

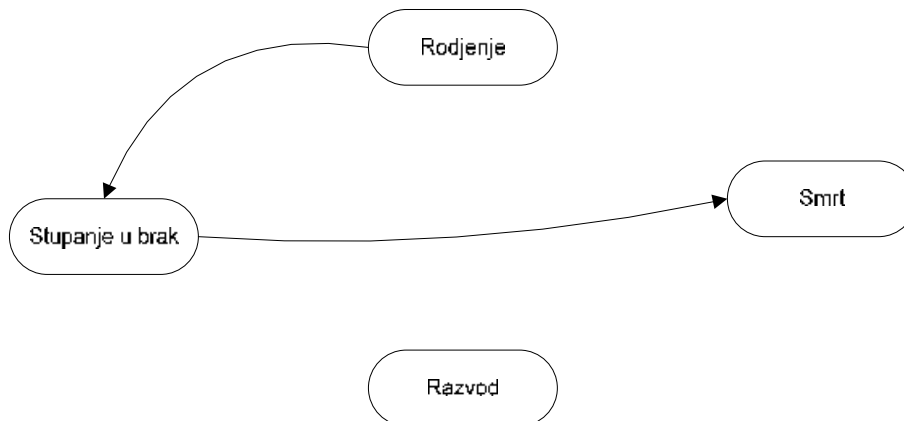
Pracenje Promena Stanja

Jedan od nacina da pratimo promene u nekom sistemu jeste da pratimo promene stanja. Sitem nekom akcijom prelazi iz jednog stanja u drugo ili iz jedne faze u neku drugu.

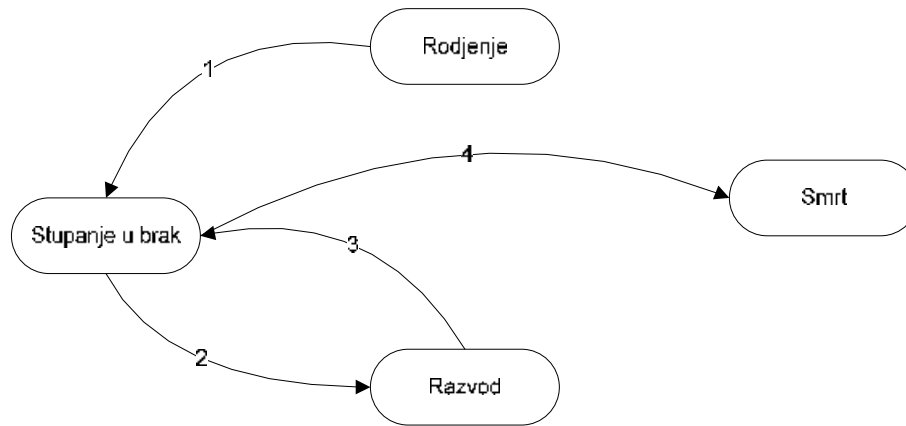
Na primer, zamislimo maticara u opstini koji zapisuje vazne promene (faze) u zivotu ljudi. Sto se maticara tice, faze u zivotu osobe su 'rodjenje', 'stupanje u brak', 'razvod', 'smrt'. Ako maticar uredno zpiše kad su se ove faze desile, uvek mozete da od njega dobijete rodni list (krstenicu) ili vencani list. Kada covek umre, i to se zapiše u maticne knjige. Maticna knjiga je dakle baza podataka u kojoj se cuvaju podaci o vaznim fazama u zovotu ljudi. Cetiri faze za koje je zainteresovan maticar mogu se prikazati an dijagramu, ovako:



Prav faza je uvek rodjejne. Posle toga moguće su razne varijante. "Normalno", osoba se rodi, pa u nekom momentu stupid u brak, srećno zivi i na kraju umre. To se može ovako poakzati na dijagramu:

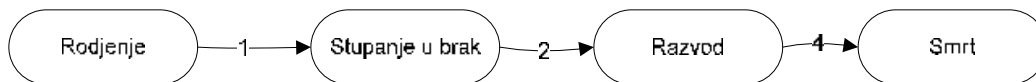


No, nisus vi brakovi srećni, pa dodje do razvoda. Posle toga neko se preuda ili ozeni ponovo, a neko zivi sam do kraja zivota. Evo kako to izgleda sa razvodom i preudajom:

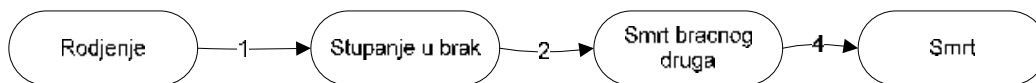


Na sterlicama smo upisali redosld desavanja. Primetite da se koraci 2 i 3 mogu desiti i vise puta. Jedan moj drug se zenio 9 (devet) puta, od toga dva puta sa istom zenom ☺. Broj 4 ga jos nije stigao.

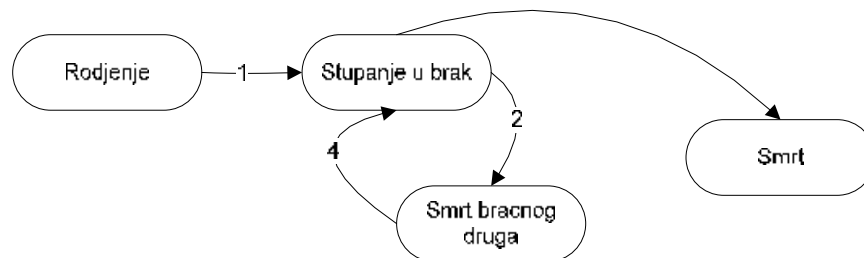
Ko ostane da zivi sam, njegov dijagram izgleda ovako:



Kad bolje razmislím, ima i drugih kombinacija. Moja baka je zivela ovako:



Jedan rodjak je ziveo ovako:

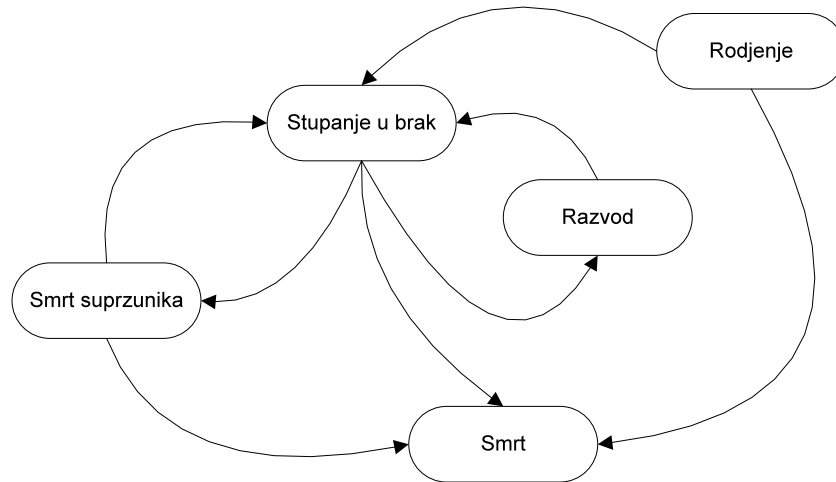


Sta svi dijagrami imaju zajednick? Pocetna faza ili dogadhja je „Rodjenje“ i završna faza je „Smrt“. U medjuvremenu, svasta moze da se desi. To 'svasta' je medjutim ograniceno. Niko ne moze da se razvede ili ostane udovac pre nego se ozeni. Ako neko umre, ne moze posle da stupid u brak. Deluje cudno da uopste o ovome raspravljamo, ali koliko ste puta culi da je osoba pereminula i zatim pozvana da recimo bude u poroti na sudu? Ili joj je posolat racun za struju i TV pretplatu.

Poceli smo sa cetiri faze i negde na pola primera uveli smo fazu 'smrt bracnog druga'. Bas kao sto se desaav u anlizi sistema, pocnemo sa nekim saznanjima in a pola posla se promene zahtevi...

Nacrtacemo dijagram sa svim fazama is vim mogucim dogadjajima.

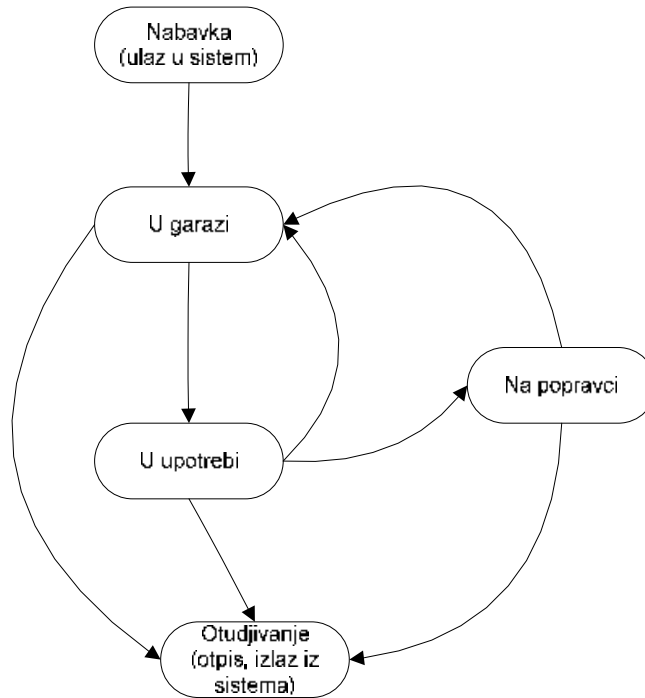
Kako maticni ured vidi faze u zivotu jedne osobe



Od rodjenja do smrti osoba moze da prodje kroz medju-faze na razlicite nacine. Neke faze se mogu ponavljati vise puta, ali uvek u pravcu strelica.

