Objektno orijentirane baze podataka

Šimić, Tonči

Undergraduate thesis / Završni rad

2015

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Pula / Sveučilište Jurja Dobrile u Puli**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:137:487508

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-20**

Repository / Repozitorij:

Digital Repository Juraj Dobrila University of Pula

Sveučilište Jurja Dobrile u Puli
Fakultet ekonomije i turizma
«Dr. Mijo Mirković»

Tonči Šimić

OBJEKTNO ORIJENTIRANE BAZE PODATAKA

Završni rad

Pula, 2015.

Sveučilište Jurja Dobrile u Puli
Fakultet ekonomije i turizma
«Dr. Mijo Mirković»

Tonči Šimić

OBJEKTNO ORIJENTIRANE BAZE PODATAKA

Završni rad

JMBAG: 0303037228 , redoviti student
Studijski smjer: Poslovna informatika

Predmet: Baze podataka
Mentor: prof.dr.sc. Vanja Bevanda

Pula, rujan 2015.

IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisani _____, kandidat za prvostupnika _____ovime izjavljujem da je ovaj Završni rad rezultat isključivo mojega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio Završnog rada nije napisan na nedozvoljen način, odnosno da je prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

Student:

U Puli, . . 2015. _____

UVOD
1. OPĆENITO O BAZAMA PODATAKA 1
1.1. OSNOVNI POJMOVI 1
1.2. BAZE PODATAKA 2
1.3. VRSTE BAZA PODATAKA 4
1.4. RAZVOJNI CIKLUS BAZE PODATAKA 6
1.5. SUSTAV ZA UPRAVLJANJE BAZOM PODATAKA 7
2. MODELIRANJE PODATAKA 10
2.1. KONCEPTUALNO MODELIRANJE PODATAKA 11
2.1.1. MODEL ENTITETI – VEZE 11
2.1.1.1. ENTITETI 11
2.1.1.2. VEZE 12
2.1.1.3. ATRIBUTI 13
2.2. LOGIČKO MODELIRANJE PODATAKA 15
2.3. FIZIČKO MODELIRANJE PODATAKA 16
3. LOGIČKI MODELI PODATAKA 17
3.1. HIJERARHIJSKI MODEL PODATAKA 17
3.2. MREŽNI MODEL PODATAKA 18
3.2.1. DBTG CODASYL mrežni model 19
3.3. RELACIJSKI MODEL PODATAKA 20
3.4. OBJEKTNI MODEL PODATAKA 22
4. OBJEKTNO ORIJENTIRANE BAZE PODATAKA 23
4.1. GRAĐA OBJEKTNOG MODELA 24
4.2. PREDNOSTI I NEDOSTACI OBJEKTNOG MODELA 26
4.3. ODMG STANDARD 26
5. UNIFIED MODELING LANGUAGE (UML) 27
6. RELACIJSKO – OBJEKTNE DB NA PRIMJERU ORACLE DB 29
7. ZAKLJUČAK 32
8. LITERATURA 33
a) KNJIGE 33
b) INTERNET 33
9. POPIS SLIKA 34

UVOD

U današnje vrijeme informacije su jedan od najvažnijih resursa u poslovnom svijetu. Uz informacije, važan je i informacijski sustav koji prikuplja, obrađuje i distribuira informacije. Konačni cilj informacijskog sustava je osigurati sve informacije koje su potrebne za obavljanje poslova i donošenje odluka. Tu se dolazi do važnosti baza podataka. Baze podataka olakšavaju pretraživanje, organiziranje, odnosno uporabu podataka neophodnih za uspješno poslovanje. Samim time, baze podatak omogućuju kvalitetnije, efikasnije, brže i bolje poslovanje. Sve su to preduvjeti za uspjeh u današnjem poslovnom svijetu. Kada se spomene pojam baze podatak uglavnom se to odnosi na relacijske baze podataka. One su danas najzastupljenije upravo zbog svoje jednostavnosti. U relacijskom modelu podaci i veze među podacima prikazani su pomoću tablica koje se sastoje o redaka i stupaca. Ipak, ovaj rad je orijentiran na baze podataka zasnovane na objektnom modelu. Objektne baze su dugo bile u sjeni relacijskih baza, čak su 1999. godine izbačene iz uporabe, ali naglim razvojem informacijske tehnologije objektne baze ponovno izbijaju u prvi plan. Cilj je utvrditi zašto relacijske baze podataka danas prevladavaju, te u čemu se uopće razlikuju od objektnih baza. Nadalje, cilj je objasniti osnovne pojmove vezane uz baze podataka, te utvrditi posjeduju li objekte baze podataka potencijal da u budućnosti nadvladaju relacijske baze.

Završni rad sastoji se od šest poglavlja u kojima se razrađuju objektno orijentirane baze podataka. Prvo poglavlje kreće od općenitog poimanja baze podataka, te se poslije proteže na osnovne pojmove koji su ključni za razumijevanje materije. Potom se definira baza podataka, te se obrađuju pojedine specifičnosti vezane za istu. Nakon toga, detaljno se objašnjavaju vrste baza podataka, da bi se na kraju poglavlja objasnio razvojni ciklus baze podatak, te teoretski obradio sustav za upravljanje bazom podataka. Poglavlje dva odnosi se na modeliranje podatak koje podrazumijeva konceptualno, logičko i fizičko modeliranje podataka. U sklopu drugog poglavlja prikazan je primjer relacijskog i ER dijagrama. Poglavlje tri donosi definiranje sljedećih modela podataka: hijerarhijski, mrežni, relacijski i objektni model. Svaki je model vizualno prikazan. Četvrto poglavlje donosi razradu objektno orijentiranih baza podataka. U ovom poglavlju najprije se govori o objektno orijentiranim baza podataka, potom se objašnjava građa objektnog modela, te se na kraju izdvajaju prednosti i nedostaci tog modela. Peto poglavlje odnosi se na Unfied modeling language. U ovom poglavlju teoretski se razlaže UML, te je priložen primjer klasnog dijagrama. Na samom kraju poglavlja prikazano je kreiranje objektnih tipova i tablica.

1. OPĆENITO O BAZAMA PODATAKA

1.1. OSNOVNI POJMOVI

U današnje vrijeme informacije se sve više smatraju jednim od najvažnijih resursa poduzeća. Općenito, informacija je novo znanje, koje primatelju donosi nove činjenice. Informacije imaju karakter novosti, otklanjaju neizvjesnost i služe kao podloga za odlučivanje[1].

Uz informaciju, vrlo važan pojam je informacijski sustav koji se brine o prikupljanju, obradi te distribuciji informacija. Po općoj teoriji sustava, sustav je skup komponenata organiziranih na određeni način, koje radeći zajedno postižu određene ciljeve.

Internacionalna federacija za obradu podataka (IFIP) definira informacijski sustav na sljedeći način: Informacijski sustav jest sustav koji prikuplja, pohranjuje, čuva, obrađuje i isporučuje informacije važne za organizaciju i društvo, tako da budu dostupne i upotrebljive za svakog tko ih želi koristiti, uključujući poslovodstvo, klijente, osoblje i ostale. Informacijski sustav aktivni je društveni sustav koji može, ali ne mora, koristiti suvremenu informacijsku teoriju[2].

Informaciju predstavlja rečenica „Tonči je rođen 19.03.1993“. U ovoj informaciji Tonči je objekt promatranja, datum rođenja je promatrano obilježje koje ima vrijednost 19.03.1993. U ovom primjeru vidi se da je informacija predstavljena n-torkom < Tonči, datum rođenja, 19.03.1993. što ubiti predstavlja < naziv objekta, obilježje objekta, vrijednost obilježja, vrijeme >. Da bi informacija ostala sačuvana potrebno ju je zapisati. Taj zapis se zove podatak. Podatak je skup prepoznatljivih znakova na određenom mediju.

Dobar informacijski sustav je mozak poslovnog svijeta. Ukoliko određeni poslovni sustav želi opstati, potreban mu je informacijski sustav. Poslovni sustav prima informacije iz raznih izvora, vanjskih ili unutarnjih, koje potom informacijski sustav obrađuje u nove i korisne informacije. Konačni cilj informacijskog sustava je osigurati poslovnom sustavu sve informacije koje su mu potrebne za obavljanje poslova i donošenje odluka.

[1] Varga, Mladen, Baze podataka, DRIP, Zagreb, 1994., str. 2.
[2] loc. cit.

1.2. BAZE PODATAKA

Baza podataka je skup povezanih podataka informacijskog sustava odnosno organizacije. Sadrži određene podatke različitih poslovnih sustava poput ugovora, usluga, poslovnih partnera i slično. Bolje rečeno, baza podataka je organiziran skup podataka koji služe jednoj ili više aplikacija, neovisno o programu koji ih obrađuje, te je pristup njima kontroliran.

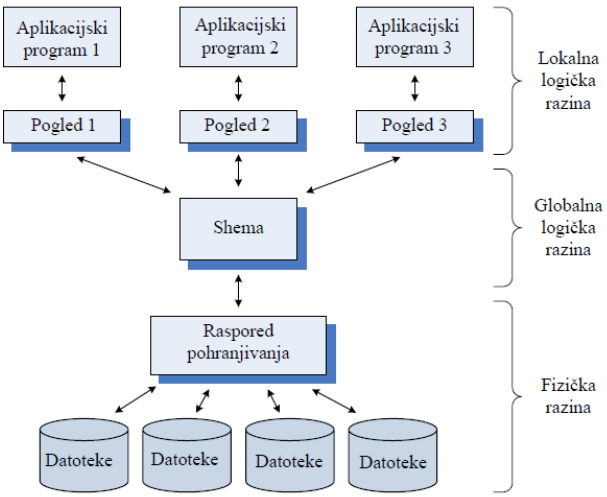
Osnovna ideja baze podataka je da pojedina aplikacija ne stvara svoje vlastite datoteke na disku, već sve aplikacije koriste zajedničku kolekciju podataka. Aplikacije ne pristupaju izravno podacima na disku, već koriste određene specijalizirane softvere koji su zaduženi za zajedničku kolekciju podataka. Zajednička kolekcija podataka je baza podataka, dok je specijalizirani softver sustav za upravljanje bazom podataka - DBMS. Temelj dobro oblikovane baze podataka je cjelovit konzistentan i neredundantan konceptualni, logički i fizički model podataka. Temeljem toga baza podataka predstavlja cjelovitu, konzistentnu i neredundantnu kolekciju podataka. U današnje vrijeme još nije određeno na kojem će se mediju nalaziti baza podataka. Tako da postoje baze podataka koje nisu spremljene na računalu već na papiru. Takve baze ne predstavljaju dobro oblikovane baze podataka jer su često redundantne i nekonzistentne. Baze podataka koriste se u mnogim aplikacijama. One su poželjna metoda spremanja velikih korisničkih aplikacija gdje je potrebna koordinacija između mnogo korisnika.

