



Python and Database Access on IBM i

MIKE PAVLAK



mikepavlak@gmail.com



Today's schedule

12:00 - Introduction to Python for RPGers (Mike)

1:30 - Python and Data Access (Mike)

3:00 - RPG Makes Friends with Open Source Apps (Richard)

4:30 - Python Modules for Profitability (Mike)

7:00 - Pass the Flask & Quickly Pour IBM I Python Web Apps & Services (Richard)



Agenda

Set review and modules

ODBC & PyODBC

SQLite

Postgresql

Summary and wrap up



Set review and Modules

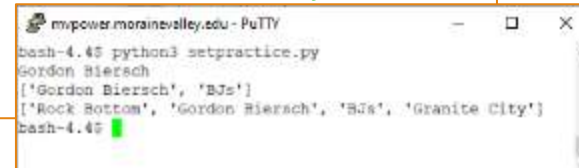


Python sets

List

- Mutable, integer index

```
myList = ['Rock Bottom', 'Gordon Biersch', 'BJs', 'Granite City']  
print(myList[1])  
print(myList[1:3])  
print(myList)
```

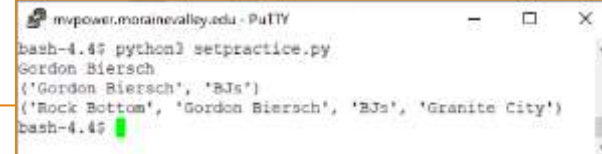


A terminal window titled 'mvpower.morainevalley.edu - PuTTY' showing the execution of a Python script. The output is: bash-4.4\$ python3 setpractice.py
Gordon Biersch
['Gordon Biersch', 'BJs']
['Rock Bottom', 'Gordon Biersch', 'BJs', 'Granite City']
bash-4.4\$

Tuple

- Immutable, integer index

```
myTuple = ('Rock Bottom', 'Gordon Biersch', 'BJs', 'Granite City')  
print(myTuple[1])  
print(myTuple[1:3])  
print(myTuple)
```

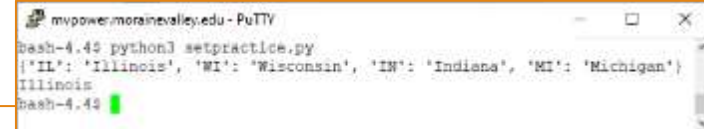


A terminal window titled 'mvpower.morainevalley.edu - PuTTY' showing the execution of a Python script. The output is: bash-4.4\$ python3 setpractice.py
Gordon Biersch
('Gordon Biersch', 'BJs')
('Rock Bottom', 'Gordon Biersch', 'BJs', 'Granite City')
bash-4.4\$

Dictionary

- Mutable, character index

```
statesByMichigan = {'IL': 'Illinois', 'WI': 'Wisconsin', 'IN': 'Indiana',  
                   'MI': 'Michigan'}  
print(statesByMichigan)  
print(statesByMichigan['IL'])
```

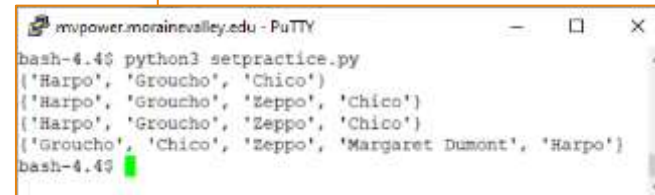


A terminal window titled 'mvpower.morainevalley.edu - PuTTY' showing the execution of a Python script. The output is: bash-4.4\$ python3 setpractice.py
{'IL': 'Illinois', 'WI': 'Wisconsin', 'IN': 'Indiana', 'MI': 'Michigan'}
Illinois
bash-4.4\$

Set

- Mutable, integer index
- Only allows distinct values

```
marxBros = set(['Groucho', 'Chico', 'Harpo'])  
print(marxBros)  
marxBros.add('Zeppo')  
print(marxBros)  
marxBros.add('Groucho')  
print(marxBros)  
marxBros.add('Margaret Dumont')  
print(marxBros)
```



A terminal window titled 'mvpower.morainevalley.edu - PuTTY' showing the execution of a Python script. The output is: bash-4.4\$ python3 setpractice.py
{'Harpo', 'Groucho', 'Chico'}
{'Harpo', 'Groucho', 'Zeppo', 'Chico'}
{'Harpo', 'Groucho', 'Zeppo', 'Chico'}
{'Groucho', 'Chico', 'Zeppo', 'Margaret Dumont', 'Harpo'}
bash-4.4\$

What is a module

Code library – hunks of Python code

They come from the Python community

- PSL: Python Standard Library
- Home grown
- Third party

Usually Python, but not necessarily...

Designed to augment, not replace

Think of them as copybooks

Tested, trusted, useful...

The letters 'OPC' are rendered in a bold, three-dimensional font with a brown, metallic texture. The letters are slightly shadowed, giving them a sense of depth and weight. They are positioned on the right side of the slide, centered vertically relative to the text on the left.

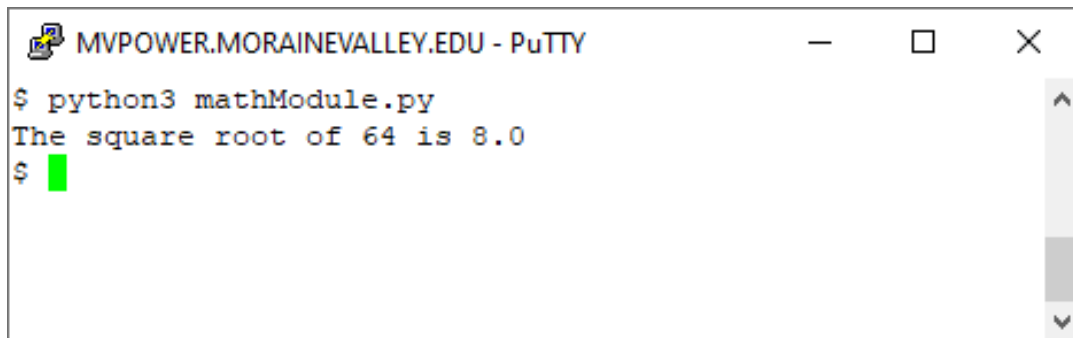
Math module example

Let's say we want to do a square root or use pi

```
import math

x = 64
y = math.sqrt(x)

print("The square root of", x, "is", y)
```

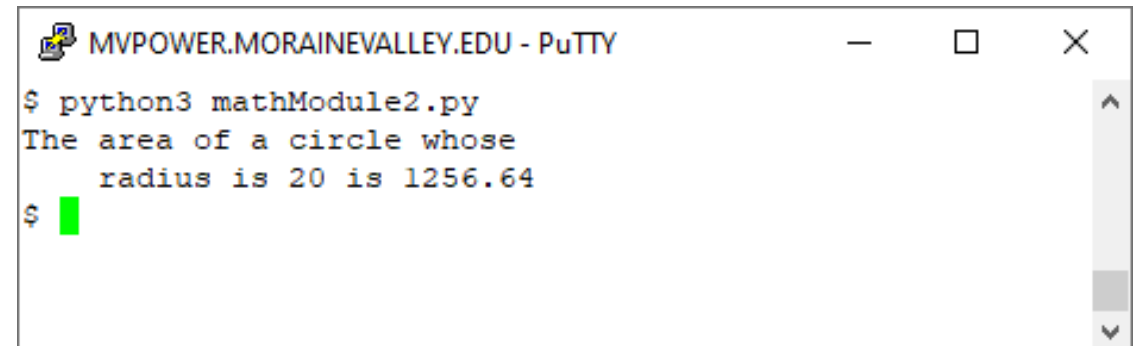


```
MVPOWER.MORAIN VALLEY.EDU - PuTTY
$ python3 mathModule.py
The square root of 64 is 8.0
$
```

```
import math

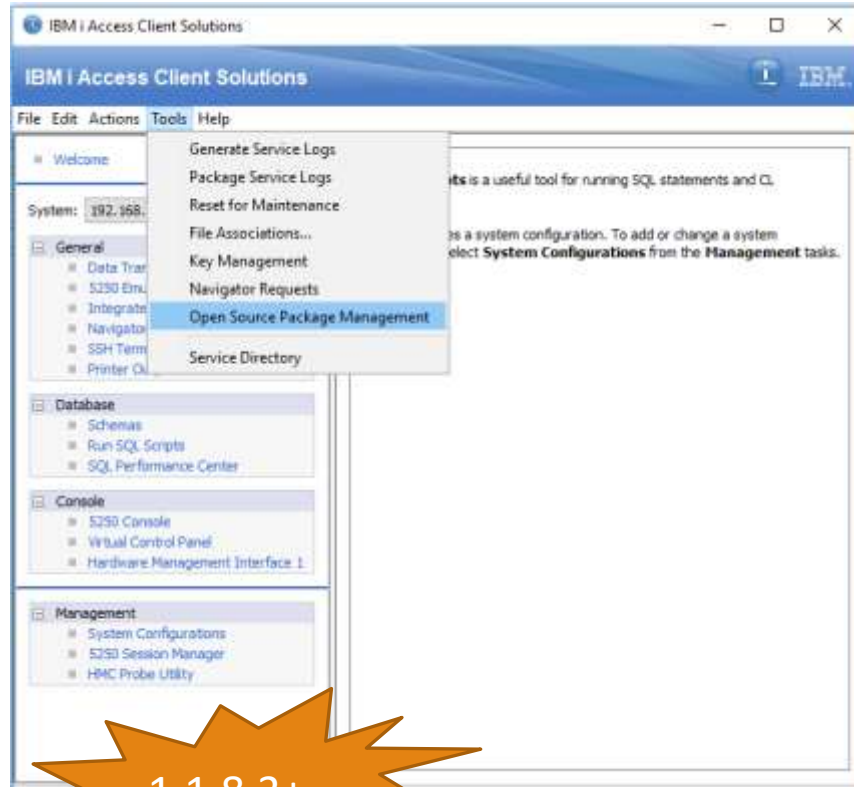
# Calculate the area of circle
radius = 20
area = round((math.pi * radius**2),2)

print("""The area of a circle whose
radius is""", radius, "is", area)
```

