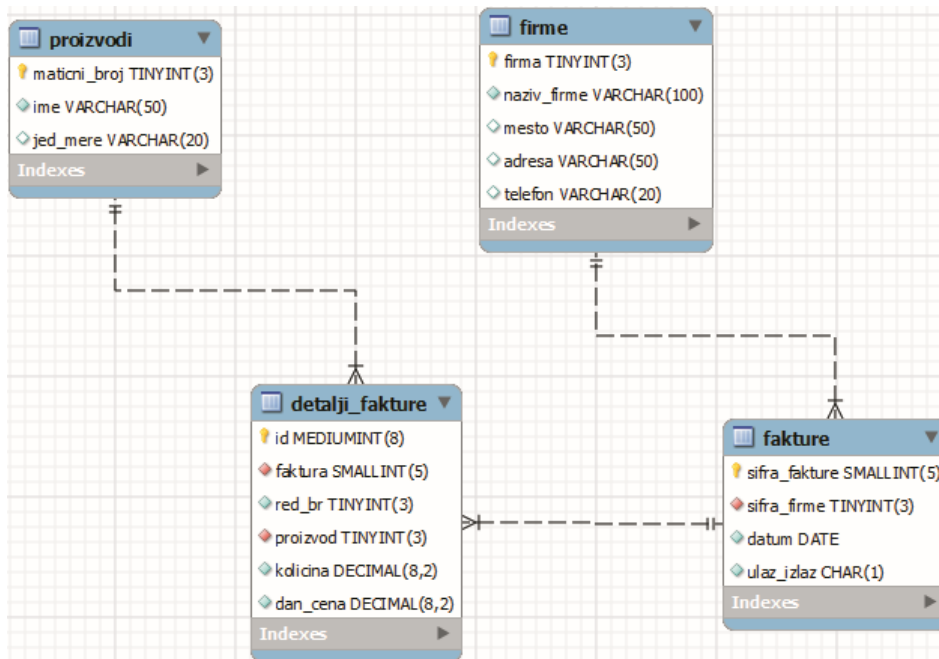


1. Baza podataka poslovanje

Izabran je primer baze podataka **poslovanje** koja služi za praćenje poslovanja jedne firme koja radi sa određenom grupom proizvoda (kupuje i prodaje proizvode) i saraduje sa određenim brojem drugih firmi. Glavni dokument koji prati poslovanje firme je faktura.

Baza podataka **poslovanje** ima četiri tabele i to: **proizvodi**, **firme**, **fakture** i **detalji_fakture**. Na sledećoj slici je prikazan EER dijagram baze podataka **poslovanje**. Može se videti da su sve uspostavljene veze tipa jedan-prema-više. Takođe se može videti da je **detalji_fakture** asocijativni entitet koji je iskorišćen za predstavljanje veze više-prema-više između entiteta **fakture** i **proizvodi**.



2. Data Manipulation Language (DML)

Jezik za rad sa podacima (*Data Manipulation Language (DML)*) služi za umetanje, brisanje i ažuriranje podataka u bazi.

2.1 Umetanje novih zapisa u tabelu – INSERT sintaksa

```
INSERT  
[INTO] tbl_name [(col_name,...)]  
VALUES ({expr | DEFAULT},...),(...),...
```

ili:

```
INSERT  
[INTO] tbl_name  
SET col_name={expr | DEFAULT}, ...
```

ili:

```
INSERT  
[INTO] tbl_name [(col_name,...)]  
SELECT ...
```

Komanda INSERT umeće zapise u postojeću tabelu. INSERT ... VALUES i INSERT ... SET forme iskaza umeću zapise na osnovu eksplicitno specificiranih vrednosti. INSERT ... SELECT forma umeće zapise selektovane iz jedne ili više tabela.

tbl_name je naziv tabelle u koju se umeću zapisi. Kolone za koje iskaz obezbeđuje vrednosti mogu biti specificirane na sledeći način:

- Može se proslediti lista naziva kolona razdvojenih zarezima iza naziva tabelle. U ovom slučaju vrednost za svaku imenovanu kolonu u listi mora biti obezbeđena VALUES listom ili SELECT iskazom.
- Ako se ne specificira lista naziva kolona za INSERT ... VALUES ili INSERT ... SELECT, vrednosti za svaku kolonu tabelle moraju biti obezbeđene VALUES listom ili SELECT iskazom
- SET klauzula omogućava da se izričito zadaju kolone u koje treba umetnuti podatke

Kada se koristi prvi oblik komande INSERT (INSERT ... VALUES) za svaki novi zapis tabelle se mora zadati lista vrednosti koje su poređane istim redosledom kao odgovarajuće ciljne kolone tabelle. Ovaj oblik omogućava da se jednom komandom INSERT umetne više novih zapisa u tabelu (više lista vrednosti u zagradama međusobno razdvojenih zarezima).

```
INSERT INTO tbl_name (a,b,c) VALUES (1,2,3),(4,5,6),(7,8,9);
```

Prethodni iskaz umeće tri zapisa u tabelu.

Sledeći iskaz je loš zato što broj vrednosti u listi ne odgovara broju naziva kolona:

```
INSERT INTO tbl_name (a,b,c) VALUES (1,2,3,4,5,6,7,8,9);
```

Drugi oblik komande INSERT omogućava da se izričito zadaju kolone u koje treba umetnuti podatke. Ovaj oblik omogućava da se unese samo jedan zapis po komandi, ali ne moraju se zadati vrednosti za sve kolone.

```
INSERT INTO firme
```

```
SET firma = 1,
```

```
naziv_firme = 'BALKAN';
```

Ovde se dodaje novi zapis u tabelu *firme*, ali se prosleđuju podaci samo za kolone *firma* i *naziv_firme*.

Primer za treći oblik komande INSERT (INSERT ... SELECT):

```
INSERT INTO fakture (sifra_firme)
```

```
SELECT firma FROM firme WHERE naziv_firme = 'STIL';
```

Ovde se dodaje novi zapis u tabelu *fakture* i prosleđuje se samo vrednost za kolonu *sifra_firme*, a ta vrednost se dobija iz tabelle *firme* tako što se selektuje vrednost za kolonu *firma* u zapisu u kome je vrednost za kolonu *naziv_firme* STIL. Ovo znači dodavanje nove fakture za firmu STIL.

Kolone za koje nisu zadate vrednosti će preuzeti podrazumevane vrednosti (u kolonama u kojima su takve vrednosti definisane) ili vrednost NULL.

Komanda INSERT ima nekoliko neobaveznih odredaba:

- Može se izričito zadati da kolona treba da preuzme svoju podrazumevanu vrednost ako se umesto vrednosti za kolonu zada opcija DEFAULT

Moguće je specificirati izraz *expr* koji će obezbediti vrednost za kolonu.

```
INSERT INTO tbl_name (col1,col2) VALUES (15,col1*2);
```

Dodaje se jedan zapis tabeli sa vrednostima za kolonu *col1* 15 i kolonu *col2* 15*2=30.

Ako se prilikom umetanja jednog zapisa prosledi NULL vrednosti koloni koja je deklarirana kao NOT NULL iskaz neće biti izvršen i javiće se poruka o grešci. Prosleđivanjem NULL vrednosti koloni koja je deklarirana kao NOT NULL prilikom umetanja većeg broja zapisa ili kada se koristi INSERT ... SELECT oblik naredbe vrednost u koloni se podešava na implicitnu podrazumevanu vrednost za tip podataka u toj koloni. Ovo je 0 za numeričke tipove, prazan string (") za znakovne tipove i „nulta“ vrednost za datumske i vremenske tipove podataka.

Za kolone tipa AUTO_INCREMENT je moguće izričito zadati vrednost ili prepustiti MySQL-u da sam generiše neku vrednost.

2.1.1. Umetanje zapisa u tabele baze podataka poslovanje

Prilikom unošenja podataka u tabele baze podataka **poslovanje** treba voditi računa o tome da se podaci prvo unose u primarne (roditelj) tabele pa onda u sekundarne (dete) tabele. Podaci se mogu unositi u tabele ovim redosledom: **proizvodi, firme, fakture, detalji_fakture**.

