

8-1 Modelovanje istorijskih podataka

Smisao

- Koliko si bio visok sa 5 godina ? A sa 10 ? Koliko si visok sada ?
- Da je neko zapisao ove podatke tada sada bi imao istorijske podatke.
- Biznisi moraju da vode evidenciju istorijskih podataka.
- Pomaže im da pronađu trendove i šeme koji su osnova za biznis inovacije ili poboljšanja procesa
- Npr, istorija iznajmljivanja filma je značajna za prodavnicu videa. To kaže menadžeru koji filmovi su popularni a koji bi trebalo da se stave na police u pozadini

Model podataka tokom vremena

- Kada je neophodno modelovati podatke tokom vremena ?
- Treba pitati klijenta: Da li je audit trail zahtevano ? Mogu li se vrednosti atributa promeniti tokom vremena ? Mogu li se relacije promeniti tokom vremena ? Da li trebate izveštaje o proizvodnji sa starim datumima ? Da li trebate zadržati prethodne verzije podataka ? Ako da koliko dugo ?
- Istorijski podatak : Podataka koji je zapis o promenama tokom vremena.
- Audit trail: Korak po korak zapis o promenama podataka
- Trebalo bi dogovoriti zahteve za smeštanje istorijskih podataka sa korisnikom. Smeštanje nepotrebnih istorijskih podataka može biti skupo. Traži prostor – ako je orman sa fajlovima ili u db. Traži resurse zaposlenih, jer neko mora biti odgovoran za updejtovanje istorijskih zapisa sa najnovijim podacima, da bi se osiguralo da ima dovoljno prostora kako podaci dolaze i za alokaciju više prostora ili pomeranje podataka ako je neophodno.

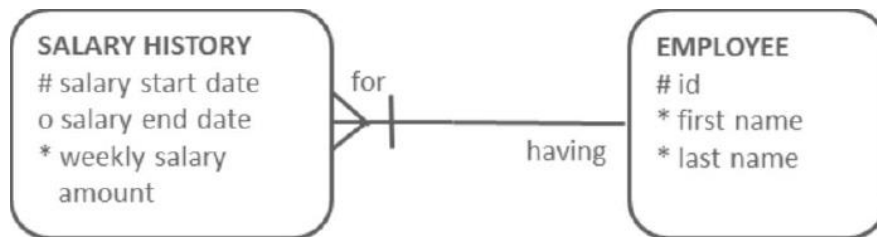
Primer podataka tokom vremena (Data Over Time)

- Organizacija treba da čuva podatke o platama zaposlenih; svi zaposleni se plaćaju nedeljno
- U početku, modelovan je EMPLOYEE entitet
- Dodatni zahtevi sada traže da organizacija vodi istorijske zapise kako i kada su plate zaposlenih se menjale tokom rada u firmi
- Model na slici neće raditi bazirano na novim zahtevima, pošto postoji samo jedna vrednost za salary se može čuvati. Ako se plata menjala, ne postoji mogućnost zapisa pre nego je updejtovana. Ako bi smo smestili ranije plate svih zaposlenih kao svotu novca i početni datum ovo bi bila ponavljajuća grupa više vrednosnih atributa, pa bi prekršila 1NF.



Promene modela Salary

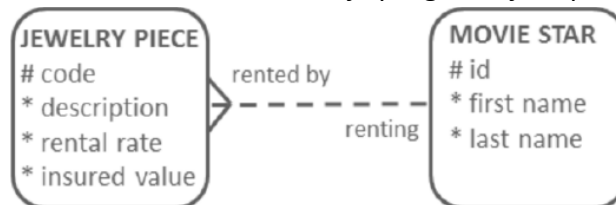
- Model plata se menja tokom vremenom dodatak je SALARY HISTORY entitet



- UID od SALARY HISTORY entiteta je relaciji EMPLOYEE id i salary start date
- Plata i datum je opcionalno pošto ako je to trenutna plata, ne postoji end date (za sada)
- Trenutna plata je ona čiji end date je null
- Takođe bi trebalo da bude programska constraint koja obezbeđuje da zaposleni ima samo jednu platu u jednom trenutku: početni datum nove plate ne može biti pre end date prethodne plate

