

8 වන පාඩම - වගු දෙකක් අතර සම්බන්ධතාව ගොඩනැගීම

මෙම පාඩම තුළින් වගු දෙකක් සම්බන්ධ කොට MySQL විමසුමක් හරහා දත්ත ලබා ගන්නේ කෙසේද යන්න පැහැදිලි කරයි.

අපගේ දත්ත සමුදායේ දැනටමත් 'personal' නමින් වගුවක් ඇති අතර තවත් වගුවක් සෑදිය යුතුය ('subjects' නමින්).

MySQL දත්ත සමුදායේ වගුවක් සෑදීම සඳහා CREATE TABLE ජරකාශය භාවිතා කරයි. මෙහිදී ඔබ වගුවේ නම සහ එක් එක් තීරුවට අඩංගු වන දත්ත වර්ගය (name සහ datatype) සඳහන් කළ යුතුය.

අපගේ Personal වගුවේ indexno සහ name ලෙස තීරු දෙකක් ඇත. එක් එක් සිසුවා ඉගෙන ගන්නා විෂයන් ඇතුළත් කිරීමට නව වගුව යොදාගනී. මෙම වගුවේ indexno සහ subject ලෙස තීරු දෙකක් ඇත. උදාහරණයක් ලෙස, නව වගුව (Subjects) පහත ආකාරයෙන් දැක්විය හැකිය.

SUBJECTS	
indexno	subject
001	Geography
002	Music
002	History
003	Science
003	Mathematics
003	Geography
004	Business Studies
004	Music
005	Geography

සටහන:

පළමු සිසුවා එක් විෂයක් හදාරයි: *Geography*, දෙවන සිසුවා විෂයන් දෙකක් හදාරයි: *Music* සහ *History*, තුන්වන සිසුවා විෂයන් තුනක් හදාරයි: *Science*, *Mathematics*, *Geography*, හතරවන සිසුවා විෂයන් දෙකක් හදාරයි: *Business Studies*, *Music*. පස්වන සිසුවා එක් විෂයක් හදාරයි: *Geography*.

මීට පෙර අපි phpMyAdmin වෙබ් අතුරුමුහුණත භාවිතා කරමින් දත්ත සමුදායක් සහ වගුවක් සකස් කොට ඊට වාර්තා එකතු කළෙමු. දැන් අපි subjects වගුව නිර්මාණය කර එයට Python හරහා වාර්තා එකතු කරමු.

subjects වගුව නිර්මාණය කිරීමට අදාළ Python කේතය:-

```
sql = 'CREATE TABLE subjects (indexno VARCHAR(3), subject  
    VARCHAR(20)) '  
mycursor.execute(sql)
```

ඔබ දත්ත සමුදාය නිර්මාණය කළ පසු, අපට subjects වගුවට වාර්තා එකතු කළ හැකිය. එහි **Python** කේතය:-

```
sql = 'INSERT INTO subjects (indexno, subject)
      VALUES (%s, %s) '
val = [
    ('001', 'Geography'),
    ('002', 'Music'),
    ('002', 'History'),
    ('003', 'Science'),
    ('003', 'Mathematics'),
    ('003', 'Geography'),
    ('004', 'Business Studies'),
    ('004', 'Music'),
    ('005', 'Geography')
]
mycursor.executemany(sql, val)
```

අවසාන වශයෙන්, ඔබ ඇතුළත් කළ දත්ත පෙන්වීම සඳහා අපට පහත කේතයක එක් කළ හැකිය.

```
sql = 'SELECT * from subjects'
mycursor.execute(sql)

myresult = mycursor.fetchall()

print(myresult)
```

ඉහත සාකච්ඡා කළ පියවර තුනම එක් **Python** වැඩසටහනකට ඇතුළත් කරමු. පහත **Python** කේතය (dbnewtable.py) අධ්‍යයනය කර වගුව නිර්මාණය කිරීමට එය ක්‍රියාත්මක කර ඉහත වාර්තා එක් කරන්න.

```
denewtable.py - C:/Users/Prabhath/OneDrive/Desktop/To Hide/Python Lesson/denewtable.p...
File Edit Format Run Options Window Help
import mysql.connector

mydb = mysql.connector.connect (
    host = 'localhost',
    database = 'student',
    user = 'root',
    password = '',
)

mycursor = mydb.cursor()

sql = 'CREATE TABLE subjects (indexno VARCHAR(3), subject VARCHAR(20))'
mycursor.execute(sql)

sql = 'INSERT INTO subjects (indexno, subject) VALUES (%s, %s)'
val = [
    ('001', 'Geography'),
    ('002', 'Music'),
    ('002', 'History'),
    ('003', 'Science'),
    ('003', 'Mathematics'),
    ('003', 'Geography'),
    ('004', 'Business Studies'),
    ('004', 'Music'),
    ('005', 'Geography')
]
mycursor.executemany(sql, val)

sql = 'SELECT * from subjects'
mycursor.execute(sql)

myresult = mycursor.fetchall()

print(myresult)
```

වගු දෙකක් හෝ ඊට වැඩි ගණනක් අතර සම්බන්ධතාව ගොඩනැගීම

Join විධානය භාවිතා කරමින්, වගු දෙකක් අතර සම්බන්ධතාව ගොඩනැගීමට එම වගු දෙකට ම පොදු සම්බන්ධිත තීරුවක් යොදාගනිමින් වගු දෙකකින් හෝ වැඩි ගණනකින් දත්ත ඒකාබද්ධ කළ හැකිය.

අපට වගු දෙකක් ඇත: **personal** සහ **subjects**.

