

Univerzitet u Kragujevcu
Fakultet inženjerskih nauka



Baze podataka

Projektni zadatak:

IS Auto Kuće

Student:

Đorđe Molnar 660/2019

Predmetni nastavnici:

Milan Erić

Aleksandar Đorđević

1. Uvod

Cilj projektnog zadatka je upoznavanje sa pojmovima vezanim za relacione baze podataka, kao i predstavljanje konkretne baze na primeru baze podataka potrebne za rad jedne Auto Kuće.

U početnom delu projektnog zadatka ukratko će biti objašnjeni osnovni pojmovi vezani za baze podataka, kao što su relacioni model, predikatska logika, teorija skupova..

2. Osnovni pojmovi baze podataka

Osnovne strukture i koncepti su:

- Relacioni model;
- Baza i šema;
- Tabele, redovi i kolone;
- Ključevi;
- Relacije.

2.1. Relacioni model

Relacioni model je semantički model za upravljanje podacima, koji se bazira na teoriji skupova i predikatskoj logici. Kreiran je od strane dr Edgara Koda, britanskog informatičara, i prva verzija ovog modela je predstavljena 1969. godine. Cilj modela je da obezbedi konzistentnost i eliminiše ponavljanje podataka.

Prema teoriji skupova, skup opisuje objekte, tačnije članove tog skupa sa zajedničkim osobinama, kao jednu celinu. Posmatranje samog skupa i njegovih osobina, predstavlja pojam entiteta u bazama podataka. Osobine koje poseduje skup se opisuju atributima jednog entiteta.

2.2. Baze podataka i šema

Baza podataka je bilo koja uredjena kolekcija podataka. To ne mora biti samo baza u elektronskoj formi, već bilo koja organizovana kolekcija podataka, kao što su kartoni pacijenata, adresari, Excel tabele ili obični tekstualni fajlovi. Baza podataka predstavlja glavni kontejner koji se koristi za grupisanje ostalih objekata unutar iste.

2.3. Tabele, redovi i kolone

U relacionim bazama, tabela predstavlja prezentaciju nekog koncepta, odnosno entiteta o kome želimo da čuvamo informacije. Tabele mogu sadržati podatke o osobama, mestima, stvarima ili nekim apstraktnim objektima i idejama. Entitet je opisan atributima, pri čemu treba imati u vidu da je tabela samo koncept – dizajn podataka – lišen bilo kakvih detalja implementacije, što je u osnovi relacionog modela podataka.

2.4. Ključevi

Tabele relacionog modela mogu sadržati samo jedinstvene instance određenog entiteta u svojim redovima. Nije dozvoljeno dupliranje instanci, jer tada ne bi postojao kriterijum za izdvajanje pojedinačnog entiteta iz baze.

Da se ne bi desilo dupliranje podataka, koriste se primarni ključevi. Ključ se može definisati za jednu ili više kolona, tako da svaka instanca ima svoju jedinstvenu vrednost ključa. Postoje dve vrste ključa – primarni i spoljni.

Primarni ključ je jedinstveni identifikator jedne instance entiteta. Može biti prirodni, kada se za ključ proglasi neki od atributa čija je vrednost garantovano jedinstvena, ili surogat, neki veštački ključ koji garantuje jedinstvenost entiteta.

2.5. Relacije

Prilikom kreiranja tabela, za dobijanje strukture podataka, ponekad je potrebno uključiti i attribute iz drugih tabela. Ovo se postiže definisanjem relacija između tabela i njihovim međusobnim povezivanjem. Povezivanje tabela se vrši preko spoljnog ključa, koji zapravo predstavlja primarni ključ nezavisne table uveden u zavisnu tabelu. U relaciji mogu biti jedna, dve ili više tabela.

3. ER dijagram

Modelovanje realnog sistema započinje se identifikovanjem entiteta koji postoje u sistemu, nakon toga se određuju veze između postojećih entiteta.

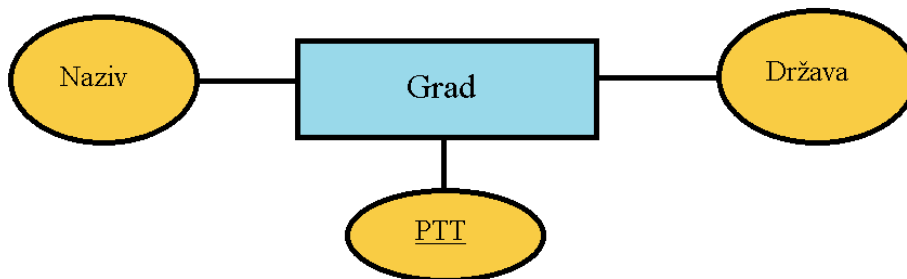
3.1 Entiteti i atributi

Entiteti su objekti modela koji su jednoznačno određeni. Grafički se prikazuju pravougaonikom u okviru koga se upisuje naziv tipa entiteta.

Entiteti poseduju svojstva (osobine), gde se opis jedne osobine sastoji od *atributa* kojim je jednoznačno određena vrsta osobine i vrednosti atribuda.

