

goit-rdb-fp

Завантажити базу

```
mysql -u root -p --local-infile=1 < db.sql
```

Виконання запитів

```
mysql -u root -p < queries.sql
```

1. Завантажте дані

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

```
DROP SCHEMA IF EXISTS `pandemic`;  
CREATE SCHEMA IF NOT EXISTS `pandemic` DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci;
```

- Оберіть її як схему за замовчуванням за допомогою SQL-команди.

```
USE `pandemic`;
```

- Імпортуйте дані за допомогою Import wizard так, як ви вже робили це у темі 3.

Дані імпортуються стандартними командами SQL з заміною пустих рядків на NULL

```

DROP TABLE IF EXISTS `infectious_cases_original`;
CREATE TABLE IF NOT EXISTS `infectious_cases_original` (
  `entity` VARCHAR(100) NOT NULL,
  `code` VARCHAR(10) NULL,
  `year` INT NOT NULL,
  `number_yaws` DECIMAL(12,2) NULL,
  `polio_cases` DECIMAL(12,2) NULL,
  `cases_guinea_worm` DECIMAL(12,2) NULL,
  `number_rabies` DECIMAL(12,2) NULL,
  `number_malaria` DECIMAL(12,2) NULL,
  `number_hiv` DECIMAL(12,2) NULL,
  `number_tuberculosis` DECIMAL(12,2) NULL,
  `number_smallpox` DECIMAL(12,2) NULL,
  `number_cholera_cases` DECIMAL(12,2) NULL
);

LOAD DATA LOCAL INFILE 'data/infectious_cases.csv'
INTO TABLE `infectious_cases_original`
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 LINES
(`entity`, `code`, `year`, @number_yaws, @polio_cases, @cases_guinea_worm,
@number_rabies, @number_malaria, @number_hiv, @number_tuberculosis,
@number_smallpox, @number_cholera_cases)
SET
  `number_yaws` = NULLIF(@number_yaws, ''),
  `polio_cases` = NULLIF(@polio_cases, ''),
  `cases_guinea_worm` = NULLIF(@cases_guinea_worm, ''),
  `number_rabies` = NULLIF(@number_rabies, ''),
  `number_malaria` = NULLIF(@number_malaria, ''),
  `number_hiv` = NULLIF(@number_hiv, ''),
  `number_tuberculosis` = NULLIF(@number_tuberculosis, ''),
  `number_smallpox` = NULLIF(@number_smallpox, ''),
  `number_cholera_cases` = NULLIF(@number_cholera_cases, '');

```

- Продивіться дані, щоб бути у контексті.

```

SELECT COUNT(*) AS total_records
FROM infectious_cases_original;


```



```

-----
total_records
10521

```

```
4 • SELECT COUNT(*) AS total_records
5 FROM infectious_cases_original;
6
```

100%  32:5

Result Grid   Filter Rows: E

	total_records	
	10521	

2. Нормалізація

Нормалізуйте таблицю `infectious_cases` до 3ї нормальної форми. Збережіть у цій же схемі дві таблиці з нормалізованими даними.

- Схеми

```

DROP TABLE IF EXISTS `entities`;
CREATE TABLE IF NOT EXISTS `entities` (
  `entity_id` INT NOT NULL AUTO_INCREMENT,
  `entity` VARCHAR(100) NOT NULL,
  `code` VARCHAR(10) NULL,
  PRIMARY KEY (`entity_id`),
  UNIQUE KEY `uk_entity_code` (`entity`, `code`)
);

DROP TABLE IF EXISTS `infectious_cases`;
CREATE TABLE IF NOT EXISTS `infectious_cases` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `entity_id` INT NOT NULL,
  `year` INT NOT NULL,
  `number_yaws` DECIMAL(12,2) NULL,
  `polio_cases` DECIMAL(12,2) NULL,
  `cases_guinea_worm` DECIMAL(12,2) NULL,
  `number_rabies` DECIMAL(12,2) NULL,
  `number_malaria` DECIMAL(12,2) NULL,
  `number_hiv` DECIMAL(12,2) NULL,
  `number_tuberculosis` DECIMAL(12,2) NULL,
  `number_smallpox` DECIMAL(12,2) NULL,
  `number_cholera_cases` DECIMAL(12,2) NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `uk_entity_year` (`entity_id`, `year`),
  FOREIGN KEY (`entity_id`) REFERENCES `entities` (`entity_id`)
);

```

- Дані

```

INSERT INTO entities (entity, code)
SELECT DISTINCT entity, code
FROM infectious_cases_original;

INSERT INTO infectious_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.entity_id, o.year,
o.number_yaws, o.polio_cases, o.cases_guinea_worm,
o.number_rabies, o.number_malaria, o.number_hiv,
o.number_tuberculosis, o.number_smallpox, o.number_cholera_cases
FROM infectious_cases_original o
JOIN entities e ON o.entity = e.entity AND COALESCE(o.code, '') = COALESCE(e.code,

```

- Перевірка

```

SELECT COUNT(*) AS total_records
FROM infectious_cases;

```

```

-----
total_records
10521

```

8

•

SELECT

COUNT(*)

AS

total_records

9

FROM

infectious_cases;


10


00%

↕

22:9

Result Grid





Filter Rows:

