

# Laboratorium 10 – Obsługa baz danych

Języki skryptowe

## Cele dydaktyczne

1. Zapoznanie się z obsługą relacyjnych baz w języku Python.
2. Zapoznanie się z mechanizmami odwzorowywania obiektowo-relacyjnego.

## Wprowadzenie

Zapoznaj się ze zbiorem danych [Przejazdy Wrocławskiego Roweru Miejskiego \(WRM\) - archiwalne](#). Zbiór danych zawiera m.in. przejazdy WRM w latach 2019-2021 miesięcznie. Następnie, pobierz pliki \*.csv z historią przejazdów za rok 2021 (w porcjach od stycznia do grudnia).



...

Struktura plików obejmuje następujące kolumny:

- UID wynajmu – identyfikator wypożyczenia roweru,
- Numer roweru – numer roweru,
- Data wynajmu – data wypożyczenia roweru,
- Data zwrotu – data zwrotu roweru,
- Stacja wynajmu – stacja wypożyczenia roweru,
- Stacja zwrotu – stacja zwrotu roweru,
- Czas trwania – czas trwania wypożyczenia w minutach.

Przed przystąpieniem do wykonywania zadań zapoznaj się z sekcją **Wersja dla ambitnych** na końcu dokumentu .

## Zadania

1. Na bazie struktury plików CSV utwórz strukturę relacyjnej bazy danych w systemie zarządzania SQLite lub dowolnym innym RDBMS, która pozwoli na wstawienie, przechowywanie oraz odpytywanie danych o wypożyczeniach rowerów. Baza danych

powinna uwzględniać co najmniej dwie tabele zawierające wypożyczenia oraz stacje. Za wsparcie może posłużyć poniższy diagram:



- a. Utwórz polecenia DDL/SQL ("CREATE TABLE...", "ALTER TABLE...") tworzące tę bazę danych w wybranym RDBMS. Skonstruuj klucze obce realizujące więzy integralności referencyjnej. Zastanów się nad odpowiednimi typami kolumn. W przypadku, gdy wybrany RDBMS nie wspiera typu czasowego, rozważ [obliczenie](#) i wykorzystanie [czasu uniksowego](#).
- b. Utwórz skrypt create\_database.py, którego wywołanie spowoduje uruchomienie opracowanych poleceń DDL/SQL, a co za tym idzie, utworzenie bazy danych. Skrypt powinien przyjmować argument linii komend z nazwą bazy. Przykładowo, dla implementacji w SQLite wywołanie skryptu

```
python create_database.py rentals
```

powinno skutkować utworzeniem pliku rentals.sqlite3 zawierającego pustą strukturę.

**Punkty: 2**

2. Opracuj skrypt load\_data.py, który jako argument przyjmie plik z historią przejazdów (np. historia\_przejazdow\_2021-02.csv) oraz plik z bazą danych, a następnie wstawi dane do bazy danych. Załaduj do bazy wszystkie pobrane pliki dla roku 2021. Przykład wywołania:

```
python load_data.py historia_przejazdow_2021-02.csv rentals
```

**Punkty: 2**

3. Utwórz skrypt z graficznym lub tekstowym interfejsem użytkownika, który pozwoli użytkownikowi wybrać stację z listy, a następnie dla wskazanej stacji skonstruuje zapytanie SQL typu "SELECT" obliczy i wyświetli:
  - a. średni czas trwania przejazdu rozpoczynanego na danej stacji,

- b. średni czas trwania przejazdu kończącego na danej stacji,
- c. liczbę różnych rowerów parkowanych na danej stacji,
- d. wynik samodzielnie zaprojektowanego, nietrywialnego zapytania.

**Punkty: 3**

### **Wersja dla ambitnych**

Zadania 1-3 wykonaj z wykorzystaniem wybranej biblioteki do odwzorowywania obiektowo-relacyjnego (ORM, np. [SQLAlchemy ORM](#), [peewee](#), ...).

**Punkty: 3**