

Adatbázisok - 5. előadás

Horváth Árpád <horvath.arpad@amk.uni-obuda.hu>

Óbudai Egyetem
Alba Regia Műszaki Kar (AMK)
Székesfehérvár

2015. október 15.

Vázlat

- 1 Database cluster
- 2 Felhasználó létrehozása
- 3 Adatbázis létrehozása
- 4 psql
- 5 SQL

Database cluster

- Database cluster: adatbázis tároló terület, egy könyvtár a merevlemezen.
- Inicializálás pl.: `initdb -D d:\kl\pg_data`
- Az inicializálást követően a database cluster tartalmaz számos állományt, így egy postgres elnevezésű és egy template1 elnevezésű adatbázist is. A postgres adatbázis az alapértelmezett adatbázis, a template1 egy adatbázis sablon. A postgresben minden létrehozott adatbázis egy sablon alapján jön létre.

Vázlat

- 1 Database cluster
- 2 Felhasználó létrehozása
- 3 Adatbázis létrehozása
- 4 psql
- 5 SQL

Felhasználó létrehozása (1)

- A PostgreSQL a felhasználók kezelésére a role koncepciót használja. Egy role, egy adatbázis felhasználót vagy a felhasználók egy csoportját jelenti, rendelkezhet adatbázis objektumokkal és jogosultságokat határozhat meg a többi role számára a saját adatbázis objektumainak használatával kapcsolatban.
- Role parancssorból így hozható létre:
`createuser felhasznalo`
- A törléshez a következő parancs használható:
`dropuser felhasznalo`

Felhasználó létrehozása (2)

- SQL nyelven így érhetjük hozhatuk létre és törölhetünk felhasználót:

```
1 CREATE ROLE felhasznalo;
```

```
2
```

```
3 DROP ROLE felhasznalo;
```

- A felhasználók listáját így tekinthetjük meg:

```
1 SELECT rolname FROM pg_roles;
```

- Vagy használhatjuk a psql programban a \du metaparancsot.

Vázlat

- 1 Database cluster
- 2 Felhasználó létrehozása
- 3 Adatbázis létrehozása**
- 4 psql
- 5 SQL

Adatbázis létrehozása

- Egy adatbázis objektumokat tartalmaz és minden objektum egy adatbázishoz tartozik. Ez alól néhány rendszertábla kivétel, pl. a `pg_database` elérhető a teljes clusterben, mindegyik adatbázisból.
- Adatbázis létrehozása és törlése a `createdb` programmal:
`createdb adatbazis`
`dropdb adatbazis`
- Adatbázis létrehozása és törlése SQL nyelven:

1 `CREATE DATABASE` adatbazis;

2
3 `DROP DATABASE` adatbazis;

Adatbázis létrehozása tulajdonos megadásával

- createdb programmal:
createdb -O felhasznalo adatbazis
- SQL nyelven:

```
1 CREATE DATABASE adatbazis OWNER felhasznalo;
```

Adatbázisok listázása

- SQL utasítással: `SELECT datname FROM pg_database ;`
- parancssorból: `psql -l`
- a psql-ben metaparanccsal: `\l`

Vázlat

- 1 Database cluster
- 2 Felhasználó létrehozása
- 3 Adatbázis létrehozása
- 4 psql**
- 5 SQL

- A psql a PostgreSQL interaktív terminál kliense.
- A psql lehetővé teszi:
 - a kapcsolódást PostgreSQL szerverekhez,
 - az adatkezelést SQL nyelven,
 - parancssori opciók és metaparancsok használatát számos feladathoz.

psql - kapcsolódás adatbázishoz

```
psql -h host -p port -d adatbazis -U felhasznalo -W
```

Az opciók jelentése:

- a `-h` után a szerveret adhatjuk meg, localhost, ha a helyi gépen fut a szerver,
- a `-p` a port száma, amelyen a szerver elérhető, ez alapértelmezetten 5432,
- az adatbázis nevét a `-d` opció után adhatjuk meg,
- a felhasználót a `-U` opciót követően írjuk be,
- a `-W` jelentése: a kapcsolódáskor a psql a jelszó beírását kéri, a `-w` opció esetén nem kér jelszót.

