**3. Нормалізація відношень при проектуванні БД**

[Наверх сторінки](https://rdb.dp.ua/uk/chapter_03#top)

Лекція

[**3.1. Процедура нормалізації**](https://rdb.dp.ua/uk/chapter_03#3.1)

[**3.2. Перша нормальна форма**](https://rdb.dp.ua/uk/chapter_03#3.2)

[**3.3. Друга нормальна форма**](https://rdb.dp.ua/uk/chapter_03#3.3)

[**3.4. Третя нормальна форма**](https://rdb.dp.ua/uk/chapter_03#3.4)

[**3.5. Четверта та п'ята нормальні форми**](https://rdb.dp.ua/uk/chapter_03#3.5)

[**3.6. Нормалізація – за й проти**](https://rdb.dp.ua/uk/chapter_03#3.6)

[**3.7. КОНТРОЛЬНІ ПИТАННЯ**](https://rdb.dp.ua/uk/chapter_03#3.7)

**Мета**

*Мета лекції – ознайомлення з процедурою нормалізації, її різними рівнями, теоретичними і практичними аспектами, критеріями визначення рівня нормалізації на практиці відповідно до вимог кінцевих користувачів.*

**3.1. Процедура нормалізації**

З одного боку, процес проектування структури БД є творчим, неоднозначним, з іншого боку, його вузлові моменти можуть бути формалізовані [4 – 7, 10, 12]. У процесі розробки логічна модель даних постійно тестується й перевіряється на відповідність вимогам користувачів. Коректність логічної моделі даних забезпечується ***процедурою нормалізації***.

Процедура нормалізації БД полягає в усуванні ***надмірності даних*** та виявленні ***функціональних залежностей***. Усунення ***надмірності даних*** гарантує компактність наборів даних за рахунок уникнення їх зайвого дублювання та унеможливлює виникнення аномалій вставки, вилучення й оновлення кортежів після фізичної реалізації БД. ***Функціональна залежність****пов'язує атрибути в одному відношенні з єдиним значенням в іншому.* Функціональну залежність для відношень ***А*** та ***B*** прийнято позначати як ***A→B***. Це поняття підводить “на один крок” до спорідненої концепції об'єднання відношень зв'язками типу один до одного (***1:1***) або один до багатьох (***1:М***).

Правила нормалізації або правила Кодда, як їх тепер називають, дуже прості й нечисленні, але досить суворі. У разі застосування до відношень кожне правило описує наступний рівень відповідності вимогам теорії реляційних БД і різні ступені нормалізації.

Існують такі рівні нормалізації: перша нормальна форма (***1НФ***), ***2НФ***, ***3НФ***, нормальна форма ***Бойса-Кодда*** (***БКНФ***), ***4НФ***, ***5НФ***. Але дотепер жодна з реляційних СКБД не надає належної підтримки усім п'яти нормальним формам. Це відбувається через жорсткі вимоги до продуктивності. Суть справи полягає в тому, що в повністю нормалізованій БД для виконання запиту треба з'єднати настільки багато таблиць, що продуктивність такої системи не зможе задовольнити користувачів. Тому на практиці використовують лише перші три рівня нормалізації – ***1НФ***, ***2НФ***, ***ЗНФ***.

**3.2. Перша нормальна форма**

*Відношення буде зведено до****першої нормальної форми****(****1НФ****) тоді й тільки тоді, коли всі його атрибути містять тільки неподільні (атомарні) значення й у ньому відсутні групи атрибутів з однаковими за змістом значеннями, які повторюються у межах одного кортежу.*

***Неподільність*** значення атрибута говорить про те, що його не можна розбити на більш дрібні частини. Наприклад, якщо у атрибуті «Прізвище, ім'я, по батькові» міститься прізвище, ім'я та по батькові читача, вимога неподільності не виконується (рис. 3.1, а). Тут необхідно виділити в окремі атрибути ім'я та по батькові. У результаті вийде три атрибути відношення «Читачі»: «Прізвище», «Ім'я» і «По батькові» (рис. 3.1, б).

