Databases : Python such as SQLite3 MySQL MongoDB

# Basic for python with built-in SQLite3

Relational database management  
doesn’t require server and can be embedded

**Connect  
Establish a connection object in order to establish a connection to a database**

Connect to a database: creates if it doesn’t exist, connects if it exists using connect method

Create connection object:   
connection\_object=sqlite3.connect(‘datab\_name.db’)

We need a cursor, interact objects to database  
cursor\_name=connection\_object.cursor()

FOREIGN

SQL: Table – rows and columns

///execute is a method used to query or statement  
///Query is in uppercase for readability and is case-Insensitive  
///always mention the data type of the data to be given inside the cell, String is TEXT, VARCHAR(len) in SQL

cursor\_name.execute(“CREATE TABLE table\_name (colum\_name1 TEXT, column\_name2 TEXT, column\_name3 INTEGER)”)

/// a 3 column table

connection\_name.commit() ///saves

connection\_name.close()

#$#$#$#$

cursor\_name.execute(“CREATE TABLE IF NOT EXISTS table\_name (colum\_name1 TEXT, column\_name2 TEXT, column\_name3 INTEGER)”)

/// IF NOT EXISTS – used when error return : table already exists , occurs if program runs more than once

#$#$#$#$#$#$

/// insert 3 rows of data inside the empty table

cursor\_name.execute(“INSERT INTO table\_name VALUES (‘column1data1’,’column2data1’,column3data1), (‘column1data2’,’column2data2’,column3data2), (‘column1data3’,’column2data3’,column3data3)”)

/// single quotes is used to define strings since the queries are passed within double quotes and viceversa

#$#$#$#$#$#$#

////Primary key or unique id for each data and is Auto incrementing

cursor\_name.execute(“CREATE TABLE table\_name (id\_name INTEGER PRIMARY KEY, colum\_name1 TEXT, column\_name2 TEXT, column\_name3 INTEGER)”)

cursor\_name.execute(“CREATE TABLE sub\_table (sub\_col\_id INTEGER PRIMARY KEY, id\_of\_another\_table INTEGER, sub\_table\_data TEXT, FOREIGN KEY (id\_of\_another\_table) REFERENCES table\_name(id\_name))”)

////// Reference Primary key of other table inside this new table

///cursor\_name.execute(“INSERT INTO table\_name VALUES (1,’marc’,’specter’,002)”)

///but since primary key is auto incrementing we can just define the other three columns by name

cursor\_name.execute(“INSERT INTO table\_name (colum\_name1, column\_name2 , column\_name3) VALUES (’marc’,’specter’,002)”)

#$#$#$#$#$#

cursor\_name.execute(“UPDATE table\_name SET column\_name1 = ‘jack’ where column\_name1= ‘marc’ “)

//// Update information in table

#$#$#$#$#$#$

///To read the data rows with data\_to\_match in column 2

cursor\_name.execute(“SELECT \* FROM table\_name WHERE column2=’data’ “)

/// \* every column , WHERE – condition

/// cursor\_name.execute(“SELECT column\_name1 FROM table\_name “)

/// for specific columns

/// to get them in specific order  
/// cursor\_name.execute(“SELECT \* FROM table\_name ORDER BY column\_name2 DESC “)

///DESC – descending order, ASC – ascending order

/// cursor\_name.execute(“SELECT \* FROM table\_name ORDER BY column\_name2 DESC LIMIT 1“)

/// returns only 1 data by limiting

rows\_variable= cursor\_name.fetchall()

/// fetchall() method in sql

print(rows\_variable)

#$#$#$#$#$

Combine OOP and databases

DATACLASSES:

Data classes ; data focussed class ; database class

Functionality : classifier classes, regression classes, etc

Data clas

#$#$#$#$#$

//// GROUP BY

cursor\_name.execute(“SELECT column\_name1 FROM table\_name GROUP BY column\_name1 “)

#$#$#$#$#$

“SELECT AVG(data) --- returns average of the numerical data

#$#$#$#$#$

Leftover queries

#$#$#$#$#$

Database operations with different data types  
#$#$#$##$

Data types and formats allowed by sqlite3

#$#$#$#$

DDL and DML

#$#$#$#$

Python syntax for sqlite3  
#$#$#$#$

cursor.executemany()

#$#$#$#$

Values(?,?,?)  
tuple with data – multiple tuples in a sequence

#$#$#$#$