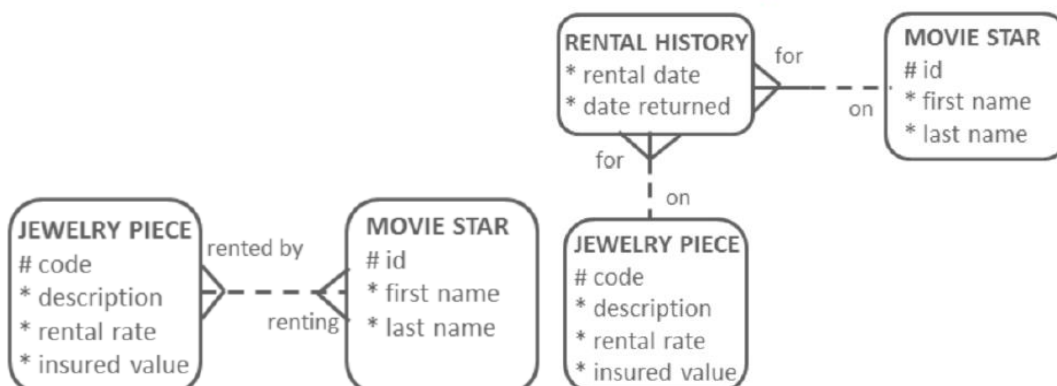
Model Rental Over Time

- Prodavnica nakita iznajmljuje komade nakita (ogrlice, narukvice...) filmskim zvezdama za posebne prilike, kao što su ceremonije proglašenja ili premijere filmova



- Žele da vode evidenciju rental history za komade nakita
- Sledeći ER model će voditi evidenciju o trenutnom iznajmljivaču (renter) komada nakita
- Kako bi se izmenila relacija za vođenje računa o istoriji iznajmljivanja ?
- Da bi se vodilo računa o istoriji iznajmljivanja trebalo bi primeniti M:M relaciju između JEWELRY PIECE i MOVIE STAR a zatim je izmeniti sa intersekcioni entitetom RENTAL HISTORY
- Sledeće bi trebalo odabrati UID za RENTAL HISTORY

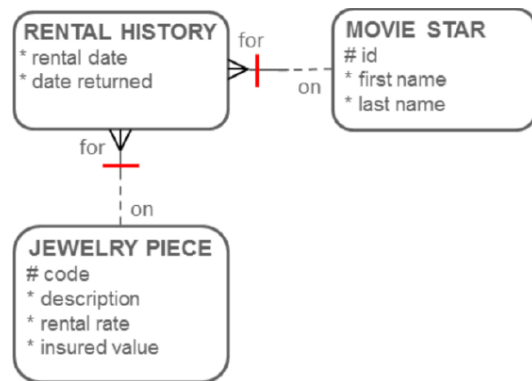
The M:M relationship is resolved with an intersection entity.



- Takođe su dodati atributi rental date i date returned za RENTAL HISTORY entitet

Određivanje UID

- Opcija 1 - zabranjene relacije: crtanje zabranjenih relacija nije baš najbolji izbor za UID ovde pošto ovo neće dozvoliti MOVIE STAR da iznajmi neki JEWELRY PIECE u različitim datumima
- Ako je Scarlet Johansson iznajmila Princess ogrlicu 14.juna, ne bi mogla da je iznajmi nekog drugog datuma, pošto UID ne bi dozvolio istu kombinaciju MOVIE STAR id i JEWELRY PIECE kod da se unese više od jednom
- Opcija 2 – zabranjene relacije i Rental Date : dodavanje rental date u UID bi omogućilo MOVIE STAR da iznajmi neki JEWELRY PIECE različitim datumima, ali bi takođe dozvolilo različite MOVIE STARS da iznajme neki JEWELRY PIECE na isti datum
- Ako Scarlet Johansson iznajmi Princess ogrlicu 12 maja ovaj UID će takođe omogućiti Cameron Diaz da iznajmi Princess ogrlicu 12 maja
- Kombinacija MOVIE STAR id, JEWELRY PIECE koda i datuma iznajmljivanja su jedinstveni za svaku instancu, tako da će model to dopustiti. Ipak, znamo da u realnosti, ovo se ne može desiti
- Opcija 3 – zabranjena relacija između MOVIE STAR i RENTAL HISTORY sa Rental Date
- Ovaj model neće dopustiti istoj MOVIE STAR da iznajmi više od jednog dela JEWELRY PIECE u jednom danu
- Ako je Angelina Jolie htela da iznajmi 20 jula Princess ogrlicu i Maharajah narukvicu, neće moći pošto MOVIE STAR id plus date rented mora biti jedinstveno
- Opcija 4 – zabranjena relacija između JEWELRY PIECE i RENTAL HISTORY sa rental date
- Ovaj model kaže da JEWELRY PIECE može da se iznajmi jednom istog dana
- Kameron Dijaz želi da iznajmi Princezu ogrlicu 5 maja. Andželina Džoli može da je iznajmi 12 maja. Kombinacija JEWELRY PIECE koda i date rented je jedinstvena



