



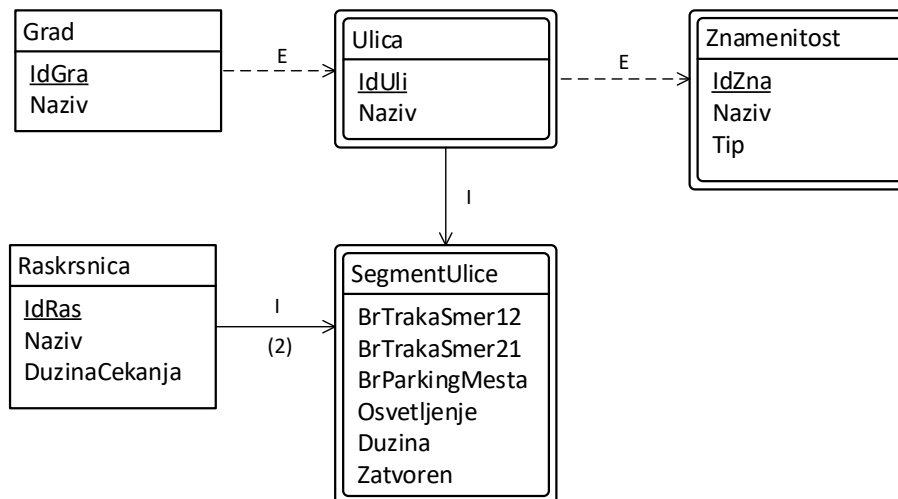
# Базе података 1

- фебруарски испитни рок -

## Група А

Посматра се део базе података саобраћајне инфраструктуре. У овој бази се чувају подаци о градовима, улицама, сегментима улица, раскрсницама и знаменитостима. Свака улица се састоји из барем једног сегмента и почиње и завршава се раскрсницом. На раскрсницама могу да постоје семафори на којима се чека одређено време.

У наставку је дата релациона шема посматраног дела базе податка.



Grad (IdGra, Naziv)

IdGra

- цео број, идентификује град, аутоматско додељивање наредног идентификатора

Naziv

- низ до 50 знакова, обавезно

Ulica (IdUli, Naziv, IdGra)

IdUli

- цео број, идентификује улицу, аутоматско додељивање наредног идентификатора

Naziv

- низ до 50 знакова, обавезно

IdGra

- страни кључ (табела Grad), обавезно

#### Raskrsnica (IdRas, Naziv, DuzinaCekanja)

IdRas	- ceo broj, identifikuje raskrsnicu, automatsko dodeljivanje narednog identifikatora
Naziv	- niz do 50 znakova
DuzinaCekanja	- ceo broj, vrednost veћа od 0

#### Напомена:

Када на раскрсници постоји семафор, у ситуацији да је на семафору упаљено црвено светло чека се DuzinaCekanja времена, у супротном одмах се пролази (нема чекања). У случају да на раскрсници не постоји семафор, DuzinaCekanja има вредност NULL.

Атрибут DuzinaCekanja је целобројног типа и дат је у минутима.

#### SegmentUlice (IdUli, IdRas1, IdRas2, BrTrakaSmer12, BrTrakaSmer21, BrParkingMesta, Osvetljenje, Duzina, Zatvoren)

IdUli	- страни кључ (табела Ulica), обавезно
IdRas1	- страни кључ (табела Raskrsnica), обавезно
IdRas2	- страни кључ (табела Raskrsnica), обавезно
BrTrakaSmer12	- ceo broj, обавезно поље, вредност већа или једнака од 0
BrTrakaSmer21	- ceo broj, обавезно поље, вредност већа или једнака од 0
BrParkingMesta	- ceo broj, вредност већа или једнака од 0
Osvetljenje	- ceo broj, могуће вредности у опсегу од 0 до 1
Duzina	- ceo broj, обавезно поље, вредност већа од 0
Zatvoren	- ceo broj, обавезно поље, могуће вредности у опсегу од 0 до 1

#### Напомене:

Значење атрибута BrTrakaSmer12: број трака у смеру IdRas1→IdRas2.

Значење атрибута BrTrakaSmer21: број трака у смеру IdRas2→IdRas1.

Претпоставити да је вредност поља IdRas1 увек мања од вредности поља IdRas2.

Значење атрибута Osvetljenje: NULL – непознато, 0 –нема осветљења, 1 – има осветљења.

Атрибут Duzina целобројног типа дат је у јединици метар.

Значење атрибута Zatvoren: 0 – отворен, 1 – затворен.

Комбинација вредности у пољима IdUli, IdRas1 и IdRas2 је увек јединствен податак на нивоу табеле SegmentUlice.

#### Znamenitost (IdZna, Naziv, Tip, IdUli)

IdZna	- ceo broj, identifikuje mesto, automatsko dodeljivanje narednog identifikatora
Naziv	- niz do 50 znakova, обавезно, јединствено
Tip	- niz do 30 znakova, обавезно
IdUli	- страни кључ (табела Ulica), обавезно

Задатак 1 [3 поена]

Потребно је направити SQL упит који исписује све улице у Београду које су осветљене у бар неком сегменту. Резултат треба сортирати растуће по називу улице.