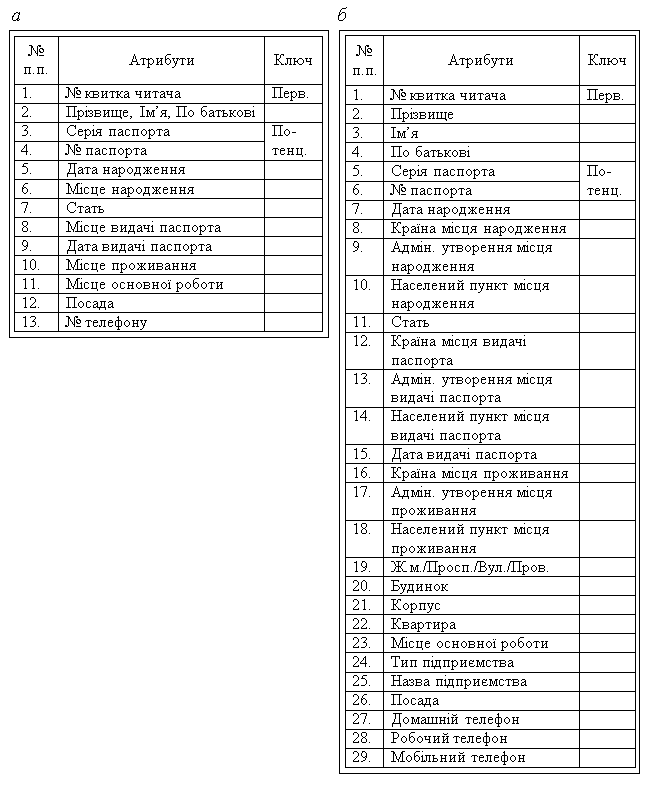


Рис. 3.1. Можливі специфікації ненормалізованого відношення «ЧИТАЧІ»

На більш дрібні частини можна розбити атрибути: «Місце народження» («Країна», «Адміністративне утворення», «Населений пункт»); «Місце видачі паспорта» («Країна», «Адміністративне утворення», «Населений пункт»); «Місце основної роботи» («Тип підприємства», «Назва підприємства»), «Місце проживання» («Країна», «Адміністративне утворення», «Населений пункт», «Житловий масив / Проспект / Вулиця / Провулок», «Будинок», «Корпус», «Квартира»), рис. 3.1, б.

Для контакту читач може визначити один, декілька або жодного номеру телефону. Таким чином у загальному випадку інформація у атрибуті «Номер телефону» може бути розділена на декілька частин, кожна з яких є окремим телефонним номером (рис. 3.1, а). На перший погляд, цю проблему можна вирішити так само, як і для прізвища, імені та по батькові, виділивши для найпоширеніших типів телефонів окремі атрибути (рис. 3.1, б). Однак у цьому випадку ми зіткнемося з групою атрибутів, що мають однакові за змістом значення в межах одного кортежу, наприклад: «Домашній телефон», «Робочій телефон», «Мобільний телефон».

Щоб відношення «ЧИТАЧІ» (рис. 3.1, б) відповідало 1НФ, треба вилучити з нього групу атрибутів з номерами телефонів, які повторюються в межах одного кортежу, в інше відношення разом з копією ключового атрибута «№ квитка читача» (рис. 3.2). Причому, для подання номера й типу телефона виділимо окремі атрибути. Це дозволяє: по-перше, урахувати не тільки три зазначені типи телефона, але й додати нові і по-друге, зазначити для кожного читача тільки ті типи телефонів, які в нього є, і по-третє, можна указувати для будь-якого читача кілька однотипних телефонів або не указувати жодного номера телефону.

