

## 【課題 22 : teacher\_db.py】

配布した teachers.html から、CS 教員の名前、詳細ページの URL、職位、専門分野を抽出し、テーブルに格納するコードを実装せよ。

採点の都合上、以下の要件を満たすこととする：

- (1) 読み込みファイル (teachers.html) と出力データベース (teachers.db) のパスをコマンドライン引数で指定する；
- (2) テーブル名は「cs」にする；
- (3) カラム (id, name, url, position, fields) を持つテーブルを CREATE する；
- (4) ただし、すでに「cs」という名前のテーブルが存在し OperationalError が発生した場合は、当該テーブルを DROP し、再度 CREATE する；
- (5) teachers.html の内容を cs テーブルに INSERT する；
- (6) 配布した confirm.py の出力結果が図 1 になるようにする；

```
PS C:\Users\admin\Desktop\ProA2> python lec10/kadai22/confirm.py lec10/kadai22/teachers.db
(38,)
```

```
(1, '大野 澄雄', 'https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1660', '教授', '情報システム、音声情報処理、デジタル信号処理、マン・マシン・インタフェース')
(2, '青木 輝勝', 'https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1732', '教授', '画像理解、画像処理、コンピュータグラフィックス')
(3, '石畑 宏明', 'https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1629', '教授', 'コンピュータアーキテクチャ・並列処理')
(4, 'Reijer Grimbergen', 'https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1661', '教授', '人工知能、認知科学、ゲームプログラミング')
(5, '佐藤 公則', 'https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1734', '教授', '画像認識、画像計測、バイオメトリクス')
(6, '瀬之口 潤輔', 'https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1736', '教授', '人工知能、機械学習モデルによる金融市場予測')
(7, '服部 聖彦', 'https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1737', '教授', '群知能、群ロボット、自律分散システム')
(8, '松下 宗一郎', 'https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1662', '教授', 'コンピュータエンタテインメント')
(9, '井上 亮文', 'https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1688', '准教授', 'ヒューマンコンピュータインタラクション、ネットワークセキュリティ')
(10, '岩下 志乃', 'https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1685', '准教授', '感性情報処理、自然言語処理、ユーザモデル')

(29, '山元 進', 'https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1632', '准教授', '電子物性、量子分子動力学、大規模数値解析')
(30, '金光 永徳', 'https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1628', '講師', '並列分散処理、タスクスケジューリング')
(31, '千葉 康生', 'https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1634', '講師', '超局所解析、代数解析、eラーニング')
(32, '堀野 康徳', 'https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1746', '講師', '知能情報、数理情報、情報教育')
(33, '董 然', 'https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1738', '助教', '認知科学、ヒューマンロボットインタラクション、コンピュータグラフィックス')
(34, '上ノ山 海沙', 'https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1747', '実験助手', 'バイオメトリクス、webアプリケーション')
(35, '梅本 暖', 'https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1744', '実験助手', 'バイオメトリクス、コンピュータエンタテインメント')
(36, '大作\u3000浩文', 'https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1739', '実験助手', '並列分散処理、P2Pネットワーク')
(37, '葛西\u3000義哉', 'https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1740', '実験助手', 'ソフトウェアテスト、プログラム理解支援')
(38, '榎内 美穂', 'https://www.teu.ac.jp/info/lab/teacher/cs/index.html?id=1745', '特任講師', 'キャリア教育、キャリアデザイン学、キャリアコンサルティング')
```

図 1 課題 22 の実行結果

## リスト 1 teacher.db.py

---

```
1 import sys
2 import re
3 import sqlite3
4
5 def read_html(file_path):
6     with open(file_path, "r", encoding="utf8") as rfo:
7         html = rfo.read().replace("\n", "")
8         pattern = re.compile(r"<a href=\"(/info/lab/teacher/cs/index.html?id=\d+).*?\">.*?<h2
9             >(.*?)</h2>.*?<h4>(.*?)</h4>.*?<p>(.*?)</p>")
10        match_lst = pattern.findall(html)
11        teachers = []
12        for i, match in enumerate(match_lst, 1):
13            name = match[1]
14            url = "https://www.teu.ac.jp"+match[0]
15            pos = match[2].split()[0]
16            field= match[3]
17            teachers.append((name, url, pos, field))
18        return teachers
19
20 if __name__ == "__main__":
21     teachers = read_html(sys.argv[1]) # lec10/data/teachers.html
22     db_path = sys.argv[2] # lec10/kadai22/teachers.db
```

