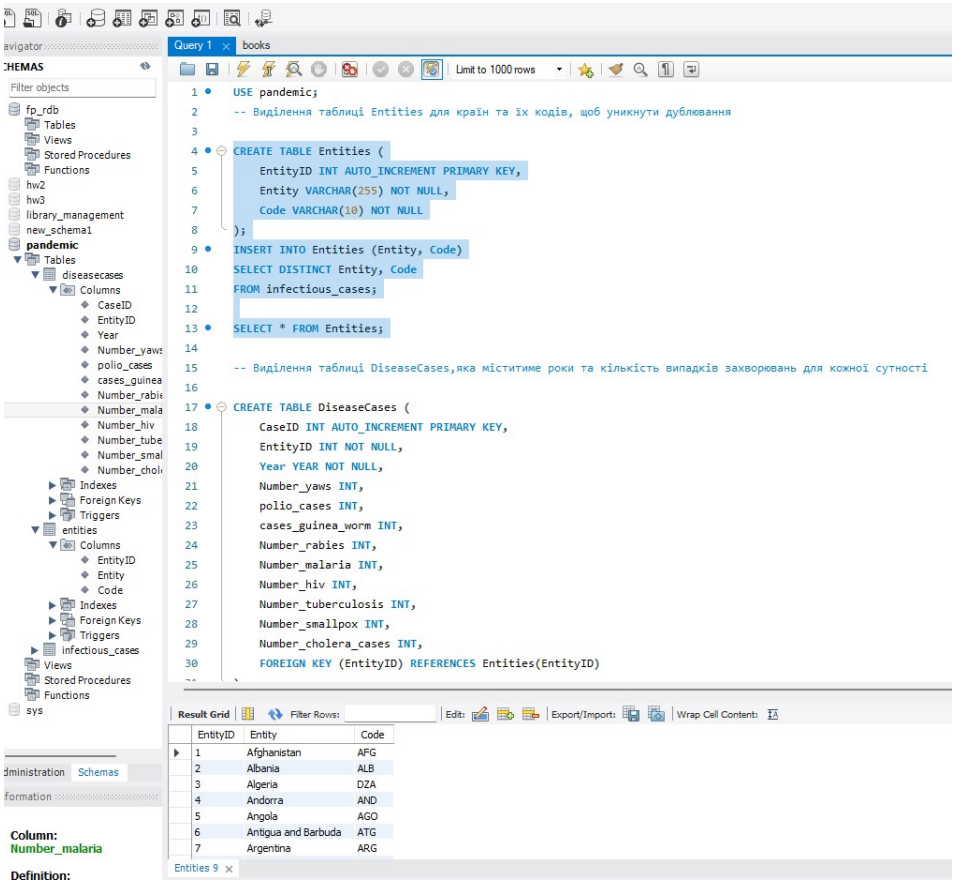
Фінальний проект РБД

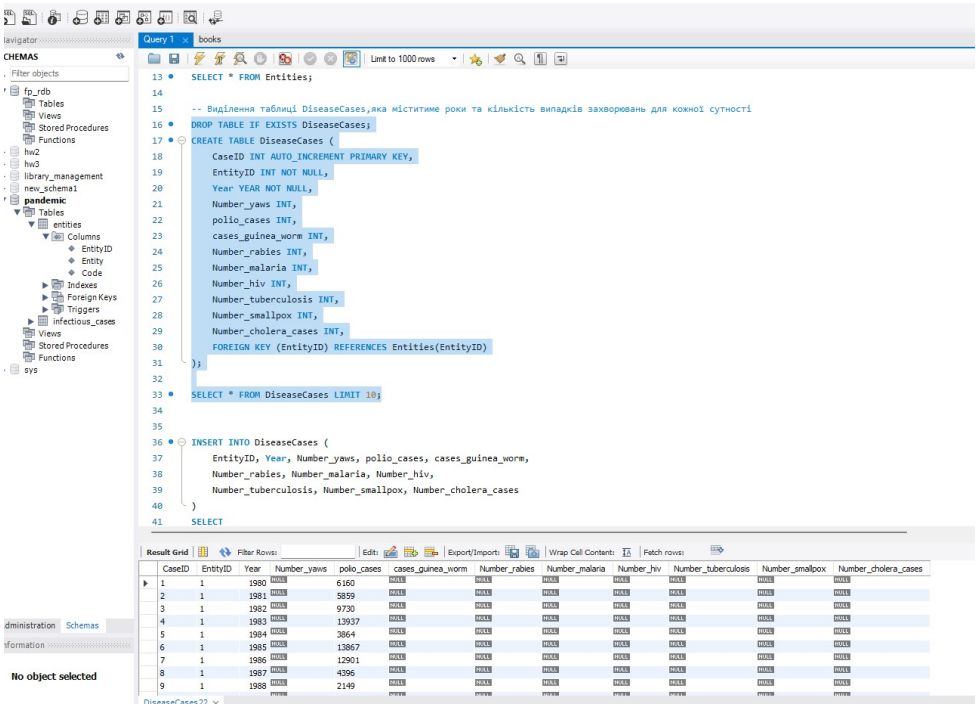
Завдання 1, 2

Завантаження даних та нормалізація таблиць до 3 нормальної форми



```
USE pandemic;
-- Виділення таблиці Entities для країн та їх кодів, щоб уникнути дублювання

CREATE TABLE Entities (
    EntityID INT AUTO_INCREMENT PRIMARY KEY,
    Entity VARCHAR(255) NOT NULL,
    Code VARCHAR(10) NOT NULL
);
INSERT INTO Entities (Entity, Code)
SELECT DISTINCT Entity, Code
FROM infectious_cases;

SELECT * FROM Entities;

-- Виділення таблиці DiseaseCases,яка міститиме роки та кількість випадків захворювань для кожної сутності

