

Praktični deo maturalnog ispita (Programiranje) – Proširenja baza podataka

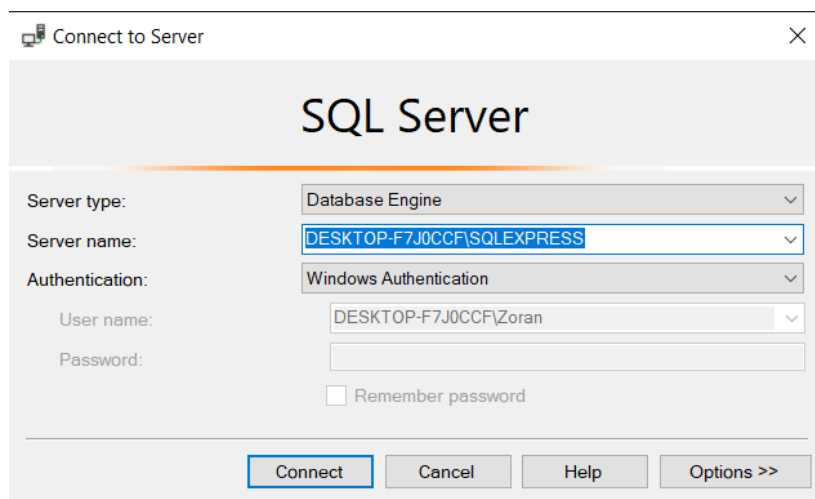
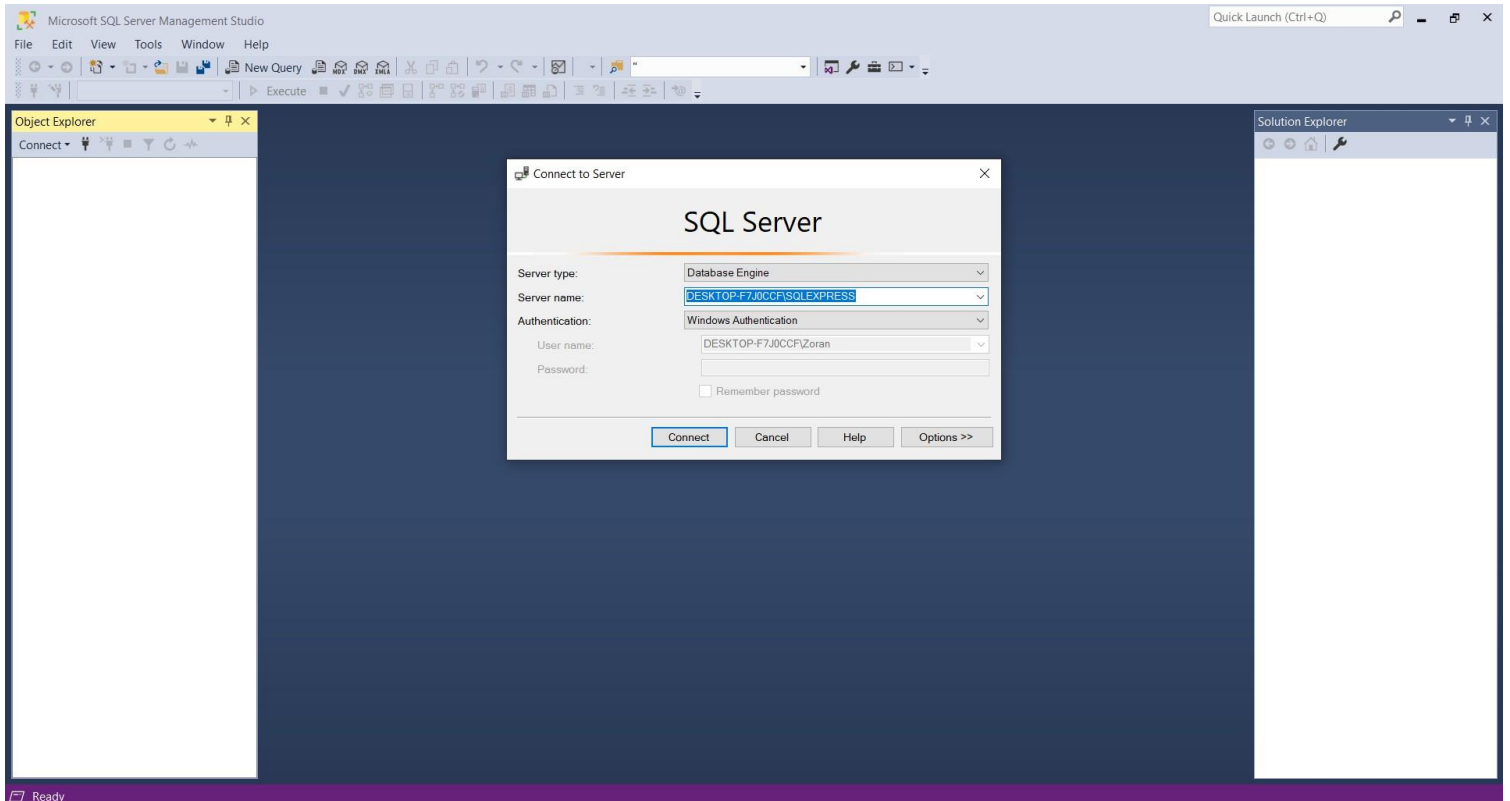
SQL Server Management Studio – početni koraci	2
CREATE TABLE	4
ALTER TABLE	4
CHECK CONSTRAINT	5
FOREIGN KEY	5
A1 – Biblioteka	6
A2 – Biblioteka	6
A3 – Evidencija radnika	7
A4 – Seoski turizam	7
A5 – Produženi boravak	8
A6 – Polovni automobili	8
A12 – Fudbalski stadioni	9
Kontrolna lista	10

