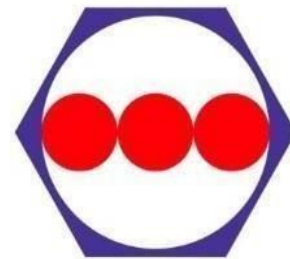




Univerzitet u Novom Sadu
FAKULTET TEHNIČKIH NAUKA



Departman za industrijsko inženjerstvo i inženjerski menadžment
Inženjerstvo informacionih sistema

INFORMACIONI SISTEM
ZA PODRŠKU POSLOVANJA PARKING SERVISA
-SISTEMI BAZA PODATAKA-

Tijana Kovačević IT54/2017

Novi Sad, 2020.

SADRŽAJ

<i>UVOD</i>	3
<i>ANALIZA PROGRAMSKOG DOMENA</i>	3
<i>ER MODEL</i>	4
<i>ER MODEL PODŠEME</i>	5
<i>TABELARNI PRIKAZ OBELEŽJA I OGRANIČENJA</i>	7
<i>RELACIONI MODEL</i>	13
<i>DDL</i>	16
<i>DML</i>	16
<i>SQL</i>	16
<i>OBJEKTI</i>	18
<i>UPITI</i>	24
<i>FUNKCIJE</i>	26
<i>PROCEDURE</i>	27
<i>TRIGERI</i>	28
<i>ZAKLJUČAK</i>	29

UVOD

Projekat koji je predstavljen u dokumentaciji podrazumeva razvoj informacionog sistema za podršku poslovanja “Parking servisa”, čiji je cilj unapređenje i podrška poslovanja službe “Parking servis” koji će doprineti efikasnijem radu službe. Takođe, zamisao je i postizanje olakšanog i ubrzanog pristupa podacima u digitalnom obliku, kao i zadovoljenje zahteva korisnika.

U daljem tekstu, priložen je ER model podataka, čijim prevođenjem je kreiran relacioni model podataka. Priložen je i tabelarni prikaz ograničenja pojave tipa, integriteta tipa i jedinstvenih vrednosti.

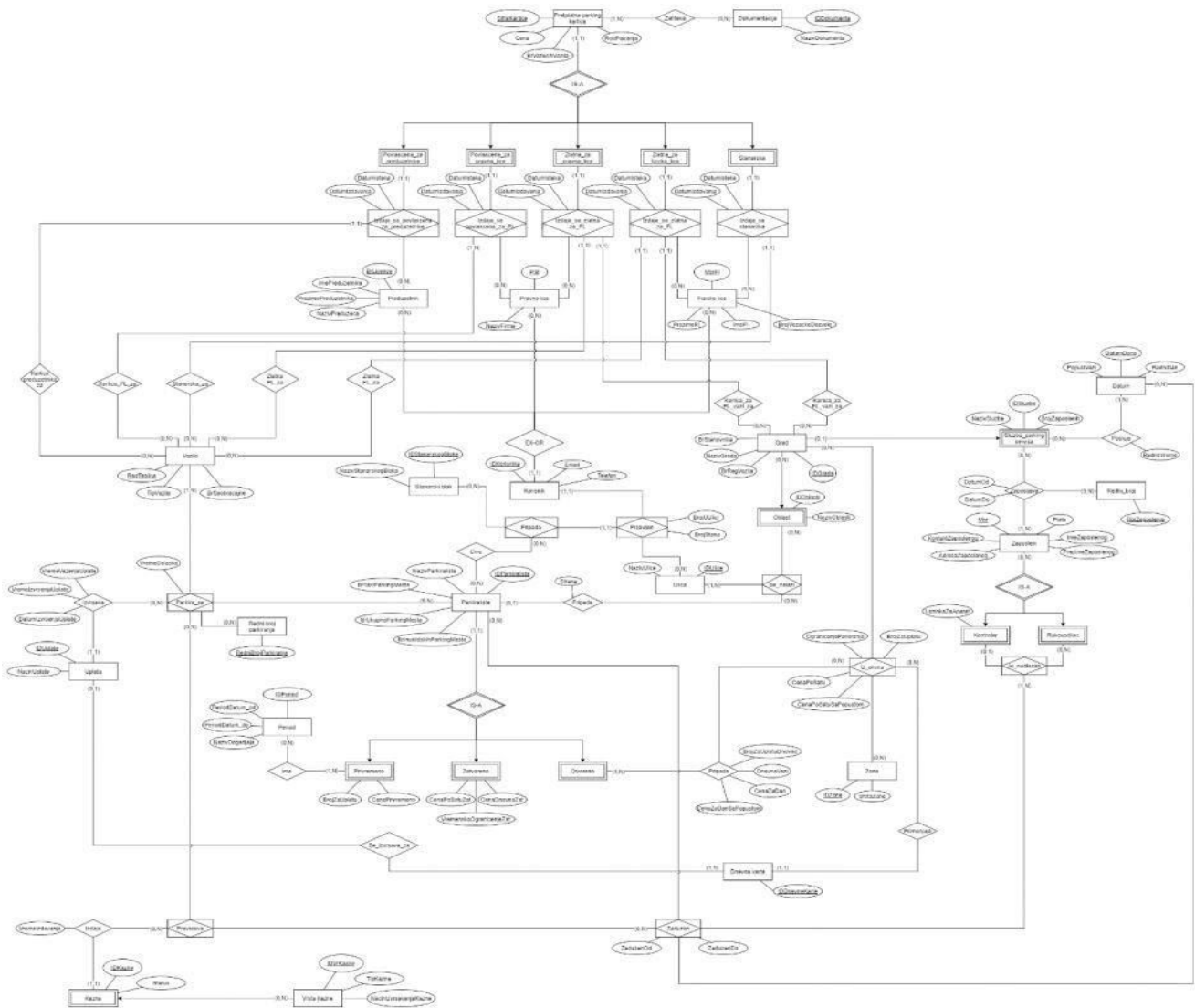
Na samom kraju, na osnovu relacionog modela podataka, izvršena je implementacija šeme baze podataka.

