

Uvod

Veliki broj firmi bavi se manipulisanjem robe – nešto radi s nekakvom robom. Roba ili materijal se od nekuda dobavlja, pa se onda može preprodati ili preraditi u neki proizvod veće složenosti, koji se na kraju prodaje. Znači, ulazni materijal ili roba putuje od dobavljača do finalnog kupca. Na tom putu roba ili materijal nekoliko puta prolazi kroz magacine ili stovarišta. Ne mora da bude uvek zgrada ili ograđeno stovarište, svako mesto gde se roba zadržava neko vreme pre nego što promeni vlasnika jeste u stvari magacin. Na ovaj način, rafovi u samousluži jesu vrsta magacina. Kamion koji dovozi lubenice u Beograd iz Sremske jeste magacin. Lubenice su utovarene i sedece na kamionu nekoliko dana, dok se ne prodaju. Gajbe piva u kaficu koje čekaju da se prodaju jesu magacin. Ova uopštenost znači da se bi informacioni sistemi za praćenje rada magacina bili slični u svim ovim slučajevima.

Poslovni proces u magacinskom poslovanju

Sta magacini rade?

Osnovna uloga magacina jeste da

- primi robu od dobavljača
- isporuci robu kupcu/korisniku

Kako magacini odradjuju ono sta rade?

Magacin je dakle veza između dobavljača i kupca. Može se reći da magacin vrši transfer robe od dobavljača prema kupcu. Da bi se taj transfer robe izvršio, magacioner mora da -

- vodi evidenciju primljene robe
- vodi evidenciju izdate robe
- zna gde se u magacinu nalazi roba

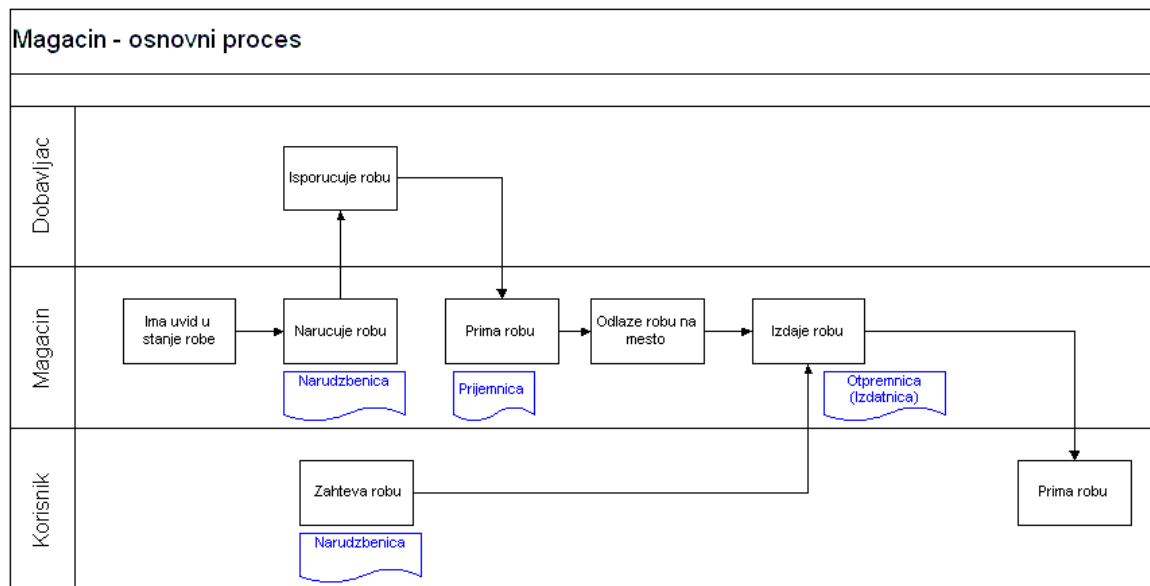
Magacioner je odgovoran pred zakonom za primljenu robu. Roba koja je primljena mora biti ili u magacinu ili izdata kupcima. Fizičko stanje u magacinu mora da se slaže sa računskim stanjem

