

**INFORMACIONI SISTEM ZA PODRŠKU ORGANIZOVANJA**

**NAUČNE KONFERENCIJE**

**-SISTEMI BAZA PODATAKA-**

**Enio Kurteši IT27g2019**

**Novi Sad, 2022.**

**SADRŽAJ**

|  |  |
| --- | --- |
| [UVOD](#page3) | [2](#page3) |
| [ANALIZA PROGRAMSKOG DOMENA](#page3) | [2](#page3) |
| [ER MODEL](#page4) | [3](#page4) |
| [ER MODEL PODŠEME](#page5) | [4](#page5) |
| [TABELARNI PRIKAZ OBELEŽJA I OGRANIČENJA](#page6) | [5](#page6) |
| [RELACIONI MODEL](#page7) | [6](#page7) |
| [DDL](#page7) | [6](#page7) |
| [DML](#page7) | [6](#page7) |
| [SQL](#page7) | [6](#page7) |
| [OBJEKTI](#page7) | [6](#page7) |
| [UPITI](#page8) | [7](#page8) |
| [FUNKCIJE](#page9) | [8](#page9) |
| [PROCEDURE](#page10) | [9](#page10) |
| [TRIGERI](#page11) | [10](#page11) |
| [ZAKLJUČAK](#page12) | [11](#page12) |

**UVOD**

Projekat koji je predstavljen u ovoj dokumentaciji reprenzetuje razvoj informacionog sistema za podršku pri organizovanju naučne konferencije. Reč je o nastavku projekta iz V semestral gde je zadatak bio isprojektovati šemu baze podataka i izvršiti prevođenje iz ER u relacioni model podataka. Izabrana podšema za ovaj projekat je pod nazivom ‘Učestvovanje na konferenciji,donatori i recezenti’, koja ima za cilj evidenciju učesnika, predavača, donatora i recezenata za sve konferencije koje se nalaze u sistemu.

Kao prilog projektu priložene će biti sledeće skripte:

• script za kreiranje i brisanje schema i tabela(DDL)

• script za insert podataka(DML)

• script sa SQL upitima

• script za kreiranje funkcija,•

• script za kreiranje procedura

• script za kreiranje trigera,

**ANALIZA PROGRAMSKOG DOMENA**

Cilj projekta jeste podrška kod organizovanja naučnih konferencija. Realizacija projekta omogućava zainteresovanim korisnicima, pre svega organizatorima konferencija da odgovore na sledeća pitanja:

1. Ko su učesnici na konferenciji?
2. Kog tipa su učesnici?
3. Ko predaje na konferenciji?
4. Da li su učesnici platili kotizaciju i kakvu?
5. U kojim salama će se održati konferencije?
6. Ko je zaslužan za odabir predavača?
7. Da li postoji neko ko bi donirao sredstva za realizaciju konferencije?

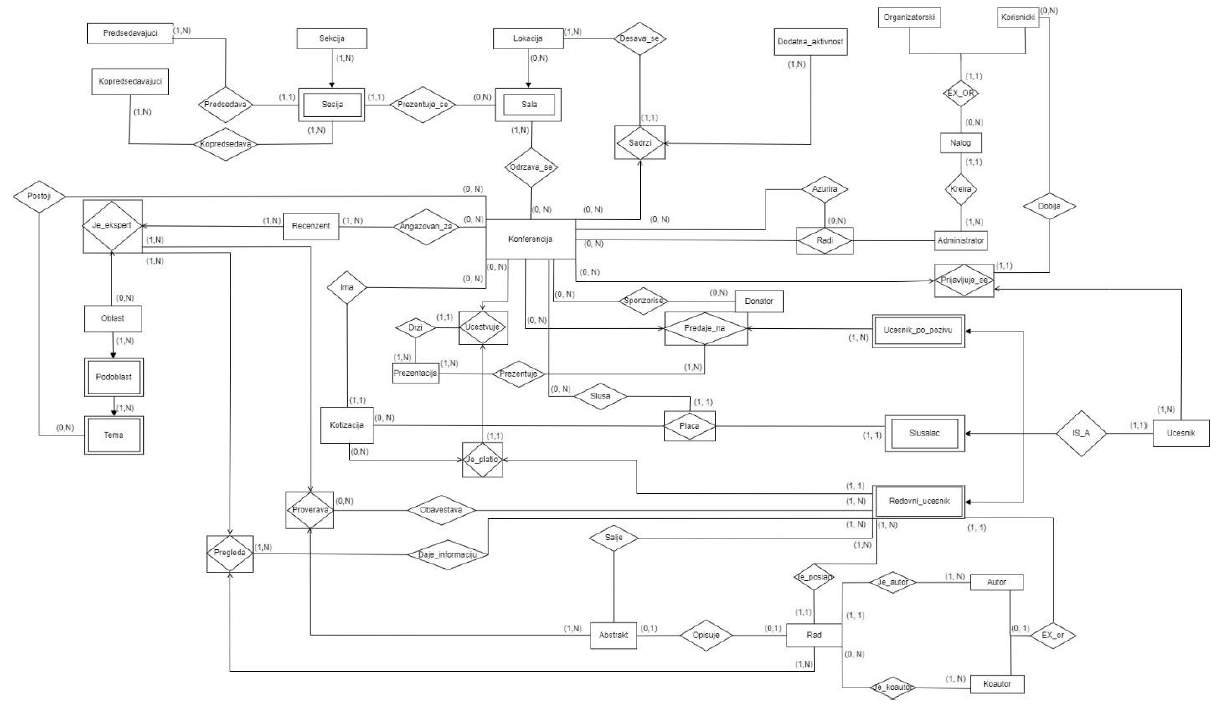
U nastavku teksta, sledi objašnjenje podšeme i koje sve probleme želimo da identifikujemo i rešimo.

**ER MODEL**

Prikaz er modela.