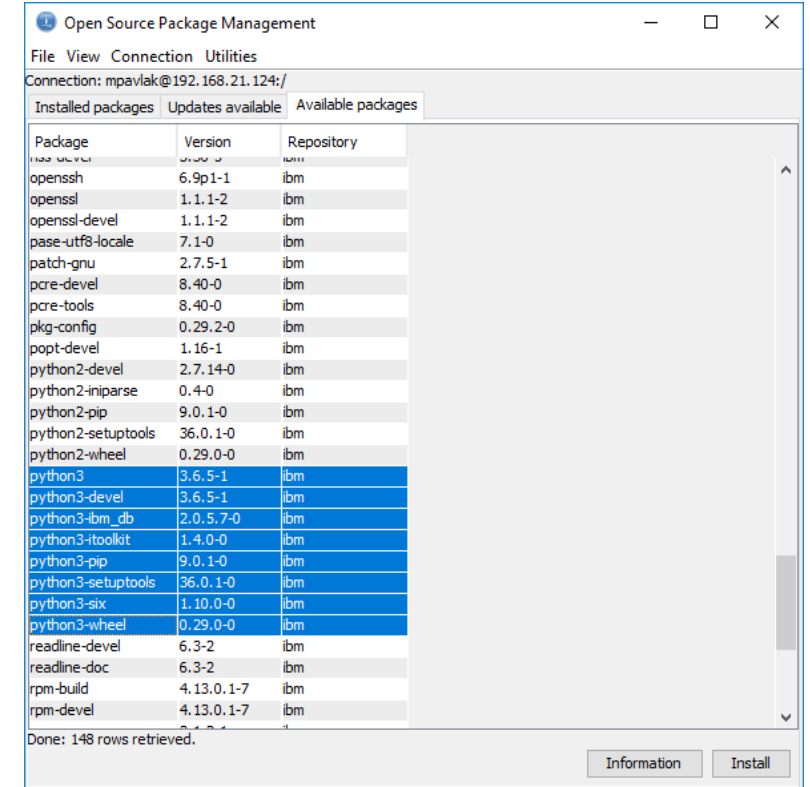
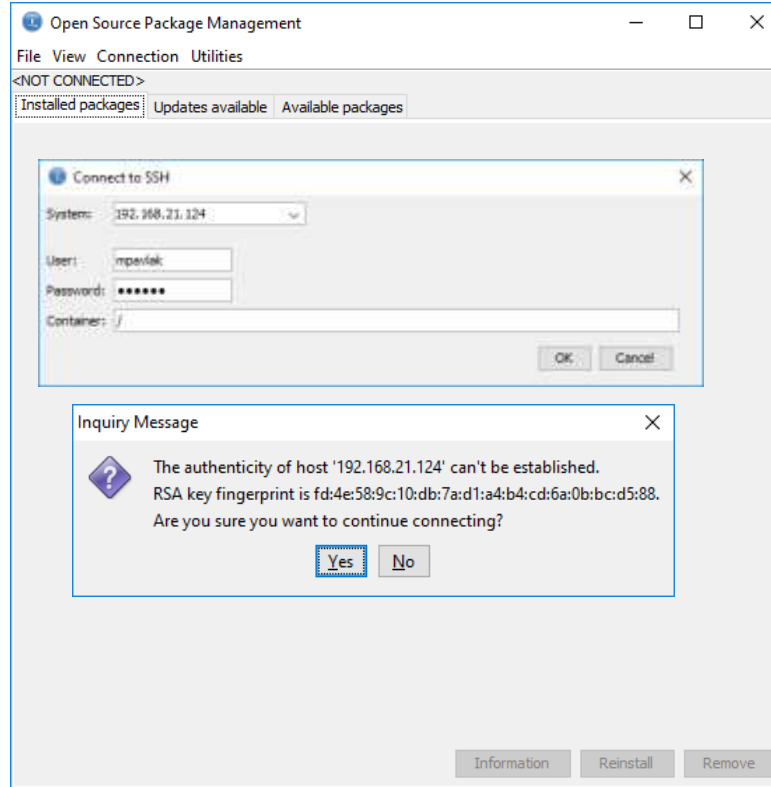


```
MVPOWER.MORAIN VALLEY.EDU - PuTTY
$ python3 mathModule2.py
The area of a circle whose
radius is 20 is 1256.64
$
```

Open Source Package Management (ACS)



1.1.8.3+



More on this in a bit...

ODBC



ODBC Needs 3 pieces to work

ODBC Driver

ODBC Driver Manager

Application specific connectors (adapter)

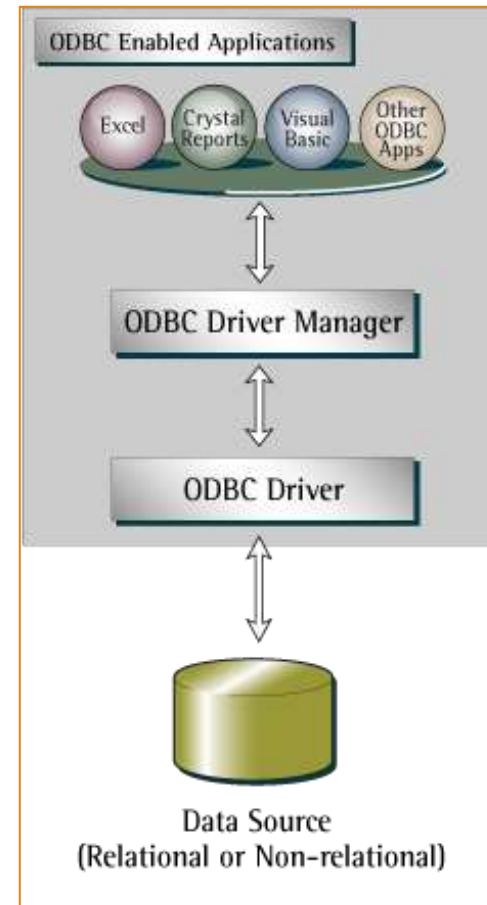
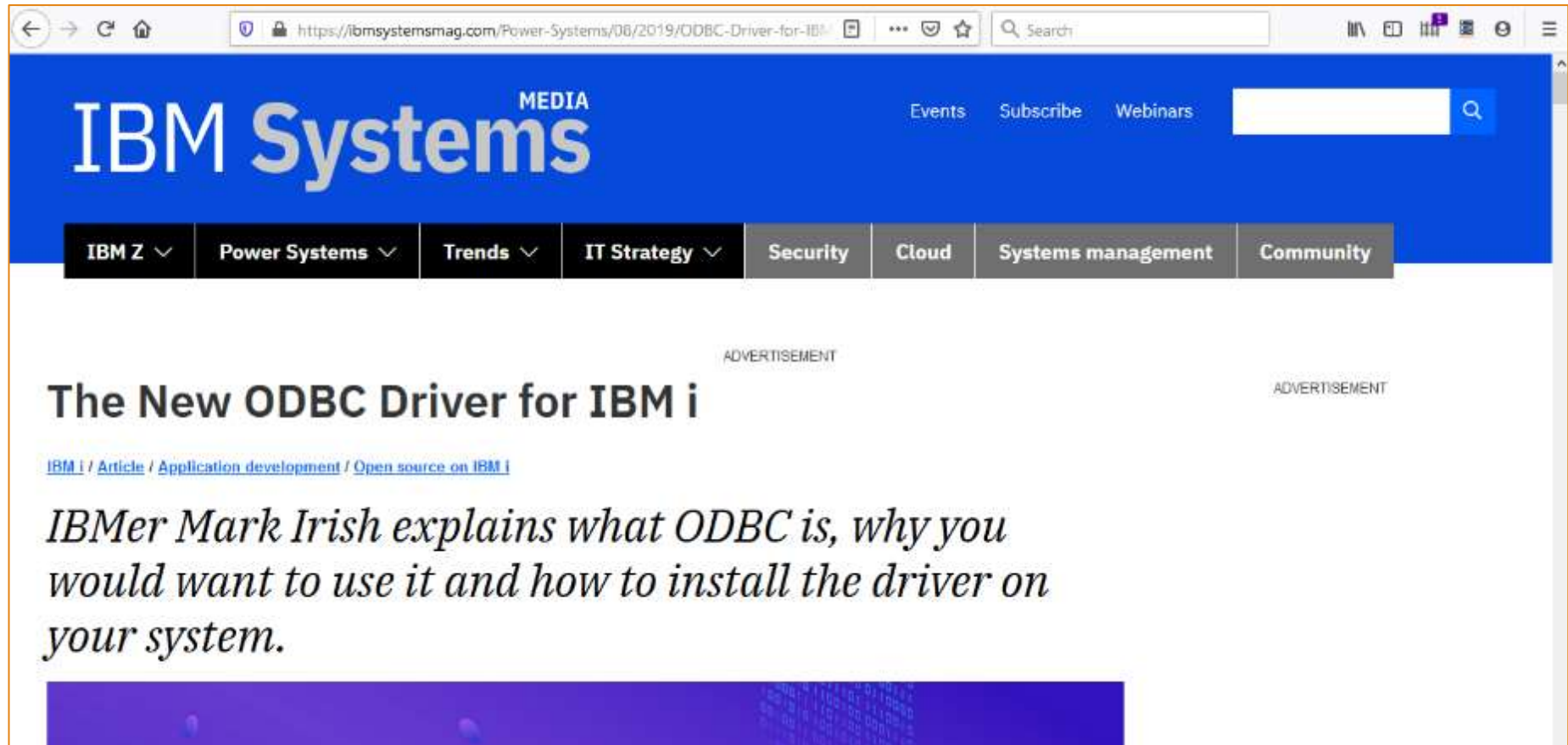


Image: Wikimedia Commons

ODBC is where it's at...

Read the article by Mark Irish

<https://ibmsystemsmag.com/Power-Systems/08/2019/ODBC-Driver-for-IBM-i>

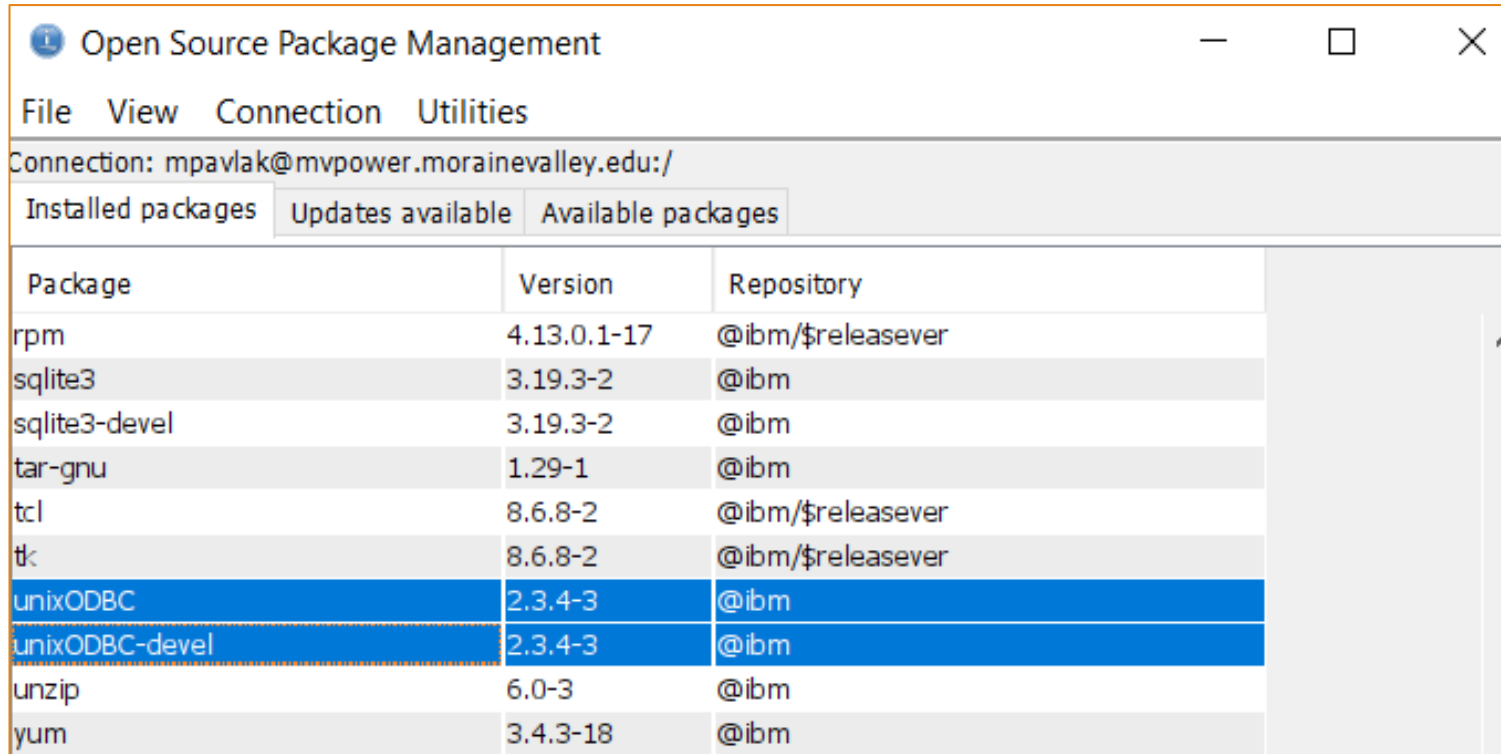


The screenshot shows a web browser displaying the IBM Systems Media website. The page features a blue header with the IBM Systems logo and navigation links for Events, Subscribe, and Webinars. Below the header is a dark navigation bar with categories like IBM Z, Power Systems, Trends, IT Strategy, Security, Cloud, Systems management, and Community. The main content area displays an article titled "The New ODBC Driver for IBM i" with a sub-header "ADVERTISEMENT". The article text reads: "IBMer Mark Irish explains what ODBC is, why you would want to use it and how to install the driver on your system." Below the text is a blue banner with a grid pattern.

ODBC: Driver Manager

Step 1

- Command line: **yum install unixODBC unixODBC-devel**
- GUI:











The screenshot shows the Open Source Package Management (OSPM) window. The title bar reads "Open Source Package Management". The menu bar includes "File", "View", "Connection", and "Utilities". The connection information is "mpavlak@mvpower.morainevalley.edu:". The "Available packages" tab is selected, displaying a table of packages.

