Okvirni projekt razvoja baze podataka - vođenje evidencije zaštitnih sredstava

Hren, Nikola

Master's thesis / Specijalistički diplomski stručni

2015

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Karlovac University of Applied Sciences / Veleučilište u Karlovcu**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:128:193543

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-25**

VELEUČILIŠTE U KARLOVCU
Karlovac University of Applied Sciences

Repository / Repozitorij:

Repository of Karlovac University of Applied Sciences - Institutional Repository

zir.nsk.hr

dabar
DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

VELEUČILIŠTE U KARLOVCU

SPECIJALISTIČKI STUDIJ SIGURNOSTI I ZAŠTITE

Nikola Hren

OKVIRNI PROJEKT RAZVOJA BAZE PODATAKA: VOĐENJE EVIDENCIJE ZAŠTITNIH SREDSTAVA

ZAVRŠNI RAD

Karlovac, 2015.

VELEUČILIŠTE U KARLOVCU

SPECIJALISTIČKI STUDIJ SIGURNOSTI I ZAŠTITE

Smjer: Zaštita na radu

OKVIRNI PROJEKT RAZVOJA BAZE PODATAKA: VOĐENJE EVIDENCIJE ZAŠTITNIH SREDSTAVA

ZAVRŠNI RAD

Mentor:
dr.sc. Damir Kralj

Student:
Nikola Hren

Karlovac, 2015.

Sažetak

U ovom završnom radu prikazan je okvirni projekt razvoja jedne baze podataka unutar informacijskog podsustava sigurnosti i zaštite na radu. Za tu svrhu prethodno je su analizirana načela, metode i tehnike projektiranja baza podataka. Detaljno je prikazan i pregled osnovnih funkcija alata Microsoft Access 2010 koji je korišten u realizaciji praktičnog dijela zadatka. Praktični dio obuhvatio je projekt i izrada baze podataka i jednostavne aplikacije za vođenje evidencija i raspodjelu sredstava zaštite na radu. Posebna pažnja posvećena izradi korisničkih sučelja za unos i pregled podataka, kao i izradi potrebitih izvješća. Analizirani su i načini na koje je moguće ovu bazu funkcionalno uklopiti u postojeći veći informacijski sustav.

Ključne riječi: baze podataka, sigurnost i zaštita na radu, zaštitna sredstva, MS Access 2010, informacijski sustavi, metode i postupci projektiranja

Summary

This final thesis presents the framework project of development of a database inside of the information subsystem of the safety and protection at work. For that purpose, in the first step we analyzed principles, methods and techniques of database design. In detail is shown an overview of the basic functions of Microsoft Access 2010, which was used in the realization of the practical task. The practical part of the task included the design and development of a database and a simple application for record-keeping and distribution of the personal protective equipment. Special attention was paid to development of the user interfaces for input and review of data, as well as to creating of the necessary reports. The ways in which it is possible to functionally integrate the new database into the existing wider information system, were also analyzed.

Keywords: databases, safety and protection at work, protective equipment, MS Access 2010, information systems, methods and procedures of design

ZAHVALA

Zahvaljujem se mentoru dr.sc. Damiru Kralju na ukazanom povjerenju, trudu i pomoći tijekom izrade ovog Završnog rada.

Zahvaljujem se i kolegi Zoranu Petliću dipl.ing. na nesebičnoj pomoći oko tehničkih detalja završnog rada.

Najveća zahvala ide mojoj obitelji ponajviše supruzi Lidiji i sinu Davidu na pruženoj podršci, razumijevanju i strpljenju tijekom mog školovanja.

Veliko HVALA svima!

Sadržaj

1. Uvod 1
2. Metodologija 2
   2.1. Teorijska analiza 2
      2.1.1. Zakonska regulativa 2
      2.1.2. Metode i postupci u analizi poslovnih sustava 5
      2.1.3. Metode i postupci u projektiranju informacijskih sustava 11
   2.2. Praktična realizacija projekta 20
3. Baza podataka informacijskog sustava 22
   3.1. Relacijski model baze podataka 25
   3.2. Strukturirani upitni jezik – SQL 27
   3.3. Pojedini tržišno dostupni programski alati i podrške 28
   3.4. Združena računalna tehnologija 30
   3.5. Otvorena spojivost podatkovnih baza 30
   3.6. SQL Server 30
4. Odabir sredstva za ostvarenje projekta 31
5. Microsoft Access 2010 33
   5.1. Tablice 33
   5.2. Upiti 34
   5.3. Obrazac 35
   5.4. Izvještaji 36
   5.5. Makronaredbe i kod 36
   5.6. Upravljanje sigurnošću baze podataka 37
      5.6.1. Konfiguracijski propusti 37
      5.6.2. Nedostatak zaštite korisničkih računa 37
      5.6.3. Kontrola pristupa tablicama 37
      5.6.4. Neprikladne metode kontroliranja rada baze podataka 37
   5.7. Mogućnosti zaštite baze podataka 38
      5.7.1. Sustav dozvoljenih IP adresa 38
      5.7.2. Programski propusti unutar SUBP-a 38
      5.7.3. Korištenje enkripcije 38

6. Izrada podsustava za raspodjelu zaštitnih sredstava 40
6.1. Primarni ključ 43
6.2. Veze baze podataka 44
6.2.1. Vrste veza 44
6.2.2. Međusobno povezivanje 45
6.2.3. Referencijalni integritet - integritet povezanosti 47
6.3. Rad s tablicama 48
6.3.1. Izrada tablice kroz razvojnu strukturu podatkovnog lista 48
6.3.2. Izrada tablice kroz razvojnu strukturu dizajna podataka 51
6.3.3. Uvoz tablice iz vanjskih izvora 52
6.3.4. Kodeksa atributa (definicija objekta) 53
6.4. Obrasci 55
6.4.1. Izrada obrasca 55
6.4.2. Unos podataka u obrasce 56
6.5. Upiti 58
6.5.1. Oblici upita 59
6.5.2. Izrada upita 60
6.5.3. Pokretanje upita 62
6.6. Izvješća 62
6.6.1. Izrada izvještaja 63
6.6.2. Kontrole 65
6.6.3. Izvoz rezultata upita - izvještaj 65
7. Analiza primjenjivosti 68
8. Zaključak 70
9. Literatura 72
10. Prilozi 74
11. Popis kratica 90
12. Popis slika 92
13. Popis tablica 94

1. Uvod

Razvojem ljudskog društva i segmenta njegovih djelatnosti koji se odnosi na provedbu Zaštite i sigurnosti pri obavljanju poslova, razvija se i potreba za time da što veća količina potrebnih podataka bude što dostupnija svim ljudima kojima je zaista potrebna. Budući da su informacijski sustavi preuzeli uloge nekadašnjih arhiva i ostalih “papirnatih“ oblika vođenja evidencija, došlo je vrijeme da i u ovom segmentu ukupnih ljudskih djelatnosti ti sustavi općenito postaju sve važniji.

Potreba za učinkovitim praćenjem raznih podataka kao što su podaci o zaposlenicima, razmjena podataka o raznim projektima, radno vrijeme zaposlenih, stanje zaštitne opreme pojedinca, stanje opreme tvrtke, uvjetuje korištenje za te poslove posebno prilagođenog informacijskog sustava, koji na kraju sve te oblike spaja u jedinstvenu bazu podataka kako bi vodstvo neke tvrtke, institucije ili organizacije moglo donositi pravovremene strateške poslovne odluke. Njihova fleksibilnost u oblikovanju poslovnih funkcija, količina podataka koja može biti u njima pohranjena te obrada tih podataka kroz pružanje različitih oblika izvještaja i dalje će biti jedna od najvažnijih njihovih značajki.

Kako je autor ovog rada svakodnevno u kontaktu s različitim oblicima primjene baza podataka za temu ovog rada izabran je prikaz izrade jednog, naizgled lako izvedivog, oblika baze podataka od samog razvojnog početka pa do završnog praktično primjenjivog oblika. Početni parametri projektnog zadatka su poznate činjenice: tema projekta je evidencija raspodjele zaštitnih sredstava po korisnicima, broj zaposlenika koji se u tom trenutku nalazi u pojedinoj organizacijskoj jedinici, te činjenica da je predmetna oprema namijenjena upotrebi od više zaposlenika kako je navedeno u Članku 9. Pravilnika o uporabi osobnih zaštitnih sredstava (NN 39/06) kao polaznog akta na kojem se zasniva planirani informacijski podsustav.

2. Metodologija

Kako se ovaj rad sastoji od teorijskog djela i od praktičnog rada koji uključuje izradu aplikacije za vođenje evidencije zaštitnih sredstava, što predstavlja praktičnu primjenu do sada stečenih znanja od strane autora, tako se i oslanja na stečenu informatičku praksu prilikom pripreme i izrade same aplikacije.

Prikupljeni podaci i znanja zasnovani su na osobnim bilješkama i priloženim materijalima u okviru kolegija "Upravljanje zaštitom na radu primjenom računala", na informatičkim tečajevima koji obuhvaćaju ovu tematiku, literaturi vezanoj za temu uporabe i razvoja baza podataka velikih poslovnih sustava, višegodišnjem iskustvu u radu s bazama podataka i vođenjem evidencija, te manjim dijelom i na sadržajima raznih internetskih stranica.

Cilj je osvrnuti se i na pojedine zakonske osnove koje možemo povezati s izradom baze podataka koje kasnije mogu evoluirati u sustave velikih razmjera, te također pojasniti osnovne pojmove za izradu aplikacije baze podataka zajedno s pojašnjenjima vezanim uz izradu i korištenje takvih aplikacija kao jednog informacijskog podsustava iz kojeg može nastati samostalni sustav ili isti može biti pripojen nekoj drugoj bazi podataka, odnosno informacijskom sustavu značajne veličine.

U radu su korištene sustavne i komparativne metode te analiza sadržaja raznih projekata razvoja baza podataka za potrebe vođenja različitih evidencija.

2.1. Teorijska analiza

2.1.1. Zakonska regulativa

"Život, zdravlje i očuvanje radne sposobnosti vrednote su od posebnog društvenog interesa u Republici Hrvatskoj." [1]. Ovo je prva rečenica Članka 5., Zakona o zaštiti na radu (N.N., br. 59/96., 94/96., 114/03. i 86/08., 116/08., 75/09. i 143/12.), iz koje se lako može zaključiti što bi trebala biti Nacionalna politika i aktivnosti.

U skladu sa Zakonom o zaštiti na radu, svaka tvrtka, mora urediti obavljanje poslova zaštite na radu. U navedenom Zakonu, u Članku. 11. Opća načela prevencije, među ostalim stoji:

- *(stavak 5) – prilagođavanja tehničkom napretku;*

- *(stavak 7) – razvoja dosljedne sveobuhvatne politike prevencije povezivanjem tehnologije, organizacije rada, uvjeta rada, ljudskih odnosa i utjecaja radnog okoliša;*

U čl. 17. Opće obveze poslodavca u vezi s organiziranjem i provođenjem zaštite na radu stoji:

- *(stavak 5) – U svrhu unaprjeđivanja sigurnosti i zaštite zdravlja radnika poslodavac je obvezan poboljšavati razinu zaštite na radu i usklađivati radne postupke s promjenama i napretkom u području tehnike, zdravstvene zaštite, egronomije i drugih znanstvenih i stručnih područja [...]* [1],

Ovo su tvrdnje kojima možemo opisati iniciranje teme za izradu ovog rada kao i poveznicama za veliki broj segmenata od kojih se sastoji ovaj rad.

Ako navedene tvrdnje povežemo s tvrdnjama, odnosno smjernicamaiz Članka. 61., Vođenje i čuvanje evidencija i isprava te davanje podataka i obavjesti:

- *(stavak 1) Poslodavac je obvezan voditi evidencije, čuvati isprave te davati obavjesti i podatke u skladu s ovim Zakonom i drugim propisima o zaštiti na radu*,

zaključak se sam nameće, a to je potreba i neophodnost izrade evidencije zaštitnih sredstava u elektroničkom obliku.

Zavisno o broju djelatnika i procjeni opasnosti za njihova radna mjesta, poslodavac može poslove zaštite na radu obavljati sam, ukoliko zapošljava stručno osposobljen kadar, ili ih može povjeriti ovlaštenoj pravnoj osobi. [1]

Vođenje poslova zaštite na radu obuhvaća slijedeće aktivnosti:

- stručni nadzor,
- stručna pomoć iz područja zaštite na radu,
- stručno vođenje evidencija iz područja zaštite na radu,
- neposredno pratiti sve poslove iz područja zaštite na radu,
- stručna pomoć poslodavcu, zaposlenicima i njihovim povjerenicima u provedbi i unapređenju zaštite na radu,
- unutarnji nadzor nad primjenom pravila zaštite na radu,
- brigu o svim rokovima, vezanim za ispitivanja sredstava rada i instalacija, osposobljenosti zaposlenika za zaštitu na radu i poticanje poslodavca da nadležne službe otklanjaju eventualno utvrđene nedostatke iz područja zaštite na radu, praćenje i raščlamba podataka u svezi sa ozljedama na radu i

profesionalnim bolestima, te izrada godišnjeg izvješća za potrebe poslodavca, suradnja sa tijelima nadležnih inspekcija rada,

- suradnju sa stručnim službama poslodavca prilikom izgradnje i rekonstrukcije objekata namjenjenih za rad, nabavu strojeva i uređaja, te osobnih zaštitnih sredstava,
- suradnju sa specijalistima medicine rada,
- na temelju praćenja stanja zaštite, predlaže poslodavcu mjere za poboljšanje stanja,
- briga o posjedovanju potrebne dokumentacije u skladu sa Zakonom o zaštiti na radu,
- pravovremeno izvještavanje poslodavca o svim relevantnim faktorima zaštite na radu te predlaganje poslodavcu rješenja obaveza temeljem odredbi Zakona.[1]

Treba istaknuti da uz velik broj normativnih akata koji opisuju pojedina područja zaštite i sigurnosti gdje za primjer možemo navesti npr.:

- zaštita na radu – 34 propisa,
- medicina rada – 31 propis,
- radni propisi – 10 propisa,
- zaštita od požara – 26 propisa,
- tehnički propisi – 8 propisa,
- kemikalije – 16 propisa,
- buka i vibracije – 8 propisa

još uvijek postoje područja koja nisu obrađena ovim propisima, tako da možemo očekivati i povećanje ukupnog broja propisa [2].

Prilikom izrade ovog rada osvrnuli smo se i na Pravilnik o uporabi osobnih zaštitnih sredstava (NN 39/06), kojim je prestao važiti dotadašnji važeći propis Pravilnik o sredstvima osobne zaštite na radu i osobnoj zaštitnoj opremi (»Službeni list SFRJ«, br. 35/69), gdje u pravilniku iz 2006. godine je navedeno kako slijedi:

"Članak 8.

1) Poslodavac pri dodjeljivanju osobnih zaštitnih sredstava radnicima poštuje načelo da su sredstva namijenjena za njihovu osobnu uporabu.

2) Ukoliko okolnosti zahtijevaju da određeno osobno zaštitno sredstvo koristi više radnika, poslodavac mora poduzeti sve što je potrebno da takva uporaba kod korisnika ne uzrokuje zdravstvene ili higijenske teškoće.

Članak 9.

1) Poslodavac mora na vlastiti trošak radnicima osigurati osobna zaštitna sredstva.
2) Poslodavac mora osigurati ispravna osobna zaštitna sredstva, odgovarajuće higijenske uvjete, potrebno održavanje te popravke i zamjenu osobnih zaštitnih sredstava.
3) Poslodavac mora osigurati da su pri radu pri kojem se upotrebljavaju osobna zaštitna sredstva na raspolaganju tehničke upute te upute za njihovu uporabu." [3]

2.1.2. Metode i postupci u analizi poslovnih sustava

U današnje vrijeme sama organizacija poslovanja sve više odstupa od nekadašnjih vertikalnih (hijerarhijskih) organizacijskih pristupa gdje su postojale samo dvije podjele - nadređeni i podređeni, te se razvija prema nekim „modernijim konceptima“ koji zastupaju vertikalni pristup organizaciji poslovanja, gdje je naglasak na samom procesu pružanja usluge ili izrade proizvoda u kojem sudjeluju svi zaposlenici tvrtke, svaki na svoj način, odnosno u segmentima svog radnog područja.

Kako bi proveli takav koncept u djelo potrebno je dati ovlasti djelatnicima tako da im omogućimo da koriste razne alate u svom radnom procesu kako bi razvijali svoje vještine i samim time bili motivirani za preuzimanje potpune odgovornosti za poslovne procese. Kada zaposlenici postanu "partneri“ u izvršavanju procesa svaki neuspjeh tvrtke smatrat će i svojim neuspjehom, a to bi im trebala biti dodatna motivacija za što kvalitetniji rad, uspješnost i odanost "svojoj“ tvrtki. Uloga informacijskih sustava unutar tvrtke bi trebala omogućiti jednostavnu, brzu i laku prilagodbu poslovnim procesima, dok je njihova uporaba jedan od alata koji zaposlenicima olakšava ostvarivanje ciljeva organizacije.

Informacijski sustav, sa stajališta funkcije u organizacijskom sustavu, ima zadaću permanentno opskrbljivati potrebnim podacima poslovne procese i sve razine upravljanja tim procesima te odlučivanja u tim procesima.
Temeljne funkcije informacijskog sustava su:

- prikupljanje i upis podataka u bazu podataka,

- obrada (procesiranje) podataka,
- prikaz i ispostavljanje (diseminacija) podataka iz baze podataka,
- čuvanje (dokumentiranje, trajno pohranjivanje) podataka. [4]

Kako bi poslovni procesi postali i ostali uspješni sve više ovise o novim tehnologijama i automatizaciji, čime se potvrđuje nužnost povezivanja različitih elemenata poslovnih procesa kako informacijskih tako i organizacijskih. Sama faza razvoja sustava se odnosi na razvoj programskih rješenja za potporu, praćenje, izvođenje i analiziranje poslovnih procesa koji mogu biti podijeljeni na nekoliko segmenata programskih rješenja:

- za povezivanje,
- za komunikaciju i povezivanje,
- orijentirana na procese analize, automatizacije, upravljanje te modeliranje.

Kako programskih rješenja u informacijskom sustavu u ponudi ima više vrsta, odluka za koju ćemo se opredijeliti ovisi o grani posla te zadatku koji bi ista trebala obavljati. Tako postoje različiti sustavi uredskog poslovanja, sustavi za obradu podataka, transakcije, sustavi za potporu, upravljački sustavi, izvještajni sustavi te sustavi za pomoć pri odlučivanju s time da postoje i sustavi koji se sastoje od dva ili više podsustava koji su međusobno povezani. Nakon odabira sustava, a prije njegova puštanja u “pogon“, potrebno je obaviti i nekoliko obveznih koraka od strane ljudi koje su zadužene za implementaciju istoga u sustav poslovnog subjekta kojem je namijenjen kao što su testiranja i sama implementacija.

Prije no što dođemo do faze testiranja u sustav je potrebno unijeti podatke koji će biti jedinstveni i ne dvosmisleni za svaki segment podatka na koji se odnose. Tu se od strane poslovnog subjekta treba dodijeliti čovjek ili cijeli tim ljudi koje znaju informatičkim stručnjacima opisati sve zadatke pojedinog poslovnog procesa poslovnog subjekta (nalogodavca) kako bi im bilo lakše izraditi tokove podataka potrebne za ispravan rad sustava.

Nakon provođenja testiranja izrađenog informacijskog sustava, pogreške koje su nastale tijekom rada je potrebno ispraviti ili ukloniti, nakon čega se provodi ponovno testiranje i ispravljanje pogrešaka u istom dok sustav ne započne s ispravnim radom bez pogrešaka. Značajnost informacijskog sustava je u tome što sadrži znanje i spoznaje (*engl. Know – how*) procesa za koje je njegova izrada namijenjena. Proces

faze implementacije osim provedbe testiranja zahtjeva obuku korisnika i njihovu prilagodbu na nova programska rješenja.

Uvođenje gotove aplikacije često je puno teže nego se to očekuje. Kod uvođenja se pojavljuju svi problemi koji su bili skriveni jer ih testiranje nije uspjelo otkriti. Uklanjanje tih nedostataka uvjet je za uvođenje. Pored objektivnih postoje i subjektivne poteškoće uvođenja, a najvažnija je otpor promjenama koje pružaju krajnji korisnici. [4]

Kako bismo izradili aplikaciju na zadovoljavajući način, čime bismo zadovoljili zahtjeve korisnika i nama olakšali samu izradu, potrebno je osmisliti i izraditi projekt.

Projekt je dobro definiran slijed događaja sa svojim početkom i krajem, koji se izvode radi postizanja nekog jasnog cilja. Projekt podrazumijeva izvođenje složenog skupa poslova. Projekt je određeni radni zadatak koji odlikuju jasni ciljevi, faze, ljudski, vremenski, materijalni i drugi resursi kao i organizacija njegova proizvođenja.[1]

Također je potrebno oformiti tim ljudi koji će biti specijalisti u svojem djelokrugu rada s kojima i za koje se obavlja izrada projekta odnosno aplikacije. Isti čovjek u timu može obavljati više uloga, ovisno o njezinom predznanju, iskustvu, specijalnosti itd. Funkcije radnika u timu sadržane su Tablici 1.

[1] Definicija projekta prema normi DIN 69901, DIN je kratica njemačkog standarda (DIN-Norm) i istoimenog instituta *Deutsches Institut für Normung* koji propisuje standarde u Njemačkoj.

Tablica 1. – Funkcije radnika u timu [4]

Naziv posla	Opis posla
Voditelj projekta	Menadžer koji može raspolagati resursima, obično iz redova korisnika
Savjetnik	Obično vanjski eksperti koji komuniciraju s neprofesionalnim korisnicima u svrhu utvrđivanja njihovih potreba
Analitičar poslovnog sustava	Poznavatelj metoda za analizu poslovanja koji izrađuju model procesa sustava
Projektant informacijskog sustava	Zajedno s analitičarima analiziraju postojeći način rada sustava te dizajniraju novi poboljšani način rada metodama za modeliranje budućeg informacijskog sustava
Dizajner softvera	Specijalist za oblikovanje softverskih alata
Dizajner weba	Specijalist za oblikovanje web-aplikacija
Tehničko osoblje	Specijalisti za održavanje hardvera i mreže
Dizajner baze podataka	Specijalist za modeliranje podataka
Administrator baze podataka	Specijalist zadužen za efikasno funkcioniranje baze podataka
Računalni programer	Specijalist za kodiranje kompjuterskih programa
CASE developer	Specijalist za korištenje CASE alata
Ispitivač	Specijalist zadužen za testiranje ispravnosti programskog proizvoda
Operater	Obučeni korisnik koji je preuzeo izvršenje obrade podataka
Korisnik za unos podataka	Upisuje podatke u bazu podataka
Korisnik za testiranje	Izabrani korisnik kojem se povjerava testiranje
Menadžer poslovne funkcije	Voditelj poslovnih područja za koje se gradi informacijski sustav

2.1.2.1. Modeliranje poslovnih procesa

Prilikom izgradnje informacijskog sustava neophodno je cjelovito poznavanje poslovnih procesa organizacije u koju će se sustav implementirati. Jedno od ponajboljih sredstava pokazale su se grafičke metode od kojih bi trebali razlikovati statičke od dinamičkih metoda modeliranja poslovnih procesa, većinu njih podržavaju programski alati koji omogućuju provedbu analize iz tako prikazanog procesa. Model procesa je strukturirani skup procesa koji mijenja stanje sustava i procesa pomoću kojih se formiraju izlazi iz sustava. Realiziran na računalu, dio modela procesa postaje skup programa za ažuriranje. Procesi za generiranje informacija iz stvarnog sustava opisuju se modelom procesa za generiranje izlaza, a programi za izvješćivanje fizička su realizacija tog modela.

Metodologija računalno potpomognutog sustavnog inženjerstva ili koje se još naziva CASE metodologija (*engl. Computer-Aided Systems Engineering*) izrađena od korporacije ORACLE, na području modeliranja procesa primjenjuje sljedeće metode:

- hijerarhijski prikaz poslovnih funkcija,
- prikaz ovisnosti između funkcija,
- povezivanje funkcija s organizacijskim jedinicama, mjestom izvođenja procesa, ulogom funkcija (koliko često, koliko brzo, koliko puno), modelom podataka,
- definicija detaljne logike funkcija,
- modeliranje procesa i crtanje dijagrama toka podataka,
- hijerarhijski prikaz procesa,
- modeliranje u stvarnom vremenu,
- strukturna karta,
- prevođenje dijagrama toka podataka u strukturnu kartu. [4]

Podatkovni modeli imaju težište na entitetima kao što su podaci ili dokumenti te na njihovoj povezanosti i strukturi. Funkcijski modeli usredotočeni su na poslove koje treba obaviti. Organizacijski modeli govore nam mjesto, vrijeme i komunikacije između aktera u procesu dok procesni model nam govori o onome što pokreće, kada, kako i kada se odrađuju aktivnosti. Pojedini oblici zajedno s pripadajućim metodama prikazani su u Tablici 2.

Tablica 2. Oblici modeliranja i pripadajuće metode [5]

PRISTUP	TEŽIŠTE	METODA
Podatkovni	- entiteti (dokument, podaci), njihova struktura i povezanost	- Dijagram toka podataka (DFD), - Dijagram entiteta -veza (ER)
Funkcijski (što, koji)	- ono što treba napraviti (koje aktivnosti) - aktivnost (poslovi) koje se izvode - podaci koji ulaze u aktivnosti i koji izlaze iz njih	- SADT dijagram, - IDEFo dijagram
Organizacijski (gdje, tko)	- mjesto izvođenja aktivnosti - tko izvodi aktivnost - komunikacijski mehanizmi	- IDEF dijagram, - UML dijagram korištenja, - UML dijagram suradnje
Procesni (zašto, kako, kada)	- ono što pokreće aktivnosti i zašto se aktivnosti izvode na odgovarajući način - slijed izvođenja aktivnosti: niz, ponavljanje, uvjeti - kada se izvode aktivnosti, - koliko traju aktivnosti	- eEPC dijagram, - UML dijagram aktivnosti, - Petrijeve mreže, - BPMN dijagram

2.1.2.2. Transakcije i podudarnosti

Većina aktualnih baza podataka pokušava provesti model transakcije baze podataka koji ima poželjna svojstva za integraciju podataka. Programska potpora baze podataka bi prema zamisli trebala provoditi ACID pravila koja označavaju Valentnost, Dosljednost, izolaciju, Postojanost (*engl. Atomicity Consistency Isolation Durability*) [6] :

- Valentnost – ili sve zadaće moraju biti obavljene ili nijedna. transakcija mora biti dovršena ili se mora poništiti (vratiti natrag);
- Dosljednost – svaka transakcija mora sačuvati integritetnu prinudnost – izričita pravila dosljednosti baze podataka. Ona ne može smjestiti podatke u kontradiktornom položaju;

- Izolacija – dvije simultane transakcije ne mogu interferirati, tj. križati se. Međurezultati unutar transakcije nisu vidljivi drugim transakcijama;
- Postojanost – dovršene transakcije se ne mogu kasnije prekinuti ili se njihovi rezultati odbaciti. One se moraju nastaviti kroz ponovno pokretanje sustava upravljanja bazom podataka (SUBP) nakon njegovog kraha.

