

ŠOLSKI CENTER VELENJE
POKLICNA IN TEHNIŠKA ELEKTRO IN RAČUNALNIŠKA ŠOLA

KNJIŽNICA

seminarska naloga

Alenka Močilnik

Velenje, maj 2000

OPIS NALOGE

Implementiraj program, ki omogoča vodenje evidence o članih, knjigah, izposoji in rezervacijah. Program naj omogoča tudi izpis opominov in raznih statističnih pregledov.

KNJIŽNICA

Alenka Močilnik

Mentor: ga. inž. Alenka Pačnik-Gabrovec

POVZETEK

Seminarska naloga obsega evidence o članih in knjigah, izposoji, rezervacijah, izpisujemo lahko opomine, različne statistične preglede (po različnih kriterijih). Seminarska naloga je narejena v okolju Delphi2. Delo sem si olajšala s programom za načrtovanje informacijskih sistemov PowerDesigner. V seminarški nalogi so na kratko opisane baze, orodja za hitro programiranje, gradnja podatkovne baze za knjižnico in generiranje Delphijsve aplikacije. Na koncu so podane tudi možne izboljšave.

ABSTRACT

I developed an application for libraries. It stores data about members, books, reservations of books etc. We can print reminders and number of other statistics. The application is written in Delphi, version 2. I used an application PowerDesigner, which makes the development of an application much easier due to the fact that it is a powerful tool for information system planning. In seminar work I described philosophy of database, RAD (Rapid Application Development) tools, work I done with PowerDesigner (logical (conceptual) model, physical model and generation of my application, respectively), appendix to generated application and short user manual. I conclude with a list of possible upgrades.

Kazalo

Opis naloge

Povzetek

Abstract

1. Uvod

- Kaj so baze?
- Zakaj so baze koristne?
- O orodjih za hitro programiranje(RAD-Raid Application Development)
- Zakaj Delphi?
- Struktura seminarske naloge

2. Programski paket POWER DESIGNER

2.1 Opis programskega paketa

2.2 Gradnja podatkovne baze za knjižnico

- Logični podatkovni model baze
- Fizični podatkovni model baze
- Avtomatsko ustvarjanje baze iz fizičnega modela
- Ustvarjanje pogledov oz. poizvedb
- Generiranje Delphijske aplikacije

3. Dodatki programu

- Kaj je bilo treba v samem Delphiju še dodati, da smo ugodili zahtevam naloge?

4. Program KnjiznicaPlus

- O uporabi programa

5. Zaključek in nadaljne delo

DODATEK A Skripta za kreiranje baze

Literatura

Zahvala

1.

UVOD

Glavni namen programa je, da bi olajšala delo uporabnikom. Zaradi prevelikega števila knjig in članov, je postalo zapisovanje podatkov v beležke prenaporno in časovno potratno, saj je iz dneva v dan več članov in knjig. Pogosto se je zgodilo, da so se podatki izgubili in prihajalo je do napak. Da bi si olajšali delo, smo za tako obširne in pomembne podatke našli rešitev. Znanost je tako napredovala, da nam ni potrebno več zapisovati podatkov na papir, ampak jih preprosto shranimo v računalnik. Odločila sem se, da naredim seminarsko nalogo, ki bo uporabna v vseh knjižnicah. Uporaba programa je enostavna in prijazna.

Program je bil razvit na osebem računalniku PC PII 300 MHz in sicer pod operacijskim sistemom Windows98. Program deluje tudi na šibkejšem računalniku, vendar le pod 32-bitnim operacijskim sistemom (Windows95 ali višja različica). Program izrablja podatkovno strukturo baze programa Borland Paradox.

Kaj je podatkovna baza?

Podatki omogočajo posamezniku ali organizaciji shraniti na pomnilniški medij tisti del svojega znanja, ki se nanaša na pomnenje dejstev. Iz takega razumevanja podatkov sledi definicija podatkovne baze:

- 1) *Podatkovna baza je model okolja, ki služi kot osnova za sprejemanje odločitev in izvajanje akcij*
- 2) *Podatkovna baza je zbirka med seboj pomensko povezanih podatkov, ki so shranjeni v računalniškem sistemu, dostop do njih je centraliziran in omogočen s pomočjo sistema za upravljanje podatkovnih baz.*
- 3) *Podatkovna baza je mehanizirana, večuporabniška, formalno definirana in centralno nadzorovana zbirka podatkov.*

V podatkovni bazi se shranjujejo podatki, ki se nanašajo na določen del sveta, v katerem so vse stvari in dogajanja med seboj povezana.

Razen podatkovnih baz za shranjevanje formalnih podatkov obstajajo tudi druge vrste podatkovnih sistemov kot npr. referenčni sistemi, sistem za obdelavo besedil, bibliografski sistemi, specializirane podatkovne baze, baze znanja in objektne podatkovne baze.

Podatkovno bazo sestavljajo podatki, uporabniki in uporabniški programi, upravitelj podatkovne baze in SUPB. Podatkovni del PB sestavljata **fizična podatkovna baza** in **metapodatkovna baza**.

Zakaj so podatkovne baze koristne?

- Podatkovna baza mora biti **sočasno** na voljo vsem uporabnikom, pri čemer sočasna raba ne sme ogroziti njene celovitosti. Uporabnikom morajo biti dostopne **vse vrste podatkov**, zato mora biti v okviru upravljanja zagotovljeno zbiranje podatkov, ki se nanašajo na samo organizacijo in njeno okolje. Uporabnikom naj bodo na voljo ne samo podatki, ki govorijo o tem, kaj se je zgodilo (zgodovina), pač pa tudi podatki o tem, kaj se utegne zgoditi (napovedi).
- Upravljanje podatkovne baze mora zagotoviti učinkovit dostop do podatkov vsem vrstam uporabnikov:
 - uporabniškimi programi za rutinske, ponavljajoče se obdelave,
 - parametričnim uporabnikom za transakcijsko delo,
 - občasnim uporabnikom za bolj ali manj enkratna povpraševanja;
 - programerjem za razvoj uporabniških programov,
 - upravitelju podatkovne baze za nadzor nad njo.

Uporabniki podatkovne baze

Med uporabnike podatkovne baze štejemo vse, ki kakorkoli uporabljajo njene zmogljivosti – bodisi da rokujejo s podatki, bodisi da uporabljajo funkcije SUPB za razvoj programov ali upravljanje podatkovne baze. Delimo jih na **neposredne** in na **posredne** uporabnike. Neposredni uporabniki komunicirajo s podatkovno bazo preko terminalov računalniškega sistema, posredni uporabniki pa le okvirno podajajo svoje podatkovne zahteve, njihovo izvedbo pa prepuščajo neposrednim uporabnikom.

O orodjih za hitro programiranje (RAD – Rapid Application Development)

Jezik Delphija je objektna različica Turbo Pascala. Delphi v sebi združuje tudi Borlandovo tehnologijo upravljanja podatkovnih baz in hiter, vizualen razvoj aplikacij, poimenovan RAD.

Zakaj Delphi?

Njegova največja prednost je v načinu ustvarjanja delujočih aplikacij. Delphi se navzven kot univerzalno orodje z mnogo obrazi. Delphi je visoko zmogljivo okolje za programiranje pod Okni. Delphi nam omogoča, da okenskih aplikacij ne pišemo z ničle, ampak nam pomaga na dva načina: vsebuje že pripravljene vizualne gradnike, ki jih povežemo v delujočo aplikacijo. Predmetna ustvarjalnost nam omogoča, da napišemo lastne gradnike. Za razliko od mnogih vizualnih orodij, Delphi celotno aplikacijo zapeče v samo eno, samostojno izvršilno datoteko s podaljškom .EXE. Poleg tega lahko z Delphijem ustvarjamo tudi deljene knjižnice DLL.

