

Elektrotehnički fakultet Osijek
Stručni studij



Tehnička podrška

Projektni zadatak iz kolegija
Baze podataka

Osijek, 2011.

Ime Prezime, Axxxx

Sadržaj

1. Opis projektnog zadatka	1
2. ER dijagram	2
3. Relacijski model	3
4. Opis odabranog rješenja	5
5. SQL naredbe za kreiranje baze	6
6. Punjenje baze podacima.....	10
7. Upiti koji se najčešće koriste.....	12
8. Zaključak	16

Marenić

1. Opis projektnog zadatka

Projektni zadatak - "Tehnička podrška"

Opis:

Potrebno je osmisliti i realizirati bazu podataka koja može učinkovito poslužiti za potrebe tehničke podrške. U fazi analize, utvrđeno je da korisnik pruža tehničku podršku za pitanja vezana uz Oracle bazu podataka, te želi imati urednu evidenciju prijava, pri čemu svaka prijava može biti prijava greške ili tehničko pitanje (oznaka/redni broj, oznaka operatera, opis prijave, vrsta prijave, datum i vrijeme prijave, status...), evidenciju učinjenih koraka po pitanju prijave (tko je kada i što učinio - datum, vrijeme, akcija, ...). Nadalje, potrebno je imati evidenciju korisnika (naziv, ...) s adekvatnom oznakom važnosti (npr. gold, silver, bronze) koja definira prioritet i evidenciju tehničara koji se bave prijavama. Osnovni zahtjevi korisnika su slijedeći:

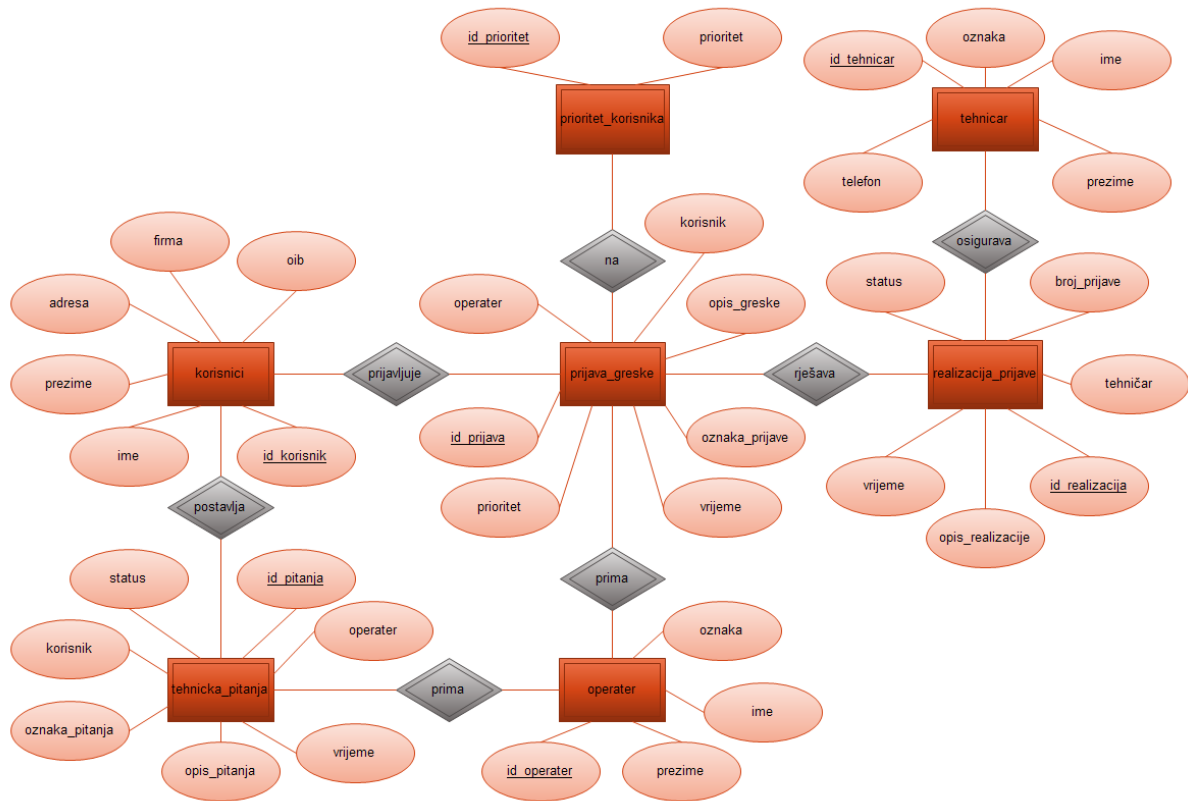
- omogućiti uvid u prijavljene greške i tehnička pitanja
- omogućiti prikaz otvorenih prijava prema važnosti (prioritetu) korisnika
- prikaz neobrađenih prijava (prema starosti, hitnosti, ...)
- prikaz riješenih prijava (količina, razdoblje, brzina rješavanja, najopterećeniji djelatnik, ...)
- prikaz trenutno zauzetih / slobodnih tehničara
- broj riješenih prijava po tehničaru za zadani vremenski period

