

**ГІБРИДНИЙ ПІДХІД ДО АУДИТУ ЗМІН ДАНИХ
ОБЛІКОВИХ КАРТОК ІНФОРМАЦІЙНОЇ СИСТЕМИ «АРХІВ»**

Голойда К.О.

e-mail: kseniia.holoida@nure.ua

Науковий керівник – асистент Цвіркун О.А.

Харківський національний університет радіоелектроніки, каф. ІУС
м. Харків, Україна

Classical relational databases overwrite previous values during updates, making change history unavailable for auditing. To address this limitation, a hybrid approach is proposed that combines a relational schema as the source of the current system state with an immutable audit log inspired by the Event Sourcing pattern. This approach allows recording all changes and tracking the users who performed them while preserving efficient access to current data. Event versioning enables point-in-time reconstruction of any record state.

Інформаційна система (ІС) «Архів» працює з обліковими картками трьох рівнів: коробки, папки та документи. Оскільки ці дані є офіційними документованими записами, у багатокористувацькому середовищі необхідно фіксувати всі зміни та відомості про користувачів, які їх виконали. У традиційних реляційних системах управління базами даних (СУБД) під час оновлення запису попередні значення зазвичай не зберігаються в основній таблиці, що ускладнює контроль історії змін і потребує використання додаткових механізмів аудиту.

Для розв'язання цієї проблеми було проаналізовано три архітектурні підходи. Реляційна модель забезпечує простий доступ до актуальних даних, проте не підтримує повноцінне збереження історії змін та відстеження їх засобами самої СУБД без додаткових рішень. Саме тому були розглянуті альтернативні підходи – часові таблиці та Event Sourcing.

Підхід на основі часових таблиць (temporal tables) дає змогу автоматично зберігати попередні версії рядків на рівні СУБД, однак не забезпечує повної фіксації автора змін і має обмеження з точки зору платформної сумісності та практичного впровадження в межах обраного технологічного стеку [1]. Зокрема, у PostgreSQL версіонування потребує ручної реалізації, MySQL таблиці з версіонуванням за системним часом (system-versioned tables) не підтримує взагалі, а комерційні СУБД з повною підтримкою (MS SQL, Oracle) є фінансово невиправданими для більшості проєктів. Event Sourcing забезпечує повне збереження історії змін у вигляді послідовності подій [2], проте ускладнює виконання оперативних запитів до поточного стану даних і потребує додаткових механізмів його відновлення [3]. Це вимагає розгортання окремої інфраструктури у вигляді проєкцій та обробників подій, що суттєво підвищує складність системи. Порівняльний аналіз зазначених підходів наведено в таблиці 1.

Таблиця 1 – Порівняльний аналіз підходів

Характеристика	Реляційна модель	Temporal Tables	Event Sourcing
Історія змін	Зберігається тільки поточний стан	Зберігається історія рядків	Повна історія всіх змін
Швидкість запитів	Прямий SQL-запит	Вбудована підтримка СУБД	Відтворення стану з подій
Повнота аудиту	Поточні значення	Немає інформації про автора змін	Кожна зміна зафіксована
Відновлення стану	Неможливе	Можливе	Можливе
Складність системи	Стандартна реляційна схема	Залежить від СУБД	Потрібні проєкції

Метою дослідження є розробка гібридного підходу, у межах якого реляційна схема використовується як джерело актуального стану даних для оперативної роботи, а паралельний журнал подій використовується як засіб повного аудиту змін.

Для досягнення поставленої мети визначено такі задачі:

- аналіз вимог до аудиту облікових карток в архівних ІС;
- дослідження патернів Event Sourcing і їх застосування в задачах аудиту;
- розробка гібридної архітектури зберігання даних;
- програмна реалізація та верифікація запропонованого підходу.

У запронованій гібридній моделі кожна зміна супроводжується двома взаємопов'язаними операціями: оновленням реляційної таблиці та додаванням незмінного запису до журналу аудиту, який містить метадані операції і JSON-подання змінених полів.

За такого підходу реляційна схема відповідає на питання про поточний стан облікової картки, тоді як журнал подій дає змогу встановити, що саме, коли і ким було змінено, а також відтворити стан картки на довільний момент часу [3]. З урахуванням специфіки архівної предметної області, де зміни відбуваються відносно рідко, до журналу також доцільно включати події доступу, зокрема перегляд і експорт даних.

Список використаних джерел:

1. Kulkarni K., Michels J.-E. Temporal features in SQL: 2011. ACM SIGMOD Record. 2012. Vol. 41, no. 3. P. 34–43. DOI: <https://doi.org/10.1145/2380776.2380786>.
2. Richardson C. Microservices patterns: with examples in Java. Manning Publications Co. LLC, 2018.
3. Richards M., Ford N. Fundamentals of software architecture: an engineering approach. O'Reilly Media, 2020. 432 p.