

Лекція 8. Формалізація зв'язків між відношеннями

У попередньому розділі розглянуто проектування СБД на рівні окремих сутностей (окремих відношень). Проте СБД, найчастіше, складається з множини сутностей, між якими існують певні зв'язки. Тому природним є питання, яким чином формалізувати (визначити) ці зв'язки в СБД. Тут також існує ряд правил, і для цього необхідно згадати, які існують типи зв'язків. Але до того визначимо ще ряд понять.

1. Іноді виникають ситуації, коли в **якому-небудь кортежі відношення не існує значення** того чи іншого атрибута. Не „значення не відоме“ користувачу, а саме „значення не існує“ (на цей час чи взагалі) або „не може існувати“.. Наприклад: сутність *Принтер*:

Тип	Фірма	Тип чорнила	Ширина стрічки, мм
MP5350AI	Seikosha	—	24
Stylus Color 600	Epson	Quick Dry	—

При виникненні такої ситуації більшість реальних СУБД допускають використання так званих **NULL-** чи **NIL-значень**.

Нехай дане відношення R . Тоді **зовнішній ключ** K' у відношенні R' — це підмножина схеми відношення R' , така що:

існує відношення R'' з *потенційним* ключем K'' ;

для будь-якого кортежу відношення R' значення K' або є *NULL-значенням*, або збігається зі значенням K'' деякого кортежу відношення R'' . Відношення R'' називається **батьківським** чи **головним**, R' — **деталізованим** чи **підлеглим**, а ключ K'' — **батьківським**.

У прикладі декомпозиції відношення R атрибут A у відношенні R'' є потенційним та батьківським ключем, а у відношенні R' — зовнішнім.

Зауваження. У більшості випадків при проектуванні СБД K'' є *первинним ключем* відношення R'' .

Рекомендація. Треба, за можливістю, уникати використання NULL-значень. Наприклад, шляхом поділу сутності на супертип і підтипи.

Тепер розглянемо правила формалізації деяких типів зв'язку [1]. Усі відношення представлені в НФБК. Це *необов'язкова вимога*, тому що:

1) проектування за допомогою ФЗ може бути виконано і на заключних етапах розробки СБД;

2) схема відношення може бути настільки великою (на практиці більш 20 атрибутів), що це істотно ускладнює визначення ФЗ і проектування за їхньою допомогою.

Друга ситуація вимагає представлення сутності у вигляді декількох відношень.

1. *Зв'язок 1:1.* Це єдиний тип зв'язку, що допускає **два варіанти представлення**. Якщо між двома сутностями існує безумовний зв'язок 1:1, то для їхнього представлення досить *одного відношення*. Причому **первинним ключем цього відношення може бути ідентифікатор кожної із сутностей**. Якщо ж за умовами задачі або системи необхідно кожну сутність представити у вигляді окремого відношення, то для **формалізації зв'язків необхідно атрибути первинного ключа одного з відношень додати в схему іншого відношення як зовнішній ключ**.

На ER-діаграмі це відіб'ється просто додаванням множини атрибутів до списку атрибутів другої сутності.

На ІМ, і в цьому її більша наочність, зовнішній ключ буде представлений допоміжними атрибутами з обов'язковою вказівкою мітки зв'язку. Приклад R1 (рис. 1).

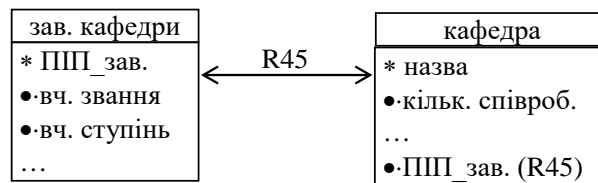


Рисунок 1 – Формалізація зв'язку 1:1

На ІМ MS Access, замість мітки зв'язку використовується її графічна інтерпретація: сама лінія зв'язку з'єднує відповідні атрибути зовнішнього та батьківського ключів. Приклад R1 в нотації MS Access (рис. 2).

