



Viktor D. Boyko
Бойко Віктор
Дмитрович



Valeriia M. Slatvinska
Слатвінська
Валерія
Миколаївна



Viacheslav I. Bevza
Бевза В'ячеслав
Ігорович

УДК 004.65:004.4

STRATEGY MANAGEMENT CHANGES SCHEMES DATABASES

СТРАТЕГІЯ УПРАВЛІННЯ ЗМІНАМИ СХЕМИ БАЗИ ДАНИХ

DOI [https://doi.org/10.15589/smi2026.1\(22\).14](https://doi.org/10.15589/smi2026.1(22).14)

Viktor D. Boyko

Бойко Віктор Дмитрович, канд. техн. наук, доцент
boyko-work@ukr.net

ORCID: 0000-0001-5929-657X

Valeriia M. Slatvinska

Слатвінська Валерія Миколаївна, доктор філософії
slatvinskaya_valeriya@ukr.net

ORCID: 0000-0002-6082-981X

Viacheslav I. Bevza

Бевза В'ячеслав Ігорович, здобувач вищої освіти
viacheslavbevza718@gmail.com

ORCID: 0009-0007-2695-969X

National University «Odesa Law Academy», Odesa

Національний університет «Одеська юридична академія», м. Одеса

Abstract. The purpose of the article is to reveal the importance of database change management (DCM) as a technical discipline in the life cycle of information systems. Operating conditions are constantly changing. To substantiate that the work has theoretical and practical value in shipbuilding and marine infrastructure, namely in the optimization and convenient use of databases. The continuous development of application services continues. Data volumes are growing. Integration relationships are becoming more complicated. In modern distributed systems and microservice architectures, changes in data structure are becoming routine. These include adding attributes, denormalization, changing data types, reorganization of indexes, migration of integrity constraints.

Methodology – the research methodology is based on a systematic analysis of existing DCM practices and a generalization of approaches to the evolution of the scheme as a managed engineering process. The work uses the following methods: (1) conceptual modeling of the DCM strategy through the allocation of functional components and data control flows; (2) comparative analysis of typical classes of scheme changes and their risk profiles to determine optimal deployment scenarios; (3) construction of change management metrics (traceability, reversibility, compatibility, execution time, impact on resource locking); (4) synthesis of recommendations for organizing phased migrations and integration with continuous deployment (CI/CD) practices.

Results. A generalized change management strategy for the schema is obtained. It consists of six interconnected structural components. These include the change log and versioning. A compatibility policy is specified separately. The use of a phased migration pattern (expand–migrate–contract) is proposed as a basic approach to reduce the risk of downtime and compatibility violations.

Scientific novelty. The scientific novelty of the work lies in improving the methodology for managing database evolution. For this purpose, a comprehensive DCM strategy has been developed. It differs from existing approaches. Versioning mechanisms are integrated. Dynamic risk assessment metrics are also used. Continuous compatibility templates are included separately.

Practical significance. An application toolkit for database architects and administrators is obtained. Change processes are standardized. System downtime is reduced during updates.

Key words: schema versioning; data migration; continuous deployment; database refactoring; structure evolution; version compatibility; update automation; administration and protection of databases.

Анотація. Мета статті – розкрити значення управління змінами схеми бази даних (Database Change Management, DCM) як технічної дисципліни в життєвому циклі інформаційних систем. Умови експлуатації змінюються постійно. Обґрунтувати що робота має теоретичну та практичну цінність у суднобудівельній та морській інфраструктурі, а саме у оптимізації та зручному використанні баз даних. Триває безперервний розвиток прикладних сервісів. Зростають обсяги даних. Ускладнюються інтеграційні зв'язки. У сучасних розподілених системах та архітектурах мікросервісів зміни структури даних стають рутинними. До них відносяться додавання атрибутів, денормалізація, зміна типів даних, реорганізація індексів, міграція обмежень цілісності.

