

1.

```
CREATE SCHEMA IF NOT EXISTS pandemic;
```

```
USE pandemic;
```

The screenshot shows a database management tool interface. On the left is a sidebar with a tree view of database objects. The 'Schemas' tab is selected, and the 'pandemic' schema is highlighted. Under 'Tables', 'infectious\_cases' is listed. The main pane on the right shows a list of SQL commands being executed:

- 1 • CREATE SCHEMA IF NOT EXISTS pandemic;
- 2 • USE pandemic;
- 3

Below the commands, there is a section for 'Action Output' showing a log of execution results:

		Time	Response
✓	484	16:32:31	S OK
✓	485	16:32:...	C OK
✓	486	16:32:...	P OK
✓	487	16:33:...	D OK
✓	488	16:34:...	S 1000 row(s) returned

2.

```
CREATE TABLE infectious_cases_data (  
  CaseID INT PRIMARY KEY AUTO_INCREMENT,  
  Year INT,  
  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  
);  
SET sql_mode =  
'STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZE  
RO,NO_ENGINE_SUBSTITUTION';
```

```
CREATE TABLE Entity_Code_mapping (  
    EntityID INT PRIMARY KEY AUTO_INCREMENT,  
    EntityName VARCHAR(255) NOT NULL,  
    Code VARCHAR(50) NOT NULL  
);
```

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

```
SELECT Year,  
    NULLIF(Number_yaws, ''),  
    polio_cases,  
    cases_guinea_worm,  
    NULLIF(Number_rabies, ''),  
    NULLIF(Number_malaria, ''),  
    NULLIF(Number_hiv, ''),  
    NULLIF(Number_tuberculosis, ''),  
    NULLIF(Number_smallpox, ''),  
    NULLIF( Number_cholera_cases, '')
```

```
FROM infectious_cases;
```

```
INSERT INTO Entity_Code_mapping (EntityName, Code)  
SELECT DISTINCT Entity, Code  
FROM infectious_cases;
```

```

1 CREATE TABLE infectious_cases_data (
2   CaseID INT PRIMARY KEY AUTO_INCREMENT,
3   Year INT,
4   Number_yaws INT,
5   polio_cases INT,
6   cases_guinea_worm INT,
7   Number_rabies INT,
8   Number_malaria INT,
9   Number_hiv INT,
10  Number_tuberculosis INT,
11  Number_smallpox INT,
12  Number_cholera_cases INT
13 );
14
15 SET sql_mode = 'STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
16
17 CREATE TABLE Entity_Code_mapping (
18   EntityID INT PRIMARY KEY AUTO_INCREMENT,
19   EntityName VARCHAR(255) NOT NULL,
20   Code VARCHAR(50) NOT NULL
21 );
22
23 INSERT INTO infectious_cases_data (Year, Number_yaws, polio_cases, cases_guinea_worm, Number_rabies, Number_malaria, Number_hiv, Number_tuberculosis, Number_smallpox, Number_cholera_cases)
24 SELECT Year,
25        NULLIF(Number_yaws, ''),
26        polio_cases,
27        cases_guinea_worm,
28        NULLIF(Number_rabies, ''),
29        NULLIF(Number_malaria, ''),
30        NULLIF(Number_hiv, ''),
31        NULLIF(Number_tuberculosis, ''),
32        NULLIF(Number_smallpox, ''),
33        NULLIF( Number_cholera_cases, '')
34 FROM infectious_cases;

```

```

22 • INSERT INTO infectious_cases_data (Year, Number_yaws, polio_cases, cases_guinea_worm, Number_rabies, Number_malaria, Number_hiv, Number_tuberculosis, Number_smallpox, Number_cholera_cases)
23 SELECT Year,
24        NULLIF(Number_yaws, ''),
25        polio_cases,
26        cases_guinea_worm,
27        NULLIF(Number_rabies, ''),
28        NULLIF(Number_malaria, ''),
29        NULLIF(Number_hiv, ''),
30        NULLIF(Number_tuberculosis, ''),
31        NULLIF(Number_smallpox, ''),
32        NULLIF( Number_cholera_cases, '')
33 FROM infectious_cases;
34
35 • INSERT INTO Entity_Code_mapping (EntityName, Code)
36 SELECT DISTINCT Entity, Code
37 FROM infectious_cases;

