

The report for 4140 Assignment 5

1. Before using python to operate database, we created the schema and the table student_info in the MySQL. We also imported data of student_info into MySQL through the csv file. However, the other table 'course' is created in the Jupyter Notebook. In addition to screenshots in this report, we also enclose the output file (ipynb file) in the submitting file.
2. Query explanation

Query 1:

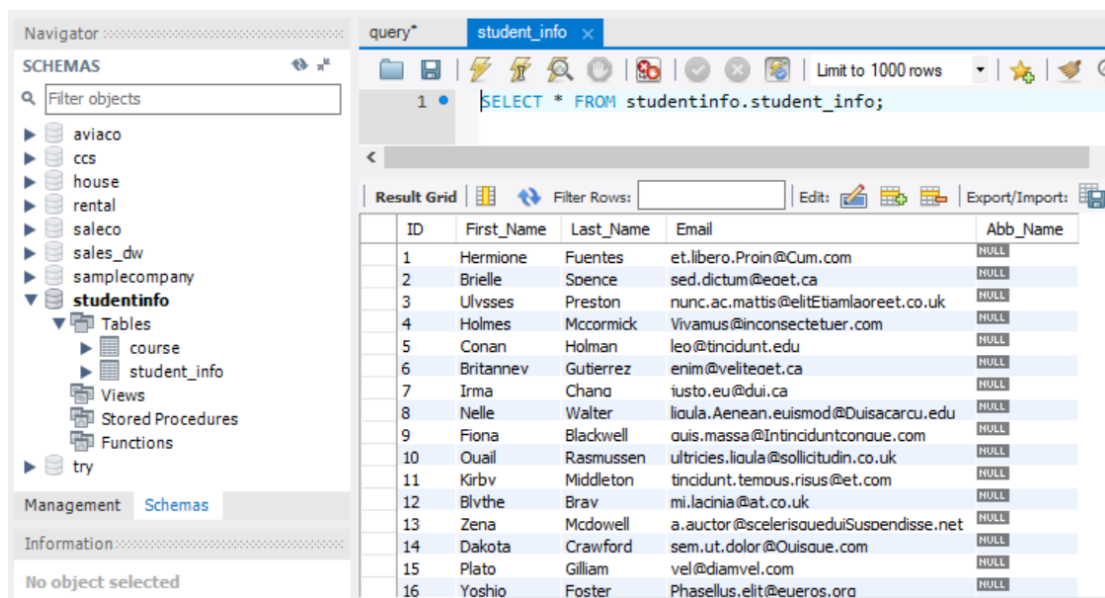
Alter table student_info add (Abb_Name varchar (10));

Explanation:

Adding a new column named 'Abb_Name' in the table student_info which contains abbreviation of the student name.

Business scenario:

By adding Abb_Name, users can access student name faster



The screenshot shows a MySQL database interface. On the left is the 'Navigator' pane with a tree view of schemas. The 'studentinfo' schema is selected, showing tables 'course' and 'student_info'. The 'student_info' table is highlighted. The main area shows a query editor with the query: `SELECT * FROM studentinfo.student_info;`. Below the query editor is the 'Result Grid' showing 16 rows of data. The columns are ID, First_Name, Last_Name, Email, and Abb_Name. The Abb_Name column contains 'NULL' for all rows.