Struktura seminarske naloge

V drugem poglavju je opisan programski paket Power Designer. To je paket, ki omogoča načrtovanje informacijskih sistemov ter generiranje aplikacije na podlagi narejenega načrta. Opisan je celoten postopek dela s programom.

V naslednjem poglavju sem opisala tiste dele Delphijske kode, ki so bili dodani ročno.

V četrtem poglavju sem opisala uporabo nastalega programa.

V zadnjem poglavju so podane možne izboljšave.

2.

Programski paket

POWER DESIGNER

2.1 Opis programskega paketa

V programsko skupino **Power Designer** spadajo naslednja orodja:

- Data Architect
Omogoča logično in fizično načrtovanje podatkovne baze
- Process Analyst
Funkcionalno generiranje aplikacije
- Application Modeler
Automatično generiranje aplikacije
- Warehouse Architect
Načrtovanje podatkovnih skladišč
- Power Designer ODBC
Open Data Base Connectivity gonilniki za povezavo z SUPB

2.2 Gradnja podatkovne baze za knjižnico

- Logičen podatkovni model baze

Logični oz. konceptualni nivo predstavlja celotno logično strukturo podatkovne baze. Vsebuje informacijo o vsebini podatkovne baze. Konceptualni nivo je neodvisen od posamezne podatkovne baze.

DEFINIRANJE PODATKOVNIH ATRIBUTOV

- Sestavljanje konceptualnega modela začnemo z definiranjem poslovnih pravil, domen in podatkovnih atributov, še prej pa definiramo poslovne aktivnosti na osnovi katerih določimo entitete, attribute entitet ter poslovna pravila.

- **Podatkovni atribut** oz. **atribut** je elementarni del informacije. **Domena** je seznam uporabniških specifikacij, ki jih lahko definiramo za nek atribut. Če spremenimo domeno, se spremenijo podatkovni tipi vseh atributov vezanih na to domeno. Domenam lahko določimo tudi parametre kot so: minimum, maksimum, privzeta vrednost,...
- **Entiteta** predstavlja objekt o katerem shranjujemo informacije.

DEFINIRANJE RELACIJ

Lastnosti relacij:

- števnost (cardinality: one-many,...)
- obveznost (existence: optional / mandatory)
- odvisnost (dependency: zapis entitete je enolično definiran šele s pomočjo ključa iz druge entitete) ((zunanji ključ))

Data architect

Data architect je močno orodje, ki nam omogoča projektiranje podatkovne baze s tako imenovanim dvo-nivojskim pristopom:

- CDM konceptualni podatkovni nivo – logični model
- PDM fizični podatkovni nivo – fizični model

S pomočjo Data Architect-a lahko:

- modeliramo informacijski sistem z entitetnim relacijskim diagramom imenovanim konceptualni podatkovni model,
- generiramo fizični podatkovni model na osnovi konceptualnega podatkovnega modela, pri čemer izberemo ciljno podatkovno bazo, oz. DBMS,
- popravljamo podatkovni nivo na fizičnem nivoju,
- generiramo skripto za generiranje podatkovne baze za določen DBMS,
- generiramo triggerje, ki omogočajo referenčno integriteto, če to ciljni DBMS podpira,
- urejamo ter tiskamo poročila o modelu,
- uporabljamo Reverse engineering
- definiramo dodatne attribute za objekte na fizičnem nivoju

KONCEPTUALNI NIVO

Konceptualni nivo predstavlja celotno loično strukturo podatkovne baze. Vsebuje informacijo o vsebini podatkovne baze. Konceptualni nivo je neodvisen od posamezne podatkovne baze. **Rezultat je logični model.**

Sestavljanje konceptualnega modela začnemo z definiranjem poslovnih pravil, domen

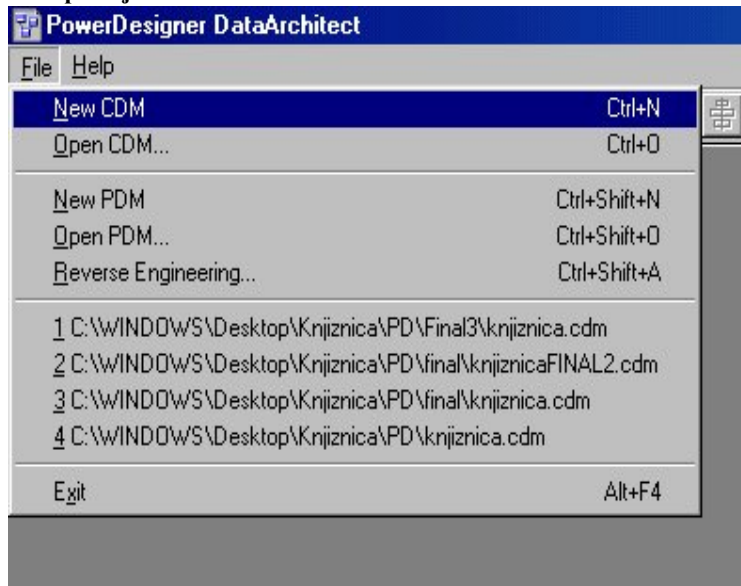
in podatkovnih atributov, še prej pa definiramo poslovne aktivnosti na osnovi katerih določimo entitete, attribute entitet ter poslovna pravila.

PODATKOVNI NIVO

Fizični podatkovni model predstavlja podatkovno strukturo v taki obliki, kot bo bila implementirana v podatkovni bazi. Kreiramo ga za določen DBMS. Data Architect prevede vse podatkovne tipe iz konceptualnega modela v podatkovne tipe, ki jih podpira ciljna podatkovna baza. Povezava med konceptualnimi in fizičnimi podatkovnimi tipi je definirana v DEF datotekah za vsak DBMS posebej in jo lahko tudi sami spreminjamo.