```

100% 1:1

Result Grid Filter Rows: Search Edit:

	EntityID	EntityName	Code	
	1	Afghanistan	AFG	
	2	Albania	ALB	
	3	Algeria	DZA	
	4	Andorra	AND	
	5	Angola	AGO	
	6	Antigua and Barbuda	ATG	
	7	Argentina	ARG	
	8	Armenia	ARM	
	9	Australia	AUS	
	10	Austria	AUT	
	11	Azerbaijan	AZE	
	12	Bahamas	BHS	
	13	Bahrain	BHR	
	14	Bangladesh	BGD	
	15	Barbados	BRB	
	16	Belarus	BLR	
	17	Belgium	BEL	
	18	Belize	BLZ	

Entity\_Code\_mapping 1

100%

1:1

Result Grid

Filter Rows:

Edit:

Export/Import:

Fetch rows:

	CaseID	Year	Number_yaws	polio_cases	cases_guinea_worm	Number_rabies	Number_malaria	Number_hiv	Number_tuberculo...	Number_smallpox	Number_cholera_cas...
	1	1980	NULL	6160	0	NULL	NULL	NULL	NULL	NULL	
	2	1981	NULL	5859	0	NULL	NULL	NULL	NULL	NULL	
	3	1982	NULL	9730	0	NULL	NULL	NULL	NULL	NULL	
	4	1983	NULL	13937	0	NULL	NULL	NULL	NULL	NULL	
	5	1984	NULL	3864	0	NULL	NULL	NULL	NULL	NULL	
	6	1985	NULL	13867	0	NULL	NULL	NULL	NULL	NULL	
	7	1986	NULL	12901	0	NULL	NULL	NULL	NULL	NULL	
	8	1987	NULL	4396	0	NULL	NULL	NULL	NULL	NULL	
	9	1988	NULL	2149	0	NULL	NULL	NULL	NULL	NULL	
	10	1989	NULL	385	0	NULL	NULL	NULL	NULL	NULL	
	11	1990	NULL	336	0	2	1219846	88	16303	NULL	NULL
	12	1991	NULL	14	0	2	1132782	104	17582	NULL	NULL
	13	1995	NULL	0	0	3	725758	144	22163	NULL	19903
	14	1997	NULL	21	0	3	605853	161	23141	NULL	4170
	15	1998	NULL	65	0	4	547045	165	23179	NULL	10000
	16	1999	NULL	167	0	4	502899	169	22944	NULL	24639
	17	2000	NULL	133	0	4	491615	178	22880	NULL	4330
	18	2001	NULL	22	0	4	511626	193	23330	NULL	4499

infectious\_cases\_data 1

3.1

SELECT

Entity,

Code,

AVG(NULLIF(Number\_rabies, '')) AS Avg\_Number\_rabies,

MIN(NULLIF(Number\_rabies, '')) AS Min\_Number\_rabies,

```

MAX(NULLIF(Number_rabies, '')) AS Max_Number_rabies,
SUM(NULLIF(Number_rabies, '')) AS Sum_Number_rabies
FROM infectious_cases
WHERE Number_rabies IS NOT NULL AND Number_rabies <> ''
GROUP BY Entity, Code;

```

The screenshot shows a database management interface with a left sidebar containing a tree view of schemas and tables. The main area displays a SQL query (Query 58) and its results in a grid format. The query is as follows:

```

38
39
40 • SELECT
41     Entity,
42     Code,
43     AVG(NULLIF(Number_rabies, '')) AS Avg_Number_rabies,
44     MIN(NULLIF(Number_rabies, '')) AS Min_Number_rabies,
45     MAX(NULLIF(Number_rabies, '')) AS Max_Number_rabies,
46     SUM(NULLIF(Number_rabies, '')) AS Sum_Number_rabies
47 FROM infectious_cases
48 WHERE Number_rabies IS NOT NULL AND Number_rabies <> ''
49 GROUP BY Entity, Code;

```

The results are displayed in a table with the following columns: Entity, Code, Avg\_Number\_rabies, Min\_Number\_rabies, Max\_Number\_rabies, and Sum\_Number\_rabies. The table contains 10 rows of data for various countries.