ID	First_Name	Last_Name	Email	Abb_Name
1	Hermione	Fuentes	et.libero.Proin@Cum.com	NULL
2	Brielle	Soence	sed.dictum@eget.ca	NULL
3	Ulvsses	Preston	nunc.ac.mattis@elitEtiamlaoreet.co.uk	NULL
4	Holmes	Mccormick	Vivamus@inconsectetuer.com	NULL
5	Conan	Holman	leo@tincidunt.edu	NULL
6	Britannev	Gutierrez	enim@veliteget.ca	NULL
7	Irma	Chano	iusto.eu@dui.ca	NULL
8	Nelle	Walter	liqua.Aenean.euismod@Duisacarcu.edu	NULL
9	Fiona	Blackwell	quis.massa@Intinciduntconque.com	NULL
10	Ouail	Rasmussen	ultrices.liqua@sollicitudin.co.uk	NULL
11	Kirby	Middleton	tincidunt.tempus.risus@et.com	NULL
12	Blvthe	Brav	mi.lacinia@at.co.uk	NULL
13	Zena	Mcdowell	a.auctor@scelerisqueuiSuspendisse.net	NULL
14	Dakota	Crawford	sem.ut.dolor@Ouisque.com	NULL
15	Plato	Gilliam	vel@diamvel.com	NULL
16	Yoshio	Foster	Phasellus.elit@eueros.org	NULL

```
In [1]: import mysql.connector
conn = mysql.connector.connect(
    user='root',
    password='RogerF1520',
    host='localhost',
    database='studentinfo')

cur = conn.cursor()

query = ("SELECT * FROM student_info")

cur.execute(query)

print("ID First_name Last_Name Email")
for (col1,col2,col3,col4) in cur:
    print("{} , {} , {} , {}".format(col1,col2,col3,col4))

cur.close()
conn.close()
```

```
ID First_name Last_Name Email
1, Hermione, Fuentes, et.libero.Proin@Cum.com
2, Brielle, Spence, sed.dictum@eget.ca
3, Ulysses, Preston, nunc.ac.mattis@elitEtiamlaoreet.co.uk
4, Holmes, McCormick, Vivamus@insectetuer.com
5, Conan, Holman, leo@tincidunt.edu
6, Britanney, Gutierrez, enim@veliteget.ca
7, Irma, Chang, justo.eu@dui.ca
8, Nelle, Walter, ligula.Aenean.euismod@Duisacarcu.edu
9, Fiona, Blackwell, quis.massa@Intinciduntcongue.com
10, Quail, Rasmussen, ultricies.ligula@sollicitudin.co.uk
11, Kirby, Middleton, tincidunt.tempus.risus@et.com
```

Query 2:

Update student_info set Abb_Name='H.F' where ID=1

Update student_info set Abb_Name='B.S' where ID=2;

Update student_info set Abb_Name='U.P' where ID=3;

Update student_info set Abb_Name='H.M' where ID=4;

Explanation:

Update data of the student_info table. It adds four new values of Abb_name according to the given ID .

Business scenario:

By adding Abb_Name, users can access student name faster. In this case, user inserts new data to existed data.

	ID	First_Name	Last_Name	Email	Abb_Name
	1	Hermione	Fuentes	et.libero.Proin@Cum.com	H.F
	2	Brielle	Spence	sed.dictum@eget.ca	B.S
	3	Ulysses	Preston	nunc.ac.mattis@elitEtiamlaoreet.co.uk	U.P
	4	Holmes	McCormick	Vivamus@insectetuer.com	H.M
	5	Conan	Holman	leo@tincidunt.edu	NULL
	6	Britanney	Gutierrez	enim@veliteget.ca	NULL
	7	Irma	Chang	justo.eu@dui.ca	NULL
	8	Nelle	Walter	ligula.Aenean.euismod@Duisacarcu.edu	NULL
	9	Fiona	Blackwell	quis.massa@Intinciduntcongue.com	NULL
	10	Quail	Rasmussen	ultrices.ligula@sollicitudin.co.uk	NULL
	11	Kirby	Middleton	tincidunt.tempus.risus@et.com	NULL
	12	Blvthe	Brav	mi.lacinia@at.co.uk	NULL
	13	Zena	Mcdowell	a.auctor@sclerisqueuiSusnendisse.net	NULL

Query 3:

```
SELECT * FROM student_info;
```

Explanation:

This query is used to select everything in the table student_info.

Business scenario:

To review all of the data in the student_info table

```
In [5]: cur = conn.cursor()

query = ("SELECT * FROM student_info")

cur.execute(query)

print("ID First_name Last_Name Email Abb_Name")
for (col1,col2,col3,col4,col5) in cur:
    print("{} , {} , {} , {} , {}".format(col1,col2,col3,col4,col5))

cur.close()
conn.commit()
```