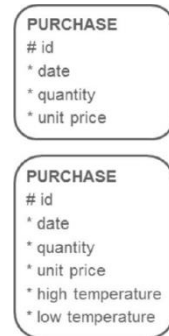
8-2 Modelovanje promena (modeling change) : Vreme

Smisao

- Vreme je bitan faktor u mnogim biznis modelima
- Istorijski podaci se često koriste od strane biznisa da bi se našli trendovi koji mogu uputiti na efikasnije načine sprovođenja biznisa
- Modelovanje vremena u biznisu omogućava da se takvi podaci čuvaju
- Izveštaji daju informacije koje mogu da se izvedu iz podataka
- Dobro dizajniran izveštaj kome dati vredne informacije koje biznisi mogu iskoristiti za unapređenje svojih operacija

Entitet DAY protiv atributa Date

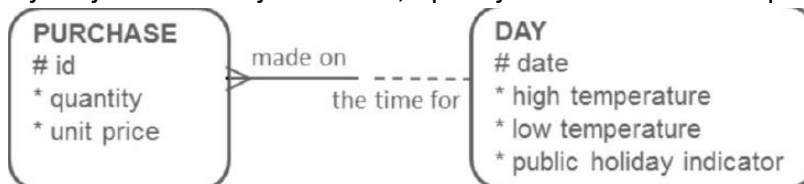
- Posmatrati entitet PURCHASE
- Uključite atribut date ako želite da znate kada je stvar kupljena
- Ipak, ako želite da identifikujete trendove, kao npr kupovina kaputa, bade mantila, patika, možda bi ste želeli znati kakav je bila temperatura u tim danima
- ako dodate temperaturu atribut u PURCHASE entitet javlja se problem



- Setite se 3NF: non-UID atribut ne može imati svoje sopstvene attribute
- Pošto high i low temperatura su atributi date, treba ih odvojiti kroz entitet DAY
- Ovo će nam omogućiti da kreiramo izveštaje sa detaljima koji proizvodi su prodati u toplim danima a koji u hladnim danima, omogućujući da se trendovi identifikuju

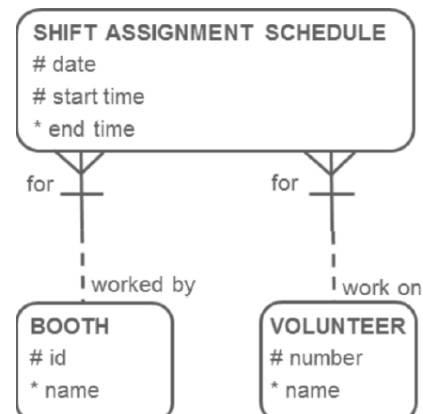


- Praveći razdvojeni DAY entitet omogućava nam da vodimo evidenciju o više informacija koje nam trebaju za biznis, npr koji dani su bili državni praznici