CREATE TABLE DiseaseCases (
    CaseID INT AUTO_INCREMENT PRIMARY KEY,
    EntityID INT NOT NULL,
    Year YEAR NOT NULL,
    Number_yaws INT,
    polio_cases INT,
    cases_guinea_worm INT,
    Number_rabies INT,
    Number_malaria INT,
    Number_hiv INT,
    Number_tuberculosis INT,
    Number_smallpox INT,
    Number_cholera_cases INT,
    FOREIGN KEY (EntityID) REFERENCES Entities(EntityID)
```

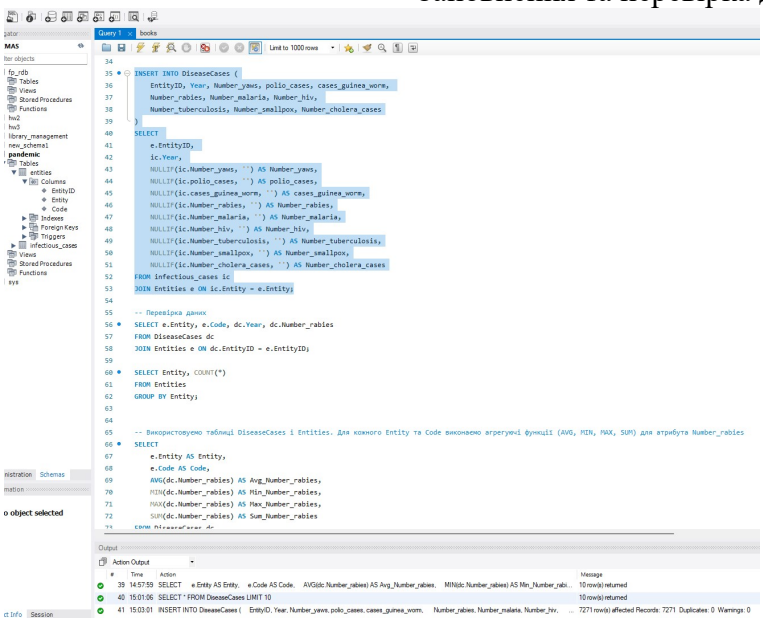
EntityID	Entity	Code
1	Afghanistan	AFG
2	Albania	ALB
3	Algeria	DZA
4	Andorra	AND
5	Angola	AGO
6	Antigua and Barbuda	ATG
7	Argentina	ARG



