



Базе података 2

– Релациона алгебра –

– Решења задатака –

Задатак 1

Дат је садржај дела базе података градског саобраћајног предузећа. У табелама **Autobus** и **Linija** 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	Vidikovac	Zeleni Venac	24
83	Zemun	Crveni Krst	27
84	Zeleni Venac	Nova Galenika	19

- а) Возачи аутобуса нису пријавили диспечеру статус аутобуса. Таквим аутобусима статус је непознат (NULL вредност). Написати упит за приказ регистрација аутобуса код којих је непознат статус.

π Registracija (σ Status=NULL (Autobus))

- б) Потребно је написати 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 по броју линије. Приказ резултата треба да буде у формату: Broj, Od, Do, BrStanica.

τ Broj (π Broj, Od, Do (σ (Od='Zeleni Venac' Do='Zeleni Venac') \wedge BrStanica>20 (Linija)))

- в) Једна острвска, источна земља је понудила донацију у виду новог аутобуса са капацитетом од 110 путника са регистарским таблицама BG-987-JA. Аутобус би требао да се ангажује на линији 53, а за сада се налази у гаражи. Написати упит (или упите) за додавање новог аутобуса (формирати нову табелу са придодатим наведеним аутобусом).

Autobus U π 'BG-987-JA' -> Registracija, 'garaza' -> Status, 110 -> Kapacitet, 53 -> Linija (Autobus)

- Г) Потребно је написати упит који за сваки аутобус проналази почетну и последњу станицу линије на којој је ангажован. Приказ резултата треба да буде у формату : RegistracijaVozila, PocetnaStanica, PoslednjaStanica.

π Registracija->RegistracijaVozila,Od->PocetnaStanica, Do->PoslednjaStanica
(Autobus \bowtie Linija=Broj Linija)

Оптималнији упит:

π Registracija->RegistracijaVozila,Od->PocetnaStanica, Do->PoslednjaStanica
(π Registracija, Linija (Autobus) \bowtie Linija=Broj π Broj, Od, Do (Linija))

- д) Потребно је написати скрипту која прави подтабелу VelikiAutobusi који као приказ даје оне аутобусе који имају капацитет преко 100 путника. Искористи подтабелу VelikiAutobusi за приказ само оних линија којима није ангажован ни један велики аутобус. Приказ резултата треба да буде у формату: Broj, Od, Do.

VelikiAutobusi = (σ Kapacitet>100 (Autobus))
 π Broj, Od, Do (Linija) \triangleright ρ Broj \leftarrow Autobus.Linija (VelikiAutobusi)

или

VelikiAutobusi = (σ Kapacitet>100 (Autobus))
 π Broj, Od, Do (Linija) \bowtie (π Broj (Linija) - π Linija (VelikiAutobusi))

- ђ) Аутобуси са регистарским таблицама BG-342-SC и BG-122-XX су се директно сударили са 100% штетом (на срећу без теже повређених). Написати упит (или упите) за брисање ових аутобуса из евиденције (формирати табелу без наведених аутобуса).

σ Registracija!='BG-342-SC' \wedge Registracija!='BG-122-XX' (Autobus)

- е) 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 возњу. Написати упит (или упите) за новонастало стање (формирати табелу).

σ \neg (Status='garaza' \wedge (Linija=53 \vee Linija=84)) (Autobus)
U
 π Registracija, 'voznja'->Status, Kapacitet, Linija (
 σ Status='garaza' \wedge (Linija=53 \vee Linija=84)) (Autobus))

Ако је битан редослед:

π Registracija (Autobus) \bowtie
(
 σ \neg (Status='garaza' \wedge (Linija=53 \vee Linija=84)) (Autobus)
U
 π Registracija, 'voznja'->Status, Kapacitet, Linija (
 σ Status='garaza' \wedge (Linija=53 \vee Linija=84)) (Autobus))
)

ж) Директор предузећа захтева информацију о просечном капацитету својих возила на линији 84. Потребно је написати упит који даје као резултат потребне податке. Приказ резултата треба да буде у формату: Prosek84

γ AVG(Капацитет)->Prosek84 (σ Linija=84 (Autobus))

з) Директор предузећа ипак захтева информацију о просечном капацитету својих возила на свакој линији ојединачно. Потребно је написати упит који даје као резултат потребне податке. Приказ резултата треба да буде у формату: Linija, Prosek

γ Linija; AVG(Капацитет)->Prosek84 (Autobus)

Задатак 2

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	Voditelj	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		
Sif	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) Потребно је написати упит који за све филмове даје скраћен приказ који треба да буде у формату : NazivFilma, Datum, VremePocetka.

