Министерство образования и науки Российской Федерации Российский государственный университет нефти и газа (национальный исследовательский университет)

имени И.М. Губкина

Факультет Автоматики и вычислительной техники Кафедра Автоматизированных систем управления

Отчёт по домашнему заданию №1

«ПРОДВИНУТАЯ РАБОТА С SQL, РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ РАБОТЫ С СУБД»

по дисциплине *Базы данных*

Выполнил: студент группы АС-23-04

Ханевский Ярослав

Проверили:

доцент кафедры АСУ, к.т.н. Волков Д.А.

ст. преп. кафедры АСУ Мухина А.Г.

Москва, 2025 г.

Ход работы:

1. Для своих таблиц выполнить несколько запросов на группировку данных, используя GROUP\_CONCAT(в Postgres STRING\_AGG) с разными разделителями.

Таблицы, с которыми мы будем работать:

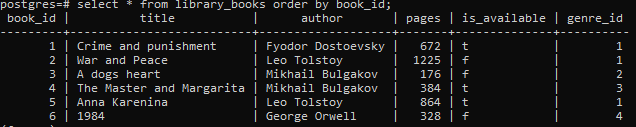


Рисунок 1. Таблица с книгами

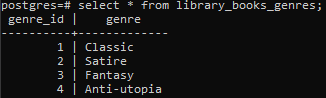


Рисунок 2. Таблица с жанрами книг

Используя агрегирующую функцию STRING\_AGG, сгруппируем книги по авторам с объединением названий книг через точку с запятой:



Рисунок 3. Использование STRING\_AGG

В результате получим следующую таблицу:

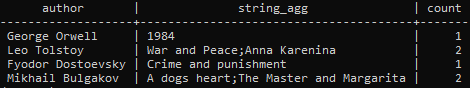


Рисунок 4. Результат вывода с STRING\_AGG

Теперь сгруппируем книги по id жанров, применив INNER JOIN для того, чтобы взять книги из таблицы library\_books и вывести их с жанрами из таблицы library\_books\_genres. Также изменим разделитель на '' -- '' и отсортируем вывод по возрастанию id жанров:

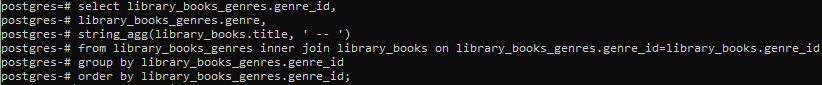


Рисунок 5. Группировка и смена разделителя

В этом случае таблица будет выглядеть следующим образом:

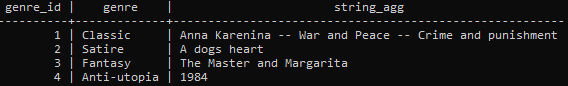


Рисунок 6. Отсортированная по жанрам таблица с разделителем '' -- ''

1. Для своих таблиц выполнить несколько запросов, содержащих подзапросы в разных операторах (WHERE, SELECT…).

Выполним запрос, содержащий подзапрос для поиска книг определенного жанра:

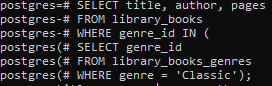


Рисунок 7. Запрос с подзапросом для поиска книг определенного жанра

Получим результат:

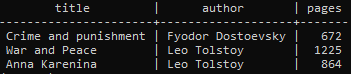


Рисунок 8. Результат запроса с подзапросом (1)

Следующий запрос, выводящий статистику по жанрам:

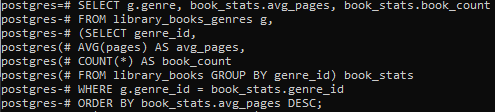


Рисунок 9. Запрос с подзапросом, выводящий нужные данные

Результат данного запроса:

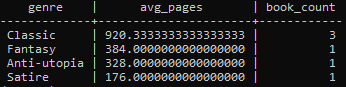


Рисунок 10. Результат запроса с подзапросом (2)

Запрос с подзапросом, отображающий самый популярный жанр и книги этого жанра через разделитель:

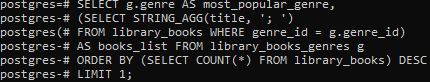


Рисунок 11. Запрос, отображающий самый популярный жанр и книги этого жанра

Получаем следующее:



Рисунок 12. Результат запроса с подзапросом (3)

1. Создать несколько транзакций, изменить данные, часть завершить успешно, часть откатить.