Baza podataka treba biti u što većoj mjeri normalizirana.

Zadaci:

1. Napraviti ER dijagram (entity-relationship diagram) dijagram sa svim bitnim elementima.
2. Napraviti relacijski model
3. Dati kraći tekstualni opis odabranog rješenja uz osvrt na eventualne specifičnosti, pretpostavke ili ograničenja u modelu.
4. Napraviti SQL naredbe za kreiranje baze podataka koja odgovara relacijskom modelu.
5. Napraviti SQL naredbe kojima se baza puni podacima za potrebe testiranja.
6. Napraviti primjer SQL upita za koje se očekuje da će biti najčešće upotrebljavani od strane korisnika baze podataka (prema zahtjevima u opisu zadatka), opisati ih riječima (koja je svrha upita?) i dati konkretni primjer rezultata kakav se dobije takvim upitom.

2. ER dijagram



Slika 1. ER dijagram baze Tehnicka podrška

3. Relacijski model

korisnici				
Polje	Vrsta	Null	Zadano	Poveznica
id_korisnik	int(11)	No		
ime	varchar(50)	No		
prezime	varchar(50)	No		
adresa	varchar(50)	No		
oib	char(11)	No		
firma	varchar(50)	No		

operater				
Polje	Vrsta	Null	Zadano	Poveznica
id_operater	int(11)	No		
oznaka	char(4)	No		
ime	varchar(50)	No		
prezime	varchar(50)	No		

tehničar				
Polje	Vrsta	Null	Zadano	Poveznica
id_tehnicar	int(11)	No		
oznaka	char(4)	No		
ime	varchar(50)	No		
prezime	varchar(50)	No		
telefon	char(12)	No		

prioritet_korisnika				
Polje	Vrsta	Null	Zadano	Poveznica
id_prioritet	int(11)	No		
prioritet	varchar(10)	No		

realizacija_prijave				
Polje	Vrsta	Null	Zadano	Poveznica
id_realizacija	int(11)	No		
tehničar	int(11)	No		tehnicar -> id_tehnicar
broj_prijave	int(11)	No		prijava_greske -> id_prijava
opis_realizacije	varchar(60)	No		
vrijeme	timestamp	No	CURRENT_TIMESTAMP	
status	char(15)	No	REALIZIRANO	

prijava_greske				
Polje	Vrsta	Null	Zadano	Poveznica
id_prijava	int(11)	No		
oznaka_prijave	char(4)	No		
opis_greske	varchar(60)	No		
korisnik	int(11)	No		korisnici -> id_korisnik
prioritet	int(11)	No		prioritet_korisnika -> id_prioritet
vrijeme	timestamp	No	CURRENT_TIMESTAMP	
operater	int(11)	No		operater -> id_operater

tehnička_pitanja				
Polje	Vrsta	Null	Zadano	Poveznica
id_pitanja	int(11)	No		
oznaka_pitanja	char(4)	No		
korisnik	int(11)	No		korisnici -> id_korisnik
opis_pitanja	varchar(50)	No		
vrijeme	timestamp	No	CURRENT_TIMESTAMP	
operater	int(11)	No		korisnici -> id_korisnik
status	char(50)	No		

