

実践

Python によるデータベース入門

MySQL, MongoDB, CouchDB の基本操作から
アプリプログラミングまで

博士 (工学) 藤野 巖【著】

コロナ社

```
(base) C:\Users\fujino\python3user\MySQL>python ch5ex2.py
information_schema
jikanwaribase
mysql
performance_schema
salesbase
sampledb
sys
testbase
todoufukibase
```

実行結果

《実行結果の解説》

サーバーにあるデータベースの一覧が表示され、その中には sampledb が作成されていることが確認できる。

5.3 プログラム例

— テーブルの作成とデータ入力 —

例題 5.2 では、データベースを作成することができたので、ここでは、その続きとして、テーブルを作成して、そのテーブルにデータを入力するプログラム例を以下に示す。

【例題 5.3】 下記仕様要求を実現する Python プログラムを作成せよ。

仕様要求

- (1) MySQL サーバーの sampledb への接続を行う。
- (2) SQL 命令「テーブルを作成する」を実行する。
- (3) SQL 命令「テーブルにデータ入力する」を実行する。
- (4) SQL 命令「テーブルのデータを表示する」を実行する。
- (5) (4) の結果を画面に表示する。
- (6) サーバーにコミットする。
- (7) 接続を切断する。

この問題について、まず基本的な解答を解答 1 に示す。そしてそれをさらに改良したものを解答 2 に示す。

【解答 1】 基本的な解答。SQL 命令文を 1 行ずつ投入する。

```
1: # データベースにテーブル作成
2: import mysql.connector
3:
4: # コネクターを作成
5: config = {
6:     'host': 'localhost',
7:     'port': '3306',
8:     'user': 'root',
9:     'password': 'rootmysql',
10:    'database': 'sampledb'
11: }
12:
13: dbconnector = mysql.connector.connect(**config)
14:
15: # 接続できているかどうか確認
16: if dbconnector.is_connected():
17:     print('データベースへの接続が成功しました。')
18: else:
19:     print('データベースへの接続が失敗しました。')
20:     exit(1)
21:
22: # cursor オブジェクトの生成
23: cursor = dbconnector.cursor(buffered=True)
24:
25: # 属性 id, name, gender, age を持つテーブルを作成, データ入力, 問い
    合わせ
26: cursor.execute("DROP TABLE IF EXISTS memberlist;")
27: cursor.execute("CREATE TABLE memberlist ( mid INT, name
    CHAR(20), gender CHAR(10), age INT, PRIMARY KEY (mid) );")
28: cursor.execute("INSERT INTO memberlist VALUES ( 1, '田中花子',
    '女', 28);")
29: cursor.execute("INSERT INTO memberlist VALUES ( 2, '山田聡',
    '男', 32);")
30: cursor.execute("INSERT INTO memberlist VALUES ( 3, '佐藤美那子',
    '女', 21);")
31: cursor.execute("INSERT INTO memberlist VALUES ( 4, '鈴木博',
    '男', 25);")
32: # コミットする
33: dbconnector.commit()
34:
35: # 問い合わせ
36: cursor.execute("SELECT * FROM memberlist;")
37: # すべてのデータを取得, 表示
38: tuples = cursor.fetchall()
```

ソースコード

```
39: print("テーブル memberlist のデータ:")
40: for tpl in tuples:
41:     print(tpl)
42:
43: # 接続を切断
44: dbconnector.close()
```

[ファイル名: ch5ex3a.py]

《ソースコードの解説》

- 2: MySQL にアクセスするためのライブラリを読み込む。
- 4~10: MySQL サーバーに接続するための接続先とアカウント情報を記述する。ここでは、例題 5.2 で作成した `sampledb` を接続先に指定する。
- 13: 所定の接続情報を使って、MySQL サーバーに接続する。
- 16~20: 接続が成功したかどうか確認する。成功した場合は、成功のメッセージを表示する。失敗した場合は、失敗のメッセージを表示して終了する。
- 23: カーソルオブジェクトを作成する。
- 26: カーソルで SQL 命令文「`DROP TABLE IF EXISTS memberlist`」を実行する。テーブル `memberlist` が存在するならば、それをいったん削除する。
- 27: カーソルで SQL 命令文「`CREATE TABLE memberlist (mid INT, name CHAR(20), gender CHAR(10), age INT, PRIMARY KEY (mid));`」を実行する。テーブル `memberlist` を生成する。
- 28: カーソルで SQL 命令文「`INSERT INTO memberlist VALUES (1, '田中花子', '女', 28)`」を実行する。テーブル `memberlist` にデータを挿入する。
- 29~31: 28 行目と同じく、ほか 3 名のデータを挿入する。
- 33: MySQL サーバーに実際に変更を書き込む。
- 36: カーソルで SQL 命令文「`SELECT * FROM memberlist`」を実行する。テーブル内にあるデータを表示する。
- 38: カーソルから実行結果を取得して、`tuples` に代入する。
- 40~41: `tuples` の中身を表示する。
- 44: 接続を切断する。

```
(base) C:\Users\fujino\python3user\MySQL>python ch5ex3a.py
データベースへの接続が成功しました。
テーブル memberlist のデータ：
(1, '田中花子', '女', 28)
(2, '山田聡', '男', 32)
```

