

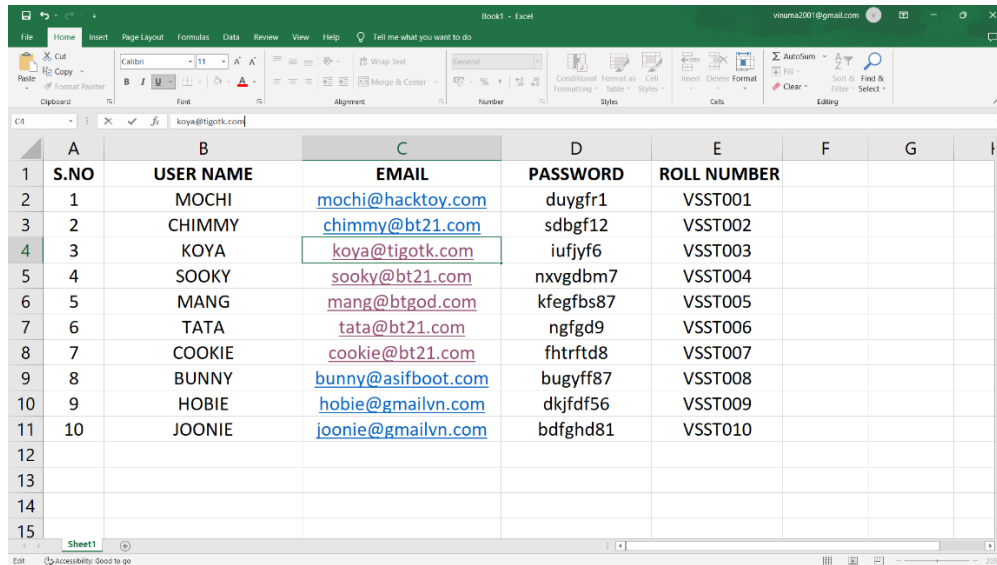
## Assignment -1

### CLOUD APPLICATION DEVELOPEMENT

Assignment Date	19 September 2022
Student Name	L.VINOTHINI
Student Roll Number	812419104074
Maximum Marks	2 Marks

#### Question-1:

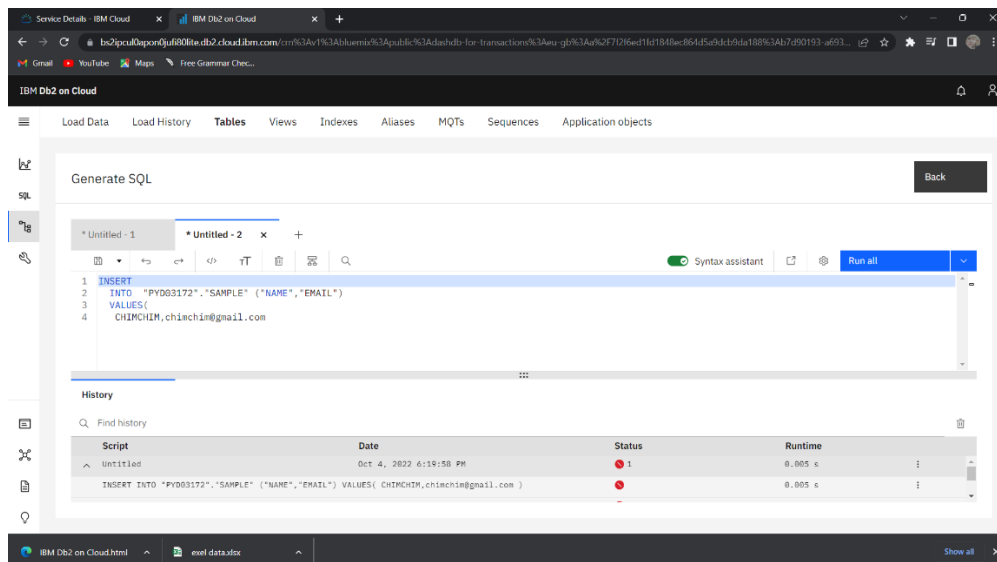
1. Create User table with email, username, roll number, password.



S.NO	USER NAME	EMAIL	PASSWORD	ROLL NUMBER
1	MOCHI	mochi@hacktoy.com	duygfr1	VSST001
2	CHIMMY	chimmy@bt21.com	sdbgf12	VSST002
3	KOYA	koya@tigotk.com	iufjyf6	VSST003
4	SOOKY	sooky@bt21.com	nxvgdbm7	VSST004
5	MANG	mang@btgod.com	kfegfbs87	VSST005
6	TATA	tata@bt21.com	ngfgd9	VSST006
7	COOKIE	cookie@bt21.com	fhtfrtd8	VSST007
8	BUNNY	bunny@asifboot.com	bugyff87	VSST008
9	HOBIE	hobie@gmailvn.com	dkjfdf56	VSST009
10	JOONIE	joonie@gmailvn.com	bdfghd81	VSST010