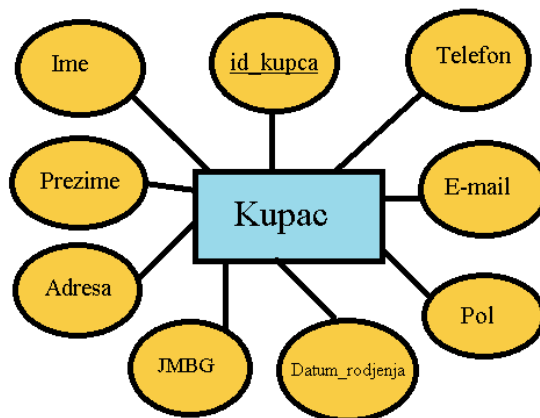
- Grad

Entitet *Grad* je predstavljen svojim identifikatorom (PTT), nazivom grada i državom u kojoj se grad nalazi.



- Kupac

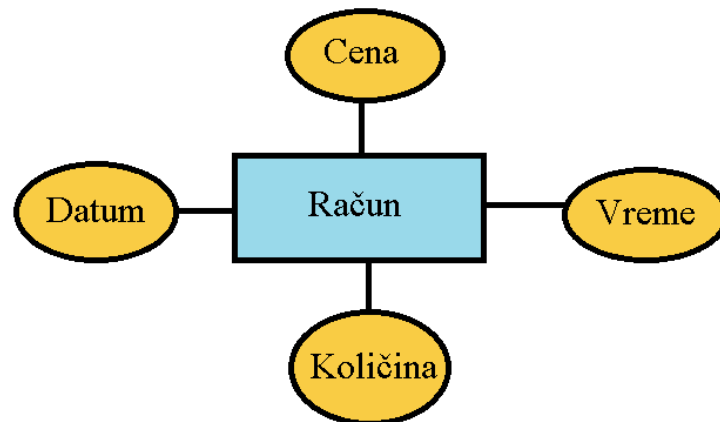
Entitet *Kupac* sadrži svoj identifikator (*id_kupca*), ime i prezime, adresu, jmbg, datum rođenja, pol, e-mail i broj telefona kupca.



- Račun

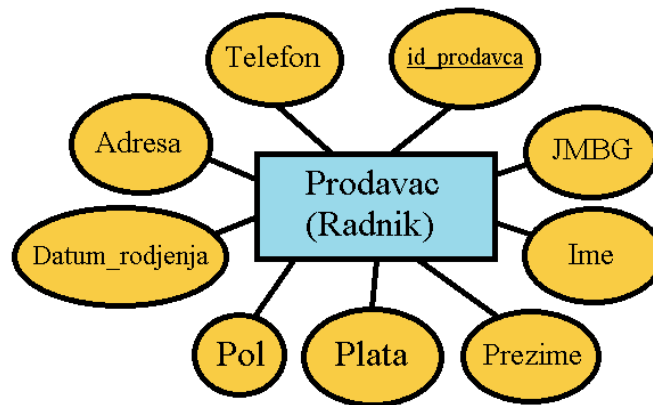
Entitet *Račun* predstavlja fiskalni račun (ili fakturu) koju kupac dobija od salona automobila.

Entitet se sastoji iz datuma i tačnog vremena kupovine, količine koju je kupac pazario, kao i ukupne cene koju kupac treba da plati.



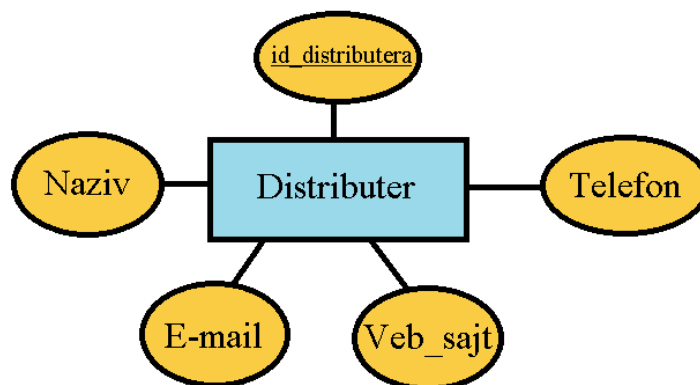
- Prodavac

Entitet *prodavac* (ili radnik) predstavlja zaposlenog radnika u salonu automobila i sadrži svoj identifikator (id_prodavca), ime i prezime, adresu, jmbg, datum rođenja, pol, broj telefona i platu.



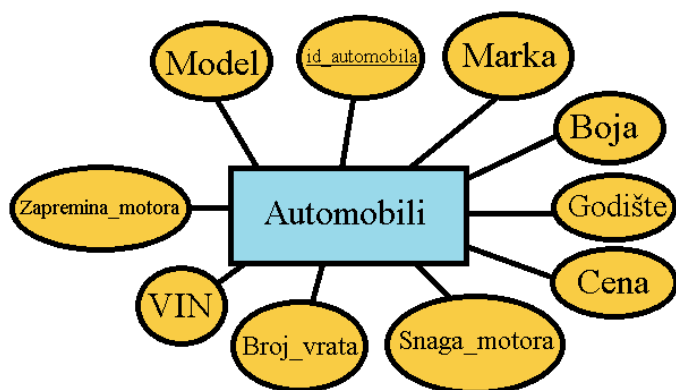
- Distributer

Entitet *distributer* je okarakterisan svojim identifikatorom (id_distributera), nazivom distributera, njegovom e-mail adresom, veb sajtom, i brojem telefona.