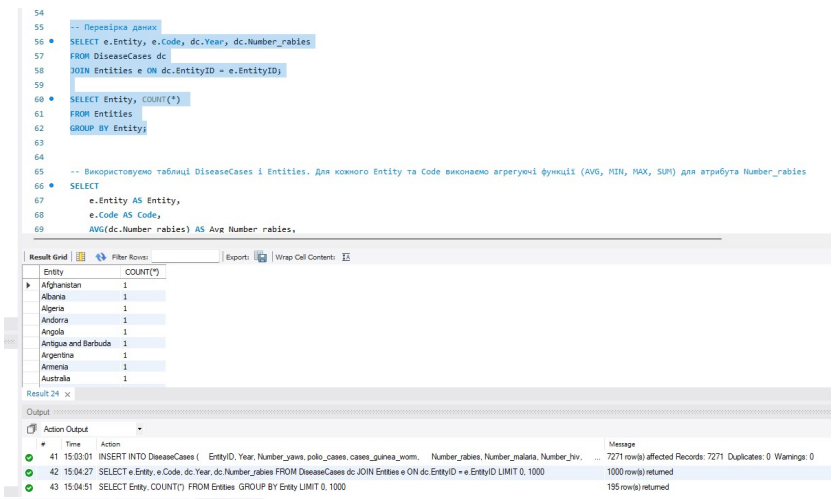
```
SELECT * FROM Entities;

-- Виділення таблиці DiseaseCases,яка міститиме роки та кількість випадків захворювань для кожної сутності
DROP TABLE IF EXISTS DiseaseCases;
CREATE TABLE DiseaseCases (
    CaseID INT AUTO_INCREMENT PRIMARY KEY,
    EntityID INT NOT NULL,
    Year YEAR NOT NULL,
    Number_yaws INT,
    polio_cases INT,
    cases_guinea_worm INT,
    Number_rabies INT,
    Number_malaria INT,
    Number_hiv INT,
    Number_tuberculosis INT,
    Number_smallpox INT,
    Number_cholera_cases INT,
    FOREIGN KEY (EntityID) REFERENCES Entities(EntityID)
);

SELECT * FROM DiseaseCases LIMIT 10;


INSERT INTO DiseaseCases (
    EntityID, Year, Number_yaws, polio_cases, cases_guinea_worm,
    Number_rabies, Number_malaria, Number_hiv,
    Number_tuberculosis, Number_smallpox, Number_cholera_cases
)
SELECT
```

CaseID	EntityID	Year	Number_yaws	polio_cases	cases_guinea_worm	Number_rabies	Number_malaria	Number_hiv	Number_tuberculosis	Number_smallpox	Number_cholera_cases
1	1	1980	NULL	6160	NULL	NULL	NULL	NULL	NULL	NULL	NULL
2	1	1981	NULL	5859	NULL	NULL	NULL	NULL	NULL	NULL	NULL
3	1	1982	NULL	9730	NULL	NULL	NULL	NULL	NULL	NULL	NULL
4	1	1983	NULL	13937	NULL	NULL	NULL	NULL	NULL	NULL	NULL
5	1	1984	NULL	3864	NULL	NULL	NULL	NULL	NULL	NULL	NULL
6	1	1985	NULL	13867	NULL	NULL	NULL	NULL	NULL	NULL	NULL
7	1	1986	NULL	12901	NULL	NULL	NULL	NULL	NULL	NULL	NULL
8	1	1987	NULL	4396	NULL	NULL	NULL	NULL	NULL	NULL	NULL
9	1	1988	NULL	2149	NULL	NULL	NULL	NULL	NULL	NULL	NULL

