

Python: ramki danych.

Wszystkie zadania rozwiązujemy bez używania pętli.

Będziemy ponownie korzystać z bazy danych `nycflights13`.

Przypomnijmy, że baza danych składa się z następujących ramek danych:

- `flights` – information about all flights that departed from NYC (e.g., EWR, JFK, and LGA) in 2013
- `airports` – airport names and locations
- `planes` – construction information about each plane
- `airlines` – translation between two letter carrier codes and names
- `weather` – hourly meteorological data for each airport.

Każde rozwiązanie uzyskane przy użyciu pakietu `pandas` porównaj z wynikiem referencyjnym otrzymanym wprost z zapytania SQL.

```
import os, os.path
import sqlite3

# ścieżka dostępu do bazy danych:
baza = os.path.join('przyklad.db')
conn = sqlite3.connect(baza)      # połączenie do bazy danych

airports.to_sql("airports", conn)  # importujemy ramkę danych do bazy danych
# ...
#
pd.read_sql_query("""
                    Zapytanie SQL
                    """, conn)
# ...
# po skończonej pracy zamykamy połączenie
#
conn.close()
```

Porównania równoważności ramek dokonaj przy użyciu metody `.equals()`.

Zadanie 7.1 [MG] Zapoznaj się z ramkami danych z bazy `nycflights13`

- wykorzystaj metodę `.info()` i `.describe()`,
- poznaj wymiar ramek danych np. `shape`,
- wypisz kilka pierwszych i ostatnich wierszy ramek danych `.head()`, `.tail()`.

Zadanie 7.2 [MG] Wykonaj poniższe polecenia SQL prz użyciu funkcji `sqldf()`. Następnie odtwórz wynikową ramkę danych przy użyciu funkcji (bazowych) R-a.

```
# a
SELECT DISTINCT engine FROM planes

# b
SELECT DISTINCT type, manufacturer FROM planes

# c
SELECT COUNT(*), engine FROM planes GROUP BY engine

# d
SELECT COUNT(*), engine, type
FROM planes
```

```
GROUP BY engine, type
```

```
# e
SELECT MIN(year) AS minYear, engine,
       manufacturer
FROM planes
GROUP BY engine, manufacturer
```

Zadanie 7.3 [MG] Wykonaj poniższe polecenia SQL prz użyciu funkcji `sqldf()`. Następnie odtwórz wynikową ramkę danych przy użyciu funkcji (bazowych) R-a.

```
# a
SELECT * FROM planes WHERE speed IS NOT NULL
# b
SELECT tailnum
FROM planes
WHERE seats BETWEEN 150 AND 190 AND year >= 2012
# c
SELECT * FROM planes
WHERE manufacturer IN ("BOEING", "AIRBUS", "EMBRAER") AND seats > 390

# d
SELECT DISTINCT year, seats
FROM planes
WHERE year >= 2012
ORDER BY year ASC, seats DESC

# e
SELECT DISTINCT year, seats
FROM planes
WHERE year >= 2012
ORDER BY seats DESC, year ASC
```

Zadanie 7.4 [MG] Wykonaj poniższe polecenia SQL prz użyciu funkcji `sqldf()`. Następnie odtwórz wynikową ramkę danych przy użyciu funkcji (bazowych) R-a.

```
# a
SELECT manufacturer, COUNT(*)
FROM planes
WHERE seats > 200
GROUP BY manufacturer

# b
SELECT manufacturer, COUNT(*)
FROM planes
GROUP BY manufacturer
HAVING COUNT(*) > 10

# c
SELECT manufacturer, COUNT(*)
FROM planes
WHERE seats > 200
GROUP BY manufacturer
HAVING COUNT(*) > 10
```

```
# d
SELECT manufacturer, COUNT(*) AS howmany
FROM planes
GROUP BY manufacturer
ORDER BY howmany
DESC LIMIT 5
```

Zadanie 7.5 [MG] Wykonaj poniższe polecenia SQL prz użyciu funkcji `sqldf()`. Następnie odtwórz wynikową ramkę danych przy użyciu funkcji (bazowych) R-a.

```
# a
SELECT flights.tailnum, flights.year, flights.month, flights.day,
       flights.carrier, planes.manufacturer, planes.type
FROM flights
  LEFT JOIN
    planes
  ON flights.tailnum=planes.tailnum

# b
SELECT planes.*, airlines.* FROM
  (SELECT DISTINCT carrier, tailnum FROM flights) AS cartail
  INNER JOIN planes ON cartail.tailnum=planes.tailnum
  INNER JOIN airlines ON cartail.carrier=airlines.carrier

# c
SELECT flights2.*, weather2.atemp, weather2.ahumid, weather2.apressure FROM
  (SELECT * FROM flights WHERE origin='EWR') AS flights2
  LEFT JOIN
    (SELECT year, month, day, AVG(temp) AS atemp,
            AVG(humid) AS ahumid, AVG(pressure) AS apressure
     FROM weather WHERE origin='EWR' GROUP BY year, month, day) AS weather2
  ON flights2.year=weather2.year
  AND flights2.month=weather2.month
  AND flights2.day=weather2.day
```

Zadanie 7.6 [MG] Niech A oznacza ramkę danych zawierającą pierwszych 10 wierszy z ramki danych `airports`. Zaś B – podzbiór wierszy od 6 do 15.

```
# a
SELECT * FROM A UNION SELECT * FROM B

# b
SELECT * FROM A UNION ALL SELECT * FROM B

# c
SELECT * FROM A INTERSECT SELECT * FROM B

# d
SELECT * FROM A EXCEPT SELECT * FROM B

# e
SELECT * FROM B EXCEPT SELECT * FROM A
```