Методика – методика дослідження базується на системному аналізі існуючих практик DCM та узагальненні підходів до еволюції схеми як керованого інженерного процесу. У роботі використано методи: (1) концептуального моделювання стратегії DCM через виділення функціональних компонентів й потоків контролю даних; (2) порівняльного аналізу типових класів змін схеми та їх ризикових профілів для визначення оптимальних сценаріїв розгортання; (3) побудови метрик керованості змін (трасованість, оборотність, сумісність, час виконання, вплив на блокування ресурсів); (4) синтезу рекомендацій щодо організації поетапних міграцій та інтеграції з практиками безперервного розгортання (CI/CD).

Результати. Отримано узагальнену стратегію управління змінами схеми. Вона складається з шести взаємопов'язаних структурних складових. До них віднесено реєстр змін й версіонування. Окремо задано політику сумісності.

Наукова новизна. Наукова новизна роботи полягає в удосконаленні методології управління еволюцією баз даних. Для цього розроблено комплексну стратегію DCM. Вона відрізняється від наявних підходів. Інтегруються механізми версіонування. Також використовуються динамічні метрики оцінки ризиків. Шаблони безперервної сумісності включено окремо.

Практична значимість. Отримано прикладний інструментарій для архітекторів та адміністраторів баз даних. Процеси внесення змін стандартизуються. Під час оновлень простої систем скорочуються.

Ключові слова: версіонування схеми; міграція даних; безперервне розгортання; рефакторинг бази; еволюція структур; сумісність версій; автоматизація оновлень; адміністрування та захист баз даних.

References

- [1] Brahmia, Z., Grandi, F., & Oliboni, B. (2024). Schema versioning in databases: A literature review. *Computing Open*. <https://doi.org/10.1142/S2972370124300024>
- [2] Schuler, R. E., & Kesselman, C. (2022). Managing database-application co-evolution in a scientific data ecosystem. In 2022 IEEE 18th International Conference on e-Science (e-Science). <https://doi.org/10.1109/eScience55777.2022.00035>
- [3] Petrie, J., Loban, A., Turton, E., Derebecka, J., North, S., Herbert, E., & Hind, D. (2024). “Reality is frequently inaccurate” A case study examining the whens and whys of post-live database changes in a UK clinical trials unit *Douglas Adams. *Contemporary Clinical Trials*, 142, Article 107573. <https://doi.org/10.1016/j.cct.2024.107573>
- [4] Nastase, M., Tîrnovanu, A. C., Şişu, J. A., & Obreja, M.-M. (2024). Mapping the field of change management – A bibliometric analysis using Vosviewer. *Proceedings of the International Conference on Business Excellence*, 18(1). <https://doi.org/10.2478/picbe-2024-0269>
- [5] Ramachandran, D. (2025). Change management strategies for seamless large-scale system migrations. *Journal of Technology and Systems*, 7(2), 22–32. <https://doi.org/10.47941/jts.2635>
- [6] Tama, E. H., & Mark, D. (2024). The future of database management in the era of big data and cloud computing. *International Journal of Information Technology & Computer Engineering*, 4(5), 48–60. <https://doi.org/10.55529/ijitc.45.48.60>
- [7] Jiang, K., Du, S., Zhao, F., Huang, Y., Li, C., & Luo, Y. (2022). Effective data management strategy and RDD weight cache replacement strategy in Spark. *Computer Communications*, 193, 1–11. <https://doi.org/10.1016/j.comcom.2022.07.008>
- [8] Nallapareddy, C. (2025). Database cloud migration strategy for a business-critical application. *Journal of Technology and Systems*, 7(2), 22–32. <https://doi.org/10.47941/jts.2635>

[9] Netinant, P., Saengsuwan, N., Rukhiran, M., & Pukdesree, S. (2023). Enhancing data management strategies with a hybrid layering framework in assessing data validation and high availability sustainability. *Sustainability*, 15(20), Article 15034. <https://doi.org/10.3390/su152015034>

