МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования «Самарский национальный исследовательский университет имени академика С.П. Королева (Самарский университет)»

Институт информатики и кибернетики

Кафедра программных систем

**ОТЧЕТ**  
  
 по лабораторному практикуму по дисциплине

«Распределённые БД»

Лабораторная работа № 3

Обучающийся группы 6131-020402D В.Д. Гижевская

Обучающийся группы 6131-020402D Д.А. Кремнёв

Обучающийся группы 6131-020402D М.Г. Алкеев

Проверил М.В. Додонов

Самара 2023

ЗАдание

Тема работы: создание представлений. Использование команд SQL для работы с данными, которые доступны через представления.

Цель работы: знакомство с возможностями СУБД по созданию представлений и запросов на основе представлений.

Подготовка к работе:

1. Изучите теоретические основы, команды создания представлений, рассмотренные в лекционном курсе, методических указаниях или других источниках.

2. Ответьте на вопросы к лабораторной работе.

3. Подготовьте отчет к оформлению хода выполнения работы.

Порядок выполнения работы:

1. Создайте представления на основе запросов из лабораторной работы No 1.

2. Проверьте работоспособность созданных представлений командой SQL Select.

3. Создайте обновляемые представления для всех таблиц.

4. Проверьте работоспособность созданных представлений командами SQL: Select,

Insert, Update и Delete.

5. Добавьте в таблицы поля NameUser, в которых будет храниться имя пользователя, создавшего запись.

6. Создайте представления с детальным контролем доступа (where NameUser=USER).

7. Приведите примеры представлений с выборкой, сортировкой, группировкой, левым,

правым и внешним объединением.

8. Оформите отчет о выполнении лабораторной работы.

вариант

Предметная область: Книжный сайт.

Основные предметно-значимые сущности: Серии книг, Авторы, Книги, Главы.

Основные предметно-значимые атрибуты сущностей:

* серии книг – название серии книг;
* авторы – ФИО автора;
* книги – название книги, описание книги, ключевые слова;
* главы – название главы, время чтения, количество страниц.

Основные требования к функциям системы:

* выбрать все книги по определённому автору;
* выбрать все главы определённой книги;
* вывести рейтинг книг по времени чтения;
* вывести все книги определённой серии;
* вывести все книги по определённому ключевому слову;
* подсчитать количество страниц нужной книги;
* подсчитать количество страниц нужной серии;
* подсчитать время чтения нужной книги;
* подсчитать время чтения выбранной серии.

1. Создайте представления на основе запросов из лабораторной работы №2

**Представление jk\_rowling\_books, которое будет содержать имена книг и автора с именем “Джоан К.Роулинг”:**

CREATE VIEW jk\_rowling\_books AS

SELECT Books.book\_name, Authors.author\_name

FROM Books

JOIN Book\_series ON Book\_series.id\_book\_series = Books.id\_book\_series

JOIN Authors ON Authors.id\_author = Book\_series.id\_author

WHERE author\_name = 'Джоан К.Роулинг';

**Представление book\_reading\_times, которое будет содержать названия книг и общее время чтения для каждой книги**:

CREATE VIEW book\_reading\_times AS

SELECT book\_name, sum(reading\_time)

FROM Books

JOIN Chapters ON Chapters.id\_book = Books.id\_book

GROUP BY book\_name

ORDER BY sum(reading\_time);

**Представление harry\_potter\_chapters, которое будет содержать названия глав и названия книги "Гарри Поттер и философский камень":**

CREATE VIEW harry\_potter\_chapters AS

SELECT Books.book\_name, Chapters.chapter\_name

FROM Books

JOIN Chapters ON Chapters.id\_book = Books.id\_book

WHERE Books.book\_name = 'Гарри Поттер и философский камень'

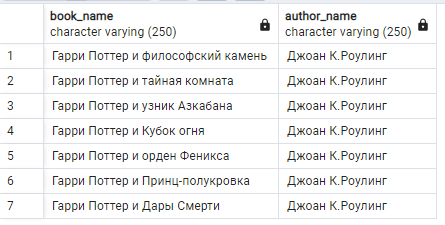
ORDER BY chapter\_number;

1. Проверьте работоспособность созданных представлений командой SQL Select.