Postopek ustvarjanja logičnega modela

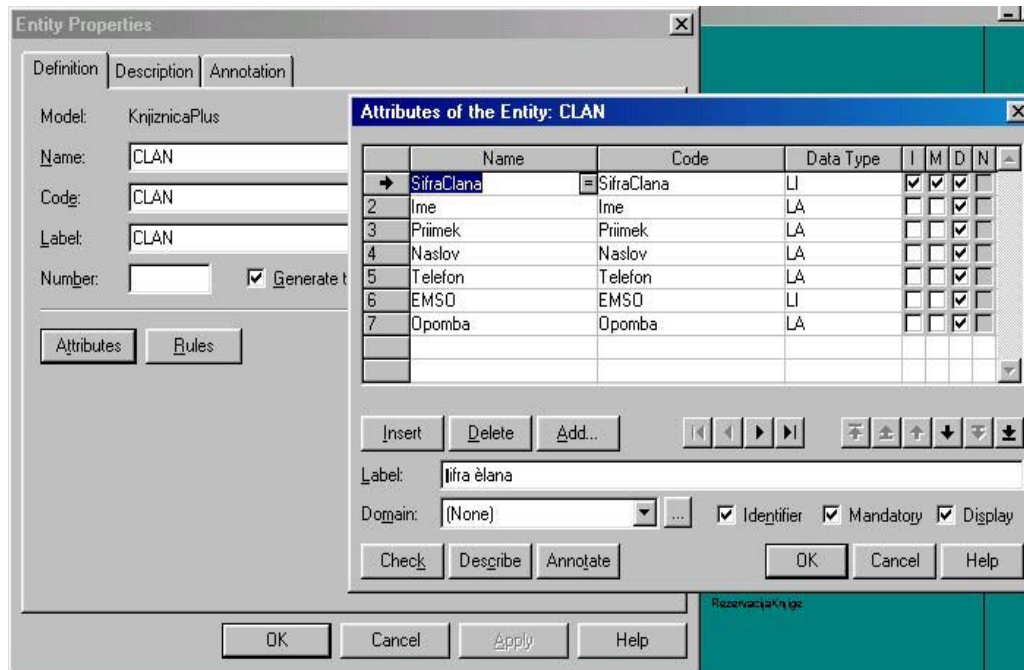
1. Odpiranje



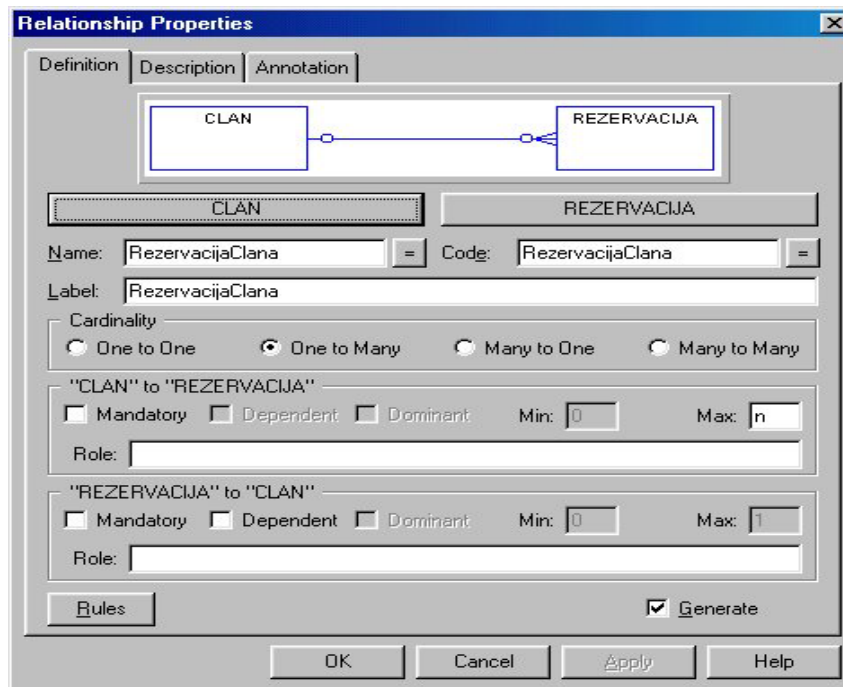
2. Orodna vrstica s katero zgeneriramo povezave



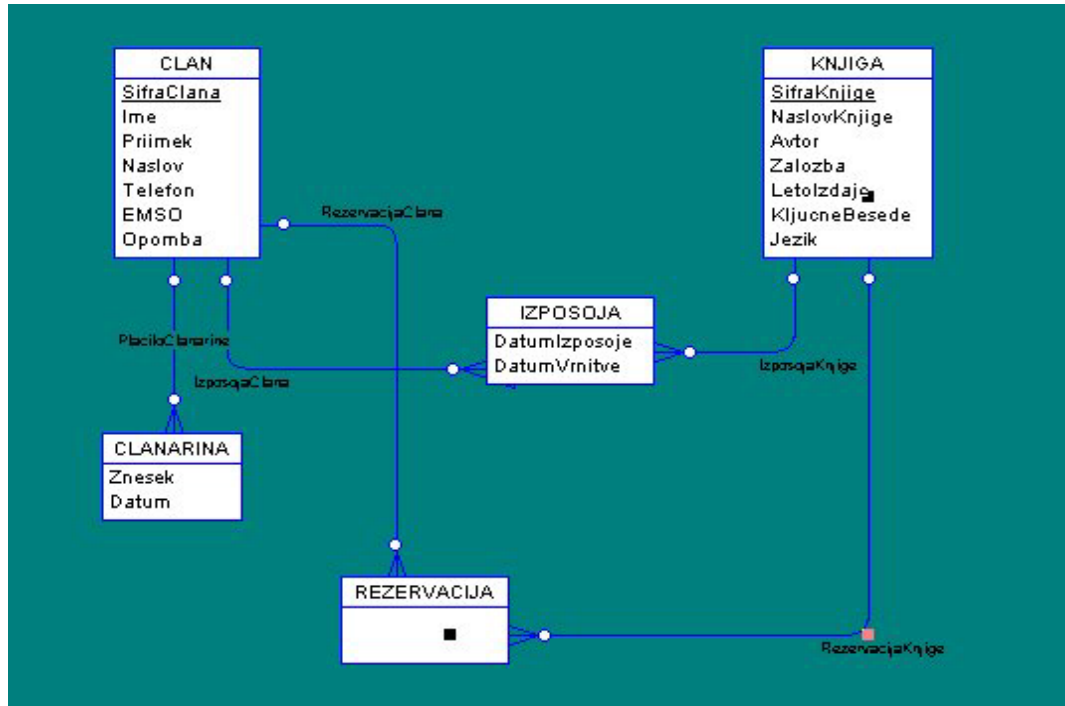
3. Kako opišemo ime in attribute (osnovno delovanje Power Designerja)