Q

Search

Exp

	total_records	
	10521	

3. Проаналізуйте дані

- Для кожної унікальної комбінації Entity та Code або їх id порахуйте середнє, мінімальне, максимальне значення та суму для атрибута Number_rabies.
- Результат відсортуйте за порохованим середнім значенням у порядку спадання.
- Оберіть тільки 10 рядків для виведення на екран.

```
SELECT
    e.entity,
    e.code,
    AVG(ic.number_rabies) AS avg_rabies,
    MIN(ic.number_rabies) AS min_rabies,
    MAX(ic.number_rabies) AS max_rabies,
    SUM(ic.number_rabies) AS total_rabies
FROM infectious_cases ic
JOIN entities e ON ic.entity_id = e.entity_id
WHERE ic.number_rabies IS NOT NULL
GROUP BY e.entity, e.code
ORDER BY avg_rabies DESC
LIMIT 10;
```

```
-----
entity                                code          avg_rabies      min_rabies      m
World                                OWID_WRL      20192.370667    14075.51
Lower Middle Income (WB)              15193.959333    10202.53
South Asia (WB)                       11729.889667    7271.28
South-East Asia Region (WHO)          11424.327667    6806.01
G20                                   10189.046000    6339.08
India                                IND           8599.172667    5425.87
Sub-Saharan Africa (WB)              4748.268333    4356.82
African Region (WHO)                 4649.702000    4283.07
Low Income (WB)                      3568.106000    3014.62
East Asia & Pacific (WB)             3427.461000    2009.94
```

```

12 • SELECT
13     e.entity,
14     e.code,
15     AVG(ic.number_rabies) AS avg_rabies,
16     MIN(ic.number_rabies) AS min_rabies,
17     MAX(ic.number_rabies) AS max_rabies,
18     SUM(ic.number_rabies) AS total_rabies
19 FROM infectious_cases ic
20 JOIN entities e ON ic.entity_id = e.entity_id
21 WHERE ic.number_rabies IS NOT NULL
22 GROUP BY e.entity, e.code
23 ORDER BY avg_rabies DESC
24 LIMIT 10;

```

100% 9:24

Result Grid

Filter Rows:

Export:

Fetch rows:

entity	code	avg_rabies	min_rabies	max_rabies	total_rabi...
World	OWID_WRL	20192.370667	14075.51	24744.66	605771.12
Lower Middle Income (WB)		15193.959333	10202.53	19182.80	455818.78
South Asia (WB)		11729.889667	7271.28	15361.88	351896.69
South-East Asia Region (WHO)		11424.327667	6806.01	15641.96	342729.83
G20		10189.046000	6339.08	13164.88	305671.38
India	IND	8599.172667	5425.87	11121.14	257975.18
Sub-Saharan Africa (WB)		4748.268333	4356.82	5017.98	142448.05
African Region (WHO)		4649.702000	4283.07	4917.78	139491.06
Low Income (WB)		3568.106000	3014.62	3873.53	107043.18
East Asia & Pacific (WB)		3427.461000	2009.94	4591.01	102823.83

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,
- атрибут, що дорівнює поточній даті,
- атрибут, що дорівнює різниці в роках двох вищезгаданих колонок.

Виконання

- Схема

```
ALTER TABLE infectious_cases
ADD COLUMN year_start_date DATE NULL,
ADD COLUMN today_date DATE NULL,
ADD COLUMN years_difference INT NULL;
```

- Дані

Поточна дата зберігається в змінну перед заповненням оскільки під час виконання CURDATE() буде повертати різне значення для різних рядків

```
SET @current_date = CURDATE();
```

```
UPDATE infectious_cases
SET
    year_start_date = STR_TO_DATE(CONCAT(year, '-01-01'), '%Y-%m-%d'),
    today_date = @current_date,
    years_difference = TIMESTAMPDIFF(YEAR, year_start_date, @current_date);
```

- Перевірка

```
SELECT
    year,
    year_start_date,
    today_date,
    years_difference
FROM infectious_cases
LIMIT 10;
```

```
-----
year      year_start_date  today_date  years_difference
1980      1980-01-01      2025-07-19  45
1981      1981-01-01      2025-07-19  44
1982      1982-01-01      2025-07-19  43
1983      1983-01-01      2025-07-19  42
1984      1984-01-01      2025-07-19  41
1985      1985-01-01      2025-07-19  40
1986      1986-01-01      2025-07-19  39
1987      1987-01-01      2025-07-19  38
1988      1988-01-01      2025-07-19  37
1989      1989-01-01      2025-07-19  36
```





```

27 • SELECT
28     year,
29     year_start_date,
30     today_date,
31     years_difference
32 FROM infectious_cases
33 LIMIT 10;

```

100% 9:33

Result Grid   Filter Rows:

	year	year_start_da...	today_date	years_differen...
	1980	1980-01-01	2025-07-19	45
	1981	1981-01-01	2025-07-19	44
	1982	1982-01-01	2025-07-19	43
	1983	1983-01-01	2025-07-19	42
	1984	1984-01-01	2025-07-19	41
	1985	1985-01-01	2025-07-19	40
	1986	1986-01-01	2025-07-19	39
	1987	1987-01-01	2025-07-19	38
	1988	1988-01-01	2025-07-19	37
	1989	1989-01-01	2025-07-19	36

5. Побудуйте власну функцію

Створіть і використайте функцію, що буде 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').

- Схема

Функція не є DETERMISTIC оскільки поточна дата є зовнішнім змінним станом і функція може повертати різне значення при тих же параметрах.

```
DELIMITER //
```

```
CREATE FUNCTION calculate_years_difference(input_year INT)
RETURNS INT
NO SQL
BEGIN
    DECLARE input_date DATE;
    SET input_date = STR_TO_DATE(CONCAT(input_year, '-01-01'), '%Y-%m-%d');
    RETURN TIMESTAMPDIFF(YEAR, input_date, CURDATE());
END //
```

```
DELIMITER ;
```

- Перевірка

```
SELECT calculate_years_difference(1996) AS years_from_1996;
```

years_from_1996

29

```
36 • SELECT calculate_years_difference(1996) AS years_from_1996;
```

27

100% 59:36

Result Grid



Filter Rows:



Search

Export:



years_from_1996

29
