

Osnovne Procedure

1. **Kreiranje procedure za prikaz svih proizvoda**

```
DELIMITER $$  
CREATE PROCEDURE `PrikaziSveProizvode`()  
BEGIN  
    SELECT * FROM Proizvod;  
END$$  
DELIMITER ;  
CALL PrikaziSveProizvode();
```

2. **Kreiranje procedure za prikaz svih kupaca**

```
DELIMITER $$  
CREATE PROCEDURE `PrikaziSveKupce`()  
BEGIN  
    SELECT * FROM Kupac;  
END$$  
DELIMITER ;  
CALL PrikaziSveKupce();
```

3. **Kreiranje procedure za prikaz ukupnog broja prodanih proizvoda**

```
DELIMITER $$  
CREATE PROCEDURE `UkupanBrojProdanihProizvoda`()  
BEGIN  
    SELECT SUM(Kolicina) AS UkupnoProdano FROM Stavka;  
END$$  
DELIMITER ;  
CALL UkupanBrojProdanihProizvoda();
```

4. Kreiranje procedure za dobivanje imena kupca na temelju ID-a

```
DELIMITER $$
```

```
CREATE PROCEDURE `DohvatilmeKupca`(IN KupacID INT)
```

```
BEGIN
```

```
    SELECT Ime FROM Kupac WHERE IDKupac = KupacID;
```

```
END$$
```

```
DELIMITER ;
```

```
CALL DohvatilmeKupca(1);
```

5. Kreiranje procedure za ažuriranje cijene proizvoda

```
DELIMITER $$
```

```
CREATE PROCEDURE `AzurirajCijenuProizvoda`(IN ProizvodID INT, IN  
NovaCijena DECIMAL(10,2))
```

```
BEGIN
```

```
    UPDATE Proizvod SET CijenaBezPDV = NovaCijena WHERE IDProizvod =  
ProizvodID;
```

```
END$$
```

```
DELIMITER ;
```

Srednje Složene Procedure

6. Kreiranje procedure za prikaz svih proizvoda određene kategorije

```
DELIMITER $$
```

```
CREATE PROCEDURE `ProizvodiPoKategoriji`(IN KategorijaNaziv  
VARCHAR(255))
```

```
BEGIN
```

```
    SELECT Proizvod.* FROM Proizvod
```

```
    JOIN Potkategorija ON Proizvod.PotkategorijaID =  
Potkategorija.IDPotkategorija
```

```
    JOIN Kategorija ON Potkategorija.KategorijaID = Kategorija.IDKategorija
```

```
    WHERE Kategorija.Naziv = KategorijaNaziv;
```

```
END$$
```

```
DELIMITER ;
```

```
CALL ProizvodiPoKategoriji('Elektronika');
```

7. **Kreiranje procedure za izračun ukupne vrijednosti svih računa**

```
DELIMITER $$
```

```
CREATE PROCEDURE `UkupnaVrijednostRacuna`()
```

```
BEGIN
```

```
    SELECT Racun.IDRacun, SUM(Stavka.Kolicina * Stavka.CijenaPoKomadu) AS  
    UkupnaVrijednost
```

```
    FROM Racun
```

```
    JOIN Stavka ON Racun.IDRacun = Stavka.RacunID
```

```
    GROUP BY Racun.IDRacun;
```

```
END$$
```

```
DELIMITER ;
```

```
CALL UkupnaVrijednostRacuna();
```

8. **Kreiranje procedure za dohvaćanje broja računa po kupcu**

```
DELIMITER $$
```

```
CREATE PROCEDURE `BrojRacunaPoKupcu`(IN KupacID INT)
```

```
BEGIN
```

```
    SELECT COUNT(*) AS BrojRacuna FROM Racun WHERE KupacID = KupacID;
```

```
END$$
```

```
DELIMITER ;
```

```
CALL BrojRacunaPoKupcu(1);
```

9. **Kreiranje procedure za prikaz proizvoda ispod određene cijene**

```
DELIMITER $$
```

```
CREATE PROCEDURE `ProizvodIspodCijene`(IN MaxCijena DECIMAL(10,2))
```

```
BEGIN
```

```
    SELECT * FROM Proizvod WHERE CijenaBezPDV < MaxCijena;
```

```
END$$
```

```
DELIMITER ;
```

```
CALL ProizvodIspodCijene(200.00);
```

10. Kreiranje procedure za ažuriranje zaliha proizvoda

```
DELIMITER $$  
  
CREATE PROCEDURE `AzurirajZalihe`(IN ProizvodID INT, IN Kolicina INT)  
  
BEGIN  
  
    UPDATE Proizvod SET MinimalnaKolicinaNaSkladistu = Kolicina WHERE  
    IDProizvod = ProizvodID;  
  
END$$  
  
DELIMITER ;  
  
CALL AzurirajZalihe(1, 100);
```

