

DBI

RPostgreSQL

Grzegorz Olejarz

Wydział Geodezji Górniczej i Inżynierii Środowiska
Akademia Górniczo-Hutnicza
2010

Biblioteki DBI i RPostgreSQL

- ▶ Wersja: 0.2–5,
 - ▶ Data: 2009–12–22,
 - ▶ Interfejs baz danych,
 - ▶ Umożliwia komunikację pomiędzy RDBMS a R,
 - ▶ Klasy DBI są abstrakcyjne,
- ▶ Wersja: 0.1–6,
 - ▶ Data: 2009–10–19,
 - ▶ Implementacja DBI dla bazy danych PostgreSQL

DBI

RPostgreSQL

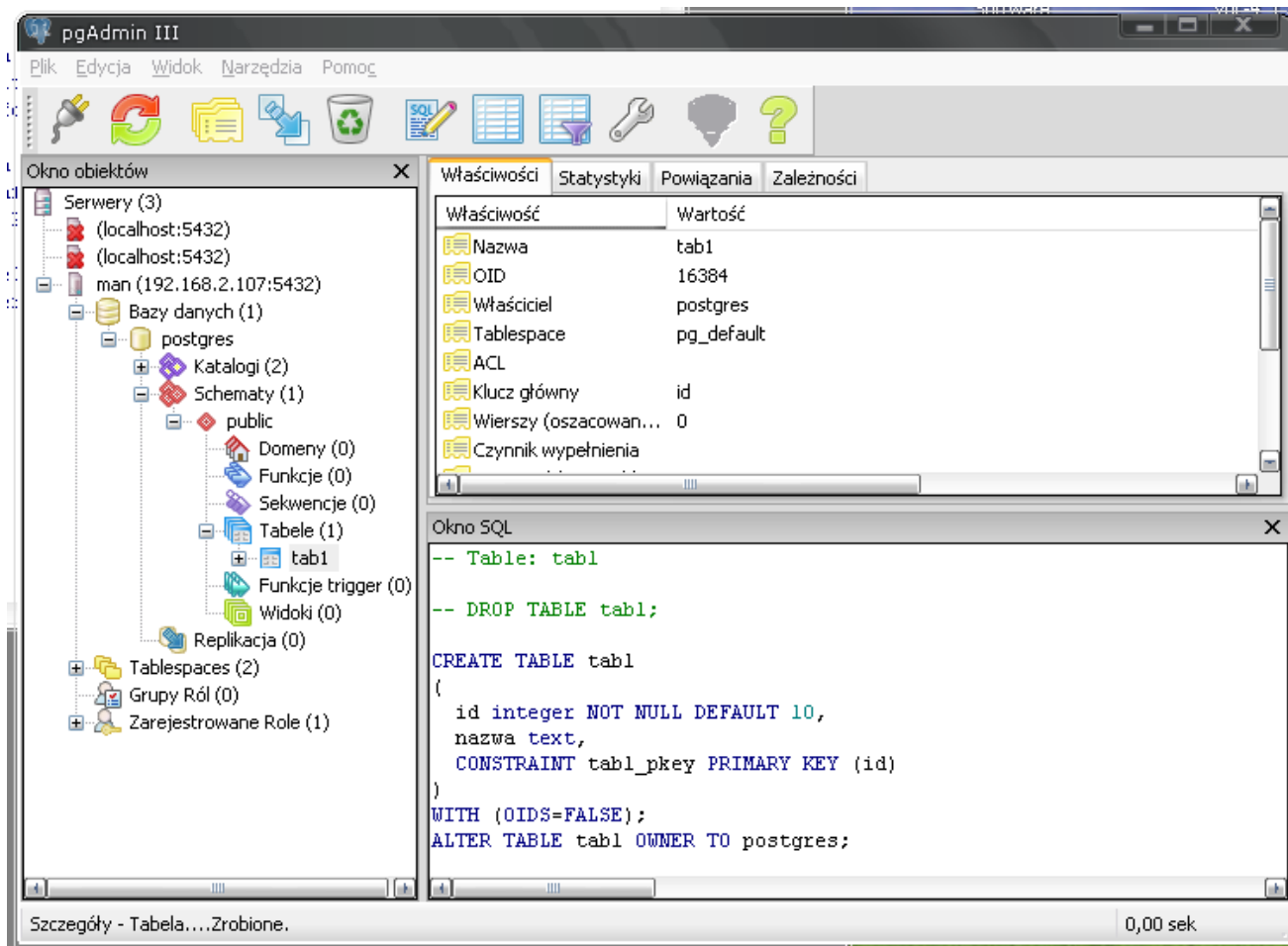
Podstawowe metody DBI

- ▶ dbConnect – ustawia połączenie z bazą,
- ▶ dbDisconnect – kończy aktywne połączenie,
- ▶ dbSendQuery – wysyła zapytanie (kwerendę) do bazy; zwraca zasób zapytania,
- ▶ fetch – przekształca zasób zapytania na data.frame,
- ▶ dbGetQuery – wykonuje zapytanie i zwraca data.frame (działa jak dbSendQuery oraz fetchjednocześnie),
- ▶ dbCommit – zatwierdza transakcję,
- ▶ dbRollback – cofa transakcje,
- ▶ dbGetInfo – meta-dane dla obiektu DBIObject,
- ▶ dbListTables – tworzy listę tabel aktualnej bazy,
- ▶ dbListFields – tworzy listę kolumn tabeli,
- ▶ dbListConnections – tworzy listę aktywnych połączeń,
- ▶ dbListResults – tworzy listę aktualnych zapytań.