Tablice: Relacijski model baze Tehnicka podrška

4. Opis odabranog rješenja

tablica korisnici

Tablica za unos korisnika. Kako je tehnička služba namijenjena za firme, tako je potrebno unijeti podatke o njoj, oib i vlasnika. Za primarni ključ postavljen je id_korisnik. Tablica je povezana sa dva strana ključa: prijava_greske.korisnik i tehnicka_pitanja.korisnik.

tablica operater

Ova tablica sadrži podatke o operateru koji se javlja. Povezana je sa dva strana ključa prijava_greske.operater i tehnicka_pitanja.operater.

tablica tehnicar

Tablica za unos tehničara koji izlazi na teren. Kao primarni ključ postavljen je id_tehnicar, a povezan je sa stranim ključem realizacija_prijave.tehnicar.

tablica prioritet_korisnika

Mala tablica koja sadrži osnovne informacije o prioritetu otklanjanja kvara (gold, silver ili bronze).

tablica realizacija_prijave

Ova tablica služi za unos obavljenih poslova. U nju podatke unosi tehničar. On daje opis posla koji je napravio i da li je posao uspješno obavljen. Tablica ima dva strana ključa: tehnicar i broj_prijave.

tablica prijava_greske

Tablica za prijavu kvara i grešaka. Nju popunjava operater. Sastoji se od tri strana ključa: korisnik, prioritet i operater.

tablica tehnicka_pitanja

Ova tablica služi za unos tehničkih pitanja od strane korisnika postavljene operateru.

5. SQL naredbe za kreiranje baze

Kreiranje baze podataka

```
CREATE DATABASE `tehnicka_podrska` DEFAULT CHARACTER SET latin1 COLLATE latin1_swedish_ci;  
USE `tehnicka_podrska`;
```

Kreiranje tablice korisnici

```
CREATE TABLE IF NOT EXISTS `korisnici` (  
  `id_korisnik` int(11) NOT NULL AUTO_INCREMENT,  
  `ime` varchar(50) NOT NULL,  
  `prezime` varchar(50) NOT NULL,  
  `adresa` varchar(50) NOT NULL,  
  `oib` char(11) NOT NULL,  
  `firma` varchar(50) NOT NULL,  
  PRIMARY KEY (`id_korisnik`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=5 ;
```

Kreiranje tablice operater

```
CREATE TABLE IF NOT EXISTS `operater` (  
  `id_operater` int(11) NOT NULL AUTO_INCREMENT,  
  `oznaka` char(4) NOT NULL,  
  `ime` varchar(50) NOT NULL,  
  `prezime` varchar(50) NOT NULL,  
  PRIMARY KEY (`id_operater`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=4 ;
```


Kreiranje tablice prijava_greske

```
CREATE TABLE IF NOT EXISTS `prijava_greske` (  
  `id_prijava` int(11) NOT NULL AUTO_INCREMENT,  
  `oznaka_prijave` char(4) NOT NULL,  
  `opis_greske` varchar(60) NOT NULL,  
  `korisnik` int(11) NOT NULL,  
  `prioritet` int(11) NOT NULL,  
  `vrijeme` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE  
CURRENT_TIMESTAMP,  
  `operater` int(11) NOT NULL,  
  PRIMARY KEY (`id_prijava`),  
  KEY `korisnik` (`korisnik`),  
  KEY `prioritet` (`prioritet`),  
  KEY `operater` (`operater`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=5 ;
```

Kreiranje tablice prioritet_korisnika

```
CREATE TABLE IF NOT EXISTS `prioritet_korisnika` (  
  `id_prioritet` int(11) NOT NULL AUTO_INCREMENT,  
  `prioritet` varchar(10) NOT NULL,  
  PRIMARY KEY (`id_prioritet`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=4 ;
```