Ako je entitet koji posmatramo osnovno sredstvo, onda mozemo da nacrtamo ovakav dijagram:

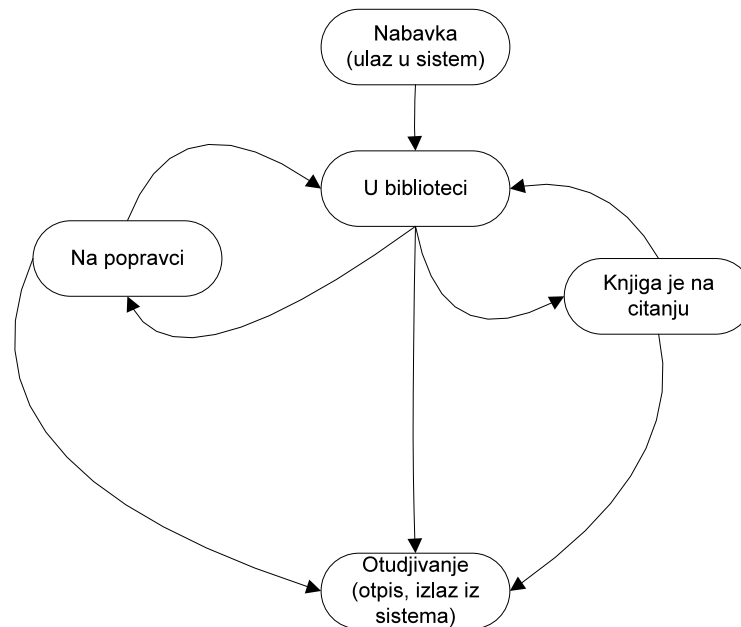
Zivotni put jednog kamiona



Kamion je kupljen. Kad ga registrujemo i dovezemo u garazu, postaje spreman za upotrebu. Neko vozi kamion - u upotrebi je. Onda ga vrati u garazu - opet je spreman za upotrebu. Uz malo sreće, tako radimo sledećih 20 godina. Posle 20 godina, kamion se vrati u garazu i onda se otpise. Ako se desi da je u kvaru, može se poslati na opravku pre nego što se vrati u garazu. Odatle ide nazad u garazu, pa opet u upotrebu. I tako u krug. A može se desiti nesreća na putu i kamion je slupan, te se otpisuje (iz faze 'upotreba' prelazi u fazu 'otudjivanje').

Slican dijagram može se napraviti za knjigu u biblioteci:

Zivotni put jedne knjige iz biblioteke



Ovako možemo da teramo do sutra. Kako bismo sve ovo zapisali u bazi podataka?

Modeliranje Promene Stanja u Relacionj Bazi Podataka

Svaka strelica predstavlja jednu promenu stanja posmatranog entiteta. Treba pamtiti strelce i to kao par {„pocetno stanje“, „završno stanje“}. Za knjigu, imali bi ova moguća stanja:

- („Nabavka“, „U biblioteci“) - knjiga je kupljena i stavili smo je na policu
- („U biblioteci“, „Na citanju“) - neko je pozajmio knjigu
- („Na citanju“, „U biblioteci“) - knjiga je vraćena
- („Na citanju“, „Otpisana“) - neko pozajmio pa zagubio ili uništio knjigu
- („U biblioteci“, „Otpisana“) - posle moga upotreba vreme je da se knjiga otpise, ili smo je zagubili u samoj biblioteci
- („U biblioteci“, „Na popravci“) - neko je vratio knjigu pocepanih korica pa smo je poslali na popravku

("Na popravci", "U biblioteci") - knjiga stigla sa popravke
("Na poravci", "otudjivanje") - mi je poslali na poravku, a oni je izgubili
ili se ne moze popraviti

Uvedemo dakle tabelu mogucih (dozvoljenih) **promena stanja**. Naravno da ce nam
trebati i tabela mogucih **stanja** (Nabava, U biblioteci, Na popravci, na
citanju, oOtpis). Tabela koja sadrzi entitete koje pratim se podrazumeva
(Zoranov PredmetPoslovanja ☺)

Promene stanja pamtili bi u tabeli koja za pocetak izgleda ovako:

Tabela PromeneStanjaEntitetaE:

{EntitetID, PocetnoStanje, ZavršnoStanje, DatumPromeneStanja}

Ocigledno je da nam treba FK na tabelu **Entiteti**, kao i na tabelu
DozvoljenePromeneStanja. Sta stavimo u tabelu DozvoljenePromeneStanja, samo
to ce smeti da udje u tabelu PromeneStanja.

A tabelu DozvoljenePromeneStanja kontrolisemo mi. Ako se nam nije dovoljno da
kazemo „Otpisana knjiga“, mozemo da uvedemo stanje „Ukradena“, „Unistena“ i
sta god hocemo jos, u bilo kom momentu, i da upisemo u
DozvoljenePromeneStanja svemoguce odnose u kojima novo stanje moze biti sa
postojecim. Ovim dobijamo mnogo na fleksibilnosti baze. Nismo ograniceni
prvom iteracijom modela poslovnog procesa.

Za sada, ogranicili smo promene stanja i time sprecili da u bazu upisemo
greskom kako je otpisana knjiga upravo daat na citanje. Ovo je potreban
uslov, ali ne i dovoljan.

*Kako da garantujemo da smo zapisali sve promene u pravilnom redosledu? Ako
smo zapisali (Knjiga1, „nabavljena“, „u biblioteci“, „10 Jan 2010“) ond sledeci
red u tabeli mora da ima PocetnoStanje = „u biblioteci“. , to jest
PocetnoStanje za jedan red mora biti isto sto i završno stanje za prethodni
red. Samo tada podaci imaju smisla. Ovako nekako:*

(Knjiga1, „nabavljena“, „u biblioteci“, „10 Jan 2010“)
(Knjiga1, „u biblioteci“, „na citanju“, „12 Jan 2010“)
(Knjiga1, „na citanju“, „u biblioteci“, „10 Feb 2010“)

Sledeci red ne bi smeo da izgleda ovako:

(Knjiga1, „na popravci“, „u biblioteci“, „16 Feb 2010“)

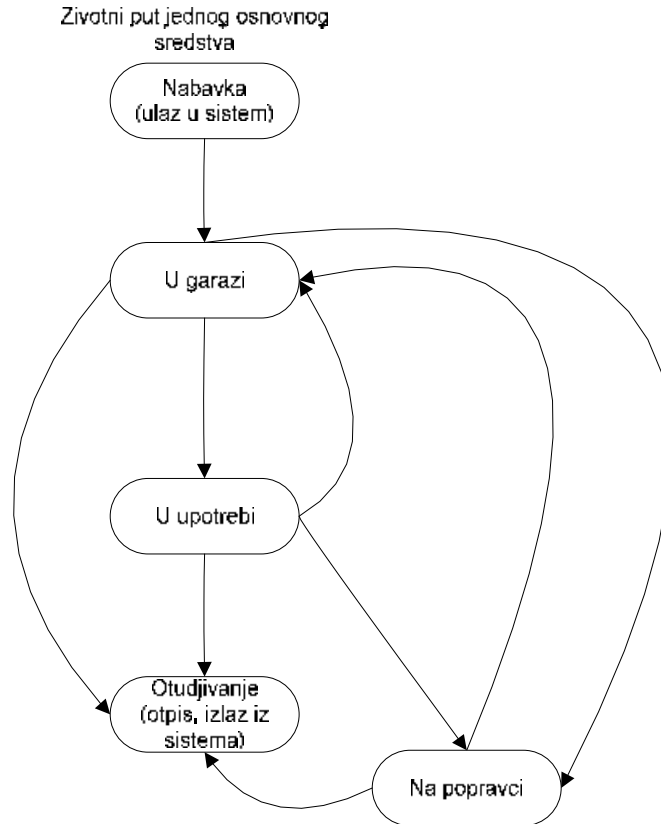
Ne moze da pocetno stanje bude „na popravci“, ako je prethodni rekord imao
završno stanje = „u biblioteci“

Do sada je bila teorija, po knjizi „Applied Mathematics for Database
professionals“, Lex de Haan and Toon Koppeliers.

Ono sto sledi je praksa, po knjizi Defensive Database Programming with SQL Server, Alex Kuznetsov.

January 18, 2011

Pogledajmo dijagram promene stanja koji opisuje zivotni put jednog osnovnog sredstva:



Strelice predstavljaju dogadjaje koji dovode do promene stanja. Ovalne figure predstavljaju stanja. Svaka strelica ima pocetno i krajnje stanje.

Vidimo ukupno 9 strelica, pa imamo 9 mogucih promene stanja. Kako ce se stanja definisati, zavisi od konkretne situacije. Ovo je samo slika pogodna za primer za ucenje. U realnom zivotu moze biti desetine i stotine stanja.

Obicno imamo bar jedno stanje od koga sve pocinje, zovemo ga *pocetno stanje celog sistema*. Pocetno stanje celog sistema nema strelica koja u njega idu, sve strelice iz takvog stanja izlaze. U nasem primeru pocetno stanje je opisano recima "Nabavka (ulaz u sistem)". Obicno postoji i bar jedno *završno stanje celog sistema*. Tu sve strelice ulaze, ne izlazi ni jedna. U nasem primeru, završno stanje sistema je opisano recima "otudjivanje (izlaz iz sistema)"

Nacrtni dijagram mozemo predstaviti tablicom (redosled je proizvoljan)

RB	Pocetno stanje	Završno stanje	Kada se desava
----	----------------	----------------	----------------

1	Nabavljeno	Nalazi se u garazi	Sredstvo je kupljeno i dovezeno je u garazu
2	Nalazi se u garazi	U upotrebi je	Sredstvo je izdato na upotrebu nekome (Kamion na putu...)
3	U upotrebi	Nalazi se u garazi	Sredstvo je bilo u upotrebi, pa je vraćeno nazad u garazu (kamion se vratio s puta...)
4	U upotrebi	Na popravci	Sredstvo je bilo na upotrebi, ali se pokvarilo i umesto u garazu, odvezeno je na popravku. Ili je otišlo na redovan servis.
5	U upotrebi	Otudjeno je	Prilikom upotrebe sredstvo je ili potpuno uništeno ili oštećeno toliko da se mora otpisati. Ili je ukradeno = otpisano.
6	U garazi je	Otudjeno je	Sredstvo se nalazi u garazi, i zastarelo je pa se otpisuje. Ili se profdaj - to je neka vrsta otudjivanja.
7	U garazi je	Na popravci	Posle upotrebe je vraćeno u garazu, ali je konstatovano da treba da se popravi, ili ide na redovan servis.
8	Na popravci je	U garazi je	Iz radionice za popravku šalje se u garazu i spremno je za dalju upotrebu
9	Na popravci je	Otudeno je	Poslali smo sredstvo na popravku ali je tamo utvrđeno da se ne vredi popravljati. Ili su ga na popravci nehotice još više oštetili ili ga je neko ukrao - u svakom slučaju, mora se otpisati.