Temeljni ciljevi baza podataka su sljedeći [3]: fizička nezavisnost podataka, logička nezavisnost podataka, fleksibilnost pristupa podacima, istovremeni pristup do podataka, čuvanje integriteta, mogućnost oporavak nakon kvara, zaštita od neovlaštene uporabe, zadovoljavajuća brzina pristupa i mogućnost podešavanja i kontrole. U fizičkoj nezavisnost podataka razdvaja se logička definicija baze od stvarne fizičke građe , odnosno ako se fizička građa promijeni to neće zahtijevati promjene u postojećim aplikacijama. Kod logičke nezavisnosti podataka, ako se globalna logička definicija promijeni (uvede se novi zapis ili slično) to neće zahtijevati promjene u postojećim aplikacijama. U praksi teže je postići fizičku nezavisnost podataka od fizičke nezavisnosti. Pod fleksibilnosti pristupa podacima danas se podrazumijeva da korisnik može slobodno pretraživati i prebirati po podacima te uspostavljati vezu među podacima. Relacijske baze podataka jedine zaista zadovoljavaju ovaj zahtjev. Sljedeći cilj je istovremeni pristup do podataka. U sklopi njega baza podataka trebala bi omogućiti da veći broj korisnika u isto vrijeme koristi istu bazu podataka, te da pritom ne

[3] Manger, Robert, Baze podataka, Element, Zagreb, 2012., str. 3.

ometaju jedan drugog i da svaki od njih ima dojam da sam radi s bazom. Čuvanjem integriteta nastoji se sačuvati konkretnost i konzistencija podataka, i to u situacijama kad postoje greške u aplikacijama. Mogućnost oporavka nakon kvara znači da bi trebala postojati zaštita baze podataka u slučaju kvara hardvera ili grešaka u radu softvera. Zaštita od neovlaštene uporabe znači da bi trebala postojati mogućnost da korisnici reguliraju ovlaštenja, odnosno što se smije, a što ne smije raditi s podacima. Zadovoljavajuća brzina pristupa znači da bi se operacije trebale odvijati dovoljno brzo u skladu s određenom aplikacijom. Brzina ovisi o odabiru određenih struktura podataka, te o algoritmima za pretraživanje. Mogućnost podešavanja i kontrole znači da baze podataka zahtijevaju stalnu brigu: reguliranje ovlaštenja korisnika, praćenje performansi, rutinsko pohranjivanje rezervnih kopija podataka i slično. Odgovorna osoba naziva se administrator baze podataka.

Arhitektura baze podataka sastoji se od tri sloja i sučelja među slojevima. Riječ je o tri razine apstrakcije: fizička razina, globalna logička razina i lokalna logička razina.[4]



SLIKA 1. Arhitektura baze podataka[5]

Na slici je prikazana arhitektura baze podataka. Sastoji se od tri razine. Fizička razina zadužena je za fizički prikaz raspored podataka na jedinicama vanjske memorije. Taj aspekt vide samo sistemski programeri koji su razvili DBMS. Fizička razina može se podijeliti na više podrazina, od sasvim konkretnih staza i cilindara na disku, do već donekle apstraktnih pojmova datoteke i zapisa kakvi se susreću u klasičnim programskim jezicima. Raspored pohranjivanja opisuje kako se elementi logičke definicije baze preslikavaju na fizičke uređaje.

[4] ibidem, str.5
[5] loc.cit.

Globalna logička razina odnosi se na logičku strukturu cijele baze. Taj aspekt vidi projektant baze odnosno njezin administrator. Shema je opis globalne logičke definicije. Shema je tekst ili dijagram koji definira logičku strukturu baze podataka u skladu sa zadanim modelom. To znači da se imenuju i definiraju svi tipovi podataka i veze među njima u skladu s pravilima zadanog modela. Također, shema može uvesti određena ograničenja kako bi sačuvala integritet podataka. Lokalna logička razina odnosi se na logičku predodžbu o dijelu baze koji rabi pojedina aplikacija. To je aspekt koji vide korisnici ili aplikacijski programeri. Opis jedne lokalne logičke definicije zove se pogled ili podshema. To je tekst ili dijagram kojim se imenuju i definiraju svi lokalni tipovi podataka i veze među njima u skladu s pravilima zadanog modela.

1.3. VRSTE BAZA PODATAKA

Većina baza podataka sadrži znakovne podatke, ali danas su sve više prisutne baze podataka koje sadrže neznakovne podatke (slikovne ili zvučne) ili čak kombinaciju znakovnih i neznakovnih podataka. Ovisno o vrsti i namjeni te načinu korištenja podataka postoje baze strukturiranih podataka, baze nestrukturiranih podataka te baze znanja.

Baze strukturiranih podataka najčešće se koriste u poslovnim primjenama. Na slici 2. prikazan je primjer takvih podataka. U tablici su vidljivi podaci kupaca, te se za sve kupce bilježe atributi u istim formatima. Podaci jednog objekta, odnosno kupca pohranjeni su u jednom retku tablice. Ovisno o tome kako su izgrađene logičke i fizičke strukture podataka, postoje starije baze podataka koje su građene prema hijerarhijskom ili mrežnom modelu, te suvremene baze podataka koje su građene po relacijskom, objektnom i dimenzijskom modelu. Danas se najčešće koristi relacijska građa jer se podaci u njima prikazuju tablicama koje su međusobno povezane ključevima.

Šifra kupca	Naziv kupca	Adresa kupca
432	Furnir Otok d.o.o.	Otok, Josipa Kozarca 56
532	Boso d.o.o.	Vinkovci, Školska 39
190	Ivan Mikulić	Zagreb, Vladimira Nazora 35
212	Ante Ivić	Pula, Nobileova 33

SLIKA 2. Prikaz baze strukturiranih podataka

(izvor: izradio student)

Baze nestrukturiranih podataka sadrže nestrukturirane ili polustrukturirane znakove te multimedijske podatke. Baze nestrukturiranih podataka pohranjuju objekte koji mogu biti novinski, znanstveni, stručni članci, dokumenti, slike, zvukovi, videozapisi i slično. Logično je da se takvi podaci ne mogu odgovarajuće pohraniti pomoću tablica već ih je potrebno

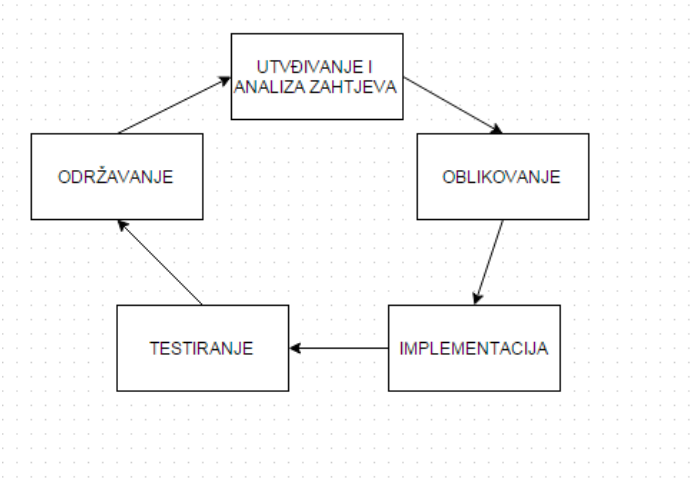
pohraniti kao skup dokumenata. Za to su zadužene baze dokumenata. Jedna takva baza dokumenata je hrcak.srce.hr koja sadrži razne znanstvene časopise Republike Hrvatske.
Pretraživanje u bazama dokumenata odvija se pretragom ključnih riječi od kojih se očekuje da se nalaze u dokumentu. Postoje dva načina pretraživanja: logičko pretraživanje i pretraživanje rangiranjem dokumenata. Logičko pretraživanje provodi se tako da se uvjet pretraživanja opisuje riječima koje se potom kombiniraju logičkim operatorima AND, OR i NOT (I, ILI, NE). Za svaki se dokument ispituju zadani uvjeti te ako je uvjet ispunjen dokument se smatra pronađenim. Pretraživanje rangiranjem dokumenata provodi se tako da se izračuna udaljenost između upita i svakog pojedinog dokumenta u bazi. Ispisuju se samo oni dokumenti čija je udaljenost manja od zadanog praga. Kod ispisa, najprije se prikazuju dokumenti koji su bliže traženom upitu, te su zbog toga važniji dokumenti ispred manje važnih. Takav način koristi se i pri pretraživanju dokumenata na internetu, ali pretraživanja dokumenata putem specijaliziranih baza podataka je preciznije od pretraživanja putem interneta. Razlog leži u tomu što se dokumenti u bazama dokumenata redovito stručno klasificiraju. Također, baze dokumenata se uglavnom služe sređenim popisom ključnih riječi što kod pretraživanja putem interneta nije slučaj. Ipak, specijalizirane baze podataka su dostupne putem interneta, ali se ne mogu pretraživati putem internetskih pretraživača (Google, Yahoo i slično) već njihovi dokumenti pripadaju tzv. dubinskom webu.

Baze znanja sadrže znanje prikazano u različitim oblicima, npr. u obliku pravila, semantičkih mreža, okvira ili scenarija. Tako prikazano znanje može se upotrijebiti korištenjem različitih mehanizama zaključavanja. Tako se u ekspertnim sustavima i drugim sustavima temeljenim na znanju mogu rješavati različiti problemi kao što su dijagnoza uzroka grešaka u složenim sustavima, financijska predviđanja, konfiguriranje računalnih sustava i slično. [6]

[6] Varga, Mladen, Informacijska tehnologija u poslovanju, Element, Zagreb, 2004., str. 381.

1.4. RAZVOJNI CIKLUS BAZE PODATAKA

Prilikom uvođenja baze podataka u određenu organizaciju ili ustanovu potrebno je primijeniti odgovarajuće metode i alate, te koristiti timski rad stručnjaka različitih profila. Razvojni ciklus baze podataka podijeljen je u pet aktivnosti:



SLIKA 3. Razvojni ciklus baze podataka

(Izvor: izradio student)

Slika 3. prikazuje razvojni ciklus baze podataka. Prvi korak priliko uvođenja baze podataka je utvrđivanje i analiza zahtjeva. U ovom koraku analiziraju se dokumenti, razgovara se s korisnicima, prati se radni proces i slično, te se na osnovu toga utvrđuju zahtjevi. Također, uočavaju se podaci koji će biti pohranjeni, te utvrđuju veze među njima. Provodi se analiza operacija i postupaka koji će se obavljati s podacima jer to ima iznimno jaki utjecaj na sadržaj i konačni oblik baze podataka. Rezultat utvrđivanja i analize zahtjeva je dokument koji se naziva specifikacija. On definira najvažnije operacije s podacima, a često i cijele aplikacije.

