

Лабораторна робота 5

Нормалізація бази даних

1. Аналіз поточної схеми

Ми починаємо з бази даних, створеної у попередніх лабораторних роботах. Основні таблиці:

- Publisher(publisher_id, name_and_surname, address, phone)
- Author(author_id, full_name, birth_date)
- Book(book_id, isbn, title, year_published, genre, publisher_id)
- BookAuthor(book_id, author_id)
- Copy(copy_id, book_id, condition, is_available)
- Member(member_id, full_name, email, phone, address, member_type, join_date)
- Loan(loan_id, copy_id, member_id, loan_date, due_date, return_date, fine_amount)

Проблеми / надлишковість

1. **Таблиця Loan:** містить fine_amount разом із даними про позики. Це створює **транзитивну залежність**, бо штраф залежить від позики, але зберігається в тій самій таблиці.
2. Таблиці Copy та Book в основному структуровані правильно.
3. Інші таблиці не містять повторюваних груп чи багатовартісних атрибутів.

2. Функціональні залежності (FD)

Таблиця Loan (до нормалізації)

- loan_id → copy_id, member_id, loan_date, due_date, return_date, fine_amount
- copy_id, member_id, loan_date → fine_amount (непряма залежність)

Таблиця Copy

- copy_id → book_id, condition, is_available

Таблиця Book

- book_id → isbn, title, year_published, genre, publisher_id

Таблиця Member

- member_id → full_name, email, phone, address, member_type, join_date

Таблиця Publisher

- publisher_id → name_and_surname, address, phone

Таблиця Author

- author_id → full_name, birth_date

Таблиця BookAuthor

- (book_id, author_id) – складений первинний ключ, додаткових залежностей немає

3. Перевірка нормальних форм

3.1 Таблиця Loan

До нормалізації: Loan містить fine_amount.

- **1NF:** усі атрибути атомарні.
- **2NF:** первинний ключ простий (loan_id), тому часткових залежностей немає
- **3NF:** fine_amount залежить від loan_id, але концептуально штрафи повинні зберігатися окремо. Включення їх у таблицю Loan створює **транзитивну залежність**.

3.2 Інші таблиці

- Book, Member, Copy, Author, Publisher вже відповідають 3NF.

3.3 BookAuthor

- Таблиця зв'язків з композиційним ключем; додаткових атрибутів немає, 3NF

4. Кроки нормалізації

Крок 1: Усунення повторюваних груп (1NF)

Усі таблиці вже атомарні, багатовартісних атрибутів немає.

Крок 2: Усунення часткових залежностей (2NF)

Важливо для таблиць зі складеним ключем.

- BookAuthor вже у 2NF; обидва стовпці формують ключ, інших атрибутів немає.

Крок 3: Усунення транзитивних залежностей (3NF)

- Відокремлюємо Fine від Loan:
- Оригінал: Loan(loan_id, copy_id, member_id, loan_date, due_date, return_date, fine_amount)
- Нормалізовано:
- Loan(loan_id, copy_id, member_id, loan_date, due_date, return_date)
- Fine(fine_id, loan_id, amount)

Тепер кожен неключовий атрибут залежить **безпосередньо** від первинного ключа.

5. Підсумок змін

Loan	Видалено fine_amount; зберігає лише інформацію про позики.
Fine	Нова таблиця для зберігання штрафів; loan_id → amount.
Інші таблиці	Змін не потребують, вже нормалізовані.

6. Переваги нормалізації

1. **Цілісність даних:** штрафи тепер пов'язані безпосередньо з позиками, що виключає неконсистентні значення.
2. **Менше надлишковості:** fine_amount більше не дублюється для кожної позики.
3. **Легше обслуговування:** оновлення штрафів або видалення позики не створює ризику неконсистентності.
4. **Відповідність 3NF:** кожен неключовий атрибут залежить лише від первинного ключа; відсутні часткові та транзитивні залежності.

7. Функціональні залежності після нормалізації

Loan

loan_id → copy_id, member_id, loan_date, due_date, return_date

Fine

fine_id → loan_id, amount

Інші таблиці зберігають ті ж залежності, що й раніше.