Primjena baze podataka u zaštiti na radu

Perković, Boris

Master's thesis / Diplomski rad

2015

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:697590

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-16**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
ELEKTROTEHNIČKI FAKULTET

Sveučilišni studij

PRIMJENA BAZE PODATAKA U ZAŠTITI NA RADU

Diplomski rad

Boris Perković

Osijek, 2015. godina.

SADRŽAJ

1. UVOD 1
   1.1 Zadatak 1
2. TEORIJSKI DIO 2
   2.1 Baza podataka 2
3. Arhitektura baze podataka 5
   3.1 Logička razina 5
   3.2 Fizikalna razina 5
   3.3 Jezik SQL 6
4. IZRADA BAZE PODATAKA ZA ZAŠTITU NA RADU 9
   4.1 Konceptualni model beze podataka 9
   4.2 Logički model baze podataka 10
   4.3 ER - dijagram 11
   4.4 Logički model 12
   4.5 Fizički model 15
      4.5.1 Unos podataka u bazu podataka 15
      4.5.2 Upiti koji se najčešće koriste 15
   4.6 Izrada baze podataka 17
   4.7 Popunjavanje baze podataka 21
5. ZAKLJUČAK 22

LITERATURA 23

SAŽETAK 24

USAGE OF DATABASE IN SAFETY AT WORK 24

ŽIVOTOPIS 25

PRILOG 27

1. UVOD

Kada se govori o bazi podataka, govori se o skupu međusobno povezanih podataka, pri čemu se podaci dijele i koriste unutar ili između pojedinih programa odnosno aplikacija različitih namjena pod različitim uvjetima. Pojedini uvjeti kod baza podataka su da se kontrolira pojedini pristup te unos, brisanje, dohvaćanje i izmjenjivanje podataka. [Manger, Baze podataka, 2013] Sama izrada baze podataka je pojednostavljena te će ova dokumentacija predstaviti samu izradu baza podataka kroz logički i fizički model te će prikazati predstavljanje sučelja korisnicima uz sam slikovit prikaz. Korisnicima podataka nije postavljen uvjet poznavanja detalja same strukture podataka što znači da je olakšano samo korištenje bez nužnog poznavanja cijelokupne strukture. Baza podataka čini skup podataka koji zadovoljavaju potrebe korisnika odnosno potrebe o pojedinim informacijama. No, korisnici baza podataka su u današnje vrijeme apsolutno svi, jer je trenutna struktura i korištenje računala nezamislivo bez baza podataka, od samih evidencija i skupova do posebno čuvanih baza podataka s velikim količinama informacija.

1.1 Zadatak

Projekt baze podataka u svrhu zaštite na radu, primjena baze podataka kao evidencija djelatnika u određenim poslovima, evidencija opreme pojedinih djelatnika u svrhu nadgledanja trajnosti opreme odnosno evidencija promjene opreme za pojedine djelatnike, te evidentiranje ozljeda ako ih je bilo. Projekt će biti izrađen uz pomoć jezika SQL.

2. TEORIJSKI DIO

2.1 Baza podataka

Baza podataka je skup međusobno povezanih podataka koji su spremljeni bez štetne redundancije na jedinice vanjskih memorija, gdje postoji mogućnost istovremenog korištenja podataka od strane više korisnika (aplikacija), gdje su podaci neovisni o aplikacijama kojima se obrađuju i gdje postoji mehanizam zaštite podataka i kontrola pristupa . Podaci u bazi su logički organizirani u skladu s nekim modelom podataka. Model podataka je skup podataka koja određuju kako može izgledati logička struktura baze i čini osnovu za koncipiranje, projektiranje i implementiranje baze podataka. Model podataka je formirani sustav koji mora imati barem sljedeće tri komponente:

A) Skup objekata koji su osnovni elementi baze podataka,
B) Skup operacija koje možemo izvoditi nad objektima definiranim pod A i kojima se mogu pretraživati, dobivati i modificirati podaci o tim objektima,
C) Skup općih pravila integriteta podataka koja implicitno ili eksplicitno definiraju skup konzistentnih stanja podataka ili promjena stanja, ili oboje i koja su općenita u smislu da su primjenjiva na bilo koju bazu podataka koja koriste taj model. [Manager, Baze podataka, 2013]

Model podataka se koristi za:

- razvijanje sustava za upravljanje bazom podataka,
- razvijanje programskih jezika za rad s podacima u bazi podataka,
- razvijanje općih teorija oblikovanja baza podataka,
- istraživanje svojstava podataka, odnosno svojstava baza podataka. [Manger, Baze podataka, 2013]

Sustav za upravljanje bazom podataka (*eng. Data Base Management System- DBMS*) je programski sustav koji osigurava osnovne funkcije odabranog modela podataka u postupku kreiranja i korištenja baze podataka. DBMS osigurava softverske alate potrebne za kreiranje, primjenu, pristupanje i ažuriranje sadržaja baze podataka. On, također, upravlja svim ulazno-izlaznim operacijama, a na većim sustavima vodi računa o kontroli pristupa, odnosno zaštiti i

integritetu podataka te rješava probleme nastale usljed istovremenog pristupanja istim podacima. [Manger, Baze podataka, 2013]

Komunikacija korisnika odnosno aplikacijskog programa i DBMS-a odvija se pomoću specijaliziranih jezika. Ti se jezici tradicionalno dijele na sljedeće kategorije:

- jezik za opis podataka (*eng. Data Description Language- DLL*)
- jezik za manipuliranje podacima (*eng. Data Manipulation Language- DML*)
- jezik za postavljanje upita (*eng. Query Language- QL)* [Manger, Baze podataka, 2013]

Jezik za opis podataka:

- služi projektantu baze podataka ili administrantu radi zapisivanja sheme (ili pod sheme),
- tim se jezikom definiraju podaci i veze među njima,
- naredbe DDL podsjećaju na naredbe za definiranje složenih tipova podataka u jezicima kao što su Pascal, PL/1. [Manger, Baze podataka, 2013]

Jezik za manipuliranje podacima:

- služi za uspostavljanje veza između aplikacijskih programa i baze,
- naredbe DML omogućuju upis, promjene, brisanje, čitanje i općenito manipuliranje podacima,
- u nekim DBMS, MDL je biblioteka potprograma tj. naredbe u jeziku se pozivni na potprograme, dok je drugdje riječ o posebnom jeziku. [Manger, Baze podataka, 2013]

Jezik za postavljanje upita:

- služi korisniku za direktno pretraživanje baze podataka (tzv. interaktivni rad),
- naredbe su takve da definiraju što se želi dobiti, ali ne i kako do toga doći (algoritmi). Ovo svojstvo se naziva neproceduralnost. [Manger, Baze podataka, 2013]

Sustav za upravljanje se sastoji od integrirane kolekcije programske podrške koja omogućava:

- opis i manipulaciju podacima pomoću posebnog jezika ili posebnih jezika
- visoki nivo sučelja prema podacima neovisno o strukturi podataka u računalu
- efikasno korištenje i razumijevanje informacija pohranjenih u bazi podataka, zahvaljujući skupu programskih alata (pomagala). [Manger, Baze podataka, 2013]

Svaki sustav za upravljanje bazom podatak je zasnovan na nekom od sljedećih modela:

- Hijerarhijski model - zasniva se na ideji povezivanja podataka u hijerarhijske strukture pri čemu starijim zapisu odgovara više zapisa nižeg nivoa, ali zapisu nižeg nivoa odgovara samo jedan zapis višeg nivoa. Na vrhu hijerarhije nalazi se korijeni zapis iz kojega se kasnije kroz hijerarhijske čvorove razvijaju ostali zapisi i na taj način pokušava obuhvatiti stanje sustava. Baza je predočena stablom ili skupom stabala. Veze između tipova zapisa su izražene hijerarhijskim odnosom nadređeni-podređeni. [Varga, Baze podataka, 1994.]
- Mrežni model – zasniva se na ideji mrežne strukture. Osnovni koncepti ovog modela su polja, slogovi i setovi. Za predstavljanje veza između objekata realnog sustava koristi se set. Set predstavlja vezu između vlasnika seta (jedan zapis) i jednog ili više setova. Ovaj model prati proceduralni jezici pri čemu nisu odvojeni logički i fizički koncept baze podataka. Baza je predočena usmjerenim grafom. Čvorovi su tipovi zapisa a lukovi veze među tipovima. [Varga, Baze podataka, 1994.]
- Relacijski model - zasnovan je na matematičkom pojmu relacije. Baza podataka se sastoji od skupa pravokutnih tablica ili relacija. Relacijski model je danas najviše primjenjivan model podataka. Većina poznatih sustava za upravljanje bazom podataka zasnovani su na relacijskom modelu (Oracle, Informix, MS Access). Kao glavne prednosti ovoga modela navode se jednostavnost strukture, što omogućava primjenu matematičkih formalizama u postupku projektiranja na konceptualnom nivou kao i lak interaktivni pristup uz pomoć nekih od poznatijih upitnih jezika (SQL). [Varga, Baze podataka, 1994.]
- Objektni model - je inspiriran objektno-orijentiranim progrmskim jezikom. Baza je skup trajno pohranjenih objekata koji se sastoje od svojih internih podataka i metoda koje vrše operacija nad njima. Svaki objekt pripada nekoj klasi. Između klase se uspostavljaju veze nasljeđivanja, agregacije, odnosno međusobnog korištenja operacija. [Varga, Baze podataka, 1994.]

3. Arhitektura baze podataka

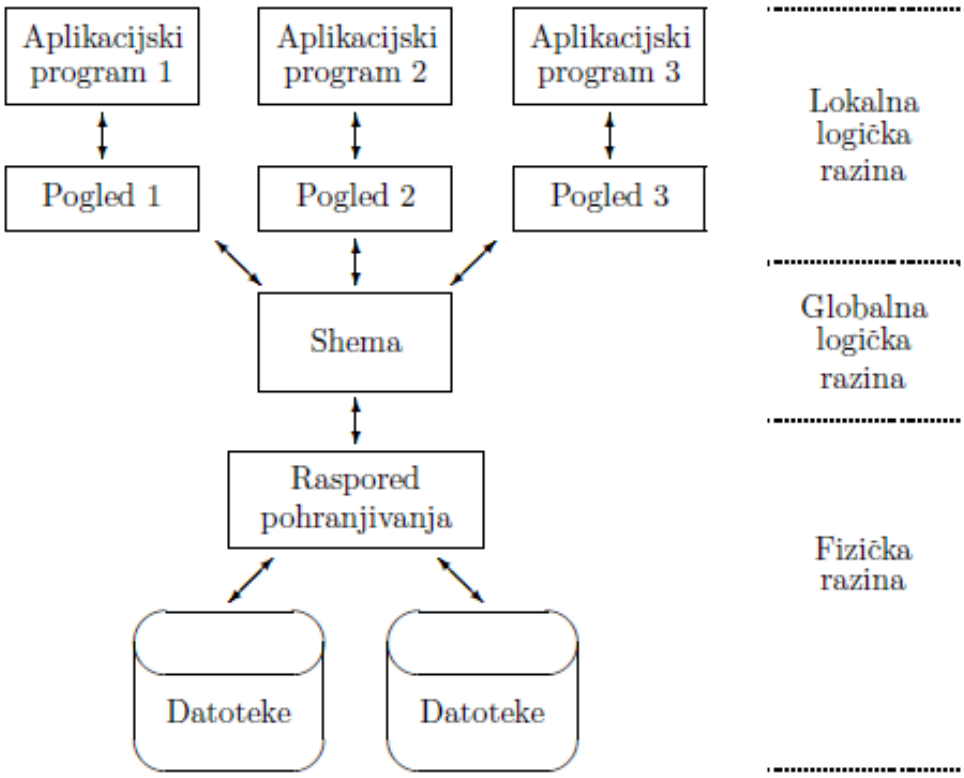
3.1 Logička razina

Lokalna logička razina - odnosi se na logičku predodžbu o dijelu baze kojeg koristi pojedina aplikacija. To je aspekt kojeg vidi korisnik ili aplikacijski programer. Zapis jedne lokalne logičke definicije zove se pogled ili pod-shema. Pogledom se zadaje preslikavanje kojim se iz globalnih podataka i veza izvode lokalni.

Globalna logička razina - odnosi se na logičku predođbu o cijeloj bazi. To je aspekt kojeg vidi projektant baze odnosno njezin administrator. Zapis logičke definicije cjelokupne baze podataka zove se shema. To je tekst ili dijagram kojim se imenuju i definiraju svi podaci, veze među podacima, te pravila (ograničenja) kojima se čuva integritet baze podataka. [Manger, Baze podataka, 2013.]

3.2 Fizikalna razina

Fizička razina - odnosi se na fizički prikaz i raspored podataka na jedinicama vanjske memorije. To je aspekt kojeg vide samo sistemski programeri (oni koji su razvili DBMS). Raspored pohranjivanja opisuje kako se se elementi logičke definicije baze preslikavaju na fizičke uređaje. Rad je napravljen na temelju troslojne arhitekture baze podataka. [Manger, Baze podataka, 2013.]



Sl.2.1. Troslojna (trorazinska) arhitektura baze podataka

3.3 Jezik SQL

SQL (*eng. Structured Query Language)* je jezik za rad s relacijskom bazom podataka. SQL se koristi za obavljanje zadataka kao što su ažuriranje podataka o bazi podataka, dohvatiti podatke iz baze podataka, dodavanje novih podataka u bazu, brisanje postojećih podataka iz baze, izmjenu postojećih podataka u bazi, administraciju korisnika i prava pristupa podacima, kreirati nove baze podataka i stvoriti nove tablice u bazi podataka. SQL objedinjuje funkcije jezika za definiciju podataka (*eng. Data Definition Language-DLL*) i jezika za upravljanje podacima (*eng. Data Manipulation Language-DML*). [Manger, Baze podataka, 2013.]

Osnovni objekti u SQL jeziku:

- baza podataka
- tablica
- stupac
- pogled ili virtualna tablica
- sinonim
- indeks
- pohranjena procedura
- okidač

Postoje 3 grupe naredbi:

1. naredbe za 4 osnovne operacije nad podacima u bazi:

- pronalaženje
- upis
- brisanje
- promjena

2. naredbe za definiranje relacijske sheme baze podataka, opis ograničenja i definiranje pogleda na bazu podataka

3. naredbe za upravljanje bazom podataka

SQL DML (*eng. Data Manipulation Language*)

- SELECT - naredba služi za selekciju i ispis podataka jednog upita
- UPDATE - služi za izmjenu postojećih podataka
- DELETET - služi za brisanje postojećih podataka
- INSERTE - za unos novih n-torki: upisom vrijednosti atributa i unosom jedne n-torke, selektiranjem n-torki i njihovih atributa iz postojeće relacije i njihova prijenosa u relaciju

SQL DDL (*eng. Data Definition Language*)

Definicija objekata u bazi:

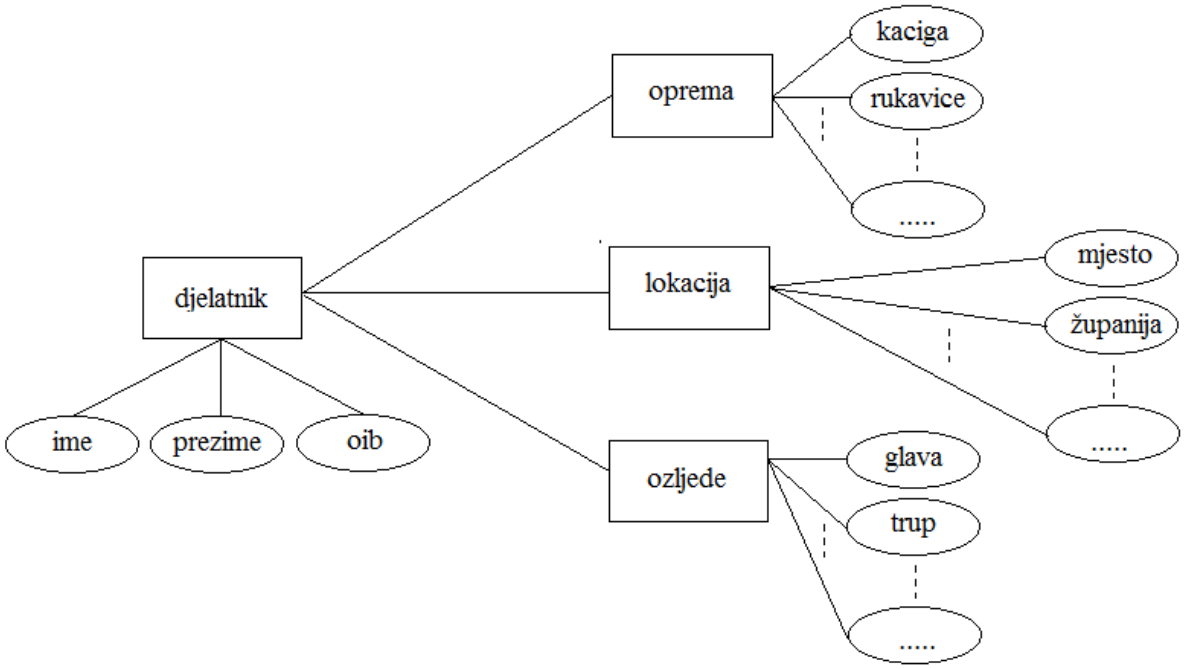
- CREATE - naredbom se kreira nova relacija, odnosno opisuje njena relacijska shema

- DROP - naredbom se relacijska shema izbacuje iz kataloga, a pripadna relacija iz baze podataka
- ALTER - naredbom se mijenja definicija relacijske sheme. Moguće je dodati novi atribut ili redefinirati postojeći
- GRANT - vlasnik relacije je korisnik koji ju je definirao s ovom naredbom. Samo je on ima pravo koristiti dok to pravo ne prenese na drugog. Pravo korištenja se prenosi naredbom GRANT
- REVORKE - pravo korištenja relacije se ukida ovom naredbom. [Manger, Baze podataka, 2013.]

4. IZRADA BAZE PODATAKA ZA ZAŠTITU NA RADU

4.1 Konceptualni model beze podataka

U stvarnom svijetu mnogi poslodavci se, a i ja sam, susreću u praksi s ogromnim listama i evidencijama koje su vezane za određene djelatnike. Kako bih pomogao i olakšao sebi, a i drugima, došao sam na ideju ovog projekta. Kako svaki radnik ima svoje podatke, popis opreme koju je zadužio za određeno radno mjesto konceptualno sam zamislio bazu podataka koja se sastoji od 3 tablice vezane za njegovo radno mjesto, zaduženu opremu i osobne podatke. Ali kako je moguće da se radnik ozljedi na radnom mjestu, vrlo nam je bitno voditi i evidenciju o tome tako da na kraju imamo bazu podataka od 4 tablice. Glavna tablica bi se zvala „djelatnik“ koja sadržava osnovne podatke o radniku(npr. ime, prezime, oib, kontakt). Na glavnu tablicu vezale bi se ostale, tablica „oprema“ sa popisom opreme koju je radnik zadužio, tablica lokacija vezana za lokaciju gdje je radnik obavljao svoj posao te tablica „ozljede“ u kojoj bi evidentirali ozljede radnika. Konceptualno bi to izgledalo kao na slici 3.1.



Slika 3.1. Konceptualni prikaz baze podataka

4.2 Logički model baze podataka

Baza podataka napravljena uz pomoć 4 tablice. Baza kreirana je uz pomoć programa PHPMyAdmin-a uz pomoć programa XXAMP za simuliranje poslužitelja na osobnom računalu. Baza sadrži 4 tablice pod nazivima: - djelatnik, lokacija, oprema te ozljede. Nastavak opisuje svrhu pojedinih tablicu u ovom projektu.

Tablica `djelatnik`

Tablica „djelatnik“ mora sadržavati najosnovnije podatke djelatnika kako bi korisnik baze podataka na koncu mogao vidjeti podatke djelatnika. Ove informacije su jako bitne i korisne u industriji i u tvrtkama koje imaju veliku broj djelatnika. Naime, podaci u ovoj tablici povezani su s ostalim tablicama odnosno ova tablica čini osnovu u cijeloj bazi podataka. Tablica sadrži najosnovniji podatak a to je „id“ djelatnika što će biti osnova za pronalazak svih informacija djelatnika.

Tablica `lokacija`

Tablica „lokacija“ služi u svrhu spremanja podataka djelatnika o mjestu djelovanja. Mjesto djelovanja je također bitno jer se mora voditi evidencija o tome na kojim je mjestima djelatnik sve bio, a osnovu u ovoj tablici činit će „koordinate“ po kojima će korisnik tablice imati konkretan podatak direktno s uređaja s lica mjesta.

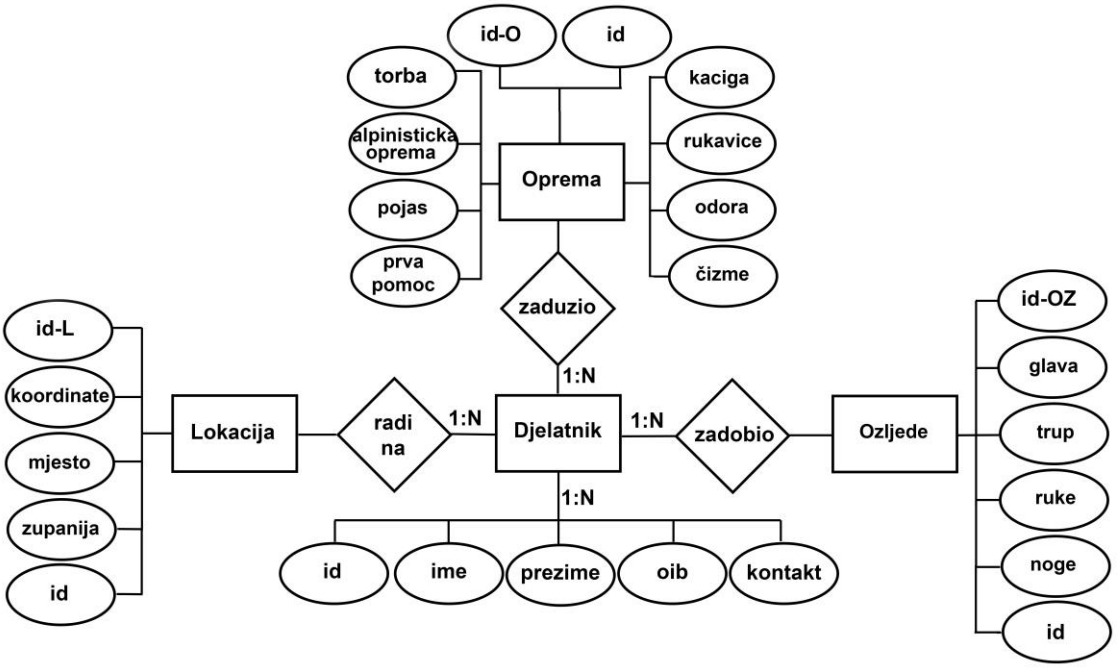
Tablica `oprema`

Tablica „oprema“ čini najveću tablicu u ovoj bazi podataka, kao što samo ime kaže radi se o opremi koju djelatnik pod određenim „id“-om zadužuje, odnosno radi se o broju preuzete opreme različitih namjena.

