அலகு V

பாடம் 15

SQL மூலம் தரவுகளைக் கையாளுதல்

2369H7

கற்றலின் நோக்கங்கள்

இந்த பாடப்பகுதியைக் கற்றபின் மாணவர்கள், பைத்தான் ஸ்கிரிப்ட்டைப் பயன்படுத்தி,

- ஒரு தரவு அட்டவணையை உருவாக்கி அதில் புதிய வரிசைகளைத் தரவுத் தளத்தில் சேர்த்தல்.
- ஒரு தரவு அட்டவணையில் உள்ள பதிவுகளை மேம்படுத்துதல் மற்றும் நீக்குதல்.
- ஒரு தரவு அட்டவணையில் வினவல்.
- CSV கோப்பில் வினவலை எழுதுதல்.

15.1 அறிமுகம்

ஒரு தரவுத்தளம் என்பது ஒருங்கிணைக்கப்பட்ட தரவுகளின் தொகுப்பாகும். "தரவுத்தளம்" என்ற கூறு தரவுகளை அல்லது தரவுத்தள மேலாண்மை அமைப்பைக் குறிக்கும். பயனர்கள் மற்றும் தரவுத்தளத்திற்கு இடையே உள்ள தொடர்புக்கு பயன்படும் மென்பொருள் பயன்பாடே தரவுத்தள மேலாண்மை அமைப்பாகும். பயனர்கள் மனிதர்களாக இருக்க வேண்டிய அவசியமில்லை, பிற நிரல்களாகவோ அல்லது பயன்பாடுகளாகவோ இருக்கலாம். பைத்தான் நிரல் எவ்வாறு SQL தரவுத்தளத்துடன் பயனராக தொடர்பு கொள்கிறது என்பதை இனி கற்கலாம்.

15.2 SQLite

SQLite என்பது எளிய உறவுநிலை தரவுத்தள அமைப்பாகும். இது தரவுகளை முறையான தரவுக்கோப்புகளாகவும் கணினியின் உட்புற நினைவகத்தில் கூட சேமித்து வைக்கும். இது MySQL அல்லது Oracle போன்று தனித்த தரவுத்தள சேவையக நிரலாக இல்லாமல் உள்ளிணைந்த பயன்பாடாக வடிவமைக்கப்பட்டுள்ளது. வேகமாகவும், மிகுந்த சோதிக்கப்பட்டதாகவும் மற்றும் நெகிழ்வானதாகவும் உள்ளதால் SQLite-ல் வேலை செய்வது எளிதாகும். SQLite ற்காக பைத்தான் சிறப்பான நூலகத்தைக் கொண்டுள்ளது.

SQLite யைப் பயன்படுத்த,

- படிநிலை 1: sqlite3யை இணைக்கவும்
- படிநிலை 2: connect( ) வழிமுறையைப் பயன்படுத்தி இணைப்பை உருவாக்கி தரவுத்தளத்தின் பெயரை அனுப்பவும்.
- படிநிலை 3: cursor= connection.cursor( ) என்றக் கூற்றைப் பயன்படுத்தி cursor என்னும் பொருளை அனுப்பவும்.

Chapter 15.indd 295 30-01-2020 20:58:04

- படிநிலை 2 ல் தரவுத்தளத்தை இணைத்தல் என்பது, அணுக வேண்டிய தரவுத்தளத்தின் பெயரைஅனுப்புதல் என்பதாகும். அவ்வாறு அனுப்பும்போது, அத்தரவுத்தளம் ஏற்கனவே இருக்குமாயின் அது இணைக்கப்படும். இல்லாவிடில், பைத்தான், கொடுக்கப்பட்ட பெயரில் ஒரு புதிய தரவுத்தளத்தை உருவாக்கும்.
- படிநிலை3ல் cursor என்பது ஒரு கட்டுப்பாட்டு அமைப்பாகும், இது தரவுத்தளப் பதிவுகளை இணைக்கப் பயன்படுகிறது.
- பைத்தானில், cursor மிக முக்கிய பங்குவகின்றது. அனைத்து கட்டளைகளும் cursor பொருள் மூலமே மட்டுமே இயக்கப்படும்.

ஒரு அட்டவணையை உருவாக்க, தரவுத்தளத்தில் ஒரு பொருளை உருவாக்கி அதற்கான SQL கட்டளைகளை எழுத வேண்டும்.

எடுத்துக்காட்டு:- sql_comm = "SQL statement"

கட்டளைகளை நிறைவேற்ற cursor வழிமுறையைப் பயன்படுத்தி sql கட்டளையின் பெயரை அளபுருக்களாக அனுப்ப வேண்டும். sql_comm-ல் நிறைய கட்டளைகள் சேமிக்கப்பட்டு அவை நிறைவேற்றப்படும். அனைத்து செயல்களையும் முடித்த பின்னர் மாற்றங்களைக் கோப்பில் சேமித்து பின்னர் இணைப்பை மூட வேண்டும்.

15.3 SQLite டைப் பயன்படுத்தி தரவுத்தளத்தை உருவாக்குதல்

பின்வரும் எடுத்துக்காட்டு sqlite3 பயன்படுத்தி பைத்தானில் ஒரு தரவு தளத்தை எவ்வாறு இணைப்பது என்பதை விளக்குகிறது.

```
# Python code to demonstrate table creation and  insertions with SQL
# importing module
import sqlite3
# connecting to the database
connection = sqlite3.connect ("Academy.db")
# cursor
cursor = connection.cursor()
```

மேற்கண்ட எடுத்துக்காட்டில், "Academy" என்ற பெயரில் தரவுத்தளம் உருவாக்கப்பட்டுள்ளது. இது SQL-ன் "CREATE DATABASE Academy;" என்ற கட்டளைக்கு இணையானது. அனுப்புவது. "sqlite3.connect ('Academy.db')" என்ற கூற்றை அழைக்கும் போது ஏற்கனவே உருவாக்கிய தரவுத்தளத்தைத் திறக்கும்.

15.3.1 அட்டவணையை உருவாக்குதல்

ஒரு வெற்று தரவுத்தளத்தை உருவாக்கிய பிறகு, முடிந்த வரை ஒன்று அல்லது அதற்கு மேற்பட்ட அட்டவணைகளைச் சேர்க்கலாம். அதற்கான கட்டளை அமைப்பு

CREATE TABLE Student (

RollnoINTEGER, SnameVARCHAR(20), GradeCHAR(1), gender CHAR(1),

Chapter 15.indd 296 30-01-2020 20:58:04

Average float(5.2), birth_date DATE, PRIMARY KEY (Rollno));

மேற்காண் இந்த முறையில் SQL கட்டளைகளில் செய்யப்படுவதைப் போலவே நேரடியாக பைத்தானில் செய்யலாம். SQL அல்லது SQLiteக்கு கட்டளைகளை அனுப்ப cursor பொருள்தேவை. பொதுவாக SQL மற்றும் தரவுத்தளத்திலுள்ள cursor என்பது, தரவுத்தளப் பதிவுகளின் மீது செயல்படும் ஒரு கட்டுப்பாட்டு அமைப்பாகும்.

குறிப்பு

SQL-ன் அனைத்து கட்டளைகளையும் செயல்படுத்த cursor பயன்படுகிறது.

இணைப்புக்கான cursor( ) வழிமுறையை அழைத்து cursor பொருளை உருவாக்கலாம்.

Cursor விடைத்தொகுப்பில் இருந்து பதிவுகளை எடுக்கப்பயன்படுகிறது.

பைத்தானில் SQL கட்டளைகளை மூன்று மேற்கொள் அடைப்புக் குறிகளுக்குள் சரமாக வரையறுக்க வேண்டும்.