Slika ER modela

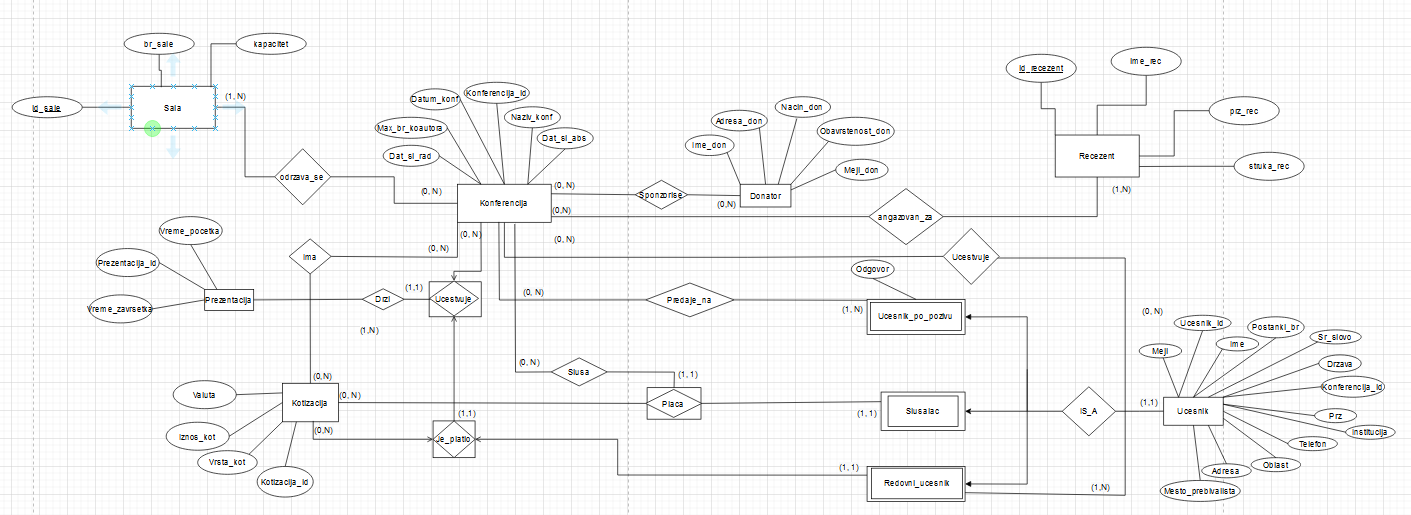
Slika 1 ER model



Slika 1: ER model

**ER MODEL PODŠEME**

Prikaz podšeme koja se koristi u projektu **uz objašnjenje podšeme**.



Slika 2: ER podšema

U podšemi **Doniranje i učestvovanje na konferenciji** učestvuju tipovi entiteta:

* UČESNIK
* SLUSALAC
* REDOVAN UČESNIK
* UČESNIK PO POZIVU
* KOTIZACIJA
* DONATOR
* PREZENTACIJA
* SALA
* RECEZENT

U podšemi postoji tip entiteta Učesnik koji se deli na Slušaoca, Redovnog učesnika i Učesnika po pozivu. Za učestvovanje u konferenciji neophodno je uplatiti kotizaciju. Redovni učesnik koji je platio kotizaciju, učestvuje na konferenciji. Slušalac, takođe, mora da izvrši uplatu da bi mogao da sluša konferenciju. Sa druge strane, učesnik po pozivu daje odgovor na poziv koji dobija i predaje na temu za koju je stručan bez ikakve uplate kotizacije.U ovoj podšemi, vodi se evidencija i o donatorima, čija je uloga da uz pomoć svog doprinosa, bilo novčanog, robnog, ili nekog drugog vida sponzorstva, potpomognu realizaciju konferencije.Da bi podšema bila bogatija, uveden je tip entiteta recezent, koji u ovako kreiranom sistemu ima zadatak

da bira ko će biti učesnik po pozivu, odnosno predavač. Uveden je, takođe, tip entiteta

Sala koja je povezana sa Konferencijom I govori nam u kojoj Sali se održava koja

Konferencija.

**TABELARNI PRIKAZ OBELEŽJA I OGRANIČENJA**

Tabela 1. Tip poveznika Angazovan\_za

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Redni broj** | **Naziv obeležja** | **Opis obeležja** | **Tip podataka** | **Dužina podataka** | **Null** | **Uslov** |
| **1.** | ID\_RECEZENT | Identifikacioni broj  recezenta | NUMERIC | 8 | ⊥ | ∆ |
| **2.** | ID\_KONF | Identifikacioni  broj  konferencije | NUMERIC | 8 | ⊥ | ∆ |
| **Ključ** | **K={ID\_RECEZENT + ID\_KONF}** | | | | | |

Tabela 2. Tip entiteta Donator

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Redni broj** | **Naziv obeležja** | **Opis obeležja** | **Tip podataka** | **Dužina podataka** | **Null** | **Uslov** |
| **1.** | ID\_DON | Identifikacioni broj  donatora | INTEGER | 8 | ⊥ | ∆ |
| **2.** | IME\_DON | Ime donatora | VARCHAR | 18 | ⊥ | ∆ |
| **3.** | ADRESA\_DON | Adresa donatora | VARCHAR | 30 | ⊥ | ∆ |
| **4.** | MAIL\_DON | Mail donatora | VARCHAR | 30 | ⊥ | ∆ |
| **5.** | NACIN\_DON | Način doniranja | CHAR | 30 | ⊥ | d = {‘robna’, ‘novcana’} |
| **6.** | OBAVESTENOST\_DON | Obaveštenost  donatora | VARCHAR | 2 | ⊥ | d = { ‘da’,’ne’} |
| **Ključ** | **K={ID\_DON}** | | | | | |

Tabela 3. Tip poveznika Ima

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Redni broj** | **Naziv obeležja** | **Opis obeležja** | **Tip podataka** | **Dužina podataka** | **Null** | **Uslov** |
| **1.** | ID\_KOT | Identifikacioni broj  kotizacije | NUMERIC | 8 | ⊥ | ∆ |
| **2.** | ID\_KONF | Identifikacioni  broj  konferencije | NUMERIC | 8 | ⊥ | ∆ |
| **Ključ** | **K={ID\_KOT + ID\_KONF}** | | | | | |

