

Univerzitet u Kragujevcu  
Fakultet inženjerskih nauka



## Seminarski rad iz predmeta Baze podataka

**Tema:**

**Razvoj baze podataka informacionog sistema za elektronski školski dnevnik**

Student:  
Nikola Mitrevski 603/2017

Predmetni profesor:  
Milan Erić

Kragujevac 2020.

## Sadržaj:

1	Uvod.....	2
2	ER dijagram .....	3
2.1	Entiteti.....	3
2.2	Veze .....	6
2.3	Kompletan ER dijagram.....	11
3	Logička šema relacione baze podataka i međurelaciona ograničenja.....	12
3.1	Prevođenje tipova entiteta.....	12
3.2	Prevođenje veza .....	12
3.3	Međurelaciona ograničenja .....	15
3.4	Kompletna logička šema .....	16
4	Fizički model relacione baze podataka .....	17
5	Implementacija projektovane baze podataka sa testnim podacima u MySQL.....	18
6	Literatura.....	27

## **1 Uvod**

Cilj projekta elektronskog školskog dnevnika je efikasnije vođenje evidencije o obrazovno-vaspitnom radu učenika.

Efikasnost se ogleda u smanjenju administrativnog rada nastavnika, jer se sve potrebne evidencije nalaze na jednom mestu.

Svakodnevne evidencije koje nastavnik vodi u elektronskom školskom dnevniku su: evidencija o prisustvu učenika, oceni učenika, vladanju učenika, održanim časovima, temama za pismene zadatke, vannastavnim aktivnostima itd.

Zahvaljujući upotrebi elektronskog školskog dnevnika u nastavi, svi rokovi moraju biti ispoštovani jer su kriterijumi unapred isprogramirani, tako da je svako odstupanje od njih odmah vidljivo, te nastavnik tačnije i pouzdanije izvršava svoje radne obaveze.

Zakasnela opravdanja, neobjektivno ocenjivanje, neprisustvovanje nastavi, zadaci i obaveze koje nisu urađene postaju vidljive svim nastavnicima, ali i roditeljima.

Posebne opcije za evidentiranje pohvala, kazni, disciplinskih mera, dostupne svim nastavnicima i roditeljima, takođe doprinose objektivnosti u radu sa učenicima što pruža tačan uvid u to kako se učenik ponaša te se dobija potpunija i objektivnija slika o radu učenika.

Ovakav sistem nesumnjivo pozitivno utiče i na nastavnike i na učenike, usmeravajući ih na tačnost i preciznost u obavljanju školskih obaveza.

Da bi sistem mogao da se koristi bez sumnje su potrebni računari(desktop računari, lap topovi, android uređaji,...) tako da je prethodno opisan sistem zapravo informacioni sistem elektronskog školskog dnevnika.

Svi podaci koje obuhvata informacioni sistem elektronskog školskog dnevnika se nalaze u bazi podataka koja je implementirana i opisana u daljem delu rada.

## 2 ER dijagram

Modelovanje realnog sistema započinje se identifikovanjem entiteta koji učestvuju u sistemu, zatim se odrede veze između postojećih entiteta.

### 2.1 Entiteti

U sistemu su identifikovani sledeći entiteti:

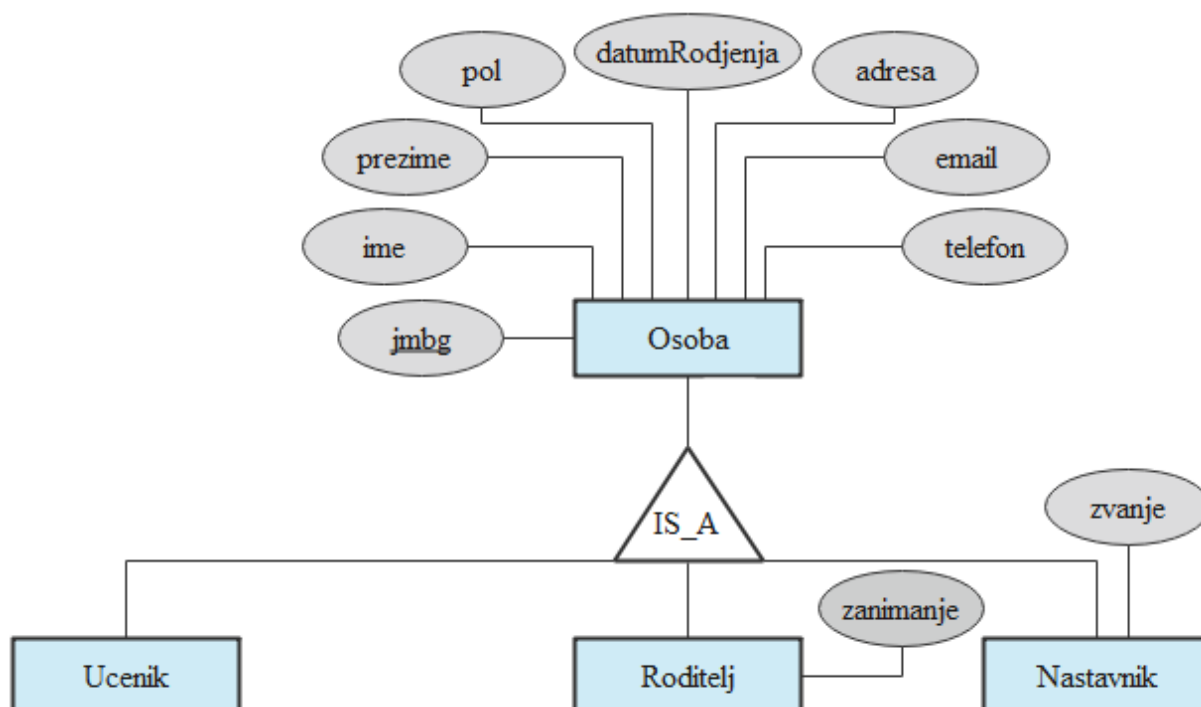
- **Entitet Osoba** i uloga osobe:

Entitet koji se zove “Osoba” predstavlja osobu koja ima svoj lični identifikator jmbg, ime, prezime, pol, datum kada je osoba rođena, adresu osobe i kontakte u vidu emaila i telefona.

Svaka osoba može da ima neku od sledeće tri uloge u informacionom sistemu elektronskog školskog dnevnika:

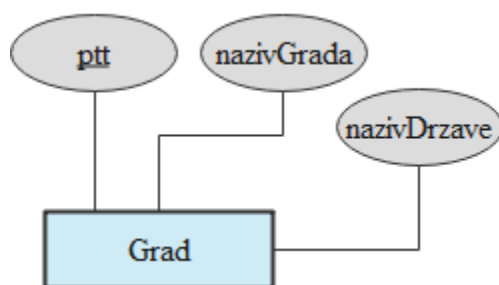
- da je osoba učenik (predstavljena entitetom Ucenik);
- da je osoba roditelj (predstavljena entitetom Roditelj);
- da je osoba nastavnik (predstavljena entitetom Nastavnik).

Što se tiče roditelja on ima svoje zanimanje, a što se tiče nastavnika on ima svoje zvanje. Entiteti kao i njihovi atributi za ovu tačku prikazani su na slici 1.



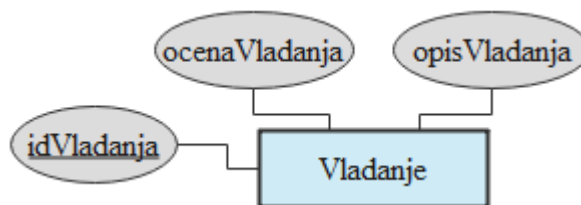
Slika 1 Prikaz nadentiteta Osoba i podentiteta Ucenik, Roditelj i Nastavnik i njihovih atributa

- **Entitet Grad:** Kada je u pitanju entitet “Grad” on predstavlja grad za koji se uvek zna poštanski broj(pty), njegov naziv, naziv drzave u kojoj se taj grad nalazi. Entitet kao i njegovi atributi za ovu tačku prikazani su na slici 2.



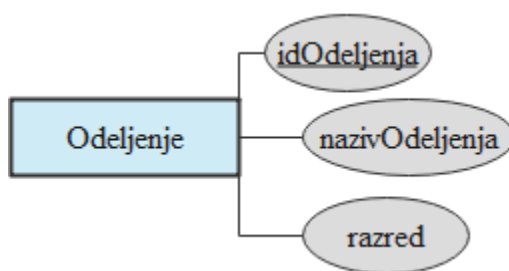
Slika 2 Prikaz entiteta Grad i njegovih atributa

- **Entitet Vladanje:** Kada je u pitanju entitet “Vladanje” on predstavlja vladanje koje je sastavljeno od identifikacionog broja vladanja, ocene iz vladanja kao i opisa vladanja. Entitet kao i njegovi atributi za ovu tačku prikazani su na slici 3.