U praksi mnogi SUBP *(engl. DataBase Management System, DBMS)* dopuštaju većini ovih pravila da se selektivno ublaže radi boljih performansi. SUBP mora osigurati da su dopušteni linearni, nadopunjujući redoslijedi, te da ni jedna radnja počinjenih transakcija nije izgubljena prilikom poništenja prekinutih transakcija.

2.1.3. Metode i postupci u projektiranju informacijskih sustava

Informacijski sustavi predstavljaju skup podataka, metoda, tehničkih sredstava i organizacije za pružanje informacija, kroz komunikaciju, prikupljanje, arhiviranje i obradu, podataka potrebnih za donošenje poslovnih odluka a sve u cilju boljeg funkcioniranja cjelokupnog organizacijskog sustava. Informacijski sustav se sastoji od programa koji prikupljaju informacije o stanju sustava te ih upisuju u bazu podataka, programa koji čitaju podatke unesene u bazu podataka te oblikuju izvješća sukladno korisnikovim zahtjevima.

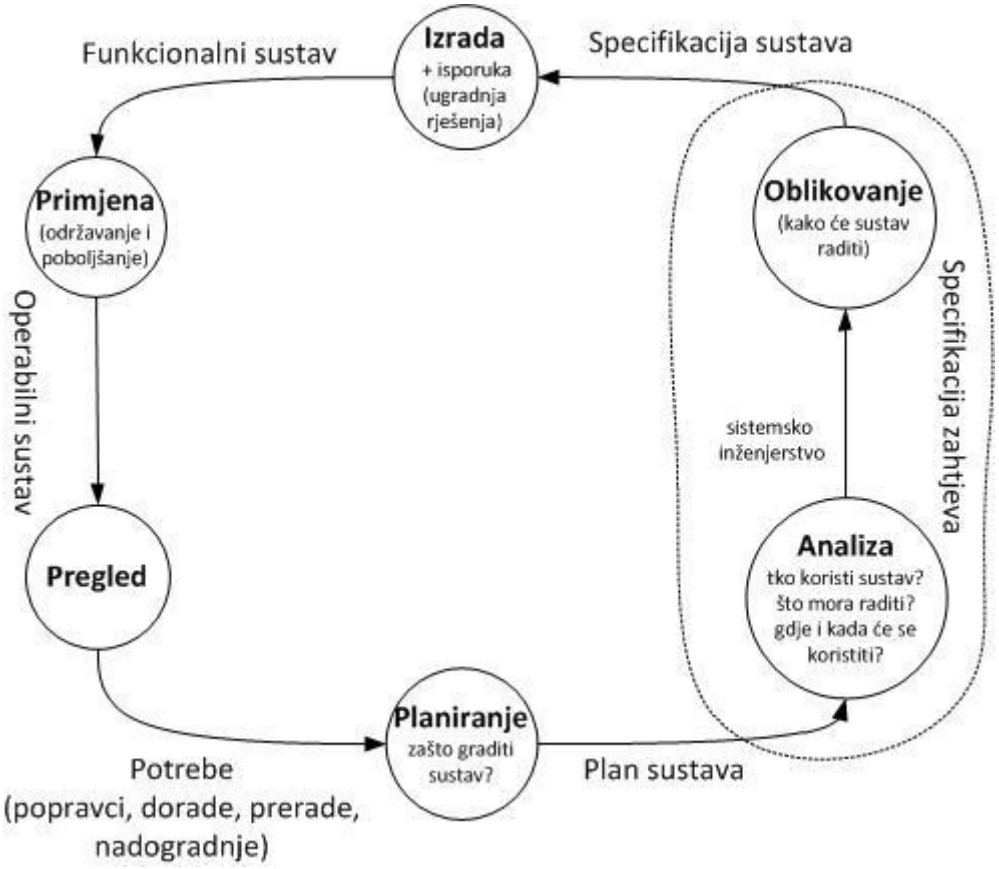
Komponente Informacijskog sustava su:

- programska potpora (*engl. Software*) – namjenska, sustavska,
- računalna (strojna) potpora (*engl. Hardware*) – računala, mrežna oprema,
- ljudski čimbenici (*engl. Lifeware*) – informatičari, korisnici,
- organizacijska potpora (*engl. Orgware*) – postupci povezivanja dijelova u cjelinu.

S gledišta organizacijskih funkcija ista se može sistematizirati prema potrebama istih u različite informacijske podsustave koji ima jedinstvene informacijske potrebe što ujedno znači da je potrebno dizajnirati sustave koji bi odgovarali njihovim potrebama, kao što su: ljudski resursi, nabava, prodaja, marketing, financije, računovodstvo itd.. Međutim važno je naglasiti kako je neophodno da pojedini dijelovi ili funkcije informacijskog sustava poslovnog subjekta budu povezani u jedan cjeloviti, integrirani sustav.

2.1.3.1. Projektiranje informacijskog sustava

Životni ciklus informacijskog sustava definira faze i zadatke koje nužno treba obaviti tijekom razvoja neovisno o veličini sustava koji se izrađuje. Taj ciklus osiguravaju "kontrolne točke", kako je prikazano na Slici 1., uz pomoć kojih pratimo napredak, procjenjujemo postignute rezultate i donosimo odluke o daljnjim koracima.



Slika 1. – Životni ciklus informacijskog sustava

Prikazane kontrolne točke nam pomažu u određivanju sljedećeg:

- Planiranje je vezano uz određivanje poslovnih ciljeva, otkrivanje problema, pronalaženje rješenja i određivanje zahtjeva vezanih za sustav.
- Analizom zahtjeva se određuju granice projekta, poslovnih zahtjeva i modela sustava u razvoju.
- Oblikovanje podrazumijeva modeliranje i dizajniranje sustava te donošenje odluka o načinu gradnje i funkcioniranja sustava.
- Izrada je provedba ugradnje oblikovanih rješenja u nekoliko segmenata kao što su: kodiranje, ugradnja baze podataka, testiranje.

- Integracijom se udružuju dijelovi sustava i provjerava cjelina kako bi se dokazalo da sustav uistinu odrađuje ono što je korisnik zahtijevao.
- Uvođenje u primjenu označava prijenos podataka iz starog u novi sustav zajedno s radnim aktivnostima odnosno procesima, nakon čega se provodi analiza o radu sustava te uvode se eventualna poboljšanja.
- Održavanje označava izmjene unutar sustava kako bi se zadržale ili poboljšale radne performanse (karakteristike) odnosno obavile prilagodbe ili poboljšanja samog načina uporabe.
- Pregled je posljednja faza razvojnog ciklusa a traje cijelo vrijeme od perioda uvođenja sustava a odnosi se na preispitivanje cijelog sustava kada je potrebno obaviti izmjene zbog promjena u poslovanju.

2.1.3.2. Semantičko modeliranje baze podataka

Semantičko modeliranje podataka započinje istovremeno s prikupljanjem podataka koji su potrebni za izgradnju baze podataka, jer dok prikupljamo podatke moramo znati koja je njihova uloga i značenje kako bi mogli odrediti njihovu ulogu u budućoj bazi podataka [7]. Svi podaci u bazi podataka podaci bi trebali biti međusobno povezani direktno ili indirektno. U poglavljima ovog rada, koja slijede, pojedini segmenti semantičkog modeliranja baze podataka će biti detaljnije obrađeni. U stvari, semantičko modeliranje predstavlja označavanje podataka (entiteta) kroz određeni model podataka. Što se tiče entiteta isti mogu biti samostalni, opisni ili sadržavati neka druga osobna svojstva koje možemo nazvati atributi odnosno karakteristike.

Odnosi između entitetima koji se kasnije reflektiraju na način rada baze i njene postupke definirani su kroz *agregaciju* (asocijacija), specijalizaciju (podtip) i generalizaciju (nadtip).

Kod *agregacije* (asocijacije ili pridruživanja) prikazani su odnosi između različitih oblika veza i njihovih osobnih atributa na način kao da je jedan podatak sastavni dio drugog podatka. Npr. odnos N:1 između *ZAPOSLENIK* i *TVRTKA* odnos je *ZAPOSLENJE* dok svojstva označavaju vrijednosti kao što je *DATUM, RADNO-PRAVNI STATUS, PRIMANJA, POLOŽAJ U TVRTCI*. Također je bitno za napomenuti kao u *agregaciji* entiteti bilo koje razine imaju svoje atribute koji ne moraju biti međusobno povezani niti biti isti npr. DATUM ZAPOSLENJA. Odnos

između generalizacije koja predstavlja entitete više razine (nadtip) i specijalizacije koja predstavlja entitete niže razine (podtip) možemo zamisliti kao odnos između naziva *ČOVJEK* i ostalih naziva kao što je *STUDENT*, *ZAPOSLENIK*, *UMIROVLJENIK* i td., možemo primijetiti da nazivom *ČOVJEK* možemo zahvatiti puno širi obim ljudi nego s ostalim oblicima naziva. Stoga zaključujemo kako je nadtip unija svih podtip-ova koji proširuju opis nekog entiteta nadtip-a dodatnim atributima.

Postoje razne metode za precizno definiranje podataka i njihovih veza. Prema CASE metodologiji na području modeliranja podataka primjenjuju se sljedeće metode:

- metoda entiteti – veze
- relacijska metoda
- entiteti – veze
- metoda mrežne organizacije podataka
- metoda hijerarhijske organizacije podataka.[4]

2.1.3.3. Logičko modeliranje baze podataka

Prilikom logičkog modeliranja koristi se, ranije izrađena, semantički modelirana baza podataka kao podloga za izradu logičkih elemenata baze podataka. Ti elementi baze podataka okarakterizirani su u obliku tablica, veza, međusobne zavisnosti i td. Što se tiče međusobne zavisnosti svojstvo jednog ili više atributa da jednostavno određuju vrijednost jednog ili više preostalih atributa naziva se funkcionalnom zavisnošću [7]. Skup atributa (Y) je funkcionalno zavisan o drugom setu atributa (X) te se takva zavisnost označava s X→Y.

Također zavisnosti mogu biti i višeznačne kada niti jedan set atributa ne ovisi jedan o drugome te ih označavamo s dvije uzastopne strelice X→→Y ili čak trivijalne višeznačne vrijednosti za koje vrijedi da je Y podskup od X ili da je unija X i Y jednaka skupu svih atributa u tablici.

2.1.3.4. Dijagram toka podataka

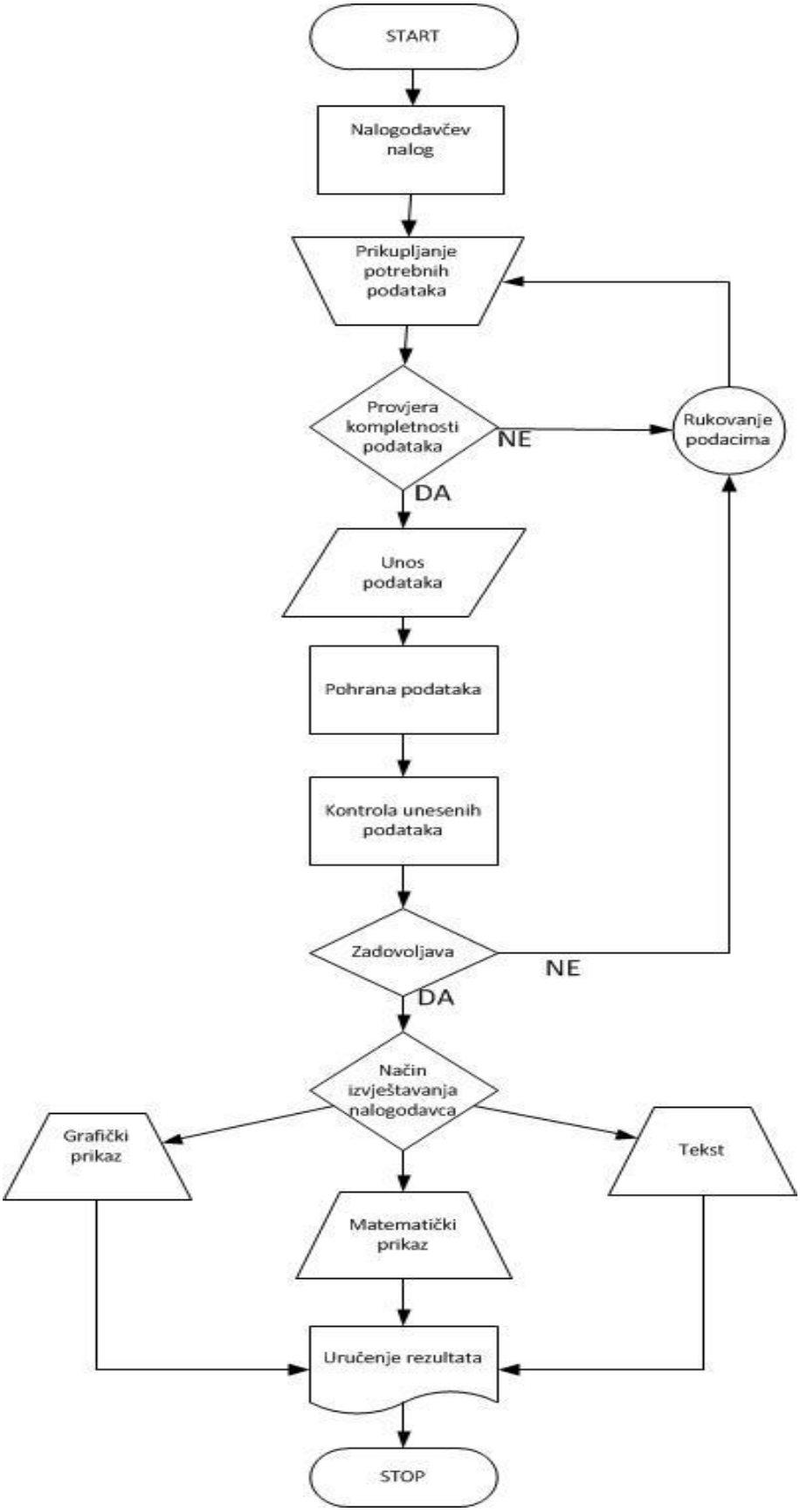
Dijagram toka podataka (DTP, *engl. Data Flow Diagram, DFD*) je statičan prikaz nekog procesa koji nam grafički prikazuje tokove podataka unutar sustava kako bi lakše mogli shvatiti odnosno razumjeti tko ili što je izvor ili izvorišta podatka, koji su

procesi koje nam oni opisuju te koja su odredišta na koja oni tim procesom djeluju ali njime ne možemo vidjeti i dinamiku samog procesa.

Postoji više inačica za grafički prikaz dijagrama toka podataka kao što su :

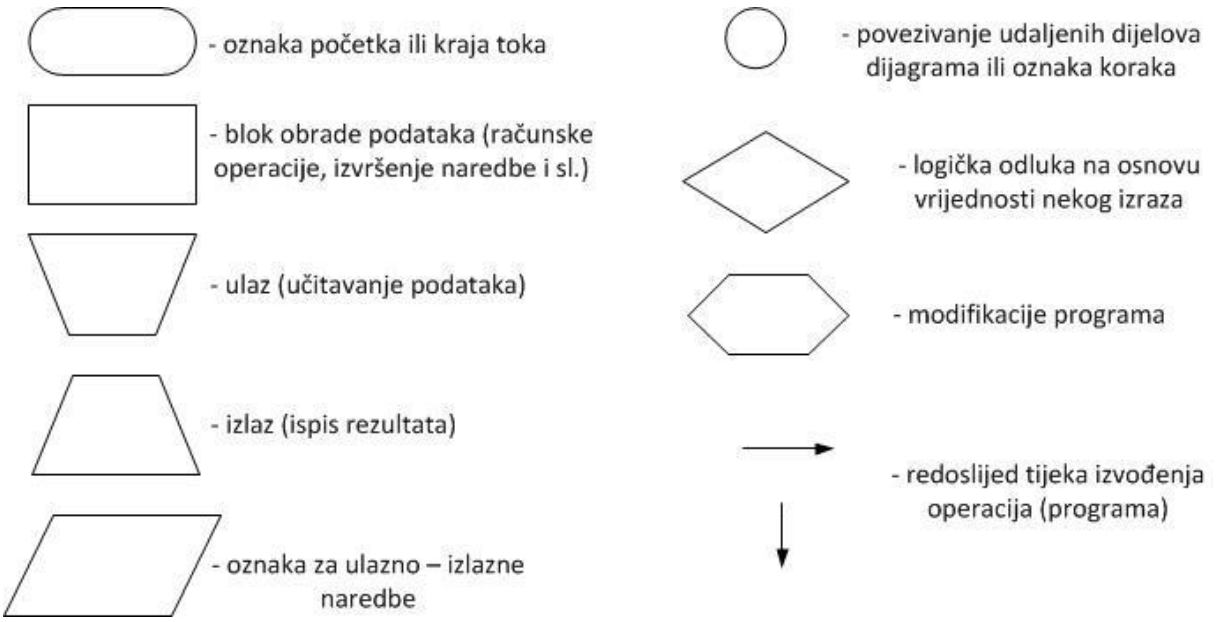
- Gane (Gane – Sarsonova metoda),
- DeMarco,
- Yourdon,
- ANSI standard, itd. [4, 5]

Svaki dijagram se sastoji od nekoliko osnovnih elemenata kao što su: izvorišta podataka, procesa, spremišta podataka te odredišta podataka. Sam tok podataka čini skup kretanja pojedinih podataka kroz sustav tako da skup ulaznih tokova podataka pretvara u skup izlaznih tokova podataka. Sukladno tome svaki proces mora imati najmanje jedan ulazni i jedan izlazni tok podataka. Kako bi pohranili ulazne i izlazne podatke nekog procesa moramo imati i spremište podataka. Dijagram toka podataka gotovo da možemo poistovjetiti sa dijagramom entiteta veza EV (*engl. Entity Relationship, ER*) jer se oni najčešće koriste pri razvoju informacijskih sustava te su ponajprije usmjereni na podatke a prikazuju tok poslovnih procesa, kao što je prikazano na Slici 2.



Slika 2. – Dijagram toka podataka

Prilikom izrade dijagrama toka podataka potrebno je pridržavati se općepriznatih simbola, ovisno prema čijem grafičkom obliku (tvorcu grafičkog oblika) radimo, dok smo mi pri izradi koristili znakove koji su prikazani na Slici 3.



Slika 3. – Neki od simbola dijagrama toka podataka prema ANSI standardu

2.1.3.5. UML dijagrami

UML (*engl. Unified Modeling Language*) je grafički jezik za vizualizaciju, specifikaciju, razvoj i dokumentiranje programskih rješenja pri čemu nije određena metodologija razvoja programa, već sam način prikaza. To je integrirani alat za modeliranje i razvoj objektno orijentiranih programskih aplikacija, koji se i dalje razvija i nadopunjuje. Prilikom korištenja UML dijagrama postoje i pet pravila kojih se je potrebno pridržavati:

1. Odlučite o čemu želite prvotno iskomunicirati pa prema tome usmjerite svoj opis. Ekstrahirajte detalje koji su vam potrebni a ostatak anulirajte. UML je poput svakog drugog jezika u koji možete zakomplicirati raznim detaljima te zbog njih na kraju izgubite svoj glavni cilj. Budite koncizni;
2. Svaki UML dijagram bi trebao biti veličine od jedne stranice. Dijagram ćete lakše razumjeti ako ga možete sagledati kao cjelinu istovremeno. To ne znači da morate ograničiti sebe, nego razdijeliti i organizirati sadržaj u razumnim, razumljivim dijelovima. Koristite pakete kako bi si organizirali preglednost. U koliko imate mnogo sjajnih ideja za prenijeti, počnite s dijagramom više razine

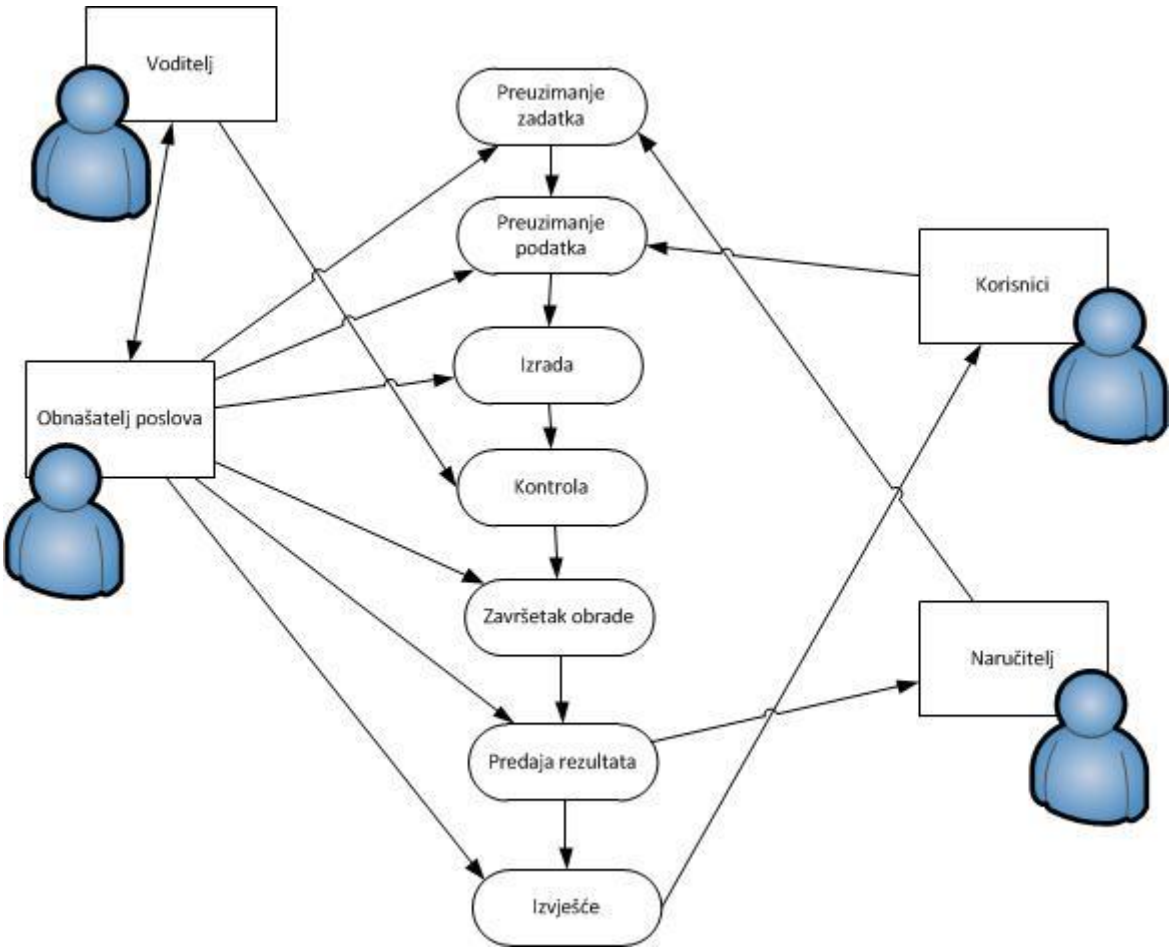
koji prikazuje “širu sliku“. Zatim započnite izrađivati dijagrame koji su posvećeni svakoj od vaših ideja;

3. UML uporabite kada vam je od koristi. Nemojte osjećati obvezu napisati UML dokument samo zato što mislite da bi ste ga trebali imati. UML nije sam sebi svrha, nego je odlično dizajniran alat za pomoć pri odgovarajućim problemima;
4. Popratite svoje dijagrame s tekstualnim opisima, kojima si pojašnjavajte vaše namjere. Osim toga, prisjetite se da su neki ljudi orijentirani verbalnom govoru dok su drugi pak vizualnom prikazu. Uspjehom se pokazalo kombinirati prirodni jezik s UML dijagramom;
5. Pazite da jasno organizirate svaki dijagram. Izbjegavajte unakrsna povezivanja. Grupirajte elemente međusobno ako u vašim mislima postoji razlog za to. Dva UML dijagrama mogu sadržavati iste elemente i veze, dok jedan bi mogao biti u strašnom “vizualnom“ neredu dok je drugi savršeno pregledan. Oba će prenijeti isto značenje u UML, ali je jasno da će te preglednijom verzijom biti uspješniji pri komunikaciji o pitanjima dizajna. [8]

Trenutno razlikujemo trinaest vrsta različitih dijagrama koji su podijeljeni u tri skupine:

- Strukturni dijagrami;
    - Dijagrami klasa,
    - Dijagrami objekta,
    - Dijagrami komponenata,
    - Dijagrami složene strukture,
    - Paketni dijagram,
    - Dijagrami rasporeda,
- Dijagrami ponašanja;
    - Dijagrami korištenja,
    - Dijagrami aktivnosti,
    - Dijagrami stanja,
- Dijagrami međudjelovanja;
    - Dijagram komunikacije,
    - Dijagram slijeda,
    - Vremenski dijagram,
    - Dijagram pregleda međudjelovanja. [5]

Dijagram korištenja, prikazan na Slici 4., zorno nam prikazuje sve sudionike procesa i njihove akcije unutar aplikacije. Prikazuje se međusobna interakcija sudionika a ukoliko postoje dodatne akcije izrađuje se onoliko scenarija koliko je to moguće.



Slika 4. – Dijagram korištenja

2.1.3.6. Normalizacija podataka

Relacijske tablice baze podataka, bilo da su izrađene na osnovi EV ili UML dijagrama, ponekad mogu skrivati neku naizgled prilično bezazlenu anomaliju koja se u kasnijem radu može pokazati kao ozbiljna prijetnja ispravnom radu baze podataka, naročito u pogledu performansi, zadržavanja integriteta ili održavanja. Ako cijelu bazu podataka zamislimo kao veliku tablicu, to može rezultirati velikom količinom redundantnih podataka i dugotrajnih pretraživanja čak i za najmanju količinu

potrebnih podataka. To također može rezultirati do dugotrajnih i skupih ažuriranja i brisanja, što na kraju kao nuspojava može rezultirati nestankom korisnih podataka. [7, 8, 9]

U koliko imamo metodu razbijanja tako velike tablice na manje cjeline ovakva vrsta problema bi bila eliminirana a naša baza podataka bi bila znatno učinkovitija i pouzdanija. Normalizacija se postiže analizom međuovisnosti pojedinih atributa povezanih unutar tih tablica.

