

Funkcije – sintaksa

```
CREATE [ OR REPLACE ] FUNCTION
  funcname ( [ [ argmode ]
    [ argname ] argtype [, ...] ] )
  [ RETURNS rettype ]
AS 'definition'
LANGUAGE langname
```

+ dodatni atributi po potrebi

Upute

- U nastavku slijedi niz primjera implementacija funkcija.
- Svaku implementaciju dodajte u skriptu kreiranje.sql u okviru transakcije koja je kreirala tablice.
- Za svaku funkciju dodajte odgovarajuće brisanje u skriptu brisanje.sql

Primjer

- Funkcija **upisi_u_grupu** koja prima email korisnika i naziv grupe te upisuje u clanstvo da je zadani korisnik član zadane grupe

Rješenje

```
CREATE OR REPLACE FUNCTION upisi_u_grupu( email
TEXT, naziv_grupe TEXT )
  RETURNS VOID
  AS '
    INSERT INTO clanstvo
    SELECT $1, sifra
    FROM grupa
    WHERE naziv = $2
  '
  LANGUAGE SQL;
```

Isprobavanje

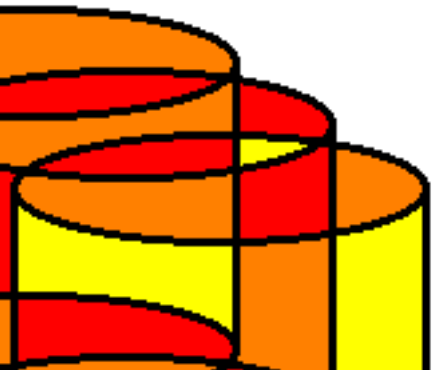
```
SELECT upisi_u_grupu(  
    'markus.schatten@foi.hr',  
    'Baze podataka' );
```

Brisanje

```
DROP FUNCTION upisi_u_grupu( TEXT,  
TEXT );
```

Upute

Ubuduće brisanje implementirajte samostalno!



Primjer

- Funkcija **ispisi_iz_grupe** koja prima email korisnika i naziv grupe te ispisuje zadanog korisnika iz zadane grupe

Rješenje

```
CREATE OR REPLACE FUNCTION ispisi_iz_grupe(  
    email TEXT, naziv_grupe TEXT )  
    RETURNS VOID  
    AS '  
        DELETE FROM clanstvo  
        WHERE clan = $1  
        AND grupa = (  
            SELECT sifra  
            FROM grupa  
            WHERE naziv = $2  
        )  
    '  
LANGUAGE SQL;
```

Isprobavanje

```
SELECT ispisi_iz_grupe(  
    'markus.schatten@foi.hr',  
    'Baze podataka' );
```

Zadatak

- Implementirajte funkciju **posalji_poruku** koja prima naslov i sadržaj poruke, e-mail primatelja, email posiljatelja, te poruku zapisuje u tablicu poruka
- Rješenje također dodajte u skriptu kreiranje.sql odnosno brisanje.sql

Isprobavanje

```
SELECT posalji_poruku(  
  'markus.schatten@foi.hr',  
  'mirko.malekovic@foi.hr',  
  'Postovanje',  
  'Kako ste?' );
```

Primjer

- Funkcija **moje_grupe** koja prima e-mail korisnika i vraća nazive grupa kojih je isti član

Rješenje

```
CREATE OR REPLACE FUNCTION moje_grupe(  
    email TEXT )  
    RETURNS SETOF TEXT  
    AS '  
        SELECT naziv  
        FROM grupa  
        WHERE sifra IN (  
            SELECT grupa  
            FROM clanstvo  
            WHERE clan = $1  
        )  
    ;  
LANGUAGE SQL;
```

Isprobavanje

```
SELECT moje_grupe(  
    'markus.schatten@foi.hr' );
```

Primjer

- Funkcija **moje_grupe_2** koja prima e-mail korisika i vraća šifre i nazive grupa kojih je isti član

Rješenje

```
CREATE OR REPLACE FUNCTION moje_grupe_2(  
    email TEXT )  
    RETURNS SETOF grupa  
    AS '  
        SELECT *  
        FROM grupa  
        WHERE sifra IN (  
            SELECT grupa  
            FROM clanstvo  
            WHERE clan = $1  
        )  
    '  
LANGUAGE SQL;
```

Isprobavanje

```
SELECT  
  moje_grupe_2( 'markus.schatten@foi.hr' );
```

Isprobavanje

```
SELECT  
  moje_grupe_2( 'markus.schatten@foi.hr' );
```

```
SELECT *  
FROM moje_grupe_2( 'markus.schatten@foi.hr' );
```

Isprobavanje

```
SELECT  
  moje_grupe_2( 'markus.schatten@foi.hr' );
```

```
SELECT *  
FROM moje_grupe_2( 'markus.schatten@foi.hr' );
```

```
SELECT sifra  
FROM moje_grupe_2( 'markus.schatten@foi.hr' );
```