Veza magacina sa ostatkom firme

Magacin je samo deo poslovanja neke firme i ne može se posmatrati izolovano od ostatka firme. Magacin komunicira sa drugim delovima firme. Zavisno od organizacije firme, uloga magacina može biti različita. U nekim firmama, magacin šalje izveštaje o stanju robe u direkciju, a odande stizu nalozi za nabavku. U nekim firmama magacin sam donosi odluku o nabavci. Negde magacin samo prima i idaje robu, a neko drugi vrši plaćanje i naplatu. Negde magacin vrši i plaćanje i naplatu. Što manja firma, to više funkcija vrši magacin. Magacin ne mora da se zove magacin da bi vršio sve ove funkcije. U malim firmama, sve ove funkcije obavlja veoma mali broj ljudi, često samo jedan čovek. U većim firmama to rade citavi sektori, ali se zahteva veća kooperacija između sektora. Bilo da je firma velika ili mala, mora se magacin posmatrati kao deo celine.

Karta procesa zamisljenog magacina

Ovako može da funkcioniše magacin :



Slika 1 : Poslovni proces

Pored aktivnosti prikazani su prateci dokumenti.

Informacioni sistem za pracenje magacinskog poslovanja

Definisimo prvo pojam informacioni sistem, za potrebe ovog dokumenta. Neka IS bude skup sledecih elemenata:

- baza podataka (tabele, sirovi podaci)
- jedna ili vise programa koji pristupaju bazi

Znaci, treba da napravimo dve stvari koje rade zajedno – bazu podataka i aplikaciju (ili vise njih).

Baza Podataka Magacin

Ponovimo « Sta magacini rade" :

Osnovna uloga magacina jeste da

- *primi robu od dobavljacka*
- *isporuci robu kupcu/korisniku*

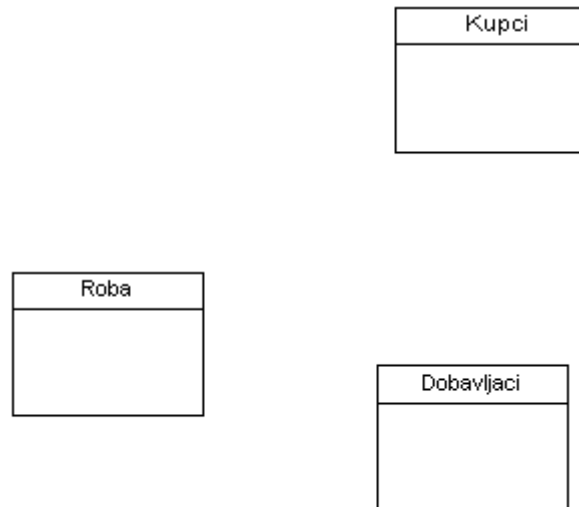
Dizajn baze podataka pocinje uocavanjem vaznih imenica u opisu problema kojim se bavimo. Ovde imamo imenice:

roba
dobavljacki
korisnici

Glagoli u opisu problema obicno ukazuju na veze izmedju imenica:

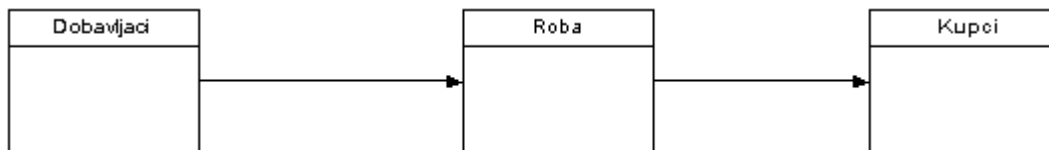
primi
isporuci

Onda se za vazne imenice naprave tabele. Mi bismo imali :



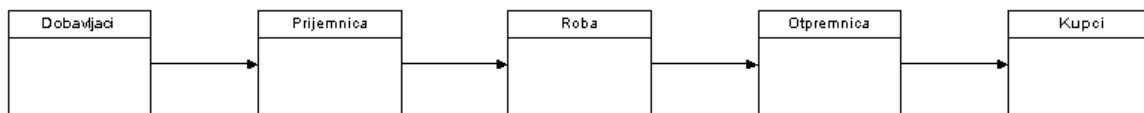
Slika 2: Objekti koji ucestvuju u procesu (imenice, entiteti)