Az opciók elhagyhatók, ekkor az alapértelmezett értékekkel kapcsolódik a psql a szerverhez.

psql - prompt, help, kilépés

- Az interaktív terminál promptja `adatbázisnev=#` formátumú, ha superuserként kapcsolódtunk, egyébként `adatbázis=>` formátumban jelenik meg. A prompt után írhatjuk be a psql parancsait vagy az SQL utasításokat.
- Az SQL parancsokat `;`-vel zárjuk, a psql metaparancsait nem.
- Néhány metaparancs:
 - `\?` - psql parancsainak listája
 - `\h` - SQL help
 - `\q` - kilépés

Vázlat

- 1 Database cluster
- 2 Felhasználó létrehozása
- 3 Adatbázis létrehozása
- 4 psql
- 5 SQL

SQL

- DDL - Data Definition Language (CREATE ..., ALTER ..., DROP ...)
- DML - Data Manipulation Language (SELECT, UPDATE, DELETE, INSERT INTO)
- DCL - Data Control Language (GRANT, REVOKE)

CREATE TABLE

```
1 CREATE TABLE hallgatok (  
2     id integer,  
3     nev text  
4 );
```

CREATE TABLE - elsődleges kulccsal

```
1 CREATE TABLE pelda1 (  
2     a integer PRIMARY KEY,  
3     b integer  
4 );  
5  
6  
7 CREATE TABLE pelda2 (  
8     a integer,  
9     b integer  
10    PRIMARY KEY (a, b)  
11 );
```

CREATE TABLE - idegen kulccsal

```
1 CREATE TABLE termekek (  
2     termék_kod integer PRIMARY KEY,  
3     nev text NOT NULL,  
4     ar numeric DEFAULT 1000  
5 );  
6  
7 CREATE TABLE megrendelesek (  
8     megrendeles_kod integer PRIMARY KEY,  
9     mennyiség integer NOT NULL,  
10    termék_id integer REFERENCES termekek (termék_kod)  
11 );
```

Constraints - kényszerek

- NOT NULL
- UNIQUE
- PRIMARY KEY
- FOREIGN KEY
- CHECK

CHECK

```
1 CREATE TABLE termekek (  
2     termék_kod integer PRIMARY KEY,  
3     nev text NOT NULL,  
4     ar numeric DEFAULT 1000 CHECK (ar > 0)  
5 );
```

ALTER - oszlop hozzáadása

```
1 ALTER TABLE termekek ADD COLUMN leiras text;
```

ALTER - oszlop törlése

```
1 ALTER TABLE termekek DROP COLUMN leiras;
```

ALTER - constraint hozzáadása

```
1 ALTER TABLE termekek ADD CHECK (nev <> '');  
2  
3 ALTER TABLE termekek ADD CONSTRAINT egyedi_termekszam  
4 UNIQUE (termekszam);  
5  
6 ALTER TABLE megrendelesek ADD FOREIGN KEY (termek_id)  
7 REFERENCES termekek;  
8  
9 ALTER TABLE termekek ALTER COLUMN nev SET NOT NULL;
```


ALTER - constraint törlése

- kényszer nevének megtekintése: \d tabla

```
1 ALTER TABLE tabla DROP CONSTRAINT constraint_nev;
```

ALTER - oszlop átnevezése

- 1 `ALTER TABLE termekek RENAME COLUMN termekszam`
- 2 `TO termek_kod;`

ALTER - tábla átnevezése

```
1 ALTER TABLE diakok RENAME TO hallgatok;
```

Tábla törlése

```
1 DROP TABLE pelda1;
```

INSERT INTO (1)

- hallgatok(id, nev)

```
1 INSERT INTO hallgatok VALUES (1, 'Kiss Alex');  
2 INSERT INTO hallgatok VALUES (2, 'Nagy Anna');
```

INSERT INTO (2)

```
1 CREATE TABLE tanarok (  
2     id serial,  
3     nev text  
4 );  
5  
6 INSERT INTO tanarok (nev) VALUES ('Nagy Alex');  
7 INSERT INTO tanarok (nev) VALUES ('Kiss Anna');
```

UPDATE

```
1 UPDATE termek SET ar = 2000 WHERE ar = 1000;
```

DELETE

- 1 `DELETE FROM termekek WHERE price = 2000;`
- 2 `DELETE FROM hallgatok;`