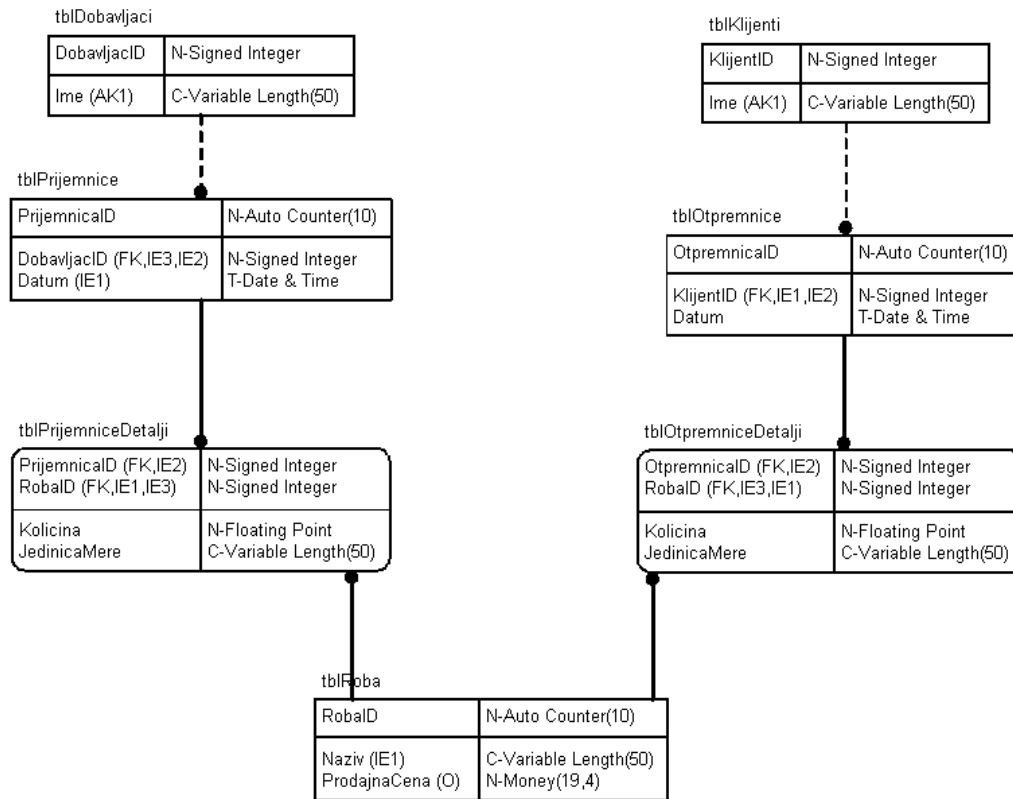


Magacin – Fizicki Model Baze Podataka

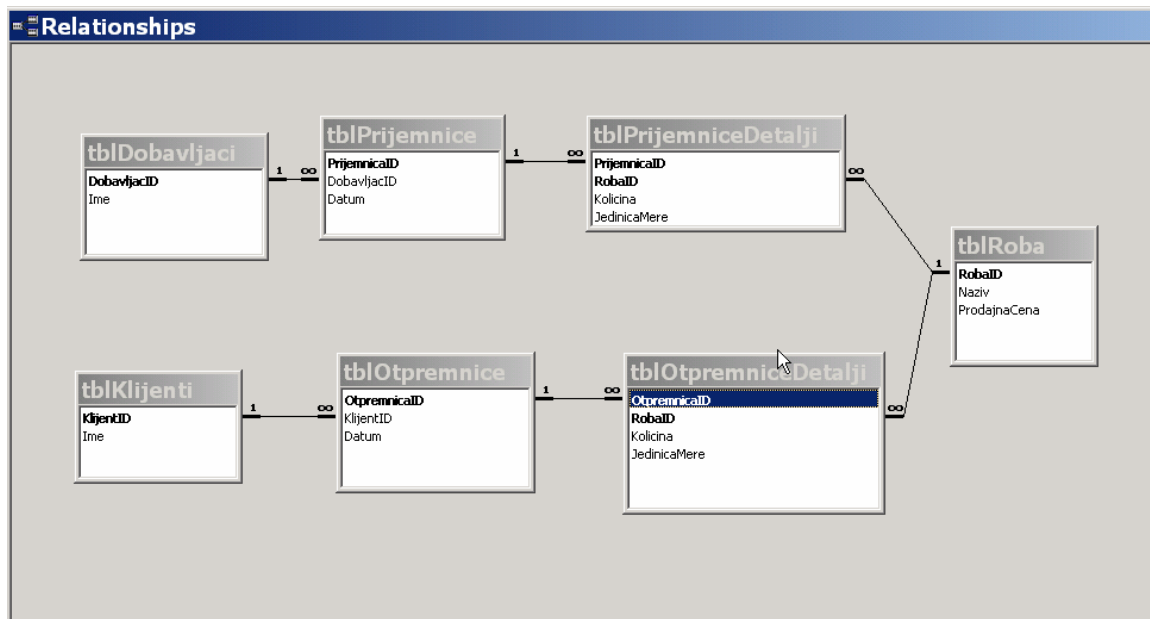
Projekat: Magacinsko Poslovanje Konceptualni Model Baze Podataka



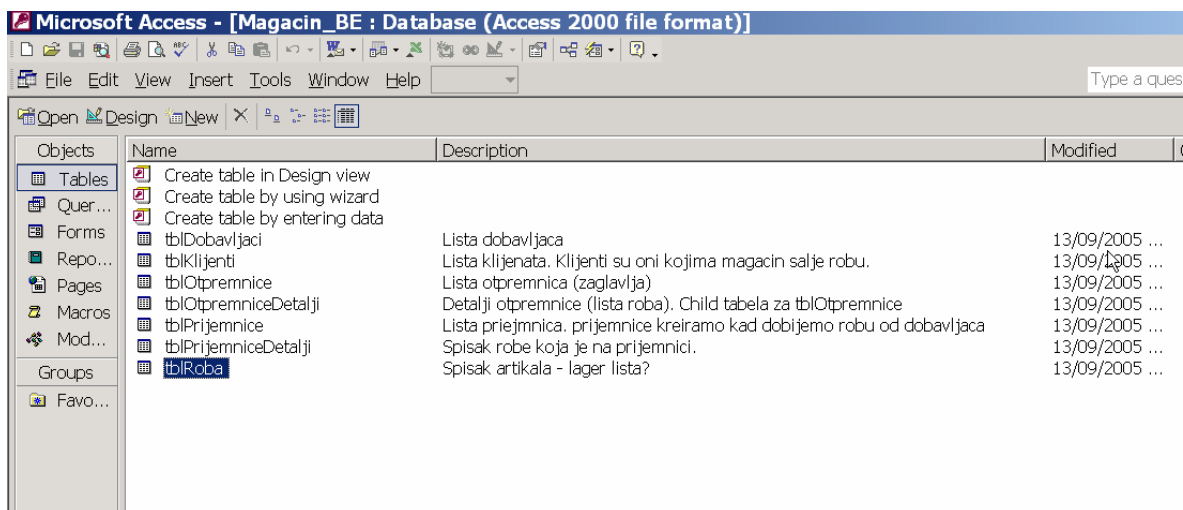
Ovo je nas usvojeni model podataka. Rado bih ga nazvao ‘Minimalni model’ jer zaista poseduje samo minimalni deo tabela i polja potrebnih za realni zivot. Minimalan ili ne, posluzece dobro da ilustrujemo postupak dizajna aplikacije koji ja zovem ‘Pokrivanja Dijagrama Baze’

U Accessu, to izgleda ovako:

Fizicki model baze podataka, Access



U database window to izgleda ovako:



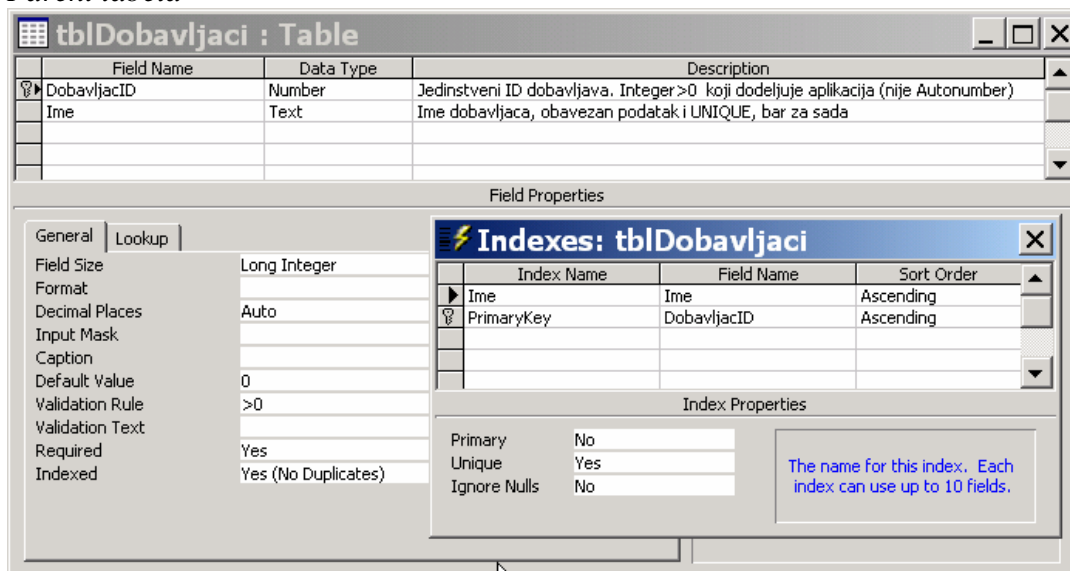
Primitite kako su u Description sekciji isisani komentari – sta koja tabela radi. Ovako **dokumentujemo** svoj rad usput, dok radimo.

Fizicki dizajn tabela

Skrećem pažnju naneke sitnice koje ne kostaju mnogo vremena u fazi dizajna tabela, a uštede mnogo nevolja kasnije:

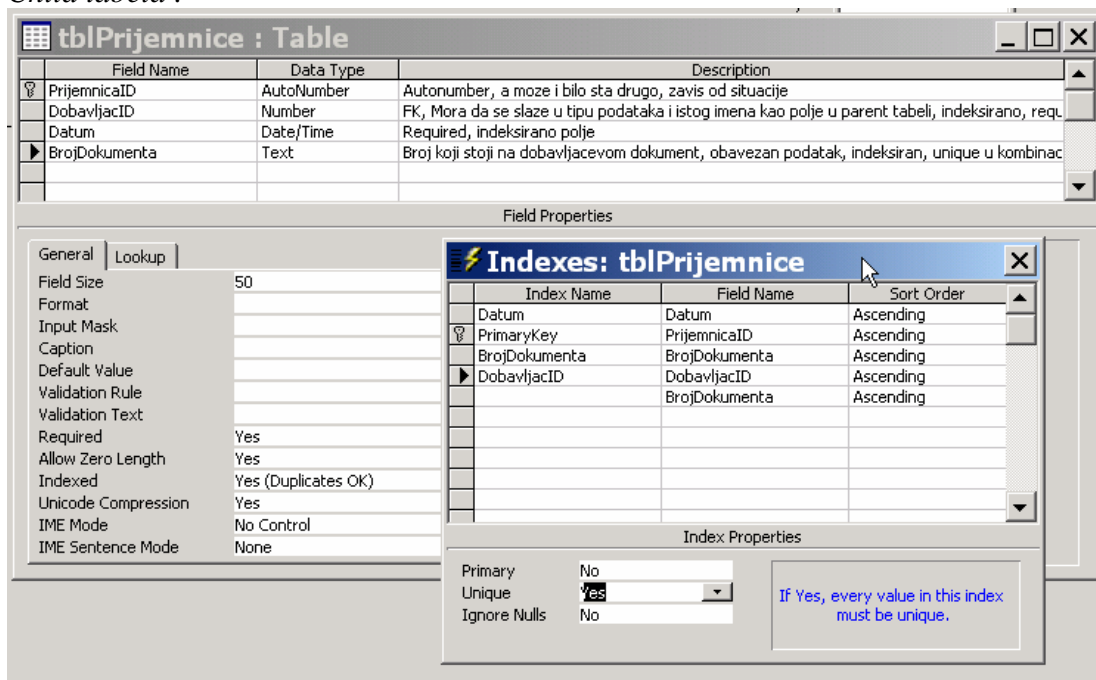
- Potsvaljene su relacije između tabela. To se često zaboravlja (!???)
 - Polja koja služe kao FK (foreign key) u relacijama imaju ista imena kao odgovarajući PK (Primary Key) u parent tabelama
 - Sva FK polja su indeksirana
 - Vecina polja je Required, FK obavezno
 - Uz sva polja je stavljen komentar.
 - Properties Required, Default, Validation rule su maksimalno korisne u vecini polja
 - ako bas nema default value za neko polej gde ocekujete slobodan unos, stavite 'N/A' – not available, nije raspoloživo u momentu unosa, tek da izbegnem NULL vrednosti
- Evo primera parent i child tabela :