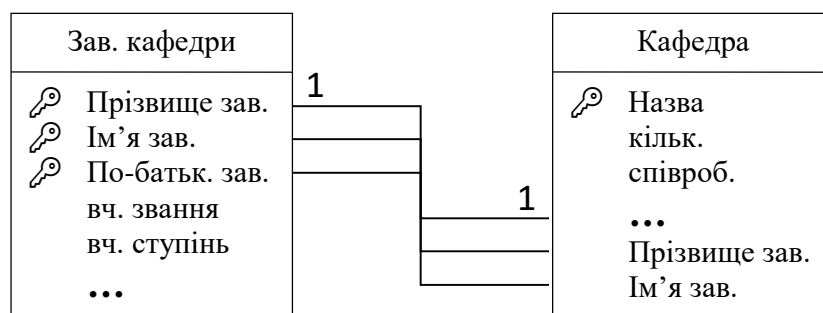


Рисунок 2 – Формалізація зв'язку 1:1 в нотації MS Access

2. Зв'язок 1:1у. Якщо між сутностями встановлений зв'язок 1:1у, то

а) кожна сутність представляється **окремим відношенням**. Причому **ID сутності стає первинним ключем відповідного відношення**.

б) ID сутності, що відповідає умовності зв'язку, додається як **допоміжний атрибут** (на ІМ ще й з міткою зв'язку) в іншу сутність і, відповідно, як зовнішній ключ у схему її відношення.

Приклад R4 (рис. 3):

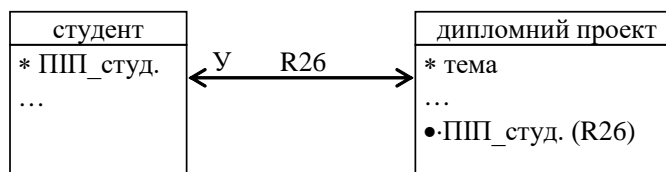


Рисунок 26 – Формалізація зв'язку 1у:1

До речі, допоміжний атрибут у цьому випадку буде одним з потенційних ключів цього ж відношення і може бути використаний для пошуку інформації.

3. Зв'язок 1у:1у. Для **формалізації** зв'язку цього типу **необхідно** використовувати **3 відношення**: по одному для кожної сутності й одне для зв'язку. На ІМ останнє відношення представляється **асоціативним об'єктом (сутністю)**. Атрибутами цього відношення обов'язково будуть первинні ключі двох інших. Приклад R8 (рис. 4):

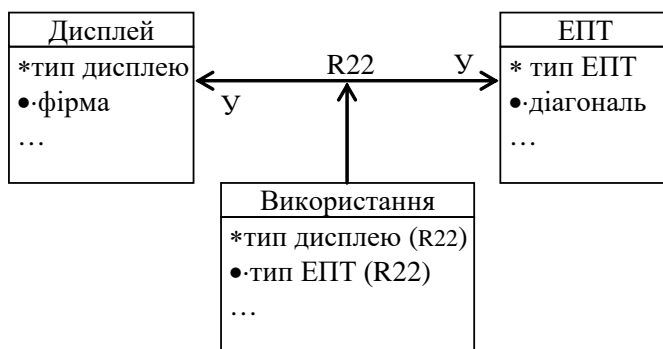


Рисунок 4 – Формалізація зв'язку 1y:1y

При цьому типі зв'язку обидві множини атрибутів асоціативного об'єкта є потенційними ключами. Причому, якщо один з них вибирається як первинний ключ цього відношення, то другий автоматично вважається зовнішнім. Крім того, асоціативний об'єкт може мати і множину власних атрибутів.

4. Зв'язки 1:N і 1y:N. Для формалізації зв'язків типу 1:N не має значення, чи є зв'язок з боку однозв'язної сутності умовним чи безумовним.