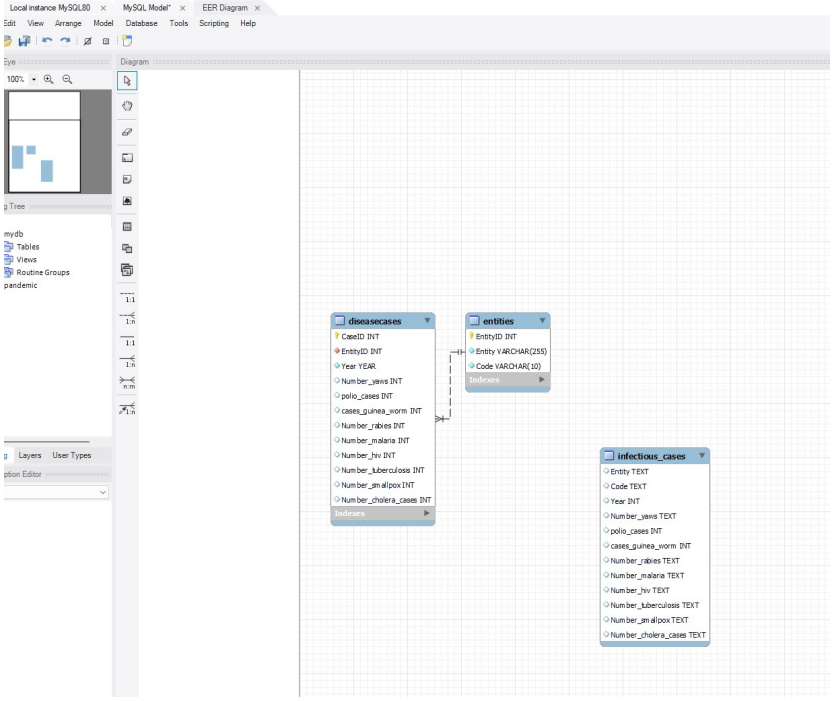
Заповнення та перевірка даних



```
INSERT INTO DiseaseCases (
    EntityID, Year, Number_yaws, polio_cases, cases_guinea_worm,
    Number_rabies, Number_malaria, Number_hiv,
    Number_tuberculosis, Number_smallpox, Number_cholera_cases
)
SELECT
    e.EntityID,
    ic.Year,
    NULLIF(ic.Number_yaws, '') AS Number_yaws,
    NULLIF(ic.polio_cases, '') AS polio_cases,
    NULLIF(ic.cases_guinea_worm, '') AS cases_guinea_worm,
    NULLIF(ic.Number_rabies, '') AS Number_rabies,
    NULLIF(ic.Number_malaria, '') AS Number_malaria,
    NULLIF(ic.Number_hiv, '') AS Number_hiv,
    NULLIF(ic.Number_tuberculosis, '') AS Number_tuberculosis,
    NULLIF(ic.Number_smallpox, '') AS Number_smallpox,
    NULLIF(ic.Number_cholera_cases, '') AS Number_cholera_cases
FROM infectious_cases ic
JOIN Entities e ON ic.Entity = e.Entity;

-- Перевірка даних
SELECT e.Entity, e.Code, dc.Year, dc.Number_rabies
FROM DiseaseCases dc
JOIN Entities e ON dc.EntityID = e.EntityID;

SELECT Entity, COUNT(*)
FROM Entities
GROUP BY Entity;


-- Використовуємо таблиці DiseaseCases і Entities. Для кожного Entity та Code виконаємо агрегуючі функції (AVG, MIN, MAX, SUM) для атрибута Number_rabies
SELECT
    e.Entity AS Entity,
    e.Code AS Code,
    AVG(dc.Number_rabies) AS Avg_Number_rabies,
    MIN(dc.Number_rabies) AS Min_Number_rabies,
    MAX(dc.Number_rabies) AS Max_Number_rabies,
    SUM(dc.Number_rabies) AS Sum_Number_rabies
FROM DiseaseCases dc
```



Entity	COUNT(*)
Afghanistan	1
Albania	1
Algeria	1
Andorra	1
Angola	1
Antigua and Barbuda	1
Argentina	1
Armenia	1
Australia	1