4. Kako določimo relacije (logična shema)

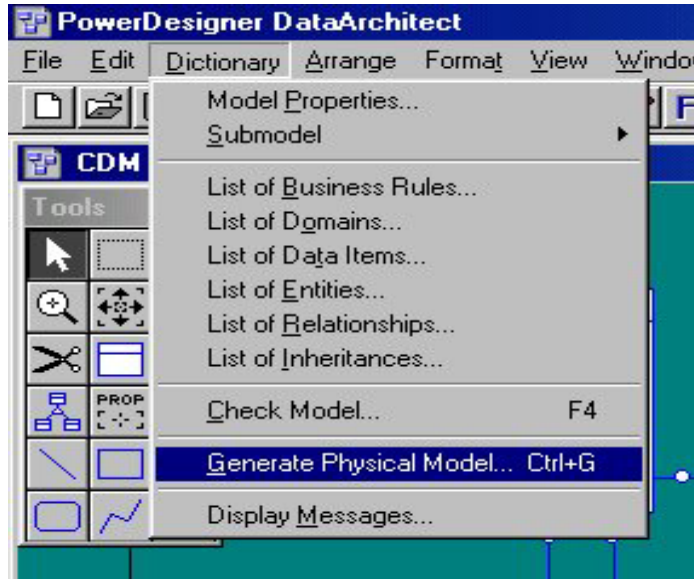


5. Slika logičnega modela

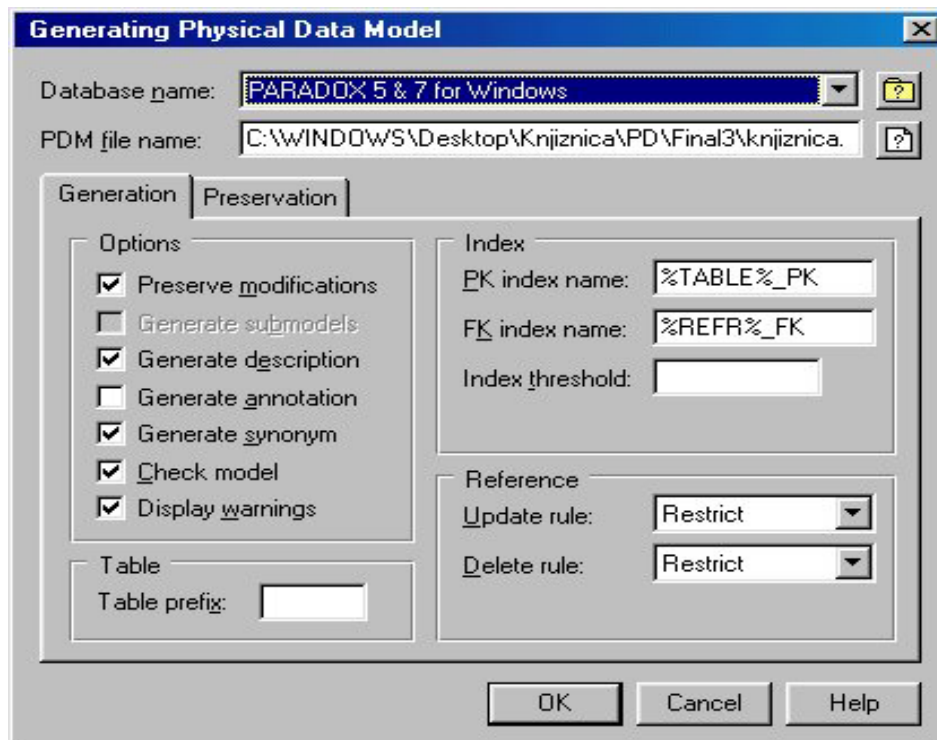


- Fizični podatkovni model baze

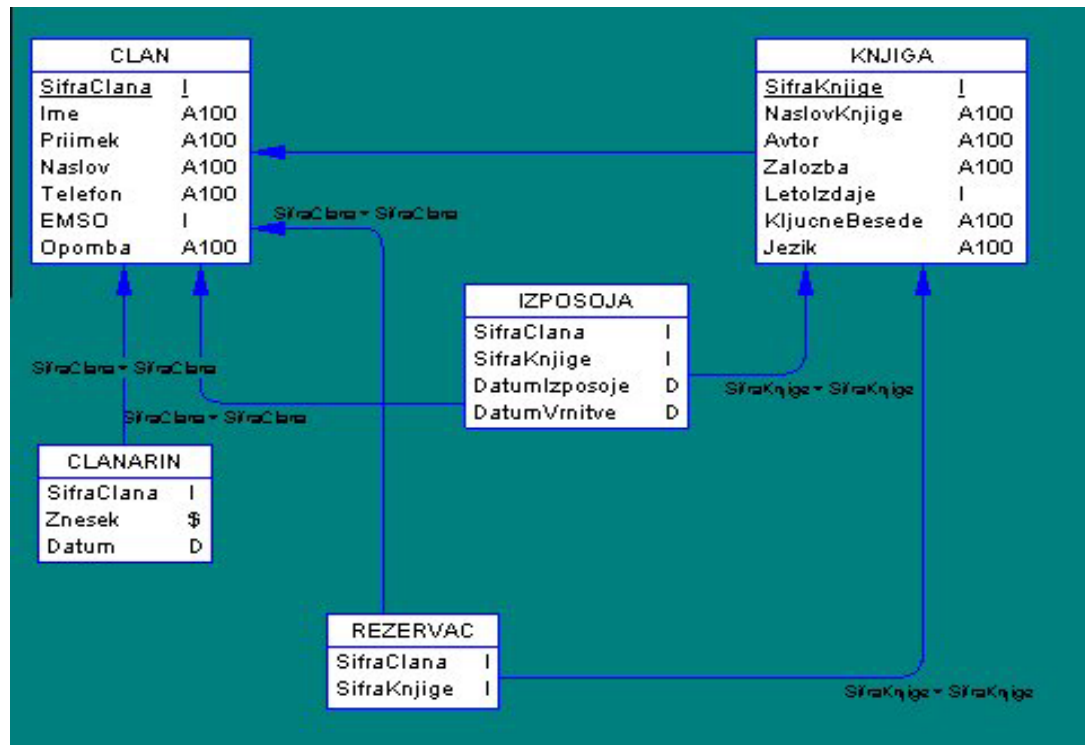
6. Generiranje fizičnega modela



7. Izbira baze (kako izbereš bazo)



8. Fizični model

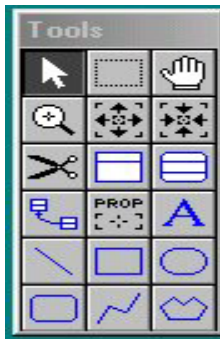


- Ustvarjanje pogledov oz. poizvedb

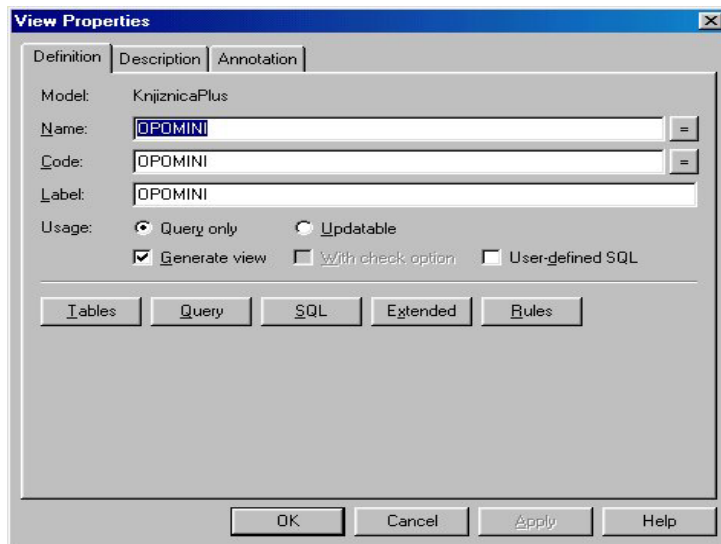
POGLED (VIEW)

Pogled je alternativen način gledanja na podatke ene ali večih tabel. Pogled vključuje podmnožico kolon ene ali večih tabel.

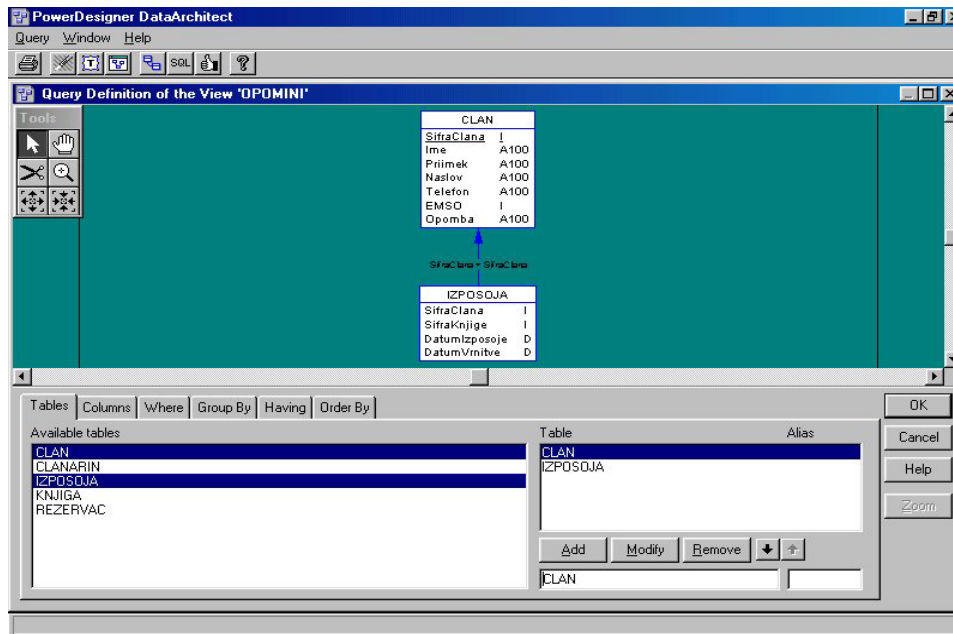
9. Orodna vrstica (Wiev-poizvedba)