Sve je dobro za sada. Kako bismo ove tri imenice doveli u vezu? Mozda ovako :



Slika 3: Veza izmedju objekata

Eto, vidi se kako roba putuje od dobavljacka do kupca. Lepo, ali nam ne pomaze bas mnogo. Pogledajmo dijagram procesa i obratimo paznju na dokumente u procesu. Uz prijem robe ide *prijemnica*, a izdavanje parti *otpremnic* . Dodajmo ih na sliku :



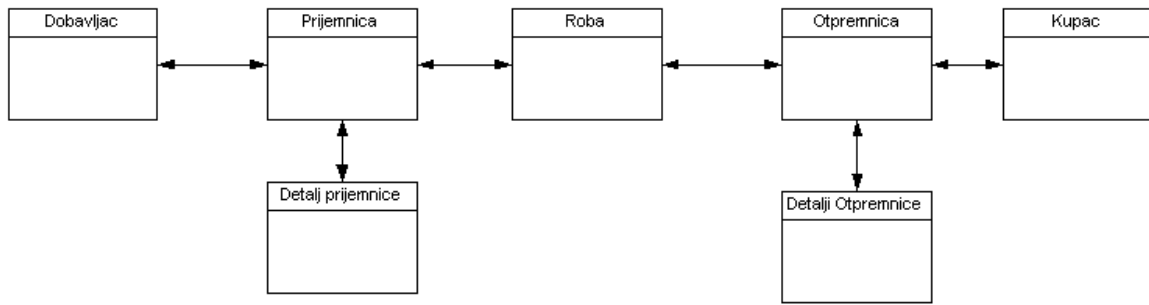
Slika 4: Potpunija slika objekata i veza

Naravno, ako se nesto evidentira, to su dokumenti. Znaci, evidentiracemo prijemnice i otpremnice.

Kao ovu sliko objekata pretvoriti u bazu podataka? Skoro sam negde procitao, da se objekti (stvari, dogadjaji) predstavljau tabelama relacionih baza. A osobine objekata (atributi) predstavljaju se poljima tabela. Do polja jos nismo stigli, pozabavimo se za sada tabelama.

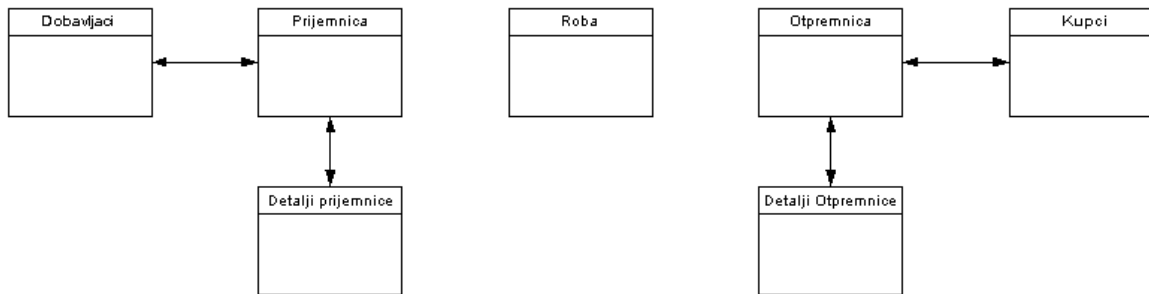
Koji je najbolji nacin za cuvanje podataka o prijemnicam i otpremnicama?

