

Лабораторна робота № 7
“Нормалізація бази даних”
з курсу “Організація баз даних та знань”

Виконав:

Студент групи ФЕС-21с

Шавало А. А.

Викладач: асист. Галяткін О. О.

Львів 2024

Лабораторна робота №7. Нормалізація бази даних

Мета роботи: навчитись аналізувати структуру бази даних для визначення її нормальної форми та здійснювати нормалізацію бази даних у разі необхідності.

Хід роботи

1. Запустити аналіз розробленої на попередніх лабораторних роботах бази даних на відповідність наступним нормальним формам:

- a) перша нормальна форма (1NF);
- b) друга нормальна форма (2NF);
- c) третя нормальна форма (3NF);
- d) нормальна форма Бойса-Кодда (BCNF);
- e) четверта нормальна форма (4NF).

Для таблиці - cases

	case_id	client_id	lawyer_id	case_type	article_code	punishment_min	punishment_max	fee	case_status	case_result	received_punishment
▶	1	1	1	Кримінальна справа	ККУ ст. 115	7	15	10000.00	завершена	виправдальний вирок	0
	2	2	2	Цивільна справа	ЦКУ ст. 203	0	0	5000.00	завершена		NULL
	3	3	3	Адміністративна справа	КУпАП ст. 173	0	0	3000.00	завершена	штраф	500
	4	4	4	Кримінальна справа	ККУ ст. 187	5	10	12000.00	в процесі		NULL
	5	5	5	Цивільна справа	ЦКУ ст. 215	0	0	4000.00	завершена	мировий договір	0
	6	6	6	Кримінальна справа	ККУ ст. 289	3	7	8000.00	в процесі		NULL
	7	7	7	Адміністративна справа	КУпАП ст. 122	0	0	2000.00	завершена	штраф	200
	8	8	8	Цивільна справа	ЦКУ ст. 376	0	0	6000.00	завершена	відмова у позові	0
	9	9	9	Кримінальна справа	ККУ ст. 186	4	8	9500.00	в процесі		NULL
	10	10	10	Кримінальна справа	ККУ ст. 187	5	10	10000.00	в процесі	NULL	0
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

a) Перша нормальна форма (1NF)

- Дані відповідають вимогам 1NF, оскільки всі поля мають атомарні значення
- Кожен рядок є унікальним через унікальний case_id

б) Друга нормальна форма (2NF)

Таблиця відповідає 1NF, тому перевіряємо залежності:

- Усі неключові атрибути повністю залежать від `case_id`.
- Немає часткових залежностей

с) Третя нормальна форма (3NF)

Дані відповідають 2NF.

- Однак, тут можна виявити можливі транзитивні залежності, наприклад:
- `punishment_min` і `punishment_max` можуть залежати від `case_type`, якщо тип справи визначає можливі покарання. Це може означати, що `punishment_min` і `punishment_max` не повинні бути безпосередньо пов'язані з `case_id`, а прописані через `case_type`.

	case_id	client_id	lawyer_id	article_code	fee	case_status	case_result	received_punishment	case_type_id
	1	1	1	ККУ ст. 115	10000.00	завершена	виправдальний вирок	0	1
	2	2	2	ЦКУ ст. 203	5000.00	завершена		NULL	2
	3	3	3	КУпАП ст. 173	3000.00	завершена	штраф	500	3
	4	4	4	ККУ ст. 187	12000.00	в процесі		NULL	4
	5	5	5	ЦКУ ст. 215	4000.00	завершена	мировий договір	0	5
	6	6	6	ККУ ст. 289	8000.00	в процесі		NULL	6
	7	7	7	КУпАП ст. 122	2000.00	завершена	штраф	200	7
	8	8	8	ЦКУ ст. 376	6000.00	завершена	відмова у позові	0	8
	9	9	9	ККУ ст. 186	9500.00	в процесі		NULL	9
▶	10	10	10	ККУ ст. 187	10000.00	в процесі	NULL	0	10
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

	case_type_id	case_type	punishment_min	punishment_max
	1	Кримінальна справа	3	15
	2	Цивільна справа	0	0
	3	Адміністративна справа	0	0
▶	4	Кримінальна справа	3	15
	5	Цивільна справа	0	0
	6	Адміністративна справа	0	0
	7	Кримінальна справа	3	5
	8	Цивільна справа	1	2
	9	Адміністративна справа	2	5
	10	Кримінальна справа	1	5
*	NULL	NULL	NULL	NULL

д) Нормальна форма Бойса-Кодда (BCNF)

Відповідає: Для всіх залежностей, лівий бік є суперключем.

