

## Навчання баз даних в основній школі: методика й типові помилки

Завадський Ігор Олександрович,  
доктор фіз.-мат. наук, доцент КНУ ім. Т. Шевченка,  
гол. наук. співробітник Інституту педагогіки НАПН України;  
[ihorza@gmail.com](mailto:ihorza@gmail.com)

**Анотація.** У статті досліджено типові помилки в означенні й тлумаченні фундаментальних концепцій баз даних, яких часто припускаються українські вчителі та автори підручників з інформатики. Продемонстровано змістові зв'язки між темами, що становлять ядро компетенції «опрацювання таблиць як наборів однотипних об'єктів» — «Опрацювання табличних даних», «Бази даних. Системи керування базами даних» та «Алгоритми та програми» в курсі інформатики 9 класу основної школи. На прикладі онлайн-ового інтерактивного підручника «ІТ-книга» продемонстровано методику навчання основних понять баз даних на пропедевтичному рівні, що відповідає вимогам чинної програми з інформатики для 9 класу.

**Ключові слова:** бази даних, інформатика, основна школа, розвивальний.

### Вступ

У пояснювальній записці до чинної навчальної програми з інформатики для основної школи [1] зазначено, що в основу курсу інформатики покладено **розвивально-компетентнісний** підхід. Таке формулювання передбачає певний дуалізм, який характерний для української шкільної інформатики загалом. Найчастіше під розвивальною складовою розуміють розвиток алгоритмічного мислення, який має забезпечуватися завдяки вивченню теми «Алгоритми та програми», а решту курсу відносять до «компетентнісної складової». Безперечно, розвиток алгоритмічного мислення є одним із найважливіших завдань курсу інформатики. Однак досить часто він неправомірно проголошується самоціллю, а прикладні програми розглядаються як теми «другого сорту», що тільки заважають вчитися програмувати. Найприкріше, коли у категорію «вигнанців» потрапляє така технологія, як бази даних, хоча насправді її вивчення є не менш розвивальним, ніж програмування. Цього не помічають тому, що вона дає змогу розвинути інший, ніж алгоритмічний, тип мислення, а саме, мислення *структурне*. До цього типу мислення, згідно з [1], належать такі вміння:

- визначати параметри об'єктів та їх можливі значення;
- класифікувати явища та об'єкти;
- знаходити структурні зв'язки між класами об'єктів, класифікувати знайдені зв'язки;
- подавати дані в табличному та графічному вигляді, інтерпретувати дані, подані графічно;
- формулювати задачі з опрацювання структур даних і формалізувати їх з метою подальшого автоматизованого розв'язування з використанням ІКТ-засобів.

Образно кажучи, мова йде про вміння «розкладати дані по полочках», яке в сьогоденнішньому переповненому інформацією світі надзвичайно важливе. Технологію створення й обробки баз даних категорично не можна перелічувати через кому в ряді «офісних» технологій. Вона незрівнянно фундаментальніша, ніж, скажімо, обробка текстів чи створення презентацій. СКБД по суті не є офісною програмою, а MS Access включено в пакет Office лише з маркетингових міркувань.

Звичайно, основи структурного мислення закладаються не винятково під час вивчення основ баз даних. Так, табличний процесор містить цілий комплекс засобів для опрацювання таблиць як наборів однотипних об'єктів (сортування, фільтрація, підсумки тощо), а також є якнайкращим засобом розвитку вміння «подавати дані в табличному та графічному вигляді, інтерпретувати дані, подані графічно». Крім того, вивчення основ об'єктно-орієнтованого програмування у 8 класі та методів опрацювання структур даних у 9 класі в темі «Алгоритми та програми» також є істотним внеском у розвиток структурного мислення.

Загалом 3 теми курсу інформатики 9 класу — «Опрацювання табличних даних», «Бази даних. Системи керування базами даних» та «Алгоритми та програми» взаємопов'язані і є центральними у формуванні структурного мислення в курсі шкільної інформатики. У розділі 2 даної статті ми спробуємо описати підходи до інтегрованого вивчення цих тем, а в розділі 3 запропонуємо методику навчання проектуванню баз даних — першої і, можливо, найважливішої змістової частини теми «Бази даних», навівши численні приклади з інтерактивного онлайн-ового підручника «ІТ-книга» [2]. Однак перед тим мусимо вказати на деякі помилки в тлумаченні фундаментальних концепцій теорії баз даних, які, на превеликий жаль, вкорінилися в інформатичній освіті. Цьому присвячено перший розділ.