Kreiranje tablice realizacija_prijave

```
CREATE TABLE IF NOT EXISTS `realizacija_prijave` (  
  `id_realizacija` int(11) NOT NULL AUTO_INCREMENT,  
  `tehničar` int(11) NOT NULL,  
  `broj_prijave` int(11) NOT NULL,  
  `opis_realizacije` varchar(60) NOT NULL,  
  `vrijeme` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE  
CURRENT_TIMESTAMP,  
  `status` char(16) NOT NULL DEFAULT 'REALIZIRANO',  
  PRIMARY KEY (`id_realizacija`),  
  KEY `tehničar` (`tehničar`),  
  KEY `broj_prijave` (`broj_prijave`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=4 ;
```

Kreiranje tablice tehnicar

```
CREATE TABLE IF NOT EXISTS `tehnicar` (  
  `id_tehnicar` int(11) NOT NULL AUTO_INCREMENT,  
  `oznaka` char(4) NOT NULL,  
  `ime` varchar(50) NOT NULL,  
  `prezime` varchar(50) NOT NULL,  
  `telefon` char(12) NOT NULL,  
  PRIMARY KEY (`id_tehnicar`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=4 ;
```

Kreiranje tablice tehnicka_pitanja

```
CREATE TABLE IF NOT EXISTS `tehnicka_pitanja` (  
  `id_pitanja` int(11) NOT NULL AUTO_INCREMENT,  
  `oznaka_pitanja` char(4) NOT NULL,  
  `korisnik` int(11) NOT NULL,  
  `opis_pitanja` varchar(60) NOT NULL,  
  `vrijeme` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE  
CURRENT_TIMESTAMP,  
  `operater` int(11) NOT NULL,  
  `status` char(50) NOT NULL,  
  PRIMARY KEY (`id_pitanja`),  
  KEY `korisnik` (`korisnik`),  
  KEY `operater` (`operater`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=3 ;
```

Ograničenja

```
ALTER TABLE `prijava_greske`  
  ADD CONSTRAINT `prijava_greske_ibfk_1` FOREIGN KEY (`korisnik`)  
REFERENCES `korisnici` (`id_korisnik`) ON DELETE CASCADE ON UPDATE CASCADE,  
  ADD CONSTRAINT `prijava_greske_ibfk_2` FOREIGN KEY (`operater`)  
REFERENCES `operater` (`id_operater`) ON DELETE CASCADE ON UPDATE CASCADE,  
  ADD CONSTRAINT `prijava_greske_ibfk_3` FOREIGN KEY (`prioritet`)  
REFERENCES `prioritet_korisnika` (`id_prioritet`) ON DELETE CASCADE ON UPDATE  
CASCADE;
```

```
ALTER TABLE `realizacija_prijave`  
    ADD CONSTRAINT `realizacija_prijave_ibfk_1` FOREIGN KEY (`broj_prijave`)  
REFERENCES `prijava_greske` (`id_prijava`) ON DELETE CASCADE ON UPDATE  
CASCADE,  
    ADD CONSTRAINT `realizacija_prijave_ibfk_2` FOREIGN KEY (`tehničar`)  
REFERENCES `tehnicar` (`id_tehnicar`) ON DELETE CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE `tehnicka_pitanja`  
    ADD CONSTRAINT `tehnicka_pitanja_ibfk_1` FOREIGN KEY (`korisnik`)  
REFERENCES `korisnici` (`id_korisnik`) ON DELETE CASCADE ON UPDATE CASCADE,  
    ADD CONSTRAINT `tehnicka_pitanja_ibfk_2` FOREIGN KEY (`operater`)  
REFERENCES `operater` (`id_operater`) ON DELETE CASCADE ON UPDATE CASCADE;
```

Marehnic

6. Punjenje baze podacima

Punjenje tablice korisnici

```
INSERT INTO `korisnici` (`id_korisnik`, `ime`, `prezime`, `adresa`, `oib`, `firma`) VALUES  
(1, 'Tomislav', 'Kovac', 'Ulica Bagrema 6', '56895362561', 'TK Promet'),  
(2, 'Ivan', 'Maric', 'Kralja Tomislava 8', '47836889633', 'RTK'),  
(3, 'Vlado', 'Grgic', 'Trg Velikana 9', '74586159632', 'SF Nabava'),  
(4, 'Pero', 'Peric', 'Crkvena 11', '78512365333', 'TradeIN');
```