е) Четверта нормальна форма (4NF)

Відповідає: Немає множинних незалежних залежностей.

Для таблиці - clients

	id	full_name	birth_date	address	phone	status
▶	1	Коваленко Марія Петрівна	1990-03-22	м. Одеса, вул. Дерибасівська, 15	+380631234567	фізична особа
	2	Товариство "Юридична допомога"	2005-07-11	м. Харків, вул. Сумська, 100	+380661234567	юридична особа
	3	Іваненко Василь Петрович	1980-04-25	м. Київ, вул. Володимирська, 18	+380501234768	фізична особа
	4	ООО "Медсервіс"	2010-05-18	м. Дніпро, вул. Леніна, 20	+380671234668	юридична особа
	5	Захарченко Антон Сергійович	1985-10-17	м. Львів, вул. Стрийська, 9	+380931234568	фізична особа
	6	ПП "Будмонтаж"	2012-08-22	м. Одеса, вул. Фонтанська, 3	+380661234568	юридична особа
	7	Степаненко Марія Вікторівна	1992-02-14	м. Вінниця, вул. Соборна, 21	+380711234568	фізична особа
	8	ООО "Агросоюз"	2015-01-12	м. Полтава, вул. Жовтнева, 55	+380991234568	юридична особа
	9	Сергієнко Олексій Іванович	1983-12-03	м. Чернігів, вул. Кошобинського, 16	+380951234568	фізична особа
	10	Невідомий Клієнт	1980-01-01	м. Київ, вул. Незалежності, 1	+380991112233	фізична особа
✱	NULL	NULL	NULL	NULL	NULL	NULL

а) Перша нормальна форма (1NF):

Не відповідає:

- Кожен атрибут має одне значення. Full_name має декілька значень

```

1 CREATE TABLE clients_normalized (
2     id SMALLINT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
3     last_name VARCHAR(255) DEFAULT NULL,
4     first_name VARCHAR(255) DEFAULT NULL,
5     middle_name VARCHAR(255) DEFAULT NULL,
6     company_name VARCHAR(255) DEFAULT NULL,
7     birth_date DATE,
8     address VARCHAR(255),
9     phone VARCHAR(20),
10    status ENUM('фізична особа', 'юридична особа')
11 );
12
13 INSERT INTO clients_normalized (id, last_name, first_name, middle_name, birth_date, address, phone, status)
14 SELECT
15     id,
16     SUBSTRING_INDEX(full_name, ' ', 1) AS last_name,
17     SUBSTRING_INDEX(SUBSTRING_INDEX(full_name, ' ', 2), ' ', -1) AS first_name,
18     SUBSTRING_INDEX(full_name, ' ', -1) AS middle_name,
19     birth_date,
20     address,
21     phone,
22     status
23 FROM clients
24 WHERE status = 'фізична особа';
25
26 INSERT INTO clients_normalized (id, company_name, birth_date, address, phone, status)
27 SELECT
28     id,
29     full_name AS company_name,
30     birth_date,
31     address,
32     phone,
33     status
34 FROM clients
35 WHERE status = 'юридична особа';

```

[illegible]

b) Друга нормальна форма (2NF):

Для відповідності **2NF**, таблиця повинна відповідати 1NF і не містити часткових залежностей — тобто жоден неключовий атрибут не може залежати тільки від частини первинного ключа (якщо ключ складний).

- У даній таблиці первинний ключ — це поле **id**, яке є простим (один стовпець).
- Всі інші поля (ім'я, адреса, дата народження тощо) функціонально залежать від **id**.

Висновок: Таблиця **відповідає** другій нормальній формі.

c) Третя нормальна форма (3NF):

Для відповідності **3NF**, таблиця повинна відповідати 2NF і жоден неключовий атрибут не може залежати від інших неключових атрибутів (тобто не повинно бути транзитивних залежностей).

