

# **Задания к работе №2 по базам данных и программированию.**

Все задания выполняются на языке программирования SQL для СУБД postgresql версии 13 и выше.

Каждое задание должно быть выполнено в виде SQL-скрипта, выполнение которого приводит к решению поставленной задачи. Не допускается выполнение заданий посредством средств IDE.

Выполнение заданий подразумевает наличие отношений, реализованных в виде таблиц в базе данных t01\_library на основе заданий к работе №1.

Реализация сценария подразумевает написание скрипта, содержащего один и более запросов уровня DML и удовлетворяющего описанным бизнес-требованиям.

1. В базе данных t01\_library реализовать сценарии добавления/удаления/редактирования авторов. Результатом запроса на добавление должны являться сгенерированные системой значения атрибутов сущности, входящие в ограничение первичного ключа. Ключевым параметром запросов на удаление/редактирование должны являться значения атрибутов сущности, входящие в ограничение первичного ключа.
2. Аналогично заданию 1, реализовать сценарии добавления/удаления/редактирования издательств.
3. Аналогично заданию 1, реализовать сценарии добавления/удаления/редактирования информации о книге.
4. Аналогично заданию 1, реализовать сценарии добавления/удаления/редактирования информации о читателе.
5. Аналогично заданию 1, реализовать сценарии добавления/удаления/редактирования информации об экземпляре книги (таблица public.book\_instance).
6. Реализовать сценарий выдачи экземпляра книги заданному пользователю.
7. Реализовать сценарий возврата ранее выданного заданному пользователю экземпляра книги.
8. Реализовать представление, результатом выполнения которого является информация о всех выданных читателям экземплярах книг: ФИО пользователя, автор книги, название книги, состояние экземпляра, дата выдачи.
9. Реализовать представление, результатом выполнения которого является информация обо всех просроченных выданных (не возвращённых) на текущий момент времени экземплярах книг: ФИО пользователя, автор книги, название книги, время просрочки.

10. Модифицировать сценарий выдачи экземпляра книги заданному пользователю: выдача запрещается при наличии у пользователя хотя бы одной просрочки (используйте представление из задания 9).
11. Реализовать сценарий бронирования экземпляра книги для заданного читателя.  
*Бронирование выполняется на 3 календарных дня, включительно день бронирования.*
12. Реализовать сценарий отмены забронированного экземпляра книги для заданного читателя.
13. Модифицировать сценарий выдачи экземпляра книги заданному читателю: выдача конкретного экземпляра запрещается, если на текущий момент времени этот экземпляр забронирован за другим читателем.
14. Реализовать функционал (в виде хранимой процедуры или функции), результатом выполнения которого является информация обо всех местоположениях заданной книги, с сортировкой по состоянию (от лучшего к худшему).
15. Реализовать представление, результатом выполнения которого является информация обо всех не выданных и не забронированных экземплярах книг, сгруппированных по информации о книге и состоянию экземпляра с указанием количества экземпляров.
16. Реализовать представление, результатом выполнения которого является информация обо всех не возвращённых книгах, с момента выдачи которых прошло более одного календарного года.
17. В базе данных `t01_library` создайте таблицу `logs` уровня схемы `public`, структура которой репрезентирует лог: `id` лога (PK), дата и время лога, контекст логгирования (название таблицы), содержимое лога (строка).
18. Реализуйте автоматическое заполнение таблицы логов при выполнении запросов уровня DML (кроме `SELECT`) к таблицам уровня базы данных `t01_library` с указанием в содержимом лога характера внесённых изменений в данные. Для автоматизации заполнения используйте механизм триггеров.