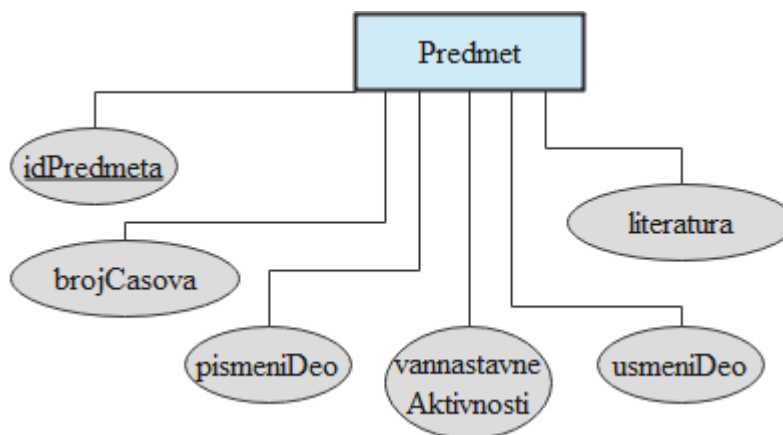
Slika 3 Prikaz entiteta Vladanje i njegovih atributa

- **Entitet Odeljenje:** Kada je u pitanju entitet “Odeljenje” on predstavlja odeljenje gde za svako odeljenje važi da ima svoju identifikacionu oznaku, prema kojoj se zna tačan naziv odeljenja, kao i kom razredu je odeljenje namenjeno. Entitet kao i njegovi atributi za ovu tačku prikazani su na slici 4.



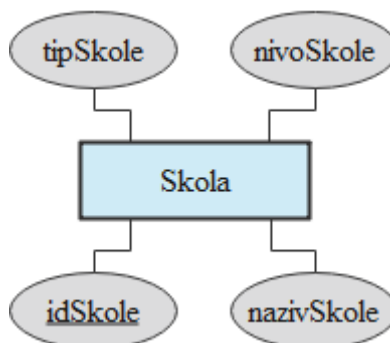
Slika 4 Prikaz entiteta Odeljenje i njegovih atributa

- **Entitet Predmet:** Kada je u pitanju entitet “Predmet” on predstavlja predmet gde svaki predmet ima svoju identifikacionu oznaku na osnovu koje se zna broj predviđenih časova iz tog predmeta, onda način na koji se vrši ocenjivanje iz tog predmeta, kao i literatura koja se koristi iz tog predmeta. Entitet kao i njegovi atributi za ovu tačku prikazani su na slici 5.



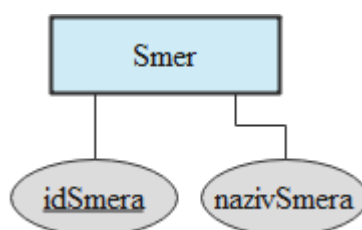
*Slika 5 Prikaz entiteta Predmet i njegovih atributa*

- **Entitet Skola:** Kada je u pitanju entitet “Skola” on predstavlja školu za koju se na osnovu identifikacione oznake zna tačan naziv škole, tip škole(gimnazija, tehnička, ekonomska,...) kao i nivo škole(osnovna ili srednja). Entitet kao i njegovi atributi za ovu tačku prikazani su na slici 6.



*Slika 6 Prikaz entiteta Skola i njegovih atributa*

- **Entitet Smer:** Kada je u pitanju entitet “Smer” on predstavlja smer za koji se na osnovu identifikacione oznake zna tačan naziv tog smer. Entitet kao i njegovi atributi za ovu tačku prikazani su na slici 7.



*Slika 7 Prikaz entiteta Smer i njegovih atributa*

## 2.2 Veze

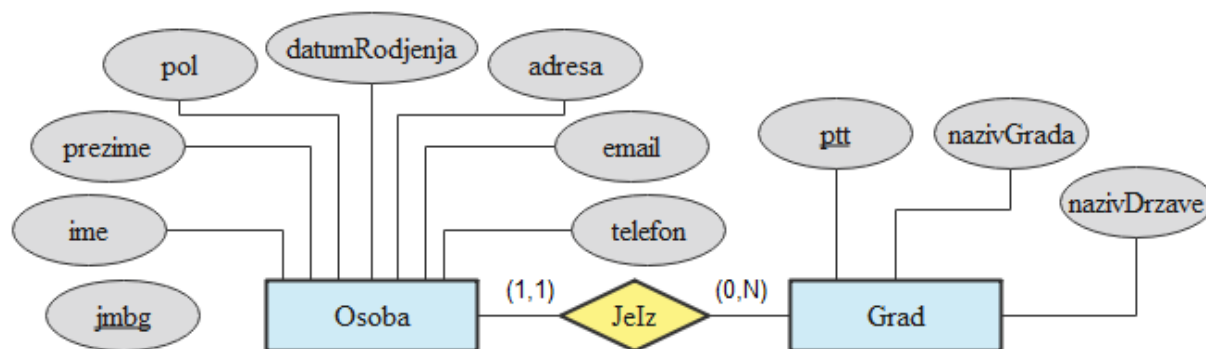
U prethodnom poglavlju je prikazana relacija nasleđivanja između entiteta “Osoba” i njegovih podentiteta “Ucenik”, “Roditelj” i “Nastavnik”.

Priroda relacije nasleđivanja je takva da podentiteti, pored atributa, nasleđuju i sve veze nadentiteta.

Sve veze koje se odnose na entitet “Osoba” takođe se odnose i na entitete “Ucenik”, “Roditelj” i “Nastavnik”.

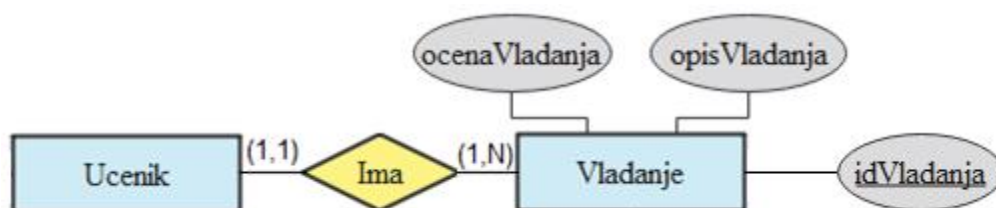
U sistemu su identifikovani sledeće veze:

- **Veza Osoba-Grad:** Svaka osoba pripada tačno jednom gradu. Jedan grad može da ima više osoba, a ne mora ni jednu. Neka je naziv ove veze “JeIz”. Veza je prikazana na slici 8.



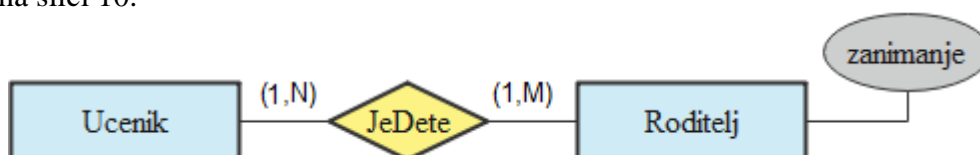
Slika 8 Prikaz veze "JeIz"

- **Veza Ucenik-Vladanje:** Svakom učeniku pripada tačno jedno vladanje. Vladanje je namenjeno barem jednom učeniku. Neka je naziv ove veze “Ima”. Veza je prikazana na slici 9.



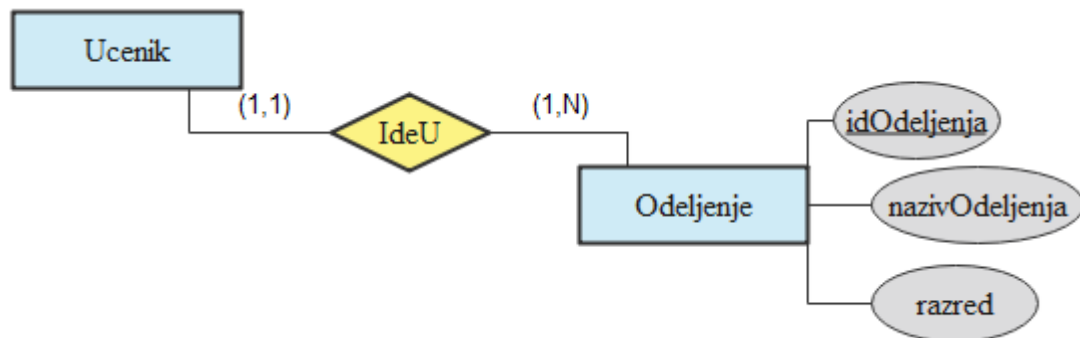
Slika 9 Prikaz veze "Ima"

- **Veza Ucenik-Roditelj:** Jedan učenik mora da ima bar jednog roditelja. Jedan roditelj da bi bio identifikovan kao roditelj u elektronskom školskom dnevniku mora da je roditelj barem jednom učeniku. Neka je naziv ove veze “JeDete”. Veza je prikazana na slici 10.