[10] Verma, P., & Tyagi, A. D. (2025). Agile methodologies in database management: Discussing the application of Agile Scrum and Kanban in database administration. *International Journal of Research in Management, Economics, Engineering, Education and Technology*, 13(4). <https://doi.org/10.63345/ijrmeet.org.v13.i4.13>

Постановка задачі. Сучасні інформаційні системи працюють у режимі постійних зростаючих вимог. Це неминуче тягне модифікацію структур зберігання даних. Традиційне адміністрування баз даних часто спирається на ручне застосування SQL-скриптів. Також використовуються тривалі вікна обслуговування для оновлень. Вимоги до режиму роботи 24/7 посилюються. Тривалі зупинки сервісів для реструктуризації таблиць або індексів стають неприпустимими. Помилки під час зміни схеми мають високу ціну. Можлива катастрофічна втрата даних. Також можлива тривала недоступність критичних бізнес-функцій.

Актуальність дослідження зумовлена необхідністю розробки науково обґрунтованої стратегії управління змінами схеми бази даних, яка б дозволяла поєднати швидкість впровадження інновацій із гарантіями цілісності, продуктивності та стабільності системи, забезпечуючи автоматизований та безпечний процес еволюції даних. Актуальність дослідження

Аналіз останніх досліджень та публікацій. Проблема управління еволюцією схем баз даних розглядається у багатьох сучасних наукових працях.

Брахмія З., Гранді Ф., Олібоні Б. (Brahmia Z., Grandi F., Oliboni B.) [1, с. 6] проводять ґрунтовний огляд літератури щодо версіонування схем, акцентуючи на теоретичних аспектах підтримки темпоральних даних.

Шулер Р. Е., Кессельман К. (Schuler R. E., Kesselman C.) [2, с. 2] досліджують питання спільної еволюції (co-evolution) баз даних та застосунків у наукових екосистемах, підкреслюючи складність синхронізації змін.

Петрі, Дж., Лобан, А., Тертон, Е., Дербека, Дж., Норт, С., Герберт, Е., Хайнд, Д. (Petrie, J., Loban, A., Turton, E., Derebecka, J.,

North, S., Herbert, E., Hind, D.) [3, с. 107] на прикладі клінічних випробувань аналізують причини та наслідки змін у базах даних після запуску систем у промислову експлуатацію, вказуючи на розрив між проектною реальністю та фактичними потребами.

Загальні підходи до управління змінами в ІТ-проектах досліджують Nastase M., Târnovanu A. C., Şişu JA A., Obreja M.-M. (Nastase M., Tîrnovanu A. C., Şişu J. A., Obreja M.-M.) [4, с. 360], використовуючи бібліометричний аналіз для картування поля.

Рамачандран Д. (Ramachandran D.) [5, с. 25] фокусується на стратегіях управління змінами для безшовних міграцій великомасштабних систем, що є критичним для забезпечення безперервності бізнесу.

Тама Е. Х., Марк Д. (Tama E. H., Mark D.) [6, с. 52] розглядають майбутнє управління базами даних в еру великих даних та хмарних обчислень, де традиційні методи DCM потребують перегляду.

Цзян, К., Д У, С., Чжао, Ф., Хуанг, Ю., Лі, К., Лю, Ю. (Jiang, K., Du, S., Zhao, F., Huang, Y., Li, C., & Luo, Y.) [7, с. 5] пропонують ефективні стратегії управління даними в розподілених системах типу Spark, що впливає на підходи до схеми.

Наллапаредді К. (Nallapareddy C.) [8, с. 28] деталізує стратегії міграції баз даних у хмару для критично важливих застосунків, де зміни схеми є невід'ємною частиною процесу.

Нетінант П., Саенгсуван Н., Рухіран М., Пукдесрі С. (Netinant P., Saengsuwan N., Rukhiran M., Pukdesree S.) [9, с. 8] пропонують гібридний фреймворк для покращення стратегій управління даними з фокусом на валідацію та високу доступність.

