Tehnologija upravljanja podatkov

Programski dostop do podatkovnih baz

Univerzitetni študij, 2. in 3. letnik
Magistrski študij, 1. in 2. letnik

Matjaž Kukar, 2015

ODBC - open data base connectivity

- Proceduralni programski vmesnik za dostop do podatkovne baze
- Nastal je leta 1992 v sodelovanju Microsofta s podjetjem Simba Technologies
- Sloni na različnih standardnih Call Level Interface (CLI) specifikacijah iz SQL Access Group, X/Open in ISO/IEC
- Leta 1995 je ODBC 3.0 postal del standarda ISO/IEC 9075-3 -- Information technology -- Database languages -- SQL -- Part 3: Call-Level Interface (SQL/CLI).

Značilnosti ODBC

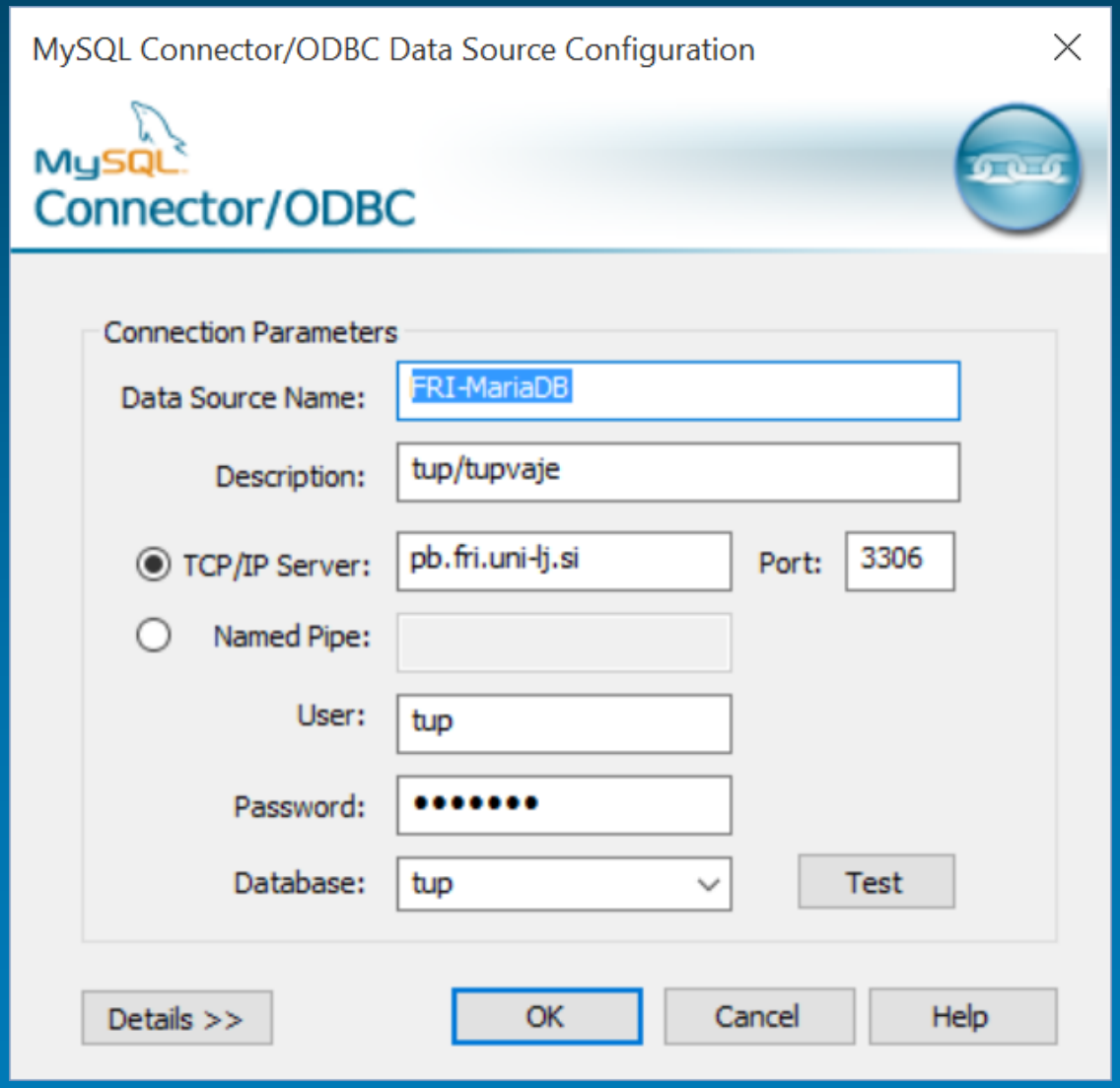
- Omejitev ODBC: delo z SQL standardom, kot ga definira ODBC
- Dostop do specifičnih razširitev SQL: omogočen s pomočjo dodatnih funkcij
- Kaj potrebujemo za delo:
  - ODBC aplikacijski vmesnik za naš OS in izbran programski jezik (npr. pyodbc za Python)
  - ODBC gonilnik za naš OS in uporabljano PB (npr. Connector/ODBC za MariaDB in MySQL)

Predpriprava na uporabo ODBC

- SUPB s podatki
- Aplikacijo, ki zna uporabljati ODBC (npr. Microsoft Excel)
- ODBC gonilnik za izbrani OS (32/64 bit)in SUPB
  - MySQL: Connector/ODBC
  - Oracle: Oracle Instant Client
  - PostgreSQL: psqlodbc
  - ...

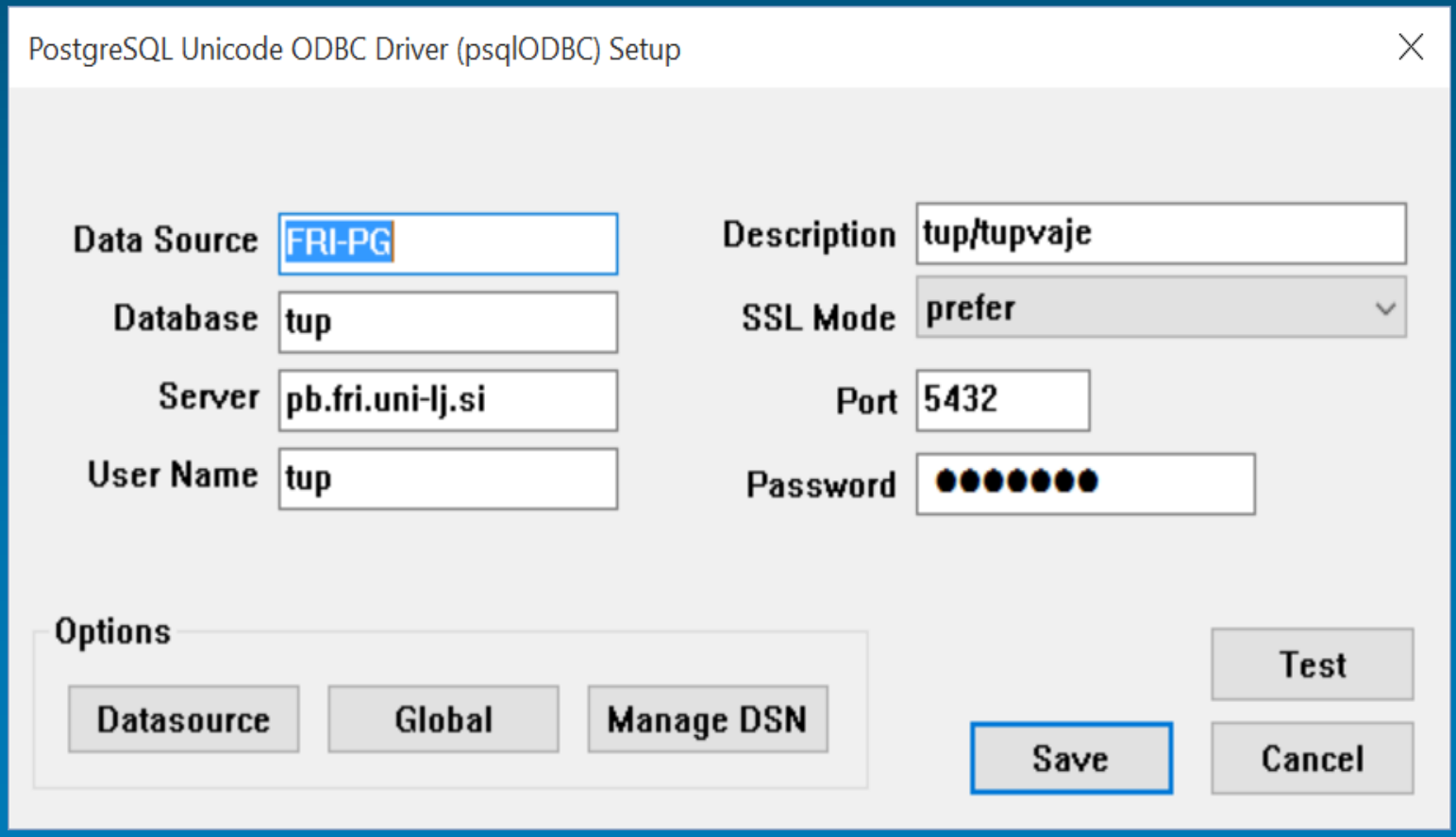
Priprava podatkovnega vira (MariaDB, MySQL)

