

Molekularbiologische Datenbanken

Übungen Aufgabe 3



Silke Trißl
Ulf Leser



Wissensmanagement in der
Bioinformatik

Dataload – CSV files

- Plattform
 - Array-Submitter.csv
 - Angaben über Array Submitter
 - Array.csv
 - Alle bis jetzt genutzten Arrays, wobei die ID = GPLxx ist.
 - GPLxx.csv
 - Jede Datei steht für einen Array. In diesen Dateien ist angegeben, welche
 - Proben auf dem Array sind und noch die zusätzlichen Daten über
 - GenBank-Accession und GeneOntology.

Dataload – CSV files – 2 –

- Samples
 - Organisation.csv
 - Daten über die beteiligten Labors
 - Submitter.csv
 - Daten über den Submitter von Samples und Meßergebnissen
 - Sample.csv
 - Daten über die Samples, wobei ID = GSMxxxx.csv ist.

Dataload – CSV files – 3 –

- Samples
 - Sample-Protokoll.csv
 - Welche Probe ist welchem Protokoll behandelt worden
 - GSMxxxx.csv
 - Jede Datei steht für ein Sample. In diesen Dateien sind die ID der Probe
 - und die Meßwerte für jede Probe auf dem Array angegeben.

Dataload – CSV files – 4 –

- Protokolle
 - Protokolle.csv
 - Daten über Protokoll-ID und die dazugehörigen Angaben.
- Aufgabe:
 - Laden der Daten (verfügbar über web-page)
 - SQL-Queries

PostgreSQL 7.3

- Download von postgresql-7.3.2.tar.gz über <http://www.postgresql.org/>
- Installation unter Linux (schwierig bei Windows)
- Mit root-rechten

```
./configure --with-perl (--with-java, --with-python)
```

PostgreSQL 7.3 – 2 –

• Ohne Root-rechte: lokale Installation

```
./configure --prefix=/home/name/.../pgsql
gmake
gmake install
mkdir /home/name/.../pgsql/data
chmod g+rwx /home/name/.../pgsql/data
/home/name/.../pgsql/bin/initdb -D /home/name/.../pgsql/data
/home/name/.../pgsql/bin/postmaster -D /home/name/.../pgsql/data
>logfile 2>&1 &

/home/name/.../pgsql/bin/createdb test - erstellt Datenbank
/home/name/.../pgsql/bin/psql test - shell, um mit Datenbank zu
arbeiten
```

SQL

```
CREATE TABLE test
(t_id int NOT NULL,
 some_text varchar,
 foreign_id int NOT NULL,
 CONSTRAINT t_id PRIMARY KEY (subid));
```

```
ALTER TABLE test
ADD FOREIGN KEY (foreign_id)
REFERENCES foreign (id);
```

SQL – 2 –

```
SELECT count(*) FROM array  
WHERE array_id LIKE '%34%';
```

```
INSERT INTO protokoll (prot_id, des)  
VALUES ('EMBL12', 'some text');
```

```
DELETE FROM submitter  
WHERE name LIKE 'Smith';
```