ID	First_name	Last_Name	Email	Abb_Name
1	Hermione	Fuentes	et.libero.Proin@Cum.com	H.F
2	Brielle	Spence	sed.dictum@eget.ca	B.S
3	Ulysses	Preston	nunc.ac.mattis@elitEtiamlaoreet.co.uk	U.P
4	Holmes	Mccormick	Vivamus@inconsectetuer.com	H.M
5	Conan	Holman	leo@tincidunt.edu	None
6	Britanney	Gutierrez	enim@veliteget.ca	None
7	Irma	Chang	justo.eu@dui.ca	None
8	Nelle	Walter	ligula.Aenean.euismod@Duisacarcu.edu	None
9	Fiona	Blackwell	quis.massa@Intinciduntcongue.com	None
10	Quail	Rasmussen	ultrices.ligula@sollicitudin.co.uk	None
11	Kirby	Middleton	tincidunt.tempus.risus@et.com	None
12	Blythe	Bray	mi.lacinia@at.co.uk	None
13	Zena	Mcdowell	a.auctor@scelerisqueuiSuspendisse.net	None
14	Dakota	Crawford	sem.ut.dolor@Quisque.com	None
15	Plato	Gilliam	vel@diamvel.com	None
16	Yoshio	Foster	Phasellus.elit@eueros.org	None

Query 4:

```
Insert into student_info values(101,'Jiali','Jin','jialijin@dal.ca','J.F');
```

```
Insert into student_info values(102,'Ada','Wong','ada@gmail.ca','A.W')
```

Explanation:

These queries are used to insert new rows in the table of student_info. These two rows' ID are 101 and 102.

Business scenario:

Adding new data I the table when there are new students.

The screenshot shows a database management interface with a 'student_info' table. The table has 16 rows. The last two rows, ID 101 and 102, are highlighted in blue, indicating they are the newly inserted data. The table structure is as follows:

ID	First_Name	Last_Name	Email	Abb_Name
92	Sandra	Coffey	nec@Praesenteu.co.uk	NULL
93	Channing	Ortiz	pharetra.nibh@Aenean.co.uk	NULL
94	Chase	Brvan	lorem.Donec.elementum@malesuadaid...	NULL
95	Roonev	Truillo	nonummv.ultrices.ornare@amet.ca	NULL
96	Kameko	Miles	dictum@Crasoellentesque.co.uk	NULL
97	Drake	Nichols	vitae@lacinia.ca	NULL
98	Orlando	Pearson	eleifend@loulaelitoretum.net	NULL
99	Daquan	Wong	Aliquam@nuncrisus.com	NULL
100	Rashad	Robertson	et@lectus.com	NULL
101	Jiali	Jin	jialijin@dal.ca	J.F
102	Ada	Wong	ada@gmail.ca	A.W

```

88, Reagan, Morgan, Nam.consequat@estvitae.ca, None
89, Morgan, Kirby, at.libero.Morbi@neque.org, None
90, Stuart, Reyes, Cras.convallis@commodoipsum.ca, None
91, Gisela, Stanton, ut.aliquam.iaculis@Sed.co.uk, None
92, Sandra, Coffey, nec@Praesenteu.co.uk, None
93, Channing, Ortiz, pharetra.nibh@Aenean.co.uk, None
94, Chase, Bryan, lorem.Donec.elementum@malesuadaiderat.org, None
95, Rooney, Trujillo, nonummy.ultrices.ornare@amet.ca, None
96, Kameko, Miles, dictum@Craspellentesque.co.uk, None
97, Drake, Nichols, vitae@Iacinia.ca, None
98, Orlando, Pearson, eleifend@ligulaelitpretium.net, None
99, Daquan, Wong, Aliquam@nuncrisus.com, None
100, Rashad, Robertson, et@lectus.com, None
101, Jiali, Jin, jialijin@dal.ca, J.F
102, Ada, Wong, ada@gmail.ca, A.W

```

Query 5:

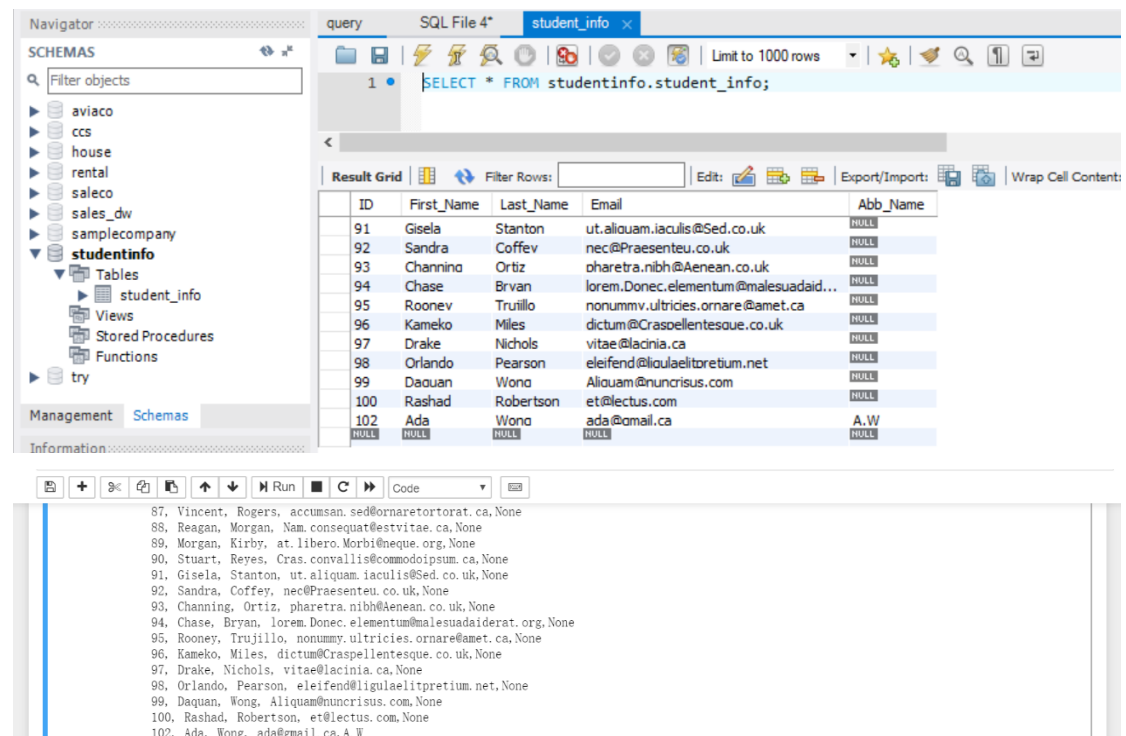
delete from student_info where ID='101';

Explanation:

This query is used to delete a row whose student ID is 101.

Business scenario:

Delete certain data as required.



The screenshot shows the SQL Developer interface. The 'Navigator' pane on the left displays the database schema, including tables like 'student_info'. The 'query' window shows the SQL statement: `SELECT * FROM studentinfo.student_info;`. The 'Result Grid' displays the following data:

ID	First_Name	Last_Name	Email	Abb_Name
91	Gisela	Stanton	ut.aliquam.iaculis@Sed.co.uk	NULL
92	Sandra	Coffey	nec@Praesenteu.co.uk	NULL
93	Channing	Ortiz	pharetra.nibh@Aenean.co.uk	NULL
94	Chase	Bryan	lorem.Donec.elementum@malesuadaid...	NULL
95	Rooney	Trujillo	nonummy.ultrices.ornare@amet.ca	NULL
96	Kameko	Miles	dictum@Craspellentesque.co.uk	NULL
97	Drake	Nichols	vitae@Iacinia.ca	NULL
98	Orlando	Pearson	eleifend@ligulaelitpretium.net	NULL
99	Daquan	Wong	Aliquam@nuncrisus.com	NULL
100	Rashad	Robertson	et@lectus.com	NULL
101	Ada	Wong	ada@gmail.ca	A.W
102	Ada	Wong	ada@gmail.ca	A.W

The 'Code' window below shows the same data as a text list, with row 101 highlighted in blue.