ANALIZA PROGRAMSKOG DOMENA

Modernizacija i unapređenje kulture parkiranja, primena novog sistema zasnovanog na organizaciji parkirališta u više zona i sasvim drugačiji sistem naplate od do sada primenjivanog svakako predstavljaju smernicu kako ovaj projekat može izgledati. Svako poslovanje, kao i pomenuto, treba da bude zasnovano na poštovanju zakona i pravila, a u slučaju njihovog kršenja od strane korisnika sistema neophodno je izdati odgovarajuće kazne za načinjeni prekršaj.

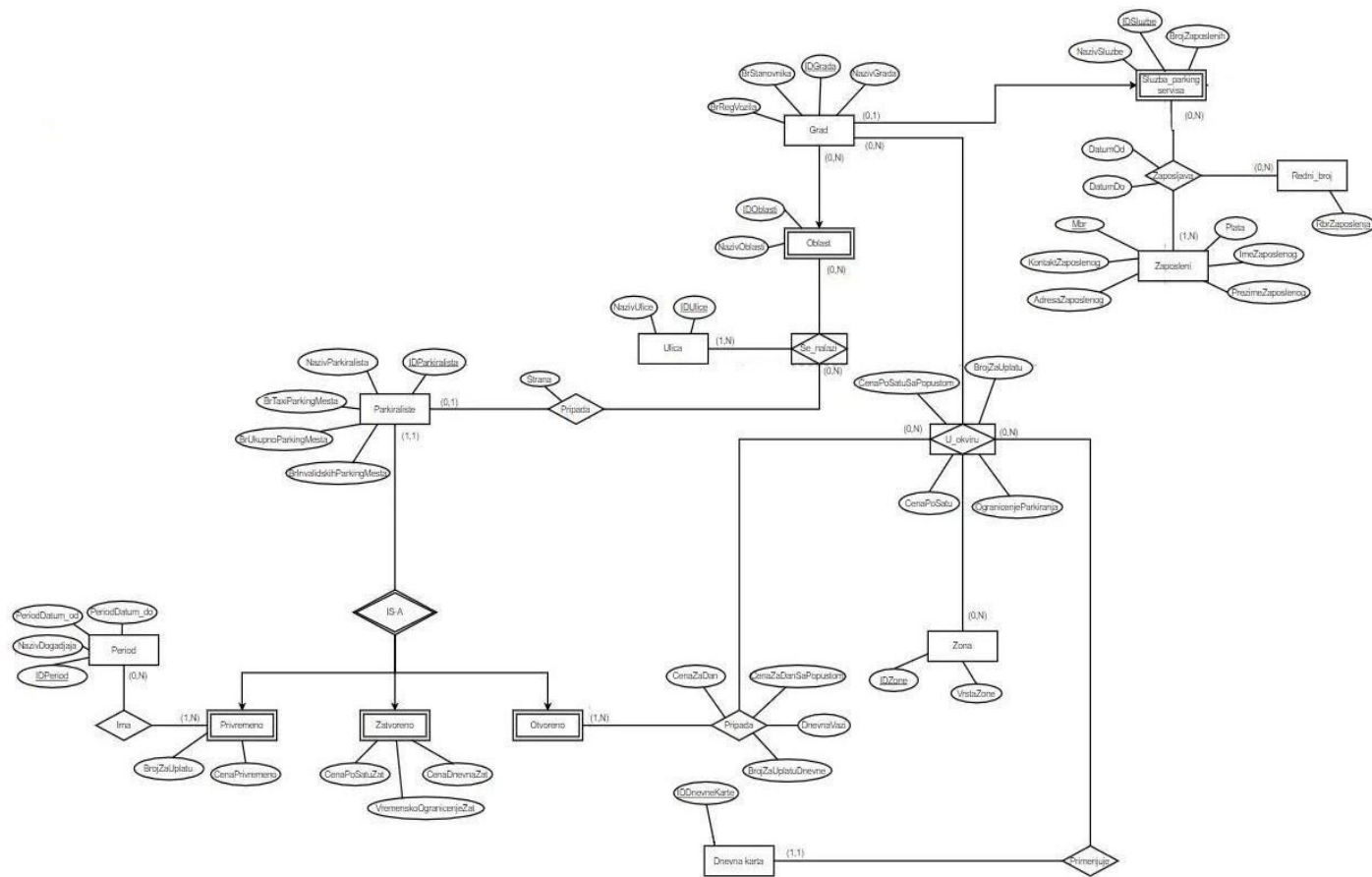
Značaj budućeg informacionog sistema za podršku poslovanja preduzeća “Parking servis” ogleda se u organizaciji rada preduzeća. Od velike važnosti su informacije prema kojima se postupa odgovorno i profesionalno.

ER MODEL



ER MODEL PODŠEME

Podšema *Parkiralište*



Grad modelujemo tipom entiteta Grad koji čuva podatke o identifikacionom broju grada, nazivu grada, kao i broju prijavljenih stanovnika u okviru grada i broju registrovanih vozila na teritoriji grada. Grad je podeljen na različite oblasti, koje se modeluju tipom entiteta Oblast, a identifikuju na osnovu identifikacionog broja grada i sopstvenog identifikacionog broja. Takođe se pamti i naziv oblasti. Oblastima pripadaju različite ulice, pri čemu se tip entiteta Ulica identifikuje na osnovu ID-a ulice, i beleži se podatak o nazivu ulice. U nekoj ulici u okviru određene oblasti, odnosno na određenoj strani ili obe strane ulice, mogu se nalaziti parkirališta, i ona se modeluju tipom entiteta Parkiralište. Neka od značajnih obeležja parkirališta su ID parkirališta, naziv parkirališta, ukupan broj parking mesta, kao i broj parking mesta za taksi službe, i broj invalidskih mesta. Parkiralište može da bude ili otvoreno ili zatvoreno ili privremeno. Privremena parkirališta jesu privremeno uređeni prostori za parkiranje vozila, najčešće za vreme državnih praznika i različitih manifestacija u gradu. Zbog toga, period tokom kojeg privremeno parkiralište postoji modelujemo tipom entiteta