Postoji sedam normalnih formi (NF) u kojima se mogu nalazit tablice:

1NF – označava tablicu u kojoj svaki stupac ima samo jednu vrijednost za svaki red u tablici, dok te vrijednosti pojedinačno mogu sačinjavati vanjski ključ, zajedno mogu sačinjavati kompozitni primarni ključ;

2NF – isključivo označava tablicu koja zadovoljava kriterije 1NF te ne postoji trivijalna parcijalna zavisnost neključnog atributa od ključa;

3NF – tablica je u trećoj normalnoj formi ako i samo ako ne postoji tranzitivna zavisnost neključnog atributa od ključa;

BCNF – tablica je u Boyce-Coddovoj normalnoj formi (BCNF) ako za svaku netrivijalnu funkcijsku zavisnost vrijedi da lijeva strana zavisnosti sadrži ključ te tablice;

4NF – samo ako za svaku netrivijalnu višeznačnu zavisnost vrijedi da lijeva strana zavisnosti sadrži ključ te tabele, onda se tablica nalazi u toj formi, ukoliko tablica nema netrivijalnih višeznačnih zavisnosti ona je u 4NF;

5NF – samo ako svaka netrivijalna zavisnost spoja ima svojstvo da sve njene komponente sadrže ključ;

6NF – samo ako ne postoji netrivijalna zavisnost spoja.

Na osnovi stečenog znanja i iskustva, lako je zaključiti kako tablica koja ima primarni ključ, bio on jednostavan ili složen, i jednu funkcionalnu zavisnost kojom ključ implicira ostale atribute, zadovoljava petu normalnu formu, a time i sve niže normalne forme, neovisno o broju stupaca. [7]

2.2. Praktična realizacija projekta

Osnova praktičnog dijela ovog rada obuhvaća poglavlja u kojima se kao sredstvo koristi razvojni alat MS Access 2010 pri čemu se detaljno opisuje način korištenja alata, mogućnosti koje on pruža, odnosno mogućnosti koje su korištene prilikom

izrade aplikacije za raspodjelu sredstava. Dakle, rad se sastoji od praktičnog dijela koji uključuje izradu aplikacije za vođenje evidencije zaštitnih sredstava, što predstavlja praktičnu primjenu do sada stečenih znanja od strane autora i oslanja se na informatičku praksu prilikom pripreme, izrade i korištenja takvih aplikacija kao jednog segmenta koji može funkcionirati samostalno ili može biti sastavni dio nekog većeg sustava tj. povezan s nekom drugom bazom podataka u veći sustav.

Realizacija praktičnog dijela zadatka obuhvaća provođenje slijedećih aktivnosti rada:

- okvirno projektiranje informacijskog sustava, odnosno projektiranje baze podataka kao osnove sustava za vođenje evidencija zaštitnih sredstava,
- analiza postojećih rješenja i alata s kojim se osnovna zamisao može povezati,
- osnovno funkcionalno, podatkovno te aplikativno modeliranje konkretno primjenjivog rješenja,
- analiza daljnjeg razvoja sustava za vođenje evidencija i njegova analiza uklopivosti u standardne sustave.

Metode koje su korištene u oblikovanju ovog završnog rada su:

- komparativna metoda s ciljem uspoređivanja upravljačkih modela između poslovnih sustava i informacijskih sustava,
- logičke metode analize i sinteze te metode indukcije i dedukcije koristiti će se u obradi podataka i izvođenju zaključaka,
- metodom analize sadržaja normativnih akata odrediti će se osnovne značajke o zadacima, pravima i obvezama sudionika odnosno izvršioca,
- kroz osobna iskustva,
- metoda ispitivanja, u cilju prikupljanja podataka na temelju iskustava ispitanika gdje su se koristile tehnike pojedinačnog i grupnog intervjua sa ljudima i korisnicima koje se neposredno bave razvojem i implementacijom baza podataka.

3. Baza podataka informacijskog sustava

U bazi podataka su pohranjeni podaci od dijelu realnih podataka radi kojih je i razvijen informacijski sustav. Sadrži međusobno povezane podatke s ciljem da posluže u izradi različitih upitnih oblika zahtijevanih od strane korisnika. Svi podaci se pohranjuju u bazu koristeći odgovarajući model podataka, bez nepotrebne zalihosti koja bi dodatno opterećivala sustav, dok se podaci mogu mijenjati a postojeći premještati.

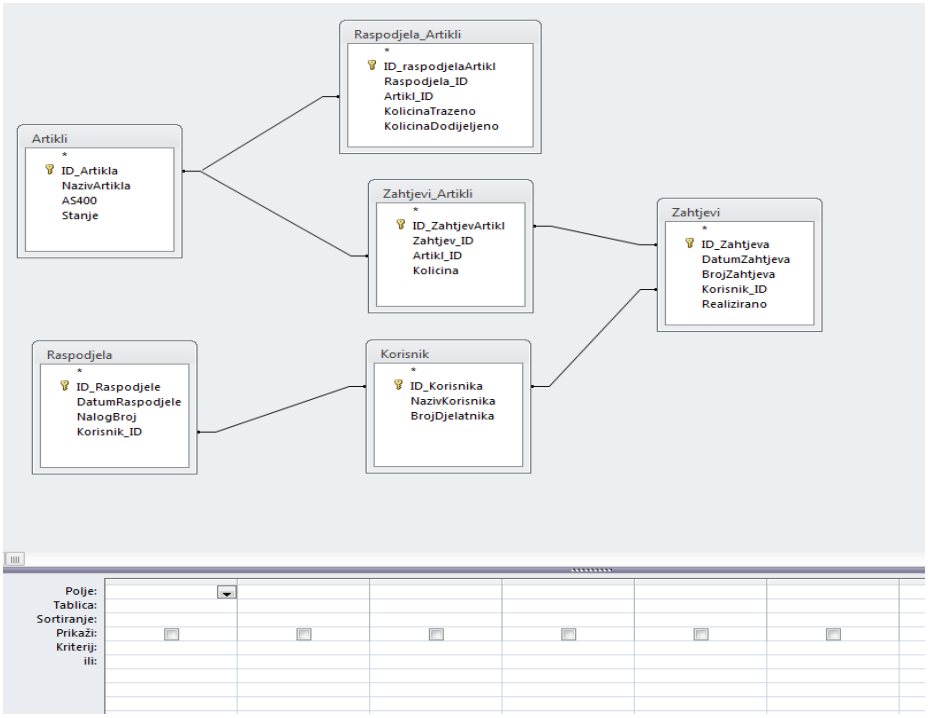
Arhitektura baze podataka zasnovana je na tri apstraktne razine i njihovim međusobnim sučeljima:

- logička lokalna razina se odnosi na logičku predodžbu o dijelu baze koji se koristi u pojedinoj aplikaciji;
- logička globalna razina se odnosi na logičku strukturu cijele baze čiji zapisi se nazivaju shemama te se imenuju i definiraju svi tipovi podataka kao i njihove veze u skladu s pravilima korištenog modela;
- fizička razina odnosi se na prikaz i raspored podataka na jedinicama vanjske memorije dok raspored pohranjivanja opisuje se kao elementi logičke definicije baze preslikavaju na fizičke uređaje.

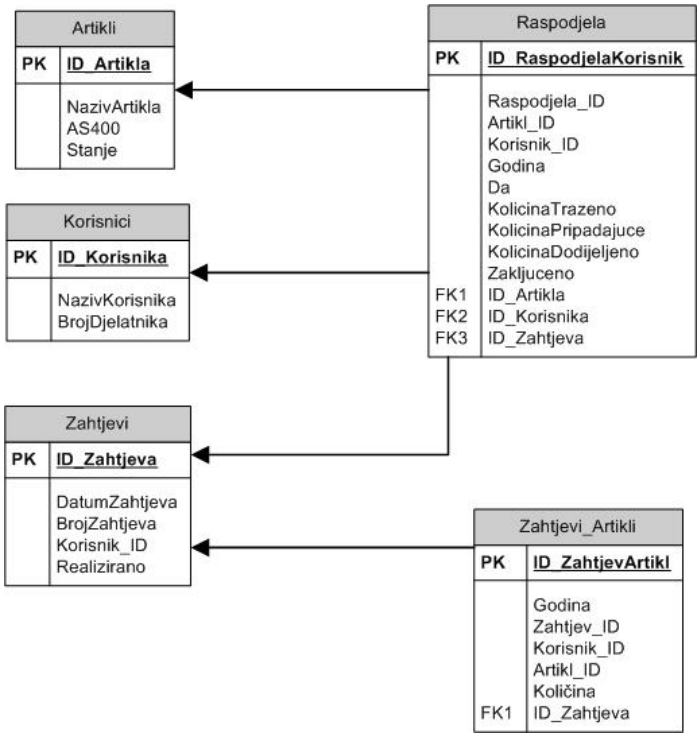
Kod stvaranja baze podataka potrebno je znati samo shemu i poglede a program SUBP tada automatski izrađuje fizičku bazu i raspored pohranjivanja podataka unutar nje bez da upisujemo ili poznajemo SQL (*engl. Structured Query Language*) naredbe. SUBP upravlja fizičkim smještajem podataka na optimizirani način kako bi čitanje i pisanje bilo što brže, procesuira upite za izvještavanjem ili izvršavanjem, upravlja fizičkom sigurnošću podataka kroz pohranu (*engl. Backup*), osigurava logičku sigurnost podataka (*engl. Security*), upravlja integritetom podataka koji moraju biti jedinstveni i potpuni, upravlja transakcijama koje su u biti skup operacija nad podacima koji se smatraju jednom cjelinom, upravlja konkurentnim pristupom kada bazi pristupa više korisnika u isto vrijeme te upravlja poslovnim pravilima. Informatičar ili administrator može samo jednim djelom utjecati na fizičku građu podešavajući one parametre koji su njemu dostupni. Tako niti korisnici niti programi ne pristupaju izravno fizičkom zapisu baze nego sve zadaće odrađuju putem SUBP.

Današnji SUBP podržavaju tri osnovna modela:

- relacijski model (Slika 5. i Slika 6.) – zasnovan na matematičkom pojmu relacije – podaci i veze između njih se prikazuju pomoću dvodimenzionalnih tablica koje mogu biti prikazani (izrađeni) u različitim oblicima

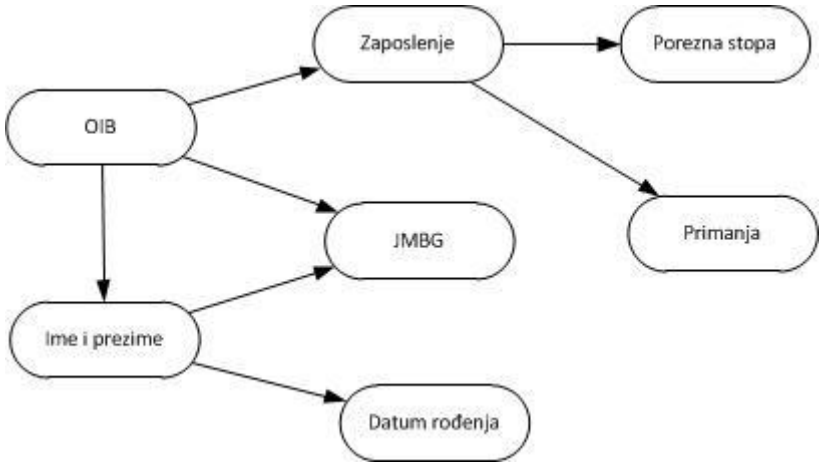


Slika 5. – Relacijski model baze podataka izrađen u MS Access-u 2010



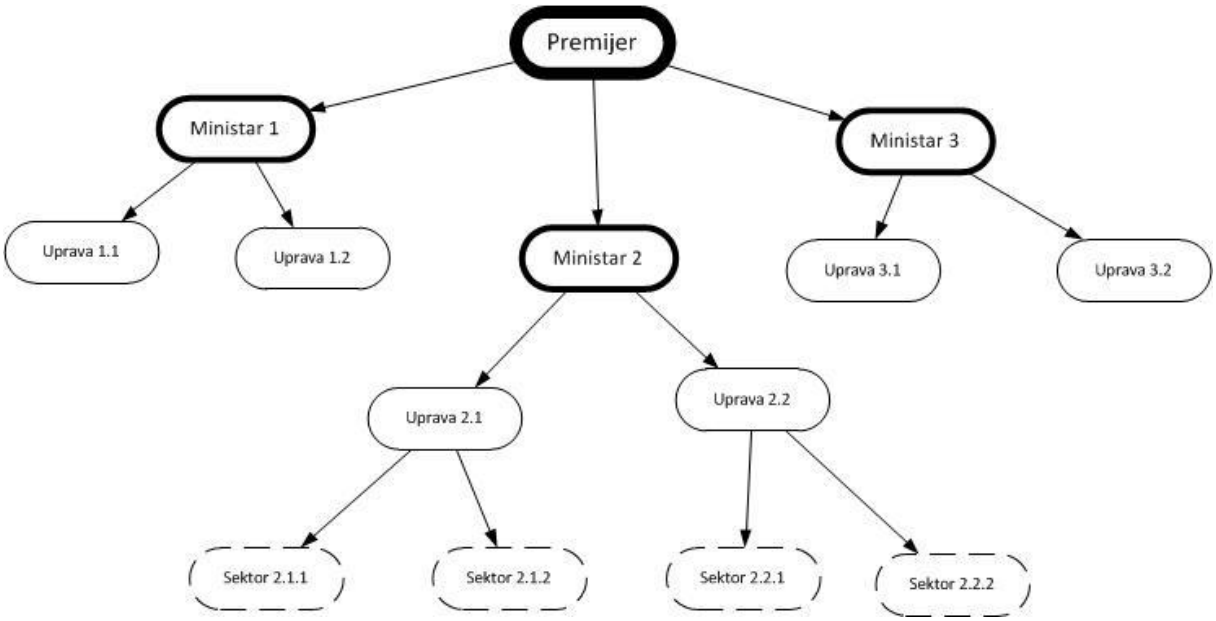
Slika 6. – Relacijski model baze podataka izrađen u MS Visio 2007

- mrežni model (Slika 7.) – može biti predočen putem mrežnog grafa u kojem su čvorovi podaci a lukovi među njima definiraju veze među podacima



Slika 7. – Mrežni model

- hijerarhijski model (Slika 8.) – sačinjavaju ga hijerarhijski organizirani podaci



Slika 8. – Hijerarhijski model

- objektni model – inspiriran je programskim jezicima koji su objektno orijentirani, ta baza je skup trajno pohranjenih objekata koji se sastoje od svojih “internih podataka i metoda“ za rukovanje s tim podacima. Svaki objekt

pripada nekoj klasi između kojih se uspostavljaju veze nasljeđivanja, generiranja, akumuliranja te međusobne uporabe različitih operacija.

Model procesa opisuje dinamiku podataka informacijskog sustava i propisuje skup procesa (funkcija) kojima se mijenjaju podaci unutar informacijskog sustava. Kada se informacijski sustav implementira tada se modeli procesa ostvaruju kroz odgovarajuće aplikacije (programsku podršku). Komunikacija između baze podataka i SUBP-a odvija se pomoću posebnih programskih jezika koji se dijele na nekoliko kategorija:

- DDL (*engl. Data Definition Language*) – jezik za stvaranje strukture podataka (opis), brisanje strukture podatka, dodavanje postojeće strukture podatka;
- DML (*engl. Data Manipulation Language*) – jezik koji koristimo kao primarni mehanizam za određivanje naredbi za upravljanje podacima koji se odnose na bazu podataka;
- DIL (*engl. Data Integrity Language*) – jezik koji se koristi pri određivanju ograničenja koja se odnose na integritet podataka;
- DCL (*engl. Data Control Language*) – jezik koji je dio pod-jezika baze podataka koji je dizajniran da se koristi od strane administratora baze podataka pri određivanju ovlaštenih korisnika baze podataka i određivanju njihovih mogućnosti rada u samoj bazi podataka;
- QL/SQL (*engl. Query Language*) – jezik za programiranje koji nije u potpunosti jezik za razvoj aplikacija, formalno je to jezik koji je namijenjen za rad baze podataka, postao standardni prikazani jezik za rad relacijskih i ne-relacijskih baza podataka, najčešće ga koristimo za postavljanje upita. [9]

3.1. Relacijski model baze podataka

Ustroj relacijskog modela je takav da se baza sastoji od skupa dvodimenzionalnih tablica (relacija). Kako bi relacije međusobno razlikovali svaka ima svoj, po mogućnosti jedinstveni naziv. Jedan stupac relacije sadrži vrijednost jednog atributa za entitet ili vezu radi čega se stupac poistovjećuje s atributom i obratno. Atribut ima svoje ime po kojem ga razlikujemo od ostalih unutar relacije (tablice) a njegova vrijednost je u podacima istoga tipa. Svaka vrijednost atributa treba biti čim jednostavnija po mogućnosti jednostruka tako da se ne da rastaviti na

daljnje segmente ili dijelove. U rijetkim slučajevima može se dopustiti da vrijednost atributa nije unesena. Skup svih atributa unutar pojedine relacije nazivamo relacijska shema.

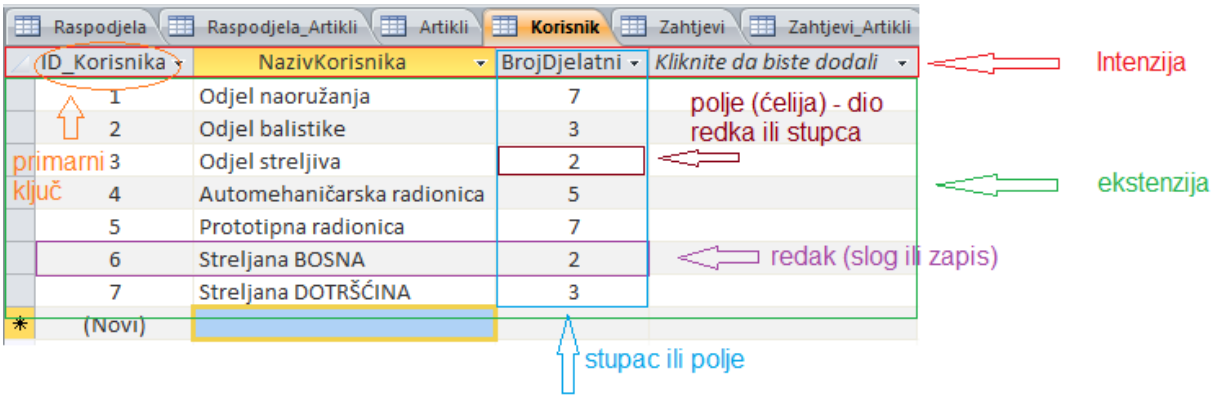
Jedan red unutar relacije najčešće predstavlja jedan primjerak entiteta ili bilježi oblik veze između više njih, taj red nazivamo i n-torka i u jednoj relaciji ne smiju postojati dvije identične n-torke.

Entiteti su predstavljeni nekim **n** brojem atributa (intenzijom) i nekim **m** brojem n-torki koje svojim svojstvima i vrijednostima opisuju entitet (ekstenzijom). Intenzija predstavlja osnovu tzv. "kodeksa atributa" ili strukturu tablice dok sam kodeks atributa osim naziva atributa (naziva stupca tablice ili polja retka [n-torke]) sadrži i podatke o vrsti i formatu podataka (metapodatke). [10]

Pri radu s relacijama se koriste operacije relacijske algebre u čije osnove spadaju: unija, razlika, selekcije, spajanje, presjek, dijeljenje, projekcija, Kartezijev produkt, a rezultat bilo koje operacije relacijske algebre je nova relacija.

Kako bi se izbjegla mogućnost da se u jednoj relaciji pojave dvije jednake n-torke prethodno moramo definirati skup atributa koji će nam jednoznačno definirati svaku pojedinu n-torku unutar relacije. Jednoznačnim određivanjem pojavljivanja svake n-torke (retka u relaciji) se definira primarni ključ koji označava atribut ili njihov skup čije su vrijednosti unutar relacije jedinstvene kako bi se osigurala njena "sigurnost" od umnožavanja, a vrijednost tog primarnog atributa ne smije niti u jednoj n-torki ostati neupisana.

U konačnici, baza podataka je skup relacija kod kojih ne postoji fizička veza već su one povezane preko stranih ključeva a samo nepostojanje fizičke veze je osnovna razlika između relacijskog i ostalih podatkovnih modela.



Slika 9. – Prikaz podataka u tablici

3.2. Strukturirani upitni jezik – SQL

SQL je skup instrukcija koji se koristi za interakciju s relacijskom bazom podataka. U biti, SQL je jedini jezik koji razumije većina baza podataka. svaki put kada komunicirate s bazom podataka, programska podrška (softver) prevodi vaše zapovjedi u SQL oblik koji baza podataka može protumačiti. [18]

Rukovanje podacima unutar relacijske baze podataka obavlja se uz pomoć SQL jezika na dva načina:

- direktnim pisanjem upita u bazu podataka (*engl. Interactive SQL*), ili gdje se upiti mogu postavljati u obliku obrazaca (*engl. query by forms*);
- pomoću gotovih aplikacija.

Također jezici sadrže tri grupe naredbi:

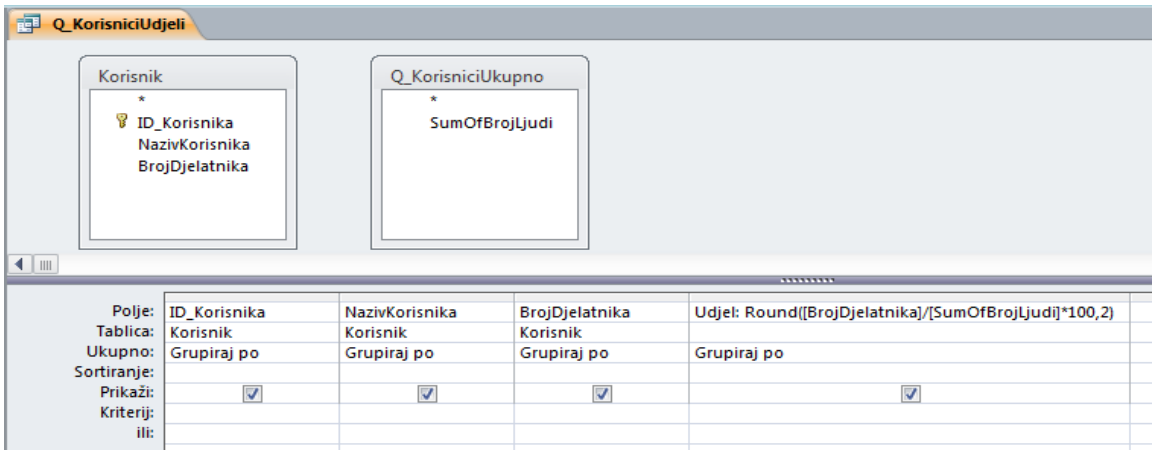
- naredbe za osnovne operacije nad podacima;
    - pronalaženje nepostojećih n-torki,
    - upis novih n-torki,
    - izbacivanje ili brisanje podataka n torki,
    - promjena podataka n-torki,
- naredbe za upravljanje bazom podataka;
- naredbe za definiranje relacijske sheme baze podataka.

Gotovo svi današnji programski paketi podržavaju relacijski model i SQL, svaki od njih sadrži svoj “menadžment sustav“ te ostale uobičajene alate za razvoj aplikacija. Svaka verzija računalne platforme (operacijskog sustava) zahtjeva sebi prikladni oblik i inačicu podatkovnog paketa. Na primjer, identični podaci iz nekog Upita mogu biti prikazani u više oblika od kojih ćemo jedan prikazati i u SQL obliku.

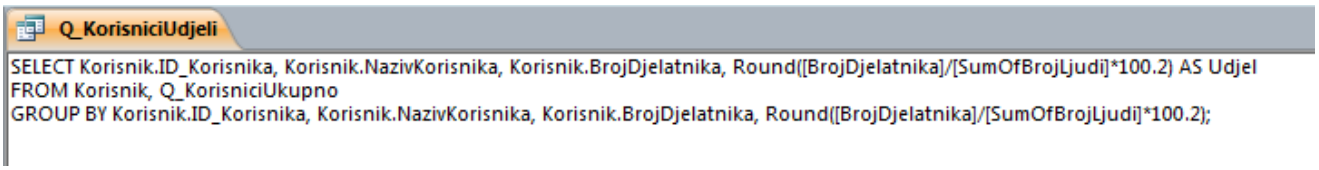
Q_KorisniciUdjeli

ID_Korisn	NazivKorisnika	BrojDjelatni	Udjel
1	Odjel naoružanja	7	24
2	Odjel balistike	3	10
3	Odjel streljiva	2	7
4	Automehaničarska radionica	5	17
5	Prototipna radionica	7	24
6	Streljana BOSNA	2	7
7	Streljana DOTRŠĆINA	3	10

Slika 10. – Prikaz podatkovnog lista



Slika 11. – Podaci u obliku prikaza dizajna



Slika 12. – Podaci prikazani u SQL obliku

3.3. Pojedini tržišno dostupni programski alati i podrške

Operacijski sustav mora osigurati nesmetanost provedbe svih procesa koji se provode na računalu. Osnovno sredstvo sustava je procesor čiji je zadatak omogućiti istodobnu funkcionalnost više procesa. Kako bi se to riješilo uobičajeno se koristi raspoređivač poslova koji upravlja dodjelom procesora metodom kružne podjele vremena (*engl. Round Robin, RR*).

Osim procesora, razni procesi programi mogu istodobno tražiti korištenje nekog drugog sredstva. Takvi se problemi rješavaju korištenjem sinkronizacijskih mehanizama kojima se raspoređuje pristup tim sredstvima (npr. jedan po jedan). Uobičajeni načini komunikacije su korištenjem poruka, zajedničkih spremničkih lokacija, datotečnog sustava te mrežnog sustava tako da ponekada mogu komunicirati i programi s različitih računala unutar nekog istog sustava [12]

Tabela 3. – Neki od poznatih programskih paketa za rad s bazama podataka [11]

Proizvođač	Produkt	Operacijski sustav	Jezici
IBM Corporation	DB2	Linux, UNIX (razni), MS Windows NT/2000/XP, VMS, MVS, VM, OS/400	SQL, COBOL, Java, ...
Oracle Corporation	Oracle	MS Windows (razni), Mac OS, UNIX (razni), Linux i drugi	SQL, Java i drugi
IBM Corporation (prije : Informix Software Inc.)	Informix	UNIX (razni), Linux, MS Windows NT/2000/XP	SQL, Java i drugi
Microsoft	MS SQL Server	MS Windows NT/2000/XP	SQL, C++, ...
MySQL AB	MySQL	Linux, UNIX (razni), MS Windows (razni), Mac OS	SQL, C, PHP, ...
Sybase Inc.	Sybase SQL Server	MS Windows NT/2000, OS/2, Mac, UNIX (razni), UNIXWare	SQL, COBOL, ...
Hewlett Packard Co.	Allbase/SQL	UNIX (HP-UX)	SQL, 4GL, C, ...
Cincom Systems Inc.	Supra	MS Windows NT/2000, Linux, UNIX (razni), VMS, MVS, VM	SQL, COBOL, ...
Microsoft Corporation	MS Access	MS Windows (razni)	Access Basic, SQL

Svaki program je namijenjen za obavljanje jednog ili više zadataka. Današnji operacijski sustavi za osobna računala, poslužitelje pa čak i ručna računala omogućuju pokretanje i rad više od jednog programa paralelno. Svaki zadatak bi se trebao odrađivati u zasebnom procesu koji je nastao pokretanjem programa. Pojedini se zadatak može sastojati od nekolicine zadataka niže razine (podzadataka). Npr. reprodukcija sadržaja s mrežnih stranica sastoji se od podzadatka koji dohvaća sadržaj, podzadatka koji sadržaj prikazuje te podzadatka koji očitava naredbe korisnika. [12]

3.4. Združena računalna tehnologija

Ovakav oblik označava tehnologiju koji ima ugrađeno nekoliko proizvoda Microsoft-a. Ta tehnologija je podsustav baza podataka, a nju sačinjavaju i upotpunjuju pojedini sustavni podaci pohranjeni u računalu. Za potrebe većih baza podataka, Jet baze podataka mogu biti nadograđivane ili uvećane odnosno podaci mogu biti migrirani na SQL Server.