Unos podataka u tabelu **proizvodi**:

INSERT INTO **proizvodi** VALUES

(1, 'Četka', 'kom.'),
 (2, 'Lak', 'lit.'),
 (3, 'Stiropor', 'm2'),
 (4, 'Gips', 'kg'),
 (5, 'Destilovana voda', 'kg'),
 (6, 'Šmirgla', 'kom.);

matični_broj	ime	jed_mere
1	Četka	kom.
2	Lak	lit.
3	Stiropor	m2
4	Gips	kg
5	Destilovana voda	kg
6	Šmirgla	kom.

Unos podataka u tabelu **firme**:

INSERT INTO **firme** VALUES

(1, 'BALKAN', 'Niš', 'Mokranjčeva 13', '018-522-854'),
 (2, 'STIL', 'Beograd', 'Takovska 10', '011-562-365'),
 (3, 'KOKOMAX', 'Niš', 'Dušanova 33', '018-352-666'),
 (4, 'HELIO', 'Subotica', 'Nikole Tesle 55', '027-555-125');

firma	naziv_firme	mesto	adresa	telefon
1	BALKAN	Niš	Mokranjčeva 13	018-522-854
2	STIL	Beograd	Takovska 10	011-562-365
3	KOKOMAX	Niš	Dušanova 33	018-352-666
4	HELIO	Subotica	Nikole Tesle 55	027-555-125

Unos podataka u tabelu **fakture**:

INSERT INTO **fakture** VALUES

(1, 2, '2006-10-25', '1'),
 (2, 4, '2006-10-28', '1'),
 (3, 1, '2006-10-25', '2'),
 (4, 3, '2006-10-29', '2'),
 (5, 3, '2006-10-25', '1'),
 (6, 1, '2006-11-06', '2'),

```
(7, 4, '2006-11-04', '2'),
(8, 2, '2006-10-23', '1'),
(9, 1, '2006-11-02', '2'),
(10, 4, '2006-10-08', '1'),
(11, 3, '2006-10-15', '1');
```

↑ sifra_fakture	sifra_firme	datum	ulaz_izlaz
1	2	2006-10-25	1
2	4	2006-10-28	1
3	1	2006-10-25	2
4	3	2006-10-29	2
5	3	2006-10-25	1
6	1	2006-11-06	2
7	4	2006-11-04	2
8	2	2006-10-23	1
9	1	2006-11-02	2
10	4	2006-10-08	1
11	3	2006-10-15	1

Unos podataka u tabelu **detalji_fakture**:

INSERT INTO **detalji_fakture** (faktura, red_br, proizvod, kolicina, dan_cena) VALUES

```
(1, 1, 3, 50.00, 200.00),
(2, 1, 5, 50.00, 25.00),
(2, 2, 6, 35.00, 60.00),
(2, 3, 3, 40.00, 220.00),
(3, 1, 1, 20.00, 75.00),
(3, 2, 2, 35.00, 300.00),
(3, 3, 3, 60.00, 200.00),
(3, 4, 4, 50.00, 170.00),
(4, 1, 4, 50.00, 190.00),
(4, 2, 1, 50.00, 80.00),
(5, 1, 5, 120.00, 20.00),
(5, 2, 4, 100.00, 150.00),
(5, 3, 6, 25.00, 80.00),
(6, 1, 4, 75.00, 200.00),
(6, 2, 6, 50.00, 85.00),
(6, 3, 2, 24.00, 330.00),
(6, 4, 3, 70.00, 230.00),
(6, 5, 5, 80.00, 30.00),
(7, 1, 2, 26.00, 310.00),
(8, 1, 6, 45.00, 65.00),
(8, 2, 3, 50.00, 250.00),
(9, 1, 1, 50.00, 70.00),
(9, 2, 5, 25.00, 25.00),
(9, 3, 2, 14.00, 320.00),
(10, 1, 4, 150.00, 210.00),
(10, 2, 5, 200.00, 35.00),
(10, 3, 1, 125.00, 90.00),
(10, 4, 2, 100.00, 280.00),
(10, 5, 6, 60.00, 55.00),
(11, 1, 1, 5.00, 75.00);
```