### 1. Бази даних: міфи шкільної інформатики

Доводиться констатувати, що в українській шкільній інформатиці склалися негарні традиції переплутування та хибної інтерпретації деяких ключових понять, що належать до галузі моделювання предметних областей — основи будь-якого навчання баз даних. А коли йдеться про реалізацію моделей предметних областей у реляційних базах даних, до цього часто додається несвідоме наслідування авторами підручників та вчителями деяких концепцій найдавнішої ієрархічної моделі даних, які насправді незастосовні до реляційних баз. Отже, перш ніж розглядати методику вивчення баз даних в основній школі, спробуємо розвіяти деякі найпоширеніші міфи, що побутують у шкільній інформатичній освіті стосовно цієї теми.

**Міф 1, кумедний.** «Основною ознакою реляційної моделі даних є наявність зв'язків (relations) між таблицями. Звідси походить і назва моделі».

Насправді «зв'язок» англійською буде «relationship», а «relation» означає «відношення» — математичний об'єкт, підмножину декартового добутку кількох множин. Відношення зручно зображувати у вигляді таблиць і в теорії реляційних баз даних терміни «таблиця» і «відношення» майже еквівалентні. Тому правильніше було б назвати цю модель даних «табличною», а не калькувати її назву з англійської. Подання структур даних у вигляді таблиць і є визначальною ознакою реляційної моделі даних.

А кумедність міфу полягає в тому, що реляційна модель — єдина серед усіх моделей даних, у якій не передбачено жодних механізмів безпосереднього зв'язування об'єктів. В ієрархічній, мережевій та об'єктно-орієнтованій моделях даних зв'язки створюються за допомогою **вказівників** — полів, що належать одним об'єктам та вказують на місця в пам'яті, де розташовано інші об'єкти. У реляційній моделі зв'язки моделюються непрямо, за допомогою **зовнішніх ключів** — додаткових, штучно створених полів, які безпосередньо нікуди не вказують, а лише містять значення, за якими можна знайти зв'язані записи в іншій таблиці. Тому однією з характерних ознак реляційної моделі є якраз **відсутність** безпосередніх зв'язків між об'єктами, що зберігаються в базі.

**Міф 2, абсурдний.** «Є 4 типи зв'язків: один-до-одного, один-до-багатьох, багато-до-одного та багато-до-багатьох».

Цілком очевидно, що «один-до-багатьох» і «багато-до-одного» — це один той самий тип зв'язку. «Відмінність» між цими «типами зв'язків» суто лінгвістична: якщо сказати

«учень вчиться у школі», вийде зв'язок «багато-до-одного», а якщо «у школі вчатьс я учні» – «один-до-багатьох», хоча в дійсності йдеться про один той самий зв'язок.

Ця помилка походить від переплутування концепцій реляційної та ієрархічної моделей даних, адже в реляційній моделі даних зв'язки симетричні (якщо школа 17 зв'язана з учнем Петровим, то і учень Петров – зі школою 17), а в ієрархічній моделі – спрямовані (з того, що школу 17 зв'язано з учнем Петровим ще не випливає, що учень Петров зв'язаний зі школою 17).

**Міф 3, найпоширеніший і найстрашніший.** «Таблиці А і В з'єднано зв'язком «один-до-одного», якщо одному запису А відповідає один запис В. Таблиці А і В з'єднано зв'язком «один-до-багатьох», якщо одному запису А відповідає багато записів В. Таблиці А і В з'єднано зв'язком «багато-до-багатьох», якщо багатьом записам А відповідає багато записів В».

Хибність такого означення зв'язку «багато-до-багатьох» стає очевидною, якщо зауважити, що зв'язки «один-до-одного» та «один-до-багатьох» йому також задовольняють. Справді, як би не були зв'язані таблиці, а **багатьом** записам однієї таблиці завжди відповідатимуть багато записів іншої.

Хибність наведеного означення зв'язку «один-до-багатьох» впливає з того, що йому задовольняє і зв'язок «багато-до-багатьох». Наприклад, є зв'язок «читач прочитав книжку» – зв'язок типу «багато-до-багатьох» між читачем і книжкою. Але, оскільки один читач прочитав багато книжок, цей зв'язок, згідно з наведеним вище означенням, можна було б класифікувати як зв'язок типу «один-до-багатьох».

Наведене вище означення зв'язку «один-до-одного» є неповним (порівняйте його з поданим нижче коректним означенням).

На жаль, спілкування автора з учителями інформатики свідчить, що більшість із них дають учням саме ці категорично хибні, неприпустимі означення, вони наведені і в рекомендованому МОН підручнику для 10-11 класів, що поширюється наймасовішим накладом. Звичайно, якщо у фундамент будови закладати такі «міни», ні про яке коректне та ґрунтовне подальше вивчення баз даних не йдеться.

Для коректного визначення типу зв'язку між сутностями (таблицями) А і В слід відповісти на два запитання:

- Зі скількома об'єктами сутності А може бути зв'язано кожен об'єкт сутності В?
- Зі скількома об'єктами сутності В може бути зв'язано кожен об'єкт сутності А?

