

3. Modelovanje baze podataka

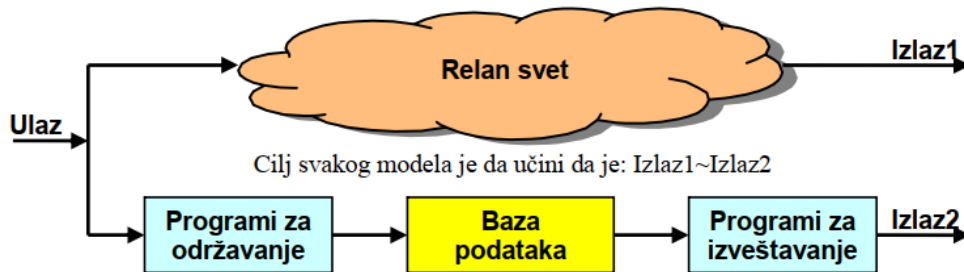
Informacioni sistemi pojedinih firmi omogućavaju upravljanje podacima koji su bitni za njeno poslovanje. Međutim, broj internih podataka i podataka iz okruženja je ogroman te je nemoguće sve podatke i sve uočene detalje opisati i sačuvati unutar informacionog sistema.

Postupkom selekcije identifikuju se i čuvaju samo **relevantni podaci**. Time se dolazi do pojma **modela podataka**. On je izraz i posledica zahteva za obradom podataka relevantnih za određeno područje primene.

Modeli su čovekovo sredstvo pojednostavljivanja problema i njegovo posmatranje samo sa stanovišta bitnih za ciljeve analize.

Objekt posmatranja (npr. automobil) ima uvek više osobina (**atributa**) od kojih u datom trenutku analize može biti dovoljan samo njihov manji broj (npr. samo *registarski broj, tip automobila, ime i prezime vlasnika*). To su najvažniji atributi potrebni u postupku pretraživanja i pronalaženja vlasnika vozila na osnovu registarskog broja vozila unutar jednog informacionog sistema.

Ostali atributi kao što su *boja, godina proizvodnje, broj sedišta* i sl. nisu bitni (mogu se zanemariti) za takav postupak. Čovek, obdaren sposobnostima apstraktnog načina mišljenja, stvara jedan apstraktni model realnog sveta. Takav model realnog sveta (objekta posmatranja) zasniva se na simbolima i zove se **konceptualni model podataka**.



Slika 1.11 Realan svet i njegov model

Objekti iz realnog sveta se u računarskoj primeni opisuju pomoću podataka. Podaci su zato apstrakcija realnosti, tj. sredstva za kodiranje osobina objekata iz realnog sveta.

3.1. Relacioni model -E/R model

Najčešće korišćeni model u praksi je **model objekti-veze (MOV) -Entity/Relations model (E/Rmodel)**

Relacioni model je svakako najpopularniji i najrasprostranjeniji model podataka danas i predstavlja osnovu za relacione baze podataka koje dominiraju na tržištu. Relacione baze podataka dominiraju na tržištu već skoro 40 godina!

Relacioni model je predložio E.F. Codd 1970 godine, dok je radio u IBMu.

System R je prvi sistem koji je koristio relacioni model, nakon toga IBM je implementirao svoj sistem poznat kao DB2. Nakon toga je Oracle realizovao svoj sistem zasnovan na ovom modelu, ... i sve ostalo je istorija.

Relacija, kao osnovni koncept relacionog modela je zapravo matematička relacija, i ima jednostavnu reprezentaciju u obliku tabele sa podacima.

To je grafička prezentacija povezanih entiteta i ograničenja koja čine dati dizajn odnosno projekat.

Kao i kod ostalih vizuelno orijentisanih dizajn metodologija, on pruža grafički sažetak strukture baze podataka koji je veoma koristan dizajneru - ne samo u procenjivanju tačnosti, odnosno pravilnosti dizajna, nego i za savetovanje sa kolegama i za objašnjavanje programerima koji će je koristiti.

E/R model se upotrebljava za konceptualni dizajn. Sastoji se od tri osnovne komponente:

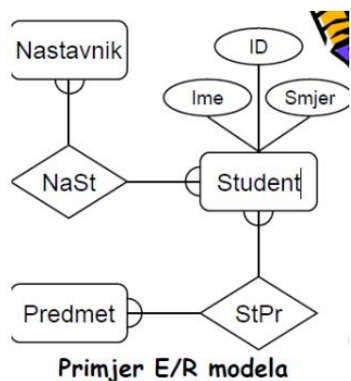
- Entiteti (objekti - stavke od interesa).
- Atributi (osobine entiteta).
- Relacije (veze između entiteta).

