



УДК 378.147:004.655

[https://doi.org/10.52058/2786-6300-2026-5\(47\)-3314-3325](https://doi.org/10.52058/2786-6300-2026-5(47)-3314-3325)

**Сікора Ярослава Богданівна** доктор педагогічних наук, доцент, професор кафедри комп'ютерних наук та інформаційних технологій, Житомирський державний університет імені Івана Франка, м. Житомир, <https://orcid.org/0000-0003-2621-6638>

## СПІРАЛЬНА МОДЕЛЬ ПРОЄКТУВАННЯ БАЗ ДАНИХ ЯК МЕТОДИЧНА ОСНОВА ПІДГОТОВКИ МАЙБУТНІХ ІТ-ФАХІВЦІВ

**Анотація.** У статті розглянуто проблему вдосконалення методики навчання баз даних у закладах вищої освіти в умовах цифрової трансформації суспільства та зростання потреби у фахівцях, здатних здійснювати проектування, адміністрування й супровід сучасних інформаційних систем. Обґрунтовано актуальність переходу від традиційного навчання, орієнтованого переважно на засвоєння синтаксису SQL і виконання типових лабораторних робіт, до компетентнісно спрямованої моделі підготовки, що забезпечує формування системного професійного мислення майбутнього розробника баз даних.

Визначено основні недоліки поширених підходів до викладання баз даних, серед яких недостатня увага до аналізу предметної області, концептуального моделювання та практичного застосування знань у реальних професійних ситуаціях.

Проаналізовано сучасні наукові підходи до організації навчання баз даних, зокрема практико-орієнтоване, проблемне та активне навчання, а також використання компетентнісних технологій у підготовці ІТ-фахівців. На основі аналізу запропоновано методику навчання баз даних, що ґрунтується на інтеграції спіральної моделі проектування та багатоетапної організації освітнього процесу. Методика передбачає послідовне проходження навчально-ознайомчого та практико-орієнтованого етапів із поступовим ускладненням завдань і переходом від вивчення базових понять до розробки прототипів інформаційних систем для реальних підприємств.

Окрему увагу приділено використанню спіральної моделі проектування баз даних як засобу формування навичок системного аналізу. Визначено основні етапи такої моделі: аналіз предметної області, концептуальне та даталогічне моделювання, верифікація і нормалізація структури бази даних, фізична реалізація та створення користувацького інтерфейсу. Показано, що ітераційний характер спіральної моделі дозволяє студентам поступово поглиблювати розуміння предметної області, уникати типових помилок проектування та формувати цілісне бачення архітектури інформаційної системи.



Розкрито можливості інтеграції таксономії Б. Блума в систему оцінювання результатів навчання, що забезпечує перехід від перевірки репродуктивних знань до оцінювання здатності студентів аналізувати, проєктувати та створювати працездатні програмні продукти. Запропонована методика сприяє підвищенню професійної мотивації здобувачів освіти, розвитку їхньої практичної підготовки та адаптації до реальних потреб ІТ-галузі. Перспективи подальших досліджень пов'язано з адаптацією навчання баз даних до хмарних технологій, NoSQL-архітектур і систем інтелектуального супроводу освітнього процесу.

**Ключові слова:** спіральна модель, проєктування баз даних, методика навчання, вища освіта, компетентнісний підхід, підготовка ІТ-фахівців, інформаційні системи.

**Sikora Yaroslava** Doctor of Pedagogical Sciences, Associate Professor, Professor at the Department of Computer Science and Information Technology, Zhytomyr Ivan Franko State University, Zhytomyr, <https://orcid.org/0000-0003-2621-6638>

## **THE SPIRAL MODEL OF DATABASE DESIGN AS A METHODOLOGICAL BASIS FOR TRAINING FUTURE IT SPECIALISTS**

**Abstract.** The article addresses the problem of improving database teaching methodology in higher education institutions under the conditions of the digital transformation of society and the growing demand for specialists capable of designing, administering, and maintaining modern information systems. The relevance of transitioning from traditional teaching focused mainly on mastering SQL syntax and performing standard laboratory tasks to a competency-based training model that ensures the development of systematic professional thinking in future database developers is substantiated.

The main shortcomings of common approaches to teaching databases are identified, including insufficient attention to domain analysis, conceptual modeling, and the practical application of knowledge in real professional situations.

