

Poslovna aplikacija za sajam knjiga Interliber nad relacijskom bazom podataka - Clarion

Rigler, Silvija

Undergraduate thesis / Završni rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka / Sveučilište u Rijeci**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:195:143328>

Rights / Prava: [Attribution 4.0 International](#)/[Imenovanje 4.0 međunarodna](#)

Download date / Datum preuzimanja: **2025-02-22**



Repository / Repozitorij:

[Repository of the University of Rijeka, Faculty of Informatics and Digital Technologies - INFORI Repository](#)



Sveučilište u Rijeci, – Odjel za informatiku

Preddiplomski jednopredmetni studij informatike

Silvija Rigler

Poslovna aplikacija za sajam knjiga
Interliber nad relacijskom bazom
podataka - Clarion

Završni rad

Mentor: prof.dr. sc. Patrizia Pošćić

Rijeka, rujan 2020.

SADRŽAJ

1. SAŽETAK.....	4
2. KLJUČNE RIJEČI.....	4
1. UVOD	5
1.1. Što je aplikacija?	6
1.2. Što je baza podataka, a što relacijska baza podataka?	6
1.3. Što je relacijski model?.....	6
1.4. Što je DEV?	7
2. Izrada modela.....	8
3. Izrada aplikacije u Clarionu.....	11
3.1. Izrada rječnika.....	11
3.1.1. Izrada .dct datoteke	11
3.1.2. Stvaranje tablica i stupaca	11
3.1.3. Određivanje ključeva	17
3.1.4. Dodavanje veza između tablica	21
3.2. Izrada aplikacije	24
3.2.1. Izrada procedure glavniizbornik te procedura Popis i Ažuriranje	24
3.2.2. Dizajn aplikacije	31
3.3. Izvještaji.....	33
4. ZAKLJUČAK.....	36
5. LITERATURA	37
6. Popis slika.....	38

Rijeka, 18. lipnja 2020.

Zadatak za završni rad

Pristupnik: Silvija Rigler

Naziv završnog rada: Poslovna aplikacija za sajam knjiga Interliber nad relacijskom bazom podataka - Clarion

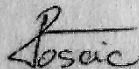
Naziv završnog rada na eng. jeziku: Business application with relational database for book fair Interliber - Clarion

Sadržaj zadatka:

Baza podataka (BP) predstavlja kolekciju podataka, ograničenja i operacija koji reprezentiraju neke aspekte realnoga svijeta. Dakle, BP je model neke aplikacijske domene. Cilj rada je izgraditi aplikaciju za potrebe procesa upravljanja sastancima u nekom poduzeću (sajam knjiga Interliber). Aplikacija će biti izgrađena nad relacijskom BP pomoću RAD alata Clarion. Od studentice se očekuje da kreira model podataka i odgovarajuću poslovnu aplikaciju.

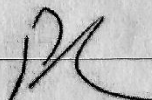
Mentor

Prof. dr.sc. Patrizia Poščić

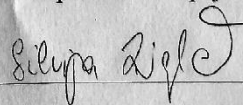


Voditelj za završne radove

Dr. sc. Miran Pobar



Zadatak preuzet: 19. lipnja 2020.



(potpis pristupnika)

1. SAŽETAK

Završni rad prikazuje izradu poslovne aplikacije za sajam knjiga Interliber nad relacijskom bazom podataka u programskom alatu Clarion. Aplikacija izvršava osnovne CRUD operacije (pregled, dodavanje, brisanje i izmjena zapisa. Navedene su i opisane sve metode koje su korištene u izradi. Detaljan opis završnog rada nalazi se u uvodu, a rezultat završnog rada je funkcionalna poslovna aplikacija

2. KLJUČNE RIJEČI

Clarion, Baza podataka, Model, DEV, Rječnik, Aplikacija, Ključ, Veza, Dizajn, Izvještaj

1. UVOD

Možemo vidjeti kako se poslovni svijet sve više okreće korištenju tehnologije, a time i korištenju baze podataka u poslovanju. Uglavnom je to zbog jednostavnosti praćenja poslovnog sustava, pristupačnosti te sigurnosti. Iz tog razloga će ovaj završni rad pratiti izradu sustava koji se oslanja na baze podataka. Korisniku će biti omogućen unos, izmjena, ažuriranje te brisanje podataka. Opisuje se planiranje, dizajn i razvoj poslovne aplikacije „Interliber“ nad relacijskom bazom podataka „Clarion“. U radu će biti opisano kako se definira rječnik poslovnog sustava, a nakon toga i sama aplikacija u programskom jeziku Clarion. Clarion je komercijalni programski alat tvrtke Soft Velocity koji je i sam dizajniran za ovakvu vrstu aplikacije te je kompatibilan sa SQL jezikom. Aplikacija se razvija pomoću gotovih elemenata i uz smanjenu upotrebu pisanja dugačkih linija koda.

Interliber je hrvatski sajam knjiga. Svrha sajma je promocija knjiga, čitanja i povezivanja autora s nakladničkim industrijama te tehnologijom koja je u današnje doba veoma bitan aspekt poslovanja. S obzirom na to, bilo bi dobro imati bazu podataka koja bi čuvala informacije o izlagačima, nakladnicima, posjetiteljima, izlošcima i računima.

Za početak će biti opisani osnovni termini koji su korišteni u izradi ove poslovne aplikacije, poput objašnjenja što je i zašto se koristi relacijski model te koja pravila treba pratiti prilikom izrade. Nakon toga opisuje se izrada rječnika. Unos tablice, stupaca i veza, u koje će se kasnije moći upisivati željeni podaci, su temelj u izradi rječnika, a kasnije i same aplikacije. Za kraj je ostavljena izrada aplikacije u RAD alatima gdje se opisuje način stvaranja i dizajn svih prozora aplikacije. RAD alati nastaju zbog želje za lakšom i kvalitetnijom izradom aplikacije. Temelje se na brzini izrade prototipa i povratne informacije korisnika. Do gotove aplikacije dolazi se u kratko vrijeme.

1.1. Što je aplikacija?

Aplikacija je računalni program stvoren kako bi obavljao neku zadaću, odnosno programski proizvod nad bazom podataka koji obavlja određenu poslovnu funkciju (Pavlič, 2011.).

Može se reći da aplikacija pruža pomoć korisnicima u izvršavanju zadataka iz različitih područja ljudske djelatnosti (primarno, sekundarno, tercijarno i kvartarno područje). Unutar aplikacije postoji skup naredbi ili uputa kojima se upravlja ovisno o vrsti aktivnosti. Obrađuje tekst, brojeve i grafiku, a izrađuje se prema zahtjevima određenog poslovnog sustava. Poslovni sustavi kroz izradu poslovne aplikacije povećavaju produktivnost i prate obavljanje zadataka unutar sustava.

1.2. Što je baza podataka, a što relacijska baza podataka?

Baza podataka je model određenog segmenta stvarnog svijeta na najnižoj razini apstrakcije (Tkalac, 1988.). Predstavlja kolekciju podataka, ograničenja i operacija koji reprezentiraju neke aspekte realnog svijeta. Unutar baze podaci se organiziraju (modeliraju) u obliku povezanih tablica koje se sastoje od stupaca i redova.

Relacijska baza podataka je skup relacija definiranih relacijskom shemom baze podataka (Varga, 1994.). Zasniva se na relacijskom modelu koji će biti opisan kasnije. Podaci su međusobno povezani relacijama, a strukturiraju se na način da bi se postigla ažurnost, arhiviranje, administriranje i točnost pohranjenih podataka. Pošto je spomenuta relacija, ona se može opisati kao imenovana tablica koja se sastoji od atributa ili stupaca tablica, odnosno n-torki ili redaka tablice. Veza između relacija se ostvaruje isključivo preko vrijednosti atributa. Svaka relacija obavezno mora sadržavati primarni ključ koji jedinstveno identificira svaku n-torku relacije. Oblik relacije opisuje se relacijskom shemom, a sastoji se od imena relacije i popisa atributa.

1.3. Što je relacijski model?

Postoje različite kategorije relacijskog modela koji sadrži podatke i interpretaciju podataka, poput strukture modela, ograničenja u modelu i skupa operatora u modelu. Struktura relacijskog modela definira se shemom relacijskog modela podataka koja ima oblik: *naziv_rel(naziv_atr1, naziv_atr2, ...)*. (Tkalac 1988.) Model sadrži opis baze podataka i opisuje osnovnu ideju za izradu aplikacije. Jednostavan je u uporabi i razumijevanju.

Ključ jedinstveno identificira n-torke relacije. Relacija može imati različite vrste ključeva. Razlikujemo primarni i vanjski ključ. Vanjski ključ je ključ druge relacije u modelu koja je povezana sa prvom. Od niza kandidata za ključ bira se jedan koji se naziva primarni ključ. Atribut relacije je ključ ako je jednoj vrijednosti atributa u relaciji pridružena samo jedna vrijednost ostalih atributa relacije, odnosno ako su svi neključni atributi funkcijski ovisni o ključu.

Prema Tkalcu osnovni elementi relacijskog modela su:

- Skup objekata koji čine osnovne elemente baze podataka
- Skup operacija koje se izvode nad objektima modela
- Skup pravila integriteta baze podataka (Tkalac,1988.)

Znanje krajnjeg korisnika o relacijskom modelu treba biti jednostavno i razumljivo, a to bi trebao biti i cilj kojeg informatičar mora dostići. Informatičari bi trebali poznavati relacijski model da bi mogli uspješno izvršiti korisničke.

1.4. Što je DEV?

DEV ili dijagram entiteti-veze opisuje model entiteta, a prikazuje se grafički. Dijagram u velikoj mjeri olakšava razumijevanje modela podataka. Model opisuje poslovna pravila određenog poslovnog sustava.

„Entitet je stvaran ili apstraktan predmet ili događaj o kojemu se u informacijskom sustavu prikupljaju podaci.“ (Varga, 1994.) Entitet je podatak koji je važan za svaki poslovni sustav. Opis jednog svojstva sastoji se od atributa kojima se određuje vrsta svojstva i vrijednosti atributa. Atributi opisuju određeno svojstvo, a mijenjaju se u ovisnosti o vremenu nakon čega je potrebno ažurirati bazu podataka. Svojstvo je osnova prema kojem se entiteti svrstavaju u skupove. Vrijednost atributa jednoznačno određuje entitet u promatranom skupu. Ne postoje dva entiteta s jednakom vrijednosti atributa nekog svojstva. Odnosno, atribut bi trebao ispunjavati uvjete jedinstvenosti i uvjet neredundantnosti kako bi se mogao smatrati ključem relacije. Ako postoji više takvih atributa, oni se nazivaju kandidati za ključ. Primarni ključ relacije ispunjava sljedeći kriterij: svi ne ključni atributi su funkcijski ovisni o ključu relacije (jedna vrijednost atributa u relaciji pridružena je samo jednoj vrijednosti ostalih atributa iste relacije). Vanjski ključ je ključ neke druge relacije u modelu, a ključ za sortiranje služi za razvrstavanje podataka u tablici.

Također postoji i podjela tipa entiteta na slabi i agregirani tip entiteta. Slab tip entiteta ne može postojati ako nije vezan uz neki tip entiteta i uvijek ima složen primarni ključ nadtipa i ključ podtipa. Agregacija je apstrakcija u kojoj se tip veze između dvaju ili više tipova uzima kao novi tip entiteta. Radi se o još jednom složenom ključu i to dvaju entiteta koje povezuje.

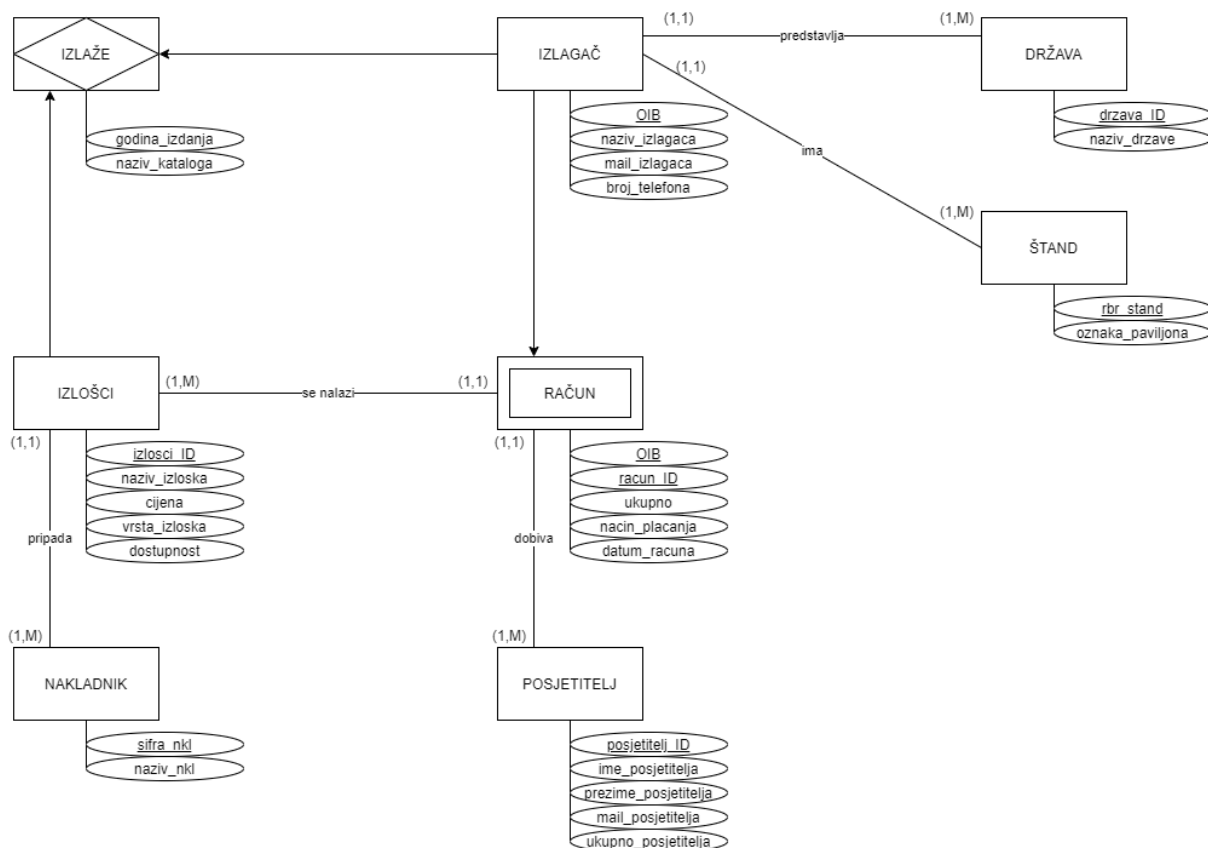
Svaki odnos između entiteta ima određena svojstva, a s obzirom na broj entiteta određene vrste koji su u određenom odnosu, možemo imati različite vrste odnosa što nazivamo brojnost tipa veze. Zapisuje se kao uređen par brojeva (DG:GG), DG je donja granica, a GG je gornja granica. Upisuje se na vezu u blizini entiteta o kojemu je riječ.

2. Izrada modela

Model se izrađuje pomoću dijagrama entiteti-veze. Dijagram entiteti-veze dolazi uz koncepte kao što su: entiteti, atributi, veze, specijalne veze, slabi entitet i agregacija. Svaki od tih koncepata ima svoj vlastiti oblik po kojemu se prepoznaje.

Entitet je prikazan pomoću pravokutnika unutar kojeg se upisuje naziv velikim tiskanim slovima. Atributi se nalaze ispod entiteta u obliku elipsa, atribut koji je ujedno i primarni ključ podcrtava se punom linijom, a vanjski ključ se podcrtava isprekidanom linijom. Veze su ravne linije na koje se upisuju brojnosti u blizini entiteta kojega opisuju i dobivaju naziv. Specijalne veze su prikazane usmjerenom linijom (strelicom) prema slabom tipu entiteta ili agregaciji. U dvostrukom pravokutniku nalazi se slabi tip entiteta, a romb unutar pravokutnika označava agregaciju.

Primjer dijagrama entiteti-veze prikazan je na slici 1.4.1, a nakon njega je prikazana i relacijska shema. Zajedno daju cjeloviti prikaz relacijskog modela koji je detaljniji i jasniji za pregled i razumijevanje. Imena tablica su ovdje navedena sa dijakritičkim znakovima dok će u Clarionu biti navedena bez njih.