Верма П., Тягі А. Д. (Verma P., Tyagi A. D.) [10, с. 4] обговорюють застосування гнучких методологій (Agile) в адмініструванні баз даних, що вимагає нових підходів до DCM.

Відокремлення невирішених раніше частин загальної проблеми. Консолідована технічна стратегія саме для зміни схеми реляційних й гібридних баз даних описується рідко, особливо за суворих вимог до high availability. Наявні підходи зазвичай розглядають або теорію версіонування, або окремі інструменти міграції. У єдину систему ці частини часто не зводяться. Ризики блокувань при виконанні міграцій враховуються недостатньо. Стратегії відкату (rollback) й забезпечення зворотної сумісності коду застосунку під час міграції також подаються фрагментарно. Окремо залишається проблема формалізації критеріїв вибору методу міграції залежно від типу структурних змін.

Мета дослідження. Розробка та обґрунтування комплексної стратегії управління змінами схеми бази даних та зручному використанні для морської та суднобудівельної інфраструктури, яка забезпечує мінімізацію ризиків, гарантує цілісність даних та підтримує безперервність роботи інформаційних систем в умовах частих оновлень.

Методи, об'єкт та предмет дослідження. Об'єктом дослідження є процеси супроводу та розвитку життєвого циклу баз даних у корпоративних інформаційних системах. Предметом дослідження є методи, моделі та стратегії управління змінами структури (схеми) бази даних. Методологічну основу роботи складають методи системного аналізу для декомпозиції процесу управління змінами, методи моделювання бізнес-процесів для побудови схеми взаємодії компонентів стратегії, а також методи класифікації та ризик-аналізу для розробки матриці прийняття рішень щодо міграцій. У дослідженні використано підхід «infrastructure as code» (інфраструктура як код) як базову парадигму для автоматизації змін БД.

Виклад основного матеріалу. Ефективна стратегія управління змінами схеми бази даних (DCM) базується на відмові від ad-hoc змін на користь суворо регламентованого, версіюваного та автоматизованого процесу.

Як зазначають Брахмія З., Гранді Ф., Олібоні Б. (Brahmia Z., Grandi F., Oliboni B.) [1, с. 6], еволюція схеми вимагає підтрим-

ки множинності версій для забезпечення коректної роботи історичних запитів та сумісності. В основі запропонованої стратегії лежить принцип, згідно з яким схема бази даних розглядається як програмний код, що підлягає версіонуванню у системах контролю версій (VCS), проходженню через етапи CI/CD та автоматизованому тестуванню.

Шулер Р. Е., Кессельман К. (Schuler R. E., Kesselman C.) [2, с. 2] наголошують на важливості коеволюції схеми та коду застосунку. Розсинхронізація цих компонентів є головною причиною інцидентів. Для вирішення цієї проблеми стратегія передбачає обов'язкове забезпечення зворотної сумісності (backward compatibility) для будь-якої зміни.

Для пояснення взаємодії компонентів системи та послідовності етапів управління змінами на рисунку 1 зображено структурну модель процесу DCM.



Рис. 1. Структурна модель компонентів стратегії управління змінами схеми БД

Як видно з рисунка 1, основні залежності між етапами мають послідовну структуру з обов'язковими точками контролю (Quality Gates) між фазами, що дозволяє виявити потенційні проблеми до моменту впливу на продуктивне середовище.

Петрі, Дж., Лобан, А., Тертон, Е., Дербека, Дж., Норт, С., Герберт, Е., Хайнд, Д. (Petrie, J., Loban, A., Turton, E., Derebecka, J., North, S., Herbert, E., Hind, D.) [3, с. 107] вказують, що значна частина змін відбувається постфактум, тому наявність чіткого конвеєра (pipeline) є критичною для стабільності.

