**Фінальний проєкт**

**1. Завантажте дані:**

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

CREATE SCHEMA 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;

CREATE TABLE countries(

id INT PRIMARY KEY AUTO\_INCREMENT,

code VARCHAR(8) UNIQUE,

country VARCHAR(32) NOT NULL UNIQUE

);

INSERT INTO countries (code, country)

SELECT DISTINCT code, entity FROM infectious\_cases;

CREATE TABLE cases\_normalized

AS SELECT \* FROM infectious\_cases;

ALTER TABLE cases\_normalized

ADD id INT PRIMARY KEY AUTO\_INCREMENT FIRST,

ADD country\_id INT AFTER id,

ADD CONSTRAINT fk\_country\_id FOREIGN KEY (country\_id) REFERENCES countries(id);

UPDATE cases\_normalized i

JOIN countries c ON i.code = c.code

SET i.country\_id = c.id

WHERE i.id > 0;

ALTER TABLE cases\_normalized

DROP COLUMN entity,

DROP COLUMN code;

SELECT \* FROM cases\_normalized;

**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 Entity, Code,

AVG(Number\_rabies) AS average\_value,

MAX(Number\_rabies) AS max\_value,

MIN(Number\_rabies) AS min\_value,

SUM(Number\_rabies) AS sum\_value

FROM cases\_normalized

WHERE Number\_rabies IS NOT NULL AND Number\_rabies <> ''

GROUP BY Entity, Code

ORDER BY average\_value 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’.**

USE pandemic;

SELECT Entity, Code, Year,

CONCAT(Year, "-01-01") AS year\_start\_date

FROM infectious\_cases;

**атрибут, що дорівнює поточній даті,**

USE pandemic;

SELECT Entity, Code, Year,

CURDATE() AS year\_current\_date

FROM infectious\_cases;

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

**Перераховувати всі інші атрибути, такі як Number\_malaria, не потрібно.**

**Для пошуку необхідних вбудованих функцій вам може знадобитися матеріал до теми 7.**

USE pandemic;

SELECT Entity, Code, Year,

TIMESTAMPDIFF(YEAR, CONCAT(Year, '-01-01'), CURDATE()) AS difference\_year

FROM infectious\_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’).**

DROP FUNCTION IF EXISTS fn\_subtract\_now\_year;

DELIMITER //

CREATE FUNCTION fn\_subtract\_now\_year(year INT)

RETURNS int

DETERMINISTIC

NO SQL

BEGIN

DECLARE result INT;

SET result = YEAR(CURDATE()) - year;

RETURN result;

END //

DELIMITER ;

SELECT fn\_subtract\_now\_year(1994)

DROP FUNCTION IF EXISTS fn\_calc\_illnesses;

DELIMITER //

CREATE FUNCTION fn\_calc\_illnesses(num\_illnesses\_per\_year DOUBLE, period INT)

RETURNS DOUBLE

DETERMINISTIC

NO SQL

BEGIN

DECLARE result DOUBLE;

SET result = num\_illnesses\_per\_year / period;

RETURN result;

END //

DELIMITER ;

SELECT fn\_calc\_illnesses(20000, 2);