Package	Version	Repository
rpm	4.13.0.1-17	@ibm/\$releasever
sqlite3	3.19.3-2	@ibm
sqlite3-devel	3.19.3-2	@ibm
tar-gnu	1.29-1	@ibm
tcl	8.6.8-2	@ibm/\$releasever
tk	8.6.8-2	@ibm/\$releasever
unixODBC	2.3.4-3	@ibm
unixODBC-devel	2.3.4-3	@ibm
unzip	6.0-3	@ibm
yum	3.4.3-18	@ibm

ODBC Driver is on ACS site

Step 2:

- Dowload
- Install

Description	Filename	Size	Action
IBM i Access Client Solutions	IBMiAccess_v1r1.zip	134107575 B	Download 
Readme file for 5733-XJ1	Readme.txt	5 KB	Download 
Quick Start Guide	QuickStartGuide_en.html	10 KB	Download 
Getting Started	GettingStarted_en.html	119 KB	Download 
ACS EHLAPI - Version 8.20.104	acshllapi.exe	986862	Download 
ACS Windows App Pkg English (64bit)	IBMiAccess_v1r1_WindowsAP_English.zip	53311376 B	Download 
ACS Linux App Pkg	IBMiAccess_v1r1_LinuxAP.zip	20797233 B	Download 
ACS PASE App Pkg	IBMiAccess_v1r1_PASE_AP.zip	8577229 B	Download 

ODBC Driver is on ACS site

```
mvpower.morainevalley.edu - PuTTY
login as: mpavlak
mpavlak@mvpower.morainevalley.edu's password:
$ cd /mvccutil
$ ls
IBMiAccess_v1r1_PASE_AP.zip
$ unzip IBMiAccess_v1r1_PASE_AP.zip
Archive:  IBMiAccess_v1r1_PASE_AP.zip
  creating: License/
  inflating: License/LA_cs.txt
  inflating: License/LA_de.txt
  inflating: License/LA_el.txt
  inflating: License/LA_en.txt
  inflating: License/LA_es.txt
  inflating: License/LA_fr.txt
  inflating: License/LA_in.txt
  inflating: License/LA_it.txt
  inflating: License/LA_ja.txt
  inflating: License/LA_ko.txt
  inflating: License/LA_lt.txt
  inflating: License/LA_pl.txt
  inflating: License/LA_pt.txt
  inflating: License/LA_ru.txt
  inflating: License/LA_sl.txt
  inflating: License/LA_tr.txt
  inflating: License/LA_zh.txt
  inflating: License/LA_zh_TW.txt
  inflating: License/cs.html
  inflating: License/de.html
```

ODBC Install

YUM command works here

This was my example, values will change

```
mvpower.morainevalley.edu - PuTTY
$ pwd
/mvccutil/ppc64
$ ls
ibm-iaccess-1.1.0.13-0.ibm7.2.ppc64.rpm
$ yum install ibm-iaccess-1.1.0.13-0.ibm7.2.ppc64.rpm
Setting up Install Process
Examining ibm-iaccess-1.1.0.13-0.ibm7.2.ppc64.rpm: ibm-iaccess-1.1.0.13-0.ppc64
Marking ibm-iaccess-1.1.0.13-0.ibm7.2.ppc64.rpm as an update to ibm-iaccess-1.1.0.11-0.ppc64
Resolving Dependencies
--> Running transaction check
---> Package ibm-iaccess.ppc64 0:1.1.0.11-0 will be updated
---> Package ibm-iaccess.ppc64 0:1.1.0.13-0 will be an update
--> Finished Dependency Resolution

Dependencies Resolved

-----
Package      Arch    Version      Repository      Size
-----
Updating:
ibm-iaccess  ppc64  1.1.0.13-0   /ibm-iaccess-1.1.0.13-0.ibm7.2.ppc64  43 M

Transaction Summary
-----
Upgrade      1 Package

Total size: 43 M
```

ODBC or upgrade

Upgrade works here, too

```
Upgrade      1 Package
Total size: 43 M
Is this ok [y/N]: y
Downloading Packages:
Running Transaction Check
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Updating   : ibm-iaccess-1.1.0.13-0.ppc64                1/2
  Cleanup    : ibm-iaccess-1.1.0.11-0.ppc64                2/2
Updated:
  ibm-iaccess.ppc64 0:1.1.0.13-0
Complete!
```


Test from command line

Use “isql” part of ODBC

User is defaulted to current

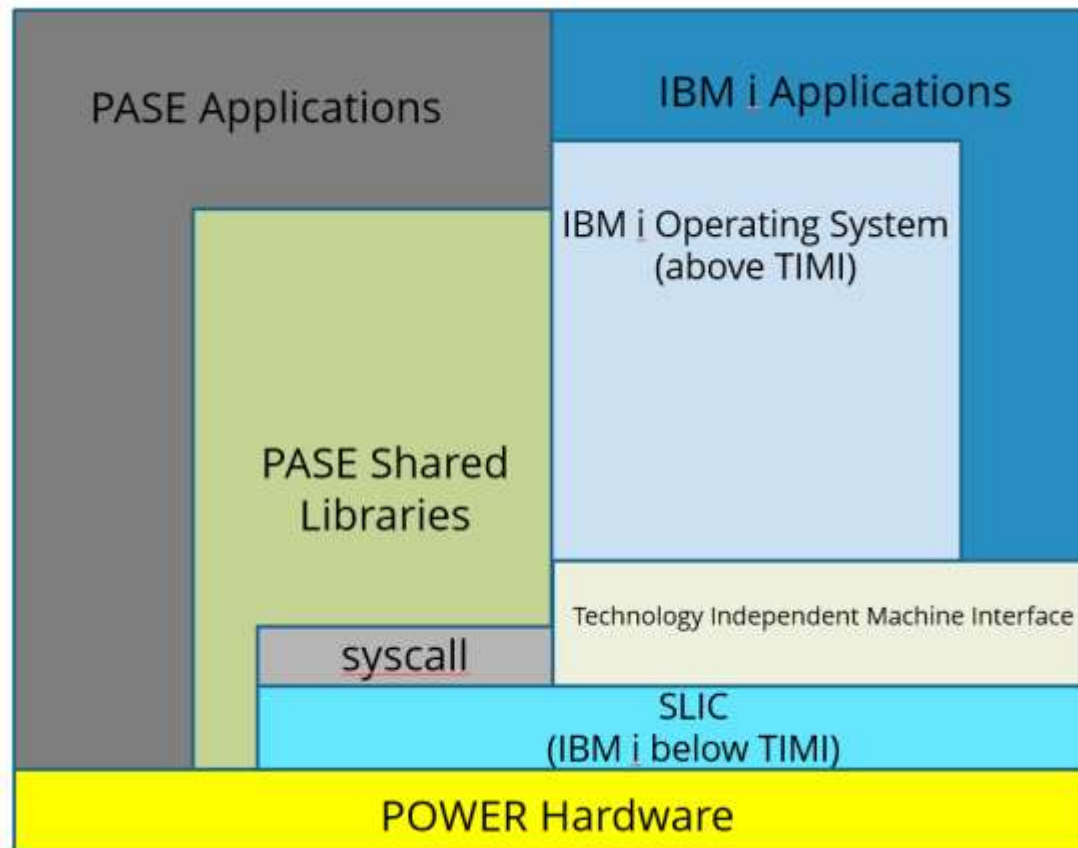
[.odbc.ini](#) file auto created

- Can be modified

```
mvpower.morainevalley.edu - PuTTY
bash-4.4$ isql '*LOCAL'
+-----+
| Connected!
|
| sql-statement
| help [tablename]
| quit
|
+-----+
SQL> select * from qiws.qcustcdt
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| CUSNUM | LSTNAM | INIT| STREET          | CITY  | STATE| ZIPCOD | CDTLMT| CHGCOD| BALDUE | CTDUE |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 938472 | Henning| G K | 4859 Elm Ave   | Dallas| TX   | 75217  | 5000  | 3     | 37.00  | 0     |
| 839283 | Jones  | B D | 21B NW 135 St | Clay  | NY   | 13041  | 400   | 1     | 100.00 | 0     |
| 392859 | Vine   | S S | PO Box 79     | Broton| VT   | 5046   | 700   | 1     | 439.00 | 0     |
| 938485 | Johnson| J A | 3 Alpine Way  | Helen | GA   | 30545  | 9999  | 2     | 3987.50| 33.50 |
| 397267 | Tyron  | W E | 13 Myrtle Dr  | Hector| NY   | 14841  | 1000  | 1     | 0       | 0     |
| 389572 | Stevens| K L | 208 Snow Pass| Denver| CO   | 80226  | 400   | 1     | 58.75  | 1.50  |
| 846283 | Alison | J S | 787 Lake Dr   | Isle  | MN   | 56342  | 5000  | 3     | 10.00  | 0     |
| 475938 | Doe    | J W | 59 Archer Rd  | Sutter| CA   | 95685  | 700   | 2     | 250.00 | 100.00|
| 693829 | Thomas | A N | 3 Dove Circle| Casper| WY   | 82609  | 9999  | 2     | 0       | 0     |
| 593029 | Williams| E D | 485 SE 2 Ave  | Dallas| TX   | 75218  | 200   | 1     | 25.00  | 0     |
| 192837 | Lee    | F L | 5963 Oak St   | Hector| NY   | 14841  | 700   | 2     | 489.50 | .50   |
| 583990 | Abraham| M T | 392 Mill St   | Isle  | MN   | 56342  | 9999  | 3     | 500.00 | 0     |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
SQLRowCount returns -1
12 rows fetched
SQL> █
```

ODBC for all!!!

The ODBC Driver is available to connect PASE to the native environment for ANY application!



PyODBC: Application Adapter for Python

Step 3:

- Use IBM Open Source package Manager in ACS
 - Highlight
 - Click install
 - Type “y”
- Driver is strategic to python data access.
- Indirectly supported

python3	3.6.10-1	@ibm
python3-Pillow	5.0.0-5	@ibm
python3-asn1crypto	0.24.0-1	@ibm
python3-bcrypt	3.1.4-6	@ibm
python3-cffi	1.11.5-3	@ibm
python3-cryptography	2.8-0	@ibm
python3-dateutil	2.7.5-1	@ibm
python3-devel	3.6.10-1	@ibm
python3-ibm_db	2.0.5.12-0	@ibm
python3-idna	2.8-1	@ibm
python3-itoolkit	1.6.1-1	@ibm
python3-lxml	4.2.1-4	@ibm
python3-numpy	1.15.4-0	@ibm/\$releasever
python3-pandas	0.22.0-5	@ibm
python3-paramiko	2.6.0-1	@ibm
python3-pip	9.0.1-3	@ibm
python3-psutil	5.5.1-0	@ibm
python3-pyparser	2.19-2	@ibm
python3-pynacl	1.2.1-4	@ibm
python3-pyodbc	4.0.27-0	@ibm
python3-pytz	2018.5-3	@ibm
python3-pyzmq	17.1.2-0	@ibm
python3-rpm	4.13.0.1-18	@ibm
python3-scikit-learn	0.19.1-7	@ibm
python3-scipy	1.1.0-1	@ibm
python3-setuptools	36.0.1-3	@ibm
python3-six	1.10.0-1	@ibm
python3-tkinter	3.6.10-1	@ibm
python3-wheel	0.29.0-3	@ibm