Slika 2.1. Dijagram entiteti veze

Relacijska shema definira strukturu relacijskog modela.

Relacijska shema:

IZLAŽE (izlosci_ID, OIB, godina_izdanja, naziv_kataloga)

IZLOŠCI (izlosci_ID, naziv_izloska, cijena, vrsta_izloska, dostupnost, sifra_nkl)

IZLAGAČ (OIB, naziv_izlagaca, mail_izlagaca, broj_telefona, drzava_ID, rbr_stand)

RAČUN (racun_ID, OIB, ukupno, nacin_placanja, datum_racuna, izlosci_ID)

ŠTAND (rbr_stand, oznaka_paviljona)

POSJETITELJ (posjetitelj_ID, ime_posjetitelja, prezime_posjetitelja, mail_posjetitelja, ukupno_posjetitelja, racun_ID, OIB)

NAKLADNIK (sifra_nkl, naziv_nkl)

DRZAVA (drzava_ID, naziv_drzave)

Prvi entitet koji se opisuje je IZLAGAČ kojemu je primarni ključ *OIB*, uz attribute *naziv_izlagaca*, *br_tel*. *OIB* jedinstveno identificira svakog pojedinog izlagaca, a uz kojeg obavezno stoji i njegov naziv. Kako bi se izlagača moglo kontaktirati, stavljeni su i kontakt podaci poput mail adrese i broja telefona. Izlagač predstavlja neku od država i dodjeljuje mu se štand, a to je određeno vanjskim ključem kojeg se prema pravilu ne upisuje na dijagram.

Prvi vanjski ključ *drzava_ID* je primarni ključ entiteta DRZAVA. Tablica prikazuje attribute *naziv_drzave* raspoređene prema primarnom ključu. Entiteti IZLAGAČ i DRZAVA povezuju se običnom vezom s brojnostima (1,1):(1,M) koje znače da jedan izlagač predstavlja jednu ili više država.

Entitetu STAND pripada drugi vanjski ključ *rbr_stand*. U toj tablici nalazi se atribut *oznaka_paviljona*(5, 6, 7A, KON, ZIMSKI) koji predstavlja oznaku štanda prema kojemu se može saznati gdje se nalazi izlagač kojeg tražimo. Tablice su opet povezane običnom vezom s brojnostima (1,1):(1,M).

Sljedeći je slabi tip entiteta RACUN, a kojemu je primarni ključ složen od atributa jakog tipa entiteta *OIB* i atributa slabog tipa entiteta *racun_ID*. Ostali atributi *su nacin_placanja* (kartično, gotovinsko), *ukupno* (zbroy cijena kupljenih izložaka) i *datum_racuna*. Pošto se radi o složenom ključu, uklanjanjem jakog tipa entiteta dolazi do uklanjanja slabijeg. Znači, ako uklonimo tablicu IZLAGAČ, nestat će i tablica RACUN jer izlagaci izdaju racun. Brojnosti ne upisujemo na dijagram jer je riječ o specijalnoj vezi.

Vanjski ključ tablice RACUN je *posjetitelj_ID*. Ujedno je i primarni ključ tablice POSJETITELJ. U toj tablici se doznaju podaci o posjetitelju (*ime_p*, *prezime_p*, *mail_p*) i ukupan broj posjetitelja (*br_pos*). Sa tablicom RACUN povezana je s brojnosti (1,M):(1,1), jedan posjetitelj može dobiti više računa.

Tablica RACUN ima još jedan vanjski ključ, a to je *izlosci_ID* iz tablice IZLOSCI. Ovdje je prikazan i primjer ključa za sortiranje pod nazivom *naziv_izloska*, ukoliko se traži određeni

naziv izložka može se prema tom ključu i tražiti. Tu su i atributi *cijena*, *vrsta_izloska* (može biti riječ o udžbeniku za srednje škole, knjiga, časopis,...), *dostupnost* (provjerava se da li je izložak dostupan). Tablice su povezane vezom (1,M):(1,1). Na jednom računu se nalazi najmanje jedan, a najviše mnogo izložaka.

Primarni ključ tablice NAKLADNIK *sifra_nkl*, vanjski je ključ tablice NAKLADNIK o kojemu se saznaje naziv nakladničke kuće iz atributa *naziv_nkl*, a koja izdaje određeni izložak. Brojnosti su određene vezom (1,1):(1,M). Jedan nakladnik pripada najmanje jednom i najviše mnogo izložaka.

Agregacija IZLAŽE tvori se od primarnih ključeva *izlosci_ID* i *OIB*, a uz njih su atributi *godina_izdanja* i *naziv_kataloga*. Tablica IZLAŽE se može zamisliti kao katalog preko kojeg se prati dostupnost izložaka te kod kojeg izlagača se izložak nalazi. Ponavljam da se brojnosti specijalnih tipova veza ne upisuju.

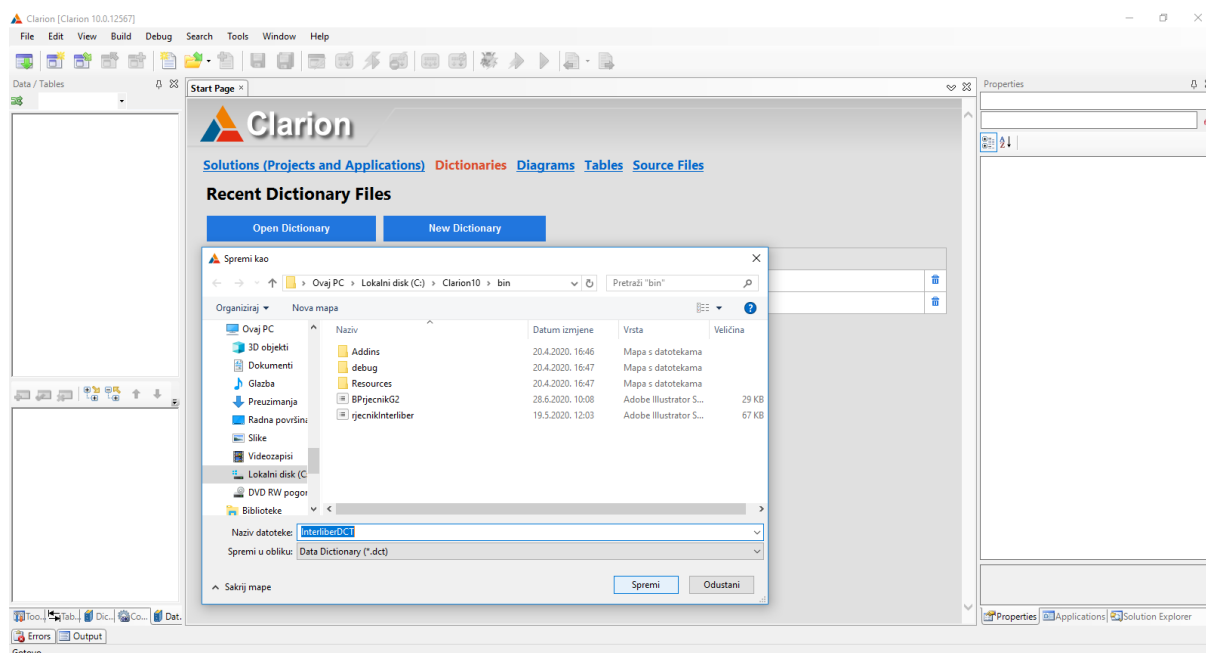
3. Izrada aplikacije u Clarionu

3.1. Izrada rječnika

Izrada aplikacije u Clarionu počinje od rječnika koji se radi na osnovi modela koji je objašnjen u prethodnom koraku. On daje uputu softveru kako oblikovati podatke koje će korisnik unositi u bazu podataka. Unutar rječnika prvo se izrađuje nova .dct datoteku, naprave se tablice, stupci i njihova svojstva, odrede se ključevi i postave veze između tablica.

3.1.1. Izrada .dct datoteke

Prvo otvoriti Clarion i na početnom zaslonu među ostalim projektima odabrati *Dictionaries*. Nakon toga pritisnuti tipku *New dictionary*. Otvara se prozor za odabir naziva rječnika i odabir mjesta gdje će se spremiti u obliku .dct datoteke. Prikaz možemo vidjeti na slici ispod (Slika 3.1.1.).



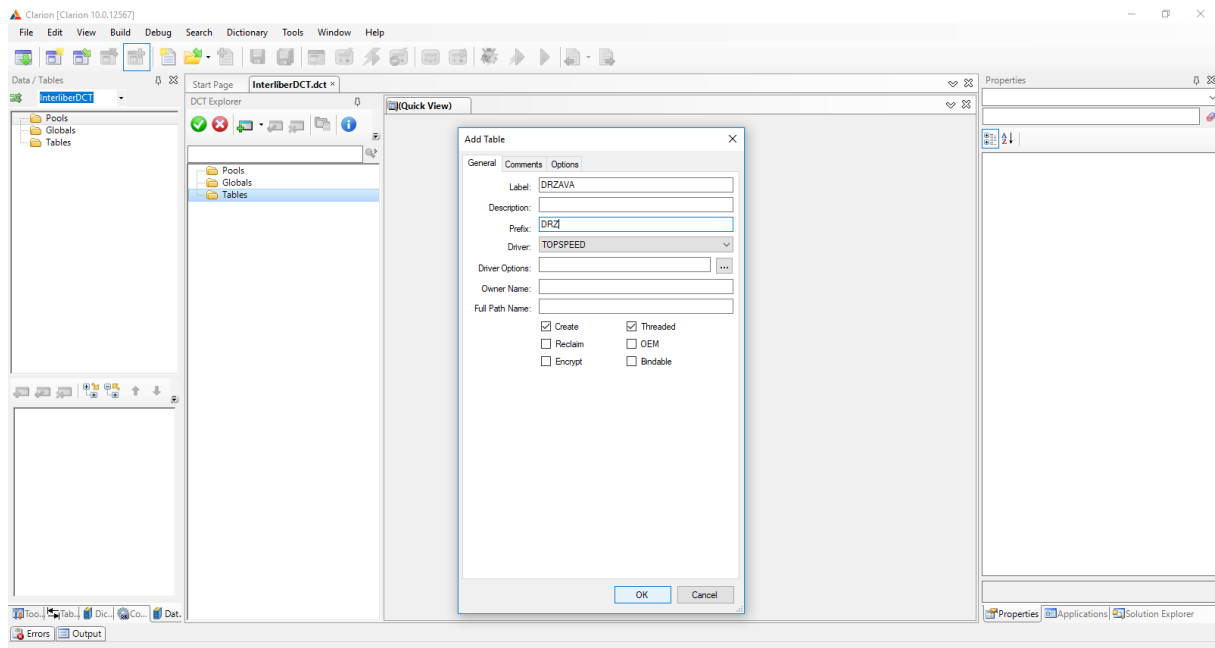
Slika 3.1. Stvaranje .dct datoteke i spremanje rječnika

3.1.2. Stvaranje tablica i stupaca

Spremanjem rječnika, otvara se Clarionov *DCT Explorer* sa praznim mjestima za unošenje *Pools*, *Globals* i *Tables* gdje se klikom na mapu *Tables* odabire *Add Table* za izradu prve tablice.

Otvorio se prozor u koji se upisuju svojstva koja želimo da ima naša tablica. Počinje se od tablica koje sadrže samo svoj primarni ključ, nakon toga unose se tablice koje imaju svoj

primarni ključ i vanjski ključ, a na kraju slabi tip entiteta i agregacija. Prefiks se automatski kreira nakon unošenja imena tablice, a uzimaju se prva tri slova naziva tablice. Bilo bi poželjno promijeniti ih kako ne bi došlo do miješanja dviju tablica (IZLAŽE, IZLOŠCI, IZLAGAČ). Ukoliko želimo može se staviti *Description*, opis tablice, odnosno što ona predstavlja, ukoliko nije dovoljno jasno iz naziva. Ostale postavke biraju se po defaultu, klikne se *OK* i nastavlja na isti način sa svim ostalim tablicama. (Slika 3.2.1.)

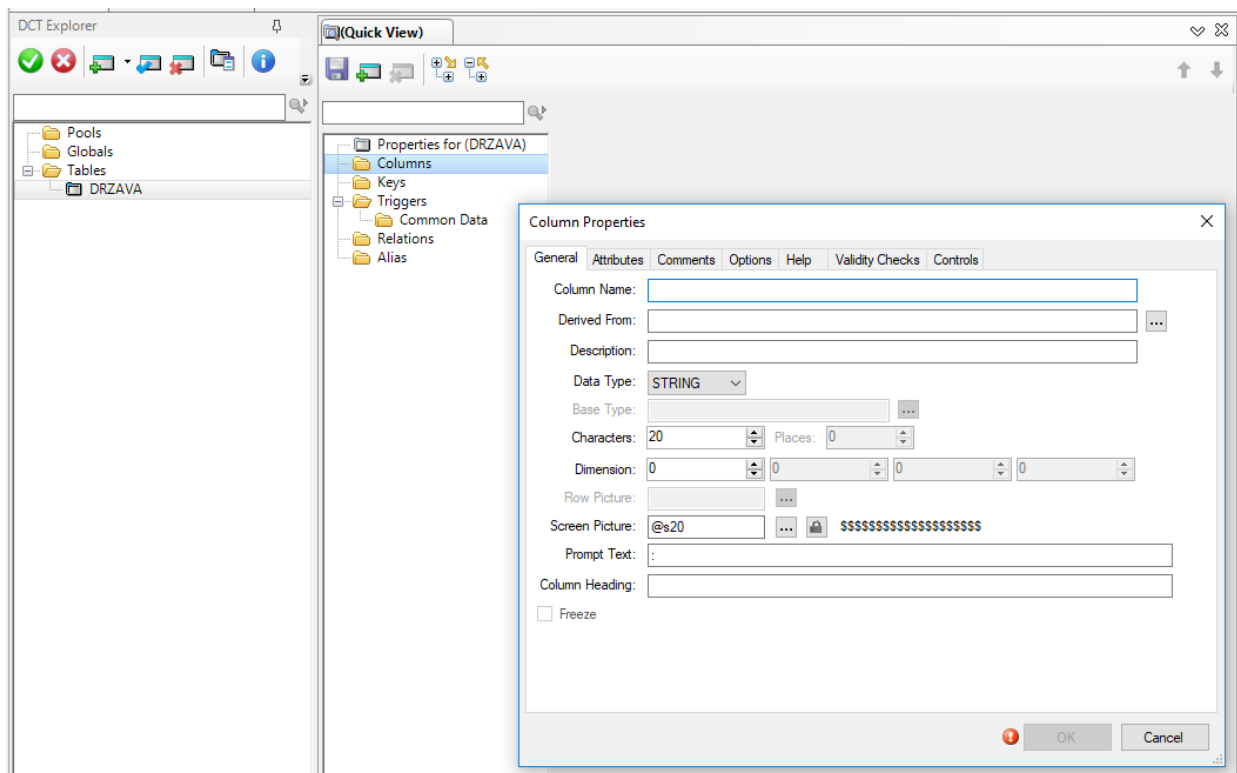


Slika 3.2. dodavanje tablice

Prilikom dodavanja stupaca tablicama često se dolazi do grešaka pa treba pripaziti na parametre koji se dodaju. Tablica se dodaje pritiskom na jednu od tablica, u mom slučaju tablica DRZAVA, zatim desni klik na mapu *Columns* i odabrati *Add Column*.

Otvora se prozor *Column Properties* gdje se odabire kartica *General* i upisuje *Column Name* (npr. *drzava_ID*), *Data Type* (*String* ili *Short*), a u polje *Screen Picture* upisuje se oblik u kojem podatak želimo upisati, npr. *ID države* želimo upisati kao troznamenasti broj pa se upisuje *@P####P* i taj tip uzorka nadjačava tip podatka koji je stavljen u *Data Type*. Najčešće se u *Screen Picture* upisuje npr. *OIB* koji je određen sa točno 11 znamenki ili broj telefona od 15 znamenki (koristeći crtice koje razdvajaju znamenke, a mogu se staviti i početne znamenke broja, npr. 00385). Polje *Prompt Text* je način na koji se u aplikaciji prikazuje ime stupca, ukoliko je nepotpuno može se promijeniti i zapisati u cjelosti, u ovom slučaju ne treba mijenjati. Ostala su još neka polja koja se ne upisuju za svaki stupac, a to je *Derived From* za polja koja se povlače iz druge tablice (radi se o vanjskim ključevima). U *Characters* se upisuje željeni broj znakova za određeni tip podatka i *Dimension* je oznaka dimenzije (1D, 2D, 3D) koju mi nećemo upotrebljavati.