Tablica `ozljede`

Tablica „ozljede“ imat će informacije svih djelatnika o određenim ozljedama koje su zadobili na radnom mjestu. Ozljede su kategorizirane u 4 grupe kako bi se vođenje evidencije svih djelatnika pod određenim „id“-om tijekom određene akcije ili rada olakšalo korisniku te se imalo saznanja o kojoj se ozljedi tijela radi.

4.3 ER - dijagram



Sl. 3.2. ER dijagram baze podataka „zastita“

Slika 3.2. prikazuje ER – dijagram baze podataka „zastita“. Veliku ulogu pri izradi ER – dijagrama imaju „entiteti“. Entiteti kao prepoznatljive tablice u ovoj bazi podataka čine: djelatnik, oprema, lokacija i ozljede. Svaki entitet sadrži pojedine atribute koje formiraju bazu podataka, te kako bi naknadno bilo prostora za popunjavanje tablica. Atributi entiteta djelatnik su: id, ime, prezime, oib, kontakt. Atributi entiteta oprema čine: id-O, kaciga, odora, cizme, rukavice, torba, brdska oprema, pojas, prva pomoc. Atributi u entitetu lokacija su: id-L koordinate, mjesto, zupanija. Zadnja tablica odnosno zadnji entitet čini ozljede s atributima: id-OZ, glava, trup, noge te ruke. Ovaj ER dijagram ujedno predstavlja i shematski prikaz baze podataka. Ključne riječi korištene u ER dijagramu su ID kod pozivanja podataka o korisniku, ID-L za informacije o lokaciji gdje je korisnik pod određenim ID-em radio, ID-O za informacije o opremi koju je određeni korisnik zadužio, te ID-OZ gdje možemo pozvati tablicu sa podacima o ozljedi djelatnika.

4.4 Logički model

Kao što je predstavljeno na ER – dijagramu atributi pojedinih tablica međusobno su povezani. Logički model predstavlja detaljniji opis entiteta odnosno svake tablice posebno, prema prikazanim entitetima posebno je zapisan svaki atribut te detalji o svakom. Svaka tablica opisana je sa 5 elemenata – „Polje“, „Vrsta“, „Null“, „Zadano“ i „Poveznica“. Najvažniji element u logičkom oblikovanju je „Poveznica“ koja označava povezanost jedne tablice sa drugom. Povezivanje se izvršava pomoću primarnog i stranog ključa, koji čine poveznicu na pojedine atribute iz jedne u drugu tablicu.

djelatnik				
Polje	Vrsta	Null	Zadano	Poveznica
id	int(20)	No		djelatnik => oprema
ime	varchar (30)	No		
prezime	varchar (30)	No		
oib	varchar (11)	No		
kontakt	varchar (50)	No		

Tab. 3.1. Tablična struktura za tablicu *djelatnik*

Struktura tablice „djelatnik“ sadrži sve bitne informacije o djelatniku koji će u konačnici imati određen broj zadužene opreme i biti na određenim lokacijama te ako je imao ozljedu biti će navedena o kojem dijelu tijela je riječ. Atribut „id“ je primarni ključ unutar ove tablice, dok atributi ime, prezime, oib, kontakt su ostali atributi odnosno faktori za popunjavanje informacijama.

lokacija				
Polje	Vrsta	Null	Zadano	Poveznica
id-l	int(20)	No		lokacija => djelatnik
koordinate	varchar(100)	No		
mjesto	varchar(40)	No		
zupanija	varchar(40)	No		

Tab.3.2. Tablična struktura za tablicu *lokacija*

Tablica odnosno tablična struktura „lokacija“ čini parametar kojim se evidentiraju mjesta na kojima je djelatnik bio uz pomoć zadanih atributa: koordinate, mjesto te županija.

ozljede				
Polje	Vrsta	Null	Zadano	Poveznica
Id-oz	int(20)	No		ozljede => djelatnik
glava	varchar(100)	No		
trup	varchar(100)	No		
noge	varchar(100)	No		
ruke	varchar(100)	No		

Tab.3.3. Tablična struktura za *ozljede*

Tablična struktura „ozljede“ sadrži atribute „id“ kao glavni čimbenik prema kojemu korisnik baze podataka vidi o kojem je djelatniku riječ te ostale atribute koji upućuju na to koji dio tijela je ozlijeđen: glava, trup, noge ili ruke ako je došlo do ozljede.

oprema				
Polje	Vrsta	Null	Zadano	Poveznica
Id-o	int(20)	No		oprema => djelatnik
kaciga	smalint(10)	No		
odora	smalint(10)	No		
cizme	smalint(10)	No		
rukavice	smalint(10)	No		
torba	smalint(10)	No		
brdska_oprema	smalint(10)	No		
pojas	smalint(10)	No		
prva_pomoc	smalint(10)	No		

Tab. 3.4. Tablična struktura za *opremu*

Tablica 3.4. predstavlja tabličnu strukturu „oprema“ te se također kao glavni atribut uzima „id“ kako bi se moglo znati o kojem se djelatniku radi odnosno koji je djelatnik preuzeo određenu opremu i u kojoj količini do sada. Ostatak atributa uz „id“ čine sljedeći: kaciga, odora, čizme, rukavice, torba, brdska oprema, pojas te prva pomoć.

4.5 Fizički model

Naredbe „SQL“ čine fizički model baze podataka. Fizički model se sastoji od „SQL“ naredbi za kreiranje baze podataka te za kreiranje tablica tj. atributa unutar istih. Sam početak čini izradu baze podataka naredbom „CREATE DATABASE“ gdje se određuje ime baze podataka, te sustav znakova koji se koristi za izradu. Nakon toga se kreiraju tablice sa svojim atributima i entitetima. Baza je kreirana u PHPMyAdmin-u tako da se kod nije ručno pisao već pri izradi tablice dobiven u textfield-u. Kodovi gdje se kreira baza podataka i definiraju naredbe za definiranje tablica ispisani su u dodatku na kraju rada.

4.5.1 Unos podataka u bazu podataka

Zadane naredbe dobivene su nakon unošenja pojedinih podataka putem grafičkog sučelja te su na isti način dobivene naredbe za unošenje podataka u pojedine atribute pojedinih tablica. U sljedećem poglavlju biti će slikovito prikazan unos podataka. Treba naglasiti da prilikom popunjavanja tablica treba pripaziti na razmake koji se ne smiju koristiti ali se kod više riječi postavlja znak “_“ kao razmak, ovo pravilo vrijedi u svim segmentima izrade od same izrade baze i tablica do popunjavanja informacijama. Nastavak prikazuje naredbe za unos podataka u pojedine tablice.

4.5.2 Upiti koji se najčešće koriste

Ispis svih podataka iz tablice oprema:

*SELECT * FROM oprema;*

Upit služi za prikaz svih podataka u tablici oprema.

Ispis svih podataka iz tablice djelatnik:

*SELECT * FROM djelatnik;*

Ovaj upit služi za dobivanje svih podataka iz tablice djelatnik.

Ispis svih podataka iz tablice ozljede:

*SELECT * FROM ozljede;*

Ovaj upit ispisuje podatke „Id“-a djelatnika te ako je bilo ozljeda određenih ekstremiteta označeno je polje s „da“ a time će biti prikazana ta informacija.

