Povezava do podatkovne baze

- modul pyodbc
- povezovalni niz (eksplicitno, ali z uporabo DSN)

In [6]:
```
import pyodbc
from __future__ import print_function   # Kompatibilnost s Pythonom 2.7 in 3.x

# Eksplicitna prijava brez DSN
ConnectionStringEX = 'Driver={MySQL ODBC 5.3 UNICODE Driver}; \
                      Server=pb.fri.uni-lj.si;Database=tup; \
                      User=tup;Password=tupvaje'
cnxnEX = pyodbc.connect(ConnectionStringEX)

# Privzete vrednosti DSN
ConnectionStringPG = 'DSN=Vaje-PG'
cnxnPG = pyodbc.connect(ConnectionStringPG)
# cnxnPG.cursor().execute("SET SCHEMA 'tup'")  # Lahko izvedemo v definiciji D
SN pod "connect settings"

ConnectionStringMA = 'DSN=Vaje'
cnxnMA = pyodbc.connect(ConnectionStringMA)
```

Kurzor

- Osnovni element interakcije s PB.
- Naredimo ga na osnovi konkretne povezave.
- Istočasno imamo lahko več kurzorjev.

In [7]:
```
cursor = cnxnMA.cursor()
```

In [8]:
```
rez = cursor.execute("SELECT * FROM jadralec")
```

In [9]:
```
# Glava in vsebina (prvi poskus)
for g in rez.description:
    print(g[0],end="\t")
print()
for r in rez:
    for a in r:
        print(a,end="\t")
    print()
```

```
jid     ime     rating  starost
22      Darko   7       45.0
29      Borut   1       33.0
31      Lojze   8       55.5
32      Andrej  8       25.5
58      Rajko   10      35.0
64      Henrik  7       35.0
71      Zdravko 10      16.0
74      Henrik  9       35.0
85      Anze    3       25.5
95      Bine    3       63.5
```

In [10]:
```
rez = cursor.execute("SELECT * FROM jadralec")
telo = rez.fetchall()
#telo = rez.fetchone()
#telo = rez.fetchmany(2)
```

In [11]:
```
# Glava
for g in rez.description:
    print(g[0],end="\t")
print()
```

```
jid     ime     rating  starost
```

In [12]:
```
# Vsebina in tipi
for r in telo:
    for a in r:
        print(a,type(a),end="\t")
    print()
```

```
22 <class 'int'>        Darko <class 'str'>     7 <class 'int'> 45.0 <class
'float'>
29 <class 'int'>        Borut <class 'str'>     1 <class 'int'> 33.0 <class
'float'>
31 <class 'int'>        Lojze <class 'str'>     8 <class 'int'> 55.5 <class
'float'>
32 <class 'int'>        Andrej <class 'str'>    8 <class 'int'> 25.5 <class
'float'>
58 <class 'int'>        Rajko <class 'str'>     10 <class 'int'>        35.0
<class 'float'>
64 <class 'int'>        Henrik <class 'str'>    7 <class 'int'> 35.0 <class
'float'>
71 <class 'int'>        Zdravko <class 'str'>   10 <class 'int'>        16.0
<class 'float'>
74 <class 'int'>        Henrik <class 'str'>    9 <class 'int'> 35.0 <class
'float'>
85 <class 'int'>        Anze <class 'str'>      3 <class 'int'> 25.5 <class
'float'>
95 <class 'int'>        Bine <class 'str'>      3 <class 'int'> 63.5 <class
'float'>
```

In [13]:
```
# Glava in vsebina (drugi poskus)
print("Vseh vrstic je", rez.rowcount)
for g in rez.description:
    print(g[0],end="\t")
print("\n"+"-"*31)
# Vsebina
for r in telo:
    for a in r:
        print(a,end="\t")
    print()
```

```
Vseh vrstic je 10
jid     ime     rating  starost
-------------------------------
22      Darko   7       45.0
29      Borut   1       33.0
31      Lojze   8       55.5
32      Andrej  8       25.5
58      Rajko   10      35.0
64      Henrik  7       35.0
71      Zdravko 10      16.0
74      Henrik  9       35.0
85      Anze    3       25.5
95      Bine    3       63.5
```

Življenjska doba vsebine kurzorja

- le ena iteracija!

Metode kurzorja:

- fetchall(): vrne seznam vseh vrstic
- fetchone(): vrne naslednjo neprebrano vrstico (**pozor: to ni seznam!**)
- fetchmany(*n*): vrne naslednjih *n* neprebranih vrstic

Naloga: poišči šifre najkrajših čolnov!

In [14]:
```
naj = cursor.execute("SELECT cid, dolzina FROM coln")

mind = 1000 # Nekaj velikega
mins = -1   # Nekaj neveljavnega

for (s,d) in naj:
    if d < mind:
        (mins, mind) = (s,d)

print (mins, mind)
```

```
101 34
```

Bolje: kombinacija SQL in Pythona ¶

In [16]:
```
naj = cursor.execute("""
        SELECT cid, dolzina
        FROM coln
        WHERE dolzina = (SELECT MIN(dolzina) FROM coln)
      """)


for (s,d) in naj:
   print (s, d)
```

```
101 34
102 34
```

Obravnava izjem

In [ ]:
```
naj = cursor.execute("""
        SELECT cid, dolzina
        FROM coln
        WHERE dolzina = (SELECT MIN(dolzina) FROM coln)
      """)
```

In [17]:
```
try:
    ConnectionStringBLA = 'Bla Bla'
    cnxnPG = pyodbc.connect(ConnectionStringBLA)
except Exception as e:
    print("NAPAKA v povezav!\n",e)

try:
    naj = cursor.execute(naj)
except pyodbc.DatabaseError as e:
    print("NAPAKA v poizvedbi!\n",e)
```

```
NAPAKA v povezav!
 ('IM002', '[IM002] [Microsoft][ODBC Driver Manager] Data source name not fou
nd and no default driver specified (0) (SQLDriverConnect)')
NAPAKA v poizvedbi!
 ('42000', "[42000] [MySQL][ODBC 5.3(w) Driver][mysqld-5.5.5-10.1.18-MariaDB-
1~trusty]You have an error in your SQL syntax; check the manual that correspo
nds to your MariaDB server version for the right syntax to use near 'WHEN dol
zina = (SELECT MIN(dolzina)                                              FR' at
line 1 (1064) (SQLExecDirectW)")
```

Zaključek dela: vedno zaprite povezave (po možnosti z obravnavo napak)!

Poženi dvakrat.

In [19]:
```
try:
    cnxnEX.close()
except Exception as e:
    print("Zapiranje povezave:",e)

try:
    cnxnPG.close()
except Exception as e:
    print("Zapiranje povezave:",e)

try:
    cnxnMA.close()
except Exception as e:
    print("Zapiranje povezave:",e)
```

```
Zapiranje povezave: Attempt to use a closed connection.
Zapiranje povezave: Attempt to use a closed connection.
Zapiranje povezave: Attempt to use a closed connection.
```