Clarion u *Data Type* daje razne tipove podataka, poput: BLOB, BYTE, CSTRING, DATE, DECIMAL, GROUP, LIKE, LONG, MEMO, PICTURE, PSTRING, REAL, SHORT, SREAL, STRING, TIME, ULONG, USHORT. Uglavnom se najviše koriste podebljani tipovi koji će biti objašnjeni u nastavku. CSTRING je sličan tipu STRING samo što uvijek upisuje jednu znamenku više, ali i jedan i drugi znače upisivanje nekog znakovnog niza. DATE koristimo kod upisivanja datuma gdje se u Screen Picture može odrediti točan uzorak koji je potreban u prikazu datuma. DECIMAL se uzima kod navođenja cijena izložaka, a može se koristiti i kod navođenja nekih drugih mjera koje je najbolje prikazati decimalnim zapisom. LONG je duži zapis (4B) za upis više od 8 brojeva, a SHORT je kraći zapis brojeva (2B). TIME se koristi kod navođenja vremena, ukoliko želimo kod tiskanja računa navesti i njegovo vrijeme, pored datuma.

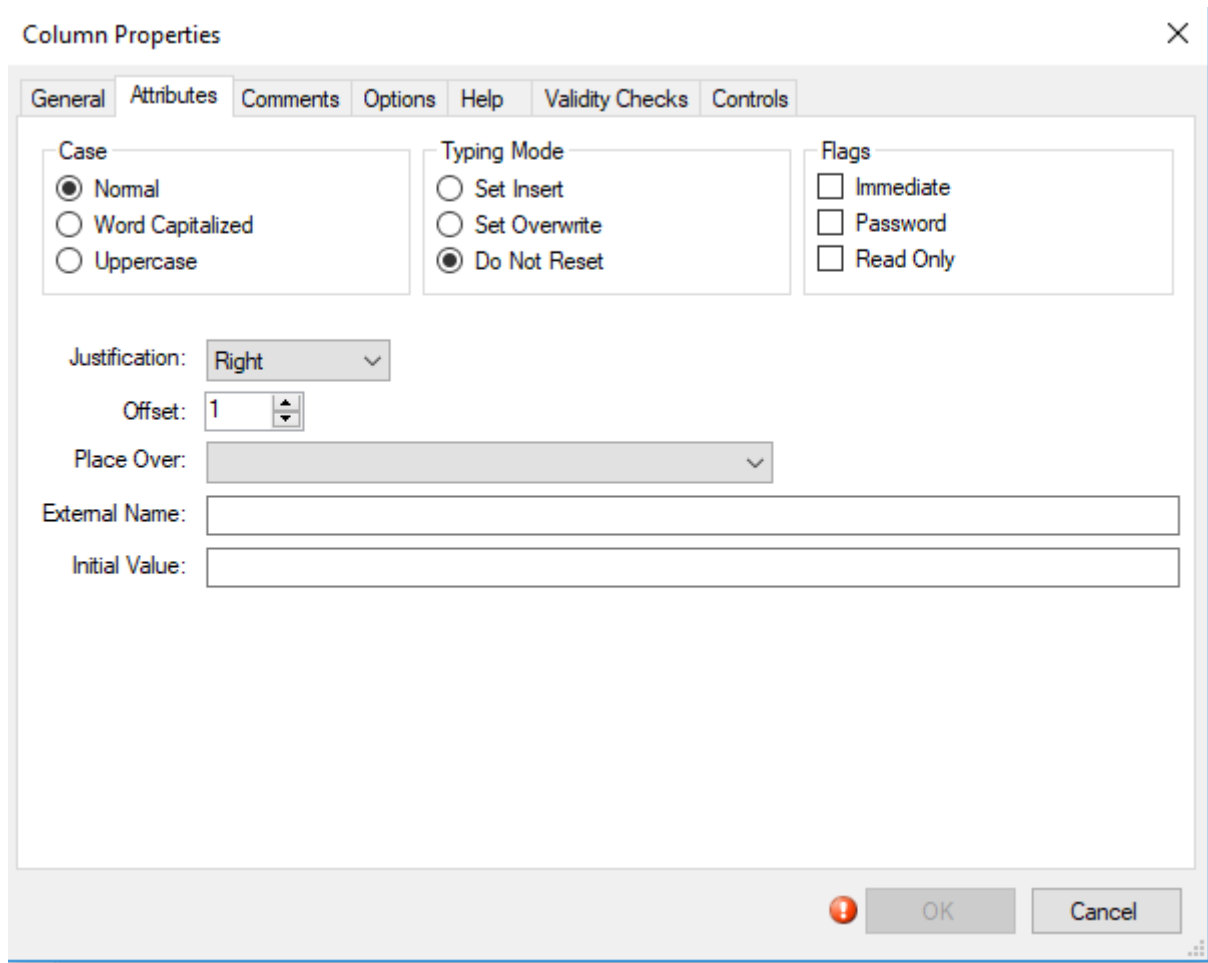


Slika 3.3. Unos općenitih svojstva stupcu

Nakon kartice *General* nastavlja se na karticu *Attributes* gdje se može uočiti polje *Case* i unutar njega načine zapisivanja teksta pa ukoliko postoji neki naziv kojeg treba pisati velikim početnim slovom, označi se *Word Capitalized*, tako da kasnije korisniku automatski olakša posao u slučaju zabune i promjeni malo početno slovo u veliko. *Normal* je za uobičajeno i *Uppercase* za sva velika slova. Ostali podaci do *Initial Value* ne predstavljaju neku važnost, može se jedino poravnanje staviti po svojoj želji, npr. ja ću stavljati centralno poravnanje (*Justification: Center*).

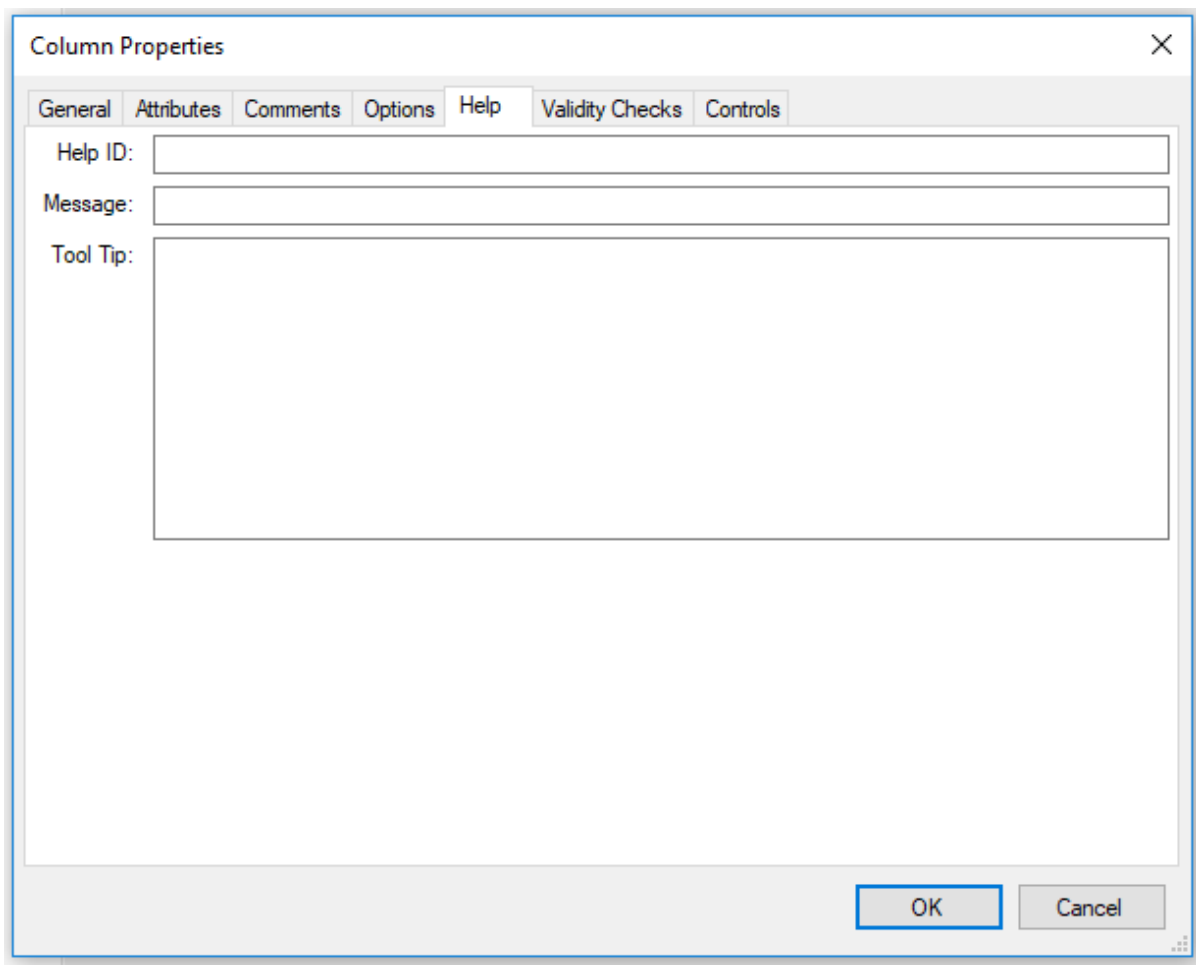
Initial Value popunjavamo ukoliko imamo neku početnu vrijednost, u mom slučaju to će kasnije u tablicama biti atribut datum_računa, a zapisujemo na sljedeći način: za datum

DATE() , ako želimo današnji datum TODAY(), za vrijeme CLOCK() ili neki drugi podatak koji se ne mijenja ili se mijenja vrlo rijetko.



Slika 3.4. Unos atributa određenom stupcu

Sljedeća kartica koja se popunjava je kartica *Help. Tool Tip* može korisniku ostaviti neku kratku poruku („oblačić“) od 5 sekundi kako bi znao na koji način popuniti određeno polje. Na primjer ukoliko se stavi određeni uzorak (engl. *Pattern*) u kartici *General* (*broj telefona*, *OIB*, format za unos datuma...), korisnik neće odmah znati o kojem je točno uzorku riječ, a taj oblačić će mu biti pomoć za brže upisivanje podataka.



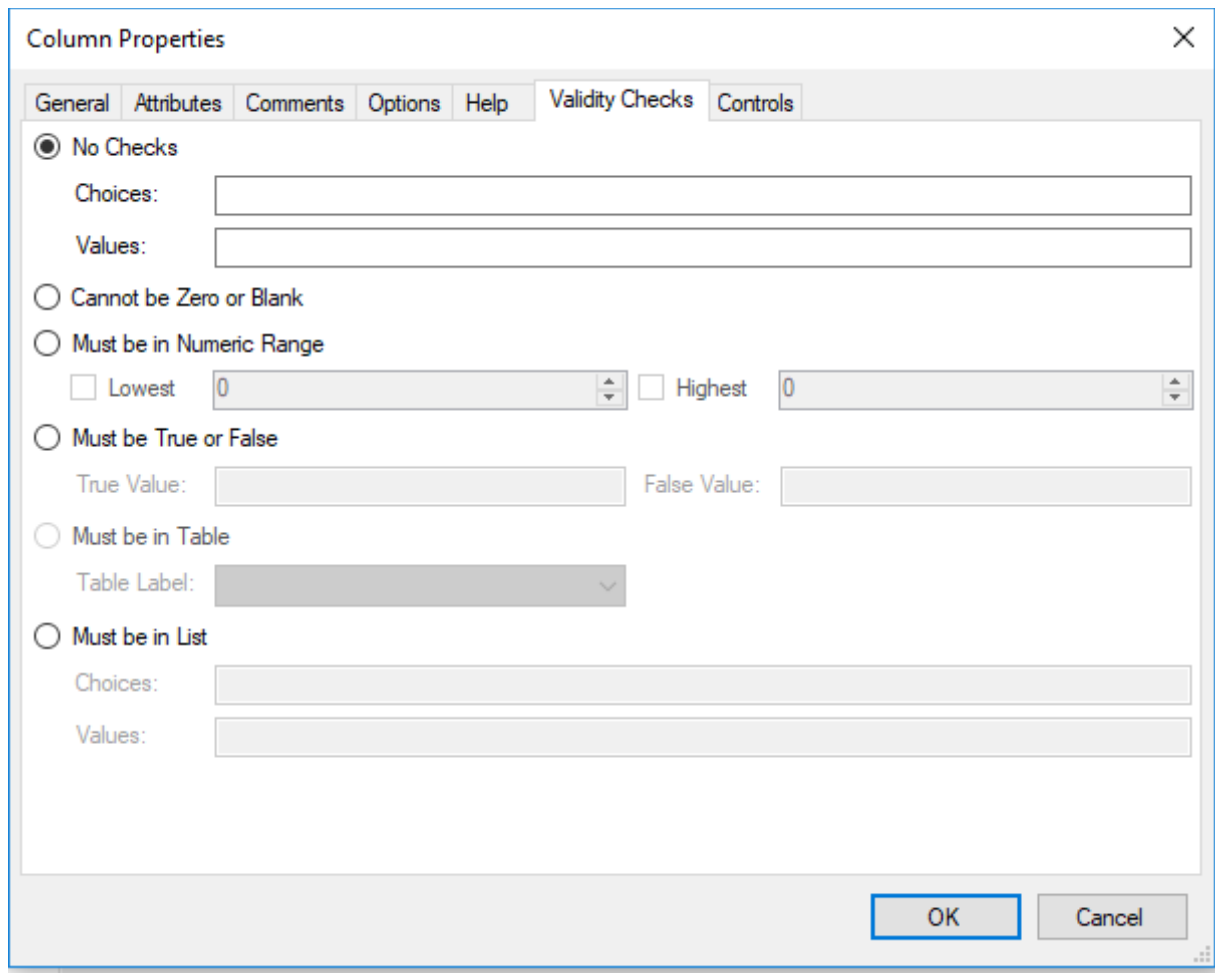
Slika 3.5. Pomoć pri unosu stupca

Kartica *Validity Checks* daje razne opcije odabira. Za primarne ključeve uvijek odabrati *Cannot be Zero or Blank* jer taj podatak nikada ne smije poprimiti *null* vrijednost. Za attribute koji nisu od posebne važnosti staviti *No checks*. Atributu s nekim rasponom brojeva dodjeliti treću opciju i uz nju označiti koje vrijednosti se uzimaju u obzir.

Must be True or False se koristi kod provjere dostupnosti određenog izloška (tablica IZLOŠCI, atribut *dostupnost*) gdje se kao *True Value* stavlja *Dostupno*, a kao *False Value*, *Nedostupno*. *Must be in Table* se označi tek kada se kreiraju sve veze, a pomaže pri određivanju neispravne veze (opciju neće biti moguće odabrati). Opciju odabiremo kod

tablica s vanjskim ključevima, slabim tipom entiteta i agregacijom. Opcija prikazuje iz koje tablice dolazi određeni ključ.

Ako postoji određena lista unutar atributa, onda će se označiti *Must be in List* i upisati *Choices* i *Values* s opcijama koje se mogu odabrati (5|6|7A|KON|ZIMSKI, tablica STAND, atribut *oznaka_paviljona*).