Period čime se obezbeđuje da jedno isto privremeno parkiralište bude može da bude aktivno više puta u toku godine. Ono se jedinstveno identifikuje sa obeležjima ID perioda, naziv događaja koji se odigrava tokom tog perioda, kao i datum početka i datum završetka tog događaja. Kod samog tipa entiteta Privremeno pamte se podaci o ceni parkiranja vozila na privremenom parkiralištu bez obzira na vreme zadržavanja i podaci o broju na koji je moguće izvršiti uplatu za parkirano vozilo. Kod tipa entiteta Zatvoreno čuvaju se podaci o ceni po započetom satu zatvorenog parkirališta i ceni dnevne karte za zatvoreno parkiralište kao i podatak o vremenskom ograničenju parkiranja nakon kog prestaje da se obračunava cena po započetom satu, već se naplaćuje dnevna karta bez obzira na vreme zadržavanja vozila. Na nivou grada se definišu zone, i one su iskazane tipom entiteta Zona. Neka od značajnih obeležja svake zone jesu njen identifikacioni broj i vrsta zone, a u zavisnosti od grada razlikuju se cena po satu parkiranja u zoni, cena po satu sa popustom koja važi tokom različitih državnih praznika i manifestacija, broj na koji je moguće uplatiti parking u toj zoni putem SMS poruke, kao i maksimalno vremensko ograničenje parkiranja u zoni. Otvoreno parkiralište pripada barem jednoj zoni. Moguće je da na nekim parkiralištima važi dnevna karta i takva parkirališta poseduju dopunsku tablu na kojoj je naznačeno da se parkiranje može platiti i dnevnom kartom, a evidentira se i broj za uplatu dnevne karte putem SMS poruke, cena karte na dnevnom nivou kao i cena sa popustom na dnevnom nivou koja važi tokom različitih državnih praznika i manifestacija. Dnevna karta se modeluje tipom entiteta Dnevna_karta i identifikuje se na osnovu sopstvenog identifikacionog broja. Služba parking servisa posluje na nivou grada. Tip entiteta Grad se jedinstveno identifikuje pomoću id_grada, a takođe su poznati naziv, broj stanovnika i broj registrovanih vozila u tom gradu, dok se tip entiteta Služba identifikuje pomoću id_grada i sopstvenog ID-ja, a još se pamte njen naziv i broj zaposlenih. Služba parking servisa zapošljava određeni broj zaposlenih, koji se modeluju pomoću tipa entiteta Zaposleni i identifikuju pomoću matičnog broja (Mbr), a takođe se evidentiraju podaci o imenu i prezimenu zaposlenog, njegovoj adresi, kontakt telefonu i plati. Ono što može da se dogodi jeste da zaposleni koji je jednom napustio posao u službi, ponovo odluči da se zaposli tu, pa je bilo potrebno modelovati tip entiteta Redni_broj, da bi se vodila evidencija o tome koje mu je to zaposlenje po redu i od kad je zaposlen. U slučaju trenutnog zaposlenja obeležje datum_do bi imalo vrednost null ali su u evidenciji i sva prethodna zaposlenja sa rednim brojem, datumom od kada je bio zaposlen i obaveznim datumom do kada.

TABELARNI PRIKAZ OBELEŽJA I OGRANIČENJA

Tabela Zona

Rbr.	Naziv Obeležja	Opis Obeležja	Tip podatka	Dužina podatka	Null	Uslov
1.	ID_zone	Identifikacioni broj zone	NUMERIC	10	⊥	d>0
2.	Vrsta_zone	Vrsta zone	NVARCHAR	20	⊥	Δ
Ključ	K={ID_zone}					

Tabela Grad

Rbr.	Naziv Obeležja	Opis Obeležja	Tip podatka	Dužina podatka	Null	Uslov
1.	ID_grada	Identifikacioni broj grada	NUMERIC	10	⊥	d>0
2.	naziv_grada	Naziv grada	NVARCHAR	50	⊥	Δ
3.	broj_stan	Broj stanovnika u gradu	NUMERIC	10	⊥	d>=0
4.	broj_reg_vozila	Broj registrovanih vozila u gradu	NUMERIC	10	T	d>=0
Ključ	K={ID_grada}					
Ne postoje 2 grada sa istom vrednošću naziv_ grada						

Tabela Dnevna karta

Rbr.	Naziv Obeležja	Opis Obeležja	Tip podatka	Dužina podatka	Null	Uslov
1.	id_dnevne_karte	Identifikacioni broj dnevne karte	NUMERIC	10	⊥	d>0
Ključ	K={id_dnevne_karte}					

Tabela Oblast

Rbr.	Naziv Obeležja	Opis Obeležja	Tip podatka	Dužina podatka	Null	Uslov
1.	id_oblasti	Identifikacioni broj oblasti	NUMERIC	10	⊥	d>0
2.	id_grada	Naziv oblasti	NVARCHAR	50	⊥	Δ
Ključ	K={id_oblasti +id_grada}					

Tabela Ulica

Rbr.	Naziv Obeležja	Opis Obeležja	Tip podatka	Dužina podatka	Null	Uslov
1.	id_ulice	Identifikacioni broj ulice	NUMERIC	10	⊥	d>0
2.	naziv_ulice	Naziv ulice	NVARCHAR	50	⊥	Δ
Ključ	K={id_ulice}					