Svaki objekat, odnosno **entitet**, poseduje neka **svojstva(atribute)**. Na primer, entitet "vozilo" ima vlasnika, registarski broj, datum registracije, godinu proizvodnje, proizvođača, marku, boju, tip motora, i dodatnu opremu. Entitet „učenik” ima svoje atribute: JMBG, ime, prezime, datum rođenja, ocene... Svojstva ili atributi objekta će biti predstavljena kolonama u odgovarajućoj tabeli. Objekti međusobno mogu biti **povezani** različitim odnosima odnosno **relacijama**. Svaka takva relacija može da poseduje posebna svojstva. Relacije se mogu iskoristiti kod pretraživanja međusobno povezanih podataka, na primer, kod pretraživanja podataka o registrovanim vozilima i njihovim vlasnicima. Svaki objekat je definisan atributima, a atribut ima svoj tip.

Primer:

Na Univerzitetu, baza podataka bi mogla imati entitete za studente, predmete i nastavnike. Entitet Student može imati atribute kao ID, ime, i smer, i može biti povezan sa entitetima Predmet i Nastavnik.

E/R modeli se često predstavljaju preko **E/R dijagrama**.



Kreiranje E/R modela - Prilikom pravljenja E/R modela, iz opisa “problema” potrebno je prepoznati:

- o Entitete
- o Atribute
- o Veze

1. Entitet

Pod entitetom se podrazumeva sve što se može jednoznačno odrediti, identifikovati i razlikovati. Tako široko postavljena definicija pokazuje da entitet može biti svaki “realan” ili “apstraktan” objekt o kojem u određenom trenutku razmišljamo. Entitet je realan ako fizički, stvarno postoji (osoba, radnik, učenik, knjiga). Primer za apstraktan – radon iskustvo. Entitet se predstavlja kao pravougaonik. Unutar pravougaonik se upisuje ime entiteta.

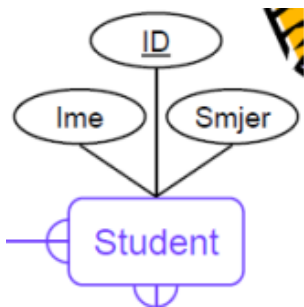
2. Atributi

Atribut opisuje entitet. Svaki entitet uočen u realnom sistemu ima svoje osobine koje ga čine složenim i njihove vrednosti omogućavaju razlikovanje entiteta. Svojstvo entiteta uključuje dva elementa - atribut i vrednost atributa

Npr. entitet *Student* ima atribute: *Ime*, *Prezime*, *Broj indeksa*, *Adresu*, *Telefon* i sl. i vrednosti *Marko*, *Marković*, *123/03*, *Danijelova*, *15*, *011/376-543* respektivno). Svaki put kada se promeni vrednost atributa, potrebno je promenu evidentirati, tj. ažurirati tu vrednost atributa za dati entitet.

Atributi mogu biti - Prosti i složeni atributi. Prosti atributi: *Visina (cm)*, *Ocena*, *Smer* itd. Složeni atributi: *Adresa (Ulica, Broj, Mesto, ...)*, *DatumRođenja (Dan, Mesec, Godina)*.

Predstavljaju se elipsama. Svaki je pravom linijom povezan sa entitetom kojem pripada. Naziv atributa je upisan unutar elipse.



Domen atributa je skup svih mogućih vrednosti koje atribut može poprimiti.

3. Relacije -Veze između entiteta

Baza podataka se ne odnosi samo na pojedinačne objekte nego i na odnose između objekata. U realnom sistemu objekti nisu međusobno izolovani, nego se nalaze u međusobnoj interakciji. Student se upisuje na fakultet, sluša predavanja iz pojedinih predmeta, prijavljuje polaganje ispita, polaže ispit itd. To su primeri logičkih i realnih veza između objekata, koje slede iz realnih odnosa u posmatranom sistemu studiranja na jednom fakultetu.

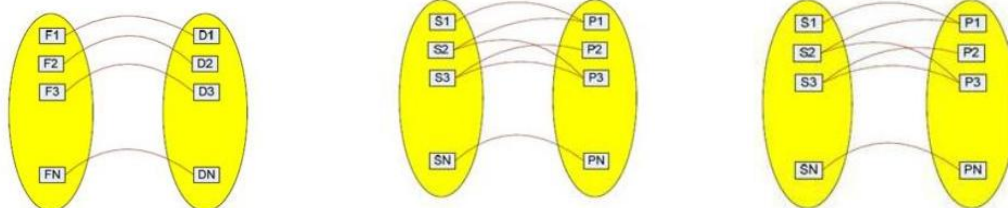
Najjednostavniji odnos između ta dva tipa objekata naziva se **preslikavanje 1:1** (veza 1:1). Kod takvog preslikavanja svaki se element skupa X može preslikati na najviše jedan element skupa Y. Istovremeno, i svaki element skupa Y može biti preslikan na najviše jedan element skupa X.