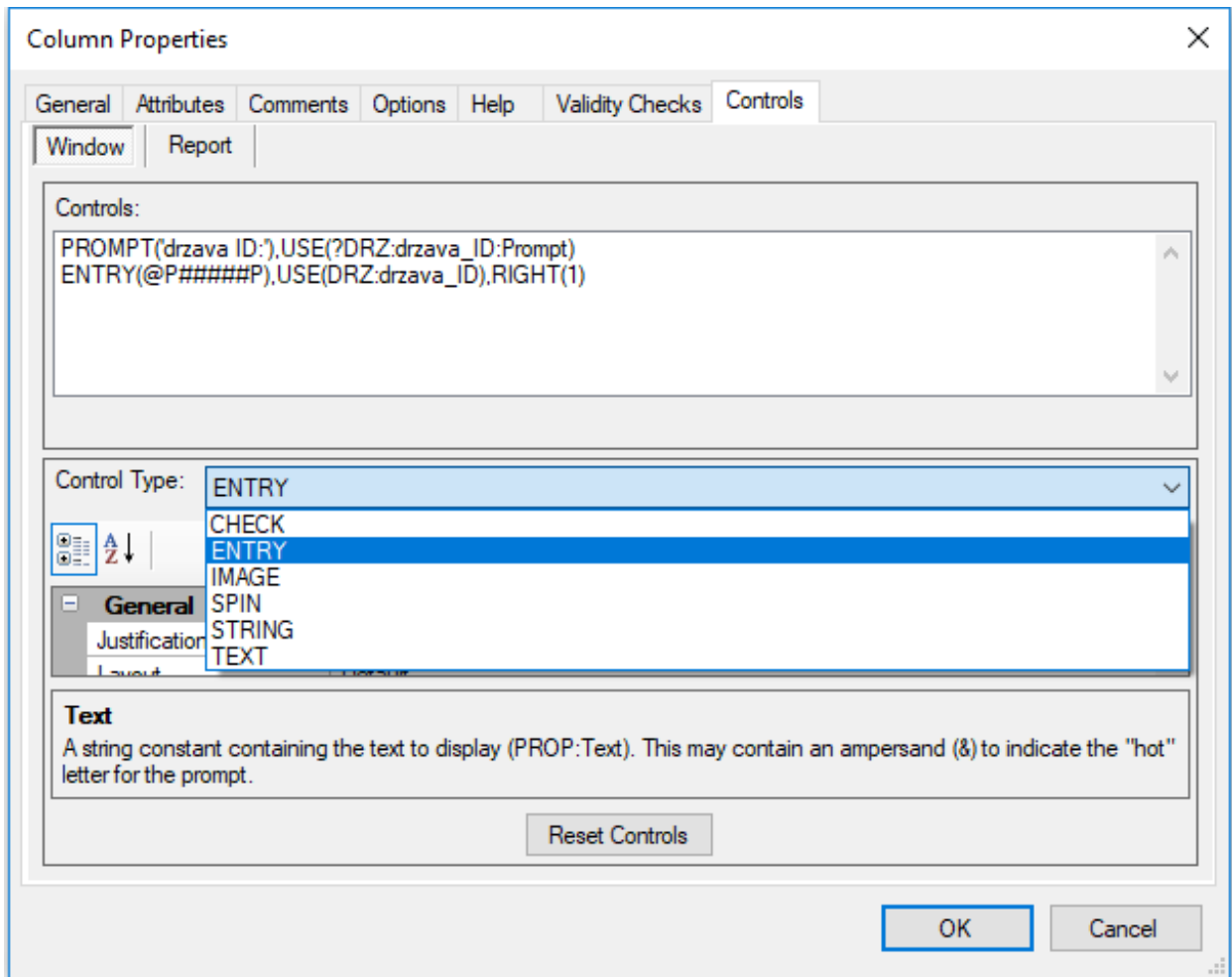


The image shows a 'Column Properties' dialog box with the 'Validity Checks' tab selected. The 'No Checks' radio button is selected. Below it are two empty text boxes for 'Choices' and 'Values'. Other options include 'Cannot be Zero or Blank', 'Must be in Numeric Range' (with 'Lowest' and 'Highest' set to 0), 'Must be True or False' (with empty 'True Value' and 'False Value' fields), 'Must be in Table' (with a 'Table Label' dropdown), and 'Must be in List' (with empty 'Choices' and 'Values' fields). 'OK' and 'Cancel' buttons are at the bottom right.

Slika 3.6. Provjera ispravnosti stupca

U posljednjoj kartici *Controls* može se označiti kojem tipu polja pripada atribut. *CHECK* je za odabir opcija TRUE or FALSE s uvjetom da sami unosimo njihove vrijednost (DA/NE), *ENTRY* za unos kratkog teksta, *IMAGE* za unos slike, *SPIN/LIST* za odabir elementa liste kroz koju se može listati (vrti se pomoću strelica gore dolje i odabire željeni element),

OPTION za pritisak na jednu od opcija, STRING za unos znakova, TEXT za unos tekstualnog polja.

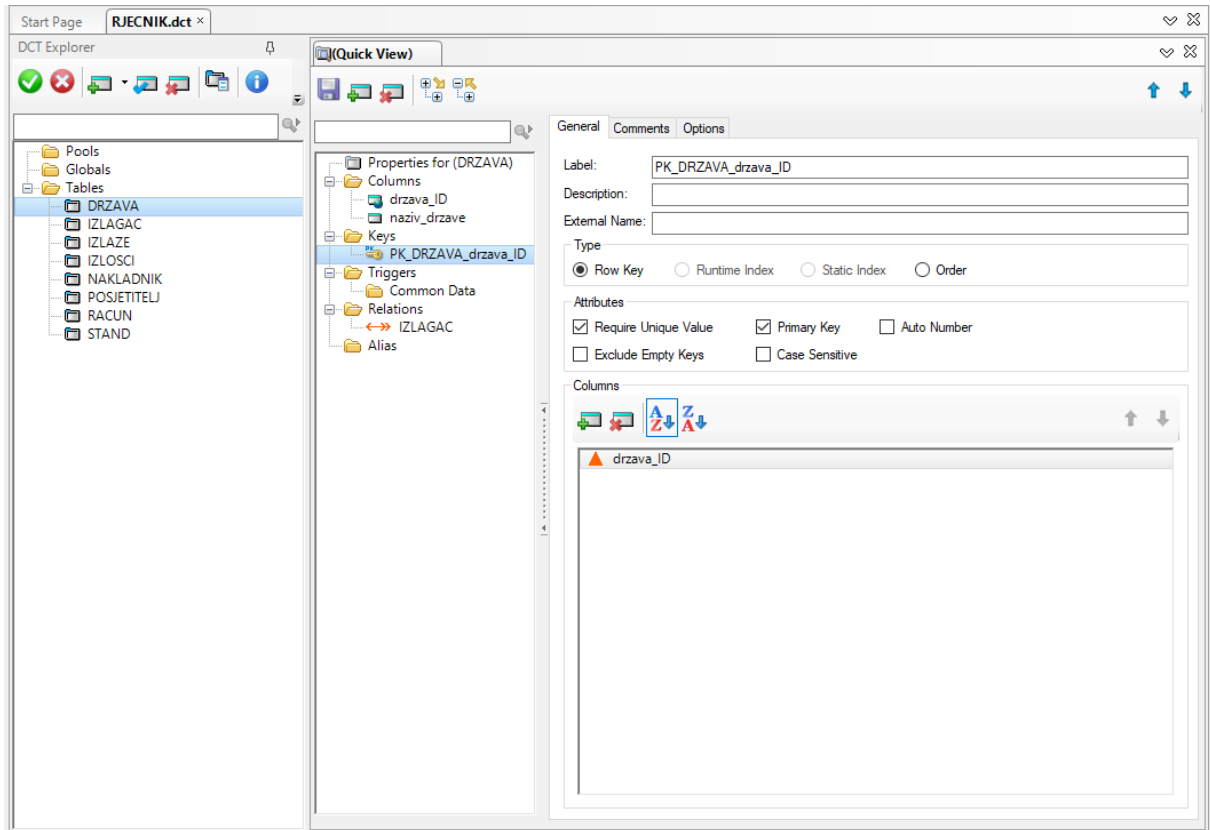


Slika 3.7. Upravljanje stupcima

3.1.3. Određivanje ključeva

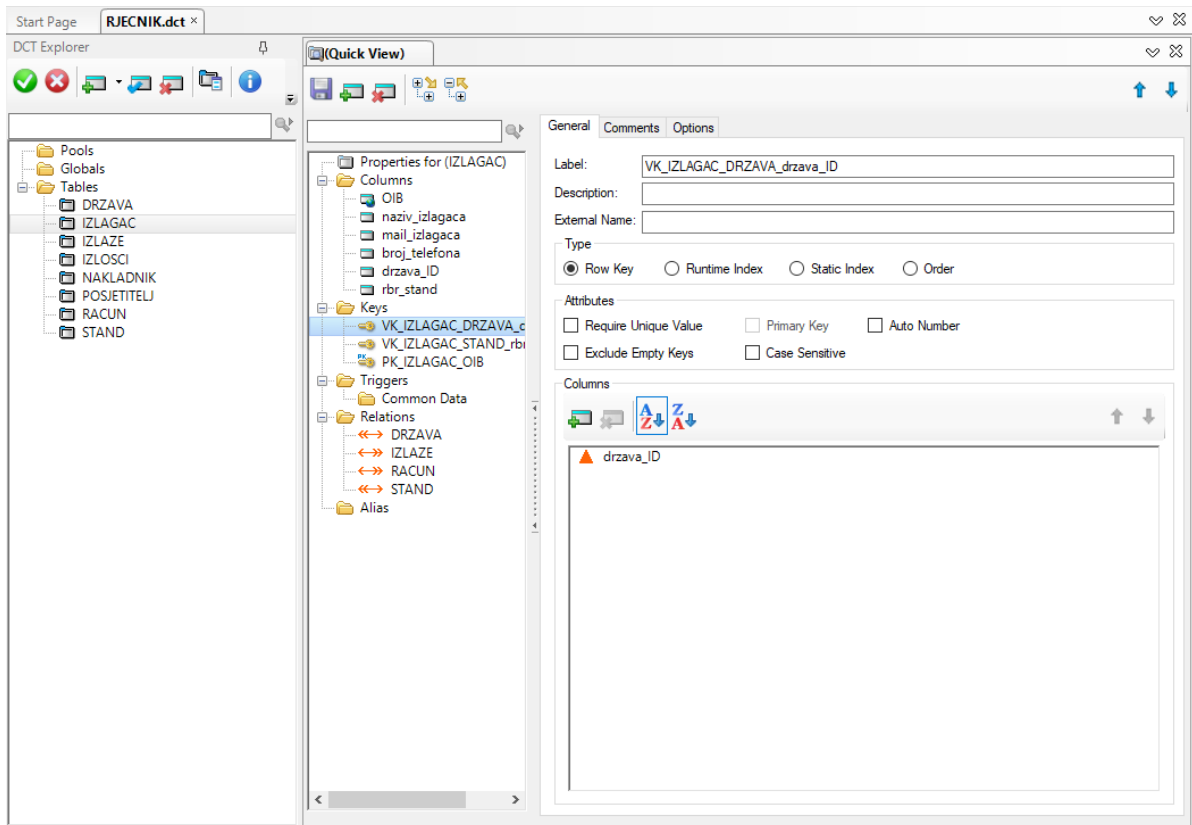
Nakon što se naprave sve tablice i pripadajući stupci, mogu se odrediti primarni, vanjski i ključevi za sortiranje.

Prva je tablica DRZAVA gdje se dodaje samo primarni ključ *drzava_ID*. Dodaje se ime ključa, tip i *Attributes*, a nakon toga ga treba izabrati u polju *Columns*.

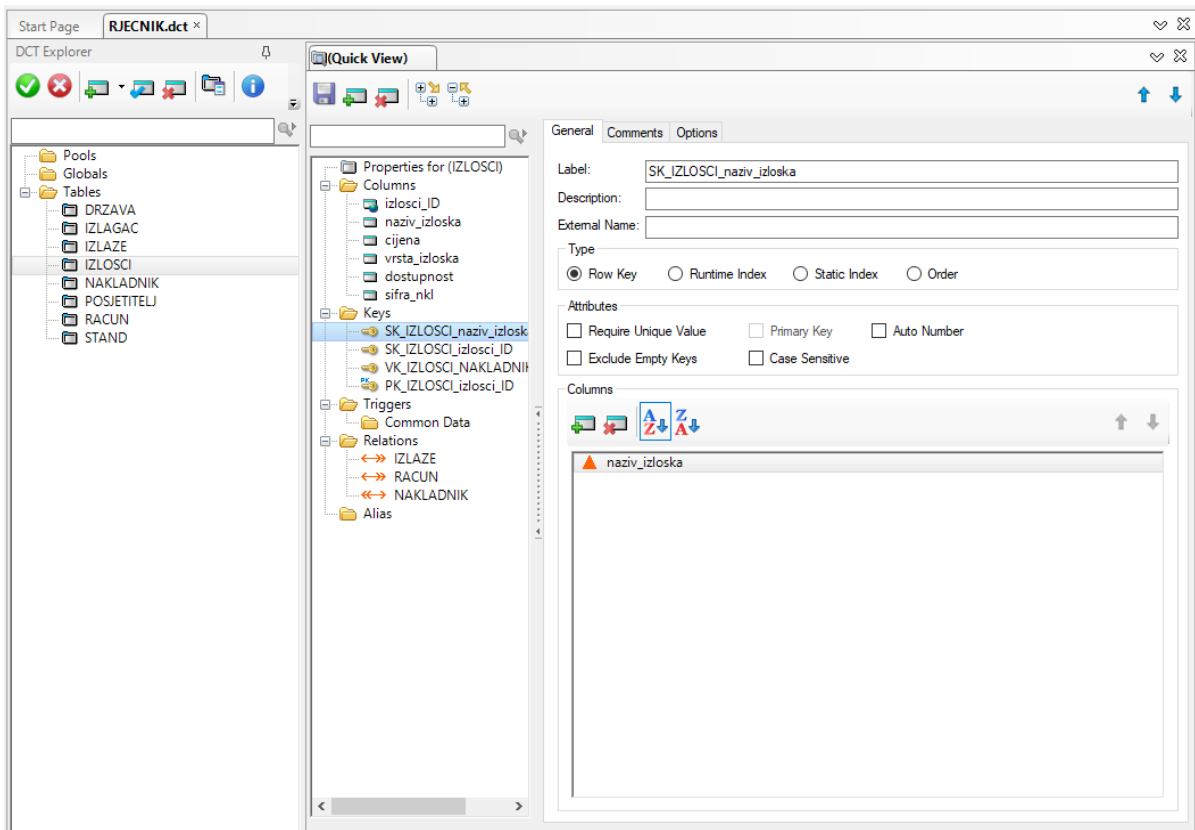


Slika 3.8. Dodavanje primarnog ključa

Na isti način dodaju se ključevi u tablicama POSJETITELJ, NAKLADNIK, STAND, IZLAGAC i IZLOSCI, sa jednom razlikom da se u tablicama IZLAGAC i IZLOSCI nalazi vanjski ključ i ključ za sortiranje. Vanjski ključ i ključ za sortiranje dodaju se slično kao primarni, samo tu se ne bira opcija pod poljem *Attributes*, nego se ostavlja prazno, a na kraju se dodaju stupci. Ključ za sortiranje može se dodati na bilo kojem retku koji nije vanjski ključ kako bi mogli brže pristupiti traženom podatku.



