

## STEPS TO CREATE IBM DB2 AND CONNECTION WITH PYTHON

|              |                                    |
|--------------|------------------------------------|
| Date         | 18 November 2022                   |
| Team ID      | PNT2022TMID03947                   |
| Project Name | Skills/Job Recommender Application |

### STEP 1: Import the `ibm_db` Python library:

```
!pip install --force-reinstall ibm_db==3.1.0 ibm_db_sa==0.3.7 import  
ibm_db
```

### STEP 2: Identify the database connection credentials:

```
"connection": {  
  "cli": {  
    "arguments": [  
      [  
        "-u",  
        "mzw99208",  
        "-p",  
        "X50Cas8TwAsNxiZ9",  
        "--ssl",  
        "--sslCAFile",  
        "1cbbb1b6-3a1a-4d49-9262-3102a8f7a7c8",  
        "--authenticationDatabase",  
        "admin",  
        "--host",  
        "54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733"  
      ]  
    ],  
    "bin": "db2",  
    "certificate": {  
      "certificate_base64": "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSB0tLS0tCk1JSURFakNDQWZxZ0F3SUJBZ0lKQVA1S0R3ZTNCTkxiTUEwR0NTcUdTSWlZRFFFQkN3VUFNQjR4SERBYUJnTIYKQkFNTUUbENUU0JEYkc5MVpDQkVZWVJ0WWV1GelpYTXdlGNOTWpBd01qSTVNRFF5TVRBeVdoY05NekF3TWpJMgpNRF5TVRBeVdqQWVNUnd3R2dZRFZRUUREQk5KUWswZ1EyeHZkV1FnUkdGMFIxSmhjMlZ6TUIJQklqQU5CZ2txCmhraUc5dzBCQVFFRkFBT0NBUThtBTUIJQkNnS0NBUEVBdXUvbWpWW9xdkdGNU8xSGpEalpsK25iYjE4UkR4ZGwKTzRUL3FoUGMxMTREY1FUK0plRXdhG13aGljTGxaQnF2QWFMb1hnbmhmSVFOMG01L0x5YzdBY291VXNmSGR0QWpDVGcrSUxsbjBrdDMrTHM3d1dTakxqVE96N3M3MlZUSU5yYmx3cnRIRUlvM1JWTKV6SkNHYW5LSXZdZWZVSUtrClcNMIROSDI5cnFsSGN0Z2pIUlFmRkVTRmlYaHJiODhSQmd0amIva0xtVGpCaTFBeEVadWNobWZ2QVRmNEN0Y3EKY21QcHNqdDBPTnI0YnhjMVRYUWxEemNiN1hMSFBrWW91SUprdnVzMUZvaTEySmRNM1MrK3labFZPMUZmZkU3bWpKMjhUdGJoZ3JGOGtIU0NMSkYvTTFsZ3FPZG90Vm5QOC9EOWZhamNNN0lWd2V4a0lSOTNKR1FjREFRQUJvMU13ClVUQWRZC05WSFE0RUZnUVVlQ3JZanFJQzc1VUpXVmZEMDh1ZWdqeDZiUmN3SHdZRFZSMGpCQmd3Rm9BVWVdClkKaNFJQzc1VUpXVmZEMDh1ZWdqeDZiUmN3RHdZRFZSMFRBUUgVqkFVd0F3RUlvekFOQmdrcWhraUc5dzBCQVFFZGpBU9DQVFFQUkyRTBUOUt3MlN3RjJ2MXBqaHV4M0lkWWV2SGFVSkRMb0tPd0hSRnFSOHgxZ2dRcGVFcFBNMk5SCkx3R08yek85SWZUMmhLaWd1d2orWnJ5SGxcHlxQ0pLOHJEU28xZUVPEklyWmE2S1YrQTVScEttMWdjV3VHYzMkK1UrVTFzTDdlUjd3ZFFuVjU0TVU4aERvNi9sVHRMRVB2Mnc3VlNPSlFDK013ejgrTFJMdjVHSW5BNlJySWNhKwozM0wxNnB4ZEttd1pLYThWcnBnMXJ3QzRnY3dlYUhyMUNEWE42K0JlbzhvWG5YWkh6UG91cldYS1BoaGdXZ2J5CkNDcUdIK0NWNnQ1eFg3b05NS3VNSUNqRV
```

```

ZndnNLWnRqeTQ5VW5iNVZZbHQ0b1J3dTFlbGdzRDNjekltbjlLREQKNHB1REFvYTZyMktZZE4xVlxuN3
F3VG1TbDITU05RPT0KLS0tLS1FTkQgQ0VSVElGSUNBVEUtLS0tLQo=",
  "name": "1cbbb1b6-3a1a-4d49-9262-3102a8f7a7c8"
},
"composed": [