- Поля, такі як ім'я, прізвище, адреса і телефон, залежать тільки від первинного ключа **id**.
- Немає транзитивних залежностей між неключовими атрибутами.

Висновок: Таблиця **відповідає** третій нормальній формі.

d) Нормальна форма Бойса-Кодда (BCNF):

Таблиця відповідає **BCNF**, якщо вона відповідає 3NF і кожен детермінант є кандидатом на первинний ключ.

- У таблиці немає атрибутів, які детермінують інші, окрім первинного ключа **id**.

Висновок: Таблиця **відповідає** нормальній формі Бойса-Кодда.

e) Четверта нормальна форма (4NF):

Для відповідності **4NF**, таблиця не повинна містити багатозначних залежностей. Це стосується ситуацій, коли один атрибут може мати множинні значення для одного значення первинного ключа.

[illegible]

b)2NF: Усі неключові атрибути залежать від первинного ключа (стовпець id є PK).

c)3NF: Всі атрибути не мають транзитивних залежностей (наприклад, дані про освіту та посаду не залежать один від одного).

d)BCNF: Таблиця відповідає вимогам, оскільки кожна залежність базується на суперключі.

e)4NF: Відсутність багатозначних залежностей — кожен атрибут зберігає одну атомарну інформацію.

4. Навести список запитів, що використовуються найчастіше.

query	executions	total_time_seconds
SELECT `Favorite` . `id` , `Favorite` . `UserId` , `Favorite` . `RecipeId` , `Favorite` . `createdAt` , `Favorite` . `u...	15265	6858.5262
SELECT `Favorite` . `id` , `Favorite` . `UserId` , `Favorite` . `RecipeId` , `Favorite` . `createdAt` , `Favorite` . `u...	12532	6200.6348
CREATE TABLE IF NOT EXISTS `Users` (`id` INTEGER NOT NULL AUTO_INCREMENT , `username` VARCHARACTER (?)...	211	713.6021
CREATE TABLE IF NOT EXISTS `Likes` (`id` INTEGER NOT NULL AUTO_INCREMENT , `userId` INTEGER NOT NULL , `c...	213	572.0164
CREATE TABLE IF NOT EXISTS `Comments` (`id` INTEGER NOT NULL AUTO_INCREMENT , `commentBody` TEXT NOT ...	213	570.0211
CREATE TABLE IF NOT EXISTS `Favorites` (`id` INTEGER NOT NULL AUTO_INCREMENT , `createdAt` DATETIME NOT ...	92	502.6774
CREATE TABLE IF NOT EXISTS `Recipes` (`id` INTEGER NOT NULL AUTO_INCREMENT , `title` VARCHARACTER (?) NO...	147	380.9501
CREATE TABLE IF NOT EXISTS `Favorites` (`id` INTEGER NOT NULL AUTO_INCREMENT , `UserId` INTEGER NOT NULL...	114	361.4893
SHOW INDEX FROM `Users` FROM `kursova`	213	347.9902
SHOW INDEX FROM `Favorites` FROM `kursova`	213	307.4223
SHOW TABLES	110	306.0612
SELECT `id` , `title` , `description` , `instructions` , `username` , `cookingTime` , `ingredients` , `category` , `phot...	655	292.2805
SHOW INDEX FROM `Recipes` FROM `kursova`	213	286.2301
SHOW INDEX FROM `Likes` FROM `kursova`	213	254.9711
SHOW INDEX FROM `Comments` FROM `kursova`	213	245.1516
SELECT `id` , `createdAt` , `updatedAt` , `RecipeId` , `UserId` FROM `Favorites` AS `Favorite` WHERE `Favorite` ...	424	231.5854
SET `time_zone` = ?	741	217.6190
COMMIT	76	209.5847
SHOW SESSION VARIABLES LIKE ?	13	201.8412
CREATE TABLE IF NOT EXISTS `Recipes` (`id` INTEGER NOT NULL AUTO_INCREMENT , `title` VARCHARACTER (?) NO...	58	194.8535
DELETE FROM `Favorites` WHERE `userId` = ? AND `recipeId` = ?	31	193.8855
SELECT `id` , `title` , `description` , `instructions` , `username` , `cookingTime` , `ingredients` , `category` , `phot...	272	122.5788
SHOW INDEX FROM `kursova` . `favorites`	72	117.5299
SHOW CHARACTER SET WHERE CHARSET = ?	56	116.4065
EXPLAIN `cases`	50	115.4470
SET NAMES ? COLLATE ?	105	108.3244
SELECT `id` , `username` , `password` , `createdAt` , `updatedAt` FROM `Users` AS `Users` WHERE `Users` . `us...	199	101.3939
SELECT `st` . * FROM `performance_schema` . `events_statements_current` `st` JOIN `performance_schema` . `th...	87	79.3249