Завдання 3

Аналіз даних

```
62    GROUP BY Entity;
63
64    -- Використовуємо таблиці DiseaseCases і Entities. Для кожного Entity та Code виконаємо агрегуючі функції (AVG, MIN, MAX, SUM) для атрибута Number_rabies
65    SELECT
66        e.Entity AS Entity,
67        e.Code AS Code,
68        AVG(dc.Number_rabies) AS Avg_Number_rabies,
69        MIN(dc.Number_rabies) AS Min_Number_rabies,
70        MAX(dc.Number_rabies) AS Max_Number_rabies,
71        SUM(dc.Number_rabies) AS Sum_Number_rabies
72    FROM DiseaseCases dc
73    JOIN Entities e ON dc.EntityID = e.EntityID
74    WHERE dc.Number_rabies IS NOT NULL
75    GROUP BY e.Entity, e.Code
76    ORDER BY Avg_Number_rabies DESC
77    LIMIT 10;
78
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

Entity	Code	Avg_Number_rabies	Min_Number_rabies	Max_Number_rabies	Sum_Number_rabies
World	OWID_WRL	20192.3667	14076	24745	1211542
India	IND	8599.2333	5426	11121	515954
Pakistan	PAK	1582.2000	1177	1882	94932
Nigeria	NGA	1335.4667	1074	1441	80128
China	CHN	1248.0000	744	2076	74880
Ethiopia	ETH	1145.2333	758	1323	68714
Myanmar	MMR	972.9000	561	1450	58374
Bangladesh	BGD	785.7333	180	1735	47144
Nepal	NPL	661.1667	338	1042	39670

Result 25

Output

Action Output

#	Time	Action	Message
42	15:04:27	SELECT e.Entity, e.Code, dc.Year, dc.Number_rabies FROM DiseaseCases dc JOIN Entities e ON dc.EntityID = e.EntityID LIMIT 0, 1000	1000 row(s) returned
43	15:04:51	SELECT Entity, COUNT(*) FROM Entities GROUP BY Entity LIMIT 0, 1000	195 row(s) returned
44	15:05:44	SELECT e.Entity AS Entity, e.Code AS Code, AVG(dc.Number_rabies) AS Avg_Number_rabies, MIN(dc.Number_rabies) AS Min_Number_rabi...	10 row(s) returned

Завдання 4

Різниця в роках

```
78
79
80    SELECT DISTINCT Year FROM DiseaseCases;
81
82    SELECT
83        `Year`,
84        STR_TO_DATE(CONCAT(`Year`, '-01-01'), '%Y-%m-%d') AS Date_1st_Jan,
85        CURDATE() AS `Current_Date`,
86        TIMESTAMPDIFF(YEAR, STR_TO_DATE(CONCAT(`Year`, '-01-01'), '%Y-%m-%d'), CURDATE()) AS Year_Difference
87    FROM DiseaseCases
88    LIMIT 10;
89
90
91
92
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

Year	Date_1st_Jan	Current_Date	Year_Difference
1980	1980-01-01	2024-12-01	44
1981	1981-01-01	2024-12-01	43
1982	1982-01-01	2024-12-01	42
1983	1983-01-01	2024-12-01	41
1984	1984-01-01	2024-12-01	40
1985	1985-01-01	2024-12-01	39
1986	1986-01-01	2024-12-01	38
1987	1987-01-01	2024-12-01	37
1988	1988-01-01	2024-12-01	36