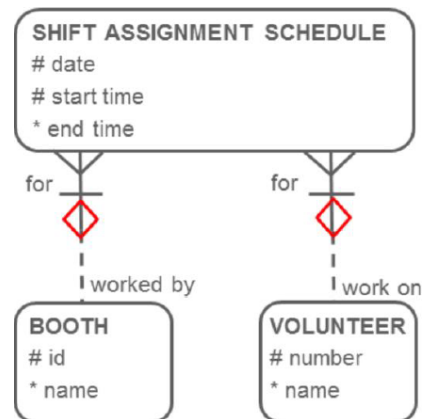


Ograničenja vezana vremenom

- Treba biti pažljiv sa ograničenjima koja rezultuju iz potrebe za vođenjem evidencije o datumima i vremenu
- Primer: na školskom vašaru su različiti štandovi; menadžer spaja dobrovoljce da rade različite smene na različitim štandovima; štand ima samo jednog dobrovoljca u jednom momentu; neki dobrovoljci mogu raditi nekoliko sati, drugi manje sati u zavisnosti od njihovog slobodnog vremena; raspored mora biti definisan unapred tako da menadžer zna koja vremena nisu popunjena sa dobrovoljcima
- Time-relateds constraints: ograničenja ili vremenske restrikcije koje rezultuju od vremenske dimenzije
- Ovo je selekcija ograničenja vezana vremenom koja se moraju uzeti u obzir za ovaj model
- Očigledan: smensko end time mora biti kasnije od smenskog start time
- Smene ne smeju da se preklapaju



- Start time za smenu za dobrovoljca ne bi smela biti između start time i end time drugog dobrovoljca na istom štandu
 - Isto važi za end time
 - Iako su ograničenja vezana vremenom očigledna u realnom životu, moraju biti sprovedena logičkim programiranjem u db. Zato se moraju dokumentovati
- Kondicional ne-transferabilan**
- Start time za smenu može biti updejtovan na later time, sem ako je smena već počela
 - Conditional nontransferability: Odnosi se na relacije koje mogu ili ne mogu biti transferabilne, u zavisnosti od vremena
 - Verovatno ne biste dopustili smeni da se dodeli novom dobrovoljcu ili drugom štandu sem ako smena nije već počela
 - Ovo je primer kondicionala ne-transferabilnog
 - Ne-transferabilnost: SHIFT ASSIGNMENT ne može biti promenjen na drugi BOOTH (ili na drugog VOLUNTEER)
 - Netransferabilne relacije se prikazuju sa dijamantom u ERD
 - Kondicionalna ne-transferabilnost: SHIFT ASSIGNMENT može ponekad da se promeni, u ovom slučaju, ako smena još nije počela
 - Ove relacije ne mogu biti predstavljene u dijagramu ali moraju biti dokumentovane



8-3 Modelovanje promena (modeling change) : Cena

Smisao

- Istorijske cene su važne kada se posmatraju trendovi, gleda rast ili smanjenje vrednosti stvari ili dobija refundacija za stvar kupljenu u prošlosti sa prethodnom cenom
- Mnogi biznisi vode evidenciju o promenama – ko ih je promenio, kada itd
- Primer: ako je promenjena ocena studenta, korisno je zapisati kada je promenjena, kakva je bila stara ocena, koja je nova ocena i ko je promenio ocenu
- Appreciation: porast vrednosti ili cene tokom vremena
- Depreciation: smanjenje ili gubitak u vrednosti zbog starenja, iznošenosti ili uslova na tržištu

Vажnost promena cena

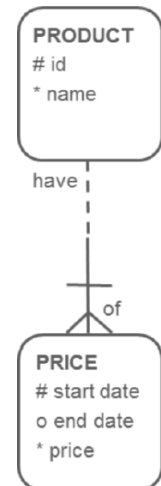
- Promene u cenama često su važne posebno kod biznis uslova
- Primeri: berza cene se menjaju u sekundi i koji faktori na to utiču; industrija goriva i cene su bitne pri kupovini kola ili načina kućnog grejanja; biznis građevine, cene su bitne za petogodišnje modelovanje izgradnje mosta

Kakva je cena danas

- Cene su proizvod promena tokom vremena (rastu, padaju i fluktuiraju)
- Hrana, odeća i troškovi školovanja su skuplji sada nego pre dvadeset godina
- Tehnologija je jeftinija tokom vremena
- Zlato, srebro, valute imaju cene koje variraju

Model Istorijskih cena

- Često je dobro imati informacije o cenama u prošlosti
- Ovde pokazan model vodi evidenciju o istorijskim cenama proizvoda
- Da li je zaista potreban end date ? Ako različiti periodi cena proizvoda su kontinualne, onda vam ne treba end date
- End date se može izvući iz start date sledeće cene pridružene stvari. Ipak ako proizvod nije uvek dostupan, poput sezonskog voća i tržišta povrća onda je end date potreban
- U ovom slučaju, moguće je imati proizvod koji još uvek nema cenu, i zato je relacija opcionalna na PRODUCT kraju

