PYTHON DATABASE

Introduction

- Data
- Database
- DBMS
- RDBMS
- SQL
- No-SQL
- Big Data

SQL

Structured **Q**uery **L**anguage is a standard Database language which is used to create, maintain and retrieve the relational database.

SQL is case insensitive. But it is a recommended practice to use keywords in capital letters

Types of SQL:

- DDL
- DML
- DCL
- TCL

DDL

Data Definition Language (DDL) statements are used to define the database structure or schema.

CREATE - to create objects in the database
ALTER - alters the structure of the database
DROP - delete objects from the database
TRUNCATE - remove all records from a table permanently
RENAME - rename an object

DML and DQL

Data Manipulation Language (DML) statements are used for managing data within schema objects.

INSERT - insert data into a table
UPDATE - updates existing data within a table
DELETE - deletes unwanted/all records from a table

Data Query Language

SELECT - To Project the data

DCL

Data Control Language (DCL) statements are used for granting and denying access privileges to other users

GRANT - gives user's access privileges to database

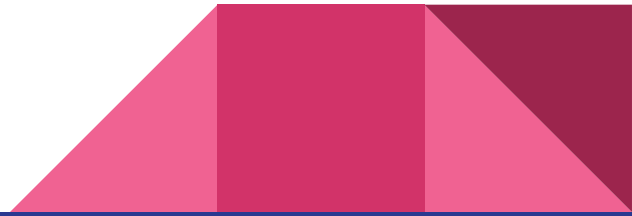
REVOKE - withdraw access privileges given with the GRANT command

TCL

Transaction Control (TCL) statements are used to manage the changes made by DML statements. It allows statements to be grouped together into logical transactions.

COMMIT - save work done

ROLLBACK - restore database to original since the last COMMIT SAVEPOINT - identify a point in a transaction to which you can later roll back



Getting Started With MySql

Prerequisites:

MySql Database download:

Windows User:
https://dev.mysql.com/get/Downloads/MySQLInstaller/mysql-installer-web-community-8.0.25.0.msi
Mac and Linux User:
https://dev.mysql.com/downloads/mysql/

Install MySql Driver:
pip install mysql-connector-python

Database Connection

```
import mysql.connector                  #Importing mysql connector

con = mysql.connector.connect(
  host="localhost",                     #DB Host
  user="root",                          #DB Username
  password="admin"                      #DB Password
)
print(con)
```

Create Database and Table

Create Database:

```
cur=con.cursor()
cur.execute("CREATE DATABASE test_db")
```

Once this gets executed, test_db will be created in MySql Database.

Connection With Database Name:

```
con = mysql.connector.connect(
  host="localhost",
  user="root",
  password="admin",
  database='test_db'
)
```

Create Table:

```
cur=con.cursor()
cur.execute("CREATE TABLE emp (eid INTEGER,ename VARCHAR(50),esal INTEGER)")
```

Working with DMLs

Insert:

```
cur=con.cursor()
cur.execute("INSERT INTO emp(eid,ename,esal) VALUES(101,'King',50000)")    #Direct Query
sql="INSERT INTO emp(eid,ename,esal) VALUES (%s,%s,%s)"
val=(102,'Gopal',30000)
cur.execute(sql,val)                #Execute Query with Value
con.commit()                        #Commit to save the records
```

Select:

```
cur=con.cursor()
cur.execute("SELECT * FROM emp")
res=cur.fetchall()        #it return table data as list of tuples
for row in res:
  print(row)
```

Update:

```
cur=con.cursor()
cur.execute("UPDATE EMP SET esal=70000 WHERE eid=101")
con.commit()                              #To save the changes
print(cur.rowcount,"Rows affected")       #Return the affected rows
```

Delete:

```
cur=con.cursor()
cur.execute("DELETE FROM emp WHERE eid=101")
con.commit()
print(cur.rowcount,"Rows affected")              #Return the nos of deleted rows
```

Drop:

```
cur=con.cursor()
cur.execute("DROP TABLE emp")
```

JSON

What is JSON?

- JSON stands for Java String Object Notation
- It is a standard format for information interchange.
- JSON Contains key-value pair, key and value are separated by colon (:)
- Each key-values pairs are separated by comma (,)
- String is JSON objects are surrounded by double quote (“)

A Sample JSON:

```
{
  "name" : "King",
  "age" : 24,
  "loc" : "Nuapada"
}
```

Read JSON in Python

In Python JSON can be converted into Dictionary by importing json module.

Let's try to read one json file:

```
import json                    #Import json module
f=open("simple.json")
data=f.read()
simple_dict=json.loads(data)
print(type(simple_dict))
```

```
Op:
<class 'dict'>
```

So here JSON file is converted to a dictionary. This can be accessed using dictionary attributes.