The study analyzes modern scientific approaches to organizing database education, including practice-oriented, problem-based, and active learning, as well as the use of competency-based technologies in the training of IT specialists. Based on the analysis, a database teaching methodology grounded in the integration of the spiral design model and a multi-stage organization of the educational process is proposed. The methodology involves the sequential implementation of introductory-learning and practice-oriented stages with the gradual complication of tasks and the transition from studying fundamental concepts to developing prototypes of information systems for real enterprises.



Particular attention is paid to the use of the spiral database design model as a means of developing systems analysis skills. The main stages of this model are defined as follows: domain analysis, conceptual and logical modeling, verification and normalization of the database structure, physical implementation, and user interface development. It is demonstrated that the iterative nature of the spiral model enables students to gradually deepen their understanding of the domain, avoid common design errors, and develop a holistic vision of information system architecture.

The possibilities of integrating B. Bloom's taxonomy into the system of learning outcomes assessment are revealed, ensuring a transition from evaluating reproductive knowledge to assessing students' ability to analyze, design, and develop functional software products. The proposed methodology contributes to increasing students' professional motivation, improving their practical training, and adapting their competencies to the real needs of the IT industry. Prospects for further research are associated with adapting database education to cloud technologies, NoSQL architectures, and intelligent educational support systems.

**Keywords:** spiral model, database design, teaching methodology, higher education, competency-based approach, IT specialists training, information systems.

**Постановка проблеми.** Сучасний етап розвитку глобальної інформаційної економіки характеризується фундаментальними змінами у підходах до обробки, зберігання та аналізу великих масивів даних. В умовах цифрової трансформації суспільства проєктування та розробка баз даних перетворюються на важливу сферу діяльності фахівців з інформаційних технологій. Значущість цього завдання, яке було теоретично обґрунтоване ще у праці К. Дж. Дейта [1], сьогодні набуває нового змісту через необхідність підготовки дедалі більшої кількості ІТ-фахівців, здатних проєктувати та адмініструвати бази даних, писати оптимізовані SQL-запити й аналізувати набори даних.

Заклади вищої освіти постають перед викликом не просто навчити студента синтаксису мови структурованих запитів (SQL), а сформувати цілісну фахову компетентність, що охоплює весь життєвий цикл бази даних – від концептуального аналізу предметної області до фізичної реалізації та супроводу. У зв'язку з цим особливо актуалізується проблема підготовки майбутніх фахівців, яка полягає в необхідності подолання розриву між академічними знаннями та практичними запитами ІТ-індустрії.

**Аналіз останніх досліджень і публікацій.** Використання компетентнісних технологій навчання у системі вищої освіти характеризується насамперед готовністю майбутнього кваліфікованого фахівця ефективно та самостійно діяти у професійній діяльності.

У дослідженні [2, с. 17] науковцями виявлено ключові проблеми навчання баз даних у провідних університетах світу: пріоритетність теоретичної підго-



товки порівняно з практичною, що ускладнює формування прикладних навичок і застосування знань; попри використання сучасних методик і технологій, зберігаються застарілі педагогічні підходи, які обмежують розвиток ініціативності та здібностей студентів; навчання баз даних зводиться до вузького кола завдань, далеких від реальних професійних ситуацій. Вони рекомендують впровадження практико-орієнтованого навчання через прикладні проєкти, що сприяє формуванню практичних умінь і підвищенню ефективності навчання.

Б. Шебаро [3] також обґрунтував ефективність стратегій активного навчання, які позитивно впливають на засвоєння матеріалу.

На думку дослідників А. Uzun, А. Onur, S. Alabay, використання проблемного навчання у викладанні дисципліни «Системи управління базами даних» позитивно впливає на залученість і задоволеність студентів освітнім процесом, а ефективність такого підходу значною мірою залежить від належного проєкування навчального середовища та рівня підготовки здобувачів освіти [4].

Незважаючи на значний досвід викладання курсу «Бази даних», у багатьох закладах вищої освіти спостерігається низка системних недоліків. Основна проблема полягає у недостатній увазі до найбільш складних та інтелектуально містких етапів – аналізу предметної області та концептуального проєкування.