Кожна сутність, яка бере участь у зв'язку типу 1:N чи 1y:N, представляється своїм відношенням. При цьому первинний ключ відношення, що відповідає однозв'язній сутності, додається як зовнішній ключ у схему відношення N-зв'язної сутності.

Приклад R2 (рис. 5).

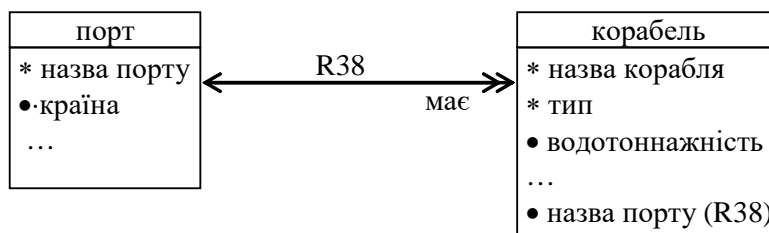


Рисунок 5 – Формалізація зв'язку 1:N, 1y:N

Зрозуміло, що зовнішній ключ відношення N-зв'язної сутності не може бути потенційним ключем цієї сутності.

5. Зв'язки 1:Ny і 1y:Ny. Для визначення зв'язку 1:N, умовного з боку N-зв'язної сутності, необхідно сформулювати по одному відношенню для кожної сутності й одне відношення для зв'язку, у схему відношення якого повинні бути включені первинні ключі відношень сутностей.

Приклад R9 представлено на рис. 6.

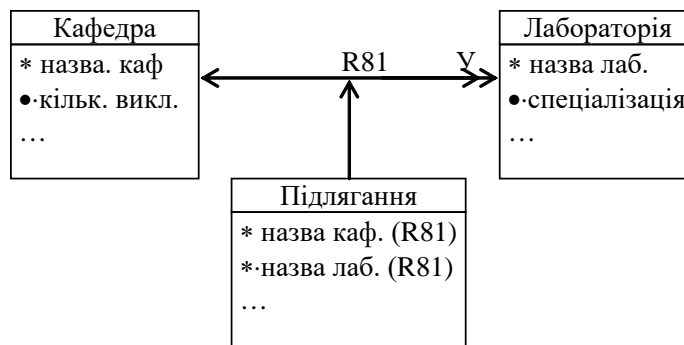


Рисунок 6 – Формалізація зв'язку 1:Ny і 1y:Ny

Потенційними ключами відношення, що відповідає асоціативній сутності, може бути ключ відношення N-зв'язної сутності або множина, що складається з ключів обох відношень.

6. Зв'язок $M:N$ будь-якої умовності (БУ, У, 2У) також **вимагає трьох відношень для формалізації: по одному для кожної сутності й одне для зв'язку. Причому, останнє повинно мати в схемі відношення первинні ключі двох інших.** Більш того, якщо асоціативний об'єкт не має чи не повинен мати власного атрибута-ідентифікатора, то первинним ключем може бути тільки множина, що містить обидва зовнішні ключі. Якщо при цьому відсутні і будь-які інші власні атрибути цього відношення, то кажуть, що воно *цілком є ключем*.

Приклад R3 представлено на рис. 7.

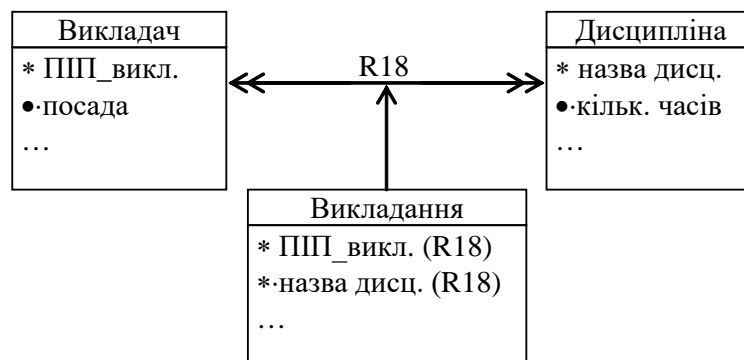


Рисунок 7 – Формалізація зв'язку $M:N$

7. При формалізації зв'язків „супертип-підтип“ **відношення підтипів повинні включати як первинні ключі первинний ключ відношення супертипу, що показано на розглянутому раніше прикладі.** Крім того, між підтипами, у свою чергу, можуть існувати зв'язки, які формалізуються за вже розглянутими правилами.