Start slow...


```
import pyodbc

cnxn = pyodbc.connect('DSN=*LOCAL')
print(cnxn)
```

```
import pyodbc

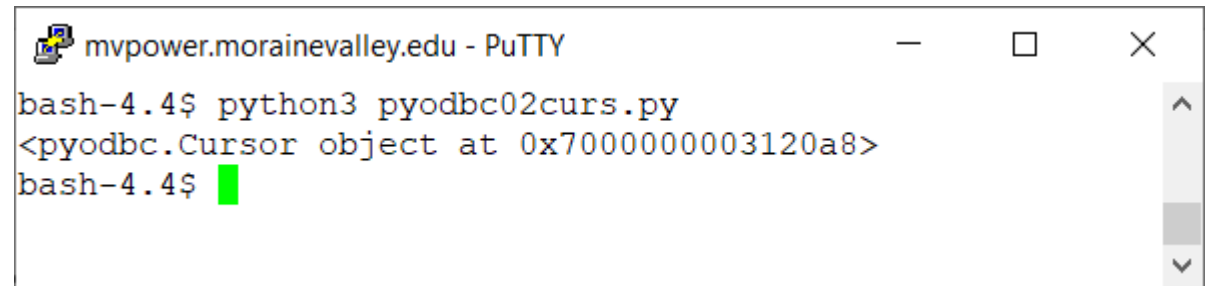
cnxn = pyodbc.connect('DSN=*LOCAL')

cursor = cnxn.cursor()
print(cursor)
```



A terminal window titled "mvpower.morainevalley.edu - PuTTY" showing the execution of a Python script. The prompt is "bash-4.4\$". The command "python3 pyodbc01conn.py" is entered, and the output is "<pyodbc.Connection object at 0x7000000002f81e0>". The prompt returns to "bash-4.4\$".

```
mvpower.morainevalley.edu - PuTTY
bash-4.4$ python3 pyodbc01conn.py
<pyodbc.Connection object at 0x7000000002f81e0>
bash-4.4$
```



A terminal window titled "mvpower.morainevalley.edu - PuTTY" showing the execution of a Python script. The prompt is "bash-4.4\$". The command "python3 pyodbc02curs.py" is entered, and the output is "<pyodbc.Cursor object at 0x7000000003120a8>". The prompt returns to "bash-4.4\$".

```
mvpower.morainevalley.edu - PuTTY
bash-4.4$ python3 pyodbc02curs.py
<pyodbc.Cursor object at 0x7000000003120a8>
bash-4.4$
```

PyODBC: Simple script

```
import pyodbc

cnxn = pyodbc.connect('DSN=*LOCAL')

cursor = cnxn.cursor()

#Sample select query
cursor.execute('''SELECT CUST_ID, COMPANY, FIRSTNAME, LASTNAME, COUNTRY
                from zendphp7.sp_cust''')

row = cursor.fetchone()
while row:
    print(row)
    row = cursor.fetchone()
```



```
mvpower.morainevalley.edu - PuTTY
bash-4.4$ python3 pyodbc03select.py
(Decimal('1221'), 'Kauai Dive Shoppe', 'LINA', 'Norman', 'US')
(Decimal('1231'), 'Unisco', 'George', 'Weathers', 'Bahamas')
(Decimal('1351'), 'Sight Diver', 'Phyllis', 'Spooner', 'Cyprus')
(Decimal('1354'), 'Cayman Divers World Unlimited', 'Joe', 'Bailey', 'British West Indies')
(Decimal('1356'), 'Tom Sawyer Diving Centre', 'Chris', 'Thomas', 'US Virgin Islands')
(Decimal('1380'), 'Blue Jack Aqua Center', 'Ernest', 'Barratt', 'US')
(Decimal('1384'), 'VIP Divers Club', 'Russell', 'Christopher', 'US Virgin Islands')
(Decimal('1510'), 'Ocean Paradise', 'Paul', 'Gardner', 'US')
(Decimal('1513'), 'Fantastique Aquatica', 'Susan', 'Wong', 'Columbia')
(Decimal('1551'), 'Marmot Divers Club', 'Joyce', 'Marsh', 'Canada')
(Decimal('1560'), 'The Depth Charge', 'Sam', 'Witherspoon', 'US')
(Decimal('1563'), 'Blue Sports', 'Theresa', 'Kunec', 'US')
(Decimal('1624'), 'Makai SCUBA Club', 'Donna', 'Siaus', 'US')
(Decimal('1645'), 'Action Club', 'Michael', 'Spurling', 'US')
(Decimal('1651'), 'Jamaica SCUBA Centre', 'Barbara', 'Harvey', 'West Indies')
(Decimal('1680'), 'Island Finders', 'Desmond', 'Ortega', 'US')
(Decimal('1984'), 'Adventure Undersea', 'Gloria', 'Gonzales', 'Belize')
(Decimal('2118'), 'Blue Sports Club', 'Harry', 'Bathbone', 'US')
(Decimal('2135'), 'Frank's Divers Supply', 'Lloyd', 'Fellows', 'US')
(Decimal('2156'), 'Davy Jones' Locker', 'Tanya', 'Wagner', 'Canada')
(Decimal('2163'), 'SCUBA Heaven', 'Robert', 'Michelind', 'Bahamas')
(Decimal('2165'), 'Shangri-La Sports Center', 'Frank', 'Paniagua', 'Bahamas')
(Decimal('2315'), 'Divers of Corfu, Inc.', 'Charles', 'Lopez', 'Greece')
```

For DML type transactions

```
import pyodbc

cnxn = pyodbc.connect('DSN=*LOCAL;CommitMode=0') #Turn off commitment control!

cursor = cnxn.cursor()

#Sample select query
cursor.execute("""update zendphp7.sp_cust
    set FIRSTNAME = 'Mike' where CUST_ID = 9841""")
```

Before:

CUST_ID	COMPANY	FIRSTNAME	LASTNAME	CIVIL	ADDRESS
9841	Neptune's Trident Supply	MIKE	Franks	2	PO Box 129

After:

CUST_ID	COMPANY	FIRSTNAME	LASTNAME	CIVIL	ADDRESS
9841	Neptune's Trident Supply	Mike	Franks	2	PO Box 129

IBM ODBC Documentation

https://www.ibm.com/support/knowledgecenter/ssw_ibm_i_74/rzaik/connectkeywords.htm

Table 2. IBM i Access ODBC connection string keywords for Server properties

Keyword	Description
Connection String: CMT ODBC.INI: CommitMode	Specifies the default transaction isolation level. Possible values: <ul style="list-style-type: none">• 0 = Commit immediate (*NONE)• 1 = Read committed (*CS)• 2 = Read uncommitted (*CHG)• 3 = Repeatable read (*ALL)• 4 = Serializable (*RR) Default: 2

SQLite

What is SQLite?



SQLite is an in-process library that implements a [self-contained](#), [serverless](#), [zero-configuration](#), [transactional](#) SQL database engine.

It is persistent, like any other file in the IFS

Platform agnostic (typically embedded)

- Since it's a file, it can reside anywhere
- Any system with a C compiler can manipulate it

Think of it as the **QTEMP** of the open source world

Was born in 2000 but current plans are mapped through 2050

Brief history

Created by D. Richard Hipp while at GD working on software for guided missile destroyers for the US Navy

- Replaced HP-UX based Informix

Goal to allow database access without server or admin

Syntax based on PostgreSQL 6.5



Who's using it?

Most browsers and operating systems

Web application frameworks

Most applications that need an embedded database like

- Skype
- Evernote
- Zend Server (Statistics and event info)
- Millions more...

How it works

EASY PEASY



SQLite3 Installer

Open Source Package Management

File View Connection Utilities

Connection: mpavlak@mvpower.morainevalley.edu:/

Installed packages	Updates available	Available packages																																																																																				
		<table border="1"><thead><tr><th>Package</th><th>Version</th><th>Repository</th></tr></thead><tbody><tr><td>popt-devel</td><td>1.16-1</td><td>ibm</td></tr><tr><td>python2</td><td>2.7.15-1</td><td>ibm</td></tr><tr><td>python2-devel</td><td>2.7.15-1</td><td>ibm</td></tr><tr><td>python2-iniparse</td><td>0.4-0</td><td>ibm</td></tr><tr><td>python2-itoolkkit</td><td>1.4.0-1</td><td>ibm</td></tr><tr><td>python2-pip</td><td>9.0.1-0</td><td>ibm</td></tr><tr><td>python2-setuptools</td><td>36.0.1-0</td><td>ibm</td></tr><tr><td>python2-wheel</td><td>0.29.0-0</td><td>ibm</td></tr><tr><td>readline-devel</td><td>6.3-2</td><td>ibm</td></tr><tr><td>readline-doc</td><td>6.3-2</td><td>ibm</td></tr><tr><td>rpm-build</td><td>4.13.0.1-10</td><td>ibm</td></tr><tr><td>rpm-devel</td><td>4.13.0.1-10</td><td>ibm</td></tr><tr><td>rsync</td><td>3.1.2-1</td><td>ibm</td></tr><tr><td>sed-gnu</td><td>4.4-0</td><td>ibm</td></tr><tr><td>slang</td><td>2.3.1a-0</td><td>ibm</td></tr><tr><td>slang-devel</td><td>2.3.1a-0</td><td>ibm</td></tr><tr><td>sqlite3</td><td>3.19.3-0</td><td>ibm</td></tr><tr><td>sqlite3-devel</td><td>3.19.3-0</td><td>ibm</td></tr><tr><td>tar-gnu</td><td>1.29-0</td><td>ibm</td></tr><tr><td>unixODBC</td><td>2.3.4-1</td><td>ibm</td></tr><tr><td>unixODBC-devel</td><td>2.3.4-1</td><td>ibm</td></tr><tr><td>unzip</td><td>6.0-0</td><td>ibm</td></tr><tr><td>wget</td><td>1.17.1-1</td><td>ibm</td></tr><tr><td>xz</td><td>5.2.3-0</td><td>ibm</td></tr><tr><td>xz-devel</td><td>5.2.3-0</td><td>ibm</td></tr><tr><td>zip</td><td>3.0-0</td><td>ibm</td></tr><tr><td>zlib-devel</td><td>1.2.11-1</td><td>ibm</td></tr></tbody></table>	Package	Version	Repository	popt-devel	1.16-1	ibm	python2	2.7.15-1	ibm	python2-devel	2.7.15-1	ibm	python2-iniparse	0.4-0	ibm	python2-itoolkkit	1.4.0-1	ibm	python2-pip	9.0.1-0	ibm	python2-setuptools	36.0.1-0	ibm	python2-wheel	0.29.0-0	ibm	readline-devel	6.3-2	ibm	readline-doc	6.3-2	ibm	rpm-build	4.13.0.1-10	ibm	rpm-devel	4.13.0.1-10	ibm	rsync	3.1.2-1	ibm	sed-gnu	4.4-0	ibm	slang	2.3.1a-0	ibm	slang-devel	2.3.1a-0	ibm	sqlite3	3.19.3-0	ibm	sqlite3-devel	3.19.3-0	ibm	tar-gnu	1.29-0	ibm	unixODBC	2.3.4-1	ibm	unixODBC-devel	2.3.4-1	ibm	unzip	6.0-0	ibm	wget	1.17.1-1	ibm	xz	5.2.3-0	ibm	xz-devel	5.2.3-0	ibm	zip	3.0-0	ibm	zlib-devel	1.2.11-1	ibm
Package	Version	Repository																																																																																				
popt-devel	1.16-1	ibm																																																																																				
python2	2.7.15-1	ibm																																																																																				
python2-devel	2.7.15-1	ibm																																																																																				
python2-iniparse	0.4-0	ibm																																																																																				
python2-itoolkkit	1.4.0-1	ibm																																																																																				
python2-pip	9.0.1-0	ibm																																																																																				
python2-setuptools	36.0.1-0	ibm																																																																																				
python2-wheel	0.29.0-0	ibm																																																																																				
readline-devel	6.3-2	ibm																																																																																				
readline-doc	6.3-2	ibm																																																																																				
rpm-build	4.13.0.1-10	ibm																																																																																				
rpm-devel	4.13.0.1-10	ibm																																																																																				
rsync	3.1.2-1	ibm																																																																																				
sed-gnu	4.4-0	ibm																																																																																				
slang	2.3.1a-0	ibm																																																																																				
slang-devel	2.3.1a-0	ibm																																																																																				
sqlite3	3.19.3-0	ibm																																																																																				
sqlite3-devel	3.19.3-0	ibm																																																																																				
tar-gnu	1.29-0	ibm																																																																																				
unixODBC	2.3.4-1	ibm																																																																																				
unixODBC-devel	2.3.4-1	ibm																																																																																				
unzip	6.0-0	ibm																																																																																				
wget	1.17.1-1	ibm																																																																																				
xz	5.2.3-0	ibm																																																																																				
xz-devel	5.2.3-0	ibm																																																																																				
zip	3.0-0	ibm																																																																																				
zlib-devel	1.2.11-1	ibm																																																																																				

Done: 134 rows retrieved.