Punjenje tablice operater

```
INSERT INTO `operater` (`id_operater`, `oznaka`, `ime`, `prezime`) VALUES  
(1, 'MF01', 'Marija', 'Fijacko'),  
(2, 'TK02', 'Tomislav', 'Kovacic'),  
(3, 'AP03', 'Ankica', 'Premasunac');
```

Punjenje tablice prijava_greske

```
INSERT INTO `prijava_greske` (`id_prijava`, `oznaka_prijave`, `opis_greske`, `korisnik`,  
`prioritet`, `vrijeme`, `operater`) VALUES  
(1, 'PG01', 'serveru ne radi redundancija', 1, 1, '2011-02-07 19:29:54', 2),  
(2, 'PG02', 'printer se nakon pola ispisane stranice ugasi', 4, 3, '2011-02-07 19:33:54',  
1),  
(3, 'PG03', 'potrebno redovito odrzavanje racunala', 3, 2, '2011-02-08 12:29:54', 2),  
(4, 'PG04', 'ne radi internet', 2, 3, '2011-02-08 19:29:54', 3),  
(5, 'PG05', 'racunalo R1 se ne pali', 2, 2, '2011-02-08 15:28:31', 1);
```

Punjenje tablice prioritet_korisnika

```
INSERT INTO `prioritet_korisnika` (`id_prioritet`, `prioritet`) VALUES  
(1, 'gold'),  
(2, 'silver'),  
(3, 'bronze');
```

Punjenje tablice realizacija_prijave

```
INSERT INTO `realizacija_prijave` (`id_realizacija`, `tehničar`, `broj_prijave`,  
`opis_realizacije`, `vrijeme`, `status`) VALUES  
(2, 2, 1, 'zamjenjen disk', '2011-02-07 19:38:34', 'REALIZIRANO'),  
(3, 1, 2, 'kvar nepoznat, doneseno u servis', '2011-02-08 12:38:42', 'NIJE  
REALIZIRANO'),  
(4, 2, 5, 'nepoznat kvar', '2011-02-08 15:29:17', 'NIJE REALIZIRANO');
```

Punjenje tablice tehnicar

```
INSERT INTO `tehnicar` (`id_tehnicar`, `oznaka`, `ime`, `prezime`, `telefon`) VALUES  
(1, 'IP04', 'Ivan', 'Peric', '0989998686'),  
(2, 'JH05', 'Josip', 'Horvat', '0988009915'),  
(3, 'HF07', 'Hrvoje', 'Ficko', '0981118562'),  
(4, 'BB08', 'Boris', 'Boris', '0911569724');
```

Punjenje tablice tehnicka_pitanja

```
INSERT INTO `tehnicka_pitanja` (`id_pitanja`, `oznaka_pitanja`, `korisnik`,  
`opis_pitanja`, `vrijeme`, `operater`, `status`) VALUES  
(1, 'TP07', 3, 'kako zamjeniti toner u printeru', '2011-02-07 19:37:19', 1, 'odradeno,  
korisnik zadovoljan'),  
(2, 'TP08', 4, 'nemogucnost otvaranja pdf dokumenata', '2011-02-08 10:54:40', 2, 'nije  
odradeno, nuzan odlazak tehnicara'),  
(3, 'TP09', 4, 'da li moze zamjeniti disk DD02 za DD03', '2011-02-08 15:30:12', 1,  
'odradeno, korisnik zadovoljan');
```

7. Upiti koji se najčešće koriste

Ispis podataka iz tablice „korisnici“

```
SELECT `korisnici`.`ime`, `korisnici`.`prezime`, `korisnici`.`adresa`, `korisnici`.`oib`,  
`korisnici`.`firma`  
FROM `korisnici`
```

Naredbom SELECT dohvatili smo sve podatke osim id_korisnik. Nisu potrebne nikakve dodatne naredbe jer tablica ne sadrži strani ključ.

Ispis podataka iz tablice „operater“

```
SELECT `operater`.`oznaka`, `operater`.`ime`, `operater`.`prezime`  
FROM `operater`
```

Slično kao i gornja tablica. Također ispis bez id_operater i tablica nema strani ključ.

