## Univerzitet u Kragujevcu

## Fakultet inženjerskih nauka



# Baze podataka

### Projektni zadatak:

#### Razvoj baza podataka informacionog sistema za hotelski sistem

#### upravljanja

### Student: Profesor:

#### Đorđe Ilić 648/2017 Milan Erić

Sadržaj

[1.Uvod 3](#_Toc44174122)

[1.1 Poslovne i informacione potrebe realnog sistema 3](#_Toc44174123)

[2. ER Dijagram 4](#_Toc44174124)

[2.1 Entiteti 4](#_Toc44174125)

[2.2 Veze 10](#_Toc44174126)

[2.3 Kompletan ER dijagram 16](#_Toc44174127)

[3. Logička šema relacione baze podataka 17](#_Toc44174128)

[3.1 Entiteti 17](#_Toc44174129)

[3.2 Identifikacione veze slabih entiteta 17](#_Toc44174130)

[3.3 Veze 18](#_Toc44174131)

[3.4 Međurelaciona ograničenja 18](#_Toc44174132)

[3.5 Kompletna logička šema 19](#_Toc44174133)

[4.Fizička šema relacione baze podataka 21](#_Toc44174134)

[5.Implementacija projektovane baze podataka u MySQL 22](#_Toc44174135)

[6. Literatura 36](#_Toc44174136)

## 1.Uvod

Glavni cilj ovog projekta je kreiranje sistema za uravljanje bazama podataka hotela.U nastavku će biti objasnjeno i izvrseno modelovanje baze podataka koja je pogodna za skladištenje informacija koje se koriste u hotelskom sistemu upravljanja.Pre svega će biti prikazan i objasnjen ER dijagram sistema,relaciona(logička)šema i na kraju fizička šema i njen SQL koji je implementira.

### Poslovne i informacione potrebe realnog sistema

Hotel je ugostiteljski objekat za smeštaj,otvorenog tipa,namenjen različitim kategorijama korisnika,koji posluje po komercionalnim principima,u tehničko-tehnološkom i ogranizacionom smislu koji obezbeđuje pružanje usluga smeštaja,hrane i dodatnih usluga.Hotel može imati više lanaca koji mogu imati više hotela.Zato nam treba organizovan sistem upravljanja,koji može lako da upravlja nad svim operacijama i podacima hotelskih lanaca i hotela respektivno. Mi ćemo upravljati sledećim oblastima hotelskog sistema za upravljanje bazama podataka:Lanci hotela i njihovi detalji,hoteli u svakom lancu i njihovi detalji i ostale informacije poput soba i njihov opis i popusti(itd),informacije o zaposlenima i odeljenjima u kojima rade,informacije o gostima,upravljanje rezervacijama i ostalim uslugama koje gosti koriste.Sve navedene informacije će biti smeštene u bazi podataka.

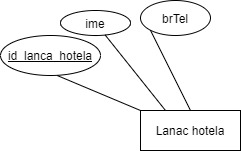
## 2. ER Dijagram

Modelovanje realnog sistema se započinje identifikovanjem entiteta,a zatim se određuju veze između entiteta.Osnovni elementi ER modela su entiteti,veze i atributi.Veza je odnos između dva ili više entiteta,entitet je događaj ili objekat o kome je potrebno čuvati podatke,a atribut je osobina entiteta ili veze.

### 2.1 Entiteti

* **Lanac hotela**

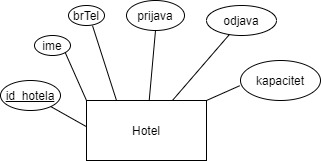
Entitet *Lanac hotela* sadrzi informacije vezane za lance hotela.Identifikator ovog entiteta je *id\_lanca\_hotela,*pored identifikatora sadrži i atribute:*ime i brTel*.Lanac hotela poseduje niz hotela koji ga sačinjavaju i predstavlja jednog od nosećih entiteta ovog modela.



Slika br.1

* **Hotel**

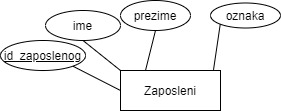
Entitet *Hotel* sadrži informacije vezane za hotel.Njegovi atributi su *id\_hotela,ime,brTel,prijava,odjava,kapacitet i broj\_zvezdica*.Identifikator ovog entiteta je atribut *id\_hotela* .



Slika br.2

* **Zaposleni**

Entitet *Zaposleni* sadrži informacije za zaposlenog koji radi u hotelu.Njegovi atributi su:*id\_zaposlenog,ime i prezime.*Identifikator ovog entiteta je atribut *id\_zaposlenog*.



Slika br.3

* **Odeljenje**

Entitet *Odeljenje* sadrži informacije za deo hotela koji obavlja određene fukcije.Njegovi atributi su *id\_odeljenja,ime,opis.*Identifikator ovog entiteta je atribut *id\_odeljenja,*ostala dva atributa predstavljaju neke osnovne informacije o ovom entitetu.

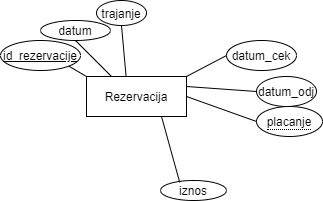
****

Slika br.4

* **Rezervacija**

Entitet *Rezervacija* sadrži informacije vezane za rezervacije koje su prijavljene od strane gostiju u hotelu.Atributi ovog entiteta su *id\_rezervacije,datum,trajanje,datum\_cek,iznos,*

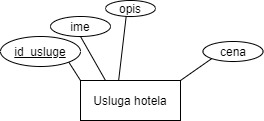
*datum\_odj i placanje.*Identifikator ovog entiteta je atribut *id\_rezervacije.*



Slika br.5

* **Usluga hotela**

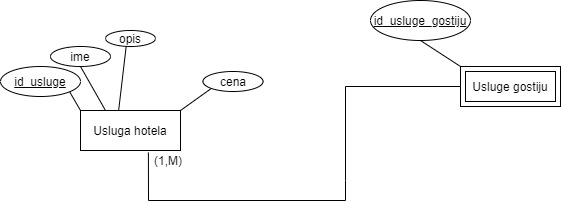
Entitet *Usluga hotela* sadrži informacije vezane za usluge koje hotel pruža gostima.Atributi entiteta su:*id\_usluge,ime,opis i cena.*Identifikator entiteta je atribut *id\_usluge.*



Slika br.6

* **Usluga gostiju**

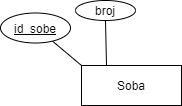
Entitet *Usluga gostiju* sadrži informacije vezane za usluge hotela koje koriste gosti i ovaj entitet je slab entiet jer zavisi od entiteta *Usluga hotela*.Ovaj entitet ima atribut *id\_usluge\_gostiju.*Identifikatori ovog entitea su atributi *id\_usluge\_gostiju* i identifikator *id\_usluge* entiteta *Usluga hotela.*Uslugu hotela može da koristi jedan ili više gostiju.