query	executions	total_time_seconds
SELECT 'id', 'UserId', 'RecipeId', 'createdAt', 'updatedAt' FROM 'Favorites' AS 'Favorite' WHERE 'Favorite'...	145	65.4489
DELETE FROM 'Favorites' WHERE 'UserId' = ? AND 'RecipeId' = ?	21	63.3354
SHOW FULL FIELDS FROM 'kursova' . 'comments'	26	62.3186
EXPLAIN 'lawyers'	45	61.8316
SHOW FULL TABLES FROM 'kursova'	22	59.7490
SELECT 'id', 'commentBody', 'username', 'createdAt', 'updatedAt', 'RecipeId' FROM 'Comments' AS 'Comm...	119	58.6351
EXPLAIN 'clients'	45	57.9803
INSERT INTO 'clients' ('full_name', 'birth_date', 'address', 'phone', STATUS) VALUES (...)	36	52.6514
SHOW INDEX FROM 'lab5' . 'cases'	23	52.3863
SELECT 'st' , * FROM 'performance_schema' . 'events_statements_current' `st` JOIN 'performance_schema' . 'th...	57	51.6420
SET 'autocommit' = ?	56	51.4179
SHOW SESSION VARIABLES LIKE ?	40	51.2182
SHOW SESSION STATUS LIKE ?	12	49.6626
SHOW FULL FIELDS FROM 'kursova' . 'favorites'	31	48.7868
SHOW FULL FIELDS FROM 'kursova' . 'recipes'	25	48.1911
SHOW GLOBAL STATUS	12	40.8147
SHOW FULL FIELDS FROM 'kursova' . 'users'	25	40.4253
SHOW PROCEDURE STATUS WHERE 'Db' = ?	23	40.3314
SELECT 'id', 'title', 'description', 'instructions', 'username', 'cookingTime', 'ingredients', 'category', 'phot...	77	40.1881
SELECT * FROM 'lab5' . 'clients' LIMIT ?, ...	8	37.7043
SHOW SCHEMAS	22	37.5373
SELECT CURRENT_USER ()	56	34.3149
SELECT * FROM 'kursova' . 'favorites' LIMIT ?, ...	73	34.2957
SHOW FULL FIELDS FROM 'kursova' . 'likes'	24	33.8453
SELECT 'id', 'title', 'description', 'instructions', 'username', 'cookingTime', 'ingredients', 'category', 'phot...	80	33.6241
SELECT * FROM 'lab5' . 'cases' LIMIT ?, ...	23	32.6735
SHOW FULL FIELDS FROM 'lab5' . 'cases'	16	30.5707
SELECT 'st' . * FROM 'performance_schema' . 'events_stages_history_long' `st` WHERE 'st' . 'nesting_event_id' ...	87	30.4403
query	executions	total_time_seconds
SELECT 'VERSION' () AS 'version'	213	29.4702
SELECT 'st' . * FROM 'performance_schema' . 'events_waits_history_long' `st` WHERE 'st' . 'nesting_event_id' = ?	87	29.3162
SELECT 'id', 'title', 'description', 'instructions', 'username', 'cookingTime', 'ingredients', 'category', 'phot...	64	28.2864
SHOW FULL FIELDS FROM 'lab5' . 'clients'	15	25.3842
SHOW FUNCTION STATUS WHERE 'Db' = ?	23	23.5376
SHOW PROCEDURE STATUS WHERE 'Db' = ?	15	22.7518
SELECT * FROM 'cases' WHERE 'punishment_min' > ?	18	19.7171
SHOW FULL TABLES FROM 'lab5'	8	19.4925
SHOW SESSION VARIABLES LIKE ?	10	19.1557
SHOW INDEX FROM 'kursova' . 'recipes'	9	18.8024
SHOW INDEXES FROM 'kursova' . 'favorites'	11	18.3463
UPDATE 'cases' SET 'case_status' = ? WHERE 'case_id' = ?	36	17.5833
SHOW FULL FIELDS FROM 'lab5' . 'case_types'	10	17.2908
SHOW INDEXES FROM 'lab5' . 'cases'	8	16.8831
