

Министерство науки и высшего образования РФ
ФГБОУ ВО «Ульяновский государственный университет»
Факультет математики, информационных и авиационных технологий

Петровичева Ю.В.

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ
ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ ПО
ДИСЦИПЛИНЕ «СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ»**

Для студентов специалитета по специальности 10.05.03, 10.05.01 очной
формы обучения

Ульяновск, 2022

Методические указания для самостоятельной работы студентов по дисциплине «Системы управления базами данных» / составитель: Ю.В. Петровичева. - Ульяновск: УлГУ, 2022. Настоящие методические указания предназначены для студентов специалитета по специальности 10.05.03, 10.05.03 очной формы обучения. В работе приведены литература по дисциплине, основные темы курса и вопросы в рамках каждой темы, задания для лабораторных работ. Студентам очной формы обучения они будут полезны при подготовке к лекциям, лабораторным работам и к зачёту по данной дисциплине.

Методические указания рекомендованы к введению в образовательный процесс Ученым советом факультета математики, информационных и авиационных технологий УлГУ (протокол №3/22 от 19.04.2022 г.).

Содержание

1. Литература для изучения дисциплины.....	4
2. Методические указания.....	6
2.1. Раздел 1. Структурирование баз данных.....	6
2.1.1. Тема 1. Оптимизация структуры базы данных.....	6
2.1.2. Тема 2. Распределение баз данных в сети.....	6
2.2. Раздел 2. Ведение баз данных	7
2.2.1. Тема 3. Проектирование баз данных.....	7
2.2.2. Тема 4. Внедрение и эксплуатация баз данных.....	8
2.3. Раздел 3. Проектирование баз данных.....	11
2.3.1. Тема 5. Постановка задачи на проектирование.....	12
2.3.2. Тема 6. Разработка проекта базы данных.....	12

1. ЛИТЕРАТУРА ДЛЯ ИЗУЧЕНИЯ ДИСЦИПЛИНЫ

1. Стружкин, Н.П. Базы данных: проектирование: учебник для вузов / Н.П. Стружкин, В.В. Годин. — Москва: Издательство Юрайт, 2021. — 477 с. — (Высшее образование). — ISBN 978-5-534-00229-4. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/469021>
2. Кукарцев, В. В. Теория баз данных : учебник / В. В. Кукарцев, Р. Ю. Царев, О. А. Антамошкин. — Красноярск: Сибирский федеральный университет, 2017. — 180 с. — ISBN 978-5-7638-3621-9. — Текст: электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/84153.html> — Режим доступа: для авторизир. пользователей
3. Разработка баз данных: учебное пособие / А. С. Дорофеев, Р. С. Дорофеев, С. А. Рогачева, С. С. Сосинская. — Саратов: Ай Пи Эр Медиа, 2018. — 241 с. — ISBN 978-5-4486-0114-9. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: <https://www.iprbookshop.ru/70276.html> — Режим доступа: для авторизир. пользователей - DOI: <https://doi.org/10.23682/70276>
4. Стружкин, Н. П. Базы данных: проектирование. Практикум : учебное пособие для вузов / Н. П. Стружкин, В. В. Годин. — Москва : Издательство Юрайт, 2021. — 291 с. — (Высшее образование). — ISBN 978-5-534-00739-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/470023>.
5. Базы данных: учебное пособие — Саратов: Научная книга, 2012. — 158 с. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: <https://www.iprbookshop.ru/6261.html> — Режим доступа: для авторизир. пользователей.
6. Основы современных баз данных: методическая разработка к выполнению лабораторных работ (№1-3) / Золотухин сост. [и др.]. — Липецк: Липецкий государственный технический университет, ЭБС АСВ, 2013. — 37 с. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: <https://www.iprbookshop.ru/22906.html> - Режим доступа: для авторизир. пользователей
7. Самуйлов, С. В. Базы данных: учебно-методическое пособие для выполнения лабораторной и контрольной работы / С. В. Самуйлов. — Саратов: Вузовское образование, 2016. — 50с. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: <https://www.iprbookshop.ru/47276.html> — Режим доступа: для авторизир. пользователей
8. Братченко, Н. Ю. Распределенные базы данных: лабораторный практикум / Н. Ю. Братченко. — Ставрополь: Северо-Кавказский федеральный университет, 2014. — 180 с. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: <https://www.iprbookshop.ru/63129.html> — Режим доступа: для авторизир.