SQL Server Management Studio – početni koraci

Pokretanje Microsoft SQL Server Management Studio okruženja:

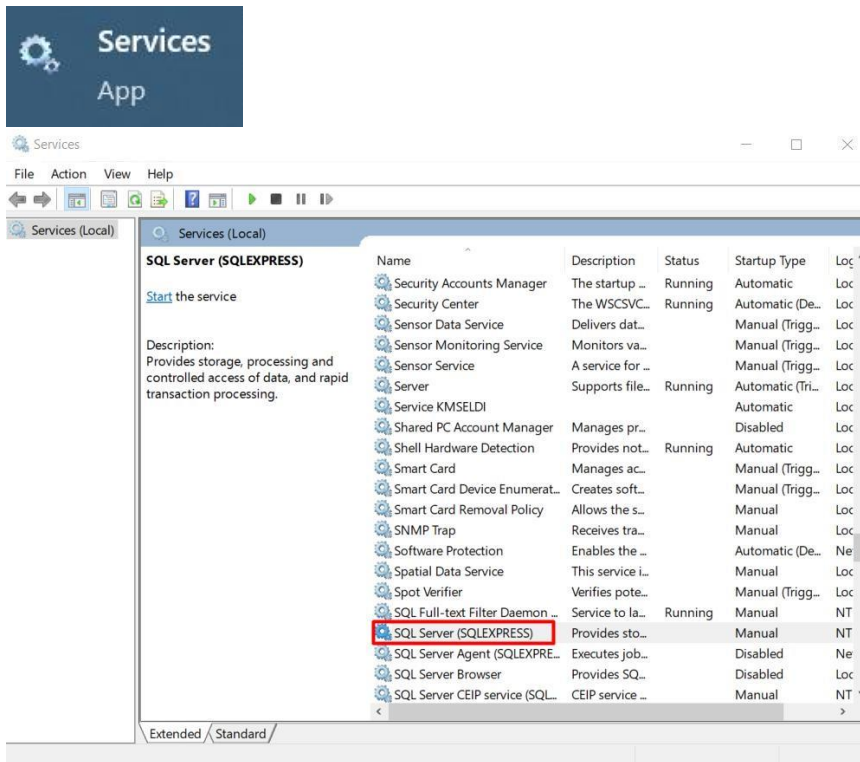
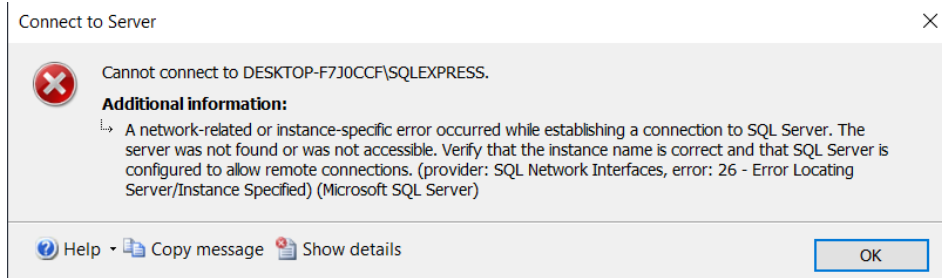


Nakon pokretanja programa dobija se radno okruženje i otvoren prozor *Connect to Server*:



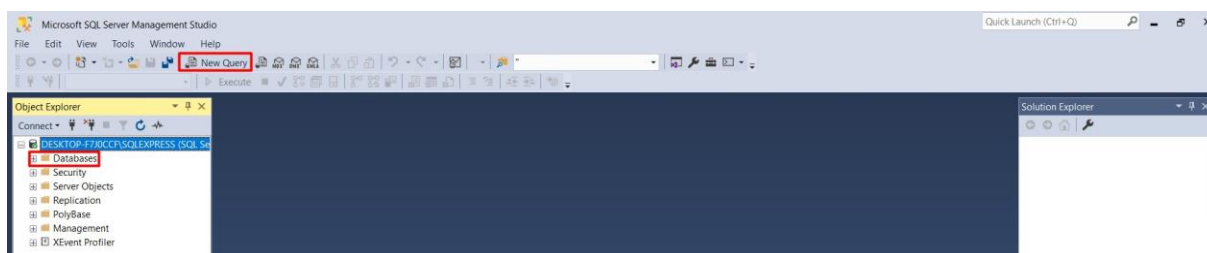
U prozoru *Connect to Server* potrebno je izabrati opciju *Connect*. Na mestu *Server name* će stojati ime_računara\SQLEXPRESS ako prilikom instalacije nije drugačije naznačeno.