Information Install

Package Installation

```
clear;exec /00openSys/pkg/bin/yum install 'sqlite3' 'sqlite3-devel'
$ Setting up Install Process
Resolving Dependencies
--> Running transaction check
----> Package sqlite3.ppc64 0:3.19.3-0 will be installed
----> Package sqlite3-devel.ppc64 0:3.19.3-0 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package                Arch      Version      Repository    Size
=====
Installing:
sqlite3                ppc64    3.19.3-0     ibm            1.3 M
sqlite3-devel          ppc64    3.19.3-0     ibm            138 k
=====
Transaction Summary
-----
Install      2 Packages

Total download size: 1.4 M
Installed size: 5.5 M
Is this ok [y/N]: Y
```

Package Installation

```
----> Package sqlite3.ppc64 0:3.19.3-0 will be installed
----> Package sqlite3-devel.ppc64 0:3.19.3-0 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package                Arch      Version      Repository    Size
=====
Installing:
sqlite3                ppc64    3.19.3-0     ibm            1.3 M
sqlite3-devel          ppc64    3.19.3-0     ibm            138 k
=====
Transaction Summary
-----
Install      2 Packages

Total download size: 1.4 M
Installed size: 5.5 M
Is this ok [y/N]: Y
Downloading Packages:
(1/2): sqlite3-3.19.3-0.ibm17.1.ppc64.rpm | 1.3 MB 00:00
(2/2): sqlite3-devel-3.19.3-0.ibm17.1.ppc64.rpm | 138 kB 00:00
-----
Total                                     566 kB/s | 1.4 MB 00:02
Running Transaction Check
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : sqlite3-devel-3.19.3-0.ppc64                1/2
  Installing : sqlite3-3.19.3-0.ppc64                    2/2

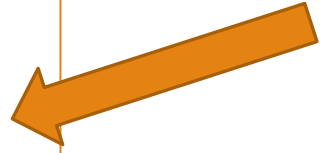
Installed:
  sqlite3.ppc64 0:3.19.3-0                sqlite3-devel.ppc64 0:3.19.3-0

Complete!
```

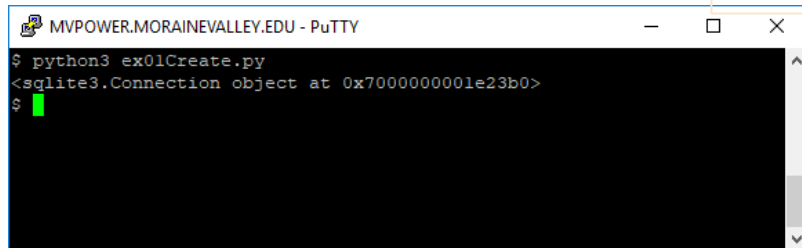
Creating a database

```
#Create SQLite DB
```

```
import sqlite3  
myconnection = sqlite3.connect('Mikey1.db')  
  
print(myconnection)
```





Before:



```
MVPOWER.MORAINESVALLEY.EDU - PuTTY  
$ python3 ex01Create.py  
<sqlite3.Connection object at 0x7000000001e23b0>  
$
```

After:

Icon	Name	Size (KB)	Last Modified
	ex01Create.py		1 10 July, 2018 4:33:15 PM CDT

Icon	Name	Size (KB)	Last Modified
	ex01Create.py		1 10 July, 2018 4:38:33 PM CDT
	Mikey1.db		1 10 July, 2018 4:38:53 PM CDT

Let's add a table and a row...

```
#Create SQLite DB...
```

```
import sqlite3  
myconnection = sqlite3.connect('Mikey1.db')
```

```
#Create a cursor...
```

```
c = myconnection.cursor()
```

```
#Create a table
```

```
c.execute(''' CREATE TABLE IF NOT EXISTS beers  
          (Beer text, Brewery text, Country text, qty int, price real)  
          ''')
```

```
#Create a row...
```

```
c.execute("INSERT INTO beers VALUES ('Boston Lager', 'Sam  
Adams', 'USA', 100, 35.14)")
```

```
myconnection.commit()
```

```
myconnection.close()
```

How can we see this?

Tools for SQLite

SQL (Based on PostgreSQL)

Special . (dot) commands

- Type .help at any time

3. Special commands to sqlite3 (dot-commands)

Most of the time, sqlite3 just reads lines of input and passes them on to the SQLite library for "commands" are typically used to change the output format of queries, or to execute certain

For a listing of the available dot commands, you can enter ".help" at any time. For example:

```
sqlite> .help
.archive ...           Manage SQL archives: ".archive --help" for details
.auth ON|OFF          Show authorizer callbacks
.backup ?DB? FILE     Backup DB (default "main") to FILE
                      Add "--append" to open using appendvfs.
.bail on|off          Stop after hitting an error.  Default OFF
.binary on|off        Turn binary output on or off.  Default OFF
.cd DIRECTORY         Change the working directory to DIRECTORY
.changes on|off       Show number of rows changed by SQL
.check GLOB           Fail if output since .testcase does not match
.clone NEWDB          Clone data into NEWDB from the existing database
.databases            List names and files of attached databases
.dbconfig ?op? ?val? List or change sqlite3_db_config() options
.dbinfo ?DB?         Show status information about the database
.dump ?TABLE? ...    Dump the database in an SQL text format
                      If TABLE specified, only dump tables matching
                      LIKE pattern TABLE.
.echo on|off          Turn command echo on or off
.eqp on|off|full      Enable or disable automatic EXPLAIN QUERY PLAN
.excel                Display the output of next command in a spreadsheet
.exit                Exit this program
.expert              EXPERIMENTAL. Suggest indexes for specified queries
.fullschema ?--indent? Show schema and the content of sqlite_stat tables
.headers on|off       Turn display of headers on or off
.help                Show this message
.import FILE TABLE   Import data from FILE into TABLE
```

How about adding a few rows at a time?

```
import sqlite3
myconnection = sqlite3.connect('Mikey1.db')

#Create a cursor...
c = myconnection.cursor()

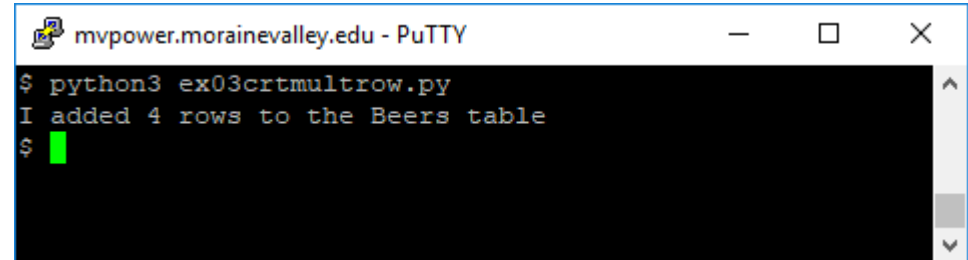
#Create a few rows...
beers = [('Honkers Ale', 'Goose Island', 'USA', 96, 17.85),
         ('Winter Welcome', 'Samuel Smith', 'UK', 150, 35.14),
         ('Spotted Cow', 'New Glarus', 'USA', 100, 35.14),
         ('Zombie Dust', 'Three Floyds', 'USA', 100, 35.14)]

c.executemany('INSERT INTO beers VALUES (?, ?, ?, ?, ?)', beers)

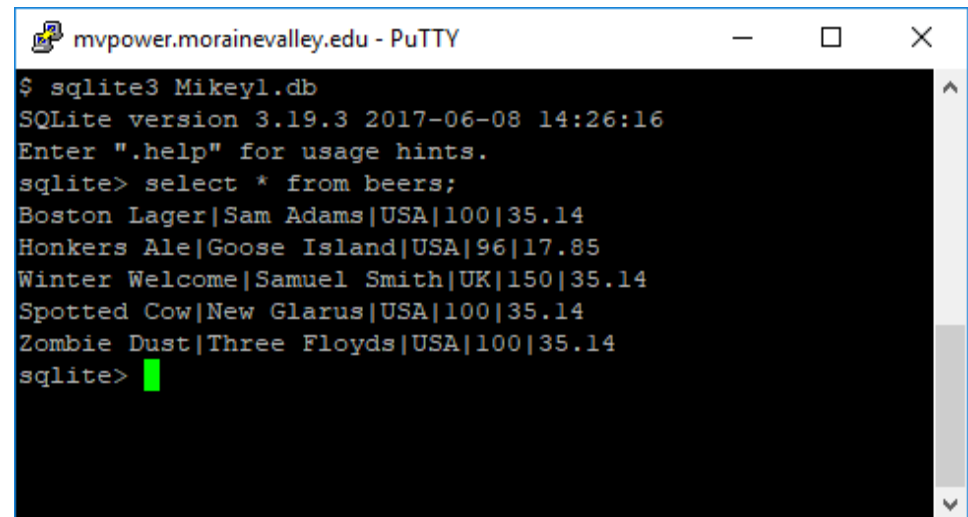
print("I added {} rows to the Beers table".format(c.rowcount))

myconnection.commit()

myconnection.close()
```



```
mvpower.morainevalley.edu - PuTTY
$ python3 ex03crtmultrow.py
I added 4 rows to the Beers table
$
```



```
mvpower.morainevalley.edu - PuTTY
$ sqlite3 Mikey1.db
SQLite version 3.19.3 2017-06-08 14:26:16
Enter ".help" for usage hints.
sqlite> select * from beers;
Boston Lager|Sam Adams|USA|100|35.14
Honkers Ale|Goose Island|USA|96|17.85
Winter Welcome|Samuel Smith|UK|150|35.14
Spotted Cow|New Glarus|USA|100|35.14
Zombie Dust|Three Floyds|USA|100|35.14
sqlite>
```

Read from SQLite

```
import sqlite3
myconnection = sqlite3.connect('Mikey1.db')

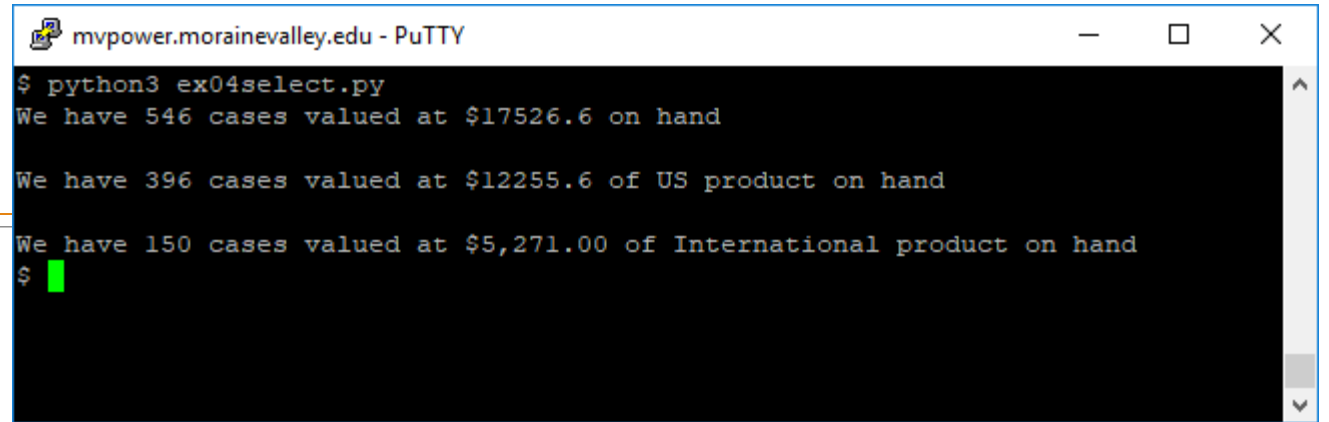
#Create a cursor...
c = myconnection.cursor()

#Build the SQL Statement
sqlStatement = 'SELECT SUM(qty), sum(qty*price) from beers'
c.execute(sqlStatement)
result = c.fetchone()
print("We have {} cases valued at ${} on hand".format(result[0], result[1]))

#US Inventory
sqlStatement2 = sqlStatement + " where country = 'USA'"
c.execute(sqlStatement2)
result = c.fetchone()
print("\nWe have {} cases valued at ${} of US product on hand".format(result[0], result[1]))

#International Inventory
sqlStatement2 = sqlStatement + " where country <> 'USA'"
c.execute(sqlStatement2)
result = c.fetchone()
print("\nWe have {} cases valued at {} of International product on hand".format(result[0],
    '{:,.2f}'.format(result[1])))

myconnection.close()
```



```
mvpower.morainevalley.edu - PuTTY
$ python3 ex04select.py
We have 546 cases valued at $17526.6 on hand

We have 396 cases valued at $12255.6 of US product on hand

We have 150 cases valued at $5,271.00 of International product on hand
$
```