Result 35

Output

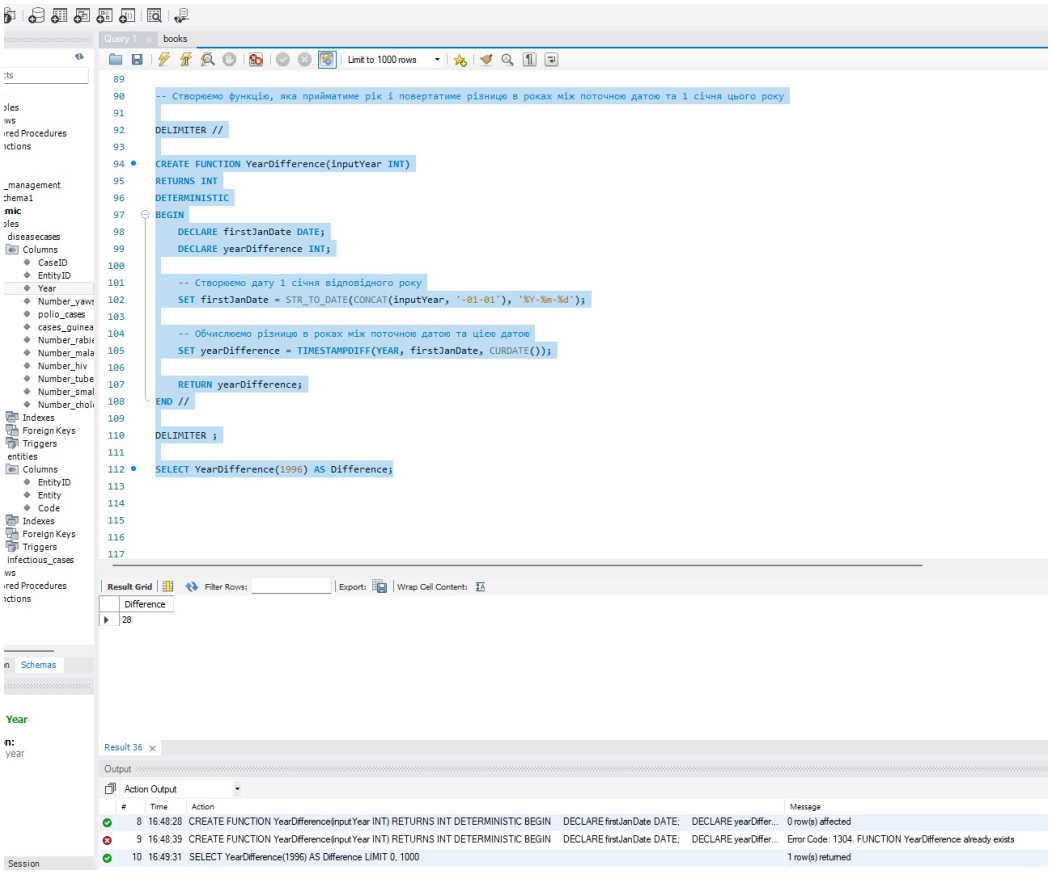
Action Output

#	Time	Action	Message
5	15:23:26	SELECT `Year`, STR_TO_DATE(CONCAT(`Year`, '-01-01'), '%Y-%m-%d') AS Date_1st_Jan, CURDATE() AS `Current_Date`, TIMESTAMPDIF...	10 row(s) returned
6	15:23:57	SELECT DISTINCT Year FROM DiseaseCases LIMIT 0, 1000	41 row(s) returned
7	15:24:08	SELECT `Year`, STR_TO_DATE(CONCAT(`Year`, '-01-01'), '%Y-%m-%d') AS Date_1st_Jan, CURDATE() AS `Current_Date`, TIMESTAMPDIF...	10 row(s) returned