Entity	Code	Avg_Number_rabies	Min_Number_rabies	Max_Number_rabies	Sum_Number_rabies
Afghanistan	AFG	3.469552092307692	2.145772	3.9896567	90.20835439999999
Albania	ALB	0.01645030461666665	0.0102469465	0.024485007	0.49350913849999994
Algeria	DZA	1.5353912600000001	1.3461609	1.7746575	46.0617378
Andorra	AND	0.0000968316723809525	0.000071659604	0.00013845737	0.0020933465120000004
Angola	AGO	12.907648586206896	10.417461	9.800037	374.321809
Antigua an...	ATG	0.00163001972	0.0010932273	0.0032946493	0.0326003944
Argentina	ARG	0.6520745666666667	0.51903343	1.5078027	19.562237000000003
Armenia	ARM	8.898946555517242	0.34852454	8.69053	258.06945011
Australia	AUS	0.02497951083333333	0.010655328	0.055543594	0.749385325
Austria	AUT	0.00724464257333332	0.005048178	0.009986834	0.21733927719999996
Azerbaijan	AZE	10.630334596551723	10.785144	9.632432	308.2797033
Bahamas	BHS	0.0014024708950000002	0.0010848407	0.0021752398	0.028049417900000005

3.2

```

SELECT
    Entity,
    Code,
    AVG(NULLIF(Number_rabies, '')) AS Avg_Number_rabies,
    MIN(NULLIF(Number_rabies, '')) AS Min_Number_rabies,
    MAX(NULLIF(Number_rabies, '')) AS Max_Number_rabies,
    SUM(NULLIF(Number_rabies, '')) AS Sum_Number_rabies
FROM infectious_cases
WHERE Number_rabies IS NOT NULL AND Number_rabies <> ''
GROUP BY Entity, Code
ORDER BY Avg_Number_rabies DESC;

```

51

52 • **SELECT**

53     Entity,

54     Code,

55     AVG(NULLIF(Number\_rabies, '')) AS Avg\_Number\_rabies,

56     MIN(NULLIF(Number\_rabies, '')) AS Min\_Number\_rabies,

57     MAX(NULLIF(Number\_rabies, '')) AS Max\_Number\_rabies,

58     SUM(NULLIF(Number\_rabies, '')) AS Sum\_Number\_rabies

59 **FROM** infectious\_cases

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

61 **GROUP BY** Entity, Code

62 **ORDER BY** Avg\_Number\_rabies **DESC**;

100% 22:61

**Result Grid** Filter Rows: Search Export:

Entity	Code	Avg_Number_rabies	Min_Number_rabi...	Max_Number_rabi...	Sum_Number_rabies
Pakistan	PAK	1582.1696266666663	1177.1449	1881.7257	47465.08879999999
Nigeria	NGA	1335.4154866666667	1073.9458	1441.1188	40062.46460000001
China	CHN	1247.906733	1044.7965	977.9138	37437.20199
Ethiopia	ETH	1145.1725223333333	1008.1017	987.63824	34355.17567
Myanmar	MMR	972.9036866666668	1022.31464	992.23914	29187.110600000004
Bangladesh	BGD	785.7088230000002	1033.1205	911.6814	23571.264690000004
Nepal	NPL	661.2508643333334	1019.73895	997.43427	19837.525930000003
Philippines	PHL	618.6910056666668	358.0602	898.071	18560.730170000003
Indonesia	IDN	267.71476083333334	124.621735	481.37482	8031.442825
Tanzania	TZA	254.6958748275862	169.13283	335.68637	7386.18037
Ghana	GHA	190.67501733333333	164.81816	211.28151	5720.25052
Niger	NER	190.1975356666667	161.07932	225.82402	5705.926070000001