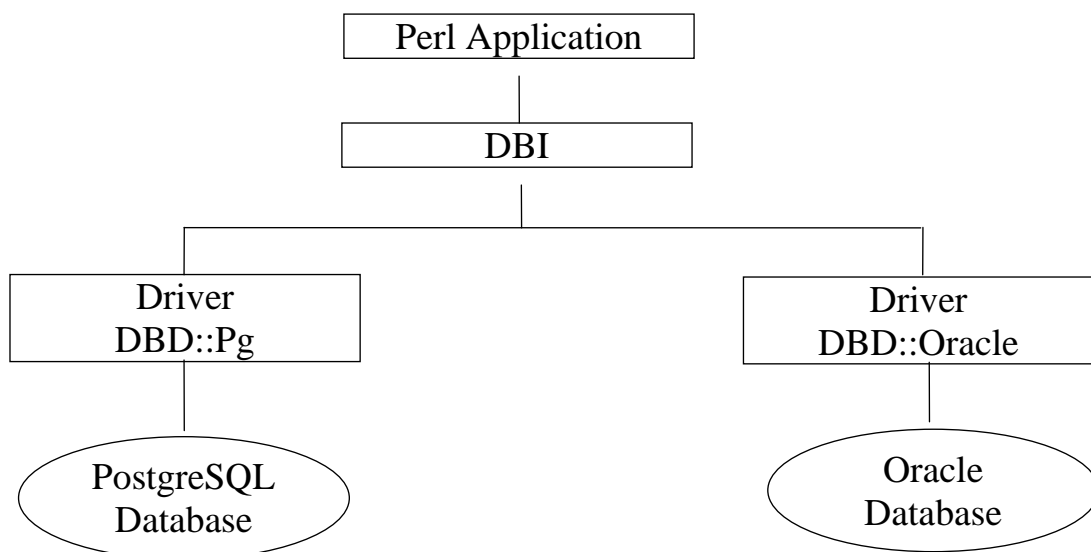
Meta-commands

- \h help
- \dT list data types
- \i <file> read and execute queries from <file>
- \d{t|i|s|v} list tables/indices/sequences/views
- \d <table> describe table (or index, ...)
- \q quit psql

Eigenheit von PostgreSQL

- Bei Verwendung von Anführungszeichen ("):
 - CREATE TABLE "SOME" ("S_Id" int, ...
 - SELECT "SOME". "S_Id", ... FROM "SOME", ...
- Besser:
 - CREATE TABLE SOME (S_Id int, ...
 - SELECT SOME.S_Id, ... FROM SOME, ... oder
 - SELECT some.s_id, ... FROM some, ...

Perl



```

#!/usr/bin/perl -w
#
# Perl script to
# Connect to PostgreSQL DB
# with INSERT INTO and SELECT
#

use strict;
use DBI;

my $sqlstat;

my $dbh = DBI->connect("dbi:PgPP:microarray",'','')
    or die "Could not connect to database: ".DBI->errstr;

# insert into database
$sqlstat = "INSERT INTO protokoll VALUES ('EMBST-23', 'Some Text', 'Image')";
my $insert_handle =
    $dbh->prepare( $sqlstat );
    die "Could not prepare queries; aborting" unless defined $insert_handle;

    $insert_handle->execute();

```

```

# select from database
$sqlstat = "SELECT * FROM array WHERE arsubid LIKE 'Affy'";
my $sth = $dbh->prepare ( $sqlstat )
    or die "Could not prepare statement: " . $dbh->errstr;

    $sth->execute( )
    or die "Could not execute statement: " . $sth->errstr;

print_results ( $sth );

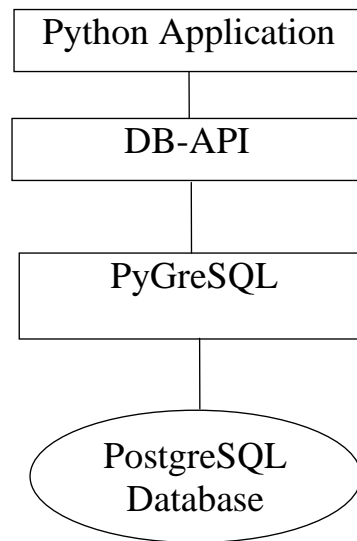
$dbh->disconnect();

sub print_results {
    my $sth = shift;

    while( my $vals = $sth->fetchrow_arrayref() ) {
        foreach my $val ( @$vals ) {
            print $val."\t";
        }
        print "\n";
    }
}

```

Python



```
#!/usr/bin/python
#
#
# Python script to
# Connect to PostgreSQL DB
# with INSERT INTO
#

import _pg
import string

def main( ):
    try:
        connection = _pg.connect( 'microarray' )
    except Exception, e:
        print str( e )
        exit

    try:
        sqlstat = "INSERT INTO protokoll VALUES ('EMBST-28', 'Some Text', 'Image')"
        connection.query( sqlstat )
        print "INSERT"
    except StandardError, e:
        print str( e )

    connection.close()
    exit

main()
```

```

#!/usr/bin/python
#
# Python script to
# Connect to PostgreSQL DB
# with SELECT
#
import pgdb
import string

def main( ):

    try:
        connection = pgdb.connect( database = 'microarray' )
    except Exception, e:
        print str( e )
        exit

    cur = connection.cursor()
    try:
        sqlstat = "SELECT * FROM array WHERE arsubid LIKE 'Affy'"
        cur.execute( sqlstat )
        process_results( cur )
    except StandardError, e:
        print str( e )

    cur.close()
    connection.close()

    exit