U našem sistemu imamo mnogo osnovnih sredstava, označenih nekakvim ID oznakama. Za nas prijemr, pretpostavićemo da u to brojevi, cifarske oznake,

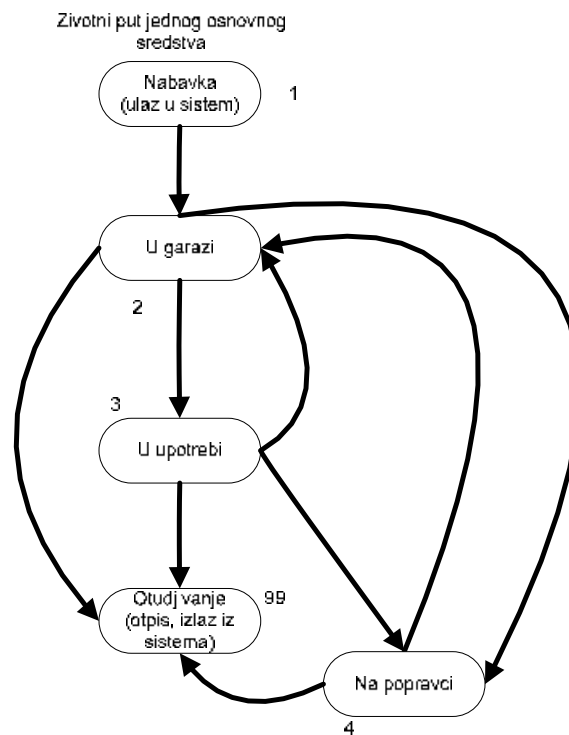
zbog jednostavnosti. U praksi može biti bilo šta, sve zavisi od konkretne situacije.

Zbog lakšeg pisanja se i stanja mogu označiti brojevima, pa imamo

Stanje	Opis
1	Nabavljeno
2	U garazi
3	U upotrebi
4	Na popravci
99	Otudjeno

Brojeve koji označavaju stanja možemo dodeliti proizvoljno. Meni je zgodno da 1 bude *početno stanje celog sistema* i da neki veliki broj, recimo 99 ili 999 ili još više, bude *završno stanje celog sistema*. Ovo je samo zbog lakšeg pisanja. Potpuno je OK dodati stanje 2356 koje nije ni početno ni završno, ako nam to tako sviđa, nije abranjeno ali nije baš zgodno.

Imamo dakle ovo:



Onda bi se dozvoljene promene stanja mogle prikazati ovako:

Tabela DozvoljenePromeneStanja

StaroStanje	NovoStanje
1	2
2	3
2	99
2	4
3	2

3	4
3	99
4	99
4	2

Kaze se da tabela koja prati zivot osnovnih sredstava treba da izgleda nekako ovako:

SerijskiBroj	StaroStanje	NovoStanje	DatumPromene
123456789	1	2	10 Januar 2003
123456789	2	3	15 Januar 2003
123456789	3	2	22 Januar 2003
123456789	2	3	22 Januar 2003
123456789	3	4	25 Januar 2003

Uocite kako je u svakom redu vrednost [StaroStanje] jednaka vrednosti [NovoStanje] iz prethodnog reda. I datumi rastu u nekom logicnom sledu. Nesto ipak nedostaje. Kad je sredstvo u slo u NovoStanje = 3 (u upotrebi), ne pise nigde ko je preuzeo sredstvo. Dodajmo i to.

Tabela: PromeneStanjaOsnovnogSredstva

SerijskiBroj	StaroStanje	NovoStanje	DatumPromene	KoKoristi
123456789	1	2	10 Januar 2003	
123456789	2	3	15 Januar 2003	Zika
123456789	3	2	22 Januar 2003	
123456789	2	3	22 Januar 2003	Laza
123456789	3	4	25 Januar 2003	

Primetite da kolona KoKoristi ima vrednosti samo u onim redovima gde je NovoStanje = 3. Sve ostalo ne bi imalo smisla.

Tabela lepo izgleda, samo treba nekako garantovati integritet podataka. A to znaci da postavimo neka pravila. Evo nekih ogranicenja:

1. SerijskiBroj NOT NULL, mora postojati u tabeli OsnovnaSredstva ako PK
2. StaroStanje NOT NULL
3. NovoStanje NOT NULL
4. Par (StaroStanje,NovoStanje) mora postojati u tabeli DozvoljenePromeneStanja. Ovo je jako vazno. Kad jednom otpisemo sredstvo prevodimo ga u NovoStanje = 99. Posto 99 nemamo u tabeli DozvoljenaStanja.StaroStanje ne bi trebalo dozvolitimozemo da unesemo novi red za dato osnovno sredstvo.
5. KoKoristi moze biti NULL, osim kad je NovoStanje = 3 (sredstvo je u upotrebi), jer zelimo da znamo ko koristi sredstvo
6. Ako je sredstvo na upotrebi, moramo zapisati ko koristi sredstvo, ako nije u upotrebi ne smemo upisati nista u kolonu KoKoristi - ovako nekako: ([KoKoristi] IS NULL kada NovoStanje <> 3) AND (KoKoristi IS NOT NULL kada je NovoStanje = 3))
7. Ako KoKoristi nije NULL, mora postojati u tabeli Zapolenikao PK

8. StaroStanje u svakom redu mora biti isto kao NovoStanje u prethodnom redu.

Zahtevi od 1 do 7 se relativno lako garantuju pomocu NULL/NOT NULL, spoljnjih kljuceva (FK) i validacijom na nivou tabele (CHECK constraints). Validacija na nivou tabele je u Accesu malo nezgodna ali ovde moze, pokazacemo kasnije i kako.

Pravilo 8 je nezgodno. Trazi se da svaki red bude zavistan od prethodnog reda. Po relacionoj teoriji, svaki red je nezavisan od bilo kog drugog reda u tabeli, otuda ono da redosled nije bitan. Mi ovde imamo situaciju gde je redosled i te kako bitan, cak kljucan za reenje naseg problema.

Ako je red zavisian od nekog prethodnog reda, to je kao neka hijerarhija, prethodni redovi su neka vrsta roditelja za sledese redove. U hijerarhiji, svaki red ima tacno jednog roditelja, a roditelji mogu imati vise dece. U nasem slucaju, zahteva se da svaki red ima tacno jedno „dete“ i da svaki red ima tacno jednog roditelja. Ovo je strozije nego obicna hijerarhija, ali u svakom slucaju vazi „svaki red mora imati roditelja“

Da bi uspostavili hijerarhijsku zavisnost, sagradicemo "self referencing Foreign Key". Za to nam treba pogodan Primary Key u tabeli PromeneStanjaOsnovnogSredstva.

Ako pogledamo u test podatkom vidimo das u kandidati za PK su kombinacije

(SerijskiBroj, StaroStanje, datumPromene) i
(SerijskiBroj, NovoStanje, DatumPromene)

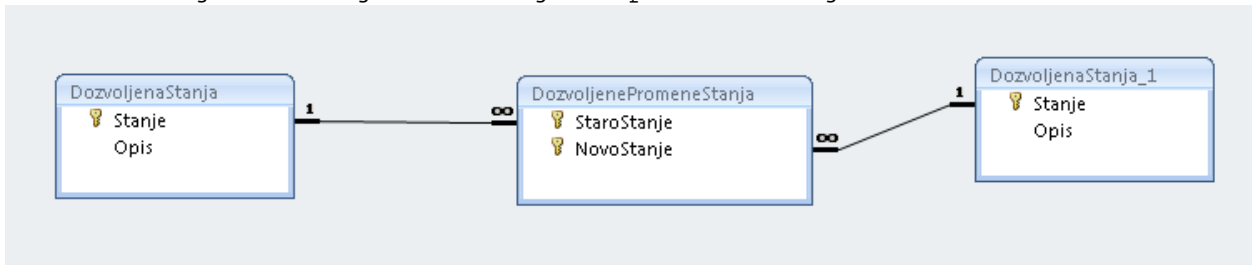
Poston am treba da je NovoStanje iz prethodnog reda roditelj za StaroStanje u novom redu, onda cemo za PK izabrati kombinaciju (SerijskiBroj, NovoStanje, DatumPromene). Ovo je potreban, ali ne i dovoljan uslov za ogranicenje broj 8.