Baza danych PostgreSQL

- ▶ PostgreSQL to najbardziej zaawansowany system relacyjnych baz danych Open Source.
- ▶ Model relacyjno–obiektowy,
- ▶ Otwarte źródła,
- ▶ Darmowa,
- ▶ Ponad 15 lat aktywnego rozwoju,
- ▶ Wieloplatformowa.

Klient pgAdmin III



The screenshot displays the pgAdmin III application window. The left pane shows a tree view of the database structure, including servers, databases, catalogs, schemas, and tables. The right pane is divided into two sections: 'Właściwości' (Properties) and 'Okno SQL' (SQL Window).

The 'Właściwości' section shows the following table properties:

Właściwość	Wartość
Nazwa	tab1
OID	16384
Właściciel	postgres
Tablespace	pg_default
ACL	
Klucz główny	id
Wierszy (oszacowan...)	0
Czynnik wypełnienia	

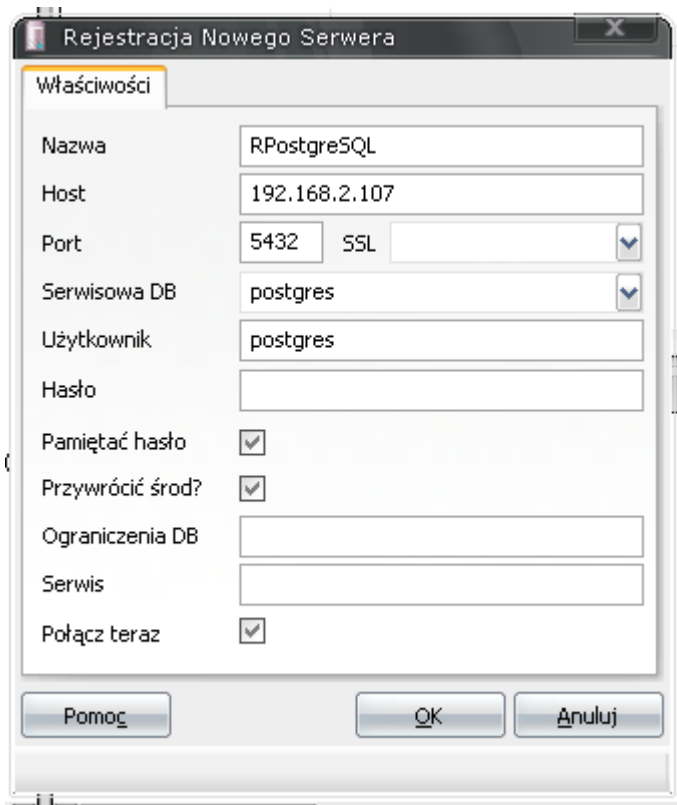
The 'Okno SQL' section contains the following SQL code:

```
-- Table: tab1
-- DROP TABLE tab1;

CREATE TABLE tab1
(
  id integer NOT NULL DEFAULT 10,
  nazwa text,
  CONSTRAINT tab1_pkey PRIMARY KEY (id)
)
WITH (OIDS=FALSE);
ALTER TABLE tab1 OWNER TO postgres;
```

The status bar at the bottom indicates 'Szczegóły - Tabela....Zrobione.' and '0,00 sek'.