3.5. Otvorena spojivost podatkovnih baza

ODBC je oblik komunikacijske tehnologije (platforme) uz pomoć koje svi korisnici ili klijenti drugih tehnologija, koje nisu izgrađene uz pomoć ili kompatibilne sa sustavom Windows, komuniciraju sa SQL Serverom na kojem je eventualno smještena neka baza podataka. MS SQL Server i Sybase/ASE komuniciraju uz pomoć aplikacijski niveliranog protokola TDS (*engl. Tabular Data Stream*) koji također omogućava da što više aplikacija može komunicirati s navedenim bazama podataka. Nakon 2005. godine također je izdan i certificiran JDCB API za komunikaciju s Java aplikacijama.

3.6. SQL Server

MS SQL Server je prva baza podataka koja je koristila korisničko sučelje. To je relacijska baza podataka kojoj je primarni jezik za upite Transact SQL (T-SQL) što znači da osim klasičnih SQL upita dozvoljava i složenije stvari poput mijenjanja programskog toka i sl. Po svojoj veličini SQL Server je baza podataka srednje veličine. [14]

4. Odabir sredstva za ostvarenje projekta

Svaki sustav izgrađen je od više sastavnica ili komponenti koje mogu biti raspoređene i u nekoliko slojeva. Također je prilikom odabira dobro znati koja će biti razina dostupnosti tog sustava, dali će to biti s razine osobnog računala koje je namijenjeno radu jednog korisnika ili s razine poslužitelja (*engl. Server*) koji istovremeno može koristiti više korisnika i td..

Računala prema namjeni možemo podijeliti na:

- poslužitelje (serveri),
- radne stanice:
    - grafičke radne stanice
    - multimedijske radne stanice
    - mrežne radne stanice
    - laboratorijske i mjerne radne stanice,
- samostojeća računala,
- prijenosna računala:
    - laptop (uobičajeno prijenosno računalo)
    - netbook (mali prijenosnici za pristup Internetu)
    - dlanovnici i inteligentni mobiteli,
- POS (*engl. Point of Sale*) računala,
- tanki klijenti - s poluvodičkom memorijom,
- grozdovi računala (*engl. Clusters*),
- distribuirani računalni sustavi (*engl. Datagrids*). [10]

Svako računalo mora sadržavati i neki od programa pomoću kojeg će obavljati zadane mu operacije i mehanizme za pokretanje tog istog programa koji bi u najmanjoj mjeri trebali uključivati:

- interakciju korisnika s programom,
- učitavanje programa s diska u spremnik,
- rezervaciju spremničkog prostora za program,
- zaštitu pokrenutog programa od drugih ranijih programa iz sustava i sl.

Takvi mehanizmi ostvaruju se pomoćnim programima koji omogućuju izvođenje operacija na računalu a svi oni objedinjeni čine operacijski sustav čija je osnovna zadaća upravljanje svim elementima sustava. Kako za osobna računala tako i za

poslužitelje na ovim prostorima su najraširenij te sukladno tome i najkorišteniji operativni sustavi koje čine razne inačice Microsoft Windows operacijskih sustava. Također ali nešto manje zastupljeniji su operativni sustavi tvrtke Apple Inc. kao što je Mac OS X ili besplatni operacijski sustav zasnovan na Linux jezgri npr. Ubuntu, Suse i td.. Bez obzira na namjenu u svim operacijskim sustavima osnovna funkcionalnost podsustava je vrlo slična.

Prilikom odabira operacijskog sustava koji će upravljati sklopovljem i programima a korisniku ponuditi neki od oblika standardiziranog sučelja npr. korisnik programe koristi preko korisničkog sučelja, operacijski sustav preko njegovog grafičkog korisničkog sučelja (*engl. Graphical User Interface, GUI*) i td.. Također prilikom odabira potrebno je obratiti pozornost na daljnje održavanje naše baze podataka, mogućnost njenog rasta i razvoja te mogućnost kompatibilnosti s ostalim operacijskim sustavima koji se koriste na poslužiteljskim ili korisničkim radnim stanicama.

Nakon kratkotrajnog promišljanja zaključeno je kako MS Access-om možemo zadovoljiti gotovo sve zahtjeve koji se stavljaju pred bazu podataka koja je u nastajanju. Združena računalna tehnologija na strani klijentske radne stanice (*engl. Joint Engine Technology, Jet / JetBase*) je sastavni mehanizam upravljanja relacijskim bazama podataka u Access-u zajedno sa svojim grafičkim korisničkim sučeljem [6], uz mogućnost otvorene spojivosti podatkovnih baza (*engl. Open DataBase Connectivity, ODBC*) pristupa bazama podataka drugih proizvođača. [13]

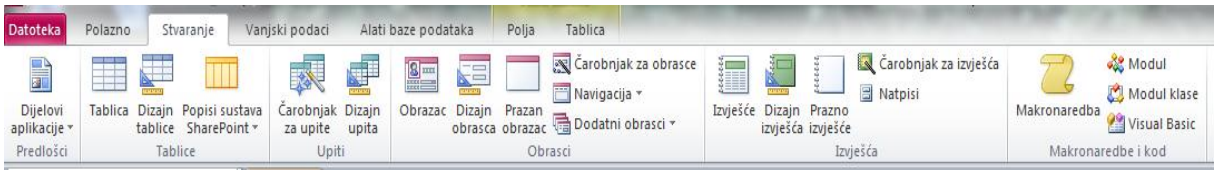
5. Microsoft Access 2010

Microsoft Access je sustav namijenjen upravljanju s relacijskim bazama podataka SURBP (*engl. Relation DataBase Menagement System, RDBMS*), proizveden od tvrtke Microsoft te je sastavni dio uredskog paketa Microsoft Office 2010.

Najraniji oblik Access-a (1.1) razvijen je 1992. godine nakon čega su ubrzo počele izlaziti novije verzije koje su bile podržane Access Basic programskim jezikom. Verzijama novijim od (8.0) u Access-u 97, taj programski jezik je zamijenjen VBA (*engl. Visual Basic for Applications*) jezikom. Sljedeća značajna promjena dogodila se pojavom Access-a 2007 kada je predstavljen novi format baze podataka .ACCDB koji je oblik baze podataka koji je podešen tako da automatizmom pohranjuje podatke koji su uneseni u nju, za razliku od prethodnog .MDB oblika, te podržava unos više vrijednosti u jedno polje, pruža mogućnost dodavanja raznih priloga bazi podataka itd. [13, 17]

Sam Access je jedan od najprilagodljivijih programa za izradu baza podataka koji omogućava sakupljanje, pohranjivanje i organiziranje informacija kao i izradu različitih izvještaja gdje dolazi do izražaja njegova jednostavnost uporabe.

Elementi MS Access-a su prikazani na Slici 13., koja sadrži: Tablice (*engl. Table*), Upiti (*engl. Query*), Obrasci (*engl. Form*), Izvješća (*engl. Report*), Makronaredbe i kod (*engl. Macro i Modul*).



Slika 13. – Prikaz kartica MS Access-a 2010

MS Access je napredni razvojni sustav koji nam omogućava da oblikujemo kodeks atributa paralelno sa fizičkom izradom tablica s time da svi podaci koje unesemo se automatski dokumentiraju u bazi tako da u slučaju eventualne potrebe možemo obaviti ispis kodeksa atributa u obliku dokumenta.

5.1. Tablice

Tablica s podacima je “srce“ svake relacijske baze podataka, izgledom je vrlo slična radnoj stranici MS Excel tablice kako je vidljivo na Slici 14. Access pohranjuje

svaki unos u tablice u svoje zasebni redak koji nazivamo slog ili zapis (*engl. Record*) koji označava skup informacija o jednoj stvari (entitetu) dok svaki detaljan opis koji je sadržan u svom stupcu nazivamo polje (*engl. Field*) s time da u djelu polja ili retka nalazi se jedinstven podatak za određeni slog kojeg nazivamo polje ili ćelija (*engl. Cell*). [15]

Korisnik

ID Korisnik	Naziv Korisnik	BrojLjudi	Kliknite da biste dodali
1	Odjel naoružanja	7	
2	Odjel balistike	3	
3	Odjel streljiva	2	
4	Automehaničarska radionica	5	
5	Prototipna radionica	7	
6	Streljana BOSNA	2	
7	Streljana DOTRŠĆINA	3	
(Novi)			

Slika 14. – Prikaz tablice

5.2. Upiti

Upit je segment uz pomoć kojeg potražujemo ili analiziramo određeni oblik podataka iz jedne ili više tablica koje se nalaze unutar naše baze podataka a ujedno se mogu koristiti kao podloga kod izrade forme ili izvještaja. Upit se može vrlo jednostavno izraditi uz pomoć Čarobnjaka za upite koji pomaže u izradi samog upita ili možemo izraditi vlastiti dizajn upita, bez Čarobnjakove pomoći budući da Čarobnjak nudi nekoliko vrsta već predodređenih upita kao što su:

- jednostavni upiti (*engl. Select*) – stvara odabrani upit iz polja koja ste odabrali, traži podatke koji ispunjavaju zadane uvjete i obavlja zadane zadatke nad tim podacima;
- unakrsni upiti (*engl. Crosstab*) – stvara unakrsni upit koji prikazuje podatke u sažetom obliku nalik proračunskoj tablici, prikazuje vrijednosti kao što su zbrojevi, prebrojavanja ili presjeci jednog polja;
- pronalaženje dvostrukih upita (*engl. Find duplicates*) – pronalazi zapise s dvostrukim vrijednostima polja u jednostrukoj tablici ili upitu;
- pronalaženje ne uparenih upita (*engl. Find unmatched*) – pronalazi zapis (redak) u jednoj tablici koji nemaju srodnih zapisa u drugoj tablici koja je zasnovana na uspoređivanju određenih polja.

Nužno je voditi računa o činjenici da korisnik ili njegov nalogodavac u najvećem broju slučajeva govore hrvatski jezik dok sustav raspoznaje samo SQL jezik koji je zasnovan na engleskom jeziku. Svaki korisnik Acces-a bi ga trebao poznavati u tolikoj mjeri da može bez problema iz njega dobiti potrebne mu podatke.

Svaki daljnji rad s bazom podataka u Access-u zahtjeva oblikovanje objekata (obrazaca ili izvještaja) koji se ne mogu izraditi bez upita ili tablica, pa se njihovo oblikovanje može znatno ubrzati samo pod uvjetom da su izvori podataka koje sačinjavaju tablice ili upiti izrađeni u ispravom obliku.

5.3. Obrazac

Obrazac (*engl. Form*) je dio baze podataka koji se koristi za manipulaciju podacima iz tablica ili upita kroz npr.: dodavanje podataka, mijenjanje ili brisanje podataka ili njihov pregled. On omogućava unos i prikazivanje podataka u prikladnom formatu koji izgledom podsjeća na tiskane obrasce u kojima treba popuniti prazna mjesta. Obrasci mogu biti jednostavni ili složeni u različitim formama kao što su npr.: obična forma, podijeljena forma, višestruka forma, prazna forma, dijalog forma ili graf forma i različitim mogućnostima automatskog pretraživanja, koje unos podataka čine lakšim i bržim.

Obrazac obično prikazuje samo jedan zapis tablice ili upita dok je za prikaz više zapisa istovremeno potrebno je da su tablice u relacijskoj vezi. Obrasci mogu sadržavati i druge pod-obrasce (*engl. Subforms*) koji omogućuju istovremeni unos podataka u više tablica.

Postoji nekoliko načina stvaranja obrazaca:

- opcija "Čarobnjak za obrasce" – pomaže u stvaranju istog putem okvira za dijalog i odabirom polja i stilova za formu;
- opcija "Prazan obrazac" – je najteži način stvaranja koja nudi mrežu u koju ćete postavljati željena polja a ujedno nudi i najviše mogućnosti;
- opcija "Dodatni obrasci" - nudi više oblika već ranije definiranih obrazaca.

Sam upit u koliko otvorimo u obliku „dizajn“ on će nam omogućiti razna modeliranja upita sukladno našim željama odnosno potrebama kroz mijenjanje postavki upita i razna uređivanja.

5.4. Izvještaji

Izvještaj (*engl. Report*) *se* koristi kada je potrebno neki grupirani segment podataka pripremiti za ispis. Može se izraditi iz tablice ili upita. Ne razlikuje se značajno od obrasca, jedino što je u izvještaju moguće isključivo samo gledati podatke. Mogu biti jednostavni ili složeni.

Izvještaji potrebne podatke pronalaze unutar tablica ali još češće se referiraju na upite dok je njihov osnovni zadatak da te podatke prikažu u obliku koji je lako razumljiv. Nakon što je izvještaj oblikovan njegov stvarni izgled je moguće pregledati u obliku kakav će biti ispisan na pisaču (*engl. Print preview*) što se može iskoristiti za ispravljanje eventualnih pogrešaka ili ispravljanje izgleda samog izvještaja. Osim ispisa izvještaja isti se može pohraniti u neku od datoteka, u različitim formama, te naknadno ispisati.

5.5. Makronaredbe i kod

Makronaredba je alat koji omogućava automatizaciju zadatka i dodavanje funkcionalnosti obrascima, izvješćima i kontrolama. Najčešće se koristi kada je potrebno ,određenim redoslijedom, nekoliko naloga (akcija) pozvati na više mjesta unutar Access aplikacije, najčešće iz oblika forme. Sama makronaredba sastoji se od niza naloga koje nudi Access, koje korisnik ne mijenja nego ih uporabljuje: za prijenos podataka, za rad s objektima, za izvršenje pojedinih zadataka, za rad s podacima forme i izvještajima, i sl..

Naprednije se makronaredbe oblikuju zadavanjem uvjeta od kojeg ovisi što će makronaredba učiniti. Uvjet može sadržavati operatore, identifikatore, funkcije, literale i konstante. Poželjno je da se svako često ponavljanje neke operacije zamijeni makronaredbom, što će uz točnost i efikasnost omogućiti uvijek istovjetno obavljanje zadanih operacija nad podacima ili objektima unutar baze podataka.

Kod kao i makronaredba omogućava automatizaciju, ali mnogo napredniju, jer se u njemu, kao razvojna platforma koristi se VBA programskim jezikom, u obliku procedura koje su na raspolaganju cijelom projektu.

Kako su tablice izvori podataka, tako Access datoteka u slučaju da ima tablice predstavlja bazu podataka, dok se u Access aplikacijama uglavnom nalaze ostali

objekti, a veza sa tablicama se uspostavlja linkom. Koncepcija odvajanja baze od aplikacije se obvezno primjenjuje u profesionalnim projektima.

5.6. Upravljanje sigurnošću baze podataka

Sigurnost baze podataka nam je jedan od ključnih elemenata u koliko želimo da podaci u bazi podataka budu ispravno ažurirani, da njihovi podaci ostanu nepromijenjeni, u koliko to ne učini “ovlašteni“ korisnik baze, te da nam ta baza na kraju daje odgovarajuće rezultate. [16]

Mogućnosti za provedbu sigurnosti nad bazom podataka su mnogobrojne kao i njihova međusobna kombiniranja, tako da ćemo spomenuti samo neke od njih kao što su:

5.6.1. Konfiguracijski propusti

Iako SUBP često ne podržavaju sigurnosna rješenja kao drugi sustavi, ispravno oblikovanje i određivanje postojećih mogućnosti može povećati razinu sigurnosti cjelokupne baze podataka te samim time ukloniti i određeni broj ranjivosti.

5.6.2. Nedostatak zaštite korisničkih računa

Velika većina SUBP nema mogućnost zaštite korisničkih računa koje se odnose na nemogućnost kontrole zaporki i nemogućnost određivanja roka valjanosti korisničkog računa.

5.6.3. Kontrola pristupa tablicama

To je najčešće najzanemarivaniji element zaštite baza podataka zbog toga što je njena implementacija složena i zahtjeva suradnju sustavnog administratora i razvojnog programera baze podataka. Primjeri ovakve kontrole su onemogućavanje čitanja tablice na istom sastanku na kojem je u nju obavljen neki upis ili dozvoljavanje čitanja samo određenog tipa tablica.

5.6.4. Neprikladne metode kontroliranja rada baze podataka

Kontroli rada SUBP često su nametnuti visoki zahtjevi u pogledu performansi i korištenja diskovnog prostora. Zbog toga je umanjena učinkovitost forenzičke analize i otežano utvrđivanje odgovornosti.

Ispravne metode nadzora su ključne u razumijevanju napada na SUBP jer bilježe aktivnosti izravno vezane uz pohranjene podatke.

5.7. Mogućnosti zaštite baze podataka

Postoje alati kao što je Microsoft Accesas koji omogućuje pristup bazi podataka pomoću ODBC-a (*engl. Open DataBase Connectivity*) ili nekog drugog protokola koji u potpunosti zaobilazi sigurnosne provjere ugrađene u aplikacije, tako da jedina pouzdano sigurnosna ograničenja su ona koja su ugrađena izravno u SUBP.

5.7.1. Sustav dozvoljenih IP adresa

SUBP bi trebao posluživati isključivo sigurne IP (*engl. Internet Protocol*) adrese. Ako se radi o radu s dinamičkim internetskim stranicama onda to treba biti samo IP adresa internetskog poslužitelja, a ako se baza podataka koristi u sprezi s nekom lokalno korištenom aplikacijom onda pristup treba biti dozvoljen isključivo s IP adresa koje pripadaju lokalnoj mreži.

5.7.2. Programski propusti unutar SUBP-a

Oni uključuju razne pogreške koje mogu udaljenim zlonamjernim korisnicima omogućiti izvođenje napada zasnovanih na uskraćivanju resursa (*engl. Denial of Service, DOS*) napada ili unošenje programskog koda koji može rezultirati različitim posljedicama.

5.7.3. Korištenje enkripcije

Obzirom na podatke koji se kriptiraju i na razinu na kojoj se kriptiranje obavlja, postoji nekoliko vrsta enkripcije primjenjivih u zaštiti baza podataka kao što su:

- korištenje enkripcije tijekom prijenosa podataka (*engl. Data-inmotion*) - većina SUBP podržava komunikaciju uporabom SSL (*engl. Secure Socets Layer*) zaštitnog protokola;
- korištenje enkripcije za podatke u mirovanju (*engl. Data-at-rest*), gdje nije potrebno kriptirati sve podatke, već samo najosjetljivije;
- korištenje enkripcije datoteka (*engl. File-based*), ali takav pristup ne štiti od napada kroz SUBP;
- enkripcija na razini programskog sučelja (*engl. API - Application Programming Interfaces*) podrazumijeva kriptiranje komunikacije među pojedinim podsustavima SUBP-a;

- “transparent“ enkripcija je enkripcija koja se automatski primjenjuje pri svakoj promjeni ili unosu potencijalno osjetljivih podataka.

Čim SUBP posjeduje više funkcionalnosti automatizmom se uvećava i mogućnost pogreške tako da možemo zaključiti kako je najsigurniji onaj sustav kojeg administrator najbolje poznaje kroz arhitekturu, funkcionalnost i međusobne odnose što mu omogućuje da pravovremeno poduzme sve mjere opreza i spriječiti pojavu ranjivosti u našoj bazi podataka.

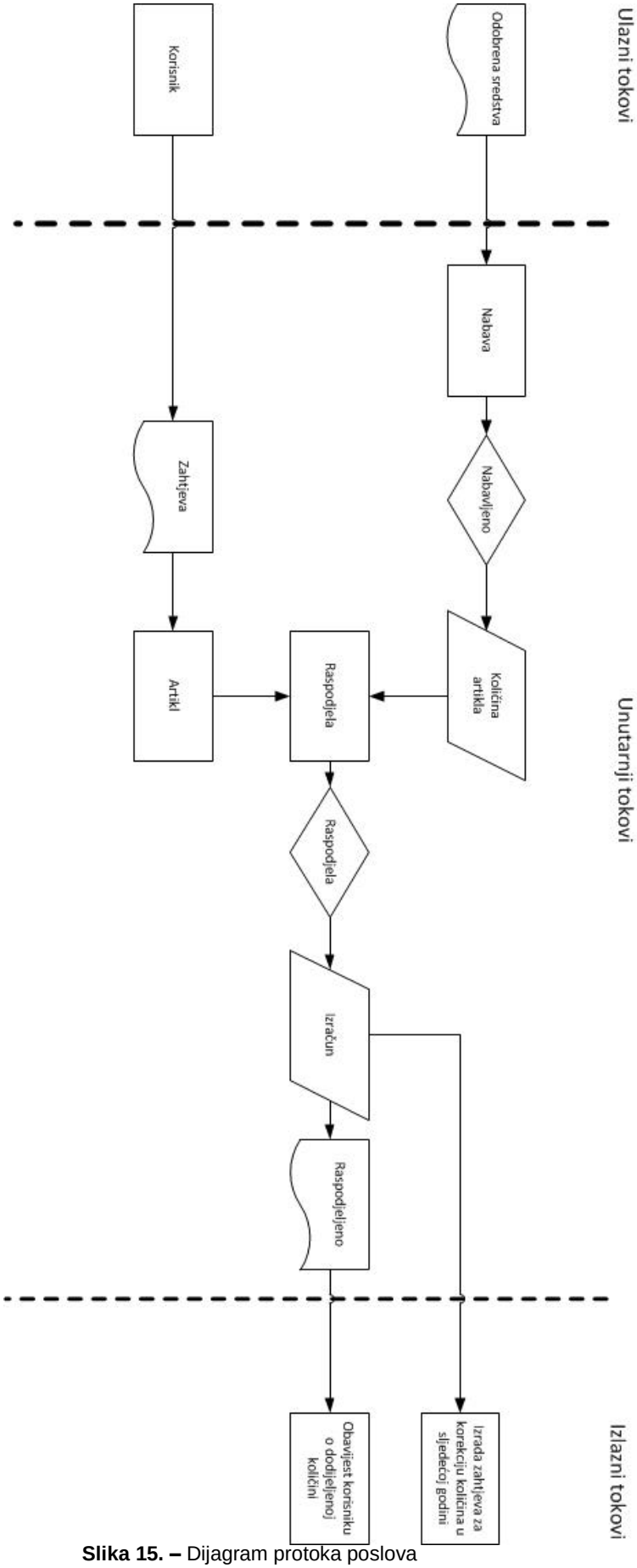
6. Izrada podsustava za raspodjelu zaštitnih sredstava

Nakon obavljene analize dostupnih alata koji se nalaze na tržištu odlučili smo koristiti MS Access 2010 koji je jednostavan i snažan alat čija prvenstvena namjena je pomoć pri uredskom poslovanju zajedno s pripadajućim SUBP MS JetBase kao razvojnom platformom na kojoj će podsustav biti izrađen. Informacijski podsustav ovog završnog rada zamišljen je kao poslovni informacijski sustav za transparentno obavljanje raspodjele zaštitnih sredstava unutar nekog poslovnog subjekta koji uz određenu nadogradnju lako može postati jedan od sustava za podršku bilo koje tvrtke.

Kod izrade podsustava baze podataka procedura je jednaka kao i kod izrade cjelovitog sustava baze podataka. Potrebno je obaviti planiranje koje će nam odgovoriti na pitanje radi čega izrađujemo podsustav, potrebno je obavit analizu na način da dobijemo odgovore na pitanja kao što su tko će koristiti sustav, koje su funkcije sustava (što mora raditi) te gdje, kada i kako će biti korišten da bi kroz dobivene odgovore mogli uopće oblikovati sustav.

Sva ta saznanja tvore određeni oblik specifikacije, koja kada je u cijelosti definirana omogućuje i olakšava nam izradu. Nakon izrade i obavljenih potrebnih testiranja izrađenog podsustava možemo obaviti i isporuku odnosno ugraditi to podsustavno rješenje u veći sustav kako bi dobili funkcionalni sustav unutar kojeg je nužno provoditi pravovremeno održavanje i poboljšavanje sustava kako bi isti bio konstantno operabilan.

U ovom dijelu rada opisat ćemo razvoj i izradu pojedinih dijelova aplikacije za Vođenje evidencije zaštitnih sredstava odnosno njihovu raspodjelu. Budući da u novije vrijeme uglavnom zbog “recesije“ ili povećanja dobiti tvrtke, glavna smjernica smanjenje troškova u svim segmentima poslovanja, kako u tvrtkama privatnog tako je to još više izraženo u poslovnim subjektima društvenog vlasništva. Jedan od primjera protoka poslova prikazan je na Slici 15.



Slika 15. – Dijagram protoka poslova

Kako bismo pokrenuli MS Access 2010 (u daljnjem tekstu Access), na računalima koja sadrže kompatibilan operativni sustav, potrebno je pokazivač miša postaviti na ikonu i duplim klikom na lijevu tipku miš-a je pokrenuti.

Nakon što nam se otvori Access u svom početnom obliku moći ćemo se odlučiti između nekolicine već unaprijed predodređenih oblika baze podataka. Budući da predodređeni oblici koji su nam unaprijed definirani u ovom slučaju neće odgovarati odabrat ćemo oblik Prazna baza podataka, upisat ćemo njen naziv u za to ponuđenu rubriku, odabrati mjesto u računalu gdje će biti smještena, nakon čega ćemo potvrdom na ikonu Stvori stvoriti platformu naše baze podataka.

Kako su tablice "srce" naše baze podataka iste možemo izraditi na nekoliko načina:

- tablicu stvorimo sami:
    - izrada tablice kroz razvojni prikaz (prikaz dizajna);
    - izrada tablice kroz sadržajni prikaz (prikaz podatkovnog lista);
- uvoz tablice iz nekih vanjskih podataka (najčešće iz MS Excel oblika).