Іноді процес вивчення дисципліни зводиться до виконання лабораторних робіт за методичними посібниками, де структура бази даних уже визначена, поля ключових таблиць вказані та встановлені зв'язки. Здобувач у такому випадку виконує лише технічну роботу з перенесення готової моделі у середовище конкретної системи управління базами даних (СУБД), не розуміючи причин вибору саме такої архітектури. Це призводить до того, що на етапі дипломного проєкування або реальної роботи здобувач освіти виявляється неспроможним самостійно структурувати хаотичну інформацію реального підприємства у цілісну модель. Такий підхід «навчання на прикладі» виправданий лише на початковому етапі знайомства з інструментарієм, але він не сприяє формуванню професійного мислення розробника. У більшості запропонованих методик практично відсутні основні етапи проєкування баз даних: етап аналізу предметної області; етап концептуального проєкування.

Відома низка методик, у яких основну увагу приділяють формальним принципам низхідного проєкування на основі функціональних залежностей та проведення процесу нормалізації [5; 6; 7; 8]. Ці методики дають змогу розглядати проблему проєкування з формальної математичної точки зору та забезпечують студентам глибоке розуміння математичних і логічних засад баз даних.

У ряді публікацій розглядаються окремі питання проєкування баз даних, такі як методики опитування потенційних користувачів авторів [9], поради з окремих питань проєкування баз даних [10; 11; 12], питання підвищення продуктивності при фізичній реалізації бази даних [13]. Ці методики можуть



бути результативними у навчанні проєктування баз даних у поєднанні з методиками, що охоплюють весь цикл проєктування баз даних.

**Мета статті** – теоретично обґрунтувати методику навчання баз даних на основі інтеграції спіральної моделі їх проєктування.

**Виклад основного матеріалу.** Реалізація компетентнісного підходу вимагає дотримання низки принципів при відборі та структуруванні змісту дисципліни:

1) принцип автентичності задач: навчальні завдання мають максимально відтворювати реальні виробничі ситуації (наприклад, аналіз бази даних реального інтернет-магазину чи медичної інформаційної системи);

2) принцип міждисциплінарності: зміст дисципліни пов'язується з курсами з програмування, системного аналізу, математичної логіки та теорії алгоритмів;

3) принцип наскрізної практики: практична складова охоплює не менше 60% навчального часу, причому всі практичні завдання вибудовуються навколо єдиного наскрізного проєкту;

4) принцип рефлексії: здобувачі систематично аналізують власні досягнення і прогалини шляхом формування та наповнення портфоліо.

На основі зазначених принципів пропонується методика, що ґрунтується на багатоетапному інтерактивному навчанні та спіральній моделі проєктування. Її сутність полягає у послідовному проходженні здобувачем освіти двох основних етапів: навчально-ознайомчого та практико-орієнтованого (табл. 1).

*Навчально-ознайомчий етап* розподіляється на початковий та поглиблений рівні. На початковому рівні студенти знайомляться з еволюцією баз даних, реляційною моделлю, поняттями сутності, атрибута та ключа. Практична робота передбачає ознайомлення з можливостями СУБД на прикладі Microsoft Access, де студенти виконують лабораторні завдання зі створення основних об'єктів бази даних. Паралельно здійснюється базове опанування SQL, яке поглиблюється на наступних етапах. Важливим компонентом є формування розуміння принципів цілісності реляційної моделі даних і правил організації зв'язків між таблицями. Студенти знайомляться з інструментами моделювання інформаційних систем, зокрема Oracle SQL Developer Data Modeler, що дозволяє опанувати базові принципи концептуального та логічного проєктування баз даних у контексті сучасних підходів до розробки інформаційних систем.



Таблиця 1

**Етапи навчання проєктування баз даних**

Етапи/рівні навчання	Зміст діяльності студента	Інструментарій
Початковий	Вивчення основ, реляційної алгебри, створення об'єктів у локальних СУБД	MS Access, теоретичний курс.
Поглиблений	Робота в клієнт-серверних середовищах, складні SQL-запити, моделювання	PostgreSQL / MySQL, SQL Developer.
Практичний	Аналіз реального підприємства, проєктування та розробка прототипу інформаційної системи	Oracle, MySQL, PHP/Java для інтерфейсу.

Другий крок навчально-ознайомчого етапу передбачає поглиблене вивчення дисципліни: опрацювання етапів проєктування баз даних у межах конкретної предметної області в інтерактивній взаємодії з викладачем, включаючи аналіз, концептуальне, даталогічне та фізичне проєктування. Паралельно поглиблюються знання з SQL, реляційної алгебри та реляційного числення.