எடுத்தக்காட்டு 15.3.1

```
import sqlite3
sql_command = """
CREATE TABLE Student (
Rollno INTEGER PRIMARY KEY ,
Sname VARCHAR(20),
Grade CHAR(1),
gender CHAR(1),
Average DECIMAL(5,2),
birth_date DATE);"""
```

மேலேயுள்ள எடுத்துக்காட்டில்,

Emp_no என்ற புலம் "INTEGER PRIMARY KEY" (முழு எண் முதன்மைத் திறவுகோல்) என வரையறுக்கப்பட்டுள்ளது. SQLite3 ல் இது போல் பெயரிடப்பட்ட நெடுவரிசையானது தானாகவே அதிகரித்துக் கொள்ளும்.

வேறு விதமாக கூற வேண்டுமெனில், அட்டவணையில் உள்ள ஒரு நெடுவரிசை INTEGER PRIMARY KEY, என்று அறிவிக்கப்பட்டு, எப்பொழுதெல்லாம் NULL என்ற மதிப்பு உள்ளீடாகப் பயன்படுத்தப்படுகிறதோ, அந்த NULL மதிப்பு தானாகவே அந்த நெடுவரிசையில் இதுவரை பயன்படுத்தப்பட்ட மிக உயர்ந்த மதிப்பைவிட ஒன்று மிகுந்து முழு எண்ணாக இருக்கும். வெற்று அட்டவணை எனில் 1 என்ற மதிப்பு பயன்படுத்தப்படும்.

15.3.2 பதிவுகளைச் சேர்த்தல் (Adding Records)

இப்பொழுது அட்டவணையுடன் கூடிய ஒரு தரவுத்தளம் எந்த தரவுகளும் இல்லாமல் உள்ளது. "INSERT" கட்டளையை SQLite ல் அனுப்புவதன் மூலம் அட்டவணையில் தரவுகளை உள்ளிடலாம். execute() செயற்கூறு கொடுக்கப்பட்ட SQL கட்டளையை செயல்டுத்தும்

எடுத்தக்காட்டு 15.3.2 -1

```
import sqlite3
connection = sqlite3.connect ("Academy.db")
cursor = connection.cursor()
# delete
cursor.execute ("""DROP TABLE Student;""")
sql_command = """
CREATE TABLE Student (
Rollno INTEGER PRIMARY KEY , Sname VARCHAR(20), Grade CHAR(1),
gender CHAR(1), Average DECIMAL (5, 2), birth_date DATE);"""
cursor.execute(sql_command)
    sql_command = """INSERT INTO Student (Rollno, Sname, Grade, gender, Average, birth_date) VALUES (NULL, "Akshay", "B", "M","87.8", "2001-12-12");""" cursor.execute(sql_command)
    sql_command = """INSERT INTO Student (Rollno, Sname, Grade, gender, Average, birth_date) VALUES (NULL, "Aravind", "A", "M","92.50","2000-08-17");""" cursor.execute(sql_command)
# never forget this, if you want the changes to be saved:
connection.commit()
connection.close()
print("STUDENT TABLE CREATED")
```

வெளியீடு

```
STUDENT TABLE CREATED
```

பின்வரும் எடுத்துக்காட்டு 15.3.2 ஒரு முழுமையான உதாரணமாகும். இந்த நிரலை இயக்க "Academy.db" கோப்பை வேண்டும் அல்லது SQL கட்டளையில் உள்ள DROP TABLE என்ற வரியின் குறிப்புரையாக்குக.

பெரும்பாலான நிகழ்வுகளில் கருத்தியலாக ஒரு SQL அட்டவணையில் தரவுகளை இடையில் செருக முடியாது. Insert கூற்றைப்பயன்படுத்தி, சில பைத்தான் தரவு இனங்களில் உள்ள பலத்தரவுகளை உள்ளிடலாம்.

பின்வரும் எடுத்துக்காட்டில், Academy.db என்ற தரவுத்தளம் மற்றும் Student அட்டவணை ஏற்கனவே உள்ளது எனக்கொள்க. நபர்களின் தரவுகளைக்கொண்ட List-யைINSERT கூற்றைப் பயன்படுத்தி உள்ளிடப்பட்டுள்ளது.

Chapter 15.indd 298 30-01-2020 20:58:05

எடுத்தக்காட்டு 15.3.2-2

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
student_data = [("BASKAR", "C", "M","75.2","1998-05-17"),
    ("SAJINI", "A", "F","95.6","2002-11-01"),
    ("VARUN", "B", "M","80.6","2001-03-14"),
    ("PRIYA", "A", "F","98.6","2002-01-01"),
    ("TARUN", "D", "M","62.3","1999-02-01") ]
for p in student_data:
    format_str = """INSERT INTO Student (Rollno, Sname, Grade, gender,Average, birth_
    date)  VALUES (NULL,"{name}", "{gr}", "{gender}","{avg}","{birthdate}");"""
    sql_command=format_str.format(name=p[0],gr=p[1],gender=p[2],avg=p[3],
    birthdate = p[4])
cursor.execute(sql_command)
connection.commit()
connection.close()
print("RECORDS ADDED TO STUDENT TABLE ")
```

வெளியீடு

```
RECORDS ADDED TO STUDENT TABLE
```

15.4 பைத்தானை பயன்படுத்தி SQL வினவல்

இப்பொழுது student அட்டவணையில் வினவலைப் பயன்படுத்துதலைப்பற்றி ,0கற்கலாம். பதிவுகளைச் சேர்ப்பதைப் போல அவற்றை திரும்ப பெறுவதும் மிக எளிது. Execute செயற்கூறு SQL கட்டளைகளை பயன்படுத்தி அனைத்து தரவுகளையும் அட்டவணையில் இருந்து பெறுகிறது.

15.4.1 SELECT வினவல்

SELECT கூற்று **SQL**-ல் மிக அதிகமாக பயன்படும் கூற்று ஆகும். இந்த கூற்று தரவுத் தளத்திலிருந்து தரவை பெற பயன்படுத்தப்படுகிறது. இந்த கூற்றுக்கான கட்டளை அமைப்பு **"Select * from table_name"** அட்டவணையின் அனைத்து தரவுகளையும் பட்டியலாக ஒரு பொருளில் பெற முடியும்.

"sql_Academy_query.py" என்ற சேமிக்கப்பட்ட நிரலை இயக்கினால் மெய்யான தரவைப் பொறுத்து பின்வரும் வெளியீடு தோன்றும்.

பைதான் கோப்பு உருவாக்கப்படும் கோப்புறையில் தான் தரவுத்தள கோப்புகளும் உருவாக்கப்படும் என்பதை கவனத்தில் கொள்ள வேண்டும். கோப்பின் பாதையை மாற்ற விரும்பினால் கோப்பை திறக்கும் போது மாற்றம் செய்ய வேண்டும்.

எடுத்தக்காட்டு 15.4.1-1

```
#save the file as "sql_Academy_query.py"
# cursor object
crsr = connection.cursor()
# execute the command to fetch all the data from the table Student
crsr.execute("SELECT * FROM Student")
# store all the fetched data in the ans variable
ans= crsr.fetchall()
# loop to print all the data
for i in ans:
        print(i)
```

15.4.1.1 fetchall() பயன்படுத்தி அனைத்து பதிவுகளையும் காண்பித்தல்

fetchall() செயற்கூறு அனைத்து வரிசைகளையும் தரவுத்தள அட்டவணையில் இருந்து பெற பயன்படுகிறது.

எடுத்தக்காட்டு 15.4.1.1-1

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT * FROM student")
print("fetchall:")
result = cursor.fetchall()
for r in result:
    print(r)
```

வெளியீடு

```
fetchall:
(1, 'Akshay', 'B', 'M', 87.8, '2001-12-12')
(2, 'Aravind', 'A', 'M', 92.5, '2000-08-17')
(3, 'BASKAR', 'C', 'M', 75.2, '1998-05-17')
(4, 'SAJINI', 'A', 'F', 95.6, '2002-11-01')
(5, 'VARUN', 'B', 'M', 80.6, '2001-03-14')
(6, 'PRIYA', 'A', 'F', 98.6, '2002-01-01')
(7, 'TARUN', 'D', 'M', 62.3, '1999-02-01')
```

குறிப்பு

cursor.fetchall() -fetchall () செயற்கூறு அனைத்து வரிசைகளையும் பெற பயன்படுகிறது.

cursor.fetchone() - fetchone () செயற்கூறு வினவல் முடிவுத் தொகுதியின் உள்ளே உள்ள அடுத்த வரிசையைக் கொடுக்கும் (அல்லது) எந்த வரிசையும் இல்லை என்றால் None என்ற மதிப்பை விடையாகக் கொடுக்கும்.

cursor.fetchmany() செயற்கூறு முடிவுத் தொகுதியில் மீதம் உள்ள வரிசைகளின் எண்ணிக்கையைக் கொடுக்கும்.