пользователей

9. Системы управления базами данных: лабораторный практикум / составители Д. Л. Осипов, М. Г. Огур. — Ставрополь: Северо-Кавказский федеральный университет, 2017. — 148 с. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: <https://www.iprbookshop.ru/75595.html> — Режим доступа: для авторизир. пользователей

2. МЕТОДИЧЕСКИЕ УКАЗАНИЯ

2.1. РАЗДЕЛ 1. СТРУКТУРИРОВАНИЕ БАЗ ДАННЫХ

2.1.1. ТЕМА 1. ОПТИМИЗАЦИЯ СТРУКТУРЫ БАЗЫ ДАННЫХ

Основные вопросы:

1. Оптимизация структуры базы данных.
2. Основные характеристики базы данных: избыточность, полнота заполнения, уровень сложности.
3. Первая, вторая и третья нормальные формы базы данных.
4. Задача оптимизации структуры базы данных.

Лабораторная работа 1. Транзакции

Задание:

1. Добавить в таблицу 1 000 000 текстовых записей на русском языке, каждая длиной не менее 100 символов UNICODE.
2. Написать запрос на изменение этих данных содержащих подстроку. Количество изменяемых строк должно быть не менее 10 000.
3. Определить выполнение запроса внутри транзакции.
4. Произвести считывание данных из таблицы во время выполнения транзакций.
5. Установить различные уровни изоляции:
READ Uncommitted
READ Committed
Repeatable READ
Serializable
6. Продемонстрировать график зависимости времени на выполнение транзакции при использовании различных типов изоляции.
7. Продемонстрировать график зависимости времени чтения данных другим пользователем во время выполнения транзакции.

2.1.2. ТЕМА 2. РАСПРЕДЕЛЕНИЕ БАЗ ДАННЫХ В СЕТИ

Основные вопросы:

1. Распределение баз данных в сети.
2. Локальные сети и типы блокировок данных.
3. Создание сетей в режиме клиент - сервер: установка режима, язык SQL - сетевые функции, администрирование баз данных.

Лабораторная работа 2. Курсоры

Операции в реляционной базе данных выполняются над множеством строк. Например, набор строк, возвращаемый инструкцией SELECT, содержит все строки, которые удовлетворяют условиям, указанным в предложении WHERE инструкции. Такой полный набор строк,

возвращаемых инструкцией, называется результирующим набором. Приложения, особенно интерактивные, не всегда эффективно работают с результирующим набором как с единым целым. Им нужен механизм, позволяющий обрабатывать одну строку или небольшое их число за один раз. Курсоры являются расширением результирующих наборов, которые предоставляют такой механизм.

Курсоры позволяют усовершенствовать обработку результатов:

- позиционируясь на отдельные строки результирующего набора;
- получая одну или несколько строк от текущей позиции в результирующем наборе;
- поддерживая изменение данных в строках в текущей позиции результирующего набора;
- поддерживая разные уровни видимости изменений, сделанных другими пользователями для данных, представленных в результирующем наборе;
- предоставляя инструкциям SQL в скриптах, хранимых процедурах и триггерах доступ к данным результирующего набора.

Задание:

1. Сделать выборку из таблицы с миллионом записей при помощи курсора. Найти минимальное или максимальное значение любого столбца таблицы.

2. Поменять местами имена всех фамилий на букву 'А' с именами всех фамилий на букву 'К'. Например: Андреев Игорь, Кузнецов Владислав → Андреев Владислав, Кузнецов Игорь

2.2. РАЗДЕЛ 2. Ведение баз данных

2.2.1. ТЕМА 3. Проектирование баз данных

Основные вопросы:

1. Проектирование баз данных.
2. Система управления базами данных, архитектура, состав, реляционные структуры, характеристики, архитектура клиент - сервер.
3. Создание пользовательского интерфейса: экраны, меню, отчёты, запросы.
4. Организация многопользовательского доступа к базам данных: техника блокировки, временные маркеры, тупиковые ситуации и способы их разрешения.
5. Этапы автоматизированного проектирования.
6. Оптимизация баз данных.

Лабораторная работа 3. Резервное копирование базы данных

Задание:

1. Создать резервную копию базы данных.
2. Удалить базу данных.
3. Восстановить базу данных из резервной копии.

4. Создать инкрементальный бекап базы со следующими параметрами:
5. Дифференциальный - в понедельник в 8:00.
6. Инкрементальный - каждые 4 часа
7. Журнал транзакций - каждые 10 минут.
8. Остановить базу данных.
9. Восстановить базу данных на 15:30 вторника.

2.2.2. ТЕМА 4. Внедрение и эксплуатация баз данных

Основные вопросы:

1. Внедрение и эксплуатация баз данных.
2. Этапы внедрения: параллельные и последовательные планы, генерация и модернизация системы, установление связей с инородными системами.
3. Разработка систем помощи и инструкций пользователя.

Лабораторная работа 4. Моделирование нагрузки и оценка производительности реляционных баз данных

Задание:

Цель: разработка приложения с максимальным быстродействием, которое принимает HTTP-запросы от клиентов, выполняет действие, предусмотренное вариантом задания и записывает результаты в БД.