- Automobili

Entitet *automobili* predstavlja detaljan opis automobila koji se nalaze u ponudi salona i sastoji se od identifikatora (id_automobila), marke i modela, boje, godista, zapremine i snage motora, broja vrata, VIN oznake (broj šasijske) i cene automobila.

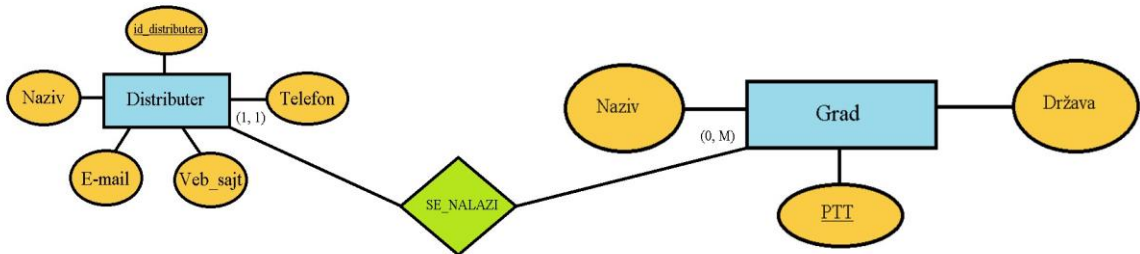


3.2 Veze

Nakon definisanja entiteta kao I njihovih atributa, sada je na redu definisanje povezanosti entiteta, tj. veza. Priroda relacije nasleđivanja je takva da podentitet pored atributa njihovih nadtipova, nasleđuju sve veze koje se odnose na njihove nadtipove. Sledi opis veza između entitet u sistemu IS Auto kuće.

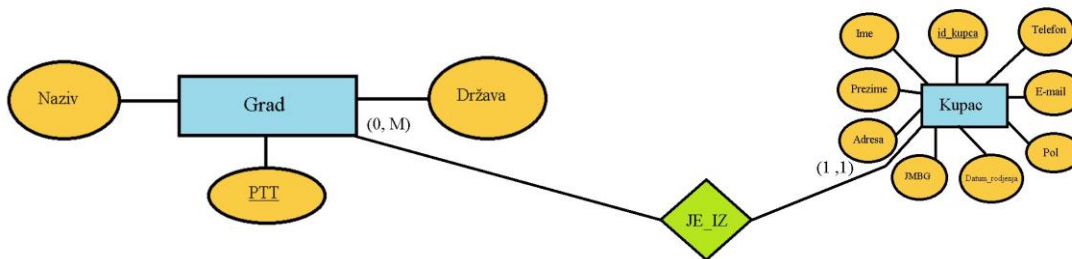
- Veza “SE_NALAZI”

Veza *SE_NALAZI* povezuje entitet *Grad* sa entitetom *Distributer*. Distributer se nalazi u jednom i samo jednom gradu, dok jedan grad može imati više distributera.



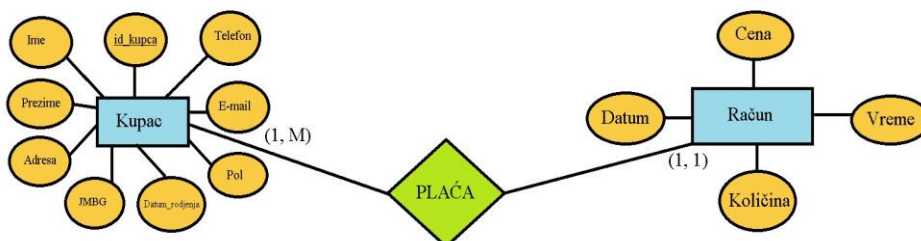
- Veza „JE_IZ“

Veza *JE_IZ* povezuje entitet *Grad* sa entitetom *Kupac*, tako da je svaki kupac pojedinačno iz jednog i samo jednog grada, a jedna grad ne mora da ima niti jednog kupca, a može ih imati i više.



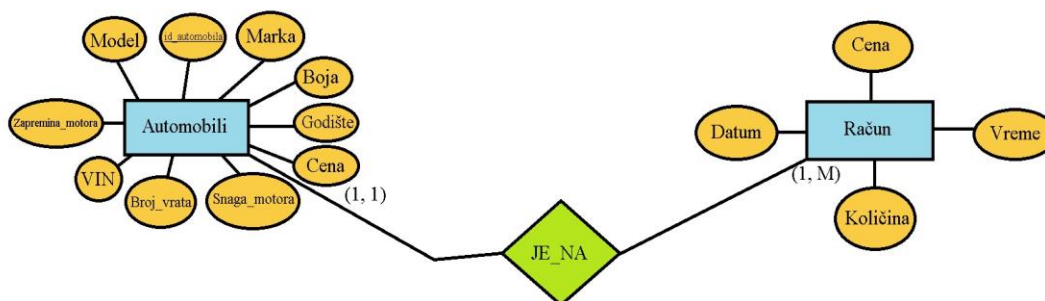
- Veza „PLAĆA“

Veza *PLAĆA* povezuje entitet *Kupac* sa entitetom *Račun*, tako da svaki kupac pojedinačno plaća jedan ili više računa, dok jedan račun može platiti samo jedan kupac.