---

## 【課題 23 : pokemon\_db.py】

配布した poke\_names.txt と base\_stats.txt の内容をテーブルに格納するコードを実装せよ。

採点の都合上、以下の要件を満たすこととする：

- (1) 読み込みファイル (poke\_names.txt, base\_stats.txt) と出力データベース (pokemon.db) のパスをコマンドライン引数で指定する；
- (2) テーブル名は「name\_stats」にする；
- (3) 最初に、カラム (id, name, types, evolvs) を持つテーブルを CREATE する；
- (4) ただし、すでに「name\_stats」という名前のテーブルが存在し OperationalError が発生した場合は、当該テーブルを DROP し、再度 CREATE する；
- (5) poke\_names.txt の内容を name\_stats テーブルにバルクインサートする；
- (6) カラム (hp, atk, dfs, sp\_atk, sp\_dfs, spd) を追加する；
- (7) base\_stats.txt の内容を name\_stats テーブルに反映させる；
- (8) 配布した confirm.py の出力結果が図 2 になるようにする；

```
PS C:\Users\admin\Desktop\ProA2> python lec10/kadai23/confirm.py lec10/kadai23/pokemon.db (251,)
```

```
(1, 'フシギダネ', 'くさ どく', 'フシギソウ', 45, 49, 49, 65, 65, 45)
(2, 'フシギソウ', 'くさ どく', 'フシギバナ', 60, 62, 63, 80, 80, 60)
(3, 'フシギバナ', 'くさ どく', '', 80, 82, 83, 100, 100, 80)
(4, 'ヒトカゲ', 'ほのお', 'リザード', 39, 52, 43, 60, 50, 65)
(5, 'リザード', 'ほのお', 'リザードン', 58, 64, 58, 80, 65, 80)
(6, 'リザードン', 'ほのお ひこう', '', 78, 84, 78, 109, 85, 100)
(7, 'ゼニガメ', 'みず', 'カメール', 44, 48, 65, 50, 64, 43)
(8, 'カメール', 'みず', 'カメックス', 59, 63, 80, 65, 80, 58)
(9, 'カメックス', 'みず', '', 79, 83, 100, 85, 105, 78)
(10, 'キャタピー', 'むし', 'トランセル', 45, 30, 35, 20, 20, 45)

(241, 'ミルタンク', 'ノーマル', '', 95, 80, 105, 40, 70, 100)
(242, 'ハピナス', 'ノーマル', '', 255, 10, 10, 75, 135, 55)
(243, 'ライコウ', 'でんき', '', 90, 85, 75, 115, 100, 115)
(244, 'エンテイ', 'ほのお', '', 115, 115, 85, 90, 75, 100)
(245, 'スイクン', 'みず', '', 100, 75, 115, 90, 115, 85)
(246, 'ヨーギラス', 'いわ じめん', 'サナギラス', 50, 64, 50, 45, 50, 41)
(247, 'サナギラス', 'いわ じめん', 'バンギラス', 70, 84, 70, 65, 70, 51)
(248, 'バンギラス', 'いわ あく', '', 100, 134, 110, 95, 100, 61)
(249, 'ルギア', 'エスパー ひこう', '', 106, 90, 130, 90, 154, 110)
(250, 'ホウオウ', 'ほのお ひこう', '', 106, 130, 90, 110, 154, 90)
(251, 'セレビィ', 'エスパー くさ', '', 100, 100, 100, 100, 100, 100)
```

図 2 課題 23 の実行結果

## リスト 2 pokemon\_db.py

---

```
1 import sqlite3
2 import sys
3
4 def read_names(file_path):
5     names = []
6     with open(encoding="utf8", file=file_path, mode="r") as rfo:
7         for row in rfo:
8             id_, nam, typ, *evo_lst = row.rstrip().split("\t")
9             evo = " ".join(evo_lst)
10            names.append((int(id_), nam, typ, evo))
11    return names
12
13
14 def read_stats(file_path):
15     stats = []
16     with open(file_path, "r") as rfo:
17         for row in rfo:
18             row = row.rstrip()
19             stats.append([int(col) for col in row.split(" ")])
20    return stats
21
22
23 if __name__ == "__main__":
24     names = read_names(sys.argv[1]) # lec10/data/poke_names.txt
25     stats = read_stats(sys.argv[2]) # lec10/data/base_stats.txt
26     db_path = sys.argv[3] # lec10/kadai23/pokemon.db
```