  "db2 -u mzw99208 -p X50Cas8TwAsNxiZ9 --ssl --sslCAFile 1cbbb1b6-3a1a-4d49-9262-3102a8f7a
7c8 --authenticationDatabase admin --host 54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lq
de00.databases.appdomain.cloud:32733"
],
"environment": {},
"type": "cli"
},
"db2": {
  "authentication": {
    "method": "direct",
    "password": "X50Cas8TwAsNxiZ9",
    "username": "mzw99208"
  },
  "certificate": {
    "certificate_base64": "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSB0tLS0tCk1JSURFakNDQWZxZ0F3SUJBZ
0lKQVA1S0R3ZTNCTkxiTUEwR0NTcUdTSWlZRFFFQkN3VUFNQjR4SERBYUJnTiYKQkFNTUuwbENUU0
JEYkc5MVpDQkVZWVJoWW1GelpYTXdlaGNOTWpBd01qSTVNRFF5TVRBeVdoY05NekF3TWpJMgpNRF
F5TVRBeVdqQWVNUnd3R2dZRFZRUUREQk5KUWswZ1EyeHZkV1FnUkdGMFlXSmhjMlZ6TUIJQklqQU5
CZ2txCmhraUc5dzBCQVFFRkFBT0NBUThBTUIJQkNnS0NBUEVBdXUvbWpWW9xdkdGNU8xSGpEalpsK2
5iYjE4UkR4ZGwKTzRUL3FoUGMxMTREY1FUK0plRXdhG13aGljTGxaQnF2QWFMb1hnbmhmSVFOMG0
1L0x5YzdBY291VXNmSGR0QwpDVGcrSUsxbjBrdDMrTHM3d1dTakxqVE96N3M3MlZUSU5yYmx3cnRI
RUlvM1JWTKV6SkNHYW5LSXZdZWVZVSUtrClNMIROSDl5cnFsSGN0Z2pIUlFmRkVTRmlYaHJiODhSQm
d0amlva0xtVGpCaTFBeEVadWNobWZ2QVRmNEN0Y3EKY21QcHNqdDBPTnI0YnhJMVRyUWxEemNiN
1hMSFBrWW91SUprdnVzMUZvaTEySmRNM1MrK3labFZPMUZmZkU3bwpKMjhUdGJoZ3JGOgtIU0NMS
kVjTTFsz3FPZG90Vm5QOC9EOWZhamNNN0lWd2V4a0lSOTNKR1FjREFRQUJvMU13ClVUQWRZC05WS
FE0RUZnUVVlQ3JZanFJQzc1VUpxVmZEMDh1ZWdqeDZiUmN3SHdZRFZSMGpCQmd3Rm9BVWVDclKkKa
nFJQzc1VUpxVmZEMDh1ZWdqeDZiUmN3RHdZRFZSMFRBUUgVqkFVd0F3RUlvekFOQmdrcWhraUc5dz
BCQVFzRgpBQU9DQVFFQkYkRTBUOUt3MIN3RjJ2MXBqaHV4M0lkWWV2SGFVSkRmB0tPd0hSRnFSOH
gxZ2dRcGVFcFBNMk5SCkx3R08yek85SWZUMmhLaWd1d2orWnJ5SGxxcHlxQ0pLOHJEU28xZUVPEkly
WmE2S1YrQTVscEttMWdjV3VHYzMkk1UrrVTFzTDdlUjd3ZFFuVjU0TVU4aERvNi9sVHRMRVB2Mnc3Vl
NPSlFDK013ejgrTFJMdjVHSW5BNlJySWNhKwozM0wxNnB4ZEttd1pLYThWcnBnMXJ3QzRnY3dlYUhmY
UNEWE42K0JlbzhvWG5YWkh6UG91cldYS1BoaGdXZ2J5CkNDcUdIK0NWNnQ1eFg3b05NS3VNSUNqRV
ZndnNLWnRqeTQ5VW5iNVZZbHQ0b1J3dTFlbGdzRDNjekltbjlLREQKNHB1REFvYTZyMktZZE4xVlxuN3
F3VG1TbDITU05RPT0KLS0tLS1FTkQgQ0VSVElGSUNBVEUtLS0tLQo=",
  "name": "1cbbb1b6-3a1a-4d49-9262-3102a8f7a7c8"
},
"composed": [
  "db2://mzw99208:X50Cas8TwAsNxiZ9@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu
0lqde00.databases.appdomain.cloud:32733/bludb?authSource=admin&replicaSet=replset"
],
"database": "bludb",
"host_ros": [
  "54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:305
92"
],
"hosts": [
  {
    "hostname": "54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdom
ain.cloud",
    "port": 32733
  }
],
"jdbc_url": [
  "jdbc:db2://54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomai
n.cloud:32733/bludb:user=<userid>;password=<your_password>;sslConnection=true;"
],