Tabela Se_nalazi

Rbr.	Naziv Obeležja	Opis Obeležja	Tip podatka	Dužina podatka	Null	Uslov
Ključ	K={id_lice + id_oblasti + id_grada}					

Tabela Parkiraliste

Rbr.	Naziv Obeležja	Opis Obeležja	Tip podatka	Dužina podatka	Null	Uslov
1.	id_parkiralista	Identifikacioni broj parkirališta	NUMERIC	10	⊥	d>0
2.	naziv_parkiralista	Naziv parkirališta	NVARCHAR	50	⊥	Δ
3.	br_ukupno_parking_mesta	Ukupan broj parking mesta na parkiralištu	NUMERIC	10	⊥	d>0
4.	br_invalidskih_parking_mesta	Ukupan broj invalidskih parking mesta na parkiralištu	NUMERIC	10	⊥	d>=0
5.	br_taxi_parking_mesta	Ukupan broj taxi mesta na parkiralištu	NUMERIC	10	⊥	d>=0
Ključ	K={id_parkiralista}					

Tabela Otvoreno

Rbr.	Naziv Obeležja	Opis Obeležja	Tip podatka	Dužina podatka	Null	Uslov
1.	id_parkiralista	Identifikacioni broj parkirališta	NUMERIC	10	⊥	d>0
Ključ	K={id_parkiralista}					

Tabela Redni broj zaposlenja

Rbr.	Naziv Obeležja	Opis Obeležja	Tip podatka	Dužina podatka	Null	Uslov
1.	rbr_zaposlenja	Redni broj zaposlenja zaposlenog	NUMERIC	3	⊥	d>0
Ključ	K={rbr_zaposlenja}					

Tabela Sluzba parking servisa

1.	id_sluzbe	Jedinstveni identifikacioni broj sluzbe	NUMERIC	10	⊥	d>0
2.	naziv_sluzbe	Naziv sluzbe	NVARCHAR	50	⊥	Δ
3.	br_zaposlenih	Broj zaposlenih koji rade u sluzbi	NUMERIC	10	⊥	d>=0
Ključ	K={id_grada+id_sluzbe}					

Tabela Zaposleni

Rbr.	Naziv Obeležja	Opis Obeležja	Tip podatka	Dužina podatka	Null	Uslov
1.	Mbr	Matični broj kao jedinstveno identifikaciono obeležje zaposlenog	NUMERIC	8	⊥	d>0
2.	ime_zap	Ime zaposlenog	NVARCHAR	15	⊥	Δ
3.	preizme_zap	Prezime zaposlenog	NVARCHAR	20	⊥	Δ
4.	kontakt_zap	Kontakt zaposlenog	NVARCHAR	10	⊥	Δ
5.	adresa_zap	Adresa zaposlenog	NVARCHAR	20	⊥	Δ
Ključ	K={Mbr}					
Ne postoje 2 zaposlena sa istom vrednošćukontakt_zap,						
Vrednost Mbr mora da ima tračno 8 numeričkih znakova						

Tabela Ima

Rbr.	Naziv Obeležja	Opis Obeležja	Tip podatka	Dužina podatka	Null	Uslov
Ključ	K={id_parkalista + id_period}					

Tabela Pripada

Rbr.	Naziv Obeležja	Opis Obeležja	Tip podatka	Dužina podatka	Null	Uslov
1.	Dnevna_vazi	Provera da li se dnevna karta primenjuje	BIT	1	⊥	Δ
2.	cena_za_dan	Cena dnevne karte	NUMERIC	5	T	d>0
3.	br_za_uplatu_dnevne	Broj za uplatu dnevne karte	NVARCHAR	10	T	Δ
	cena_za_dan_sa_popustom	Cena dnevne karte sa popustom	NUMERIC	5	T	d>0
Ključ	K={id_parkiralista + id_grada +id_zone}					
cena_za_dan i br_za_uplatu_dnevne postoje samo ako je obeležje Dnevna_vazi true						
cena_za_dan_sa_popustom može da postoji samo kada je obeležje Dnevna_vazi true						

Tabela Zaposljava

Rbr.	Naziv Obeležja	Opis Obeležja	Tip podatka	Dužina podatka	Null	Uslov
1.	Datum_od	Datum početka zaposlenja	DATE		⊥	Δ
2.	Datum_Do	Kranji datum zaposlenja	DATE		T	Δ
Ključ	K={IDSluzbe + IDGrada + Mbr + RbrZaposlenja}					
Vrednost Datum od mora da bude pre Datum do						

Tabela U_okviru

Rbr.	Naziv Obeležja	Opis Obeležja	Tip podatka	Dužina podatka	Null	Uslov
1.	Broj_za_uplatu	Broj za uplatu	NVARCHAR	10	\perp	Δ
2.	Ogranicenje_parkiranja	Maksimalno zadržavanje u zoni	NVARCHAR	30	T	Δ
3.	cenaaposatu	Cena jednočasovnog zadržavanja na parkiralištu	NUMERIC	5	\perp	d>0
4.	cenaaposatusa_popustom	Cena jednočasovnog zadržavanja na parkiralištu sa popustom	NUMERIC	5	T	Δ
Ključ	K={id_grada + id_zone}					

Tabela Pripadaa

Rbr.	Naziv Obeležja	Opis Obeležja	Tip podatka	Dužina podatka	Null	Uslov
1.	Strana	Strana ulice na kojoj se parkiralište nalazi 0 - parna strana 1 - neparna strana 2 - na obe strane	NUMERIC	1	\perp	$d \in \{0, 1, 2\}$
Ključ						

RELACIONI MODEL

Relacioni model je jedna od vrsta modela podataka. Zasniva se na predikatskom računu i teoriji skupova. Predstavlja najrasprostranjeniji model za modeliranje realnih problema i dinamično upravljanje podacima.

