

1

```
-- Створення схеми pandemic  
CREATE SCHEMA IF NOT EXISTS pandemic;
```

```
-- Вибір схеми pandemic як схеми за замовчуванням  
USE pandemic;
```

2.

```
-- Перевірка кількості відсутніх значень у кожній колонці  
SELECT  
    COUNT(*) AS Total_Rows, -- Загальна кількість рядків  
    SUM(CASE WHEN Number_yaws IS NULL OR Number_yaws = " THEN 1 ELSE 0 END)  
AS Missing_Number_yaws,  
    SUM(CASE WHEN polio_cases IS NULL OR polio_cases = " THEN 1 ELSE 0 END) AS  
Missing_polio_cases,  
    SUM(CASE WHEN cases_guinea_worm IS NULL OR cases_guinea_worm = " THEN 1  
ELSE 0 END) AS Missing_cases_guinea_worm,  
    SUM(CASE WHEN Number_rabies IS NULL OR Number_rabies = " THEN 1 ELSE 0  
END) AS Missing_Number_rabies,  
    SUM(CASE WHEN Number_malaria IS NULL OR Number_malaria = " THEN 1 ELSE 0  
END) AS Missing_Number_malaria,  
    SUM(CASE WHEN Number_hiv IS NULL OR Number_hiv = " THEN 1 ELSE 0 END) AS  
Missing_Number_hiv,  
    SUM(CASE WHEN Number_tuberculosis IS NULL OR Number_tuberculosis = " THEN 1  
ELSE 0 END) AS Missing_Number_tuberculosis,  
    SUM(CASE WHEN Number_smallpox IS NULL OR Number_smallpox = " THEN 1 ELSE  
0 END) AS Missing_Number_smallpox,  
    SUM(CASE WHEN Number_cholera_cases IS NULL OR Number_cholera_cases = "  
THEN 1 ELSE 0 END) AS Missing_Number_cholera_cases  
FROM infectious_cases;
```

```
-- Заміна некоректних значень на NULL  
UPDATE infectious_cases  
SET Number_yaws = NULL WHERE NOT (Number_yaws REGEXP '^-[0-9]+(\.[0-9]+)?$');  
UPDATE infectious_cases  
SET polio_cases = NULL WHERE NOT (polio_cases REGEXP '^-[0-9]+(\.[0-9]+)?$');  
UPDATE infectious_cases  
SET cases_guinea_worm = NULL WHERE NOT (cases_guinea_worm REGEXP  
'^-[0-9]+(\.[0-9]+)?$');  
UPDATE infectious_cases  
SET Number_rabies = NULL WHERE NOT (Number_rabies REGEXP  
'^-[0-9]+(\.[0-9]+)?$');  
UPDATE infectious_cases  
SET Number_malaria = NULL WHERE NOT (Number_malaria REGEXP  
'^-[0-9]+(\.[0-9]+)?$');  
UPDATE infectious_cases  
SET Number_hiv = NULL WHERE NOT (Number_hiv REGEXP '^-[0-9]+(\.[0-9]+)?$');
```

```

UPDATE infectious_cases
SET Number_tuberculosis = NULL WHERE NOT (Number_tuberculosis REGEXP
'^-[0-9]+(\.[0-9]+)?$');
UPDATE infectious_cases
SET Number_smallpox = NULL WHERE NOT (Number_smallpox REGEXP
'^-[0-9]+(\.[0-9]+)?$');
UPDATE infectious_cases
SET Number_cholera_cases = NULL WHERE NOT (Number_cholera_cases REGEXP
'^-[0-9]+(\.[0-9]+)?$');

```

-- Перевірка таблиці на відсутність некоректних даних

```

SELECT *
FROM infectious_cases
WHERE NOT (Number_yaws REGEXP '^-[0-9]+(\.[0-9]+)?$' OR Number_yaws IS NULL)
  OR NOT (polio_cases REGEXP '^-[0-9]+(\.[0-9]+)?$' OR polio_cases IS NULL)
  OR NOT (cases_guinea_worm REGEXP '^-[0-9]+(\.[0-9]+)?$' OR cases_guinea_worm IS
NULL)
  OR NOT (Number_rabies REGEXP '^-[0-9]+(\.[0-9]+)?$' OR Number_rabies IS NULL)
  OR NOT (Number_malaria REGEXP '^-[0-9]+(\.[0-9]+)?$' OR Number_malaria IS NULL)
  OR NOT (Number_hiv REGEXP '^-[0-9]+(\.[0-9]+)?$' OR Number_hiv IS NULL)
  OR NOT (Number_tuberculosis REGEXP '^-[0-9]+(\.[0-9]+)?$' OR Number_tuberculosis
IS NULL)
  OR NOT (Number_smallpox REGEXP '^-[0-9]+(\.[0-9]+)?$' OR Number_smallpox IS
NULL)
  OR NOT (Number_cholera_cases REGEXP '^-[0-9]+(\.[0-9]+)?$' OR
Number_cholera_cases IS NULL);

