

pymssql notes

Here are some notes on making a connection to your Cypress SQL Server database using pymssql. The example is being run in IDLE (the Python Interpreter), connecting to CSIL via Remote Desktop. Of course, in practice I'd write a Python module – the interpreter is mainly for investigation.

I started by opening IDLE in the Python 2.7 folder in the Start menu. Lines beginning with the >>> prompt are what I typed. Lines in red are editorial comments about what I'm doing.

```
>>> import pymssql #import python ms sql library

>>> conn = pymssql.connect(host='cypress.csil.sfu.ca',
user='s_johnwill', password='<your SQL Server Password>',
database='cmpt354_bank')

#If you want to avoid frustration you need to read this section
#carefully.

#Replace 's_johnwill' with 's_<your SFU ID>'

#Replace password with '<your SQL Server password>'.

#It's important that this is not your SFU password, and that you
#are very unlikely to know the value of this password!

#You can find it in your database's dbo.helpdesk table. Navigate
#to this table in SQL Server (expand your database, then expand
#tables), right click on it and select "Select top 1000 rows".
#There is, in fact, only one row which shows your user id, and
#your hideously ugly password.

#Replace 'cmpt354_bank' with your database name (<username354>)

>>> mycursor = conn.cursor()

#I've created a cursor, now let's attach it to something

>>> mycursor.execute('SELECT * from account')

#Now let's print the contents of the Account table, row by row

>>> row = mycursor.fetchone()

>>> while row:
    print row
    row = mycursor.fetchone()

#And you get some ugly output
(1, u'SAV', Decimal('118231.13'), 1)
...
(298, u'SAV', Decimal('92525.41'), 4)

>>> mycursor.close()
```

pymssql notes

```
#Don't re-use the cursor until you've closed it. Then define it  
#again and attach it to another query.
```

```
#Let's make the output less ugly
```

```
>>> mycursor = conn.cursor()
```

```
>>> mycursor.execute('SELECT * from account')
```

```
>>> row = mycursor.fetchone()
```

```
>>> print "Account %d: type = %s, balance = $%f, branch = %d"  
%(row[0], row [1], row[2], row[3])
```

```
Account 1: type = SAV, balance = $118231.130000, branch = 1
```

```
>>>
```