“Advanced Python Programming for Everybody”

Instructor: Ernest Bonat, Ph.D.
Senior Software Engineer
Senior Data Scientist
ebonat@15itresources.com
Cell: 503.730.4556

Module 3“Database Access and Data Manipulation”

1. Check Programs Installation:

- MySQL Local Engine (Select VERSION();)
- Database Management IDE
- MySQL Connector/Python (print(mysql.connector.__version__))

2. GitHub: https://github.com/ebonat/intel_module_3

Required Open Source Programs:

1. Download and Install MySQL database engine (https://dev.mysql.com/downloads/installer/). Make sure admin username and password are setup properly. You may need to write them down, so you'll not forget them.

2. Download and Install Connector/Python for MySQL (https://dev.mysql.com/downloads/connector/python/). If you're using Anaconda Distribution Package in your local laptop you'll not need to install it. Feel free to install it first and then update it using "conda" in the command prompt ("conda install mysql-connector-python", "conda update mysql-connector-python").

3. Create a simple Python program to test your connection to your local MySQL database engine. This must be done before next class (Module 3) starts for sure. Follow the instruction at "5.1 Connecting to MySQL Using Connector/Python" (https://dev.mysql.com/doc/connector-python/en/connector-python-example-connecting.html)

4. Feel free to read the MySQL Connector/Python Developer Guide (https://dev.mysql.com/doc/connector-python/en/)

5. Download and Install MySQL Workbench (https://dev.mysql.com/downloads/workbench/). Feel free to use any MySQL database management program if you would like to. I'm using Toad for MySQL in my laptop now (http://community-downloads.quest.com/toadsoft/MySQL/ToadforMySQL_Freeware_8.0.0.296.zip). Make sure that

your program can connect to your local MySQL database before the class (Module 3) starts - very important!

SQL Tutorials:

- https://www.w3schools.com/sql/
- https://www.tutorialspoint.com/sql/index.htm
- http://www.sqltutorial.org/

SQL Programming (CRUD Operations)

1. **SQL** (Structured Query Language) is a computer language aimed to store, manipulate, and query data stored in the databases engines like MySQL, SQL Server, Oracle, etc.

2. **CRUD** represents an acronym for the database operations **Create** (**Insert**), **Read** (**Select**), **Update**, and **Delete.**

3. **Stored Procedure** is a subroutine like a subprogram in a regular computing language, stored in database. A procedure has a name, a parameter list, and SQL statement(s). Why Stored Procedures? **Fast** (pre-compiled SQL in the database engine with catching) and **Security**.

Using the stored procedures instead of the SQL statements to build dynamic queries reduces the potential for SQL injection attacks from user input data because everything placed into a parameter gets quoted in the process.

MySQL database engine uses **SQL:2003** programming language. SQL:2003 is the fourth revision of the SQL database query language. The standard consists of 9 parts which are described in detail in SQL. It was updated by **SQL:2006**.

SQL Examples:

- Select
- Insert
- Update
- Delete

MySQL Connector/Python

1. Check for connection first!

```
import mysql.connector

def main():
    config = {
    “user”: “root”,
    “password”: “your_password”,
    “host”: “127.0.0.1”, # or “localhost”
    “database”: “magazine”,
    “raise_on_warnings”: True,
    }
    mysql_connector = mysql.connector.connect(**config)
    if (mysql_connector.is_connected()):
        print(“YES Connection!”)
        mysql_connector.close()
    else:
        print(“NO Connection!”)

if __name__ == '__main__':
    main()
```