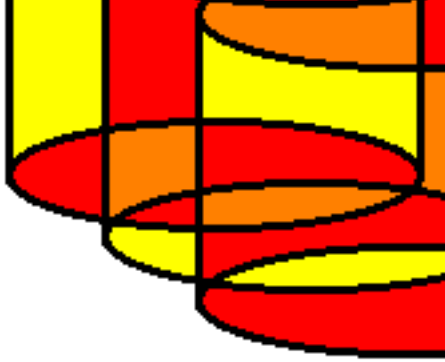
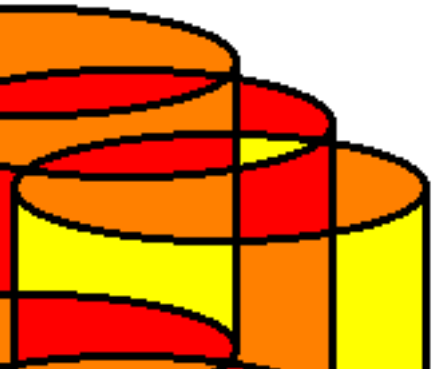
Poopćene relacijske baze podataka

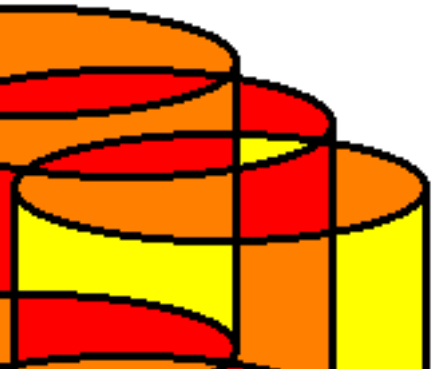


- PostgreSQL omogućava korištenje kompleksnijih objekata
  - BLOB
  - polja (1D, 2D)
  - pobrojene vrijednosti (ENUM)
  - složeni tipovi

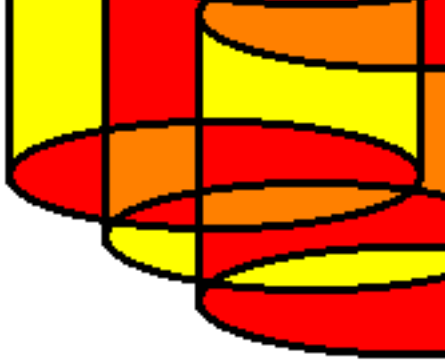


BLOB – Binary Large OBject

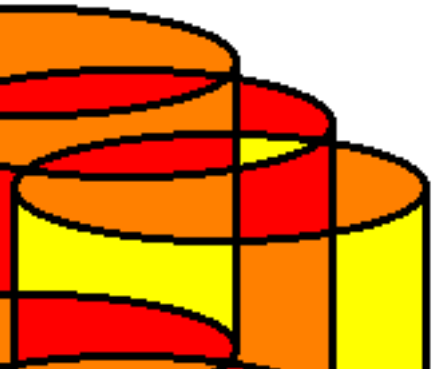
- Koristi se za pohranu raznih binarnih podataka poput različitih dokumenata, slika, videa ...
- Za takve objekte koristimo tip podataka **`OID`** (Object ID) te funkcije:
  - **`lo_import`** – za unos objekata
  - **`lo_export`** – za čitanje pohranjenih objekata
  - **`lo_unlink`** – za brisanje objekata



Primjer

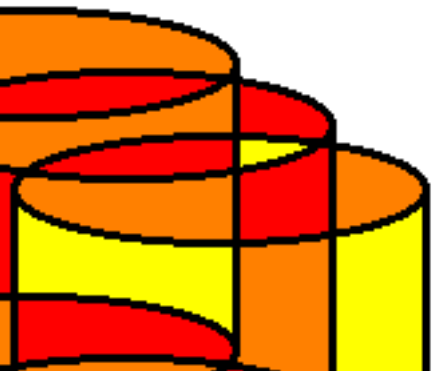


```
CREATE TABLE fotoalbum(
    sifra SERIAL PRIMARY KEY,
    opis TEXT,
    tip VARCHAR(3),
    slika OID
);
```



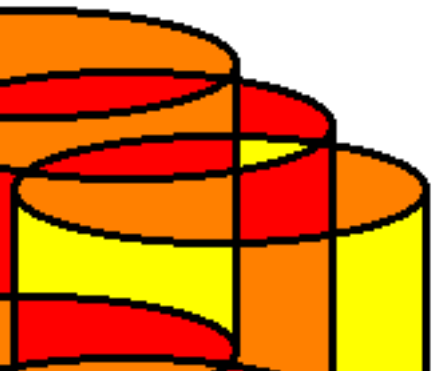
Primjer

```
INSERT INTO fotoalbum( opis, tip, slika )
  VALUES ( 'kupanje na plazi', 'png',
  lo_import(
  '/home/vjezbefoi/Documents/slika.png' )
  );
```