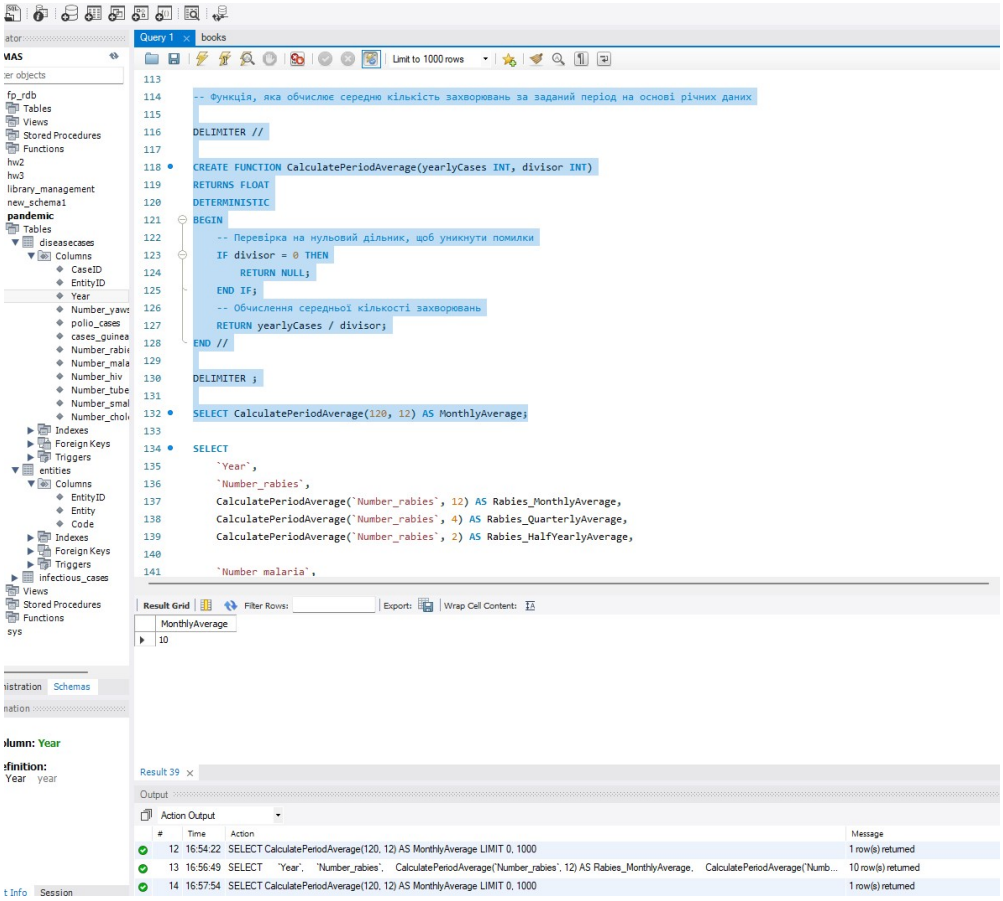
Завдання 5

Створення функцій

Функція, яка приймає рік і повертає різницю в роках між поточною датою та 1 січня цього року



Функція, яка обчислює середню кількість захворювань за заданий період на основі річних даних

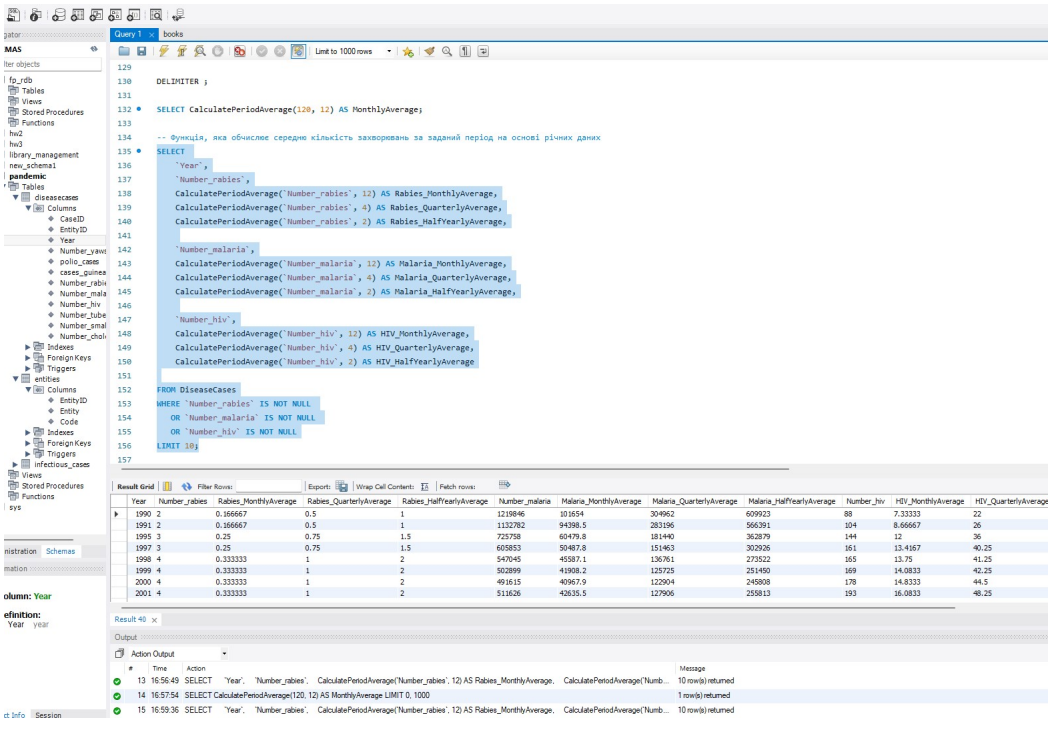


```
-- Функція, яка обчислює середню кількість захворювань за заданий період на основі річних даних

DELIMITER //

CREATE FUNCTION CalculatePeriodAverage(yearlyCases INT, divisor INT)
RETURNS FLOAT
DETERMINISTIC
BEGIN
    -- Перевірка на нульовий дільник, щоб уникнути помилки
    IF divisor = 0 THEN
        RETURN NULL;
    END IF;
    -- Обчислення середньої кількості захворювань
    RETURN yearlyCases / divisor;
END //

DELIMITER ;

SELECT CalculatePeriodAverage(120, 12) AS MonthlyAverage;

SELECT
    `Year`,
    `Number_rabies`,
    CalculatePeriodAverage(`Number_rabies`, 12) AS Rabies_MonthlyAverage,
    CalculatePeriodAverage(`Number_rabies`, 4) AS Rabies_QuarterlyAverage,
    CalculatePeriodAverage(`Number_rabies`, 2) AS Rabies_HalfYearlyAverage,

    `Number_malaria`,
```