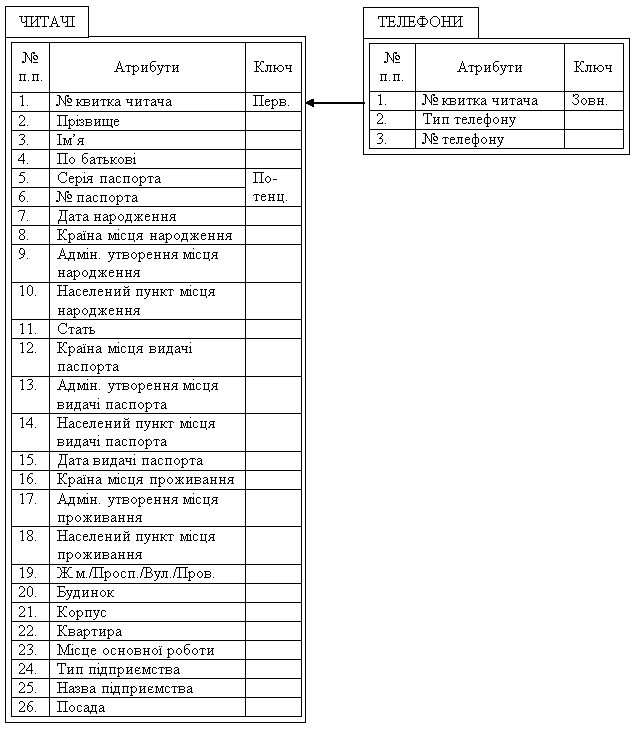


Рис. 3.2. Приклад відношення «ЧИТАЧІ», яке зведено до форми 1НФ

**3.3. Друга нормальна форма**

*Відношення буде зведено до****другої нормальної форми****(****2НФ****) тоді й тільки тоді, коли воно є в****першій нормальній формі****, і кожний неключовий атрибут повністю визначається****первинним ключем****, тобто щоб первинний ключ однозначно визначав кортеж і не був надлишковим (збігався із суперключем). Ті атрибути, які залежать тільки від частини суперключа, мають бути виділені в окремі таблиці.*

Відношення «ЧИТАЧІ» (рис. 3.2) не зведено до форми 2НФ. У ньому кожен кортеж однозначно ідентифікується такими атрибутами: «№ квитка читача», «Серія паспорта» і «№ паспорта». Сукупність цих атрибутів є ***суперключем*** цього відношення. Він складається з двох ***потенційних ключів***, кожен з яких окремо може ідентифікувати кортежі відношення. Із двох потенційних ключів за первинний вибирається той, довжина якого мінімальна. Наявність обох потенційних ключів обумовлена вимогами користувачів.

Звести відношення «ЧИТАЧІ» до 2НФ можна, якщо винести в окреме відношення атрибути 2 – 22, які стосуються паспортних даних, і копію первинного ключа «№ квитка читача» (рис. 3.3). Однак у результаті ми одержимо відношення «ПАСПОРТНІ ДАНІ», яке має такий самий суперключ, як і відношення «ЧИТАЧІ» до його зведення до 2НФ. У нашому випадку подальша нормалізація відношення «ПАСПОРТНІ ДАНІ» неможлива.