Izmjena podatka u tablici djelatnik:

UPDATE `djelatnik`` SET `id` = 12341244 *WHERE `ime` = boris ;*

Upit služi da za izmjenu podatka kojemu je ime „boris“

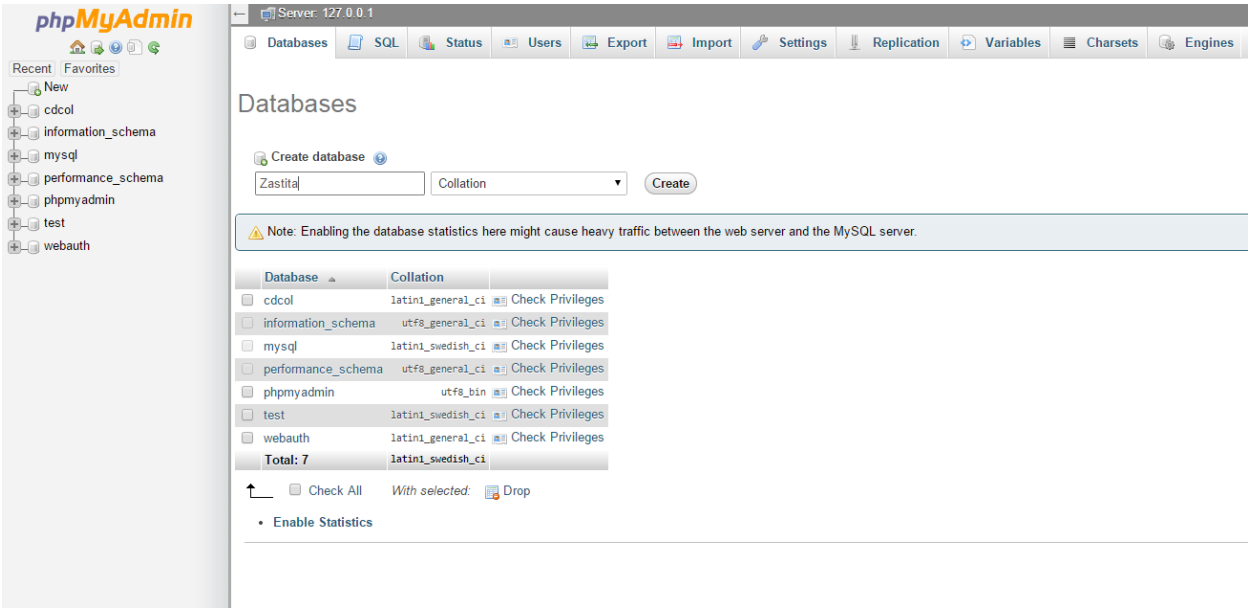
Brisanje pretplatnika:

DELETE FROM `djelatnik` WHERE id='2';

Upit briše red podataka iz tablice djelatnik pod id = 2.

4.6 Izrada baze podataka

Nastavak rada predstavlja slikovit prikaz izrade baze podataka uz pomoć zadanog programa „PHPMyAdmin“ koji uz pomoć programa „XXAMP“ simulira poslužitelj uz pomoć servisa „Apache“ te „MySQL“, na osobnom računalu.

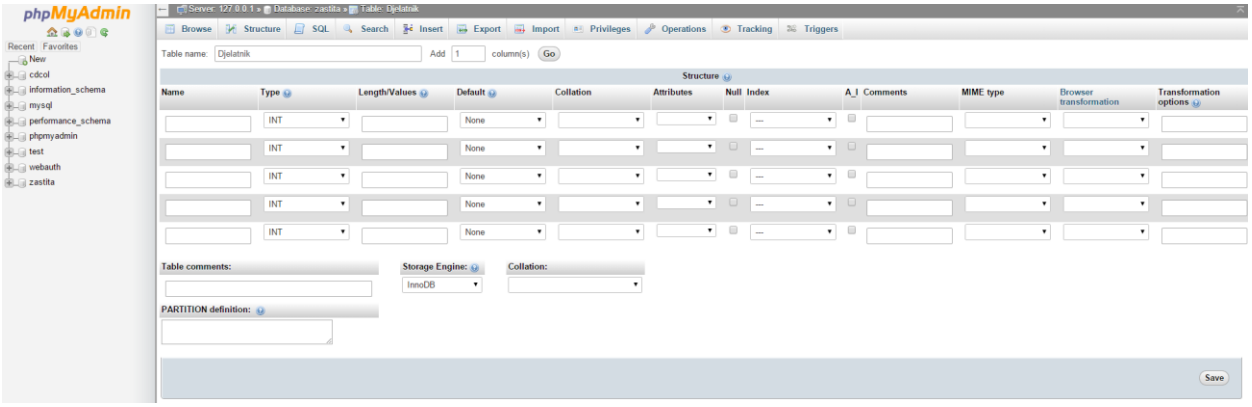


Sl. 3.3. prikaz izrade baze podataka

Sl. 3.4. prikaz izrade tablice „djelatnik“

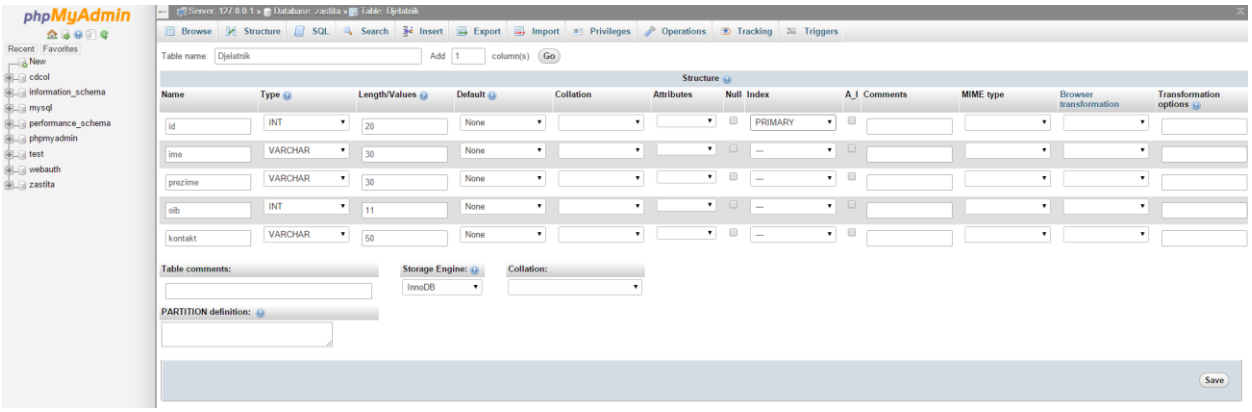
Na slici 3.3. i 3.4. prikazano je glavno početno sučelje programa „PHPMyAdmin“ kojemu se uz prethodno pokrenute servise „Apache“ i „MySQL“, pristupa upisom „localhost/phpmyadmin“ unutar web preglednika, za ovaj projekt koristio se web preglednik „Google Chrome“. Prva slika prikazuje polje za unos imena baze podataka te gumb „Create“ za samu izradu, dok druga slika predstavlja izradu prve tablice, što sam program ponudi nakon unosa imena baze podataka, dakle

prikazuje se izrada tablice, unošenje imena tablice te odabir broja atributa. Treba naglasiti da je prikazana izrada tablice djelatnik s 4 atributa te je naknadno dodan još jedan atribut.

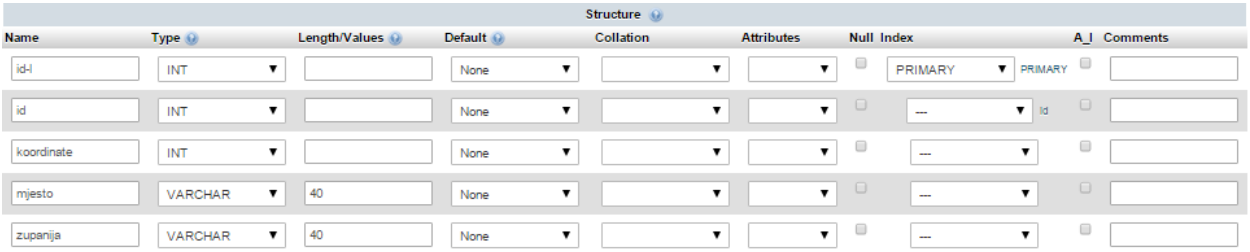


Sl. 3.5. prikaz praznih atributa tablice „djelatnik“

Slika 3.5. prikazuje polja zadanih atributa, prikazana su prazna polja kako bi korisnik mogao unijeti imena atributa te sve bitne segmente istih, od samoga tipa podataka do određivanja primarnog ključa. Na slici 3.6 prikazani su isti atributi ali s unesenim podacima zadanih atributa.