15.4.1.2 fetchone()-னைப் பயன்படுத்தி ஒரு பதிவைக் காண்பித்தல்

fetchone() செயற்கூறு வினவல் முடிவுத் தொகுதியின் உள்ளே உள்ள அடுத்த வரிசையைக் கொடுக்கும் (அல்லது) எந்த வரிசையும் இல்லை என்றால் None என்ற மதிப்பை கொடுக்கும்.

எடுத்தக்காட்டு 15.4.1.2-1

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT * FROM student")
print("\nfetch one:")
res = cursor.fetchone()
print(res)
```

வெளியீடு

```
fetch one:
(1, 'Akshay', 'B', 'M', 87.8, '2001-12-12')
```

15.4.1.3 fetchone()-னைப் பயன்படுத்தி அனைத்து பதிவுகளையும் காண்பித்தல்

while மடக்கு மற்றும் fetchone() செயற்கூறு பயன்படுத்தி அட்டவணையில் உள்ள அனைத்து பதிவுகளையும் காண முடியும்.

எடுத்தக்காட்டு 15.4.1.3 -1

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT * FROM student")
print("fetching all records one by one:")
result = cursor.fetchone()
while result is not None:
    print(result)
    result = cursor.fetchone()
```

வெளியீடு

```
fetching all records one by one:
(1, 'Akshay', 'B', 'M', 87.8, '2001-12-12')
(2, 'Aravind', 'A', 'M', 92.5, '2000-08-17')
(3, 'BASKAR', 'C', 'M', 75.2, '1998-05-17')
(4, 'SAJINI', 'A', 'F', 95.6, '2002-11-01')
(5, 'VARUN', 'B', 'M', 80.6, '2001-03-14')
(6, 'PRIYA', 'A', 'F', 98.6, '2002-01-01')
(7, 'TARUN', 'D', 'M', 62.3, '1999-02-01')
```

15.4.1.4 fetchmany() பயன்படுத்தி பதிவுகளைக் காண்பித்தல்

குறிப்பிட்ட எண்ணிக்கையிலான பதிவுகளைக் காண்பிக்க fetchmany() செயற்கூறு பயன்படுகிறது. இந்த செயற்கூறு முடிவுத் தொகுதியில் மீதம் உள்ள வரிசைகளின் எண்ணிக்கையைக் கொடுக்கும்.

எடுத்தக்காட்டு 15.4.1.4-1: பயன்படுத்தி பதிவுகளின் பொருளடக்கத்தைக் காண்பிக்கும் நிரல்

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT * FROM student")
print("fetching first 3 records:")
result = cursor.fetchmany(3)
print(result)
```

வெளியீடு

```
fetching first 3 records:
[(1, 'Akshay', 'B', 'M', 87.8, '2001-12-12'), (2, 'Aravind', 'A', 'M', 92.5, '2000-08-17'), (3, 'BASKAR',
'C', 'M', 75.2, '1998-05-17')]
```

எடுத்தக்காட்டு 15.4.1.4-2: மடக்கு பயன்படுத்தாமல், பதிவுகளின் பொருளடக்கத்தை புதியவரியில் காண்பிக்கும் நிரல்

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT * FROM student")
print("fetching first 3 records:")
result = cursor.fetchmany(3)
print(*result,sep="\n")
```

வெளியீடு

```
fetching first 3 records:
(1, 'Akshay', 'B', 'M', 87.8, '2001-12-12')
(2, 'Aravind', 'A', 'M', 92.5, '2000-08-17')
(3, 'BASKAR', 'C', 'M', 75.2, '1998-05-17')
```

குறிப்பு

மடக்குகளைப் பயன்படுத்தாமல் இடைவெளிகளுடன் கூடிய உறுப்புகளின் பட்டியலை ஒரே வரிசையில் அச்சிடுவதற்கு குறியீடு =”\n” or sep=”, ” பயன்படுகிறது.

15.4.2 SQL-ல் துணைநிலை கூற்று (CLAUSES)

SQL வழங்கும் பலவகையான துணை நிலை கூற்றுகள் SELECT கூற்றுகளில் பயன்படுத்தப்படுகிறது. SQLite ல் ஏறக்குறைய அனைத்து வகை துணை நிலை கூற்றுகளும் செயல்படுகிறது. அதிகமாக பயன்படுத்தப்படும் துணை நிலை கூற்றுகள் பின்வருமாறு,

Chapter 15.indd 302 30-01-2020 20:58:05

- DISTINCT
- WHERE
- GROUP BY
- ORDER BY.
- HAVING

15.4.2.1 SQL DISTINCT துணைநிலை கூற்று

ஒரு குறிப்பிட்ட நெடுவரிசை அல்லது அட்டவணையில் உள்ள இரட்டிப்பு மதிப்புகளைத் தவிர்ப்பதற்காக DISTINCT துணை நிலை கூற்று பயன்படுகிறது. இந்த சிறப்பு சொல்லைப் பயன்படுத்தும் போது தனித்த மதிப்புகளை பெற முடியும்.

பின்வரும் எடுத்துக்காட்டில், “student table” என்ற அட்டவணையில் இருந்து மாணவர்கள் பெற்ற தரவரிசை காண்பிக்கப்பட்டுள்ளது.

எடுத்தக்காட்டு 15.4.2.1-1

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT DISTINCT (Grade) FROM student")
result = cursor.fetchall()
print(result)
```

வெளியீடு

```
[('B',), ('A',), ('C',), ('D',)]
```

மேற்கண்ட எடுத்துக்காட்டில் “distinct” என்ற சிறப்பு சொல்லைப் பயன்படுத்தவில்லை எனில் 4 பதிவுகளுக்குப் பதிலாக 7 பதிவுகள் காண்பிக்கப்பட்டிருக்கும். ஏனெனில், மூல அட்டவணையில் இரட்டிப்பு மதிப்புகளுடன் 7 பதிவுகள் உள்ளன.

15.4.2.2 SQL WHERE துணைநிலை கூற்று

குறிப்பிட்ட நிபந்தனைகளை நிறைவேற்றும் பதிவுகளை மட்டுமே பிரித்தெடுக்க WHERE துணைநிலை கூற்று பயன்படுகிறது. பின்வரும் எடுத்துக்காட்டில் “student table” தரவுத்தளத்தில் இருந்து மாணவர்களின் தரவரிசையை மட்டுமே பிரித்தெடுப்பதைக் காணலாம்.

Chapter 15.indd 303 30-01-2020 20:58:05

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT DISTINCT (Grade) FROM student where gender='M'")
result = cursor.fetchall()
print(*result,sep="\n")
```

வெளியீடு

```
('B',)
('A',)
('C',)
('D',)
```

15.4.2.3 SQL Group By துணை நிலை கூற்று

GROUP BY துணை நிலை கூற்றானது குறிப்பிட்ட பதிவுகளைச் சுருக்கமான வரிசைகளைக் கொண்ட குழுவாக சேர்க்கிறது. இது ஒவ்வொரு குழுவிற்கும் ஒரு பதிவை கொடுக்கிறது. இது மதிப்பீட்டு செயற்கூறுகள் (COUNT, MAX, MIN, SUM, AVG) விடைத்தொகுதிகளை ஒன்று அல்லது அதற்கு மேற்பட்ட நெடுவரிசை குழுக்களாக மாற்ற பயன்படுகிறது. பின்வரும் எடுத்துக்காட்டு student அட்டவணையில் இருந்து ஆண் மற்றும் பெண் எண்ணிக்கையைக் கணக்கிட்டு விடையைக் காண்பிக்கிறது.