---

## 【課題 24 : stock\_db.py】

配布した ufj.html から、日付、始値、高値、安値、終値、出来高、調整後終値を抽出し、テーブルに格納するコードを実装せよ。

採点の都合上、以下の要件を満たすこととする：

- (1) 読み込みファイル (ufj.html) と出力データベース (stocks.db) のパスをコマンドライン引数で指定する；
- (2) テーブル名は「ufj」にする；
- (3) カラム (date, hajimene, takane, yasune, owarine, dekidaka, owarine2) を持つテーブルを CREATE する；
- (4) ただし、すでに「ufj」という名前のテーブルが存在し OperationalError が発生した場合は、当該テーブルを DROP し、再度 CREATE する；
- (5) ufj.html の内容を ufj テーブルに INSERT する；
- (6) 集計関数用の集計クラス MaxAverage を定義する；
  - (a) イニシャライザで、最大値用のインスタンス変数を定義し、仮の最大値として 0 を代入する；
  - (b) step メソッドで、引数として渡された 2 つの値の平均値を計算し、仮の最大値と比較・更新する；
  - (c) finalize メソッドで、最大値を return する；
- (7) MaxAverage クラスを「max\_avg」という集計関数名で Connection オブジェクトに登録する；
- (8) max\_avg 関数に hajimene と owarine を渡すことで平均値を計算させ、その最大値を集計し、出力する；
- (9) 「#以下は変更しないこと」以下を変更しない；
- (10) 出力結果が図 3 になるようにする；

```
PS C:\Users\admin\Desktop\ProA2> python lec10/kadai24/stock_db.py lec10/data/ufj.html lec10/kadai24/stocks.db
(712.9,)
(715.0, 710.8, 712.9)
(707.3, 710.7, 709.0)
(710.0, 708.5, 709.25)
(707.3, 710.8, 709.05)
(708.0, 703.8, 705.9)
(699.7, 702.0, 700.85)
(702.0, 702.3, 702.15)
(705.4, 703.3, 704.3499999999999)
(694.4, 699.5, 696.95)
(697.0, 687.5, 692.25)
(709.5, 691.1, 700.3)
(711.5, 709.4, 710.45)
(712.0, 710.1, 711.05)
(705.0, 705.3, 705.15)
(688.0, 695.3, 691.65)
(687.9, 693.0, 690.45)
(681.9, 688.7, 685.3)
(684.9, 681.9, 683.4)
(669.2, 675.3, 672.25)
(658.9, 659.2, 659.05)
```

図 3 課題 24 の実行結果

## リスト 3 stock\_db.py

```
1 import sys
2 import re
3 import sqlite3
4
5 def read_html(file_path):
6     with open(file_path, "r", encoding="utf8") as rfo:
7         html = rfo.read().replace("\n", "")
8         datepattern = r"<th scope=\"row\".*?>(\d{4}年\d{1,2}月\d{1,2}日)</th>"
9         pricepattern = r"<td.*?><span.*?><span .*?><span .*?>([0-9.]{1,})</span></span></span></td>"
10        pattern = re.compile(datepattern+pricepattern*6)
11        stocks = list()
12        for match in pattern.findall(html):
13            stocks.append([match[0]]+[float(m.replace(",","")) for m in match[1:]])
14        return stocks
15
16
17 if __name__ == "__main__":
18     stocks = read_html(sys.argv[1]) # lec10/data/ufj.html
19     db_path = sys.argv[2] # lec10/kadai24/stocks.db
20
21
22
23
24
25
26 # 以下は変更しないこと
27 sql = f"SELECT hajimene, owarine, (hajimene+owarine)/2 FROM ufj"
28 cur.execute(sql)
29 for row in cur.fetchall():
30     print(row)
31
32 con.close()
```