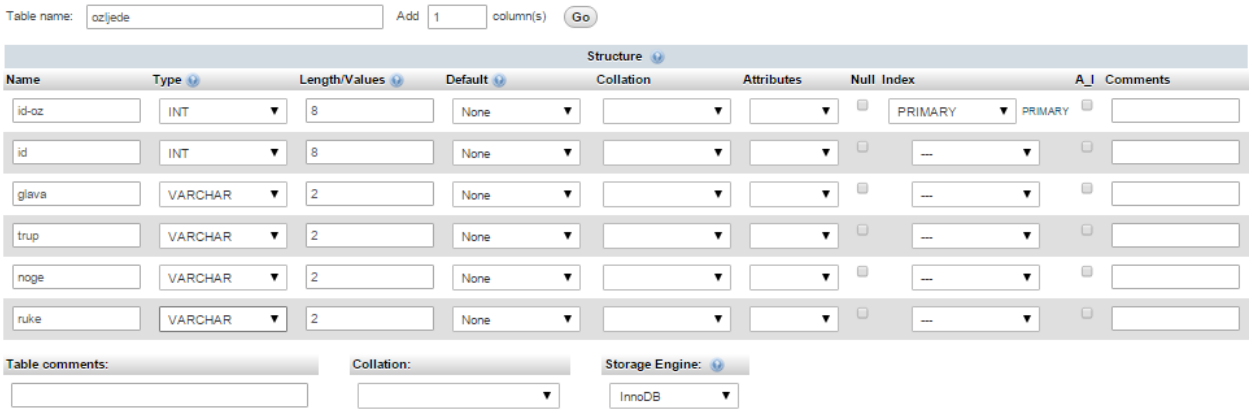


Sl. 3.6. prikaz izrade atributa tablice „djelatnik“



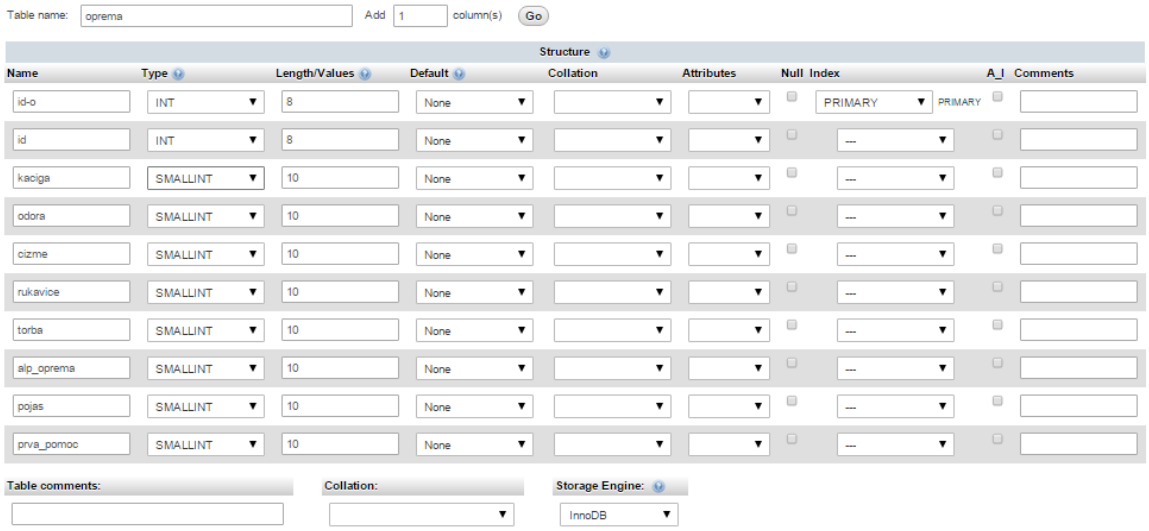
Sl. 3.7. prikaz izrade atributa tablice „lokacija“

Slika 3.7. prikazuje unošenje atributa u tablicu lokacije, vidljivo je unošenje tipa atributa, dali je int ili varchar ili neki drugi tip ovisno je o podacima koje unosimo, slika 3.8. također prikazuje unos atributa za tablicu ozljede.



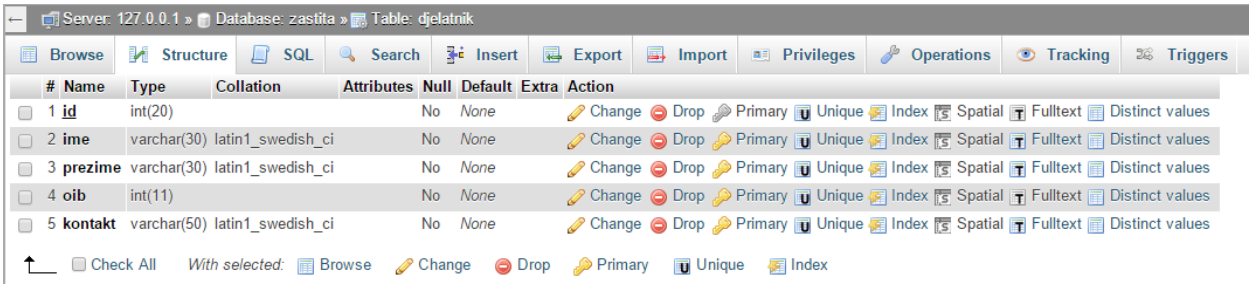
Sl. 3.8. prikaz izrade atributa tablice „ozljede“

Slika 3.9. prikazuje najveću tablicu s najviše atributa pod nazivom oprema.

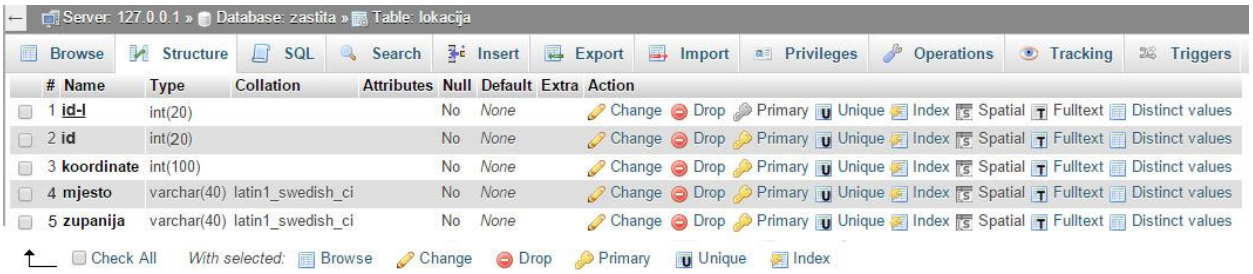


Sl. 3.9. prikaz izrade atributa tablice „oprema“

Unutar programa „PHPMyAdmin“ postoji mogućnost pregleda strukture pojedinih tablica putem gornjeg izbornika koji nudi različite mogućnosti i različite preglede do krajnjeg spremanja baze podataka u različitim oblicima i formatima po želji korisnika uz pomoć „Export“ menija. Slika 3.10. prikazuje strukturu tablice djelatnik, dok slika 3.11. prikazuje strukturu tablice lokacija.