На практичних заняттях студенти самостійно моделюють бази даних із використанням CASE-засобів проєктування та реалізують їх у сучасному середовищі СКБД засобами SQL, що сприяє формуванню стійких практичних навичок

*Практико-орієнтований етап* розбивається на декілька складових: виробничу практику на конкретному підприємстві, проєктування бази даних підприємства; фізичну реалізацію бази даних та розробку інтерфейсу користувача.

Це створює умови для вирішення реальних завдань та підвищує мотивацію, оскільки студент бачить практичну цінність своєї роботи.

Проєктування бази даних пропонується розглядати як ітераційний процес, що описується спіральною моделлю. Спіральна модель (spiral model), запропонована Б. Боемом [14], традиційно використовується для керування ризиками у проєктах програмного забезпечення, проте у навчанні баз даних вона дозволяє здобувачу поступово заглиблюватися у складність предметної області. Кожен виток спіралі – це повернення до тих самих питань проєктування, але на новому рівні глибини та деталізації.

*Перший виток спіралі – визначення сутності і меж предметної області.*

На цьому етапі здобувач освіти за допомогою викладача проводить аналіз діяльності підприємства. Інтерактивна взаємодія з викладачем на цьому етапі



спрямована на звуження меж предметної області та виділення конкретного бізнес-процесу, який підлягає автоматизації. Студент має чітко відповісти на запитання: «Хто буде користувачем цієї бази даних?» та «Які завдання вона вирішуватиме?».

*Другий виток спіралі – концептуальне та даталогічне моделювання.*

На другому етапі відбувається деталізація обраного процесу з більш повним описом його складових. Здобувач розробляє концептуальну модель (найчастіше у форматі ER-діаграм), де визначаються сутності та логічні зв'язки між ними. Тут важливо перейти від словесного опису до формальних визначень атрибутів, типів даних та первинних ключів. Викладач допомагає усунути логічні помилки, які на цьому етапі ще легко виправити без переписування коду.

*Третій виток спіралі – верифікація та нормалізація.*

Наступним етапом є уточнення складових бази даних шляхом «розгортання» таблиць із наповненням їх тестовими даними, що дозволяє виявити та скоригувати помилки в атрибутах і типах зв'язків, зокрема трансформацію зв'язків «багато-до-багатьох» у «один-до-багатьох». Паралельно виконується нормалізація логічної моделі до нормальних форм із обов'язковою перевіркою викладача.

Після усунення помилок на етапі проєктування студенти можуть самостійно розробляти базу даних на фізичному рівні, оскільки необхідні навички сформовано раніше. Реалізація здійснюється з використанням середовища розробки: студенти створюють основні об'єкти бази даних (таблиці, запити, індекси, подання, тригери тощо).

Завершальним етапом є *розробка користувацького інтерфейсу*.

Вибір інструментарію має відповідати етапам навчання та вимогам ринку. Хоча MS Access залишається популярним для початкового рівня через візуальну зрозумілість, підготовка професійного розробника вимагає опанування складніших систем (табл. 2).

Важливим компонентом є безпосередня робота з мовою SQL через командний рядок або спеціалізовані клієнтські інструменти (зокрема Oracle SQL Developer).

Таблиця 2

**Середовища розробки**

Технологія / СУБД	Цільове призначення	Переваги для здобувача
MS Access	Ознайомлення з реляційними зв'язками, формами та простими звітами.	Швидкий результат, наочність архітектури.



Технологія / СУБД	Цільове призначення	Переваги для здобувача
<b>Oracle Database XE</b>	Глибоке вивчення SQL, PL/SQL, адміністрування, тригерів та процедур.	Промисловий стандарт, висока надійність, CASE-підтримка.
<b>MySQL / MariaDB</b>	Розробка вебдодатків, робота з PHP, налаштування локальних серверів.	Відкритий код, легкість інтеграції у вебпроекти.
<b>CASE-засоби</b>	Автоматизація проектування, генерація DDL-скриптів, візуалізація моделей.	Системне бачення структури великих проєктів.

Такий підхід мінімізує залежність від візуальних засобів проектування та сприяє формуванню у студентів глибокого розуміння синтаксису мов DDL, DML, DQL та DCL.

Використання спіральної моделі дозволяє уникнути ситуації, коли студент намагається спроектувати все і відразу, що зазвичай призводить до появи непрацездатних систем.