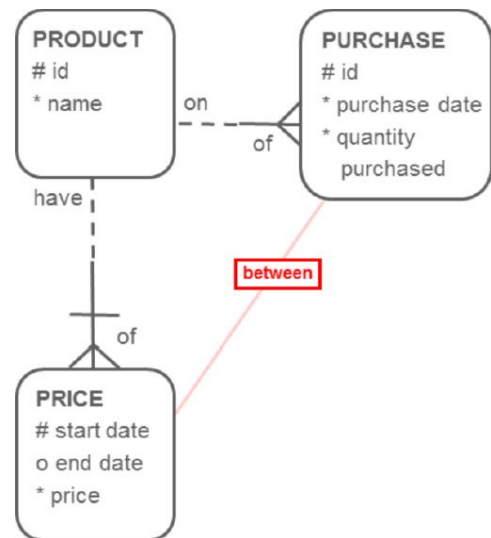


Promena istorijskih cena

- Razmotriti šta se dešava kada se vrati stvar u prodavnicu; kupljena je po određenoj ceni, ili se i dalje nalazi u prodaji

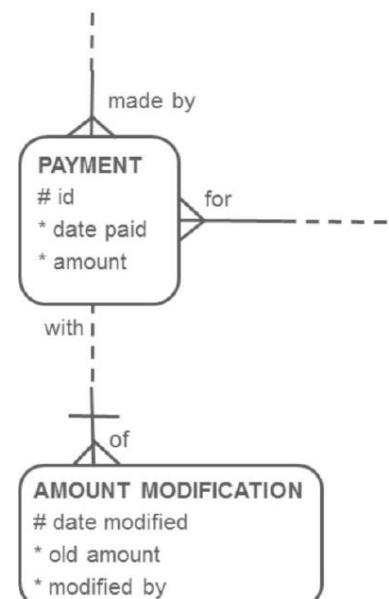
Praćenje promena cena

- Biznis često treba da čuva zapise o promenama cena
- U ovom modelu, pretpostavljamo da svaki PURCHASE se odnosi samo na jedan proizvod
- Cena koja je plaćena može da se nađe upoređivanjem sa datumom kupovine između start date i end date od PRICE
- Ne može se modelovati derived between relacija; ona mora biti dokumentovana i implementirana sa dodatnim kodom u sistemu
- PURCHASE entitet može takođe sadržati druge attribute ili relacije na CUSTOMER i EMPLOYEE koji izvršavaju transakciju; zbog jednostavnosti ovde nisu uključeni



Journaling

- Kada god sistem dozvoli korisniku da modifikuje ili skloni određenu informaciju, postavlja se pitanje Da li stare vrednosti treba da budu zapisane ?
- Ovo se naziva logging ili journaling
- Ovo je često problem kada je informacija finansijska ili osetljive prirode (promena ocena studenta)
- Journaling: Održavanje zapisa transakcija
- PAYMENT entitet može da ima dodatne attribute i relacije za PAYOR i PAYEE ali su izostavljeni iz dijagrama zbog jednostavnosti
- Osim uticaja na konceptualan data model, sistem traži poseban journaling funkcionalnost: bilo koji biznis funkcija koja dopušta update od PAYMENT sume treba

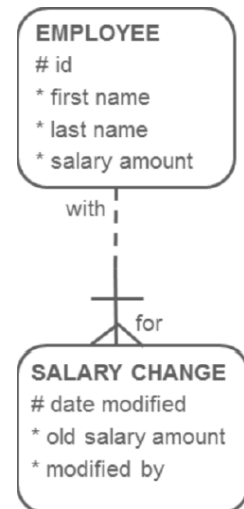


da rezultuje traženim updejtom čak i kreiranjem nove instance od AMOUNT MODIFICATION sa odgovarajućim vrednostima

