ΠΡΟΑΠΑΙΤΟΥΜΕΝΑ:

1. Σειρές: Python, SQL
2. Python Advanced: Μαθήματα 1-3 (PIP, Iterators, Generators)
3. Python Modules: Μάθημα 1.3 - Decimal

ΠΕΡΙΕΧΟΜΕΝΑ:

1. Σύνδεση με MySQL
2. Εκτέλεση Ερωτημάτων
   1. Result set ως λεξικό
   2. fetchone() και fetchmany()

Ευάγγελος Μ.

Σμαραγδένιος Χορηγός Μαθήματος

Βασίλης Σ.

Ασημένιος Χορηγός Μαθήματος

ΜΑΘΗΜΑ 1: Σύνδεση - Εκτέλεση Ερωτημάτων

1. Σύνδεση με MySQL

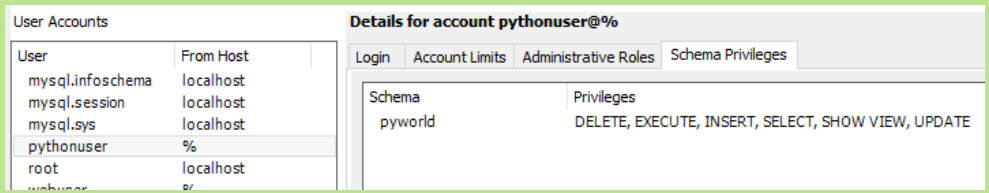
MySQL psounis

- Το πακέτο που απαιτείται για να συνδεθούμε με μία MySQL βάση δεδομένων είναι το **mysql-connector-python**
- Το εγκαθιστούμε με το PIP:

```
> pip install package mysql-connector-python
```

Προετοιμασία Βάσης Δεδομένων:

- Δημιουργούμε νέα βάση δεδομένων τρέχοντας το script py_world.sql στην MySQL (βλ.βίντεο)
  - (αποτελεί αντίγραφο της βάσης δεδομένων world που είδαμε στα μαθήματα της SQL)
- Δημιουργούμε το χρήστη με στοιχεία username: πχ "pythonuser", password: "pwd1234" με δικαιώματα εκτέλεσης (αλλά όχι διαχείρισης), {βλ. βίντεο}



Η μέθοδος connect() συνδέει την εφαρμογή με τη MySQL:

- Δέχεται ως ορίσματα (με λέξεις-κλειδιά):
  - **host** (συμβολοσειρά, για τοπική εφαρμογή: localhost)
  - **user** (συμβολοσειρά, το username)
  - **password** (συμβολοσειρά, το password)
  - **database** (συμβολοσειρά, το όνομα της βάσης δεδομένων)
- Επιστρέφει αντικείμενο τύπου MYSQLConnection.

Παρατήρηση:

- Κανονικά πρέπει να ανοίγουμε και να κλείνουμε (μέθοδος close()) τη σύνδεση με τη βάση.
- Ωστόσο παρέχεται ένα πιο εύκολο συντακτικό με χρήση της with (βλ. και Python - Μάθημα 15)

Παράδειγμα 1: connection.py

```
import mysql.connector
MYSQL = mysql.connector

try:
   with MYSQL.connect(
      host="localhost",
      user="pythonuser",
      password="pwd1234",
      database="pyworld"
   ) as connection:
      # program code goes here
      print(connection)
except MYSQL.Error as e:
   print(e)
```

Παρατήρηση:

- Οι εντολές μας πρέπει να εντίθεται try…except block
- Παράγονται εξαιρέσεις (με γενικότερη τη MySQL.error) που μπορούν να μας καθοδηγήσουν για λάθη που έχουν συμβεί.

ΜΑΘΗΜΑ 1: Σύνδεση - Εκτέλεση Ερωτημάτων

2. Εκτέλεση Ερωτημάτων

MySQL psounis

- Για να τρέξουμε ένα ερώτημα (SELECT) στη βάση δεδομένων:
  - Κατασκευάζουμε ένα αντικείμενο **MySQLCursor** (μέλος του MySQLConnection - αυτό που επέστρεψε η connect):

κατασκευαστής	επεξήγηση
cursor()	κατασκευάζει αντικείμενο τύπου cursor

- Το αντικείμενο cursor περιτυλίσσει ένα SQL ερώτημα.
- Πρέπει να το εκτελέσουμε (**execute**) και έπειτα να πάρουμε (**fetch**) τα αποτελέσματα.

- Εκτελούμε ένα ερώτημα (συμβολοσειρά), με τη μέθοδο **execute** του αντικειμένου MySQLCursor:

μέθοδος	επεξήγηση
execute(string)	Εκτελεί το ερώτημα (string)

  - (προσοχή, ότι η execute δεν προκαλεί την επιστροφή γραμμών, αλλά εκτελεί το ερώτημα και μπορεί να προκαλέσει σφάλμα σε περίπτωση που κάτι πάει στραβά)
- Παίρνουμε τις γραμμές-απάντηση του ερωτήματος με τη μέθοδο **fetchall()** του αντικειμένου MySQLCursor:

μέθοδος	επεξήγηση
fetchall()	Επιστρέφει το αποτέλεσμα του ερωτήματος

  - επιστρέφει <u>λίστα από tuples</u> (κάθε γραμμή είναι ένα tuple)

μέθοδος	επεξήγηση
close()	Κάνει όλες τις απαραίτητες ενέργειες για το κλείσιμο του cursor

<u>Παράδειγμα 2: query</u>

```
import mysql.connector
from connect import open_connection, close_connection
MYSQL = mysql.connector

conn = open_connection()
try:
   cursor = conn.cursor()
   query = "SELECT * FROM country"
   cursor.execute(query)
   rows = cursor.fetchall()
   print(rows)
   close_cursor(cursor)
except MYSQL.Error as e:
   print(e)
close_connection(conn)
```

<u>Άσκηση 1:</u>

Κατασκευάστε συνάρτηση με όνομα print_country που να παίρνει ως όρισμα μια γραμμή (σαν αυτή που επιστρέφει το ερώτημα του παραδείγματος 1) και να την τυπώνει μορφοποιημένα. Έπειτα εκτυπώστε ‘κομψά’ όλες τις χώρες του πίνακα country.