**Для представления jk\_rowling\_books:**

SELECT \*

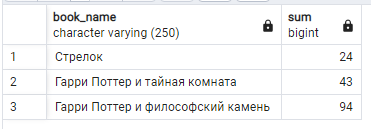
FROM jk\_rowling\_books;



**Для представления book\_reading\_times:**

SELECT \*

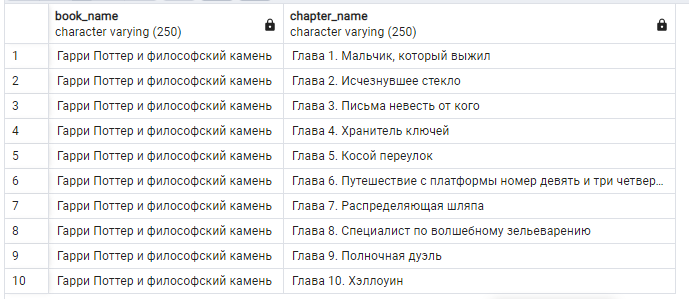
FROM book\_reading\_times;



**Для представления harry\_potter\_chapters:**

SELECT \*

FROM harry\_potter\_chapters;



1. Создайте обновляемые представления для всех таблиц

**Создание представления для таблицы Authors:**

CREATE OR REPLACE VIEW authors\_view AS

SELECT \*

FROM Authors;

**Создание представления для таблицы Book\_series**:

CREATE OR REPLACE VIEW book\_series\_view AS

SELECT \*

FROM Book\_series;

**Создание представления для таблицы Books:**

CREATE OR REPLACE VIEW books\_view AS

SELECT \*

FROM Books;

**Создание представления для таблицы Chapters:**

CREATE OR REPLACE VIEW chapters\_view AS

SELECT \*

FROM Chapters;

1. Проверьте работоспособность созданных представлений командами SQL: Select, Insert, Update и Delete.

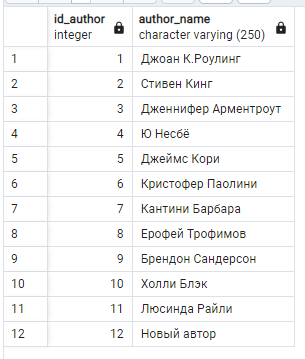
**Для представления authors\_view:**

INSERT INTO authors\_view (author\_name)

VALUES ('Новый автор');

SELECT id\_author, author\_name

FROM authors\_view;



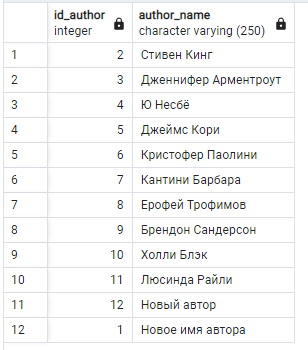
UPDATE authors\_view

SET author\_name = 'Новое имя автора'

WHERE id\_author = 1;

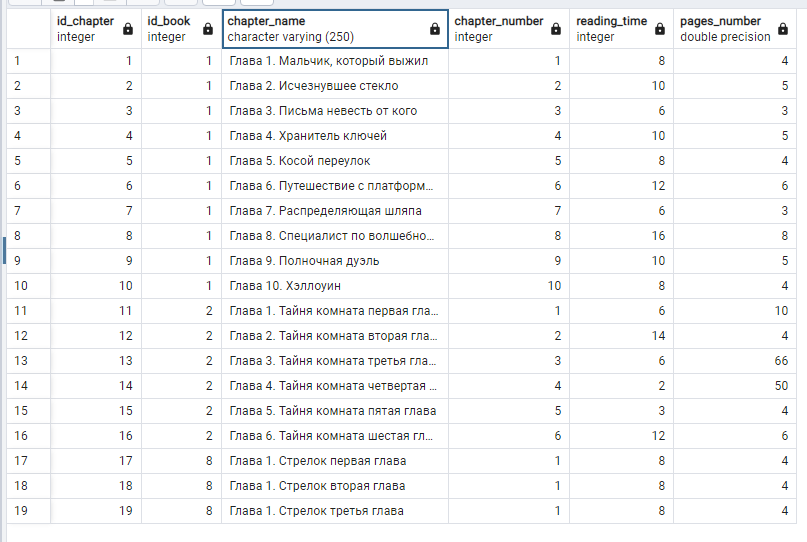
SELECT id\_author, author\_name

FROM authors\_view;