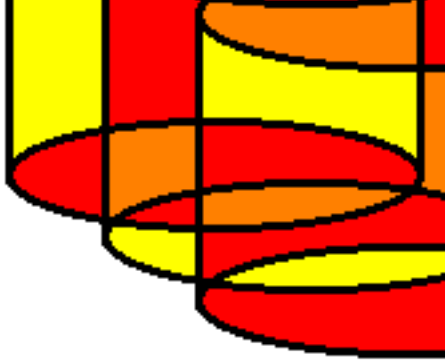


Primjer

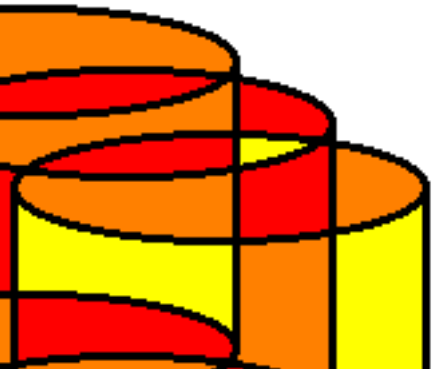
```
INSERT INTO fotoalbum( opis, tip, slika )
  VALUES ( 'kupanje na plazi', 'png',
  lo_import(
  '/home/vjezbefoi/Documents/slika.png' )
  );
SELECT * FROM fotoalbum;
```



Primjer

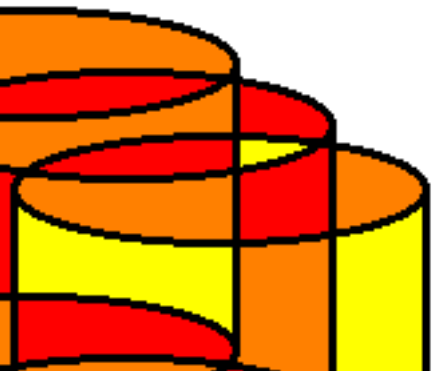


```
INSERT INTO fotoalbum( opis, tip, slika )
  VALUES ( 'kupanje na plazi', 'png',
  lo_import(
  '/home/vjezbefoi/Documents/slika.png' )
  );
SELECT * FROM fotoalbum;
SELECT lo_export( slika,'/tmp/mojaslika.'
  || tip ) FROM fotoalbum WHERE sifra = 1;
```

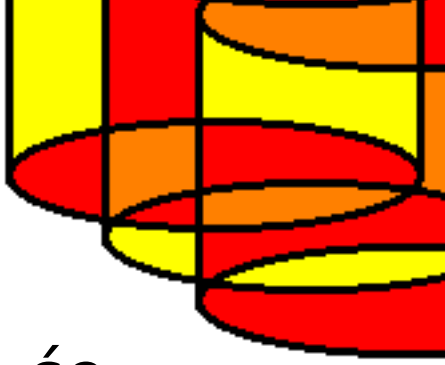


Primjer

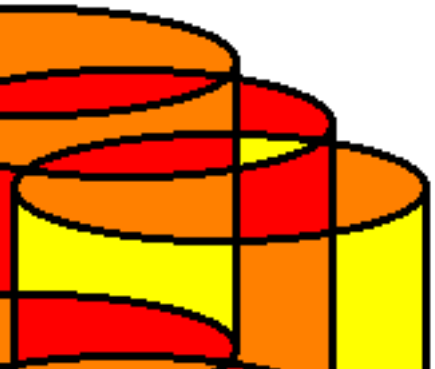
```
INSERT INTO fotoalbum( opis, tip, slika )
  VALUES ( 'kupanje na plazi', 'png',
  lo_import(
  '/home/vjezbefoi/Documents/slika.png' )
  );
SELECT * FROM fotoalbum;
SELECT lo_export( slika,'/tmp/mojaslika.'
  || tip ) FROM fotoalbum WHERE sifra = 1;
SELECT lo_unlink( slika ) FROM fotoalbum;
```



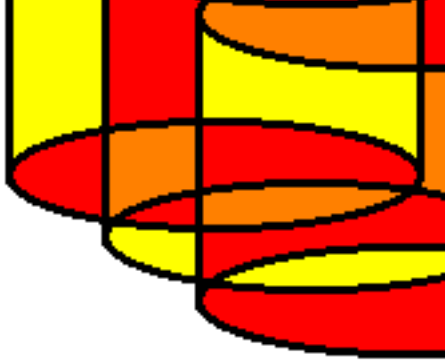
Zadatak



- Implementirajte tablicu **vlasnik_slike** koja će povezati tablicu **osoba** s tablicom **fotoalbum**.
- Implementirajte funkciju **nova_slika** koja će primati e-mail adresu korisnika, opis slike, tip slike i putanju do slike te zapisati u bazu podataka novu sliku i zabilježiti da je korisnik s tom e-mail adresom vlasnik nove slike.
- Implementaciju tablica **fotoalbum** i **vlasnik_slike** te funkcije **nova_slika** također dodajte u skripte kreiranje.sql odnosno brisanje.sql.