Połączenie z bazą przy pomocy pgAdmin



Rys. Parametry połączenia

Edit Data - man (192.168.2.107:5432) - postgres - tab1

Plik Edycja Widok Pomoc

Bez limitu

	id [PK] integer	nazwa text
1	1	cos
2	2	tekst
3	3	test
*		

Rys. Przykładowa tabela testowa

R, DBI oraz RPostgreSQL CRUD – Select

```
> local({pkg <- select.list(sort(.packages(all.available = TRUE)))
+ if(nchar(pkg)) library(pkg, character.only=TRUE)})
Loading required package: DBI
> drv = dbDriver("PostgreSQL")
> con = dbConnect(drv, user="postgres", password="postgres", dbname="postgres", host="192.168.2.107")
> res = dbSendQuery(con, "SELECT * FROM tab1")
> data = fetch(res)
> data
  id nazwa
1  1   cos
2  2 tekst
3  3  test
> |
```

CRUD – Insert

```
> res = dbSendQuery(con, "INSERT INTO tab1(id, nazwa) VALUES (4, 'R') ")
> res = dbSendQuery(con, "SELECT * FROM tab1")
> data = fetch(res)
> data
  id nazwa
1  1   cos
2  2  tekst
3  3   test
4  4     R
```


CRUD – Delete

```
> res = dbSendQuery(con, "DELETE FROM tab1 WHERE id=4")
> res = dbSendQuery(con, "SELECT * FROM tab1")
> data = fetch(res)
> data
  id nazwa
1  1   cos
2  2 tekst
3  3  test
```

CRUD – Update

```
> res = dbSendQuery(con, "UPDATE tab1 SET nazwa='update from R' WHERE id=3")
> res = dbSendQuery(con, "SELECT * FROM tab1")
> data = fetch(res)
> data
```

	id	nazwa
1	1	cos
2	2	tekst
3	3	update from R

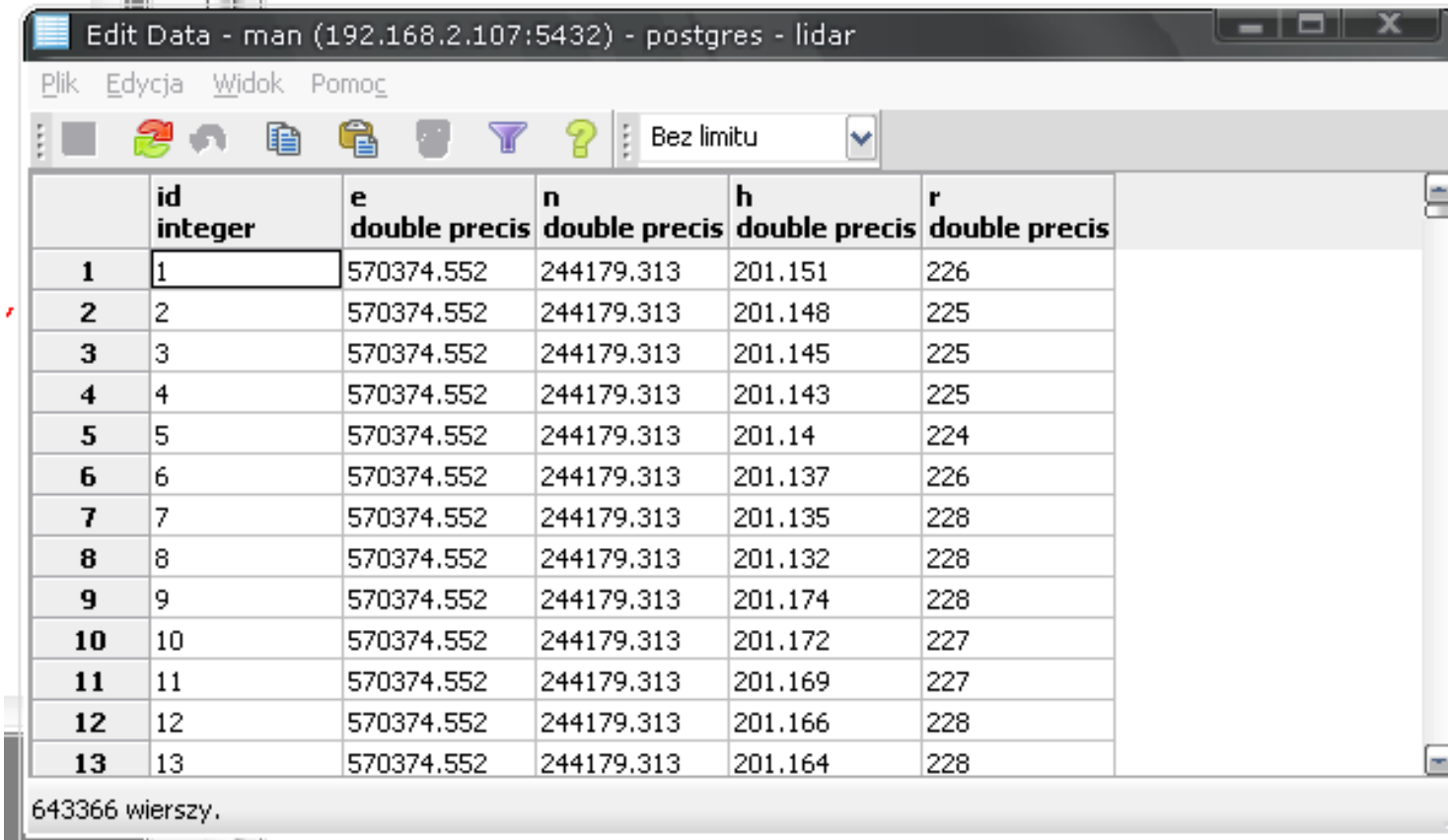
Próba załadowania do bazy dużej ilości danych

```
lid_df = matrix(scan("lidar.asc"), ncol=4, byrow=T)
Read 2573464 items
lid = data.frame(lid_df)

res = dbSendQuery(con, "CREATE TABLE lidar (id integer,E
  double precision,N double precision,H double precision,R
  double precision)")

names(lid) = c("e","n","h","r")
cc=1
for(i in 0:length(lid$e)){
  st=c("INSERT INTO lidar(id,e,n,h,r)
    VALUES (",cc, ", ", lid$e[i], ", " ,lid$n[i], ", "
    ,lid$h[i], ", " ,lid$r[i], ")") ;
  dbSendQuery(con,paste(st,collapse=""));
  cc=cc+1
}
```