```

```

    "path": "/bludb",
    "query_options": {
      "authSource": "admin",
      "replicaSet": "replset"
    },
    "replica_set": "replset",
    "scheme": "db2",
    "type": "uri"
  }
},
"instance_administration_api": {
  "deployment_id": "crn:v1:bluemix:public:dashdb-for-transactions:us-south:a/e6391f45afa647b8ae19f77ecfab2ec0:afb2bc32-4b27-4bac-8318-9379cd61e6d5::",
  "instance_id": "crn:v1:bluemix:public:dashdb-for-transactions:us-south:a/e6391f45afa647b8ae19f77ecfab2ec0:afb2bc32-4b27-4bac-8318-9379cd61e6d5::",
  "root": "https://api.db2.cloud.ibm.com/v5/ibm"
}
}

```

### STEP 3: Create the DB2 database connection:

```

dsn = (
    "DRIVER={0};"
    "DATABASE={1};"
    "HOSTNAME={2};"
    "PORT={3};"
    "PROTOCOL={4};"
    "UID={5};"
    "PWD={6};"
    "SECURITY={7};").format(dsn_driver, dsn_database, dsn_hostname, dsn_port, dsn_protocol,
dsn_uid, dsn_pwd,dsn_security) print(dsn)

```

Now establish the connection to the database

try:

```
conn = ibm_db.connect(dsn, "", "")      print ("Connected to database: ", dsn_database, "as user:  
", dsn_uid, "on host: ", dsn_hostname)
```

```
except:
```

```
    print ("Unable to connect: ", ibm_db.conn_errormsg())
```

```
server = ibm_db.server_info(conn)
```

```
print ("DBMS_NAME: ", server.DBMS_NAME)
```

```
print ("DBMS_VER: ", server.DBMS_VER)
```

```
print ("DB_NAME: ", server.DB_NAME)
```

```
client = ibm_db.client_info(conn)
```

```
print ("DRIVER_NAME: ", client.DRIVER_NAME) print
```

```
("DRIVER_VER: ", client.DRIVER_VER) print
```

```
("DATA_SOURCE_NAME: ", client.DATA_SOURCE_NAME) print
```

```
("DRIVER_ODBC_VER: ", client.DRIVER_ODBC_VER) print
```

```
("ODBC_VER: ", client.ODBC_VER)
```

```
print ("ODBC_SQL_CONFORMANCE: ", client.ODBC_SQL_CONFORMANCE)
```

```
print ("APPL_CODEPAGE: ", client.APPL_CODEPAGE) print
```

```
("CONN_CODEPAGE: ", client.CONN_CODEPAGE)
```