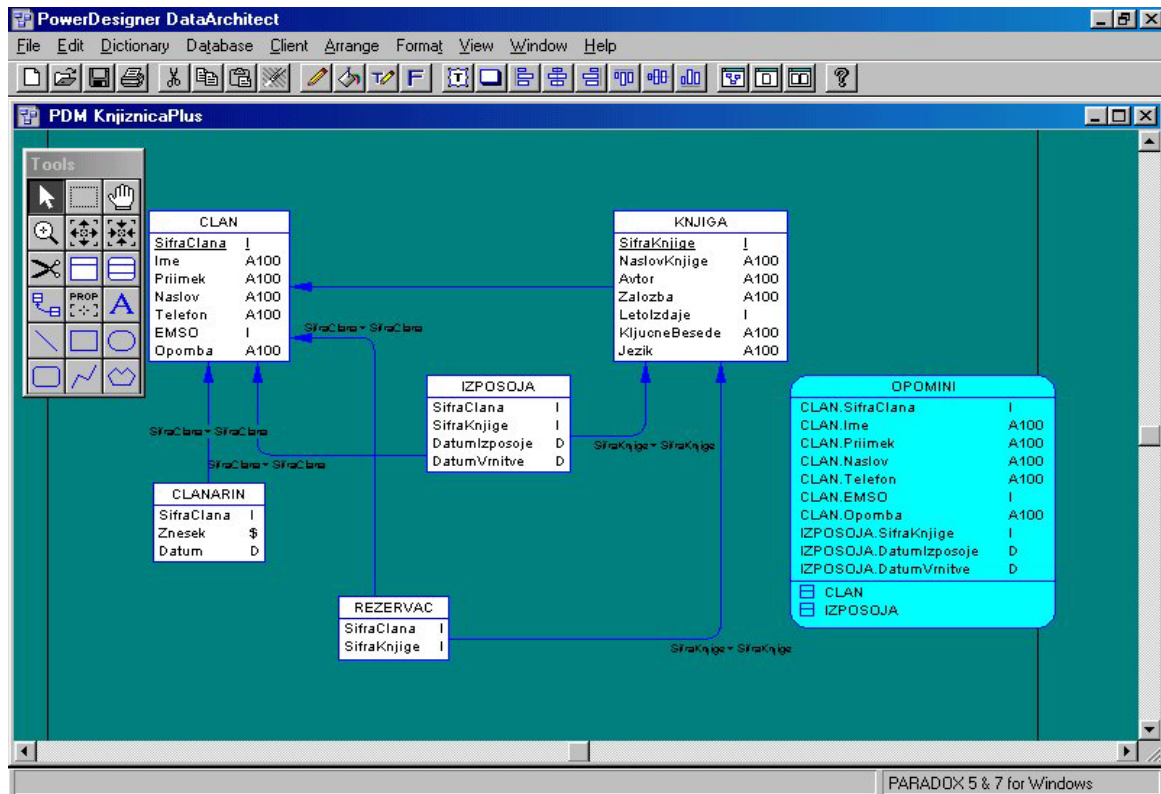
10. Določimo vse lastnosti poizvedbe (ime, uporaba tabel, SQL)



11. Opisovanje podatkov za avtomatsko generiranje SQL stavka

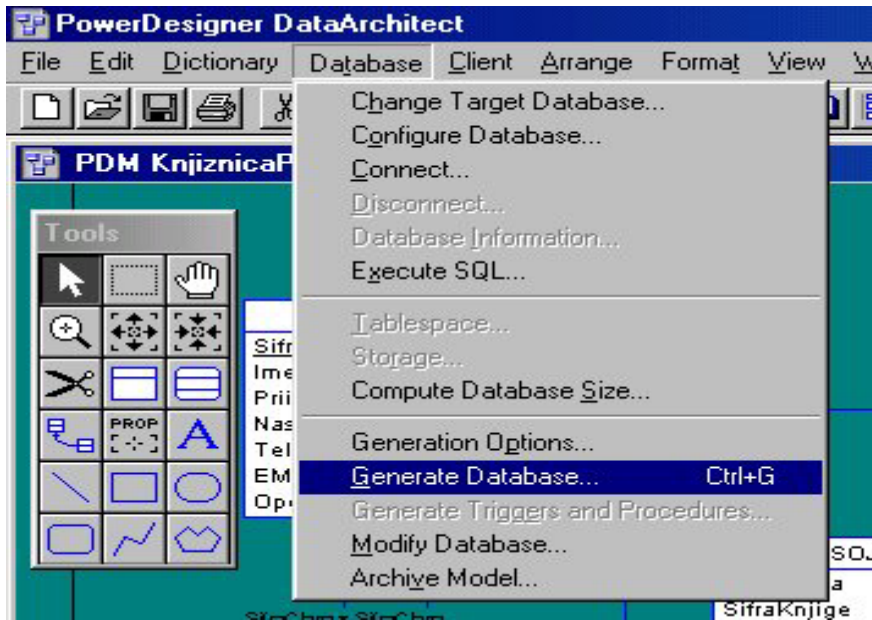


12. Celoten fizični model (poizvedba + fizični izgled baze)

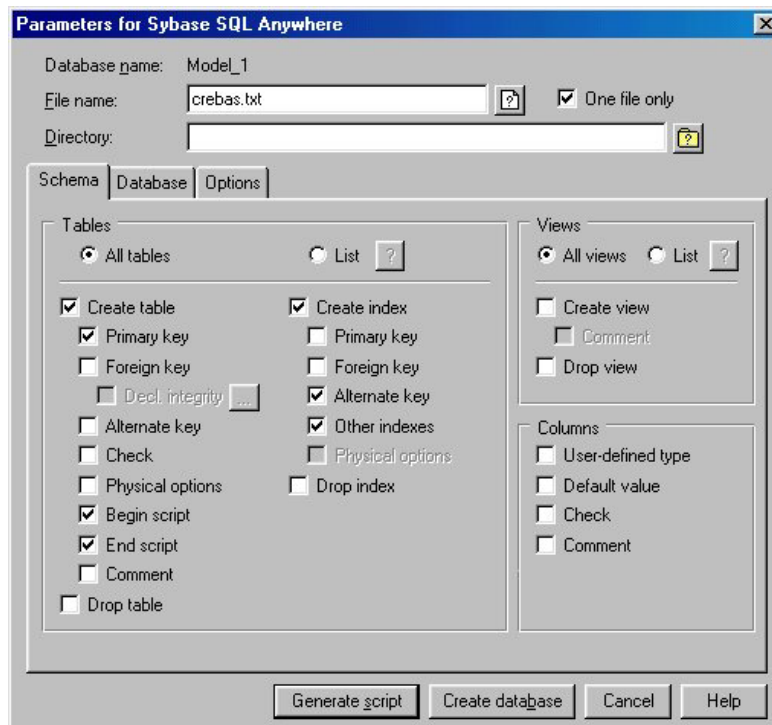


- Avtomatsko ustvarjanje baze iz fizičnega modela

13. Generiranje baze



14. Nastavimo ime skripte in pot, kamor jo bomo shranili



Tako nam program DataArchitect zgenerira skripto (glej dodatek A), ki jo moramo zagnati v Paradoxu. Navodila, kako to narediti, nam da program sam:

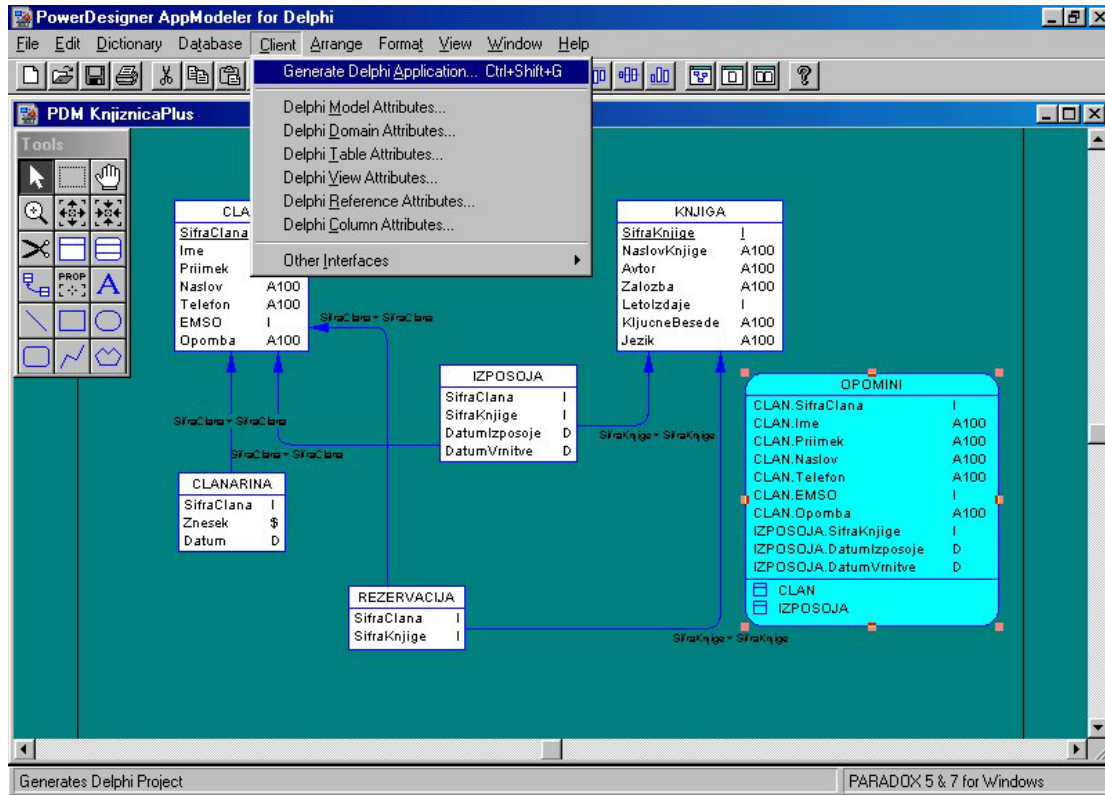
Usage:

- (1) Start PARADOX for Windows*
- (2) With the "Working directory ..." option of the "File" menu, change the working directory to the directory in which you will create files*
- (3) With the "New" option of the "File" menu, create a new script*
- (4) Delete the default code*
- (5) With the "Paste from file" option of the "Edit" menu, load the script crebas.txt*
- (6) Run this script*

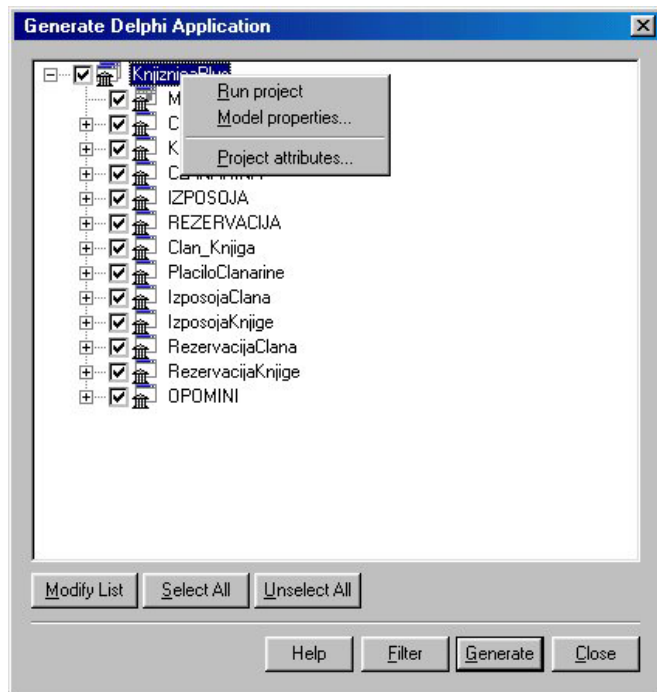
Tako dobimo bazo.

- Generiranje Delphijeve aplikacije

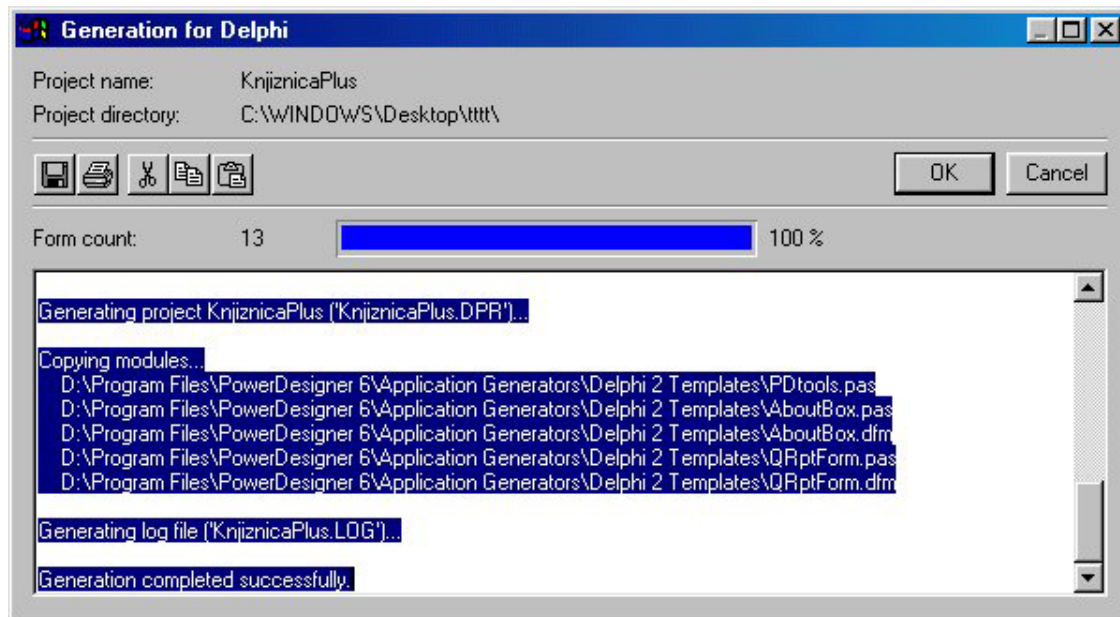
15. Naložimo fizični model in izberemo opcijo za generiranje aplikacije



16. Izberemo stvari, ki naj bodo vključene v aplikacijo



17. Po uspešnem generiranju...



3.