Создадим транзакцию с произвольными действиями и завершим её:

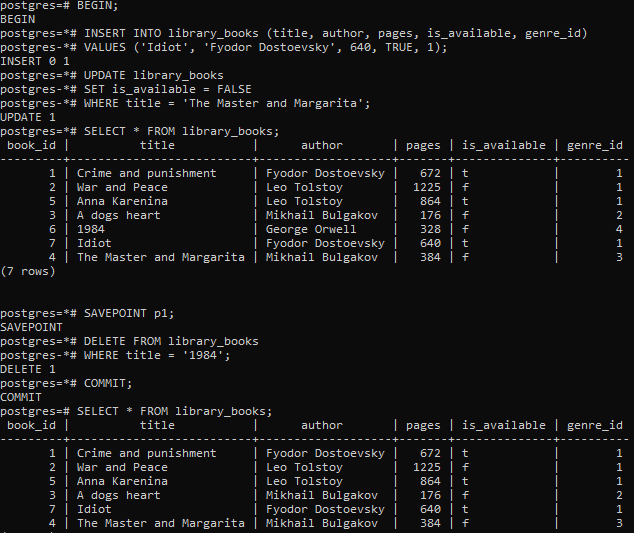


Рисунок 13. Успешная транзакция

Транзакция завершена успешно, данные обновлены.

Осуществим следующую транзакцию с внедрением точек сохранения SAVEPOINT и отката к ним ROLLBACK TO и ROLLBACK:

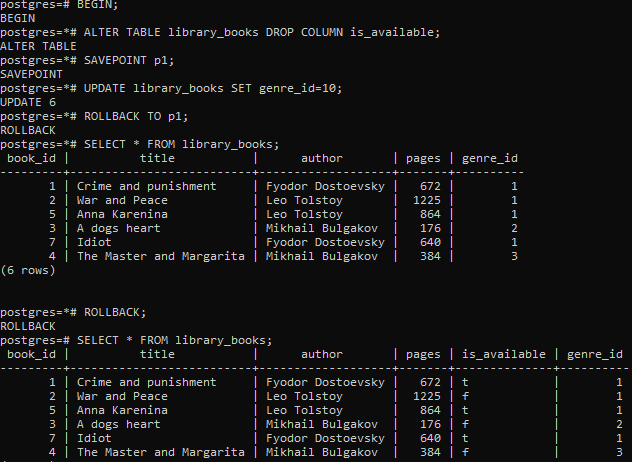


Рисунок 14. Транзакция с точками сохранения и откатом к ним

В этой транзакции сначала происходит обновление данных по заданному требованию, а затем таблица приводится к первоначальному виду с помощью отката к точке сохранения.

1. Создать несколько функций, триггеров.

В PostgreSQL процедуры – это набор SQL-инструкций, которые объединены вместе для выполнения определенной задачи или операции. Они могут принимать параметры в качестве входных данных и возвращать значения в качестве выходных данных.

Создадим процедуру с помощью CREATE PROCEDURE, предназначенную для добавления новых книг в библиотечную систему с автоматической проверкой и созданием жанров, если они не существуют:

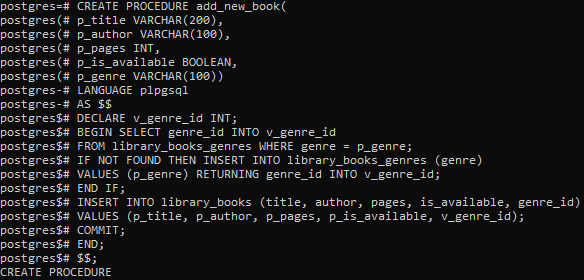


Рисунок 15. Создание процедуры

Вызовем процедуру через CALL:

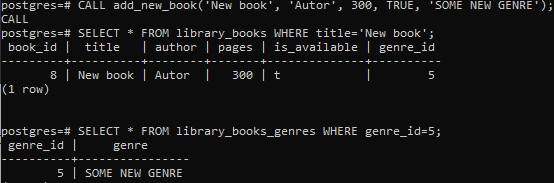


Рисунок 16. Результат работы процедуры

Триггеры в PostgreSQL представляют собой специальные хранимые процедуры, которые выполняются автоматически при определенных событиях, таких как вставка, обновление или удаление данных в таблице. Они позволяют реагировать на изменения данных в реальном времени и выполнять определенные действия в ответ на эти изменения. Триггеры часто используются для обеспечения целостности данных, аудита изменений, автоматизации сложных бизнес-логик и соблюдения бизнес-правил.

Создадим триггер, обновляющий время последнего редактирования записи о книге:

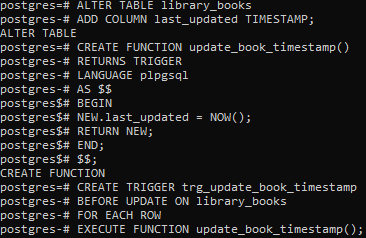


Рисунок 17. Создание триггера

Проведем проверку, обновим некоторые данные:

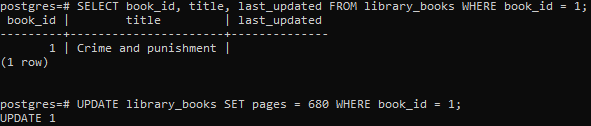


Рисунок 18. Обновление данных в таблице



Рисунок 19. Результат работы триггера

1. Выполнить несколько запросов с оконными функциями.