Opis entiteta koji sudjeluju u procesima “naše“ baze podataka u grubo prema tipovima entiteta su prikazani u Tablici 4., dok kodeks njihovih atributa prikazani su u Prilogu A

Tablica 4. Opis entiteta “naše“ baze podataka

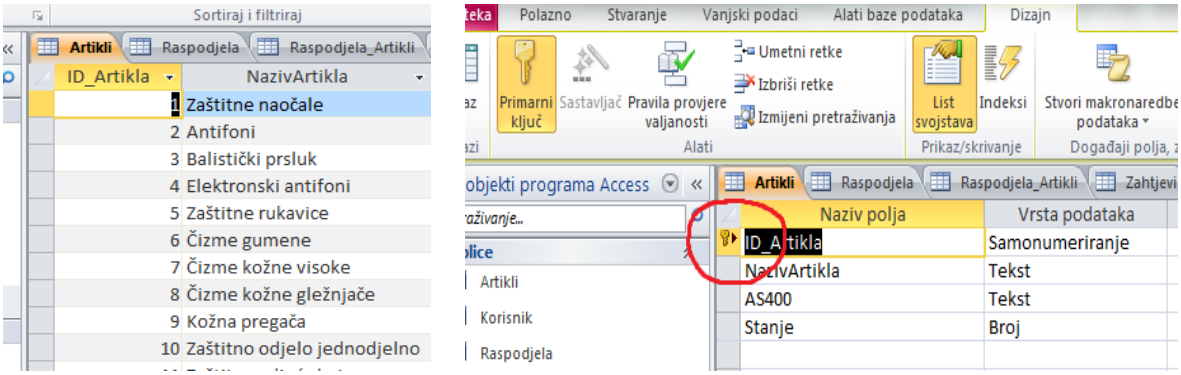
Tip entiteta	Entitet	Opis entiteta
Ljudi	Korisnik	Korisnik svojom ulogom započinje u trenutku kada uputi Zahtjev za dodjelom sredstava te se tada za tog konkretnog korisnika započinje s procesom Raspodjele.
Stvar ili fizički objekt	Artikl	Artikl predstavlja stvar, opremu ili neki drugi predmet za kojim Korisnik ima potrebu, koji mu omogućava da obavlja ili izvršava svoje “planirane“obveze.
Predmet poslovanja	Zahtjevi	Zahtjev je pisani dokument kojim nam se Korisnik obraća kako bi na temelju njega Korisnika uključili u Raspodjelu. Za Zahtjev, osim elemenata propisanim poslovima uredskog poslovanja, poželjno da sadrži količinu traženog artikla te ukratko opisnu namjenu za koju će isti biti korišten.

Poslovni događaj	Raspodjela	Raspodjela je “konačni događaj“ po radi kojeg je izrađena “naša“ baza podataka, po obavljenoj raspodjeli Korisnik dobiva pisanu obavijest o Artiklima i njihovim količinama koje su mu dodijeljene.

6.1. Primarni ključ

Primarni ključ (PK) je polje ili skup polja u tablici koji programu Access pružaju jedinstveni identifikator za svaki redak. U relacijskoj bazi podataka, informacije su podijeljene u zasebne tablice gdje se koristite odnosi između tablica i primarni ključevi kako bi se odredio način spajanja informacija. Kada jednom definirate primarni ključ, možete ga koristiti i u drugim tablicama za upućivanje na tablicu s primarnim ključem.

U prikazu podatkovne tablice polje je po zadanom skriveno, ali ga možete vidjeti ako prijeđete na prikaz dizajna. Najčešće se za primarni ključ koristi neki od jedinstvenih identifikacijskih brojeva za kojeg znamo da se neće ponoviti, poput ID broja ili serijskog broja ili šifre, itd. Na primjer, ako imate tablicu Artikli koja sadrži i ostale “opisne“ podatke i gdje svaki artikl ima jedinstveni ID artikla. Polje ID artikla je primarni ključ.



Slika 16. – Prikazi sadržajnog i razvojnog oblika tablice s podacima

Dobar primarni ključ ima nekoliko karakteristika:

- jedinstveno identificira svaki redak
- nikad nije prazno polje ili nula (uvijek sadrži neku vrijednost)
- rijetko (najbolje nikada) se ne mijenja.

Primjer loše odabranog primarnog ključa su podaci neke dinamičke kategorije kao što su nazivi ulica ili adresa koje se mogu promijeniti s vremenom. U nekim slučajevima možda ćete željeti koristiti dva ili više polja koja zajednički daju primarni ključ za tablicu. Kad primarni ključ koristi više od jednog stupca onda se on naziva složeni ključ.

6.2. Veze baze podataka

Kako se baze podataka najčešće sastoje od više tablica u kojima se nalaze različiti ali korisniku jednako bitni podaci, tablice moraju biti međusobno povezane (imati relaciju) jedna s drugom.

Veze koje ponekad nazivamo i relacije, među tablicama najčešće se zasnivaju uz pomoć primarnih ključeva i po principu jednakosti polja što znači da neko polje koje je u jednoj tablici označeno kao tekst nećemo moći povezati s poljem iz druge tablice koje je označeno kao broj. Također je nužno da tablice budu međusobno povezane u slučaju kada želimo izraditi neki oblik upita, izvješća ili obrazaca.

6.2.1. Vrste veza

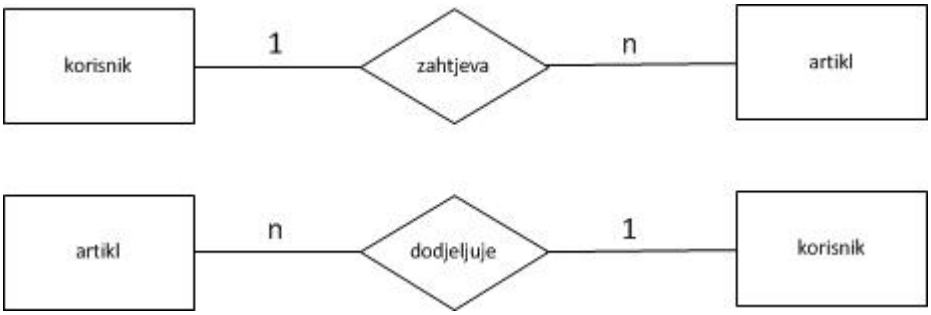
Jedna baza podataka se sastoji od jednog ili više entiteta koji mogu biti povezani na više načina:

- veza 1:1 (*engl. one to one*) predstavlja najobičniji oblik veze, oba povezana polja u relaciji su primarni ključevi, kao primjer nam može poslužiti slika 17. gdje se vidi da ministar može biti na čelu samo jednog ministarstva, odnosno u drugom smjeru da ministarstvo može voditi samo jedan ministar;

Slika 17. – Prikaz veze 1:1

- veza 1:n (n:1) (*engl. one to many*) u ovoj relaciji je samo jedno polje primarni ključ, to je najčešći oblik veze, a dali je ta veza 1:n ili n:1 ovisi o kutu gledišta. Na primjernoj, Slici 18., prikazano je kako neki korisnik može

zahtijevati više artikala gdje su entiteti korisnik i artikl u vezi 1:n jer više artikala može biti dodijeljeno jednom korisniku;



Slika 18. – Prikaz veze 1:n (n:1)

- veza n:n (*engl. many to many*) ovakav tip veze se vrlo često javlja u modelima, prilikom ovakvog načina povezivanja stvara se treća tablica čiji primarni ključ se sastoji od dva polja u kojima su strani ključevi iz tablica koje povezujemo. Problem ovog tipa veze između dva entiteta može se riješiti na način da se ovakva veza razdjeli na dvije veze u obliku 1:n tako da prikazana veza službenik : projekt (n:n) se dijeli na projekt : radi na (1:n) i radi na : službenik (n:1) gdje je radi na novi entitet (tablica u bazi) koja osigurava strogu kontrolu modela.

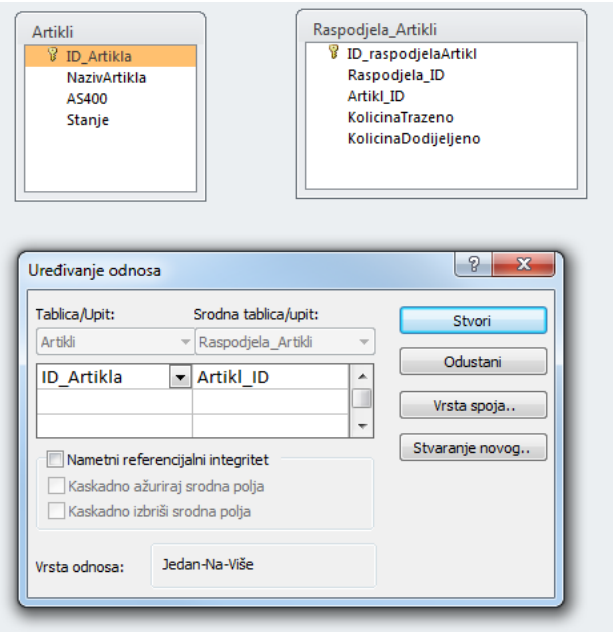
Slika 19. – Prikaz veze n:n

6.2.2. Međusobno povezivanje

Prilikom izrade i oblikovanja veza možemo se poslužiti alatom Prikaz tablice koji se nalazi u grupi Alati na kartici Dizajn. Tada će nam se otvoriti dijaloški okvir iz kojeg dodajemo tablice koje želimo prikazati u prozoru Odnosi. Također navedene tablice možemo postaviti u prozor Odnosi i pomoću povuci i pusti metode.

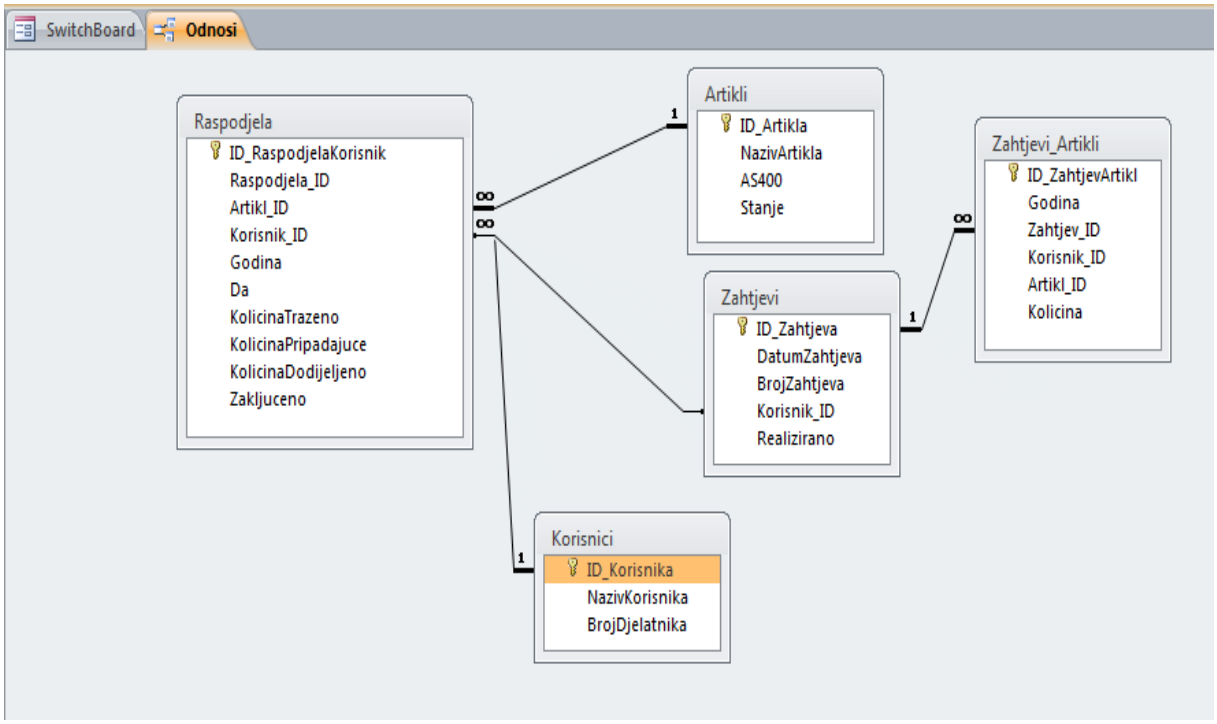
Kada izrađujemo vezu (relaciju) između tablica najčešće se služimo metodom povuci i pusti na način da kursor smjestimo na ID polje koje nam je primarni ključ u tablici Artikli, pritisnemo lijevu tipku miša, držeći tu tipku cijelo vrijeme ulazimo u prostor sljedeće tablice "Raspodjela_artikli" sve dok ne dođemo do polja "Artikli_ID",

koje nam je istoznačno kao i “ID_Artikli“ iz tablice “Artikli“, kome samo radi lakšeg razlikovanja izmijenili redoslijed u nazivu, nad kojim otpuštamo tipku.



Slika 20. – Stvaranje veze između tablica

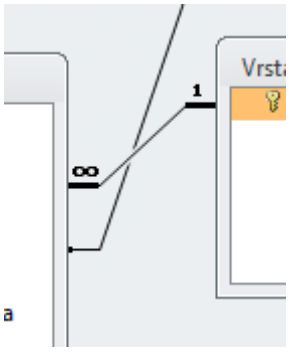
Nakon provedbe ovog postupka otvorit će nam se dijaloški okvir Uredi odnose te u koliko pritisnemo tipku Stvori prikazati će nam se vizualno, veza između te dvije tablice.



Slika 21. – Prikaz veza između tablica “naše“ baze

6.2.3. Referencijalni integritet - integritet povezanosti

Nametanje referencijalnog integriteta označava način uz pomoć kojega možemo osigurati cjelovitost svih zapisa u našoj bazi podataka. Primjenom tog oblika veze potvrđujemo valjanost odnosa između tablica i onemogućavamo slučajno brisanje ili zamjenu podataka na način da ukoliko pokušamo obrisati neki artikl iz tablice Artikli (koja je povezana sa tablicom Raspodjela_Artikli) Access aplikacija će nas o tome obavijestiti i upozoriti. Također nam referencijalni integritet omogućava kaskadno upisivanje podataka u polja koja su vanjski ključevi u relacijski povezanim tablicama ovisno o vrijednostima primarnog ključa osnovne tablice podataka.



Slika 22. – Prikaz veze s nametnutim referencijalnim integritetom

Bitno je napomenuti da za referencijalan integritet nije od presudne važnosti za sam rad baza podataka, u koliko korisnik izvrsno poznaje način njezinog rada i odnose koji su unutar nje predodređeni. Također pojedini dizajneri baza podataka ne koriste referencijalne integritete dok god je baza u fazi razvoja ili testiranja (beta verzija[2]), da bi lakše mogli vidjeti kakve se poteškoće javljaju i kakva je njihova učestalost pri radu baze podataka, te da u kasnijim koracima iste mogu ispraviti i ukloniti ili na neki drugi način ograničiti njihov utjecaj na rada baze podataka.

Da je nametnut referencijalni integritet u nekom odnosu možemo prepoznati tako što kod vizualnog prikaza veze na jednom kraju dobijemo oznaku 1 dok na drugom kraju dobijemo oznaku beskonačnosti.

Konkretno kod naše baze podataka referencijalni integritet će se nametati nad svim tablicama koje sadrže popis artikala gdje će polazišna “baza“ (referenca) integriteta biti tablica “Artikli“, također za nazive korisnika kao referentna će biti

[2] Beta verzija predstavlja nedovršenu inačicu novog programa u kojem su popravljene sintaktičke i logičke pogreške pronađene pri početnom ispitivanju.

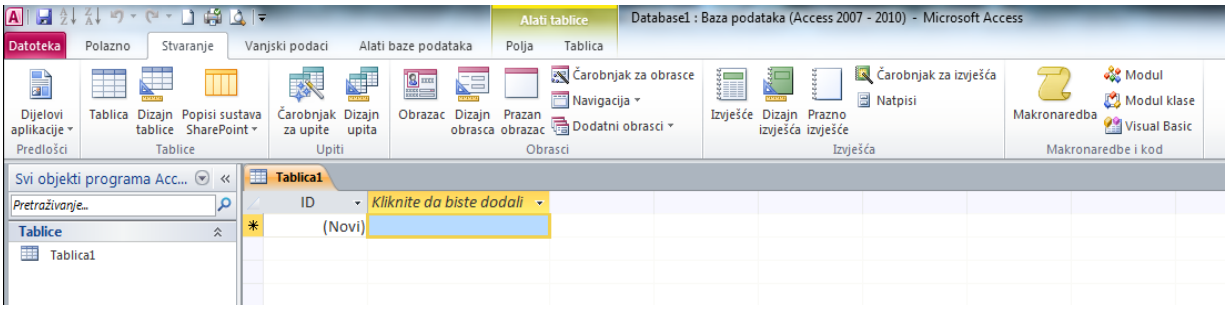
zadana tablica “Korisnici“, te nam je još jedna tablica bitna u kojoj će biti pohranjeni zahtjevi korisnika koja se nalazi poda nazivom “Zahtjevi“. Sve ostale veze unutar baze podataka će biti dinamičke kategorije i većinom će se bazirati nad upitima.

6.3. Rad s tablicama

6.3.1. Izrada tablice kroz razvojnu strukturu podatkovnog lista

Kada otvorimo praznu bazu podataka da bismo izradili tablicu kroz prikaz dizajna, odabiremo karticu Stvaranje pa u ponudi između ostalih oblika ponude nam se i Tablice gdje odaberemo Tablica. Početni naziv novonastale tablice je Tablica1 koju lako možemo preimenovati sukladno našim željama ili potrebama na dva načina.

Jedan način preimenovanja tablice je taj da prije no što spremimo podatke koje smo u ovaj oblik tablice unijeli sam sustav nas pita kako da spremi tablicu pod kojim nazivom, dok drugi oblik je da kada je tablica zatvorena iz prikaza odaberemo onu kojoj želimo promijeniti naziv u pomoćnom prozoru gdje su nam prikazani svi objekti programa, pritisnemo desnu tipku na mišu i onda odaberemo opciju Preimenuj, te upišemo željeni naziv i željeno potvrdimo s tipkom unos (*engl. enter*).



Slika 23. – Izgled prazne tablice u obliku podatkovnog lista

Kod obavljanja preimenovanja tablica s podacima nad kojima smo proveli nekakve radnje koje označavaju spajanje ili neke druge međusobne odnose može se pojaviti mogućnost da dobijemo grešku u sustavu stoga što će sustav samo pojedine odnose automatski ažurirati s novim nazivom tablice. Bitno je još napomenuti kako tablice koje su izrađene na ovaj način, oblik i vrstu podataka koja je u nju unesena sustav će pokušati prepoznati automatski, npr. dali se radi o datumu, nekoj brojčanoj vrijednosti ili tekstu, što ne znači da je to i ispravan oblik. Također se na ovaj način ne mogu uvesti neki drugi oblici ograničenja za unesene podatke kao što je na

primjer njihova veličina, odnosno broj polja koje zauzimaju u sustavu, tako da izrada tablice kroz ovaj oblik ostavlja veliku mogućnost nastanka nenamjerne pogreške koji nam u daljnjem radu baze podataka može zadavati znatne probleme.

Kako naša aplikacija koju ćemo izraditi sadržavati će sedam različitih tablica, abecednim redom kako slijedi, čiji sadržaj bi nam trebao omogućiti sljedeće:

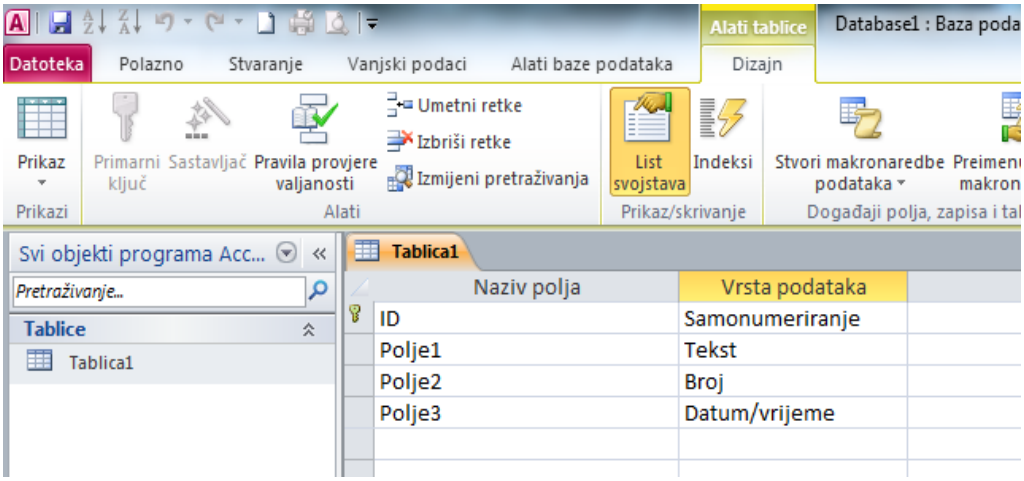
- tablica “Artikli“ – sadrži stupac “ID_Artikla“ što je PK koji će nam u daljnjem radu omogućavati lakše praćenje u ostalim podatkovnim oblicima o kojem se artiklu radi, u polje “NazivArtikla“ upisat ćemo cjeloviti naziv artikla koji kasnije možemo koristiti prilikom ispisa ili prikaza u nekom od izvještaja koje ćemo izraditi, stupac “AS400“ je stupac u kojem će biti unijeta interna numerička oznaka poslovnog subjekta za taj artikl i polje “Stanje“ u koje ćemo točno u određena polja za pojedini artikl upisivati količine sredstava koje su nam dostupne za raspodjelu;
- tablica “ArtiklKorisnik“ sadrži stupce “ID_AK“ koji nam je PK određenog artikla, odnosno poveznica da li se na nekog korisnika odnosi (raspoređuje) određeni artikl ili ne što ćemo unijeti kroz stupac “Da“ kroz potvrdni okvir, te stupac “Korisnik _ID“ u kojem će nam biti prikazani svi korisnici iz tablice “Korisnici“ kroz numeričku oznaku PK;
- tablica “Korisnici“ sadržava stupac “ID_Korisnika“ koji je označen kao stupac za samonumeriranje i bit će nam PK, stupac “NazivKorisnika“ dati će nam detaljan naziv o kome se to radi, te stupac “BrojDjelatnika“ u koji upisujemo broj koliko djelatnika je zaposleno na poslovima za koje je potrebno koristiti zaštitna sredstva iz čega će naša aplikacija u kasnijem djelu izračunati odnose i predložiti nam količinu određenih sredstava koja bi trebala biti dodijeljena toj jedinici naše tvrtke ili poslovnog subjekta;
- tablica “Raspodjela“ će biti jedna od važnijih tablica naše aplikacije stoga što njezini sastavni elementi će biti stupci “ID_RaspodjelaKorisnik“ i “Raspodjela_ID“ koji će nam označavati vanjski ključ (VK) raspodjele po pojedinom korisniku, u stupac “Artikl_ID“ biti će nam upisani VK artikla koji su bili raspodijeljeni, u stupac “Korisnik_ID“ će nam biti upisani VK korisnika koji su sudjelovali u raspodjeli, u stupac “Godina“ biti će upisana godina u kojoj se obavila određena raspodjela, s time da postoji mogućnost upisivanja cjelovitog datuma ali to u našem slučaju neće biti

potrebno pa ćemo se zadovoljiti samo s godinom raspodjele, u stupac „“Da“ biti će upisano dali je neki korisnik sudjelovao u raspodjeli određenog artikla kroz potvrdni okvir, stupac “KolicinaTraženo“ prikazivat će nam količinu koju su kao potrebu iskazali korisnici sredstava, stupac “KolicinaPripadajuce“ prikazivat će nam količinu izračunatu i predloženu od strane aplikacije iskazanu u numeričkom obliku na dvije decimale, u stupac “KolicinaDodijeljeno“ biti će upisan podatak o stvarno dodijeljenoj količini nekom korisniku, budući da je poznato kako većinu artikala kao što su npr. zaštitne naočale vrlo teško mogli podijeliti na 21,88 komada nekom od korisnika, te posljednji stupac ove tablice “Zakljuceno“ kroz potvrdni okvir će nam prikazati dali smo s tom raspodjelom uistinu zaključili;

- tablica “Raspodjele“ osim što se od prethodne razlikuje u posljednjem slovu sadržajno se razlikuje zbog stupca “ID_Raspodjele“ koji nam označava PK raspodjele, “DatumRaspodjele“ nam označava datum kada je zahtjev od pojedinog korisnika dostavljen, “NalogBroj“ označava urudžbeni broj zahtjeva od strane korisnika, “Godina“ označava godinu raspodjele, “Artikl_ID“ nam označava VK artikla koji je raspodijeljen;
- tablica “Zahtjevi“ označava nam tablicu koja će sadržavati podatke iz zahtjeva korisnika i isto evidentirati kroz polja: “ID_Zahtjeva“, “DatumZahtjeva“, “BrojZahtjeva“, “Korisnik_ID“ te “Realizirano“ gdje će kroz potvrdni okvir biti evidentirano dali je određeni zahtjev realiziran;
- tablica “Zahtjevi_Artikli“ biti će “sumarna“ tablica u koju će biti upisano kroz polja “ID_ZahtjevArtikli“, „“Godina“, “Zahtjev_ID“, “Korisnik_ID“, “Artikl_ID“ i “Količina“ ono što smo stvarno dodijelili korisnicima.

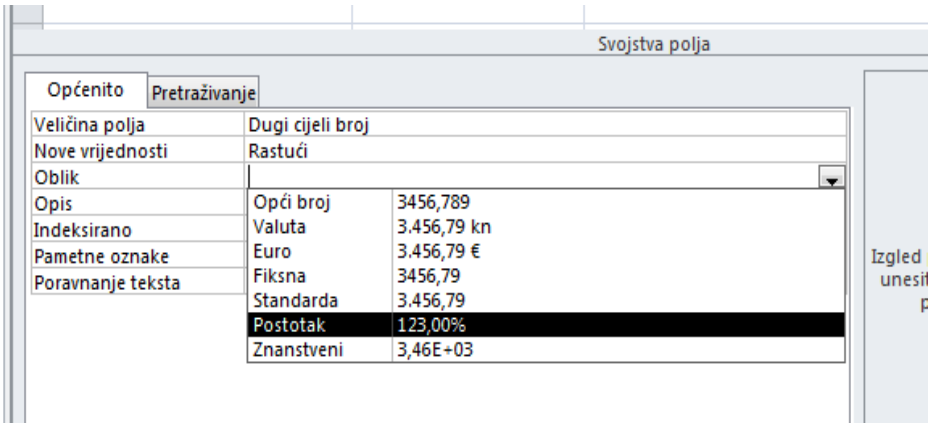
6.3.2. Izrada tablice kroz razvojnu strukturu dizajna podataka

Kada izrađujemo ili dorađujemo već postojeću tablicu u obliku podatkovnog lista tada možemo zadati ostale parametre, ovisno o potrebi, koji bi se odnosili na podatke iz naše tablice.



Slika 24. – Prikaz razvojnog oblika tablice kroz prikaz dizajna

Tim podacima možemo predodrediti veličinu polja, nove vrijednosti, oblik, indeksiranost na način dali da dozvoli duple unose iste vrijednosti ili ne, neke od pametnih oznaka koje se mogu odnositi na zakazivanje sastanaka ili označavanja na našem kalendaru na način da ih potakne unos nekog datuma, imena osobe ili telefonskog broja u našu tablicu koju izrađujemo.