Relaciona baza podataka je poseban tip baze podataka kod kojeg se organizacija podataka zasniva na relacionom modelu. Podaci se u ovakvim bazama organizuju u skup relacija između kojih se definišu određene veze. Relacija se definiše kao skup n-torki sa istim atributima, definisanih nad istim domenima iz kojih mogu da uzimaju vrednosti. U relacionim bazama podataka, svaka relacija mora da ima definisan primarni ključ, koji predstavlja atribut pomoću kojeg se jedinstveno identifikuje svaka n-torka. Relacija može imati i strani ključ, preko kojeg se ostvaruje veza sa drugim relacijama.

Grad ($\{IDGrada, NazivGrada, BrStanovnika, BrRegVozila\}, \{IDGrada\}$)

Oblast ($\{IDOblasti, IDGrada, NazivOblasti\}, \{IDOblasti + IDGrada\}$)

$Oblast[IDGrada] \subseteq Grad[IDGrada]$

Ulica ($\{IDUlice, NazivUlice\}, \{IDUlice\}$)

Se_nalazi ($\{IDUlice, IDGrada, IDOblasti\}, \{IDUlice + IDGrada + IDOblasti\}$)

$Se_nalazi[IDUlice] \subseteq Ulica[IDUlice]$

$Ulica[IDUlice] \subseteq Se_nalazi[IDUlice]$

$Se_nalazi[IDOblasti + IDGrada] \subseteq Oblast[IDOblasti + IDGrada]$

$Korisnik[IDUlice] \subseteq Ulica[IDUlice]$

$Null(Korisnik, IDUlice) = \perp$

Parkiraliste ($\{IDParkiralista, NazivParkiralista, BrTaxiParkingMesta, BrUkupnogParkingMesta, BrInvalidskihParkingMesta, IDUlice, IDOblasti, IDGrada, Strana\}, \{IDParkiralista\}$)

$Parkiraliste[IDUlice + IDGrada + IDOblasti] \subseteq Se_nalazi[IDUlice + IDGrada + IDOblasti]$

Privremeno ($\{IDParkiralista, BrojZaUplatu, CenaPrivremeno\}, \{IDParkiralista\}$)

Zatvoreno ($\{IDParkiralista, CenaPoSatuZat, CenaDnevnaZat, VremenskoOgranicenjeZat\}, \{IDParkiralista\}$)

Otvoreno ($\{IDParkiralista\}, \{IDParkiralista\}$)

$Privremeno[IDParkiralista] \subseteq Parkiraliste[IDParkiralista]$

$Zatvoreno[IDParkiralista] \subseteq Parkiraliste[IDParkiralista]$

$Privremeno[IDParkiralista] \subseteq Parkiraliste[IDParkiralista]$

$\text{Parkiraliste}[\text{IDParkiralista}] \subseteq \text{Privremeno}[\text{IDParkiralista}] \cup \text{Zatvoreno}[\text{IDParkiralista}]$
 $\cup \text{Otvoreno}[\text{IDParkiralista}]$

$\text{Privremeno}[\text{IDParkiralista}] \cap \text{Zatvoreno}[\text{IDParkiralista}] = \emptyset$

$\text{Privremeno}[\text{IDParkiralista}] \cap \text{Otvoreno}[\text{IDParkiralista}] = \emptyset$

$\text{Otvoreno}[\text{IDParkiralista}] \cap \text{Zatvoreno}[\text{IDParkiralista}] = \emptyset$

$\text{Otvoreno}[\text{IDParkiralista}] \cap \text{Zatvoreno}[\text{IDParkiralista}] \cap \text{Zatvoreno}[\text{IDParkiralista}] = \emptyset$

$\text{Period}(\{\text{IDPeriod}, \text{PeriodDatum_od}, \text{PeriodDatum_do}, \text{NazivDogadjaja}\}, \{\text{IDPeriod}\})$

$\text{Ima}(\{\text{IDPeriod}, \text{IDParkiralista}\}, \{\text{IDPeriod} + \text{IDParkiralista}\})$

$\text{Ima}[\text{IDPeriod}] \subseteq \text{Period}[\text{IDPeriod}]$

$\text{Ima}[\text{IDParkiralista}] \subseteq \text{Privremeno}[\text{IDParkiralista}]$

$\text{Privremeno}[\text{IDParkiralista}] \subseteq \text{Ima}[\text{IDParkiralista}]$

$\text{Zona}(\{\text{IDZone}, \text{VrstaZone}\}, \{\text{IDZone}\})$

$\text{U_okviru}(\{\text{IDGrada}, \text{IDZone}, \text{BrojZaUplatu}, \text{OgranicenjeParkiranja}, \text{CenaPoSatu}, \text{Popust}\},$
 $\{\text{IDGrada} + \text{IDZone}\})$

$\text{U_okviru}[\text{IDGrada}] \subseteq \text{Grad}[\text{IDGrada}]$

$\text{U_okviru}[\text{IDZone}] \subseteq \text{Zona}[\text{IDZone}]$

$\text{Pripada}(\{\text{IDParkiralista}, \text{IDGrada}, \text{IDZone}, \text{DnevnaVazi}, \text{CenaZaDan}, \text{BrojZaUplatuDnevne},$
 $\text{CenaZaDanSaPopustom}\}, \{\text{IDParkiralista} + \text{IDGrada} + \text{IDZone}\})$

$\text{Pripada}[\text{IDParkiralista}] \subseteq \text{Otvoreno}[\text{IDParkiralista}]$

$\text{Otvoreno}[\text{IDParkiralista}] \subseteq \text{Pripada}[\text{IDParkiralista}]$

$\text{Pripada}[\text{IDGrada} + \text{IDZone}] \subseteq \text{U_okviru}[\text{IDGrada} + \text{IDZone}]$