Cilj oblikovanje je da se pomoću, i u skladu sa specifikacijom, oblikuje baza podataka. Prethodni korak odredio je koje će vrste podataka baza sadržavati, te što bi se s njima trebalo moći raditi, dok oblikovanje definira na koji će se način podaci odgovarajuće strukturirati, grupirati i međusobno povezati. Rezultat oblikovanja je shema cijele baze. Oblikovanje se provodi u tri faze: Konceptualno, logičko i fizičko oblikovanje. Implementacija je fizička realizacije baze podataka na poslužiteljskom računalu. Najprije se pokreću SQL[7] naredbe koje predstavljaju fizičku shemu baze te se nakon toga kreiraju prazne SQL tablice na disku. Nakon toga se popunjavaju prazne tablice s početnim podacima. Iako postoje alati koji bi taj proces olakšali, postupak je obično težak i kompliciran zbog potrebe čišćenja, usklađivanja i

[7] Structured Query Language – upitni jezik

uređivanja podataka. Kako bi se omogućilo testiranje potrebno je kreirati aplikacija koja će obavljati transakcije s podacima. Testiranje se provodi tako da korisnici rade s bazom, te provjeravaju radi li baza onako kako se to od nje očekuje. Znači, pokreću se transakcije s podacima, prati se njihov performans i slično. Cilj testiranja je otkriti greške koje su nastale u prethodnim koracima te ih ispraviti. Greške u ranijim fazama su problematičnije od grešaka u kasnijim fazama jer se greške u ranijim fazama provlače i kroz ostale faze. Održavanje je kontinuirani proces u kojem su baza i njeni dokumenti podvrgnuti promjenama. Postoji nekoliko vrsta održavanja : Korekcijsko održavanje – popravak grešaka koje nisu bile uočene prilikom testiranja, Perfekcijsko održavanje – mijenjanje sheme baze kako bi se prilagodilo novim aplikacijama koje nisu postojale tijekom utvrđivanja i analize zahtjeva, Adaptacijsko održavanje – koristi se onda kada se bazu želi prilagoditi novom sustavu za upravljanje bazom podataka koji se nije koristio tijekom oblikovanja i implementacije. Održavanje je potrebno provoditi tijekom cijelog života baze.

1.5. SUSTAV ZA UPRAVLJANJE BAZOM PODATAKA

Baza podataka je skup međusobno povezanih podataka koji su spremljeni u vanjskoj memoriji računala. Brisanje, mijenja, upravljanje te čitanje podataka vrši se pomoću posebnog softvera odnosno sustava za upravljanje bazom podataka (eng. Data Base Management System – DBMS). Sustav za upravljanje bazom podataka ima niz funkcija : oblikuje fizički prikaz baze, obavlja u ime klijenta sve operacije s podacima, u stanju je podržati razne baze od kojih svaka može imati svoju logičku strukturu, brine se za sigurnost podataka i automatizira administrativne poslove i slično. DBMS omogućuje definiranje baze podataka (spremište podataka), upis podataka u bazu, ispis podataka iz baze i obradu podataka smještenih u bazi.

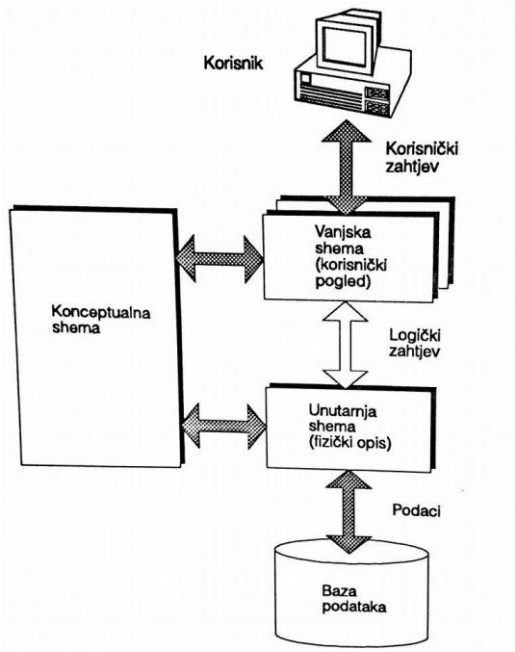
DBMS ima niz funkcija[8]:

- Funkcije za definiranje baze podataka : ova funkcija ostvaruje se standardnim jezikom za rad s bazom podataka, kao što je SQL kod relacijske baze podataka, ili zasebnim jezikom kod mrežnih i hijerarhijskih baza podataka. U oba slučaja jezikom se opisuje shema baze podataka koja sadrži logički opis baze podataka (naziv, tip, forma), kao i njezin fizički opis (fizičku organizaciju podataka) .

[8] Varga, M., op.cit., str. 11.

- Funkcije za manipulaciju podacima u bazi podataka: ostvaruje se zasebnim, neproceduralnim jezikom kao što je SQL ili dodavanjem DML naredbi u standardne programske jezike COBOL, PL/I, FORTRAN, C ili Pascal.
- Upravljačke funkcije – funkcije sigurnosti podataka, odnosno zaštite od neovlaštenog korištenja, funkcije očuvanja integriteta baze podataka, funkcije statističkog praćenja baze podataka.

Shema baze podataka



SLIKA 4. Shema baze podataka[9]

Na slici četiri prikazana je shema baze podataka. Shema baze podataka sadrži opis baze podataka. ANSI/SPARC[10] definira 3 sheme. Konceptualna shema je potpun konceptualni i logički model podataka čitave baze. Hoće li opis pripadati logičkoj ili konceptualnoj razini ovisi o korištenom sustavu za upravljanje bazom podataka. Vanjska shema ili korisnički pogled je opis samo jednog dijela logičkog modela koji je primjeren određenom korisniku. Sukladna je s konceptualnom shemom. Za svakog korisnika može se definirati vanjska shema. Naziva se još i podshema jer predstavlja samo dio podataka u bazi. Unutarnja shema je opis fizičkih podataka u bazi. Također, sukladna je s konceptualnom shemom.

[9] ibidem, str.12
[10] Komitet Američkog instituta za standardizaciju (American National Institute / Standards Planning and Requirements Committee)

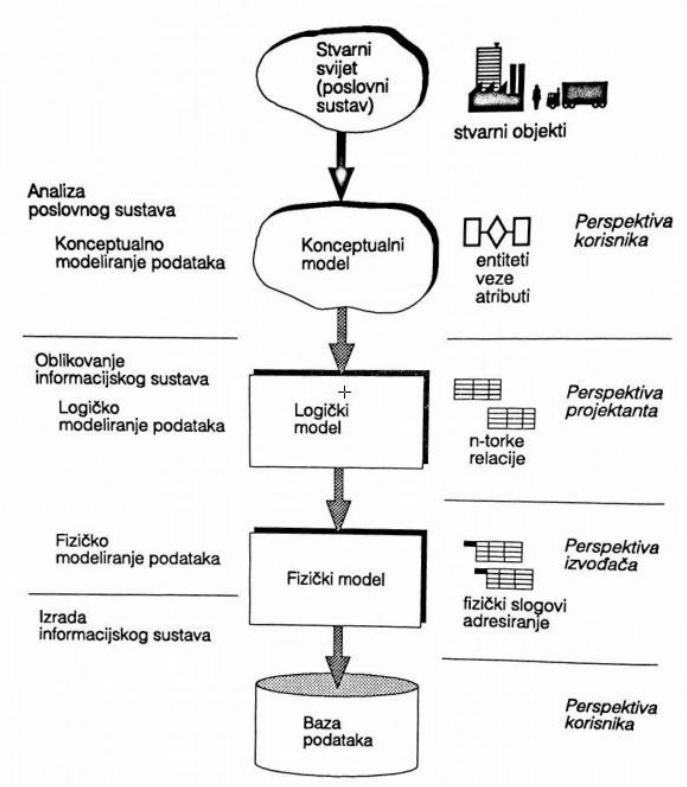
Komercijalni sustavi i sustavi otvorenog koda:

- MySQL – besplatan, open source sustav za upravljanje bazom podataka. Čest je izbor za projekte otvorenog koda te se distribuira kao sastavni dio Linux-a, ali postoje inačice i za ostale operacijske sustave (Windows, Mac Os i slično). Slobodan je za većinu uporaba. Vrlo je stabilan, ima dobro dokumentirane module i ekstenzije za podršku brojnih jezika: PHP, Perl, Java i slično. MySQL baze su relacijskog tipa, koji se pokazao kao jedan od boljih načina skladištenja i pretraživanja velike količine podataka. Jedan od razloga zašto je MySQL osvojio veliki dio tržišta leži u time što se može besplatno koristiti.
- Microsoft SQL Server – relacijska baza podataka koja osim klasičnih SQL upita omogućuje i složenije mijenja programskog toka.
- DB2 - Proizvod tvrtke IBM, namijenjen prvenstveno velikim mainframe računalima. Koristi relacijski model.
- Oracle - Proizvod istoimene tvrtke, pokriva gotovo sve računalne platforme, na primjer UNIX, Linux i MS Windows.
- Oracle RDBMS – kombinira relacijski model s elementima objektnog modela.
- Microsoft Office Access – kombinira alate za razvoj softvera i relacijsku bazu podataka s jednostavnim korisničkim sučeljem.
- Ostali: Oracle Lite, Versant, Postgre SQL, Interbase, Rdb...

DBMS spada u temeljni softver koji većina korisnika i organizacija ne razvija samostalno, već ga kupuju. Svi ovi proizvodi uz sam DBMS uključuju u sebi i dodatne alate za razvoj aplikacija, administriranje baze i slično.

2. MODELIRANJE PODATAKA

Modeliranje podataka počinje u fazi planiranja, a detaljno se razvija u fazi analize što znači da se modeliranje podataka odvija kroz sve faze razvoja informacijskog sustava. Najprije se pomoću konceptualnog modeliranja razvija konceptualni model podataka, potom se u fazi oblikovanja pomoću logičkog ili implementacijskog modeliranja transformira u logički ili implementacijski model, te se na kraju fizičkim modeliranjem pretvara u fizički model. Realizacijom fizičkog modela nastaje baza podataka.



SLIKA 5. Od stvarnog svijeta do baze podataka[11]

Kao što je prikazano na slici pet, modeliranje podataka odvija se kroz konceptualno modeliranje, logičko ili implementacijsko modeliranje te fizičko modeliranje.

Ciljevi modeliranja podataka su:

- dokumentiranje informacijskih zahtjeva,
- izgradnja baze podataka koja ima minimalnu redundanciju, maksimalnu integriranost i konzistentnost podataka, odgovarajuću stabilnost i fleksibilnost te dobar pristup i iskoristivost,
- povećanje vrijednosti podatkovnih resursa.

[11] ibidem, str. 40

2.1. KONCEPTUALNO MODELIRANJE PODATAKA

Konceptualno modeliranje podataka je prva faza tijekom oblikovanja baze podataka, a glavni cilj je stvoriti konceptualnu shemu koja je sastavljena od entiteta, veza i atributa. Konceptualna shema razumljiva je ljudima svih struka, i može služiti kao sredstvo komunikacije između projektanta i korisnika. Konceptualni model podataka jest cjelovit, konzistentan i neredundantan opis podataka informacijskog sustava. Njega izgrađuju analitičari odnosno projektanti u suradnji s korisnicima koji dobro razumiju potrebe za informacijama.

2.1.1. MODEL ENTITETI – VEZE

Ovaj model promatra svijet kroz entitete, njihove veze te njihove atribute. Još se koristi postupcima apstrakcije, generalizacije, klasifikacije i agregacije. Apstrakcija je uočavanje općeg, nužnog, bitnog ili glavnog, te ispuštanje sporednog, posebnog, slučajnog ili nebitnog. Apstrakcija je proces teoretskog uopćavanja. U klasifikacije se entiteti opisuju, klasificiraju i grupiraju u klase, razrede odnosno tipove. Na primjer, svi studenti (Ivan, Marko, Janko) se mogu klasificirati u tip entiteta student. Pri tome svi oni imaju nekakva zajednička obilježja kao što su ime, prezime, datum i slično, a međusobno se razlikuju po vrijednostima tih obilježja. Kod generalizacije se niži tipovi entiteta uopćavaju tipom entiteta više razine. Na primjer tipovi entiteta radnik, direktor, umirovljenik, student se generaliziraju tipom entiteta osoba. Postoji nadtip i podtip, nadtip je entitet više razine, dok je podtip entitet niže razine. Agregacija je formiranje novog pojma , višeg stupnja, na temelju postojećih pojmova. Model entiteti – veze opisan je u više varijanata koje se bitno ne razlikuju, a služe se konceptima entitet, veza i atribut za opis konceptualnog modela podataka. Postoji više grafičkih prikaza u obliku dijagrama entiteti – veze. Pritom, postupci modeliranja nisu strogo propisani. Najkorištenija dva prikaza su: Chenovi dijagrami prema proširenom modelu entiteti – veze i Martinovi dijagrami. Uz njih postoji i niz njihovih izvedenica.