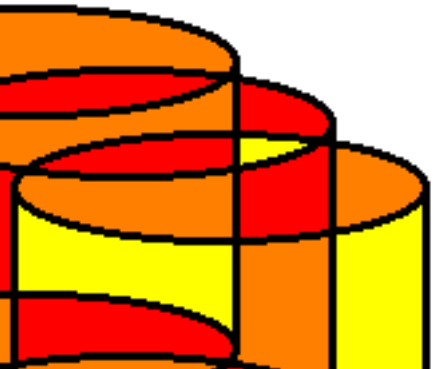


Polja

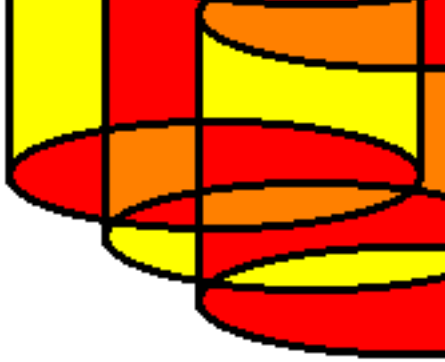


- Jednodimenzionalna polja

```
tip_podatka[]
tip_podatka[ dimenzija ]
```

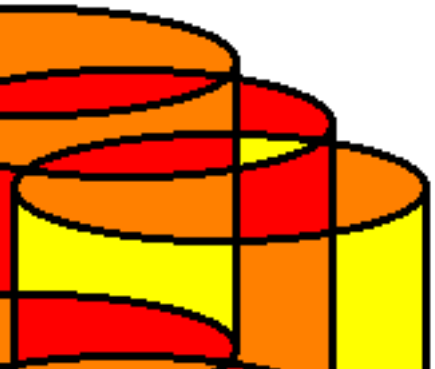


Polja



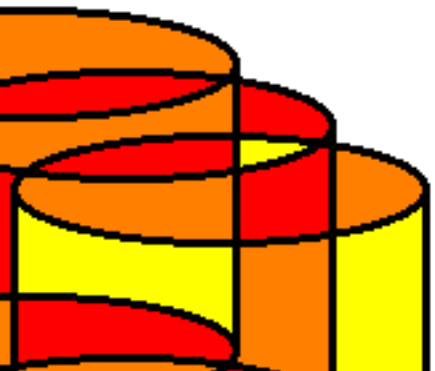
- Dvodimenzionalna polja

```
tip_podatka[][]

tip_podatka[ dimenzija1 ]
           [ dimenzija2 ]
```



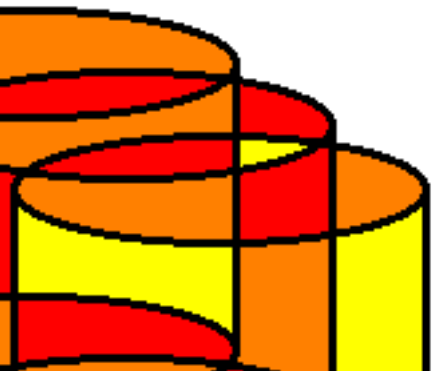
Primjer

```
CREATE TEMP TABLE
  evidencija_nastave(
     student VARCHAR( 20 ),
     dolasci BOOLEAN[]
);
```



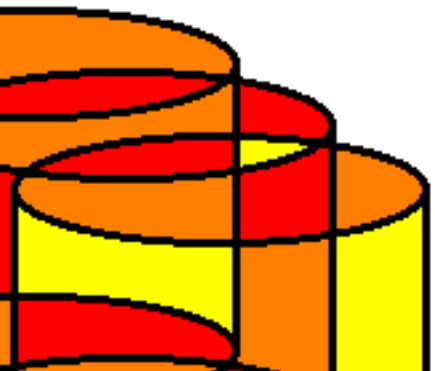
Primjer

```
INSERT INTO evidencija_nastave VALUES
  ( 'Anic', '{ TRUE, FALSE, FALSE }' );
```



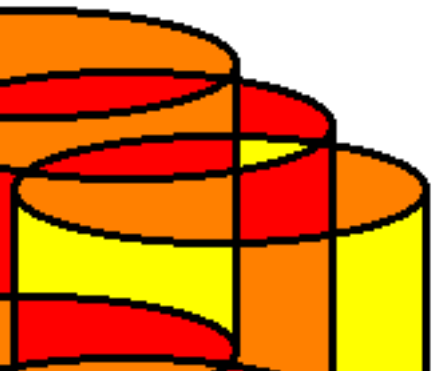
Primjer

```
INSERT INTO evidencija_nastave VALUES
  ( 'Anic', '{ TRUE, FALSE, FALSE }' );

INSERT INTO evidencija_nastave VALUES
  ( 'Ivic', '{ TRUE, TRUE, FALSE }' );
```



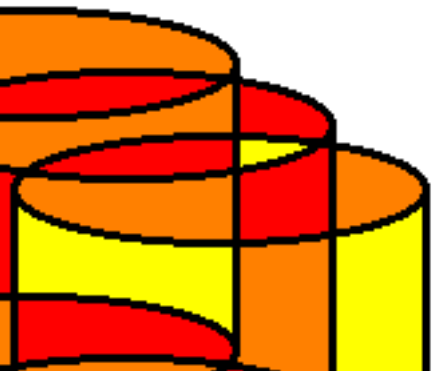
Primjer

```
SELECT * FROM evidencija_nastave;
```



Primjer

```
SELECT * FROM evidencija_nastave;


UPDATE evidencija_nastave
  SET dolasci[ 3 ] = TRUE
  WHERE student = 'Ivic';
```

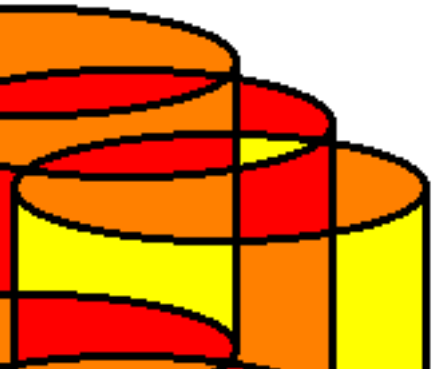


Primjer

```
SELECT * FROM evidencija_nastave;
```

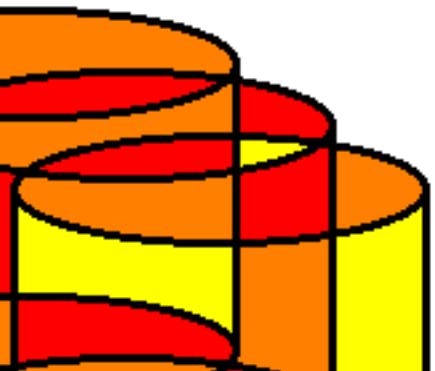
```
UPDATE evidencija_nastave
  SET dolasci[ 3 ] = TRUE
  WHERE student = 'Ivic';
```

```
SELECT * FROM evidencija_nastave;
```



