**Лабораторная работа № 7**

«Использование транзакций»

Перед выполнением этой работы рекомендуется прочитать раздел 5 «Использование хранимых функций и процедур» и раздел 6 «Использование транзакций» [книги](https://svyatoslav.biz/database_book/) и посмотреть все видеоролики из раздела 5 «Использование хранимых функций и процедур» и раздела 6 «Использование транзакций» в [Moodle](https://lms2.bsuir.by/course/view.php?id=2807).

**Решите любые пять задач.**

Напишите SQL-код для решения следующих задач:

1. Создать хранимую процедуру, которая:
   1. добавляет каждой книге два случайных жанра;
   2. отменяет совершённые действия, если в процессе работы хотя бы одна операция вставки завершилась ошибкой в силу дублирования значения первичного ключа таблицы «m2m\_books\_genres» (т.е. у такой книги уже был такой жанр).
2. Создать хранимую процедуру, которая:
   1. увеличивает значение поля «b\_quantity» для всех книг в два раза;
   2. отменяет совершённое действие, если по итогу выполнения операции среднее количество экземпляров книг превысит значение 50.
3. Написать запросы, которые, будучи выполненными параллельно, обеспечивали бы следующий эффект:
   1. первый запрос должен считать количество выданных на руки и возвращённых в библиотеку книг и не зависеть от запросов на обновление таблицы «subscriptions» (не ждать их завершения);
   2. второй запрос должен инвертировать значения поля «sb\_is\_active» таблицы subscriptions с «Y» на «N» и наоборот и не зависеть от первого запроса (не ждать его завершения).
4. Написать запросы, которые, будучи выполненными параллельно, обеспечивали бы следующий эффект:
   1. первый запрос должен считать количество выданных на руки и возвращённых в библиотеку книг;
   2. второй запрос должен инвертировать значения поля «sb\_is\_active» таблицы «subscriptions» с «Y» на «N» и наоборот для читателей с нечётными идентификаторами, после чего делать паузу в десять секунд и отменять данное изменение (отменять транзакцию).

Исследовать поведение СУБД при выполнении первого запроса до, во время и после завершения выполнения второго запроса, повторив этот эксперимент для всех поддерживаемых СУБД уровней изолированности транзакций.

1. Написать код, в котором запрос, инвертирующий значения поля «sb\_is\_active» таблицы «subscriptions» с «Y» на «N» и наоборот, будет иметь максимальные шансы на успешное завершение в случае возникновения ситуации взаимной блокировки с другими транзакциями.
2. Создать на таблице «subscriptions» триггер, определяющий уровень изолированности транзакции, в котором сейчас проходит операция обновления, и отменяющий операцию, если уровень изолированности транзакции отличен от REPEATABLE READ.
3. Создать хранимую функцию, порождающую исключительную ситуацию в случае, если выполняются оба условия (подсказка: эта задача имеет решение только для MS SQL Server):
   1. режим автоподтверждения транзакций выключен;
   2. функция запущена из вложенной транзакции.
4. Создать хранимую процедуру, выполняющую подсчёт количества записей в указанной таблице таким образом, чтобы она возвращала максимально корректные данные, даже если для достижения этого результата придётся пожертвовать производительностью.

Форма отчётности: файл с SQL-запросами, в котором в виде комментариев перед запросом указан текст соответствующего задания. Для пункта 4 результаты исследования описать в комментариях к написанным запросам.