У цьому контексті доцільно на різних рівнях складності підготовки при складанні практичних навчальних завдань виявляти та оцінювати отримані знання та вміння.

Процес розв'язання будь-якого завдання з теми «Бази даних» здійснюється у п'ять етапів:

1. Аналіз завдання – розуміння змісту завдання та визначення очікуваних результатів.
2. Вибір потрібної моделі – вибір моделі, яка забезпечить розв'язання задачі залежно від її змісту.
3. Розробка алгоритму – визначення послідовності виконання дій для розв'язання задачі.
4. Computer Expression – виконання задачі на комп'ютері та отримання результату.
5. Обґрунтування рішення – аналіз отриманих результатів і їх інтерпретація.

Ефективність навчання базам даних не може бути оцінена лише за кількістю вивчених команд. Важливо враховувати когнітивні рівні засвоєння матеріалу, які описуються таксономією Б. Блума (табл. 3).

Для об'єктивного оцінювання рекомендується використовувати багаторівневі тестові та практичні завдання.



Перший рівень контрольних завдань передбачає перевірку знань, засвоєних здобувачами через запам'ятовування, а також аналіз навчального матеріалу та інформації, отриманої в процесі навчання.

Таблиця 3

**Структурування навчальних результатів курсу  
бази даних за таксономією Б. Блума**

<b>Рівень за Б. Блумом</b>	<b>Діяльність здобувача у курсі БД</b>	<b>Приклад завдання</b>
<b>Знання</b>	Відтворення термінів, правил нормалізації, команд SQL.	Перелічити типи зв'язків у реляційній моделі.
<b>Розуміння</b>	Інтерпретація схем БД, пояснення принципів цілісності.	Описати різницю між зв'язками 1:1 та 1:М.
<b>Застосування</b>	Використання методів проектування у нових ситуаціях.	Спроектувати БД для шкільної бібліотеки.
<b>Аналіз</b>	Виокремлення частин цілого, виявлення функціональних залежностей.	Провести нормалізацію таблиці до 3НФ.
<b>Синтез</b>	Комбінування елементів для створення нової архітектури.	Розробити прототип ІС для реального підприємства.
<b>Оцінювання</b>	Порівняння моделей, обґрунтування вибору СУБД.	Оцінити переваги SQL та NoSQL підходів для проєкту.

Допоміжні контрольні завдання (другий рівень) базуються на репродуктивному рівні: здобувачі аналізують зміст навчальних матеріалів і перевіряють знання, сформовані в результаті самостійної роботи.

Третій рівень передбачає виконання завдань, спрямованих на розв'язання нестандартних і проблемних ситуацій на основі знань продуктивного рівня, що потребує самостійного мислення та прийняття рішень. При цьому рівень складності матеріалу не є надто високим, однак завдання вимагають осмисленого опрацювання змісту та логічно узгоджених висновків.

Контрольні завдання четвертого рівня мають творчий характер і спрямовані на розвиток креативного мислення, формування творчих здібностей та пошук нестандартних рішень на основі пізнавального й інноваційного підходів.



Наприклад, завдання 1-го рівня перевіряють здатність створити таблицю, тоді як 4-го рівня вимагають підготовки SQL-запитів у режимі конструктора та коду з порівнянням їхньої ефективності.

Цифровізація освіти зумовлює необхідність упровадження інноваційних підходів, які виходять за межі традиційного лекційного викладання. Ефективним у цьому контексті є застосування змішаного навчання та моделі «перевернутого класу». Під час вивчення баз даних це дає змогу перенести первинне опрацювання теоретичного матеріалу та синтаксичних конструкцій у самостійну роботу здобувачів освіти, тоді як аудиторний час використовується для обговорення та аналізу архітектурних і проєктних рішень.

Серед актуальних трендів, що можуть бути інтегровані у вивчення баз даних, виділяють:

- навчання за допомогою відео та віртуальних лабораторій: використання онлайн-середовищ для роботи з віддаленими серверами БД.
- технології штучного інтелекту: інтеграція штучного інтелекту уможливорює персоналізовані освітні досвіди й підготовку студентів до конкретних ролей у динамічному і непередбачуваному професійному середовищі. Для курсу баз даних це означає не лише використання ШІ-асистентів для написання запитів, а й формування критичної компетентності — здатності верифікувати, оптимізувати та пояснювати ШІ-генерований код.
- гейміфікація: впровадження ігрових елементів у процес вивчення SQL для підвищення залученості студентів.
- мікронавчання та персоналізація: підвищує мотивацію та залученість студентів завдяки засвоєнню матеріалу невеликими логічно завершеними порціями; воно тісно пов'язане з персоналізованим навчанням, що передбачає адаптацію змісту до індивідуального рівня знань і досвіду.