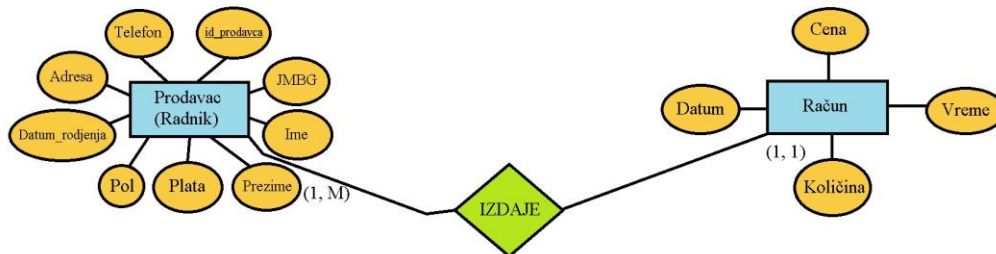
- Veza “JE_NA”

Veza *JE_NA* povezuje entitet *Automobili* sa entitetom *Račun*, tako da se jedan automobil nalazi samo na jednom računu, dok se na jednom računu može nalaziti više automobila.

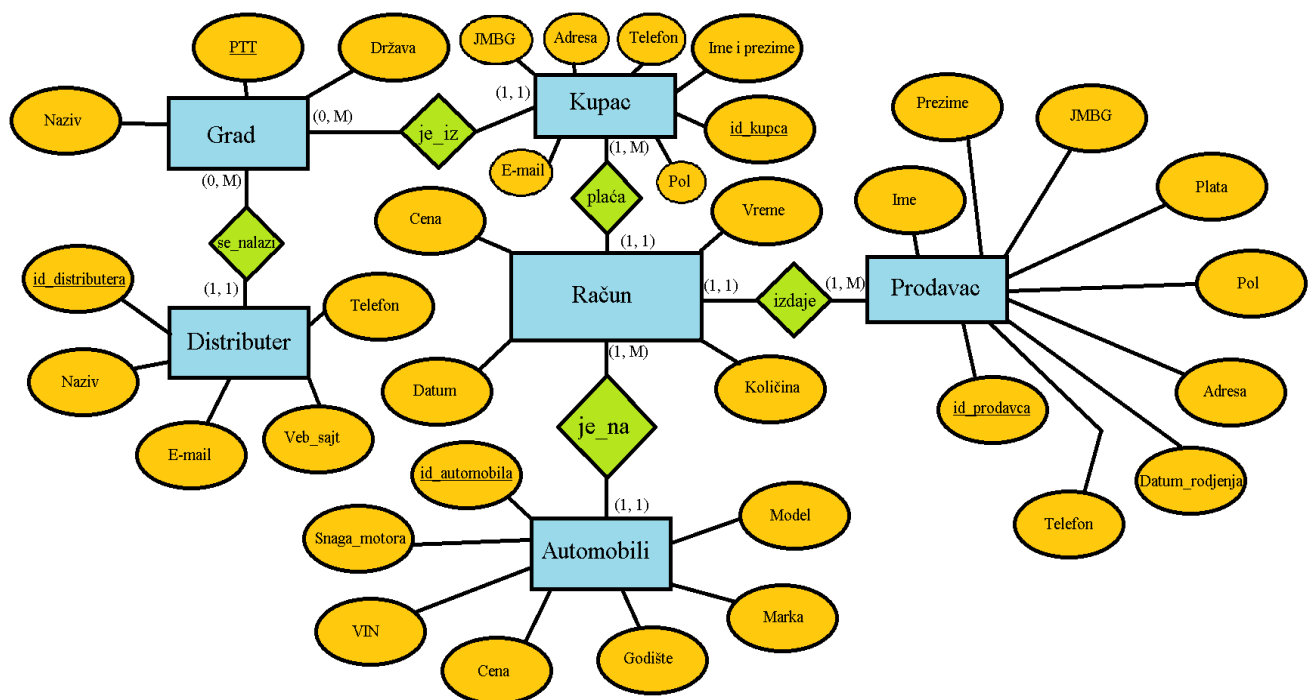


- Veza „IZDAJE“

Veza *IZDAJE* povezuje entitet *Prodavac* sa entitetom *Račun*, tako da jedan prodavac može izdati jedan ili više računa, dok jedan račun može biti izdat samo od strane jednog prodavca.



3.3 Kompletan ER dijagram



4. Logička šema relacione baze podataka

Da bi se dobila logička šema relacione baze podataka, moraju svi entiteti i veze da se prevedu iz modela objekti-veze na njihov ekvivalentni oblik u relacionom modelu – relacije i međurelaciona ograničenja. Elementi će biti prevedeni u relacioni oblik sledećim redosledom.

- Entiteti
- Veze nasleđivanja
- Gerundi
- Preostale veze

4.1 Entiteti

Svi entiteti postaju nezavisne šeme relacija. Ime tipa entiteta postaje ime šeme relacije, obeležja tipa objekta su obeležja šeme relacije (atributi). Identifikator entiteta postaje primarni ključ šeme relacije.