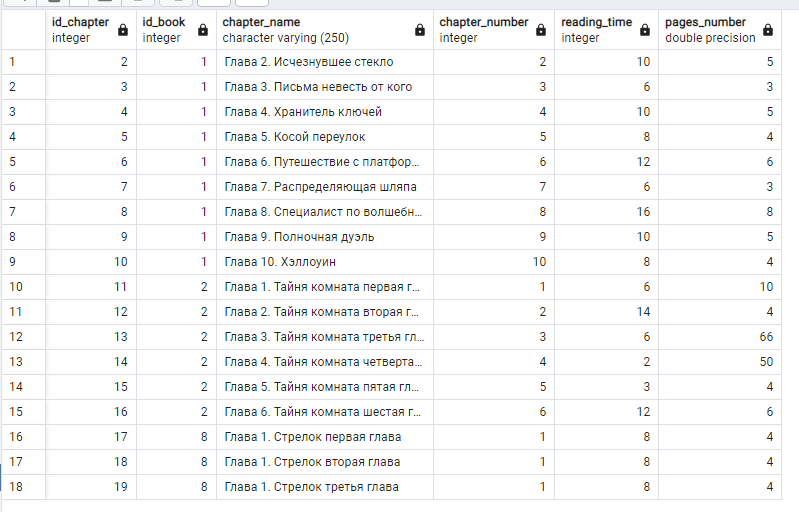
**Для представления chapters\_view:**

SELECT \* FROM chapters\_view;



DELETE FROM chapters\_view

WHERE chapter\_name = 'Глава 1. Мальчик, который выжил';



1. Добавьте в таблицы поля NameUser, в которых будет храниться имя пользователя, создавшего запись.

**Для таблицы authors:**

ALTER TABLE IF EXISTS authors

ADD COLUMN "NameUser" character varying(50) DEFAULT USER;

**Для таблицы book\_series:**

ALTER TABLE IF EXISTS book\_series

ADD COLUMN "NameUser" character varying(50) DEFAULT USER;

**Для таблицы books:**

ALTER TABLE IF EXISTS books

ADD COLUMN "NameUser" character varying(50) DEFAULT USER;

**Для таблицы chapters:**

ALTER TABLE IF EXISTS chapters

ADD COLUMN "NameUser" character varying(50) DEFAULT USER;

1. Создайте представления с детальным контролем доступа (where NameUser=USER).

**Для таблицы authors:**

CREATE VIEW authors\_with\_access\_control

AS

SELECT \*

FROM authors

where "NameUser" = USER;

**Для таблицы book\_series:**

CREATE VIEW book\_series\_with\_access\_control

AS

SELECT \*

FROM book\_series

where "NameUser" = USER;

**Для таблицы books:**

CREATE VIEW books\_with\_access\_control

AS

SELECT \*

FROM books

where "NameUser" = USER;

**Для таблицы chapters:**

CREATE VIEW chapters\_with\_access\_control

AS

SELECT \*

FROM chapters

where "NameUser" = USER;

1. Приведите примеры представлений с выборкой, сортировкой, группировкой, левым, правым и внешним объединением

**Представление с выборкой:**

CREATE VIEW books\_view AS

SELECT \* FROM books

WHERE book\_name = 'Гарри Поттер';

**Представление с сортировкой:**

CREATE VIEW books\_view AS

SELECT \* FROM books

ORDER BY book\_name;

**Представление с группировкой:**

CREATE VIEW author\_book\_counts AS

SELECT id\_author, COUNT(\*) AS book\_count

FROM Book\_series

GROUP BY id\_author;

**Представление с левым объединением:**

CREATE VIEW authors\_with\_books AS

SELECT a.author\_name, COUNT(b.id\_book) AS book\_count

FROM Authors a

LEFT JOIN Book\_series bs ON a.id\_author = bs.id\_author

LEFT JOIN Books b ON bs.id\_book\_series = b.id\_book\_series

GROUP BY a.author\_name;

**Представление с правым объединением:**

CREATE VIEW books\_with\_chapters AS

SELECT b.book\_name, COUNT(c.id\_chapter) AS chapter\_count

FROM Books b

RIGHT JOIN Chapters c ON b.id\_book = c.id\_book

GROUP BY b.book\_name;

**Представление с внешним объединение:**

CREATE VIEW authors\_with\_theirs\_books AS

SELECT a.author\_name, COUNT(DISTINCT bs.id\_book\_series) AS book\_series\_count, COUNT(DISTINCT b.id\_book) AS book\_count

FROM Authors a

LEFT JOIN Book\_series bs ON a.id\_author = bs.id\_author

LEFT JOIN Books b ON bs.id\_book\_series = b.id\_book\_series

GROUP BY a.author\_name;