Tabela 4. Tip entiteta Konferencija

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Redni broj** | **Naziv obeležja** | **Opis obeležja** | **Tip podataka** | **Dužina podataka** | **Null** | **Uslov** |
| **1.** | ID\_KONF | Identifikacioni broj  konferencije | NUMERIC | 8 | ⊥ | d > 0 |
| **2.** | NAZIV\_KONF | Naziv konferencije | VARCHAR | 18 | ⊥ | d > 0 |
| **3.** | DATUM\_KONF | Datum početka konferencije | DATE |  | ⊥ | ∆ |
| **4.** | KRAJ\_KONF | Datum kraja konferencije | DATE |  | ⊥ | ∆ |
| **5.** | BR\_ORGANIZATORA | Broj organizatora | NUMERIC | 7 | ⊥ | ∆ |
| **6.** | MAX\_BR\_KOAUT | Max broj koautora | NUMERIC | 4 | ⊥ | ∆ |
| **7.** | MAX\_BR\_RAD | Max broj radova | NUMERIC | 4 | ⊥ | ∆ |
| **8.** | DAT\_SL\_ABS | Datum slanja abstrakta | DATE |  | ⊥ | dat\_sl\_abs < dat\_sl\_rad |
| **9.** | DAT\_SL\_RAD | Datum slanja rada | DATE |  | ⊥ | ∆ |
| **Ključ** | **K={ID\_KONF}** | | | | | |

Tabela 5. Tip entiteta Kotizacija

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Redni broj** | **Naziv obeležja** | **Opis obeležja** | **Tip podataka** | **Dužina podataka** | **Null** | **Uslov** |
| **1.** | ID\_KOT | Identifikacioni broj  kotizacije | NUMERIC | 8 | ⊥ | ∆ |
| **2.** | IZNOS\_KOT | Iznos kotizacije | NUMERIC | 8 | ⊥ | d > 0 |
| **3.** | VRSTA\_KOT | Vrsta kotizacije | VARCHAR | 15 | ⊥ | d = {‘studentska, ‘redovna’} |
| **4.** | VALUTA | Valuta | VARCHAR | 20 | ⊥ | d = {‘euro’,’din} |
| **Ključ** | **K={ID\_KOT}** | | | | | |

Tabela 6. Tip poveznika Odrzava\_se

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Redni broj** | **Naziv obeležja** | **Opis obeležja** | **Tip podataka** | **Dužina podataka** | **Null** | **Uslov** |
| **1.** | ID\_KONF | Identifikacioni broj  konferencije | NUMERIC | 8 | ⊥ | ∆ |
| **2.** | ID\_SALE | Identifikacioni  broj  sale | NUMERIC | 8 | ⊥ | ∆ |
| **Ključ** | **K={ID\_SALE + ID\_KONF}** | | | | | |

Tabela 7. Tip poveznika Predaje\_na

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Redni broj** | **Naziv obeležja** | **Opis obeležja** | **Tip podataka** | **Dužina podataka** | **Null** | **Uslov** |
| **1.** | ID\_UCESNIK\_POZIV | Identifikacioni broj  učesnika po pozivu | NUMERIC | 8 | ⊥ | ∆ |
| **2.** | ID\_KONF | Identifikacioni broj konferencije | NUMERIC | 8 | ⊥ | d > 0 |
| **3.** | ID\_PREZENTACIJA | Identifikacioni broj prezentacije | NUMERIC | 8 | ⊥ | ∆ |
| **Ključ** | **K={ID\_UCESNIK\_POZIV + ID\_KONF}** | | | | | |

Tabela 8. Tip entiteta Prezentacija

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Redni broj** | **Naziv obeležja** | **Opis obeležja** | **Tip podataka** | **Dužina podataka** | **Null** | **Uslov** |
| **1.** | ID\_PREZENTACIJA | Identifikacioni broj prezentacije | NUMERIC | 8 | ⊥ | ∆ |
| **2.** | VREME\_POCETKA | Vreme početka prezentacije | DATE | 8 | ⊥ | vreme\_pocetka < vreme\_zavrsetka |
| **3.** | VREME\_ZAVRSETKA | Vreme završetka prezentacije | DATE | 8 | ⊥ | ∆ |
| **Ključ** | **K={ID\_PREZENTACIJA}** | | | | | |

Tabela 9. Tip entiteta Recezent

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Redni broj** | **Naziv obeležja** | **Opis obeležja** | **Tip podataka** | **Dužina podataka** | **Null** | **Uslov** |
| **1.** | ID\_RECEZENT | Identifikacioni broj  recezenta | NUMERIC | 8 | ⊥ | ∆ |
| **2.** | IME\_REC | Ime recezenta | VARCHAR | 18 | ⊥ | ∆ |
| **3.** | PRZ\_REC | Prezime recezenta | VARCHAR | 18 | ⊥ | ∆ |
| **4.** | STRUKA\_REC | Struka recezenta | VARCHAR | 30 | ⊥ | ∆ |
| **Ključ** | **K={ID\_RECEZENT}** | | | | | |

Tabela 10. Tip poveznika Redovni\_Ucesnik

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Redni broj** | **Naziv obeležja** | **Opis obeležja** | **Tip podataka** | **Dužina podataka** | **Null** | **Uslov** |
| **1.** | ID\_REDOVNI\_UCESNIK | Identifikacioni broj  redovnog učesnika | NUMERIC | 8 | ⊥ | ∆ |
| **2.** | ID\_KOT | Identifikacioni  broj  kotizacije | NUMERIC | 8 | ⊥ | ∆ |
| **Ključ** | **K={ID\_REDOVNI\_UCESNIK}** | | | | | |

Tabela 11. Tip entiteta Sala

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Redni broj** | **Naziv obeležja** | **Opis obeležja** | **Tip podataka** | **Dužina podataka** | **Null** | **Uslov** |
| **1.** | ID\_SALE | Identifikacioni broj  sale | NUMERIC | 8 | ⊥ | ∆ |
| **2.** | BR\_SALE | Broj sale | NUMERIC | 8 | ⊥ | ∆ |
| **3.** | KAPACITET | Kapacitet sale | NUMERIC | 7 | ⊥ | d > 0 |
| **Ključ** | **K={ID\_SALE}** | | | | | |