π Naziv->NazivFilma, Datum, Od->VremePocetka (σ Tip='Film' (Program))

- б) Потребно је написати упит (или упите) 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: Naziv, Datum, Vreme, Gledanost, Voditelj.

τ Datum DESC, Vreme DESC, Gledanost ASC (
 π Naziv, Datum, Od->Vreme, Gledanost, Ime->Voditelj (Program \bowtie Voditelj=Sif Voditelj)
)

Оптималнији упит:

τ Datum DESC, Vreme DESC, Gledanost ASC (
 π Naziv, Datum, Od->Vreme, Gledanost, Ime->Voditelj (
 π Naziv,Datum,Od,Voditelj, Gledanost (Program) \bowtie Voditelj=Sif π Sif,Ime (Voditelj)
)

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

π Naziv,Datum,Tip (σ Voditelj=NULL (Program))

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

InfoJutro = σ Tip='Info' \wedge Od>=600 \wedge Od<=2000 (Program)
 π Sif, Ime (σ Staz<30 (Voditelj) \bowtie π Voditelj->Sif (InfoJutro))

- д) Одлучено је да се укину квизови који имају мању гледаност од 20. Написати упит (или упите) које реализују брисање описаних емисија (формирати табелу без наведених емисија).

$\sigma \neg(\text{Tip}='Kviz' \wedge \text{Gledanost}<20)$ (Program)

- ђ) Водитељ Мирко Алвировић препушта своју емисију да води његов син Младен Алвировић. Написати упит (или упите) за промену водитеља емисије и при том за гледаност емисије поставити вредност 0.

$X = \text{Program} \bowtie \text{Voditelj} = \text{Sif } \sigma \text{ Ime}='Mirko\ Alvirović'$ (Voditelj)

Program - π Naziv, Datum, Od, Voditelj, Gledanost, Tip (X)

\cup

π Naziv, Datum, Od, 6->Voditelj, 0->Gledanost, Tip (X)

Ако је битан редослед:

$X = \text{Program} \bowtie \text{Voditelj} = \text{Sif } \sigma \text{ Ime}='Mirko\ Alvirović'$ (Voditelj)

π Naziv (Program) \bowtie (

Program - π Naziv, Datum, Od, Voditelj, Gledanost, Tip (X)

\cup

π Naziv, Datum, Od, 6->Voditelj, 0->Gledanost, Tip (X)

)

- е) Директор предузећа захтева информацију о просечној гледаности филмова који су били у јануару у 2018 години. Написати упит који враћа као резултат претходни захтев. Приказ резултата треба да буде у формату: FilmJan2018.

γ AVG(Gledanost)->FilmJan2018 (σ Tip='Film' (Program))

- ж) 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 упит који додаје нову емисију (формирати табелу са наведеном емисијом).

Program

\cup {

Naziv:string, Datum:number, Od:number, Voditelj:number, Gledanost:number, Tip:string

'Oko', NULL, NULL, 6, NULL, 'Info'

}

Задатак 3

1. Дат је део релационе шеме базе података једне социјалне мреже.

User (IDUser, Username, Password)

- IDUser: целобројна величина, јединствено, идентификује корисника
- Username: текстуални тип, јединствено, обавезно поље
- Password: текстуални тип, обавезно поље

Content (IDContent, Time, OwnerID)

- IDContent: целобројна величина, јединствено, идентификује садржај
- Time: целобројна величина, јединствено, обавезно поље
- OwnerID: страни кључ (табела User), обавезно поље

Friend (IDContent, IDUserResponse, Accepted)

- IDContent: страни кључ (табела Content), идентификује пријатељство
- IDUserResponse: страни кључ (табела User), обавезно поље
- Accepted: логички тип, обавезно поље

Status (IDContent, Text)

- IDContent: страни кључ (табела Content), идентификује статус
- Text: текстуални тип, обавезно поље, није празна реч

Like (IDUser, IDContent)

- IDUser: страни кључ (табела User), корисник који је “лајковао”, део идентификације табеле Like
- IDContent: страни кључ (табела Content), садржај који се “лајковао”, део идентификације табеле Like

а) Написати упит или упите за проверу ограничења на нивоу базе, који обезбеђује валидно “лајковање”, а то значи да је могуће “лајковати” само своје и садржаје својих пријатеља. Провера треба да буде у виду исписа невалидних “лајковања”.