Slika br.7

* **Soba**

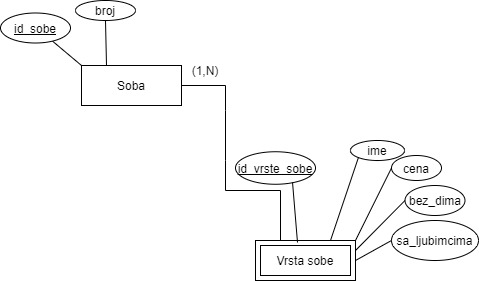
Entitet *Soba* sarži informacije vezane za sobe hotela.Atributi ovog entiteta su:*id\_sobe i broj.*Identifikator ovog entiteta je atribut *id\_sobe.*



Slika br.8

* **Vrsta sobe**

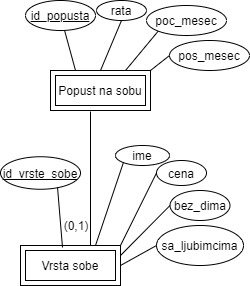
Entitet *Vrsta sobe* sadrži informacije o vrsti sobe.Atributi entiteta su:*id\_vrste\_sobe,ime,cena,bez\_dima i sa\_ljubimcima.*Ovaj entitet je slab entitet jer zavisi od entiteta *Soba.*Identifikatori ovog entiteta su atributi *id\_vrste\_sobe i id\_sobe* koji je atribut entiteta *Soba.*Sobe hotela imaju najjmanje jednu vrstu soba,a obično imaju i više.



Slika br.9

* **Popust na sobu**

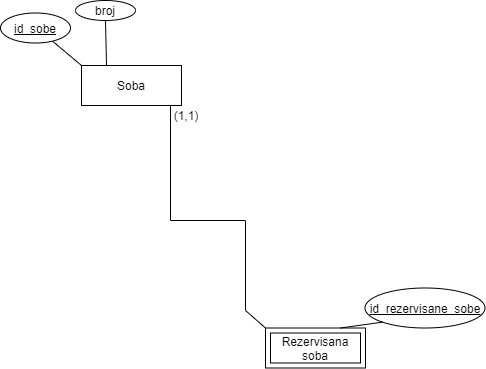
Entitet *Popust na sobu* sadrži informacije o popustima koje su ostvarene na datu sobu.Atributi ovog entiteta su:*id\_popusta,rata,poc\_mesec,pos\_mesec.*Ovaj entitet je slab entitet jer zavisi od entiteta *Vrsta sobe*.Identifikatori ovog entiteta su atributi:*id\_popusta, id\_vrste\_sobe i id\_sobe* koji su atribut entiteta *Soba i*  *Vrsta sobe.*Za jednu vrstu sobe može biti samo jedan popust,a ne mora ni jedan.



Slika br.10

* **Rezervisana soba**

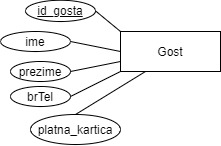
Entitet *Rezervisana soba* sadrži informacije o sobama koje su rezervisane.Atribut ovog entiteta je id\_rezervisane\_sobe.Ovaj entitet je slab entitet jer zavisi od entiteta *Soba.*Identifikatori ovog entiteta su atributi *id\_rezervisane\_sobe* i *id\_sobe* koji je atribut entiteta *Soba.*Za jednu sobu važi samo jedna rezervacija.



Slika br.11

* **Gost**

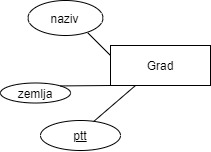
Entitet *Gost* sadrži informacije vezano za gosta hotela.Njegovi atributi su:*id\_gosta,ime,prezime,brTel,platna\_kartica.*Identifikator ovog entiteta je atribut *id\_gosta.*

**

Slika br.12

* **Grad**

Entitet *Grad* sadrži informacije o gradovima hotela,zaposlenih i gostiju.Njegovi atributi su:*ptt,naziv,zemlja.*Identifikator ovog entiteta je atribut *ptt.*

**

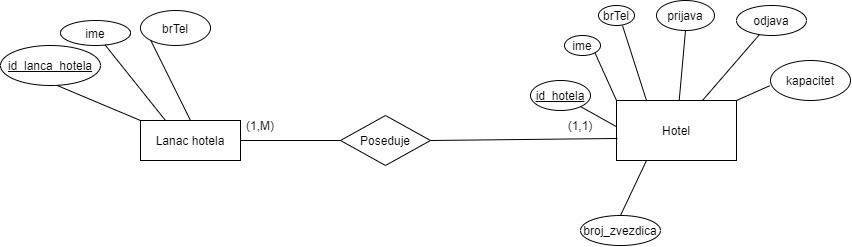
Slika br.13

### 2.2 Veze

U nastavku definisaćemo sve veze koje povezuju date entitete u modelu sistema:

* **Veza Lanac hotela-Hotel (*Poseduje)***

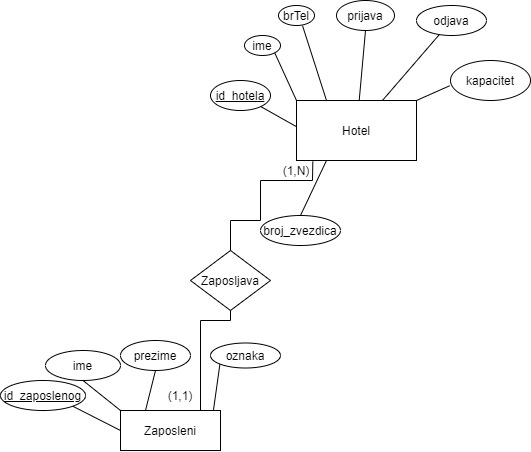
Ova veza povezuje entitet *Lanac hotela* sa entitetom *Hotel.*Lanac hotela poseduje najjmanje jedan hotel,ali obično ih ima više(1,M),dok hotel može pripadati isključivo jednom lancu hotela(1,1).



Slika br.14

* **Veza Hotel-Zaposleni (*Zapošljava)***

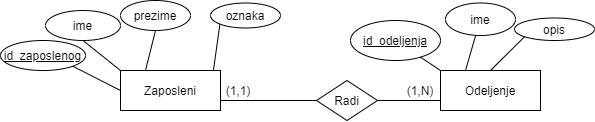
Ova veza povezuje entitet *Hotel* sa entitetom *Zaposleni.*Hotel zapošljava jednog ili više radnika(1,N),dok radnik(zaposleni) može biti zaposlen samo u jednom hotelu,ne može raditi na dva mesta.(1,1).



Slika br.15

* **Veza Zaposleni-Odeljenje *(Radi)***

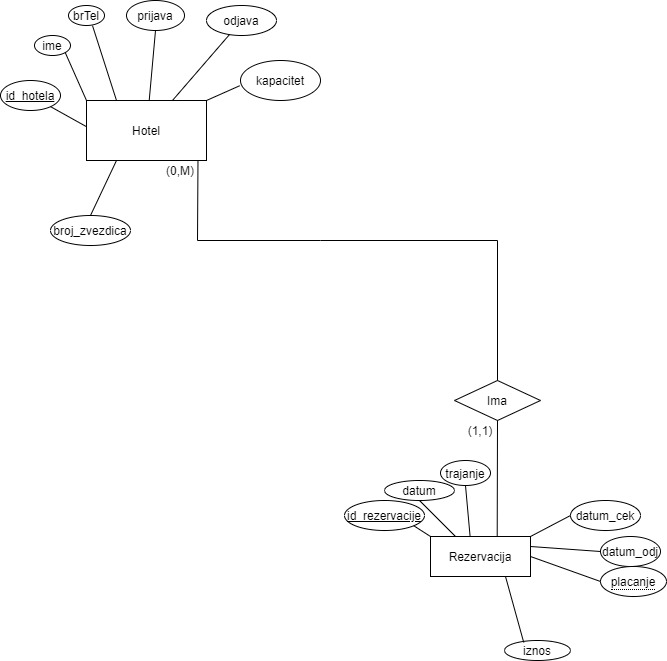
Ova veza povezuje entitete *Zaposleni i Odeljenje.*Isključivo jedan zaposleni može raditi u jednom odeljenju(1:1),dok u odeljenju može raditi najjmanje jedan zaposleni,a može radititi više od jenog(1,N).