Tabela 12. Tip entiteta Slušalac

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Redni broj** | **Naziv obeležja** | **Opis obeležja** | **Tip podataka** | **Dužina podataka** | **Null** | **Uslov** |
| **1.** | ID\_SLUŠALAC | Identifikacioni broj  slušaoca | NUMERIC | 8 | ⊥ | ∆ |
| **2.** | ID\_KOT | Identifikacioni  broj  kotizacije | NUMERIC | 8 | ⊥ | ∆ |
| **Ključ** | **K={ID\_SLUŠALAC}** | | | | | |

Tabela 13. Tip poveznika Sponzoriše

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Redni broj** | **Naziv obeležja** | **Opis obeležja** | **Tip podataka** | **Dužina podataka** | **Null** | **Uslov** |
| **1.** | ID\_DON | Identifikacioni broj  donatora | NUMERIC | 8 | ⊥ | ∆ |
| **2.** | ID\_KONF | Identifikacioni  broj  konferencije | NUMERIC | 8 | ⊥ | ∆ |
| **Ključ** | **K={ID\_DON + ID\_KONF}** | | | | | |

Tabela 14. Tip entiteta Ucesnik

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Redni broj** | **Naziv obeležja** | **Opis obeležja** | **Tip podataka** | **Dužina podataka** | **Null** | **Uslov** |
| **1.** | ID\_UCESNIK | Identifikacioni broj  učesnika | INTEGER | 8 | ⊥ | ∆ |
| **2.** | IME | Ime učesnika | VARCHAR | 18 | ⊥ | d > 0 |
| **3.** | SR\_SLOVO | Srednje slovo učesnika | VARCHAR | 3 | ⊥ | ∆ |
| **4.** | PRZ | Prezime učesnika | VARCHAR | 18 | ⊥ | ∆ |
| **5.** | MAIL | Mail ucesnika | VARCHAR | 30 | ⊥ | ∆ |
| **6.** | ADRESA | Adresa učesnika | VARCHAR | 30 | ⊥ | ∆ |
| **7.** | DRZAVA | Drzava učesnika | VARCHAR | 20 | ⊥ | ∆ |
| **8.** | POSTANSKI\_BR | Poštanski br | NUMERIC | 10 | ⊥ | d > 5 |
| **9.** | INSTITUCIJA | Institucija učesnika | VARCHAR | 30 | ⊥ | ∆ |
| **10.** | MESTO\_PREBIVALISTA | Mesto prebivalista ucesnika | VARCHAR | 30 | ⊥ | ∆ |
| **12.** | TELEFON | Telefon ucesnika | VARCHAR | 20 | ⊥ | ∆ |
| **13.** | DATUM\_RODJENJA | Datum rodjenja ucesnika | DATE |  | ⊥ | ∆ |
| **Ključ** | **K={ID\_UCESNIK}** | | | | | |

Tabela 15. Tip entiteta Ucesnik\_Poziv

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Redni broj** | **Naziv obeležja** | **Opis obeležja** | **Tip podataka** | **Dužina podataka** | **Null** | **Uslov** |
| **1.** | ID\_UCESNIK\_POZIV | Identifikacioni broj  učesnika po pozivu | NUMERIC | 8 | ⊥ | ∆ |
| **2.** | ODGOVOR | Odgovor učesnika | VARCHAR | 2 | ⊥ | d = {‘da’,’ne’} |
| **3.** | CENA | Cena učesnika | DECIMAL | 9 | ⊥ | ∆ |
| **Ključ** | **K={ID\_UCESNIK\_POZIV}** | | | | | |

Tabela 16. Tip poveznika Ucestvuje

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Redni broj** | **Naziv obeležja** | **Opis obeležja** | **Tip podataka** | **Dužina podataka** | **Null** | **Uslov** |
| **1.** | ID\_UCESNIK | Identifikacioni broj  ucesnika | NUMERIC | 8 | ⊥ | ∆ |
| **2.** | ID\_KONF | Identifikacioni  broj  konferencije | NUMERIC | 8 | ⊥ | ∆ |
| **Ključ** | **K={ID\_UCESNIK + ID\_KONF}** | | | | | |

**RELACIONI MODEL**

**Relacione baze podataka** obezbeđuju okruženje iz kojeg se podacima može pristupiti na različite načine bez potrebe za reorganizacijom tabela baze podataka. Svaka tabela ima jedinstveni identifikator ili primarni ključ, koji identifikuje informacije u tabeli, a svaki red sadrži jedinstvenu instancu podataka za kategorije definisane kolonama.

Relacioni model relacione baze podataka razdvaja logičke strukture podataka od fizičkih struktura skladištenja, omogućavajući administratorima baze podataka da upravljaju fizičkim skladištenjem podataka bez uticaja na pristup tim podacima kao logičkoj strukturi. Razlika se takođe odnosi na operacije baze podataka – logičke operacije omogućavaju aplikaciji da specificira sadržaj koji joj je potreban, a fizičke operacije određuju kako treba pristupiti tim podacima, a zatim izvršavaju zadatak.

Glavna prednost relacione baze podataka je njena formalno opisana, tabelarna struktura, iz koje se podaci mogu lako skladištiti, kategorizovati, ispitivati i filtrirati bez potrebe za reorganizacijom tabela baze podataka. Dodatne prednosti relacionih baza podataka uključuju:

**Skalabilnost**: Novi podaci se mogu dodati nezavisno od postojećih zapisa.

**Jednostavnost**: Složeni upiti su laki za korisnike za izvršavanje pomoću SKL-a.

**Tačnost podataka**: Procedure normalizacije eliminišu anomalije dizajna

**Sigurnost**: Podaci u tabelama unutar SUBP-a mogu ograničiti pristup određenim korisnicima.

**Konkurentnost**: Više korisnika može pristupiti istoj bazi podataka istovremeno.

**DDL**

Jezik definicije podataka (DDL) je računarski jezik koji se koristi za kreiranje i modifikovanje strukture objekata baze podataka u bazi podataka. Ovi objekti baze podataka uključuju poglede, šeme, tabele, indekse.

Termin DDL je u nekim kontekstima poznat i kao jezik opisa podataka, jer opisuje polja i zapise u tabeli baze podataka.