SHOW FUNCTION STATUS WHERE 'Db' = ?	15	15.9567
SHOW INDEX FROM 'lab5' . 'clients'	8	15.5832
SET 'autocommit' = ?	121	14.7717
SHOW FULL FIELDS FROM 'lab5' . 'lawyers'	10	14.5494
SELECT * FROM 'cases'	24	14.5452
SET NAMES 'utf8'	54	14.0695
SELECT 'st' . * FROM 'performance_schema' . 'events_stages_history_long' `st` WHERE 'st' . 'nesting_event_id' ...	57	14.0351
SHOW INDEX FROM 'lab5' . 'case_types'	10	13.2583
SELECT 'st' . * FROM 'performance_schema' . 'events_waits_history_long' `st` WHERE 'st' . 'nesting_event_id' = ?	57	11.8728
SET CHARACTER SET 'utf8'	54	11.2791
SELECT ? AS ?, CATALOG_NAME AS ?, SCHEMA_NAME AS ?, SCHEMA_NAME AS ? FROM 'information_schema' . 'sch...	7	10.8768
SELECT * FROM 'cases' ORDER BY 'fee' DESC	32	8.8466
SELECT 'id', 'createdAt', 'updatedAt', 'RecipeId', 'UserId' FROM 'Favorites' AS 'Favorite' WHERE 'Favorite'...	14	7.7535
SELECT 'case_type' AS 'Тип справи', 'article_code' AS 'Код статті', 'punishment_max' AS 'Максимальний тер...	23	6.2975
SELECT 'CONNECTION_ID' ()	12	5.8792
SELECT @@'lower_case_table_names'	7	5.8519
SELECT 'case_type' AS 'Тип справи', 'article_code' AS 'Код статті', 'punishment_max' AS 'Максимальний тер...	16	5.5211
SET NAMES ?	56	5.3716
SELECT * FROM 'cases' WHERE 'punishment_min' > ? AND 'punishment_max' < ?	18	4.7071
SELECT * FROM 'kursova' . 'recipes' LIMIT ?, ...	9	4.6085
SELECT * FROM 'lab5' . 'case_types' LIMIT ?, ...	10	3.5961
SHOW CREATE SCHEMA 'mysql'	7	2.6332
SHOW CREATE SCHEMA 'law'	7	2.1755
SHOW CREATE SCHEMA 'sys'	7	1.8699
SET 'autocommit' = ?	16	1.6490
SHOW CREATE SCHEMA 'b'task_8'	7	1.6440
SHOW CREATE SCHEMA 'law_firm_db'	7	1.3884
SHOW CREATE SCHEMA 'performance_schema'	7	1.3287
USE 'kursova'	10	1.2340
SHOW CREATE SCHEMA 'information_schema'	7	0.9277

5. Оцінити швидкість і зручність роботи з нормалізованою базою даних, у випадку виявлення недоліків, здійснити денормалізацію бази даних.

Нормалізація бази даних покращує цілісність і структурованість даних, зменшуючи дублювання та підвищуючи швидкість запитів на вибірку. Однак, із збільшенням кількості таблиць, які потрібно об'єднувати, можуть виникнути затримки у виконанні складних запитів. У таких випадках може бути доцільно розглянути денормалізацію, яка спростить запити, але може призвести до дублювання даних і ускладнення їхнього оновлення.

s

Висновок: Нормалізація бази даних є важливим етапом проектування, що забезпечує її ефективність, зменшуючи ризики аномалій при оновленні даних. Однак, важливо враховувати, що надмірна нормалізація може призводити до зниження швидкодії при виконанні запитів. Тому досягнення оптимального балансу між нормалізацією і денормалізацією є критично важливим для забезпечення ефективної роботи бази даних у реальних умовах.