Personal

```
indexno: '001', name: 'Tharindu'
indexno: '002', name: 'Supun'
indexno: '003', name: 'Bhagya'
indexno: '004', name: 'Sachini'
indexno: '005', name: 'Hiruni'
indexno: '006', name: 'Janani'
indexno: '007', name: 'Pasindu'
```

```
indexno: '008', name: 'Lahiru'
indexno: '009', name: 'Isuru'
indexno: '010', name: 'Fathima'
```

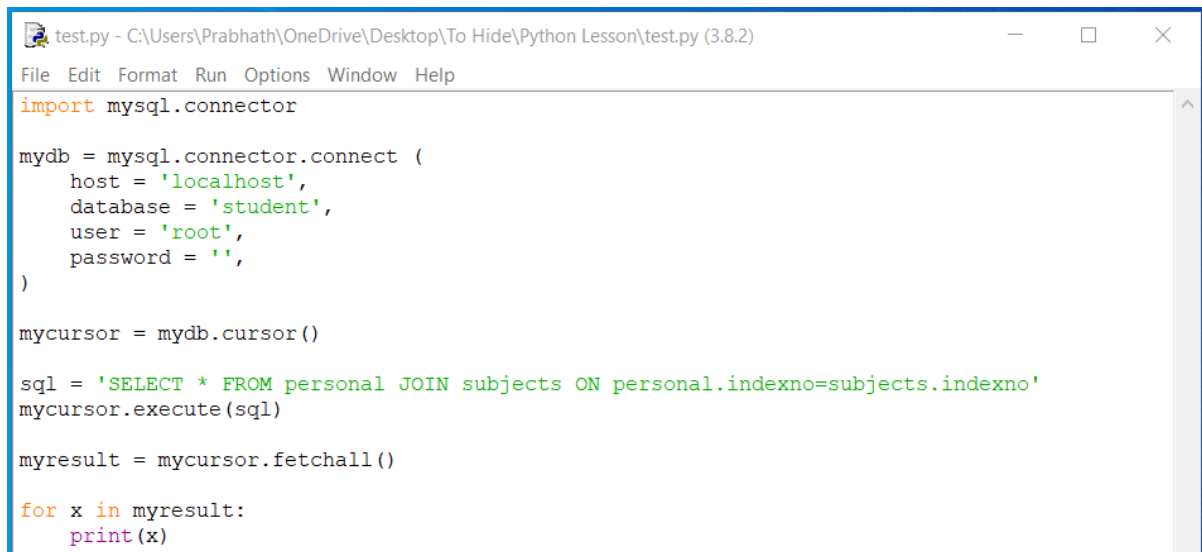
Subjects

```
indexno: '001', subject: 'Geography'
indexno: '002', subject: 'Music'
indexno: '002', subject: 'History'
indexno: '003', subject: 'Science'
indexno: '003', subject: 'Mathematics'
indexno: '003', subject: 'Geography'
indexno: '004', subject: 'Business Studies'
indexno: '004', subject: 'Music'
indexno: '005', subject: 'Geography'
```

Personal වගුවේ indexno සහ **subjects** වගුවේ indexno කීරු යොදා ගනිමින් වගු දෙක අතර සම්බන්ධතාව ගොඩනැගිය හැකිය.

```
SELECT * FROM personal JOIN subjects ON personal.indexno=subjects.indexno
```

සම්පූර්ණ **Python** කේතය පහත දැක්වේ



```
test.py - C:\Users\Prabhath\OneDrive\Desktop\To Hide\Python Lesson\test.py (3.8.2)
File Edit Format Run Options Window Help
import mysql.connector

mydb = mysql.connector.connect (
    host = 'localhost',
    database = 'student',
    user = 'root',
    password = '',
)

mycursor = mydb.cursor()

sql = 'SELECT * FROM personal JOIN subjects ON personal.indexno=subjects.indexno'
mycursor.execute(sql)

myresult = mycursor.fetchall()

for x in myresult:
    print(x)
```

ඉහත වැඩසටහනේ පේරාකිවිල පහත දැක්වේ.

```
===== RESTART: C:\Users\Prabhath\OneDrive\Desktop\To Hide\Python Lesson\test.py =====
('001', 'Tharindu', '001', 'Geography')
('002', 'Supun', '002', 'Music')
('002', 'Supun', '002', 'History')
('003', 'Bhagya', '003', 'Science')
('003', 'Bhagya', '003', 'Mathematics')
('003', 'Bhagya', '003', 'Geography')
('004', 'Sachini', '004', 'Business Studies')
('004', 'Sachini', '004', 'Music')
('005', 'Hiruni', '005', 'Geography')
>>> |
```

Ln: 99 Col: 4

වඩා පැහැදිලි ප්‍රතිඵලයක් ලබා ගැනීම සඳහා GROUP_CONCAT විධානය භාවිතා කරන්න.

```
SELECT personal.indexno, personal.name, \
GROUP_CONCAT(subjects.subject) \
FROM personal \
JOIN subjects \
ON personal.indexno=subjects.indexno \
GROUP BY personal.indexno
```

පහත ආකාරයෙන් ප්‍රතිඵල ලබා ගැනීම සඳහා ඉහත SQL ප්‍රකාශය යොදාගනිමින් ඔබේ කේතය වෙනස් කරන්න.

```
===== RESTART: C:\Users\Prabhath\OneDrive\Desktop\To Hide\Python Lesson\test3.py =====
001 Tharindu - Geography
002 Supun - History,Music
003 Bhagya - Mathematics,Science,Geography
004 Sachini - Business Studies,Music
005 Hiruni - Geography
>>> |
```

Ln: 120 Col: 4