## Міграція Бази Даних

**Задача**: написати програму в стилі Layered Architecture, в якій би була описана структура БД за допомогою ORM моделі (<https://www.fullstackpython.com/object-relational-mappers-orms.html>). Запрограмувати зчитування з файлу та рефакторинг-міграції погодної бази даних, що знаходиться за цим посиланням: <https://www.kaggle.com/datasets/nelgiriyewithana/global-weather-repository>. За бажанням, можна вибрати іншу базу даних та виконати аналогіні описаним нижче завдання.

Міграції мають дозволяти, як створити базу з нуля, так і мігрувати існуючу базу. При міграції бази дані не мають бути втрачені.

## Завдання

1. Відібрати колонки, які будуть описані в ORM. Серед колонок мають бути в кількості хоча б одна

* текстова (country)
* з цілим числом (wind\_degree)
* з дробним числом (wind\_kph)
* з еnumetaion (wind\_direction)
* з датою (last\_updated)
* з часом (sunrise)

Також взяти в опрацювання всі колонки стосовно однієї категорії для подальшого оцінювання в залежності від кратності варіанта вашому номеру в списку:  
1. вітру,

2. стану повітря,

3. осадів

4. сходу-заходу небесних тіл

1. Щоб погодні дані не були всі в одній купі, треба впорядкувати БД - зробити рефакторинг. Для цього дані стосовно вашої категорії необхідно винести в окрему таблицю. Це робиться інструментом для міграції, вибраним з цього списку: <https://www.ropstam.com/best-database-schema-migration-tools/>. Наприклад, Flyway, Liquibase, Alembic
2. Створити нову колонку у новоствореній таблиці типу булеан, яка б відображала відповідь на питання “Чи варто виходити на вулицю”. Для заповнення колонки придумайте формулу, яка б на виході базуючись на ваших даних видавала відповіді “так” чи “ні”. Наприклад, вітер більше 10 м/c, краще не виходити.
3. Дайте можливість користувачу задавши країну та дату, можливо ще додаткові параметри, побачити всю інформацію стосовно погоди, яка у вас є. Інтерфейс не обов’язковий, достатньо консолі.
4. Мігрувати з однієї БД - Postgres на іншу - MySQL, або навпаки. Таким чином, всі скрипти, які ви написали мають накотитись на нову базу без проблем, та ви матимете 2 однакові бази.

Кожен етап бажано зберігати в окремій гілці гітхаб щоб в разі необхідності показати необхідних етап та стан бази викладачу. Також бажано описати в записці особливості переходу з кожного етапу на наступний.

### Запропоновані технології (за бажанням, можна спробувати й інші)

Мова - Python, Java

БД - Postgres, MySQL

Migration Tool: Flyway, Liquibase

ORM - SQLAlchemy, Hibernate

Клієнт БД - pgAdmin, MySQL Workbench

### Питання до роботи

1. Чи можна завантажувати всю базу, а описувати в ORM моделі тільки деякі колонки?
2. Чи є дефолтне значення при створенні у колонці “Чи варто виходити на вулицю”? Поясніть?
3. У скільки етапів ви зробили 3тє завдання?
4. На яку частину застосунку ви поклали фунцію заповнення колонки “Чи варто виходити на вулицю” даними?
5. Чи легко було накочувати міграції, перейти з однієї бази на іншу? Які проблеми траплялись?