

## Лабораторна робота №13

**Тема.** Система управління базами даних MySQL. Створення таблиць для розробки Web- проекту.

### Завдання до лабораторної роботи №13

Для розробленого Web- проекту в *лабораторній роботі №1-12*, на сервері баз даних **MySQL** створити таблиці та організувати зв'язки між таблицями. Дані представити в трьох і більше таблицях. Таблиці повинні необхідно привести до третьої нормальної форми. У кожному з таблиць ввести по 10 записів. (Для створення та наповнення бази даних та таблиць використати засіб **phpMyAdmin** комплексу програм(сервер **Apache**, база даних **mysql**, **php** та **Perl** інші складові) **xampplite**.

Приклад.

#### Створення бази даних.

1. У браузері набираємо адресу : **localhost**;
2. Переходимо в **phpMyAdmin**;
3. Створюємо базу даних **book**;
4. Створюємо таблицю **books**, кількість **полів – 4**;
5. Вводимо назви та параметри полів num (INT, cp\_1251\_ukrainian, ключове), Author(VARCHAR, 30, cp\_1251\_ukrainian) ,title (VARCHAR, 40, cp\_1251\_ukrainian) та isdn (VARCHAR, 30, cp\_1251\_ukrainian);
6. В таблицю додаємо декілька записів(закладка **Вставити**).
7. Переходимо в каталог **c:\WebServer\xampp\htdocs** ( кореневий каталог Web- сервера, відповідає шляху **localhost**);
8. Створюємо каталог mybook;

Таблиця 1а

	Вид діяльності	Обов'язкові поля, які повинні бути у таблицях бази даних <b>MySQL</b> .
1	Продаж комп'ютерів	Назва комп'ютера, тип процесора; тактова частота, обсяг оперативної пам'яті; обсяг вінчестера, відео карта, звукова карта.
2	Продаж комп'ютерів	Назва комп'ютера, виробник процесора, тактова частота, відео- карта, обсяг оперативної пам'яті, звукова карта, обсяг вінчестера.
3	Продаж комп'ютерів	Назва комп'ютера, тип процесора, тактова частота, відео карта, тип пристрою CDROM/CDRW звукова карта обсяг вінчестера.
4	Комп'ютерні аксесуари	Процесор ( тип, виробник, тактова частота) Монітор ( тип, виробник, частота) Вінчестер (тип, виробник, об'єм)
5	Комп'ютерні аксесуари	Графічний адаптер ( тип, виробник, пам'ять). Пристрій CDROM/CDRW ( тип, виробник, частота). Вінчестер (тип, виробник, об'єм)
6	Продаж книг	Автор, видавництво, ціна.
7	Виробництво молочних виробів	Вид, категорія, термін придатності, ціна
8	Продаж сигарет	Марка, виробництво, нікотин, ціна.
9	Продаж косметики	Вид, марка, виробник, ціна.

10	Продаж алкогольних напоїв	Вид, марка, виробник, ціна
11	Продаж слабо - алкогольних напоїв	Вид, марка, виробник, ціна
12	Продаж меблів	Вид, марка, виробник, ціна
13	Продаж автомашин	Вид, марка, виробник, ціна
14	Виробництво слабо - алкогольних напоїв	Вид, марка, виробник, ціна
15	Виробництво меблів	Вид, марка, матеріал, ціна

### Теоретичні відомості.

**Нормальна форма** - вимога, пропонована до структури таблиць у теорії реляційних баз даних для усунення з бази надлишкових функціональних залежностей між атрибутами (полями таблиць). Процес перетворення бази даних до виду, що відповідає нормальним формам, називається нормалізацією.

#### *Виникнення та призначення нормальних форм*

Поняття нормальної форми було введено Едгаром Коддом при створенні реляційної моделі БД. Основне призначення нормальних форм - приведення структури бази даних до виду, що забезпечує мінімальну надмірність. Усунення надмірності виробляється за рахунок декомпозиції відносин (таблиць) таким чином, щоб звести до мінімуму функціональні залежності між їхніми атрибутами (полями). Поняття функціональної залежності тут визначається в такий спосіб:

У парі атрибутів одного відношення,  $X$  й  $Y$ , атрибут  $Y$  функціонально залежить від атрибута  $X$ , якщо в даному відношенні одному значенню  $X$  відповідає в точності одне значення  $Y$ .

Повна відсутність функціональних залежностей у відносинах неможливо (та й не потрібно), але частина таких залежностей надлишкова, оскільки їх можна обчислити з інших залежностей, наявних у БД.