Як відповіді на ці запитання визначають типи зв'язків, і як вони графічно позначаються на моделі «сутність-зв'язок», показано в табл. 1 і детально описано в [3], [4].

Таблиця 1. Різновиди зв'язків

Зі скількома об'єктами А може бути зв'язано один об'єкт В?	Зі скількома об'єктами В може бути зв'язано один об'єкт А?	Тип зв'язку	Графічне позначення
з одним	з багатьма	один-до-багатьох	
з одним	з одним	один-до-одного	
з багатьма	з багатьма	багато-до-багатьох	

Найпоширеніша помилка з цього питання у шкільній інформатиці полягає в тому, що замість двох запитань для визначення типу зв'язку ставлять одне. Насправді

відповідь на одне запитання визначає множинність зв'язку тільки з одного боку (тільки один символ "1" чи "∞"). Тому коректні означення типів зв'язку мають враховувати відповіді на два запитання і можуть бути такими.

- Сутності А і В з'єднано зв'язком «один-до-одного», якщо кожному об'єкту А відповідає не більше одного об'єкта В, а кожному об'єкту В – не більше одного об'єкта А.
- Сутності А і В з'єднано зв'язком «один-до-багатьох» (зображеним у першому рядку табл. 1), **якщо кожному об'єкту В відповідає не більше одного об'єкта А**, але кожному об'єкту А може відповідати довільна кількість об'єктів В.
- Сутності А і В з'єднано зв'язком «багато-до-багатьох», якщо кожному об'єкту А може відповідати довільна кількість об'єктів В, а кожному об'єкту В – довільна кількість об'єктів А.

Найчастіше «забувають» виділену жирним частину означення зв'язку «один-до-багатьох», хоча вона є важливішою за іншу його частину. Щоб зрозуміти це, поглянемо на зв'язки з іншої точки зору: зв'язкам «один-до-одного» та «один-до-багатьох» відповідають певні обмеження на кількість об'єктів, які можуть бути зв'язані з одним об'єктом іншої сутності, причому накладаються вони саме множинністю «1». Так, обмеження, що накладається зв'язком «один-до-багатьох», полягає в тому, що кожен об'єкт В не можна зв'язувати більше, ніж з одним об'єктом А. А у зворотний бік (символ «∞») обмеження немає: кількість об'єктів В, з яким може бути зв'язано кожен об'єкт А, необмежена. Тому зв'язки можна було б класифікувати так: у зв'язку «один-до-одного» обмеження на кількість об'єктів, з якими зв'язано один об'єкт іншої сутності, діє в обидва боки, у зв'язку «один-до-багатьох» – в один бік, а у зв'язку «багато-до-багатьох» – не діє зовсім. Це обмеження є різновидом обмежень цілісності для бази даних і тип зв'язку визначається саме тим, як воно підтримується чи не підтримується (тому на схемі даних у MS Access тип зв'язку відображується тільки після встановлення прапорця «з підтримкою обмежень цілісності»). Годі й казати, що в означенні типу зв'язку не можна забувати про визначальну ознаку цього типу.

Зауважимо також, що в загальному випадку, коли зв'язок між сутностями А і В має з боку сутності А множинність «один», кожен об'єкт В зв'язано не точно з одним, а не більше, ніж з одним об'єктом А (тобто об'єкт В зв'язано з якимось об'єктом А не обов'язково). Якщо ж кожен об'єкт В повинен бути зв'язаний з якимось об'єктом А, зв'язок називається **обов'язковим** з боку сутності В. У MS Access обов'язковість зв'язку задається за допомогою властивості «Обов'язкове поле» зовнішнього ключа тієї сутності, з боку якої зв'язок обов'язковий.

**Міф 4, термінологічний.** «Зовнішній ключ – це ключ іншої таблиці, з якою зв'язано дану таблицю».

Насправді зовнішній ключ – це додаткове поле (або набір полів), значення якого належать множині значень первинного ключа іншої таблиці. Наприклад, якщо є зв'язані таблиці «Учні» та «Класи», то зовнішнім ключем таблиці «Учні» буде не ключ таблиці «Класи», а поле «клас» таблиці учнів (рис. 1). Зазначимо, що причиною помилкового тлумачення цього поняття є його невдала назва: природне значення слова «зовнішній» (англ. foreign) – той, який не належить розглядуваному об'єкту – у даному випадку незастосовне. Правильніше зовнішні ключі було б назвати полями зв'язку.