Ukoliko prilikom konektovanja na server dođe do greške prikazane na slici ispod, to je znak da SQL Server nije pokrenut na računaru.



Da bi se SQL Server pokrenuo pristupa se servisima (*Services*) i pronalazi se servis *SQL Server (SQLEXPRESS)* i pokreće se desnim klikom na njega i izborom opcije *Start*.

Nakon pokretanja potrebnog servisa ponovo biramo opciju *Connect* u prozoru *Connect to Server* i dobijamo radno okruženje sa *Object Explorer*-om na levoj, *Solution Explorer*-om na desnoj strani i praznim mestom u sredini. Upite kreiramo izborom opcije *New Query* iz gornje trake sa opcijama (ili kombinacijom tastera *CTRL* i *N*). Unutar *Object Explorer*-a će se nalaziti sve baze podataka koje kreirate, zajedno sa njihovim tabelama, pogledima itd.



Nakon kreiranja novog upita možemo birati nad kojom bazom podataka želimo da ga izvršimo, izborom baze iz padajućeg menija u gornjoj traci sa alatima. Izborom opcije *Execute* izvršavamo kreirani upit.

Nakon uspešno kreiranih proširenja potrebno je sačuvati bekap (*backup*) baze podataka desnim klikom na željenu bazu u *Object Explorer*-u i izborom opcije *Tasks -> Back Up...*.

CREATE TABLE u SQL Serveru je SQL naredba koja se koristi za kreiranje nove tabele u bazi podataka. Tabela je osnovna struktura za čuvanje podataka u SQL Serveru, i obično se sastoji od kolona (ili polja) koje definišemo, a svaka kolona ima svoj tip podatka. Evo kako izgleda osnovna sintaksa CREATE TABLE naredbe:

```
CREATE TABLE ime_tabele
(
    kolona1_tip_podatka,
    kolona2_tip_podatka,
    ...
);
```

Evo jednostavnog primera CREATE TABLE naredbe:

```
CREATE TABLE Korisnici
(
    ID INT PRIMARY KEY,
    Ime NVARCHAR(50),
    Prezime NVARCHAR(50),
    Email NVARCHAR(100),
    DatumRodjenja DATE
);
```

U ovom primeru, kreiramo tabelu pod nazivom "Korisnici" sa pet kolona:

1. "ID" je kolona tipa INT i označena je kao PRIMARY KEY, što znači da će sadržati jedinstvene vrednosti i služiti kao jedinstveni identifikator za svakog korisnika.
2. "Ime" je kolona tipa NVARCHAR(50), koja čuva korisnikovo ime.
3. "Prezime" je kolona tipa NVARCHAR(50), koja čuva korisnikovo prezime.
4. "Email" je kolona tipa NVARCHAR(100), koja čuva korisnikovu email adresu.
5. "DatumRodjenja" je kolona tipa DATE, koja čuva datum rođenja korisnika.

Nakon što izvršimo ovu CREATE TABLE naredbu, SQL Server će kreirati tabelu "Korisnici" u bazi podataka sa definisanim kolonama i njihovim tipovima podataka. Ova tabela će biti spremna za unos podataka i izvršavanje različitih SQL upita nad njom.

ALTER TABLE u SQL Serveru je SQL naredba koja se koristi za izmenu postojeće tabele u bazi podataka. Ova naredba vam omogućava da dodate, izbrišete ili promenite kolone u tabeli, kao i da izvršite druge strukturne promene u tabeli.

Dodavanje nove kolone:

```
ALTER TABLE ime_tabele
ADD nova_kolona tip_podatka;
```

Na primer:

```
ALTER TABLE Korisnici
ADD Telefon NVARCHAR(20);
```

Ova naredba će dodati novu kolonu "Telefon" tipa NVARCHAR(20) u tabelu "Korisnici".

Kada koristite ALTER TABLE naredbu, budite pažljivi jer promene u strukturi tabele mogu uticati na postojeće podatke i aplikacije koje koriste tu tabelu.

U SQL Serveru, **CONSTRAINT** (ograničenje) je pravilo ili uslov koji se primenjuje na tabelu kako bi se osigurala tačnost, integritet i konzistentnost podataka u bazi. Ograničenja se koriste za definisanje pravila za vrednosti koje se mogu uneti u kolone tabele.

CHECK CONSTRAINT (ograničenje provere) u SQL Serveru je vrsta ograničenja koja se koristi da biste definisali uslove ili izraze koje moraju ispunjavati vrednosti koje se unose u određenu kolonu tabele. Ovo ograničenje osigurava da podaci u toj koloni zadovoljavaju određeni set pravila ili uslova, čime se održava integritet podataka. Ako vrednost koja se unosi ne ispunjava uslov definisan CHECK ograničenjem, SQL Server će sprečiti unos tog reda ili promenu te vrednosti.