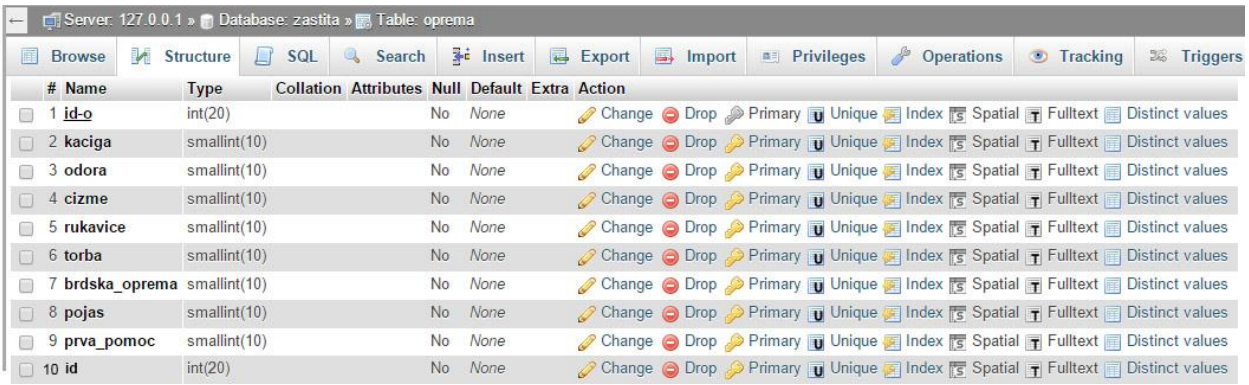


Sl. 3.10. prikaz strukture tablice „djelatnik“

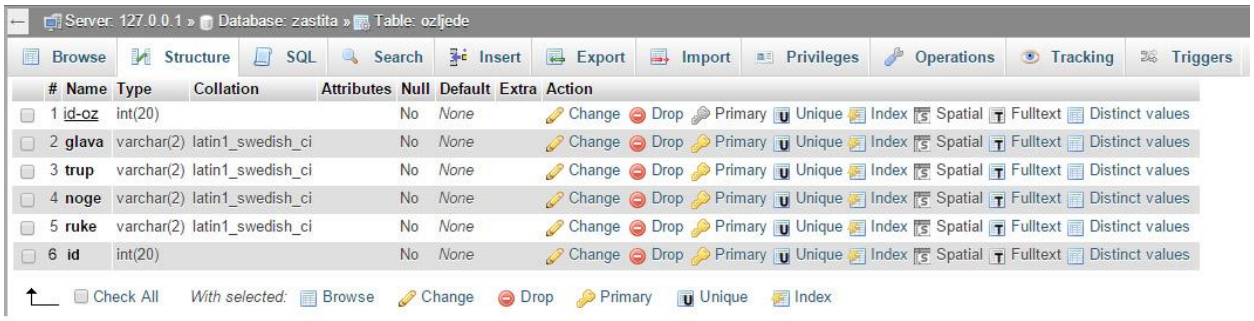


Sl. 3.11. prikaz strukture tablice „lokacija“

Kao i u prethodnim slikama slika 3.12. prikazuje strukturu tablice oprema, te slika 3.13. prikazuje strukturu tablice ozljede.



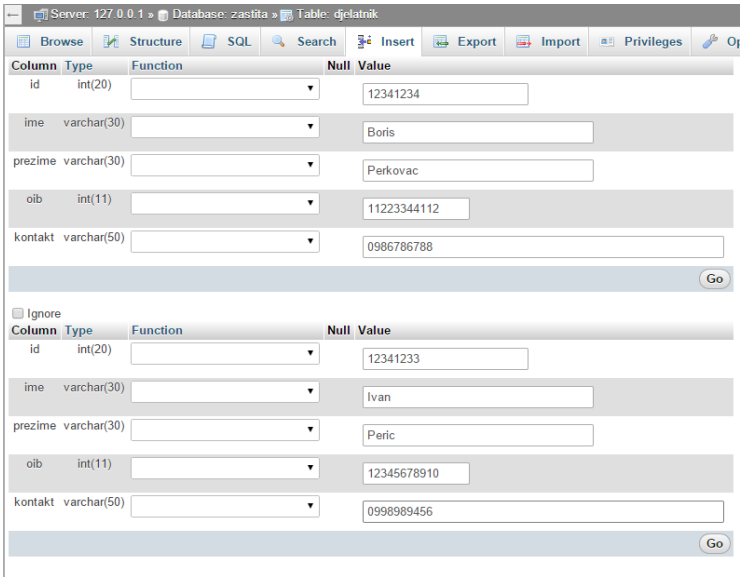
Sl. 3.12. prikaz strukture tablice „oprema“



Sl. 3.13. prikaz strukture tablice „ozljede“

4.7 Popunjavanje baze podataka

Nakon cjelokupnog prikaza struktura svih tablica započinje se s popunjavanjem tablica, prva tablica koja se popunjava je tablica djelatnik što je prikazano na slici 3.14. gdje je vidljivo sučelje za popunjavanje pojedinih podataka u pojedine atribute tablice djelatnik.



Sl. 3.14. prikaz popunjavanja atributa tablice „djelatnik“

Također kao i na slici 3.14., popunjavanje ostalih tablica je na isti princip i isto je sučelje samo se radi o tablici lokacija čijim je atributima omogućeno unošenje koordinata te se tako znalo točnu lokaciju na kojoj se djelatnik nalazi. U tablici za ozljede korisnik unosi samo krajnje „da“ ili „ne“ ako se radi o pojedinim ozljedama koje atributi nude. Tablica oprema nudi mogućnost zapisa podatka o tome koju je djelatnik opremu uzeo i u kolikim količinama.

5. ZAKLJUČAK

Baza podataka čini pohranu podataka nad kojima se mogu izvršiti pojedine aktivnosti. Baza podataka ne služi samo za pohranu već i za analizu i pregled istih ali valja naglasiti da uz to nudi i sve moguće izmjene i brisanja odnosno bolje rečeno upravljanje podacima. Ovaj dokument predstavlja opis i prikaz izrade baze podataka kao jedno rješenje u svrhu zaštite na radu, te se u konačnici može koristiti i u stvarnim potrebama.

Kako bi bilo moguće izraditi bazu podataka koristili su se programi „PHPMyAdmin“ koji se pokreće unutar web preglednika te se tako nudi korisnicima mogućnost same izrade baze podataka. Da bi sve to bilo moguće potrebno je instalirati program „XXAMP“ koji nudi mogućnost simuliranja poslužitelja na računalu uz pomoć servisa „Apache“ te „MySQL“-a. MySQL je sustav za upravljanje SQL bazama podataka. MySQL je vrlo siguran za korištenje..

Baza podataka „zastita“ sastoji se od 4 tablice unutar kojih se spremaju podatci:

- Podatci o djelatniku,
- Podatci o opremi,
- Podatci o lokaciji,
- Podatci o ozljedama.

LITERATURA

[1] R. Manger 'Baze podataka' – prvo izdanje, izdano: ožujka, 2013. godine

[2] R. Manger 'Baze podataka' – drugo izdanje, izdano: ožujka, 2013. godine

[3] M. Varga: Baze podataka – konceptualno, logičko i fizičko modeliranje podataka, DRIP, Zagreb, 1994.

[4] N. Slavek, Baze podataka - predavanja, izdano: veljača, 2013. godine

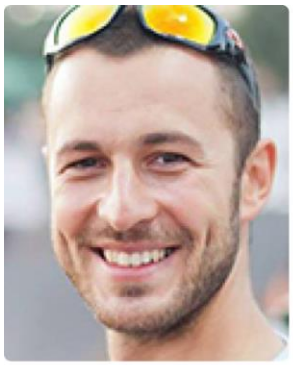
SAŽETAK

Tema rada je primjena baze podataka u svrhu zaštite na radu. Dokument predstavlja prvenstveno izradu baze podataka te fizički i logički model izrađene baze podataka, no uz to predstavlja i jedno rješenje u svrhu zaštite na radu. Baza podataka namijenjena je korisnicima koji se bave zaštitom na radu odnosno koji mogu pratiti svoje djelatnika u svim bitnim segmentima zaštite. Baza podataka može učinkovito poslužiti u evidentiranju podataka o samoj opremi djelatnika a također i o lokacijama koje je djelatnik pohađao te ako se ozlijedio o kakvoj se ozlijedi radilo, ukoliko je do nje došlo. Za izradu baze podataka iskorišteno je ranije stečeno znanje te uz samo zanimanje i bavljenje zaštitom na radu proizašao je ovaj projekt. Za samu izradu baze podataka bilo je potrebno znati jezik SQL te rad u programima XXAMP i PHPMyAdmin.