Slika br.16

* **Veza Hotel-Rezervacija *(Ima)***

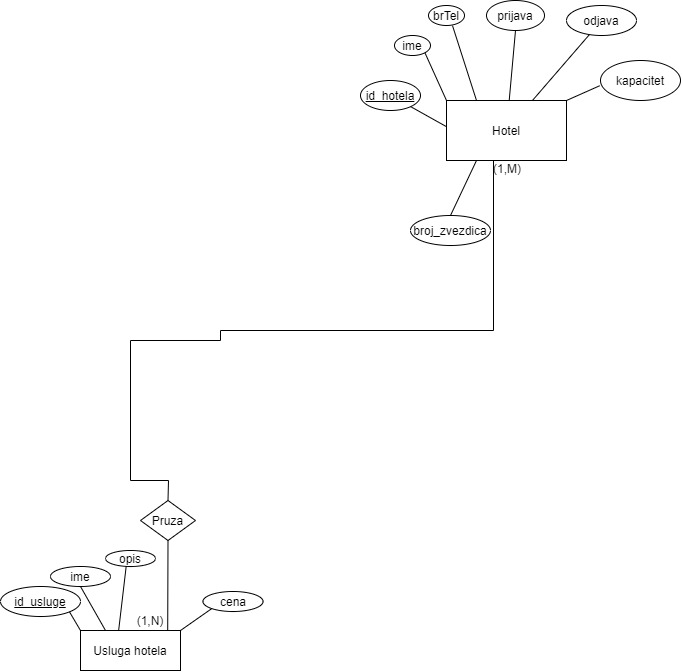
Ova veza povezuje entitete *Hotel i Rezervacija.*Hotel može imati više rezervacija,a ne mora da ima ni jednu(0,M),dok rezervacija može pripadati samo jednom hotelu(1,1).



Slika br.17

* **Veza Hotel-Usluga hotela *(Pruza)***

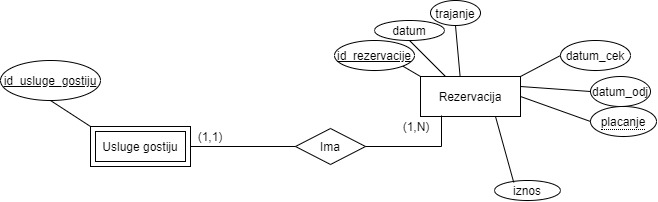
Veza *Pruza* povezuje entitete *Hotel* i *Usluga hotela*.Hotel može da pruži najjmanje jednu uslugu hotela,a obično pruža više(1,M),dok usluga hotela može biti pružena u jednom ili više hotela(1,N).



Slika br.18

* **Veza Usluge gostiju-Rezervacija *(Ima)***

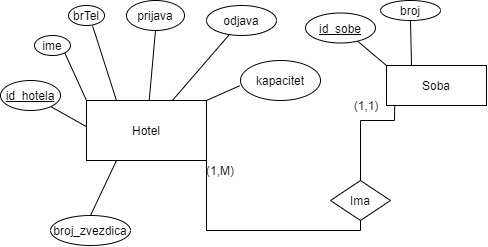
Ova veza povezuje entitete *Usluge gostiju i Rezervacija.*Usluga gostiju može isključivo imati jednu rezervaciju(ne može se rezervisati dva puta ista usluga)(1:1),dok rezervacije imaju najjmanje jednu uslugu hotela koju koriste gosti,a mogu imati više(1,N).



Slika br.19

* **Veza Hotel-Soba *(Ima)***

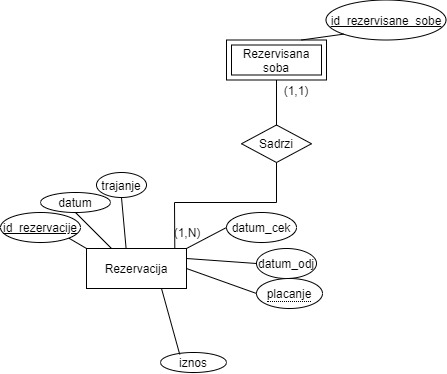
Ova veza povezuje entitete *Hotel i Soba.*Hotel ima najjmanje jednu sobu,a može imati i više soba(1,M),dok soba pripada isključivo samo jedndom hotelu(ne može se jedna soba nalaziti u dva ili više hotela,1:1).



Slika br.20

* **Veza Rezervisana soba-Rezervacija *(Sadrži)***

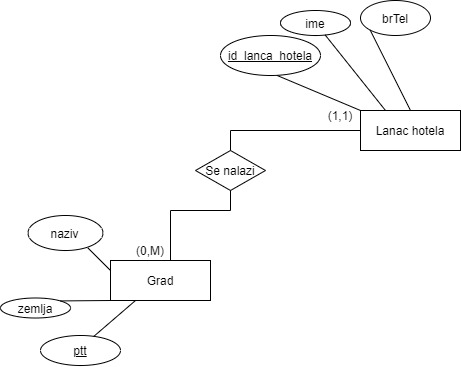
Veza *Sadrzi* povezuje entitete *Rezervisana soba* i *Rezervacija*.Svaka rezervisana soba isključivo sadrži samo jednu rezervaciju(soba je rezervisana pod jednom rezervacijom,a ne na više,1:1),dok rezervacija sadrži najjmanje jednu,a može imati više soba(pod jednom rezervacijom se može rezervisati više soba,1:N).



Slika br.21

* **Veza Lanac Hotela-Grad *(Se nalazi)***

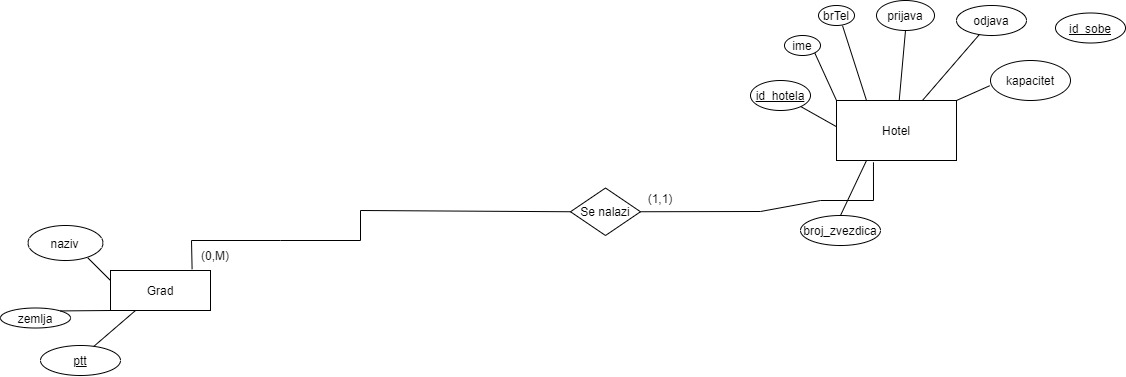
Veza *Se nalazi* povezuje entitete *Lanac hotela i Grad.*Svaki lanac hotela se može nalaziti samo u jednom gradu(1,1),dok se u gradu može nalaziti više lanaca hotela,a ne mora ni jedan(0,M).