Primjer

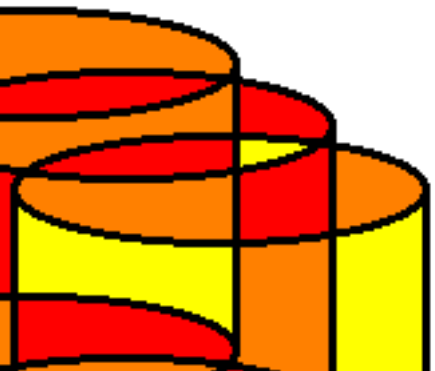
```
SELECT student, dolasci[ 1 ]
  FROM evidencija_nastave;
```



Primjer

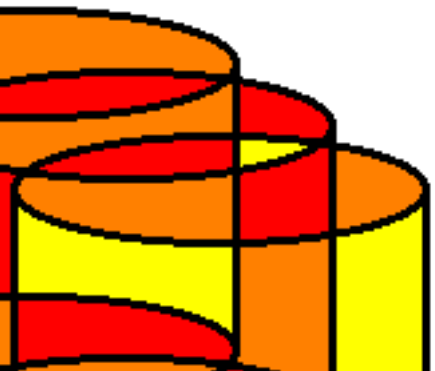
```
SELECT student, dolasci[ 1 ]
  FROM evidencija_nastave;
```

```
SELECT student, dolasci[ 1:2 ]
  FROM evidencija_nastave;
```



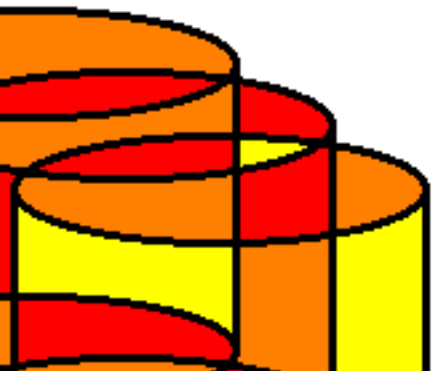
Primjer

```
SELECT student, dolasci[ 1 ]
  FROM evidencija_nastave;


SELECT student, dolasci[ 1:2 ]
  FROM evidencija_nastave;


UPDATE evidencija_nastave
  SET dolasci[ 2:3 ] = '{ TRUE, TRUE }'
  WHERE student = 'Ivic';
```



Primjer – funkcije za rad s poljima

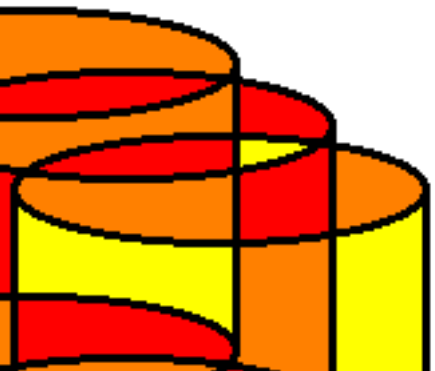
```
SELECT array_dims( dolasci )
  FROM evidencija_nastave;
```



Primjer – funkcije za rad s poljima

```
SELECT array_dims( dolasci )
  FROM evidencija_nastave;
```

```
SELECT ARRAY[ 1, 2 ] || ARRAY[ 3, 4 ]
    AS spojeno;
```

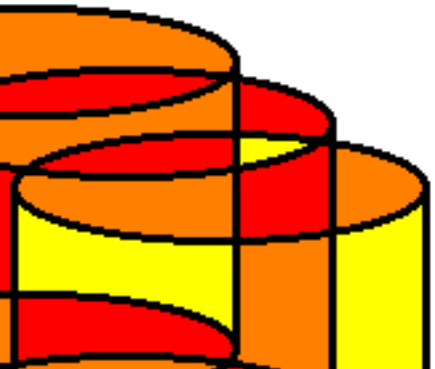


Primjer – funkcije za rad s poljima

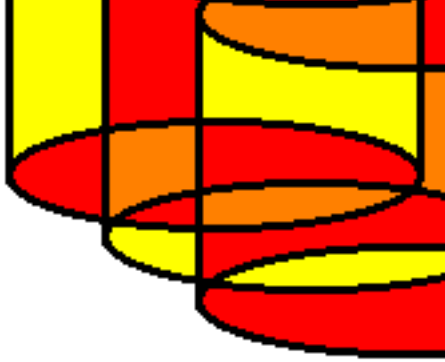
```
SELECT array_dims( dolasci )
  FROM evidencija_nastave;
```

```
SELECT ARRAY[ 1, 2 ] || ARRAY[ 3, 4 ]
    AS spojeno;
```

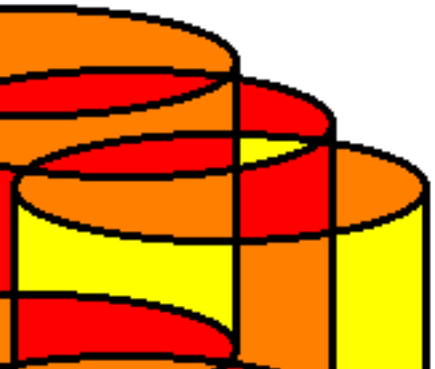
```
SELECT array_prepend( 1, '{ 2, 3 }' );
```