Primjer

- Funkcija **inbox** koja će primiti email korisnika i vraćati imena i prezimena osoba koje su tom korisniku slale poruke kao i šifre, naslove i vremena poruka sortirano prema vremenu slanja

Definicija povratnog tipa

```
CREATE TYPE inbox_popis AS (  
    ime TEXT,  
    prezime TEXT,  
    sifra INT,  
    naslov TEXT,  
    vrijeme TIMESTAMP  
);
```

Rješenje

```
CREATE OR REPLACE FUNCTION inbox( email TEXT )
  RETURNS SETOF inbox_popis
AS '
  SELECT o.ime, o.prezime, p.sifra,
         p.naslov, p.vrijeme
  FROM osoba o, poruka p
  WHERE o.email = p.posiljatelj
  AND p.primatelj = $1
  ORDER BY p.vrijeme
  '
LANGUAGE SQL;
```

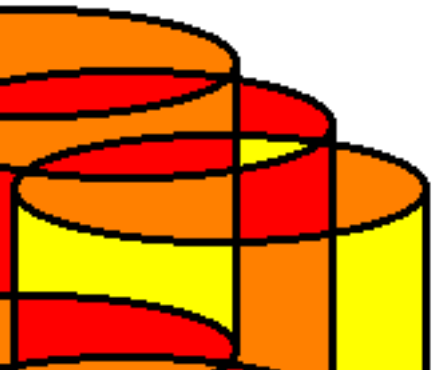
Isprobavanje

```
SELECT * FROM  
  inbox( 'mirko.malekovic@foi.hr' );
```


Primjer



- Funkcija **zamolba** koja šalje zamolbu za prijateljstvom za zadani e-mail korisnika i e-mail potencijalnog prijatelja, ali prije toga provjerava da takva zamolba/veza jos nije poslana
- Ako takva veza već postoji funkcija ne dopušta kreiranje i vraća pogrešku!



Proceduralni jezici

- Za vraćanje pogreški potreban nam je neki proceduralni jezik, npr. u PostgreSQL-u imamo:
 - C
 - PL/pgSQL
 - PL/Python
 - PL/Tcl
 - PL/Perl
 - PL/Java
 - PL/Ruby
 - PL/sh
 - PL/R
 - ...

Rješenje

1/3

```
CREATE OR REPLACE FUNCTION
zamolba( pošiljatelj TEXT, primatelj TEXT,
naziv_vrste TEXT )
RETURNS VOID
AS $$
DECLARE postoji BOOLEAN;
DECLARE sifra_vrste INT;
BEGIN
    sifra_vrste := (
        SELECT sifra
        FROM vrsta_veze
        WHERE naziv = naziv_vrste
    );
```

Rješenje

2/3

```
postoji := EXISTS(  
    SELECT veza  
    FROM veza  
    WHERE poslao = posiljatelj  
    AND prihvatio = primatelj  
    AND vrsta = sifra_vrste  
);  
IF NOT postoji THEN  
    INSERT INTO veza  
    VALUES ( posiljatelj, primatelj,  
             sifra_vrste, 'poslano' );
```

Rješenje

3/3

ELSE

**RAISE EXCEPTION '%', 'Vec postoji
takva veza';**

END IF;

END;

\$\$

LANGUAGE plpgsql;

Isprobavanje

```
SELECT zamołba(  
  'markus.schatten@foi.hr',  
  'mirko.malekovic@foi.hr',  
  'suradnik' );
```

Primjer

- Funkcija **nova_grupa** koja ce primiti e-mail kreatora i naziv nove grupe te kreirati novu grupu pod zadanim nazivom ukoliko takva ne postoji te postaviti kreatora za prvog člana
- Ako takva grupa već postoji javiti pogrešku

Rješenje

1/2

```
CREATE OR REPLACE FUNCTION
nova_grupa( kreator TEXT, naziv_grupe
TEXT )
  RETURNS VOID
  AS $$
  DECLARE postoji BOOLEAN;
  BEGIN
    postoji := EXISTS(
      SELECT naziv
      FROM grupa
      WHERE naziv = naziv_grupe
    );
```


Rješenje

2/2

```
IF NOT postoji THEN
    INSERT INTO grupa( naziv )
    VALUES ( naziv_grupe );
    PERFORM upisi_u_grupu(
        kreator, naziv_grupe );
ELSE
    RAISE EXCEPTION '%', 'Vec postoji
    grupa pod nazivom ' || naziv_grupe;
END IF;
END;
$$
LANGUAGE plpgsql;
```

Isprobavanje

```
SELECT nova_grupa(  
  'markus.schatten@foi.hr',  
  'Nastavnici' );
```