Performance implications

Db2 & MariaDB for enterprise

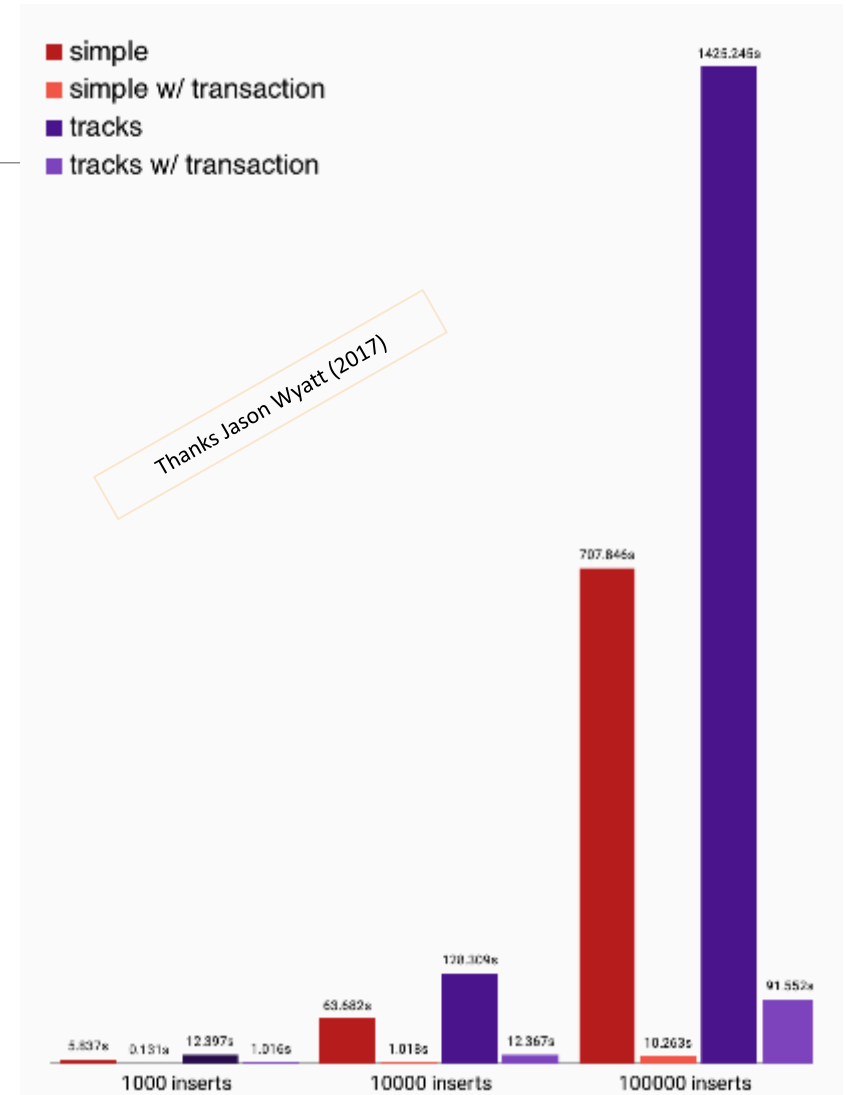
- Optimizer, multi-tenant

SQLite is great for smaller projects

- Methods exist to optimize
- Single tenant

Both have MANY advanced features

- Atomicity
- Transactions
- Persistence
- RI, functions, etc.



Use cases (wrapup)

Internal scratch pad

Not to be used to replace Db2!!!

Great for open source

Data transport (ETL?)

Performs well



Postgresql



What is it and why should I care?

Possibly the most popular Open Source database after MySQL

Key features:

- Multi-user
- ACID compliant
- Scalable

Why use it?

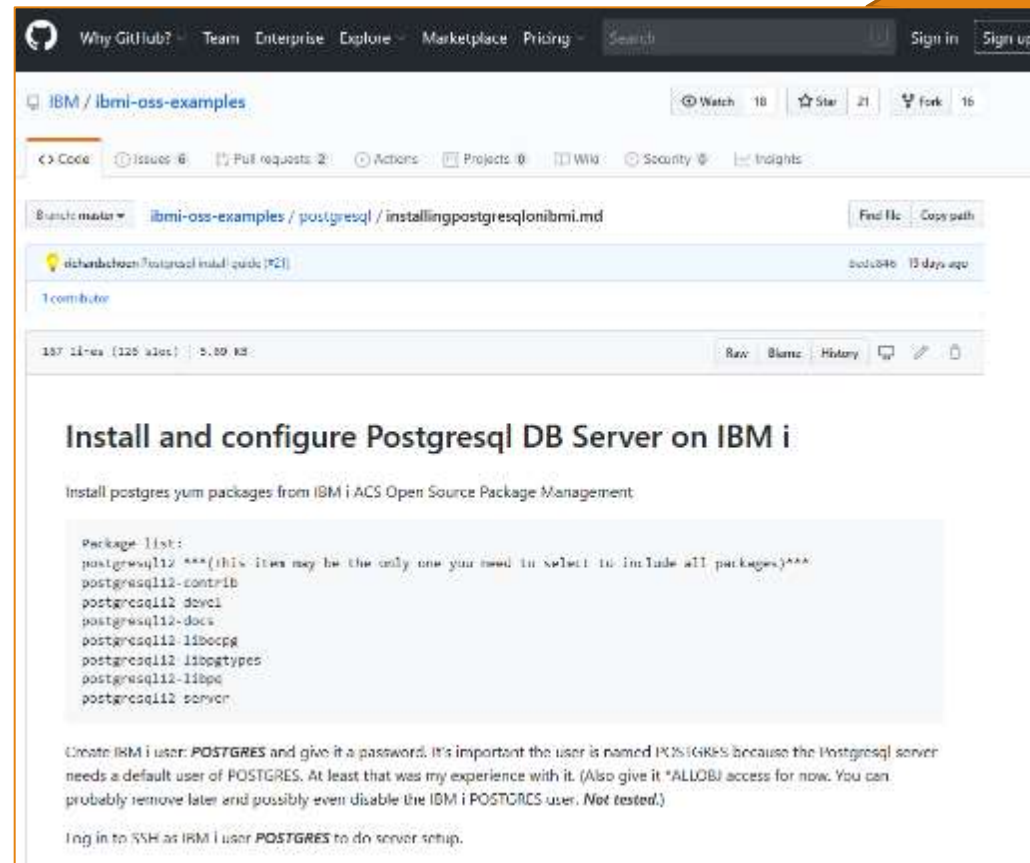
- Some open source applications are designed for it.
- Object relations vs. MySQL, etc.
- Full support for nearly every known data type
 - Network Address storage
 - Multidimensional array
 - Geometric data
 - Schema-less JSON support
 - Create your own
 - Max sizes are HUGE
 - Data Integrity



GIT page on Postgres Install

Thanks Richard Schoen!!!

Read the directions through once before starting



The screenshot shows a GitHub repository page for 'IBM / ibmi-oss-examples'. The file 'installingpostgreslonibmi.md' is selected, showing a commit by 'richenschoen' from 'postgresl-install-guide' on Oct 2, 2016. The file content includes a package list and instructions for creating a user and logging in via SSH.

```
Package List:
postgres112 *** (this item may be the only one you need to select to include all packages)***
postgres112-contrib
postgres112-devel
postgres112-docs
postgres112-libpq
postgres112-libpqtypes
postgres112-libpq
postgres112-server
```

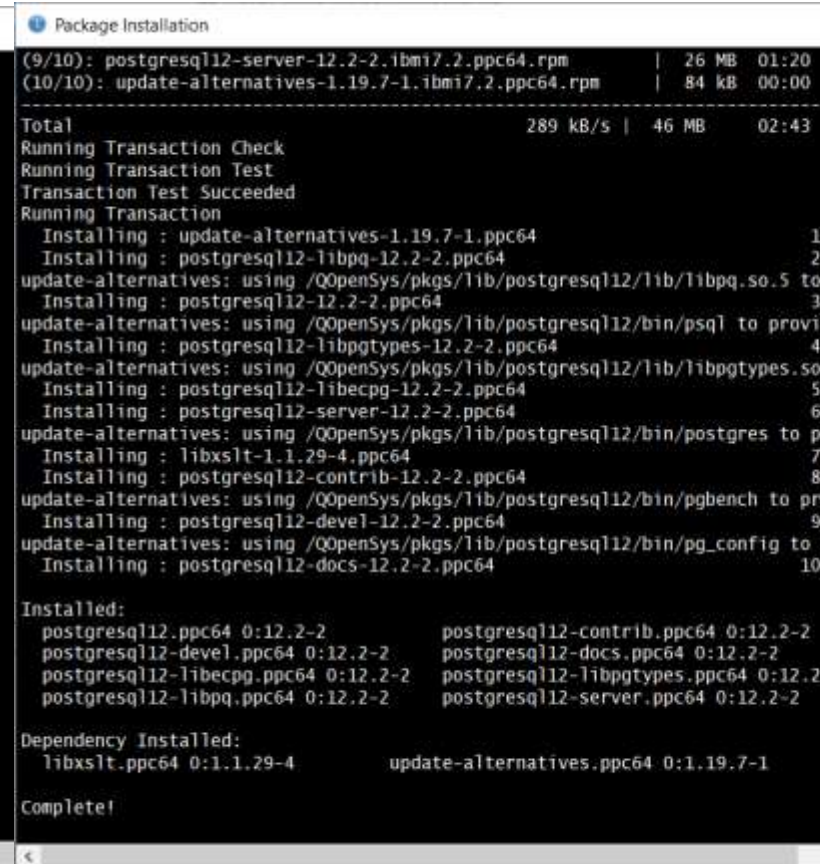
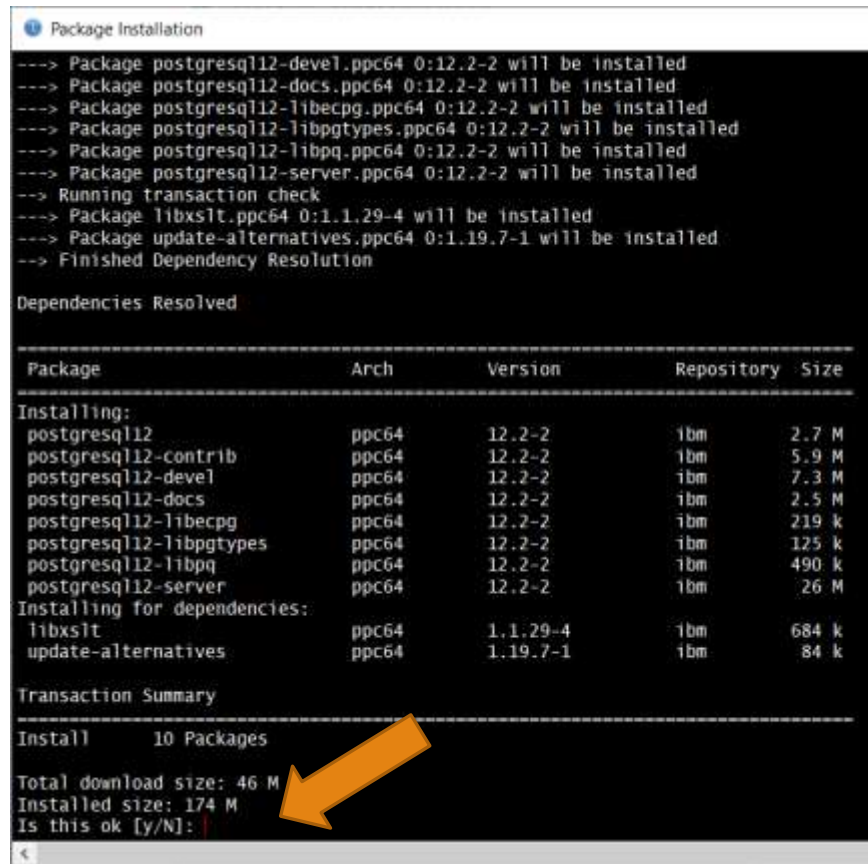
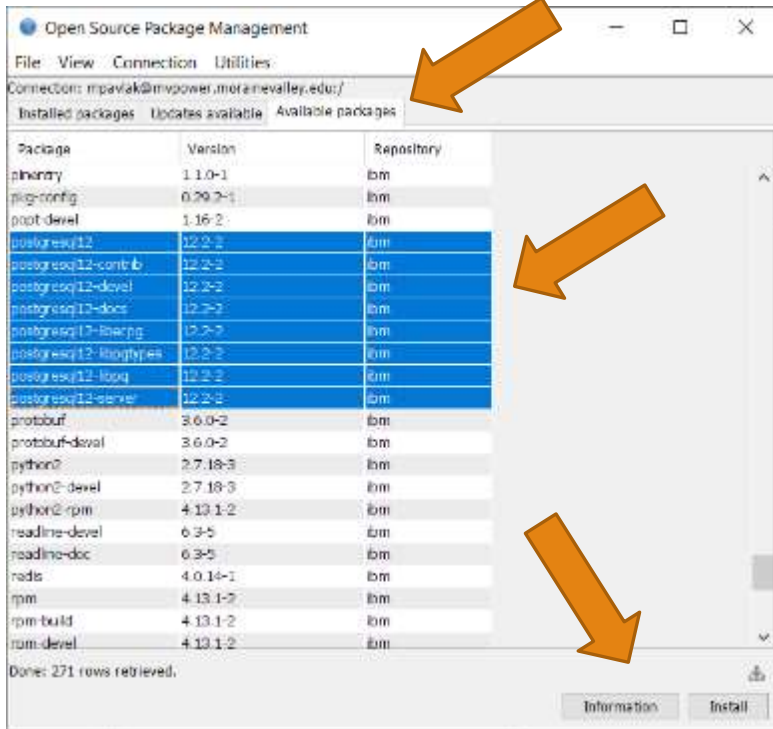
Create IBM i user: **POSTGRES** and give it a password. It's important the user is named POSTGRES because the Postgresql server needs a default user of POSTGRES. At least that was my experience with it. (Also give it *ALLOBJ access for now. You can probably remove later and possibly even disable the IBM i POSTGRES user. *Not tested.*)