Slika 25. – Prikaz mogućnosti za oblikovanje podataka

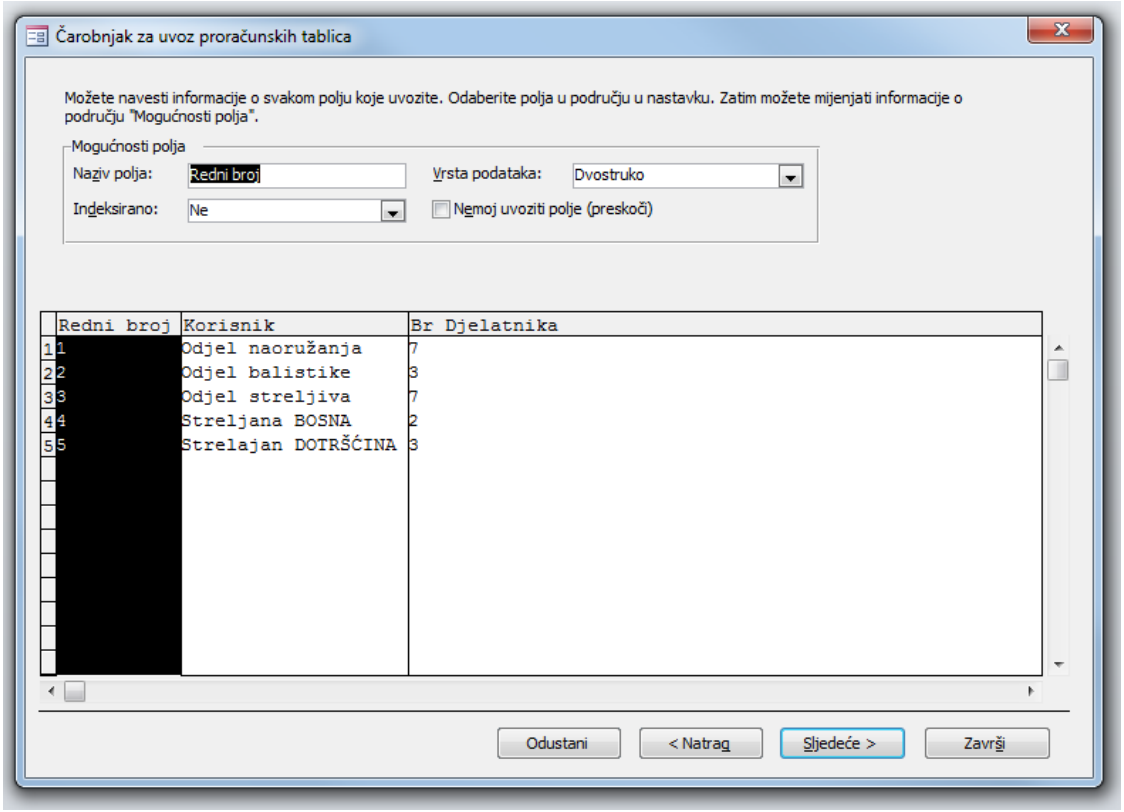
Također i u ovom obliku možemo obaviti razne izmjene kao što je izmjena naziva polja ili vrstu podatka koje nam to polje predstavlja.

6.3.3. Uvoz tablice iz vanjskih izvora

Da bi smo uvezli podatke iz nekog vanjskog izvora ili odredišta podataka potrebno je odabrati karticu Vanjski podaci gdje nam se nudi uvoz i povezivanje s preko nekoliko vrsta i oblika podataka. Nama u slučaju izrade ove baze podataka najzanimljiviji su oni u MS Excel obliku, pošto postoji vjerojatnost da je netko već ranije izradio nekakav oblik pomoćne tablice čiji podaci bi odgovarali u nekom djelu ili možda čak u cijelosti bazi podataka koju izrađujemo. U ovom našem konkretnom slučaju izrade "baze podataka", kako bi izbjegli ponovnu izradu tablice pod nazivom "Korisnici" koja je već ranije bila izrađena u MS Excel obliku, od strane korisnika, koja sadrži stupce čiji sadržaj označava jedinstvenu numeraciju svake od ustrojstvenih jedinica unutar tvrtke, zajedno s njihovim cjelovitim nazivima i brojem zaposlenika u svakoj od njih, te nam također odgovara i sam raspored stupaca unutar tablice, istu ćemo "uvesti" u našu bazu podataka. Kod iniciranja uvoza Excel podataka sustav nam automatski ponudi da izaberemo lokaciju smještaja datoteke, odnosno izvor podataka koji želimo prebaciti u naš program kroz nekoliko načina:

- uvezi izvorišne podatke u novu tablicu u trenutnoj bazi podataka,
- dodaj kopiju zapisa u tablicu (po našoj želji),
- poveži s izvorom podataka stvaranjem povezane tablice.

Budući da ne želimo narušiti integritet i sadržaj dosadašnje tablice, u našem slučaju sada nam izvorišne baze podataka, niti želimo nepotrebno opteretiti sustav s još jednom redundantnom tablicom odabiremo karticu "Vanjski podaci", nakon čega u grupi "Uvoz i povezivanje" odaberemo ikonu "Excel". Nakon odabranog načina uvoza podataka otvara nam se "Čarobnjak za uvoz proračunskih tablica" (Slika 26.) gdje možemo odabrati određeni raspon podataka ili neki od radnih listova. Nakon odabira određenog lista ili raspona podataka čarobnjak nas pita dali naša tablica koju smo odabrali u prvom retku sadrži zaglavlja (nazive) stupaca, u koliko da potrebno je u za to određenu "kućicu" staviti potvrdni znak jer u protivnom prilikom prebacivanja podataka doći će do greške pogotovo u koliko nam stupac sadrži nekakve numeričke znakove a sam naziv se sastoji od slovnih oznaka.



Slika 26. – Izgled Čarobnjaka za uvoz podataka u jednom segmentu

Kada smo to označili (ili nismo) ovisno o podacima u tablici za svako polje koje ćemo prebaciti sustav nam nudi izmjenu naziva polja, vrstu podataka, indeksiranje vezano uz duplikate te “kućicu“ koju je potrebno potvrdno označiti u koliko uopće ne želimo da nam se podaci iz tog polja uvezu u našu tablicu. Na kraju svih gore navedenih koraka čarobnjak nas pita za određivanje primarnog ključa i nudi nam opcije:

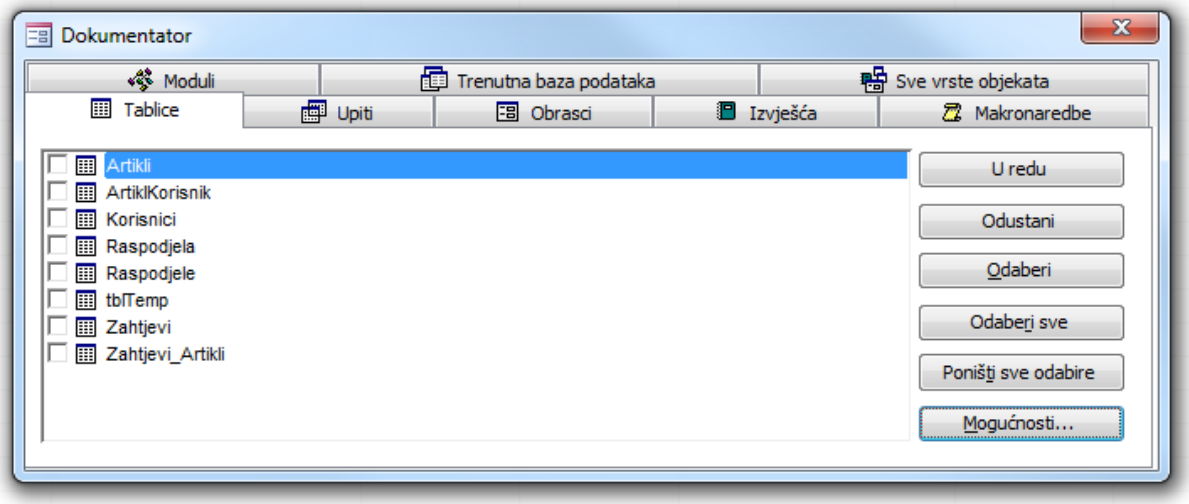
- neka Access doda primarni ključ,
- sam ću odrediti primarni ključ,
- bez primarnog ključa.

Po završetku gore navedene mogućnost izmjene naziva tablice na način da nam nudi opciju Uvezi u tablicu, te također nam nudi da čarobnjak obavi analizu tablice nakon uvoza podataka.

6.3.4. Kodeksa atributa (definicija objekta)

Pregledom kodeksa može se saznati gotovo sve o našoj ili nekoj drugoj aplikaciji. Taj pregled nam omogućava uvid u samu srž vrste i tipa podataka koji se nalaze unutar aplikacije, među ostalim prikazuje veze i njene karakteristike, prikazuje vrstu

podataka, veličinu prostora koja je predviđena da ju zauzme određeni podatak u našoj bazi, i td.. Do podataka tog tipa u Access-u se dolazi na način da u karici “Alati baze podataka“ u odjeljku “Analiza“ odaberemo funkciju “Dokumentator“ baze podataka“ nakon čega će nam se otvoriti “čarobnjak“, koji je prikazan na Slici 27., u kojem ćemo odabrati ono što nas zanima a to je kartica “Tablice“ gdje ćemo kroz potvrdne okvire odabrati one tablice koje nas zanimaju.



Slika 27. – Izgled “čarobnjaka“ Dokumentatora

Nakon što smo označili tablice i odabrali ono što nas zanima sadržaj bi trebao izgledati kako je prikazano u Prilogu B.

6.4. Obrasci

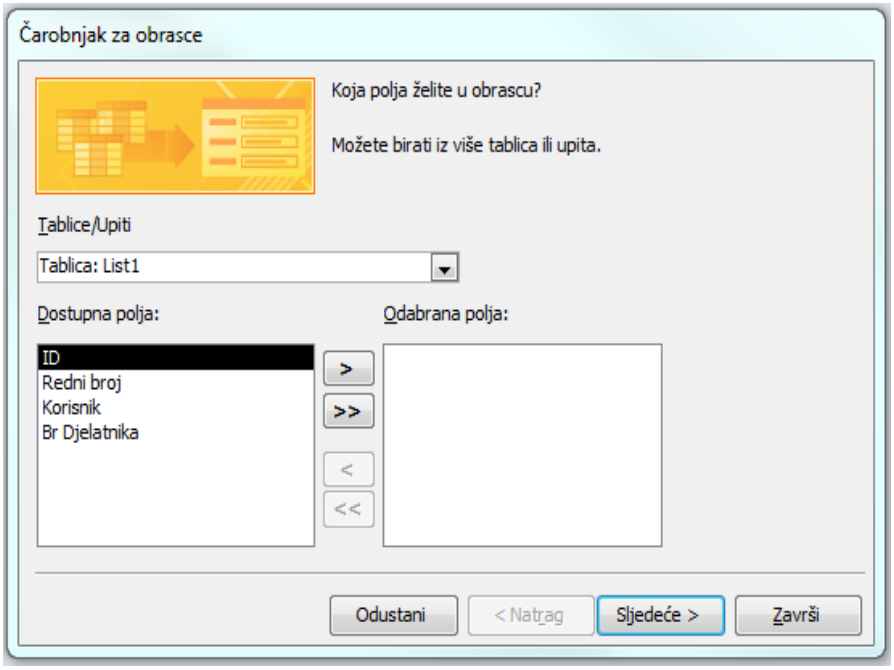
Kako bi olakšali korisniku rad s aplikacijom i da bi se što duže sačuvao integritet same baze podataka potrebno je onemogućiti korisnika da unosi ili ispravlja podatke u samim tablicama, s time da je i najpraktičnije korisniku da ne koristi cijelu tablicu u njenom tabličnom prikazu nego samo ona polja koja su mu potrebna prilikom upisa određenog oblika podataka.

Access nam za takve potrebe omogućava izradu i oblikovanje posebnog podatkovnog oblika koje nazivamo Obrasci a oni su najbolje i najlakše grafičko okruženje za korisnika koje realno prikazuje sve potrebne podatke za koje je nužno obaviti unos. Obrasci mogu biti grupirani i prema vrstama podataka koje sadrže te im se može pristupiti i s osnovnog sučelja.

6.4.1. Izrada obrasca

Prilikom izrade obrasca možemo se koristiti Čarobnjakom za obrasce (Slika 28.) ili kroz neki drugi oblik kao što je Prazan obrazac ili neki od Dodatnih oblika obrasca izraditi svoj specifičan obrazac sukladno našim potrebama. Radi lakšeg rada i snalaženja prilikom izrade obrasca, kada god to mogućnosti dopuštaju, koristim se Čarobnjakom za obrasce.

Bitno je napomenuti da u koliko izrađujemo obrazac koji će se odnositi na više tablica istovremeno, te tablice moraju imati zadan nekakav međusobni odnos, moraju biti povezane ili nećemo moći izraditi funkcionalan obrazac.

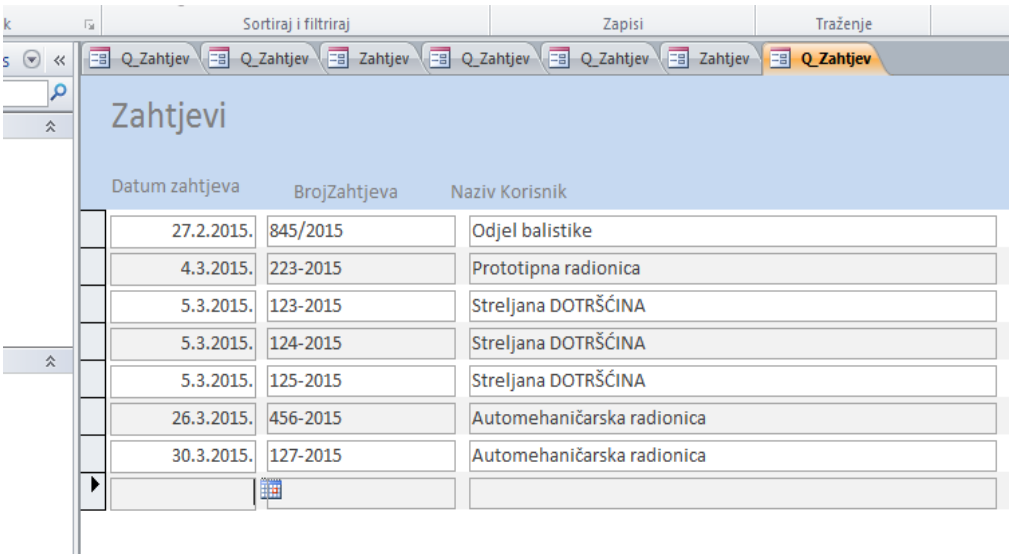


Slika 28. – Izgled Čarobnjaka za obrasce podataka u jednom segmentu

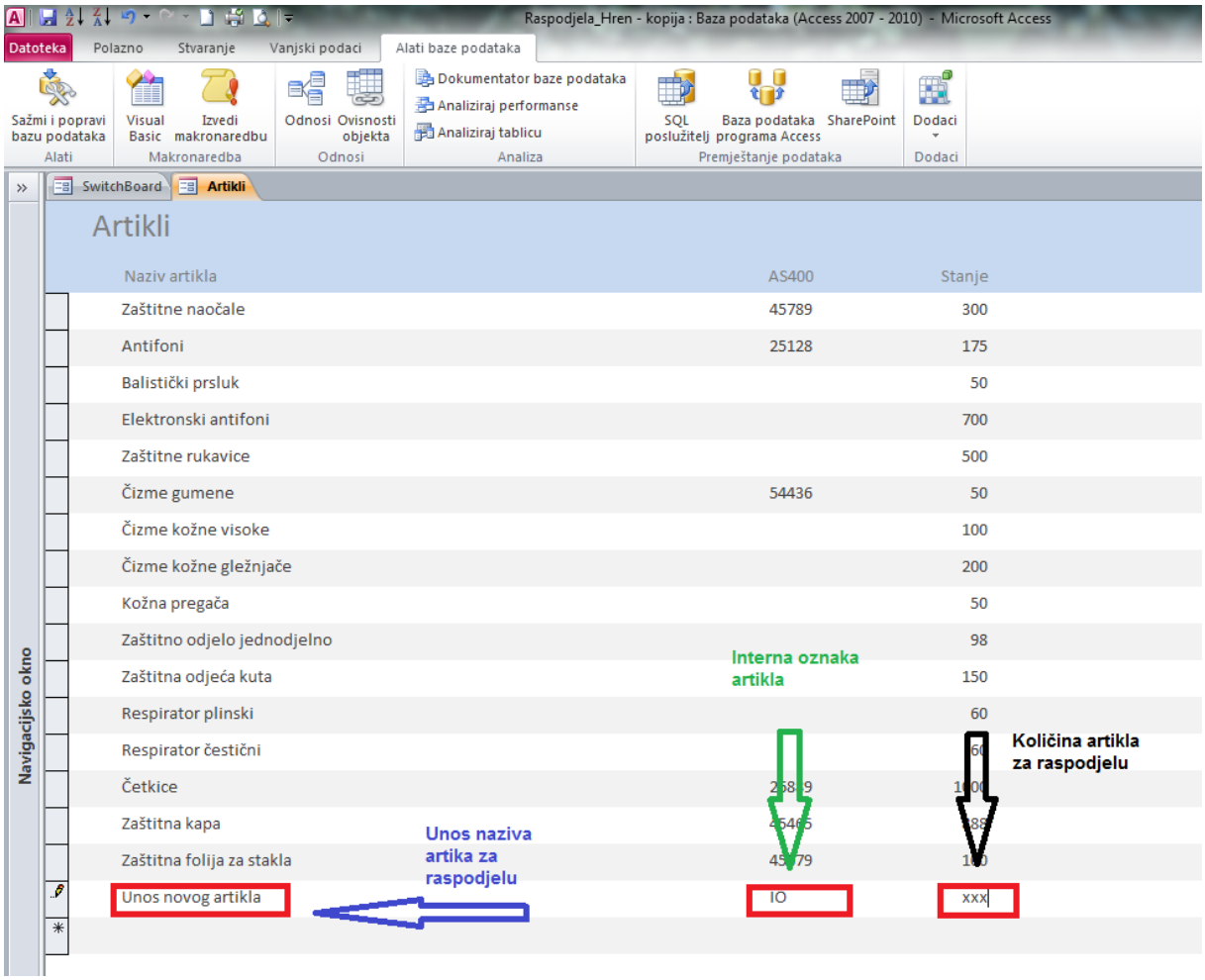
Nakon odabira izvora tablica i odabira polja koja želimo da naš obrazac sadrži postavlja nam se pitanje kakav izgled obrasca želimo dali da isti izgleda Stupčasto, Tablično kao Podatkovna tablica ili Obostrano. Po završetku odabira podataka, izgleda i oblika obrasca, sustav nam automatski ponudi mogućnost da upišemo naziv obrasca te nam nudi mogućnosti da otvori obrazac za prikaz ili unos podataka ili da izmijenimo dizajn obrasca. U koliko smo zadovoljni svime što smo naveli odabiremo mogućnost Završi što nam gotovo istovremeno omogućava da imamo prikaz našeg novonastalog obrasca na sučelju.

6.4.2. Unos podataka u obrasce

Sama procedura oko unosa podataka u obrasce je vrlo jednostavna stoga što se bazira na popunjavanju za to predviđenih polja (Slika 29. i Slika 30.). Prilikom obavljanja popunjavanja polja mogu se izraditi različite mogućnosti kao što su ta da sustav pamti zadnje unesene parametre na tom obrascu osim nekog “ključnog podatka“ kao što je na primjer serijski ili inventarski broj nekog predmeta, u koliko je datum izrade ili oznaka modela jednaka itd.



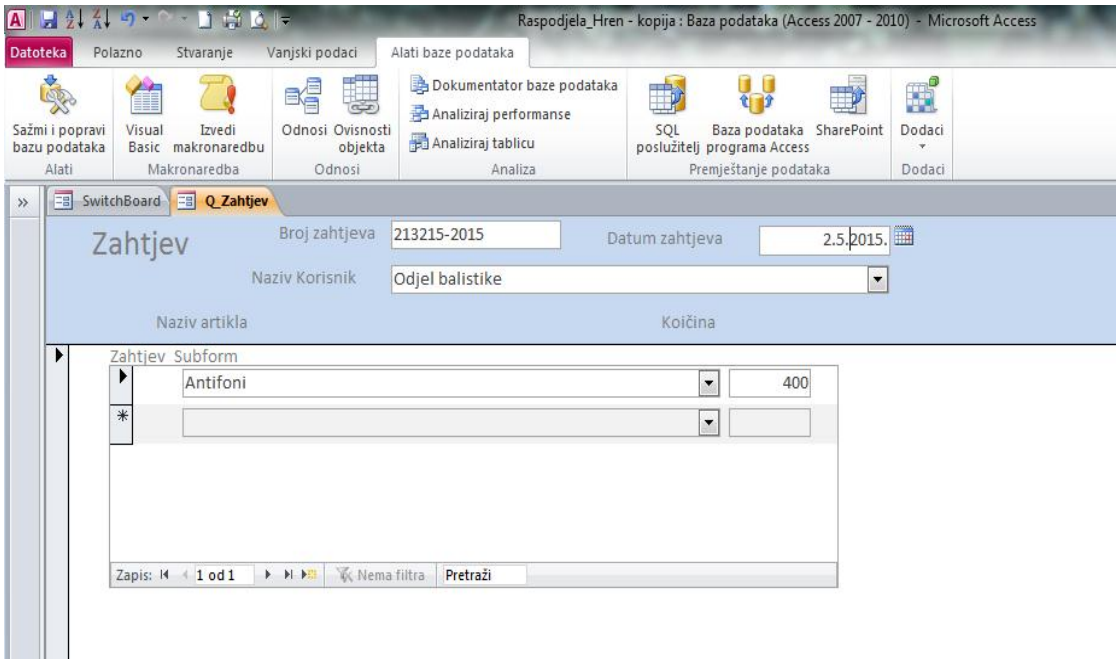
Slika 29. – Izgled obrazaca za unos Zahtjeva korisnika



Slika 30. – Izgled obrazaca za unos Artikla i njihove količine za raspodjelu

Postoje mnogobrojne mogućnosti za lakši unos podataka, kako bi ubrzali sam unos, sačuvali jednoobraznost podataka koja će nam olakšati daljnji rad s bazom kao što je na primjer mogućnost padajućeg izbornika koji će nam ponuditi samo unaprijed definirane oblike podataka, pojedini su vidljivi na Slici 31., kako bi smanjili mogućnost pogreške prilikom unosa i sl. Također možemo postaviti obrasce na način da se isti popunjavaju po određenim razinama odnosno da tek po cjelovitom popunjavanju prethodnog oblika obrasca nam se otvori sljedeći itd. U koliko nismo zadovoljni izgledom obrasca kroz različite načine ga možemo dorađivati. Možemo mijenjati font, boju, oblik, veličinu polja, slova i gotovo svih ostalih segmenata od kojih se sastoji obrazac.

U koliko će nam navedeni obrazac koristiti za neke druge baze kao što su podaci o zaposlenicima ili sl. možemo postaviti u njega i sliku zaposlenika (ili neke opreme), možemo napraviti raspodjelu polja unutar obrasca ili njihovo poravnanje itd.



Slika 31. – Izgled obrazaca za unos podataka iz Zahtjeva korisnika

6.5. Upiti

Sam Access nudi gotovo neograničene mogućnosti za dizajn upita koji se prilagođavaju potrebama i željama krajnjeg korisnika, povezani su s izvješćima tako

da se rezultat svakog upita prikazuje putem izvješća. Isto tako rezultati koji su dobiveni iz nekog upita mogu se koristiti kao izvor podataka za obrasce i izvještaje.

Oni služe kao posredni izvor podataka, kako bismo došli do podataka koji su nama potrebni, kroz određeni oblik procedure koju smo mu zadali. Te procedure mogu sadržavati razne oblike filtriranja, pretraživanja, grupiranja, zbrajanja podataka, praćenje njihovih odnosa pa čak do te razine da izrađuju nove zapise ili brišu postojeće. Upiti su sekundarni izvori podataka i temelja za obrasce koji prikazuju njihove rezultate na sučelju ili kroz izvještaje.

U koliko koristimo Čarobnjak za upite tada dobijemo mogućnost za izradu:

- jednostavnog upita,
- unakrsnog upita,
- pronalaženje dvostrukih upita,
- pronalaženje ne uparenih upita.

Također, kao i u prethodnim slučajevima, ukoliko se izrađuje upit koji se odnosi na podatke iz dvije ili više tablica, iste moraju biti međusobno povezane. Nakon odabira željenog oblika koji nam je ponudio Čarobnjak postavlja nam se pitanje dali želimo detaljan koji sadrži sva polja ili sažeti oblik upita koji nam može pružiti zbroj, prosječnu veličinu, maksimalnu ili minimalnu veličinu itd.

6.5.1. Oblici upita

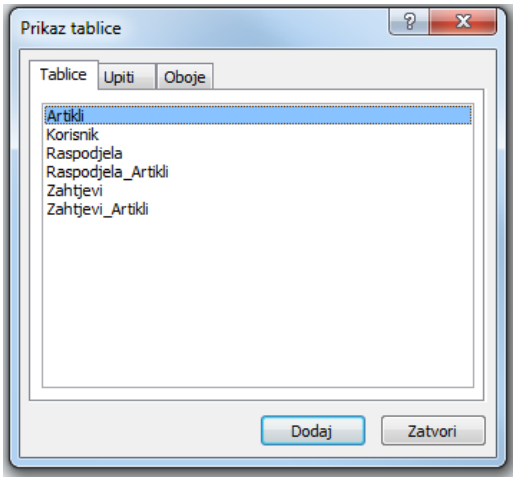
MS Access podržava slijedeće oblike upita:

- (*engl. Select query*) – je najčešći i najjednostavniji oblik upita kojim pregledavaju i analiziraju podaci iz tablica;
- (*engl*. *Crosstab query*) – u ovom obliku podaci su grupirani u kolone i redove na način koji je uobičajen za unakrsne tablice. Ovaj oblik je pogodan za sažimanje velikog broja informacija u lako preglednom obliku;
- (*engl*. *Action query*) – je jedinstvena vrsta upita koji mijenja veću količinu podataka jednom operacijom, dok osnovni tipovi akcijskih upita su:
  - (*engl. Make table query*) – nam omogućava izradu novih tablica izborom polja iz postojećih tablica,
  - (*engl. Delete query*) – pomoću kojeg brišemo podatke iz tablica, prema zadanim uvjetima,

    - (*engl. Append query*) – obavlja funkciju dodavanja izabranih podataka jedne tablice drugoj,
    - (*engl. Update query*) – mijenja (zanavlja) sadržaj odabranih polja u tablici;
- (*engl. Union query*) – nam omogućava nadovezivanje koristeći odgovarajuća polja iz više tablica ili upita u jedinstveno dobiveni oblik podataka;
- (*engl. SQL specific*) – je vrsta upita koji je karakterističan za SQL jezik koji se ne oblikuju u standardnom Prikazu dizajna nego u SQL prikazu;
- (*engl. Pass-through query*) – šalje naloge (zapovjedi) na SQL DataBase server te omogućava da se s tablicama radi preko servera;
- (*engl. Data-definition query*) – je vrlo sličan tipu (action query) a on nam omogućava mijenjanje i oblikovanje novih tablica u bazi.