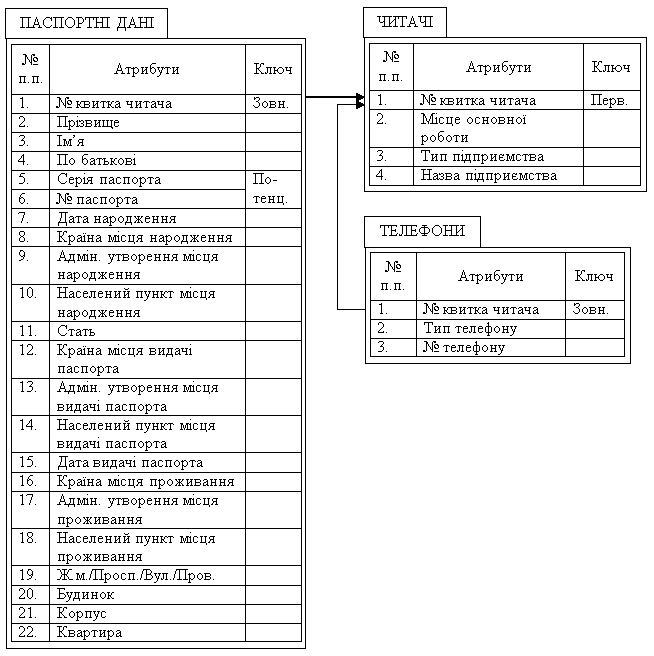


Рис. 3.3. Приклад відношень «ТЕЛЕФОНИ» і «ЧИТАЧІ», що зведені до 2НФ

У відношенні «ТЕЛЕФОНИ» (рис. 3.3) на перший погляд здається, якщо в опис логічної моделі додати номер телефону читача, включаючи код країни, оператора або населеного пункту, то потенційним буде ключ, у складі якого всього один атрибут – «№ телефона». Однак у двох різних читачів може бути однаковий однаковим номер його домашнього або робочого телефона. Отже, однозначно, ідентифікувати кортежі можливо за складеним суперключем, до якого входять атрибути «№ квитка читача» та «№ телефону». Ми бачимо, що суперключ не є надлишковим. Його зміст збігається з первинним ключем. Виходить, що відношення «ТЕЛЕФОНИ» презентовано у другій нормальній формі.

**3.4. Третя нормальна форма**

Загалом 1НФ і 2НФ розглядаються як проміжні етапи в процесі нормалізації БД. Більша частина СКБД орієнтована на досягнення наступного ступеня нормалізації – третьої нормальної форми (3НФ). Це пов’язано з тим, що зведення відношень до 3НФ цілком відповідає майже усім практичним задачам. При розробці винятково великих систем на надшвидкодіючих комп'ютерах, коли необхідно забезпечити максимальне зменшення обсягів даних, бажано виконати подальшу нормалізацію відношень.

*Відношення буде зведено до****третьої нормальної форми****(****3НФ****) тоді й тільки тоді, коли воно є у****другій нормальній формі****і у ньому немає транзитивних залежностей між неключовими атрибутами, тобто значення будь-якого атрибута відношення, що не входить до первинного ключа, не залежить від значення іншого атрибута, що не входить до первинного ключа.*

Це визначення – усього лише оригінальний спосіб відобразити необхідність подання системи пов'язаних відношень у такому вигляді, щоб значення атрибутів кожного відношення безпосередньо визначалися або ***суперключем***, або ***потенційним ключем*** цього відношення.

Існує метод розрахунків мінімального числа відношень, яке необхідно для зведення всіх відношень БД до 3НФ. У тому випадку, якщо Ви склали список усіх функціональних залежностей для Ваших даних, то можна застосувати алгоритм Бернштейна, який описаний у будь-якому підручнику реляційної алгебри.

Розглянемо простий приклад з обліку товару на складі. Здається логічним, щоб у відношення з оперативною інформацією про товар на складі входили у число інших такі атрибути: «Кількість товару», «Ціна за одиницю товару» та «Загальна вартість товару». Вони не є ключовими. Для одержання значення атрибута «Загальна вартості товару» необхідно перемножити між собою значення, що перебувають у атрибутах: «Кількість товару» та «Ціна за одиницю товару», тобто значення атрибута «Загальної вартості товару» залежить від величин двох атрибутів, що не входять до складу первинного ключа. Це суперечить визначенню 3НФ. Щоб дане відношення відповідало третій нормальній формі з нього треба вилучити атрибут «Загальна вартість товару».

Відзначимо, що відношення «ТЕЛЕФОНИ» і «ЧИТАЧІ» (рис. 3.3) відповідають третій нормальній формі. Це випливає з того, що вони зведені до другої нормальної форми й у них немає транзитивних залежностей.

Нижче наведений варіант визначення 3НФ, яка називається ***нормальною формою Бойса-Кодда*** (Воусе – Codd) – ***БКНФ***, де встановлюються більш суворі вимоги.