- Odprite Control Panel->Administrative tools ->Data Sources (ODBC)
- V zavihku User DSN izberite Add in nato določite ODBC gonilnik:
  - MySQL ODBC 5.x driver
  - Vnesite vrednosti s slike: DSN je lahko poljuben.
  - Lahko vnesete uporab. ime in geslo



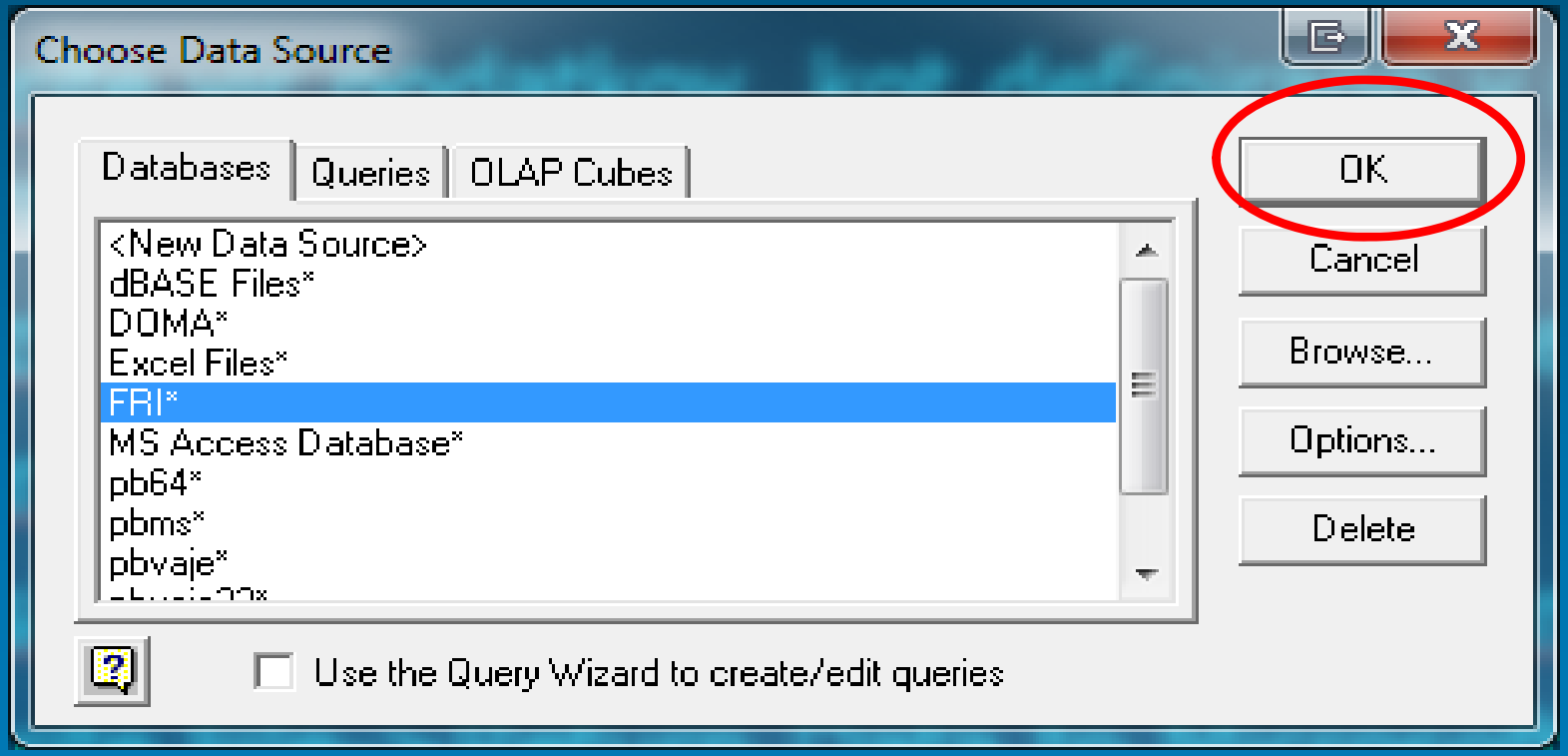
Priprava podatkovnega vira (PostgreSQL)

- Odprite Control Panel->Administrative tools ->Data Sources (ODBC)
- V zavihku User DSN izberite Add in nato določite ODBC gonilnik:
  - PostgreSQL UNICODE (priporočeno) ali
  - PostgreSQL ANSI instantclient
  - Vnesite vrednosti s slike: DSN je lahko poljuben.



Microsoft Excel in ODBC

- Izberite
  Data->From Other Sources->From Microsoft Query
- Izberite vir podatkov, kot definirano v ODBC Data Sources (npr. FRI-MariaDB)

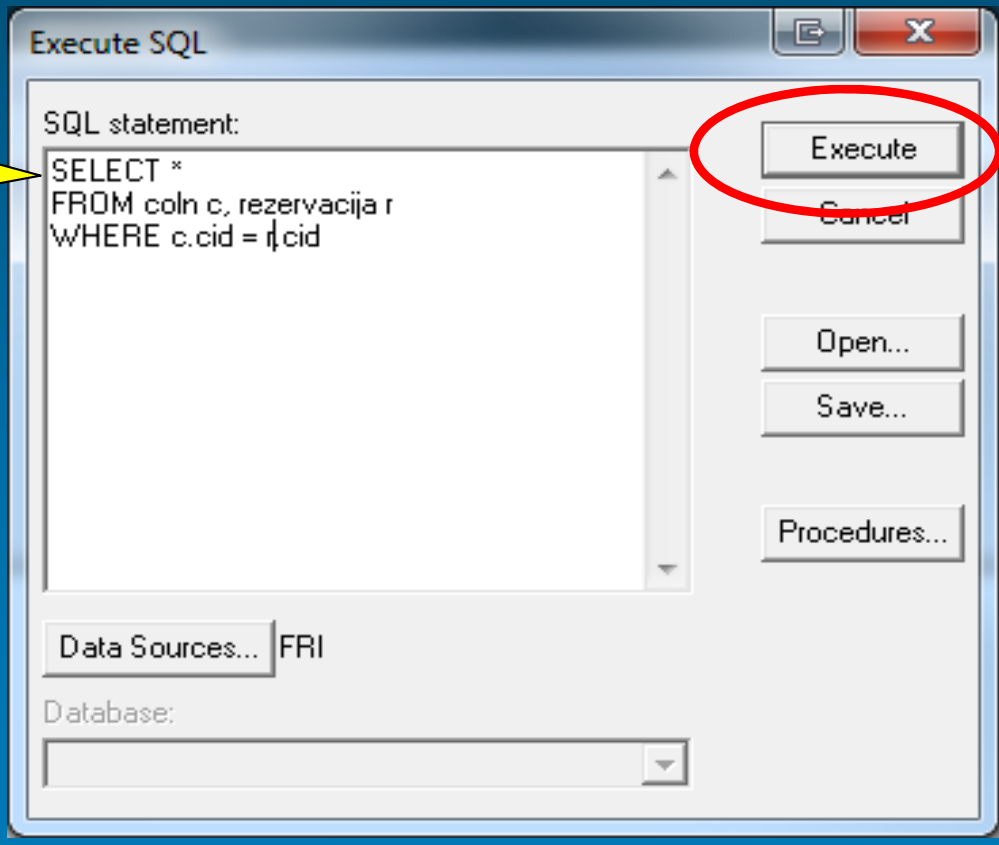


```
Jadralec(jid, ime, rating, starost)
Coln(cid, ime, dolzina, barva)
Rezervacija(jid, cid, dan)
```

