

## Методические рекомендации по лабораторной работе

Тема: Имитация аномалии «Потерянное обновление» на примере управления запасами книг в библиотеке

### Цель работы

Демонстрировать, как при недостаточно строгом уровне изоляции транзакций в базе данных возможно возникновение аномалии «потерянное обновление». Показать на практике, как параллельные операции могут влиять на корректность данных, и какие методы существуют для предотвращения этой аномалии.

### Предметная область

Рассматривается система учёта количества книг в филиалах библиотеки. Каждая книга может находиться в одном или нескольких филиалах одновременно, и в каждом из них хранится определённое количество экземпляров. В системе реализована возможность перемещения книг из одного филиала в другой — такая операция уменьшает количество книг в филиале-отправителе и увеличивает их в филиале-получателе.

В лабораторной работе студенты моделируют ситуацию, при которой два параллельных процесса одновременно пытаются переместить книги из одного и того же филиала. Это может привести к ошибочному уменьшению количества книг, если обновления "перетрут" друг друга — типичная ситуация, иллюстрирующая аномалию потерянного обновления.

### Задачи лабораторной работы

#### 1. Проектирование и создание структуры базы данных

В начале работы необходимо создать следующие сущности:

- Таблица `books` содержит список книг, каждая строка описывает одну книгу, включая её уникальный `id` и `name`.
- Таблица `branches` представляет собой список филиалов библиотеки. Каждый филиал имеет уникальный `id` и `name`.
- Таблица `stock` предназначена для учёта количества экземпляров каждой книги в каждом филиале. Связь между книгой и филиалом задаётся через составной первичный ключ, состоящий из `book_id` и `branch_id`. Поле `quantity` указывает, сколько экземпляров книги доступно в конкретном филиале.
- Таблица `movements` служит для логирования перемещений книг между филиалами. Каждая операция фиксируется в виде отдельной записи, где указывается `book_id`, количество (`quantity`),

from\_branch\_id, to\_branch\_id, а также время выполнения операции (movement\_time).

## **2. Реализация функции перемещения книг:**

Необходимо реализовать функцию (или скрипт), осуществляющую перемещение книг между двумя филиалами. Эта функция должна:

- Принимать параметры: идентификатор книги, идентификаторы филиала-отправителя и филиала-получателя, количество перемещаемых экземпляров, а также уровень изоляции транзакции.
- Перемещение должно происходить в рамках одной транзакции: сначала уменьшение остатка у отправителя, затем — увеличение у получателя.
- Фиксировать операцию перемещения в таблице movements.

## **3. Имитация параллельных операций:**

Необходимо создать два параллельных потока или процесса, которые одновременно инициируют перемещение одного и того же количества книг из одного и того же филиала. Для этого можно воспользоваться средствами многопоточности в Python (threading). Это создаёт ситуацию конкурентного изменения одних и тех же строк в таблице stock, и позволяет наблюдать:

- Успешную потерю обновления при недостаточном уровне изоляции (если СУБД это позволяет).
- Конфликт транзакций и ошибки сериализации (например, could not serialize access due to concurrent update) при более строгих уровнях (REPEATABLE READ, SERIALIZABLE).