*Відношення X буде зведено до****нормальної форми Бойса-Кодда****тоді, коли в кожній нетривіальній функціональній залежності В→А В є суперключем.*

**3.5. Четверта та п'ята нормальні форми**

Перш ніж закінчити розгляд правил Кодда, Вам буде запропонований короткий огляд двох останніх форм відношень реляційних БД. Вони призначені для усунення ще двох аномалій: багатозначна залежність та об'єднувальна залежність.

У відношенні *X* існує багатозначна залежність *А→В* тоді, коли в ньому можна виявити ситуації, де пара кортежів містить значення *А*, які дублюються, і одночасно існують інші пари кортежів, що отримані шляхом перестановки значень *В*, які присутні у першій парі.

Насамперед, для існування багатозначної залежності потрібна наявність пар кортежів. *А* й *В* можуть бути як окремими атрибутами, так і об'єднанням деякого набору атрибутів. Тривіальна багатозначна залежність для *А→В* існує тоді, і тільки тоді, коли *В* є підмножиною *А* або *А* поєднує *B = XS* (більше відношення містить вихідне відношення).

Існування багатозначної залежності породжує аномалію оновлення. **4НФ** усуває нетривіальну багатозначну залежність у відношенні за допомогою створення менших відношень. Процес нормалізації полягає в створенні як можна більшого числа дрібних відношень з метою зменшення надмірності даних.

*Відношення X буде зведено до****четвертої нормольної форми****(****4НФ****) тоді й тільки тоді, коли воно є у****БКНФ****і для будь-якої багатозначної залежності А→В у цьому відношенні можна сказати, що вона є або тривіальною, або А є суперключем таблиці X.*

***П’ята нормальна форма****(****5НФ****) досягається в тому випадку, коли відношення не може далі розбиватися на більш малі відношення за допомогою операції проектування.*

Під операцією проектування розуміється декомпозиція даних (без їх утрати), при якій відношення розбивається на частини (кожна з них є окремим відношенням) таким чином, щоб залишалася можливість об'єднати ці частини у вихідне відношення.

**3.6. Нормалізація – за й проти**

Метою нормалізації відношень БД є усунення надлишкової інформації. Як видно з наведених вище прикладів, нормалізовані відношення БД містять тільки один елемент надлишкових даних – це атрибути зв'язку, які присутні одночасно в батьківському та дочірньому відношеннях. Оскільки надлишкові дані у відношеннях не зберігаються, то заощаджується місце на носіях інформації. Однак у нормалізованій БД є й недоліки, насамперед, практичного характеру.

Чим більше суб'єктів (сутностей) охоплюється предметною областю, тим з більшого числа відношень буде складатися нормалізована БД. Бази даних у складі великих систем, які задіяні у життєдіяльності великих організацій та підприємств, можуть містити сотні зв'язаних між собою відношень. Оскільки поріг людського сприйняття не дозволяє одночасно охопити велику кількість об'єктів з урахуванням їх взаємозв'язків, то можна стверджувати, що зі збільшенням числа нормалізованих відношень цілісне сприйняття БД як системи взаємозалежних даних зменшується. Тому при розробці й експлуатації великих систем існують ситуації, коли кожний співробітник уявляє собі процеси, що відбуваються тільки в частині системи. Відомі випадки еволюційного створення таких систем, коли принципи їх функціонування згодом виходили за межі розуміння.

Іншим недоліком нормалізованої БД є необхідність зчитування з відношень зв'язані дані при виконанні складних запитів, які надають інформацію про взаємодію сутностей технологічного процесу між собою. При великих обсягах це призводить до збільшення часу доступу до даних.

Третя нормальна форма та нормальна форма Бойса-Кодда є теоретичними конструкціями, у той час як більшість розроблювачів БД працюють у реальному світі. Тому доречно зробити кілька зауважень про недоліки, що властиві відношенням, які зведені до 3НФ. Існують варіанти, коли має сенс поділити відношення на декілька дрібних, якщо частина наведених у ньому даних непостійна й часто оновлюється (оперативна інформація), а інші дані пасивні та змінюються не так часто (довідкова інформація). Також є сенс у разі необхідності об'єднати відношення, щоб забезпечити високу швидкість реакції на запит. Можна навіть піти на дублювання даних у відношеннях, якщо це дозволить зменшити витрати на обробку запитів, хоча формально не варто було б цього робити.