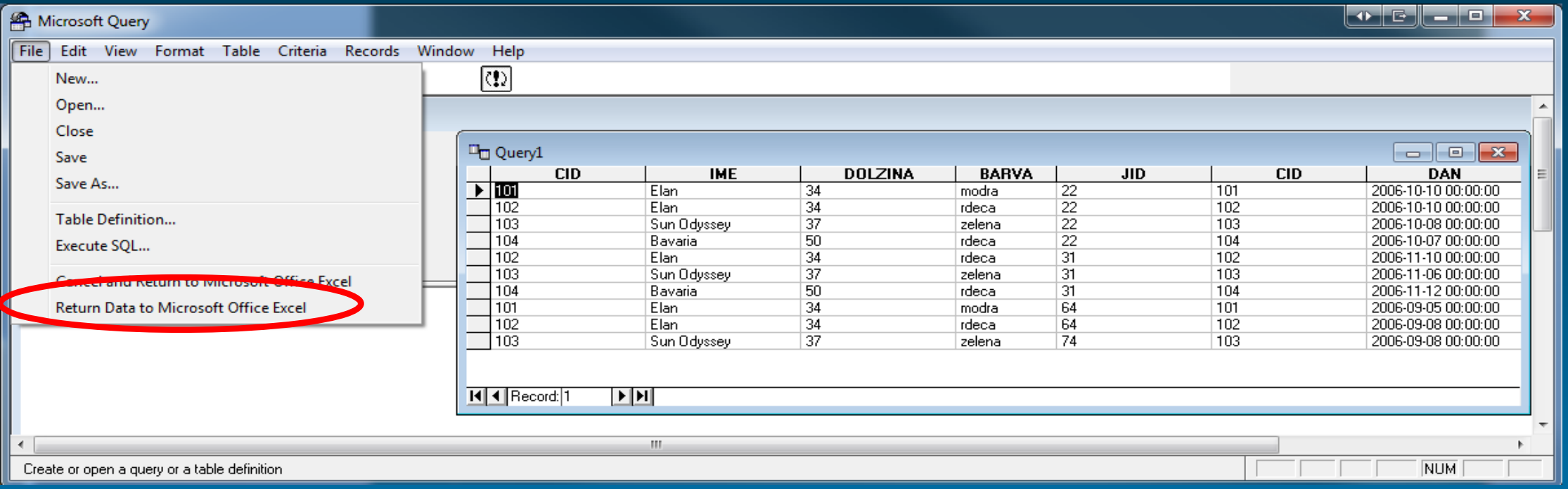
Microsoft Excel in ODBC

- V orodju Microsoft Query ne izberite nobene tabele (Close)
- Izberite File->Execute SQL
- Vnesite SQL poizvedbo in pritisnite gumb Execute

SQL poizvedbo je smiselno napisati in preveriti v za to namenjenem okolju (SQL Developer, MySQL Workbench) ob upoštevanju ODBC omejitev.

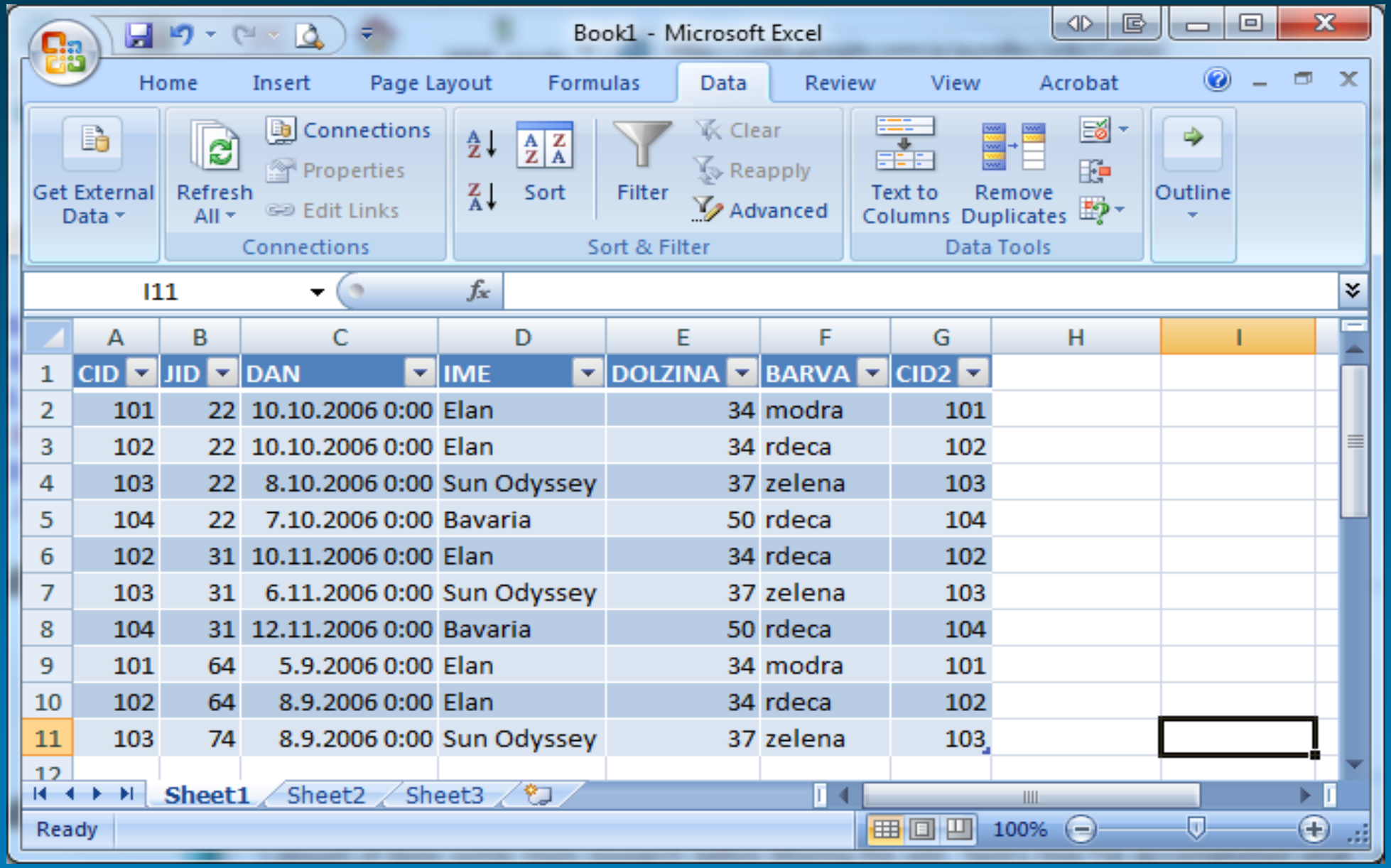


Microsoft Excel in ODBC



- Izberite File->Return Data to Microsoft Excel
- V Excelu dobite tabelo z rezultatom
- Odvisno od definicije DSN (z ali brez gesla) je občasno potrebno vnesti ime in geslo za dostop do podatkovne baze