```

```

def process_results( cur ):

    rows = cur.fetchall()

    print_values ( cur, rows )

def print_values( cur, rows ):

    r = 0

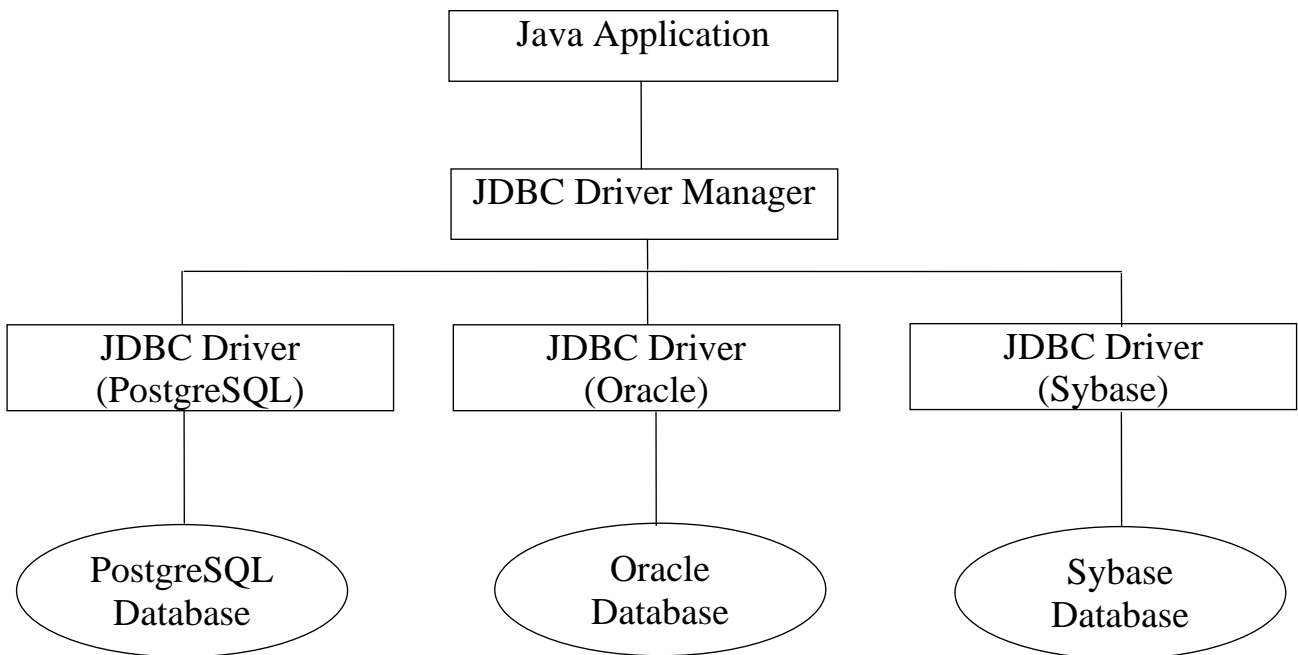
    for row in rows:
        c = 0

        for col in row:
            print str(col) + "\t",
            c = c + 1
        r = r + 1
        print

main()

```

Java



```
// Java code to
// Connect to PostgreSQL DB
// with INSERT INTO and SELECT
//

import java.sql.*;

class PostgreSQLTest {
    public static void main (String[] args) {
        try {
            Driver driver = (Driver)
                Class.forName("postgresql.Driver").newInstance();
            DriverManager.registerDriver(driver);

            String url = "jdbc:postgresql:microarray";
            Connection con = DriverManager.getConnection(url, "postgres", "");
            Statement stm = con.createStatement();

            stm.setQueryTimeout(10);
            ResultSet rs = stm.executeQuery("SELECT * FROM array WHERE arsubid LIKE 'Affy'");

            rs.next();

            System.out.println(rs.getString(1));

        } catch (SQLException e) {

            System.out.println("Exception!");
            System.out.println(e.toString());

        }
    }
}
```

Literatur

- PostgreSQL-Homepage:
 - <http://www.postgresql.org/docs/>
 - Administrators Guide
 - Users Guide
 - Programmers Guide
- Bücher:
 - Douglas & Douglas: PostgreSQL, Developers Library 2003

Fragen?

- Aufgabe 3 bei Goya oder auf der Web-page
- Daten auf der Web-page
- Lösung der SQL-Queries bis 04.06., 17 Uhr