- Ova funkcionalnost se mora implementirati programiranjem

Journal Content

- Journal obično se sastoji od i modifikovane vrednosti i informacije o tome ko je uradio modifikaciju i kada je ona izvedena
- Ova dodatna informacija može biti i proširena po želji
- Ponovo, ovaj sistem traži posebnu journaling funkcionalnost: bilo koja biznis funkcija koja omogućava updejt EMPLOYEE plate treba rezultovati traženim updejtom, plus kreiranjem entiteta instance SALARY CHANGE sa određenim vrednostima
- Alternativni način za model salary changes (i za druge primere vođenja evidencije o promenama tokom vremena) bio bi uključivanje i stare i nove salary količine kao atributa od SALARY CHANGE, i skidanja salary amount atributa iz EMPLOYEE. Relacija 1:M bi onda bila obavezna na oba kraja. Kako bi smo pronašli trenutnu platu zaposlenog ? To je nova plata u SALARY CHANGE instanci sa most recent date modified



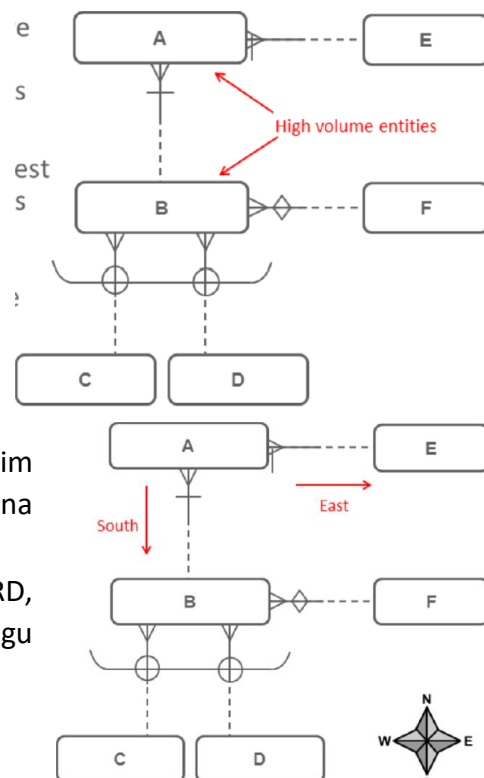
8-4 Crtanje konvencija za čitljivost

Smisao

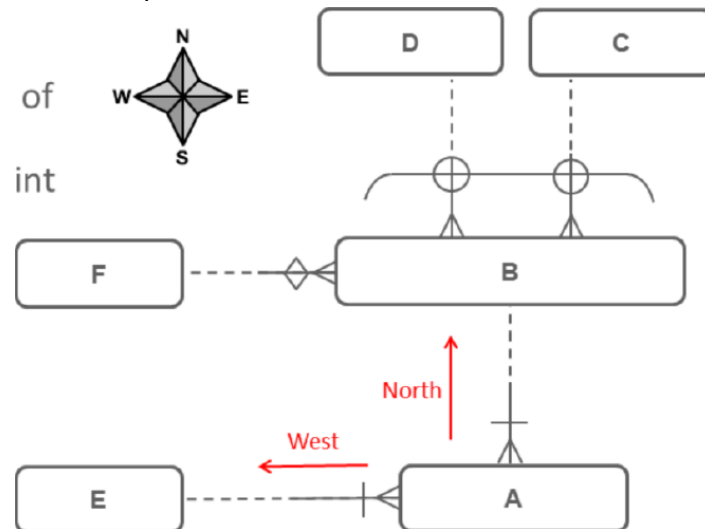
- Šta bi bilo kada bi proizvođači cipela izmislili neke svoje veličine ?
- Šta bi bilo kada bi svaki arhitekta koristio drugačiji sistem za crtanje planova za zgradu?
- Praćenjem istih konvencija čini pojednostavljenje posla pri radu u timu

Velike ERD konvencije pri crtanju

- Što je ERD veći i kompleksniji, postaje više izazovnije da prikaže svoje delove u čistom i čitljivom formatu
- Postoje dve crtajuće konvencije: jedna postavlja velike entitete prema gornjem levom uglu strane a druga u donji desni ugao strane
- Treba izabrati jednu i nje se držati
- Veliki (high-volume) entitet je entitet koji ima veliki broj instanci
- Veliki entiteti su često centralni ili važniji u ERD
- Imaju najveći broj relacija prema drugim entitetima i najveći broj biznis funkcija će uticati na podatke smeštene u njima
- Kada su veliki entiteti u gornjem levom delu ERD, crows feet ima tendenciju pokazivanja prema jugu i istoku



- Kada veliki entiteti su u donjem desnom delu ERD, crows feet imaju tendenciju da pokazuju na sever i zapad