Parent tabela



Za prikazanu tabelu primetite Default value=0 a validation rule kaze >0. Ovo vam nece dozvoliti da zaboravite da unesete vrednost i pokusate da sacuvate record. Ako stavite za Default neku nedozvoljenu vrednost, sistem ce vam uvek traziti da nesto unesete, i to nesto morace da bude dobar podatak.

Child tabela :



Primetite das u sva polja required. Primetite da index DobavljacID ide po dva polja, (DobavljacID,BrojDokumenta). Ovo zato sto predpostavljamo da dobavljac ima svoj system numerisanja dokumenata koji dolaze uz isporucenu robu (BrojDokumanta). Moze se desiti da dobavljac koristi gotove blokove racuna, iz knjizare, i ako dva dobavljava kupe isti blok, eto moguceg duplikata,. Zato smo ukljucili BrojDokumenta zajedno sa DobavljacId u jedan UNIQUE indeks. Istovremeno je indeksiran nezavisno i sam BrojDokumenta, da bi ubrzali pretrazivanje ako zatreba u buducnosti.

Dizajn Aplikacije

Da bismo dizajnirali aplikaciju, potrebno je da znamo nekoliko stvari:

- dijagram baze podataka
- mapa poslovnog procesa
- pravila poslovanja (“business rules”) koja nisu pokrivena dizajnom bazom podataka
- korisnikove zelje i specijalni zahtevi u vezi sa interfejsom i reportima

Dijagram baze imamo, i konceptualni i fizicki. Mapu procesa dali smo na pocetku. Jedino pravila poslovanja nismo do sada nigde naveli, pa ni ona koja su pokrivena modelom baze. Ucinicemo to malo kasnije. Zeleji i zahteve u vezi interfejsa cemo pretpostaviti u nasem slucaju.

Kako koristimo model baze i pravila poslovanja u dizajnu aplikacije? Dijagram baze, uz kartu process, ce nam pomoci da odredimo [b]koje ce forme i osnovni reporti biti napravljeni[/b]. To je ono sto ja zovem ‘postupak pokrivanja dijagrama formama i reportima’. Pravila poslovanja nam kazuju [b]kako ce da funkcionisu[/b] forme i reproti. Znaci, dijagrami ce nam reci [b]Sta[/b] treba da napravimo, a proslovna pravila nam kazu [b]kako[/b] da to nesto napravimo.