6.5.2. Izrada upita

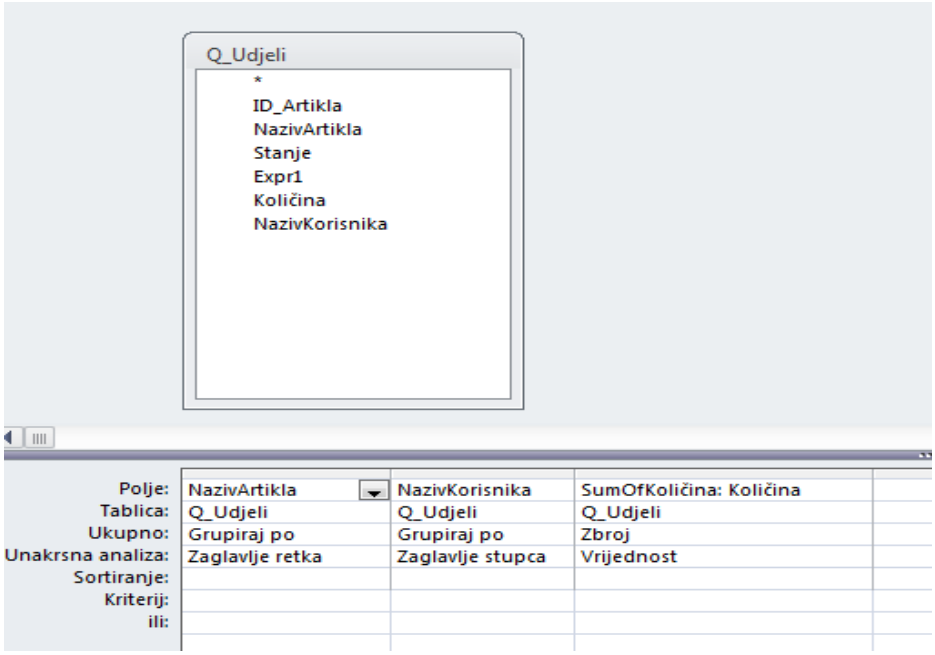
Pri izradi novog upita automatizmom se otvara dijaloški okvir (Slika 32.) u kojem se nalazi popis svih tablica ili prethodno pohranjene upite koje koristimo te nam isti mogu poslužiti kao izvor podataka. Nakon što smo odabrali željeni izvor ili izvore podataka odabiremo pojedina polja koja želimo koristiti u upitu.



Slika 32. – Prikaz dijaloškog okvira pri izradi upita

Najjednostavniji postupak kojim možemo to odraditi je da uz pomoć povuci i pusti metode (*engl. drag and drop*) ili na drugi način koji uključuje rad uz pomoć izbornika

koji se pojavi kada kliknemo u jedno od polja. U ovom slučaju dodat ćemo NazivArtikla, NazivKorisnika i Sumu(iz)Količine, kao što je prikazano na Slici 33..



Slika 33. – Odabir polja i funkcija kod izrade upita

Nakon što smo odredili što nam treba tada možemo i zadati neki od kriterija koji nam je potreban na način da prilikom upisivanja tekstualnih kriterija koristimo se navodnim znacima.

Prilikom korištenja nekog od uvjeta pretraživanja, u koliko pretražujemo podatke iz više tablica, iste moraju biti međusobno povezane. Kod postavljanja uvjeta upita potrebno je poznavati određene znakove (operatore) kao što su:

= - znak jednakosti,

< > - znakovi za različito od,

< - znak manje od,

> - znak veće od,

< = - znak za manje ili jednako od,

> = - znak za veće ili jednako od. [15]

Također možemo koristiti i neke od logičkih operatora kao što su:

I – kojim zadajemo više traženih opcija,

ILI – kojim se određujemo između nekih opcija,

NE- kojim zadajemo kriterij koji želimo izostaviti.

Isto tako da bi smo pronašli ili zamijenili neki od podataka iz naše baze, kroz dijalog Traži i zamjeni, možemo koristiti i pojedine oblike džoker znakova, kao što su:

* koji nam može zamijeniti bilo koji broj znakova naše nepoznanice ili neke riječi, npr.: u koliko u našoj bazi imamo podatke registracijskih pločica vozila i mi smo zapamtilo samo posljednju slovnu oznaku *BO aplikacija će pronaći ZG1234BO ili KA123BO;

? se može umetnuti na mjesto bilo koje slovne oznake,

[ ] podudarat će se sa bilo kojim znakom u zagradama,

! podudarat će se sa bilo kojim znakom koji nije u zagradama,

- podudara se sa bilo kojim znakom u opsegu s uvjetom da opseg mora biti naveden u rastućem nizu;

će se podudarati s bilo kojim numeričkom oznakom koja nam nedostaje u određenom dijelu broja.

6.5.3. Pokretanje upita

Kako bismo pokrenuli upit da nam odradi ono što smo mu zadali, u koliko smo to zadali na ispravan način i da nam se ne pojavljuju slučajne pogreške u zadanim kriterijima moramo ga pokrenuti iz odjeljka Alati upita na kartici Dizajn u grupi Rezultati, pritiskom na naredbu Izvedi.

Rezultati će nam biti prikazani u podatkovnoj tablici na način kako su odabrani i posloženi iz zadanog upita (iz jedne ili više tablica) u obliku tablice u kojoj nije ograničen (nedozvoljen) rad nego ih možemo mijenjati ili unositi nove podatke u istu.

6.6. Izvješća

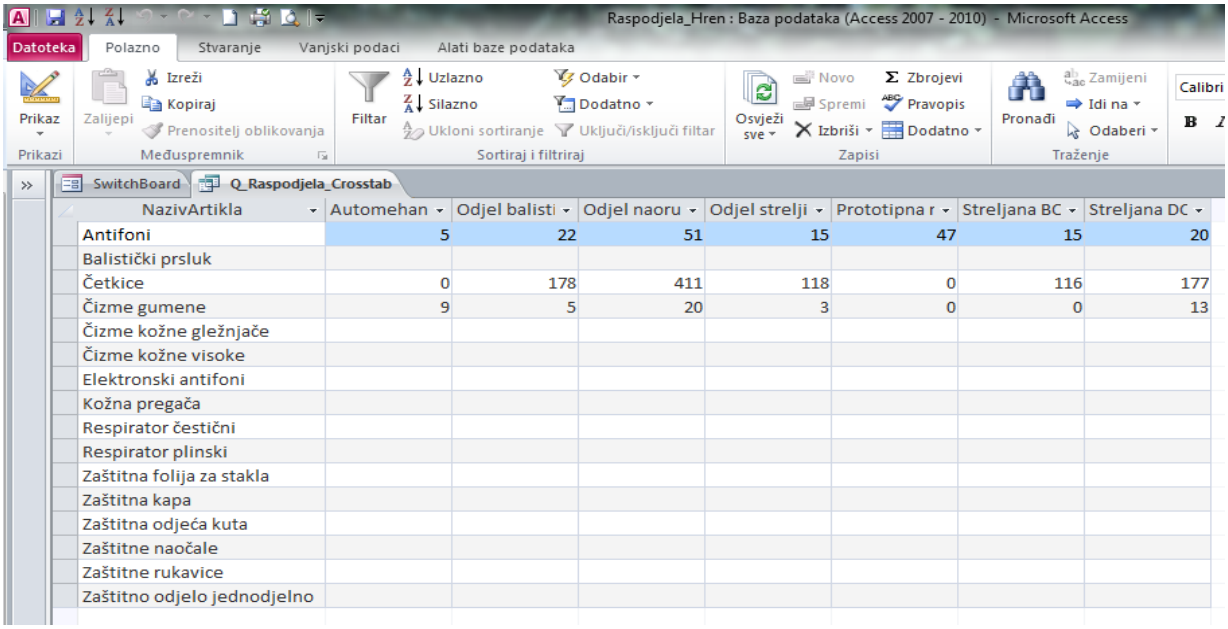
Ako se prisjetimo da prilikom samog oblikovanja baze podataka, jedan od osnovnih koraka je bio da se točno definira ona vrsta i tip podataka koja nam je potrebna, te da je to kriterij i radi čega uopće izrađujemo bazu podataka, odnosno ti podaci bi trebali biti sadržani u nekom od izvještaja koje ćemo izraditi. Dobiveni podaci iz izvještaja mogu biti prikazani na zaslonu u obliku prikaza dizajna (obrasca) ili ispisani pomoću pisaća. Izvještaji nam omogućuju da se prikaz dobivenih podataka formira na način koji je nama najprikladniji, da se obave određeni oblici kalkulacija, grupiranje podataka, vizualiziraju podaci uz pomoć dijagrama itd.

6.6.1. Izrada izvještaja

Ukoliko imamo označenu neku od tablica ili neki upit, kao i kod obrazaca, kada pritisnemo lijevu tipku miša na funkciju Izvješće u kartici Stvaranje aplikacija će nam sama generirati izvještaj nakon čega, u koliko želimo možemo koristiti Čarobnjak za izvještaje ili sam dizajnirati svoj vlastiti oblik izvještaja kroz Dizajn izvješća.

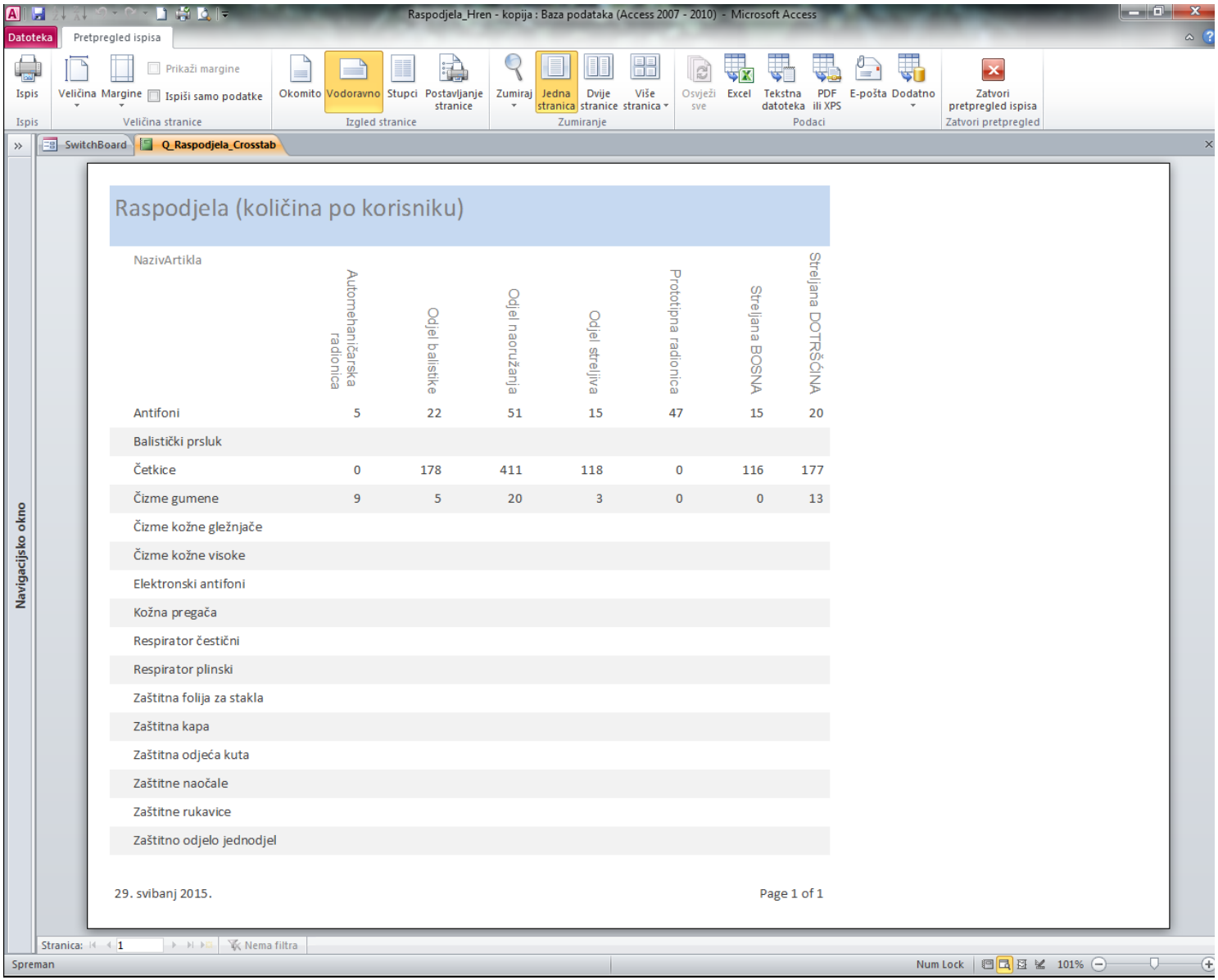
Samo mijenjanje rasporeda polja s podacima najlakše obavljamo uz pomoć metode povuci i pusti u obliku Prikaz dizajna. U koliko želimo da naš izvještaj ima neko polje za zbroj, prosjek, broj redaka i sl., pritiskom na gumb Zbrojevi na kartici Dizajn, u Podnožju izvještaja pojavit će se polje sa odabranom funkcijom koje onda možemo premjestiti i postaviti na poziciju koja nam odgovara.

Ovisno o željama korisnika izvještaj možemo prikazati i u formi tablice Upita, koju kasnije možemo preoblikovati u konačan oblik izvještaja s eventualnom malom doradom na samim podacima (sakrivanje pojedinih dijelova ili njihova vizualna dorada).



NazivArtikla	Automehan	Odjel balisti	Odjel naoru	Odjel strelji	Prototipna r	Streljana BC	Streljana DC
Antifoni	5	22	51	15	47	15	20
Balistički prsluk							
Četkice	0	178	411	118	0	116	177
Čizme gumene	9	5	20	3	0	0	13
Čizme kožne gležnjače							
Čizme kožne visoke							
Elektronski antifoni							
Kožna pregača							
Respirator čestični							
Respirator plinski							
Zaštitna folija za stakla							
Zaštitna kapa							
Zaštitna odjeća kuta							
Zaštitne naočale							
Zaštitne rukavice							
Zaštitno odjelo jednodjelno							

Slika 34. – Izgled izvještaja u “tabličnom obliku“



Slika 35. – Izgled izvještaja Slike 34. u "Print preview" obliku Pretpregleda

6.6.2. Kontrole

Postoje tri vrste objektnih kontrola (*engl. Controls*) koje se mogu nalaziti na izvještaju:

- vezane kontrole (*engl. bound control*) – označava objekt koji je povezan s nekim od polja u tablici ili upitu a isti se koristi u izvještaju a oni mogu sadržavati podatke u obliku teksta, brojeva, datum i vremena, logičke podatke…;
- nevezane kontrole (*engl. unbond control*) – sadrži neki od oblika koji nije povezan sa sadržajem u tablicama kao što su slike, pravokutnici, linije, tekst, broj…;
- proračunske kontrole (engl. calculated control) – koji označavaju objekte čiji sadržaj je izveden iz sadržaja tablice kao što je izračun iz nekakvih odnosa i sl..

6.6.3. Izvoz rezultata upita - izvještaj

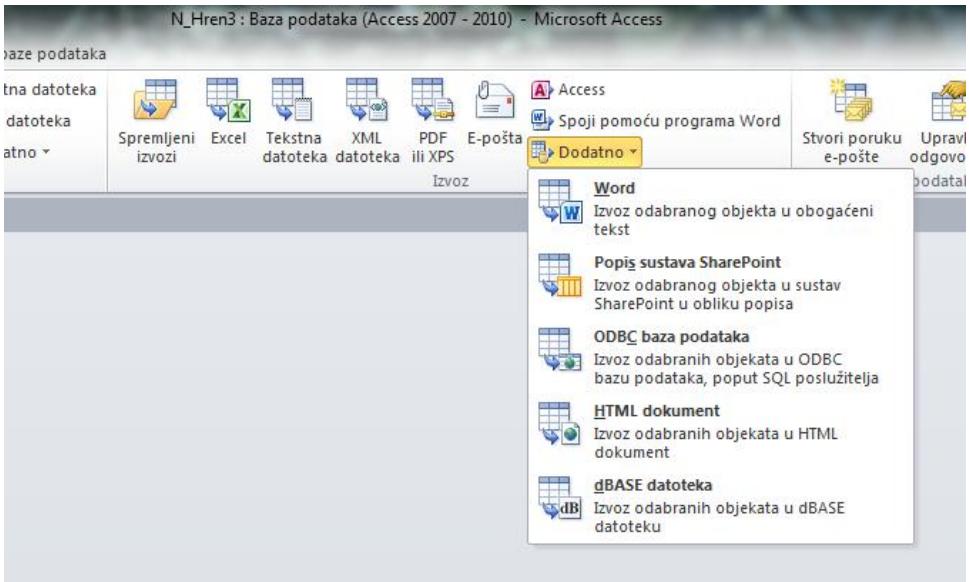
Kao što smo ranije spomenuli, podatke koje dobijemo iz upita možemo preoblikovati u neki drugi, korisniku primjereniji oblik. Da bismo to učinili potrebno je koristiti za to već predviđene alate koji se nalaze u grupi Izvoz na kartici Vanjski podaci gdje pritiskom na gumb Pretraži odabiremo lokaciju za smještaj tog radnog lista koji ćemo izvesti.

Odabir oblika u kojem želimo da naši podaci budu izvezeni odabiremo na kartici Vanjski podaci u grupi Izvoz (Slika 36.). Opcije među ponuđenim oblicima izvoza su u slijedećim formatima:

- Excel – oblik datoteke koji uobličuje naše podatke u proračunske tablice;
- Txt – sprema radnu knjigu kao tekstualnu datoteku razdvojenu tabulatorom za uporabu u drugom operacijskom sustavu;
- XML – (*engl. EXtensible Markup Language*) je jezik za označavanje podataka nudi pohranu podataka, koristi se određenim skupom zadanih oznaka, podaci u XML obliku se mogu pohraniti kao proračunske tablice ili XML oblik podataka;
- PDF ili XPS – oblik PDF (*engl. Portable Document Format*) nam omogućava čuvanje oblika dokumenta, omogućava nam zajedničko korištenje datoteke. Podatke datoteke u ovom obliku nije moguće lako izmijeniti, ako se datoteka u

ovom obliku ispisuje ili objavljuje putem mrežnih stranica zadržava postojeći oblik, također je koristan za dokumente koji se reproduciraju kroz komercijalne metode tiska: - oblik XPS (*engl. XML Paper Specification*) ima gotovo jednake karakteristike kao i ranije opisani PDF oblik;

- e-Pošta – opcija koja nam omogućava slanje i primanje odabranih segmenata aplikacije putem interneta;
- Word – oblik koji koristi MS sustav za obradu teksta;
- SharePoint – programska platforma s mnogobrojnim mogućnostima, oblik koji nam služi za rješavanje trenutnih poslovnih problema, sastoji se od šest osnovnih dijelova kao što su web mjesta, platforme za suradnju, sadržaj, tražilica, poslovna inteligencija i poslovna rješenja;
- ODBC – (*engl. Open DataBase Connectivity*) program za komunikaciju s bazom podataka;
- HTML – (*engl. HyperText Markup Language*) kada se kopira tekst iz nekog drugog programa, MS Excel ga lijepi u obliku HTML, neovisno o obliku izvornog teksta, u ovom obliku integriraju se umetnuta grafika, povezani dokumenti, apleti (mali Java programi) i sl.;
- dBASE – programski paket koji omogućuje određivanje strukture baze podataka, upis, promjenu, brisanje, pretraživanje, izvješćivanje, pripremu vlastitih podataka za ulaz u druge programske pakete.



Slika 36. – Mogućnosti za izvoz podataka

Prilikom izvoza podataka možemo se odlučiti za nekoliko mogućnosti kao što su:

- izvezi podatke sa oblikovanjem i rasporedom – koji nam u većoj mjeri zadržavaju početni izgled i oblikovanje podataka u tablici;
- otvori odredišne datoteke nakon završetka operacije izvoza – u koliko odaberemo ovu opciju sustav nam automatizmom otvara radni list u koji smo izvezli podatke;
- izvezi samo odabrane zapise – označava opciju koja prebacuje samo one podatke koje odaberemo a ista je dostupna samo u kombinaciji s opcijom izvoza oblikovanih podataka.

7. Analiza primjenjivosti

Za svaku tvrtku ili bilo koju drugu organizaciju uvođenje novih sustava predstavlja vrlo rizičan pothvat zbog mogućnosti da novi sustav ispravno ne profunkcionira a stari sustav se prijevremeno ukloni. Moguća su tri načina promjene starog sustava u novi:

- izravna promjena
- paralelna promjena
- fazna (postupna) promjena. [4]

Nakon određenog razdoblja, u prosjeku šest mjeseci, pristupa se procjeni funkcioniranja sustava (uspješnosti projekta) i obavlja se izrada izvješća o uspjehu projekta u kojem se obrađuju neimplementirane funkcionalnosti, uspoređuju planirane i ostvarene funkcionalnosti te nepostignuti ciljevi. Prilikom provedbe analize primjenjivosti naše baze podataka, ukoliko se ravnamo prema određenim smjernicama koja se sastoji od procjene i analize sljedećih elemenata:

- kadrovske strukture
- aplikacije
- baze podataka
- korištenje programskih jezika i CASE alata
- pokrivenosti poslovanja postojećim aplikacijama
- postojeće platforme (računalo, operacijski sustav, mreža)
- tehnologija posluživanja programa
- međusobna povezanost različitih aplikacija
- integracija podataka
- stupanj informatičke izobrazbe korisnika [4],

Ukoliko bi davali odgovore samo kroz potvrdni i negacijski oblik, to jest kroz oblik generalizacije, lako bi mogli zaključiti kako naša aplikacija u većini slučajeva zadovoljava sve potrebne kriterije te da njena funkcija ima značajnu ulogu u radu tvrtke koji se kao takav pozitivan može odraziti na cjelokupan sustav u kojem se ona nalazi. Ako nas zanima kompatibilnost Access-a s nekim velikim sustavima gdje se pomoću ODBC može “naša“ baza lako implementirati u njihov rad nije potrebno obavljati velike zahvate na samoj bazi. No, ako postoji mogućnost da će ju biti potrebno povezati s nekom drugom bazom koja nije kompatibilna s Access-om, ili je

zbog eventualne “osjetljivosti“ podataka pristup istoj restriktivan ili zbog nekog drugog razloga nema poveznosti, tada je potrebno obaviti migraciju podataka.

Kada govorimo o migraciji podataka na taj način, tada je prvo potrebno izraditi “skriptu“ za usporedbu podataka između “naše“ baze i baze podataka na koju želimo migrirati podatke. Po obavljenoj analizi koja uspoređuje međusobne podatke računalo nam najčešće ispisuje koliko je podataka “upareno“ a za sve ostale nam javlja grešku. Greške se analiziraju kako bi se utvrdilo da li su nastale zbog ljudskog faktora ili je posrijedi nešto drugo, te se ručno ispravljaju. Kada se obave svi potrebni ispravci provede se još jednom analiza te u koliko je sve u redu, novonastala baza započinje sa svojim radom.

8. Zaključak

Sigurnost u području Zaštite na radu ima svoju budućnost u svim oblicima poslovanja, kako u proizvodnji tako i u pružanju usluga, ne samo što ju propisuju normativni i pravni akti, već ju predstavlja i potreba za što kvalitetnijim životom i po završetku radnog vremena. Činjenica je da zaštita na radu kao znanstveno područje mora biti u korak s vremenom kako bi opstala i dalje se razvijala u svijetu koji sada predstavlja globalna informatizacija.

Računala, njihovi operacijski sustavi i aplikacije postala su sastavni dio našeg svakodnevnog života, tako da se informatička pismenost podrazumijeva i postala je jedan od osnovnih zahtjeva od strane poslodavca prilikom zapošljavanja, jer rad s računalom je postao osnova za gotovo svaki ozbiljniji rad.

MS Access 2010 je sustav za upravljanje bazama podataka, također predstavlja alat za prikupljanje, analizu, prikazivanje podataka te automatizaciju procesa koji se često ponavljaju. Ne može konkurirati profesionalnim bazama podataka, niti po količini podataka koje može zaprimiti, niti po njihovoj međusobnoj distribuciji između više korisnika, ali nam znatno može olakšati poslovanje u “privatnom“ poslovnom okruženju, što je i njegova prvenstvena namjena. Kao cjelovita aplikacija, Access omogućava upravljanje relacijskom bazom podataka i njenim integritetom s razine računala na kojem je smještena kao baza podataka. Za kvalitetnu izradu bilo kakvog oblika baze podataka potrebna je suradnja između najmanje dvije osobe od kojih je jedna korisnik, koji se razumije u procese za koje je potrebno izraditi bazu podataka, dok je druga informatičar, koji će znati kako pretočiti potrebe i zahtjeve korisnika u odgovarajući informatički oblik koji je potreban za izradu potencijalne baze podataka.

U koliko se baza podataka pravovremeno nadograđuje i održava zadovoljit će se još jedan preduvjet u cjelokupnom ciklusu njenog razvoja, a to je da će istu biti moguće integrirati u neki veći sustav, a sve u cilju da se izradi snažan i cjelovit sustav za podršku poslovanja našeg korisnika ili više njih. Budući da izrađena baza podataka omogućava unos, pregled, pretraživanje i određeni oblik vizualizacije podataka te omogućava i jednostavno postavljanje već ranije definiranih upita, u čemu i jest najveća prednost aplikacija zasnovanih na bazi podataka, korisnik koji se njome služi ne treba biti informatički stručnjak da bi došao do podataka koji su mu potrebni. Za uspješno rukovanje potreban je tek relativno kratak period obuke i uvježbavanja.

Također je bitno napomenuti kako se baza podataka prikazana u ovom radu uz vrlo male preinake može koristiti za raspoređivanje bilo kakvih drugih sredstava (novca, razne opreme, naoružanja, streljiva i sl.).

9. Literatura

[1] ZAKON HR “Zakon o zaštiti na radu“ (pročišćeni tekst), dostupno na: http://www.zakon.hr/z/167/Zakon-o-za%C5%A1titi-na-radu (9. ožujka 2015.)

[2] Preventa, Centar za integralnu sigurnost d.o.o. “Propisi zaštite na radu“, dostupno na: http://www.preventa.hr/dodatak/propisi/propisi-zastite-na-radu.html (11. ožujak 2015.)

[3] Narodne novine “Pravilnik o uporabi osobnih zaštitnih sredstava“ (NN39/06), dostupno na: http://narodne-novine.nn.hr/clanci/sluzbeni/2006_04_39_958.html (11. ožujak 2015.)