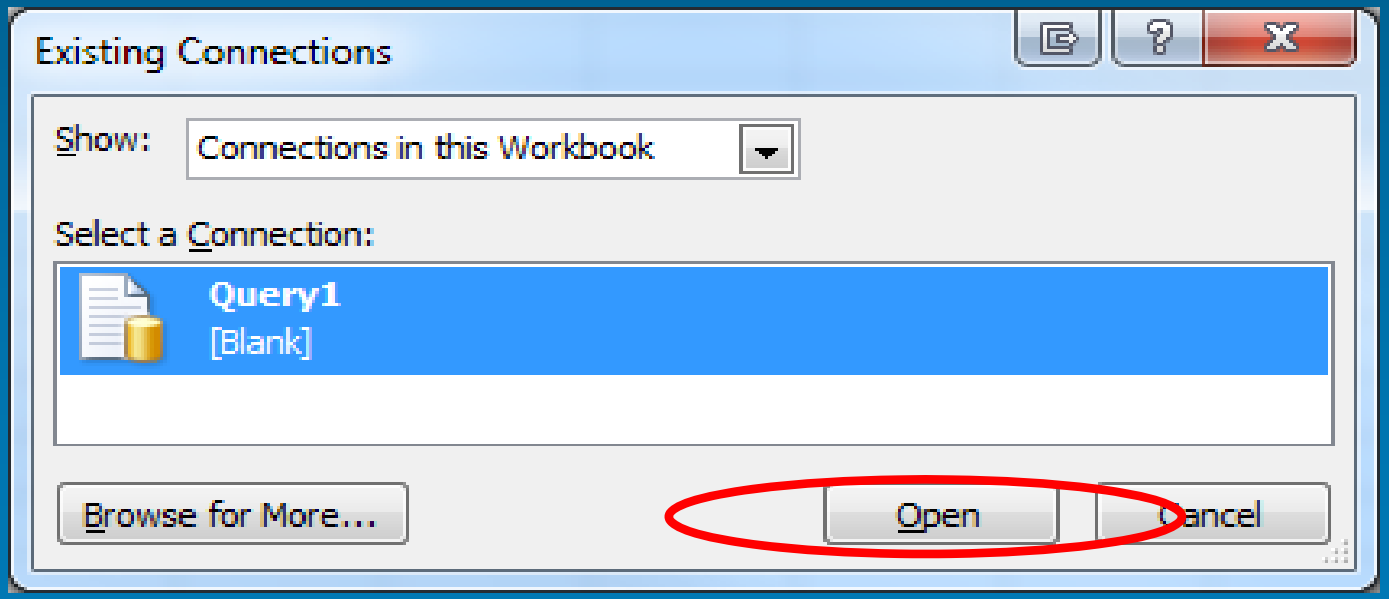
Rezultat poizvedbe v Excelu



	A	B	C	D	E	F	G	H	I
1	CID	JID	DAN	IME	DOLZINA	BARVA	CID2		
2	101	22	10.10.2006 0:00	Elan	34	modra	101		
3	102	22	10.10.2006 0:00	Elan	34	rdeca	102		
4	103	22	8.10.2006 0:00	Sun Odyssey	37	zelena	103		
5	104	22	7.10.2006 0:00	Bavaria	50	rdeca	104		
6	102	31	10.11.2006 0:00	Elan	34	rdeca	102		
7	103	31	6.11.2006 0:00	Sun Odyssey	37	zelena	103		
8	104	31	12.11.2006 0:00	Bavaria	50	rdeca	104		
9	101	64	5.9.2006 0:00	Elan	34	modra	101		
10	102	64	8.9.2006 0:00	Elan	34	rdeca	102		
11	103	74	8.9.2006 0:00	Sun Odyssey	37	zelena	103		
12									

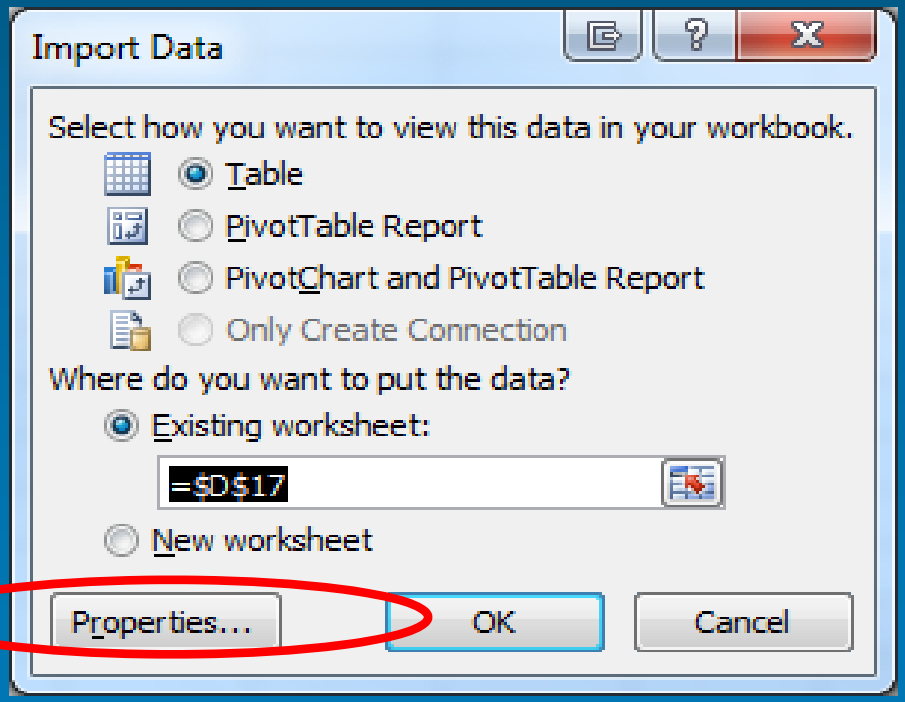
Microsoft Excel in ODBC

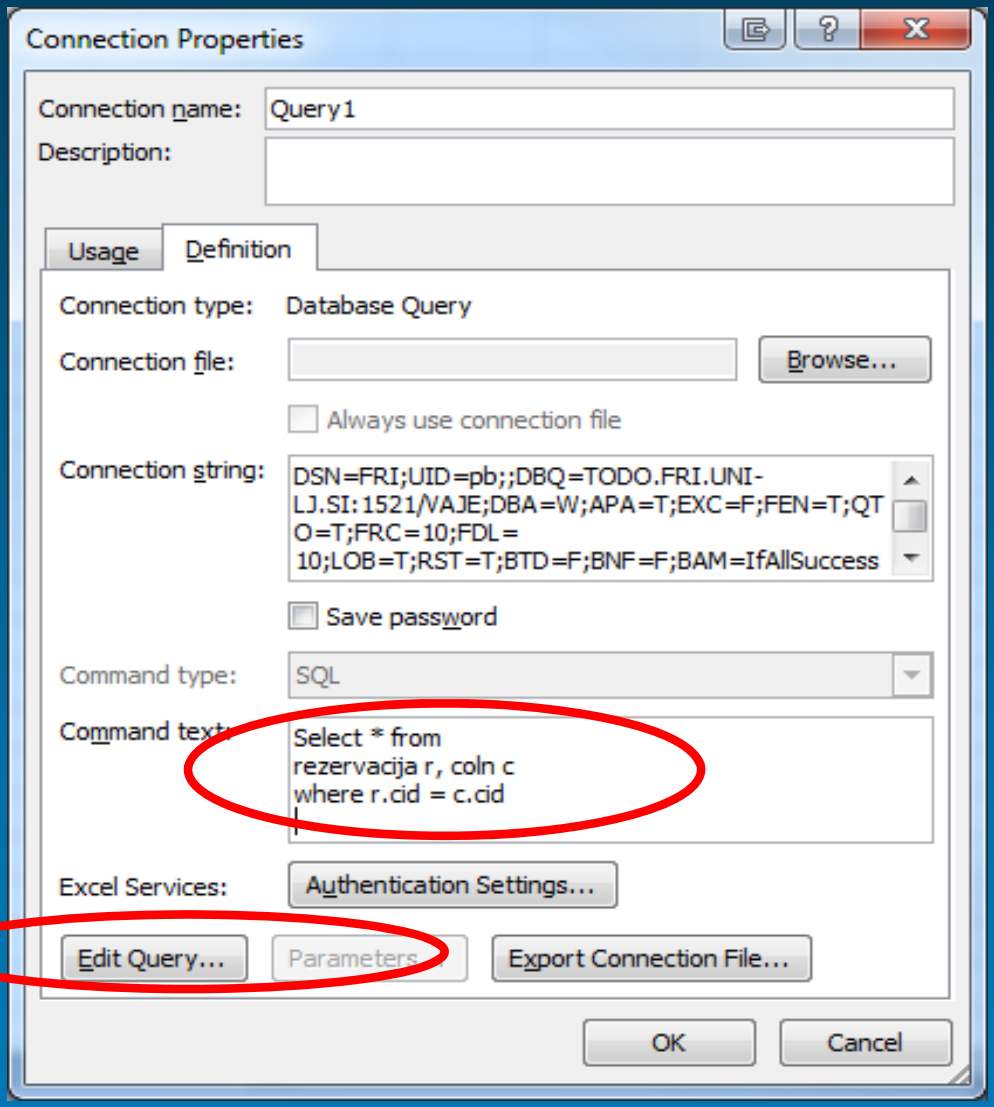
- Povezave s podatkovno bazo so "žive"
  - S pritiskom na Data->Refresh All osvežimo vsebino rezultata poizvedbe
- Urejanje poizvedb
  - Pritisnite Data->Existing Connections
  - Izberite ustrezno poizvedbo in pritisnite Open