Slika br.22

* **Veza Hotel-Grad *(Se nalazi)***

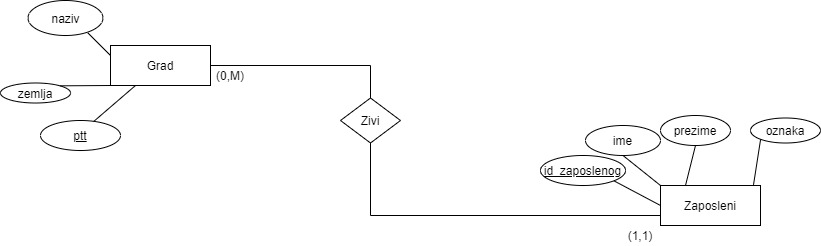
Veza *Se nalazi* povezuje entitete *Hotel i Grad.*Svaki hotel se može nalaziti samo u jednom gradu(1,1),dok se u gradu može nalaziti više hotela,a ne mora ni jedan(0,M).



Slika br.23

* **Veza Zaposleni-Grad *(Zivi)***

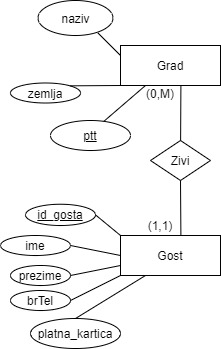
Ova veza povezuje entitete *Zaposleni i Grad.*Svaki zaposleni živi isključivo u jednom gradu(1:1),dok u gradu može da živi više zaposlenih,a ne mora ni jedan.



Slika br.24

* **Veza Gost-Grad *(Zivi)***

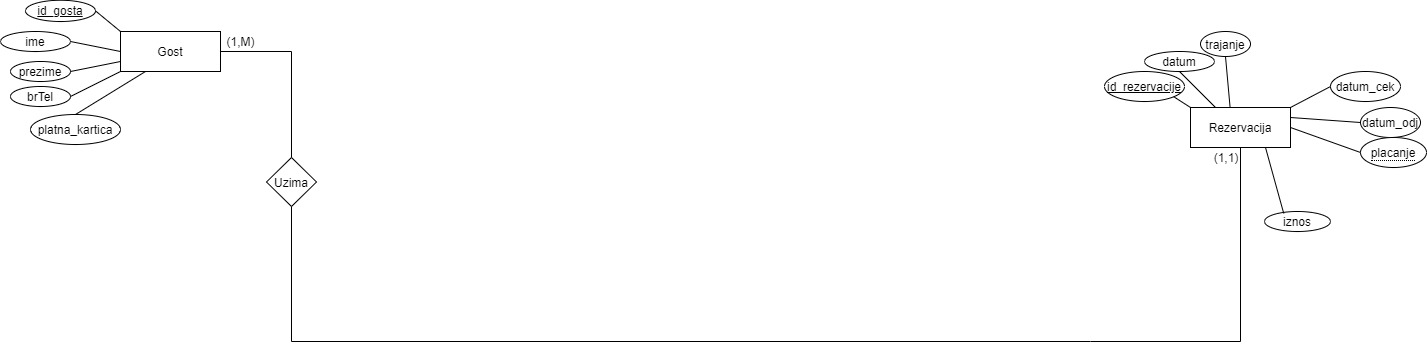
Ova veza povezuje entitete *Gost i Grad.*Svaki gost živi u jednom gradu(1,1),dok u gradu može živeti više gostiju,a ne mora ni jedan(0,M).



Slika br.25

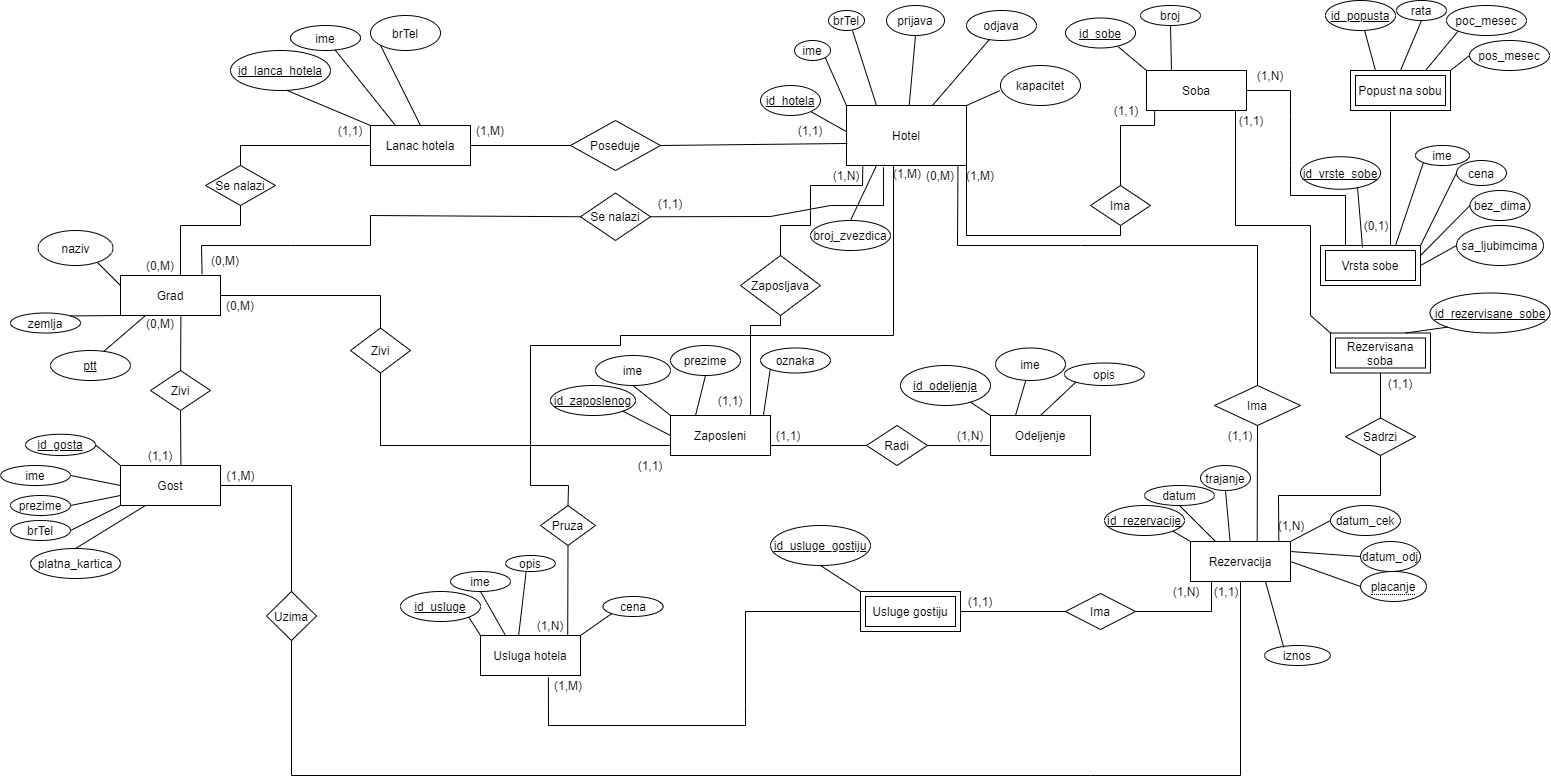
* **Veza Gost-Rezervacija *(Uzima)***

Ova veza povezuje entitete *Gost i Rezervacija.*Gost može da uzme jednu ili više od jedne rezervacije(više rezervacija,1,M),dok jedna rezervacija može biti uzeta samo od jednog gosta(1,1).