Dla pewności sprawdzimy w bazie ;-)



Edit Data - man (192.168.2.107:5432) - postgres - lidar

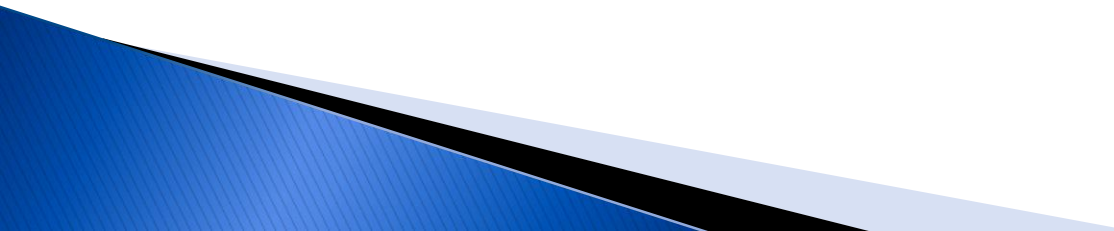
Plik Edycja Widok Pomoc

Bez limitu

	id integer	e double precis	n double precis	h double precis	r double precis
1	1	570374.552	244179.313	201.151	226
2	2	570374.552	244179.313	201.148	225
3	3	570374.552	244179.313	201.145	225
4	4	570374.552	244179.313	201.143	225
5	5	570374.552	244179.313	201.14	224
6	6	570374.552	244179.313	201.137	226
7	7	570374.552	244179.313	201.135	228
8	8	570374.552	244179.313	201.132	228
9	9	570374.552	244179.313	201.174	228
10	10	570374.552	244179.313	201.172	227
11	11	570374.552	244179.313	201.169	227
12	12	570374.552	244179.313	201.166	228
13	13	570374.552	244179.313	201.164	228

643366 wierszy.

Podsumowanie i wnioski

- ▶ Szybki dostęp do dużej ilości danych,
 - ▶ Lazy Loading – porcjowanie danych (brak konieczności wczytywania całych plików),
 - ▶ Możliwość pracy wielu osób na tych samych danych,
 - ▶ Aktualne dane,
 - ▶ Niezbędna znajomość SQL
- 

Dziękuję!

Grzegorz Olejarz