

```

CREATE TABLE firme (
  firma TINYINT(3) UNSIGNED NOT NULL,
  naziv_firme VARCHAR(100) NOT NULL,
  mesto VARCHAR(50) NOT NULL,
  adresa VARCHAR(50) NOT NULL,
  telefon VARCHAR(20) NOT NULL,
  PRIMARY KEY (firma)
);

CREATE TABLE fakture (
  sifra_fakture SMALLINT(5) UNSIGNED NOT NULL,
  sifra_firme TINYINT(3) UNSIGNED NOT NULL,
  datum DATE NOT NULL,
  ulaz_izlaz CHAR(1) NOT NULL,
  PRIMARY KEY (sifra_fakture),
  KEY k_sifra_firme (sifra_firme),
  CONSTRAINT fk_fakture_firme FOREIGN KEY (sifra_firme) REFERENCES firme (firma)
  ON DELETE CASCADE ON UPDATE CASCADE
);

ALTER TABLE fakture
ADD COLUMN primio VARCHAR(50);

CREATE INDEX k_naziv_firme ON firme (naziv_firme);

DROP INDEX k_naziv_firme ON firme;

CREATE VIEW v_fakture AS
SELECT fakture.sifra_fakture AS faktura, firme.naziv_firme AS firma,
fakture.datum AS datum,
IF(fakture.ulaz_izlaz = '1', 'ulazna', 'izlazna') AS tip
FROM fakture, firme
WHERE fakture.sifra_firme = firme.firma;

ALTER VIEW v_fakture AS
SELECT fakture.sifra_fakture AS faktura, firme.naziv_firme AS firma,
fakture.datum AS datum,
fakture.ulaz_izlaz AS tip
FROM fakture, firme
WHERE fakture.sifra_firme = firme.firma;

DROP VIEW v_fakture;

CREATE SCHEMA AUTHORIZATION oe
  CREATE TABLE new_product
    (color VARCHAR2(10) PRIMARY KEY, quantity NUMBER)
  CREATE VIEW new_product_view
    AS SELECT color, quantity FROM new_product WHERE color = 'RED'
  GRANT SELECT ON new_product_view TO hr;

DROP SCHEMA oe CASCADE;

```

```

CREATE PROCEDURE vrsta_faktura
(IN Tip VARCHAR(12), IN Ime VARCHAR(15))
NOT DETERMINISTIC
SQL SECURITY DEFINER
SELECT firme.naziv_firme, fakture.sifra_faktura,
fakture.datum, fakture.ulaz_izlaz
FROM faktura, firme
WHERE fakture.sifra_firme = firme.firma
AND fakture.ulaz_izlaz = Tip
AND firme.naziv_firme = Ime;

CALL vrsta_faktura(2, 'BALKAN');

DROP PROCEDURE vrsta_faktura;

DELIMITER //
CREATE FUNCTION factorial (n DECIMAL(3,0))
RETURNS DECIMAL(20,0)
DETERMINISTIC
BEGIN
DECLARE factorial DECIMAL(20,0) DEFAULT 1;
DECLARE counter DECIMAL(3,0);
SET counter = n;
factorial_loop: REPEAT
SET factorial = factorial * counter;
SET counter = counter - 1;
UNTIL counter = 1
END REPEAT;
RETURN factorial;
END //
DELIMITER ;

DROP FUNCTION factorial;

CREATE TRIGGER firme_after_del_tr AFTER DELETE ON firme
FOR EACH ROW
DELETE FROM faktura WHERE fakture.sifra_firme = old.firma;

DELETE FROM firme WHERE firma = 4;

DROP TRIGGER firme_after_del_tr;

GRANT SELECT, INSERT ON poslovanje.firme TO 'petar'@'160.99.22.200';

REVOKE INSERT ON poslovanje.firme TO 'petar'@'160.99.22.200';

INSERT INTO firme VALUES
(1, 'BALKAN', 'Niš', 'Mokranjčeva 13', '018-522-854'),
(2, 'STIL', 'Beograd', 'Takovska 10', '011-562-365');

DELETE FROM firme WHERE mesto = 'Niš';

```

```
UPDATE firme
SET adresa = 'Obrenovićeva 10',
telefon = '011-463-375'
WHERE naziv_firme = 'STIL';
```

```
SELECT naziv_firme AS naziv
FROM firme;
```

```
SELECT naziv_firme, adresa, telefon
FROM firme
WHERE mesto = 'Niš';
```

```
SELECT * FROM fakture
WHERE ulaz_izlaz = 1 AND datum > '2006-10-23';
```

```
SELECT DISTINCT mesto FROM firme;
```

```
SELECT COUNT(*) AS broj_faktura, sifra_firme
FROM fakture
GROUP BY sifra_firme;
```

```
SELECT COUNT(*) AS broj_faktura, sifra_firme
FROM fakture
GROUP BY sifra_firme
HAVING broj_faktura > 2;
```

```
SELECT *
FROM firme
ORDER BY mesto ASC, naziv_firme DESC;
```

```
SELECT fakture.sifra_faktura, firme.naziv_firme, fakture.datum, fakture.ulaz_izlaz
FROM fakture JOIN firme
ON fakture.sifra_firme = firme.firma;
```

```
SELECT firme.naziv_firme, firme.mesto, firme.adresa, fakture.sifra_faktura, fakture.datum,
fakture.ulaz_izlaz
FROM firme LEFT JOIN fakture
ON firme.firma = fakture.sifra_firme;
```