Slika br.26

### 2.3 Kompletan ER dijagram



Slika br.27

## 3. Logička šema relacione baze podataka

Da bi se dobila logička šema relacione baze podataka,moraju svi entiteti i sve veze da se prevedu iz modela objekti-veze na njihov ekvivalentni oblik u relacionom modelu –relacije i međurelaciona ograničenja.Elementi ER dijagrama se prevode u relacioni oblik sledećim redosledom: Entiteti,identifikacione veze slabih entiteta,gerundi i na kraju ostale veze.

### 3.1 Entiteti

Svi entiteti postaju nezavisne šeme relacija.Ime tipa entiteta postaje ime šeme relacije.Obeležja tipa objekta su obeležja šeme relacije.Identifikator entiteta postaje primarni ključ šeme relacije.

Lanac hotela(id\_lanca\_hotela,ime,brTel)

Hotel(id\_hotela,ime,brTel,prijava,odjava,kapacitet,broj\_zvezdica)

Zaposleni(id\_zaposlenog,ime,prezime,oznaka)

Odeljenje(id\_odeljenja,ime,opis)

Rezervacija(id\_rezervacije,datum,trajanje,datum\_cek,datum\_odj,placanje,iznos)

Usluga hotela(id\_usluge,ime,opis,cena)

Soba(id\_sobe,broj)

Gost(id\_gosta,ime,prezime,brTel,platna\_kartica)

Grad(ptt,naziv,zemlja)

### 3.2 Identifikacione veze slabih entiteta

Usluge gostiju(id\_usluge,id\_usluge\_gostiju)

Vrsta sobe(id\_sobe,id\_vrste\_sobe,ime,cena,bez\_dima,sa\_ljubimcima)

Popust na sobu(id\_sobe,id\_vrste\_sobe,id\_popusta,rata,poc\_mesec,pos\_mesec)

Rezervisana soba(id\_sobe,id\_rezervisane\_sobe)

### 3.3 Veze

Usled pravila kardinalnosti pojedine veze će prouzrokovati korigovanje određenih relacija,a pojedine veze će postati zasebne relacije.

Korigovane relacije:

Lanac hotela(id\_lanca\_hotela,ime,brTel,ptt)

Hotel(id\_hotela,ime,brTel,prijava,odjava,kapacitet,broj\_zvezdica,id\_lacna\_hotela,ptt)

Zaposleni(id\_zaposlenog,ime,prezime,oznaka,id\_hotela,id\_odeljenja,ptt)

Rezervacija(id\_rezervacije,datum,trajanje,datum\_cek,datum\_odj,placanje,iznos,id\_gosta,id\_hotela)

Soba(id\_sobe,broj,id\_hotela)

Gost(id\_gosta,ime,prezime,brTel,platna\_kartica,ptt)

Veze koje postaju posebne relacije:

Pruza(id\_hotela,id\_usluge)

### 3.4 Međurelaciona ograničenja

Međurelaciona ograničenja se definišu za sve nasleđene atribute neke relacije-da moraju biti podskup skupa vrednosti nasleđenog atributa u originalnoj relaciji od koje je taj atribut nasleđen.To uključuje sve strane ključeve svake relacije.

Sada ćemo definisati međurelaciona ograničenja:

Lanac hotela(ptt) ⊆ Grad(ptt)

Hotel(id\_lanca\_hotela) ⊆ Lanac hotela(id\_lanca\_hotela)

Hotel(ptt) ⊆ Grad(ptt)

Zaposleni(id\_hotela) ⊆ Hotel(id\_hotela)

Zaposleni(id\_odeljenja) ⊆ Odeljenje(id\_odeljenja)

Zaposleni(ptt) ⊆ Grad(ptt)

Rezervacija(id\_hotela) ⊆ Hotel(id\_hotela)

Rezervacija(id\_gosta) ⊆ Gost(id\_gosta)

Soba(id\_hotela) ⊆ Hotel(id\_hotela)

Vrsta sobe(id\_sobe) ⊆ Soba(id\_sobe)

Popust na sobu(id\_vrste\_sobe) ⊆ Vrsta sobe(id\_vrste\_sobe)

Popust na sobu(id\_sobe) ⊆ Soba(id\_sobe)

Rezervisana soba(id\_sobe) ⊆ Soba(id\_sobe)

Rezervisana soba(id\_rezervacije) ⊆ Rezervacija(id\_rezervacije)

Usluge gostiju(id\_usluge) ⊆ Usluga hotela(id\_usluge)

Usluge gostiju(id\_rezervacije) ⊆ Rezervacija (id\_rezervacije)

Gost(ptt) ⊆ Grad(ptt)

### 3.5 Kompletna logička šema

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

S={Odeljenje(id\_odeljenja,ime,opis),

Usluga hotela(id\_usluge,ime,opis,cena),

Grad(ptt,naziv,zemlja),

Lanac hotela(id\_lanca\_hotela,ime,brTel,ptt),

Hotel(id\_hotela,ime,brTel,prijava,odjava,kapacitet,broj\_zvezdica,id\_lacna\_hotela,ptt),

Zaposleni(id\_zaposlenog,ime,prezime,oznaka,id\_hotela,id\_odeljenja,ptt),

Rezervacija(id\_rezervacije,datum,trajanje,datum\_cek,datum\_odj,placanje,iznos,id\_gosta,id\_hotela),