2.1.1.1. ENTITET

Entitet je nešto što postoji i nešto što se u stvarnom svijetu može identificirati[12]. Entitet je bilo koji objekt u sustavu kojeg se želi modelirati ili o kojem se žele sačuvati informacije. Može biti stvar, biće, događaj ili pojava kao: polaganje ispita, pisanje rada, student, umirovljenik, radnik, auto, kuća i slično. Svi entiteti imaju određeno obilježje

[12] ibidem, str. 45.

odnosno atribute. Prilikom određivanja naziva entiteta koriste se imenice i to u jednini jer se imenuje tip entiteta, a ne skup entiteta. Naziv bi trebao biti kratak i najbolje je uzeti naziv koji se već koristi u poslovanju.

Primjeri tipova entiteta:

- ljudi : OSOBA, DJELATNIK, KUPAC
- organizacije: PODUZEĆE, AGENCIJA
- stvari: PROIZVOD, REZERVNI DIO, AUTOMOBIL
- koncepti: RAČUN, RUTA
- mjesta: GRAD, URED
- događaji: NARUDŽBA, PROJEKT, LET

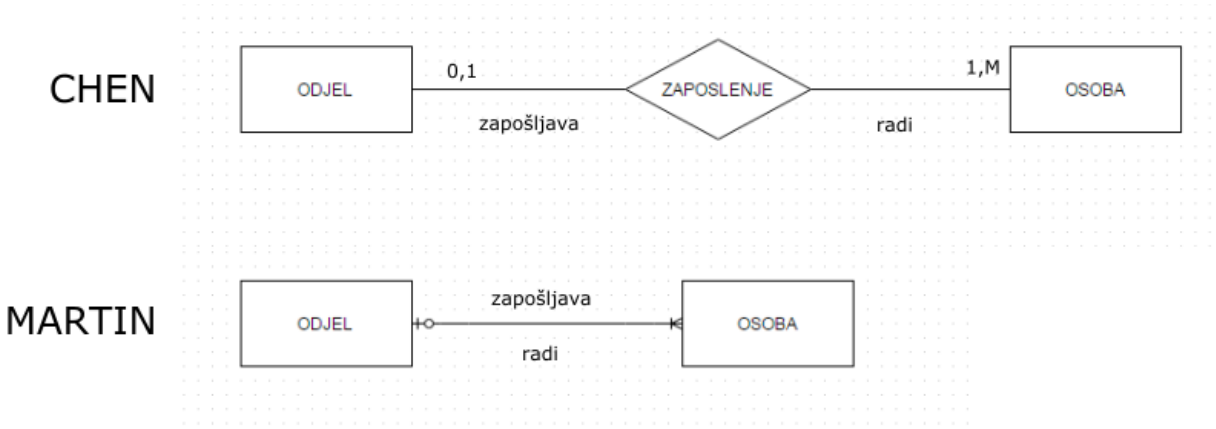
SLIKA 6. Chenov i Martinov prikaz entiteta

(Izvor: izradio student)

Na slici šest prikazan je Chenov i Martinov prikaz entiteta. Slabi entiteti grafički se različito prikazuju. Identifikacijski slab entitet označava se oznakom I uz vezu koja ga povezuje s jakim entitetom. Entiteti se jednako prikazuju u Martinovom prikazu, samo što se slabi entiteti posebno ne razlikuju.

2.1.1.2. VEZA

Veza je odnos između entiteta. S obzirom na to koliko postoji entiteta u vezi razlikuju se unarne veze, binarne veze i ternarne veze. Binarne veze su veze između dva entiteta, unarne veze su poseban slučaj binarnih u kojim na obje strane veze sudjeluju pojave istog tipa entiteta, dok su ternarne veze veze između tri entiteta.



SLIKA 7. Binarne veze

(izvor: izradio student)

Slika sedam prikazuje binarne veze. U Chenovom prikazu obično se navode nazivi veza, a ispuštaju uloge entiteta u vezi. Naziv veze stavlja se u znak veze odnosno romb, a uloga entiteta upisuje se uz liniju veze u kojoj entitet sudjeluje. U Martinovom prikazu veze se crtaju običnom linijom koja povezuje dva entiteta. Veza se ne imenuje, nego se navode uloge entiteta koji u njoj sudjeluju. Tekst iznad linije imenuje ulogu entiteta na lijevoj strani, a ispod linije ulogu entiteta na desnoj strani. Čitanje veze: svaki ODJEL mora zapošljavati jednu ili više osoba; svaka OSOBA može biti djelatnik u jednom ili ni u jednom odjelu.

Osnovni tipovi veza opisuju se kardinalnošću. Kardinalnost je omjer između povezanih entiteta. Pošto je kardinalnost nemoguće točno izraziti određuju se intervali u obliku donje i gornje granice. Uobičajene kardinalnosti su : jedan-prema-jedan (1:1), jedan-prema-više (1:M) i više-prema-više (M:M).

2.1.1.3. ATRIBUT

Atribut je svojstvo entiteta , odnosno tip entiteta opisan je skupom atributa. Svaka je osoba opisana atributima: maticni_broj, ime, prezime, datum_rođenja i slično. Atribut je funkcija koja skupu entiteta pridružuje odgovarajući skup vrijednosti određene domene. Domena definira moguće vrijednosti atributa. Vrijednost svakog atributa može se mijenjati tijekom vremena. Veze također mogu biti opisane atributima, ali i ne moraju.

Primjer entitet, veza i atribut:

Tip entiteta: STUDENT, PROFESOR, PREDMET

Veze: upisao između STUDENT I PREDMET, predaje između PROFESOR I PREDMET

Atributi: DATUM UPISA, GODINA STUDIJA, ECTS, OCJENA, IME, PREZIME

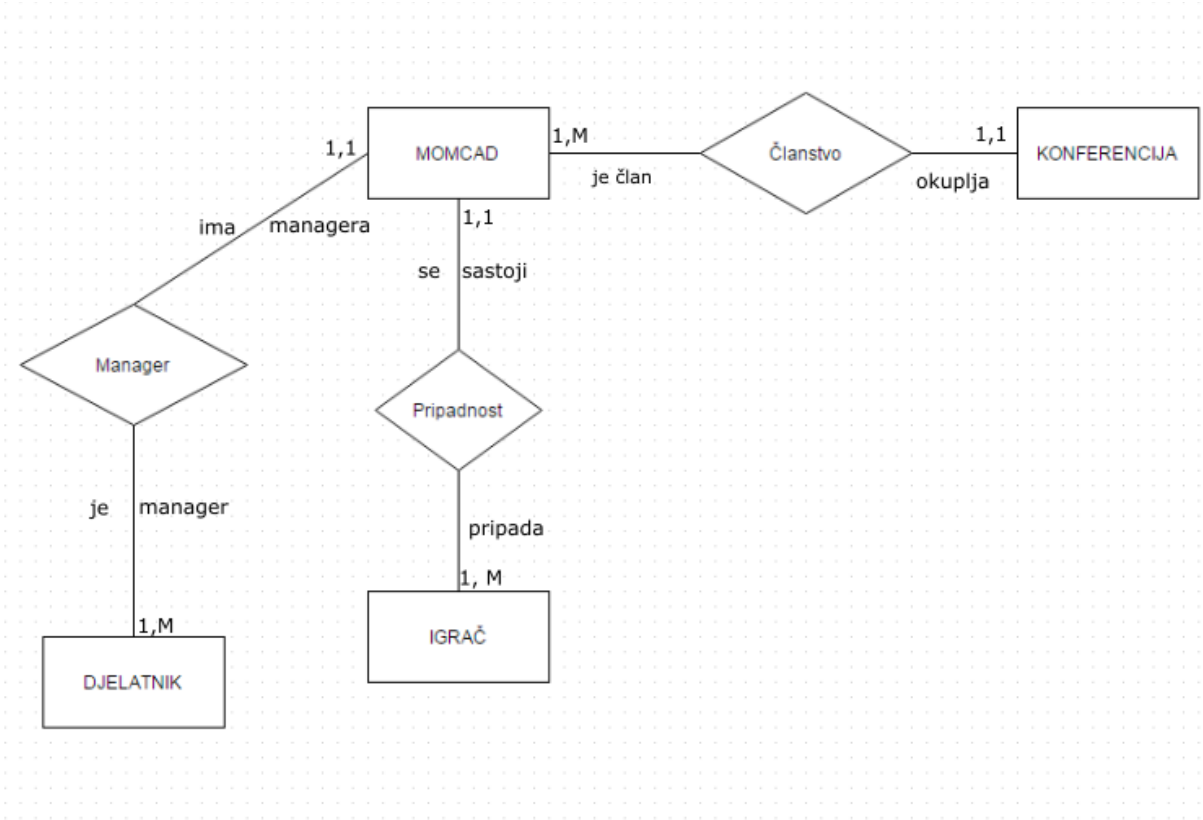
SLIKA 8. Prikaz atributa

(Izvor: izradio student)

Slika osam prikazuje atribute. Atributi se crtaju unutar elipse ili kruga. Ključni atributi mogu se podcrtati. Najčešće se atributi ne crtaju, jer velik broj atributa čini dijagram nepreglednim. Popis atributa daje se posebno ili se navodi unutar entiteta.

Ključ se koristi za identifikaciju pojava entiteta. Pojave entiteta razlikuju se po vrijednostima svojih atributa pa je dovoljno pronaći skup atributa čija vrijednost omogućava jednoznačnu identifikaciju svake pojave entiteta. Na primjer, matični broj omogućuje identifikaciju osobe, to vrijedi i za kombinaciju imena, prezimena i matičnog broja. S druge strane kombinacija imena i prezimena nije ključ jer ima više osoba koje se jednako zovu.

Dijagram entiteti – veze



SLIKA 9. Dijagram entiteti – veze

(Izvor: izradio student)

Slika devet predstavlja dijagram entiteti – veze na primjeru NBA lige. Svaka MOMCAD sastoji se od jednog ili više IGRAČA. Svaki IGRAČ pripada jednoj i samo jednoj

MOMCADI. Svaka MOMCAD je član jedne i samo jedne KONFERENCIJE. Svaka KONDERENCIJA okuplja jednu ili više MOMCADI. Svaka MOMCAD mora imati kao managera jednog ili više DJELATNIKA. Svaki DJELATNIK može biti manager u jednoj i samo jednoj MOMCADI. Dijagram je nacrtan u jednostavnom prikazu, jer su nacrtani samo entiteti i njihove veze. U potpunom modelu definirani su i atributi entiteta (MOMCAD ima atribute: id_momcadi, naziv_momcadi, broj_igraca, naziv_arene, kapacitet_arene).
Dijagram je izrađen pomoću web stranice draw.io.