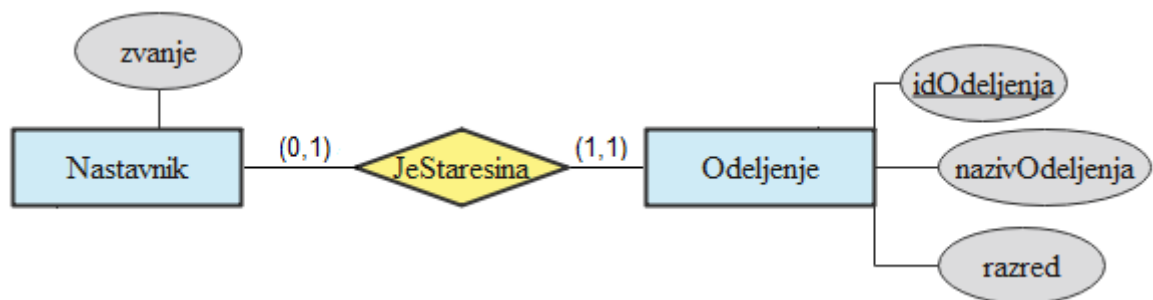
Slika 10 Prikaz veze "JeDete"

- **Veza Ucenik-Odeljenje:** Svaki učenik ide u tačno jedno odeljenje. Jedno odeljenje da bi bilo identifikovano kao odeljenje u elektronskom školskom dnevniku mora da ima barem jednog učenika. Neka je naziv ove veze “IdeU”. Veza je prikazana na slici 11.



Slika 11 Prikaz veze "IdeU"

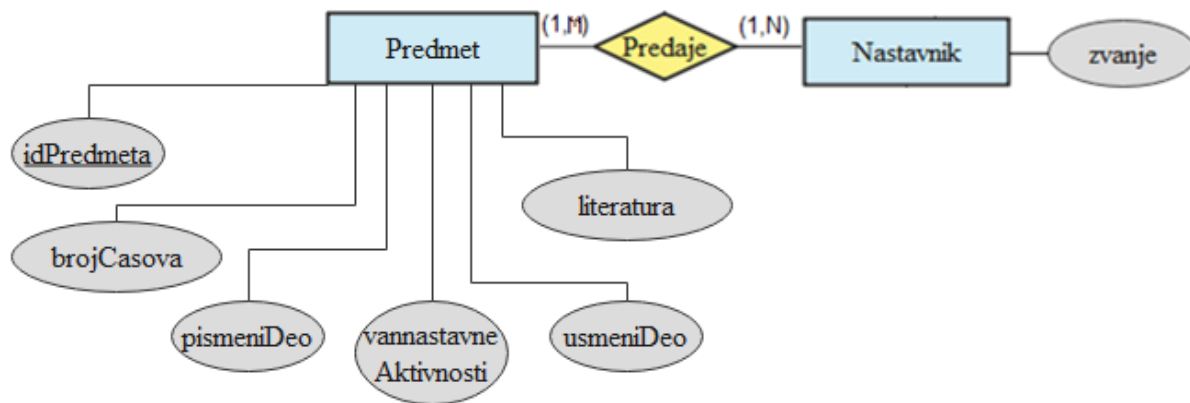
- **Veza Nastavnik-Odeljenje:** Jedan nastavnik može da bude razredni starešina maksimalno jednom odeljenju, a ne mora da bude ni jednom. Svako odeljenje mora da ima tačno jednog razrednog starešinu. Neka je naziv ove veze “JeStaresina”. Veza je prikazana na slici 12.



Slika 12 Prikaz veze "JeStaresina"

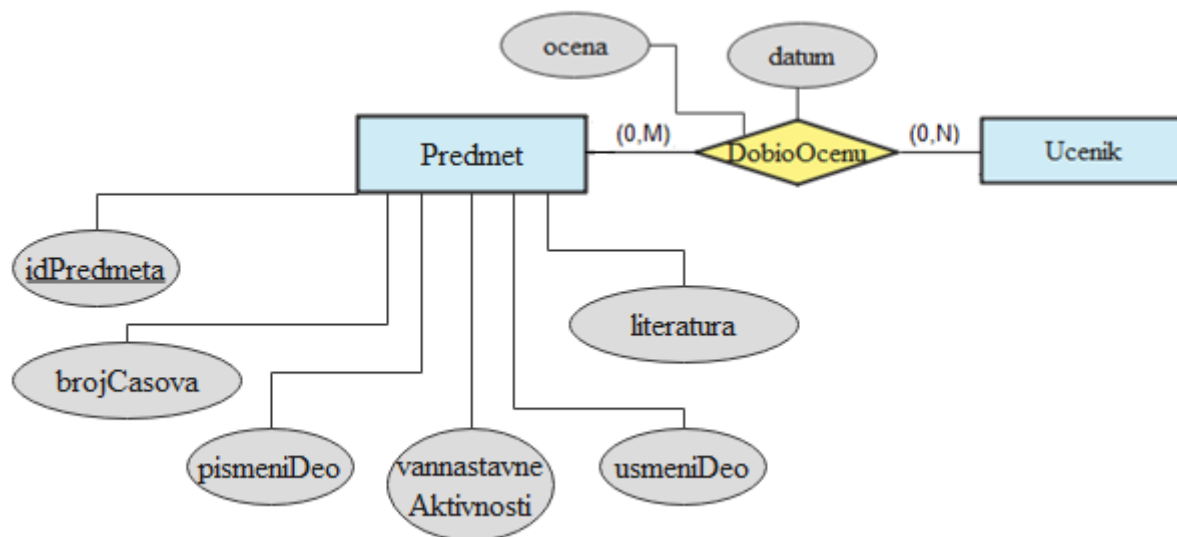


- **Veza Predmet-Nastavnik:** Jedan predmet predaje barem jedan nastavnik. Jedan nastavnik da bi bio identifikovan kao nastavnik u elektronskom školskom dnevniku mora da predaje barem jedan predmet. Neka je naziv ove veze "Predaje". Veza je prikazana na slici 13.



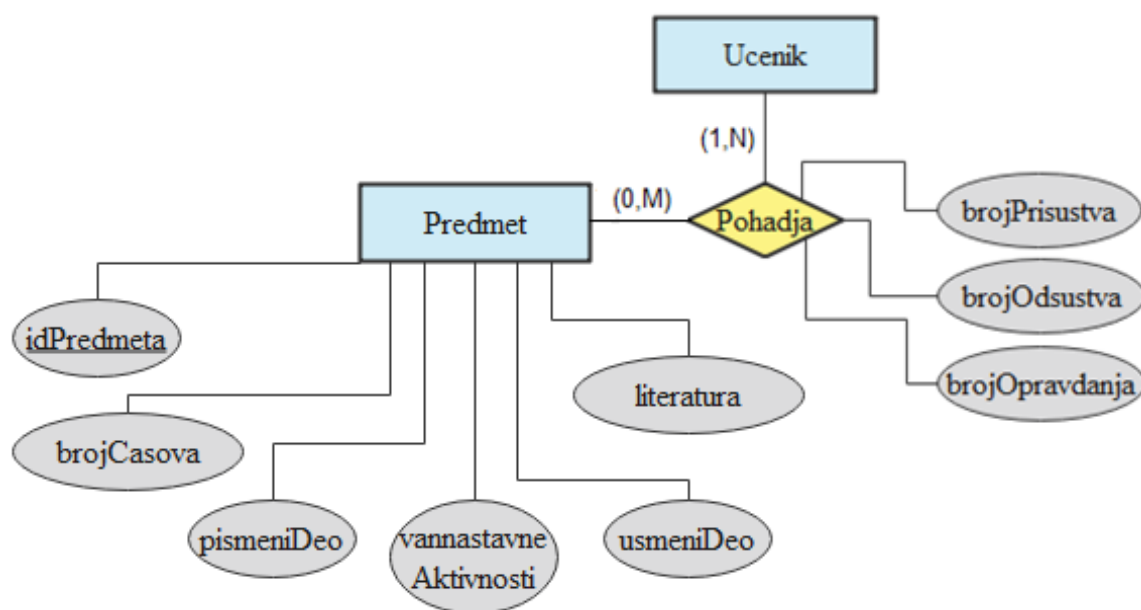
Slika 13 Prikaz veze "Predaje"

- **Veza Predmet-Ucenik:** Iz jednog predmeta postoji mogućnost da je više učenika dobilo ocenu, ali isto tako postoji mogućnost da niko nije dobio ocenu iz tog predmeta. Sa druge strane postoji mogućnost da je jedan učenik ocenjen iz više predmeta, a može da postoji mogućnost da nije niti iz jednog predmeta ocenjen. Neka je naziv ove veze "DobioOcenu". Veza je prikazana na slici 14.