Složene Procedure

11. Kreiranje procedure za izračun ukupnog prometa po kategoriji

```
DELIMITER $$  
  
CREATE PROCEDURE `UkupniPrometPoKategoriji`(IN KategorijaID INT)  
  
BEGIN  
  
    SELECT Kategorija.Naziv, SUM(Stavka.Kolicina * Stavka.CijenaPoKomadu) AS  
    UkupniPromet  
  
    FROM Stavka  
  
    JOIN Proizvod ON Stavka.ProizvodID = Proizvod.IDProizvod  
  
    JOIN Potkategorija ON Proizvod.PotkategorijaID =  
    Potkategorija.IDPotkategorija  
  
    JOIN Kategorija ON Potkategorija.KategorijaID = Kategorija.IDKategorija  
  
    WHERE Kategorija.IDKategorija = KategorijaID  
  
    GROUP BY Kategorija.Naziv;  
  
END$$  
  
DELIMITER ;  
  
CALL UkupniPrometPoKategoriji(1);
```

12. Kreiranje procedure za dodavanje novog proizvoda

```
DELIMITER $$

CREATE PROCEDURE `DodajNoviProizvod`(IN Naziv VARCHAR(255), IN Cijena
DECIMAL(10,2), IN KategorijaID INT)

BEGIN

    INSERT INTO Proizvod (Naziv, CijenaBezPDV, PotkategorijaID) VALUES (Naziv,
Cijena, KategorijaID);

END$$

DELIMITER ;

CALL DodajNoviProizvod('Novi Proizvod', 99.99, 1);
```

13. Kreiranje procedure za brisanje proizvoda po ID-u

```
DELIMITER $$

CREATE PROCEDURE `ObrisiProizvod`(IN ProizvodID INT)

BEGIN

    DELETE FROM Proizvod WHERE IDProizvod = ProizvodID;

END$$

DELIMITER ;

CALL ObrisiProizvod(1);
```

14. Kreiranje procedure za izlistavanje kupaca i broja njihovih računa

```
DELIMITER $$

CREATE PROCEDURE `KupcilBrojRacuna`()

BEGIN

    SELECT Kupac.Ime, Kupac.Prezime, COUNT(Racun.IDRacun) AS BrojRacuna
    FROM Kupac
    LEFT JOIN Racun ON Kupac.IDKupac = Racun.KupacID
    GROUP BY Kupac.IDKupac;

END$$

DELIMITER ;

CALL KupcilBrojRacuna();
```

15. Kreiranje procedure za ažuriranje statusa stalnog zaposlenika za prodavače

DELIMITER \$\$

CREATE PROCEDURE `AzurirajStatusZaposlenika` (IN KomercijalistID INT, IN Status TINYINT)

BEGIN

UPDATE Komercijalist SET StalniZaposlenik = Status WHERE IDKomercijalist = KomercijalistID;

END\$\$

DELIMITER ;

CALL AzurirajStatusZaposlenika(1, 1);

16. Provjeri Zalihe i Naruči Proizvod

```
DELIMITER $$

CREATE PROCEDURE NaruciProizvod(
    IN ProizvodID INT,
    IN NarucenaKolicina INT,
    OUT StatusNarudzbe VARCHAR(50)
)
BEGIN
    DECLARE RaspolozivaKolicina INT;
    -- Dohvaćanje trenutne količine na zalihi za proizvod
    SELECT MinimalnaKolicinaNaSkladistu INTO RaspolozivaKolicina
    FROM Proizvod
    WHERE IDProizvod = ProizvodID;
    -- Provjera dostupnosti proizvoda
    IF RaspolozivaKolicina >= NarucenaKolicina THEN
        -- Ažuriranje zaliha ako je proizvod dostupan
        UPDATE Proizvod
        SET MinimalnaKolicinaNaSkladistu = MinimalnaKolicinaNaSkladistu -
        NarucenaKolicina
        WHERE IDProizvod = ProizvodID;
        -- Postavljanje statusa narudžbe na 'Uspješno'
        SET StatusNarudzbe = 'Uspješno';
    ELSE
        -- Postavljanje statusa narudžbe na 'Neuspješno' ako nema dovoljno
        zaliha
        SET StatusNarudzbe = 'Neuspješno - Nema dovoljno zaliha';
    END IF;
END$$

DELIMITER ;

CALL NaruciProizvod(1, 10, @StatusNarudzbe);

SELECT @StatusNarudzbe AS Status;
```