Dodatki programu

Da smo zadovoljili zahtevam naloge, sem morala ustvarjeni aplikaciji dodati še nekaj kode:

- Obrazec za pisanje teksta opomina.
- Popraviti sem morala SQL stavek za poizvedbo o opominih:

Prej (v Power Designerju):

```
select CLAN.SifraClana, CLAN.Ime, CLAN.Priimek, CLAN.Naslov,  
CLAN.Telefon, CLAN.EMSO,  
CLAN.Opomba, IZPOSOJA.SifraKnjige, IZPOSOJA.DatumIzposoje,  
IZPOSOJA.DatumVrnitve  
from CLAN, IZPOSOJA  
where CLAN.SifraClana = IZPOSOJA.SifraClana  
and IZPOSOJA.DatumIzposoje > TodayMOneMonth
```

Sedaj (v Delphiju):

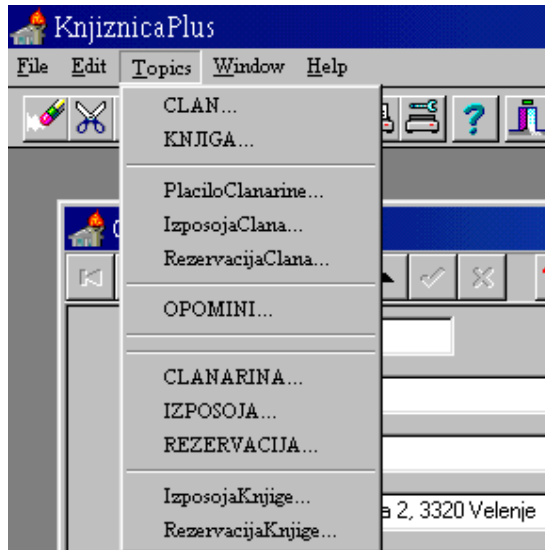
```
Select  
CLAN.SifraClana,  
CLAN.Ime,  
CLAN.Priimek,  
CLAN.Naslov,  
CLAN.Telefon,  
CLAN.EMSO,  
CLAN.Opomba,  
IZPOSOJA.SifraKnjige,  
IZPOSOJA.DatumIzposoje,  
IZPOSOJA.DatumVrnitve  
From  
CLAN,  
IZPOSOJA  
Where  
(CLAN.SifraClana = IZPOSOJA.SifraClana) And  
(IZPOSOJA.DatumIzposoje < :datum) And  
(IZPOSOJA.DatumVrnitve = '1.1.1999')
```

- Datum vrnitve se ob izposoji postavi na **1.1.1999**. S tem preprečim izpis opominov za knjige, ki so že vrnjene. Načeloma bi lahko namesto tega datuma uporabila poljuben datum, ki ustreza dejstvu, da je datum iz preteklosti in programa tedaj še nismo uporabljali.
- IZpisovanje opominov članom, ki bi morali knjige že vrniti.
- Estetske izboljšave; na primer dodajanje črt med opcije menuja Topics, rdeča barva polja DatumVrnitve v obrazcu OPOMINI, ipd.

4.

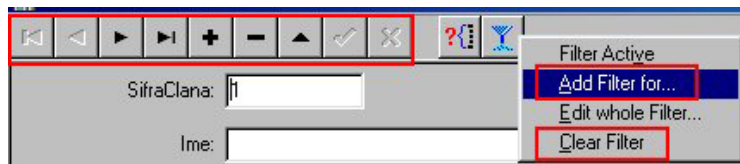
Program KnjiznicaPlus

1. Dosegljive tabele in povpraševanja. Z miško se postavimo na želeno mesto in tam kliknemo. Odpre se iskana tabela z vsemi potrebnimi podatki.

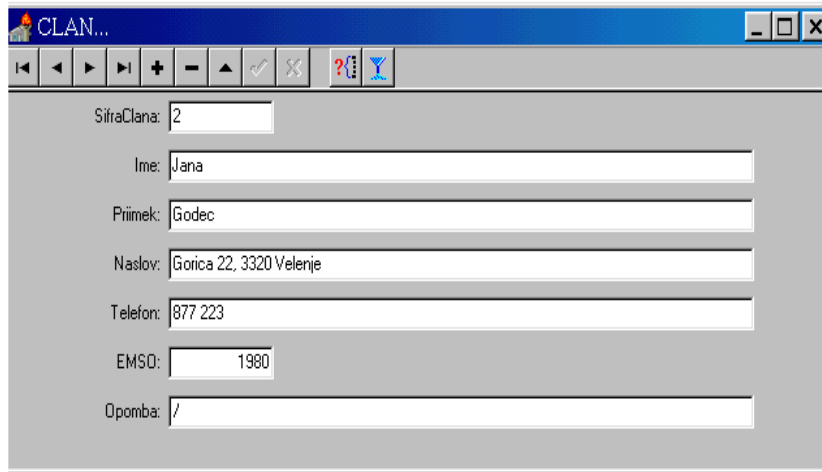


2. Pomen pomembnih (rdeči okvirčki) navigacijskih gumbov:

1. gumb: skoči na prvi zapis tabele
 2. gumb: skoči zapis nazaj
 3. gumb: skoči zapis naprej
 4. gumb: skoči na zadnji zapis
 5. gumb: dodaj zapis
 6. gumb: briši trenutni zapis
 7. gumb: popravi trenutni zapis
 8. gumb: potrditev sprememb
 9. gumb: prekličiči spremembe
- zadnji gumb, 2. opcija: iskanje po izbranem polju
zadnji gumb, 4. opcija: ponovno prikaži celotno vsebino tabele



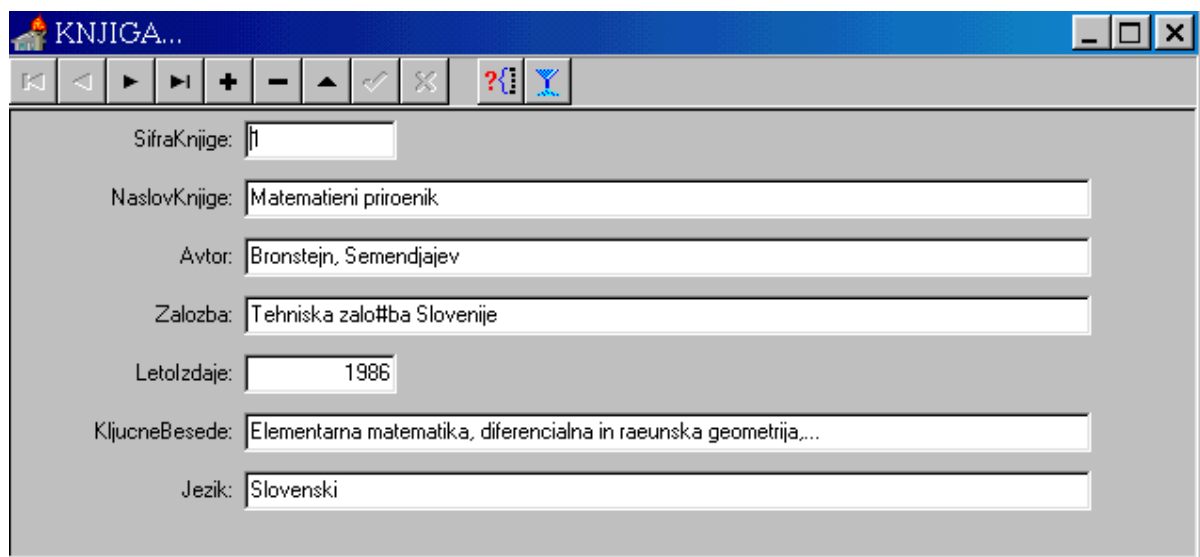
3. V tabeli najdemo podatke o članih. Lahko dodajamo nove člane, jih brišemo in po potrebi popravljamo že obstoječe podatke o članih.