Query 6:

create table course (num_of_courses int, course_ID int primary key);

Explanation:

(we also insert data in this table and selected all in the table:

query:

Insert into course values(2,1);

Insert into course values(5,2);

Insert into course values(3,3);

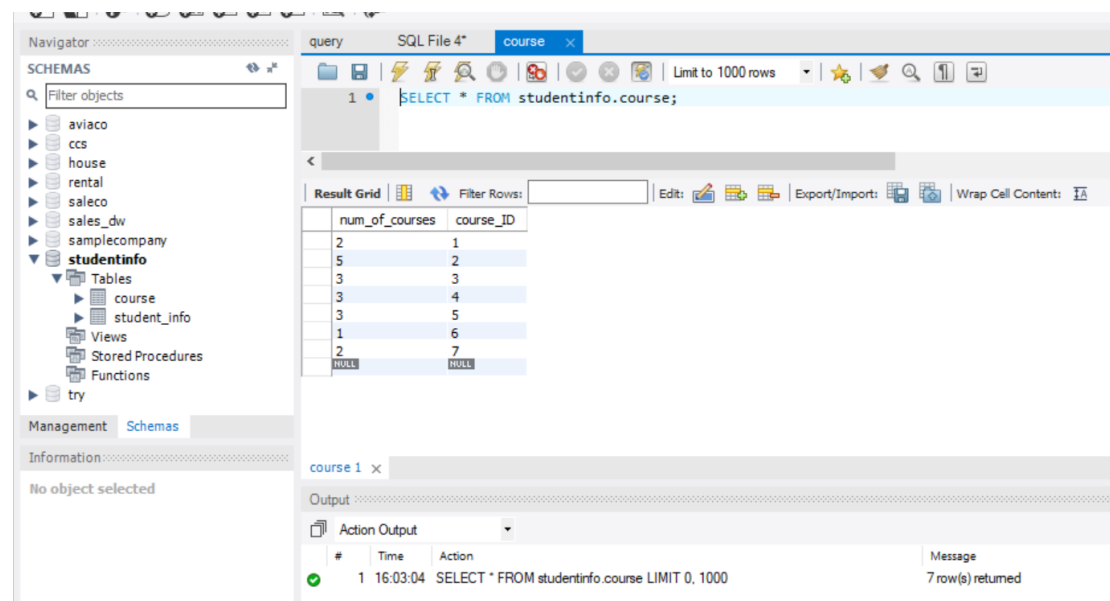
Insert into course values(3,4);

Insert into course values(3,5);
 Insert into course values(1,6);
 Insert into course values(1,7);
 SELECT * FROM course)

This query is used to create a new table named class. This new table has columns of 'num_of_courses' and course_ID. The courses_ID is the primary key.

Business scenario:

Creating a new table named class to store data of class information.



The screenshot shows a SQL query window with the query `SELECT * FROM studentinfo.course;` executed. The result grid displays the following data:

num_of_courses	course_ID
2	1
5	2
3	3
3	4
3	5
1	6
2	7
NULL	NULL

The bottom of the screenshot shows the output window with the message: "1 16:03:04 SELECT * FROM studentinfo.course LIMIT 0, 1000 7 row(s) returned".

```

In [14]: cur = conn.cursor()

query = ("SELECT * FROM course")

cur.execute(query)

print("num_of_courses course_ID")
for (col1,col2) in cur:
    print("{} , {}".format(col1,col2))

cur.close()
conn.commit()
  
```

```

num_of_courses course_ID
2, 1
5, 2
3, 3
3, 4
3, 5
1, 6
2, 7
  
```

2 . Structure of tables

The table of student_info has four columns. They are ID, First_Name, Last_Name and Email. The ID is the primary key and its datatype is integer. The datatype of First_Name, Last_Name and Email are all varchar. The size of the first tow varchar are 55. The size of Email is 255 as email may need larger size to store.

The table of course has two columns. They are 'number_of_courses' and 'ID'. Both of their datatype are integer and the ID is the primary key.

3.Feedback of all the tools used in the assignment

In this assignment, we use Python, Jupyter Notebook and MySQL. Jupyter Notebook is a useful tool. It offers a convenient way to code Python. Unlike last assignment that we need to upload files to get feedbacks and results when we used Nodejs, we can straightly get results when we run codes in the Jupyter Notebook. The whole process is also less complicated. Jupyter Notebook is easier to test or debug. As for Python, we feel like it is much easier to use when comparing with Nodejs.

4.Reference

Connecting to MySQL Using Connector/Python

<https://dev.mysql.com/doc/connector-python/en/connector-python-example-connecting.html>