Microsoft Excel in ODBC

- Izberite Properties in nato zavihek Definition
- V okencu Command text ali s klikom na Edit Query lahko popravimo poizvedbo





Python – hitra ponovitev

- Ni deklaracije spremenljivk
- Nizi znakov: 'xyz' ali "xyz"
- Struktura programa z zamikanjem
- Zelo obširna standardna knjižnica, na voljo veliko dodatnih modulov
- Pogojni stavki
- Zanke
- Funkcije in procedure

Python – hitra ponovitev

```
# Komentar:
# -  Ni deklaracij spremenljivk
# -  Zamik določa strukturo
# -  Iteracija po razponu (range)
print "Postevanka"
for x in range(1,11):
    for y in range(1,11):
        print x*y,
    print
```

Rezultat

```
Postevanka
1 2 3 4 5 6 7 8 9 10
2 4 6 8 10 12 14 16 18 20
3 6 9 12 15 18 21 24 27 30
4 8 12 16 20 24 28 32 36 40
5 10 15 20 25 30 35 40 45 50
6 12 18 24 30 36 42 48 54 60
7 14 21 28 35 42 49 56 63 70
8 16 24 32 40 48 56 64 72 80
9 18 27 36 45 54 63 72 81 90
10 20 30 40 50 60 70 80 90 100
```

Python – hitra ponovitev

- Formatiranje nizov

```
print "Postevanka"
for x in range(1,11):
    for y in range(1,11):
        izpis = "%3d" % (x*y)
        print izpis,
    print
```

Rezultat

Postevanka

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100

Python – hitra ponovitev

- Moduli

```
from math import pow, sqrt
# Lahko tudi: from math import *

print "Potence in koreni"
for x in range(1,11):
    print x, pow(x,2), sqrt(x)
```

Rezultat

```
Potence in koreni
1 1.0 1.0
2 4.0 1.41421356237
3 9.0 1.73205080757
4 16.0 2.0
5 25.0 2.2360679775
6 36.0 2.44948974278
7 49.0 2.64575131106
8 64.0 2.82842712475
9 81.0 3.010 100.0 3.16227766017
```

Python – hitra ponovitev

- Moduli

```
import math

print "Potence in koreni"
for x in range(1,11):
      # Nadaljevanje vrstice z \
      print "%d\t%3d\t%2.2f" % \
         ( x, math.pow(x,2), math.sqrt(x) )
```

Rezultat

```
Potence in koreni
1         1  1.00
2         4  1.41
3         9  1.73
4        16  2.00
5        25  2.24
6        36  2.45
7        49  2.65
8        64  2.83
9        81  3.00
10      100  3.16
```

Python – podprogrami: procedura

```
def postevanka(n = 10):
    print "Postevanka do %d" % n
    for x in range(1,n+1):
        for y in range(1,11):
            izpis = "%3d" % (x*y)
            print izpis,
        print

postevanka(5)
```

Python – podprogrami: funkcija in seznam

```
def postevanka(n = 10):  # privzet param.
    p = []  # prazen seznam
    for x in range(1,n+1):
        v = []
        for y in range(1,n+):
            v.append(x*y) # dodajanje
        p.append(v)  # na konec seznama
    return p

print postevanka(5) # izpis seznama
```

Rezultat

```
[ [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
  [2, 4, 6, 8, 10, 12, 14, 16, 18, 20],
  [3, 6, 9, 12, 15, 18, 21, 24, 27, 30],
  [4, 8, 12, 16, 20, 24, 28, 32, 36, 40],
  [5, 10, 15, 20, 25, 30, 35, 40, 45, 50]
]
```

Python – podprogrami: funkcija in seznam

```
rezultat = postevanka(5)
# lepsi izpis seznama
for vrstica in rezultat:
    for produkt in vrstica
        print produkt,
    print
```

Rezultat

```
1 2 3 4 5
2 4 6 8 10
3 6 9 12 15
4 8 12 16 20
5 10 15 20 25
```

Predpriprava na uporabo pyodbc

- SUPB s podatki
- Python (3.x) in pyodbc – povezave na učilnici
- Delovno okolje: IPython, PyCharm ali drugo
- ODBC gonilnik za izbrani OS (32/64 bit) in SUPB
- MariaDB in MySQL: Connector/ODBC
  - za Windows (povezava na učilnici)
  - za ostale sisteme: www.mysql.com
- PostgreSQL:
  - za Windows (povezava na učilnici)
  - za ostale sisteme: https://odbc.postgresql.org

pyodbc – implementacija ODBC za Python

- pyodbc je modul za Python ki omogoča dostop do poljubnega SUPB (ki podpora ODBC)
- implementira Python Database API Specification v2.0 z dodatki, ki poenostavljajo delo s podatkovno bazo
- pyodbc je odprtokoden, uporablja MIT licenco, in ga lahko zastonj uporabljamo tako v pridobitne, kot nepridobitne namene (vključno z izvorno kodo)
- domača stran in dokumentacija:
  http://code.google.com/p/pyodbc/
  (seli se na GitHub)
  https://github.com/mkleehammer/pyodbc
-

Osnovni gradniki pyodbc

- Uvoz modula:
  import pyodbc
- Najpomembnejši razredi:
  - Povezava (connection)
  - Kurzor (cursor)
  - Podatkovni tipi in njihovi konstruktorji
  - Obravnava napak

pyodbc: povezava

- Povezavo c ustvarimo z ukazom:
  c = pyodbc.connect(ConnectionString)
- ConnectionString določa povezavo, npr.

  ConnectionString = 'DSN=FRI;UID=pb;PWD=pbvaje'

  ali

  ConnectionString = 'DSN=DOMA;UID=pb;PWD=pbvaje'
- ConnectionString bi lahko napisali tudi brez definiranega DSN:

  ConnectionString = 'DRIVER={MySQL ODBC 5.1 driver};
  SERVER=localhost;DATABASE=vaje;UID=pb;PWD=pbvaje;
  CHARSET=UTF8'

pyodbc: povezava

- Povezava c ponuja metode:
- close(): zapri povezavo, enako pri destruktorju objekta
  - c.close()
- commit(): uveljavi transakcijo (če SUPB podpira)
  - c.commit()
- rollback(): razveljavi transakcijo (če SUPB podpira)
  - c.rollback()
- cursor(): vrne nov kurzorski objekt, ki uporablja povezavo c
  - cursor = c.cursor()

pyodbc: kurzor

- Kurzor x ustvarimo z ukazom:
  x = c.cursor()
- Nekateri atributi:
  - description: opis stolpcev rezultata (shema)
  - rowcount: število vrstic rezultata
- Nekatere metode:
  - execute(ukaz, [parametri]): izvede ukaz z opcijskimi parametri
  - fetchall(): prenese vse vrstice rezultata
  - fetchone(), fetchmany(size): preneseta eno ali več vrstic

```
Jadralec(jid, ime, rating, starost)
Coln(cid, ime, dolzina, barva)
Rezervacija(jid, cid, dan)
```

pyodbc: kurzor

- Po kurzorju lahko iteriramo, vendar samo enkrat:

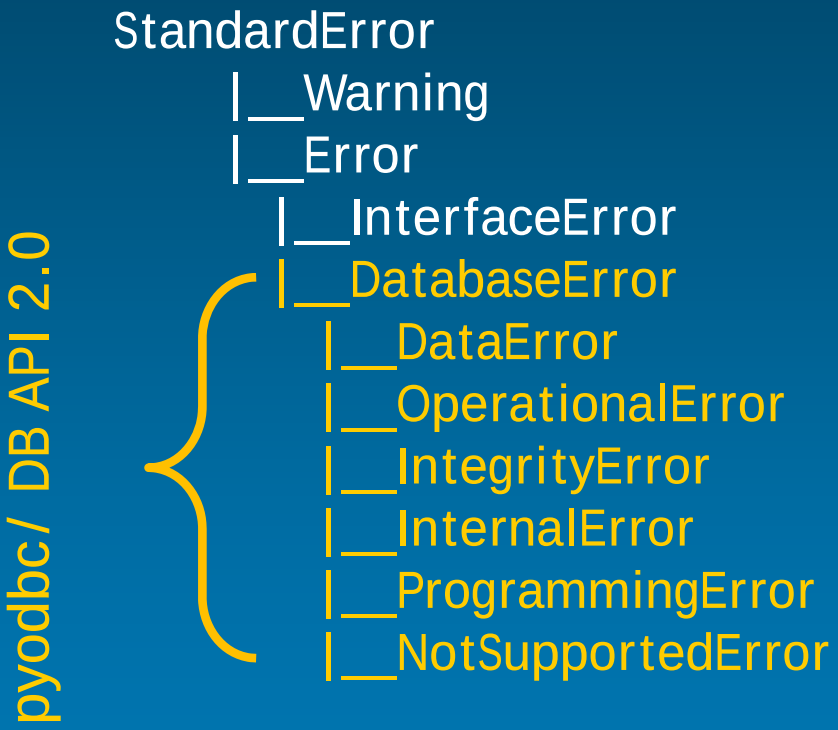
  x.execute(SQLukaz)

  for r in x: print r

- Več iteracij: v = x.fetchall(), nato iteriramo po v
- Parametri v SQL ukazih:
  - Primer: SELECT * FROM jadralec
  - SQL ukaz kot niz znakov: Pythonov način parametrizacije
    - x.execute('SELECT %s FROM %s' % ('*', 'jadralec'))
  - pyodbc prenos parametrov v metodi execute:
    - ? označuje parameter
    - seznam parametrov za ukazom
    - x.execute('SELECT ? FROM ?, '*', 'jadralec')
    - x.execute('SELECT ? FROM ?, ('*', 'jadralec'))

pyodbc: obravnava napak

- Razredi pyodbc ob napakah javljajo naslednje izjeme:
  - DatabaseError
  - DataError
  - OperationalError
  - IntegrityError
  - InternalError
  - ProgrammingError
  - NotSupportedError



pyodbc: obravnava napak

```
try:
        x.execute ( SQLukaz)
        ...
except pyodbc.DataError:
        -- obravnava napake
        pass
...
except pyodbc.DatabaseError:
        -- obravnava napake
        pass
except:
        -- obravnava ostalih napak
        pass
```

pyodbc: preslikava med ODBC/SQL in Pythonovimi podatkovnimi tipi

ODBC	Python
char varchar longvarchar GUID	string
wchar wvarchar wlongvarchar	unicode
smallint integer tinyint	int
bigint	long
decimal numeric	decimal
real float double	double
date	datetime.date
time	datetime.time
timestamp	datetime.datetime
bit	bool
binary varbinary longvarbinary	buffer
SQL Server XML type	unicode

```
Jadralec(jid, ime, rating, starost)
Coln(cid, ime, dolzina, barva)
Rezervacija(jid, cid, dan)
```

Primer programa

- Naloga: postaraj jadralce za eno leto
- Izvedba:
  - ustvari novo tabelo: postarani
  - v vsaki vrstici povecaj starost za 1
- Vse to lahko naredimo direktno v SQL-u. Kako?

Primer programa

```
import pyodbc
cnxn = pyodbc.connect('DSN=FRI;UID=pb;PWD=pbvaje')
cursor  = cnxn.cursor()
updater= cnxn.cursor()
try:
    cursor.execute("DROP TABLE postarani")
except pyodbc.DatabaseError:
    pass
cursor.execute("CREATE TABLE postarani AS SELECT * FROM jadralec")
cnxn.commit()
```

Izbriši, če že obstaja tabela.

Primer programa

```
cursor.execute("SELECT * from postarani")
print "PRED"
for r in cursor:
    print r
cursor.execute("SELECT * from postarani")
for r in cursor:
     updater.execute("UPDATE postarani SET starost =? WHERE jid =?",
                              r.STAROST + 1, r.JID)
cursor.execute("SELECT * from postarani")
print "PO"
for r in cursor:
    print r
cnxn.commit()
```

Zakaj še en SELECT?

Pazite na velikost črk pri imenih atributov

COMMIT zares zapiše v bazo

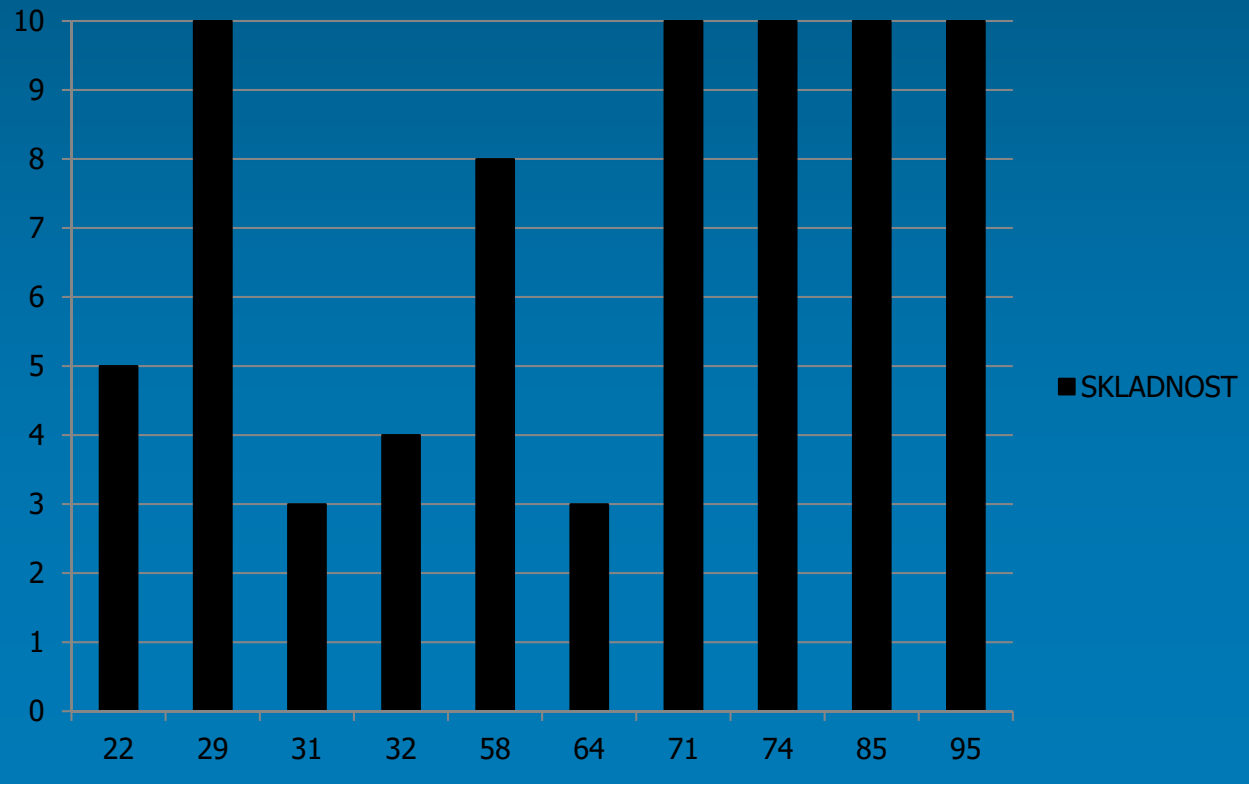
NALOGA

- V Pythonu napišite program, ki za vsakega jadralca izpiše šifro najbolj skladnega čolna.
- Rezultate zapišite v tabelo
  optimal(jid, cid, skladnost)
- Najbolj skladen čoln *cid* za jadralca *jid* je tisti, ki po formuli *(jid+cid) % 11* daje najvišjo vrednost.