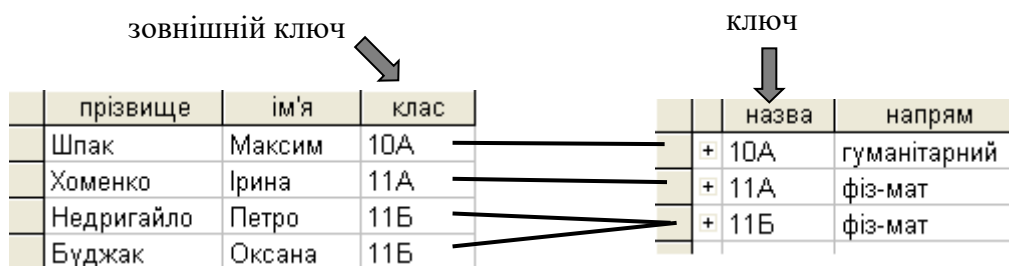


Рис. 1. Зв'язування записів за допомогою зовнішнього ключа

## 2. Компетенція «опрацювання таблиць як наборів однотипних об'єктів» в курсі інформатики 9 класу

Як уже зазначалося, останні три теми курсу інформатики 9 класу пов'язані наскрізно змістовою лінією, яку можна назвати «робота з наборами однотипних об'єктів». Традиційно електронні таблиці розглядаються як набори клітинок із даними. Така інтерпретація передбачалася в першій частині теми «Опрацювання табличних даних», що вивчалася в 7 класі, а також доречно під час вивчення певного матеріалу з однойменної теми 9 класу, такого як абсолютні та мішані посилання, функції, діаграми та умовне форматування.

Однак електронну таблицю можна розуміти також як набір рядків, у кожному з яких містяться відомості про певний об'єкт. Ці об'єкти однотипні, оскільки складаються з однакового набору параметрів, яким відповідають стовпці таблиці. На практиці електронні таблиці найчастіше використовують саме в такому розумінні. Це різноманітні каталоги, розклади руху, електронні щоденники тощо. Така інтерпретація передбачає виконання операцій не з окремими клітинками, а з цілими рядками, і для цього в табличному процесорі передбачено спеціальні засоби: фільтрацію, сортування, обчислення підсумкових характеристик. Важливо, щоб учень усвідомлював «об'єктну природу» більшості електронних таблиць, що й передбачено у знанневій та ціннісній складових компетентності у відповідній темі навчальної програми.

Щойно описане тлумачення табличних даних стає єдино можливим, коли йдеться про реляційні бази даних. Електронна таблиця-каталог фактично є однотобличною реляційною базою даних і цей факт полегшує перехід до вивчення такої загалом складної теми як «Бази даних. Системи керування базами даних». У 9-му класі ця тема має пропедевтичний характер. Метою її вивчення є підготовка учнів до більш ґрунтовного опанування технології в старшій школі та виконання вимог Державного стандарту базової і повної загальної середньої освіти, затвердженого Постановою КМУ №1392 від 23.11.11 року, на якому базується чинна навчальна програма з інформатики для 5–9 класів. У 9 класі в цій темі йдеться навіть не про створення цілісного уявлення про технологію баз даних, а про засвоєння кількох фундаментальних понять: таблиця, поле, запис, ключ, тип даних, та простих операцій над відповідними об'єктами. По суті це ті самі операції, що й над однотипними об'єктами в електронній таблиці: додавання, видалення, фільтрація, сортування тощо. Навчальна програма явним чином не вказує на необхідність створення учнями багатотабличних баз даних і роботи з ними, а така фундаментальна концепція, як зв'язок між таблицями, у програмі навіть не згадується. Однак, якщо обмежуватися лише однотобличними базами даних, важко сформулювати таку ціннісну складову компетентності, як «усвідомлює переваги використання баз даних в інформаційних системах», оскільки переваги однотобличної бази даних над електронною таблицею незначні. Тому варто пояснювати учням деякі принципи функціонування технології на прикладі вже готових багатотабличних баз

даних та/або самостійно створювати просту двотабличну базу даних, яка, наприклад, може містити інформацію про сутності, що пов'язані зв'язком «один-до-багатьох».

У 9 класі не обов'язково моделювати зв'язки на схемі даних в MS Access, однак необхідно пояснити учням, що об'єкти з різних таблиць взаємопов'язані і опрацювати створення найпростішого зв'язку безпосередньо між записами, завдяки введенню даних у таблиці. Зауважимо, що схема даних у MS Access є тільки унаочненням, графічним відображенням зв'язків, вона не містить зв'язків як таких. Наприклад, створення зв'язку між учнем та класом може полягати у введенні в поле «клас» запису, що стосується учня Петрова, значення «10А», а схему даних з цією метою можна було б не використовувати взагалі, адже в більшості реляційних СКБД такого засобу немає. Проте, крім презентаційної ролі, схема даних в MS Access має ще й функціональне призначення, адже саме під час відображення зв'язків на схемі даних створюють **обмеження цілісності**, яким мають задовольняти зовнішні ключі (див. розділ 1). І, хоча поняття обмеження цілісності також не згадується в навчальній програмі для 9 класу, його вивчення в основній школі може бути доцільним з огляду на його фундаментальність. У такому випадку доцільним стає й відображення найпростішого зв'язку «один-до-багатьох» на схемі даних.