[4] Pavlić, M., Informacijski sustavi, Odjel za informatiku Sveučilišta u Rijeci, Digital Point tiskara d.o.o. Rijeka, 2009.g., ISBN 978-953-96970-7-3.

[5] Bosilj Vukšić, V., Hernaus, T., Kovačić, A., Upravljanje poslovnim procesima – Organizacijski i informacijski pristup, Školska knjiga d.d. Zagreb, 2008.g., ISBN 978-953-30348-5.

[6] Wikipedia, Microsoft Jet Database Engine, dostupno na: http://en.wikipedia.org/wiki/Microsoft_Jet_Database_Engine (06.05. 2015.)

[7] Mileusnić, V., Završni rad, Modeliranje i dizajn baze podataka, FOI, Varaždin 2011., dostupno na: https://moodle.srce.hr/eportfolio/artefact/file/download.php?file=31207&view=7460 (2.svibnja 2015.)

[8] Teorey, T., Lightstone, S., Nadeau, T., Database Modeling & Design: Local Design, Fourth edition, Morgan Kaufmann Publishers, ISBN 13: 978-0-12-685352-0.

[9] Beynon-Davies, P., Database systems, Third edition, Palgrave Macmillan, 2004.g., tisak i uvez u Kini, ISBN 1-4039-1601-2.

[10] Kralj, D., Skripta, Upravljanje ZNR i ZOP primjenom računala, Veleučilište u Karlovcu, 2014. g.

[11] Manger, R., Skripta, Baze podataka, Prvo izdanje, Prirodoslovno Matematički Fakultet, Zagreb, 2003.g.

[12] Jelenković, L., i dr., Operacijski sustavi – uvod, Algebra, Zagreb, 2010.g.

[13] Wikipedia, Microsoft Access, dostupno na: http://hr.wikipedia.org/wiki/Microsoft_Access (30.04.2015)

[14] Wikipedia, SQL Server, dostupno na: http://hr.wikipedia.org/wiki/Microsoft_SQL_Server (05.05.2015.)

[15] Čelebić, G., Dujlo, M., Priručnik, Baze podataka – Microsoft Access 2010, ODRAZI, Zagreb, 2011.g., ISBN 978-953-56758-4-6.

[16] CARNet CERT, Sigurnost sustava za upravljanje bazama podataka, PUBDOC-2006-10-171, dostupno na: http://www.cis.hr/www.edicija/LinkedDocuments/CCERT-PUBDOC-2006-11-171.pdf (20. ožujka 2015.)

[17] Techopedia, ACCDB File Format, dostupno na: http://www.techopedia.com/definition/22/accdb-file-format (30. travanj 2015.)

[18] About, Structured Query Language (SQL), dostupno na: http://databases.about.com/od/sql/a/What-Is-Sql.htm (15. ožujak 2015.)

10. Prilozi

Prilog A

Entitet: Korisnik

Redni broj	Naziv atributa	Opis	Vrsta podatka	Veličina polja	Ograničenja
1	ID_Korisnika	Jedinstvena brojčana oznaka Korisnika	Samonumeriranje	Dugi cijeli broj	Da (bez duplikata)
2	NazivKorisnika	Cjeloviti naziv Korisnika	Tekst	255	Nema
3	BrojDjelatnika	Količina zaposlenih djelatnika pri korisniku	Broj	Dugi cijeli broj	Nema

Entitet: Artikli

Redni broj	Naziv atributa	Opis	Vrsta podatka	Veličina polja	Ograničenja
1	ID_Artikla	Jedinstvena brojčana oznaka Artikla	Samonumeriranje	Dugi cijeli broj	Da (bez duplikata)
2	NazivArtikla	Cjeloviti naziv Artikla	Tekst	255	Nema
3	AS400	Interna oznaka u sustavu	Tekst	6	Nema
4	Stanje	Količina sredstava namijenjena raspodjeli	Broj	Jednostruko	Nema

Entitet: Artikl_Korisnik

Redni broj	Naziv atributa	Opis	Vrsta podatka	Veličina polja	Ograničenja
1	ID_AK	Jedinstvena brojčana oznaka Raspodjele	Samonumeriranje	Dugi cijeli broj	Da (bez duplikata)
2	Artikl_ID	Jedinstvena brojčana oznaka Artikla	Broj	Dugi cijeli broj	Da (duplikati u redu)
3	Korisnik_ID	Jedinstvena brojčana oznaka Korisnika	Broj	Dugi cijeli broj	Da (duplikati u redu)
4	Da	Dali je Korisnik u Zahtjevu naveo da mu je potreban taj Artikl	Da/Ne		Nema

Entitet: Raspodjela

Redni broj	Naziv atributa	Opis	Vrsta podatka	Veličina polja	Ograničenja
1	ID_RaspodjelaKorisnik	Jedinstvena brojčana oznaka Raspodjele prema Korisniku	Samonumeriranje	Dugi cijeli broj	Da (bez duplikata)
2	Raspodjela_ID	Jedinstvena brojčana oznaka Raspodjele	Broj	Dugi cijeli broj	Da (duplikat u redu)
3	Artikl_ID	Jedinstvena brojčana oznaka Artikla	Broj	Dugi cijeli broj	Da (duplikat u redu)
4	Korisnik_ID	Jedinstvena brojčana oznaka Korisnika	Broj	Dugi cijeli broj	Da (duplikat u redu)
5	Godina	Godina u kojoj je provedena Raspodjela	Broj	Cijeli broj	Nema
6	Da	Oznaka dali je određen Artikl dodijeljen određenom Korisniku	Da/Ne		Nema
7	KolicinaTrazeno	Količina Artikla zahtijevana od Korisnika	Broj	Jednostruko	Nema
8	KolicinaPripadajuce	Količina Artikla pripadajuća Korisniku temeljem izračuna “baze“	Broj	Jednostruko	Nema
9	KolicinaDodijeljeno	Količina Artikla dodijeljena Korisniku	Broj	Jednostruko	Nema
10	Zakljuceno	Dali je okončana raspodjela Artikla za tu Godinu	Da/Ne		Nema

Entitet: Raspodjele

Redni broj	Naziv atributa	Opis	Vrsta podatka	Veličina polja	Ograničenja
1	ID_Raspodjele	Jedinstvena brojčana oznaka Raspodjele	Samonumeriranje	Dugi cijeli broj	Da (bez duplikata)
2	DatumRaspodjele	Datum Raspodjele	Datum/Vrijeme		Nema
3	NalogBroj	Urudžbeni broj Zahtjeva Korisnika	Tekst	20	Nema
4	Godina	Godina u kojoj je provedena Raspodjela	Tekst	4	Nema
5	Artikl_ID	Jedinstvena brojčana oznaka Artikla	Broj	Dugi cijeli broj	Da (duplikati u redu)

Entitet: Zahtjevi

Redni broj	Naziv atributa	Opis	Vrsta podatka	Veličina polja	Ograničenja
1	ID_Zahtjeva	Jedinstvena brojčana oznaka Zahtjeva	Samonumeriranje	Dugi cijeli broj	Da (bez duplikata)
2	DatumZahtjeva	Datum Zahtjeva	Datum/Vrijeme		Nema
3	BrojZahtjeva	Urudžbeni broj Zahtjeva Korisnika	Tekst	50	Da (bez duplikata)
4	Korisnik_ID	Jedinstvena brojčana oznaka Korisnika	Broj	Dugi cijeli broj	Da (duplikati u redu)
5	Realizirano	Dali su u cijelosti ispunjeni zahtjevi Korisnika	Da/Ne		Nema

Entitet: Zahtjevi_Artikli

Redni broj	Naziv atributa	Opis	Vrsta podatka	Veličina polja	Ograničenja
1	ID_ZahtjevArtikl	Jedinstvena brojčana oznaka Zahtjeva po Artiklima	Samonumeriranje	Dugi cijeli broj	Da (bez duplikata)
2	Godina	Godina u kojoj je provedena Raspodjela	Broj	Cijeli broj	Nema
3	Zahtjev_ID	Jedinstvena brojčana oznaka Zahtjeva	Broj	Dugi cijeli broj	Da (duplikat u redu)
4	Korisnik_ID	Jedinstvena brojčana oznaka Korisnika	Broj	Dugi cijeli broj	Da (duplikat u redu)
5	Artikl_ID	Jedinstvena brojčana oznaka Artikla	Broj	Dugi cijeli broj	Da (duplikat u redu)
6	Kolicina	Stvarna Količina dodijeljenog Artikla	Broj	Dugi cijeli broj	Nema

Prilog B

Svojstva

AlternateBackShade:	95	AlternateBackThemeColorInd	1
AlternateBackTint:	100	BackShade:	100
BackTint:	100	DatasheetForeThemeColorInd	0
DatasheetGridlinesThemeColo		3	DateCreated: 28.5.2015. 16:09:11
DefaultView:	2	DisplayViewsOnSharePointSite	1
FilterOnLoad:	False	GUID:	{guid {E4E0C811-67DE-4125-BC76-E846A706A3EB}}
HideNewField:	False	LastUpdated:	29.5.2015. 17:06:02
NameMap:	Dugi binarni podaci	OrderByOn:	False
OrderByOnLoad:	True	Orientation:	Slijeva nadesno
PublishToWeb:	1	ReadOnlyWhenDisconnected:	False
RecordCount:	16	ThemeFontIndex:	1
TotalsRow:	False	Updatable:	True

Stupci

Naziv	Vrsta	Veličina
ID_Artikla	Dugi cijeli broj	4
NazivArtikla	Tekst	255
AS400	Tekst	6
Stanje	Jednostruko	4

Odnosi

ArtikliRaspodjela

Artikli			Raspodjela
ID_Artikla	1	∞	Artikl_ID

Attributes:	Prisilno provedeno
RelationshipType:	Jedan-Na-Više

Indeksi tablice

Naziv	Broj polja
PrimaryKey	1
Polja:	
ID_Artikla	Uzlazno

Svojstva

AlternateBackShade:	95	AlternateBackThemeColorInd	1
AlternateBackTint:	100	BackShade:	100
BackTint:	100	DatasheetForeThemeColorInd	0
DatasheetGridlinesThemeColo		3	DateCreated: 28.5.2015. 16:09:11
DefaultView:	2	DisplayViewsOnSharePointSite	1
Filter:	([ArtiklKorisnik].[Artikl_ID]=8)	FilterOnLoad:	False
GUID:	{guid {04866079-2637-4628-BEB5-E375E2068FD8}}	HideNewField:	False
LastUpdated:	29.5.2015. 15:17:53	NameMap:	Dugi binarni podaci
OrderBy:	[ArtiklKorisnik].[Artikl_ID]	OrderByOn:	True
OrderByOnLoad:	True	Orientation:	Slijeva nadesno
PublishToWeb:	1	ReadOnlyWhenDisconnected:	False
RecordCount:	112	ThemeFontIndex:	1
TotalsRow:	False	Updatable:	True

Stupci

Naziv	Vrsta	Veličina
ID_AK	Dugi cijeli broj	4
Artikl_ID	Dugi cijeli broj	4
Korisnik_ID	Dugi cijeli broj	4
Da	Da/Ne	1

Indeksi tablice

Naziv	Broj polja
Artikl_ID	1
Polja:	
Artikl_ID	Uzlazno
Korisnik_ID	1
Polja:	
Korisnik_ID	Uzlazno
PrimaryKey	1
Polja:	
ID_AK	Uzlazno

Svojstva

AlternateBackShade:	95	AlternateBackThemeColorInd	1
AlternateBackTint:	100	BackShade:	100
BackTint:	100	DatasheetForeThemeColorInd	0
DatasheetGridlinesThemeColo		3	DateCreated: 28.5.2015. 16:09:11
DefaultView:	2	DisplayViewsOnSharePointSite	1
FilterOnLoad:	False	GUID:	{guid {09C1B2DC-64A4-4950-96F5-475CFE4A0351}}
HideNewField:	False	LastUpdated:	29.5.2015. 17:06:11
NameMap:	Dugi binarni podaci	OrderByOn:	False
OrderByOnLoad:	True	Orientation:	Slijeva nadesno
PublishToWeb:	1	ReadOnlyWhenDisconnected:	False
RecordCount:	7	ThemeFontIndex:	1
TotalsRow:	False	Updatable:	True

Stupci

Naziv	Vrsta	Veličina
ID_Korisnika	Dugi cijeli broj	4
NazivKorisnika	Tekst	255
BrojDjelatnika	Dugi cijeli broj	4

Odnosi

KorisniciRaspodjela

Korisnici			**Raspodjela**
ID_Korisnika	1	∞	Korisnik_ID

Attributes:	Prisilno provedeno
RelationshipType:	Jedan-Na-Više

Indeksi tablice

Naziv	Broj polja
PrimaryKey	1
Polja:	
ID_Korisnika	Uzlazno

Svojstva

AlternateBackShade:	95	AlternateBackThemeColorInd	1
AlternateBackTint:	100	BackShade:	100
BackTint:	100	DatasheetForeThemeColorInd	0
DatasheetGridlinesThemeColo	3	DateCreated:	28.5.2015. 16:09:11
DefaultView:	2	DisplayViewsOnSharePointSite	1
FilterOnLoad:	False	GUID:	{guid {A050B40F-A4D5-420C-BA62-888BA133243F}}
HideNewField:	False	LastUpdated:	29.5.2015. 17:06:11
NameMap:	Dugi binarni podaci	OrderByOn:	False
OrderByOnLoad:	True	Orientation:	Slijeva nadesno
PublishToWeb:	1	ReadOnlyWhenDisconnected:	False
RecordCount:	112	ThemeFontIndex:	1
TotalsRow:	False	Updatable:	True

Stupci

Naziv	Vrsta	Veličina
ID_RaspodjelaKorisnik	Dugi cijeli broj	4
Raspodjela_ID	Dugi cijeli broj	4
Artikl_ID	Dugi cijeli broj	4
Korisnik_ID	Dugi cijeli broj	4
Godina	Cijeli broj	2
Da	Da/Ne	1
KolicinaTrazeno	Jednostruko	4
KolicinaPripadajuce	Jednostruko	4
KolicinaDodijeljeno	Jednostruko	4
Zakljuceno	Da/Ne	1

Odnosi

ArtikliRaspodjela

Artikli		**Raspodjela**
ID_Artikla	1 ∞	Artikl_ID

Attributes:	Prisilno provedeno
RelationshipType:	Jedan-Na-Više

KorisniciRaspodjela

Korisnici			Raspodjela
ID_Korisnika	1	∞	Korisnik_ID

Attributes:	Prisilno provedeno
RelationshipType:	Jedan-Na-Više

ZahtjeviRaspodjela

Zahtjevi	Raspodjela
Korisnik_ID	Korisnik_ID

Attributes:	Nije prisilno provedeno
RelationshipType:	Napredniji

Indeksi tablice

Naziv	Broj polja
Artikl_ID	1
Polja:	
Korisnik_ID	Uzlazno
Artikl_ID1	1
Polja:	
Artikl_ID	Uzlazno
ArtikliRaspodjela	1
Polja:	
Artikl_ID	Uzlazno
KorisniciRaspodjela	1
Polja:	
Korisnik_ID	Uzlazno
PrimaryKey	1

Polja:
ID_RaspodjelaKorisnik Uzlazno

Zahtjev_ID 1
Polja:
Raspodjela_ID Uzlazno

Svojstva

AlternateBackShade:	95	AlternateBackThemeColorInd	1
AlternateBackTint:	100	BackShade:	100
BackTint:	100	DatasheetForeThemeColorInd	0
DatasheetGridlinesThemeColo	3	DateCreated:	28.5.2015. 16:09:11
DefaultView:	2	DisplayViewsOnSharePointSite	1
FilterOnLoad:	False	GUID:	{guid {B417CFD7-DF2D-4108-9D51-E3AA24174EDF}}
HideNewField:	False	LastUpdated:	28.5.2015. 16:09:12
NameMap:	Dugi binarni podaci	OrderByOn:	False
OrderByOnLoad:	True	Orientation:	Slijeva nadesno
PublishToWeb:	1	ReadOnlyWhenDisconnected:	False
RecordCount:	7	ThemeFontIndex:	1
TotalsRow:	False	Updatable:	True

Stupci

Naziv	Vrsta	Veličina
ID_Raspodjele	Dugi cijeli broj	4
DatumRaspodjele	Datum/vrijeme	8
NalogBroj	Tekst	20
Godina	Tekst	4
Artikl_ID	Dugi cijeli broj	4

Indeksi tablice

Naziv	Broj polja
Korisnik_ID	1
Polja:	
Artikl_ID	Uzlazno
PrimaryKey	1
Polja:	
ID_Raspodjele	Uzlazno

Svojstva

AlternateBackShade:	95	AlternateBackThemeColorInd	1
AlternateBackTint:	100	BackShade:	100
BackTint:	100	DatasheetForeThemeColorInd	0
DatasheetGridlinesThemeColo		3	DateCreated: 28.5.2015. 16:09:12
DefaultView:	2	DisplayViewsOnSharePointSite	1
FilterOnLoad:	False	GUID:	{guid {0C253FF9-9CE3-4033-9465-2A5353B9BC8C}}
HideNewField:	False	LastUpdated:	29.5.2015. 17:06:19
NameMap:	Dugi binarni podaci	OrderByOn:	False
OrderByOnLoad:	True	Orientation:	Slijeva nadesno
PublishToWeb:	1	ReadOnlyWhenDisconnected:	False
RecordCount:	18	ThemeFontIndex:	1
TotalsRow:	False	Updatable:	True

Stupci

Naziv	Vrsta	Veličina
ID_Zahtjeva	Dugi cijeli broj	4
DatumZahtjeva	Datum/vrijeme	8
BrojZahtjeva	Tekst	50
Korisnik_ID	Dugi cijeli broj	4
Realizirano	Da/Ne	1

Odnosi

ZahtjeviRaspodjela

Zahtjevi		Raspodjela
Korisnik_ID		Korisnik_ID

Attributes: Nije prisilno provedeno
RelationshipType: Napredniji

ZahtjeviZahtjevi_Artikli

Zahtjevi		Zahtjevi_Artikli
ID_Zahtjeva	1 ∞	Zahtjev_ID

Attributes: Prisilno provedeno
RelationshipType: Jedan-Na-Više

Indeksi tablice

Naziv	Broj polja
Broj zahtjeva	1
Polja:	
BrojZahtjeva	Uzlazno
Korisnik ID	1
Polja:	
Korisnik_ID	Uzlazno
PrimaryKey	1
Polja:	
ID_Zahtjeva	Uzlazno

Svojstva

AlternateBackShade:	95	AlternateBackThemeColorInd	1
AlternateBackTint:	100	BackShade:	100
BackTint:	100	DatasheetForeThemeColorInd	0
DatasheetGridlinesThemeColo		3	DateCreated: 28.5.2015. 16:09:12
DefaultView:	2	DisplayViewsOnSharePointSite	1
FilterOnLoad:	False	GUID:	{guid {9CAB9E51-A076-4AD1-951B-0ED57762045C}}
HideNewField:	False	LastUpdated:	29.5.2015. 17:06:19
NameMap:	Dugi binarni podaci	OrderByOn:	False
OrderByOnLoad:	True	Orientation:	Slijeva nadesno
PublishToWeb:	1	ReadOnlyWhenDisconnected:	False
RecordCount:	41	ThemeFontIndex:	1
TotalsRow:	False	Updatable:	True

Stupci

Naziv	Vrsta	Veličina
ID_ZahtjevArtikl	Dugi cijeli broj	4
Godina	Cijeli broj	2
Zahtjev_ID	Dugi cijeli broj	4
Korisnik_ID	Dugi cijeli broj	4
Artikl_ID	Dugi cijeli broj	4
Kolicina	Dugi cijeli broj	4

Odnosi

ZahtjeviZahtjevi_Artikli

Zahtjevi		**Zahtjevi_Artikli**
ID_Zahtjeva	1 ∞	Zahtjev_ID

Attributes:	Prisilno provedeno
RelationshipType:	Jedan-Na-Više

Indeksi tablice

Naziv	Broj polja
Artikl_ID	1

Polja:

Artikl_ID	Uzlazno

PrimaryKey	1	
Polja:		
ID_ZahtjevArtikl	Uzlazno	
Zahtjev_ID	1	
Polja:		
Korisnik_ID	Uzlazno	
Zahtjev_ID1	1	
Polja:		
Zahtjev_ID	Uzlazno	
ZahtjeviZahtjevi_Artikli	1	
Polja:		
Zahtjev_ID	Uzlazno	

11. Popis kratica

ACCDB	format baze podataka
ACID	pravila (*engl. Atomicity Consistency Isolation Durability*)
API	programsko sučelje (*engl. Application Programming Interfaces*)
BCNF	Boyce-Coddovoj normalna forma
CASE	metodologija (*engl. Computer-Aided Systems Engineering*)
DCL	(*engl. Data Control Language*) – jezik koji se koristi od strane administratora baze podataka pri određivanju ovlaštenih korisnika baze podataka i određivanju njihovih mogućnosti rada u samoj bazi podataka
DDL	(*engl. Data Definition Language*) – jezik za stvaranje strukture podataka (opis), brisanje strukture podatka, dodavanje postojeće strukture podatka
DIL	(*engl. Data Integrity Language*) – jezik koji se koristi pri određivanju ograničenja koja se odnose na integritet podataka
DML	(*engl. Data Manipulation Language*) – jezik koji koristimo kao primarni mehanizam za određivanje naredbi za upravljanje podacima koji se odnose na bazu podataka
DOS	uskraćivanje resursa (*engl. Denial of Service*)
DTP	dijagram toka podataka (*engl. Data Flow Diagram, DFD*)
EV	entiteti veza (*engl. Entity Relationship, ER*)
GUI	grafičko korisničko sučelje (*engl. Graphical User Interface*)
HTML	oblik koji integrira umetnutu grafiku i povezuje dokumente (*engl. HyperText Markup Language*)
ID	jedinstveni indetifikator (*engl. Identification*)
IP	(*engl. Internet Protocol*)
Jet/JetBase	združena računalna tehnologija (radne stanice) (*engl. Joint Engine Technology*)
MDB	format baze podataka
MS	skraćenica za Microsoft
NF	normalna forma
ODBC	otvorene spojivosti podatkovnih baza (*engl. Open DataBase Connectivity)*

PDF	oblik dokumenta (*engl. Portable Document Format*)
PK	primarni ključ – jedinstveni indetifikator nekog sloga iz tabela
QL/SQL	*engl. Query Language*) – jezik koji je namijenjen za rad baze podataka, postao standardni prikazani jezik za rad relacijskih i ne-relacijskih baza podataka
SQL	(*engl. Structured Query Language*)
SSL	zaštitni protokol za komunikaciju (*engl. Secure Socets Layer*)
SUBP	sustavi upravljanja bazom podataka (*engl. DataBase Management System, DBMS*)
SURBP	sustav namijenjen upravljanju relacijskim bazama podataka (*engl. Relation DataBase Menagement System, RDBMS*)
T- SQL	(*engl. Transact SQL*) što znači da osim klasičnih SQL upita dozvoljava i složenije stvari poput mijenjanja programskog toka i sl
TDS	protokol (*engl. Tabular Data Stream*) koji omogućava da što više aplikacija može komunicirati s bazama podataka
Txt	oblik koji sprema radnu knjigu kao tekstualnu datoteku razdvojenu tabulatorom za uporabu u drugom operacijskom sustavu
UML	(*engl. Unified Modeling Language*) je grafički jezik za vizualizaciju
VBA	jezik koji pomaže da jednim klikom ili istovremenim pritiskom kombinacije nekoliko gumba sustav izvede neku naredbu (*engl. Visual Basic for Applications*)
VK	vanjski ključ (*engl. foreign key*, *FK*) označava nam primarni ključ neke druge tablice koju međusobno povezujem s našom trenutnom tablicom
XML	jezik za označavanje podataka koji nudi pohranu podataka, koristi se određenim skupom zadanih oznaka (*engl. EXtensible Markup Language*)
XPS	oblik gotovo jednakih karakteristika kao PDF (*engl. XML Paper Specification*)

12. Popis slika

Slika 1. Životni ciklus informacijskog sustava, str. 12

Slika 2. Dijagram toka podataka, str. 16

Slika 3. Neki od simbola dijagrama toka podataka prema ANSI standardu, str. 17

Slika 4. Dijagram korištenja, str. 19

Slika 5. Relacijski model baze podataka izrađen u MS Access-u 2010, str. 23

Slika 6. relacijski model baze podataka izrađen u MS Visio 2007, str. 23

Slika 7. Mrežni model, str. 24

Slika 8. Hijerarhijski model, str. 24

Slika 9. Prikaz podataka u tablici, str. 26

Slika 10. Prikaz podatkovnog lista, str. 27

Slika 11. Podaci u obliku prikaza dizajna, str. 28

Slika 12. Podaci prikazani u SQL obliku, str. 28

Slika 13. Prikaz kartica MS Access-a 2010, str. 33

Slika 14. Prikaz tablice, str. 34

Slika 15. Dijagram protoka poslova, str. 41

Slika 16. Prikazi sadržajnog i razvojnog oblika tablice s podacima, str. 43

Slika 17. Prikaz veze 1:1, str. 44

Slika 18. Prikaz veze 1:n (n:1), str. 45

Slika 19. Prikaz veze n:n, str. 45

Slika 20. Stvaranje veze između tablica, str. 46

Slika 21. Prikaz veza između tablica “naše“ baze, str. 46

Slika 22. Prikaz veze s nametnutim referencijalnim integritetom, str. 47

Slika 23. Izgled prazne tablice u obliku podatkovnog lista, str. 48

Slika 24. Prikaz razvojnog oblika tablice kroz prikaz dizajna, str. 51

Slika 25. Prikaz mogućnosti za oblikovanje podataka, str. 51

Slika 26. Izgled Čarobnjaka za uvoz podataka u jednom segmentu, str. 53

Slika 27. Izgled „“čarobnjaka“ Dokumentatora, str, 54

Slika 28. Izgled Čarobnjaka za obrasce podataka u jednom segmentu, str. 56

Slika 29. Izgled obrazaca za unos Zahtjeva korisnika, str. 57

Slika 30. Izgled obrasca za unos Artikla i njihove količine za raspodjelu, str. 57

Slika 31. Izgled obrasca za unos podataka iz Zahtjeva korisnika, str. 58

Slika 32. Prikaz dijaloškog okvira pri izradi upita, str. 60

Slika 33. Odabir polja i funkcija kod izrade upita, str. 61

Slika 34. Izgled izvještaja u “tabličnom obliku“, str. 63

Slika 35. Izgled izvještaja Slike 33. u “Print preview“ obliku Pretpregleda, str. 64

Slika 36. Mogućnosti za izvoz podataka, str. 66

13. Popis tablica

Tablica 1. Funkcije radnika u timu, str. 8

Tablica 2. Oblici modeliranja i pripadajuće metode, str. 10

Tablica 3. Neki od poznatih programskih paketa za rad s bazama podataka, str. 28

Tablica 4. Opis entiteta “naše“ baze podataka, str. 42