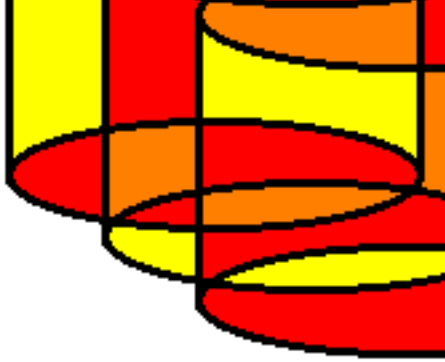
Primjer – funkcije za rad s poljima



```
SELECT array_append( '{ 1, 2, 3 }', 4 );
```

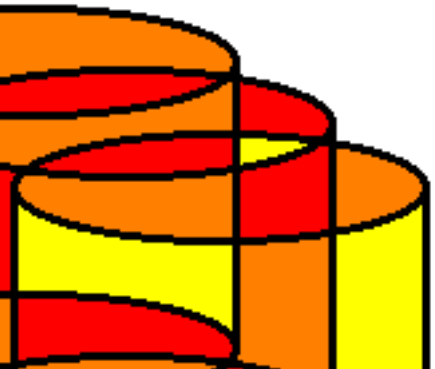


Primjer – funkcije za rad s poljima



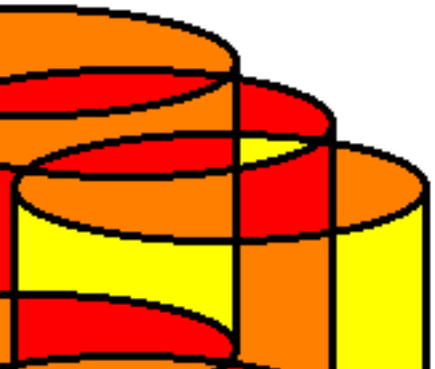
```
SELECT array_append( '{ 1, 2, 3 }', 4 );
```

```
SELECT array_cat(
  ARRAY[ [ 1, 2 ], [ 3, 4 ] ],
  ARRAY[ 5, 6 ] );
```



Primjer – ANY & ALL

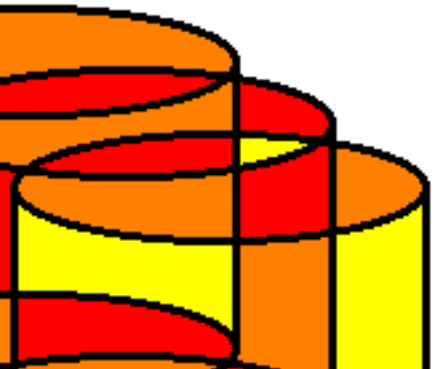
```
SELECT * FROM evidencija_nastave
  WHERE FALSE = ANY( dolasci );
```



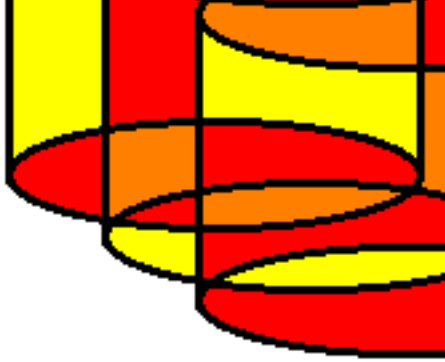
Primjer – ANY & ALL

```
SELECT * FROM evidencija_nastave
  WHERE FALSE = ANY( dolasci );
```

```
SELECT * FROM evidencija_nastave
  WHERE TRUE = ALL( dolasci );
```

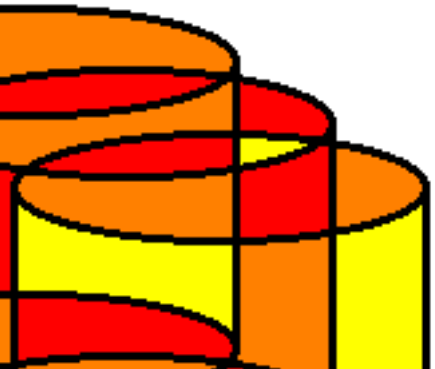


Pobrojene vrijednosti



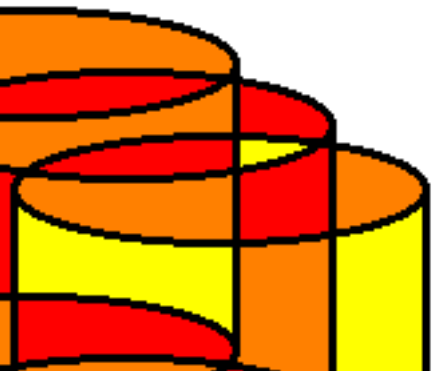
- Sintaksa

```
CREATE TYPE naziv AS ENUM
  ( 'vrijednost1', ... );
```



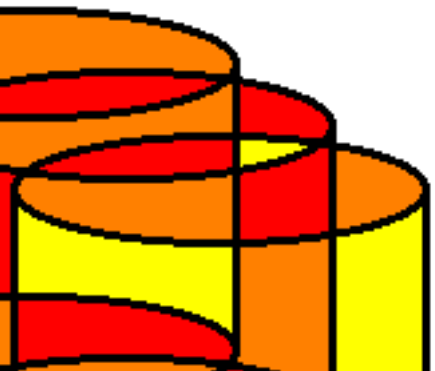
Primjer

```
CREATE TYPE status AS ENUM (
  'na vezi',
  'nije na vezi',
  'nepoznat'
  );
```



