

# ASSIGNMENT-2

## IBM CLOUD DBS

Student Name	S.Esvarapandiyar
Student Roll Number	951919CS022

### Questions:

1. Create User table with user with email, username, roll number, password

The screenshot displays the IBM Db2 on Cloud web interface. The 'Tables' tab is selected, showing a list of tables. The 'STUDENTS' table is selected, and its definition is shown on the right. The table has four columns: USERNAME (CHAR, 6, nullable), ROLLNO (INTEGER, nullable), EMAIL ID (VARCHAR, 32, nullable), and PASSWORD (VARCHAR, 32, nullable). A 'View data' button is visible at the bottom of the table definition panel.

Name	Data type	Nullable	Length	Scale
USERNAME	CHAR	Y	6	0
ROLLNO	INTEGER	Y		0
EMAIL ID	VARCHAR	Y	32	0
PASSWORD	VARCHAR	Y	32	0

2. Perform UPDATE, DELETE Queries with user table.

**INSERT TABLE:-**

IBM Db2 on Cloud

Data objects Saved objects

Filter objects

PTM66823

SQL

History

Find history

Script	Date	Status	Runtime
Untitled - 1	Nov 9, 2022 10:11:23 AM	1	0.009 s
insert into user values('n.sanjayspidey@gmail.com','SANJAI KUMAR',1901084,123);			

27°C Partly sunny

ENG IN 10:11 09-11-2022

IBM Db2 on Cloud

Load Data Load History Tables Views Indexes Aliases MQTs Sequences Application objects

PTM66823.USER

Back

Export to CSV

EMAIL	USERNAME	ROLL.NO	PASSWORD
n.sanjayspidey@gmail.com	SANJAI KUMAR	1901084	123

27°C Partly sunny

ENG IN 10:11 09-11-2022

UPDATE TABLE:-

IBM Db2 on Cloud

Data objects Saved objects

Filter objects

PTM66823

Untitled - 1

```
1 insert into user values('n.sanjayspidey@gmail.com','SANJAI KUMAR',1901084,123);
```

History

Find history

Script	Date	Status	Runtime
Untitled - 1	Nov 9, 2022 10:11:23 AM	1	0.009 s
insert into user values('n.sanjayspidey@gmail.com','SANJAI KUMAR',1901084,12...			0.009 s

27°C Partly sunny

ENG IN 10:11 09-11-2022

IBM Db2 on Cloud

Load Data Load History **Tables** Views Indexes Aliases MQTs Sequences Application objects

PTM66823.USER

Back

Export to CSV

EMAIL	USER NAME	ROLL NO	PASSWORD
n.sanjayspidey@gmail.com	SANJAI KUMAR	1901084	21052002

27°C Partly sunny

ENG IN 10:20 09-11-2022

**DELETE TABLE:-**

The screenshot shows the IBM Db2 on Cloud console interface. On the left, there's a sidebar with navigation icons. The main area is titled "Data objects" and shows a list of objects, including "PTM66823". The right pane shows the SQL editor with three lines of SQL code:

```
1 insert into user values('n.sanjayspidey@gmail.com','SANJAI KUMAR',1901084,123);
2 update user SET password=21052002 where email='n.sanjayspidey@gmail.com';
3 delete from user where email='n.sanjayspidey@gmail.com';
```

Below the editor is a "History" table showing the execution of these queries:

Script	Date	Status	Runtime
Untitled - 1	Nov 9, 2022 10:22:58 AM	✓ 1	0.011 s
delete from user where email='n.sanjayspidey@gmail.com'		✓	0.011 s
Untitled - 1	Nov 9, 2022 10:19:46 AM	✓ 1	0.018 s
update user SET password=21052002 where email='n.sanjayspidey@gmail.com'		✓	0.018 s
Untitled - 1	Nov 9, 2022 10:19:32 AM	✓ 1	0.010 s
insert into user values('n.sanjayspidey@gmail.com','SANJAI KUMAR',1901084,12...		✓	0.010 s

The screenshot shows the IBM Db2 on Cloud console interface, specifically the "Tables" tab. The table "PTM66823.USER" is selected. The table structure is displayed as follows:

EMAIL	USER NAME	ROLL NO	PASSWORD
There is no data here yet			

### 3. CONNECT PYTHON CODE TO DB2.

**NOTE:-** Question 4 contains Question 3 answer

4. Create a flask app with registration page, login page and welcome page. By default load the registration page once the user enters all the fields store the data in database and navigate to login page authenticate user username and password. If the user is valid show the welcome page

```

from flask import Flask, render_template, request, redirect, url_for,
session import ibm_db import re
app = Flask(__name__) app.secret_key

=

conn =
    ibm_db.connect( "DATABASE=Db2New;HOSTNAME=19af6446
-6171-4641-8aba-
9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=30699;USERNAM
E=kgb92696;PASSWORD=3IlWyGYU3bMIvwnX;SECURITY=SSL;SSLSERVERCERTIFICATE=DigiCer

@app.route("/", methods=['GET', 'POST']) def
register(): msg = '' if request.method ==
'POST':
    username = request.form['username'] email =
    request.form['email'] password =
    request.form['password'] sql = "SELECT * FROM
users WHERE username =?" stmt =
    ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt, 1, username)
    ibm_db.execute(stmt) account =
    ibm_db.fetch_assoc(stmt) print(account) if
account:
    msg = 'Account already exists !'
elif not re.match(r'^@+@[^@]+\.[^@]+', email): msg
    = 'Invalid email address !' elif not
re.match(r'[A-Za-z0-9]+', username):
    msg = 'name must contain only characters and numbers !'

```