- Grad(ptt, naziv, država)
- Kupac(id_kupca, telefon, e-mail, jmbg, ime, prezime, datum_rođenja, adresa, pol)
- Račun(datum, vreme, kolicina, cena)
- Prodavac(id_prodavca, jmbg, ime, prezime, telefon, datum_rođenja, adresa, pol, plata)
- Distributer(id_distributera, naziv, telefon, e-mail, veb_sajt)
- Automobil(id_automobila, marka, model, boja, godiste, cena, zapremina_motora, snaga_motora, broj_vrata, vin)

4.2 Gerundi

Agregirani objekti (mešoviti tip objekat veza, tj. Gerund) se posmatra na isti način kao i odgovarajuća veza. To je tip entiteta koji se ponaša kao veza ili tip veze koja može da učestvuje u drugim vezama – da se ponaša kao entitet. Gerund uvažava samo tipove entiteta koje povezuje prilikom prevođenja u relacioni oblik, dok tipove veza ignoriše.

U ovom ER modelu ne postoji Gerund.

4.3 Preostale veze

Prilikom prevođenja u relacioni model takođe je potrebno obratiti pažnju na sve ostale veze u modelu.

Veza *SE_NALAZI* povezuje entitet *Grad* sa entitetom *Distributer*. Distributer se nalazi u jednom i samo jednom gradu, dok jedan grad može imati više distributera.

SE_NALAZI (ptt, id_distributera)

Kardinalnost je (0, M) : (1, 1).

Veza *JE_IZ* povezuje entitet *Grad* sa entitetom *Kupac*, tako da je svaki kupac pojedinačno iz jednog i samo jednog grada, a jedna grad ne mora da ima niti jednog kupca, a može ih imati i više.

JE_IZ (ptt, id_kupca)

Kardinalnost je (0, M) : (1, 1).

Veza *PLAĆA* povezuje entitet *Kupac* sa entitetom *Račun*, tako da svaki kupac pojedinačno plaća jedan ili više računa, dok jedan račun može platiti samo jedan kupac.

PLACA (id_kupca, id_kupca)

Kardinalnost je (1, M) : (1, 1).

Veza *JE_NA* povezuje entitet *Automobili* sa entitetom *Račun*, tako da se jedan automobil nalazi samo na jednom računu, dok se na jednom računu može nalaziti više automobila.

JE_NA (id_automobila, id_automobila)

Kardinalnost je (1, 1) : (1, M).

Veza *IZDAJE* povezuje entitet *Prodavac* sa entitetom *Račun*, tako da jedan prodavac može izdati jedan ili više računa, dok jedan račun može biti izdat samo od strane jednog prodavca.

IZDAJE (id_prodavca, id_prodavca)

Kardinalnost je (1, M) : (1, 1).

4.4 Korigovanje relacija radi poštovanja normalnih formi

Primećuje se da u trenutnom sistemu ne postoji nijedan složeni atribut, dakle sve vrednosti su atomične i normalizovane, tj. poštuju prvu normalnu formu (1NF). Relacija je u drugoj normalnoj formi ako i samo ako je već u prvoj normalnoj formi i ako svi njeni neključni atributi potpuno i funkcionalno zavise od primarnog ključa. U ovom slučaju je i druga normalna forma ispunjena, jer je u većini slučajeva primarni ključ samo jedan atribut (nije složeni ključ) ili ako je složeni ključ sastoji se od dva atributa i relacije nema neključne attribute. Da bi treća normalna forma bila ispunjena potrebno je da je relacija u 2NF i da svi neključni atributi netranzitivno funkcionalno zavise od primarnog ključa, što je i ispunjeno. Pošto su sve tri normalne forme ispunjene neće biti korigovanja relacija.

4.5 Međurelaciona ograničenja

Međurelaciona ograničenja se definišu za sve nasleđene attribute neke relacije – da moraju biti podskup skupa vrednosti nasleđenog atributa u originalnoj relaciji od koje je taj atribut nasleđen.

- $\text{grad}[\text{ptt}] \subseteq \text{distributer}[\text{postanski_br_mesta}]$
- $\text{grad}[\text{ptt}] \subseteq \text{kupac}[\text{postanski_br_mesta}]$
- $\text{distributer}[\text{id_distributera}] \subseteq \text{automobili}[\text{id_distributera}]$
- $\text{automobili}[\text{id_automobila}] \subseteq \text{racun}[\text{id_automobila}]$
- $\text{kupac}[\text{id_kupca}] \subseteq \text{racun}[\text{id_kupca}]$
- $\text{prodavac}[\text{id_prodavca}] \subseteq \text{racun}[\text{id_prodavca}]$

4.6 Kompletna logička šema

$S = \{$

- $\text{Grad}(\text{ptt}, \text{naziv}, \text{država})$
- $\text{Kupac}(\text{id_kupca}, \text{telefon}, \text{e-mail}, \text{jmbg}, \text{ime}, \text{prezime}, \text{datum_rodjenja}, \text{adresa}, \text{pol})$
- $\text{Račun}(\text{datum}, \text{vreme}, \text{kolicina}, \text{cena})$
- $\text{Prodavac}(\text{id_prodavca}, \text{jmbg}, \text{ime}, \text{prezime}, \text{telefon}, \text{datum_rodjenja}, \text{adresa}, \text{pol}, \text{plata})$
- $\text{Distributer}(\text{id_distributera}, \text{naziv}, \text{telefon}, \text{e-mail}, \text{veb_sajt})$
- $\text{Automobil}(\text{id_automobila}, \text{marka}, \text{model}, \text{boja}, \text{godiste}, \text{cena}, \text{zapremina_motora}, \text{snaga_motora}, \text{broj_vrata}, \text{vin})$