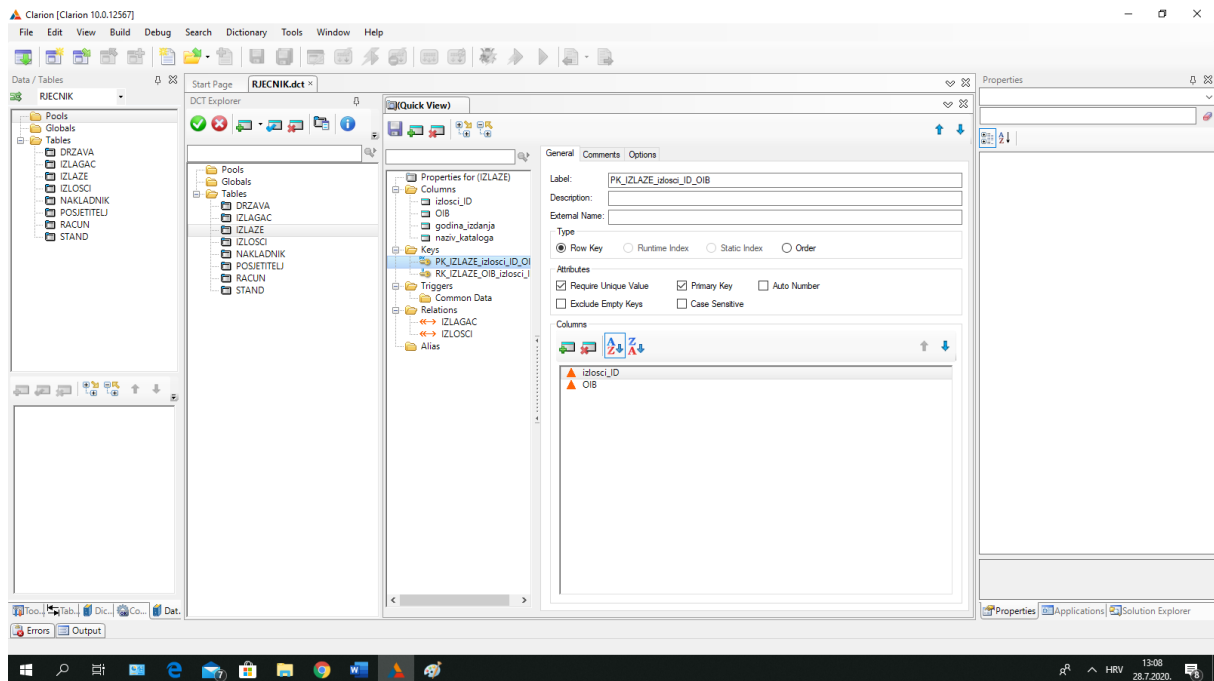
Slika 3.9 Dodavanje vanjskog ključa



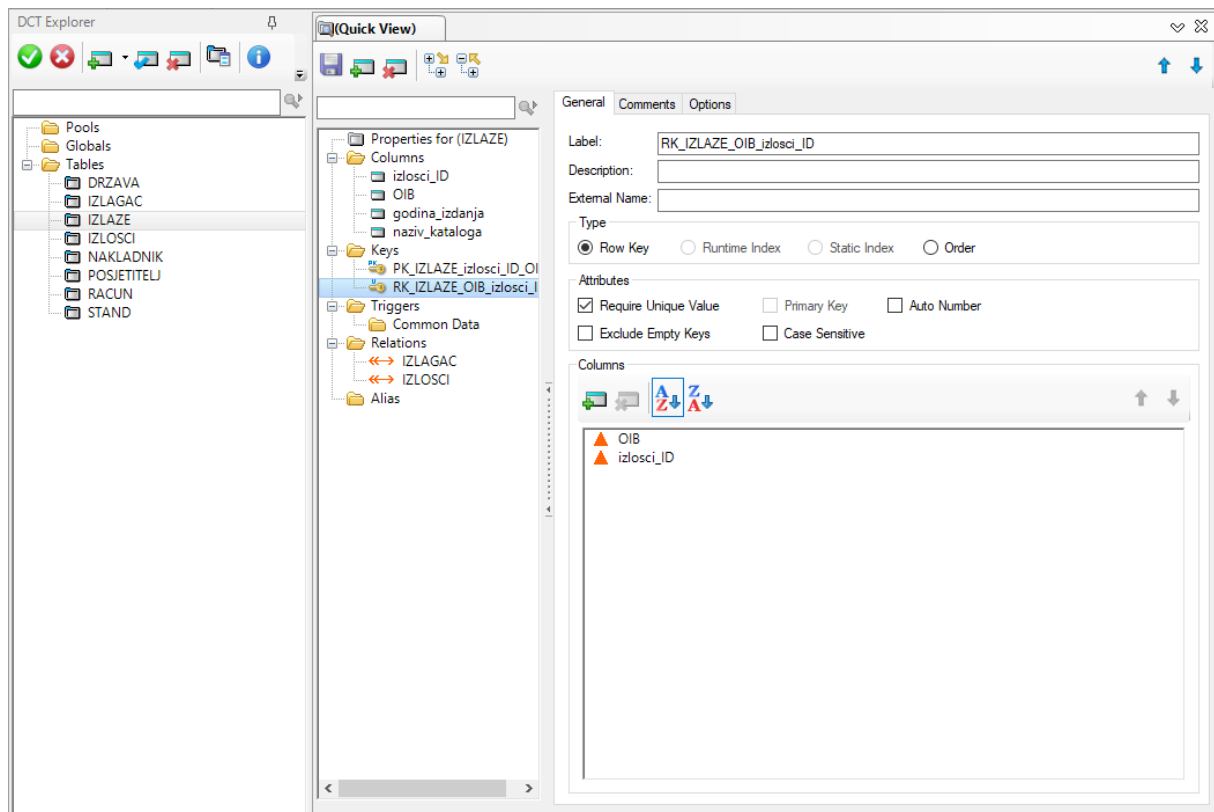
Slika 3.10. Dodavanje ključ za sortiranje (SORT KEY)

Preostaju još slabi tip entiteta (RACUN) i agregacija (IZLAZE). Kod tablice RACUN unutar polja *Columns* treba odabrati dva primarna ključa koja će se koristiti i to prvo ključ jakog tipa entiteta, a tek onda ključ slabog tipa entiteta.

Agregacija ima novu vrstu ključa, a naziva se rezervni ključ agregacije. Primarni ključ agregacije sastoji se od primarnog ključa tablice IZLAGAC i primarnog ključa tablice IZLOSCI. Prvo se dodaje primarni ključ gdje u polju *Columns* opet biramo oba ključa, a onda se dodaje i rezervni ključ, tako da treba izabrati oba ključa, samo u obrnutom redoslijedu. U agregaciji su oba primarna ključa od jednake važnosti, stoga nema prednosti prilikom dodavanja primarnog i rezervnog ključa.



Slika 3.11. Primarni ključ AGREGACIJE



Slika 3.12. Rezervni ključ AGREGACIJE

Kod slabog tipa entiteta rezervni ključ se ne koristi, koristi se isključivo u pojavi agregacije. Nakon što se dodaju svi ključevi, prelazi se na povezivanje tablica.

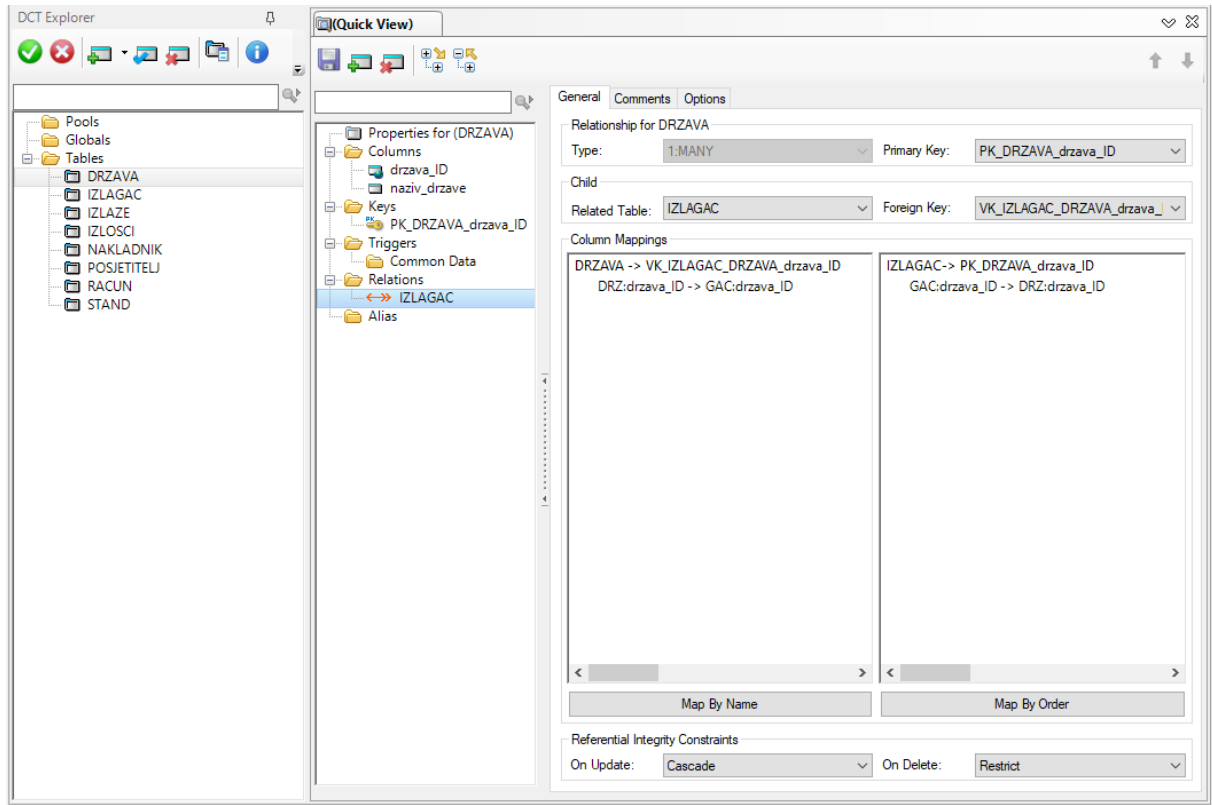
3.1.4. Dodavanje veza između tablica

Veze u Clarionu unose se prema sljedećem redosljed, prvo tablice iz kojih izlazi najviše veza, a nakon toga i sve ostale. Razlog tome je da neke veze kasnije možemo samo promijeniti, ima manje posla i brže se odradi.

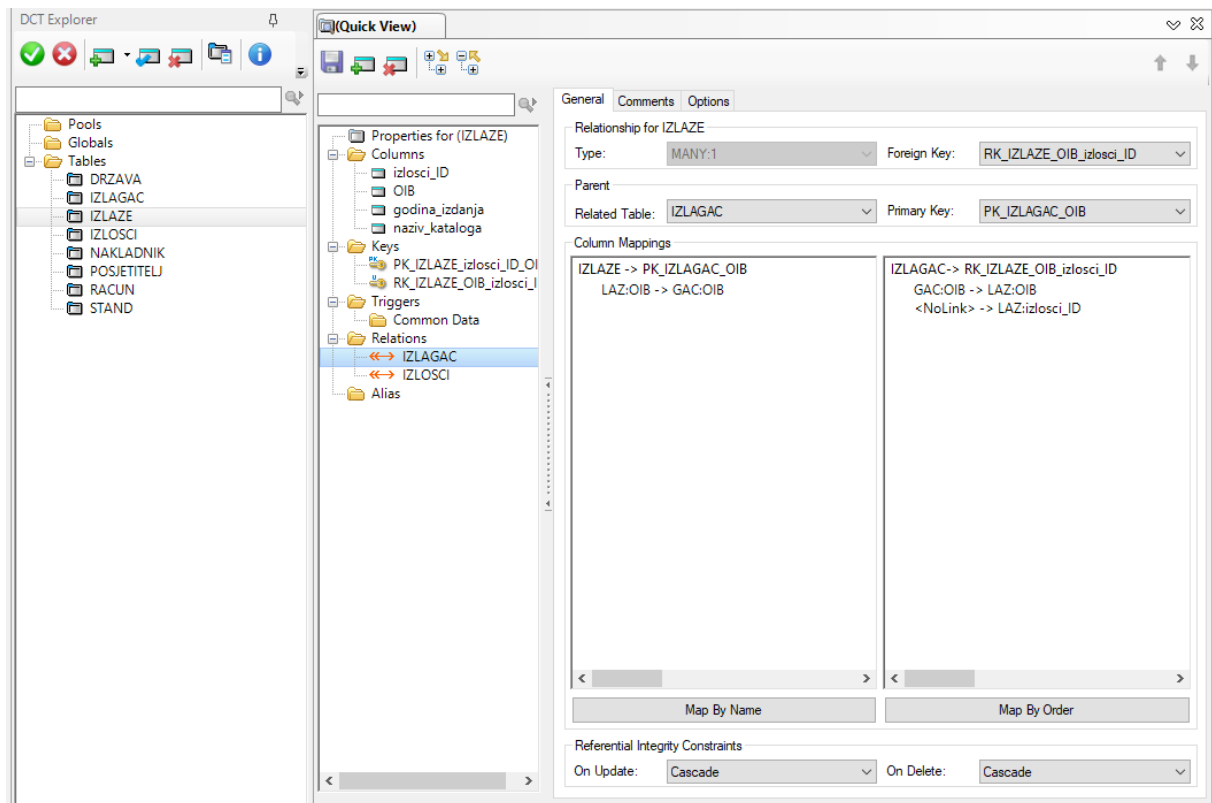
Za početak ćemo analizirati vezu DRZAVA – IZLAGAC koja ima brojnosti (1,M):(1,1), a prilikom stavljanja veze gleda se gornja granica tablice DRZAVA, a ona je MANY pa se prema tome stavlja *Type*: 1:MANY. Primarni ključ je ključ tablice u kojoj vezu radimo. *Related table* je tablica s kojom se povezuje i *Foreign key* je vanjski ključ te tablice. Unutar *Column Mappings* treba povezati ključeve pomoću njihovih imena ili po redosljed (Map by Name ili Map by Order). Najbolje je izabrati prema imenu da se slučajno ne dogodi greška u redosljed tablica. Opcije se unose na sljedeći način ON UPDATE: CASCADE i ON DELETE: RESTRICT što bi značilo da se prilikom ažuriranja ili brisanja primarnog ključa, vanjski ključ prilagođava primarnom ključu i u slučaju ažuriranja se automatski ažurira i on, a

u slučaju brisanja se ništa ne događa. Kod svih veza između jakih tipova se taj dio postavlja na isti način.

Veze kao što su jaki tip entiteta i slabi tip entiteta (i agregacija) poprimaju vrijednost CASCADE na oba dijela iz razloga da se kod povezanih ključeva dogodi ažuriranje i brisanje određenog retka. Ostale postavke postavljaju se na isti način kako je spomenuto u prethodnom tipu veze. Veza slabog tipa entiteta je po gornjoj granici uvijek 1.



Slika 3.13. Veza DRZAVA – IZLAGAC



Slika 3.14. Veza agregacije

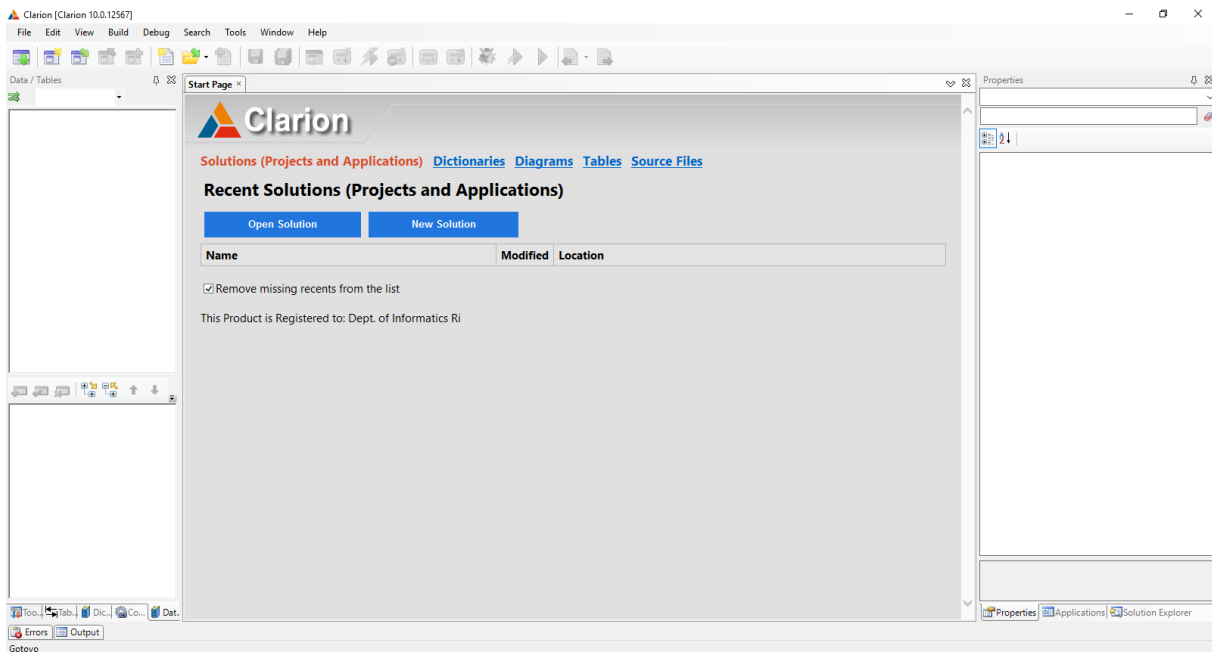
Da bi finalizirali rječnik, nakon što se kreiraju sve veze, postavljaju se ograničenja na sve vanjske ključeve, na cijeli primarni ključ agregacije i na dio primarnog ključa slabog tipa entiteta (onaj koji je posuđen od jakog). To radimo vraćanjem na postavku *Must be in Table* gdje se provjerava mogućnost odabira opcije za sve vanjske ključeve, primarne ključeve agregacije i primarne ključeve slabog tipa entiteta. Ako nije moguće, treba se vratiti i provjeriti veze te po potrebi izbrisati vezu i ponovno napraviti. S time smo završili naš rječnik i možemo se okrenuti izradi aplikacije, odnosno izradi aplikacijskog stabla.