Log in to SSH as IBM i user **POSTGRES** to do server setup.

<https://github.com/IBM/ibmi-oss-examples/blob/master/postgresql/installingpostgreslonibmi.md>

AKA: Postgres

Install via ACS menus



Step by step...

Create a user profile for POSTGRES, add a password and *ALLOBJ for now

Open SSH shell, good to Bash

Create a directory off the root: /postgres

Export the path: export PGDATA=/postgres

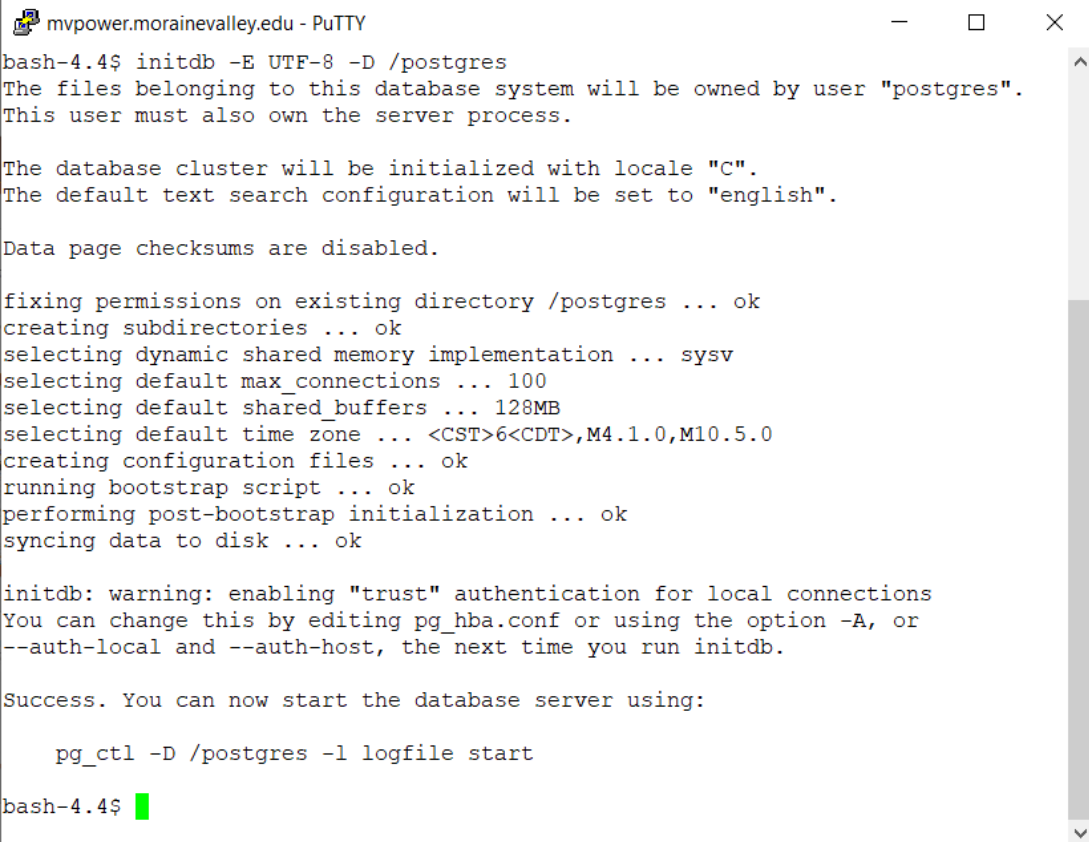
Create the cluster: initdb -E UTF-8 -D /postgres

Update postgres.conf

- IP & Port

Start it up: pg_ctl -D /postgres -l logfile start

```
bash-4.4$ pg_ctl -D /postgres -l logfile start
waiting for server to start.... done
server started
bash-4.4$ █
```



```
mvpower.morainevalley.edu - PuTTY
bash-4.4$ initdb -E UTF-8 -D /postgres
The files belonging to this database system will be owned by user "postgres".
This user must also own the server process.

The database cluster will be initialized with locale "C".
The default text search configuration will be set to "english".

Data page checksums are disabled.

fixing permissions on existing directory /postgres ... ok
creating subdirectories ... ok
selecting dynamic shared memory implementation ... sysv
selecting default max_connections ... 100
selecting default shared_buffers ... 128MB
selecting default time zone ... <CST>6<CDT>,M4.1.0,M10.5.0
creating configuration files ... ok
running bootstrap script ... ok
performing post-bootstrap initialization ... ok
syncing data to disk ... ok

initdb: warning: enabling "trust" authentication for local connections
You can change this by editing pg_hba.conf or using the option -A, or
--auth-local and --auth-host, the next time you run initdb.

Success. You can now start the database server using:

    pg_ctl -D /postgres -l logfile start

bash-4.4$ █
```

See jobs

From Native:

- QUSRWRK

```
____ QPOZSPWP  POSTGRES  BCI      .0  PGM-postgres  SELW
____ QPOZSPWP  POSTGRES  BCI      .0  PGM-postgres  SELW
____ QPOZSPWP  POSTGRES  BCI      .0  PGM-postgres  SELW
____ QPOZSPWP  POSTGRES  BCI      .0  PGM-postgres  SELW
____ QPOZSPWP  POSTGRES  BCI      .0  PGM-postgres  SELW
____ QPOZSPWP  POSTGRES  BCI      .0  PGM-postgres  SELW
____ QPOZSPWP  POSTGRES  BCI      .0  PGM-postgres  SELW
```

From PASE:

- ps ax

```
mvpower.morainevalley.edu - PuTTY
bash-4.4$ ps ax
  PID  TTY  STAT  TIME  COMMAND
   10   -   A     1:16 /QOpenSys/QIBM/ProdData/JavaVM/jdk80/32bit/jre/lib/pp
   27   -   A     3:27 [cimserve]
   34   -   A     1:24 /QOpenSys/QIBM/ProdData/JavaVM/jdk80/32bit/jre/lib/pp
   65   -   A     3:18 /QIBM/ProdData/OS/SLP/bin/lslp-kernel -f/QIBM/UserDat
   77   -   Z     0:00 <defunct>
   81   -   A     3:59 /QOpenSys/QIBM/ProdData/JavaVM/jdk80/64bit/jre/lib/pp
  110   -   A     6:43 /QOpenSys/QIBM/ProdData/SC1/OpenSSH/sbin/sshd
495609 -   A     0:00 /QOpenSys/QIBM/ProdData/SC1/OpenSSH/sbin/sshd -R
495617 pts/1 A     0:00 -bsh
495618 pts/1 A     0:00 bash
495819 -   A     0:00 /QOpenSys/QIBM/ProdData/SC1/OpenSSH/sbin/sshd -R
496420 -   A     0:00 /QOpenSys/pkgs/lib/postgresql12/bin/postgres -D /post
496422 -   A     0:00 postgres: checkpointer
496423 -   A     0:00 postgres: background writer
496424 -   A     0:00 postgres: walwriter
496425 -   A     0:00 postgres: autovacuum launcher
496426 -   A     0:00 postgres: stats collector
496427 -   A     0:00 postgres: logical replication launcher
496442 -   A     0:00 /QOpenSys/QIBM/ProdData/SC1/OpenSSH/sbin/sshd -R
496466 pts/1 A     0:00 ps ax
bash-4.4$
```

Miscellaneous

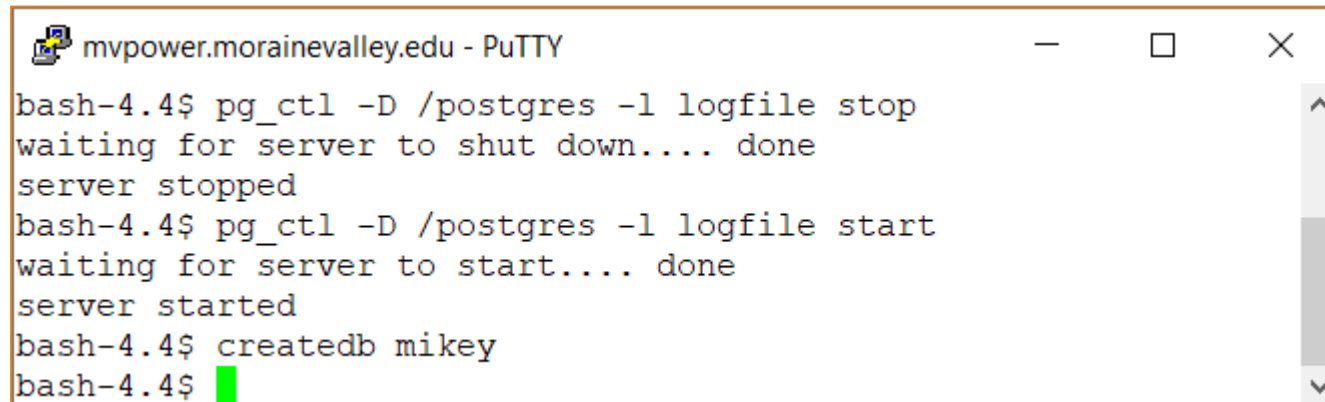
Stop the server: **pg_ctl -D /postgres -l logfile stop**

Open the server for remote access

- Edit **pg_hba.conf**
- Add: **host all all 0.0.0.0/0 password**
- Restart the server