$PrijateljjiJednostrano = \pi IDUserResponse, IDOwner (\sigma Accepted=1 (Friend) \bowtie Content)$

$PrijateljjiObostrano = PrijateljjiJednostrano \cup \pi IDOwner, IDUserResponse (PrijateljjiJednostrano)$

$DozvoljenoLajkovati = PrijateljjiObostrano \cup \pi IDUser \rightarrow IDUser1, IDUser \rightarrow IDUser2 (User)$

$\pi IDOwner, IDUser (Content \bowtie Like) - DozvoljenoLajkovati$

- б) Потребно је написати упит (или упите) који проналази корисничка имена корисника са највећим бројем “лајкованих” статуса.

$$\text{ContentBrojLajkova} = \gamma \text{ IDContent; COUNT(*) } \rightarrow \text{BrojLajkova (Like} \bowtie \text{Status)}$$

$$\text{MaxBrojLajkova} = \gamma \text{ MAX(BrojLajkova)} \rightarrow \text{BrojLajkova (ContentBrojLajkova)}$$

$$\pi \text{ Username ((ContentBrojLajkova} \bowtie \text{MaxBrojLajkova)} \bowtie \text{Content} \bowtie \text{IDOwner=IDUser User)}$$

- в) Потребно је написати упит (или упите) који за сваког корисника исписује колико има пријатеља.

$$\text{PrijateljiJednostrano} = \pi \text{ IDUserResponse} \rightarrow \text{IDUser, IDOwner } (\sigma \text{ Accepted}=1 \text{ (Friend)} \bowtie \text{Content})$$

$$\text{PrijateljiObostrano} = \text{PrijateljiJednostrano} \cup \pi \text{ IDOwner, IDUser (PrijateljiJednostrano)}$$

$$\gamma \text{ IDUser; COUNT(*)} \rightarrow \text{BrojPrijatelja (PrijateljiObostrano)}$$

- г) Потребно је написати упит (или упите) који приказује кориснике са бар два пријатеља.

$$\text{PrijateljiJednostrano} = \pi \text{ IDUserResponse} \rightarrow \text{IDUser, IDOwner } (\sigma \text{ Accepted}=1 \text{ (Friend)} \bowtie \text{Content})$$

$$\text{PrijateljiObostrano} = \text{PrijateljiJednostrano} \cup \pi \text{ IDOwner, IDUser (PrijateljiJednostrano)}$$

$$\pi \text{ IDUser } (\sigma \text{ BrojPrijatelja} \geq 2 \text{ (} \gamma \text{ IDUser; COUNT(*)} \rightarrow \text{BrojPrijatelja (PrijateljiObostrano))})$$

Ако је битан редослед:

$$\text{PrijateljiJednostrano} = \pi \text{ IDUserResponse} \rightarrow \text{IDUser, IDOwner } (\sigma \text{ Accepted}=1 \text{ (Friend)} \bowtie \text{Content})$$

$$\text{PrijateljiObostrano} = \text{PrijateljiJednostrano} \cup \pi \text{ IDOwner, IDUser (PrijateljiJednostrano)}$$

$$\pi \text{ IDOwner (PrijateljiObostrano} \bowtie \text{IDOwner=IDOwner1} \wedge \text{IDUser} \neq \text{IDUser1} \\ \pi \text{ IDOwner} \rightarrow \text{IDOwner1, IDUser} \rightarrow \text{IDUser1 (PrijateljiObostrano))}$$

- д) Потребно је написати упит (или упите) који приказује објаве корисника са идентификатором 2 које су лајковали сви његови пријатељи.

