1.

CREATE SCHEMA IF NOT EXISTS pandemic;

USE pandemic;

2.

SELECT COUNT(\*) FROM infectious\_cases;

CREATE TABLE entities (

entity\_id INT AUTO\_INCREMENT PRIMARY KEY,

entity VARCHAR(255),

code VARCHAR(50)

);

CREATE TABLE infectious\_normalized (

id INT AUTO\_INCREMENT PRIMARY KEY,

entity\_id INT,

year YEAR,

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(entity\_id)

);

INSERT INTO entities (entity, code)

SELECT DISTINCT Entity, Code FROM infectious\_cases;

INSERT INTO infectious\_normalized (

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.entity\_id,

c.Year,

NULLIF(ic.Number\_yaws, ''),

NULLIF(ic.polio\_cases, ''),

NULLIF(ic.cases\_guinea\_worm, ''),

NULLIF(ic.Number\_rabies, ''),

NULLIF(ic.Number\_malaria, ''),

NULLIF(ic.Number\_hiv, ''),

NULLIF(ic.Number\_tuberculosis, ''),

NULLIF(ic.Number\_smallpox, ''),

NULLIF(ic.Number\_cholera\_cases, '')

FROM infectious\_cases ic

JOIN entities e ON ic.entity = e.entity and ic.code = e.code

3.

SELECT

e.entity,

e.code,

ROUND(AVG(in.Number\_rabies), 2) AS avg\_rabies,

MIN(in.number\_rabies) AS min\_rabies,

MAX(in.number\_rabies) AS max\_rabies,

SUM(in.number\_rabies) AS total\_rabies

FROM infectious\_normalized in

JOIN entities e ON in.entity\_id = e.entity\_id

WHERE in.number\_rabies IS NOT NULL

GROUP BY e.entity, e.code

ORDER BY avg\_rabies DESC

LIMIT 10;

4.

SELECT

id,

year,

MAKEDATE(Year, 1) AS year\_date,

CURDATE() AS c\_date,

TIMESTAMPDIFF(YEAR, MAKEDATE(Year, 1), CURDATE()) AS year\_diff

FROM infectious\_normalized;

5.

DROP FUNCTION IF EXISTS year\_diff;

DELIMITER //

CREATE FUNCTION year\_diff(Year INT)

RETURNS INT

DETERMINISTIC

NO SQL

BEGIN

RETURN TIMESTAMPDIFF(YEAR, MAKEDATE(Year, 1)), CURDATE());

END //

DELIMITER ;

SELECT

id,

year,

year\_diff(year) AS year\_difference

FROM infectious\_normalized;

5. Additional

DROP FUNCTION IF EXISTS cases\_per\_period;

DELIMITER //

CREATE FUNCTION cases\_per\_period(

cases\_per\_year FLOAT,

divider FLOAT

)

RETURNS FLOAT DETERMINISTIC NO SQL

BEGIN

IF cases\_per\_year IS NULL OR divider = 0 THEN

RETURN NULL;

ELSE

RETURN cases\_per\_year / divider;

END IF;

END;

//

DELIMITER ;

SELECT

id,

entity\_id,

year,

number\_rabies,

CASES\_PER\_PERIOD(number\_rabies, 12) AS rabies\_per\_month,

CASES\_PER\_PERIOD(number\_rabies, 4) AS rabies\_per\_quarter,

CASES\_PER\_PERIOD(number\_rabies, 2) AS rabies\_per\_halfyear

FROM infectious\_normalized

WHERE number\_rabies IS NOT NULL;