3.2. Izrada aplikacije

Nakon što je rječnik definiran slijedi izrada same aplikacije. Prilikom kreiranja aplikacije moći ćete uočiti da je korištenje koda svedeno na minimum, no javljanje grešaka se ne može izbjeći ni ovdje tako da je najbolje spremati aplikaciju što češće da ne bi došlo do njezinog rušenja.

Kao i u rječniku, na početnom prikazu ekrana se nalaze opcije *Solutions*, *Dictionaries* i ostale mogućnosti odabira, a ovaj put umjesto *Dictionaries* bira se opcija *Solutions* i nakon toga *New Solution*.

U prozoru koji se otvara odabire se *Clarion for Windows, Application* te se unosi ime i putanja gdje će se aplikacija nalaziti. *Auto create project subdir* može se odznačiti i stisnuti *Create*. Sljedeći prozor traži imenovanje procedure, pripadajućeg rječnika, početnog zaslona (*glavniizbornik*) i umjesto Clarionovih koriste se *ABC* predložci. *Application Wizard* se ne koristi jer sami izrađujemo aplikaciju i na kraju stisnuti *OK*.



Slika 3.15 Početni prozor za izradu aplikacije

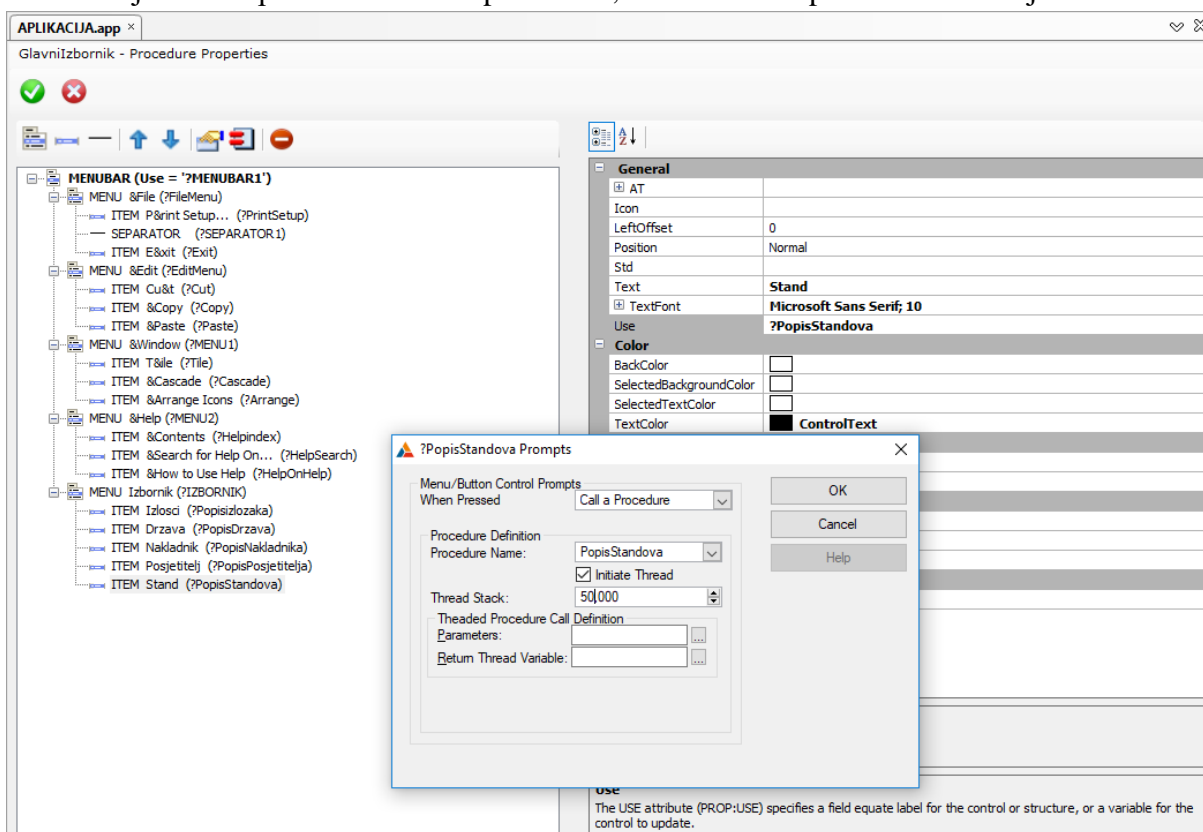
3.2.1. Izrada procedure glavniizbornik te procedura Popis i Ažuriranje

Prva procedura koju radimo je *glavniizbornik*, dvostrukim klikom na nju pod karticom *Defaults* bira se *Default MDI Frame* nakon čega se otvara *Procedure Properties*. Pritiskom na gumb *Actions* otvori se prozor gdje se u polje *Splash Procedure* upisuje *skocniprozor*, to je prozorčić koji se prikazuje na početku i otvoren je određeno vrijeme (5 sekundi). Podesiti

postavke u Date and Time, odnosno označiti sva polja i stilizirati vrijeme i datum prema želji. Izaći iz prozora pritiskom na gumb *OK* i ući u prozor *Window* ili glavni dio aplikacije.

Dizajn prozora će se obaviti na samom kraju, a sada izrađujemo gumbe, polja i ostale mogućnosti.

Desnim klikom na Alatnu traku i pritiskom na *Edit Menu* otvara se prikaz sa padajućim izbornicima. Ovdje se izrađuje *New Menu* i *New Item* (elementa dodajemo onoliko koliko ima tablica, bez agregacije i slabog tipa entiteta). Novi meni imenujemo, npr. *Popis*, a elemente menija imenujemo prema tablicama: *Drzava*, *Nakladnika*, *Posjetitelja*, *Standova*, *Izlagaca*, *Izlozaka*. *Use* upisujemo na sljedeći način: *?PopisDrzava*, tako redom sve tablice. Kako bi se procedure mogle koristiti, treba ih pozvati desnim klikom na *Item*, odabrati *Actions* i u novootvorenom prozoru odabrati *When Pressed: Call a Procedure*, zadati ime proceduri koje mora biti jednako polju *Use*, odabrati *thread* i postaviti ga na 50000 procesnih niti. Na taj način napraviti sve ostale procedure, stisnuti *OK* i spremiti dosadašnji rad.



Slika 3.16. Kreiranje procedura

Na početku u *Application Tree* nalaze se sve dodane procedure tako da se može krenuti s njihovim uređivanjem.

Dvostrukim klikom na proceduru *skocniprozor* otvara se prozor gdje odabiremo *Default Splash Window*, otvara se *Actions* i *timer* se postavlja na 5 sekundi i označava checkbox kako bi se prozorčić zatvorio nakon što korisnik klikne na njega. Dizajn ćemo ostaviti za kraj kao i

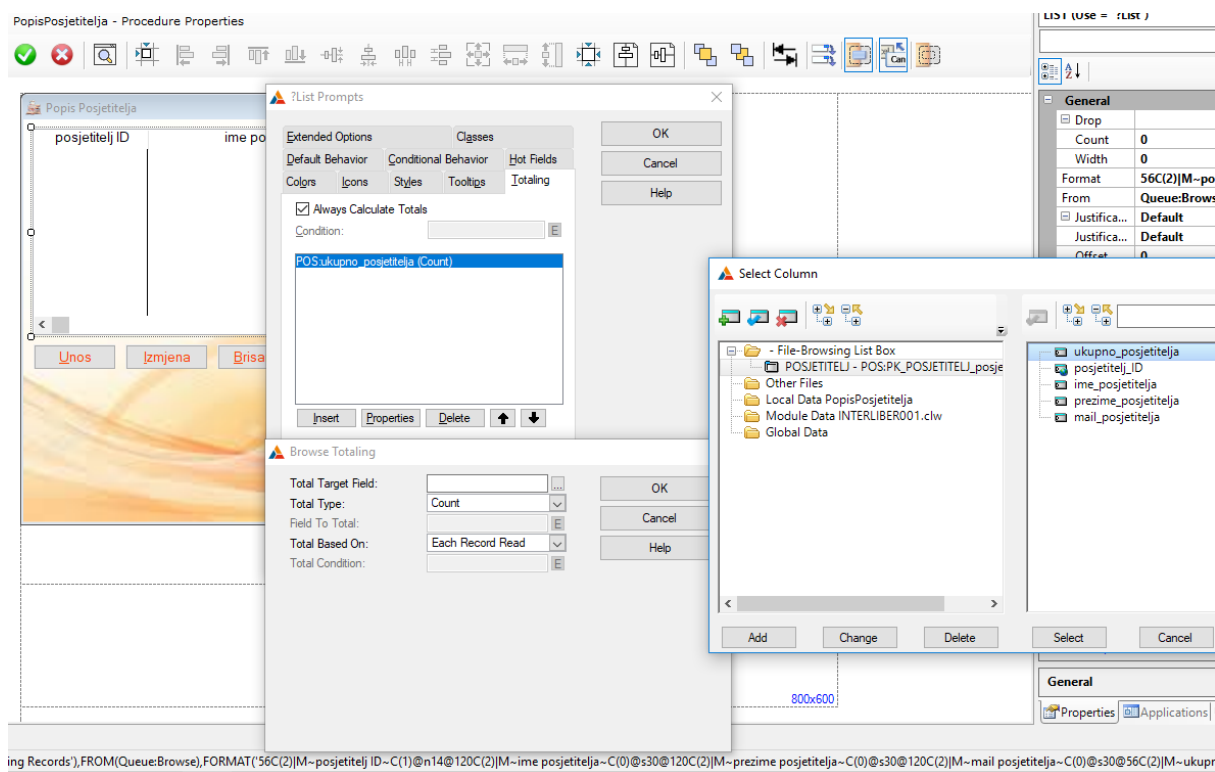
za sve ostale procedure. Za sad smo gotovi sa skočnim prozorom pa možemo krenuti sa jednostavnijim procedurama.



Slika 3.17. skočni prozor

Dvoklik na *Popis Drzava* gdje u prozoru treba odabrati *Browse With Update and Select, Actions* i u *Update Procedure* upisati *Azuriranje Drzava*. Nakon toga otvoriti *Window* i prvo ući u *Data/Tables* dvostrukim klikom na *ToDo* i odabrati tablicu *DRZAVA* te nakon toga ponovno dvokliknuti na tablicu *DRZAVA* i izabrati njezin primarni ključ *drzava_ID*. Desnim klikom na listbox dolazi se do opcije *List Box Formatter* gdje se mogu odabrati stupci koji će biti prikazani, stupce dodati na opciji *Add Field* i centrirati. Na taj način se počinje sa radom na svim ostalim tablicama.

Procedura *PopisPosjetitelja* je posebna po pojavi funkcije *Totaling* na stupcu *ukupno_posjetitelja*. Stupac treba izvući u prozor pored List Box-a i desnim klikom na *List Box* odabrati *Totaling*, upisuje se ime stupca, a tip treba postaviti na *Count* i spremiti.



Slika 3.18. Ukupno

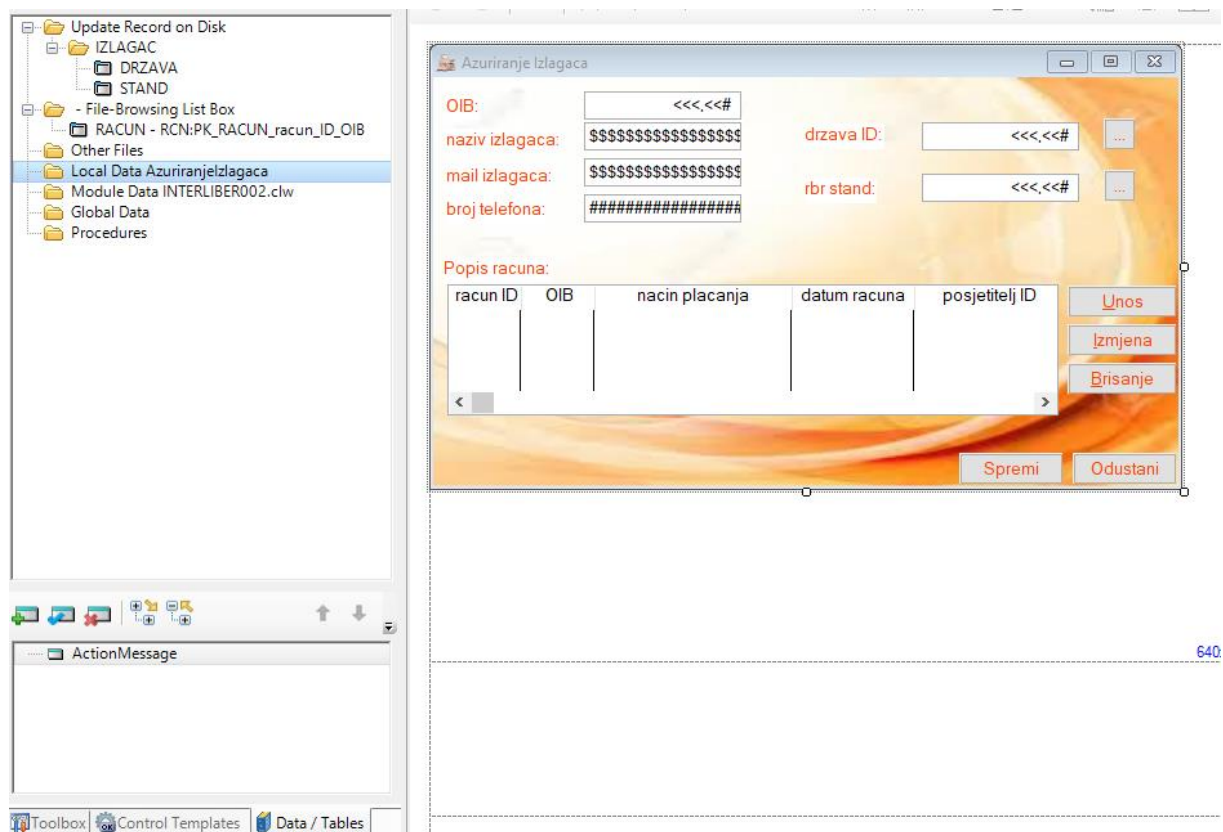
Application Tree stvara novu proceduru *AzuriranjeDrzava* tako da dvostukim klikom na nju možemo odabrati *FORM(Add/Edit/Delete)* u polju *Defaults*. U polju *Actions* možemo promijeniti *Messages and Titles* na hrvatski (Pregled, Unos, Izmjena i Brisanje zapisa), stisnuti *OK* i nastaviti na *Window*. Ovaj put u *Data/Tables* moguće je odabrati samo tablicu *DRZAVA*, bez primarnog ključa i uvući stupce u prozor. Stupac *naziv_drzave* smo u rječniku postavili na tip *LIST*, tako da sada u polju *Properties*, *Count* treba postaviti na 19, onoliko koliko imamo država. Na isti način mogu se kreirati ažuriranja za tablice *NAKLADNIK*, *POSJETITELJ* i *STAND*, osim elemenata kao što je *naziv_drzave*. U jednoj od tablica, kao što je *STAND*, naići ćemo na polje *SPIN* za atribut *oznaka_paviljona* gdje je u *Properties* već sve sređeno, tako da se ne treba brinuti o postavkama za taj dio, mogu se odraditi provjere da se ponište greške.

Ostale procedure će biti malo kompliciranije za srediti pošto su u pitanju vanjski ključevi, ključ agregacije i slabog tipa entiteta.