எடுத்தக்காட்டு 15.4.2.3 -1

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT gender,count(gender) FROM student Group BY gender")
result = cursor.fetchall()
print(*result,sep="\n")
```

வெளியீடு

```
('F', 2)
('M', 5)
```

15.4.2.4 SQL ORDER BY துணை நிலைக் கூறு

ORDER BY துணை நிலைக்கூற்று குறிப்பிட்ட புலங்களில் உள்ள தரவுகளை வரிசையாக்க SELECT கூற்றுடன் சேர்ந்து பயன்படுகிறது. விடைத் தொகுதியை ஏறு வரிசை அல்லது இறங்கு வரிசையில் வரிசையாக்கம் செய்ய உதவுகிறது. பின்வரும் எடுத்துக்காட்டு, மாணவர் பெயர் மற்றும் வரிசை எண் ஆகியவற்றை அகர வரிசைப்படி காண்பிக்கிறது.

Chapter 15.indd 304 30-01-2020 20:58:05

எடுத்தக்காட்டு 15.4.2.4 -1

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT Rollno,sname  FROM student Order BY sname")
result = cursor.fetchall()
print(*result,sep="\n")
```

வெளியீடு

```
(1, 'Akshay')
(2, 'Aravind')
(3, 'BASKAR')
(6, 'PRIYA')
(4, 'SAJINI')
(7, 'TARUN')
(5, 'VARUN')
```

15.4.2.5 SQL HAVING துணை நிலைக்கூறு

குழு சார்புகளைப் பொறுத்து தரவுகளை வடிகட்ட HAVING துணை நிலைக்கூற்று பயன்படுகிறது. இது WHERE நிபந்தனை கூற்றை ஒத்ததாகும் ஆனால் குழு சார்புகளுடன் பயன்படுகிறது. குழு சார்புகளை WHERE துணை நிலைக் கூற்றில் பயன்படுத்த முடியாது. ஆனால், HAVING துணை நிலைக் கூற்றில் பயன்படுத்த முடியும்.

எடுத்தக்காட்டு 15.4.2.5 -1

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT GENDER,COUNT(GENDER) FROM Student GROUP BY
GENDER HAVING COUNT(GENDER)>3")
result = cursor.fetchall()
co = [i[0] for i in cursor.description]
print(co)
print(result)
```

வெளியீடு

```
['gender', 'COUNT(GENDER)']
[('M', 5)]
```

15.5 SQL AND, OR மற்றும் NOT செயற்குறிகள்

WHERE துணைநிலைக் கூற்று AND, OR, மற்றும் NOT செயற்குறிகளுடன் இணைந்து செயல்படும்.

AND மற்றும் OR செயற்குறிகள் பதிவுகளை ஒன்றிற்கு மேற்பட்ட நிபந்தனைகளைப் பொறுத்து வடிகட்ட பயன்படுகிறது. பின்வரும் எடுத்துக்காட்டு, student அட்டவணையில் இருந்து 'A' அல்லது 'B' தரவகை பெற்ற மாணவர்களைத் தவிர மற்ற மாணவர்களின் விவரங்களைக் காண்பிக்கிறது.

WHERE துணை நிலைக் கூற்று NOT செயற்குறியுடன் உள்ள எடுத்துக்காட்டு

எடுத்தக்காட்டு15.5 -1

```
import sqlite3

connection = sqlite3.connect("Academy.db")

cursor = connection.cursor()

cursor.execute("SELECT * FROM student where grade<>'A' and Grade<>'B'")

result = cursor.fetchall()

print(*result,sep="\n")
```

வெளியீடு

```
(3, 'BASKAR', 'C', 'M', 75.2, '1998-05-17')

(7, 'TARUN', 'D', 'M', 62.3, '1999-02-01')
```

WHERE துணை நிலைக்கூற்று AND செயற்குறியுடன் உள்ள எடுத்துக்காட்டு

பின்வரும் எடுத்துக்காட்டு, சராசரியாக 80 லிருந்து 90 விழுக்காடு (இரண்டு வரம்பையும் சேர்த்து) வாங்கிய மாணவர்களின் பெயர், பதிவு எண், அவர்களின் சராசரி ஆகியவற்றைக் காண்பிக்கிறது.

எடுத்தக்காட்டு 15.5 -2

```
import sqlite3

connection = sqlite3.connect("Academy.db")

cursor = connection.cursor()

cursor.execute("SELECT Rollno,Sname,Average  FROM student WHERE
(Average>=80 AND Average<=90)")

result = cursor.fetchall()

print(*result,sep="\n")
```

வெளியீடு

```
(1, 'Akshay', 87.8)

(5, 'VARUN', 80.6)
```

WHERE துணை நிலைக்கூற்று OR செயற்குறியுடன் உள்ள எடுத்துக்காட்டு

பின்வரும் எடுத்துக்காட்டில் 2001ல் பிறந்த மாணவர்களின் பெயர் மற்றும் தரவரிசை காண்பிக்கப்பட்டுள்ளது.

எடுத்தக்காட்டு 15.5 -3

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT Rollno,sname  FROM student WHERE (Average<60 OR Average>70)")
result = cursor.fetchall()
print(*result,sep="\n")
```

OUTPUT

```
(1, 'Akshay')
(2, 'Aravind')
(3, 'BASKAR')
(4, 'SAJINI')
(5, 'VARUN')
(6, 'PRIYA')
```

15.6 தேதி உள்ள நெடுவரிசையில் வினவல்

பின்வரும் எடுத்துக்காட்டில் 2001ல் பிறந்த மாணவர்களின் பெயர் மற்றும் தரவரிசை காண்பிக்கப்பட்டுள்ளது.

எடுத்தக்காட்டு 15.6 -1

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT Rollno,sname FROM student
WHERE(Birth_date>='2001-01-01' AND Birth_date<='2001-12-01')")
result = cursor.fetchall()
print(*result,sep="\n")
```

வெளியீடு

```
(5, 'VARUN')
```

15.7 மதிப்பீட்டுச் சார்புகள் (Aggregate Functions)

இந்த சார்புகள் நெடுவரிசையில் உள்ள மதிப்புகளைக் கொண்டு செயல்பாடுகளைச் செய்து ஒரே ஒரு மதிப்பை விடையாகக் கொடுக்கும்.

- COUNT()
- AVG()
- SUM()
- MAX()
- MIN()

Chapter 15.indd 307 30-01-2020 20:58:05

15.7.1 COUNT() சார்பு

WHERE துணை நிலை கூற்றில் குறிப்பிட்டுள்ள நிபந்தனையை நிறைவேற்றும், அட்டவணையில் உள்ள வரிசைகளின் எண்ணிக்கையை SQL COUNT() சார்பு விடையாகக் கொடுக்கும்.

எடுத்தக்காட்டு 15.7.1-1

எடுத்தக்காட்டு 1 : இந்த எடுத்துக்காட்டில் வரிசைகளின் எண்ணிக்கை கணக்கிடப்பட்டுள்ளது.

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT COUNT(*)  FROM student ")
result = cursor.fetchall()
print(result)
```

வெளியீடு

```
[(7,)]
```

எடுத்தக்காட்டு 15.7.1-2

எடுத்துக்காட்டு 2 : இந்த எடுத்துக்காட்டில் குறிப்பிட்ட நெடுவரிசையைக் கொண்ட பதிவுகளின் எண்ணிக்கை கணக்கிடப்பட்டுள்ளது.