MonthlyAverage
10

Output

#	Time	Action	Message
12	16:54:22	SELECT CalculatePeriodAverage(120, 12) AS MonthlyAverage LIMIT 0, 1000	1 row(s) returned
13	16:56:49	SELECT `Year`, `Number_rabies`, CalculatePeriodAverage(`Number_rabies`, 12) AS Rabies_MonthlyAverage, CalculatePeriodAverage(`Numb...	10 row(s) returned
14	16:57:54	SELECT CalculatePeriodAverage(120, 12) AS MonthlyAverage LIMIT 0, 1000	1 row(s) returned



```
DELIMITER ;

SELECT CalculatePeriodAverage(120, 12) AS MonthlyAverage;

-- Функція, яка обчислює середню кількість захворювань за заданий період на основі річних даних
SELECT
    `Year`,
    `Number_rabies`,
    CalculatePeriodAverage(`Number_rabies`, 12) AS Rabies_MonthlyAverage,
    CalculatePeriodAverage(`Number_rabies`, 4) AS Rabies_QuarterlyAverage,
    CalculatePeriodAverage(`Number_rabies`, 2) AS Rabies_HalfYearlyAverage,

    `Number_malaria`,
    CalculatePeriodAverage(`Number_malaria`, 12) AS Malaria_MonthlyAverage,
    CalculatePeriodAverage(`Number_malaria`, 4) AS Malaria_QuarterlyAverage,
    CalculatePeriodAverage(`Number_malaria`, 2) AS Malaria_HalfYearlyAverage,

    `Number_hiv`,
    CalculatePeriodAverage(`Number_hiv`, 12) AS HIV_MonthlyAverage,
    CalculatePeriodAverage(`Number_hiv`, 4) AS HIV_QuarterlyAverage,
    CalculatePeriodAverage(`Number_hiv`, 2) AS HIV_HalfYearlyAverage

FROM DiseaseCases
WHERE `Number_rabies` IS NOT NULL
  OR `Number_malaria` IS NOT NULL
  OR `Number_hiv` IS NOT NULL
LIMIT 10;
```

Year	Number_rabies	Rabies_MonthlyAverage	Rabies_QuarterlyAverage	Rabies_HalfYearlyAverage	Number_malaria	Malaria_MonthlyAverage	Malaria_QuarterlyAverage	Malaria_HalfYearlyAverage	Number_hiv	HIV_MonthlyAverage	HIV_QuarterlyAverage
1990	2	0.166667	0.5	1	1219846	101654	304962	609923	88	7.33333	22
1991	2	0.166667	0.5	1	1132782	94398.5	283196	566391	104	8.66667	26
1995	3	0.25	0.75	1.5	725758	60479.8	181440	362879	144	12	36
1997	3	0.25	0.75	1.5	605853	50487.8	151463	302926	161	13.4167	40.25
1998	4	0.333333	1	2	547045	45587.1	136761	273522	165	13.75	41.25
1999	4	0.333333	1	2	502899	41908.2	125725	251450	169	14.0833	42.25
2000	4	0.333333	1	2	491615	40967.9	122904	245808	178	14.8333	44.5
2001	4	0.333333	1	2	511626	42635.5	127906	255813	193	16.0833	48.25

Output

#	Time	Action	Message
13	16:56:49	SELECT `Year`, `Number_rabies`, CalculatePeriodAverage(`Number_rabies`, 12) AS Rabies_MonthlyAverage, CalculatePeriodAverage(`Numb...	10 row(s) returned
14	16:57:54	SELECT CalculatePeriodAverage(120, 12) AS MonthlyAverage LIMIT 0, 1000	1 row(s) returned
15	16:59:36	SELECT `Year`, `Number_rabies`, CalculatePeriodAverage(`Number_rabies`, 12) AS Rabies_MonthlyAverage, CalculatePeriodAverage(`Numb...	10 row(s) returned