```

-- Видалити існуючі таблиці перед створенням

```

DROP TABLE IF EXISTS cases;
DROP TABLE IF EXISTS entities;

```

-- Створення таблиці entities

```

CREATE TABLE entities (
  id INT AUTO_INCREMENT PRIMARY KEY,
  Entity VARCHAR(255),
  Code VARCHAR(10),
  UNIQUE(Entity, Code)
);

```

-- Наповнення таблиці entities

```

INSERT INTO entities (Entity, Code)
SELECT DISTINCT Entity, Code
FROM infectious_cases;

```

-- Створення таблиці cases

```

CREATE TABLE cases (
  id INT AUTO_INCREMENT PRIMARY KEY,

```

```

entity_id INT,
Year INT NOT NULL,
Number_yaws FLOAT,
polio_cases FLOAT,
cases_guinea_worm FLOAT,
Number_rabies FLOAT,
Number_malaria FLOAT,
Number_hiv FLOAT,
Number_tuberculosis FLOAT,
Number_smallpox FLOAT,
Number_cholera_cases FLOAT,
FOREIGN KEY (entity_id) REFERENCES entities(id)
);

```

-- Наповнення таблиці cases

```

INSERT INTO cases (entity_id, Year, Number_yaws, polio_cases, cases_guinea_worm,
Number_rabies, Number_malaria, Number_hiv, Number_tuberculosis, Number_smallpox,
Number_cholera_cases)

```

```

SELECT
    e.id,
    ic.Year,
    ic.Number_yaws,
    ic.polio_cases,
    ic.cases_guinea_worm,
    ic.Number_rabies,
    ic.Number_malaria,
    ic.Number_hiv,
    ic.Number_tuberculosis,
    ic.Number_smallpox,
    ic.Number_cholera_cases
FROM infectious_cases ic
JOIN entities e
ON ic.Entity = e.Entity AND ic.Code = e.Code;

```

3

-- Аналіз даних для Number\_rabies

```

SELECT
    e.Entity,
    e.Code,
    AVG(c.Number_rabies) AS Avg_Number_rabies,
    MIN(c.Number_rabies) AS Min_Number_rabies,
    MAX(c.Number_rabies) AS Max_Number_rabies,
    SUM(c.Number_rabies) AS Sum_Number_rabies
FROM cases c
JOIN entities e ON c.entity_id = e.id
WHERE c.Number_rabies IS NOT NULL -- Фільтрація порожніх значень
GROUP BY e.Entity, e.Code
ORDER BY Avg_Number_rabies DESC -- Сортуння за середнім значенням

```

LIMIT 10; -- Вибір перших 10 рядків

4

-- Додати колонку різниці в роках

SELECT

c.Year,

STR\_TO\_DATE(CONCAT(c.Year, '-01-01'), '%Y-%m-%d') AS Year\_01\_01, -- Дата 1 січня  
відповідного року

CURDATE() AS Current\_Date, -- Поточна дата

TIMESTAMPDIFF(YEAR, STR\_TO\_DATE(CONCAT(c.Year, '-01-01'), '%Y-%m-%d'),  
CURDATE()) AS Years\_Difference -- Різниця в роках

FROM cases c

LIMIT 10; -- Вибір перших 10 рядків для демонстрації

5

-- Створення функції для обчислення різниці в роках

DELIMITER \$\$

CREATE FUNCTION YearDifference(input\_year INT)

RETURNS INT

DETERMINISTIC

BEGIN

DECLARE year\_start DATE;

DECLARE years\_diff INT;

-- Формуємо дату 1 січня відповідного року

SET year\_start = STR\_TO\_DATE(CONCAT(input\_year, '-01-01'), '%Y-%m-%d');

-- Обчислюємо різницю в роках між поточною датою і створеною датою

SET years\_diff = TIMESTAMPDIFF(YEAR, year\_start, CURDATE());

RETURN years\_diff;

END\$\$

DELIMITER ;

-- Використання функції

SELECT

c.Year,

YearDifference(c.Year) AS Years\_Difference

FROM cases c

LIMIT 10; -- Вибір перших 10 рядків