Koristi se za uspostavljanje i modifikovanje strukture objekata u bazi podataka baveći se opisima šeme baze podataka. Za razliku od komandi jezika za upravljanje podacima (DML) koje se koriste u svrhe modifikacije podataka, DDL komande se koriste za promenu strukture baze podataka kao što je kreiranje novih tabela ili objekata zajedno sa svim njihovim atributima (tip podataka, naziv tabele, itd.).

Uobičajeni DDL u SQL upitima su **CREATE, ALTER, DROP i TRUNCATE**.

**DML**

Jezik za manipulaciju podacima (DML) je porodica računarskih jezika uključujući komande koje dozvoljavaju korisnicima da manipulišu podacima u bazi podataka. Ova manipulacija uključuje umetanje podataka u tabele baze podataka, preuzimanje postojećih podataka, brisanje podataka iz postojećih tabela i modifikovanje postojećih podataka. DML je uglavnom ugrađen u SKL baze podataka.

Komanda **SELECT** se koristi za preuzimanje podataka iz baze podataka.

Ova komanda omogućava korisnicima baze podataka da preuzmu specifične informacije koje žele iz operativne baze podataka.

Vraća skup rezultata zapisa iz jedne ili više tabela.

Sintaksa : **SELECT \* FROM** <**ime\_tabele**>;

Naredba **INSERT** se koristi za unos podataka u tabelu.

Koristeći ovu komandu, može se dodati jedan ili više zapisa u bilo koju pojedinačnu tabelu u bazi podataka.

Sintaksa : **INSERT INTO <ime\_tabele> (`ime\_kolone` <tip\_podatka>, `ime\_kolone2` <tip\_podatka>, . . . )VALUES (`vrednost1`, `vrednost2`, ...);**

Komanda **UPDATE** se koristi za izmenu zapisa prisutnih u postojećoj tabeli.

Ova komanda ažurira postojeće podatke u tabeli.

Menja podatke jednog ili više zapisa u tabeli.

Sintaksa:

**UPDATE <ime\_tabele>**

**SET <ime\_kolone = vrednost>**

**WHERE uslov;**

Komanda **DELETE** se koristi za brisanje nekih ili svih zapisa iz postojeće tabele.

Briše sve zapise iz tabele.

Sintaksa:

**DELETE FROM <ime\_tabele> WHERE <uslov>;**

**SQL**

Jezik strukturiranih upita (SQL) je standardni jezik orijentisan na podatke koji se prvenstveno koristi za skladištenje, preuzimanje i manipulaciju podacima pomoću jednostavnih isečaka koda(code snippet), nazvanih upiti, u SUBP-u (sistem za upravljanje relacionim bazama podataka).

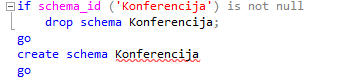
Podaci se čuvaju u SUBP-u na struktuiran način, gde postoje odnosi između različitih entiteta i varijabli u podacima. Ovi odnosi su definisani šemom baze podataka koja ne samo da specificira odnos između različitih entiteta već i organizaciju podataka povezanih sa tim entitetima u bazi podataka.

**OBJEKTI**

**SCHEMA**

U SQL bazama podataka, termin šema baze podataka može imati i drugo, specifičnije značenje. Termin „šema“ se ponekad koristi za predstavljanje imenovane grupe objekata baze podataka. Ova vrsta šeme se preciznije naziva šema definisana od strane korisnika.

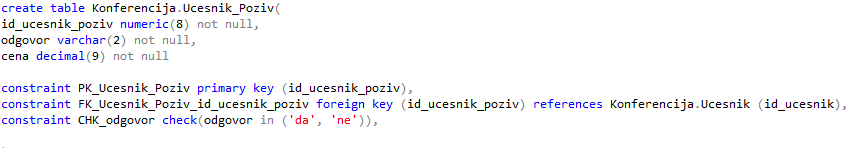
Jedna baza podataka može imati više korisnički definisanih šema, a različite korisnički definisane šeme mogu se koristiti da se različitim korisnicima ili korisničkim grupama da pristup različitim delovima baze podataka.



Slika 3: Kreiranje schema-e

**TABELA**

Prilikom kreiranja tabele koriste se rezervisane reči CREATE TABLE, a zatim se navodi i šema u koju se objekat tipa tabele smešta.. Zatim se navode nazivi obeležja, a za svako obeležje trebalo bi navesti tip podatka i dužinu polja. Praksa je da se ograničenja nad tabelom implementiraju nakon definicije obeležja. Neka od ograničenja su: CHECK, DEFAULT, PRIMARY KEY, FOREIGN KEY, UNIQUE,...



Slika 4: Kreiranje tabele i ograničenja

**SEKVENCA**

Sekvenca je korisnički definisani šemom povezan objekat koji generiše niz numeričkih vrednosti prema specifikaciji sa kojom je sekvenca kreirana. Niz numeričkih vrednosti se generiše u rastućem ili opadajućem redosledu u definisanom intervalu i može se konfigurisati da se ponovo pokrene (ciklus) kada dođe do kraja.

**CREATE SEQUENCE** ime\_sekvence

**START WITH** početna\_vrednost

**INCREMENT BY** inkrement

**MINVALUE** minimalna vrednost

**MAXVALUE** maksimalna vrednost

**CYCLE|NOCYCLE ;**

**PROCEDURA**

Skladištena procedura u SQL-u je grupa SQL naredbi koje su zajedno uskladištene u bazi podataka. Na osnovu iskaza u proceduri i parametara koje se prosleđuju , može se izvršiti jedna ili više DML operacija na bazi podataka.