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT COUNT(AVERAGE)  FROM student ")
result = cursor.fetchall()
print(result
```

வெளியீடு

```
[(7,)]
```

குறிப்பு

NULL மதிப்புகள் கணக்கிடப்படமாட்டாது. மேலே உள்ள எடுத்துக்காட்டில் ஒரு வேளை பதிவில் NULL மதிப்பு இருந்தால் வெளியீடு 6 என இருக்கும்.

15.7.2AVG():

பின்வரும் பைதான் நிரலில் உள்ள கூற்று அனைத்து மாணவர்களின் சராசரி மதிப்பெண்ணைக் கண்டு பிடிக்கிறது.

Chapter 15.indd 308 30-01-2020 20:58:05

எடுத்தக்காட்டு 15.7.2-1

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT AVG(AVERAGE)  FROM student ")
result = cursor.fetchall()
print(result)
```

வெளியீடு

```
[(84.65714285714286,)]
```

குறிப்பு

NULL மதிப்புகள் கணக்கிடப்படமாட்டாது.

15.7.3 SUM():

பின்வரும் பைதான் நிரலில் உள்ள SQL கூற்று சராசரி புலத்தில் உள்ள சராசரிகளின் கூட்டுத் தொகையைக் கணக்கிடுகிறது.

எடுத்தக்காட்டு 15.7.1-3

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT SUM(AVERAGE)  FROM student ")
result = cursor.fetchall()
print(result)
```

வெளியீடு

```
[(592.6,)]
```

குறிப்பு

NULL மதிப்புகள் கணக்கிடப்படமாட்டாது.

15.7.4 MAX() மற்றும் MIN() செயற்கூறுகள்

The MAX() சார்பு தேர்ந்தெடுக்கப்பட்ட நெடுவரிசையில் உள்ள மிகப்பெரிய மதிப்பை விடையாகக் கொடுக்கும்.

The MIN() சார்பு தேர்ந்தெடுக்கப்பட்ட நெடுவரிசையில் உள்ள மிகச்சிறிய மதிப்பை விடையாகக் கொடுக்கும் .

சராசரி மதிப்பைக் கொண்ட மாணவரின் பெயர்களைக் காண்பிக்கிறது.

எடுத்தக்காட்டு 15.7.4-1

```
import sqlite3
connection = sqlite3.connect("Organization.db")
cursor = connection.cursor()
print("Displaying the name of the Highest Average")
cursor.execute("SELECT sname,max(AVERAGE)  FROM student ")
result = cursor.fetchall()
print(result)
print("Displaying the name of the Least Average")
cursor.execute("SELECT sname,min(AVERAGE)  FROM student ")
result = cursor.fetchall()
print(result)
```

வெளியீடு

```
Displaying the name of the Highest Average
[('PRIYA', 98.6)]
Displaying the name of the Least Average
[('TARUN', 62.3)]
```

15.8 பதிவுகளைப் புதுப்பித்தல்

பைதான் ஸ்கிரிப்ட் மூலமாக பதிவுகளைப் புதுப்பிக்க முடியும். பின்வரும் எடுத்துக்காட்டில் "student" அட்டவணையில் "Priya" என்ற பெயர் "Priyanka" என புதுப்பிக்கப்படுகிறது.

எடுத்தக்காட்டு 15.8 -1

```
# code for update operation
import sqlite3
# database name to be passed as parameter
conn = sqlite3.connect("Academy.db")
# update the student record
conn.execute("UPDATE Student SET sname ='Priyanka' where Rollno='6' ")
conn.commit()
print ("Total number of rows updated :", conn.total_changes)
cursor = conn.execute("SELECT * FROM Student")
for row in cursor:
 print (row)
conn.close()
```

வெளியீடு

```
Total number of rows updated : 1
(1, 'Akshay', 'B', 'M', 87.8, '2001-12-12')
(2, 'Aravind', 'A', 'M', 92.5, '2000-08-17')
(3, 'BASKAR', 'C', 'M', 75.2, '1998-05-17')
(4, 'SAJINI', 'A', 'F', 95.6, '2002-11-01')
(5, 'VARUN', 'B', 'M', 80.6, '2001-03-14')
(6, 'Priyanka', 'A', 'F', 98.6, '2002-01-01')
(7, 'TARUN', 'D', 'M', 62.3, '1999-02-01')
```

குறிப்பு

இந்தப் பாடப்பகுதி முழுவதும் SQL வினவல்களை விளக்குவதற்காக student என்ற தரவு அட்டவணை எடுத்துக்காட்டாக பயன்படுத்தப்பட்டுள்ளது என்பதை நினைவில் கொள்க. எடுத்துக்காட்டு 15.3.2-1. பதிவுகள் கொண்ட student தரவு அட்டவணை பயன்படுத்தப்பட்டுள்ளது.

15.9 நீக்குதல் செயல்பாடு

SQL கட்டளைகளைக் கொண்டு பதிவுகளை நீக்குவது போல பைதானிலும் நீக்க முடியும். பின்வரும் எடுத்துக்காட்டில் “student ” அட்டவணையில் Rollno 2 புலத்தில் உள்ள பொருளடக்கத்தை நீக்குவது காட்டப்பட்டுள்ளது.

எடுத்தக்காட்டு 15.9-1

```
# code for delete operation
import sqlite3
  # database name to be passed as parameter
conn = sqlite3.connect("Academy.db")
# delete student record from database
conn.execute("DELETE from Student where Rollno='2'")
conn.commit()
print("Total number of rows deleted :", conn.total_changes)
cursor =conn.execute("SELECT * FROM Student")
for row in cursor:
        print(row)
conn.close()
```

வெளியீடு

```
Total number of rows deleted : 1
(1, 'Akshay', 'B', 'M', 87.8, '2001-12-12')
(3, 'BASKAR', 'C', 'M', 75.2, '1998-05-17')
(4, 'SAJINI', 'A', 'F', 95.6, '2002-11-01')
(5, 'VARUN', 'B', 'M', 80.6, '2001-03-14')
(6, 'Priyanka', 'A', 'F', 98.6, '2002-01-01')
(7, 'TARUN', 'D', 'M', 62.3, '1999-02-01')
```

15.10 பயனரால் உள்ளிடப்படும் தரவு

பின்வரும் எடுத்துக்காட்டில் இயங்கு நேரத்தில் பைதான் input() கட்டளையைப் பயன்படுத்தி தரவுகளைப் பெற்று அதை "Person" என்ற அட்டவணையில் எழுதுதல்

எடுத்தக்காட்டு 15.10 -1

```
# code for executing query using input data
import sqlite3
# creates a database in RAM
con =sqlite3.connect("Academy.db")
cur =con.cursor()
cur.execute("DROP Table person")
cur.execute("create table person (name, age, id)")
print("Enter 5 students names:")
who =[input() for i in range(5)]
print("Enter their ages respectively:")
age =[int(input()) for i in range(5)]
print("Enter their ids respectively:")
p_id =[int(input())for i in range(5)]
n =len(who)
for i in range(n):
        # This is the q-mark style:
        cur.execute("insert into person values (?, ?, ?)", (who[i], age[i], p_id[i]))
# And this is the named style:
cur.execute("select * from person")
# Fetches all entries from table
print("Displaying All the Records From Person Table")
print (*cur.fetchall(), sep='\n' )
```


Chapter 15.indd 312 30-01-2020 20:58:05

வெளியீடு