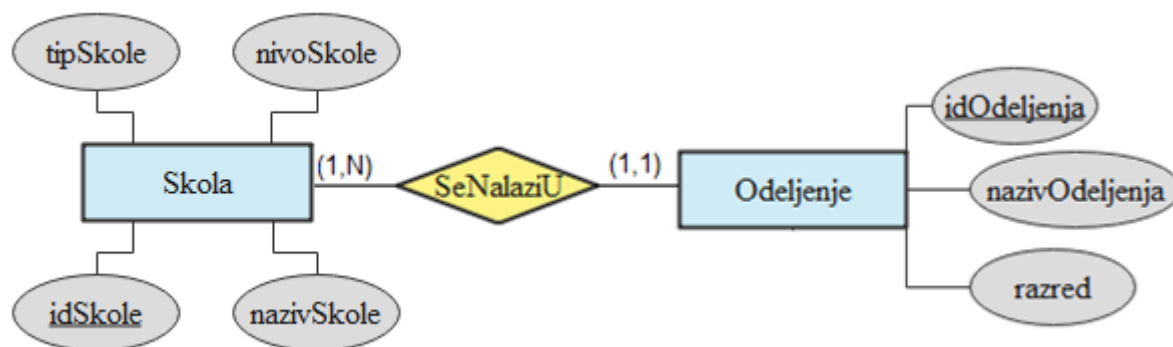
Slika 14 Prikaz veze "DobioOcenu"

- **Veza Predmet-Ucenik II:** Jedan predmet može da pohađa više učenika, a ne mora ni jedan. Jedan učenik ide minimalno na jedan predmet, a može i na više. Neka je naziv ove veze "Pohadja". Veza je prikazana na slici 15.



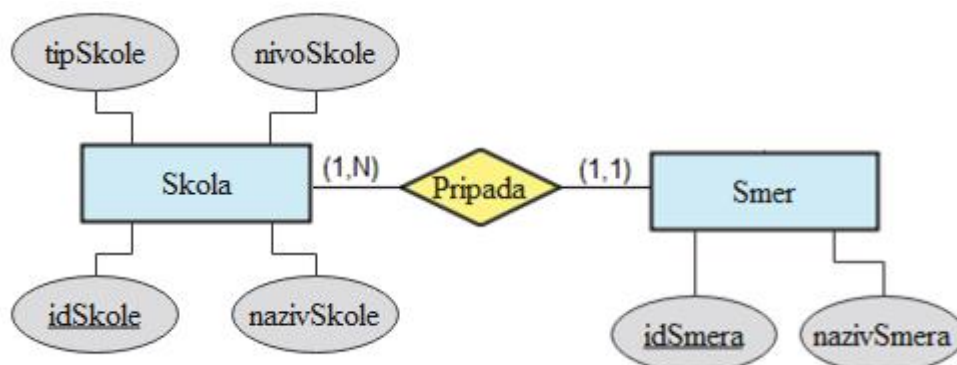
Slika 15 Prikaz veze "Pohadja"

- **Veza Skola-Odeljenje:** Jedna škola da bi se identifikovala u elektronskom školskom dnevniku kao škola mora da ima barem jedno odeljenje. Jedno odeljenje pripada tačno jednoj školi. Neka je naziv ove veze "SeNalaziU". Veza je prikazana na slici 16.



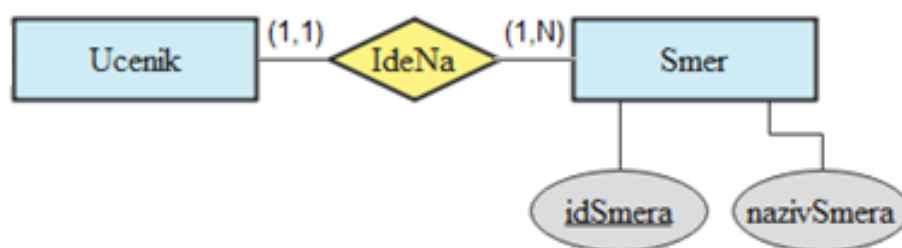
Slika 16 Prikaz veze "SeNalaziU"

- **Veza Skola-Smer:** Jedna škola da bi se identifikovala u elektronskom školskom dnevniku kao škola mora da ima barem jedan smer. Jedan smer pripada tačno jednoj školi. Neka je naziv ove veze “Pripada”. Veza je prikazana na slici 17.



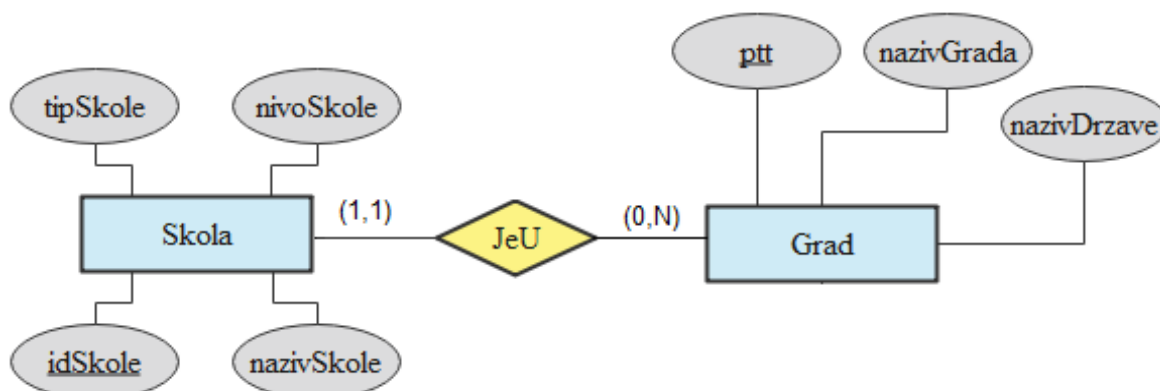
Slika 17 Prikaz veze "Pripada"

- **Veza Ucenik-Smer:** Jedan učenik pripada tačno jednom smeru. Jedan smer da bi se identifikovao u elektronskom školskom dnevniku kao smer mora da ima barem jednog učenika. Neka je naziv ove veze “IdeNa”. Veza je prikazana na slici 18.



Slika 18 Prikaz veze "IdeNa"

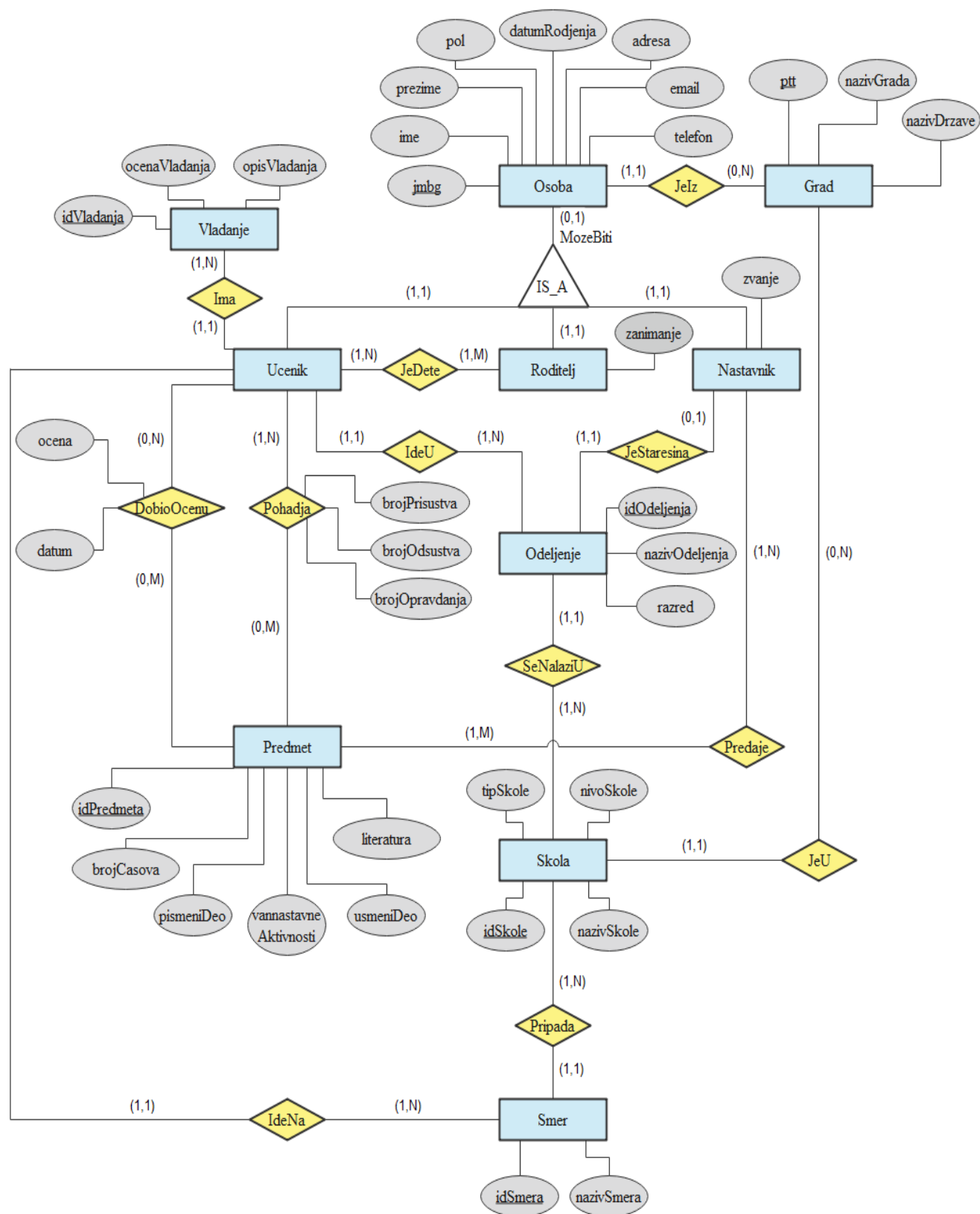
- **Veza Skola-Grad:** Jedna škola se nalazi u tačno jednom gradu. Jedan grad može da ima više škola, a ne mora ni jednu. Neka je naziv ove veze “JeU”. Veza je prikazana na slici 19.