↑ id	faktura	red_br	proizvod	kolicina	dan_cena
1	1	1	3	50.00	200.00
2	2	1	5	50.00	25.00
3	2	2	6	35.00	60.00
4	2	3	3	40.00	220.00
5	3	1	1	20.00	75.00
6	3	2	2	35.00	300.00
7	3	3	3	60.00	200.00
8	3	4	4	50.00	170.00
9	4	1	4	50.00	190.00
10	4	2	1	50.00	80.00
11	5	1	5	120.00	20.00
12	5	2	4	100.00	150.00
13	5	3	6	25.00	80.00
14	6	1	4	75.00	200.00
15	6	2	6	50.00	85.00
16	6	3	2	24.00	330.00
17	6	4	3	70.00	230.00
18	6	5	5	80.00	30.00
19	7	1	2	26.00	310.00
20	8	1	6	45.00	65.00
21	8	2	3	50.00	250.00
22	9	1	1	50.00	70.00
23	9	2	5	25.00	25.00
24	9	3	2	14.00	320.00
25	10	1	4	150.00	210.00
26	10	2	5	200.00	35.00
27	10	3	1	125.00	90.00
28	10	4	2	100.00	280.00
29	10	5	6	60.00	55.00
30	11	1	1	5.00	75.00

2.2. Brisanje zapisa iz tabele – DELETE sintaksa

```
DELETE FROM tbl_name
[WHERE where_condition]
[ORDER BY ...]
[LIMIT row_count]
```

DELETE iskaz briše zapise iz tabele *tbl_name* i vraća broj obrisanih redova. Ako se upotrebi WHERE klauzula brišu se samo zapisi koji zadovoljavaju uslov naveden u njoj. Ako se ne koristi WHERE klauzula brišu se svi zapisi. Ako je specificirana ORDER BY klauzula zapisi se brišu po redosledu koji je specificiran. LIMIT klauzula određuje maksimalan broj zapisa koji komanda DELETE sme da izbriše. Korisna je u kombinaciji sa klauzulom ORDER BY ili kada se želi da se spreči brisanje prevelikog broja zapisa. ORDER BY se koristi u kombinaciji sa LIMIT kada se na primer želi da se u tabeli izbriše samo n najstarijih zapisa.

```
DELETE FROM firme WHERE mesto = 'Niš';
```

Ovde se brišu zapisi iz tabele *firme* u kojima je vrednost u koloni *mesto* Niš (drugim rečima brišu se sve firme iz Niša). Ovde se brišu zapisi iz tri tabele baze podataka zbog opcije ON DELETE CASCADE za sve strane ključeve u svim tabelama baze!!!

2.3. Brisanje svih zapisa iz tabele – TRUNCATE sintaksa

```
TRUNCATE [TABLE] tbl_name;
```

Ova komanda je brža od komande DELETE jer radi tako što najpre fizički uklanja celu tabelu, a zatim pravi istu takvu ali praznu. Ovo je brže od brisanja svih zapisa red po red.

2.4. Ažuriranje zapisa u tabeli – UPDATE sintaksa

```
UPDATE tbl_name  
SET col_name1=expr1 [, col_name2=expr2 ...]  
[WHERE where_condition]  
[ORDER BY ...]  
[LIMIT row_count]
```

```
UPDATE firme  
SET adresa = 'Obrenovićeva 10',  
telefon = '011-463-375'  
WHERE naziv_firme = 'STIL';  
Ovde se menjaju adresa i telefon za firmu STIL u tabeli firme.
```

Komanda UPDATE je po mnogo čemu slična komandi DELETE. Može se dodati neobavezna opcija WHERE da bi se ažurirali samo određeni zapisi, a ako se izostavi biće ažurirani svi zapisi tabele:

```
UPDATE user  
SET password = 'test';  
Ovde se za sve korisnike postavlja vrednost test u koloni password.
```

```
ALTER TABLE detalji_fakture  
ADD COLUMN iznos DECIMAL(12,2);  
UPDATE detalji_fakture  
SET iznos = kolicina * dan_cena;  
Ovde se dodaje kolona iznos tabeli detalji_fakture i nakon toga se za sve zapise te tabele računa iznos kao kolicina * dan_cena.
```

Odredbe ORDER BY i LIMIT deluju na isti način kao u komadi DELETE.