Procedure pružaju neke ključne prednosti, a to su:

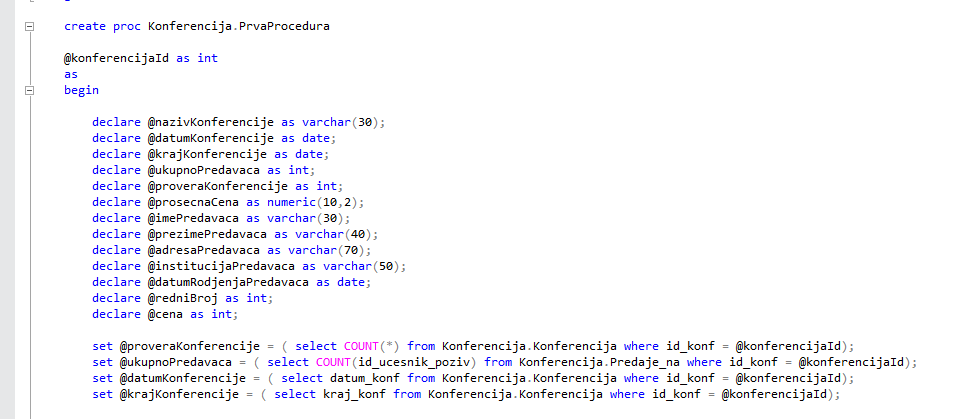
1. **Višekratna upotreba**: Kao što je pomenuto, više korisnika i aplikacija može lako da koristi uskladištene procedure jednostavnim pozivanjem.

2. **Lako se menja**: Mogu se brzo promeniti iskazi u uskladištenoj proceduri kada i kako želimo uz pomoć komande ALTER TABLE.

3. **Bezbednost**: Sačuvane procedure omogućavaju poboljšanje bezbednosti aplikacije ili baze podataka ograničavanjem korisnika direktnog pristupa tabeli.

4. **Nizak mrežni saobraćaj**: server prosleđuje samo ime procedure umesto celog upita, smanjujući mrežni saobraćaj.

5. **Povećava performanse**: Nakon prve upotrebe, plan za uskladištenu proceduru se kreira i čuva u baferu za brzo izvršavanje sledećeg puta.



Slika 5: Kreiranje procedure

**FUNKCIJA**

Korisnički definisana funkcija je rutina Transact-SQL ili Common Language Runtime (CLR) koja prihvata parametre, izvodi operaciju, kao što je složeno izračunavanje, i vraća rezultat te operacije kao vrednost. Povratna vrednost može biti ili skalarna (pojedinačna) vrednost ili tabela.

Sintaksa za kreiranje skalarne funkcije:

CREATE FUNCTION [dbo].[imeFunkcije]

(

@parametar1 int,

@parametar2 int

)

RETURNS INT –tip podatka koji vraća

AS

BEGIN

RETURN @parametar1 + @parametar2

END

Sintaksa za kreiranje funkcije koja vraća tabelu:

CREATE FUNCTION [dbo].[imeFunkcije]

(

@parametar1 int,

@parametar2 char(5)

)

RETURNS @returntable TABLE

(

[c1] int,

[c2] char(5)

)

AS

BEGIN

INSERT @returntable

SELECT @parametar1, @parametar2

RETURN

END

**TRIGER**

Triger je objekat baze podataka sličan uskladištenoj proceduri koji se automatski izvršava kada se u bazi podataka dogodi neki događaj. Postoje različite vrste događaja koji mogu da aktiviraju triger kao što je umetanje ili brisanje redova u tabeli, prijavljivanje korisnika u instancu servera baze podataka, ažuriranje kolone tabele, kreiranje, izmena ili ispuštanje tabele itd.

Postoje tri tipa trigera u SQL Serveru:

DML trigeri se automatski aktiviraju kada se u tabeli dogodi događaj INSERT, UPDATE ili DELETE.

DDL trigeri se automatski pozivaju kada se u bazi podataka dogodi događaj CREATE, ALTER ili DROP. Pokreće se kao odgovor na događaj u domenu servera ili baze podataka.

Triger za prijavu se poziva kada se pokrene događaj LOGON kada se uspostavi korisnička sesija.

CREATE TRIGGER imeTrigera –definiše se ime trigera

ON [dbo].[imeTabele] – nad kojom tabelom

FOR DELETE, INSERT, UPDATE – za koju DML operaciju

AS

BEGIN

SET NOCOUNT ON

END

**KURSOR**

Kursor u SQL Serveru je objekat baze podataka koji nam omogućava da preuzmemo svaki red u isto vreme i manipulišemo njegovim podacima. Kursor nije ništa drugo do pokazivač na red. Uvek se koristi zajedno sa naredbom SELECT.

**DECLARE <ime\_kursora> CURSOR FOR <upit>**

**OPEN <ime\_kursora>**

**FETCH NEXT FROM <ime\_kursora> [ INTO @imeVarijable [ ,...n ] ]**

**CLOSE <ime\_kursora>**

**DEALLOCATE <ime\_kursora>**

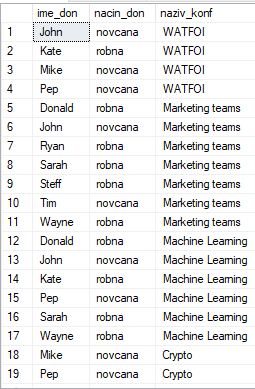
**UPITI**

**Prvi upit** :

Izlistava ime donatora, vrstu donacije, nazive konferencija na kojima je donirao.

U obzir dolaze samo one konferencije koje se održavaju u salama čiji je kapacitet manji od prosečnog kapaciteta dostupnih sala.

Rezultat upita je sortiran po nazivu konferencije opadajuće.



Slika 6: Prvi upit

**Drugi upit :**

Izlistava inicijale najmlađeg učesnika koji učestvuje na konferencijama. Za izvlačenje prvog slova imena i prezimena koristi se funkcija left(ime,1), left(prezime,1).

Izlistava se institucija odakle dolazi najmlađi učesnik, ako je platio kotizaciju, treba ispisati koju je platio, ako nije, ispisati adekvatnu poruku da je oslobodjen plaćanja.

Iz ove provere se može zaključiti da li je najmlađi učesnik došao po pozivu kao predavač, ili pripada grupi učesnika koji su prethodno morali da plate kotizaciju.

Poslednja kolona predstavlja godinu rođenja koja je izvučena uz pomoć datepart funkcije kojoj sam prosledio parametar year i datum rođenja najmlađeg učesnika.



Slika 7: Drugi upit

**Treći upit :**

Izlistava ime i prezime najskupljeg predavača na konferenciji, instituciju odakle dolazi, cenu koju treba platiti da bi održao svoje predavanje,naziv konferencije na kojoj predaje, samo za one konferencije koje se održavaju nedeljom. Izlistava i broj donatora koji su sponzorisali tu konferenciju.