$$\text{PrijateljiJednostrano} = \pi \text{ IDUserResponse} \rightarrow \text{IDUser, IDOwner } (\sigma \text{ Accepted}=1 \text{ (Friend)} \bowtie \text{Content})$$

$$\text{PrijateljiObostrano} = \pi \text{ IDUser } (\sigma \text{ IDOwner} = 2 \text{ (PrijateljiJednostrano} \cup \pi \text{ IDOwner, IDUser} \\ \text{(PrijateljiJednostrano))})$$

$$\text{StatusLajkovi} = \pi \text{ IDContent, IDUser (Content} \bowtie \text{Status} \bowtie \text{Like)}$$

$$\text{StatusLajkovi} / \text{PrijateljiObostrano}$$

- ђ) Потребно је написати упит (или упите) који прави подтабелу Wall6 (Type, IDContent, IDOwner, Time, Value, Likes). Овај поглед треба да врати податке у вези активности пријатеља корисника чија је вредност IDUser једнака 6. Активности корисника су: постављање новог статуса и остварено (прихваћено) пријатељство. Wall6 приказује активности опадајуће по времену. За објављен статус и остварено пријатељство вредност поља Type једнак је 's' и 'f', респективно. За објављен статус вредност поља Value једнак је својој вредности поља Text, а за остварено пријатељство Value је једнако IDUserResponse. Поље Likes представља број "лајкова" за дати садржај. Обратити пажњу да је остварено пријатељство двосмерно (уопштено: активност се приказује пријатељима корисника који је послао захтев, као и пријатељима корисника који је прихватио захтев за пријатељство).

Пример:

Type	IDContent	IDOwner	Time	Value	IDFriend	Likes
's'	1	1	201801011301	"ETF"	NULL	4
'f'	3	6	201801020632	NULL	3	3

Prijatelji6=

π IDOwner->IDUser (σ Accepted=1 \wedge IDUserResponse=6 (Friend \bowtie Content))

U

π IDUserResponse (σ Accepted=1 \wedge IDOwner=6 (Friend \bowtie Content))

Prijatelji6Status=

π 'S'->Type, IDContent, IDOwner, Time, Text->Value (Content \bowtie Status \bowtie IDOwner=IDUser Prijatelji6)

\bowtie {

IDFriend:number

NULL

}

Prijatelji6Prijateljstva=

π 'F'->Type, IDContent, IDOwner, Time, Value, IDUserResponse->IDFriend

(Content \bowtie σ Accepted=1 (Friend) \bowtie IDOwner=IDUser \vee IDUserResponse=IDUser Prijatelji6 \bowtie {

Value:string

NULL

})

Prijatelji6Sadrzaj = Prijatelji6Status U Prijatelji6Prijateljstva

SadrzajBrojLajkova = γ IDContent; COUNT(IDUser)->BrLajkova (Prijatelji6Sadrzaj \bowtie Like)

τ Time DESC (Prijatelji6Sadrzaj \bowtie SadrzajBrojLajkova)

- е) Потребно је направити опцију страничења садржаја Wall6 (погледати г). На једној страници могуће је приказати до 2 активности. Потребно је написати упит који приказује садржај треће стране (активности 5-6).

π Time (σ RedniBroj \geq 2 \wedge RedniBroj \leq 3 (

γ Time; COUNT(*) -> RedniBroj (Wall6 \bowtie Time<Time1 π Time->Time1 (Wall6)

)) \bowtie Wall6

ж) Написати скрипту која шаље захтев за пријатељство од стране корисника SundjerBob ка кориснику Lignjoslav, након чега Lignjoslav потврђује пријатељство и на крају SundjerBob “лажује” ово пријатељство. При прављењу новог садржаја, за IDContent узети тренутно максимални увећан за један, док за време узети 20240101.

$$SBIDUser = \pi IDUser (\sigma Username='SundjerBob' (User))$$
$$LigIDUser = \pi IDUser (\sigma Username='Lignjoslav' (User))$$
$$NextIDContent = \pi MaxIDContent+1 \rightarrow IDContent (\gamma MAX(IDContent) \rightarrow MaxIDContent (Content))$$
$$ContentNovi = Content \cup (NextIDContent \times \pi '20240101' \rightarrow Time (User) \times SBIDUser)$$
$$FriendNovi = Friend \cup NextIDContent \times LigIDUser \times \{ \\ \text{Accepted: number} \\ 0 \\ \}$$
$$FriendNovi1 = (FriendNovi \triangleright NextIDContent) \cup \\ \pi IDContent, IDUserResponse, 1 \rightarrow Accepted (\sigma IDContent = NextIDContent (FriendNovi))$$
$$LikeNovi = Like \cup SBIDUser \times NextIDContent$$