Evo kako se definiše CHECK ograničenje u SQL Serveru:

```
CREATE TABLE Imenik
(
    ID INT PRIMARY KEY,
    Ime NVARCHAR(50),
    Prezime NVARCHAR(50),
    BrojTelefona NVARCHAR(20),
    Starost INT CHECK (Starost >= 18)
);
```

U ovom primeru, kreiramo tabelu "Imenik" sa kolonom "Starost" koja ima CHECK ograničenje. Ograničenje definiše uslov da vrednost u koloni "Starost" mora biti veća ili jednaka 18. To znači da se u tabelu "Imenik" mogu dodavati samo redovi u kojima je vrednost kolone "Starost" veća ili jednaka 18. Ako pokušate dodati red sa vrednošću "Starost" manjom od 18, SQL Server će generisati grešku i sprečiti unos tog reda.

CHECK ograničenja su korisna za sprovođenje poslovne logike i osiguravanje tačnosti i konzistentnosti podataka u tabelama. Možete ih koristiti za razne provere, kao što su provere datuma, raspona vrednosti, formatiranja teksta i slično, kako biste osigurali da se u tabelu unose samo validni podaci.

Ukoliko je potrebno da izvršimo CHECK ograničenje sa upoređivanjem dva polja tabele u kojoj dodajemo ograničenje potrebno je da prvo napravimo ta polja pa da zatim preko ALTER TABLE ubacimo CHECK ograničenje na sledeći način:

```
ALTER TABLE NazivTabele
```

```
ADD CONSTRAINT NazivOgraničenja CHECK (Uslov);
```

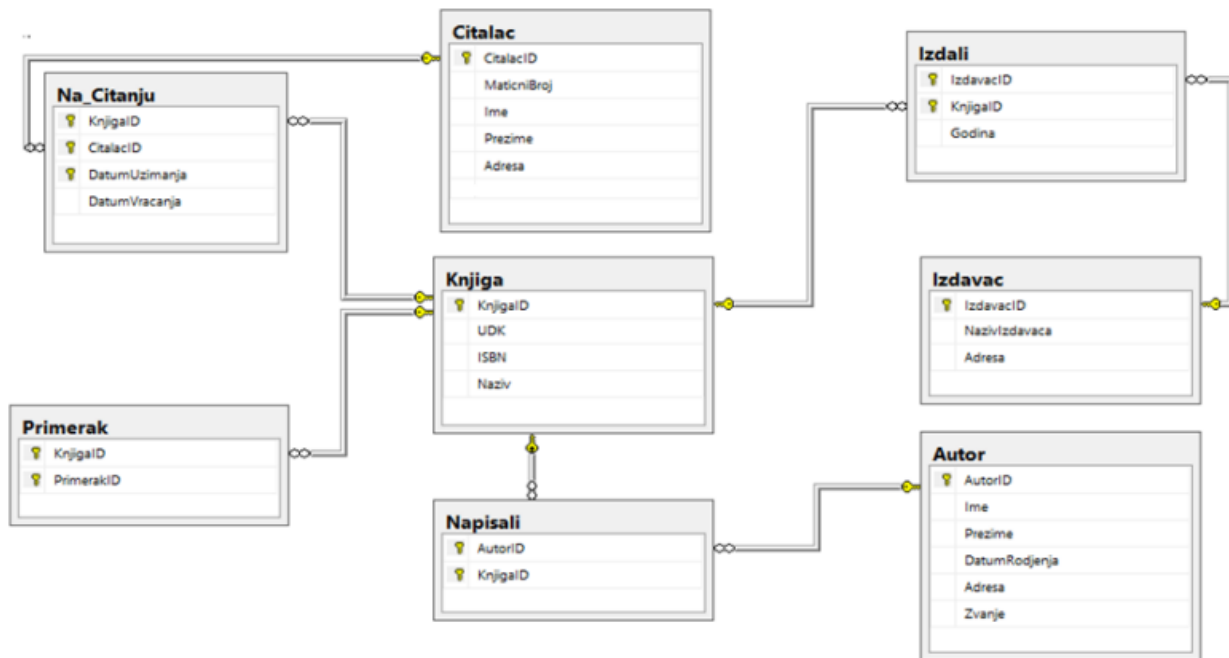
FOREIGN KEY Strani ključ (*foreign key*) odnosno polje preko kog povezujemo dve tabele se dodaje na sledeći način:

```
NazivPolja TipPodataka FOREIGN KEY REFERENCES NazivTabeleSkojomPovezujemo
(NazivPoljaSkojimSePovezujemo)
```

Na primer:

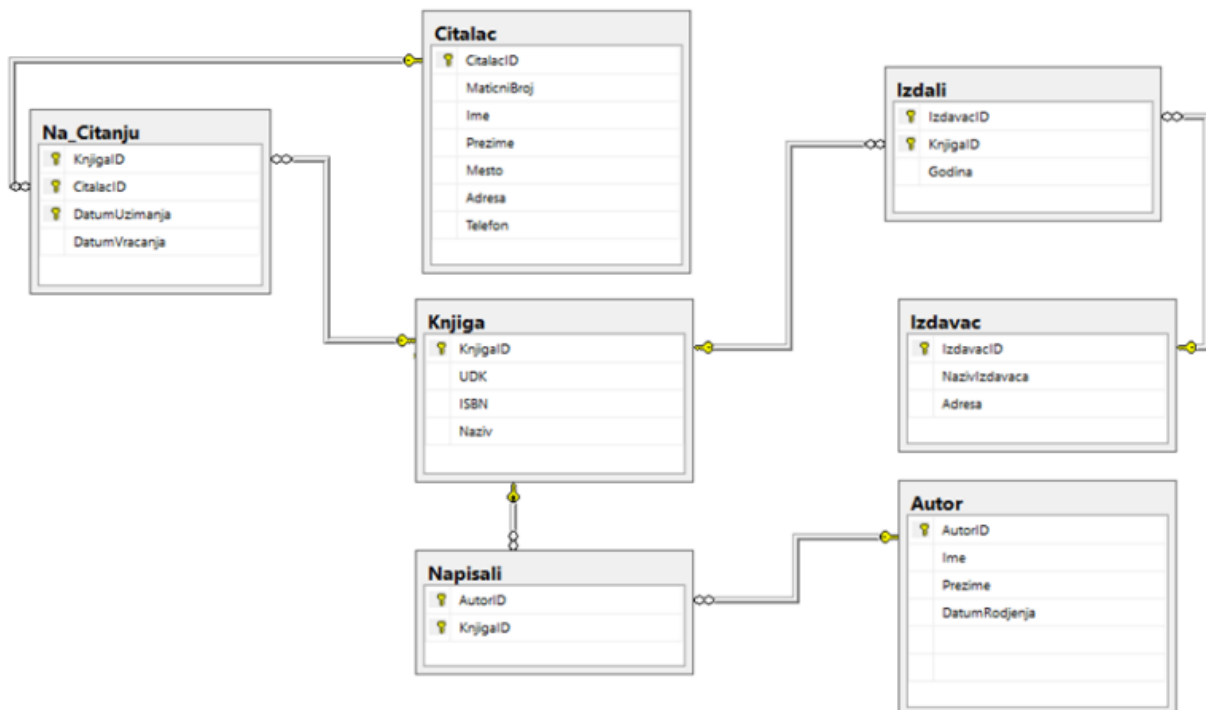
```
KnjigaID INT FOREIGN KEY REFERENCES Knjiga(KnjigaID)
```

A1 – Biblioteka



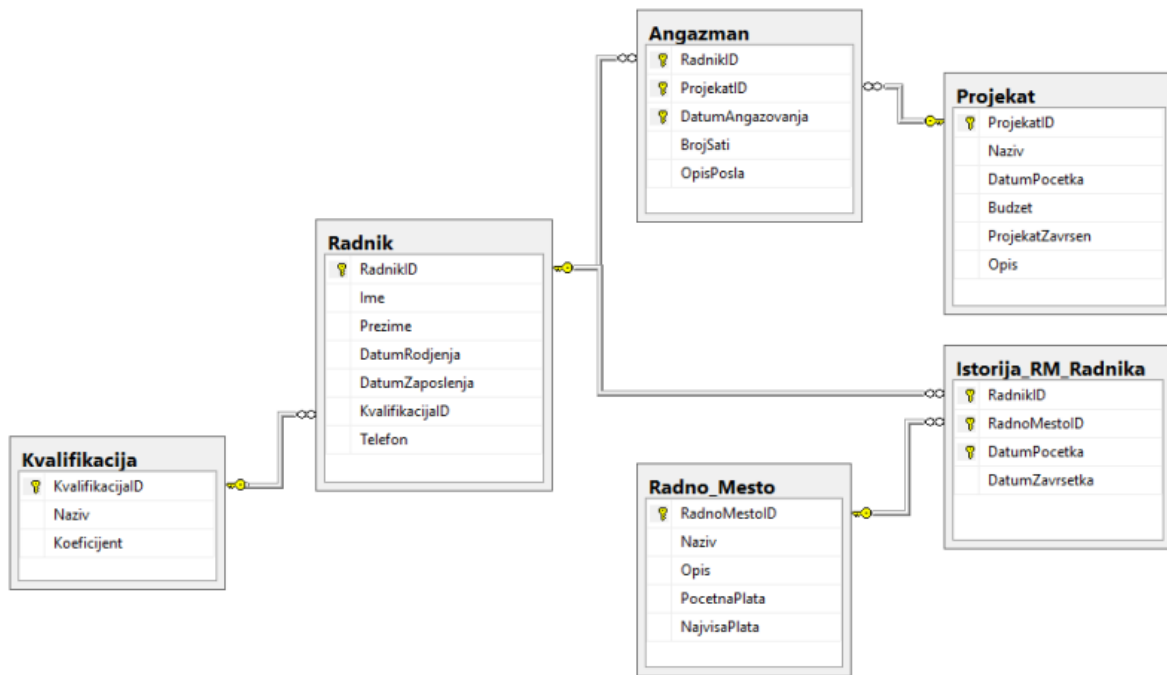
Biblioteka otvara veći broj ogranaka u školama u okolini i potrebno je čuvati podatke i o školama u kojima su otvoreni ogranci biblioteke. Za svaki primerak knjige treba pamtiti u kom ogranku se nalazi. Takođe je potrebno pamtiti i kada je knjiga nabavljena. Obezbediti kroz ograničenja u bazi da datum nabavke ne može da bude stariji od datuma osnivanja biblioteke (01.01.2000.).