Згідно з дослідженнями Nastase M., Tîrnovanu A. C., Şişu JA A., Obreja M.-M. (Nastase M., Tîrnovanu A. C., Şişu J. A., Obreja M.-M.) [4, с. 360], управління змінами вимагає чіткої класифікації типів втручань. Стратегія DCM повинна диференціювати підходи до «легких» змін (наприклад, додавання нової таблиці) та «важких» операцій (наприклад, зміна типу даних у колонці з мільярдом записів).

Рамачандран Д. (Ramachandran D.) [5, с. 25] пропонує використовувати методи «синього/зеленого» розгортання або канареечні релізи для мінімізації ризиків під час міграцій великих систем.

Тама Е. Х., Марк Д. (Tama E. H., Mark D.) [6, с. 52] підкреслюють, що в епоху Big Data неможливо покладатися на повне блокування таблиць. Тому стратегія включає використання онлайн-операцій DDL (Data Definition Language), які підтримуються сучасними СУБД, або інструментів типу `pt-online-schema-change` чи `gh-ost`.

Цзян, К., Д У, С., Чжао, Ф., Хуанг, Ю., Лі, К., Лю, Ю. (Jiang, K., Du, S., Zhao, F., Huang, Y., Li, C., & Luo, Y.) [7, с. 5] додають, що ефективно управління даними вимагає врахування кешування та реплікації, які можуть бути порушені зміною схеми.

Для систематизації підходів до різних видів модифікацій у таблиці 1 наведено класифікацію змін та рекомендовані стратегії їх безпечного впровадження

сифікацію змін та рекомендовані стратегії їх імплементації.

Як видно з таблиці 1, для змін із високим та критичним рівнем ризику необхідне застосування складніших патернів, що забезпечують доступність даних під час реструктуризації. Наллапаредді К. (Nallapareddy C.) [8, с. 28] зазначає, що при міграції в хмару такі стратегії є єдиним способом забезпечити безперервність бізнес-процесів.

Важливим елементом стратегії є валідація даних та забезпечення їх якості після змін, на що вказують Нетінант П., Саєнгсуван Н., Рухіран М., Пукдесрі С. (Netinant P., Saengsuwan N., Rukhiran M., Pukdesree S.) [9, с. 8]. Автоматизовані тести повинні перевіряти не лише успішність виконання SQL-скрипта, але й логічну цілісність даних. Верма Р., Тягі А. Д. (Verma P., Tyagi A. D.) [10, с. 4] стверджують, що використання гнучких методологій вимагає інтеграції інструментів управління міграціями (наприклад, Liquibase або Flyway) безпосередньо в спринти розробки, перетворюючи DCM на рутинну частину Agile-процесу.

Верма Р., Тягі А. Д. (Verma P., Tyagi A. D.) [10, с. 4] стверджують, що використання гнучких методологій вимагає інтеграції інструментів управління міграціями (наприклад, Liquibase або Flyway) безпосередньо в спринти розробки, перетворюючи DCM на рутинну частину Agile-процесу.

Обговорення отриманих результатів. Отримана стратегія управління змінами схеми бази даних дозволяє перейти від реактивного вирішення проблем до проактивного планування еволюції даних. Важливим результатом є підтвердження необхідності тісної інтеграції процесів розробки коду та адміністрування баз даних, де схема виступає як контракт між різними компонентами системи. Запропонований підхід створює фундамент для побудови надійних висо-

Таблиця 1. Класифікація змін схеми та стратегії їх безпечного впровадження

Джерело: авторська розробка

Тип зміни схеми	Рівень ризику	Вплив на блокування	Рекомендована стратегія міграції
Додавання нової таблиці або представлення (View)	Низький	Відсутній	Пряме виконання (Direct Execution). Безпечно виконувати у будь-який час.
Додавання колонки (nullable)	Низький/Середній	Короткочасне (залежить від СУБД)	Пряме виконання з налаштуванням тайм-аутів блокувань.
Додавання індексу (великі таблиці)	Середній	Можливе блокування запису	Використання опції CONCURRENTLY (PostgreSQL) або ONLINE (Oracle/SQL Server).
Зміна типу колонки / видалення колонки	Високий	Тривале блокування, переписування таблиці	Патерн "Expand-Migrate-Contract". Створення нової колонки, подвійний запис, міграція даних, переключення читання.
Зміна Primary Key / Partitioning	Критичний	Повна недоступність таблиці	Створення нової таблиці (Shadow Table), синхронізація через тригери/CDC, атомарне перемикання.

