "Vreme pocetka"  
FROM Program  
WHERE Tip = 'Film'
```

з) Потребно је написати SQL упит за приказ емисија и водитеља које их воде. Резултат треба сортирати опадајуће прво по датуму, па по времену почетка, а после растуће по гледаности. Приказ резултата треба да буде у формату: Datum, Vreme, Naziv, ImeVoditelja, Gledanost.

```
SELECT Datum, Od AS Vreme, Naziv, Ime AS ImeVoditelja, Gledanost  
FROM Program, Voditelj  
WHERE Program.SifV = Voditelj.SifV  
ORDER BY Datum DESC, Od DESC, Gledanost
```

и) Потребно је приказати датуме из програмске шеме када је било тачно два квиза после 18:00. Приказ резултата треба да буде у формату: *Datum*.

```
SELECT Datum  
FROM Program  
WHERE Tip = 'Kviz' AND Od > 1800  
GROUP BY Datum  
HAVING COUNT(*) = 2
```

## Задатак 2

Дат је садржај дела базе података једне болнице. У табелама *Pacijent* и *Soba* прате се подаци о пацијентима који се налазе у датим собама.

Pacijent				
Karton	ImeIPrezime	Osiguranje	Starost	BrSobe
1	Pera Perić	FALSE	37	1
2	Mika Mikić	NULL	26	1
3	Žika Žikić	TRUE	63	2
4	Miša Mišić	NULL	17	3
5	Sima Simić	TRUE	12	4
6	Laza Lazić	FALSE	54	4

Soba			
Broj	Sprat	Kategorija	BrKreveta
1	1	A	5
2	1	A	4
3	1	B	8
4	2	B	8
5	2	C	10

а) Написати *SQL* упит или упите за прављење табеле *Soba*. *Broj* је целобројна величина која идентификује собу, *Sprat* је целобројна величина која означава спрат собе, *Kategorija* је знаковни тип који представља категорију собе, *BrKreveta* је целобројна величина која означава колико се кревета налази у соби. Сва поља су обавезна. Остала ограничења треба игнорисати.

```
CREATE TABLE Soba (Broj INTEGER PRIMARY KEY,
Sprat INTEGER NOT NULL,
Kategorija CHAR(1) NOT NULL,
BrKreveta INTEGER NOT NULL)
```

б) Пацијент са бројем картона 3 је преминуо. Написати *SQL* упит или упите за брисање пацијента са картоном 3.

```
DELETE FROM Pacijent WHERE Karton = 3
```

в) Пацијент који је преминуо је био доктор ове болнице, па је управа одлучила да његову канцеларију, која се налази на трећем спрату, преуреди и направи од ње нову собу 6 (без кревета) категорије А. Написати *SQL* упит или упите за креирање описане собе.

```
INSERT INTO Soba (Broj, Sprat, Kategorija, BrKreveta) VALUES ( 6, 3, 'A', 0)
```

г) Донатори болнице су одлучили да опреме претходно направљену собу са 4 кревета. Написати *SQL* упит или упите за промену броја кревета.

```
UPDATE Soba
SET BrKreveta = 4
WHERE Broj = 6
```

д) Управи болнице је потребна информација о бројевима картона пацијената за које се не зна да ли имају или немају осигурање (NULL вредност). Написати SQL упит или упите за формирање погледа (VIEW) који омогућава приказ тражених податка.

```
CREATE VIEW PacijentiNepoznatoOsiguranje AS
SELECT Karton
FROM Pacijent
WHERE Osiguranje IS NULL
```

ђ) Потребно је написати SQL упит који за сваког пацијената проналази категорију собе у којој се налази. Приказ резултата треба да буде у формату : Karton, ImeIPrezime, Kategorija.

```
SELECT Karton, ImeIPrezime, Kategorija
FROM Pacijent, Soba
WHERE BrSobe = Broj
```

е) Директор болнице захтева информацију о укупном броју кревета у болници. Потребно је написати SQL упит који даје као резултат укупан број кревета у болници. Приказ резултата треба да буде у формату: Ukupan broj kreveta. (Назив резултата треба да има више речи).

```
SELECT SUM (BrKreveta) AS "Ukupan broj kreveta"
FROM Soba
```

ж) Потребно је написати SQL упит који даје приказ соба које имају више од 4 кревета, а нису категорије А. Резултат треба сортирати опадајуће прво по спрату собе, а после растуће по категорији. Приказ резултата треба да буде у формату: Broj, Sprat, Kategorija, BrKreveta.

```
SELECT * FROM Soba
WHERE BrKreveta > 4 AND Kategorija != 'A'
ORDER BY Sprat DESC, Kategorija
```

з) Потребно је написати SQL упит који даје приказ колико укупно има кревета за сваки спрат и сваку категорију на том спрату. Приказ резултата треба да буде у формату: Sprat, Kategorija, Broj

```
SELECT Sprat, Kategorija, SUM (BrKreveta) AS Broj
FROM Soba
GROUP BY Sprat, Kategorija
```

и) Потребно је написати SQL скрипту која прави поглед (VIEW) Deca који као приказ даје пацијенте који су деца (старост мања од 18). Искористи претходно направљен поглед за приказ само оних соба (без понављања) у којима се налази бар једно дете. Приказ резултата треба да буде у формату: Broj, Sprat.