Алгоритмічна складова технології опрацювання наборів однотипних об'єктів опановується насамперед в темі «Алгоритми та програми». Однак важливо, щоб учні засвоїли сутність операцій, які вони програмуватимуть, під час вивчення попередніх двох тем. Так, для обчислення підсумкових величин призначено статистичні функції табличного процесора, а пошук у масиві за певними критеріями — це не що інше як фільтрація. Також в базах даних та електронних таблицях повсюдно застосовується операція сортування, алгоритми виконання якої досліджуються в «алгоритмічній» темі курсу. Таким чином, теми «Бази даних. Системи керування базами даних» та «Опрацювання табличних даних» відіграють, поміж іншого, пропедевтичну роль для подальшого опанування теми «Алгоритми та програми».

### 3. Методика навчання основних понять баз даних у 9 класі

Як зазначалося вище, основною відмінністю реляційних баз даних від електронних таблиць є багатотабличність. Однак для чого взагалі може знадобитися багато таблиць? Проста і загалом правильна відповідь на це питання впливає з матеріалу теми, присвяченої електронним таблицям: оскільки таблиця — це засіб подання даних про набір однотипних об'єктів, то відомості про різнотипні об'єкти слід зберігати в різних таблицях. Проте, суто технічно, навіть великі й різнотипні дані завжди можна «умістити» в одну таблицю, створивши для кожного типу об'єктів окремий набір стовпців. Тому вищенаведене питання не пусте, і його детальніше дослідження на простих прикладах дасть учням змогу зрозуміти саму сутність інформаційної технології баз даних. В ІТ-книзі це питання розглядається на прикладі, наведеному на рис. 2. У його верхній частині зображено таблицю, що містить відомості про автобуси та маршрути, якими вони курсують. Це різнорідні дані, які, однак, належать одній предметній області. Недоліки такого способу зберігання даних неочевидні, адже всю інформацію про автобуси, маршрути та їхні зв'язки збережено: з таблиці можна однозначно дізнатися, яку модель чи рік випуску має певний автобус, яку довжину та кількість зупинок має маршрут, а також яким маршрутом який автобус курсує.

### Автобуси та маршрути

Номерний знак	Модель	Рік випуску	Маршрут	Довжина (км)	Кількість зупинок
AA3129KH	CityLAZ-12	2010	9	7,3	18
AA1684PT	Богдан А144	2017	9	7,3	18
AI4124KK	CityLAZ-12	2012	9	7,3	18
AA3312TB	МАЗ 103	2001	9	7,3	18
AI2917KT	Богдан А144	2017	12	5,4	14
AA4409KH	CityLAZ-12	2012	12	5,4	14



Рис. 2. Поділ таблиці, що містить різномірні дані

Проте в таблиці з верхньої частини рис. 2 можна зауважити й певну нерациональність: усі відомості про кожен маршрут повторюються багаторазово. Це, очевидно, призводить до зайвих витрат пам'яті, а отже, є недоліком. Утім, цей недолік не критичний, якщо зважити на обсяги пам'яті сучасних комп'ютерів. Якщо ж проаналізувати, що може трапитися в разі змінення (у термінології баз даних — **оновлення**) даних, то ситуація видаватиметься значно гіршою. Припустимо, що маршрут № 9 подовжили і тепер він містить 20 зупинок, а його нова довжина становить 9 км. Оскільки інформація про довжину та кількість зупинок 9 маршруту міститься в таблиці стільки разів, скільки на цьому маршруті є автобусів, то і зміни потрібно вносити багаторазово. А де гарантія, що ці зміни будуть узгодженим чином внесені в усі записи, які стосуються 9 маршруту? Адже оператор, який вносить зміни, може просто не знайти всіх записів, або забути змінити деякі, або, скажімо, під час внесення змін зникне струм в електромережі тощо — у результаті може так статися, що в частині записів маршрут №9 матиме 18 зупинок і довжину 7,3 км, а в частині — 20 зупинок і довжину 9 км. Тоді дані про маршрут стануть суперечливими, або, інакше кажучи, взагалі **втраченими**, адже буде неможливо визначити, яку ж довжину має насправді маршрут №9 — 7,3 км чи 9 км, скільки на ньому зупинок — 18 чи 20 (рис. 3).