実行結果

```
(3, '佐藤美那子', '女', 21)
(4, '鈴木博', '男', 25)
```

《実行結果の解説》

実行結果にはテーブル memberlist の 4 件のデータが表示されている。問題の要求が実現されていることが確認できる。

【解答 2】 解答 1 の改良版。

改良点 1：複数行の SQL 命令文をまとめて投入する。

改良点 2：データを csv ファイルに保存して、そこから読み込んで、データベースのテーブルに書き込む。

```
1: # データベースにテーブル作成
2: # SQL 命令文をまとめて実行
3: # csv ファイルからデータを読み込む
4:
5: import mysql.connector
6: import csv
7:
8: # コネクターを作成
9: config = {
10:     'host': 'localhost',
11:     'port': '3306',
12:     'user': 'root',
13:     'password': 'rootmysql',
14:     'database': 'sampledb'
15: }
16: dbconnector = mysql.connector.connect(**config)
17:
18: # 接続できているかどうか確認
19: if dbconnector.is_connected():
20:     print('データベースへの接続が成功しました。')
21: else:
22:     print('データベースへの接続が失敗しました。')
23:     exit(1)
24:
25: # cursor オブジェクトの生成
26: cursor = dbconnector.cursor(buffered=True)
27:
28: # 属性 id, name, gender, age を持つテーブルを作成, データ入力
29: operation = (
```

ソースコード

```
30:     "DROP TABLE IF EXISTS newlist;"
31:     "CREATE TABLE newlist ( mid INT, name CHAR(20), gender
    CHAR(10), age INT, PRIMARY KEY (mid) ); "
32: )
33:
34: # SQL 命令文を実行
35: results = cursor.execute(operation, multi=True)
36: for res in results:
37:     print(res)
38:
39: operation = ("INSERT INTO newlist VALUES (%s, %s, %s, %s);")
40: seqs = []
41: with open('newlistdata.csv', encoding='utf-8') AS f:
42:     reader = csv.reader(f)
43:     for row in reader:
44:         tuple=(int(row[0]), row[1], row[2], int(row[3]))
45:         seqs.append(tuple)
46:
47: #executemany() で実行する
48: cursor.executemany(operation, seqs)
49: # コミットする
50: dbconnector.commit()
51:
52: # 問い合わせ
53: cursor.execute("SELECT * FROM newlist")
54: # 問い合わせ結果の全てのデータを取得、表示
55: tuples = cursor.fetchall()
56: print("テーブル newlist のデータ：")
57: for tpl in tuples:
58:     print(tpl)
59:
60: # 接続を切断
61: dbconnector.close()
```

[ファイル名: ch5ex3b.py]

```
1, 田中花子, 女, 28
2, 山田聡, 男, 32
3, 佐藤美那子, 女, 21
4, 鈴木博, 男, 25
```

データファイル

[ファイル名: newlistdata.csv]

《ソースコードの解説》

5: MySQL にアクセスするためのライブラリを読み込む。

84 5. Python による MySQL プログラム

- 6: csv ファイル用のライブラリを読み込む。
- 8~26: 解答 1 の 5~23 行と同じなので、ここでは省略する。
- 29~32: つぎにまとめて実行する SQL 命令文を用意して、operation に代入しておく。中身は、解答 1 の 26, 27 行で実行した SQL 命令文と同じ。ただし、テーブル名を newlist に変更した。
- 35: 用意された operation を実行して、その結果を results に代入する。
- 36~37: results の中身を表示する。
- 39: つぎに実行する SQL 命令文を用意する。今回の SQL 命令は以下の通りである。ただし、%s のところに、命令の実行時に、データが代入される。
INSERT INTO newlist VALUES (%s, %s, %s, %s);
- 40~45: ファイル newlistdata.csv からデータを読み込んで、INSERT 文を実行するためのデータ seqs を作成する。
- 48: 用意された operation を実行する。ここでは execute_many() 関数を呼び出して、データをまとめたリスト seqs を、操作 operation に渡して実行した。
- 50: MySQL サーバーに実際に変更を書き込む。
- 52~61: 解答 1 の 35~44 行と同じなので、ここでは省略する。

```
(base) C:\Users\fujino\python3user\MySQL>python ch5ex3b.py 実行結果
CMySQLCursorBuffered: DROP TABLE IF EXISTS newlist
CMySQLCursorBuffered: CREATE TABLE newlist (mid INT, name CHA...
テーブル newlist のデータ:
(1, '田中花子', '女', 28)
(2, '山田聡', '男', 32)
(3, '佐藤美那子', '女', 21)
(4, '鈴木博', '男', 25)
```