Оконные функции позволяют добавить в строку значения аггрегирующих функций для группы строк в контексте окна.

Запрос, осуществляющий вывод всех строк таблицы с суммарным числом страниц в книгах:

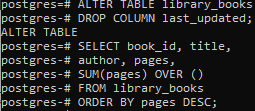


Рисунок 20. Запрос с оконной функцией SUM

Получим следующий результат:

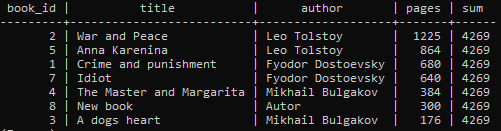


Рисунок 21. Результат работы запроса с оконной функцией SUM

Функция ранжирования RANK() назначает одинаковый ранг всем строкам с одинаковыми значениями.

Реализуем запрос с оконной агрегатной функцией RANK(), ранжирующий книги по убыванию количества страниц:



Рисунок 22. Запрос с оконной функцией RANK

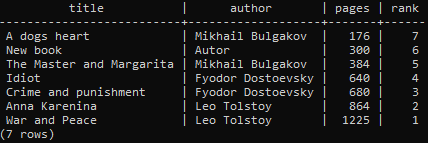


Рисунок 23. Результат работы запроса с оконной функцией RANK

Напишем запрос с функциями LAG() и LEAD:

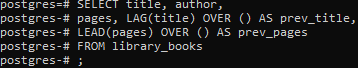


Рисунок 24. Запрос с оконными функциями LAG и LEAD

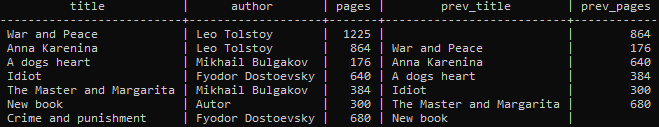


Рисунок 25. Результат работы запроса

1. Реализовать приложение.

CRUD — это аббревиатура, обозначающая четыре основных операции управления данными: создание, чтение, обновление и удаление.

Для создания программы используем python, PyQT и psycopg2.

Скриншоты программы:

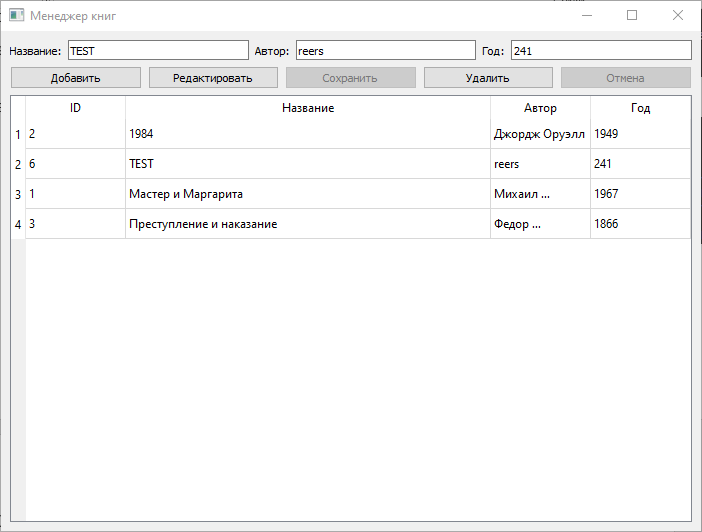


Рисунок 26. Окно программы

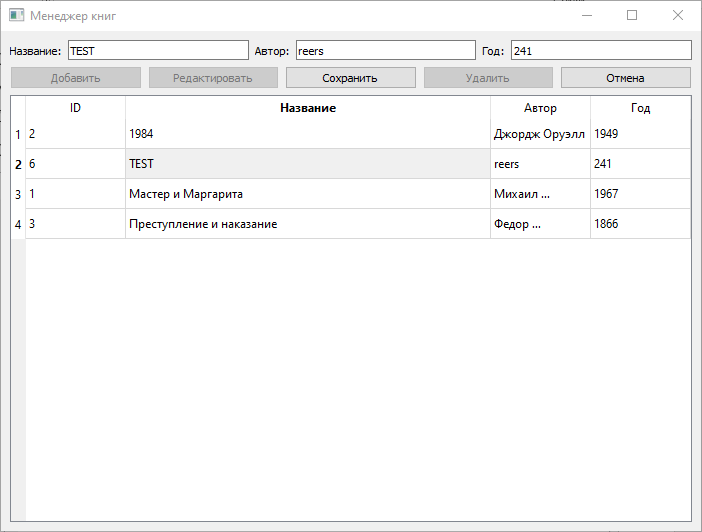


Рисунок 27. Редактирование строки

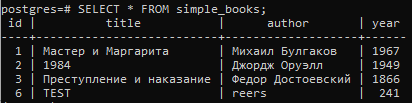


Рисунок 28. Вывод БД в консоли