### Автобуси та маршрути

Номерний знак	Модель	Рік випуску	Маршрут	Довжина (км)	Кількість зупинок
AA3129KH	CityLAZ-12	2010	9	9	20
AA1684PT	Богдан А144	2017	9	7,3	18
AI4124KK	CityLAZ-12	2012	9	7,3	18
AA3312TB	МАЗ 103	2001	9	7,3	18
AI2917KT	Богдан А144	2017	12	5,4	14
AA4409KH	CityLAZ-12	2012	12	5,4	14

Рис. 3. Суперечливі (втрачені) дані в таблиці автобусів і маршрутів

У такий спосіб учнів можна підвести до розуміння одного з основних призначень баз даних — **унеможливити втрату даних**. Причому це має забезпечуватися не завдяки «зовнішнім» механізмам, наприклад трудовій дисципліні оператора, який вводить дані, чи досвіду програміста, який створює програмне забезпечення для роботи з базою, а «всередині», завдяки самій структурі бази. Основним структурним механізмом, який унеможливило втрату даних, є поділ таблиці на кілька менших. Так, у нижній частині рис. 2 таблицю маршрутів і автобусів поділено на дві: таблицю маршрутів і таблицю автобусів (мухи — окремо, котлети — окремо) і тепер аномалій, пов'язаних з оновленням інформації про маршрути, не виникає.

Однак після такого поділу втрачено іншу важливу інформацію: тепер невідомо, який автобус яким маршрутом курсує! Яким же чином виправити цей недолік? В ІТ-книзі учням пропонується вибрати один з кількох варіантів відповіді на це питання:

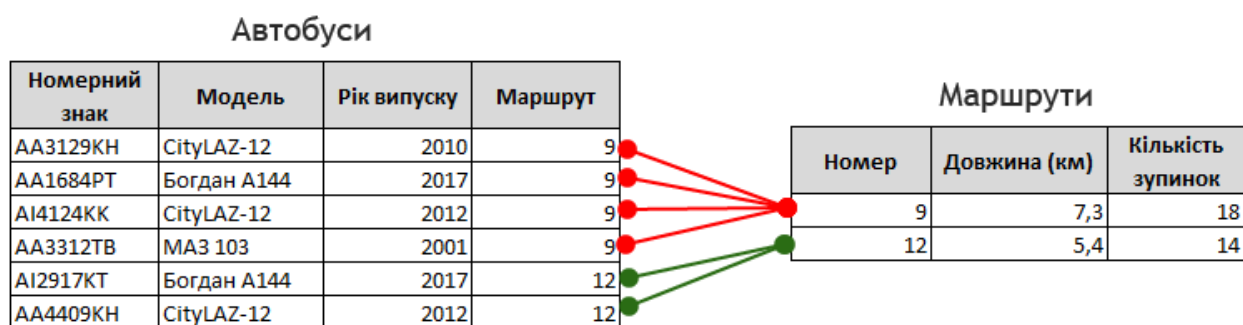
- 1) додати в таблицю «Автобуси» стовпець, де зберігатиметься № маршруту;
- 2) додати в таблицю «Маршрути» стовпець, де зберігатимуться номерні знаки автобусів;
- 3) додати в таблицю «Автобуси» стовпець, де зберігатиметься довжина маршруту;
- 4) додати в таблицю «Маршрути» стовпець, де зберігатиметься модель автобусу.

Правильною є, звичайно, перша відповідь (рис. 4а), однак учням важливо розуміти, чому кожна з інших відповідей є неправильною. Так, у разі вибору відповіді 2) утвориться ситуація, зображена на рис. 4б, з якою пов'язано два основних негаразди:

- по-перше, в клітинках стовпця «Автобуси» зберігатимуться не окремі значення, а цілі набори, які важко опрацьовувати (наприклад, щоб знайти в наборі певний елемент, потрібно писати окрему програму);

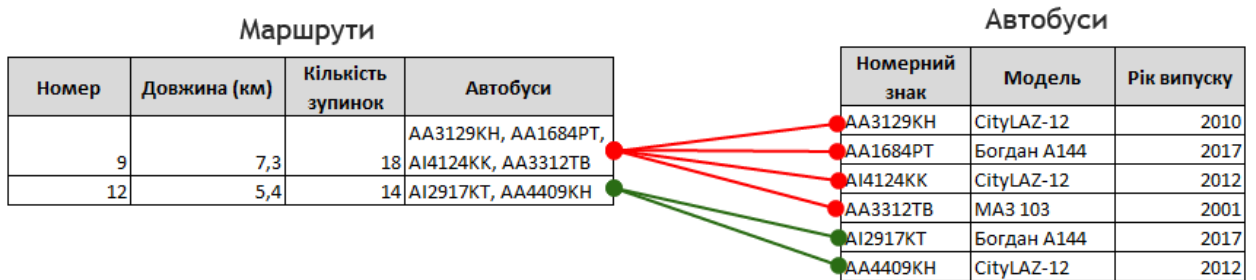
- по-друге, з'явиться можливість призначати один той самий автобус на різні маршрути (вказуючи його номерний знак у різних клітинках), що суперечить здоровому глузду, якщо база даних відображає стан автотранспортного підприємства на певний момент часу.