коінтенсивних систем, здатних до швидкої адаптації під змінні вимоги.

Висновки. У роботі проведено комплексну стратегію управління змінами схеми бази даних, яка базується на принципах версіонування, автоматизації та

безпечної міграції. Запропонована класифікація змін дозволяє обирати оптимальний метод міграції в залежності від рівня ризику, що забезпечує баланс між швидкістю розробки та стабільністю функціонування системи.

Список літератури

- [1] Brahmia, Z., Grandi, F., & Oliboni, B. (2024). Schema versioning in databases: A literature review. *Computing Open*. <https://doi.org/10.1142/S2972370124300024>
- [2] Schuler, R. E., & Kesselman, C. (2022). Managing database-application co-evolution in a scientific data ecosystem. In 2022 IEEE 18th International Conference on e-Science (e-Science). <https://doi.org/10.1109/eScience55777.2022.00035>
- [3] Petrie, J., Loban, A., Turton, E., Derebecka, J., North, S., Herbert, E., & Hind, D. (2024). “Reality is frequently inaccurate” A case study examining the whens and whys of post-live database changes in a UK clinical trials unit *Douglas Adams. *Contemporary Clinical Trials*, 142, Article 107573. <https://doi.org/10.1016/j.cct.2024.107573>
- [4] Nastase, M., Tîrnovanu, A. C., Şişu, J. A., & Obreja, M.-M. (2024). Mapping the field of change management – A bibliometric analysis using Vosviewer. *Proceedings of the International Conference on Business Excellence*, 18(1). <https://doi.org/10.2478/picbe-2024-0269>
- [5] Ramachandran, D. (2025). Change management strategies for seamless large-scale system migrations. *Journal of Technology and Systems*, 7(2), 22–32. <https://doi.org/10.47941/jts.2635>
- [6] Tama, E. H., & Mark, D. (2024). The future of database management in the era of big data and cloud computing. *International Journal of Information Technology & Computer Engineering*, 4(5), 48–60. <https://doi.org/10.55529/ijitc.45.48.60>
- [7] Jiang, K., Du, S., Zhao, F., Huang, Y., Li, C., & Luo, Y. (2022). Effective data management strategy and RDD weight cache replacement strategy in Spark. *Computer Communications*, 193, 1–11. <https://doi.org/10.1016/j.comcom.2022.07.008>
- [8] Nallapareddy, C. (2025). Database cloud migration strategy for a business-critical application. *Journal of Technology and Systems*, 7(2), 22–32. <https://doi.org/10.47941/jts.2635>
- [9] Netinant, P., Saengsuwan, N., Rukhiran, M., & Pukdesree, S. (2023). Enhancing data management strategies with a hybrid layering framework in assessing data validation and high availability sustainability. *Sustainability*, 15(20), Article 15034. <https://doi.org/10.3390/su152015034>
- [10] Verma, P., & Tyagi, A. D. (2025). Agile methodologies in database management: Discussing the application of Agile Scrum and Kanban in database administration. *International Journal of Research in Management, Economics, Engineering, Education and Technology*, 13(4). <https://doi.org/10.63345/ijrmeet.org.v13.i4.13>

© Бойко В. Д., Слатвінська В. М., Бевза В. І.

Дата першого надходження статті до видання: 24.01.2026

Дата прийняття статті до друку після рецензування: 20.02.2026

Дата публікації (оприлюднення) статті: 15.05.2026



Стаття поширюється на умовах ліцензії відкритого доступу CC BY 4.0