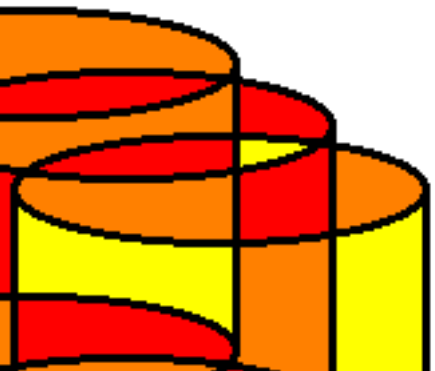
Primjer

```
CREATE TEMP TABLE korisnik(
    kor_ime VARCHAR( 20 ),
    stanje STATUS DEFAULT 'nepoznat'
);
```



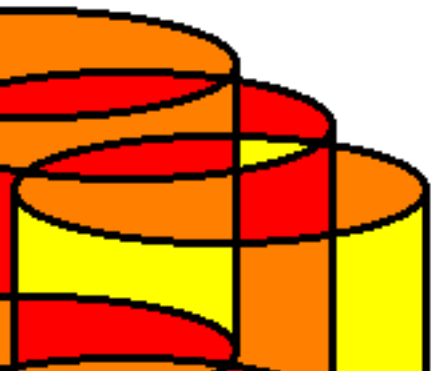
Primjer

```
INSERT INTO korisnik
   VALUES ( 'ivo', 'na vezi' );
```



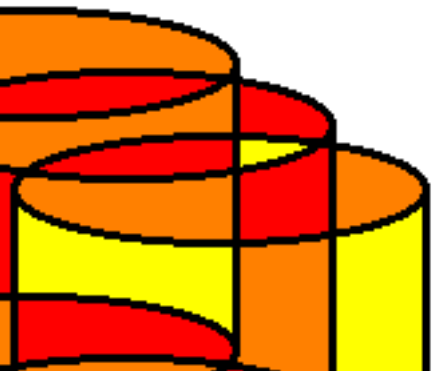
Primjer

```
INSERT INTO korisnik
   VALUES ( 'ivo', 'na vezi' );
INSERT INTO korisnik
   VALUES ( 'ana', 'nije na vezi' );
```



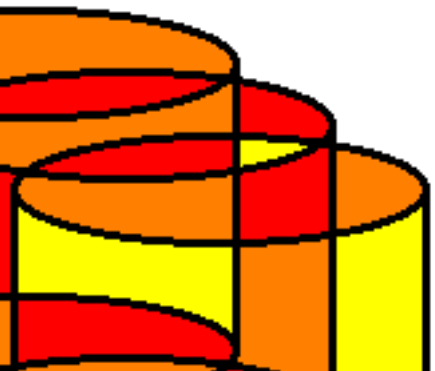
Primjer

```
INSERT INTO korisnik
   VALUES ( 'ivo', 'na vezi' );
INSERT INTO korisnik
   VALUES ( 'ana', 'nije na vezi' );
INSERT INTO korisnik
   VALUES ( 'joza' );
```



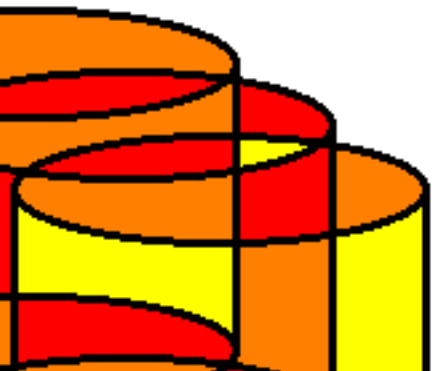
Primjer

```
INSERT INTO korisnik
   VALUES ( 'ivo', 'na vezi' );
INSERT INTO korisnik
   VALUES ( 'ana', 'nije na vezi' );
INSERT INTO korisnik
   VALUES ( 'joza' );
INSERT INTO korisnik
   VALUES ( 'barica', 'krivi status' );
```



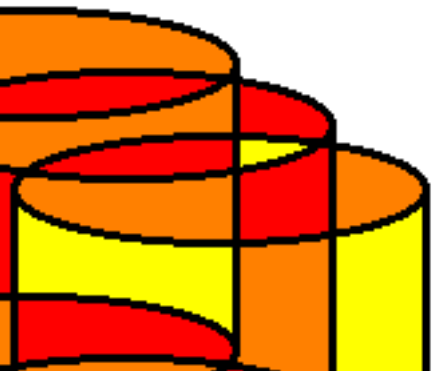
Primjer

```
SELECT * FROM korisnik
  WHERE stanje > 'na vezi';
```

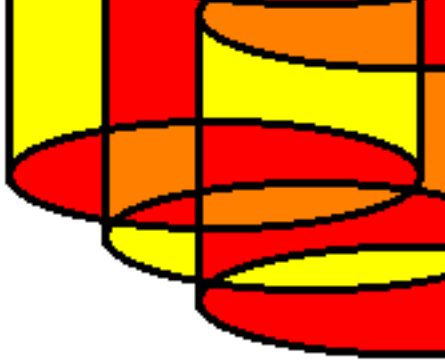


Primjer

```
SELECT * FROM korisnik
  WHERE stanje > 'na vezi';

SELECT * FROM korisnik
  ORDER BY stanje;
```