Аналізуючи ці негаразди, можна дійти двох важливих висновків: 1) значення в клітинках таблиць бази даних мають бути **атомарними** (це один з базових постулатів реляційної моделі даних); 2) таблиці «Маршрути» та «Автобуси» пов'язані зв'язком «один-до-багатьох», оскільки кожним маршрутом може курсувати багато автобусів, однак кожен автобус курсує лише одним маршрутом.



а





б

Рис. 4. Способи зв'язування таблиць автобусів і маршрутів: правильний (а) та неправильний (б)

Зауважимо, що, хоча згідно з навчальною програмою [1], класифікація зв'язків за множинністю у 9 класі не розглядається, пояснити на окремому прикладі, що означає зв'язок «один-до багатьох», все-таки варто, оскільки без цього залишається незрозумілим сам принцип пов'язування даних з різних таблиць реляційної бази.

Тепер проаналізуємо неправильний варіант відповіді 3): додати в таблицю «Автобуси» стовпець, де зберігатиметься довжина маршруту. У такий спосіб зв'язку між автобусами та маршрутами не утвориться, оскільки різні маршрути можуть мати однакові довжини й тому за довжиною неможливо *ідентифікувати* маршрут (а за номером — можна, оскільки різні маршрути не можуть мати однакових номерів). Таким чином, робимо висновок, що за значеннями деяких полів можна ідентифікувати об'єкти, а за значеннями інших — ні, й підходимо до ще одного базового поняття реляційних баз даних — *ключа* таблиці. У 9 класі можна давати означення ключа як поля таблиці, значення якого не можуть повторюватися, а отже, дають змогу ідентифікувати об'єкт. Про той факт, що ключі можуть складатися з декількох полів, для спрощення можна не згадувати.

Нарешті, неправильний варіант 4) поєднує недоліки варіантів 2) і 3): модель не ідентифікує автобус, оскільки різні автобуси можуть мати однакові моделі, а також зберігання відомостей про моделі автобусів у таблиці маршрутів порушує принцип атомарності.

Отже, правильний спосіб зв'язування таблиць «Автобуси» та «Маршрути» проілюстровано на рис. 5. Аналізуючи це зображення, учням буде неважко узагальнити принцип створення зв'язку «один-до-багатьох»: у таблицю, яка розташована з боку «багато», додати стовпець зі значеннями ключа таблиці, яка розташована з боку «1». Як зазначено в розділі 1, цей стовпець називається зовнішнім ключем, але в 9 класі цього поняття можна не вводити, обмежившись терміном «додатковий стовпець» і пам'ятаючи, що поняття зовнішнього ключа обов'язково потрібно буде ввести під час вивчення баз даних у старшій школі.

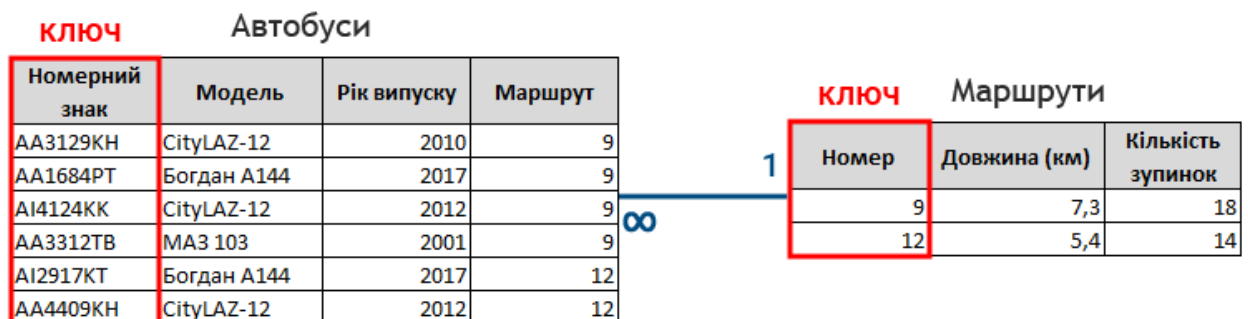


Рис. 5. Реалізація зв'язку «один-до-багатьох» між таблицями маршрутів і автобусів

Таким чином, на одному простому прикладі учні можуть не лише сформувавши уявлення про призначення баз даних й такі фундаментальні поняття, як таблиця, ключ, зв'язок між таблицями, але й зрозуміти базову логіку цієї інформаційної технології. Зауважимо, що про поняття системи керування базами даних (СКБД) у цій статті не йшлося, а отже, даний матеріал можна опановувати «теоретично», на прикладі таблиць, створених у будь-якому інструментальному середовищі. Ми вважаємо такий підхід методично доцільним, адже інакше технологічні аспекти замулюють сутність справи: учень напевно запам'ятає склад меню «Файл» у MS Access і буде тішитися з того, але не зрозуміє жодних фундаментальних понять, хоча друге незрівнянно важливіше від першого.

