1. Завантажте дані:

* Створіть схему pandemic у базі даних за допомогою SQL-команди.

| CREATE DATABASE IF NOT EXISTS pandemic; |
| --- |

* Оберіть її як схему за замовчуванням за допомогою SQL-команди.

| use pandemic; |
| --- |

2. Нормалізуйте таблицю infectious\_cases. Збережіть у цій же схемі дві таблиці з нормалізованими даними.

| use pandemic; DROP TABLE IF EXISTS cases; DROP TABLE IF EXISTS countries;  CREATE TABLE countries ( id INT PRIMARY KEY AUTO\_INCREMENT, country\_name VARCHAR(100), country\_code VARCHAR(10) );  CREATE TABLE cases ( id INT PRIMARY KEY AUTO\_INCREMENT,  country\_id INT,  year INT,  number\_yaws TEXT,  polio\_cases INT, cases\_guinea\_worm INT, number\_rabies TEXT,  number\_malaria TEXT, number\_hiv TEXT, number\_tuberculosis TEXT,  number\_smallpox TEXT, number\_cholera\_cases TEXT, FOREIGN KEY (country\_id) REFERENCES countries(id) );  INSERT INTO countries (country\_name, country\_code) SELECT Entity, Code FROM infectious\_cases GROUP BY Entity, Code;  INSERT INTO cases (country\_id, year, number\_yaws, polio\_cases,  cases\_guinea\_worm, number\_rabies, number\_malaria,  number\_hiv, number\_tuberculosis, number\_smallpox,  number\_cholera\_cases) SELECT (SELECT id FROM countries c WHERE ic.Entity=c.country\_name) country\_id,  Year, Number\_yaws, polio\_cases, cases\_guinea\_worm, Number\_rabies, Number\_malaria,  Number\_hiv, Number\_tuberculosis, Number\_smallpox, Number\_cholera\_cases  FROM infectious\_cases ic; |
| --- |

3. Проаналізуйте дані:

* Для кожної унікальної комбінації Entity та Code або їх id порахуйте середнє, мінімальне, максимальне значення та суму для атрибута Number\_rabies.

💡 Врахуйте, що атрибут Number\_rabies може містити порожні значення ‘’ — вам попередньо необхідно їх відфільтрувати.

* Результат відсортуйте за порахованим середнім значенням у порядку спадання.
* Оберіть тільки 10 рядків для виведення на екран.

| use pandemic; SELECT country\_id, AVG(number\_rabies) average, MIN(number\_rabies) min, MAX(number\_rabies) max, SUM(number\_rabies) sum FROM cases WHERE number\_rabies <> "" GROUP BY country\_id ORDER BY average DESC LIMIT 10 |
| --- |

4. Побудуйте колонку різниці в роках.

Для оригінальної або нормованої таблиці для колонки Year побудуйте з використанням вбудованих SQL-функцій:

* атрибут, що створює дату першого січня відповідного року,

💡 Наприклад, якщо атрибут містить значення ’1996’, то значення нового атрибута має бути ‘1996-01-01’.

* атрибут, що дорівнює поточній даті,
* атрибут, що дорівнює різниці в роках двох вищезгаданих колонок.

| SELECT id, year, MAKEDATE(year,1) base\_date, CURDATE() today\_date, TIMESTAMPDIFF(YEAR, MAKEDATE(year,1), CURDATE()) years\_diff FROM cases |
| --- |

5. Побудуйте власну функцію.

* Створіть і використайте функцію, що будує такий же атрибут, як і в попередньому завданні: функція має приймати на вхід значення року, а повертати різницю в роках між поточною датою та датою, створеною з атрибута року (1996 рік → ‘1996-01-01’).

| DELIMITER // CREATE FUNCTION count\_year\_diff(year INT) RETURNS INT DETERMINISTIC  NO SQL BEGIN  DECLARE result INT;  SET result = TIMESTAMPDIFF(YEAR, MAKEDATE(year, 1), CURDATE());  RETURN result; END //  DELIMITER ;  SELECT id, year, count\_year\_diff(year) years\_diff FROM cases |
| --- |