Do sada znamo iz prakse da nam za svaki document ovakvog tipa trebaju tri tabele ZaglavljeDokumenta i Detalji i Roba. Imamo do sada robu i zaglavlja dokumentata. Znaci, treba nam jos tabela:



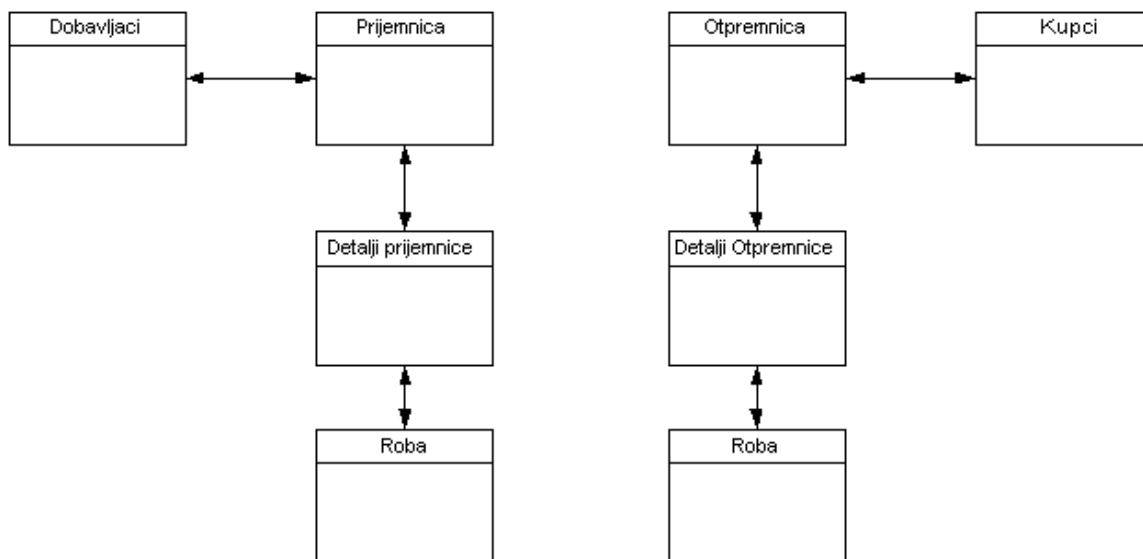
Slika 5: Objekte su us tvari tabele

Slika je pomalo cudna, priznajem. Nekako je sve povezano ali bas najjasnije. Da uprostim stvari, hajde da razdvojimo prijemnice od otpremnica. Kad roba ulazi u magacin, mi u principu ne znamo sta ce se sa robom desiti i kada, niti nas interesuje. Jedino nas interesuje da evidentiramo prijem robe. Kad izdajemo robu, takodje nas ne zanima kada je roba stigla i otkud se nasla u magacinu, zelimo samo da evidentiramo tacno koja je roba izasla i kome je predata. Dakle :



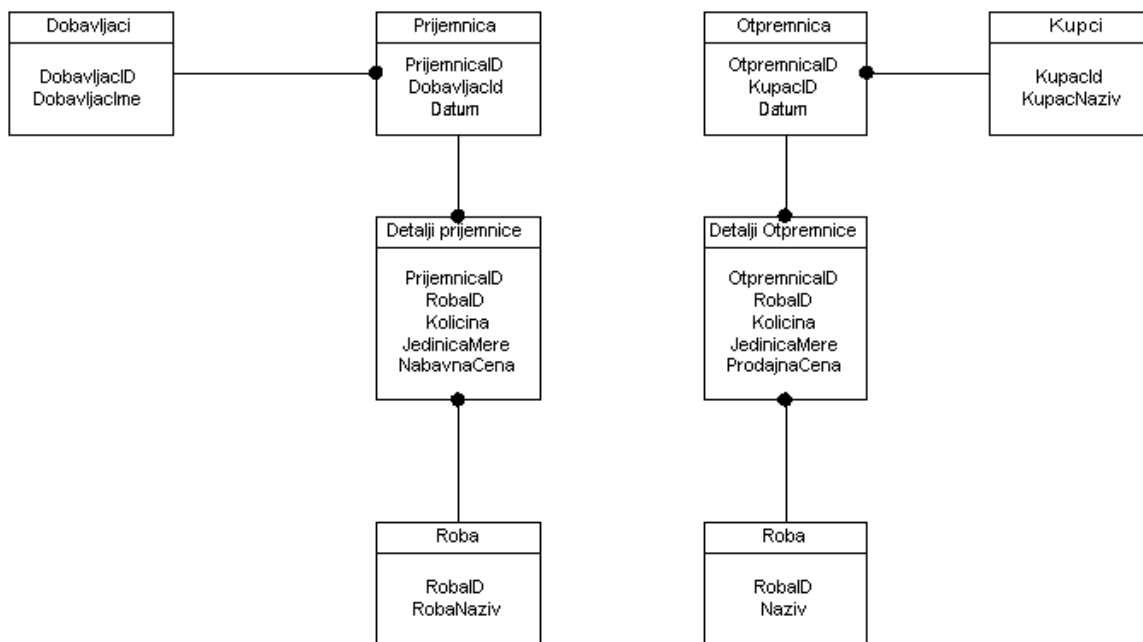
Slika 6 : Objeti i tabele, razdvojene otpremnice i prijemnice