The screenshot shows a window titled "CLAN...". The form contains the following fields:

SifraClana:	2
Ime:	Jana
Preimek:	Godec
Naslov:	Gorica 22, 3320 Velenje
Telefon:	877 223
EMSD:	1980
Opomba:	/

4. Tabela zajema podatke o knjigah, ki jih knjižnica vsebuje.



The screenshot shows a window titled "KNJIGA...". The form contains the following fields:

SifraKnjige:	11
NaslovKnjige:	Matematieni priroenik
Avtor:	Bronstein, Semendjajev
Zalozba:	Tehniska zalo#ba Slovenije
Letolz daje:	1986
KljucneBesede:	Elementarna matematika, diferencialna in raeunska geometrija,...
Jezik:	Slovenski

5. Tabela vsebuje podatke, kateri člani so plačali članarino, znesek članarine in datum plačila po letih.

Placilo Članarine...

SifraClana: 1

Ime: Jure

Priimek: Bukev

Naslov: Zidarjeva 2, 3320 Velenje

Telefon: 876 545

EMSO: 1981

Opomba: /

SifraClana	Znesek	Datum
1	400,00 SIT	2.2.00
1	400,00 SIT	2.3.99
1	400,00 SIT	2.2.98

6. Tabela prikazuje, kateri član si je izposodil katero knjigo, datum izposoje in datum, kdaj mora vrniti izposojeno.

Izposoja Člana...

SifraClana: 1

Ime: Jure

Priimek: Bukev

Naslov: Zidarjeva 2, 3320 Velenje

Telefon: 876 545

EMSO: 1981

Opomba: /

SifraClana	SifraKnjige	DatumIzposoje	DatumVrnitve
1	1	1.1.00	1.2.00
1	2	10.4.00	1.1.99

7. Tabela prikazuje, kateri član si je katero knjigo rezerviral.

RezervacijaClana...

SifraClana:

Ime:

Priimek:

Naslov:

Telefon:

EMSD:

Opomba:

SifraClana	SifraKnjige
1	1

8. Tabela vsebuje podatke o članih, ki jim je potekel rok vrnitve knjig. Omogoča tudi izpis opominov.

OPOMINI...

Tekst opomina Izpis opominov

SifraClana	Ime	Priimek	Naslov	Telefon	EMSD	Opomba	SifraKnjige	DatumIzposoje	DatumVrnitve
3	Marko	Kralj	Stari trg 2, 3310 #alec	718 439	1981	/	3	3.3.00	1.1.99
2	Jana	Godec	Gorica 22, 3320 Velenje	877 223	1980	/	1	10.3.00	1.1.99
1	Jure	Bukev	Zidarjeva 2, 3320 Velenje	876 545	1981	/	2	10.4.00	1.1.99

9. Pisanje opominov zamudnikom.

Spreminjanje teksta pisma

TEKST

+ ZADEVA: Opomir

Spoštovani,

prosimo, če nam v najkrajšem roku vrnete omenjeno knjigo in poravnate opomin v znesku 300 SIT!

Hvala in lep pozdrav Uprava knjižnice

PREKLIČI

V REDU

5.

Zaključek in nadaljne delo

Možne izboljšave:

1. avtomatsko izpisovanje šifer
2. avtomatsko izpisovanje datumov izposoje
3. lepši izpis(smiselno deljenje dolgih vrstic)
4. večkratni opomini
5. evidenca o neplačanih in plačanih opominih
6. opomin za več knjig
7. evidenca o poslanih opominih

In še beseda uporabniku: Ob koncu bi vam zaželela čim bolj uspešno delo z programom. Upam, da bo program koristil vašemu poslovanju.

DODATEK A

Skripta za kreiranje baze

```
; Database name: "KnjiznicaPlus"
; DBMS name: PARADOX 5 & 7 for Windows
; Created on: 5.5.100 18:51
; =====
; -----
; procedure for testing the existence of a file
; -----
proc canCreate(name String) logical
  message("Creating file \"\""+name+".db\" ...")
  if isTable(name) then
    if msgQuestion("Confirmation", "File \"\""+name+".db\" already exist. "+
      "Do you want to remplace it ?") <> "Yes" then
      return False
    endif
  endif
  return True
endProc

; -----
; main method
; -----
method run(var eventInfo Event)
var
  newTab Table
endVar

if canCreate("CLAN") then
; =====
; Table: "CLAN"
; =====
newTab = create "CLAN" as "Paradox" with
  "SifraClana" : "I",
  "Ime" : "A100",
  "Priimek" : "A100",
  "Naslov" : "A100",
  "Telefon" : "A100",
  "EMSO" : "I",
  "Opomba" : "A100"
  key "SifraClana"
endCreate
```

```
; end of table "CLAN"  
endif
```

```
if canCreate("KNJIGA") then
```

```
; =====  
; Table: "KNJIGA"  
; =====
```

```
newTab = create "KNJIGA" as "Paradox" with  
  "SifraKnjige"   : "I",  
    "NaslovKnjige" : "A100",  
  "Avtor"        : "A100",  
  "Zalozba"      : "A100",  
  "LetoIzdaje"   : "I",  
  "KljucneBesede" : "A100",  
  "Jezik"        : "A100"  
  key "SifraKnjige"  
endCreate
```

```
; end of table "KNJIGA"  
endif
```

```
if canCreate("CLANARINA") then
```

```
; =====  
; Table: "CLANARINA"  
; =====
```

```
newTab = create "CLANARINA" as "Paradox" with  
  "SifraClana"   : "I",  
  "Znesek"       : "$",  
  "Datum"        : "D"  
endCreate
```

```
; end of table "CLANARINA"  
endif
```

```
if canCreate("IZPOSOJA") then
```

```
; =====  
; Table: "IZPOSOJA"  
; =====
```

```
newTab = create "IZPOSOJA" as "Paradox" with  
  "SifraClana"   : "I",  
  "SifraKnjige"   : "I",  
  "DatumIzposoje" : "D",  
  "DatumVrnitve"  : "D"  
endCreate
```

```
; end of table "IZPOSOJA"  
endif
```

```
if canCreate("REZERVACIJA") then
```

```
; =====  
; Table: "REZERVACIJA"  
; =====  
newTab = create "REZERVACIJA" as "Paradox" with  
    "SifraClana"    : "I",  
    "SifraKnjige"  : "I"  
endCreate  
  
; end of table "REZERVACIJA"  
endif  
  
msgInfo("Message", "The database has been successfully created.")  
  
endMethod  
  
run
```

Literatura

- [1] T. Mohorič, UVOD V PODATKOVNE BAZE, BI-TIM, Ljubljana 1995
- [2] U. Mesojedec, DELPHI, Desk, 1996
- [3] M. Bajec, POWER DESIGNER, fri.uni-lj.si

Zahvala

Zahvaljujem se vsem, ki so pripomogli k razvoju tega programa, še posebej mentorici gospe inž. Alenki Pačnik - Gabrovec in gospodu dipl. inž. Srečku Zormanu.

Zahvaljujem se tudi dipl. inž. Petru Peeru za predstavitev programa Power Designer in koristne nasvete.

Izjava

Izjavljam, da sem seminarsko nalogo naredila sama. Izkazano pomoč drugim sem v celoti navedla v zahvali.