#### STEP 4: Close the Connection:

```
ibm_db.close(conn)
```

#### STEP 5: Integration of ibm db2 in flask app:

```
def dashhome():
    insert_sql = "INSERT INTO SKILLUSER (EMAILID, PASSWORD, NAME,PHONENUMBER) VALUES (?, ?, ?, ?)"
    prep_stmt = ibm_db.prepare(conn, insert_sql)
    email = request.form['username']
    password = request.form['password']
    name = request.form['name']
    ph = request.form['phonenum']
    var_list.append(email)
    var_list.append(password)
    var_list.append(name)
    var_list.append(ph)
    ibm_db.bind_param(prepare_stmt, 1, ph)
    ibm_db.bind_param(prepare_stmt, 2, name)
    ibm_db.bind_param(prepare_stmt, 3, email)
    ibm_db.bind_param(prepare_stmt, 4, password)
    print("giun")
    ibm_db.execute(prepare_stmt)
    return render_template('email.html')
```

IBM Db2 on Cloud

Load Data

Load History

Tables

Views

Indexes

Aliases

MQTs

Sequences

Application objects

Find schemas or tables

Refresh

SQL

Tables

New table

| Name  | Schema   | Properties |
|---|----------|------------|
| <input checked="" type="checkbox"/> CUSTOMERS | MZW99208 | ...        |
| <input type="checkbox"/> DETAILS              | MZW99208 | ...        |
| <input type="checkbox"/> SKILLUSER            | MZW99208 | ...        |
| <input type="checkbox"/> STUDENTS             | MZW99208 | ...        |

Total: 4, selected: 1

Table definition

SKILLUSER

Approximate 0 rows (0 KB)  
Updated on 2022-11-18 15:03:19

| Name        | Data type | Nullable | Length | Scale |
|-------------|-----------|----------|--------|-------|
| PHONENUMBER | INTEGER   | Y        |        | 0     |
| NAME        | VARCHAR   | Y        | 255    | 0     |
| EMAILID     | VARCHAR   | Y        | 255    | 0     |
| PASSWORD    | VARCHAR   | Y        | 255    | 0     |