**Висновки.** Навчання технології баз даних є нарізним каменем формування й розвитку структурного мислення в учнів основної та старшої школи. І хоча навчальна програма з інформатики передбачає лише пропедевтичне вивчення цієї теми в 9 класі, проблемно-орієнтована методика навчання дає можливість учням опанувати сутність більшості основних понять баз даних, не виходячи за межі вимог програми. Таку методику реалізовано в інтерактивному онлайн-овому підручнику «ІТ-книга» і детально розглянуто в цій статті стосовно моделювання баз даних, що відповідає першому уроку ІТ-книги з даної теми. Крім моделювання, усі інші аспекти баз даних, насамперед реалізація моделі та операції з даними в базі, вимагають роботи в СКБД і потребують окремих методичних досліджень.

#### Список використаних джерел

1. Інформатика для учнів 5–9 класів, які вивчали інформатику у 2–4 класах. Навчальна програма. URL: <https://mon.gov.ua/ua/osvita/zagalna-serednya-osvita/navchalni-programi/navchalni-programi-5-9-klas> (Дата звернення: 01.10.2020).
2. Інтерактивний онлайн-овий підручник «ІТ-книга». URL: <https://itknyga.com.ua> (Дата звернення: 01.10.2020).
3. Завадський І.О. Основи баз даних. Навчальний посібник. К., вид. Завадський І.О., 2011. – 192 с, [https://itknyga.com.ua/documents/Бази\\_даних.pdf](https://itknyga.com.ua/documents/Бази_даних.pdf).
4. Дейт К.Дж. Введение в системы баз данных. М.: Вильямс, 2005. – 1328 с.

#### References. Translation and transliteration

1. Informatics for 5<sup>th</sup>-9<sup>th</sup> grade students who studied informatics in 2<sup>nd</sup>-4<sup>th</sup> grades, curriculum. <https://mon.gov.ua/ua/osvita/zagalna-serednya-osvita/navchalni-programi/navchalni-programi-5-9-klas> (Date of access: 10/01/2020).
2. IT-knyga, an interactive online course. URL: <https://itknyga.com.ua> (Date of access: 10/01/2020).
3. Zavadskiy I.O. Data bases basics, textbook. Kyiv, Publisher I.O. Zavadskiy, 2011, 192 p.
4. Date C.J. An Introduction to Database Systems, 8<sup>th</sup> Edition, Addison Wesley, 2004, 1040 p.

## **Завадский Игорь Александрович. ИЗУЧЕНИЕ БАЗ ДАННЫХ В ОСНОВНОЙ ШКОЛЕ: МЕТОДИКА И ТИПИЧНЫЕ ОШИБКИ**

**Аннотация.** В статье рассматриваются типичные ошибки, допускаемые украинскими учителями и авторами учебников по информатике, в определении и трактовке фундаментальных концепций баз данных. Продемонстрированы смысловые связи между темами, составляющими ядро компетенции «использование таблиц как наборов однотипных объектов» — «Обработка табличных данных», «Базы данных. Системы управления базами данных» и «Алгоритмы и программы» в курсе информатики 9 класса основной школы. На примерах из онлайн-интерактивного учебника «ИТ-книга» продемонстрирована методика обучения основным понятиям баз данных на пропедевтическом уровне, в соответствии с требованиями действующей программы по информатике для 9 класса.

**Ключевые слова:** базы данных, информатика, основная школа, развивающий.

## **Zavadskiy Ihor. TEACHING DATA BASES IN SECONDARY SCHOOL: TEACHING TECHNIQUE AND TYPICAL MISTAKES**

**Abstract.** The typical mistakes made by Ukrainian workbook authors and teachers in defining and interpreting of fundamental data bases concepts are discussed. The semantic links between core topics of “processing tables as homogeneous object sets” competency are shown. These topics are “Processing of tabular data”, “Data bases. Data base management systems” and “Algorithms and programs” in the 9<sup>th</sup> grade curriculum of Ukrainian secondary school. The core database concepts propaedeutical teaching technique, in accordance with actual 9<sup>th</sup> grade curriculum, is shown in examples taken from online interactive course “IT-knyga”.

**Keywords:** data bases, computer science, secondary school, evolutive.