Da malo zastanemo i napravimo nekoliko tabela u Accesu:

Stanje	Opis
1	Nabavljeno
2	U garazi
3	U upotrebi
4	Na popravci
99	Otudjeno

StaroStanje	NovoStanje
1	1
1	2
2	3
2	4
2	99
3	2
3	4
3	99
4	2
4	99

Odnos dozvoljenih stanja i dozvoljenih promene stanja:

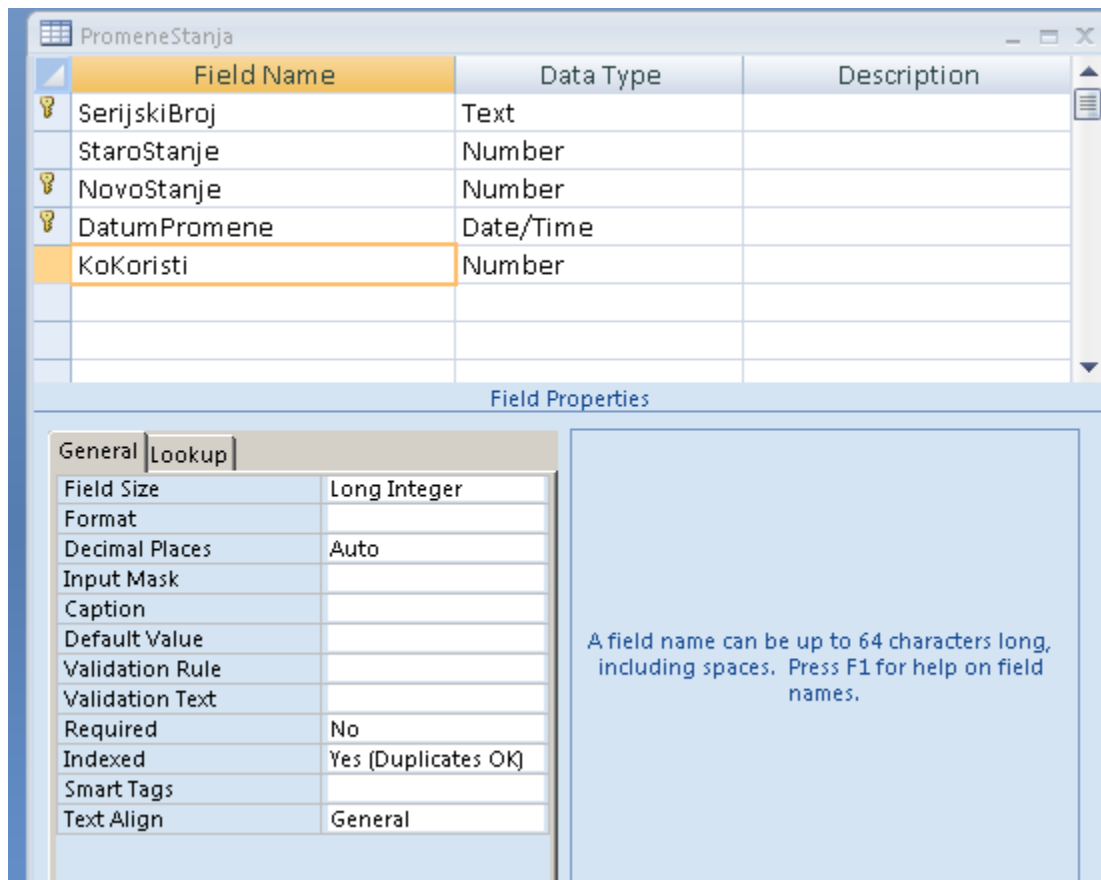


Tabele sa osnovnim sredstvima i radnicima:

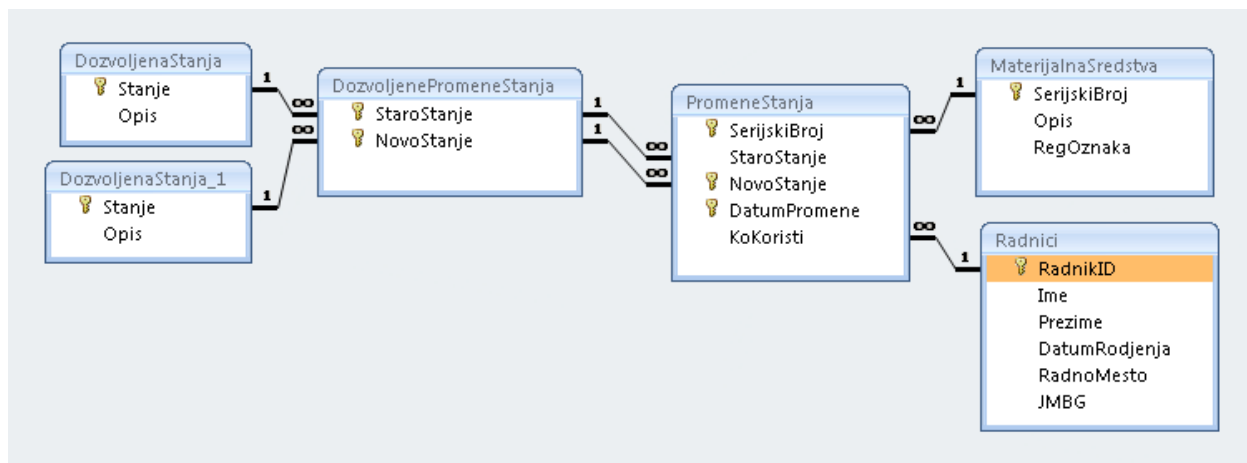
SerijskiBroj	Opis	KegOznaka
120450709	Ramion FAP 13 plavi	TZ 211-440
587654321	Ramion FAP 13 Zeleni	TZ 211-355
2234456789	Trayta Matriks	923-2WT
FUJ10C705A	Priker Fujitsu Color	

RadnikID	Ime	Prezime	DatumRođenja
1	Salko	Stralagac	
2	Mihod	Orbic	
3	Cvijetin	Markovic	
4	Evica	Zizak	
5	Miro	Markovic	
6	Dragan	Vasic	
7	Idemir	Salovic	
8	Jurica	Vrdoljak	
9	Nedžad	Orbic	
10	Risto	Dragic	
11	Amel	Alibegovic	

Tabela koju pokušavamo da napravimo:

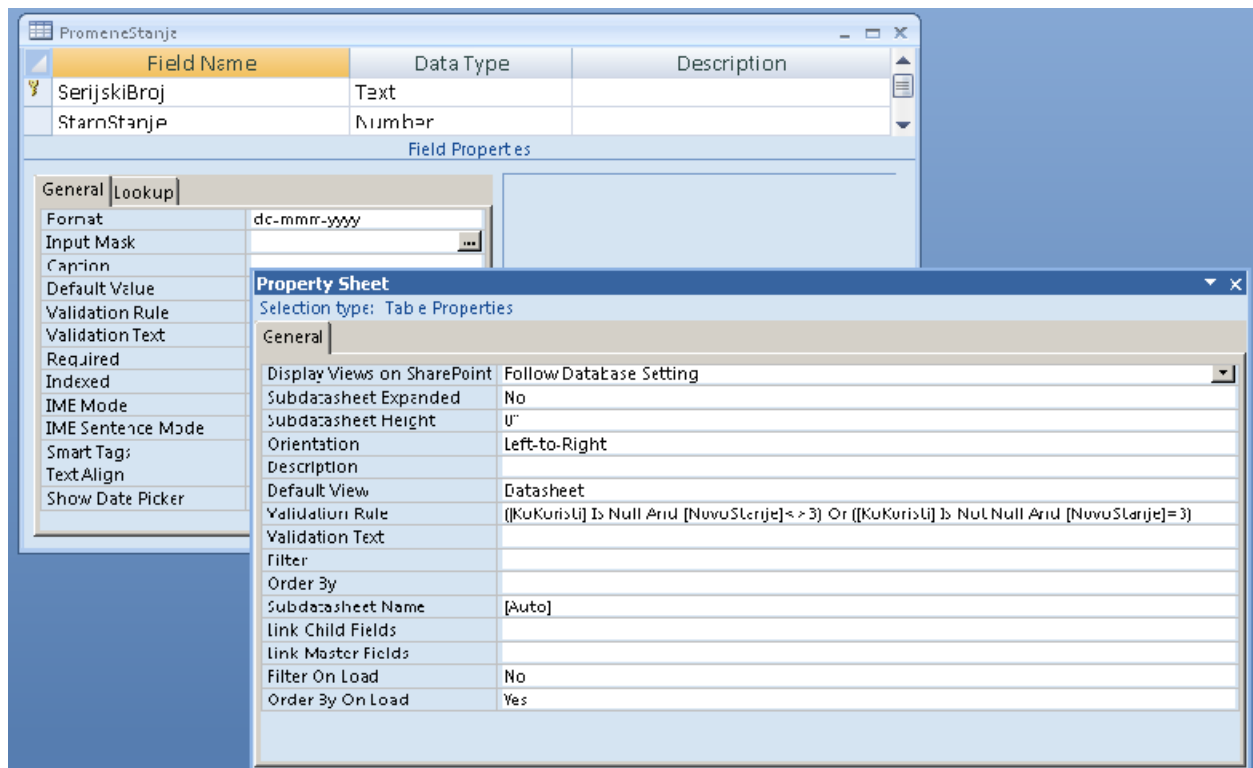


Za sada relacije izgledaju ovako:



Relacije koje smo poatavili garantuju pravila 1,2,3,4 i 7. Ostaje da resimo 5,6 i 8.

Praviloa 5,6: Ako je NovoStanje = 3 onda mora postojati vrednost [KoKoristi], a ako NovoStanje <>3 onda ko koristi MORA biti NULL. Ovo se postize tako sto se otvori tabela u design modu, otvori se properties dijalog za tabelu i upise se Validation Rule za tabelu:



Ako se ne vidi, Validation Rule za tabelu glasi:

`([KoKoristi] Is Null And [NovoStanje]<>3) Or ([KoKoristi] Is Not Null And [NovoStanje]=3)`

Uglaste zagrade su obavezne.

Pokusajmo da unesemo nekoliko test podataka, da vidimo sta prolazi a sta ne. Ovo je proslo bez problema- ispravna kombinacija (StaraStanje, NovoStanje) NovoStanje = 2, nema KoKoristi:

SerijskiBroj	StaraStanje	NovoStanje	DatumPromene	KoKoristi
123456789	1	2	11-Jan-2003	
*				

I ovo prolazi, jer je NovoStanje = 3 i imamo KoKoristi:

SerijskiBroj	StaraStanje	NovoStanje	DatumPromene	KoKoristi
123456789	1	2	11-Jan-2003	
123456789	2	3	18-Jan-2011	1
*				

Pokusajmo da vratimo sredstvo u garazu, i da zadržimo vrednost u KoKoristi - neće moći:

The screenshot shows a table with the following data:

SerijskiBr	StaroStanje	NovoStanje	DatumPromene	KoKoristi
123456789	1	2	11-Jan-2003	
123456789	2	3	18-Jan-2011	1
123456789	3	2	18-Jan-2011	1

The error dialog box contains the following text:

One or more values are prohibited by the validation rule '([KoKoristi] Is Null And [NovoStanje]<>3) Or ([KoKoristi] Is Not Null And [NovoStanje]=3)' set for 'PromeneStanja'. Enter a value that the expression for this field can accept

Mocna stvar! Sad vidite kako radi Validation Rule na nivou tabele. Uklonim KoKoristi i novi red prolazi. Za sada je sve OK, jer dajem redove u logicnom redosledu.