Slika 19 Prikaz veze "JeU"

## 2.3 Kompletan ER dijagram

Na osnovu prethodno identifikovanih entiteta i veza dobija se ER dijagram koji predstavlja kompletnu sliku modela realnog sistema, prikazan je na slici.



Slika 20 Prikaz kompletnog ER dijagrama

### 3 Logička šema relacione baze podataka i međurelaciona ograničenja

Da bi se dobila logička šema relacione baze podataka, moraju svi entiteti i sve veze da se prevedu iz ER dijagrama na njihov ekvivalentni oblik u relacionom dijagramu - *relacije i međurelaciona ograničenja*.

#### 3.1 Prevođenje tipova entiteta

- Važi sledeće pravilo:
  - 1) Prema pravilu, svi tipovi entiteta postaju nezavisne šeme relacija. Ime tipa entiteta postaje ime šeme relacije. Obeležja tipa objekta su obeležja šeme relacije. Za osnovne objekte identifikator entiteta postaje primarni ključ šeme relacije.
- Dobijaju se sledeće relacije:
  - Vladanje(idVladanja, ocenaVladanja, opisVladanja)
  - Osoba(jmbg, ime, prezime, pol, datumRodjenja, adresa, email, telefon)
  - Grad(ptt, nazivGrada, nazivDrzave)
  - Roditelj(zanimanje)
  - Nastavnik(zvanje)
  - Odeljenje(idOdeljenja, nazivOdeljenja, razred)
  - Predmet(idPredmeta, brojCasova, pismeniDeo, vannastavneAktivnosti, usmeniDeo, literatura)
  - Skola(idSkole, nazivSkole, tipSkole, nivoSkole)
  - Smer(idSmera, nazivSmera)

#### 3.2 Prevođenje veza

- U zavisnosti od kardinalnosti, na tipove veza se primenjuju različita pravila za prevođenje.
- **Veza nasleđivanja(IS\_A):**
  - Važe sledeća pravila:
    - 2) Objekat nadtip postaje šema relacije. Ime nadtipa postaje ime šeme relacije. Obeležja nadtipa su obeležja šeme relacije. Identifikator nadtipa postaje ključ šeme relacije.
    - 3) Objekat podtip takođe postaje šema relacije. Ime podtipa postaje ime šeme relacije. Obeležja podtipa su obeležja šeme relacije. Identifikator nadtipa predstavlja ključ šeme relacije.
  - Dobijaju se sledeće relacije:
    - Ucenik(jmbgUcenika)
    - Roditelj(jmbgRoditelja, zanimanje)
    - Nastavnik(jmbgNastavnika, zvanje)

- **Veza “Ima”:**
  - Zbog kardinalnosti (1,1):(1,N) važi sledeće pravilo:
    - 4) Veza ne postaje posebna šeme relacije. Identifikator objekta sa strane za koju je gornja granica kardinaliteta preslikavanja  $GG=N$  postaje obeležje šeme relacije koja odgovara objektu sa strane za koju je  $GG = 1$ .
  - Dobija se sledeća relacija:  
Ucenik(jmbgUcenika, **idVladanja**)
- **Veza “JeIz”:**
  - Zbog kardinalnosti (1,1):(0,N) važi pravilo “4”).
  - Dobija se sledeća relacija:  
Osoba(jmbg, ime, prezime, pol, datumRodjenja, adresa, email, telefon, **ptt**)
- **Veza “JeDete”:**
  - Zbog kardinalnosti (1,M):(0,N) važi sledeće pravilo:
    - 5) Veza postaje posebna šema relacije. Obeležja ove šeme relacije su identifikatori objekata koji su u vezi, a ključ šeme relacije je složeni ključ koji se sastoji od identifikatora objekata koji su u vezi.
  - Dobija se sledeća relacija:  
JeDete(jmbgUcenika, jmbgRoditelja)
- **Veza “IdeU”:**
  - Zbog kardinalnosti (1,1):(1,N) važi pravilo “4”).
  - Dobija se sledeća relacija:  
Ucenik(jmbgUcenika, **idVladanja**, **idOdeljenja**)
- **Veza “JeStaresina”:**
  - Zbog kardinalnosti (1,1):(0,1) važi sledeće pravilo:
    - 6) Vezu i objekte u vezi prevodimo u dve šeme relacije. Za svaki objekat u vezi po jedna šema relacije (prema već definisanom pravilu “1)”), s tim što se identifikator jednog od objekata koji su u vezi ubaci za obeležje druge šeme relacije. Dakle, veza se predstavlja spoljnim ključem.
  - Dobija se sledeća relacija:  
Nastavnik(jmbgNastavnika, zvanje, **idOdeljenja**)

- **Veza “SeNalaziU”:**
  - Zbog kardinalnosti (1,1):(1,N) važi pravilo “4”.
  - Dobija se sledeća relacija:  
Odeljenje(idOdeljenja, nazivOdeljenja, razred, **idSkole**)
- **Veza “Predaje”:**
  - Zbog kardinalnosti (1,1):(1,N) važi pravilo “4”.
  - Dobija se sledeća relacija:  
Predmet(idPredmeta, brojCasova, pismeniDeo, vannastavneAktivnosti, usmeniDeo, literature, **jmbgNastavnika**)
- **Veza “JeU”:**
  - Zbog kardinalnosti (1,1):(0,N) važi pravilo “4”.
  - Dobija se sledeća relacija:  
Skola(idSkole, nazivSkole, tipSkole, nivoSkole, **ptt**)
- **Veza “Pripada”:**
  - Zbog kardinalnosti (1,1):(1,N) važi pravilo “4”.
  - Dobija se sledeća relacija:  
Smer(idSmera, nazivSmera, **idSkole**)
- **Veza “IdeNa”:**
  - Zbog kardinalnosti (1,1):(1,N) važi pravilo “4”.
  - Dobija se sledeća relacija:  
Ucenik(**jmbgUcenika**, **idVladanja**, **idOdeljenja**, **idSmera**)
- **Veza “DobioOcenu”:**
  - Zbog kardinalnosti (0,N):(0,M) važi pravilo “5”.
  - Dobija se sledeća relacija:  
DobioOcenu(**jmbgUcenika**, idPredmeta, ocena, datum)
- **Veza “Pohadja”:**
  - Zbog kardinalnosti (1,N):(0,M) važi pravilo “5”.
  - Dobija se sledeća relacija:  
Pohadja(**jmbgUcenika**, idPredmeta, brojPrisustva, brojOdsustva, brojOpravdanja)

### 3.3 Međurelaciona ograničenja

Međurelaciona ograničenja se definišu za sve nasledene attribute neke relacije.

To uključuje sve strane ključeve svake relacije.

- Ovom prilikom nastaju sledeća međurelaciona ograničenja:

$\text{Osoba}[\text{ptt}] \subseteq \text{Grad}[\text{ptt}]$

$\text{Ucenik}[\text{jmbgUcenika}] \subseteq \text{Osoba}[\text{jmbg}]$

$\text{Ucenik}[\text{idVladanja}] \subseteq \text{Vladanje}[\text{idVladanja}]$

$\text{Ucenik}[\text{idOdeljenja}] \subseteq \text{Odeljenje}[\text{idOdeljenja}]$

$\text{Ucenik}[\text{idSmera}] \subseteq \text{Smer}[\text{idSmera}]$

$\text{Roditelj}[\text{jmbgRoditelja}] \subseteq \text{Osoba}[\text{jmbg}]$

$\text{Nastavnik}[\text{jmbgNastavnika}] \subseteq \text{Osoba}[\text{jmbg}]$

$\text{Nastavnik}[\text{idOdeljenja}] \subseteq \text{Odeljenje}[\text{idOdeljenja}]$

$\text{Odeljenje}[\text{idSkole}] \subseteq \text{Skola}[\text{idSkole}]$

$\text{Predmet}[\text{jmbgNastavnika}] \subseteq \text{Nastavnik}[\text{jmbgNastavnika}]$

$\text{Skola}[\text{ptt}] \subseteq \text{Grad}[\text{ptt}]$

$\text{Smer}[\text{idSkole}] \subseteq \text{Skola}[\text{idSkole}]$

$\text{JeDete}[\text{jmbgUcenika}] \subseteq \text{Ucenik}[\text{jmbgUcenika}]$

$\text{JeDete}[\text{jmbgRoditelja}] \subseteq \text{Roditelj}[\text{jmbgRoditelja}]$

$\text{DobioOcenu}[\text{jmbgUcenika}] \subseteq \text{Ucenik}[\text{jmbgUcenika}]$

$\text{DobioOcenu}[\text{idPredmeta}] \subseteq \text{Predmet}[\text{idPredmeta}]$