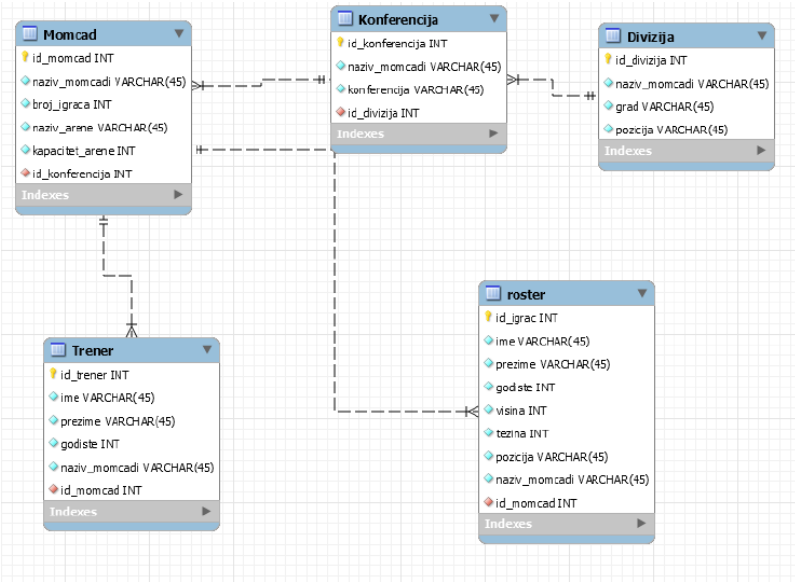
2.2. LOGIČKO MODELIRANJE PODATAKA

Logičko ili implementacijsko modeliranje dolazi odmah nakon konceptualnog modeliranja podataka i predstavlja drugu fazu oblikovanja baze podataka. Cilj je kreirati relacijsku shemu baze, odnosno shemu koja opisuje logičku strukturu baze. Važno je napomenuti da ovaj model ne rezultira razradom konačne fizičke strukture podataka. Logički model mogu opisivati više modela podataka kao što su relacijski, objektni, hijerarhijski, mrežni ili datotečni model. Danas se najviše koristi relacijski dok se mrežni, hijerarhijski i datotečni model sve manje koriste jer su u mnogim aspektima slabiji od relacijskog modela. Što se tiče objektno orijentiranog modela, on se još uvijek nalazi u fazi teoretskog oblikovanja, te se njegova ozbiljnija primjena očekuje u narednim godinama. Logički model prikazuje sve entitete i veze među njima, atribute, primarne ključeve, strane ključeve te normalizaciju podataka.

Logičko modeliranje podataka svodi se na:

- pretvorbu konceptualnog modela podataka izvedenog u obliku modela entitet-veta u logičku shemu baze podataka
- doradu logičke sheme baze podataka, ako to proizlazi iz zahtjeva za obradom podataka

Primjer relacijskog dijagrama



SLIKA 10. Relacijski dijagram

(izvor: izradio student)

Relacijski dijagram NBA lige sadrži 5 međusobno povezanih tablica. Svaka od tih tablica ima svoje primarne ključeve i strane ključeve koje su preuzeli kako bi ih se povezalo s drugim tablicama. Kao što je prikazano u tablica. Divizija je povezana s tablicom Konferencija preko stranog ključa id_divizija. Isto tako tablica Momčad je povezana sa tablicama Konferencija preko stranog ključa id_konferencija. Nadalje, tablica Momcad povezana je sa tablicom Trener i tablicom Roster preko stranog ključa id_momcad. Više momčadi može biti u jednoj Konferenciji, isto više konferencija može biti u jednoj Diviziji. Momčad može imati više igrača i više trenera iako to nije uobičajena praksa.

2.3. FIZIČKO MODELIRANJE PODATAKA

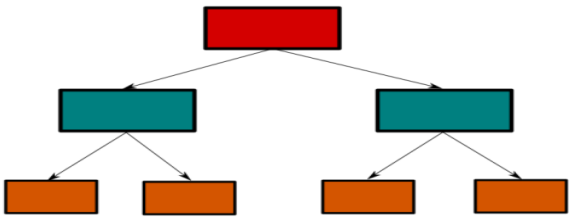
Fizičko modeliranje podataka treća je faza oblikovanja baze podataka. Cilj je stvoriti fizičku shemu odnosno opis njene fizičke građe. Fizička shema je skup naredbi koje su napisane u SQL-u ili nekom drugom programskom jeziku kojeg razumije sustav za upravljanje bazom podataka. Prilikom pokretanja tih naredbi kreira se fizička baza podataka. Fizički slog je fizička realizacija logičkog sloga. On sadrži podatke potrebne za prikaz i manipulaciju slogovima na računalu. Glavna svrha fizičkog modeliranja podataka je kreirati fizičku organizaciju podataka koja će imati brz pristup slogovima. O nekim aktivnostima koje se obavljaju tijekom fizičkog modeliranja brine sustav za upravljanje fizičkim prostorom, a o nekima administrator baze podataka. Aktivnosti su sljedeće: modeliranje strukture sloga, segmentiranje i grupiranje slogova, izbor metode pristupa slogovima i optimizacija pristupnih puteva.

3. LOGIČKI MODELI PODATAKA

Kako bi se podaci pohranili u bazu na odgovarajući način potrebno je koristiti odgovarajući model podataka. Model podataka je skupa pravila o tome kako može izgledati logička struktura baze podataka. Iz toga proizlazi da bi podaci u bazi trebali biti logički organizirani na način koji podržava ili koji može prepoznati odabrani sustav za upravljanje bazom podataka. Najvažniji modeli podataka su sljedeći: hijerarhijski, mrežni, relacijski i objektni model podataka. 60-ih i 70-ih godina 20. stoljeća u uporabi su bili hijerarhijski i mrežni model dok danas uglavnom prevladava relacijski model. Očekivalo se da će do danas već prevladavati objektni model ali za sada se to nije dogodilo. U današnje vrijeme kada se spomene baza podataka misli se na relacijsku bazu. Važno je napomenuti da svi poznati sustavi za upravljanje bazama podržavaju isključivo relacijski model.

3.1. HIJERARHIJSKI MODEL PODATAKA

Hijerarhijski model podataka prvi je razvijeni model podataka. Koristio se 60-ih i 70-ih godina prošlog stoljeća . Opisuje se skupom slogova povezanih vezama. On organizira slogove u povezane grupe podataka koje sliče obiteljskom stablu. Postoje dvije vrste slogova: roditelji i dijete. Slogovi roditelji su nadređeni slogovima djeci. Svaki slog dijete može imati samo jedan slog roditelji što je i temelj hijerarhijskog modela. Hijerarhijski model podataka prikazuje se skupom hijerarhijskih dijagrama strukture podataka jer se zbog ograničenosti hijerarhijskog prikaza mnogi problemi ne mogu prikazati samo jednim hijerarhijskim dijagramom strukture podataka. Spremanje i pronalaženje slogova počinje od osnovnog sloga (root) i kreće prema dolje. Najveća prednost hijerarhijskog modela je brzo pretraživanje i pohranjivanje podataka. Najveća mana ovog modela je odnos M:N te kada se briše slog roditelja brišu se i svi slogovi dijete. Modificiranje hijerarhijske baze podataka je komplicirano i zahtjeva iskusne programere. Danas je ovaj model podataka gotovo istisnut iz uporabe.



SLIKA 11. Hijerarhijski model podataka

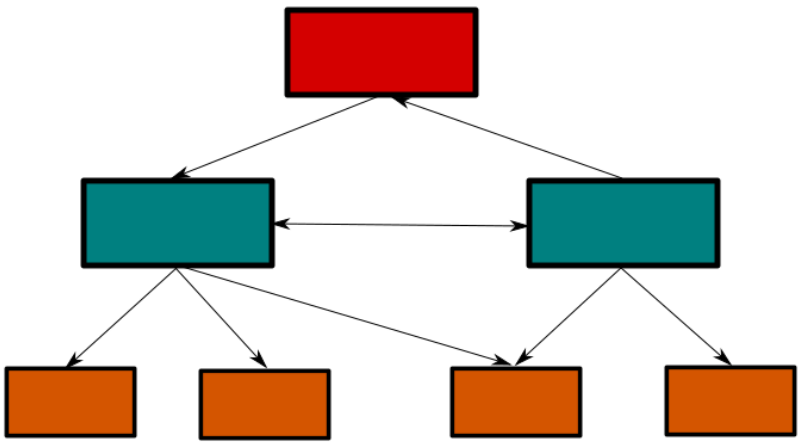
(IZVOR: izradio student)

Slika prikazuje hijerarhijski model podataka. Vidi se da svaki slog dijete može imati jedan slog roditelji

3.2. MREŽNI MODEL PODATAKA

Mrežni model opisan je skupom međusobno povezanih slogova. Glavna karakteristika mrežnog modela je to što omogućava svakom zapisu da ima višestruke roditelje i djecu, kreirajući tako mrežnu strukturu. Mrežni model je prošireni hijerarhijski model te se u njemu mogu prikazati odnosi više-prema-jedan i jedan-prema-više. Za razliku od hijerarhijskog, mrežni model može izravno uspostaviti vezu između slogova na različitim hijerarhijskim razinama. Mrežna baza podatak temelji se na mrežnom modelu podataka. U teoriji mrežna struktura podataka povezuje svaki element sa svakim. Ipak, u mrežnoj bazi podataka takva povezanost ne postoji, ona je samo teorijske naravi. U ovom modelu podaci se prikazuju pomoću skupa slogova, a njihovi odnosi prikazuju se pomoću veza. Takve veze nazivaju se pokazivači. Temeljna razlika između hijerarhijskog modela i mrežnog modela u odnosu na relacijski je u tome što relacijski model ne koristi pokazivače. Relacijski model umjesto toga relacije između slogova kreira pomoću vrijednosti atributa unutar slogova.

Struktura podataka u mrežnom modelu opisuje se dijagramom strukture podataka, čija je namjena jednaka dijagramu entitet-veza. Veza se označava linijom između dva sloga dok se slog označava pravokutnikom.



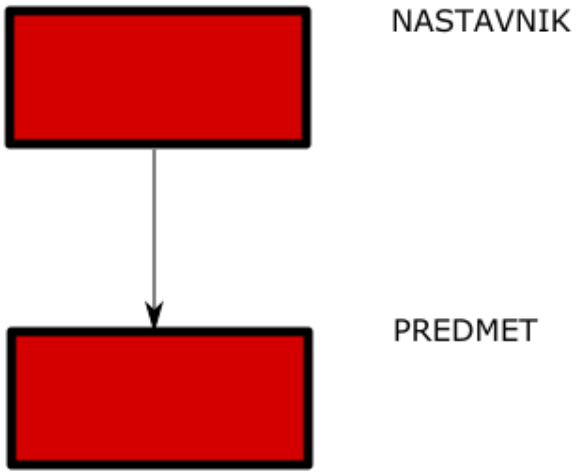
SLIKA 12. Mrežni model podataka

(IZVOR: izradio student)

Na slici je prikazan mrežni model podataka. Vidi se da je on kompleksniji od hijerarhijskog jer svaki slog dijete može imati više slogova roditelja.