```
Enter 5 students names:
        RAM
        KEERTHANA
        KRISHNA
        HARISH
        GIRISH
Enter their ages respectively:
        28
        12
        21
        18
        16
Enter their ids respectively:
        1
        2
        3
        4
        5
Displaying All the Records From Person Table
        ('RAM', 28, 1)
        ('KEERTHANA', 12, 2)
        ('KRISHNA', 21, 3)
        ('HARISH', 18, 4)
        ('GIRISH', 16, 5)
```

ஏற்கனவே உள்ள "student" என்ற அட்டவணையில் Name புலத்தில் சில மாற்றங்களைச் செய்து பதிவுகளைச் சேர்க்கலாம். இதை செய்ய Create table கூற்றைக் குறிப்புரையில் (Comment) எழுதவும்.

குறிப்பு

Execute (sql[, parameters]) :- இது ஒற்றை SQL கூற்றை இயக்குகிறது. SQL கூற்று அளபுருக்களை கொண்டிருக்கலாம் (அதாவது நிலையுருக்களுக்குப் பதிலாக இடநிரப்பிகள்). sqlite3 இரண்டு வகையான இடநிரப்பிகளை கொண்டுள்ளது. அவை, கேள்விக் குறிகள் (?) மற்றும் பெயரிடப்பட்ட இடநிரப்பிகள் (:name). எ.கா who= :who ,age=:age

15.11 பல அட்டவணைகளைப் பயன்படுத்தி வினவல்

ஒன்றுக்கு மேற்பட்ட அட்டவணைகளைச் சேர்த்து பைத்தானில் வினவலை மேற்கொள்ள முடியும். பின்வரும் எடுத்துக்காட்டில் மாணவர்களின் Rollno, Duty, Age ஆகிய விவரங்களைக் கொண்ட "Appointments" அட்டவணையை உருவாக்குவோம். "student" மற்றும் "Appointments" அட்டவணைகளை ஒன்றிணைத்து நெடுவரிசையின் தலைப்புகளுடன் கூடிய இறுதி அட்டவணையைக் காணலாம்.

எடுத்தக்காட்டு 15.11-1

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("""DROP TABLE Appointment;""")
sql_command = """
CREATE TABLE Appointment(rollnointprimarkey,Dutyvarchar(10),age int)"""
cursor.execute(sql_command)
sql_command = """INSERT INTO Appointment (Rollno,Duty ,age )
        VALUES ("1", "Prefect", "17");"""
cursor.execute(sql_command)
sql_command = """INSERT INTO Appointment (Rollno, Duty, age)
        VALUES ("2", "Secretary", "16");"""
cursor.execute(sql_command)
#  never forget this, if you want the changes to be saved:
connection.commit()
cursor.execute("SELECT student.rollno,student.sname,Appointment.
      Duty,Appointment.Age FROM student,Appointment where student.
      rollno=Appointment.rollno")
#print (cursor.description) to display the field names of the table
co = [i[0] for i in cursor.description]
print(co)
#  Field informations can be read from cursor.description.
result = cursor.fetchall()
for r in result:
        print(r)
```

வெளியீடு

```
['Rollno', 'Sname', 'Duty', 'age']
(1, 'Akshay', 'Prefect', 17)
(2, 'Aravind', 'Secretary', 16)
```

குறிப்பு

cursor.description ஒவ்வொரு நெடுவரிசையின் தலைப்புகளின் விவரங்களைக் கொண்டிருக்கும். இது பதிவாக சேமிக்கப்படும். முதல் பதிவு நெடுவரிசையின் பெயர்களைக் குறிக்கும் (index=0) இதை பயன்படுத்தி அட்டவணையின் புலங்களின் பெயர்களைக் காண்பிக்க முடியும்.

15.12 வினவலை CSV கோப்புடன் ஒருங்கிணைத்தல்

CSV கோப்பிலும் வினவலின் விடையைச் சேமிக்கலாம். வினவலின் வெளியீட்டை அட்டவணையில் காண்பிப்பதற்கு பயன்படும். பின்வரும் எடுத்துக்காட்டில் student அட்டவணையில் உள்ள gender நெடுவரிசையைப் பயன்படுத்தி இறங்கு வரிசையில் வரிசையாக்கம் செய்து பிறகு பதிவுகளை அகரவரிசைப்படி அமைக்கப்பட்டுள்ளது. இந்த வினவலின் வெளியீட்டை "SQL.CSV" என்னும் CSV கோப்பில் எழுதலாம். இந்த கோப்பினை MS-Excel மூலம் திறந்து முடிவுகளை காணலாம் அல்லது ஸ்கிரிப்ட்டைப் பயன்படுத்தியும் CSV கோப்புகளைத் திறந்து முடிவுகளைக் காணலாம்.

Chapter 15.indd 314 30-01-2020 20:58:05

எடுத்தக்காட்டு 15.12 -1

```
import sqlite3
import io                          # to access replace()
import csv
# CREATING CSV FILE
d=open('c:/pyprg/sql.csv','w')
c=csv.writer(d)
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
# a=Connection.cursor()
cursor.execute("SELECT * FROM student ORDER BY GENDER DESC,SNAME")
# WRITING THE COLUMN HEADING
co = [i[0] for i in cursor.description]
c.writerow(co)
data=cursor.fetchall()
for item in data:
        c.writerow(item)
d.close()
# Reading the CSV File
# replace() is used to eliminate the newline character at the end of each row
with open('c:/pyprg/sql.csv', "r", newline=None) as fd:
        for line in fd:
                line = line.replace("\n", "")
                print(line)
cursor.close()
connection.close()
```

வெளியீடு

```
Rollno,Sname,Grade,gender,Average,birth_date
1, Akshay, B, M, 87.8, 2001-12-12
2, Aravind, A, M, 92.5, 2000-08-17
3, BASKAR, C, M, 75.2, 1998-05-17
7, TARUN, D, M, 62.3, 1999-02-01
5, VARUN, B, M, 80.6, 2001-03-14
6, PRIYA, A, F, 98.6, 2002-01-01
4, SAJINI, A, F, 95.6, 2002-11-01
```

இதுவரை, பைத்தான் நிரல் எடுத்துக்காட்டுகளில் SQL கட்டளைகளைப் பயன்படுத்துவதைப் பார்த்தோம். ஏறக்குறைய அனைத்து SQL கட்டளைகளையும் பயன்படுத்துவதைப் பார்த்தோம். ஏறக்குறைய அனைத்து SQL கட்டளைகளும் பைதான் SQLite கூறுகளைப் பயன்படுத்தி நிறைவேற்றப்பட்டுள்ளது. SQL பாடத்தில் விவாதித்த பிற கட்டளைகளையும் முயற்சிக்கவும்.

எடுத்துக்காட்டு 15.12 -2 கோப்பினை ("sqlexcel.csv") MS-Excel மூலம் திறந்து அதனுடைய வெளியீட்டை பார்க்கவும். (எடுத்துக்காட்டு: 15.12.1 Script நிரலை ஒத்ததாகும்)

```
import sqlite3
import io                                    #to access replace()
import csv
# database name to be passed as parameter
conn = sqlite3.connect("Academy.db")
print("Content of the table before sorting and writing in CSV file")
cursor = conn.execute("SELECT * FROM Student")
for row in cursor:
 print (row)
#  CREATING CSV FILE
d=open('c:\\pyprg\\sqlexcel.csv','w')
c=csv.writer(d)
cursor = conn.cursor()
cursor.execute("SELECT *  FROM student ORDER BY GENDER DESC,SNAME")
#WRITING THE COLUMN HEADING
co = [i[0] for i in cursor.description]
c.writerow(co)
data=cursor.fetchall()
for item in data:
        c.writerow(item)
d.close()
print("sqlexcel.csv File is created open by visiting c:\\pyprg\\sqlexcel.csv")
conn.close()
```

வெளியீடு

```
Content of the table before sorting and writing in CSV file
(1, 'Akshay', 'B', 'M', 87.8, '2001-12-12')
(2, 'Aravind', 'A', 'M', 92.5, '2000-08-17')
(3, 'BASKAR', 'C', 'M', 75.2, '1998-05-17')
(4, 'SAJINI', 'A', 'F', 95.6, '2002-11-01')
(5, 'VARUN', 'B', 'M', 80.6, '2001-03-14')
(6, 'Priyanka', 'A', 'F', 98.6, '2002-01-01')
(7, 'TARUN', 'D', 'M', 62.3, '1999-02-01')
sqlexcel.csv File is created open by visiting c:\\pyprg\\sqlexcel.csv
```


Chapter 15.indd 316 30-01-2020 20:58:05

OUTPUT THROUGH EXCEL

	A	B	C	D	E	F
1	Rollno	Sname	Grade	gender	Average	birth_date
2						
3	1	Akshay	B	M	87.8	2001-12-12
4						
5	2	Aravind	A	M	92.5	2000-08-17
6						
7	3	BASKAR	C	M	75.2	1998-05-17
8						
9	7	TARUN	D	M	62.3	1999-02-01
10						
11	5	VARUN	B	M	80.6	2001-03-14
12						
13	6	PRIYA	A	F	98.6	2002-01-01
14						
15	4	SAJINI	A	F	95.6	200-11-01

15.3 அட்டவணை பட்டியல்

தரவுத்தளத்தில் உருவாக்கப்பட்ட அட்டவணைகளின் பட்டியல்களை காண்பிப்பதற்கு, பின்வரும் நிரலை (எ.கா. 15.3.1) பயன்படுத்தலாம்.