The screenshot shows the same table as before, but with a new empty row added at the bottom, indicated by an asterisk (*).

Sad cu da pokusam nedozvoljenu promenu, recimo sa 2 na 1 - iz garaze u stanje ,nabavljeno' - ni to ne prolazi:

The screenshot shows the table with the following data:

SerijskiBr	StaroStanje	NovoStanje	DatumPromene	KoKoristi
123456789	2	3	18-Jan-2011	1
123456789	3	2	18-Jan-2011	
123456789	2	1	18-Jan-2011	

The error dialog box contains the following text:

You cannot add or change a record because a related record is required in table 'DozvoljenePromeneStanja'.

Za sada smo dobri.

Sada cu da narusim redosled, da dam dozvoljenu promenu, ali koja nije logicna. Moj poslednji dobar red kaze da je vozilo u garazi (NovoStanje = 2). Sad cu da pokusam da unesem promenu „sa opravke ide vozilo na otpis“, iz stanja 4 u stanje 99:

SerijskiBr	StaroStanje	NovoStanje	DatumPromene	KoKoristi
123456789	1	2	11-Jan-2003	
123456789	2	3	18-Jan-2011	1
123456789	3	2	18-Jan-2011	
123456789	4	99	18-Jan-2011	
*				

Record: 4 of 4 No Filter Search

Access me je pustio da završim transakciju:

SerijskiBr	StaroStanje	NovoStanje	DatumPromene	KoKoristi
123456789	1	2	11-Jan-2003	
123456789	2	3	18-Jan-2011	1
123456789	3	2	18-Jan-2011	
123456789	4	99	18-Jan-2011	
*				

Record: 5 of 5 No Filter Search

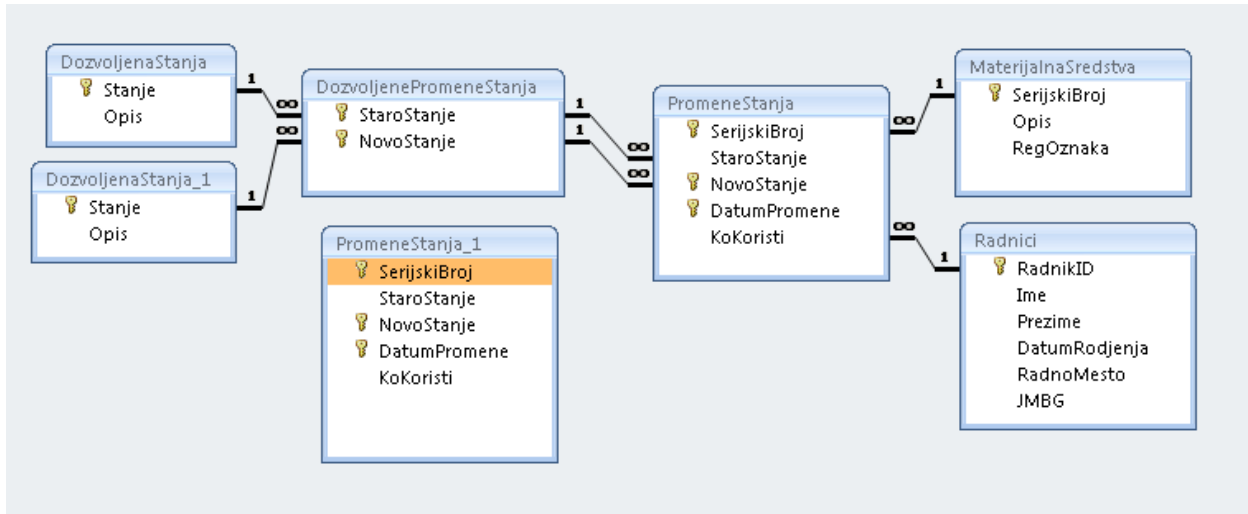
Ovim sam narusio pravilo 8 koje kaže: NovoStanje u novom rdu mora biti jednako starom stanju u prethodnom redu. Znači, umesto 4→99 trebalo bi da imam promenu 2 → X.

Da pokušam sa hijerarhijskom relacijom (prethodno sam obrisao los red), sada ponovo imam:

SerijskiBr	StaroStanje	NovoStanje	DatumPromene	KoKoristi
123456789	1	2	11-Jan-2003	
123456789	2	3	18-Jan-2011	1
123456789	3	2	18-Jan-2011	
*				

Record: 4 of 4 No Filter Search

Pocinjem da gradim relaciju sa tabele PromeneStanja na samu sebe. U shemu relacija dodajem jos jednom tabelu PromeneStanja i ona se pojavljuje kao PromeneStanja_1. Sada samo treba da spojim PK sa PromeneStanja_1 na odgovarajuće kolone na PromeneStanja (na samu sebe).



Ima jedan problem. PK za PromeneStanja_1 ima TRI kolone (SerijskiBroj, NovoStanje, DatumPromene) i ceo mora da bude deo relacije. Gde je problem? serijskiBroj ce ici na SeriskiBroj. NovoStanje ce ici na *StaroStanje* u novom redu. DatumPromene nema gde da ide. Nije garantovano da je DatumPromene za stari red isti kao i DatumPromene za novi red. Znacni, treba nam jos jedna kolona u tabeli PromeneStanja, da bismo mogli da prihvatimo kompletan PK. Inacenema nista od relacije (FK) a to nam bas treba.

Ovu zavisnost je prvi primetio i primenio i opisao u svojoj knjizi pomenuti Alex Kuznetsov.

Dodajemo novu kolonu:

Field Name	Data Type
SerijskiBroj	Text
StaroStanje	Number
NovoStanje	Number
DatumPromene	Date/Time
DatumStarogStanja	Date/Time
KoKoristi	Number

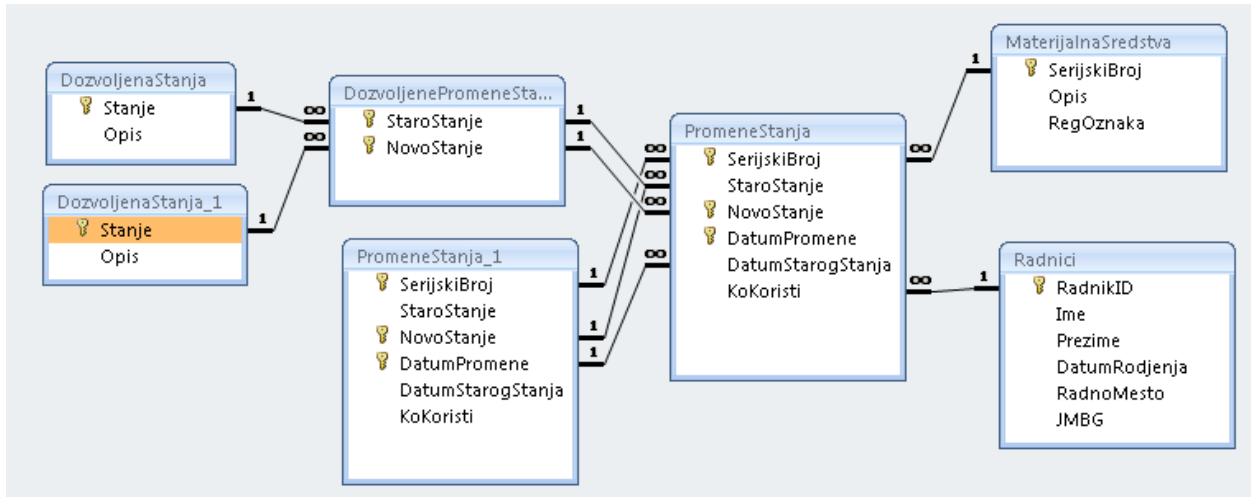
Field Properties

General | Lookup | Format

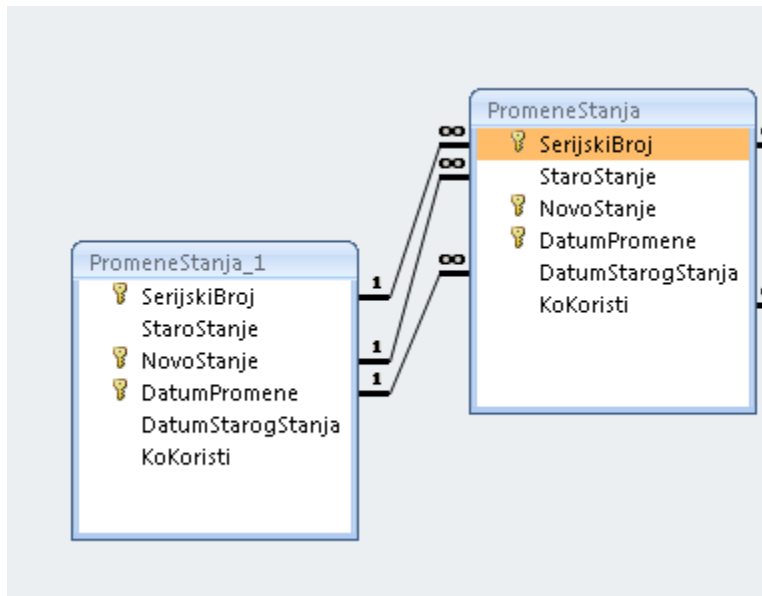
Pozicija nove kolone u design modu je bitna, zbog jasnije slike na dijagramu relacija.

Sada imamo dovoljno kolona da napravimo relaciju. Moracemo da obrisemo postojece podatke, jer ce DatumStarogStanja biti NULL svuda a to nam moze stvarati probleme. Brisemo dakle sve redove iz tyabele, pre nego pokusamo da stavimo novu relaciju (self referencing Foreign key).