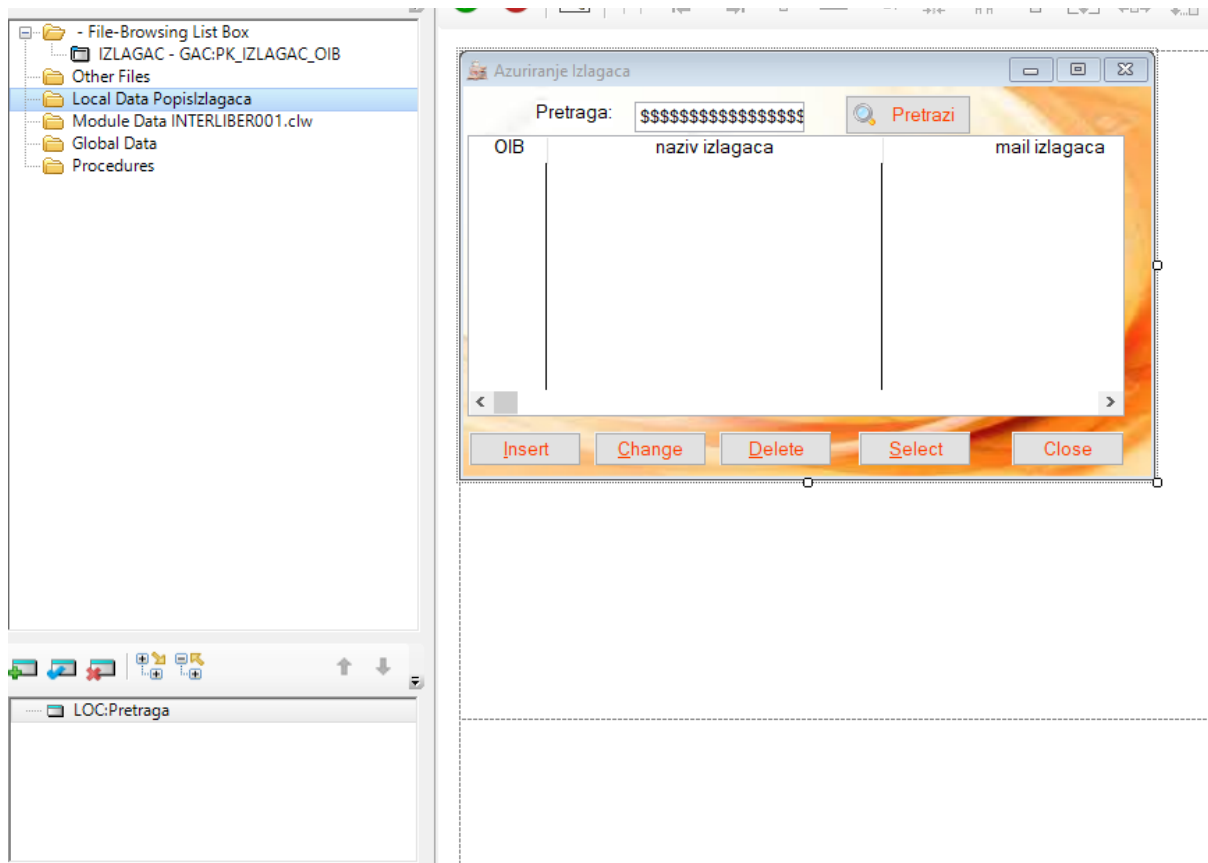
Popis Izlagaca počinje se raditi na isti način kao jednostavne tablice. Nakon toga može se dodati *Pretraga* u *Data/Tables*. Desni klik na *Local Data PopisIzlagac*, *Add Column* (naziv stupca je *LOC:Pretraga*, isto je kao stvaranje atributa u rječniku, ali ovdje se ne dira ništa drugo osim naziva), zatim kliknuti *Create* i povući atribut u prozor. Promijeniti ime atributa u *Pretraga*, kliknuti desnim klikom na *ListBox*, odabrati *Actions*, *Default Behavior* i u *Locator Behavior* u prvom polju odabrati opciju *Entry*, odabrati *override default locator behavior* i u polju *New Locator Behavior* odabrati polje *?LOC:Pretraga*. Sljedeće što treba napraviti je ubaciti gumb *Traži* pored polja *Pretraga*, a njega treba povući iz *Toolbox-a (Button)* i promijeniti naziv gumba. Preostalo je da desnim klikom na gumb, odaberemo opciju *Embeds* i

dvoklikom na *Selected (Source)* upisujemo naš prvi kod u aplikaciji : *ThisWindow.Reset(1)*. Spremiti cijelu aplikaciju i krenuti sa procedurom *AzuriranjeIzlagaca*. Unutar *Embeds* računa se i ukupna cijena izloška za tablicu RACUN (ukupno = cijena_izloska*kolicina).

Početak je isti kao u ostalim ažuriranjima, povući stupce u prozor i postaviti Lookup na vanjske ključeve. Desni klik na polje *drzava_ID*, *Actions* gdje se izvršava upis u polja, *Lookup Key*, *Lookup Field*, i *Lookup Procedure*, iz povezane tablice (znači tablica DRZAVA). Ponovno sve spremiti prije korištenja *Control Templates* odakle se povuče gumb *Field Lookup Button*. U *Actions* dolazimo do polja *Control with Lookup* i biramo primarni ključ iz trenutne tablice. Isti je postupak za vanjski ključ *rbr_stand*. Spremiti i dodati novi *List Box*, pomoću *Control Templates*, *Browse Box* povučemo u prozor i u *Properties* odabiremo horizontalne klizače. Desnim klikom na list box dodati slabi tip entiteta RACUN i njegov primarni ključ te ubaciti pripadajuće atribute. Preko polja *Range Limit Field* u *Actions* postaviti da se računi prikazuju samo za izlagače, a bira se zajednička tablica (odnosno dio primarnog ključa, u našem slučaju OIB). Nakon toga treba izabrati *File Relationship* i originalnu tablicu IZLAGAC.

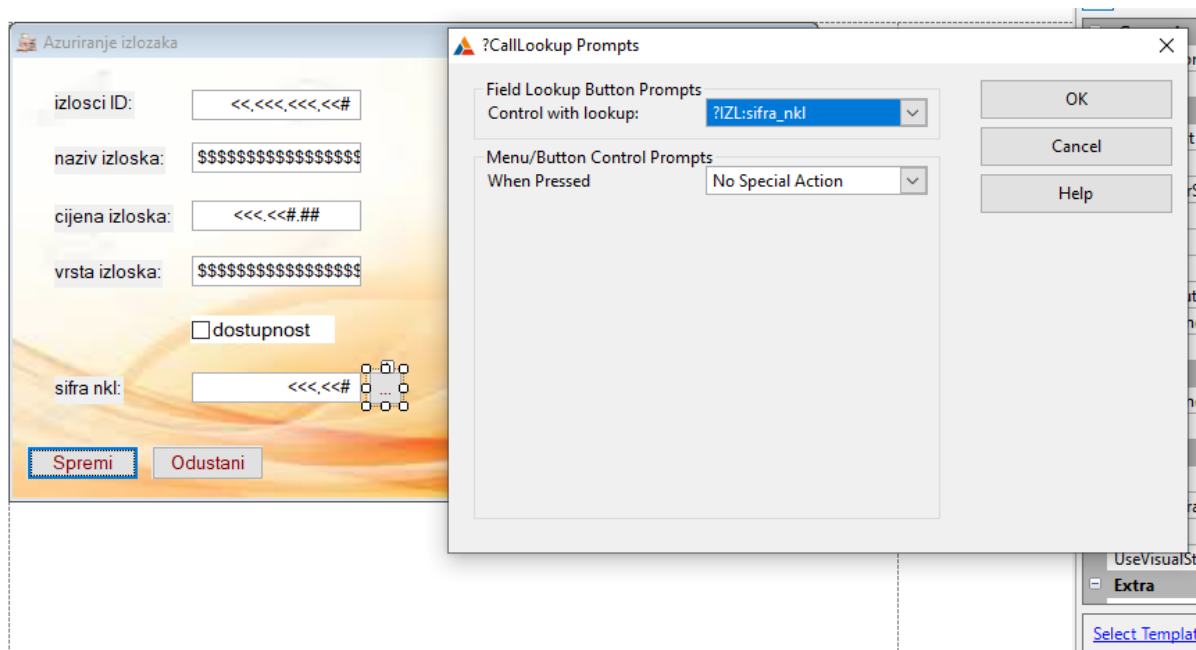


Slika 3.19. Azuriranje Izlagaca i Popis Racuna



Slika 3.20. Pretraga u Popis Izlagaca sa gumbom Pretrazi

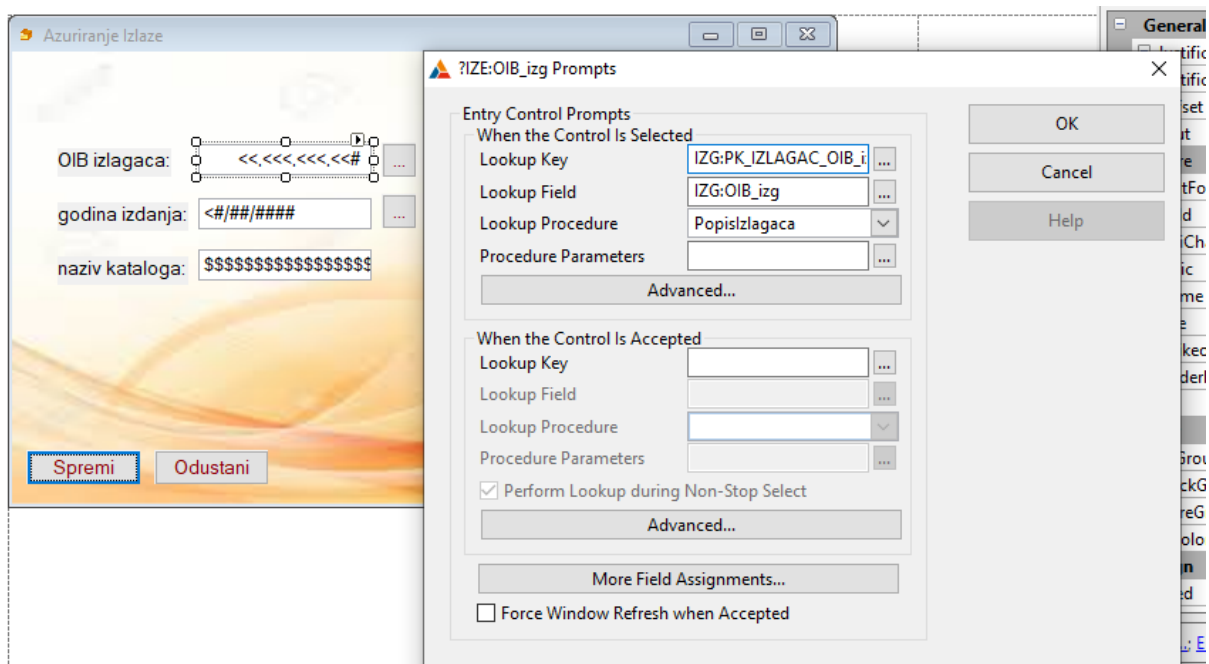
Sljedeća procedura je *Azuriranje Racuna*, nakon svih odrađenih koraka, uz povučene attribute iz drugih tablica treba napraviti *Lookup* i gumb za *Lookup*. Procedura je gotova.



Slika 3.21. Gumb za lookup

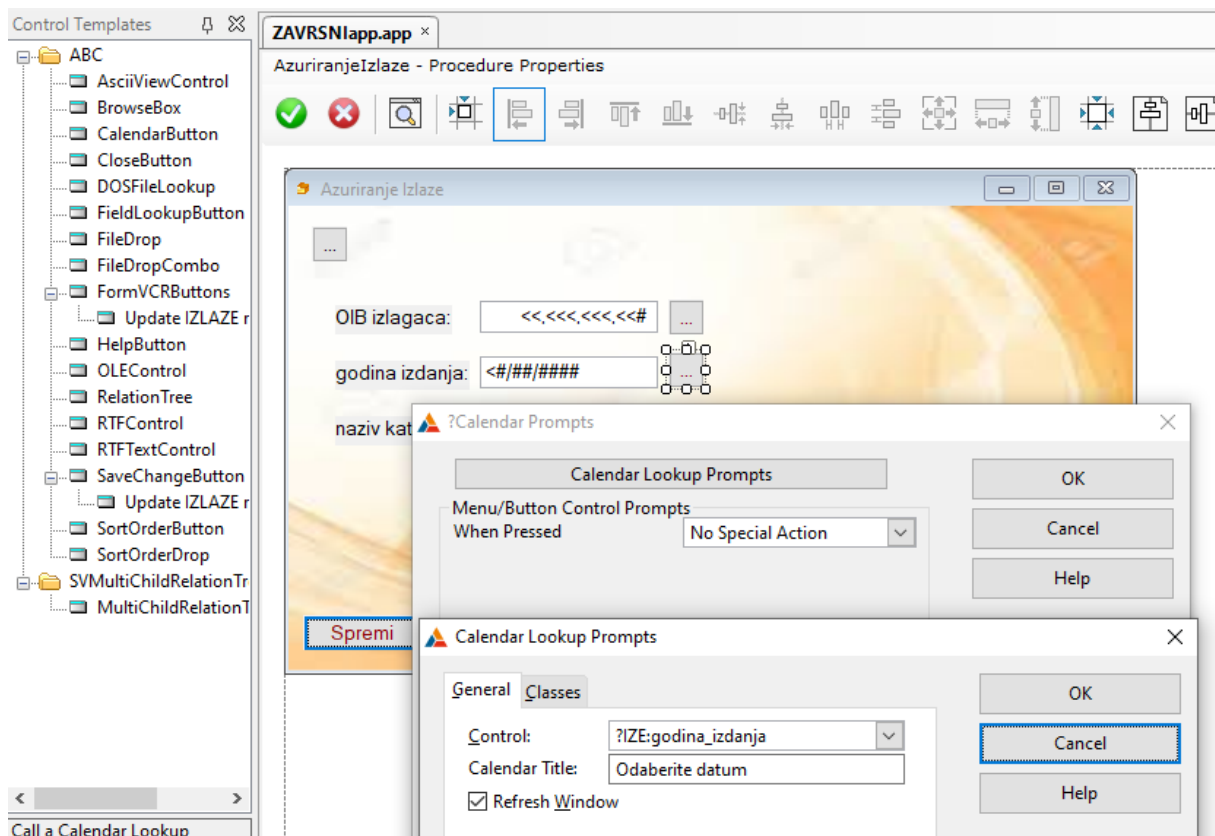
Procedura *PopisIzlozaka* trebat će za ubacivanje agregacije i pripadajuće procedure *AzuriranjeIzlaze*, procedura predstavlja Katalog za određenu godinu održavanja sajma koji sadrži podatke o izlošcima i kod kojeg izlagača se izlošci nalaze. U proceduri *PopisIzlozaka*, pored List Boxa s izlošcima dodati string s natpisom *Popis Izlaze* i dodajemo Browse Box koji će sadržavati podatke koji nisu primarni ključevi. Listbox-u moramo odrediti tablicu i pripadajući primarni ključ te *Range Limit*. Uz *Listbox* treba postaviti i gumb za ažuriranje pa ih povlačimo iz *Control Templates*. Desnim klikom na gumb Insert postaviti proceduru *AzuriranjeIzlaze* i nastaviti u *Application Tree* za uređivanje procedure.

Proceduri ne treba dodati primarni ključ tablice *IZLOSCI*, već samo tablice *IZLAGAC* i napraviti *Lookup* polje i gumb.



Slika 3.22. Lookup polje

Dodati i gumb *Calendar Button* iz *Control Templates* te desnim klikom na njega odabrati *Actions* pa *Calendar Lookup Prompts*. Odabrati atribut iz tablice po kojem želimo da se kalendar ažurira i gotovo.