E, sad nam visi roba u vazduhu. A sve ostalo deluje OK. Dobavljac nam daju prijemnice. A priemnici pise spisak robe koju primamo (Deatalji Prijemnice). Aha, tu smo dakle, robu treba vezati sa Deatljima Prijemnice. Ista prica vazi za Otpremnice i Kupce.



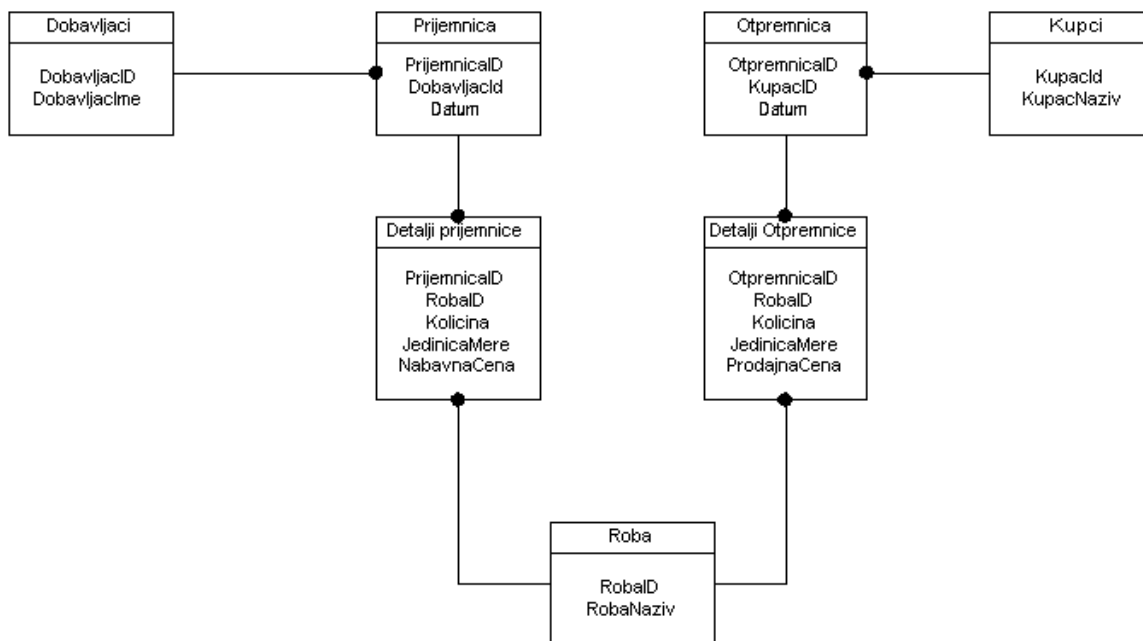
Slika 7: Objekti i veze, skoro gotovo

Vidimo jednu simetricnu sliku. I prlicno je jasno o cemu se radi. Sta mu samo znace strelice? Pa, nista, ali mogle bi. Tabele na slici su u nekoj vezi. A veze u bazama podataka se izrazavaju kao « 1 prema vise », « vise prema vise ». Umesto strelica, usvojimo simbol za « vise », kruzic. Mozemo da dodamo i neke atribute nasim tabelama (polja). Sledeca slika prikazuje sat smo dobili.