$\text{Pohadja}[\text{jmbgUcenika}] \subseteq \text{Ucenik}[\text{jmbgUcenika}]$

$\text{Pohadja}[\text{idPredmeta}] \subseteq \text{Predmet}[\text{idPredmeta}]$



### 3.4 Kompletna logička šema

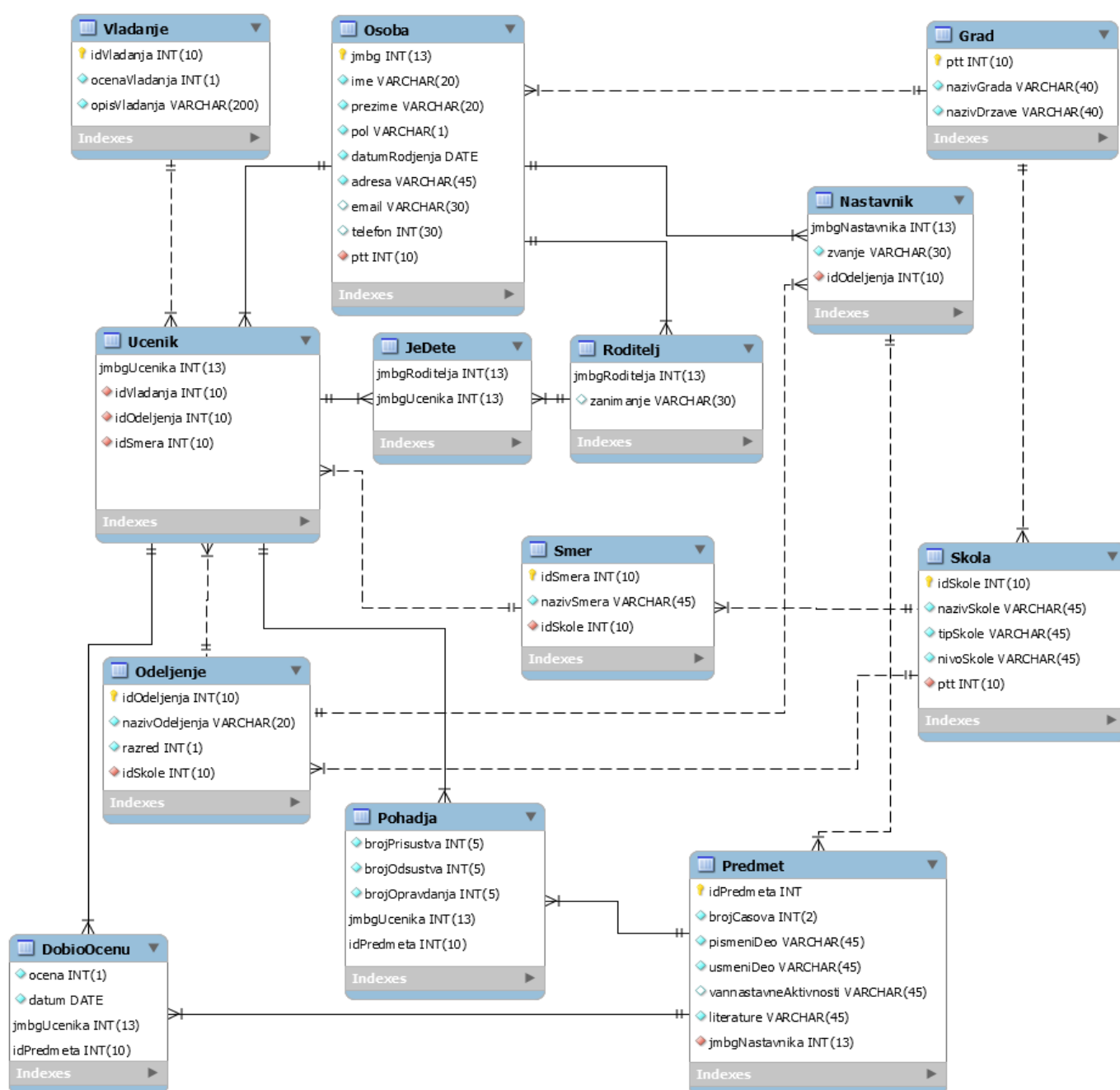
Na osnovu dosadašnjeg izlaganja dolazi se do kompletnog relacionog modela sistema:

$S = \{ \text{Vladanje}(\underline{\text{idVladanja}}, \text{ocenaVladanja}, \text{opisVladanja})$   
 $\text{Grad}(\underline{\text{ptt}}, \text{nazivGrada}, \text{nazivDrzave})$   
 $\text{Osoba}(\underline{\text{jmbg}}, \text{ime}, \text{prezime}, \text{pol}, \text{datumRodjenja}, \text{adresa}, \text{email}, \text{telefon}, \text{ptt})$   
 $\text{Ucenik}(\underline{\text{jmbgUcenika}}, \underline{\text{idVladanja}}, \underline{\text{idOdeljenja}}, \underline{\text{idSmera}})$   
 $\text{Roditelj}(\underline{\text{jmbgRoditelja}}, \text{zanimanje})$   
 $\text{Nastavnik}(\underline{\text{jmbgNastavnika}}, \text{zvanje}, \underline{\text{idOdeljenja}})$   
 $\text{Odeljenje}(\underline{\text{idOdeljenja}}, \text{nazivOdeljenja}, \text{razred}, \underline{\text{idSkole}})$   
 $\text{Predmet}(\underline{\text{idPredmeta}}, \text{brojCasova}, \text{pismeniDeo}, \text{vannastavneAktivnosti}, \text{usmeniDeo},$   
 $\text{literature}, \underline{\text{jmbgNastavnika}})$   
 $\text{Skola}(\underline{\text{idSkole}}, \text{nazivSkole}, \text{tipSkole}, \text{nivoSkole}, \underline{\text{ptt}})$   
 $\text{Smer}(\underline{\text{idSmera}}, \text{nazivSmera}, \underline{\text{idSkole}})$   
 $\text{JeDete}(\underline{\text{jmbgUcenika}}, \underline{\text{jmbgRoditelja}})$   
 $\text{DobioOcenu}(\underline{\text{jmbgUcenika}}, \underline{\text{idPredmeta}}, \text{ocena}, \text{datum})$   
 $\text{Pohadja}(\underline{\text{jmbgUcenika}}, \underline{\text{idPredmeta}}, \text{brojPrisustva}, \text{brojOdsustva}, \text{brojOpravdanja}) \}$

$I = \{ \text{Osoba}[\text{ptt}] \subseteq \text{Grad}[\text{ptt}]$   
 $\text{Ucenik}[\text{jmbgUcenika}] \subseteq \text{Osoba}[\text{jmbg}]$   
 $\text{Ucenik}[\text{idVladanja}] \subseteq \text{Vladanje}[\text{idVladanja}]$   
 $\text{Ucenik}[\text{idOdeljenja}] \subseteq \text{Odeljenje}[\text{idOdeljenja}]$   
 $\text{Ucenik}[\text{idSmera}] \subseteq \text{Smer}[\text{idSmera}]$   
 $\text{Roditelj}[\text{jmbgRoditelja}] \subseteq \text{Osoba}[\text{jmbg}]$   
 $\text{Nastavnik}[\text{jmbgNastavnika}] \subseteq \text{Osoba}[\text{jmbg}]$   
 $\text{Nastavnik}[\text{idOdeljenja}] \subseteq \text{Odeljenje}[\text{idOdeljenja}]$   
 $\text{Odeljenje}[\text{idSkole}] \subseteq \text{Skola}[\text{idSkole}]$   
 $\text{Predmet}[\text{jmbgNastavnika}] \subseteq \text{Nastavnik}[\text{jmbgNastavnika}]$   
 $\text{Skola}[\text{ptt}] \subseteq \text{Grad}[\text{ptt}]$   
 $\text{Smer}[\text{idSkole}] \subseteq \text{Skola}[\text{idSkole}]$   
 $\text{JeDete}[\text{jmbgUcenika}] \subseteq \text{Ucenik}[\text{jmbgUcenika}]$   
 $\text{JeDete}[\text{jmbgRoditelja}] \subseteq \text{Roditelj}[\text{jmbgRoditelja}]$   
 $\text{DobioOcenu}[\text{jmbgUcenika}] \subseteq \text{Ucenik}[\text{jmbgUcenika}]$   
 $\text{DobioOcenu}[\text{idPredmeta}] \subseteq \text{Predmet}[\text{idPredmeta}]$   
 $\text{Pohadja}[\text{jmbgUcenika}] \subseteq \text{Ucenik}[\text{jmbgUcenika}]$   
 $\text{Pohadja}[\text{idPredmeta}] \subseteq \text{Predmet}[\text{idPredmeta}] \}$

## 4 Fizički model relacione baze podataka

Fizički model relacione baze podataka je najniži model i on opisuje konkretnu implementaciju baze podataka. Prelaz iz logičkog u fizičko projektovanje podrazumeva promenu u fokusu i veštinama koje su zahtevane, tj. prave se izmene koje olakšavaju održavanje podataka i povećavaju efikasnost operacija sistema za upravljanje bazama podataka. Fizičko projektovanje baze podataka se često naziva i „modeliranje podataka“. Uobičajno je da se na fizičkom nivou govori o tabelama i kolonama, a ne o relacijama i atributima. Kompletan fizički model može se videti na slici 21.