3.2.1. DBTG CODASYL mrežni model

Prvi standard na području baze podataka je bio Izvještaj CODASYL DBTG 1971. godine. Posljednji važeći standard mrežnog modela je iz 1978. godine, a prijedlog standarda iz 1981. godine nikada nije službeno usvojen. U ovom modelu koriste se veze tipa jedan-prema-više. Veze više-prema-više nisu dozvoljene zbog jednostavnosti implementacije modela. Veze jedan-prema-jedan predstavljaju se vezama jedan-prema-više.



SLIKA 13. Skup
(IZVOR: izradio student)

Fizička implementacija veze jedan-prema-više sastoji se od dva sloga koji su povezani u skup. U svakom imenovanom skupu (skup PREDAJE) postoji vlasnik skupa (slog NASTAVNIK) i član skupa (slog PREDMET). Svaki skup može imati više pojava skupa, a u svakoj pojavi skupa nalazi se jedna pojava vlasnik i proizvoljni broj članova skupa. Slogovi koji sudjeluju u jednom skupu bez obzira jesu li vlasnici ili članovi skupa mogu istovremeno sudjelovati i u drugim skupovima. Modeliranje podatak i crtanje dijagrama strukture podataka je slično kao u modelu entitet-veze.

3.3. RELACIJSKI MODEL PODATAKA

U današnje vrijeme najviše se koristi relacijski model podataka zbog svoje jednostavnosti u razumijevanju i praktičnom djelovanju. Teorijski ga je zasnova E.F. Codd 60-ih godina prošlog stoljeća. Prve realizacije ovog modela bile su vrlo spore i neefikasne, ali razvojem informacijske tehnologije efikasnost i brzina su vrlo brzo rasle. Tako je relacijski model 80-ih godina postao najznačajniji, te ga danas koristi većina sustava za upravljane bazama podataka.

U relacijskom modelu baza podataka sastoji se od tablica koje se nazivaju relacije. Tablice se sastoje od više stupaca, a svaki stupac ima jedinstveno ime po kojem se razlikuje od ostalih u istoj bazi. Jedan stupac relacije sadrži vrijednost jednog atributa. Također, atribut ima svoje ime po kojem se rezlikuje od atributa na istoj relaciji. Svaki red u tablici predstavlja jedan zapis. Tablice se povezuju preko stupca koji se nalazi u obje tablice, ali važno je da taj stupac predstavlja jedinstveni podataka kao na primjer id_korisnika. U glavnoj tablici takav stupac naziva se primarni ključ, dok se u podređenoj tablici naziva strani ključ.

Osnovni ciljevi relacijskog modela su:

- omogućiti nezavisnost podataka,
- dati teorijske temelje za konzistentno semantičko postupanje s podacima i za rješavanje redundancije podataka,
- omogućiti razvoj skupovno orijentiranih jezika za obradu podataka,
- dati bogat model podataka za opis i obradu jednostavnih i kompleksnih podataka.

Prva dva cilja nisu ostvarena mrežnim i hijerarhijskim modelom dok ih relacijski model ostvaruje jednostavnim tabličnim prikazom podataka i razrađenom teorijom normalizacije podataka. Treći cilj ostvaren je uporabom operacija relacijske algebre, dok četvrti cilj, za sada, nije ostvaren na odgovarajući način.

Za rad s relacijskom bazom podataka postoji više jezika. Najpoznatiji su SQL, QUEL i QBE. Tijekom vremena kao standardni jezik izdvojio se SQL. Danas nema relacijske baze podataka koja ne koristi SQL. SQL je kratica za Structured Query Language. Nastao je od svog prethodnika SEQUEL-a, a razvijen je u IBM-u. Standardiziran je 1986. Godine. On podržava naredbe za stvaranje relacija, unos, promjenu podataka, davanje i oduzimanje ovlasti korisnicima i slično.

SLIKA 14. Relacijski model podataka

(IZVOR: Izradio student)

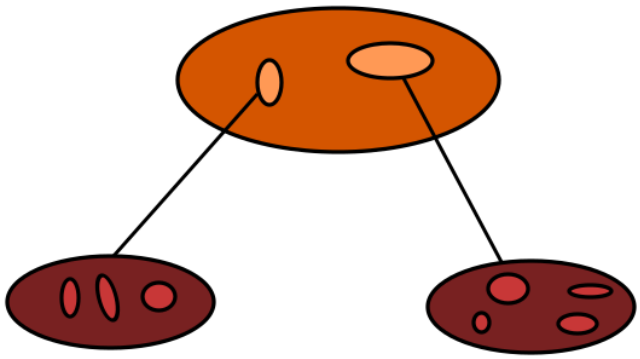
Na slici 14. prikazan je relacijski model podataka. Sastoji se od dvije tablice: zaposlenici i odjeli. Tablica zaposlenici obuhvaća sve podatke o zaposlenima kao što su ime, prezime, email, placa, id_zaposlenika i id_odjela. Tablica odjeli predstavlja informacije o odjelim. Tablice su povezane primarnim i vanjskim ključem što omogućava da se prilikom upita ispisuju informacije iz obje tablice.

3.4. OBJEKTNI MODEL PODATAKA

Relacijski model i model entiteti-veze uspješno se primjenjuju u poslovnim informacijskim sustavima u kojima su podaci jednostavnijeg tipa. Pojavom nestandardnih aplikacija kao što su uredsko poslovanje, ekspertni sustavi, baze znanja, javlja se potreba za bogatijim tipovima podataka kojih nema u gore navedenim modelima. Zbog toga se u novije vrijeme uvode novi modeli od kojih mnogi pripadaju objektno orijentiranim modelima podataka.

U objektnim modelima podaci se definiraju kroz objekte, a model se implementira kroz objektnu bazu podataka. U njoj se nalazi skup vrlo kompleksnih objekata. Korisnik upravlja kompleksnošću baze tako da dodaje ili oduzima objekte ili ih međusobno kombinira u nove kompleksne objekte. U samom objektu je opisano njegovo ponašanje i izvan njega ne postoji ništa što bi opisivalo njegovu strukturu.

Prvi prototip sustava za upravljanje bazom podataka objektnog modela javlja se 80-ih godina. Krajem 90-ih godina objektni model se prestaje koristiti, da bi 2004. doživio veliki uzlet. Zbog nedostatka standardizacije, pojedinci ili organizacije koji kreiraju objektni model definirali su skup pravila kojih bi se svatko trebao pridržavati prilikom uporabe objektnog modela. Najveći nedostatak je nepostojanje standardiziranog jezika kao što je SQL u relacijskom modelu.



SLIKA 15. Objektni model podataka

(IZVOR : izradio student)

Slika 15. prikazuje objektni model podataka. Podaci su spremljeni unutar objekata (male elipse), a objekti se nalaze unutar odgovarajuće klase (velike elipse).

4. OBJEKTNO ORIJENTIRANE BAZE PODATAKA

Objektno orijentirane baze podataka nastale su 80-ih godina kao odgovor na probleme relacijske baze podataka. Relacijske baze podataka nisu prikladne za aplikacije koje koriste složenije tipove podataka kao što su slike, multimedija i slično. Objektno orijentirane baze podataka nazivaju se još i baze objekata, te rade tako da se u bazu pohranjuju objekti. Sustav za upravljanje bazama podataka koji implementira objektni model podataka naziva se OOSUBP odnosno objektno orijentirani sustav za upravljanje bazama podataka. Implementacije objektnog sustava za upravljanje bazama podataka programirane su za specifičan programski jezik i međusobno se razlikuju.

1989. godine izdaje se Manifest o objektno-orijentiranim sustavima baza podataka odnosno izdaje se znanstveni članak o svojstvima koje bi trebao zadovoljavati OOSUBP.
Koncepti objektno-orijentiranog sustava[13]:

- razredi (klase)
- složeni objekti
- identitet objekta
- hijerarhije klasa
- učahurivanje
- Nadjačavanje, preopterećivanje i kasno vezivanje

Koncepti sustava za upravljanje bazama podataka[14]:

- perzistencija podataka
- fizička organizacija
- paralelni rad
- oporavak baze podataka
- ad hoc upitni jezik

Najpoznatiji komercijalni sustavi za upravljanje bazom podataka koji se temelje na objektnom modelu su: GemStone, ObjectDB, ObjectivitiDB, ObjectStore i Versant Object Database.

[13] (https://www.fer.unizg.hr/_download/repository/3._OOBP_i_ORBP.pdf)
[14] loc.cit.

4.1. GRAĐA OBJEKTNOG MODELA

Objekt je osnovni pojam objektnih modela podataka. On predstavlja cjelinu koja se može samostalno promatrati i obrađivati. Objekt je vrlo sličan entitetu modela entiteti-veze te se može definirati kao apstrakcija nečega u problemskoj domeni, o čemu se prikupljaju podaci i što skriva (sadrži) vrijednosti svojih atributa (obilježja) i svog ponašanja[15].

Karakteristike objekta slične su karakteristikama entiteta, pa tako svaki objekt ima više atributa koji ga opisuju. Također, svi slični objekti imaju isti skup atributa i pripadaju jednoj klasi. Objekt ima prepoznatljiv ponašanje koje je opisano operacijama odnosno servisima koji se obavljaju nad njim. Tu dolazi do temeljne razlike između entiteta i objekta. Entitet sadrži atribute dok objekt osim atributa sadrži i ponašanje.