Таким чином, сучасні педагогічні тенденції формують чіткий методичний запит: від пасивного слухання – до активного виконання автентичних завдань; від уніфікованої програми – до персоналізованої траєкторії з вимірюваними результатами. Саме ці орієнтири покладено в основу розробленої методики навчання баз даних на засадах спіральної моделі проєктування.

**Висновки.** Вивчення баз даних у закладах вищої освіти на сучасному етапі потребує відходу від статичних моделей навчання на користь динамічних, інтерактивних та практико-орієнтованих підходів. Застосування спіральної моделі проєктування дозволяє студентам долати психологічний бар'єр перед складністю реальних предметних областей та поступово формувати навички системного аналізу.

Інтеграція компетентнісного підходу та таксономії Б. Блума забезпечує прозорість та об'єктивність оцінювання результатів навчання, фокусуючи увагу не на обсязі теоретичної інформації, а на здатності студента створювати



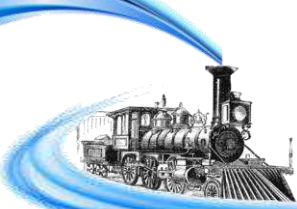
працюючі програмні продукти. Використання реальних даних підприємств як основи для навчальних проєктів є важливим фактором професійної мотивації майбутніх IT-фахівців.

Перспективи подальших досліджень у цій царині пов'язані з адаптацією навчальних програм до вимог хмарних обчислень, NoSQL архітектур та розробкою систем інтелектуального супроводу навчання, що дозволять персоналізувати освітню траєкторію кожного студента відповідно до його здібностей та професійних інтересів.

Подальше вдосконалення методик викладання сприятиме зміцненню кадрового потенціалу IT-галузі та успішній інтеграції української освіти у світовий науково-технологічний простір.

#### *Література:*

1. Date C. J. *Introduction to Database Systems*. 8th ed. Boston : Pearson, 2003. 1040 p.
2. Hamzah M. L., Rukun K., Rizal F. et al. A review of increasing teaching and learning database subjects in computer science. *Revista Espacios*. 2019. Vol. 40, № 26. P. 6.
3. Shebaro B. Using active learning strategies in teaching introductory database courses. *Journal of Computing Sciences in Colleges*. 2018. Vol. 33, № 4. P. 28–36.
4. Uzun A., Onur A., Alabay S. Students' views on database management systems course designed according to problem-based learning. *International Journal of Evaluation and Research in Education*. 2020. Vol. 9, № 1. P. 177–187. DOI: <http://doi.org/10.11591/ijere.v9i1.20501>.
5. Ahmad R., Chyi W. A., Sarlan A., Kasbon R. Guiding novice database developers in database schema creation. *IEEE Conference on e-Learning, e-Management and e-Services*. 2015. Vol. 9. P. 708–715. DOI: <https://doi.org/10.1109/IC3e.2014.7081243>.
6. Yang F.-J. A virtual tutor for relational schema normalization. *ACM Inroads*. 2011. Vol. 2, № 3. P. 38–42. DOI: <https://doi.org/10.1145/2003616.2003632>.
7. Saringat M. Z., Ibrahim R., Ibrahim N., Herawan T. On database normalization using user interface normal form. In: *Advanced Intelligent Computing Theories and Applications. ICIC 2010*. Berlin : Springer, 2010. P. 571–578. DOI: [https://doi.org/10.1007/978-3-642-14922-1\\_71](https://doi.org/10.1007/978-3-642-14922-1_71).
8. Hartmann S., Kirchberg M., Link S. Design by example for SQL table definitions with functional dependencies. *The VLDB Journal*. 2012. Vol. 21, № 1. P. 121–144. DOI: <https://doi.org/10.1007/s00778-011-0239-5>.
9. Paradis E., O'Brien B., Nimmon L., Bandiera G. Design: selection of data collection methods. *Journal of Graduate Medical Education*. 2016. Vol. 8, № 2. P. 263–264. DOI: <https://doi.org/10.4300/JGME-D-16-00098.1>.
10. Edison J. 13 blog articles on database design best practices and tips. *Vertabelo*. 2016. URL: <https://www.vertabelo.com/blog/13-blog-articles-with-database-design-tips-and-best-practices/> (accessed: 08.05.2026).
11. Djajasaputra S. Database guidelines (RDBMS/SQL). *SOA, Java, Software Development*. 2013. URL: <http://soa-java.blogspot.com/2013/01/database-guidelines-rdbmssql.html> (accessed: 08.05.2026).
12. Davidson L. Ten common database design mistakes. *RedGate Simple Talk*. 2007. URL: <https://www.red-gate.com/simple-talk/sql/database-administration/ten-common-database-design-mistakes/> (accessed: 08.05.2026).