Slika 21 Prikaz kompletanog fizičkog modela

## 5 Implementacija projektovane baze podataka sa testnim podacima u MySQL

U delu ispod je dat kompletan generisan SQL kod sa testnim podacima, koji implementira razmatrani realni sistem.

```
-- MySQL Script generated by MySQL Workbench
-- Wed Jun 24 20:42:26 2020
-- Model: New Model   Version: 1.0
-- MySQL Workbench Forward Engineering
```

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN
_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUB
STITUTION';
```

```
-- -----
-- Schema mydb
-- -----
```

```
DROP SCHEMA IF EXISTS `mydb` ;
```

```
-- -----
-- Schema mydb
-- -----
```

```
CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 ;
USE `mydb` ;
```

```
-- -----
-- Table `mydb`.`Vladanje`
-- -----
```

```
DROP TABLE IF EXISTS `mydb`.`Vladanje` ;
CREATE TABLE IF NOT EXISTS `mydb`.`Vladanje` (
  `idVladanja` INT(10) NOT NULL,
  `ocenaVladanja` INT(1) NOT NULL DEFAULT 5,
  `opisVladanja` VARCHAR(200) NOT NULL,
  PRIMARY KEY (`idVladanja`),
  UNIQUE INDEX `idVladanja_UNIQUE` (`idVladanja` ASC))
ENGINE = InnoDB;
```

```
-- Table `mydb`.`Grad`
```

```
DROP TABLE IF EXISTS `mydb`.`Grad` ;
CREATE TABLE IF NOT EXISTS `mydb`.`Grad` (
  `ptt` INT(10) NOT NULL,
  `nazivGrada` VARCHAR(40) NOT NULL,
  `nazivDrzave` VARCHAR(40) NOT NULL,
  PRIMARY KEY (`ptt`),
  UNIQUE INDEX `ptt_UNIQUE` (`ptt` ASC))
ENGINE = InnoDB;
```

```
-- Table `mydb`.`Osoba`
```

```
DROP TABLE IF EXISTS `mydb`.`Osoba` ;
CREATE TABLE IF NOT EXISTS `mydb`.`Osoba` (
  `jmbg` INT(13) NOT NULL,
  `ime` VARCHAR(20) NOT NULL,
  `prezime` VARCHAR(20) NOT NULL,
  `pol` VARCHAR(1) NOT NULL,
  `datumRodjenja` DATE NOT NULL,
  `adresa` VARCHAR(45) NOT NULL,
  `email` VARCHAR(30) NULL,
  `telefon` INT(30) NULL,
  `ptt` INT(10) NOT NULL,
  PRIMARY KEY (`jmbg`),
  UNIQUE INDEX `email_UNIQUE` (`email` ASC),
  UNIQUE INDEX `telefon_UNIQUE` (`telefon` ASC),
  INDEX `ptt_idx` (`ptt` ASC),
  UNIQUE INDEX `jmbg_UNIQUE` (`jmbg` ASC),
  CONSTRAINT `ptt`
  FOREIGN KEY (`ptt`)
  REFERENCES `mydb`.`Grad` (`ptt`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

-----  
-- Table `mydb`.`Roditelj`  
-----

```
DROP TABLE IF EXISTS `mydb`.`Roditelj` ;
CREATE TABLE IF NOT EXISTS `mydb`.`Roditelj` (
  `jmbgRoditelja` INT(13) NOT NULL,
  `zanimanje` VARCHAR(30) NULL,
  PRIMARY KEY (`jmbgRoditelja`),
  UNIQUE INDEX `jmbgRoditelja_UNIQUE` (`jmbgRoditelja` ASC),
  CONSTRAINT `jmbgRoditelja`
  FOREIGN KEY (`jmbgRoditelja`)
  REFERENCES `mydb`.`Osoba` (`jmbg`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

-----  
-- Table `mydb`.`Skola`  
-----

```
DROP TABLE IF EXISTS `mydb`.`Skola` ;
CREATE TABLE IF NOT EXISTS `mydb`.`Skola` (
  `idSkole` INT(10) NOT NULL,
  `nazivSkole` VARCHAR(45) NOT NULL,
  `tipSkole` VARCHAR(45) NOT NULL,
  `nivoSkole` VARCHAR(45) NOT NULL,
  `ptt` INT(10) NOT NULL,
  PRIMARY KEY (`idSkole`),
  INDEX `ptt_idx` (`ptt` ASC),
  UNIQUE INDEX `ptt_UNIQUE` (`ptt` ASC),
  UNIQUE INDEX `idSkole_UNIQUE` (`idSkole` ASC),
  CONSTRAINT `ptt2`
  FOREIGN KEY (`ptt`)
  REFERENCES `mydb`.`Grad` (`ptt`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-- Table `mydb`.`Odeljenje`
```

```
-----  
DROP TABLE IF EXISTS `mydb`.`Odeljenje` ;  
CREATE TABLE IF NOT EXISTS `mydb`.`Odeljenje` (  
  `idOdeljenja` INT(10) NOT NULL,  
  `nazivOdeljenja` VARCHAR(20) NOT NULL,  
  `razred` INT(1) NOT NULL,  
  `idSkole` INT(10) NOT NULL,  
  PRIMARY KEY (`idOdeljenja`),  
  INDEX `idSkole_idx` (`idSkole` ASC),  
  UNIQUE INDEX `idOdeljenja_UNIQUE` (`idOdeljenja` ASC),  
  UNIQUE INDEX `idSkole_UNIQUE` (`idSkole` ASC),  
  CONSTRAINT `idSkole`  
  FOREIGN KEY (`idSkole`)  
  REFERENCES `mydb`.`Skola` (`idSkole`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-- Table `mydb`.`Nastavnik`
```

```
-----  
DROP TABLE IF EXISTS `mydb`.`Nastavnik` ;  
CREATE TABLE IF NOT EXISTS `mydb`.`Nastavnik` (  
  `jmbgNastavnika` INT(13) NOT NULL,  
  `zvanje` VARCHAR(30) NOT NULL,  
  `idOdeljenja` INT(10) NOT NULL,  
  PRIMARY KEY (`jmbgNastavnika`),  
  INDEX `idOdeljenja_idx` (`idOdeljenja` ASC),  
  UNIQUE INDEX `jmbgNastavnika_UNIQUE` (`jmbgNastavnika` ASC),  
  CONSTRAINT `jmbgRoditelja2`  
  FOREIGN KEY (`jmbgNastavnika`)  
  REFERENCES `mydb`.`Osoba` (`jmbg`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION,  
  CONSTRAINT `idOdeljenja`  
  FOREIGN KEY (`idOdeljenja`)  
  REFERENCES `mydb`.`Odeljenje` (`idOdeljenja`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-- Table `mydb`.`Predmet`
```

```
-----  
DROP TABLE IF EXISTS `mydb`.`Predmet` ;  
CREATE TABLE IF NOT EXISTS `mydb`.`Predmet` (  
  `idPredmeta` INT NOT NULL,  
  `brojCasova` INT(2) NOT NULL,  
  `pismeniDeo` VARCHAR(45) NOT NULL,  
  `usmeniDeo` VARCHAR(45) NOT NULL,  
  `vannastavneAktivnosti` VARCHAR(45) NULL,  
  `literature` VARCHAR(45) NOT NULL,  
  `jmbgNastavnika` INT(13) NOT NULL,  
  PRIMARY KEY (`idPredmeta`),  
  INDEX `jmbgNastavnika_idx` (`jmbgNastavnika` ASC),  
  UNIQUE INDEX `idPredmeta_UNIQUE` (`idPredmeta` ASC),  
  UNIQUE INDEX `jmbgNastavnika_UNIQUE` (`jmbgNastavnika` ASC),  
  CONSTRAINT `jmbgNastavnika`  
  FOREIGN KEY (`jmbgNastavnika`)  
  REFERENCES `mydb`.`Nastavnik` (`jmbgNastavnika`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-- Table `mydb`.`Smer`
```

```
-----  
DROP TABLE IF EXISTS `mydb`.`Smer` ;  
CREATE TABLE IF NOT EXISTS `mydb`.`Smer` (  
  `idSmera` INT(10) NOT NULL,  
  `nazivSmera` VARCHAR(45) NOT NULL,  
  `idSkole` INT(10) NOT NULL,  
  PRIMARY KEY (`idSmera`),  
  INDEX `idSkole_idx` (`idSkole` ASC),  
  UNIQUE INDEX `idSmera_UNIQUE` (`idSmera` ASC),  
  UNIQUE INDEX `idSkole_UNIQUE` (`idSkole` ASC),  
  CONSTRAINT `idSkole2`  
  FOREIGN KEY (`idSkole`)  
  REFERENCES `mydb`.`Skola` (`idSkole`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-- Table `mydb`.`Ucenik`
```