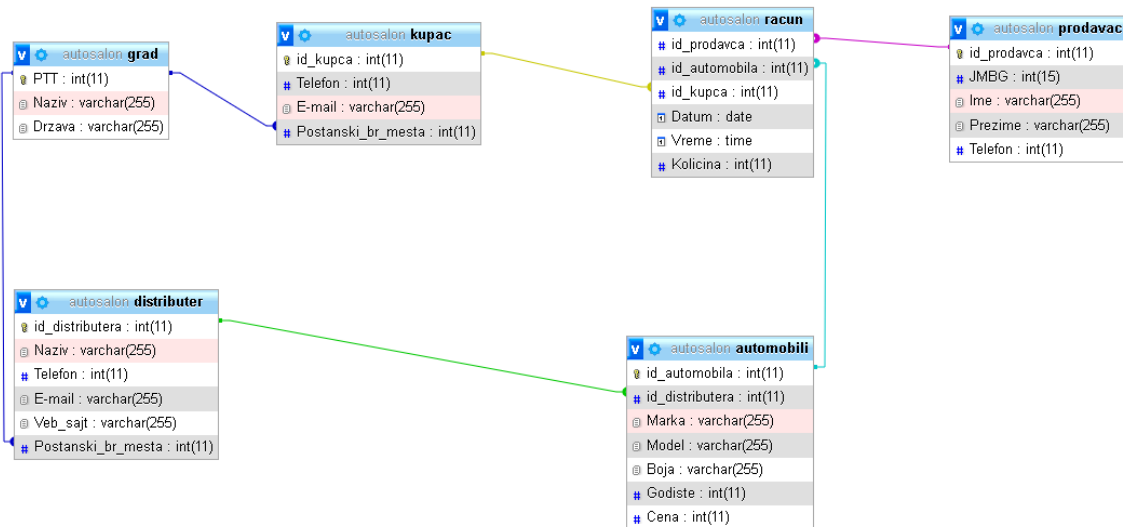
$\}$

I = {

- grad[ptt] \subseteq distributer[postanski_br_mesta]
- grad[ptt] \subseteq kupac[postanski_br_mesta]
- distributer[id_distributera] \subseteq automobili[id_distributera]
- automobili [id_automobila] \subseteq racun[id_automobila]
- kupac[id_kupca] \subseteq racun[id_kupca]
- prodavac [id_prodavca] \subseteq racun[id_prodavca]

}

5. Fizička šema relacije baze podataka



Fizička šema je najniži nivo apstrakcije podataka u bazi. Pomoću fizičke šeme se opisuje realan sistem. To je poslednji korak pre implementacije samog sistema baze podataka. Fizička šema je vrlo slična logičkoj, pa se vrši direktno mapiranje između logičke i fizičke šeme. Izuzeci su izmene koje olakšavaju održavanje podataka i povećane efikasnosti operacija sistema za upravljanje bazom podataka. Na datoj slici je prikazana kompletna fizička šema, izrađena u “phpMyAdmin” programu.

6. Implementacija projektovane baze podataka sa testnim podacima u MySQL

```

IS Auto Kuće.sql ×
D: > Faks > treca godina > baze > auto salon > IS Auto Kuće.sql
1  -- phpMyAdmin SQL Dump
2  -- version 5.1.3
3  -- https://www.phpmyadmin.net/
4  --
5  -- Host: 127.0.0.1
6  -- Generation Time: Jun 19, 2022 at 09:20 PM
7  -- Server version: 10.4.24-MariaDB
8  -- PHP Version: 7.4.29
9
10 SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
11 START TRANSACTION;
12 SET time_zone = "+00:00";
13
14
15 /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
16 /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
17 /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
18 /*!40101 SET NAMES utf8mb4 */;
19
20 --
21 -- Database: `autosalon`
22 --
23 CREATE DATABASE IF NOT EXISTS `autosalon` DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_bin;
24 USE `autosalon`;
25
26 -- -----
27
28 --
29 -- Table structure for table `automobili`
30 --
31
32 CREATE TABLE IF NOT EXISTS `automobili` (
33   `id_automobila` int(11) NOT NULL AUTO_INCREMENT,
34   `id_distributera` int(11) NOT NULL,
35   `Marka` varchar(255) COLLATE utf8mb4_bin NOT NULL,
36   `Model` varchar(255) COLLATE utf8mb4_bin NOT NULL,
37   `Boja` varchar(255) COLLATE utf8mb4_bin NOT NULL,
38   `Godiste` int(11) NOT NULL,
39   `Cena` float NOT NULL,
40   `Zapremina_motora` int(11) NOT NULL,
41   `Snaga_motora` int(11) NOT NULL,
42   `Broj_vrata` int(11) NOT NULL,
43   `VIN` varchar(255) COLLATE utf8mb4_bin NOT NULL,
44   PRIMARY KEY (`id_automobila`),
45   KEY `id_distributera` (`id_distributera`)
46 ) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_bin;
47
48 --
49 -- Dumping data for table `automobili`