Ovako ce izgledati dijagram relacija:



Da bi se bolje videlo o cemu se radi, sakricu (necu obrisati) tabelu DozvoljenePromeneStanja:



Sada smo postigli da roditelj novog red mora biti jedan od prethodnih redova za posmatrano materijalno sredstvo. To jos nije ceo uslov 8, koji trazi da roditelj bude tacno onaj red koji ima najveći datum promene stanja. To u sledecem koraku.

Pokusajmo da unesemo isti pocetni red, SerijskiBroj = 123456789, iz stanja 1 u stanje 2, kao sto smo to radili pre poslednje promene. Nazalost, ne ide:

Uz malo experimentisanja, ispostavlja se da prvi red za posmatrano materijalno sredstvo mora da bude (promena stanja iz 1 u 1, DatumpromeneStanja = DatumStarogStanja). Ovo prolazi i nista drugo osim toga, za prvi red:

	SerijskiBroj	StaroStanje	NovoStanje	DatumPromene	DatumStarogStanja	KoKoristi
	123456789	1	1	18-Jan-2011	2011-01-18	
*						

Deluje nezgrapno ali tako je. Ispada da je ovaj red sam sebi otac. Svaka druga kombinacija nema oca, pa F ne dozvoljava upis. Mislim da ovo mozemo da progutamo.

Idemo dalje. Moze li ovo da prodje?

	SerijskiBroj	StaroStanje	NovoStanje	DatumPromene	DatumStarogStanja	KoKoristi
	123456789	1	1	18-Jan-2011	2011-01-18	
	123456789	1	2	19-Jan-2011	2011-01-18	
*						

Moze:

	SerijskiBroj	StaroStanje	NovoStanje	DatumPromene	DatumStarogStanja	KoKoristi
	123456789	1	1	18-Jan-2011	2011-01-18	
	123456789	1	2	19-Jan-2011	2011-01-18	
*						

Nista drugo ne bi proslo jer iz 1 ide strelica samo u 2.


Prolazi i izdavanje vozila radniku 1:

	SerijskiBr	StaroStanje	NovoStanje	DatumPromene	DatumStarogStanja	KoKoristi
+	123456789	1	1	18-Jan-2011	2011-01-18	
+	123456789	1	2	19-Jan-2011	2011-01-18	
+	123456789	2	3	19-Jan-2011	2011-01-19	1
*						

Sad cemo da pokusamo nesto sto nemamo kao roditelja, a validna je kombinacija:

	SerijskiBr	StaroStanje	NovoStanje	DatumPromene	DatumStarogStanja	KoKoristi
+	123456789	1	1	18-Jan-2011	2011-01-18	
+	123456789	1	2	19-Jan-2011	2011-01-18	
+	123456789	2	3	19-Jan-2011	2011-01-19	1
+	123456789	4	99	18-Jan-2011	2011-01-19	
*						

Microsoft Office Access

 You cannot add or change a record because a related record is required in table 'PromeneStanja'.

OK Help

Nije proslo jer ni jedna prethodna promena nije zavrshila sa 4. Ovo je rezultat nove relacije koju smo poatavili.

Sledeca logicna promena treba da ima StaroStanje = 3 i DatumStarogStanja = 19 Jan 2011. Medjutim, pogledajte sta ce da prodje:

	SerijskiBr	StaroStanje	NovoStanje	DatumPromene	DatumStarogStanja	KoKoristi
+	123456789	1	1	18-Jan-2011	2011-01-18	
+	123456789	1	2	19-Jan-2011	2011-01-18	
+	123456789	2	3	19-Jan-2011	2011-01-19	1
+	123456789	1	2	18-Jan-2011	2011-01-18	
*						

Pokusavam da kazem da sada ponovo imam promenu iz 1 u 2 (drugi red) iako mi treba neka promena iz stanja 3

Nazalost, proslo je:

	SerijskiBroj	StaroStanje	NovoStanje	DatumPromene	DatumStarogStanja	KoKoristi
+	123456789	1	1	18-Jan-2011	2011-01-18	
+	123456789	1	2	19-Jan-2011	2011-01-18	
+	123456789	2	3	19-Jan-2011	2011-01-19	1
+	123456789	1	2	18-Jan-2011	2011-01-18	
*						

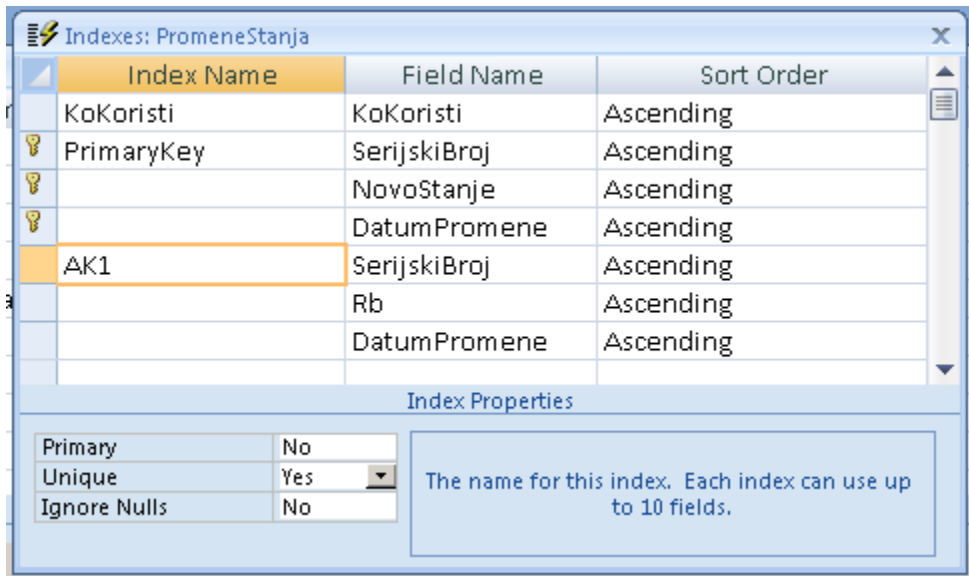
Znaci da FK na samu sebe nije dovoljno jako da garantuje i redosled.

Dodacu jos dve kolone u tabelu PromenaStanja i prosiriti Validation Rule za tabelu. Prvo kolone, Rb i StariRb, integerer, obe kolone Required = Yes.

Field Name	Data Type
SerijskiBroj	Text
StaroStanje	Number
NovoStanje	Number
DatumPromene	Date/Time
DatumStarogStanja	Date/Time
KoKoristi	Number
Rb	Number
StariRb	Number

Field Properties	
Field Size	Integer
Format	
Decimal Places	Auto
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	Yes
Indexed	No

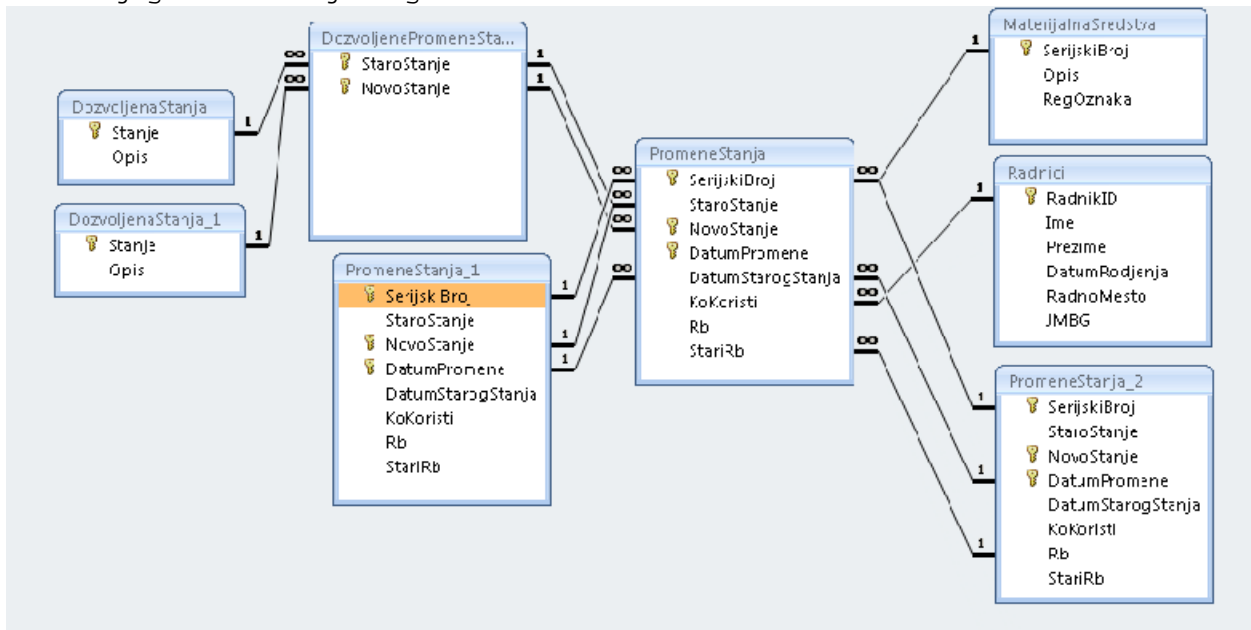
Treba mi i novi index, unique, sican primarnom klucu:



Sada cu i za taj index da napravim relaciju tabele na samu sebe.

Prvo moram da obrisem sve redove. *Ovde sam primetio da ne mogu da obrisem sve redove odjednom, jer su medjusobno povezani relacijom. Ovo i nije losa osobina, kao usputni efekat dobili smo strogu kontrolu brisanja.*

Sada dijagram relacija izgleda ovako:



Ovo se referencira u stranom kljucu: SerijskiBroj na SerisjkiBroj, Rb na StariRb i SatumPromene na DatumStarigStanja.