```
-----  
DROP TABLE IF EXISTS `mydb`.`Ucenik` ;  
CREATE TABLE IF NOT EXISTS `mydb`.`Ucenik` (  
  `jmbgUcenika` INT(13) NOT NULL,  
  `idVladanja` INT(10) NOT NULL,  
  `idOdeljenja` INT(10) NOT NULL,  
  `idSmera` INT(10) NOT NULL,  
  PRIMARY KEY (`jmbgUcenika`),  
  INDEX `idVladanja_idx` (`idVladanja` ASC),  
  INDEX `idOdeljenja_idx` (`idOdeljenja` ASC),  
  INDEX `idSmera_idx` (`idSmera` ASC),  
  UNIQUE INDEX `jmbgUcenika_UNIQUE` (`jmbgUcenika` ASC),  
  CONSTRAINT `jmbgUcenika`  
  FOREIGN KEY (`jmbgUcenika`)  
  REFERENCES `mydb`.`Osoba` (`jmbg`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION,  
  CONSTRAINT `idVladanja`  
  FOREIGN KEY (`idVladanja`)  
  REFERENCES `mydb`.`Vladanje` (`idVladanja`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION,  
  CONSTRAINT `idOdeljenja2`  
  FOREIGN KEY (`idOdeljenja`)  
  REFERENCES `mydb`.`Odeljenje` (`idOdeljenja`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION,  
  CONSTRAINT `idSmera`  
  FOREIGN KEY (`idSmera`)  
  REFERENCES `mydb`.`Smer` (`idSmera`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```



-----  
-- Table `mydb`.`JeDete`  
-----

```
DROP TABLE IF EXISTS `mydb`.`JeDete` ;
CREATE TABLE IF NOT EXISTS `mydb`.`JeDete` (
  `jmbgRoditelja` INT(13) NOT NULL,
  `jmbgUcenika` INT(13) NOT NULL,
  PRIMARY KEY (`jmbgRoditelja`, `jmbgUcenika`),
  INDEX `jmbgUcenika_idx` (`jmbgUcenika` ASC),
  UNIQUE INDEX `jmbgRoditelja_UNIQUE` (`jmbgRoditelja` ASC),
  UNIQUE INDEX `jmbgUcenika_UNIQUE` (`jmbgUcenika` ASC))
ENGINE = InnoDB;
```

-----  
-- Table `mydb`.`DobioOcenu`  
-----

```
DROP TABLE IF EXISTS `mydb`.`DobioOcenu` ;
CREATE TABLE IF NOT EXISTS `mydb`.`DobioOcenu` (
  `ocena` INT(1) NOT NULL,
  `datum` DATE NOT NULL,
  `jmbgUcenika` INT(13) NOT NULL,
  `idPredmeta` INT(10) NOT NULL,
  INDEX `jmbgUcenika_idx` (`jmbgUcenika` ASC),
  INDEX `idPredmeta_idx` (`idPredmeta` ASC),
  PRIMARY KEY (`idPredmeta`, `jmbgUcenika`),
  UNIQUE INDEX `ocena_UNIQUE` (`ocena` ASC),
  UNIQUE INDEX `idPredmeta_UNIQUE` (`idPredmeta` ASC),
  UNIQUE INDEX `jmbgUcenika_UNIQUE` (`jmbgUcenika` ASC))
ENGINE = InnoDB;
```

-----  
-- Table `mydb`.`Pohadja`  
-----

```
DROP TABLE IF EXISTS `mydb`.`Pohadja` ;
CREATE TABLE IF NOT EXISTS `mydb`.`Pohadja` (
  `brojPrisustva` INT(5) NOT NULL,
  `brojOdsustva` INT(5) NOT NULL,
  `brojOpravdanja` INT(5) NOT NULL,
  `jmbgUcenika` INT(13) NOT NULL,
  `idPredmeta` INT(10) NOT NULL,
  PRIMARY KEY (`idPredmeta`, `jmbgUcenika`),
  INDEX `jmbgUcenika_idx` (`jmbgUcenika` ASC),
  UNIQUE INDEX `idPredmeta_UNIQUE` (`idPredmeta` ASC),
  UNIQUE INDEX `jmbgUcenika_UNIQUE` (`jmbgUcenika` ASC))ENGINE = InnoDB;
```

```
SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

-----
-- Adding data to tables
-----

INSERT INTO `mydb`.`Vladanje` VALUES (223, 5, "Ucenik nije kaznjivan.");

INSERT INTO `mydb`.`Grad` VALUES (11320, "Velika Plana", "Republika Srbija");

INSERT INTO `mydb`.`Osoba` VALUES (123456, "Petar", "Petrovic", "m", "1995-06-16", "...", "petar123@gmail.com", 0652366854, 11320);
INSERT INTO `mydb`.`Osoba` VALUES (123455, "Marko", "Nedeljkovic", "m", "1975-01-11", "...", "nedeljkovic@gmail.com", 0645588741, 11320);
INSERT INTO `mydb`.`Osoba` VALUES (123433, "Mila", "Pavlovic", "z", "1985-06-18", "...", "milapavlovic@gmail.com", 0626698778, 11320);

INSERT INTO `mydb`.`Roditelj` VALUES (123456, "Ekonomista");

INSERT INTO `mydb`.`Skola` VALUES (999, "Nikola Tesla", "Tehnicka skola", "srednji nivo", 11320);

INSERT INTO `mydb`.`Odeljenje` VALUES (111, "IV-3", 4, 999);

INSERT INTO `mydb`.`Nastavnik` VALUES (123455, "Diplomirani inzenjer", 111);

INSERT INTO `mydb`.`Predmet` VALUES (2889, 60, "...", "...", "...", "...", 123455);

INSERT INTO `mydb`.`Smer` VALUES (554668, "Elektrotehnicar racunara", 999);

INSERT INTO `mydb`.`Ucenik` VALUES (123456, 223, 111, 554668);

INSERT INTO `mydb`.`JeDete` VALUES (123433, 123456);

INSERT INTO `mydb`.`DobioOcenu` VALUES (5, "2020-06-24", 123456, 2889);

INSERT INTO `mydb`.`Pohadja` VALUES (30, 15, 10, 123456, 2889);
```

-----  
-- SQL queries  
-----

/\*Prikaz liste svih nastavnika sa predmetima koje oni drže\*/

```
SELECT osoba.jmbg, osoba.ime, osoba.prezime, predmet.idPredmeta
FROM osoba INNER JOIN nastavnik ON osoba.jmbg = nastavnik.jmbgNastavnika
INNER JOIN predmet ON predmet.jmbgNastavnika = nastavnik.jmbgNastavnika;
```

/\*Prikaz liste učenika i njihovih roditelja\*/

```
SELECT R.J1 AS jmbgRoditelja, R.I1 AS imeRoditelja, R.P1 AS prezimeRoditelja,
U.J2 AS jmbgUcenika, U.I2 AS imeUcenika, U.P2 AS prezimeUcenika
FROM jedete V
INNER JOIN (SELECT osoba.jmbg AS J1, osoba.ime AS I1, osoba.prezime AS P1
FROM osoba INNER JOIN roditelj ON osoba.jmbg = roditelj.jmbgRoditelja) AS R
ON V.jmbgRoditelja = R.J1
INNER JOIN (SELECT osoba.jmbg AS J2, osoba.ime AS I2, osoba.prezime AS P2
FROM osoba INNER JOIN ucenik ON osoba.jmbg = ucenik.jmbgUcenika) AS U
ON V.jmbgUcenika = U.J2;
```

/\*Prikaz liste škola sa njihovim smerovima\*/

```
SELECT skola.idSkole, skola.nazivSkole, skola.tipSkole, skola.nivoSkole,
smer.idSmera, smer.nazivSmera
FROM skola INNER JOIN smer ON skola.idSkole = smer.idSkole;
```

/\*Prikaz liste jmbg brojeva učenika i id brojeve predmeta na koje idu, kao i jmbg brojeve nastavnika koje predaju te predmete\*/

```
SELECT ucenik.jmbgUcenika, predmet.idPredmeta, nastavnik.jmbgNastavnika
FROM ucenik INNER JOIN pohadja ON ucenik.jmbgUcenika = pohadja.jmbgUcenika
INNER JOIN predmet ON pohadja.idPredmeta = predmet.idPredmeta
INNER JOIN nastavnik ON predmet.jmbgNastavnika = nastavnik.jmbgNastavnika;
```

/\*Prikaz liste učenika i njihovih ocena iz nekog predmeta\*/

```
SELECT osoba.jmbg AS jmbgUcenika, osoba.ime AS imeUcenika, osoba.prezime AS
prezimeUcenika, dobioocenu.idPredmeta, dobioocenu.ocena
FROM osoba INNER JOIN ucenik ON osoba.jmbg = ucenik.jmbgUcenika
INNER JOIN dobioocenu ON ucenik.jmbgUcenika = dobioocenu.jmbgUcenika;
```

/\*Prikaz liste svih škola iz grada Velika Plana\*/

```
SELECT skola.idSkole, skola.nazivSkole, skola.tipSkole, skola.nivoSkole,
grd.nazivGrada FROM skola INNER JOIN
(SELECT ptt, nazivGrada FROM grad WHERE nazivGrada = "Velika Plana") AS grd
ON skola.ptt = grd.ptt;
```

## **6 Literatura**

Branislav Lazarević(2003): Baze podataka, Fakultet organizacionih nauka, Beograd