USAGE OF DATABASE IN SAFETY AT WORK

This work is about creating a database for the purpose of safety at work. The document represents primarily creating a database and physical and logical models of the database, but it also represents a solution to protect the work. The database is designed for users who deal with health and safety what can monitor their employees in all essential aspects of safety at work. The database can be effectively used to record information on the equipment staff and also to locations that the worker attended and if they get injured on what the injury was. To create the database I used previously acquired knowledge and by the interest and engagement in health and safety arising this project. For the purpose of creating a database, it was necessary to know SQL language and work in programs XXAMP and PHPMyAdmin.

ŽIVOTOPIS



Boris Perković

Vijenac Ivana Česmičkog 14,
31000 Osijek, Hrvatska

E-mail: perkovicboris@yahoo.com
Web stranica: www.borisperkovic.com,
https://vimeo.com/freshboxmedia
Telefon: +385 95 912 9627

OBRAZOVANJE

Diplomski studij računarstva 2011 — 2015
Elektrotehnički fakultet u Osijeku

Preddiplomski studij računarstva 2006 — 2011
Elektrotehnički fakultet u Osijeku

Tehničar za elektroenergetiku 2002 — 2006
Elektrotehnička i Prometna škola Osijek

PPE inspektor 2014 — 2014
Delta Plus
Kontrolor opreme za osobnu zaštitu radnika sa naglaskom na opremu za zaštitu od pada

Dizajniranje i postavljanje sustava Spider Line 2014 — 2014
Delta Plus
Dizajniranje i postavljanje sigurnosnog sustava Spider Line za spriječavanje pada sa visine, tvrtke Delta Plus.

KVALIFIKACIJE

- Rescue 3 - tehničar za spašavanje na divljim vodama
- Voditelj brodice
- Ronilac - CMAS
- Alpinist
- Biciklist
- Profesionalni fotograf i snimatelj
- Znanje stranog jezika - engleski (4/5)

- Visoko motiviran za rad u timu te kao dio pozitivnog kolektiva.
- Lako i brzo prilagođavanje i uklapanje u novo radno okruženje.
- Brzo učenje novih stvari

RADNO ISKUSTVO

Delta Plus — 2014 — Još uvijek

Instruktor

- Voditelj edukacija radnika: - korištenje osobne zaštitne opreme
- rad na visini
- spašavanje i izvlačenje

Fresh Box Media — 2013 — Još uvijek

Snimatelj

Osnivač i snimatelj

Hrvatska Gosrka Služba Spašavanja — 2011 — Još uvijek

Spašavatelj

Više tvrtki — 2012 — Još uvijek

Visinski radovi

- Pjeskarenje i nanošenje antikorozivne zaštite na silosima "Papuk Našice"
- Brušenje te nanošenje zaštite u bunarima "Vodovod Osijek"

- Kontrola instalacija i promijena signalne rasvijete na visokim objektima

- Čiščenje krovova od leda u zimskom periodu

Red Bull — 2009 — 2014

Fotograf

Extreme sport, Osijek — 2009 — 2011

Voditelj poslovnice

Voditelj poslovnice, prodavač i serviser bicikala

PRILOG

Kreiranje baze podataka:

```
CREATE DATABASE `zastita` DEFAULT CHARACTER SET latin1 COLLATE
latin1_swedish_ci;
```

Kreiranje tablice „djelatnik“

```
CREATE TABLE djelatnik(

`id` INT( 20 ) ,
`ime` VARCHAR( 30 ) NOT NULL ,
`prezime` VARCHAR( 30 ) NOT NULL ,
`oib` VARCHAR ( 11 ) ,
`kontakt` VARCHAR( 50 ) ,
PRIMARY KEY ( ` id ` )

) ENGINE = INNODB;
```

Kreiranje tablice „lokacija“

```
CREATE TABLE lokacija (
`id-l` INT( 20 ) ,
`id` INT( 20 ) ,
`koordinate` VARCHAR ( 100 ) ,
`mjesto` VARCHAR ( 40 ) NOT NULL ,
`zupanija` VARCHAR ( 40 ) NOT NULL ,
PRIMARY KEY ( ` id-l ` )
FOREGIN KEY (‘id’) REFERENCES djelatnik(‘id’)
) ENGINE = INNODB;
```

Kreiranje „ozljede“

```
CREATE TABLE `ozljede` (

`id-oz` INT( 20 ) ,
`id` INT( 20 ) ,
`glava` VARCHAR ( 100 ) NOT NULL ,
`trup` VARCHAR ( 100 ) NOT NULL ,
`noge` VARCHAR ( 100 ) NOT NULL ,
`ruke` VARCHAR ( 100 ) NOT NULL ,
PRIMARY KEY ( ` id-oz ` ),
FOREGIN KEY (‘id’) REFERENCES djelatnik(‘id’)

) ENGINE = INNODB;
```

Kreiranje tablice „oprema“

```
CREATE TABLE ` oprema ` (

`id-o` INT( 20 ) ,
`id` INT( 20 ) ,
`kaciga` SMALLINT ( 10 ) NOT NULL ,
`odora` SMALLINT ( 10 ) NOT NULL ,
`cizme` SMALLINT ( 10 ) NOT NULL ,
`rukavice` SMALLINT ( 10 ) NOT NULL ,
`torba` SMALLINT ( 10 ) NOT NULL ,
`brdska_oprema` SMALLINT ( 10 ) NOT NULL ,
`pojas` SMALLINT ( 10 ) NOT NULL ,
`prva_pomoc` SMALLINT ( 10 ) NOT NULL ,
PRIMARY KEY ( ` id-o ` ),
FOREGIN KEY (‘id’) REFERENCES djelatnik(‘id’)

) ENGINE = INNODB;
```

Unos podataka u tablicu „djelatnik“

```
INSERT INTO `zastita`.`djelatnik` (`id`, `ime`, `prezime`, `oib`
, `kontakt`) VALUES ('12341234', 'Boris', 'Perkovic', '112233441
12', '0986786788'), ('12341233', 'Ivan', 'Peric', '12345678910',
 '0998989456');
```

Unos podataka u tablicu „lokacija“

```
INSERT INTO `zastita`.`lokacija` (`id-
l`,`id`,`koordinate`, `mjesto`, `zupanija`)   VALUES ('12341234',
'54268540',    '44.794201    15.017323 ', 'Starigrad', 'splitsko-
dalmatinska'), ('44.022447      16.404991 ', 'Kijevo', 'sibensko-
kninska');
```

Unos podataka u tablicu „oprema“

```
INSERT INTO `zastita`.`oprema` (`id-o`,
`id`, `kaciga`, `odora`, `cizme`, `rukavice`, `torba`, `brdska_o
prema`, `pojas`, `prva_pomoc`) VALUES ('57487401',
'12341234', '2', '2', '4', '1', '1', '2', '1', '3'), ('12341233'
, '2', '2', '5', '1', '2', '2', '1', '4');
```

Unos podataka u tablicu „ozljede“

```
INSERT INTO `zastita`.`ozljede` (`id-oz`,
`id`, `glava`, `trup`, `noge`, `ruke`) VALUES ('98545264',
'12341234', 'ne', 'ne', 'ne', 'ne'), ('12341233', 'ne', 'ne', 'd
a', 'ne');
```