Karakterističan primer bi bio sa entitetima *Fakultet* i *Dekan*. Na jednom fakultetu može biti samo jedan dekan, a jedan dekan može biti dekan na samo jednom fakultetu. Takvi odnosi između entiteta su retki.

Druga vrsta odnosa naziva se **preslikavanje N:1** (ili 1:N). Više elementa skupa X može se preslikati na najviše jedan element skupa Y. Istovremeno jedan element skupa Y može se preslikati na više elemenata skupa X.

Pogodan primer za ovu vrstu odnosa između entiteta je odnos između entiteta *Student* i *Dekan*. Više studenata na jednom fakultetu ima samo jednog dekana, a jedan dekan je dekan za više studenata na svom fakultetu.

Najsloženije **preslikavanje je tipa M:N**. Svaki element prvog skupa može se preslikati na više elemenata drugog skupa, ali se i svaki element drugog skupa može preslikati na više elemenata prvog skupa. Karakterističan primer ovakvih veza postoji ako se uoče entiteti *Student* i *Profesor*. Jednom student predaje više profesora, a ujedno jedan profesor predaje za više studenata.



3.2. Primeri baza podataka (modela)

Cilj struktuiranja relacione baze podataka je da se napravi baza podataka pomoću koje je moguće da se modelira neki **praktičan sistem**. Znači, praktičan sistem deli u tabele i polja i određuje se odnos između dobijenih tabela. Takva baza podataka olakšava menjanje, dodavanje, uklanjanje i izdavanje podataka iz tabela.

Primer 1. Potrebno je kreirati bazu podataka za dnevni boravak dece.

O svakom detetu (ime, prezime, datum rođenja i beleške o detetu).

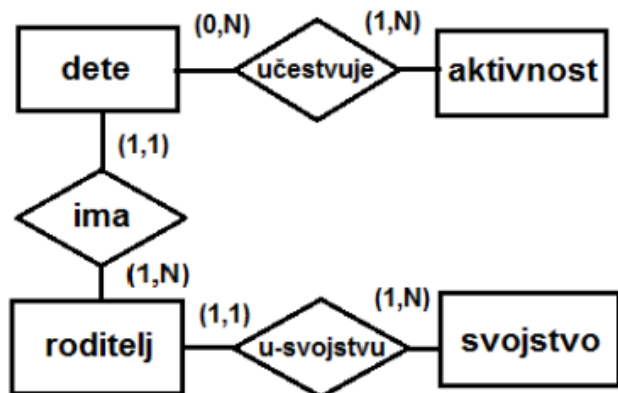
Za svako dete se zna samo jedan roditelj. Za roditelje pamtimo ime, prezime, adresu i brojeve fiksnog i mobilnog telefona. Jedan roditelj mora da ima jedno ali može da ima i više dece u dnevnom boravku. Jedan roditelj može da ima više dece koja su istovremeno u boravku.

Takođe pamtimo i podatke o svojstvu roditelja (otac, majka ili staratelj).

Deca u boravku učestvuju u aktivnostima. Za aktivnosti se zna šifra, naziv, kojim danima se odvija, u koliko sati počinje i u koliko sati se završava. Jedno dete može da učestvuje u više aktivnosti, ali ne mora da učestvuje ni u jednoj. U jednoj aktivnosti može učestvovati više dece, ali u svakoj aktivnosti mora da učestvuje bar jedno dete.

Entiteti: dete, roditelj, svojstvo, aktivnost.

Veze: dete **ima** roditelja, roditelj je **u svojstvu**, dete **učestvuje** u aktivnostima.



Primer 2. Potrebno je kreirati bazu podataka koja će da vodi evidenciju o klubovima u Fudbalskoj ligi Srbije.

Klubovi imaju šifru, naziv, ulicu i broj sedišta, telefon i e-mail, adresu zvanične web stranice, žiro račun i boje kluba.

Svaki klub dolazi iz jednog grada, a grad može imati više klubova.

O gradu se zna njegova šifra, ime, pozivni broj, poštanski broj i broj stanovnika.

Svaki klub ima jedan stadion na kome igra zvanične utakmice.

Stadion ima šifru, naziv, adresu na kojoj se nalazi (to podrazumeva i grad u kome se nalazi) i kapacitet gledalaca.

Svaki igrač može da igra samo za jedan fudbalski klub. Potrebno je pamtiti i datum od kada igrač igra za neki klub. Za igrača se pamti šifra, ime, prezime, datum rođenja, kontakt telefon i adresu boravka. Igrač može da igra samo na jednoj poziciji. Pozicija se karakteriše identifikacionim brojem, nazivom i kratkim opisom.

Entiteti: klub, grad, stadion, igrač, pozicija.

Veze: klub **dolazi iz** grada, klub **ima** stadion, igrač **igra za** klub, igrač **igra na** poziciji.