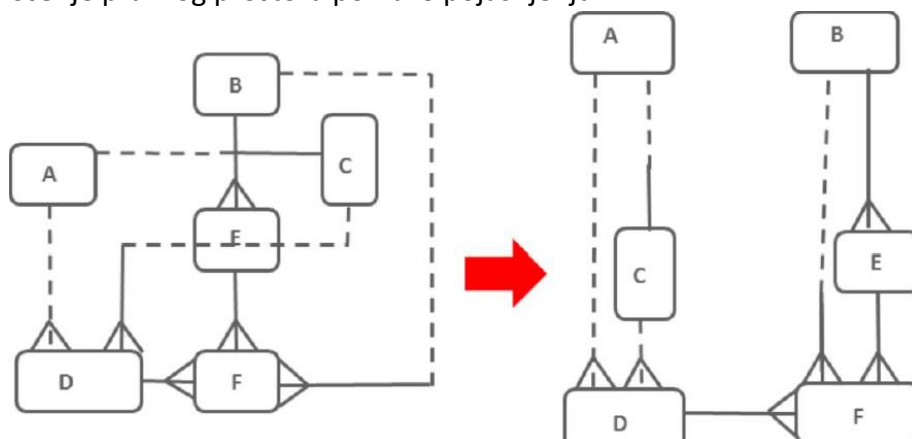


Jasnoća je cilj

- Koristi konvencije sa razumevanjem
- Glavni cilj kreiranja dijagrama je davanje reprezentacije modela koji se može koristiti za poslove komunikacije
- Ovo znači da nikad ne treba dopustiti da konvencija uskoči pre čitljivosti i jasnoće
- Često ćete imati miks konvencija, u zavisnosti od količine prostora koji imate i vaših ličnih želja
- Jasnoća i čitljivost su glavni kriterijum
- Za jasnoću i čitljivost u ERD: izbegavati prelaženje preko linija relacija, izbegavati entitete koji se preklapaju, izbegavati linije relacija koje idu preko entiteta, koristiti dosta praznog prostora, podeliti velike ERD u manje pod-dijagrame ako je to zatraženo

Prostor je potreban

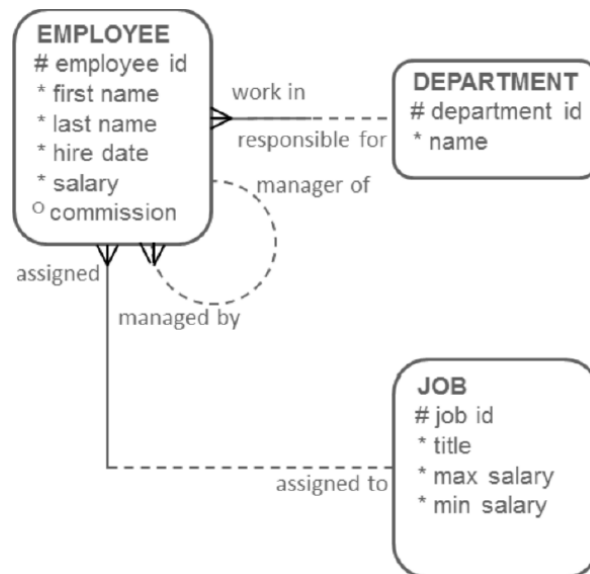
- Čitljivost zauzima prostor i deo je ukusa
- Korišćenje praznog prostora pomaže pojašnjenju ERD



- Ovi dijagrami obrađuju istu informaciju, ali onaj sa desne strane je čitljiviji; nemoj žrtvovati čitljivost zbog praznog prostora

Korišćenje pod-dijagrama

- Kada imate veliki dijagram može pomoći podela dijagrama na pod dijagrame sa funkcionalno povezanim entitetima



- Možete koristiti manje pod-dijagrame pri prezentovanju različitih grupa unutar mušterijine kompanije
- To je još uvek važno imati veliki dijagram koji pokazuje celu sliku (čak i ako mora biti odštampana preko plotera ili spojena sa manjim delovima papira)
- Mogu postojati relacije između entiteta u različitim pod-modelima i one se negde moraju pokazati
- Manji pod dijagrami mogu biti korisni tokom fizičkog modelovanja irazvoja aplikacija kao i tokom diskusija u konceptualnom modelovanju sa klijentom

