

## **ЛАБОРАТОРНАЯ РАБОТА № 5**

**Тема: Использование транзакций в базах данных.**

### **Цели и задачи**

Цель лабораторной работы состоит в изучении особенностей работы с транзакциями в реляционных системах управления базами данных (СУБД) и получении практических навыков их эффективного применения.

#### **Задачи:**

Ознакомиться с основными понятиями и характеристиками транзакций (ACID).

Научиться создавать и контролировать транзакции средствами SQL.

Решить практические задачи, демонстрируя применение транзакций в собственной учебной базе данных.

### **Теоретические сведения**

Транзакция — это группа операций над базой данных, выполняемых как единое неделимое действие. Она характеризуется четырьмя ключевыми свойствами (ACID):

Атомарность (Atomicity): все шаги транзакции выполняются полностью или вообще не применяются.

Последовательность (Consistency): любое промежуточное состояние базы данных остается скрытым от внешних пользователей.

Изоляция (Isolation): независимо протекающие транзакции не мешают друг другу.

Долговечность (Durability): результат успешной транзакции сохраняется даже после перезагрузки сервера.

#### **Основные команды для работы с транзакциями:**

BEGIN TRANSACTION; -- старт транзакции

COMMIT; -- сохранение изменений

ROLLBACK; -- отказ от изменений

SAVEPOINT sp\_name; -- создание контрольной точки

RELEASE SAVEPOINT -- удаление точки сохранения

ROLLBACK TO sp\_name; -- возврат к контрольной точке

## **Практическая часть**

Задания ориентированы на самостоятельную разработку студентами сценариев работы с транзакциями на примере своей персональной базы данных (например, телефонная книга, система учета книг, служба заказов билетов и т.п.). Студент реализует сценарии, иллюстрирующих базовые техники работы с транзакциями.

### **1. Управление изменениями в связных объектах.**

Создайте сценарий, объединяющий операции, затрагивающие сразу несколько таблиц. Например, процедура регистрации нового клиента одновременно создает новый аккаунт, формирует заказ и добавляет бонусные баллы. Сделайте это единой транзакцией, гарантирующей целостность данных.

### **2. Работа с точками восстановления.**

Используйте инструкцию `SAVEPOINT` для защиты данных от возможных сбоев.

Продемонстрируйте ситуацию, когда некоторые операции были ошибочными, но остальные выполнились успешно, и вам потребуется откатить изменения до определенного шага.

### **3. Отмена неудачных операций.**

Разработайте сценарий, который автоматически отменит изменения, если какое-либо условие нарушается. Например, передача денег между двумя пользователями возможна лишь тогда, когда сумма на счету отправителя достаточно велика. Если средств недостаточно, вся транзакция должна быть отменена.

### **4. Проверка условий перед операцией.**

Создайте скрипт, проверяющий условия до запуска основной операции. Например, прежде чем изменить цену на продукт, убедитесь, что продукт действительно существует в каталоге.

## **Контрольные вопросы**

Назовите четыре фундаментальных свойства транзакций (ACID) и раскройте смысл каждого из них.

Зачем используются контрольные точки (`SAVEPOINT`) и какова их роль в управлении транзакциями?

Опишите различия между операторами `COMMIT` и `ROLLBACK`. Когда каждый из них целесообразно применять?

Какие проблемы могут возникать при параллельном выполнении транзакций и каковы способы их предотвращения?