Result 2

3.3

SELECT

Entity,

Code,

AVG(NULLIF(Number\_rabies, '')) AS Avg\_Number\_rabies,

MIN(NULLIF(Number\_rabies, '')) AS Min\_Number\_rabies,

MAX(NULLIF(Number\_rabies, '')) AS Max\_Number\_rabies,

SUM(NULLIF(Number\_rabies, '')) AS Sum\_Number\_rabies

FROM infectious\_cases

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

GROUP BY Entity, Code

ORDER BY Avg\_Number\_rabies DESC

LIMIT 10;

Administration	Schemas	Query 58	Entity_Code_mapping	infectious_cases_data
objects				
V3				
LibraryManagement				
ydb				
andemic				
Tables				
Entity_Code_mapping				
infectious_cases				
infectious_cases_data				
Views				
Stored Procedures				
Functions				
s				

  

```

65 • SELECT
66     Entity,
67     Code,
68     AVG(NULLIF(Number_rabies, '')) AS Avg_Number_rabies,
69     MIN(NULLIF(Number_rabies, '')) AS Min_Number_rabies,
70     MAX(NULLIF(Number_rabies, '')) AS Max_Number_rabies,
71     SUM(NULLIF(Number_rabies, '')) AS Sum_Number_rabies
72 FROM infectious_cases
73 WHERE Number_rabies IS NOT NULL AND Number_rabies <> ''
74 GROUP BY Entity, Code
75 ORDER BY Avg_Number_rabies DESC
76 LIMIT 10;

```

  

Entity	Code	Avg_Number_rabies	Min_Number_rabi...	Max_Number_rabi...	Sum_Number_rabies
World	OWID_WRL	20192.370366666666	14075.508	24744.658	605771.111
India	IND	8599.173279999999	10190.644	9964.957	257975.19839999996
Pakistan	PAK	1582.1696266666663	1177.1449	1881.7257	47465.08879999999
Nigeria	NGA	1335.4154866666667	1073.9458	1441.1188	40062.46460000001
China	CHN	1247.906733	1044.7965	977.9138	37437.20199
Ethiopia	ETH	1145.1725223333333	1008.1017	987.63824	34355.17567
Myanmar	MMR	972.9036866666668	1022.31464	992.23914	29187.110600000004
Bangladesh	BGD	785.7088230000002	1033.1205	911.6814	23571.264690000004
Nepal	NPL	661.2508643333334	1019.73895	997.43427	19837.525930000003
Philippines	PHL	618.6910056666668	358.0602	898.071	18560.730170000003

  

Result 3
----------

  

Action Output
Time
Response

4.SELECT

Year,

STR\_TO\_DATE(CONCAT(Year, '-01-01'), '%Y-%m-%d') AS First\_January\_Date,

CURDATE() AS CurrentDate,

TIMESTAMPDIFF(YEAR, STR\_TO\_DATE(CONCAT(Year, '-01-01'), '%Y-%m-%d'), CURDATE()) AS  
Difference\_In\_Years

FROM pandemic.infectious\_cases\_data;