எடுத்துக்காட்டு 15.3 – 1

```
import sqlite3
con = sqlite3.connect('Academy.db')
cursor = con.cursor()
cursor.execute("SELECT name FROM sqlite_master WHERE type='table';")
print(cursor.fetchall())
```

வெளியீடு

```
[('Student',), ('Appointment',), ('Person',)]
```

மேற்கண்ட நிரல் 'Academy.db' என்ற தரவுத்தளத்தில் உருவாக்கப்பட்ட அனைத்து அட்டவணைகளின் பெயர்களை திரையில் காண்பிக்கும். நமது தரவுதள அட்டவணைகளின் முக்கிய தகவல்களை முதன்மை அட்டவணை கொண்டிருக்கும். sqlite_master எனப்படும்.

இதுவரை நாம் பைத்தான் நிரல் எடுத்துக்காட்டுகளில் SQL கட்டளைகளை பயன்படுத்துவதைப் பார்த்தோம். ஏறக்குறைய அனைத்து SRL கட்டளை பயன்படுத்துவதைப் பார்த்தோம். ஏறக்குறைய அனைத்து SQL கட்டளை பைத்தான் SQLite கூறுகளைப் பயன்படுத்தி நிறைவேற்றப்பட்டுள்ளது. SQL பாடத்தில் விவாதித்த பிற கட்டளைகளையும் முயற்சிக்கவும்.

நினைவில் கொள்க

- ஒரு தரவுத்தளம் என்பது ஒருங்கிணைக்கப்பட்ட தரவுகளின் தொகுப்பாகும்.
- தரவுத்தளத்தின் பயனர்களாக மனிதர்கள், பிற நிரல்கள் அல்லது பயன்பாடுகள் இருக்கலாம்.
- எளிய உறவுநிலை தரவுத்தள அமைப்பான இது தரவுகளை முறையான தரவுக் கோப்புகளாக சேமித்து வைக்கும்.
- தரவுத்தளத்திலுள்ள cursor என்பது தரவுத்தள பதிவுகளின் மீது செயல்படும் ஒரு கட்டுப்பாட்டு அமைப்பாகும். SQL –ன் அனைத்து கட்டளைகளையும் செயல்படுத்த இது பயன்படுகிறது.
- அட்டவணையிலுள்ள தரவு ஒற்றை அல்லது இரட்டை மேற்கோள்குறியுடன் உள்ளதால், பைத்தானிலுள்ள SQL கட்டளைகள் மூன்று மேற்கோள் குறியினால் குறிக்கப்படும்.
- SQL-ல் மிகவும் பொதுவாக பயன்படுத்தக்கூடிய கூற்று Select ஆகும்.
- தரவுத்தளத்திலுள்ள அட்டவணையிலிருந்து தரவுகளைப் பெற SQL கூற்றான Select பயன்படுகிறது.
- GROUP BY துணைநிலை கூற்று குறிப்பிட்ட பதிவுகளைச் சுருக்கமான வரிசைகளைக் கொண்ட குழுவாக சேர்க்கிறது.
- ORDER BY துணை நிலைக்கூற்று SELECT கூற்றுடன் சேர்ந்து குறிப்பிட்ட புலங்களில் உள்ள தரவுகளை முறையாக வரிசையாக்கம் செய்ய பயன்படுகிறது.
- Having துணைநிலைக்கூற்று GROUP செயற்கூறின் அடிப்படையில் தரவுகளை வடிகட்ட பயன்படுகிறது.
- Where துணை நிலைக்கூற்று 'Group by' துணை நிலைக்கூற்றுடன் பயன்படுத்த முடியாது.
- WHERE துணைநிலைக்கூற்று AND, OR, மற்றும் NOT செயற்குறிகளுடன் இணைத்துப் பயன்படுத்தலாம்.
- ஒன்றிற்கு மேற்பட்ட நிபந்தனைகளின் அடிப்படையில் பதிவுகளை வடிகட்ட 'AND' மற்றும் 'OR' செயற்குறிகள் பயன்படுகின்றன.
- மதிப்பீட்டுச் சார்புகள் நெடுவரிசையில் உள்ள மதிப்புகளைக் கொண்டு செயல்பாடுகளைச் செய்து ஒரே ஒரு மதிப்பை விடையாகக் கொடுக்கும்.
- COUNT() சார்பு அட்டவணையிலுள்ள வரிசைகளின் எண்ணிக்கையை திருப்பி அனுப்பும்.
- AVG() சார்பு அட்டவணையிலுள்ள தேர்ந்தெடுக்கப்பட்ட புலத்தின் பதிவுகளின் சராசரியை / கணக்கிடப்பயன்படுகிறது.
- SUM() சார்பு அட்டவணையிலுள்ள தேர்ந்தெடுக்கப்பட்ட புலத்தின் பதிவுகளின் கூட்டுத் தொகையைக் கணக்கிடுகிறது.
- MAX() சார்பு தேர்ந்தெடுக்கப்பட்ட புலத்தின் பெரிய மதிப்பை திருப்பி அனுப்பும்.
- MIN() சார்பு தேர்ந்தெடுக்கப்பட்ட புலத்தின் சிறிய மதிப்பை திருப்பி அனுப்பும்
- sqlite_master என்பது முதன்மை அட்டவணையாகும். இது நமது தரவுத்தள அட்டவணைகளின் முக்கிய தகவல்களை கொண்டிருக்கும்.
- பைத்தானில், கோப்பின் பாதையை '/' அல்லது '\\' குறியீடு குறிக்கப்படுகிறது. எடுத்துக்காட்டாக, பாதையை 'c:/pyprg/sql.csv', அல்லது c:\\pyprg\\sql.csv' எனக் குறிப்பிடலாம்.

Chapter 15.indd 318 30-01-2020 20:58:06

செய்முறைப் பயிற்சி

1. பயனரிடமிருந்து விவரங்களைப் பெற்று, பைத்தானைப் பயன்படுத்தி CSV கோப்பாக பின்வரும அட்டவணையில் சேமிக்க ஊடாடும் நிரலை உருவாக்கவும்

   தரவுத்தளத்தின் பெயர்;- DB1

   அட்டவணை பெயர் : Customer

Cust_Id	Cust_Name	Address	Phone_no	City
C008	Sandeep	14/1 Pritam Pura	41206819	Delhi
C010	Anurag Basu	15A, Park Road	61281921	Kolkata
C012	Hrithik	7/2 Vasant Nagar	26121949	Delhi

2. பின்வரும் GAMES அட்டவணையை எடுத்துக் கொள்வோம். (i) லிருந்து (iv) வரையிலான SQL வினவல்களுக்கான வெளியீட்டை கொடுக்கும் பைத்தான் நிரலை எழுதுக.

   Table: GAMES

Gcode	Name	GameName	Number	PrizeMoney	ScheduleDate
101	Padmaja	Carom Board	2	5000	01-23-2014
102	Vidhya	Badminton	2	12000	12-12-2013
103	Guru	Table Tennis	4	8000	02-14-2014
105	Keerthana	Carom Board	2	9000	01-01-2014
108	Krishna	Table Tennis	4	25000	03-19-2014

(i) அனைத்து Gamename மற்றும் Gcodes-களும் Schedule date-ன் இறங்கு வரிசையில் பதிவுகளை திரையிடவும்.

(ii) எந்த விளையாட்டின் பரிசுத்தொகை 7000- விட அதிகமாக இருக்கிறதா அந்த விளையாட்டுகளின் விவரங்கள் அடங்கிய பதிவுகளைத் திரையிடவும்.