Ispis podataka iz tablice „tehnicar“

```
SELECT `tehnicar`.`oznaka`, `tehnicar`.`ime`, `tehnicar`.`prezime`, `tehnicar`.`telefon`  
FROM `tehnicar`
```

Ovim uvjetom ispisali smo sve tehničare koji rade u tvrtci.

Prikaz slobodnih tehničara

```
SELECT oznaka, ime, prezime, telefon  
FROM tehnicar  
LEFT JOIN realizacija_prijave ON (tehnicar.id_tehnicar=realizacija_prijave.tehnicar)  
WHERE realizacija_prijave.id_realizacija IS NULL
```

Da bi smo prikazali koji je korisnik slobodan poslužili smo se naredbom LEFT JOIN. Uvjet je bio da ispiše sve one redove koji se ne nalaze u tablici realizacija_prijave. To je omogućeno pomoću naredbe NULL.

Prikaz zauzetih tehničara

```
SELECT DISTINCT `tehnicar`.`oznaka`, `tehnicar`.`ime`, `tehnicar`.`prezime`,  
`tehnicar`.`telefon`  
FROM `tehnicar`  
JOIN realizacija_prijave WHERE `tehnicar`.`id_tehnicar`=`realizacija_prijave`.`tehnicar`
```

Ovaj gore upit ispisuje tehničare koji su zauzeti. Pomoću SELECT DISTINCT dohvatili smo podatke koji nisu dupli jer moguće je da tehničar ima dva i više popravka.

Ispis podataka iz tablice „prioritet_korisnika“

```
SELECT * FROM prioritet_korisnika
```

Ovim upitom biti će ispisani svi podaci u tablici prioritet_korisnika.

Ispis podataka iz tablice „realizacija_prijave“

Prikaz svih prijava (realizirane/ne realizirane)

```
SELECT DISTINCT `prijava_greske`.`oznaka_prijave`, `prijava_greske`.`opis_greske`,  
`korisnici`.`firma`, `prioritet_korisnika`.`prioritet`, `prijava_greske`.`vrijeme`,  
`operater`.`oznaka`  
FROM `operater`, `prijava_greske`, `korisnici`, `prioritet_korisnika`  
WHERE prijava_greske.korisnik=korisnici.id_korisnik AND  
prijava_greske.prioritet=prioritet_korisnika.id_prioritet AND  
prijava_greske.operater=operater.id_operater
```

Upitom se ispisuju vozila koja su slobodna. To je omogućeno preko naredbe IS NULL koja provjerava koja su polja slobodna.

Prikaz riješenih prijava

```
SELECT `prijava_greske`.`oznaka_prijave`, `prijava_greske`.`korisnik`, `realizacija_prijave`.`status`  
FROM `prijava_greske`  
JOIN realizacija_prijave  
WHERE realizacija_prijave.broj_prijave = prijava_greske.id_prijava  
AND realizacija_prijave.status = "realizirano"
```

Upit ispisuje prijave koje su riješene od strane tehničara.

Prikaz otvorenih prijava prema važnosti korisnika

```
SELECT DISTINCT `prijava_greske`.`oznaka_prijave` , `prijava_greske`.`korisnik` ,  
`realizacija_prijave`.`status` , `prioritet_korisnika`.`prioritet`  
FROM `prijava_greske`  
JOIN realizacija_prijave, prioritet_korisnika  
WHERE realizacija_prijave.broj_prijave = prijava_greske.id_prijava  
AND prijava_greske.prioritet=prioritet_korisnika.id_prioritet  
AND realizacija_prijave.status = "nije realizirano"  
ORDER BY `prioritet_korisnika`.`prioritet` ASC
```

Ovim upitom ispisujemo podatke koji su još u postupku realizacije ali nisu završeni. To je omogućeno tako što je postavljen uvjet za stupac status. Ispisani podaci poredani su prema prioritetu.