NALOGA V EXCELU

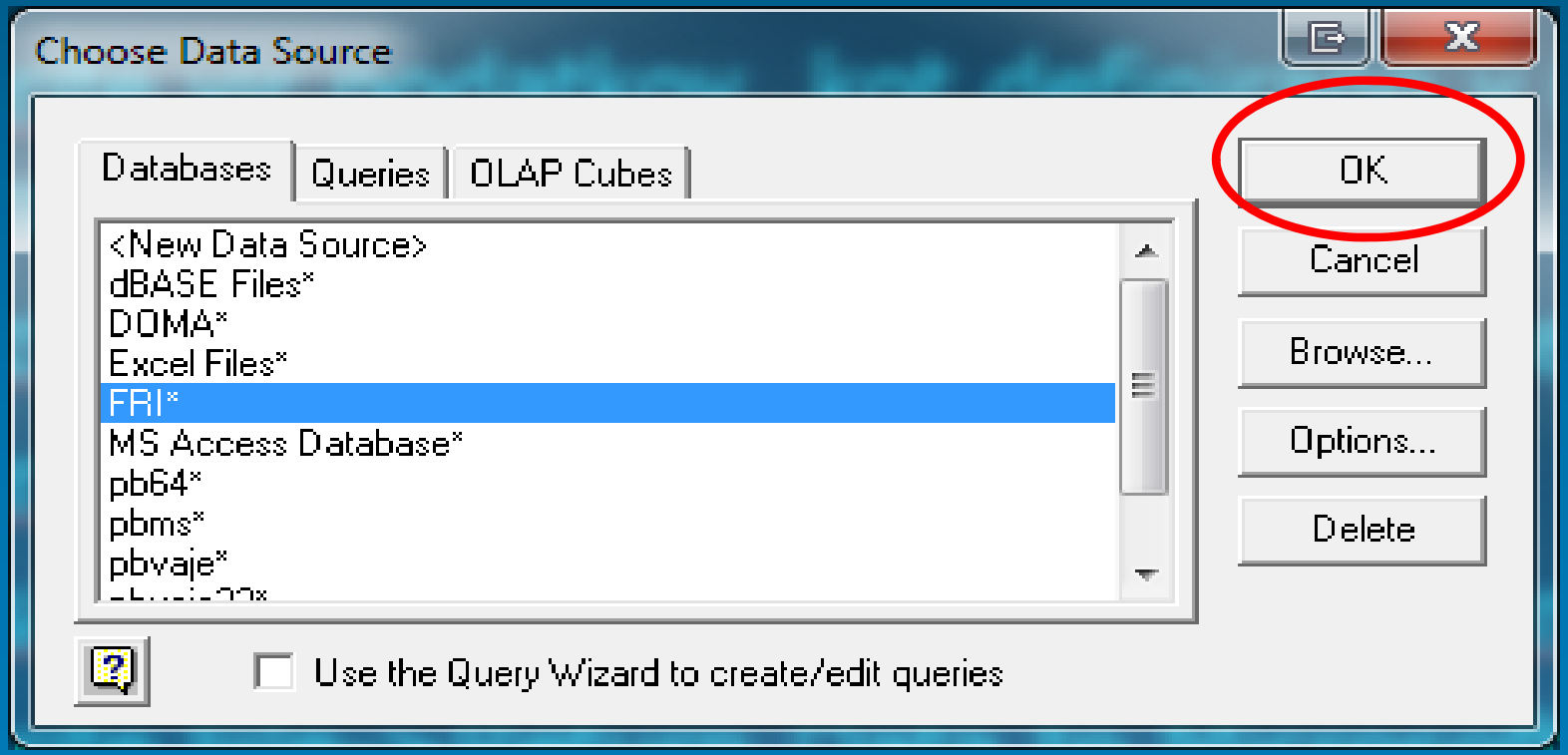
- Povežite se na bazo in prenesite tabelo optimal
- Narišite graf te tabele, kot ga kaže slika.

JID	CID	SKLADNOST
22	104	5
29	102	10
31	104	3
32	104	4
58	104	8
64	104	3
71	104	10
74	101	10
85	101	10
95	102	10



Microsoft Excel in ODBC

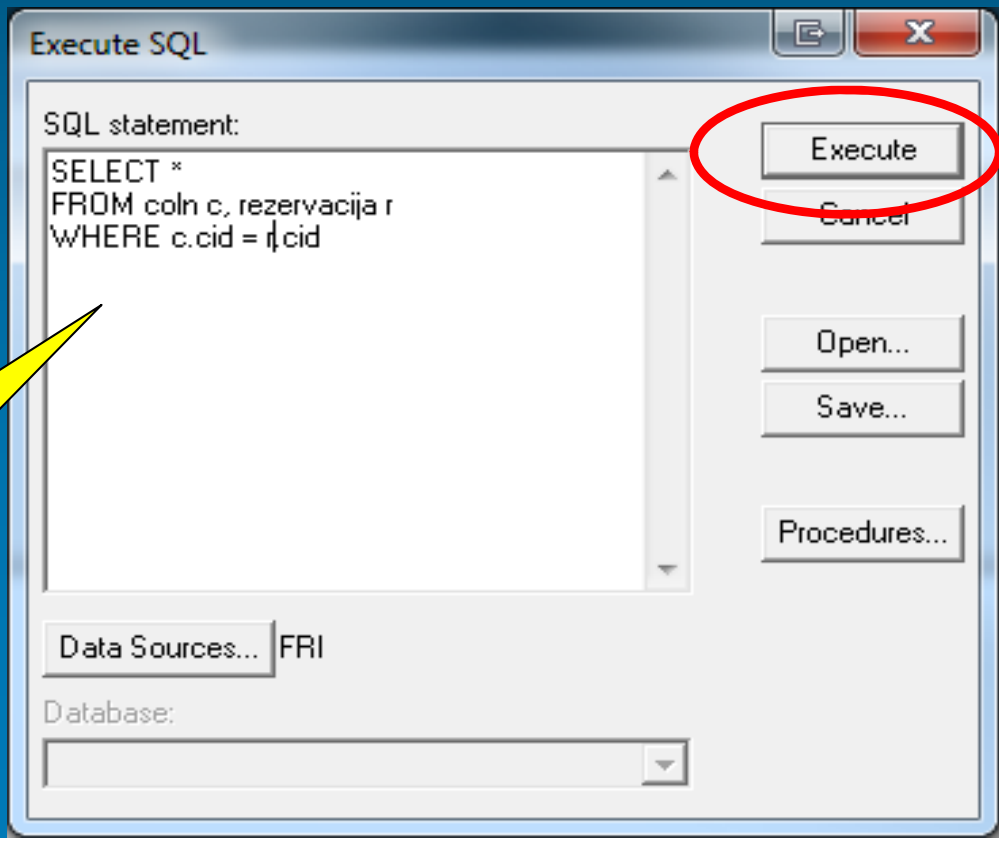
- Izberite
  Data->From Other Sources->From Microsoft Query
- Izberite vir podatkov, kot definirano v ODBC Data Sources (npr. FRI)