A2 – Biblioteka



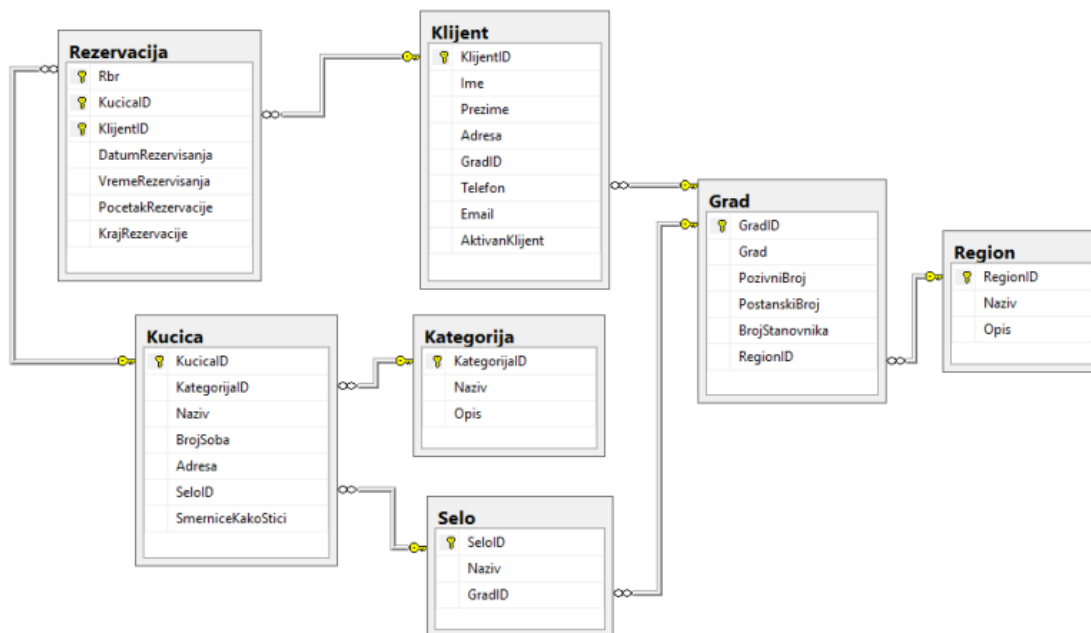
Osmisliti proširenje baze kojim se za svaki primerak knjige omogućava pamćenje formata u kome je knjiga izdata. Format može da bude tvrdi povez, meki povez, ilustrovano izdanje... Proširiti bazu tako da se za čitaoca pamti datum poslednjeg plaćanja članarine i ograničiti da se članarina ne može plaćati retroaktivno (datum plaćanja nije stariji od tekućeg datuma).