Slika 8: Treći upit

**Četvrti upit :**

Izlistava nazive svih konferencija, mesec u kom se odrzava svaka konferencija

Uz pomoc ovog upita vidimo neke osnovne informacije o konferencijama.

Funkcija Month nam vraća mesec iz datuma kada je konferencija otvorena.

U interesu nam je da izlistamo samo one konferencije koje su dobile sredstva iz Toronta kao i koju vrstu donacije su dobili. Bitno je da je broj organizatora manji od prosečnog broja organizatora na konferencijama.

Koristeći having funkciju prebrojavamo broj organizatora.

Dužina trajanja konferencije je izračunata uz pomoć datediff funkcije gde se prosleđuje parametar day, kao i datum početka i datum završetka konferencije.

Rezultati su sortirani po mesecima rastuće.



Slika 9: Četvrti upit

**Peti upit :**

U petom upitu se za svaku konferenciju izlistava predavač koji ima najpovoljniju cenu.

Ovaj upit je specifičan zato što jedini koristi privremenu tabelu

odnosno, select naredbu u from-u.



Slika 10: Peti upit

**FUNKCIJE**

**Prva funkcija** predstavlja Multi statement Table-Valued funkciju koja za prosleđenu valutu izlistava sledeće: Ime i prezime učesnika vrstu kotizacije koju je platio, iznos kotizacije,adresu i to na sledeći način: spaja obeležja poštanski broj, grad, adresu u jedno. Takođe, izlistava se broj telefona učesnika. Uz pomoć switch case-a ako korisnik dolazi iz Srbije na njegov broj će se dodati prefiks +381, za Crnu Goru je to +382 dok je za BiH reč

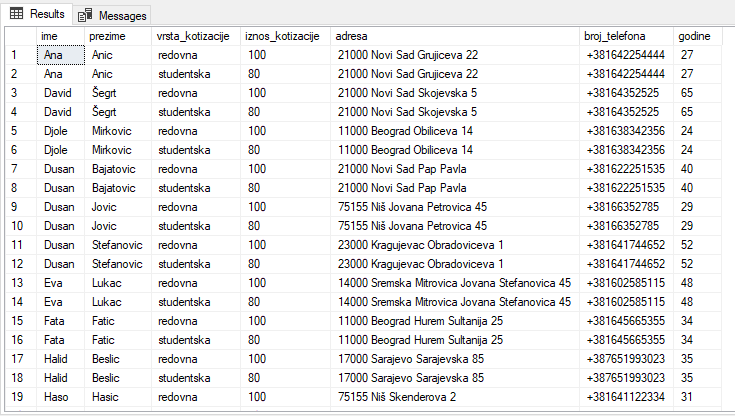
o +387 prefiksu.

Kao poslednja varijabla, izračunavaju se godine učesnika uz pomoć funkcije datediff kojoj se prosleđuje year, datum rođenja i trenutni datum.

Povratna vrednost funkcije jeste tabela, s obzirom da se radi o Multi statement Table-Valued funkciji.

Struktura tabele je definisana neposredno nakon deklarisanja varijable valuta, sa kojom će

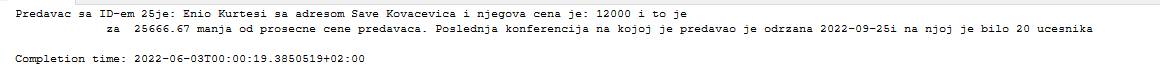
funkcija manipulisati.



Slika 11: Prva funkcija

**Druga funkcija** predstavlja skalarnu funkciju koja za prosledjeni id predavača vraća njegovo ime, prezime, instituciju,cenu predavača na konferenciji,informaciju o tome koliko je prosecna cena predavača, kolika je razlika između prosečne cene i cene prosleđenog predavača, prebrojava koliko učesnika je učestvovalo na konferenciji na kojoj je prosleđeni predavač držao prezentaciju. Bitno je napomenuti da nas u konkretnoj funkciji zanima samo poslednja konferencija na kojoj je predavač bio gost. S obzirom da funkcija vraća veći broj informacija,povratni tip podatka je varchar,dužine 400.

Kako bi se obezbedilo adekvatno izvršavanje funkcije,uz pomoć if uslova su obrađeni svi mogući slučajevi prosleđivanja parametra. Ako varijabla @proveraPredavaca ima vrednost null a @proveraUcesnika vrednost 1, znači da smo izvukli osobu koja jeste učesnik na konferenciji, ali nije predavač. Dalje, ako obe varijable imaju vrednost 0, to znači da prosleđujemo nepostojeću vrednost za ID predavača. Uz pomoć concat funkcije vršimo ispis prethodno navedenih obeležja, u slučaju da smo izvukli predavača koji postoji u bazi.

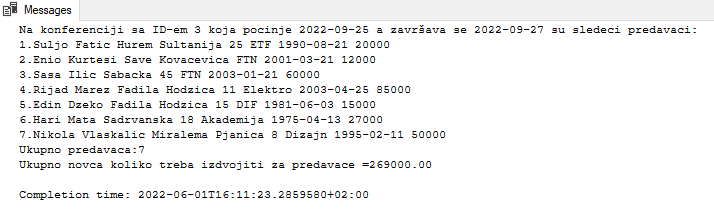


Slika 12: Druga funkcija 1

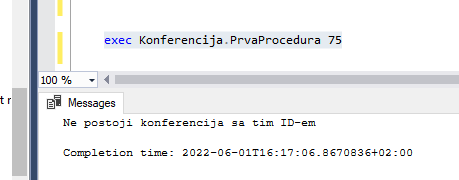
**PROCEDURE**

**Prva procedura** na osnovu prosleđenog ID-a konferencije izlistava njeno ime, početak konferencije, završetak konferencije. Ako ne postoji konferencija sa prosleđenim ID-em, ispisuje se odgovarajuća poruka. Procedura izlistava podatke o predavačima na sledeći način: Ime predavača, prezime, adresu, instituciju, datum rođenja i honorar.

Ako za prosleđenu konferenciju nema predavača, takođe ispisuje odgovarajuću poruku.