За самосталну вежбу

Задатак 4

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	ImelPrezime	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			
BrSobe	Sprat	Kategorija	BrKreveta
1	1	A	5
2	1	A	4
3	1	B	8
4	2	B	8
5	2	C	10

- a) Пацијент са бројем картона 3 је преминуо. Написати упит или упите за брисање пацијента са картоном 3 (формирати табелу без наведеног пацијента).
- б) 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 упит или упите за креирање описане собе (формирати нову табелу са придодатом наведеном собом).
- в) Донатори болнице су одлучили да опреме претходно направљену собу са 4 кревета. Написати упит или упите за промену броја кревета.
- д) Управи болнице је потребна информација о бројевима картона пацијената за које се не зна да ли имају или немају осигурање (NULL вредност). Написати упит или упите за приказ тражених податка.
- ђ) Потребно је написати упит који за сваког пацијената проналази категорију собе у којој се налази. Приказ резултата треба да буде у формату : Karton, ImelPrezime, Kategorija.
- е) Директор болнице захтева информацију о укупном броју кревета у болници. Потребно је написати упит који даје као резултат укупан број кревета у болници. Приказ резултата треба да буде у формату: UkupanBrojKreveta.

- ж) Потребно је написати 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: Број, Sprat, Kategorija, BrKreveta.
- з) Потребно је написати 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, Број
- и) Потребно је написати скрипту која прави подупит **Deca** који као приказ даје пацијенте који су деца (старост мања од 18). Искористи претходно направљен подупит за приказ само оних соба (без понављања) у којима се налази тачно два детета. Приказ резултата треба да буде у формату: Број, Sprat.
- ј) Потребно је написати скрипту која прави подупит **Neosigurani** који као приказ даје пацијенте који су неосигурани. Искористи претходно направљен подупит за приказ само оних соба (без понављања) у којима се налази бар једно неосигурано лице.
- к) Пројектанти базе података желе да промене коришћени модел, па тако желе да трансформишу релациони модел у модел једне табеле. Потребно је написати упит којим се ово остварује.
- л) Потребно је написати упит којим се приказује бројеви спратова на којима постоје празне собе (без пацијента).
- љ) Потребно је написати упит којим се пребројава за сваку собу колико има пацијената. Приказ резултата треба да буде у формату: БројSobe, БројПацијената.

Задатак 5

Посматрани систем је део базе података бродског транспорта. У бази се води евиденција о пловидбама брода. За сваки брод се прати назив и максимални капацитет путника. Релацијом, брод повезује полазну и дестинациону луку (назив). Свака релација има свој трошак превоза, при чему је могуће трошак рачунати на различите начине. Пловидба се организује над одређеној релацији. Пловидба се заказује одређеног датума и зна се када се завршава, прати се тренутни статус, као и колика је цена за купљену карту. Датум се дефинише помоћу године и дана у години (дан може да има вредност: 0-365). Статус пловидбе може да буде: N – пловидба у најави; Z – завршена пловидба; T – у току је пловидба. У систему се памте и путници (име). Путник може да купи карту или карте за једну или више пловидба и тада се памти колико је путник дао новца за купљену карту.

- а) За описани систем креирати модел базе података.
- б) Добијени модел превести у релациону шему.
- в) Потребно је написати упит или упите који као резултат даје имена свих путника који тренутно плове и треба да стигну у луку "Luka Beograd".
- г) Потребно је написати упит или упите који проналази пловидбу или пловидбе где је продата најскупља карта.
- д) Потребно је написати упит или упите који треба да смање цену карте за 10% оних пловидби које су у најави и тренутно је за њих купљено до 10 карата (формирати нову табелу са придодатом наведеном собом).
- ђ) Потребно је написати упит или упите који исписује за сваког путника у коју луку је највише путовао.
- е) Потребно је написати упит или упите који треба да прикажу податке о заради за сваку релацију (луке из које и у коју је брод пловио) и брода који се обраћао на тој релацији. Приказ поред информација о релацији и брода треба да има подате о статусу и вредности релације. Ако је зарада релације негативна, статус је "Губитак", у супротном статус је "Профит". Вредност представља апсолутну вредност зараде релације. Зарада релације се рачуна као сума само оних зарада пловидби које су завршене и код којих је брод попуњен више од 85% свог капацитета. Зарада пловидбе се рачуна као цена * број продатих карата - трошак релације те пловидбе.