《実行結果の解説》

実行結果にはテーブル newlist の 4 件のデータが表示されている。問題の要求が実現されていることが確認できる。

このようなやり方を用いると、テーブルのデータを変更するとき、プログラムのソースコード変更することなく、データファイルのみの変更で対応できるようになる。

5.4 プログラム例

— 簡単な問い合わせ —

これまでに、Python のプログラムから、データベースの作成、テーブルの

索引

【あ行】	実体関連モデル	7, 94	データモデル	5
インスタンス	射影演算	38	テーブル	7
オブジェクト	集約関数	46	問い合わせ	28
	主キー	97	等結合	59
【か行】	情報	2	トップダウンアプローチ	94
概念スキーマ	スキーマ	9	トランザクション	11
概念設計	正規化	101	【な行】	
外部キー	正規形	101	内部スキーマ	10
外部スキーマ	設計問題	93	【は行】	
関係	選択演算	38	ビュー	188
関係演算	属性	6	物理設計	93
関係スキーマ	【た行】		複合条件	33
関係データベース	第1正規形	101	部分集合	6
関係データモデル	第2正規形	106	ボトムアップアプローチ	94
関数従属性	第3正規形	107	【ま行】	
関連	タブル	7	目的表	100
基本表	単純条件	28	【や行】	
グループ化	直積集合	6	予約語	18
交差結合	データ	1	【ら行】	
【さ行】	——の耐久性	12	列名	17
サービスの永続性	データベース	4	論理設計	93
3層スキーマ構造	データベース管理システム	4		
自然結合	データベースサーバー	4		
実体	データベースシステム	4		
	データベース本体	4		

【C, E】

Chen 記法	95
CouchDB	14
ER 図	95
ER モデル	7

【J, M, N】

JSON	134
Map	188
MongoDB	14
NoSQL	14

【R, S】

Reduce	188
REST	174
SQL	12

— 著者略歴 —

1991年 東海大学大学院工学研究科博士課程修了
博士（工学）
1994年 東海大学短期大学部講師
2001年 東海大学短期大学部助教授
2004年 イギリス・サウサンプトン大学客員教授
～05年
2007年 東海大学短期大学部教授
2008年 東海大学情報通信学部教授
現在に至る
2016年 フランス海軍アカデミー招聘研究員
～17年

実践 Python によるデータベース入門

—MySQL, MongoDB, CouchDB の基本操作からアプリプログラミングまで—

Perfect Practice An Introduction to Database with Python

—MySQL, MongoDB and CouchDB from Basic Operation to Application Programming—

© Iwao Fujino 2020

2020年8月28日 初版第1刷発行

★

検印省略

著者	藤野 巖
発行者	株式会社 コロナ社
印刷所	代表者 牛来真也
製本所	壮光舎印刷株式会社
	株式会社 グリーン

112-0011 東京都文京区千石 4-46-10

発行所 株式会社 コロナ社

CORONA PUBLISHING CO., LTD.

Tokyo Japan

振替00140-8-14844・電話(03)3941-3131(代)

ホームページ <https://www.coronasha.co.jp>

ISBN 978-4-339-02912-3 C3055 Printed in Japan

(松岡)



＜出版者著作権管理機構 委託出版物＞

本書の無断複製は著作権法上での例外を除き禁じられています。複製される場合は、そのつど事前に、出版者著作権管理機構（電話 03-5244-5088, FAX 03-5244-5089, e-mail: info@jcopy.or.jp）の許諾を得てください。

本書のコピー、スキャン、デジタル化等の無断複製・転載は著作権法上での例外を除き禁じられています。購入者以外の第三者による本書の電子データ化及び電子書籍化は、いかなる場合も認めていません。落丁・乱丁はお取替えいたします。