Slika 8: Objekti i veze – relationships diagram

Lepo, ali nije bas najbolje. Sta ne valja? Pa tabela Roba javlja se dva puta, jednom za prijernice jednom za otpremnice. Ajde da to ispravimo:

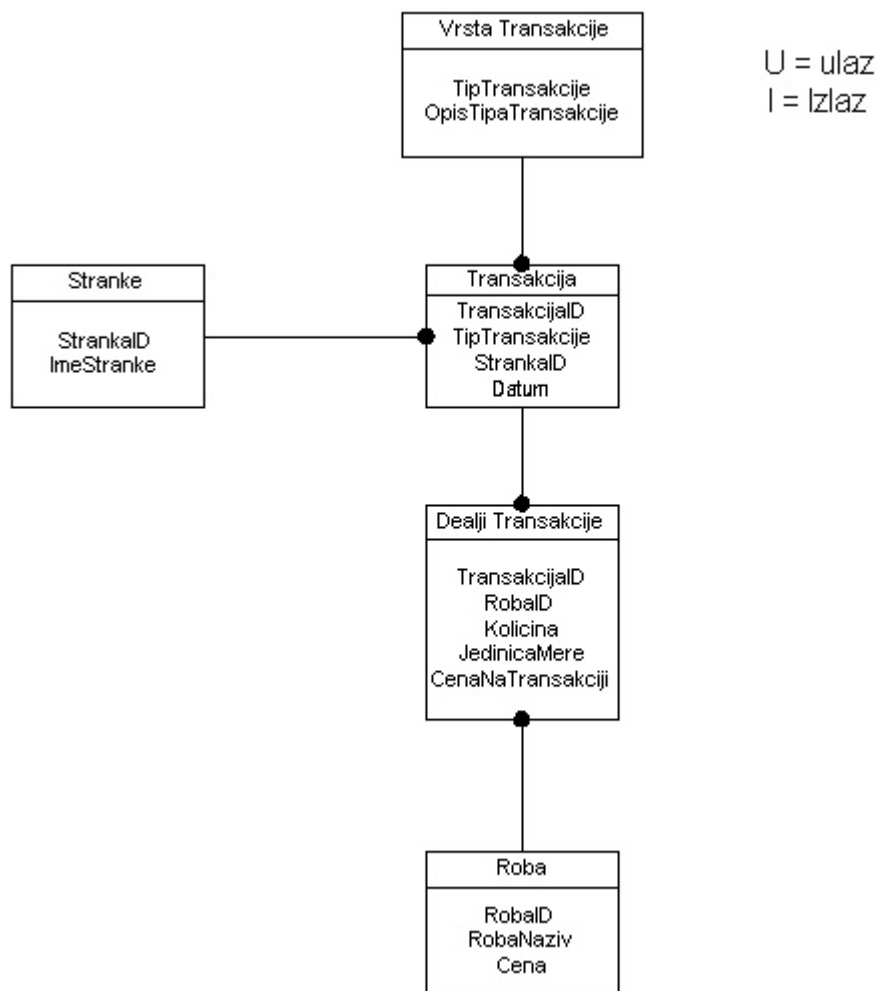


Slika 9, Magacin: logicki model baze = diagram tabela I relacija medju njima

Sad vec imamo nesto ozbiljno. Ovo vec moze da bude osnovni model ne za ‘pracenje rada magacina’ nego za uopsteniji problem ‘pracenje protoka robe’. Ovaj model moze da služi kao osnova za magacin, veletrgovinu, maloprodaju, prodaju putem interneta. Kako kad su ovo sve razlicite stvari? Tacno, razlicite stvari ali u osnovi iste. Roba se prima i prosledjuje negde. U cemu ce se videti razlika? U programima koji ce raditi sa datom bazom.