Da ponovo pokusamo da testiramo bazu. Ponovo prvi red mora da izgleda ovako i nikako drugo:

	SerijskiBr	StaroStanje	NovoStanje	DatumPromene	DatumStarogStanja	KoKoristi	Rb	StariRb
	123456789	1	1	10-Jan-2011	2011-01-10		1	1
*								

Za prvi red mora da vaze sledeca tri uslova:

- (Rb = StariRB = 1) i
- (StaroStanje = NovoStanje = 1) i
- (DatumPromene = DatumStarogStanja)

Prva dva reda:

	SerijskiBr	StaroStanje	NovoStanje	DatumPromene	DatumStarogStanja	KoKoristi	Rb	StariRb
	123456789	1	1	10-Jan-2011	2011-01-10		1	1
	123456789	1	2	12-Jan-2011	2011-01-10		2	1
*								

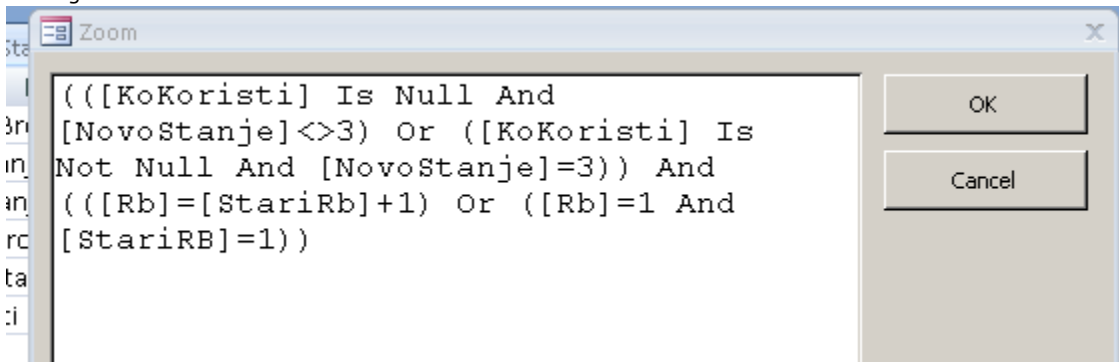
Nigde nismo rekli da Rb mora da se povecava za 1, pa je moguće i ovo:

	SerijskiBr	StaroStanje	NovoStanje	DatumPromene	DatumStarogStanja	KoKoristi	Rb	StariRb
	123456789	1	1	10-Jan-2011	2011-01-10		1	1
	123456789	1	2	12-Jan-2011	2011-01-10		2	1
	123456789	2	3	13-Jan-2011	2011-01-12	1	157	2
*								

Sledeci StariRB morace da bude 157, to je OK, ali nekako nije lepo da skocimo sa 2 na 157. Dodajmo jos jedno ogranicenje na nivou tabele - table Validation rule(table level CHECK CONSTRAINT). Prethodno brisem onaj red sa Rb = 157, i imam ponovo:

	SerijskiBr	StaroStanje	NovoStanje	DatumPromene	DatumStarogStanja	KoKoristi	Rb	StariRb
	123456789	1	1	10-Jan-2011	2011-01-10		1	1
	123456789	1	2	12-Jan-2011	2011-01-10		2	1
*								

Ovo je novo Validation Rule za tabelu:



Obratite paznju na zagrade. Ovako bi izgledalo formatirano:

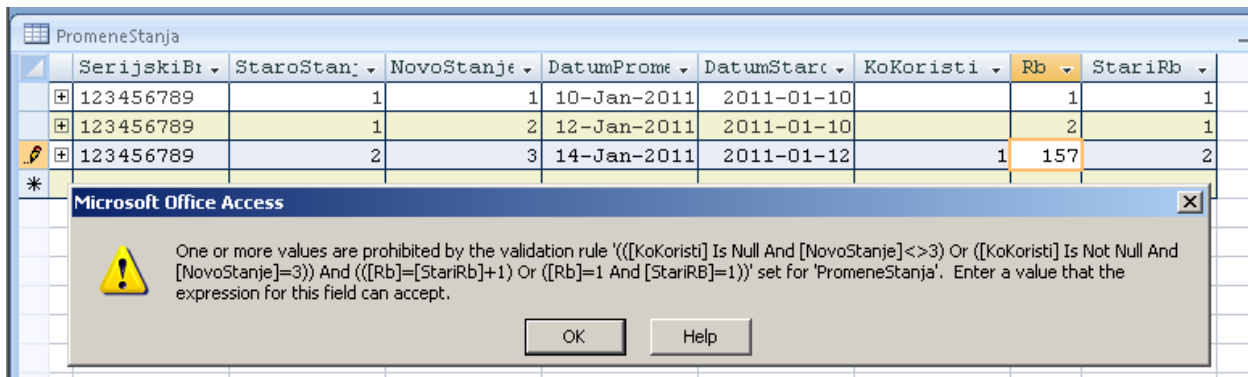
```

(
([KoKoristi] Is Null And [NovoStanje]<>3)
Or
([KoKoristi] Is Not Null And [NovoStanje]=3)
)
And
(
([Rb]=[StariRb]+1)
Or
([Rb]=1 And [StariRB]=1)
)

```

Staro Validation Rule nije imalo ovoliko zagrada.

Ako sada pokusam da unesem Rb = 157 nece mi dozvoliti:



Posto je Validation Rule sada prilično veliko i zamršeno, nece uvek biti jasno koji deo je narusen. Zato ce pretpostavlja da ce sav unos biti resen kodom id a ce krajnji korisnik videti jako malo o dove komplikovane structure. O tome verovatno sutra.

U svakom slucaju, los unos nije prosao, mora ovako ili nikako:

SerijskiBr	StaroStanje	NovoStanje	DatumPromene	DatumStaranja	KoKoristi	Rb	StariRb
123456789	1	1	10-Jan-2011	2011-01-10		1	1
123456789	1	2	12-Jan-2011	2011-01-10		2	1
123456789	2	3	14-Jan-2011	2011-01-12	1	3	2

Pokusacu da kazem da je sledeca promena 1-2 (ono u drugom redu):

SerijskiBr	StaroStanje	NovoStanje	DatumPromene	DatumStarca	KoKoristi	Rb	StariRb
123456789	1	1	10-Jan-2011	2011-01-10		1	1
123456789	1	2	12-Jan-2011	2011-01-10		2	1
123456789	2	3	14-Jan-2011	2011-01-12	1	3	2
123456789	1	2	18-Jan-2011	2011-01-12		4	2

Microsoft Office Access

One or more values are prohibited by the validation rule '([Koristi] Is Null And [NovoStanje]<>0) Or ([Koristi] Is Not Null And [NovoStanje]=3) And (([Rb]=[StariRb]+1) Or ([Rb]=1 And [StariRb]=1))' set for 'PromeneStanja'. Enter a value that the expression for this field can accept.

OK Help

Nije proslo. A bilo je proslo pre uvođenja relacije po Rb.

Za sada, deluje da smo resili sve uslove koje smo naveli. Da li je tacno? Iam li nesto sto bi moglo da pokvari sliku?

Obrisacu sve redove osim prvog (jedan po jedan naravno, jer mi drugacije ne dozvoljava ☺)

Imam dakle ovo:

SerijskiBr	StaroStanje	NovoStanje	DatumPromene	DatumStarca	KoKoristi	Rb	StariRb
123456789	1	1	10-Jan-2011	2011-01-10		1	1
*							

Pokusacu da unesem promenu 1 na 1 vise puta:

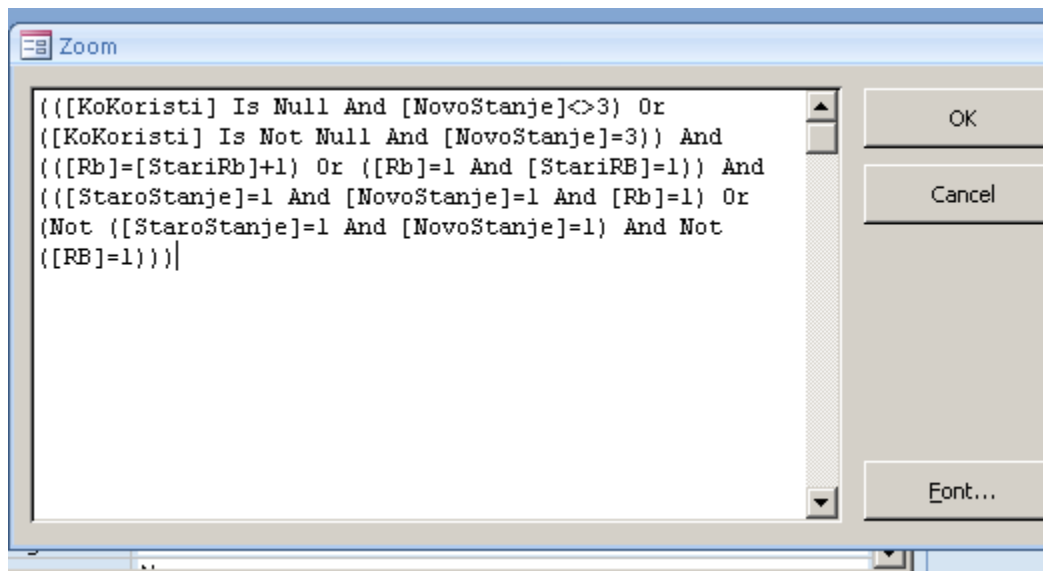
SerijskiBr	StaroStanje	NovoStanje	DatumPromene	DatumStarca	KoKoristi	Rb	StariRb
123456789	1	1	10-Jan-2011	2011-01-10		1	1
123456789	1	1	18-Jan-2011	2011-01-10		2	1
*							

Nazalost, Access mi je dozvolio da unesem promenu:

SerijskiBr	StaroStanje	NovoStanje	DatumPromene	DatumStarca	KoKoristi	Rb	StariRb
123456789	1	1	10-Jan-2011	2011-01-10		1	1
123456789	1	1	18-Jan-2011	2011-01-10		2	1
*							

Ovo nema smisla. Kako da to sprecim? Zabranicu da bude RB <> 1 ako je StaroStanje =NovoStanje = 1. To ce jos vise da zakomplikuje validation Rule na nivou tabele, ali sta se moze, takav je zivot. Dok Microsoft ne popravi SQL editor za Access, mucicemo se ovako.

Ako promenimo table Validation Rule



Pazite na zagrade!