#### Question-2:

2. Perform UPDATE, DELETE Queries with user table



Generate SQL

```

1 INSERT
2 INTO "PYD03172"."SAMPLE" ("NAME","EMAIL")
3 VALUES(
4     CHIMCHIM, chimchim@gmail.com

```

History

Script	Date	Status	Runtime
INSERT INTO "PYD03172"."SAMPLE" ("NAME","EMAIL") VALUES( CHIMCHIM, chimchim@gmail.com )	Oct 4, 2022 4:19:58 PM	1	0.005 s

Service Details - IBM Cloud x IBM Db2 on Cloud x +

IBM Db2 on Cloud

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

### Generate SQL

Back

\*Untitled - 1 x +

1 DELETE FROM "PYD03172"."SAMPLE"  
2 --Search condition (e.g. WHERE "COL1" = NULL)  
3

Syntax assistant Run all

#### History

Find history

Script	Date	Status	Runtime
Untitled - 1	Oct 4, 2022 6:23:41 PM	1	0.007 s
DELETE FROM "PYD03172"."SAMPLE" --Search condition (e.g. WHERE "COL1" = NULL) WHERE 1 = 1			0.007 s

IBM Db2 on Cloud.html excel data.xlsx Show all

Service Details - IBM Cloud x IBM Db2 on Cloud x +

IBM Db2 on Cloud

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

### Generate SQL

Back

\*Untitled - 1 x +

1 UPDATE "PYD03172"."DEMO"  
2 SET  
3 "S\_NO" = 1, --S\_NO SMALLINT  
4 "USER\_NAME" = c3immy, --USER\_NAME VARCHAR(6)

Syntax assistant Run all

#### History

Find history

Script	Date	Status	Runtime
Untitled - 1	Oct 4, 2022 6:25:51 PM	1	0.024 s

IBM Db2 on Cloud.html excel data.xlsx Show all

### Question-3:

Connect python code to db2.

The screenshot shows the IBM Db2 on Cloud console interface. At the top, there is a navigation bar with the IBM Db2 on Cloud logo and a user profile icon. Below the navigation bar, there is a red error banner that reads "Error: Invalid file type. Supported file types are .csv and .txt." In the center, there is a large blue circle with the number "11" inside it, indicating the number of rows loaded. To the right of the circle, there is a green banner that reads "The data load job succeeded. Load excel data.csv from My Computer to PYD03172.DEMO. 2022/10/04, 12:28 PM. View details". Below the circle, there is a table with the following data:

Rows read	Rows loaded	Rows rejected
11	11	0

Below the table, there is a section for "Status" and "Settings". The "Status" section shows the start time as 10/04/2022 12:28:38 PM and the end time as 10/04/2022 12:28:42 PM. The "Settings" section shows the target as PYD03172.DEMO. To the right of the status section, there is a section for "Errors" and "Warnings", which is currently empty and displays "No errors".

The screenshot shows the IBM Db2 on Cloud console interface during the data load job configuration. At the top, there is a navigation bar with the IBM Db2 on Cloud logo and a user profile icon. Below the navigation bar, there is a red error banner that reads "Error: Invalid file type. Supported file types are .csv and .txt." In the center, there is a section for "Source" and "Target". The "Source" section shows the file "excel data.csv" and the "Target" section shows the target "PYD03172.DEMO". Below the source and target sections, there is a section for "Define" and "Finalize". The "Define" section shows the code page (character encoding) as 1208 (UTF-8), the separator as a comma, the header in first row as checked, the time and date format as selected, and the detect data types as unchecked. Below the define section, there is a table with the following data:

S_NO SMALLINT	USER_NAME VARCHAR(6)	EMAIL VARCHAR(18)	PASSWORD VARCHAR(9)	ROLL_NUMBER VARCHAR(7)
1	MOCHI	mochi@hacktoy.com	duygr1	VSST001
2	CHIMMY	chimmy@bt21.com	sdbg12	VSST002
3	KOYA	koya@tigotk.com	iufjyf6	VSST003
4	SOOKY	sooky@bt21.com	nxvgdbm7	VSST004
5	MANG	mang@btgod.com	kfegfbs87	VSST005
6	TATA	tata@bt21.com	ngfgd9	VSST006
7	COOKIE	cookie@bt21.com	fhtrftd8	VSST007
8	BUNNY	bunny@asifboot.com	bugyff87	VSST008
9	HOBIE	hobie@gmailvn.com	dkjfdt56	VSST009
10	JOONIE	joonie@gmailvn.com	bdfghd81	VSST010

At the bottom right of the table, there are "Back" and "Next" buttons.

#### Question-4:

3. 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 with username and password. If the user is valid show the welcome page.

[Source code](#)

app.py

```
from flask import Flask,render_template
```

```
app=Flask(__name__)
```

```
@app.route("/")
```

```
def home():
```

```
    return "welcome to nta"
```

```
if __name__=="__main__":
```

```
    app.run(debug=True)
```

```
from flask import Flask,render_template
app=Flask(__name__)
@app.route("/")
def home():
    return "welcome to nta"
if __name__=="__main__":
    app.run(debug=True)
```