Limit to 1000 rows

```

1 • SELECT * FROM pandemic.infectious_cases_data;
2 • SELECT
3     Year,
4     STR_TO_DATE(CONCAT(Year, '-01-01'), '%Y-%m-%d') AS First_January_Dat
5     CURDATE() AS CurrentDate,
6     TIMESTAMPDIFF(YEAR, STR_TO_DATE(CONCAT(Year, '-01-01'), '%Y-%m-%d'),
7 FROM pandemic.infectious_cases_data;
8
9
10 DELIMITER //
11
12 • CREATE FUNCTION CalculateDiseaseCountPerPeriod(total_cases INT, divider
13 RETURNS DECIMAL(10, 2)
14 DETERMINISTIC
15 BEGIN
16     DECLARE count_per_period DECIMAL(10, 2);
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

100% 37:7

Result Grid Filter Rows: Search Export: Fetch rows:

	Year	First_January_Date	CurrentDate	Difference_In_Years
	1980	1980-01-01	2024-06-26	44
	1981	1981-01-01	2024-06-26	43
	1982	1982-01-01	2024-06-26	42
	1983	1983-01-01	2024-06-26	41
	1984	1984-01-01	2024-06-26	40
	1985	1985-01-01	2024-06-26	39
	1986	1986-01-01	2024-06-26	38
	1987	1987-01-01	2024-06-26	37
	1988	1988-01-01	2024-06-26	36
	1989	1989-01-01	2024-06-26	35
	1990	1990-01-01	2024-06-26	34
	1991	1991-01-01	2024-06-26	33
	1995	1995-01-01	2024-06-26	29
	1997	1997-01-01	2024-06-26	27
	1998	1998-01-01	2024-06-26	26
	1999	1999-01-01	2024-06-26	25
	2000	2000-01-01	2024-06-26	24
	2001	2001-01-01	2024-06-26	23

Result 5

5.

DELIMITER //

CREATE FUNCTION CalculateDiseaseCountPerPeriod(total\_cases INT, divider INT)

RETURNS DECIMAL(10, 2)

DETERMINISTIC

BEGIN



```
DECLARE count_per_period DECIMAL(10, 2);

-- Перевірка, щоб уникнути ділення на нуль
IF divider <= 0 THEN
    RETURN NULL;
END IF;

-- Обчислення середньої кількості захворювань на певний період
SET count_per_period = total_cases / divider;

RETURN count_per_period;
END//

DELIMITER ;

SELECT
    Year,
    Number_rabies,
    CalculateDiseaseCountPerPeriod(Number_rabies, 12) AS Avg_Cases_Per_Month
FROM infectious_cases_data
WHERE Number_rabies != '';
```

jects

ryManagement

b

lemic

bles

Entity\_Code\_mapping

infectious\_cases

infectious\_cases\_data

aws

ored Procedures

nctions

CalculateDiseaseCountPerPe...

```
7 FROM pandemic.infectious_cases_data;
8
9
10 DELIMITER //
11
12 • CREATE FUNCTION CalculateDiseaseCountPerPeriod(total_cases INT, divider INT)
13 RETURNS DECIMAL(10, 2)
14 DETERMINISTIC
15 BEGIN
16     DECLARE count_per_period DECIMAL(10, 2);
17
18     -- Перевірка, щоб уникнути ділення на нуль
19     IF divider <= 0 THEN
20         RETURN NULL;
21     END IF;
22
23     -- Обчислення середньої кількості захворювань на певний період
24     SET count_per_period = total_cases / divider;
25
26     RETURN count_per_period;
27 END//
28
29 DELIMITER ;
30
31 • SELECT
32     Year,
33     Number_rabies,
34     CalculateDiseaseCountPerPeriod(Number_rabies, 12) AS Avg_Cases_Per_Month
35 FROM infectious_cases_data
36 WHERE Number_rabies != '';
37
38
39
```

100% 1:11

Action Output

	Time	Response
622	18:24:16	C Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use
623	18:24:16	C Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use
624	18:26:01	C Error Code: 1418. This function has none of DETERMINISTIC, NO SQL, or READS SQL DATA in its declaration and binary logging is enabled (you *mi
625	18:27:21	C 0 row(s) affected
626	18:27:21	S 1000 row(s) returned

leted

Result Grid			
Filter Rows:			
SEARCH			
Export			
Year	Number_rabies	Avg_Cases_Per_Month	
1990	2	0.17	
1991	2	0.17	
1995	3	0.25	
1997	3	0.25	
1998	4	0.33	
1999	4	0.33	
2000	4	0.33	
2001	4	0.33	
2002	4	0.33	
2003	4	0.33	
2004	4	0.33	
2005	4	0.33	
2006	4	0.33	
2007	4	0.33	
2008	4	0.33	
2009	4	0.33	
2010	3	0.25	
2011	3	0.25	
Result 4			