Кожна нормальна форма являє собою певну умову, якій повинна відповідати таблиця бази даних. Якщо таблиця не відповідає нормальній формі, вона може бути наведена до неї (нормалізована) за рахунок декомпозиції, тобто розбивки на кілька таблиць, зв'язаних між собою. Теоретично, у результаті нормалізації обсяг БД повинен зменшитися. Принциповим тут є те, що нормалізація - оборотний процес, із групи таблиць, що вийшли при декомпозиції, завжди можна одержати в точності вихідну таблицю. Таким чином, нормалізація не скорочує обсяг інформації, збереженої в БД, а лише усуває інформацію, що може бути обчислена.

### Типи нормальних форм

Нормалізація може застосовуватися до таблиці, що відповідає наступним вимогам:

- Таблиця містить нуль або більше записів.
- Всі записи таблиці мають ту ж саму множину полів, причому однойменні поля ставляться до однакових типів даних.
- Таблиця не може містити двох повністю ідентичних записів.

Виділяють шість нормальних форм:

#### *Перша нормальна форма (1NF)*

Таблиця перебуває в першій нормальній формі, якщо кожен її атрибут автономний і всі рядки різні. Під виразом "атрибут автономний" розуміється, що атрибут може містити тільки одне значення. Таким чином, не відповідають 1NF таблиці, поля в яких можуть зберігатися списки значень. Для приведення таблиці до 1NF звичайно потрібно розбити таблицю на кілька окремих таблиць.

Приклад приведення таблиці до першої нормальної форми.

Вихідна, ненормалізована, таблиця:

Співробітник	Номер телефону
Іванов І. І.	283-56-82 390-57-34

Петров П. Ю.	708-62-34
--------------	-----------

Таблиця, наведена до 1NF:

Співробітник	Номер телефону
Іванов І. І.	283-56-82
Іванов І. І.	390-57-34
Петров П. Ю.	708-62-34

#### *Друга нормальна форма (2NF)*

Таблиця перебуває в другій нормальній формі, якщо вона перебуває в першій нормальній формі, і при цьому будь-який її атрибут, що не входить до складу первинного ключа, функціонально повно залежить від первинного ключа. Функціонально повна залежність означає, що атрибут функціонально залежить від усього первинного ключа, але при цьому не перебуває у функціональній залежності від якої-небудь його частини.

#### *Примітка*

Якщо первинний ключ не складової, то таблиця в 1NF автоматично перебуває й в 2NF.

#### *Третя нормальна форма (3NF)*

Таблиця перебуває в третій нормальній формі, якщо вона перебуває в другій нормальній формі, і при цьому будь-який її неключовий атрибут функціонально залежить тільки від первинного ключа.

#### *Нормальна форма Бойса-Кодда (BCNF)*

Таблиця перебуває в BCNF, якщо вона перебуває в третій нормальній формі, і при цьому відсутні функціональні залежності атрибутів первинного ключа від не-ключових атрибутів.

Дана нормальна форма — це модифікація третьої нормальної форми. Таблиця може перебувати в 3NF, але не в BCNF, тільки в одному випадку: якщо вона має, крім первинного ключа, ще принаймні один складений можливий ключ, і принаймні один з атрибутів таблиці входить у первинний, і в можливий ключі. Таке буває досить рідко, в іншому 3NF й BCNF еквівалентні.

#### *Четверта нормальна форма (4NF)*

Таблиця перебуває в 4NF, якщо вона перебуває в BCNF і не містить нетривіальних багатозначних залежностей. Багатозначна залежність не є функціональною, вона існує в тому випадку, коли з факту, що в таблиці вміщується деякий рядок X, випливає, що в таблиці обов'язково присутній деякий рядок Y.

#### *П'ята нормальна форма (5NF)*

Таблиця перебуває в 5NF, якщо вона перебуває в 4NF і будь-яка багатозначна залежність з'єднання в ній є тривіальною.

Денормалізація (denormalization) — спеціальне приведення структури бази даних у стан, що не відповідає критеріям нормалізації, звичайно проведене з метою прискорення операцій читання з бази за рахунок додавання надлишкових даних.

Теорія реляційних баз даних вимагає, щоб будь-яка база даних була нормалізована, тобто відповідала вимогам нормальних форм (як мінімум, перших трьох). Відповідність вимогам нормалізації мінімізує надмірність бази даних і забезпечує найбільшу теоретично доступну гнучкість.

Однак нормалізація вступає в протиріччя з вимогами ефективності роботи з базою даних. У результаті нормалізації цілісні таблиці розбиваються на зв'язані посиланнями набори таблиць. Запит до однієї ненормалізованої таблиці як за часом, так і по пам'яті ефективніше запиту, що вибирає ті ж дані із групи зв'язаних таблиць.

Внаслідок цього в ситуаціях, коли ефективність виявляється більше важлива, чим гнучкість БД, може проводитися денормалізація - перетворення БД, при якому зв'язані

посиланнями таблиці поєднуються для більш ефективного доступу. При денормалізації можлива поява дублювання інформації в БД