Slika 3.23. Dodavanje gumba za kalendar

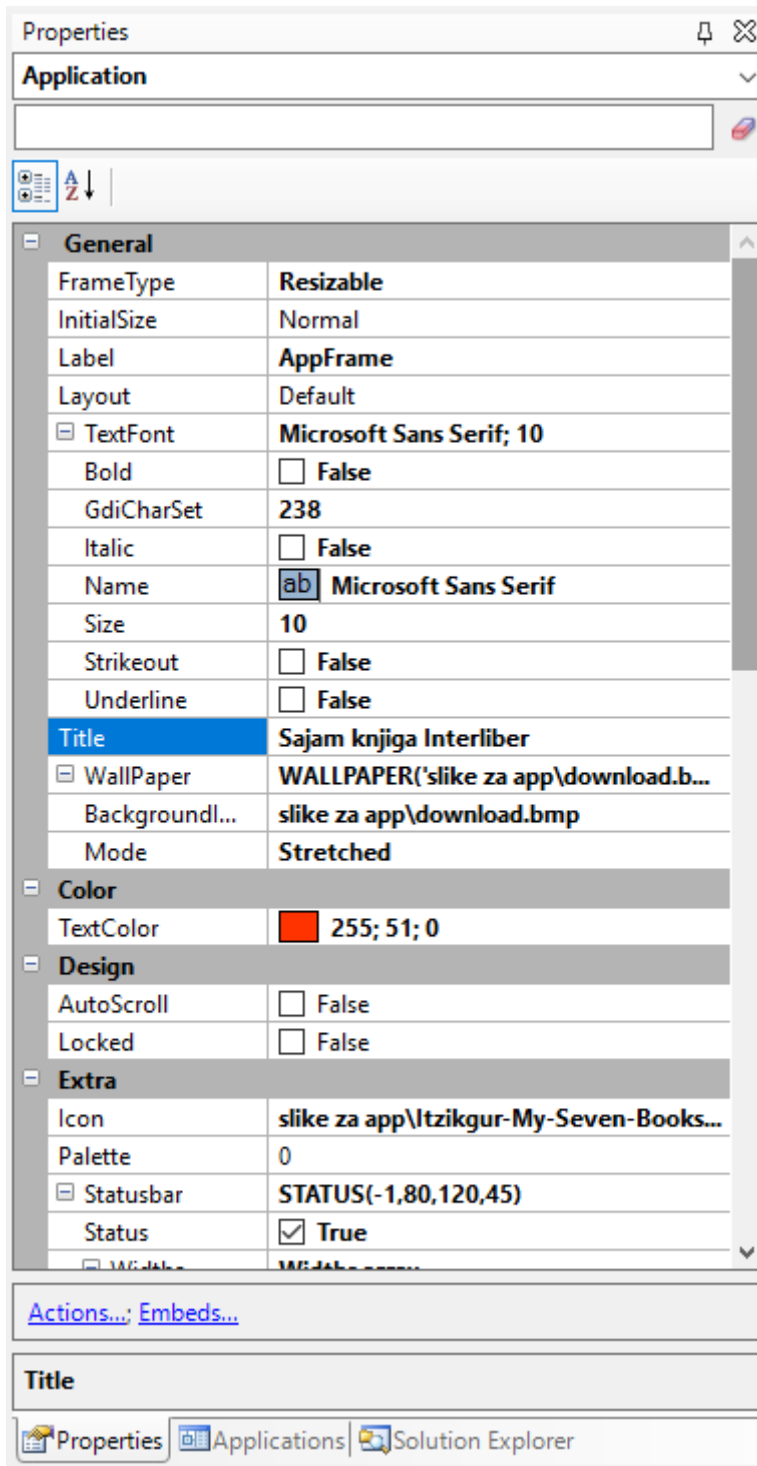
Prošli smo kroz sve procedure i istaknuli njihove najbitnije dijelove, a ostaje još dizajn aplikacije.

3.2.2. Dizajn aplikacije

Procedure se uređuju uglavnom pomoću polja *Properties* gdje se nalaze funkcije za promjenu fonta, boja, pozadinske slike, promjenu naziva gumba i procedura s engleskog na hrvatski jezik te dodavanje ikona i pozadine. Često se zna desiti da iako smo podesili neki gumb na određenu boju s određenom ikonom, Clarion uporno vraća stare postavke tako da neke postavke ne žele raditi.

Mogu se dodati novi gumbi (Toolbox->Button) i urediti tako da se postavi neka nova ikona (*Properties*->*Icon*, format .ico) iz predloženih opcija ili se mogu učitati vlastite ikone, a može se i damo promjeniti boja gumba i teksta na gumbu. U glavnom izborniku učitana je nova alatna traka i novi gumbi i promijenjen izgled. Na gumbe u *Actions* treba postaviti pozivanje određene procedure kako bi gumbi izvršavali svoju funkciju.

Skocni prozor se uređuje tako da za pozadinu u Properties odaberemo boju po želji (ili sliku u formatu .bmp), font, stavimo natpise koje želimo da se prikazuju pomoću String iz Toolbox-a i na kraju sliku pomoću Image-a iz Toolbox-a (format slike mora biti .gif). Na taj način uređujemo sve procedure, a kasnije i izvještaje.



Slika 3.24. Svojstva

3.3. Izvještaji

Izvještaji su ispisi određene skupine podataka u određenom formatu (A4) i uređeni su po želji korisnika. Prvo radimo jednostavni izvještaj na tablici IZLAGAC. Nalazimo se u *Application Tree*, ulazimo u *glavniizbornik* i desnim klikom na alatnu traku odabiremo *Edit Menu*. Ovdje će se dodati novi meni pod nazivom *Izvjestaj* te ispod njega i elementi *IzvjestajIzlagac* te *IzvjestajRacun*. Za elemente pozivamo i procedure koje se pojavljuju u *Application Tree*.

Napraviti dvostruki klik na prvu proceduru *IzvjestajIzlagac* i odabrati *Report (paper size A4 – Portrait)*, otvara se novi prozor gdje se *Actions* više uopće ne koristi, ali u Window treba ući kako bi prozor za učitavanje preveli na hrvatski i uredili po želji. Spremiti sve i kliknuti na gumb *Report* ispod gumba *Actions* i *Window*. U kartici *Data/tables* odabrati tablicu i pripadajući primarni ključ. *Page Header* se koristi za kreiranje naslova i ostalih dodataka ako želimo, a *Page Footer* za prikaz datuma i vremena našeg izvještaja (*BrowseBox->ReportDateStamp* i *ReportTimeStamp*). Zadnje ostaje kartica *Detail* gdje treba povući Stringove koje imenujemo prema podacima koje isto tako treba povući iz tablice IZLAGAC. Dizajniramo i odlazimo u proceduru *PopisIzlagaca*. Pored *ListBox*-a treba biti gumb za Ispis naših izvještaja. Gumb se povlači iz *Toolbox*-a (*Button*). U polje *Use* se upisuje *?Print*, desni klik na *Actions* i pozove se procedura *IzvjestajIzlagac*.



Slika 3.25. Jednostavni izvještaj

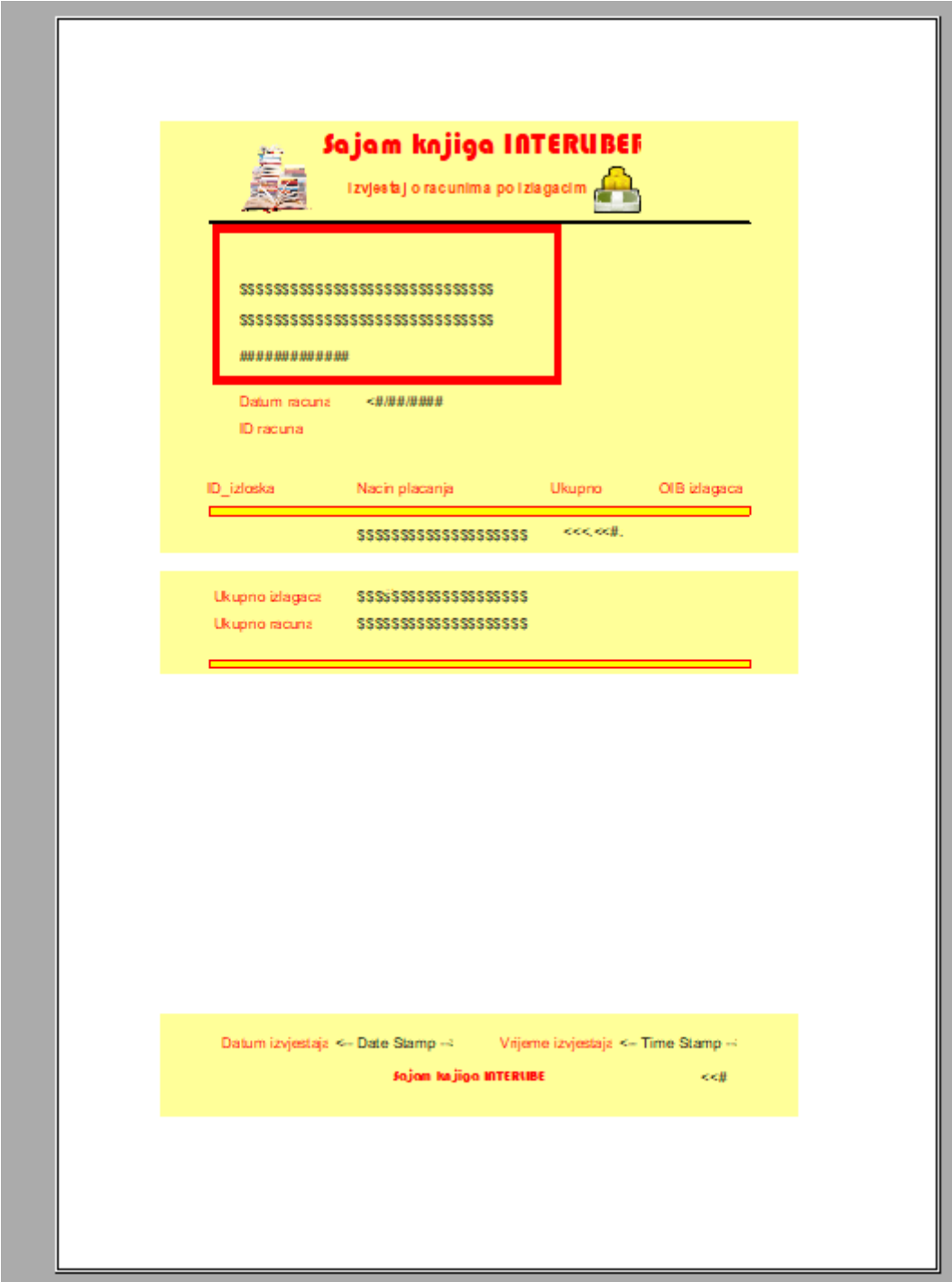
Sljedeći izvještaj je izvještaj pomoću gumba, radi se izvještaj samo za trenutno izabrani izložak iz tablice IZLOSCI. Može se samo kopirati i preimenovati prethodna procedura. Sve

ide prema istom principu, osim što ovdje unutar *Procedure Properties* ulazimo u *Actions* i biramo *Report Properties* gdje pomoću *Range Limit* ograničimo polje za pojedini izložak. Procedura *PopisIzlozaka* ponovno mora imati gumb za ispis izvještaja.



Slika 3.26. Izvjestaj preko gumba

Posljednji izvještaj je složeni, a prikazuju se svi računi po izlagačima. Potrebno je kopirati proceduru, preimenovati i dodati tablice RACUN i IZLAGAC. Ovdje je drugačiji pristup jer u polju *Details* desnim klikom treba odabrati *Surrounding Break* i *racun_ID*, a nakon toga i za *OIB*. Isto tako za obje kartice dodajemo varijable *breakIOIB* i *breakIracunID*. Još treba dodati prijelome za *Group Header* i *Group Footer* u oba slučaja i u njih povući stringove i podatke iz tablice. Na slici se može vidjeti kako to izgleda. Na kraju treba još dodati gumb za ispis i izvještaji su gotovi.



Slika 3.27. Složeni izvještaj

4. ZAKLJUČAK

Završni rad predstavlja rad u Clarionu i opisan je uz pomoć slika, uvodne teorije te objašnjenja pojmova. Pored prikazanih mogućnosti ima i puno više ostalih za istražiti, ali prikazane su one najvažnije.

Iako ova aplikacija nije nešto što bi napredne tvrtke htjele koristiti, moglo bi se još dodatno poraditi na izračunima, vezama, možda i dodatnim tablicama, ali s obzirom da je radila jedna osoba, to je na ovaj način zamišljeno i sigurno postoji nešto što bi se netko sjetio dodati. Najviše vremena oduzimaju dijagram i relacijska shema, a to su ujedno i najvažniji elementi jer njih koristimo pri izradi aplikacije. Najvažnije je upoznati se sa načinom poslovanja sustava, ukoliko je moguće popričati sa djelatnicima i korisnicima da bi dobili uvid u ono što poslovnom sustavu treba za napredovanje. Trebalo je napraviti dosta verzija dijagrama prije nego što se došlo do trenutne koja bi se mogla još dalje nadograđivati. Problemi su se javljali i u izradi rječnika. Opciju *Must be in Table* često nije bilo moguće odabrati i to je značilo da veze ili da brojnosti veza nisu odgovarale. Brojnosti veza trebalo je ispraviti na dijagramu entiteti-veze, a kasnije i u rječniku. Ovisno u tipu veze trebalo je paziti kako će povezane tablice djelovati jedna na drugu, odnosno da li će se podaci slabog tipa entiteta mijenjati (brisati) nakon promjene podataka u jakom tipu entiteta. Treba paziti i na određivanje tipa podataka za određeni stupac kako bi logički odgovaralo cijeloj aplikaciji. Problemi nisu izostali ni u kreiranju aplikacije, program se stalno rušio i iznosio neke nove greške koje je trebalo ispravljati u samom rječniku, a za neke je trebalo malo više truda i pronaći ih unutar aplikacijskog stabla. Prilikom izrade izvještaja treba paziti na organizaciju podataka i ukoliko neki podatak nije potreban može se izostaviti iz prikaza. Posebno je to važno naglasiti kod složenog izvještaja koji prikazuje podatke iz više povezanih tablica.

Unatoč svim problemima, rad u Clarionu i rad sa bazama podataka je zanimljiv. Iz tog razloga je i izabrana tema izrade poslovne aplikacije u Clarionu. Bilo je vremena naučiti i istražiti neke nove stvari koje nisu bile obrađene na kolegiju iako na kraju neke od njih nisu prikazane u radu zbog javljanja greške prilikom njihove primjene. Nadam se da ću u budućnosti izraditi aplikaciju koja će biti korištena unutar poslovnih sustava naprednijih tvrtki.

5. LITERATURA

- [1] Pavlić Mile, Oblikovanje baza podataka, Odjel za informatiku Sveučilišta u Rijeci, Rijeka, 2011.
- [2] Tkalac Slavko, Relacijski model podataka, Informator, Zagreb 1988.
- [3] Varga Mladen, Baze podataka, Konceptualno, logičko i fizičko modeliranje podataka, DRIP, Zagreb, 1994.
- [4] Pavlić Mile, Modeliranje procesa, Odjel za informatiku Sveučilišta u Rijeci, Rijeka, 2014.
- [6] <http://www.softvelocity.com/> (18.9.2020., 20.07)
- [5] <http://files.fpz.hr/Djelatnici/tcaric/Tonci-Caric-Baze-podataka.pdf> (19.9.2020., 14.53)
- [7] <https://www.zv.hr/interliber> (19.9.2020., 11.37)

6. Popis slika

Slika 2.1. Dijagram entiteti veze	8
Slika 3.1. Stvaranje .dct datoteke i spremanje rječnika.....	11
Slika 3.2. dodavanje tablice.....	12
Slika 3.3. Unos općenitih svojstva stupcu.....	13
Slika 3.4. Unos atributa određenom stupcu.....	14
Slika 3.5. Pomoć pri unosu stupca.....	15
Slika 3.6. Provjera ispravnosti stupca.....	16
Slika 3.7. Upravljanje stupcima.....	17
Slika 3.8. Dodavanje primarnog ključa.....	18
Slika 3.9 Dodavanje vanjskog ključa.....	19
Slika 3.10. Dodavanje ključ za sortiranje (SORT KEY).....	19
Slika 3.11. Primarni ključ AGREGACIJE.....	20
Slika 3.12. Rezervni ključ AGREGACIJE.....	21
Slika 3.13. Veza DRZAVA – IZLAGAC	22
Slika 3.14. Veza agregacije.....	23
Slika 3.15 Početni prozor za izradu aplikacije	24
Slika 3.16. Kreiranje procedura.....	25
Slika 3.17. skocni prozor.....	26
Slika 3.18. Ukupno	27
Slika 3.19. Azuriranje Izlagaca i Popis Racuna.....	28
Slika 3.20. Pretraga u Popis Izlagaca sa gumbom Pretrazi	29
Slika 3.21. Gumb za lookup	29
Slika 3.22. Lookup polje.....	30
Slika 3.23. Dodavanje gumba za kalendar	31
Slika 3.24. Svojstva	32
Slika 3.25. Jednostavni izvještaj.....	33
Slika 3.26. Izvjestaj preko gumba.....	34
Slika 3.27. Složeni izvještaj	35