Прилог

Задатак 1

group:Zadatak 1

Autobus = {

Registracija:string, Status:string, Kapacitet:number, Linija:number

'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:number, Od:string, Do:string, BrStanica:number

16, 'Pohorska', 'Karaburma', 22

43, 'Kotež', 'Novi Beograd', 17

53, 'Vidikovac', 'Zeleni Venac', 24

83, 'Zemun', 'Crveni Krst', 27

84, 'Zeleni Venac', 'Nova Galenika', 19

}

Задатак 2

group:Zadatak 2

Program = {

Naziv:string, Datum:number, Od:number, Voditelj:number, Gledanost:number, Tip:string

'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 = {

Sif:number, Ime:string, Staz:number

1, 'Žika Nikolić', 34

2, 'Mirko Alvirović', 32

3, 'Dejan Pantelić', 18

4, 'Jovan Memedović', 22

5, 'Jelena Obućina', 10

6, 'Mladen Alvirović', 11

}

Задатак 3

group:Zadatak 3

```
User = {
  IDUser:number, Username:string, Password:string
  1 , 'Bernier95', 'Mg05BHbNQ'
  2 , 'Laurel', 'xcwAScYb'
  3 , 'Katrina', 'nOcdJh2mi'
  4 , 'Maiya_Weber', 'IRiEYcICATX'
  5 , 'Barney', 'LbUAllgcigB'
  6 , 'Damaris', 'YihXjaV'
  7 , 'Jovanny78', '9sh6MOwgjwH6'
  8 , 'Isaac', 'kfALwF7cX3K'
  9 , 'Octavia', 'AXGQfbfdd'
  10 , 'Sidney', '6gBcSk2'
  11 , 'SundjerBob', '7pX43wQ'
  12 , 'Lignjoslav', 'R50ctI3'
}
```

```
Content = {
  IDContent:number, Time:number, IDOwner:number
  1 , 201801011301, 1
  2 , 201801011405, 2
  3 , 201801020632, 6
  4 , 201801021231, 2
  5 , 201801021343, 1
  6 , 201801031111, 3
  7 , 201801031553, 2
  8 , 201801032021, 2
  9 , 201801040234, 3
  10 , 201801040235, 3
  11 , 201801041621, 4
  12 , 201801041842, 4
  13 , 201801042037, 5
  14 , 201801042125, 5
  15 , 201801050700, 7
  16 , 201801050936, 6
  17 , 201801051455, 5
  18 , 201801051734, 6
}
```

```
Status = {
  IDContent:number, Text:string
  1 , 'Uce se baze..'
  2 , 'Mrzim relacionu algebru'
  5 , 'Uzivam u zivotu'
  6 , 'Jedva cekam more.'
  8 , 'Kada ce neki dobar koncert?'
  10 , 'Veceras zurka'
  11 , 'Verujem u vanzemaljce'
  12 , 'Prodajem bubreg'
  15 , 'Igra se CS'
  16 , 'Seta mi se po Kalemegdanu'
  17 , 'Snizeni zamrznuti pilici u lokalnoj prodavnici!'
}
```

```
Friend = {
  IDContent:number, IDUserResponse:number, Accepted:number
  3 , 1 , 1
  4 , 1 , 0
  7 , 5 , 1
  9 , 6 , 0
  13 , 6 , 1
  14 , 3 , 1
  18 , 2 , 1
}
```

```
Like = {
  IDUser:number, IDContent:number
  1 , 1
  1 , 2
  2 , 3
  3 , 1
  4 , 2
  1 , 3
  2 , 4
  6 , 1
  6 , 4
  2 , 1
  5 , 3
  8 , 2
  2 , 5
}
```


Задатак 4

group: Zadatak 4

Pacijent = {

Karton:number, ImelPrezime:string, Osiguranje:number, Starost:number, BrSobe:number

1, 'Pera Perić', 0 , 37 , 1

2, 'Mika Mikić', null , 26 , 1

3, 'Žika Žikić', 1 , 63 , 2

4, 'Miša Mišić', null , 17 , 3

5, 'Sima Simić', 1 , 12 , 4

6, 'Laza Lazić', 0 , 54 , 4

}

Soba = {

BrSobe:number, Sprat:number, Kategorija:string, BrKreveta:number

1, 1, 'A' , 5

2, 1, 'A' , 4

3, 1, 'B' , 8

4, 2, 'B' , 8

5, 2, 'C' , 10

}