Спецификация задания:

Язык разработки приложения – любой язык высокого уровня

СУБД – MySQL/PostgreSQL и MS SQL

Клиент – внешняя программа, генерирующая поток сообщений фиксированного содержания в формате HTTP-пакета (предлагается использовать Jmeter, <http://jmeter.apache.org/>)

Протокол обмена сообщениями – HTTP

Требования к клиентской части

1. Клиентский модуль должен генерировать поток HTTP-запросов с максимально возможной скоростью через запрос типа GET, например: `http://localhost/?cl_cod=m1`
2. Формат сообщения: Код клиента - символьная константа, по которой можно однозначно идентифицировать клиента, приславшего сообщение. Каждый клиент имеет свой уникальный код. Дополнительные параметры запроса, которые зависят от конкретного варианта задания (см. Таблицу 1).
3. Клиентская часть должна управляться следующими параметрами:
 - Имя модуля – идентификатор, уникально идентифицирующая клиента (код клиента). Например, `m1`.
 - Время работы модуля в секундах – параметр, определяющий время, в течении которого модуль должен генерировать пакеты.
 - Интервал задержки – параметр, определяющий время задержки между отправками пакетов в микросекундах. Если этот параметр

не указан, то модуль генерирует пакеты с максимальной интенсивностью без задержек.

4. В конце работы, модуль выдает информацию о том, сколько пакетов было отправлено на сервер.

Внимание! Все эти перечисленные требования позволяет удовлетворить Jmeter (см. Выше), поэтому реализовывать клиентскую часть не обязательно.

Требования к серверной части

Серверная часть представляет собой многопоточное приложение, которое позволяет принимать запросы от клиента, выполнять вычисление согласно варианту задания и помещать результаты работы в БД.

Алгоритм работы серверного приложения

1. Принять HTTP-запрос
2. Разобрать параметры GET запроса, выделив из него код клиента и номер пакета
3. Определить IP-адрес источника сообщения
4. Сформировать дату и время прихода запроса, как текущее время на стороне сервера
5. Выполнить действие, предусмотренное вариантом задания (см. Таблицу 1)
6. Записать в таблицу БД полученную от клиента информацию + результаты вычислений + IP адрес источника + дату и время получения запроса. (Структуру данных для хранения результатом разработать самостоятельно).

Требования к проведению эксперимента

- Эксперимент следует проводить таким образом, чтобы было запущено несколько клиентов с различных хостов.
- Цель эксперимента – «выжать» максимальное быстродействие из серверного приложения Ориентировочные показатели: Минимальное быстродействие – 1000 запросов/сек

Предоставление результата

По окончании работы предоставить рабочую программу и отчет о результатах проведения эксперимента с указанием следующих параметров:

- описание схемы проведения эксперимента
- указание количества хостов клиентов, с которых осуществлялась отправка пакетов
- максимально полученное быстродействие (запросы/сек)
- процент потерь пакетов (если они произошли в результате эксперимента)
- указать причину, по которой дальнейшее увеличение быстродействия невозможно (найти узкое место).
- Описание выполняемого действия