A3 – Evidencija radnika



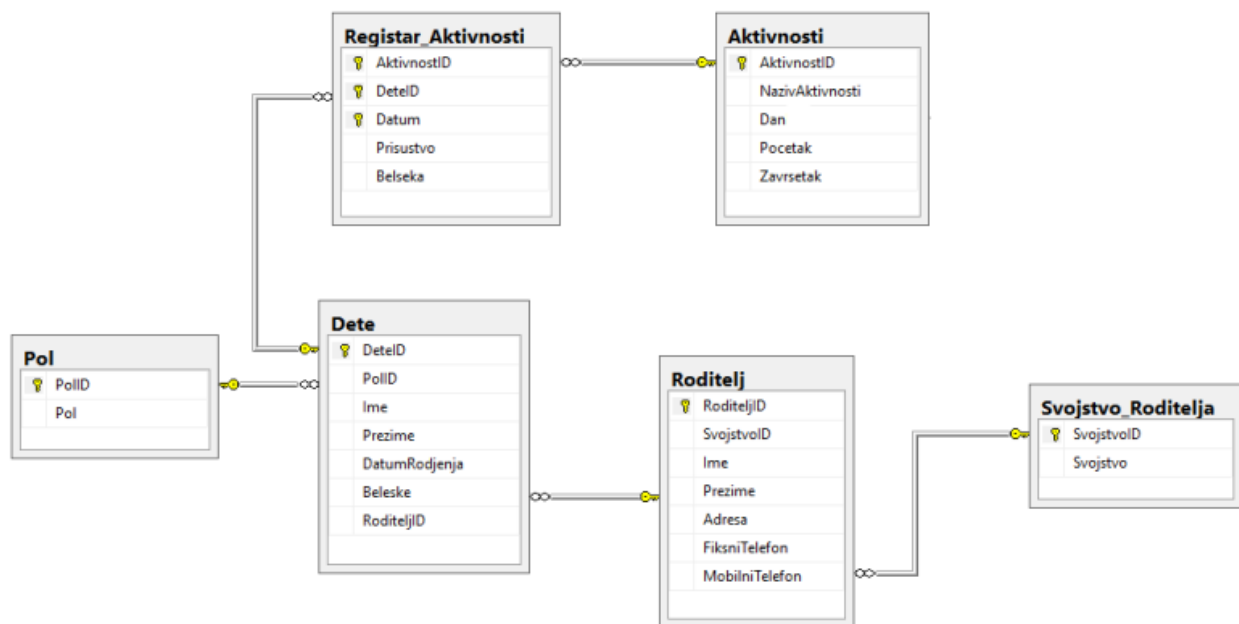
Osmisliti proširenje baze kojim se omogućava čuvanje podataka o firmama-podizvođačima angažovanim na projektu. Zbog različite prirode poslova, pored sopstvenih zaposlenih, klijent-nosilac projekta može da angažuje jednog ili više podizvođača. Za svako angažovanje treba pamtit od kog do kog datuma je podizvođač angažovan na projektu. Pri tome, direktno u bazi obezbediti da datum početka bude manji od datuma završetka projekta.

A4 – Seoski turizam



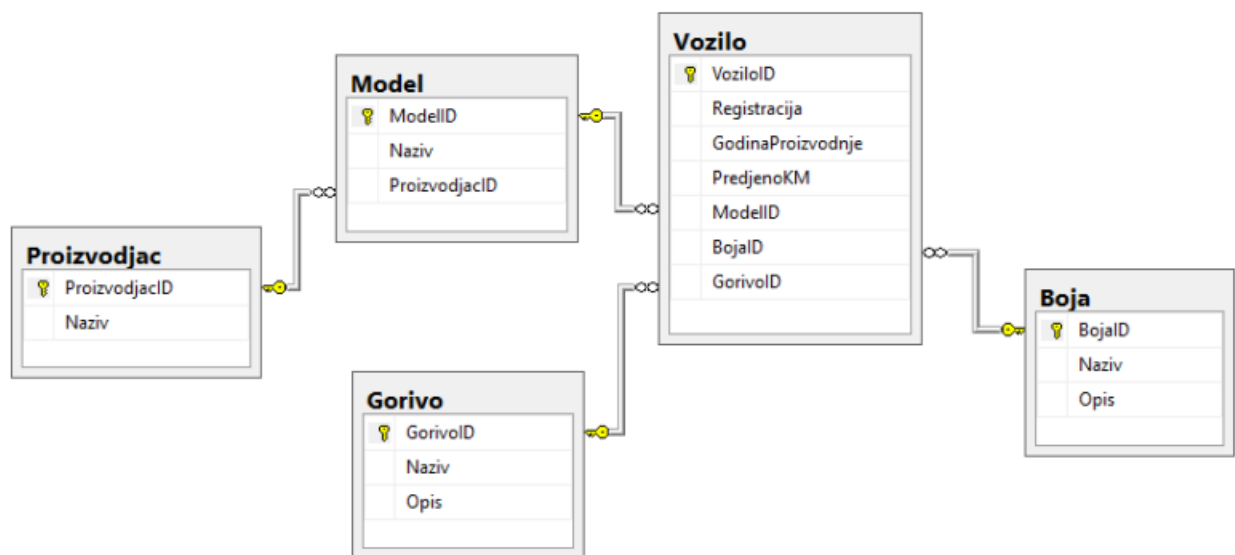
Osmisliti proširenje baze kojim se omogućava čuvanje podataka o dodatnim ponudama koje su na raspolaganju gostu (internet, TV, klima, bazen, spa...). Jedna kućica može nuditi veći broj dodatnih usluga i obrnuto, jedan tip usluge može biti na raspolaganju u većem broju objekata. Potrebno je pamtit i u kom opsegu se kreće cena usluge u konkretnom objektu (kroz dizajn baze obezbediti ispravan unos opsega cena, tj. iznos od < iznos do).