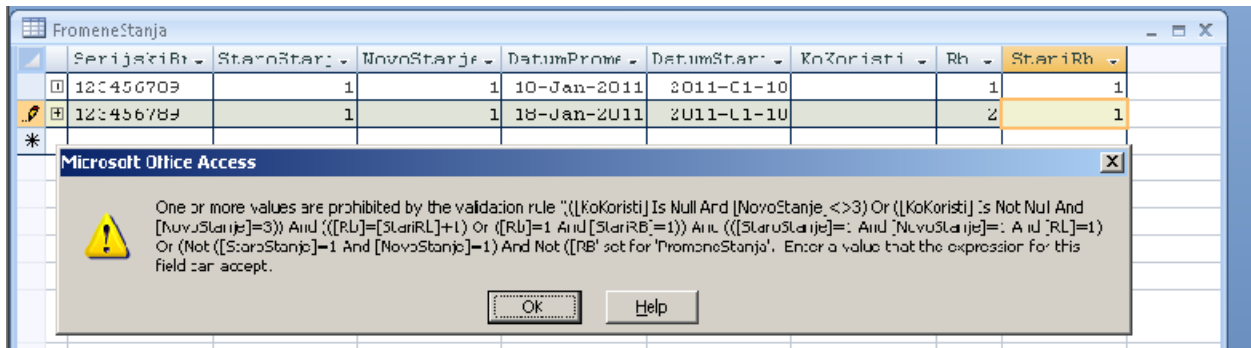
Ovako je kad se Copy/Paste iz Validation Rule property:

```
(([KoKoristi] Is Null And [NovoStanje]<>3) Or ([KoKoristi] Is Not Null And
[NovoStanje]=3)) And (([Rb]=[StariRb]+1) Or ([Rb]=1 And [StariRB]=1)) And
(([StaroStanje]=1 And [NovoStanje]=1 And [Rb]=1) Or (Not ([StaroStanje]=1 And
[NovoStanje]=1) And Not ([RB]=1)))
```

Ovako izgleda formatirano:

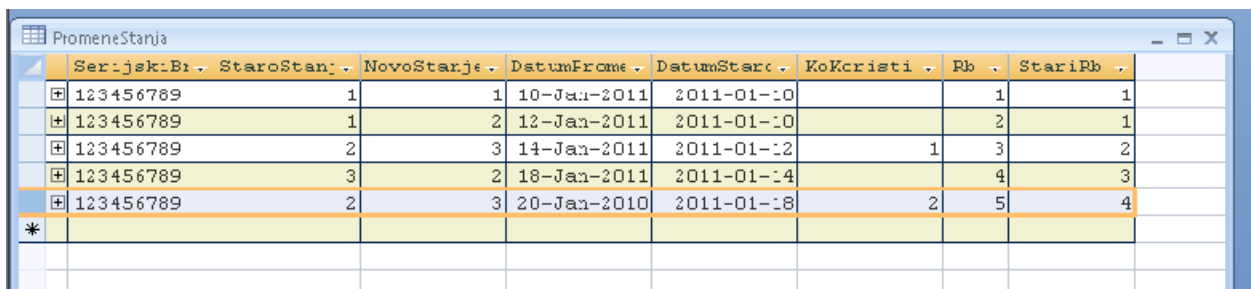
```
(
  ([KoKoristi] Is Null And [NovoStanje]<>3)
  Or
  ([KoKoristi] Is Not Null And [NovoStanje]=3)
)
And
(
  ([Rb]=[StariRb]+1)
  Or
  ([Rb]=1 And [StariRB]=1)
)
And
(
  ([StaroStanje]=1 And [NovoStanje]=1 And [Rb]=1)
  Or
  (Not ([StaroStanje]=1 And [NovoStanje]=1) And Not ([RB]=1))
)
```

Onda nas nece pustiti da unesemo ovo:



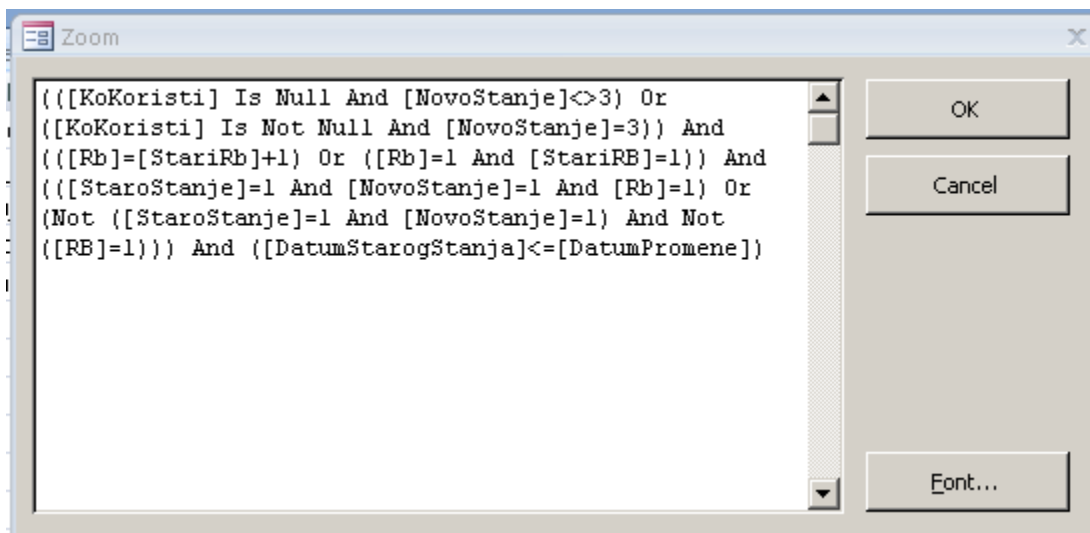
Verovatno je umesto ovog polednjeg pravila moglo da se kaže da mora biti UNIQUE (SerijskiBroj, StaroStanje, NovoStanje, StariRb), nisam probao, ali mi deluje obećavajuće. Ako je to tačno, onda je bolje engo table validation rule, ako nista ono zbog jasnoće pisanja.

Ima li jos nesto? Ima, i to da se covek obruka. Pogledajte poslednji red na slici:



DatumStarogStanaj = 18 januar **2011**, DatumPromene = 20 januar **2010**
 Ne sme biti datum novog stanja manji nego datum starog stanja. Neverovatno je koliko s eljudi opece i obruka na ovako glupavim propustima.

Moramo i to da resimo:



Dodatak na pravilo je oznacen zutom bojom:

```
(([KoKoristi] Is Null And [NovoStanje]<>3) Or ([KoKoristi] Is Not Null And [NovoStanje]=3)) And (([Rb]=[StariRb]+1) Or ([Rb]=1 And [StariRb]=1)) And (([StaroStanje]=1 And [NovoStanje]=1 And [Rb]=1) Or (Not ([StaroStanje]=1 And [NovoStanje]=1) And Not ([Rb]=1))) And ([DatumStarogStanja]<=[DatumPromene])
```

Sada ne moze pogresan datum:

The screenshot shows a Microsoft Access table named 'PromeneStanja'. The table has columns: SerijskiB, StaroStanje, NovoStanje, DatumPromene, DatumStarogStanja, KoKoristi, Rb, and StariRb. The data is as follows:

SerijskiB	StaroStanje	NovoStanje	DatumPromene	DatumStarogStanja	KoKoristi	Rb	StariRb
123456789	1	1	10-Jan-2011	2011-01-10		1	
123456789	1	2	12-Jan-2011	2011-01-10		2	
123456789	2	3	14-Jan-2011	2011-01-12	1	3	
123456789	3	2	18-Jan-2011	2011-01-14		4	
2011 01 18	2	3	20 Jan 2010	2011 01 18	2	5	

A validation error dialog box is displayed over the table. The dialog box contains the following text:

One or more values are prohibited by the validation rule '(([KoKoristi] Is Null And [NovoStanje]<>3) Or ([KoKoristi] Is Not Null And [NovoStanje]=3)) And (([Rb]=[StariRb]+1) Or ([Rb]=1 And [StariRb]=1)) And (([StaroStanje]=1 And [NovoStanje]=1 And [Rb]=1) Or (Not ([StaroStanje]=1 And [NovoStanje]=1) And Not ([Rb]=1))) And ([DatumStarogStanja]<=[DatumPromene])'. Enter a value that the expression for this field can accept.

The dialog box has 'OK' and 'Help' buttons.

Ispravimo gresku i sada moze:

The screenshot shows the same Microsoft Access table 'PromeneStanja' after the error is fixed. The data is as follows:

SerijskiB	StaroStanje	NovoStanje	DatumPromene	DatumStarogStanja	KoKoristi	Rb	StariRb
123456789	1	1	10-Jan-2011	2011-01-10		1	1
123456789	1	2	12-Jan-2011	2011-01-10		2	1
123456789	2	3	14-Jan-2011	2011-01-12	1	3	2
123456789	3	2	18-Jan-2011	2011-01-14		4	3
123456789	2	3	20-Jan-2011	2011-01-18	2	5	4

Naravno da cemo od korisnika sakriti kompleksnost. Tu sad dolazi do izrazaja programersko umece. Trba napisati procedure koje ce da obave unos, a da pre toga provere sva moguca pravila, d abi izbegli misteriozne Acessove poruke.

Lepota ovog resenja je u tome sto se ovakva jedna struktura moze dodati na postojece baze. Vase tabele sa Radnicima i osnovnim sredstvima se nece promeniti. Nove tabele su DozvoljenaStanja, DozvoljeenPromeneStanja, PromeneStanja. Sve tri tabele su genericke - struktura im ne zavisi od vaseg poslovnog procesa. Od poslovnog procesa zavisi kako cemo napraviti dijagram promene stanja, to jest koje cemo podatke ubaciti u tabelu DozvoljenePromeenStanja i DozvoljeneStanja.

Medjutim, pogresno bi bilo naprosto kopirati ovakvo resenja bez razumevanja.

Dosta za danas, preko 20 stranica a nije takvo stivo da se brzo cita.

Ako niko en nadje neke logicke falinke i greske za brukanje, mozemo za dan-
dva da predjemo na programiranje.