Soba(id\_sobe,broj,id\_hotela),

Gost(id\_gosta,ime,prezime,brTel,platna\_kartica,ptt),

Usluge gostiju(id\_usluge,id\_usluge\_gostiju),

Vrsta sobe(id\_sobe,id\_vrste\_sobe,ime,cena,bez\_dima,sa\_ljubimcima),

Popust na sobu(id\_sobe,id\_vrste\_sobe,id\_popusta,rata,poc\_mesec,pos\_mesec),

Rezervisana soba(id\_sobe,id\_rezervisane\_sobe),

Pruza(id\_hotela,id\_usluge)}

I={ Lanac hotela(ptt) ⊆ Grad(ptt),

Hotel(id\_lanca\_hotela) ⊆ Lanac hotela(id\_lanca\_hotela),

Hotel(ptt) ⊆ Grad(ptt),

Zaposleni(id\_hotela) ⊆ Hotel(id\_hotela),

Zaposleni(id\_odeljenja) ⊆ Odeljenje(id\_odeljenja),

Zaposleni(ptt) ⊆ Grad(ptt),

Rezervacija(id\_hotela) ⊆ Hotel(id\_hotela),

Rezervacija(id\_gosta) ⊆ Gost(id\_gosta),

Soba(id\_hotela) ⊆ Hotel(id\_hotela),

Vrsta sobe(id\_sobe) ⊆ Soba(id\_sobe),

Popust na sobu(id\_vrste\_sobe) ⊆ Vrsta sobe(id\_vrste\_sobe),

Popust na sobu(id\_sobe) ⊆ Soba(id\_sobe),

Rezervisana soba(id\_sobe) ⊆ Soba(id\_sobe),

Rezervisana soba(id\_rezervacije) ⊆ Rezervacija(id\_rezervacije),

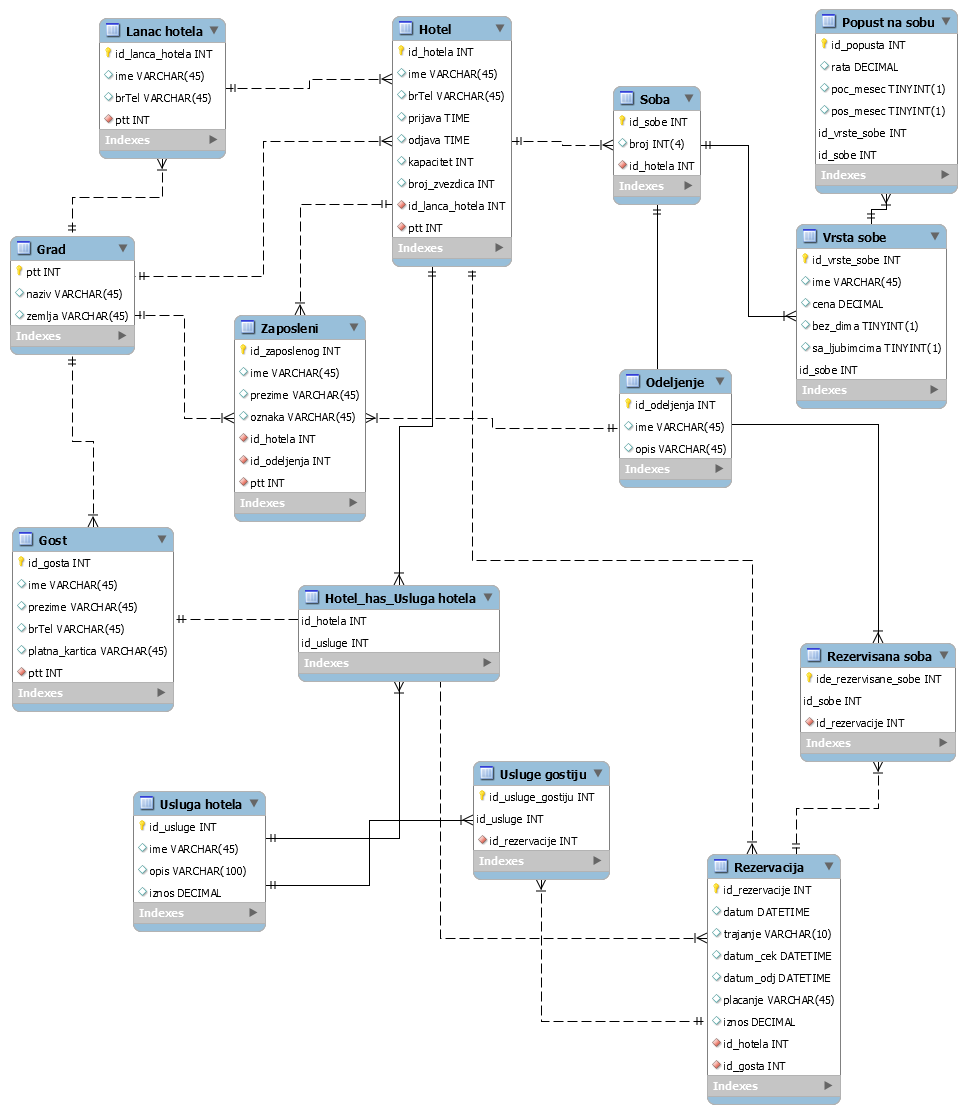
Usluge gostiju(id\_usluge) ⊆ Usluga hotela(id\_usluge),

Usluge gostiju(id\_rezervacije) ⊆ Rezervacija (id\_rezervacije),

Gost(ptt) ⊆ Grad(ptt)}

## 4.Fizička šema relacione baze podataka

U nastavku sledi fizička šema relacione baze podataka koju smo kreirali u programu koji se zove “MySqli Workbench”.



Slika br.28

## 5.Implementacija projektovane baze podataka u MySQL

MySQL Workbench ima opciju da generiše SQL kod koji implementira projektovanu fizičku šemu baze podataka.Sledi kompletan generisan SQL kod,koji implementira razmatrani realni sistem:

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\_SUBSTITUTION';

-- Schema mydb

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

USE `mydb` ;

-- Table `mydb`.`Grad`

CREATE TABLE IF NOT EXISTS `mydb`.`Grad` (

`ptt` INT NOT NULL,

`naziv` VARCHAR(45) NULL,

`zemlja` VARCHAR(45) NULL,

PRIMARY KEY (`ptt`))

ENGINE = InnoDB;

-- Table `mydb`.`Lanac hotela`

CREATE TABLE IF NOT EXISTS `mydb`.`Lanac hotela` (

`id\_lanca\_hotela` INT NOT NULL,

`ime` VARCHAR(45) NULL,

`brTel` VARCHAR(45) NULL,

`ptt` INT NOT NULL,

PRIMARY KEY (`id\_lanca\_hotela`),

INDEX `fk\_Lanac hotela\_Grad\_idx` (`ptt` ASC) ,

CONSTRAINT `fk\_Lanac hotela\_Grad`

FOREIGN KEY (`ptt`)

REFERENCES `mydb`.`Grad` (`ptt`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- Table `mydb`.`Hotel`

CREATE TABLE IF NOT EXISTS `mydb`.`Hotel` (

`id\_hotela` INT NOT NULL,

`ime` VARCHAR(45) NULL,

`brTel` VARCHAR(45) NULL,

`prijava` TIME NULL,

`odjava` TIME NULL,

`kapacitet` INT NULL,

`broj\_zvezdica` INT NULL,

`id\_lanca\_hotela` INT NOT NULL,

`ptt` INT NOT NULL,

PRIMARY KEY (`id\_hotela`),

INDEX `fk\_Hotel\_Lanac hotela1\_idx` (`id\_lanca\_hotela` ASC) ,

INDEX `fk\_Hotel\_Grad1\_idx` (`ptt` ASC) ,

CONSTRAINT `fk\_Hotel\_Lanac hotela1`

FOREIGN KEY (`id\_lanca\_hotela`)

REFERENCES `mydb`.`Lanac hotela` (`id\_lanca\_hotela`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Hotel\_Grad1`

FOREIGN KEY (`ptt`)

REFERENCES `mydb`.`Grad` (`ptt`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- Table `mydb`.`Odeljenje`

CREATE TABLE IF NOT EXISTS `mydb`.`Odeljenje` (

`id\_odeljenja` INT NOT NULL,

`ime` VARCHAR(45) NULL,

`opis` VARCHAR(45) NULL,

PRIMARY KEY (`id\_odeljenja`))

ENGINE = InnoDB;

-- Table `mydb`.`Zaposleni`

CREATE TABLE IF NOT EXISTS `mydb`.`Zaposleni` (

`id\_zaposlenog` INT NOT NULL,

`ime` VARCHAR(45) NULL,

`prezime` VARCHAR(45) NULL,

`oznaka` VARCHAR(45) NULL,

`id\_hotela` INT NOT NULL,

`id\_odeljenja` INT NOT NULL,

`ptt` INT NOT NULL,

PRIMARY KEY (`id\_zaposlenog`),

INDEX `fk\_Zaposleni\_Hotel1\_idx` (`id\_hotela` ASC) ,

INDEX `fk\_Zaposleni\_Odeljenje1\_idx` (`id\_odeljenja` ASC) ,

INDEX `fk\_Zaposleni\_Grad1\_idx` (`ptt` ASC) ,

CONSTRAINT `fk\_Zaposleni\_Hotel1`

FOREIGN KEY (`id\_hotela`)

REFERENCES `mydb`.`Hotel` (`id\_hotela`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Zaposleni\_Odeljenje1`