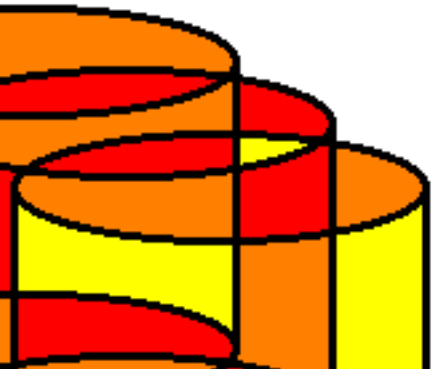


Složeni tipovi



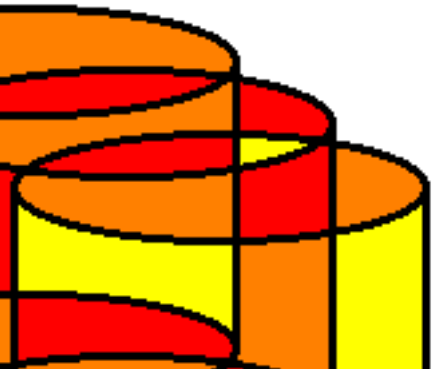
- Sintaksa

```
CREATE TYPE naziv_tipa AS
( specifikacija kolone, ... );
```



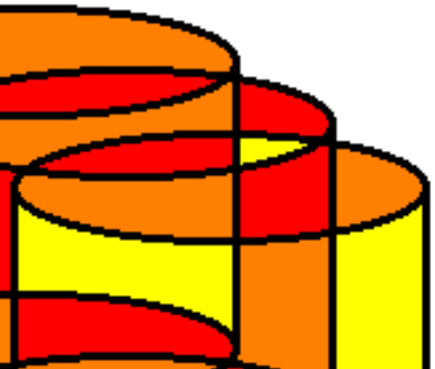
Primjer

```
CREATE TYPE tip_adresa AS (
   ulica TEXT,
   broj INTEGER,
   post_broj INTEGER,
   grad TEXT,
   drzava TEXT
);
```



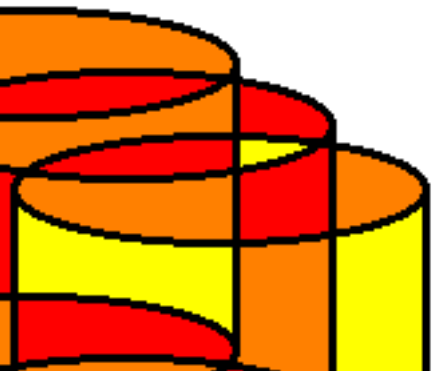
Primjer

```
CREATE TEMP TABLE kupac(
    sifra SERIAL PRIMARY KEY,
    ime TEXT,
    prezime TEXT,
    adresa tip_adresa
);
```



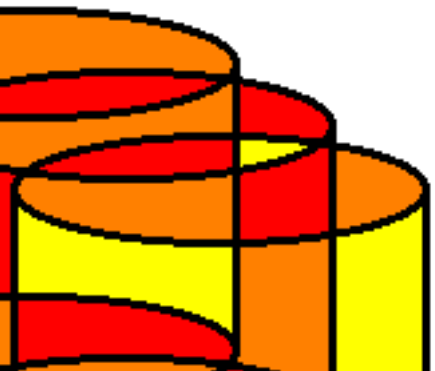
Primjer

```
INSERT INTO kupac VALUES( DEFAULT,
  'Ivek', 'Presvetli',
  ROW( 'Jalkovečka', 23, 42000,
  'Varaždin', 'Hrvatska' ) );
```



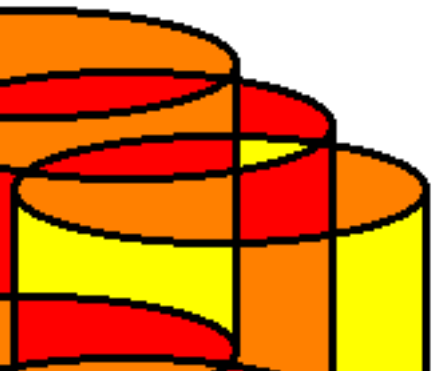
Primjer

```
INSERT INTO kupac VALUES( DEFAULT,
  'Ivek', 'Presvetli',
  ROW( 'Jalkovečka', 23, 42000,
  'Varaždin', 'Hrvatska' ) );

INSERT INTO kupac VALUES( DEFAULT,
  'Laszlo', 'Guylasz', ROW( 'Cinege
  utca', 3, 1000, 'Budapest',
  'Madarska' ) );
```

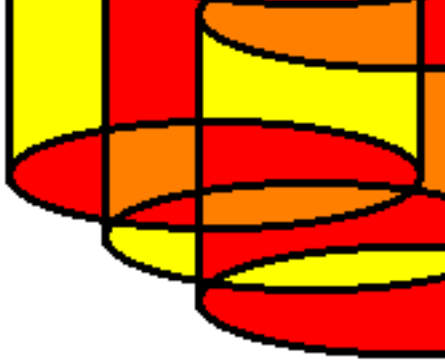


Primjer

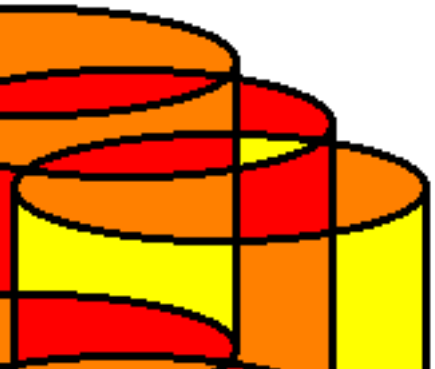
```
SELECT * FROM kupac WHERE
  (adresa).grad = 'Varaždin';
```