$\text{DnevnaKarta}(\{\text{IDDnevneKarte}, \text{IDUplate}, \text{IDZone}, \text{IDGrada}\}, \{\text{IDDnevneKarte}\})$

$\text{DnevnaKarta}[\text{IDGrada} + \text{IDZone}] \subseteq \text{U_okviru}[\text{IDGrada} + \text{IDZone}]$

$\text{DnevnaKarta}[\text{IDUplate}] \subseteq \text{Uplata}[\text{IDUplate}]$

$\text{Null}(\text{DnevnaKarta}, \text{IDUplate}) = \perp$

$\text{Null}(\text{DnevnaKarta}, \text{IDZone}) = \perp$

$\text{Null}(\text{DnevnaKarta}, \text{IDGrada}) = \perp$

$\text{Sluzba_parking_servisa}(\{\text{IDSluzbe}, \text{NazivSluzbe}, \text{BrojZaposlenih}, \text{IDGrada}\}, \{\text{IDGrada} +$
 $\text{IDSluzbe}\})$

$\text{Sluzba}[\text{IDGrada}] \subseteq \text{Grad}[\text{IDGrada}]$

$\text{Redni_broj}(\{\text{RbrZaposlenja}\}, \{\text{RbrZaposlenja}\})$

64

$\text{Zaposleni}(\{\text{Mbr}, \text{ImeZaposlenog}, \text{PrezimeZaposlenog}, \text{KontaktZaposlenog}, \text{AdresaZaposlenog},$

Plata}, {Mbr}))

Zaposljava({IDSluzbe, IDGrada, Mbr, RbrZaposlenja, DatumOd, DatumDo}, {IDSluzbe + IDGrada + Mbr + RbrZaposlenja})

Zaposljava[IDSluzbe + IDGrada] \subseteq Sluzba_parking_servisa[IDSluzbe + IDGrada]

Zaposljava[RbrZaposlenja] \subseteq Redni_broj[RbrZaposlenja]

Zaposljava[Mbr] \subseteq Zaposleni[Mbr]

Zaposleni[Mbr] \subseteq Zaposljava[Mbr]

SQL

SQL je ANSI (American National Standards Institute) standardni računarski jezik za pristup i manipulaciju podataka u relacionim bazama podataka. Naredbe SQL-a se koriste za pretraživanje i ažuriranje podataka u bazama podataka.

DDL

DDL (Data Definition Language) predstavlja jedan od četiri ključne logičke grupe SQL Server komande.

DDL komande se koriste za definisanje i izmenu objekata baze podataka. Ove komande se uglavnom brinu o dizajnu i strukturi baze podataka.

DDL komande su:

CREATE - kreiramo novu tabelu, bazu, proceduru, view...

ALTER - obično koristimo za izmjene u tabeli npr. za dodavanje ili uklanjanje atributa

DROP - koristimo za brisanje objekata iz naše baze podataka

DML

DML (Data Manipulation Language) komande se uglavnom odnose na manipulaciju sa rekordima u tabeli, pa se tako sa njima mogu uzimati/povlačiti rekordi po određenom kriterijumu, unositi novi podaci u tabelu, editovati postojeće i naravno brisati rekorde koji su nepotrebni.

DML komande su:

SELECT - selektujemo/povlačimo rekorde iz tabele u bazi podataka,

INSERT - unos novih rekorda u tabelu,

UPDATE - editovanje/izmjena već postojećih rekorda,

DELETE - brisanje postojećih rekorda u tabeli

OBJEKTI

Šema

Šema predstavlja kolekciju svih objekata koji dele isti prostor imenovanja. Može sadržati jednu ili više tabela, a svaka tabela može pripadati logički tačno jednoj šemi.

Naredba za kreiranje šeme:

CREATE SCHEMA <naziv šeme>

Izbacivanje šeme može biti CASCADE - izbacuje šemu i objekte iz nje ili

RESTRICT - brisanje šeme koja je prazna i ostvaruje se naredbom:

DROP SCHEMA <naziv šeme> CASCADE | RESTRICT

Tabela

Podaci se u SQL-u čuvaju se i prikazuju u obliku tabela. Svaka kolona tabele ima ime i tip podatka ili domen. Red je najmanja jedinica podatka koja se može uneti u tabelu ili izbrisati iz nje.

Kreiranje tabela

CREATE TABLE <naziv tabele>

(<naziv kolone1> <tip podatka> NOT NULL],

<naziv kolone> <tip podatka> [NOT NULL], ..)

Dodavanje nove kolone

ALTER TABLE <naziv tabele>

[ADD COLUMN] <definicija kolone> ;

Izmena postojeće kolone

ALTER TABLE <naziv tabele>

[ALTER COLUMN] <naziv kolone>

SET DEFAULT <vrednost> |

DROP DEFAULT;

Izbacivanje kolone iz tabele

ALTER TABLE <naziv tabele>

DROP [COLUMN] <naziv kolone> ;

Dodavanje ili izbacivanje ograničenja na vrednosti

ALTER TABLE <naziv tabele>

ADD [CONSTRAINT <naziv ograničenja>] <ograničenje tabele> |

DROP CONSTRAINT <naziv ograničenja> ;

Brisanje sadržaja tabele

DELETE TABLE <naziv tabele>;

Brisanje tabele i njenog sadržaja

DROP TABLE <naziv tabele>;

Sekvenca

Sekvenca je deljeni objekat koji automatski generiše jedinstvene brojeve. Najčešće se koristi za kreiranje PK (primary key) vrednosti.

Kreiranje sekvence:

CREATE SEQUENCE [schema_name .] sequence_name

[AS [built_in_integer_type | user-defined_integer_type]]

[START WITH <constant>]

[INCREMENT BY <constant>]

[{ MINVALUE [<constant>] } | { NO MINVALUE }]

[{ MAXVALUE [<constant>] } | { NO MAXVALUE }]

[CYCLE | { NO CYCLE }]

Brisanje sekvence:

DROP SEQUENCE [IF EXISTS] { database_name.schema_name.sequence_name |
schema_name.sequence_name | sequence_name } [,...n] [;]

Procedure

Struktura pisanja uskladištene procedure:

1. Uslovni DROP

- proverava se da li postoji procedura
- ako postoji dropuje se. Bolje rešenje od ALTER.