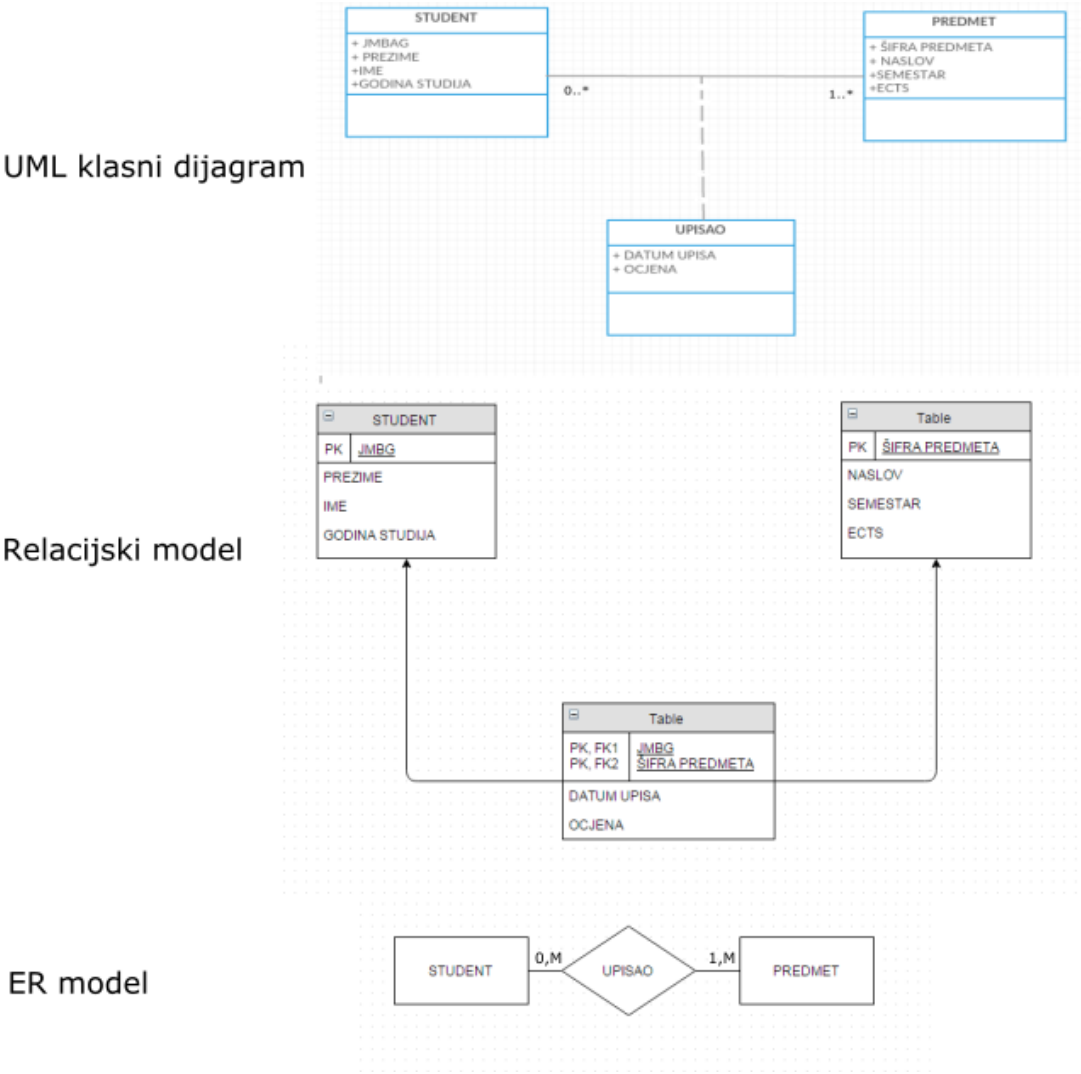
Objekti i entiteti grade različite strukture. U raspoznavanju strukture koriste se principi apstrakcije koji su već opisani u modelu entiteti – veze. To su generalizacija/specijalizacija i agregacija. Koncept atributa jednak je istom konceptu modela entiteti – veze. Za razliku od objekata za koje se smatra da su stabilni, njihovi atributi se mijenjaju pomoću servisa koji su definirani nad objektom. Preporučuje se da svaki atribut ima samo jednu vrijednost zbog toga što se tako dobivaju jednostavniji modeli. Atribut koji ima više vrijednosti eliminira se iz modela uvođenjem novog objekta.

Svaki objekt bi trebao imati mogućnost identifikacije u skupu objekata iste klase. Za to se koriste identificirajući atributi i implicitni identifikatori. U određenom modelu mogu postojati jednostavni i kompleksni objekti. Jednostavni model opisan je jednostavnim atributima te može biti temelj za kreiranje kompleksnijih objekata za manipulaciju na višem stupnju apstrakcije. Kompleksni objekt je objekt čiji atributi mogu biti i sami objekti.

Servis opisuje ponašanje objekta jedne klase. U objektnim modelima servisi opisuju operacije nad podacima. Odnosno, vrijednosti atributa objekata opisuju stanje objekata, a servisi se koriste za promjenu njihovog stanja. Potrebni servisi definiraju se: utvrđivanjem stanja objekata, utvrđivanjem potrebnih servisa, utvrđivanjem poruka potrebnih za pokretanje servisa i razradom detalja servisa.

Osnovna razlika između objektnog modela i modela entiteti – veze je u definiranju servisa. Model entiteti – veze opisuje samo podatkovnu komponentu informacijskog sustava odnosno samo razrađuje strukturu podataka. Znači, model entiteti – veze ne predstavlja konceptualni opis kompletnog informacijskog sustava dok objektni model opisuje i procesnu komponentu informacijskog sustava, pa on predstavlja konceptualni opis kompletnog sustava.

[15] Varga, M., op.cit., str. 80.



SLIKA 16. ER model, Relacijski model i UML klasni dijagram

(IZVOR: izradio student)

Na slici su prikazane razlike između različitih predložaka za prikaz sheme baze podatka. Osnovna razlika je što u relacijuskom modelu postoji primarni ključ, dok u ER modelu ne postoji. U ER modelu se atributi ne pišu zbog jednostavnosti, dok su u relacijskom modelu i UML klasnom dijagramu prikazani.

4.2. PREDNOSTI I NEDOSTACI OBJEKTNOG MODELA

Objektni model ima određene prednosti i nedostatke. Prednost objektnog modela ogleda se u tome što se pomoću njega brže i bolje upravlja sa složenim objektima i vezama te se ostvaruje brži pristup podacima, jer se objekti ne moraju izvlačiti iz tablica svaki put kada ih se želi koristiti. Također, objektni model podržava hijerarhiju, klase i nasljeđivanje. U objektnom modelu nema potrebe za primarnim ključevima jer se oni automatski generiraju. Omogućuju bolju povezanost baze podataka i aplikacije zbog toga što su kompatibilni s programskim jezicima kao što su Java i C++.

Nedostatak objektnog modela je to što nema logičke neovisnosti podataka odnosno izmjene na bazi zahtijevaju izmjene u aplikaciji i obrnuto. Nadalje, nedostatak je nepostojanje standardiziranog jezika kao što je SQL u relacijskom modelu te nemogućnost međusobnog razmjenjivanja podataka između objektne i relacijske baze.

4.3. ODMG STANDARD

ODMG standard (Object Database Managment Group) utemeljen je 1991. godine. Cilj ODMG standarda je bio kreiranje skupa standarda koji bi korisnicima omogućili razvoj prenosivih aplikacija odnosno aplikacija koje mogu koristiti objektne baze podataka.

ODMG standard sastoji se od[16]:

- objektnog modela,
- jezika za specificiranje objekata,
- objektnog upitnog jezika,
- veze na programske jezike,
- uključivanja jezika za specificiranje objekata koji je ovisan o odabranom programskom jeziku
- pružanja aplikacijskog programskog sučelja za preslikavanje tipova podataka.

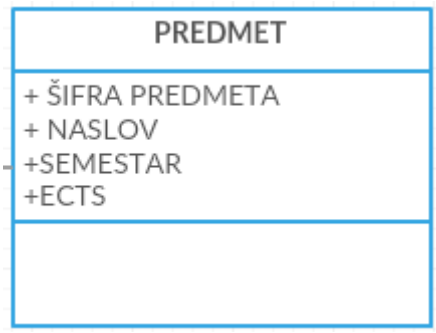
Osim toga, definirani su i standardi za vezivanje s objektno orijentiranim programskim jezicima (Java, C++). Važno je kod objektnog modela da se objekti iz prirode odnosno njihovi modeli ne rastavljaju na dijelove već čine jednu cjelinu u okviru sustava.

[16] (https://www.fer.unizg.hr/_download/repository/3._OOBP_i_ORBP.pdf)

5. UNIFIED MODELING LANGUAGE (UML)

UML je jezik za modeliranje koji se koristi grafičkim prikazom za izradu apstraktnog modela promatranog sustava. U području razvoja aplikativnih rješenja UML je standardni jezik za specifikaciju objektnog modela. On nije ograničen samo na modeliranje aplikativnih rješenja već se može upotrijebiti i za modeliranje poslovnih procesa, organizacijskih struktura i ostalih sustava. Danas je UML industrijski standard jer se javlja potreba za nekim jezikom pomoću kojeg bi dizajneri softvera mogli međusobno komunicirati. Najveći problem u prošlosti proizlazio je upravo iz komunikacije. UML se stalno poboljšava i usavršava.

Postoje strukturalni i bihevioralni UML dijagram. Strukturni dijagrami su dijagrami kojima je svrha prikazati statičke strukture sustava, elemente neovisno o vremenu, uključene klase, komponente i objekte. Bihevioralni dijagrami prikazuju dinamičko ponašanje odnosno odnose između objekata u sustavu. Oni imaju podskup koji se naziva interakcijski dijagram te je kod njih naglašenija interakcija među objektima. Klasni dijagram jedan je od standardnih UML dijagram i orginalno služi za prikaz klasa objekata i veza između tih klasa. Može se upotrijebiti za prikaz konceptualne sheme (slika 16.) na taj način da se entitet interpretira kao posebna vrsta klase koja ima atribute, ali nema operacije. Entitet se tada crta kao pravokutnik s upisanim imenom entiteta na vrhu i upisanim imenom svih atributa u sredini. Veza se crta kao spojnica između pravokutnika s upisanim imenom u sredini i upisanim oznakama kardinalnosti na krajevima. Dijagram sadržava svu potrebnu dokumentaciju te nema potrebe za tekstualnim nadopunama.

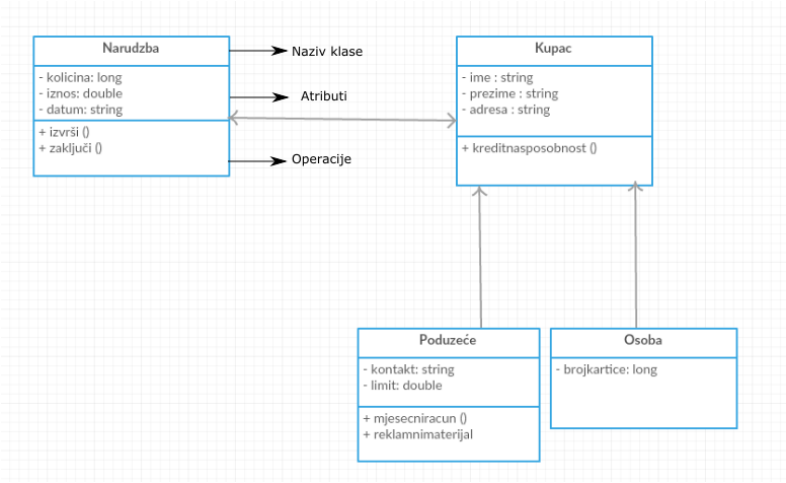


Slika 17. Entitet (klasa) za prikaz konceptualne sheme

(izvor: izradio student)

Na slici 17. je entitet koji je intepretiran kao posebna vrsta klase koja ima atribute, ali nema operacije.

Primjer 1. Online kupovina (UML klasni dijagram)

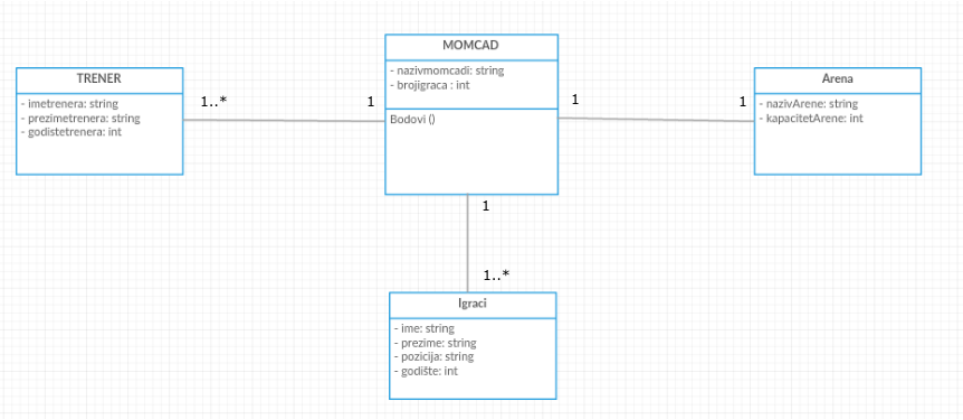


SLIKA 18. Dijagram klase

(IZVOR: izradio student)

Slika prikazuje primjer dijagrama klase. Cijeli proces počinje tako da kupac ispuni narudžbu potom se provjerava kreditna sposobnost kupca te ako je sve u redu izvršava se plaćanje pomoću broja kreditne kartice kupca. Nakon toga izdaje se račun, narudžba se izvrši te uz nju dolazi reklamni materijal. Dijagram je izrađen pomoću web stranice creately.com.