FOREIGN KEY (`id\_odeljenja`)

REFERENCES `mydb`.`Odeljenje` (`id\_odeljenja`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Zaposleni\_Grad1`

FOREIGN KEY (`ptt`)

REFERENCES `mydb`.`Grad` (`ptt`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- Table `mydb`.`Gost`

CREATE TABLE IF NOT EXISTS `mydb`.`Gost` (

`id\_gosta` INT NOT NULL,

`ime` VARCHAR(45) NULL,

`prezime` VARCHAR(45) NULL,

`brTel` VARCHAR(45) NULL,

`platna\_kartica` VARCHAR(45) NULL,

`ptt` INT NOT NULL,

PRIMARY KEY (`id\_gosta`),

INDEX `fk\_Gost\_Grad1\_idx` (`ptt` ASC) ,

CONSTRAINT `fk\_Gost\_Grad1`

FOREIGN KEY (`ptt`)

REFERENCES `mydb`.`Grad` (`ptt`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- Table `mydb`.`Rezervacija`

CREATE TABLE IF NOT EXISTS `mydb`.`Rezervacija` (

`id\_rezervacije` INT NOT NULL,

`datum` DATETIME NULL,

`trajanje` VARCHAR(10) NULL,

`datum\_cek` DATETIME NULL,

`datum\_odj` DATETIME NULL,

`placanje` VARCHAR(45) NULL,

`iznos` DECIMAL NULL,

`id\_hotela` INT NOT NULL,

`id\_gosta` INT NOT NULL,

PRIMARY KEY (`id\_rezervacije`),

INDEX `fk\_Rezervacija\_Hotel1\_idx` (`id\_hotela` ASC) ,

INDEX `fk\_Rezervacija\_Gost1\_idx` (`id\_gosta` ASC) ,

CONSTRAINT `fk\_Rezervacija\_Hotel1`

FOREIGN KEY (`id\_hotela`)

REFERENCES `mydb`.`Hotel` (`id\_hotela`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Rezervacija\_Gost1`

FOREIGN KEY (`id\_gosta`)

REFERENCES `mydb`.`Gost` (`id\_gosta`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- Table `mydb`.`Usluga hotela`

CREATE TABLE IF NOT EXISTS `mydb`.`Usluga hotela` (

`id\_usluge` INT NOT NULL,

`ime` VARCHAR(45) NULL,

`opis` VARCHAR(100) NULL,

`iznos` DECIMAL NULL,

PRIMARY KEY (`id\_usluge`))

ENGINE = InnoDB;

-- Table `mydb`.`Usluge gostiju`

CREATE TABLE IF NOT EXISTS `mydb`.`Usluge gostiju` (

`id\_usluge\_gostiju` INT NOT NULL,

`id\_usluge` INT NOT NULL,

`id\_rezervacije` INT NOT NULL,

PRIMARY KEY (`id\_usluge\_gostiju`, `id\_usluge`),

INDEX `fk\_Usluge gostiju\_Usluga hotela1\_idx` (`id\_usluge` ASC) ,

INDEX `fk\_Usluge gostiju\_Rezervacija1\_idx` (`id\_rezervacije` ASC) ,

CONSTRAINT `fk\_Usluge gostiju\_Usluga hotela1`

FOREIGN KEY (`id\_usluge`)

REFERENCES `mydb`.`Usluga hotela` (`id\_usluge`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Usluge gostiju\_Rezervacija1`

FOREIGN KEY (`id\_rezervacije`)

REFERENCES `mydb`.`Rezervacija` (`id\_rezervacije`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- Table `mydb`.`Soba`

CREATE TABLE IF NOT EXISTS `mydb`.`Soba` (

`id\_sobe` INT NOT NULL,

`broj` INT(4) NULL,

`id\_hotela` INT NOT NULL,

PRIMARY KEY (`id\_sobe`),

INDEX `fk\_Soba\_Hotel1\_idx` (`id\_hotela` ASC) ,

CONSTRAINT `fk\_Soba\_Hotel1`

FOREIGN KEY (`id\_hotela`)

REFERENCES `mydb`.`Hotel` (`id\_hotela`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- Table `mydb`.`Vrsta sobe`

CREATE TABLE IF NOT EXISTS `mydb`.`Vrsta sobe` (

`id\_vrste\_sobe` INT NOT NULL,

`ime` VARCHAR(45) NULL,

`cena` DECIMAL NULL,

`bez\_dima` TINYINT(1) NULL,

`sa\_ljubimcima` TINYINT(1) NULL,

`id\_sobe` INT NOT NULL,

PRIMARY KEY (`id\_vrste\_sobe`, `id\_sobe`),

INDEX `fk\_Vrsta sobe\_Soba1\_idx` (`id\_sobe` ASC) ,

CONSTRAINT `fk\_Vrsta sobe\_Soba1`

FOREIGN KEY (`id\_sobe`)

REFERENCES `mydb`.`Soba` (`id\_sobe`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- Table `mydb`.`Popust na sobu`

CREATE TABLE IF NOT EXISTS `mydb`.`Popust na sobu` (

`id\_popusta` INT NOT NULL,

`rata` DECIMAL NULL,

`poc\_mesec` TINYINT(1) NULL,

`pos\_mesec` TINYINT(1) NULL,

`id\_vrste\_sobe` INT NOT NULL,

`id\_sobe` INT NOT NULL,

PRIMARY KEY (`id\_popusta`, `id\_vrste\_sobe`, `id\_sobe`),

INDEX `fk\_Popust na sobu\_Vrsta sobe1\_idx` (`id\_vrste\_sobe` ASC, `id\_sobe` ASC) ,

CONSTRAINT `fk\_Popust na sobu\_Vrsta sobe1`

FOREIGN KEY (`id\_vrste\_sobe` , `id\_sobe`)

REFERENCES `mydb`.`Vrsta sobe` (`id\_vrste\_sobe` , `id\_sobe`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- Table `mydb`.`Rezervisana soba`

CREATE TABLE IF NOT EXISTS `mydb`.`Rezervisana soba` (

`ide\_rezervisane\_sobe` INT NOT NULL,

`id\_sobe` INT NOT NULL,

`id\_rezervacije` INT NOT NULL,

PRIMARY KEY (`ide\_rezervisane\_sobe`, `id\_sobe`),

INDEX `fk\_Rezervisana soba\_Soba1\_idx` (`id\_sobe` ASC) ,

INDEX `fk\_Rezervisana soba\_Rezervacija1\_idx` (`id\_rezervacije` ASC) ,

CONSTRAINT `fk\_Rezervisana soba\_Soba1`

FOREIGN KEY (`id\_sobe`)

REFERENCES `mydb`.`Soba` (`id\_sobe`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Rezervisana soba\_Rezervacija1`

FOREIGN KEY (`id\_rezervacije`)

REFERENCES `mydb`.`Rezervacija` (`id\_rezervacije`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- Table `mydb`.`Hotel\_has\_Usluga hotela`

CREATE TABLE IF NOT EXISTS `mydb`.`Hotel\_has\_Usluga hotela` (

`id\_hotela` INT NOT NULL,

`id\_usluge` INT NOT NULL,

PRIMARY KEY (`id\_hotela`, `id\_usluge`),

INDEX `fk\_Hotel\_has\_Usluga hotela\_Usluga hotela1\_idx` (`id\_usluge` ASC) ,

INDEX `fk\_Hotel\_has\_Usluga hotela\_Hotel1\_idx` (`id\_hotela` ASC) ,

CONSTRAINT `fk\_Hotel\_has\_Usluga hotela\_Hotel1`

FOREIGN KEY (`id\_hotela`)