2. Deklarisanje imena uskladištene procedure,

- sintaksa CREATE PROCEDURE (ili CREATE PROC)
- imeSeme.imeProcedure

3. Deklarisanje parametra uskladištene procedure

- struktura je vrlo slična DECLARE sintaksi za definisanje promenljivih

Nije obavezno da procedura ima parametre

Parametri mogu da budu obavezni ili opcioni.

- Opcioni parametri su oni koji imaju definisanu podrazumevanu vrednost.

Parametri mogu biti ulazni i izlazni.

- Izlazni su označeni sa OUTPUT

Parametri su tretirani kao varijable unutar procedure

4. Nakon liste parametara se definiše procedura nakon

- ključne reči AS

5. BEGIN/END označava blok naredbi koje definišu proceduru.

6. SET NOCOUNT ON:

- isključuje ispisivanje poruke na konzoli, kada se izvršava procedura
- kod često pozivanih procedura, isključivanje poruke ubrzava izvršavanje

7. RETURN

- prekida izvršavanje procedure i vraća kontrolu pozivaocu. Može biti više RETURN komandi u proceduri

Funkcije

FN - SQL skalarne funkcije

IF - SQL inline table-valued funkcije

TF - SQL table-valued funkcije

Pozivi funkcija u SQL bloku mogu biti na bilo kom mestu gde se i ugrađene funkcije mogu pozivati:

- ☐ u listi SELECT objekata,
- ☐ u uslovu kod WHERE ili HAVING klauzula,
- ☐ CONNECT BY, START WITH, ORDER BY i GROUP BY klauzula,
- ☐ VALUES klauzule u INSERT naredbi ili
- ☐ SET klauzule u UPDATE naredbi.

Ograničenja pri pozivu funkcija u SQL izrazima:

- ☐ Mogu se koristiti samo uskladištene funkcije (ne i procedure).
- ☐ Mora se koristiti poziciona notacija argumenata.
- ☐ Ne mogu se pozivati u CHECK ograničenju, kao ni pri dodeli predefinisanih vrednosti.
- ☐ Mogu imati samo IN parametre.
- ☐ Tip podataka ne može biti BOOLEAN, RECORD ili TABLE, već neki od SQL validnih tipova.
- ☐ Funkcije koje se pozivaju iz SQL izraza ne mogu da sadrže DML komande (ne može INSERT, UPDATE, DELETE u okviru funkcije).
- ☐ Funkcije koje se pozivaju iz UPDATE/DELETE komandi za tabelu T ne mogu da sadrže DML komande za istu tu tabelu T ili da vrše upit nad njom.
- ☐ Funkcije koje se pozivaju iz SQL naredbi ne mogu da sadrže naredbe koje završavaju transakciju (COMMIT ili izvršna DDL naredba).
- ☐ Funkcija ne može da poziva neki podprogram koji narušava neki od restrikcija.

Trigери

- Trigери su specijalne uskladištene procedure, koje su vezane za određene DML (i DDL) događaje.
- Trigger ne može eksplicitno da se pozove, vezani događaj je okidač (trigger) za izvršavanje triggera
- DML grupa događaja INSERT, UPDATE, DELETE
- DML trigger reaguje na jedan od ili na kombinaciju DML događaja
- SQL server podržava dve vrste triggera
 - * AFTER trigger - koji se okida nakon izvršene DML naredbe
 - * INSTEAD OF trigger - koji se izvršava UMETO DML naredbe
- AFTER trigere je moguće definisati samo nad tabelama
- INSTEAD OF trigger je moguće definisati i nad pogledima
- Trigger se izvršava samo jednom za jednu DML instrukciju, bez obzira koliko vrsta je pod uticajem instrukcije
- Trigger je deo transakcije DML instrukcije koja je dovela do okidanja triggera
- Pozivanje ROLLBACK TRANSACTION iz triggera rezultuje revertovanju svih promena (posledica triggera i dml instrukcije)
- Normalan izlazak iz triggera je poziv RETURN, kao i iz uskladištene procedure
- U telu triggera, moguće je pristupiti privremenim tabelama koje se zovu inserted i deleted. Tabele sadrže torke koje su pod uticajem DML instrukcije koja je dovela do okidanja triggera
- U slučaju INSERT i UPDATE događaja: inserted tabela sadrži novo stanje, nove torke.
- U slučaju UPDATE i DELETE događaja: deleted table sadrži staro stanje, stare torke.

Kursor

Kursor obezbeđuje mogućnost rada sa nekom tabelom ili rezultatom upita u režimu koji je slogovno orijentisan i odgovara tradicionalnom fajl-sistemu sa sekvencijalnim i direktnim datotekama. Kursor se prvo deklariše, zatim se otvara pa se koristi i na kraju se zatvara.

Deklaracija kursora:

```
DECLARE NazivKursora CURSOR FOR
```

```
SELECT upit
```

```
OPEN NazivKursora
```

```
FETCH NEXT FROM NazivKursora INTO @...
```

```
WHILE @@FETCH_STATUS = 0 *
```

0- uspesno izvršen FETCH, 1 -neuspešno izvan result seta, 2 -neuspešno red nedostaje

```
BEGIN
```

```
-- operacije nad vrstom
```

```
-- preuzimanje sledeće vrste iz kursora
```

```
FETCH NEXT FROM plata_cursor INTO @..;
```

```
END;
```

```
-- zatvaranje kursora, deklaracija, tj. referenca na kursor ostaje
```

```
CLOSE NazivKursora;
```

```
-- uklanjanje i reference na kursor DEALLOCATE NazivKursora;
```

U prilogu su predati:

- script za kreiranje tabela
- script za punjenje tabela
- script za brisanje tabela
- script sa upitima
- script sa funkcijama
- script sa procedurama
- script sa triggerima

UPITI

Prvi upit:

Prikazuje informacije o zaposlenom (ime, prezime, adresu i platu).