Primjer 2. NBA momcad (UML klasni dijagram)



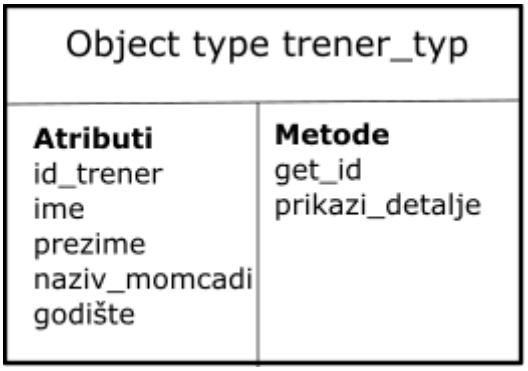
SLIKA 19. NBA dijagram klase

(Izvor: izradio student)

Na slici 16. Prikazan je UML klasni dijagram NBA momčadi. Momčad može imati jednu i samo jednu arenu. Momčad može imati jednog ili više trenera. Momčad može imati jednog ili više igrača.

6. RELACIJSKO – OBJEKTNE DB NA PRIMJERU ORACLE DB

U današnje vrijeme gotovo svi relacijski sustavi za upravljanje bazom podataka (RDBMS) imaju prefiks "O" jer se predstavljaju kao objektno – relacijski sustavi (ORDBMS)[17]. Nazivaju se objektno – relacijski zbog toga što podržavaju određene objektno orijentirane mogućnosti (definirane u SQL: 1999. I SQL: 2003. standardu)[18]. Objektno – relacijski sustavi pojavili su se 90-ih godina kao proširenje RDBMS sustava s nekim objektnim mogućnostima. Najveći doprinos objektnog pristupa relacijskom je korisnički definirani tipovi podataka odnosno klasa. Korisnički definirani tipovi podataka omogućuju trajno smještanje objekata i uključuju bitna objektno – orijentacijska svojstva. Oracle objektni tipovi su korisnički definirati tipovi podataka. Sastoje se od atributa i metoda (procedura i funkcija). Metode mogu biti one koje pripadaju objektu (member) ili cijeloj klasi (static). Posebne member metode su metode za uspoređivanje (MAP). Može se napraviti podtip odnosno potklasa koja može imati nove atribute / metode ili se postojeća metoda može pojačati. Za sada nema privatnih ili zaštićenih atributa i metoda, svi su javni. Objektni tipovi služe kao nacrt koji definira strukturu i ponašanje. Mogu se koristiti standardni tipovi podataka kao što su NUMBER ili VARCHAR2.



SLIKA 20. Objektni tip
(izvor: izradio student)

Slika 20. pokazuje primjer objektnog tipa trener_typ. On sadrži atribute id_trener, ime, prezime, naziv_momcadi i godište, te metode get_id i prikazi_detalje. Objekt na slici može se kreirati pomoću SQL izraza CREATE TYPE.

[17] (www.hroug.hr/.../303_Sirotić%20Objektno-relacijske%20mogućnosti.pdf)

[18] loc.cit.

```
CREATE TYPE trener_typ AS OBJECT (

  id_trener           NUMBER,
  ime                 VARCHAR2(20),
  prezime             VARCHAR2(25),
  naziv_momcadi       VARCHAR2(25),
  godiste             DATE,

  MAP MEMBER FUNCTION get_id RETURN NUMBER,
  MEMBER PROCEDURE prikazi_detalje ( SELF IN OUT NOCOPY trener_typ ));

/
CREATE TYPE BODY trener_typ AS
  MAP MEMBER FUNCTION get_id RETURN NUMBER IS
  BEGIN
    RETURN id_trener;
  END;
  MEMBER PROCEDURE prikazi_detalje ( SELF IN OUT NOCOPY person_typ ) IS
  BEGIN
    -- use the PUT_LINE procedure of the DBMS_OUTPUT package to display
details
   DBMS_OUTPUT.PUT_LINE(TO_CHAR(id_trener) || ' ' || first_name || ' ' ||
last_name);
   DBMS_OUTPUT.PUT_LINE(email || ' '  || phone)
   END;

END;

/
```

SLIKA 21. Kreiranje objekta

(izvor: izradio student)

Na slici 21. Prikazano je na koji način se kreira objekt trener_typ pomoću SQL izraza CREATE TYPE. U ovom primjeru definirane su specifikacije objekta i tijelo objekta.

Oracle objektno relacijske tablice su tablice kod kojih su stupci temeljeni na objektnom tipu. Objektni tipovi služe kao nacrt koji definira strukturu i ponašanje.

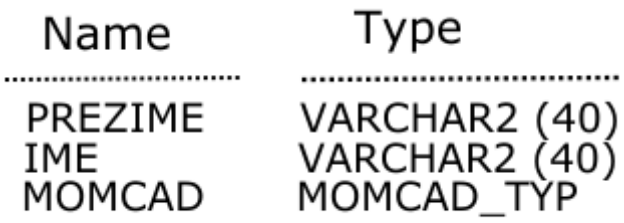
```
CREATE TYPE momcad_typ AS OBJECT(
  naziv_momcadi VARCHAR2(20),
  naziv_arene VARCHAR2(30),
  broj_igraca NUMBER
 )
 /
CREATE TABLE trener(
  prezime VARCHAR2(40) PRIMARY KEY,
  ime VARCHAR2 (40),
  momcad momcad_typ

 )
 /
```

SLIKA 22. Objektno – relacijska tablica

(izvor : izradio student)

Slika 22. prikazuje na koji način se kreiraju objektno – relacijske tablice.

Opis tablice trener pokazat će samo tri stupca:



SLIKA 23. Opis tablice

(izvor: izradio student)

Međutim, tablica trener interno ima šest stupaca,a to su vidljivi stupci IME i PREZIME, tri nevidljiva stupca koji predstavljaju komponente stupca – objekta MOMCAD (naziv_momcadi, naziv_arene, broj_igraca), te vidljivi stupac MOMCAD koji je dugačak samo 1 byte, a Oracle ga koristi za interne potrebe.

Oracle objektne tablice su one tablice kod kojih cijeli redak reprezentira jedan objekt.

```
CREATE TYPE trener_typ AS OBJECT(
  id VARCHAR2(10),
  ime VARCHAR2(20)
 )
 /
CREATE TABLE trener OF trener_typ
  ( PRIMARY KEY (id))
  OBJECT IDENTIFIER IS PRIMARY KEY

 )
 /
```

SLIKA 24. Oracle objektne tablice

(izvor: izradio student)

Slika 24. prikazuje kreiranje objektne tablice. Objektna tablica izrađuje se tako da se kreira objektni tip (trener_typ) te se potom na temelju objektnog tipa kreira objektna tablica. Iako je moguće, ovakve čiste objektne osobine u relacijskoj bazi trebalo bi izbjegavati.

7. ZAKLJUČAK

U prošlosti su se najviše koristili hijerarhijski i mrežni modeli podataka dok danas prevladava relacijski model. Unatoč tome, objektne baze podataka sve su više zastupljene u poslovnom svijetu. Razlog tomu je sve veći broj multimedijalnog sadržaja i to što relacijske baze nisu prikladne za aplikacije koje koriste složene tipove podataka. Objektne baze se mogu implementirati u poslovne sustave koji koriste kompleksne podatke. Mnogi su prepoznali potencijal objektnih baza podataka te su tako nastale objektno – relacijske baze podataka. Objektno – relacijske baze podataka su rezultat pokušaja spajanja najboljeg iz relacijskog i objektno orijentiranog pristupa. Odnosno, radi se o relacijskom modelu koji je proširen objektnom orijentacijom i konstrukcijom koje omogućuju rukovanje s novim tipovima podataka. Može se reći da je objektno – relacijski pristup kreiran kako bi se eliminirale mane dva najvažnija pristupa. Najveći nedostatak objektnog modela je nepostojanje standardizacije. Kada bi se taj problem riješio objektni model bi postao popularniji i možda bi čak nadvladao relacijski model. Objektne baze podataka zasigurno imaju potencijala, ali u isto vrijeme imaju mane preko kojih je teško prijeći. Zbog toga smatram da će u budućnosti najzastupljenije biti baze podataka s određenim objektno orijentiranim proširenjima.

8. LITERATURA

a) KNJIGE

Mladen Varga; Baze podataka: konceptualno, logičko i fizičko modeliranje podataka/ Zagreb,DRIP 1994.

Varga, Čerić: Informacijske tehnologije u poslovanju, Element, Zagreb 2004.

Manger: Baze podataka, PMF – Matematički odjel, Zagreb, 2011.

Radovan: Baze podataka Relacijski pristup i SQL, Informator, Zagreb, 1993.

b) INTERNET

1. http://e-ucenje.oet.unipu.hr/file.php/30/BP-Vjezbe-2011-03-28-notacije.pdf (09.09.2015.)

2. https://www.fer.unizg.hr/_download/repository/3._OOBP_i_ORBP.pdf (12.09.2015)

3. http://tecajevi.freeservers.com/isbaza.htm (07.09.2015)

4. http://www.etf.ucg.ac.me/materijal/1286268142Lekcija_4_-_Model_Objekti-Veze.pdf (07.09.2015

5. http://www.icentar.ba/showtopic.php?id=1095 (08.09.2015)

6. http://docs.oracle.com/cd/B28359_01/appdev.111/b28370/create_type.htm#LNPLS01375 (17.09.2015)

7. www.hroug.hr/.../303_Sirotić%20Objektno-relacijske%20mogućnosti.pd (17.09.2015)

9. POPIS SLIKA

SLIKA 1: Arhitektura baze podataka.....3
SLIKA 2: Prikaz baze strukturiranih podataka.....4
SLIKA 3: Razvojni ciklus baze podataka.....6
SLIKA 4: Shema baze podataka.....8
SLIKA 5: Od stvarnog svijeta do baze podataka.....10
SLIKA 6: Chenov i Martinov prikaz entiteta.....12
SLIKA 7: Binarne veze.....13
SLIKA 8: Prikaz atributa.....14
SLIKA 9: Dijagram entiteti – veze.....14
SLIKA 10: Relacijski dijagram.....16
SLIKA 11: Hijerarhijski model podataka.....17
SLIKA 12: Mrežni model podataka.....18
SLIKA 13: Skup.....19
SLIKA 14: Relacijski model podataka.....21
SLIKA 15: Objektni model podataka.....22
SLIKA 16: ER model, Relacijski model i UML klasni dijagram.....26
SLIKA 17: Entiteti (klasa) za prikaz konceptualne sheme.....27
SLIKA 18: Dijagram klase.....28
SLIKA 19: NBA dijagram klase.....28
SLIKA 20: Objektni tip.....29
SLIKA 21: Kreiranje objekta.....30
SLIKA 22: Objektno relacijska tablica.....30
SLIKA 23: Opis tablice.....31
SLIKA 24: Oracle objektne tablice.....31