(iii) விளையாட்டு வீரர்களின் name மற்றும் gamename-ன் ஏறுவரிசையில் பதிவுகளை திரையிடவும்.

(iv) ஒவ்வொரு விளையாட்டு குழுவின் வீர்ருக்கான பரிசுத்தொகை அடங்கிய பதிவுகளை திரையிடவும் (புலம் 4-ல் கொடுக்கப்பட்டுள்ளதைப் போல்)

(v) GameName-ன் அடிப்படையில் அனைத்து பதிவுகளையும் திரையிடவும்.

மதிப்பீடு

C9XK3F

பகுதி–அ

சரியான விடையை தேர்ந்தெடுத்து எழுதுக (1 மதிப்பெண்)

1. பின்வரும் எது ஒருங்கிணைக்கப்பட்ட தரவுகளின் தொகுப்பாகும்?

   (அ) தரவுத்தளம் (ஆ) DBMS (இ) தகவல் (ஈ) பதிவுகள்

2. SQLite எந்த தரவுத்தள அமைப்பைச் சார்ந்தது?

   (அ) ஒற்றைக் கோப்பு தரவுத்தளம் (ஆ) உறவுநிலை தரவுத்தளம்

   (இ) படிநிலை தரவுத்தளம் (ஈ) பொருள்நோக்கு தரவுத்தளம்

3. பின்வரும் எந்த கட்டுப்பாட்டு அமைப்பு தரவுத்தளத்திலிருந்து பதிவுகளைப் பெற்றுத்தர பயன்படுகிறது?

   (A) சுட்டு (ஆ) திறவுகோல்

   (இ) Cursor (ஈ) செருகும் புள்ளி

4. பதிவுகளில் உள்ள மதிப்புகளில் செய்யப்படும் மாற்றங்களை சேமிக்கப் பயன்படும் கட்டளை எது?

   (அ) Save (ஆ) Save As (இ) Commit (ஈ) Oblige

5. சில செயல்பாடுகளை SQL கட்டளைகள் செய்வதற்கு பின்வரும் எது இயக்கப்படுகிறது?

   (அ) execute() (ஆ) key() (இ) cursor() (ஈ) run()

6. பின்வரும் எந்த சார்பு அட்டவணையிலுள்ள தேர்ந்தெடுக்கப்பட்ட புலத்தின் பதிவுகளின் சராசரியைக் கொடுக்கிறது?

   (அ) Add() (ஆ) SUM() (இ) AVG() (ஈ) AVERAGE()

7. எந்த செயற்கூறு தேர்ந்தெடுக்கப்பட்ட புலத்தின் பெரிய மதிப்பைத் திருப்பி அனுப்பும்

   (அ) MAX() (ஆ) LARGE()

   (இ) HIGH() (ஈ) MAXIMUM()

8. பின்வரும் எது முதன்மை அட்டவணை?

   (அ) sqlite_master (ஆ) sql_master

   (இ) main_master (ஈ) master_main

9. SQL-ல் மிகவும் பொதுவாக பயன்படுத்தப்படும் கூற்று எது?

   (அ) cursor (ஆ) select (இ) execute (ஈ) commit

10. பின்வரும் எந்த துணை நிலைக்கூற்று நகல்களைத் தவிர்க்கும்?

   (அ) Distinct (ஆ) Remove (இ) Where (ஈ) GroupBy

Chapter 15.indd 320 30-01-2020 20:58:06

பகுதி–ஆ

அனைத்து வினாக்களுக்கும் விடையளி (2 மதிப்பெண்)

1. தரவுத்தளத்தைப் பயன்படுத்தும் பயனர்களை குறிப்பிடவும்.
2. தரவுத்தளத்தை இணைக்க பயன்படும் முறைகள் யாவை? எடுத்துக்காட்டு தருக.
3. புலத்தை "INTEGER PRIMARY KEY" என அறிவிப்பதன் நன்மை என்ன?
4. அட்டவணையில் பதிவுகளை விரிவுப்படுத்துவதற்கான கட்டளையை எழுதுக. எடுத்துக்காட்டு தருக.
5. தரவுத்தள அட்டவணையிலிருந்து அனைத்து பதிவுகளையும் பெறுவதற்கான வழிமுறை எது?

பகுதி–இ

அனைத்து வினாக்களுக்கும் விடையளி (3 மதிப்பெண்)

1. SQLite என்றால் என்ன ? இதன் நன்மைகள் யாவை?
2. fetchone() மற்றும் fetchmany() வேறுபடுத்துக.
3. Where துணைநிலைக்கூற்றின் பயன் என்ன?where கூற்றைப் பயன்படுத்தி ஒரு பைத்தான் கூற்றை எழுதவும்.
4. பின்வரும் விவரங்களை படிக்கவும். அதன் அடிப்படையில் துறைவாரியாக பதிவுகளை திரையிட பைத்தான் ஸ்கிரிப்ட்டை எழுதவும்.

   தரவுத்தள பெயர் :- organization.db

   அட்டவணை பெயர் :- Employee

   புலங்கள் :- Eno, EmpName, Esal, Dept
5. பின்வரும் விவரங்களை படிக்கவும் அதன் அடிப்படையில் பதிவுகளை Eno இறங்குவரிசையில் திரையிட பைத்தான் ஸ்கிரிப்ட்டை எழுதவும்.

   தரவுத்தள பெயர் :- organization.db

   அட்டவணை பெயர் :- Employee

   புலங்கள் :- Eno, EmpName, Esal, Dept

பகுதி ஈ

அனைத்து வினாக்களுக்கும் விடையளி (5 மதிப்பெண்)

1. SQLite பற்றி விரிவாக எழுதவும். அதனை பயன்படுத்தும் படிநிலைகளை எழுதுக.
2. fetchmany() பயன்படுத்தி பின்வரும் அட்டவணையிலுள்ள அனைத்து பதிவுகளையும் திரையிடுவதற்கான பைத்தான் ஸ்கிரிப்ட்டை எழுதவும்

Icode	ItemName	Rate
1003	Scanner	10500
1004	Speaker	3000

1005	Printer	8000
1008	Monitor	15000
1010	Mouse	700

3. HAVING துணைநிலைக்கூற்றின் பயன் யாது? எடுத்துக்காட்டு தருக.

4. பின்வரும் குறிப்புகளைக் கொண்டு ITEM என்ற அட்டவணையை உருவாக்க பைத்தான் ஸ்கிரிப்ட்டை எழுதவும்.

   அட்டவணைக்கு ஒரு பதிவை சேர்க்கவும்.

   தரவுத்தளத்தின் பெயர் :- ABC

   அட்டவணையின் பெயர் :- Item

   நெடுவரிசையின் பெயர் மற்றும் விவரங்கள் :-

Icode	:-	integer and act as primary key
Item Name	:-	Character with length 25
Rate	:-	Integer
Record to be added	:-	1008, Monitor,15000

5. பின்வரும் supplier மற்றும் Item அட்டவணைகளை கவனித்து, (i) மற்றும் (ii) வினாக்களுக்கு பைத்தான் ஸ்கிரிப்ட்டை எழுதவும்.

SUPPLIER				
Suppno	**Name**	**City**	**Icode**	**SuppQty**
S001	Prasad	Delhi	1008	100
S002	Anu	Bangalore	1010	200
S003	Shahid	Bangalore	1008	175
S004	Akila	Hydrabad	1005	195
S005	Girish	Hydrabad	1003	25
S006	Shylaja	Chennai	1008	180
S007	Lavanya	Mumbai	1005	325

i) டெல்லியில் வசிக்காத மொத்த விற்பனையாளர்களின் Name,City மற்றும் Itemname களை திரையிடவும்.

ii) அகிலாவின் suppQty யில் உள்ள மதிப்போடு 40-யை அதிகரிக்கும்

References

1. *The Definitive Guide to SQLite by Michael Owens*
2. *Programming for Beginners: 2 Manuscripts: SQL & Python by Byron Francis*
3. *Tutorialspoint.com*

Chapter 15.indd 322 30-01-2020 20:58:06