Create a database

- At the shell type: **createdb <dbname>** i.e.: **createdb mikey**

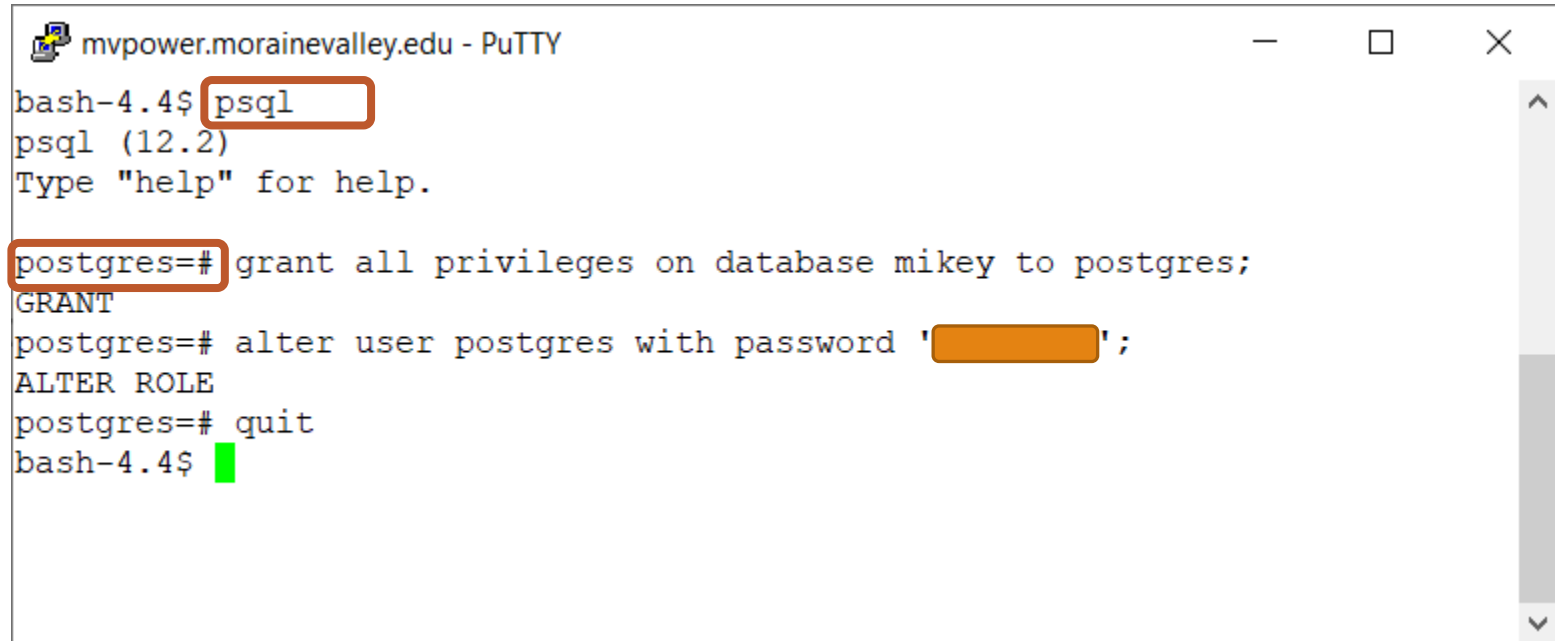


```
mvpower.morainevalley.edu - PuTTY
bash-4.4$ pg_ctl -D /postgres -l logfile stop
waiting for server to shut down.... done
server stopped
bash-4.4$ pg_ctl -D /postgres -l logfile start
waiting for server to start.... done
server started
bash-4.4$ createdb mikey
bash-4.4$ █
```

Psql utility

Bring up the Postgres command line for SQL processing

- Set auth to database
- Change default user password
- Exit the prompt



```
mvpower.morainevalley.edu - PuTTY
bash-4.4$ psql
psql (12.2)
Type "help" for help.
postgres=# grant all privileges on database mikey to postgres;
GRANT
postgres=# alter user postgres with password 'XXXXXXXXXX';
ALTER ROLE
postgres=# quit
bash-4.4$
```

pgAdmin – Like DBU for Postgres!

The screenshot displays the pgAdmin web interface. On the left, a tree view shows the server hierarchy: Servers (1) > MVCC > Databases (2) > mikey. An orange arrow labeled "Remote Connect" points to the "mikey" database. The main dashboard features several performance graphs: "Server sessions" (a flat line at 1.0), "Transactions per second" (a line graph showing spikes in Transactions, Commits, and Rollbacks), "Tuples in" (a line graph showing Inserts, Updates, and Deletes), "Tuples out" (a line graph showing Fetched and Returned tuples), and "Block I/O" (a line graph showing Reads and Hits). Below the graphs is the "Server activity" section, which includes tabs for Sessions, Locks, Prepared Transactions, and Configuration. The "Sessions" tab is active, displaying a table of database sessions.

	PID	Database	User	Application	Client	Backend start	State	Wait event	Blocking PIDs
●	497854					2020-06-21 19:37:46 CDT		Activity: CheckpointerMain	
●	497855					2020-06-21 19:37:46 CDT		Activity: BgWriterHibernate	
●	497856					2020-06-21 19:37:46 CDT		Activity: WalWriterMain	
●	497857					2020-06-21 19:37:46 CDT		Activity: AutoVacuumMain	
●	497859		postgres			2020-06-21 19:37:46 CDT		Activity: LogicalLauncherMain	
●	498352	postgres	postgres	pgAdmin 4 - DB:postgres	24.14.211.110	2020-06-21 20:07:43 CDT	active		
●	498584	mikey	postgres	pgAdmin 4 - DB:mikey	24.14.211.110	2020-06-21 20:20:45 CDT	Idle	Client: ClientRead	

SQL in the browser

The screenshot shows the pgAdmin 4 web interface in a browser. The address bar shows the URL `127.0.0.1:54220/browser/#`. The interface includes a navigation menu on the left, a central query editor, and a data output table at the bottom.

Query Editor:

```
1 select + from sales.customers;
```

Data Output Table:

	Address	City	id	name	state	zip	email
	character (20)	character (20)	[PK] integer	character (30)	character (2)	character (9)	character (50)
1	1313 Mockingbird Ln	Los Angeles		1 Herman Munster	CA	90210	herman@munster.c...
2	100 Buena Vista Blvd	Anaheim		2 Mickey Mouse	CA	90285	mickey@disney.com...
3	1 Infinite Loop	Cupertino		3 Tim Cook	CA	90129	timmy@apple.com ...



Python access from IBM i PASE

Pip install psycopg2

```
mvpower.morainevalley.edu - PuTTY
bash-4.4$ pip3 install psycopg2
Collecting psycopg2
  Downloading https://files.pythonhosted.org/packages/a8/8f/1c5690eebf148d1d1554
fc00ccf9101e134636553dbb75bdfef4f85d7647/psycopg2-2.8.5.tar.gz (380kB)
    100% |#####| 389kB 1.4MB/s
Building wheels for collected packages: psycopg2
  Running setup.py bdist_wheel for psycopg2 ... done
  Stored in directory: /home/mpavlak/.cache/pip/wheels/fb/85/a8/57f24b92b9554880
384d00a84881c0ea80cbcee02d6dcede54
Successfully built psycopg2
Installing collected packages: psycopg2
Successfully installed psycopg2-2.8.5
bash-4.4$
```

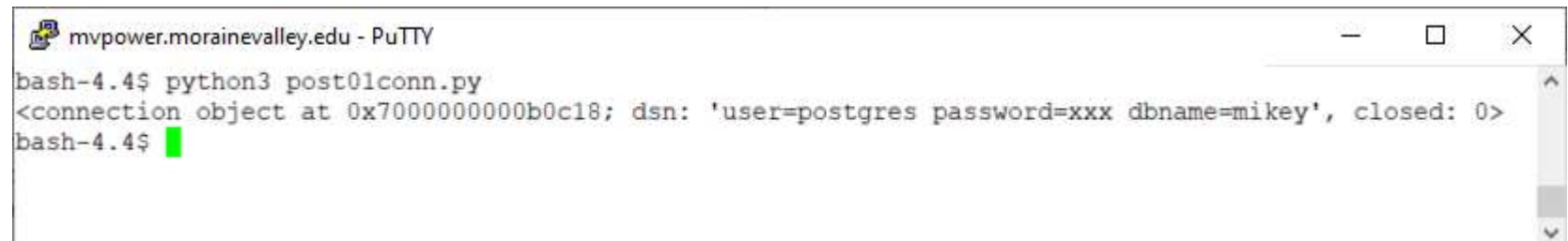
Psycopg is the most popular [PostgreSQL](#) adapter for the [Python programming language](#). Its core is a complete implementation of the [Python DB API 2.0](#) specifications. Several extensions allow access to many of the features offered by PostgreSQL.

Connection

```
import psycopg2

conn = psycopg2.connect("dbname=mikey user=postgres password=#####")

print(conn)
```



```
mvpower.morainevalley.edu - PuTTY
bash-4.4$ python3 post01conn.py
<connection object at 0x7000000000b0c18; dsn: 'user=postgres password=xxx dbname=mikey', closed: 0>
bash-4.4$ █
```

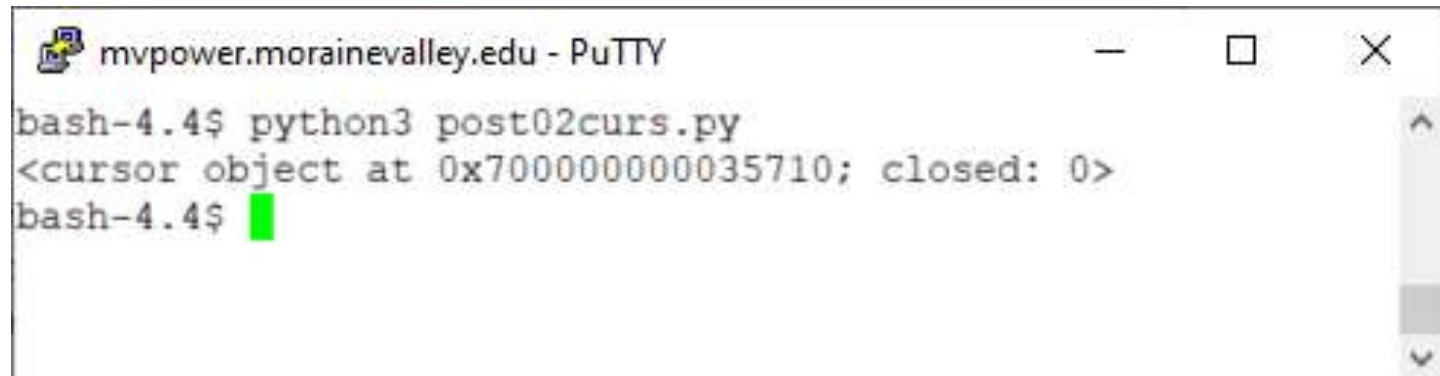
Cursor

```
import psycopg2

conn = psycopg2.connect("dbname=mikey user=postgres password=#####")

cur = conn.cursor()

print(cur)
```



The screenshot shows a PuTTY terminal window titled "mvpower.morainevalley.edu - PuTTY". The terminal output is as follows:

```
bash-4.4$ python3 post02curs.py
<cursor object at 0x700000000035710; closed: 0>
bash-4.4$ █
```

The terminal window includes standard window controls (minimize, maximize, close) and a vertical scrollbar on the right side.

Rows

```
import psycopg2

conn = psycopg2.connect("dbname=mikey user=postgres password=#####")

cur = conn.cursor()

cur.execute("""SELECT id, name, "Address", "City" , state, zip, email
              from sales.customers""")

rows = cur.fetchall()

print("\nList of Customers:\n")
for row in rows:
    print("    ", row)
```



mvpower.morainevalley.edu - PuTTY

```
bash-4.4$ python3 post03select.py

List of Customers:

      (1, 'Herman Munster', 'Los Angeles', 'herman@munster.com')
      (2, 'Mickey Mouse', 'Anaheim', 'mickey@disney.com')
      (3, 'Tim Cook', 'Cupertino', 'timmy@apple.com')

bash-4.4$
```

Summary

Many options for data

Python will invite this as a ubiquitous solution for nearly any problem

More surprises

Python Data Access

Thank You



MikePavlak@gmail.com



@MikeyPEI