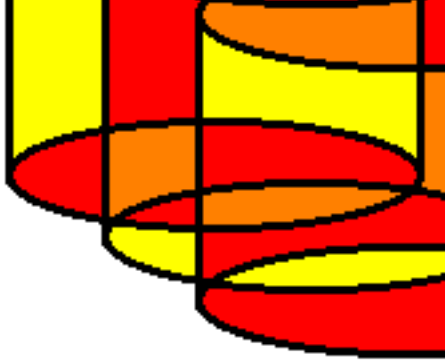
Objektno-relacijske baze podataka



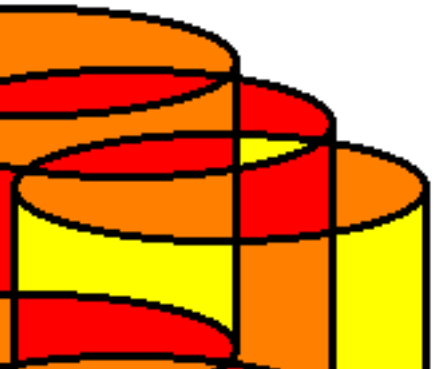
- PostgreSQL omogućava i neke koncepte iz objektno-orijentiranog pristupa



Naslijeđivanje

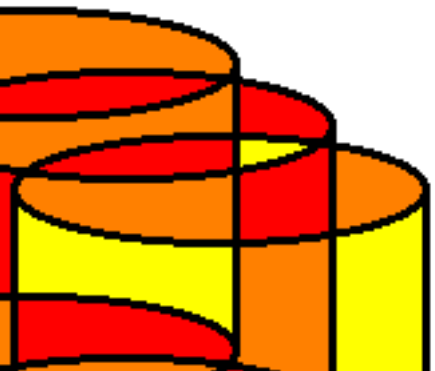


- Jedna relacija naslijeđuje sve atribute druge relacije
- Takva relacija je posebna podrelacija (podklasa) nadređene relacije (nadklasa)



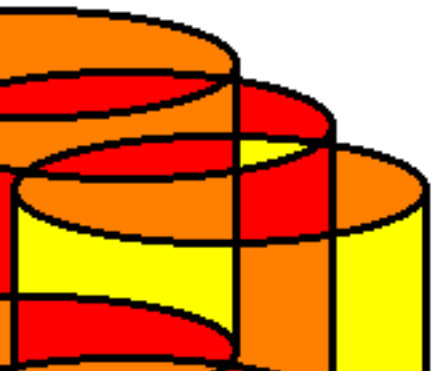
Primjer

```
CREATE TABLE admin(
    tel_br VARCHAR(10)
) INHERITS (osoba);
```



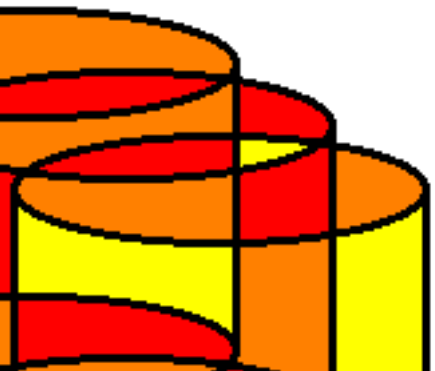
Primjer

```
CREATE TABLE admin(
    tel_br VARCHAR(10)
) INHERITS (osoba);

SELECT * FROM admin;
```



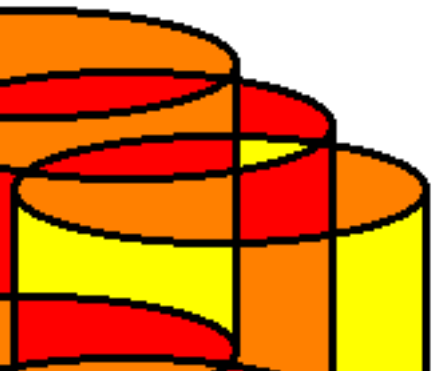
Primjer

```
INSERT INTO admin( email, ime, prezime,
  tel_br ) VALUES ( 'berny@foi.hr',
  'Bernardo', 'Golenja', '0987654321' );
```



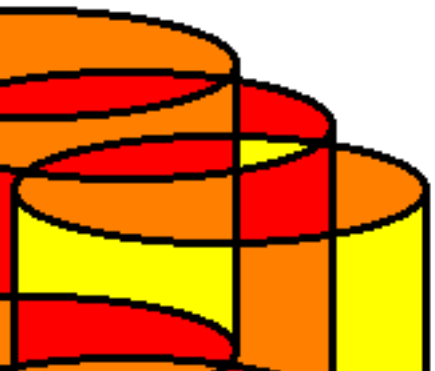
Primjer

```
INSERT INTO admin( email, ime, prezime,
  tel_br ) VALUES ( 'berny@foi.hr',
  'Bernardo', 'Golenja', '0987654321' );


SELECT * FROM admin;
```



Primjer

```
INSERT INTO admin( email, ime, prezime,
  tel_br ) VALUES ( 'berny@foi.hr',
  'Bernardo', 'Golenja', '0987654321' );


SELECT * FROM admin;

SELECT * FROM osoba;
```



Primjer

```
INSERT INTO admin( email, ime, prezime,
  tel_br ) VALUES ( 'berny@foi.hr',
  'Bernardo', 'Golenja', '0987654321' );
```

```
SELECT * FROM admin;
```

```
SELECT * FROM osoba;
```

```
SELECT * FROM ONLY osoba;
```