A5 – Produženi boravak



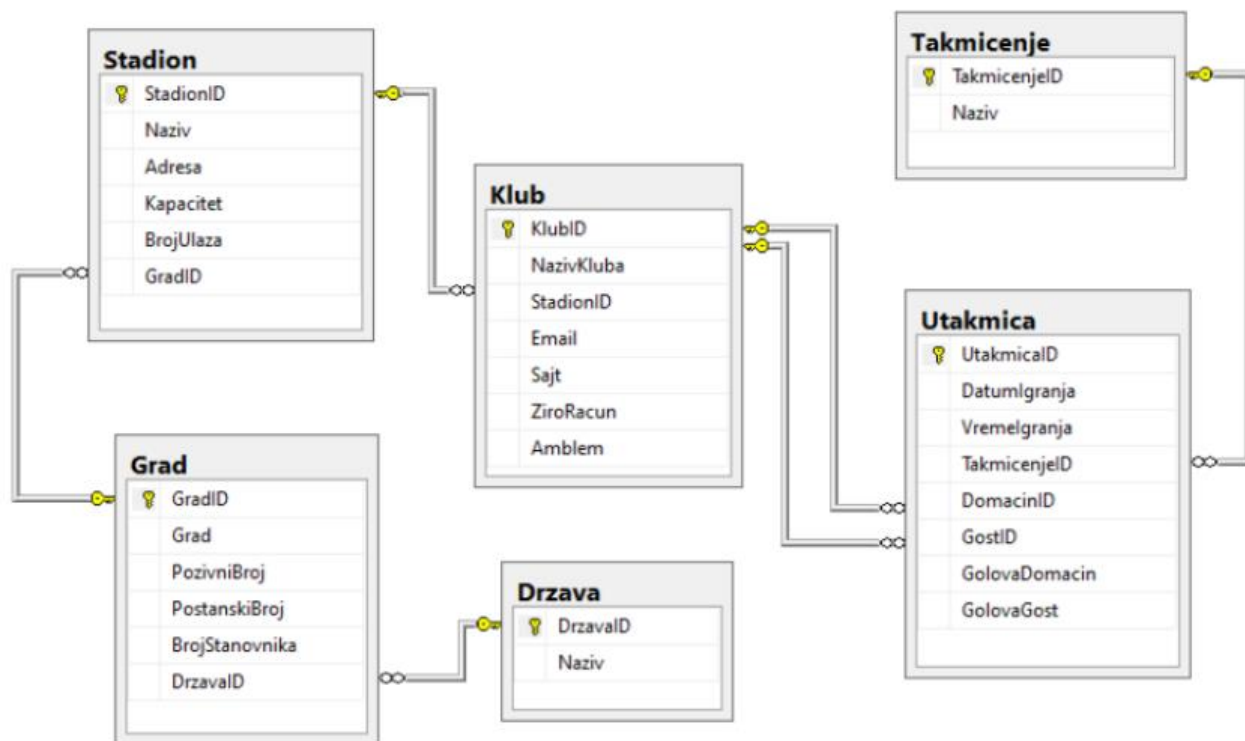
Potrebno je da se u bazi čuvaju i podaci koji se tiču nastavnika-vaspitača odgovornih za koordinaciju organizacijom aktivnosti. Za svaku aktivnost treba pamtit i ko od vaspitača je koordinator za tu aktivnost. Takođe je potrebno pamtit i datum prijema deteta u produženi boravak (obezbediti da datum prijema ne može da bude stariji od datuma rođenja).

A6 – Polovni automobili



Zbog bavljenja posredovanjem u prodaji, u bazu je potrebno dodati mogućnost da se čuvaju podaci o trenutnim vlasnicima automobila (ime, prezime, telefon, adresa, da li se radi o prvom vlasniku). Svaki automobil može da ima samo jednog vlasnika. Takođe je potrebno dodati mogućnost pamćenja početne ponuđene cene za vozilo, sa tim što treba obezbediti da se ne može uneti nelogična vrednost (cena ne može biti manja od nule).

A12 – Fudbalski stadioni



Potrebno je bazu proširiti mogućnošću pamćenja igrača. Za igrača treba pamtit i ime, prezime, datum rođenja i iz koje zemlje je poreklom. Igrač u različitim periodima može da igra za različite klubove, pa se zato pamti datum kada je počeo i datum do kada je igrao za neki tim. Prvi datum je obavezan, a drugi je prazan ukoliko igrač još uvek igra za taj tim. U samoj bazi treba obezbediti da drugi datum ili nije postavljen ili je manji od prvog datuma.

Kontrolna lista

Analiza zahteva

- Koji novi entiteti (tabele) su potrebni?
- Koji novi atributi treba da se dodaju postojećim tabelama?

Dizajn novih tabela

- Da li sam dodao primarni ključ (PRIMARY KEY)?
- Da li su nazivi kolona i tipovi podataka jasni?

Povezivanje tabela

- Da li sam dodao strani ključ (FOREIGN KEY) kada povezujem nove i stare tabele?

Ograničenja i pravila

- Da li sam dodao potrebna ograničenja (CHECK, NOT NULL, UNIQUE)?

Izmene postojećih tabela

- Da li sam ALTER TABLE koristio gde treba (dodavanje kolona)?
- Da li je ime stranog ključa i ograničenja smisljeno?

Testiranje

- Da li mogu da unesem validan podatak?
- Da li baza odbija pogrešan podatak?

Tipičan obrazac (koraci):

CREATE TABLE (ako uvodim potpuno novu tabelu)

ALTER TABLE ADD (novi atributi)

FOREIGN KEY (veze)

CHECK (ograničenja)

Test unos podataka