13. Syed A. A. Physical database design techniques to improve database performance. *International Research Journal of Engineering and Technology (IRJET)*. 2018. Vol. 5, iss. 6. P. 2335–2337. URL: <https://www.irjet.net/archives/V5/i6/IRJET-V5I6435.pdf>.

14. Boehm B. W. A spiral model of software development and enhancement. *IEEE Computer*. 1988. Vol. 21, № 5. P. 61–72.

### References:

1. Date, C. J. (2003). *Introduction to database systems* (8th ed.). Pearson.
2. Hamzah, M. L., Rukun, K., Rizal, F., et al. (2019). A review of increasing teaching and learning database subjects in computer science. *Revista Espacios*, 40(26), 6.
3. Shebaro, B. (2018). Using active learning strategies in teaching introductory database courses. *Journal of Computing Sciences in Colleges*, 33(4), 28–36.
4. Uzun, A., Onur, A., & Alabay, S. (2020). Students' views on database management systems course designed according to problem-based learning. *International Journal of Evaluation and Research in Education*, 9(1), 177–187. <https://doi.org/10.11591/ijere.v9i1.20501>.
5. Ahmad, R., Chyi, W. A., Sarlan, A., & Kasbon, R. (2015). Guiding novice database developers in database schema creation. In *Proceedings of the IEEE Conference on e-Learning, e-Management and e-Services* (Vol. 9, pp. 708–715). <https://doi.org/10.1109/IC3e.2014.7081243>.
6. Yang, F.-J. (2011). A virtual tutor for relational schema normalization. *ACM Inroads*, 2(3), 38–42. <https://doi.org/10.1145/2003616.2003632>.
7. Saringat, M. Z., Ibrahim, R., Ibrahim, N., & Herawan, T. (2010). On database normalization using user interface normal form. In *Advanced Intelligent Computing Theories and Applications. ICIC 2010* (pp. 571–578). Springer. [https://doi.org/10.1007/978-3-642-14922-1\\_71](https://doi.org/10.1007/978-3-642-14922-1_71).
8. Hartmann, S., Kirchberg, M., & Link, S. (2012). Design by example for SQL table definitions with functional dependencies. *The VLDB Journal*, 21(1), 121–144. <https://doi.org/10.1007/s00778-011-0239-5>.
9. Paradis, E., O'Brien, B., Nimmon, L., & Bandiera, G. (2016). Design: Selection of data collection methods. *Journal of Graduate Medical Education*, 8(2), 263–264. <https://doi.org/10.4300/JGME-D-16-00098.1>.
10. Edison, J. (2016). 13 blog articles on database design best practices and tips. *Vertabelo*. Retrieved from <https://www.vertabelo.com/blog/13-blog-articles-with-database-design-tips-and-best-practices/>.
11. Djajasaputra, S. (2013). Database guidelines (RDBMS/SQL). *SOA, Java, Software Development*. Retrieved from <http://soa-java.blogspot.com/2013/01/database-guidelines-rdbmssql.html>.
12. Davidson, L. (2007). Ten common database design mistakes. *RedGate Simple Talk*. Retrieved from <https://www.red-gate.com/simple-talk/sql/database-administration/ten-common-database-design-mistakes/>.
13. Syed, A. A. (2018). Physical database design techniques to improve database performance. *International Research Journal of Engineering and Technology*, 5(6), 2335–2337. Retrieved from <https://www.irjet.net/archives/V5/i6/IRJET-V5I6435.pdf>.
14. Boehm, B. W. (1988). A spiral model of software development and enhancement. *IEEE Computer*, 21(5), 61–72.

Дата першого надходження статті до видання: 27.04.2026

Дата прийняття статті до друку після рецензування: 11.05.2026