```



```

IS Auto Kuće.sql ×
D:\> Faks > treca godina > baze > auto salon > IS Auto Kuće.sql
49 -- Dumping data for table 'automobili'
50 --
51
52 INSERT INTO 'automobili' ('id_automobila', 'id_distributera', 'Marka', 'Model', 'Boja', 'Godiste', 'Cena', 'Zapremina_motora', 'Snaga_motora', 'Broj_vrata', 'VIN') VALUES
53 (1, 1, 'Zastava', '10', 'Crvena', 2007, 1999, 1242, 44, 5, 'ZFA18000000977097'),
54 (2, 2, 'Lancia', 'Delta', 'Siva', 2010, 4999, 1590, 80, 5, 'ZLA831AB003131000'),
55 (3, 3, 'Fiat', 'Bravo', 'Siva', 2010, 4299, 1360, 80, 5, 'ZFA19000004242900');
56
57 -----
58 --
59 --
60 -- Table structure for table 'distributer'
61 --
62
63 CREATE TABLE IF NOT EXISTS 'distributer' (
64   'id_distributera' int(11) NOT NULL,
65   'Naziv' varchar(255) COLLATE utf8mb4_bin NOT NULL,
66   'Telefon' int(11) NOT NULL,
67   'E-mail' varchar(255) COLLATE utf8mb4_bin NOT NULL,
68   'Veb_sajt' varchar(255) COLLATE utf8mb4_bin NOT NULL,
69   'Postanski_br_mesta' int(11) NOT NULL,
70   PRIMARY KEY ('id_distributera'),
71   KEY 'Postanski_br_mesta' ('Postanski_br_mesta')
72 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_bin;
73
74 --
75 -- Dumping data for table 'distributer'
76 --
77
78 INSERT INTO 'distributer' ('id_distributera', 'Naziv', 'Telefon', 'E-mail', 'Veb_sajt', 'Postanski_br_mesta') VALUES
79 (1, 'Nikom d.o.o', 645096351, 'nikomnikom@nikom.com', 'nikom.com', 34000),
80 (2, 'Nesa Fiat', 634781256, 'nesafiat@gmail.com', 'nesafiat.org', 11000),
81 (3, 'FlatCars', 658794503, 'flatcarsni@hotmail.com', 'flatcars.net', 10000);
82
83 -----
84 --
85 --
86 -- Table structure for table 'grad'
87 --
88
89 CREATE TABLE IF NOT EXISTS 'grad' (
90   'PTT' int(11) NOT NULL,
91   'Naziv' varchar(255) COLLATE utf8mb4_bin NOT NULL,
92   'Drzava' varchar(255) COLLATE utf8mb4_bin NOT NULL,
93   PRIMARY KEY ('PTT')
94 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_bin;
95
96 --
97 -- Dumping data for table 'grad'

```

```

IS Auto Kuće.sql ×
D:\> Faks > treca godina > baze > auto salon > IS Auto Kuće.sql
96 --
97 -- Dumping data for table `grad`
98 --
99
100 INSERT INTO `grad` (`PTT`, `Naziv`, `Drzava`) VALUES
101 (11000, 'Beograd', 'Srbija'),
102 (18000, 'Nis', 'Srbija'),
103 (34000, 'Kragujevac', 'Srbija');
104
105 -- -----
106
107 --
108 -- Table structure for table `kupac`
109 --
110
111 CREATE TABLE IF NOT EXISTS `kupac` (
112   `id_kupca` int(11) NOT NULL AUTO_INCREMENT,
113   `Telefon` int(11) NOT NULL,
114   `E-mail` varchar(255) COLLATE utf8mb4_bin NOT NULL,
115   `JMBG` varchar(255) COLLATE utf8mb4_bin NOT NULL,
116   `Ime` varchar(255) COLLATE utf8mb4_bin NOT NULL,
117   `Prezime` varchar(255) COLLATE utf8mb4_bin NOT NULL,
118   `Datum_rodjenja` date NOT NULL,
119   `Adresa` varchar(255) COLLATE utf8mb4_bin NOT NULL,
120   `Pol` char(1) COLLATE utf8mb4_bin NOT NULL,
121   `Postanski_br_mesta` int(11) NOT NULL,
122   PRIMARY KEY (`id_kupca`),
123   KEY `Postanski_br_mesta` (`Postanski_br_mesta`)
124 ) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_bin;
125
126 --
127 -- Dumping data for table `kupac`
128 --
129
130 INSERT INTO `kupac` (`id_kupca`, `Telefon`, `E-mail`, `JMBG`, `Ime`, `Prezime`, `Datum_rodjenja`, `Adresa`, `Pol`, `Postanski_br_mesta`) VALUES
131 (1, 694561237, 'draganmilosevic@gmail.com', '0102987720055', 'Dragan', 'Milosevic', '1987-02-01', 'Djordja Andrejevica Kune 2', 'M', 34000),
132 (2, 69376541, 'uros Petrovic@gmail.com', '0302000726517', 'Uros', 'Petrovic', '2000-02-03', 'Vladimira Rolovica 19', 'M', 11000),
133 (3, 634586257, 'djurovic27@gmail.com', '0408995720014', 'Ljubisa', 'Djurovic', '1995-08-04', 'Knjinska bb', 'M', 18000);
134
135 -- -----
136
137 --
138 -- Table structure for table `prodavac`
139 --
140