View data

## DATA INTEGRATION IN TABLE SKILLUSER

IBM Db2 on Cloud

Load Data

Load History

Tables

Views

Indexes

Aliases

MQTs

Sequences

Application objects

MZW99208.SKILLUSER

Back

Export to CSV

| PHONENUMBER | NAME                       | EMAILID                         | PASSWORD |
|-------------|----------------------------|---------------------------------|----------|
| 88888       | hi                         | pradeepnarayaniyappan@gmail.com | ssssss   |
| 7777777     | fff                        | sec19ec096sairamtap.edu.in      | dog      |
| 99999999    | wolf                       | pradeepnarayaniyappan@gmail.com | admin    |
| 99999999    | mycluster-dal10-           | pradeep2001                     | dddddd   |
| 99999999    | mycluster-dal10-           | sec19ec096sairamtap.edu.in      | dddddd   |
| 99999999    | mycluster-dal10-           | sec19ec096sairamtap.edu.in      | ffff     |
| 950265645   | Prem Kashyap Chilakamarthi | Prem0046                        | prem     |

```
def subapp():
    insert_sql = "INSERT INTO DETAILS (NAME,FNAME,GENDER,EID,ADDRESS,TENTHMARK,TWELTHMARK,DEG_CGPA,AADHAR,DOMAIN) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?)"
    prep_stmt = ibm_db.prepare(conn, insert_sql)
    name = request.form['yourname']
    fname = request.form['fname']
    GD = request.form['GD']
    EID = request.form['EID']
    add = request.form['AL1']
    mark1 = request.form['s5']
    mark2 = request.form['h5']
    cgpa = request.form['b5']
    AADHAR = request.form['Aadhar']
    DOMAIN = request.form['domain']

    ibm_db.bind_param(prepare_stmt, 1, name)
    ibm_db.bind_param(prepare_stmt, 2, fname)
    ibm_db.bind_param(prepare_stmt, 3, GD)
    ibm_db.bind_param(prepare_stmt, 4, EID)
    ibm_db.bind_param(prepare_stmt, 5, add)
    ibm_db.bind_param(prepare_stmt, 6, mark1)
    ibm_db.bind_param(prepare_stmt, 7, mark2)
    ibm_db.bind_param(prepare_stmt, 8, cgpa)
    ibm_db.bind_param(prepare_stmt, 9, AADHAR)
    ibm_db.bind_param(prepare_stmt, 10, DOMAIN)
```

TABLE VALUES FOR DETAILS TABLE IN IBMDB2

IBM Db2 on Cloud

Load DataLoad HistoryTablesViewsIndexesAliasesMQTsSequencesApplication objects

Find schemas or tables

Refresh

Tables

New table

| Name      | Schema   | Properties |
|-----------|----------|------------|
| CUSTOMERS | MZW99208 | ...        |
| DETAILS   | MZW99208 | ...        |
| SKILLUSER | MZW99208 | ...        |
| STUDENTS  | MZW99208 | ...        |

Total: 4, selected: 1

Table definition

DETAILS

| Name    | Data type | Nullable | Length | Scale |
|---------|-----------|----------|--------|-------|
| NAME    | VARCHAR   | Y        | 255    | 0     |
| FNAME   | VARCHAR   | Y        | 255    | 0     |
| GENDER  | VARCHAR   | Y        | 255    | 0     |
| EID     | VARCHAR   | Y        | 255    | 0     |
| ADDRESS | VARCHAR   | Y        | 255    | 0     |

View data

DATA INTEGRATION FOR DETAILS TABLE

IBM Db2 on Cloud

Load DataLoad HistoryTablesViewsIndexesAliasesMQTsSequencesApplication objects

MZW99208.DETAILS

Back

Export to CSV

| NAME    | FNAME | GENDER | EID                             | ADDRESS | TENTHMARK | TWELTHMARK | DEG_CGPA | AADHAR | DOMAIN |
|---------|-------|--------|---------------------------------|---------|-----------|------------|----------|--------|--------|
| pradeep | r     | on     | pradeepnarayaniyappan@gmail.com | no 6    | 5         | 5          | 6        | 545457 | gggg   |
| pradeep | r     | on     | pradeepnarayaniyappan@gmail.com | no 6    | 5         | 5          | 6        | 545457 | gggg   |

## Output on Creating tables using sqllite in DB2

IBM Db2 on Cloud

Load Data

Load History

Tables

Views

Indexes

Aliases

MQTs

Sequences

Application objects

Find schemas or tables

Refresh

Schemas

| <input checked="" type="checkbox"/> | Name     | Type | Tables |
|-------------------------------------|----------|------|--------|
| <input checked="" type="checkbox"/> | MZW99208 | User | 4      |

Total: 1, selected: 1

Tables

New table +

| <input checked="" type="checkbox"/> | Name      | Schema   | Properties |
|-------------------------------------|-----------|----------|------------|
| <input checked="" type="checkbox"/> | CUSTOMERS | MZW99208 | ...        |
| <input type="checkbox"/>            | DETAILS   | MZW99208 | ...        |
| <input type="checkbox"/>            | SKILLUSER | MZW99208 | ...        |
| <input type="checkbox"/>            | STUDENTS  | MZW99208 | ...        |

Total: 4, selected: 1