Upotrebom **case iskaza** vrši se procena plata i u zavisnosti od visine plate ispisuje se da li zaposleni ima minimalnu, prosečnu ili visoku platu. Izlistati informacije za sve one zaposlene koji imaju prezime sa više od (ili tačno) pet slova (**len(prezime_zap) >= 5**). Sve podatke izlistati opadajuće po plati.

Drugi upit:

Prikazuje informacije za koliko dana će se održati manifestacija u gradu (ili pre koliko dana se održala) pomoću funkcije (**DATEDIFF(day, datum1, datum2)**). Takođe, prikazati i naziv grada u kojem se održava manifestacija, koja je manifestaciju u pitanju, broj za uplatu privremenog parkirališta koji se koristi prilikom manifestacije kao i cenu privremenog parkirališta. Sve podatke izlistati po imenima rastuće.

Pomoću ovog upita se dobijaju informacije o periodu manifestacije što olakšava pretragu uz pomoć prosleđenog datuma.

Treći upit:

Prikazuje informacije o zaposlenima koji žive u bilo kom gradu sem Beograda i koji imaju platu između 60000 i 150000. Ukoliko zaposleni ne radi više u “Parking servisu” ispisati poruku 'Nije više zaposlen/a', što predstavlja karakteristiku ovog upita, koja je realizovana pomoću **iif(datum_do is null, 'Nije više zaposlen/a', naziv_grada)**.

Četvrti upit:

Prikazuje informacije o tome koliko ima dnevnih karata koje se mogu koristiti u dve ili više zona pomoću **count** grupne funkcije za sve one zone kojima se naziv završava na ‘a’. Izlistati cenu po satu i vremensko ograničenje zadržavanja u toj zoni. Sve podatke izlistati po ceni po satu opadajuće.

Peti upit:

Napisati naziv grada sa brojem oblasti u tom gradu (**count**), cenama po satu i cenama po satu sa popustom. Izračunati procenat cene sa popustom u odnosu na procenat realne cene. Ukoliko je cena snižena za 25% ispisati 75% kao procenat nove cene. Izlistati podatke gde važi da je vremensko ograničenje parkiranja tri sata. Poređati po ceni po satu rastuće.

FUNKCIJE

Prva funkcija:

Prva funkcija `NazivSluzbe` vraća naziv službe u kojem zaposleni radi na osnovu prosleđenog matičnog broja. Ukoliko nije unet odgovarajući mbr, ispisati poruku 'Uneli ste pogresan mbr!'.

Druga funkcija:

`ParkMesta_tokomManifestacije` je funkcija koja na osnovu prosleđenog datuma ispisuje ukupan broj parking mesta u privremenom parkiralištu koji je na raspolaganju tokom manifestacije, od kada do kada traje manifestacija, kao i naziv manifestacije koja se u tom periodu održava.

PROCEDURE

Prva procedura:

Prva procedura pod nazivom ZaposleniPoSluzbi ispisuje podatke o zaposlenima na osnovu prosleđenog id-a.

Podaci koji treba da budu ispisani su sledeći:

Ako postoje zaposleni u sluzbi ispisati sledece u formi:

U <naziv sluzbe> sa id-em <id sluzbe> su zaposleni:

<redni broj> <ime i prezime zaposlenog> je zaposlen/a po <redni broj zaposlenja> .put

Ukupno radnika: <broj radnika zaposlenih u toj sluzbi>

Ako ne postoji sluzba sa prosledjenim id-em ispisati poruku: 'Ne postoji sluzba sa takvom sifrom'.

Druga procedura:

Procedura pod nazivom PRIMANJA_ZAPOSLENOG proverava platu zaposlenog na osnovu prosleđenog matičnog broja i vrši upoređivanje plate u odnosu na prosečnu platu u službi u kojoj zaposleni radi.

-Ako ima platu vecu od prosecne:

<prezime radnika> sa maticnim broj <mbr> je zaposlen u <naziv sluzbe>, a plata mu je veca od prosecne

-Ako ima platu manju od prosecne:

<prezime radnika> sa maticnim broj <mbr> je zaposlen u <naziv sluzbe>, a plata mu je manja od prosecne

-I inace

<prezime radnika> sa maticnim broj <mbr> je zaposlen u <naziv sluzbe>, a plata mu je prosecna

TRIGERI

Prvi trigger:

Trigger se aktivira pri insert-u datuma zapošljavanja zaposlenog od strane službe parking servisa.

Datum zapošljavanja mora da bude stariji ili jednak današnjem datumu, ne može biti noviji.

Drugi trigger:

Trigger se aktivira svaki put kada zaposleni promeni adresu tako da se arhiviraju i prethodni podaci.

Za potrebe triggera potrebno je kreirati tabelu parkingServis.informacije u kojoj će se nalaziti ti podaci.

```
CREATE TABLE parkingServis.informacije
(
    mbr numeric(8) references parkingServis.zaposleni(mbr),
    datum date,
    staraAdresa nvarchar(20),
    novaAdresa nvarchar(20),
    constraint rk_informacije primary key(mbr)
);
```

ZAKLJUČAK

Kriterijumi projekta su zadovoljeni. Omogućeno je unošenje, modifikovanje i brisanje podataka iz baze.

Implementirani su objekti baze podataka kao što su trigeri, funkcije i procedure koje imaju za cilj da olakšaju poslovanje "Parking servisa" i da korisnicima baze podataka pruže informacije na efikasan način.