Vratimo se za trenutak na simetricnost modela. U nekim slucajevima (ali ne uvek) moze da se model generalizuje. Prijem i isporuka robe su jedna te ista operacija sa razlicitim predznacima. Zamislite da imate samo jednu polovinu modela, recimo Prijemnice a treba da unesete i prijem i izdavanje robe. Mozete recimo da kolicine koje ulaze unosite sa « + » (plus) predznakom a koje izlaze sa « - » (minus) predznakom. Upit « qryStanje » bio bi prava pesma. Kad izlistate karticu materijala, odmah se jasno vidi sta je ulaz a sta izlaz, milina bozija. Lepo, ali se ne koristi, verujte mi na rec, nije bas efikasno. Kako onda? Ovako :

Uvodjenjem pojma *Vrsta Transakcije* mozemo da spojimo simetricne grane modela u jednu. Ako kazemo da postoje dve vrste transakcije, Ulaz i Izlaz, olakasacemo sebi zivot I imacemo manje tabela da pravimo. Videcemo kako baza poataje mnogo fleksibilnija ako generalizujemo model. Pogledajmo sliku 10 :



Slika 10, Magacin, generalizovani logicki model baze podataka

Da li je generalizovani model baze podataka bolji od simetricnog? Moze da bude ali i ne mora. Mozete da pomislite da jedna grupa tabela omogucuje da koristite iste forme za kreiranje i otpremnica i prijernica. Moze ponekad, ali je programiranja znatno komplikovanije. Na primer, na prijernici cenu robe ukucavate, a na otpremnici kopirate je iz tabele Roba. Naslovi sadrzaj zaglavlja formi i reporta je razlicit. Moze da se programira da se izpisuje drugi naslov u zavisnosti od transakcije, ali je to glavobolja. Ako zelite ponekad da odstampate prijernicu/otpremnicu (a zelite, zar ne) trebace vam dva zasebna reporta, pa maker gledali u iste tabele. Dalje, na generalizovanom modelu nije vise potpuno jasno ko je kupac a ko dobavljac.

Generalizovani model dozvoljava da ista stranka bude i kupac i dobavljac, sto se desava (neka firma nam prodaje eksere, a kupuje ormane od nas). Generalizovani model omogucuje da imamo samo jednu tabelu sa adresama (namerno nisam adrese stavio u ovaj model), pa cak I da strpamo zaposlene u tabelu Stranke. Zaposleni ne kupuju I ne prodaju nista (mozda kupe proizvod svoje firme), ali ucestvuju u troskovima – isplacuju im se plate.

U generalizovani model mozemo da dodamo i nove vrste transakcija, bez mnogo problema. Na primer, narucujemo robu od dobavljacka. Ta roba nije jos stigla niti se zna da li ce stici i kada, prema tome nije u pitanju ulaz robe, a ni izlaz. Jednostavno, pravimo spisak robe koju narucujemo od nekog dobavljacka. Ako hocemo i to da pratimo, dodamo novi tip transakcije 'mi narucujemo'. Zasto ovoliko dugacko, 'mi narucujemo od dobavljacka'? Pa zato sto mogu i kupci da narucuju robu od nas, postom ili preko interneta. Onda bismo imali novu transakciju 'od nas narucilli kupci'. Nije ulaz, nije izlaz, nismo nista narucili, znaci, nesto cetvrto. Pa onda cuvene profakture, sa sve cenom. Sve to moze da stane u generalizovani model. Ali, nove forme, novi report ice morati da se rade, pa maker se gradili na osnovu postojećih.

Za pocetnika i manje projekte, verovatno je laksi simetricni model. Ako se pored magacina pokrivaju i drugi elementi poslovanja firme, generalizovani model je verovatno bolji, ali za to treba mnogo iskustva. I simetricni model moze da se siri. Jednostavno izmedju tabele roba i Kupaca/Dobavljava ubacujemo po dve tabele za svaku novu vrstu transakcije (aglavlje i detalji). Nista strasno ako se dodaju tabele na model. Strasno je ako se dodaju kriticka polja i menjaju posotojece tabele, ili se problem ne moze nikako resiti osim potpunom preradom modela.

Koji god model da usvojimo, u ovom momentu to je samo logicki (konceptualni) model baze podataka. Kad odaberemo software, napravicemo fizicku bazu. U tom procesu, mozda tu i tamo dodamo po neko polje, mozda i celu tabelu, videcemo.