```

```

IS Auto Kuće.sql x
D: > Faks > treća godina > baze > auto salon > IS Auto Kuće.sql
137 --
138 -- Table structure for table `prodavac`
139 --
140
141 CREATE TABLE IF NOT EXISTS `prodavac` (
142   `id_prodavca` int(11) NOT NULL,
143   `JMBG` varchar(255) COLLATE utf8mb4_bin NOT NULL,
144   `Ime` varchar(255) COLLATE utf8mb4_bin NOT NULL,
145   `Prezime` varchar(255) COLLATE utf8mb4_bin NOT NULL,
146   `Telefon` int(11) NOT NULL,
147   `Datum_rodjenja` date NOT NULL,
148   `Adresa` varchar(255) COLLATE utf8mb4_bin NOT NULL,
149   `Pol` char(1) COLLATE utf8mb4_bin NOT NULL,
150   `Plata` float NOT NULL,
151   PRIMARY KEY (`id_prodavca`)
152 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_bin;
153
154 --
155 -- Dumping data for table `prodavac`
156 --
157
158 INSERT INTO `prodavac` (`id_prodavca`, `JMBG`, `Ime`, `Prezime`, `Telefon`, `Datum_rodjenja`, `Adresa`, `Pol`, `Plata`) VALUES
159 (1, '2508000720055', 'Djordje', 'Molnar', 693132860, '2000-08-25', 'Mlade Bosne 7', 'M', 89450),
160 (2, '0104999720017', 'Nikola', 'Kadic', 642205580, '2000-04-01', 'Steve Knicanina 2', 'M', 87500),
161 (3, '2604968720017', 'Petar', 'Petrovic', 641236548, '1968-04-26', 'Mitra Bakica 3', 'M', 79500);
162
163 -- -----
164
165 --
166 -- Table structure for table `racun`
167 --
168
169 CREATE TABLE IF NOT EXISTS `racun` (
170   `id_prodavca` int(11) NOT NULL,
171   `id_automobila` int(11) NOT NULL,
172   `id_kupca` int(11) NOT NULL,
173   `Datum` date NOT NULL,
174   `Vreme` time NOT NULL,
175   `Kolicina` int(11) NOT NULL,
176   `Cena` float NOT NULL,
177   PRIMARY KEY (`id_prodavca`),
178   KEY `id_kupca` (`id_kupca`),
179   KEY `id_prodavca` (`id_prodavca`),
180   KEY `id_automobila` (`id_automobila`)
181 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_bin;
182
183 --
184 -- Dumping data for table `racun`

```

```

IS Auto Kuće.sql ×
D: > Faks > treća godina > baze > auto salon > IS Auto Kuće.sql

182
183 --
184 -- Dumping data for table `racun`
185 --
186
187 INSERT INTO `racun` (`id_prodavca`, `id_automobila`, `id_kupca`, `Datum`, `Vreme`, `Kolicina`, `Cena`) VALUES
188 (1, 1, 1, '2022-06-17', '15:21:28', 1, 1999),
189 (2, 2, 2, '2022-06-17', '15:38:25', 1, 4999),
190 (3, 3, 3, '2022-06-19', '21:14:13', 1, 4299);
191
192 --
193 -- Constraints for dumped tables
194 --
195
196 --
197 -- Constraints for table `automobili`
198 --
199 ALTER TABLE `automobili`
200 | ADD CONSTRAINT `automobili_ibfk_1` FOREIGN KEY (`id_distributera`) REFERENCES `distributer` (`id_distributera`);
201 |
202 --
203 -- Constraints for table `distributer`
204 --
205 ALTER TABLE `distributer`
206 | ADD CONSTRAINT `distributer_ibfk_1` FOREIGN KEY (`Postanski_br_mesta`) REFERENCES `grad` (`PTT`);
207 |
208 --
209 -- Constraints for table `kupac`
210 --
211 ALTER TABLE `kupac`
212 | ADD CONSTRAINT `kupac_ibfk_1` FOREIGN KEY (`Postanski_br_mesta`) REFERENCES `grad` (`PTT`);
213 |
214 --
215 -- Constraints for table `racun`
216 --
217 ALTER TABLE `racun`
218 | ADD CONSTRAINT `racun_ibfk_1` FOREIGN KEY (`id_kupca`) REFERENCES `kupac` (`id_kupca`),
219 | ADD CONSTRAINT `racun_ibfk_2` FOREIGN KEY (`id_prodavca`) REFERENCES `prodavac` (`id_prodavca`),
220 | ADD CONSTRAINT `racun_ibfk_3` FOREIGN KEY (`id_automobila`) REFERENCES `automobili` (`id_automobila`);
221 COMMIT;
222
223 /*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
224 /*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
225 /*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
226

```

7. Literatura

- Materijal sa Moodle portala – moodle.fink.rs
- MySQL - [MySQL :: MySQL Documentation](#)
- Youtube – youtube.com
- Lucidchart - <https://www.lucidchart.com/pages/er-diagrams>