Іноді, для прискорення пошуку інформації, у відношення супертипу додається атрибут, значення якого говорять про те, якому підтипу відповідає цей екземпляр супертипу (щось як мітки).

8. Іноді для опису предметної області **недостатньо бінарних зв'язків між сутностями.**

Для прикладу розглянемо три сутності: *Викладач*, *Дисципліна*, *Спеціальність* — *Викладач* читає конкретну *Дисципліну* конкретної *Спеціальності*:

Якби множини атрибутів цих сутностей обмежилися атрибутами приведенного відношення, то було б досить нормалізувати це відношення, використовуючи ФЗ. Однак кожна сутність має де 5-10 атрибутів, і така реалізація дуже проблематична.

Ми маємо справу з *тернарним* зв'язком, причому кожна сутність є *N-зв'язковою*. Такий

зв'язок графічно може бути показаний початково *тільки* на *ER*-діаграмі (рис. 8):

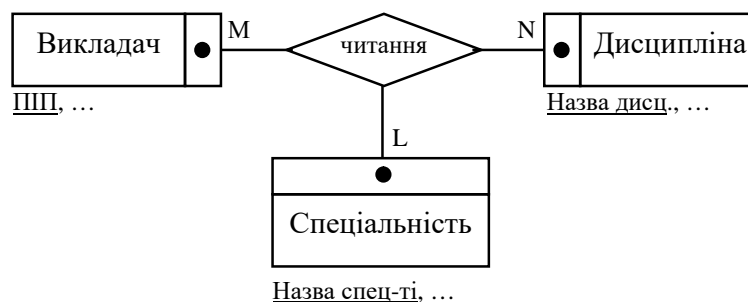


Рисунок 8 – *ER*-діаграма прикладу *n*-арного зв'язку

Для формалізації *n*-арного зв'язку необхідно *n+1* відношення: по одному для кожної сутності, що беруть участь у зв'язку, і одне для самого зв'язку. Причому відношення для зв'язку, точніше, його схема повинна включати первинні ключі всіх інших *n* відношень. Більш того, множина усіх цих ключів буде первинним ключем відношення зв'язку.

На кінець треба додати, що в нотації *MS Access* відсутня можливість графічного представлення зв'язків, формалізація яких вимагає створення асоціативної сутності. Тому при їхній формалізації треба використовувати підхід, запропонований в мережевій моделі даних. Наприклад, для зв'язку *M:N* (рис. 9):

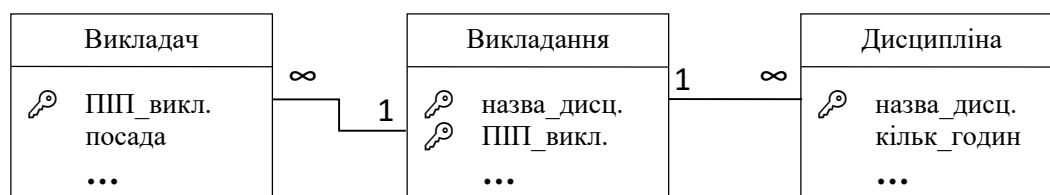


Рисунок 9 – Формалізація зв'язку *M:N* в нотації *MS Access*

Таким же чином, проте *тільки після формалізації*, може бути представлений і *n*-арний зв'язок (рис. 10):



Рисунок 10 – Представлення *n*-арного зв'язку в нотації *MS Access*