```
CREATE VIEW Deca AS  
SELECT * FROM Pacijent WHERE Starost < 18;
```

```
SELECT DISTINCT Broj, Sprat  
FROM Soba, Deca  
WHERE BrSobe = Broj;
```

### Задатак 3

1. Дат је садржај дела базе података градског саобраћајног предузећа. У табелама Аутобус и Линија прате се подаци о аутобусима који су ангажовани на датим линијама.

Autobus			
Registracija	Status	Kapacitet	Linija
BG-262-AA	garaza	86	16
BG-894-BJ	NULL	96	43
BG-732-PM	garaza	120	84
BG-122-XX	voznja	100	53
BG-342-SC	voznja	90	84
BG-459-KA	NULL	63	43

Linija			
Broj	Od	Do	BrStanica
16	Pohorska	Karaburma	22
43	Kotež	Novi Beograd	17
53	Vidkovac	Zeleni Venac	24
83	Zemun	Crveni Krst	27
84	Zeleni Venac	Nova Galenika	19

а) Написати *SQL* упит или упите за прављење табеле *Linija*. *Broj* је целобројна величина која идентификује линију, *Og/Do* представља назив почетне/последње станице која има до 25 карактера, *BrStanica* је целобројна величина. Сва поља осим *BrStanica* су обавезна. Остала ограничења треба игнорисати.

```
CREATE TABLE Linija (
    Broj INTEGER PRIMARY KEY,
    Od VARCHAR(25) NOT NULL,
    Do VARCHAR(25) NOT NULL,
    BrStanica INTEGER)
```

б) Потребно је написати *SQL* скрипту која прави поглед (*VIEW*) *VelikiAutobusi* који као приказ даје оне аутобусе који имају капацитет преко 100 путника. Искористи поглед *VelikiAutobusi* за приказ само оних линија (без понављања) на којима је ангажован бар један велики аутобус. Приказ резултата треба да буде у формату: *Broj, Od, Do*.

```
CREATE VIEW VelikiAutobusi AS
SELECT * FROM Autobus WHERE Kapacitet > 100;
```

```
SELECT DISTINCT Broj, Od, Do
FROM Linija, VelikiAutobusi
WHERE Linija = Broj;
```

в) Потребно је написати *SQL* упит који за сваки аутобус проналази почетну и последњу станицу линије на којој је ангажован. Приказ резултата треба да буде у формату : *Registracija vozila, Pocetna stanica, Poslednja stanica*. (Називи колона треба да имају више речи).

```
SELECT Registracija AS "Registracija vozila",
    Od AS "Pocetna stanica", Do AS "Poslednja stanica"
FROM Linija, Autobus
WHERE Linija = Broj
```

г) Аутобуси са регистарским таблицама *BG-342-SC* и *BG-122-XX* су се директно сударили са 100% штетом (на срећу без теже повређених). Написати *SQL* упит за брисање ових аутобуса из евиденције.

```
DELETE FROM Autobus WHERE Registracija='BG-342-SC' OR Registracija='BG-122-XX'
```

д) Због претходног судара, сви аутобуси који су у гаражи и при том су ангажовани на линијама 84 и 53 морали су да се пуне у вожњу. Написати *SQL* упит за новонастало стање.

```
UPDATE Autobus  
SET Status = 'voznja'  
WHERE (Linija = 84 OR Linija = 53) AND Status = 'garaza'
```

ђ) Једна источна земља је понудила донацију у виду два нова аутобуса са капацитетима од по 110 путника са регистарским таблицама *BG-987-JA* и *BG-142-TS*. Аутобус са регистарским таблицама *BG-987-JA* би требало да се ангажује на линији 53, док за аутобус са регистарским таблицама *BG-142-TS* не зна где ће бити ангажован. Оба аутобуса се налазе у гаражи. Написати *SQL* упит за додавање оба аутобуса у одговарајућу табелу.

```
INSERT INTO Autobus (Registracija, Status, Kapacitet, Linija)  
VALUES ('BG-987-JA', 'garaza', 110, 53),  
('BG-142-TS', 'garaza', 110, NULL)
```

е) Директор предузећа захтева информацију о просечном капацитету својих возила на линији 16. Потребно је написати *SQL* упит који даје као резултат потребне податке. Приказ резултата треба да буде у формату: *Prosek16*.

```
SELECT AVG(Kapacitet) AS Prosek16  
FROM Autobus  
WHERE Linija = 16
```

ж) Возачи аутобуса нису пријавили диспечеру статус аутобуса. Таквим аутобусима статус је непознат (*NULL* вредност). Написати *SQL* упит или упите за приказ регистрација аутобуса код којих је непознат статус.

```
SELECT Registracija  
FROM Autobus  
WHERE Status IS NULL
```



з) Потребно је написати *SQL* упит (или упите) за приказ линија чија је почетна или последња станица Зелени Венац, а при том је број станица већи од 20. Резултат треба сортирати растуће прво по броју линије, а после опадајуће по броју станица. Приказ резултата треба да буде у формату: *Broj, Od, Do, BrStanica*.

```
SELECT Broj, Od, Do, BrStanica  
FROM Linija  
WHERE (Od = 'Zeleni Venac' OR Do = 'Zeleni Venac') AND BrStanica > 20  
ORDER BY Broj, BrStanica DESC
```

и) Потребно је написати *SQL* упит (или упите) за приказ бројева линија на којима је капацитет аутобуса који су у статусу вожње већи од 150. Приказ резултата треба да буде у формату: *Linija*.

```
SELECT Linija  
FROM Autobus  
WHERE Status = 'voznja'  
GROUP BY Linija  
HAVING SUM(Kapacitet) > 150
```