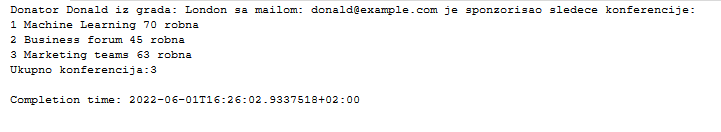


Slika 13: Prva procedura -uspešna

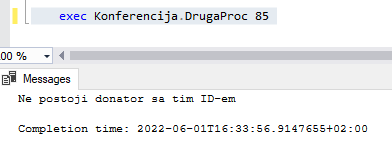


Slika 14: Prva procedura -neuspešna

**Druga procedura** na osnovu prosleđenog ID-a donatora izlistava njegovo ime i mail i grad odakle dolazi. Ako ne postoji donator sa prosleđenim ID-em, ispisuje se odgovarajuća poruka. Procedura izlistava podatke o konferencijama na sledeći način: Ime konferencije, broj organizatora, vrstu sponzorstva. Ako za prosleđen id donatora ne postoji konferencija na kojoj je donirao, ispisuje odgovarajuću poruku.



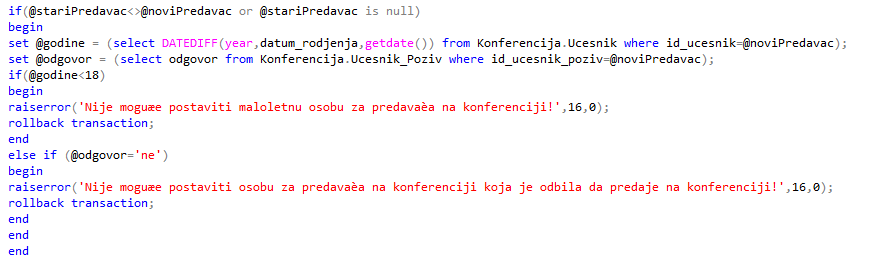
Slika 15: Druga procedura -uspešna



Slika 16: Druga procedura -neuspešna 1

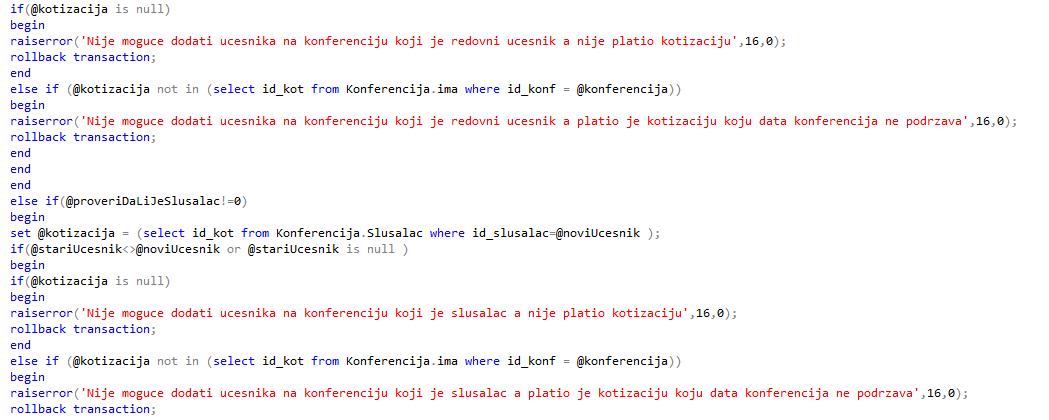
**TRIGERI**

**Prvi triger** je kreiran nad tabelom Predaje\_na i ima za cilj da implementira poslovno pravilo da se ne sme dozvoliti da predavač na konferenciji bude osoba koja je maloletna ili koja kao vrednost obeležja odgovor ima 'ne'.Kreirani triger je vrste AFTER, INSERT, UPDATE i okida se prilikom izmene nekog od gore spomenutih obeležja,ili prilikom unosa cele torke za radnika koji je maloletan, a koga želimo da predstavimo kao predavača, ubacujući njegov ID u tabelu Predaje\_na zajedno sa ID-em konferencije. U slučaju da je narušeno neko od ovih 2 pravila,triger će vratiti grešku sa adekvatnim porukama:



Slika 17: Prvi triger

**Drugi triger** je kreiran nad tabelom Ucestvuje i ima za cilj da implementira poslovno pravilo da se ne sme dozvoliti da se upiše učesnik koji nije platio kotizaciju, bilo da je reč o redovnom učesniku ili o slušaocu. Takođe, zabranjen je upis one osobe koja jeste platila kotizaciju, ali u valuti koju određena konferencija ne prihvata. Poslovni scenario je sledeći: U sistemu postoje konferencije koje kotizacije prihvataju isključivo u dinarima, dok je učesnik platio u evrima. Zbog poslovne logike, njegovo upisivanje u tabelu Učestvuje bi trebalo biti onemogućeno.Kreirani triger je vrste AFTER, INSERT, UPDATE. U slučaju da je narušeno neko od ovih 2 pravila,triger će vratiti grešku sa adekvatnim porukama:



Slika 18: Drugi triger

**ZAKLJUČAK**

Projekat koji je bio namenjen za predmet “Sistemi baza podataka”, predstavljao je krunu svega što smo radili iz oblasti baza podataka, računajući gradivo zimskog i letnjeg semestra.

Dok je sa jedne strane, projekat iz predmeta ‘’Projektovanje baza podataka’ bio timski orijentisan, na ovom projektu smo bili prepušteni sami sebi.

Bilo je potrebno dosta truda, zalaganja I vežbe da bi se projekat doveo do kraja, a da opet

predstavlja jednu kompletnu I logički ispravnu celinu.

Smatram da sam stekao značajno iskustvo i ovo će definitivno biti “odskočna daska” za sve buduće projekte, jer sam uvideo kako izgledaju sve faze ovakvog tipa projekta od same ideje do kasnije realizacije. Takođe, prilikom realizacije ovog projekta, uočio sam propuste koji su napravljeni u prethodnom semestru. Inženjerski pristup problemima, koji se zahtevao od nas, pomogao mi je da razmišljam “izvan kutije” i da donesem odluke koje su pomogle da zadatak uspešno privedem kraju.