Prikaz neobrađenih prijava prema prioritetu

```
SELECT `prijava_greske`.`oznaka_prijave` , `prijava_greske`.`korisnik` ,  
`prijava_greske`.`prioritet`  
FROM `prijava_greske`  
LEFT JOIN realizacija_prijave  
ON (realizacija_prijave.broj_prijave = prijava_greske.id_prijava)  
WHERE realizacija_prijave.id_realizacija IS NULL  
ORDER BY `prijava_greske`.`prioritet` DESC
```

Slično kao i gornji upit samo što se ispisuju podaci o prijava koje još nisu ušle u proces obrade. Dakle njima nije pristupio tehničar. To je omogućeno pomoću IS NULL naredbe. Ispis je poredan prema prioritetu.

Prikaz neobrađenih prijava prema starosti (ASC)

```
SELECT `prijava_greske`.`oznaka_prijave` , `prijava_greske`.`korisnik` , `prijava_greske`  
`.`vrijeme` , `prijava_greske`.`prioritet`  
FROM `prijava_greske`  
LEFT JOIN realizacija_prijave ON ( realizacija_prijave.broj_prijave = prijava_greske.id_  
prijava )  
WHERE realizacija_prijave.id_realizacija IS NULL  
ORDER BY `prijava_greske`.`vrijeme` ASC
```

Ovaj ispis je isti kao gore samo što je ispis poredan prema vremenu primitka prijave.

Ispis podataka iz tablice „prijava_greske“

```
SELECT `prijava_greske`.`oznaka_prijave`, `prijava_greske`.`opis_greske`,  
`korisnici`.`firma`, `prioritet_korisnika`.`prioritet`, `prijava_greske`.`vrijeme`,  
`operater`.`oznaka`  
FROM `operater`, `prijava_greske`, `korisnici`, `prioritet_korisnika`  
WHERE korisnici.id_korisnik=prijava_greske.korisnik  
AND prioritet_korisnika.id_prioritet=prijava_greske.prioritet  
AND operater.id_operater=prijava_greske.operater
```

Upit ispisuje sve podatke u tablici prijava_greske. Pomoću naredbe WHERE postavljeni su uvjeti prema id-u pa je ispis podataka ljepši, bez brojeva dodijeljenih stranim ključem.

Ispis podataka iz tablice „tehnicka_pitanja“

```
SELECT `tehnicka_pitanja`.`oznaka_pitanja`, `korisnici`.`firma`,  
`tehnicka_pitanja`.`opis_pitanja`, `tehnicka_pitanja`.`vrijeme`, `operater`.`oznaka`,  
`tehnicka_pitanja`.`status`  
FROM `tehnicka_pitanja`, `korisnici`, `operater`  
WHERE korisnici.id_korisnik=tehnicka_pitanja.korisnik  
AND operater.id_operater=tehnicka_pitanja.operater
```

Isto kao i gore samo što se ispisuju podaci iz tablice tehnicka_pitanja.

Izmjena podatka u tablici „tehnicka_pitanja“

```
UPDATE `tehnicka_podrska`.`tehnicka_pitanja` SET `status` = 'odradeno, korisnik  
zadovoljan' WHERE `tehnicka_pitanja`.`id_pitanja` =2 LIMIT 1 ;
```

Naredbom update mijenjamo podatke u tablici tehnicka_pitanja. U ovom slučaju izmijenili smo status jer je u međuvremenu tehničko pitanje riješeno.

Brisanje podatka u tablici „realizacija_prijave“

```
DELETE FROM `tehnicka_podrska`.`realizacija_prijave` WHERE  
`realizacija_prijave`.`id_realizacija` = 2 LIMIT 1
```

Ovom naredbom izbrisat ćemo realiziranu prijavu koja ima id_realizacija=2. LIMIT znači da će se izbrisati samo jedan red.

8. Zaključak

Bazu smo napravili u MySQL-u (Xampp). Ona je tipa InnoDB jer nam omogućava stvaranje stranih ključeva sa ograničenjima. Zadovoljena je većina uvjeta koji su postavljeni projektnim zadatkom.

Prilikom izrade upoznali smo se sa osnovama baza i nekim osnovnim naredbama: select, insert, update, where, join... Da bi smo ispitivali uvjete koristili smo program MySQL Query Browser. ER dijagram napravljen je u programu Edraw UML Diagram 5.1.

Marenić