Необхідно також відзначити, що відношення між собою доцільно з’єднувати по атрибутах, які зовсім звільнені від семантичної залежності, що породжується особливостями реалізації дійсних процесів у БД. Для цього використовують спеціально виділені інкрементні атрибути однозначної ідентифікації кортежів усередині відношень. Такий підхід дозволяє позбавитися необхідності процесу перевизначення зв'язків між відношеннями, що виникає в реальному житті при зміні правил обліку різних суб'єктів діяльності організації.

Наведені вище міркування не слід сприймати як заклик зовсім не нормалізувати дані. Вони лише мають довести, що при роботі з даними великого обсягу доводиться шукати компроміс між вимогами нормалізації (тобто "логічності" даних і економії місця на носіях інформації) та необхідністю збільшення швидкодії. При цьому треба звернути увагу на вимоги користувачів, щоб уникнути зайвої деталізації суб'єктів реальних процесів, які відбуваються на підприємстві.

Якщо користувачу немає сенсу деталізувати атрибути «Місце народження», «Місце видачі паспорта», «Місце основної роботи», «Місце проживання», то пошук компромісу між вимогами нормалізації й швидкодією для відношення «ЧИТАЧІ» (рис. 3.1, а) приведе до того, що, по суті, це відношення може стати не нормалізованим (рис. 3.4). Зверніть увагу, що паспортні дані читачів також не винесені в окреме відношення. Це дозволяє скоротити час виконання запиту, який видає всі наявні дані про читачів, оскільки немає сенсу з’єднувати відношення «ПАСПОРТНІ ДАНІ» та «ЧИТАЧІ».

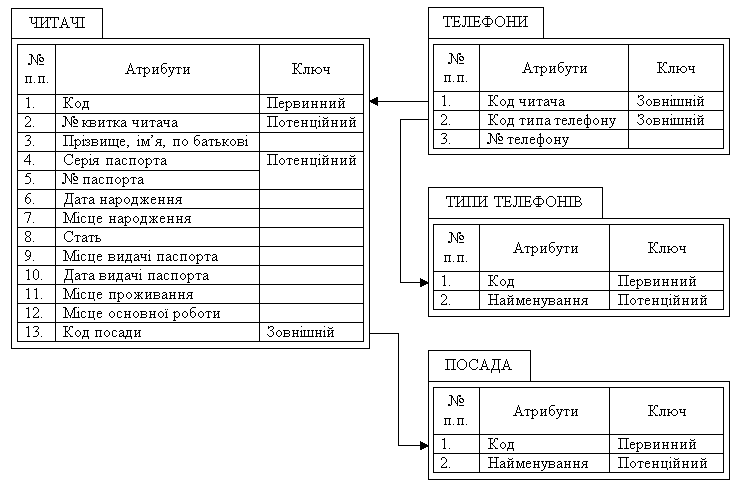


Рис. 3.4. Результат нормалізації відношення «ЧИТАЧІ»

На відміну від паспортних даних відомості про телефонні номери читачів винесені в окреме відношення (рис. 3.4). Це пов'язане з тим, що у читача може бути як кілька контактних телефонів, так і не бути їх взагалі. Інакше кажучи, зв'язок 1:М між суб'єктами обліку та діяльності організації у більшості випадків слід реалізувати за допомогою двох відношень. Зв'язок 1:1 найчастіше реалізується в одному відношенні.

На користь реалізації зв'язку 1:1 між суб'єктами обліку більш ніж в одному відношенні говорить наявність у вимогах користувачів необхідності одержання узагальнюючої інформації відносно якого-небудь атрибута. Саме це виправдовує винесення даних про тип телефона та посади читачів в окремі таблиці (рис. 3.4). У цьому разі посада та типа телефона наводяться один раз.