Варианты параметров запроса

1. Вычисление определенного интеграла $f(x) = x \cdot \sin(x)$ методом прямоугольников A и B — пределы интегрирования, h - шаг
2. Вычисление определенного интеграла $f(x) = x \cdot \cos(x)$ методом прямоугольников A и B — пределы интегрирования, h - шаг
3. Вычисление определенного интеграла $f(x) = x \cdot e^{-x}$ методом прямоугольников A и B — пределы интегрирования, h - шаг
4. Вычисление определенного интеграла $f(x) = \sin^2(x)$ методом прямоугольников A и B — пределы интегрирования, h - шаг
5. Вычисление определенного интеграла $f(x) = \cos^2(x)$ методом прямоугольников A и B — пределы интегрирования, h - шаг
6. Вычисление определенного интеграла $f(x) = x \cdot \sin(x^2)$ методом прямоугольников A и B — пределы интегрирования, h - шаг
7. Вычисление определенного интеграла $f(x) = x \cdot \cos(x^2)$ методом прямоугольников A и B — пределы интегрирования, h - шаг
8. Вычисление определенного интеграла $f(x) = x^2(e^{-2x})$ методом прямоугольников A и B — пределы интегрирования, h - шаг
9. Вычисление определенного интеграла $f(x) = x \cdot \sin(x)$ методом трапеций A и B — пределы интегрирования, h - шаг
10. Вычисление определенного интеграла $f(x) = x \cdot \cos(x)$ методом трапеций A и B — пределы интегрирования, h - шаг
11. Вычисление определенного интеграла $f(x) = x \cdot e^{-x}$ методом трапеций A и B — пределы интегрирования, h - шаг
12. Вычисление определенного интеграла $f(x) = \sin^2(x)$ методом трапеций A и B — пределы интегрирования, h - шаг
13. Вычисление определенного интеграла $f(x) = \cos^2(x)$ методом трапеций A и B — пределы интегрирования, h - шаг
14. Вычисление определенного интеграла $f(x) = x \cdot \sin(x^2)$ методом трапеций A и B — пределы интегрирования, h - шаг
15. Вычисление определенного интеграла $f(x) = x \cdot \cos(x^2)$ методом трапеций A и B — пределы интегрирования, h - шаг
16. Вычисление определенного интеграла $f(x) = x^2(e^{-2x})$ методом трапеций A и B — пределы интегрирования, h - шаг
17. Вычисление определенного интеграла $f(x) = x \cdot \sin(x)$ методом парабол (метод Симпсона) A и B — пределы интегрирования, h - шаг
18. Вычисление определенного интеграла $f(x) = x \cdot \cos(x)$ методом парабол (метод Симпсона) A и B — пределы интегрирования, h - шаг
19. Вычисление определенного интеграла $f(x) = x \cdot e^{-x}$ методом парабол (метод Симпсона) A и B — пределы интегрирования, h - шаг
20. Вычисление определенного интеграла $f(x) = \sin^2(x)$ методом парабол (метод Симпсона) A и B — пределы интегрирования, h - шаг
21. Вычисление определенного интеграла $f(x) = \cos^2(x)$ методом парабол (метод Симпсона) A и B — пределы интегрирования, h - шаг
22. Вычисление определенного интеграла $f(x) = x \cdot \sin(x^2)$ методом парабол (метод Симпсона) A и B — пределы интегрирования, h - шаг

23. Вычисление определенного интеграла $f(x) = x \cdot \cos(x^2)$ методом парабол (метод Симпсона) A и B — пределы интегрирования, h - шаг
24. Вычисление определенного интеграла $f(x) = x^2(e^{-2x})$ методом парабол (метод Симпсона) A и B — пределы интегрирования, h - шаг

2.3. РАЗДЕЛ 3. Проектирование баз данных

2.3.1. ТЕМА 5. Постановка задачи на проектирование

Основные вопросы:

1. Постановка задачи на проектирование.
2. Описание исходной информации, целей построения базы данных, описание источников и потребителей информации.
3. Формализация требований к хранению и обработке данных: конфиденциальности данных, сохранности данных, достоверности данных.
4. Определение объема и избыточности данных.

Лабораторная работа 5. Работа с не реляционными базами данных (noSQL)

Задание:

Цель: научиться использовать noSQL БД для хранения и обработки информации.

Задание: модифицировать л/р №6 таким образом чтобы в качестве СУБД использовалась noSQL технология. Провести аналогичные эксперименты и поместить результаты экспериментов в отчет.

Спецификация задания: Язык разработки приложения – любой язык высокого уровня СУБД – noSQL (Redis, Hadoop, MongoDB и т.п.) Клиент – внешняя программа, генерирующая поток сообщений фиксированного содержания в формате HTTP-пакета (предлагается использовать Jmeter, <http://jmeter.apache.org/>) Протокол обмена сообщениями – HTTP

Требования к серверной части: модифицировать серверную часть таким образом чтобы обработка запросов велась

Алгоритм работы серверного приложения

1. Принять HTTP-запрос
2. Разобрать параметры GET запроса, выделив из него код клиента и номер пакета
3. Определить IP-адрес источника сообщения
4. Сформировать дату и время прихода запроса, как текущее время на стороне сервера
5. Выполнить действие, предусмотренное вариантом задания (см. Таблицу 1)
6. Записать в таблицу БД полученную от клиента информацию + результаты вычислений + IP адрес источника + дату и время получения

запроса. (Структуру данных для хранения результатом разработать самостоятельно)

2.3.2. ТЕМА 6. Разработка проекта базы данных

Основные вопросы:

1. Разработка проекта базы данных.
2. Определение структуры хранения данных. Определение организационной структуры обслуживания базы данных.
3. Определение состава задач ведения базы данных.
4. Определение структуры пользовательского интерфейса информационной системы базы данных.
5. Определение структуры сетевой модели информационной системы.

Лабораторная работа 6. Работа с базой данных на прикладном языке программирования

Задание:

1. Разработать приложение которое будет подключаться к вашей базе данных и выполнять запросы 2 и 3 лабораторной работы.
2. Добавить возможность добавления новых данных через модели.
3. Запустить хранимые процедуры и функции.
4. Использовать разные уровни изоляции транзакций