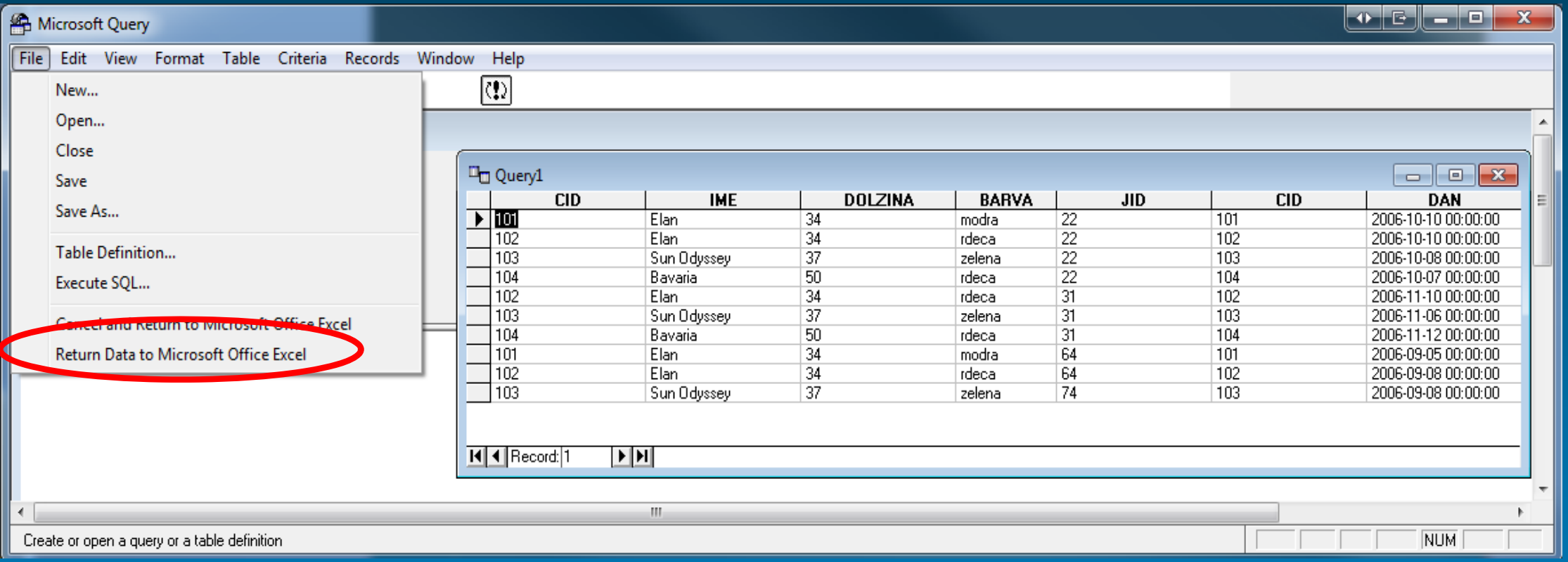
Microsoft Excel in ODBC

- Ne izberite nobene tabele (gumb Close)
- Izberite File->Execute SQL
- Vnesite SQL poizvedbo in pritisnite gumb Execute

SQL poizvedbo je smiselno napisati in preveriti v za to namenjenem okolju (SQL Developer, MySQL Workbench) ob upoštevanju ODBC omejitev.

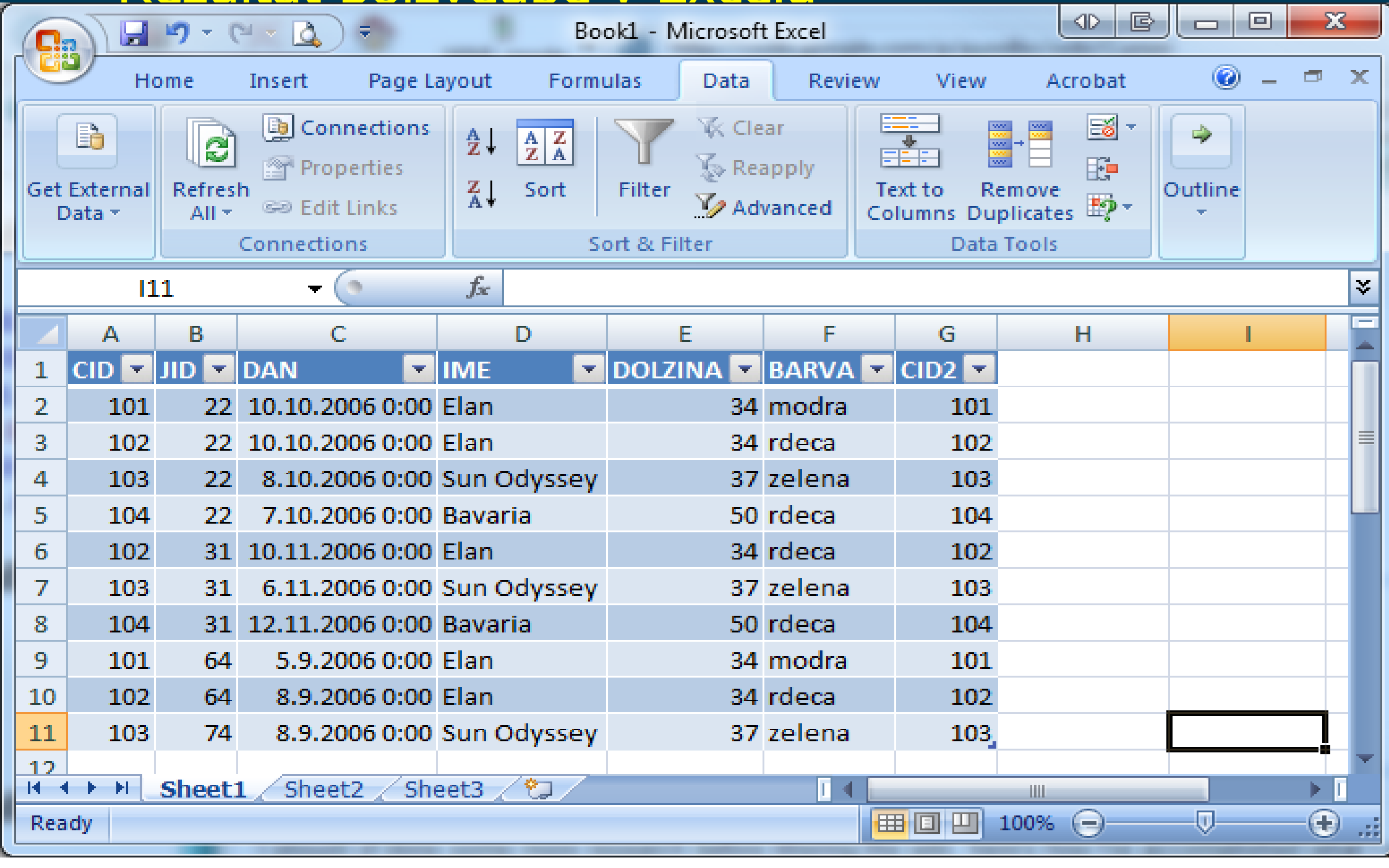


Microsoft Excel in ODBC



- Izberite File->Return Data to Microsoft Excel
- V Excelu dobite tabelo z rezultatom
- Odvisno od definicije DSN (z ali brez gesla) je občasno potrebno vnesti ime in geslo za dostop do podatkovne baze

Rezultat poizvedbe v Excelu



	A	B	C	D	E	F	G	H	I
1	CID	JID	DAN	IME	DOLZINA	BARVA	CID2		
2	101	22	10.10.2006 0:00	Elan	34	modra	101		
3	102	22	10.10.2006 0:00	Elan	34	rdeca	102		
4	103	22	8.10.2006 0:00	Sun Odyssey	37	zelena	103		
5	104	22	7.10.2006 0:00	Bavaria	50	rdeca	104		
6	102	31	10.11.2006 0:00	Elan	34	rdeca	102		
7	103	31	6.11.2006 0:00	Sun Odyssey	37	zelena	103		
8	104	31	12.11.2006 0:00	Bavaria	50	rdeca	104		
9	101	64	5.9.2006 0:00	Elan	34	modra	101		
10	102	64	8.9.2006 0:00	Elan	34	rdeca	102		
11	103	74	8.9.2006 0:00	Sun Odyssey	37	zelena	103		
12									