REFERENCES `mydb`.`Hotel` (`id\_hotela`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Hotel\_has\_Usluga hotela\_Usluga hotela1`

FOREIGN KEY (`id\_usluge`)

REFERENCES `mydb`.`Usluga hotela` (`id\_usluge`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

SET SQL\_MODE=@OLD\_SQL\_MODE;

SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;

Slede primeri popunjavanja SQL tabela sa testnim primerima:

insert into Grad (ptt,naziv,zemlja) values (11000,"Beograd","Srbija"),(34000,"Kragujevac","Srbija"),(21000,"Novi Sad","Srbija");

insert into `Lanac hotela` (id\_lanca\_hotela,ime,brTel,ptt) values (1,"Elitni hoteli","066-369678",21000),(2,"Prestižni hoteli","069-89731331",11000),(3,"Lanac hotela IZVOR","066-2456789",34000);

insert into Hotel (id\_hotela,ime,brTel,prijava,odjava,kapacitet,broj\_zvezdica,id\_lanca\_hotela,ptt) values (1,"Palic","066-369078",'12:00:00','23:00:00',50,2,1,21000),(2,"ZLatibor","063-339078",'10:00:00','21:00:00',25,3,2,11000),(3,"Gorske Zore","062-339078",'05:00:00','12:00:00',70,3,3,34000);

insert into Odeljenje(id\_odeljenja,ime,opis) values (1,'Kuhinja','Kuvanje'),(2,'Ciscenje','Brisanje i pranje'),(3,'Prednje Osoblje','Rukovanje rezervacijama i resavanje upita'),(4,'Menadzment','Obradjuje kupca i resava usluge'),(5,'Komutacija','Uzimanje i odbacivanje');

insert into Zaposleni (id\_zaposlenog,ime,prezime,oznaka,id\_hotela,id\_odeljenja,ptt) values (1,'Marko','Nikolic','Konobar',1,1,34000),(2,'Toma','Peric','Menadzer',3,3,11000),(3,'David','Lazic','Blagajnik',2,2,11000),(4,'Milomir','Antic','Kuvar',2,2,21000),(5,'Anica','Dobric','Menadzer',3,3,34000);

insert into Gost(id\_gosta,ime,prezime,brTel,platna\_kartica,ptt) values (1,'Mirko','Kodic','132-456-8564',NULL,11000),(2,'Neda','Rakic','132-456-8564',NULL,11000),(3,'Milomir','Rajovic','152-456-8564',NULL,34000),(4,'Bata','Mikic','0664789876',NULL,11000),(5,'Goran','Ristic','0653456789',NULL,21000);

insert into Rezervacija(id\_rezervacije,datum,trajanje,datum\_cek,datum\_odj,placanje,iznos,id\_hotela,id\_gosta) values (1, '2018-08-08 00:00:00', '5', '2018-08-10 12:00:00', '2018-08-15 23:00:00', 'Kes','3500',1,1),(2, '2018-07-08 00:00:00', '8', '2018-08-10 12:00:00', '2018-08-18 23:00:00', 'Kes','15000',1,4),(3, '2019-03-08 00:00:00', '12', '2018-08-10 12:00:00', '2018-08-22 23:00:00', 'Kes','25000',3,5);

insert into `Usluga hotela` (id\_usluge,ime,opis,iznos) values (1,'24-hour Room Service','Postoji usluga koja je na raspolaganju gostima 24h',2000),(2,'Menjacnica','Mogucnost razmene stranih valuta na raspolaganju',500),(3,'Perionica Vesa','Pranje,susenje vesa svakoga dana',1000),(4,'Zabavna soba','Knjige i gledanje filmova',3500),(5,'Bazen','Bazen je pristupacan svim gostima',1000),(6,'Teretana','24h',1500);

insert into `Usluge gostiju` (id\_usluge\_gostiju,id\_usluge,id\_rezervacije) values (1,1,1),(2,2,2),(3,2,1),(3,4,3),(4,4,1);

insert into Soba (id\_sobe,broj,id\_hotela) values (1,1,1),(2,17,2),(3,33,3),(4,4,1),(5,55,3),(6,16,2);

insert into `Vrsta sobe` (id\_vrste\_sobe,ime,cena,bez\_dima,sa\_ljubimcima,id\_sobe) values (1, 'Standardna soba','5000',0,1,1),(2, 'Standardna dvokrevetna soba','7000',1,1,2),(3, 'Luksuzna soba','10000',0,0,3),(4, 'Klub soba','12000',1,1,4);

insert into `Popust na sobu` (id\_popusta,rata,poc\_mesec,pos\_mesec,id\_vrste\_sobe,id\_sobe) values (1,2000,1,3,1,1),(2,3000,1,5,2,2),(3,5000,1,8,3,3),(4,5500,1,11,4,4);

insert into `Rezervisana soba`(ide\_rezervisane\_sobe,id\_sobe,id\_rezervacije) values (1,1,1);

insert into `Rezervisana soba`(ide\_rezervisane\_sobe,id\_sobe,id\_rezervacije) values (2,2,2);

insert into `Rezervisana soba`(ide\_rezervisane\_sobe,id\_sobe,id\_rezervacije) values (3,3,3);

insert into `Hotel\_has\_Usluga hotela` (id\_hotela,id\_usluge) values (1,2),(2,1),(3,3);

Slede primeri upita nad SQL testnim podacima:

Upit koji daje broj hotela:

SELECT COUNT(\*) FROM HOTEL;

Prikaz brojeva soba koji imaju id\_hotela 2:

SELECT BROJ

FROM SOBA

WHERE ID\_HOTELA=2;

Prikaz imena zaposlenih koji rade na poziciji menazdera:

SELECT IME,PREZIME

FROM ZAPOSLENI

WHERE OZNAKA="Menadzer";

Prikaz imena I prezimena gostiju koji žive u određenom gradu:

SELECT IME,PREZIME

FROM GOST G ,GRAD GR

WHERE G.PTT=GR.PTT AND NAZIV="Kragujevac";

Prikaz svih podataka rezervacije koja je zavedena u određenom hotelu sa dve zvezdice:

SELECT \*

FROM REZERVACIJA

WHERE ID\_HOTELA=(SELECT ID\_HOTELA

FROM HOTEL

WHERE BROJ\_ZVEZDICA=2);

Prikaz kapaciteta hotela koji pripadaju Lancu hotela koji imaju vrednost druge cifre ptt-a 1:

SELECT KAPACITET

FROM HOTEL H,`Lanac hotela` L

WHERE H.ID\_LANCA\_HOTELA=L.ID\_LANCA\_HOTELA

AND L.PTT LIKE "\_1%";

Prikaz broja sobe koju ima mesecnu ratu manju od 6000din:

SELECT BROJ

FROM SOBA S,`Popust na sobu` P

WHERE S.ID\_SOBE=P.ID\_SOBE AND P.RATA < 6000;

Prikaz imena i cena vrsta soba koja vraca cenu sobe u opadajucem poretku

SELECT IME,CENA

FROM `Vrsta sobe`

GROUP BY IME

ORDER BY CENA DESC;

## 6. Literatura

1.Opšta poznanstva i pojmovi o hotelu: <https://sr.wikipedia.org/wiki/%D0%A5%D0%BE%D1%82%D0%B5%D0%BB>

2.Vežbe i predavanja sa moodle-a: <http://moodle.fink.rs/course/view.php?id=977>