Якщо користувачі будуть вносити назву посади або тип телефона у відповідні атрибути відношень «ЧИТАЧІ» і «ТЕЛЕФОНИ» (рис. 3.3), то розроблювачі не зможуть гарантувати, що із БД запити будуть вибирати коректну інформацію. Це пов'язане з тим, що назву однієї й тієї ж посади або типу телефона оператор може внести із синтаксичною помилкою. У цьому разі стандартні алгоритми обробки запитів користувачів до БД будуть інтерпретувати однакові по суті значення відповідного атрибута як різні.

Запит користувача на одержання узагальнюючої інформації може бути сформульований так: «вивести на екран робочі номери телефонів читачів із вказівкою їх прізвищ, імен, по батькові, місця роботи та посади», або так: «вивести на екран контактні номери телефонів всіх асистентів із вказівкою їх прізвищ, імен, по батькові».

Уведення інкрементних атрибутів «Код» дозволить цілком відмовитися від необхідності зміни властивостей ключових атрибутів у зв'язку зі змінами правил обліку читачів у бібліотеці (рис. 3.4). Вони дозволили трохи компенсувати збільшення необхідного обсягу пам'яті для реалізації БД за рахунок зменшення місця для атрибута відношення, що пов'язує «ЧИТАЧІ» та «ТЕЛЕФОНИ» з відношеннями «ПОСАДА» та «ТИПИ ТЕЛЕФОНІВ» відповідно. Ці відношення можна було б зв'язати з атрибутами «Найменування посади» та «Найменування типу телефона» замість атрибутів «Код посади» та «Код типу телефона». Економія буде залежить тільки в кількості знаків цих назв, яка закладена згідно з вимогами користувачів. Результат нормалізації відношення «ЧИТАЧІ» (рис. 3.4) може бути іншим. Він залежить від вимог користувачів, досвіду та погляду розробника на процедуру нормалізації відношень, необхідних для вирішення поставлених задач. У нашому прикладі ми одержали з одного відношення «ЧИТАЧІ» (рис. 3.1, а) чотири відношення: «ЧИТАЧІ», «ТЕЛЕФОНИ», «ТИПИ ТЕЛЕФОНІВ» і «ПОСАДА» (рис. 3.4). Відношення «ЧИТАЧІ» ненормалізоване. Відношення «ТЕЛЕФОНИ» зведено до 3НФ. Відношення «ТИПИ ТЕЛЕФОНІВ» і «ПОСАДА» зведені до 1НФ.

Кінцевим результатом нормалізації відношень БД «БІБЛІОТЕКА» є діаграма зв'язків між ними (додаток А). Для компактності на ній наводяться лише ключові атрибути відношень. Зв'язок між логічною та фізичною моделями зручно подати після етапу фізичного проектування БД.

**3.7. КОНТРОЛЬНІ ПИТАННЯ**

1. Що забезпечує та гарантує процедура нормалізації?
2. Які нормальні форми відношень БД Вам відомі?
3. Коли відношення зведено до першої нормальної формі?
4. Коли відношення зведено до другої нормальної формі?
5. Коли відношення зведено до третьої нормальної формі?
6. Які нормальні форми відношень використовують на практиці?
7. Які нормальні форми відношень мало використовують на практиці?
8. За яких умов у БД можна залишити відношення, що не є в 1НФ?
9. Як позбутися семантичної залежності в зв’язках між відношеннями?
10. Які існують способи реалізації зв’язків 1:1 та 1:М у БД?

**Висновок**

*Необхідність проведення процедури нормалізації на етапі логічного моделювання реляційної БД є беззаперечною. При реалізації логічної моделі треба розділити теоретичний та практичний аспекти нормалізації відношень реляційної БД. На практиці процедура нормалізації є компромісом між усуненням надмірності даних, функціональних залежностей та швидкістю вибірки і обробки даних, яка задовольняє кінцевих користувачів.*