Резултат дати у форми: IdUlice, Naziv ulice  
У Sactus-у користити таб: Zadatak 1

---

Задатак 2 [3 поена]

Потребно је направити SQL упит који исписује град у коме се налази најдужа улица. У случају да постоји више таквих улица исте (највеће) дужине, исписати све градове. Резултат треба сортирати по IdGra у опадајућем редоследу.

Резултат дати у форми: IdGra, Naziv  
У Sactus-у користити таб: Zadatak 2

---

Задатак 3 [3 поена]

Потребно је направити SQL упит који исписује улице и број улица са којима се оне укрштају, за све улице које се укрштају са барем три улице . Резултат треба сортирати по IdUli растуће.

Резултат дати у форми: IdUli, Naziv, Broj Ukrstanja  
У Sactus-у користити таб: Zadatak 3

---

Задатак 4 [3 поена]

Потребно је направити SQL упит који исписује све улице које у називу имају од 3 до 5 речи и број знаменитости у њима које су типа *verski objekat*. Сматрати да у називу не постоје спојена два бланко знака. Резултат треба сортирати растуће по IdUli.

Резултат дати у форми: IdUli, Naziv, Broj verskih objekata

У Cactus-у користити таб: Zadatak 4

---

Задатак 5 [4 поена]

Потребно је направити SQL скрипту која ако постоји табела **SegmentUlice** брише табелу **SegmentUlice** из шеме, а затим формира нову табелу **SegmentUlice** која треба да има одговарајућу структуру и ограничења. Није потребно да се реализује ограничење да је вредност поља IdRas1 увек мања од вредности поља IdRas2.

У Cactus-у користити таб Zadatak 5.

---

Задатак 6 [4 поена]

Потребно је направити SQL упит који исписује све градове у којима се сва позоришта у том граду налазе у најкраћој улици у том граду (*pozoriste* је вредност типа знаменитости у табели). Градови који немају позориште не треба да буду исписани. Резултат треба сортирати по IdGra растуће.

Резултат дати у форми: IdGra, Naziv

У Сactus-у користити таб: Zadatak 6

**Није дозвољено коришћење погледа.**

---

---

Задатак 7 [4 поена]

Потребно је направити SQL упит којим се улица са називом *Jove Ilica* затвара због реконструкције, као и сви сегменти улица које се граниче са том улицом.

Након ажурираних података, исписати све податке из табеле *SegmentUlice*.

У Sactus-у користити таб: Zadatak 7

**Није дозвољено коришћење погледа.**

---

Задатак 8 [4 поена]

Потребно је направити SQL упит који исписује све улице у којима се налазе сви типови знаменитости који постоје у том граду у коме постоји барем једна знаменитост. Резултат треба сортирати по IdUli у опадајућем поретку.

Резултат дати у форми: IdUli, Naziv

У Sactus-у користити таб: Zadatak 8

**Није дозвољено коришћење погледа.**

---

Задатак 9 [5 поена]

Потребно је направити SQL скрипту која проналази све улице које се међусобно секу бар два пута. Резултат треба сортирати растуће по IdUli1, а затим растуће по IdUli2. IdUli1 је потребно да има мању вредност од IdUli2.

Резултат дати у форми: IdUli1, Naziv1, IdUli2, Naziv2

У Sactus-у користити таб: Zadatak 9

**Није дозвољено коришћење погледа.**

---

Задатак 10 [5 поена]

Потребно је направити SQL упит који исписује све кружне улице у граду. Улица је кружна ако је њен почетак уједно и њен крај. У случају да не постоји таква улица, резултат треба да садржи један податак у коме пише “*Ne postoji takva ulica*”. Резултат треба сортирати по IdUli опдајуће.

Резултат дати у форми: IdUli

У Sactus-у користити таб: Zadatak 10

**Није дозвољено коришћење погледа.**

---

Задатак 11 [6 поена]

Колегиница Мина Шекуларац жури да стигне да пожели срећу свом друштву пред испит који почиње за 10 минута. Потребно је направити SQL упит који проналази колико најдуже времена може да остане са друштвом док не почне испит. Мина полази са раскрснице под називом *Minnie raskrsnica*, и треба да дође до раскрснице под називом *ETF raskrsnica*. Њена брзина кретања је увек иста и износи . Марфијев закон каже да ће светло на првом семафору на који наиђе бити црвено и да ће бити црвено светло на сваком другом семафору. Потребно је водити рачуна да је кретање могуће само по сегментима улица које су отворене и у коректном су смеру. Обратите пажњу да семафор не мора да се налази на свакој раскрсници, као и на целобројно/реално дељење. У случају да Мина не стиже на време, резултат треба да садржи један податак у коме пише “*Ne stize*”.

На почетној раскрсници (*Minnie raskrsnica*) се не чека семафор, ако га има, док на последњој раскрсници (*ETF raskrsnica*) се чека семафор, ако га има.

Резултат дати у форми: *Vreme do ispita*

У Cactus-у користити таб: *Zadatak 11*

---



Задатак 12 [6 поена]

Потребно је направити SQL упит који исписује 7 најдужих улица у Београду. RB представља редни број улице и потребно је да буде написан речима (prva, druga, treca, cetvrta, peta, sesta, sedma), где је најдужа улица, *druga* следећа по дужини итд. Обратите пажњу да две улице могу да деле исти RB ако су исте дужине. Резултат треба сортирати по RB растуће.

Резултат дати у форми: RB, IdUli, Naziv

У Sactus-у користити таб: Zadatak 12

---