- Δίνεται η δυνατότητα να πάρουμε το **result set ως <u>λίστα από λεξικά</u>** (κάνοντας πιο εύκολη τη διαχείριση σε σχέση με το απλό tuple)
  - Απλά διοχετεύουμε τη παράμετρο-λέξη κλειδί <u>dictionary</u> με τιμή <u>True</u>.

κατασκευαστής	επεξήγηση
cursor(Dictionary=True)	κατασκευάζει αντικείμενο τύπου CursorDict. Το result set θα είναι λεξικό.

<u>Παράδειγμα 3: dictionary_manip.py</u>

```
connector = open_connection()

try:
    cursor = connector.cursor(dictionary=True)

    query = "SELECT * FROM country"
    cursor.execute(query)

    rows = cursor.fetchall()
    for row in rows:
        print(f"{row['Name']}({row['Continent']})")

except MYSQL.Error as e:
    print(e)

close_connection(connector)
```

- Προσοχή ότι (σε αντίθεση με τη MySQL) τα ονόματα των στηλών είναι case sensitive.

<u>Άσκηση 2:</u>

Κατασκευάστε πρόγραμμα που να εκτυπώνει τις χώρες και τις πόλεις ανά χώρα της ηπείρου: North America, κάνοντας κατάλληλα ερωτήματα) στη ΒΔ. Οι πρώτες 7 γραμμές της εκτύπωσης θα πρέπει να είναι:

Aruba(Oranjestad)
Anguilla(South Hill, The Valley)
Netherlands Antilles(Willemstad)
Antigua and Barbuda(Saint John´s)
Bahamas(Nassau)
Belize(Belize City, Belmopan)
Bermuda(Saint George, Hamilton)
…

<u>Άσκηση 3:</u>

Κατασκευάστε πρόγραμμα το οποίο:

- Θα διαβάζει από την είσοδο μία ήπειρο
- Θα επιστρέφει την εκτύπωση των χωρών - πόλεων της ηπείρου (όπως στην άσκηση 2)

ΜΑΘΗΜΑ 1: Σύνδεση - Εκτέλεση Ερωτημάτων — 2.2. fetchone() και fetchmany()

MySQL psounis

- Η fetchone() επιστρέφει μία - μία τις γραμμές του result set
- είναι μέρος του cursor για τον οποίο έχει εκτελεστεί ένα ερώτημα.

μέθοδος	επεξήγηση
fetchone()	Επιστρέφει την επόμενη γραμμή. None αν δεν υπάρχει επόμενη γραμμή.

Παράδειγμα 4: fetchone

```
cursor = connector.cursor(dictionary=True)

query = "SELECT * FROM country"
cursor.execute(query)

country = cursor.fetchone()
while country is not None:
    print(f"{country['Name']}({country['Continent']}). Next? Press
                                                    Enter...", end="")
    input()
    country = cursor.fetchone()
```

Άσκηση 4:

Υλοποιήστε το παράδειγμα 4, μέσω iterator (η υλοποίηση του να γίνει μέσω generator)

- Η fetchmany() δέχεται ως όρισμα πόσες γραμμές του result set θέλουμε να μας επιστραφούν (ιδανική για πολύ μεγάλα result set)

μέθοδος	επεξήγηση
fetchmany (size)	Επιστρέφει size γραμμές. None αν δεν υπάρχουν επόμενες γραμμές.

Παράδειγμα 5: fetchmany

```
cursor = connector.cursor(dictionary=True)

query = "SELECT * FROM country LIMIT 42"
cursor.execute(query)

countries = cursor.fetchmany(10)
while countries:
    for country in countries:
        print(f"{country['Name']}({country['Continent']}). ")
    print("="*10 + ". Next? Press Enter...", end="")
    input()
    countries = cursor.fetchmany(10)
```

Άσκηση 5:

Υλοποιήστε το παράδειγμα 5, μέσω iterator (η υλοποίηση του να γίνει μέσω generator)

Άσκηση 6: Αναζήτηση με κριτήρια

Κατασκευάστε πρόγραμμα που θα αναζητά χώρες με τα εξής κριτήρια:

- Σε ποια ήπειρο ανήκουν
- Εύρος στο οποίο έγιναν ανεξάρτητα
- Όνομα του αρχηγού της (με μερική αναζήτηση, δηλαδή να απαντάει και αν ο χρήστης βάλει π.χ. μόνο το επώνυμο)

Φιλοδοξία της εφαρμογής αυτής είναι να απαντάει σε ερωτήματα π.χ. του τύπου: «Ποιες χώρες ανήκουν στην …, έγιναν ανεξάρτητές μεταξύ … και … και το επώνυμο του αρχηγού του κράτους είναι ...»

Είσοδος:

- Ο χρήστης θα πληκτρολογεί τα κριτήρια, αλλά θα έχει δικαίωμα να μην εισάγει καθόλου κάποιο κριτήριο.
- Αν δεν εισάγει κανένα κριτήριο να εμφανίζει όλες τις χώρες.

Έξοδος:

- Να παρατίθενται τα αναλυτικά στοιχεία κάθε χώρας, σε ομάδες των 5 χωρών τη φορά, χρησιμοποιώντας την print_country της άσκησης 1.