

Проект по базам данных (2023-2024 уч. г.)

0. Реализуйте приложение, позволяющее выполнять операции над коллекциями данных заданных типов и контекстами их хранения (коллекциями данных).

Коллекция данных описывается набором строковых параметров (набор параметров однозначно идентифицирует коллекцию данных):

- название пула схем данных, хранящего схемы данных;
- название схемы данных, хранящей коллекции данных;
- название коллекции данных.

Коллекция данных представляет собой ассоциативный контейнер (конкретная реализация определяется вариантом), в котором каждый объект данных соответствует некоторому уникальному ключу. Для ассоциативного контейнера необходимо вынести интерфейсную и реализовать этот интерфейс. Взаимодействие с коллекцией объектов происходит посредством выполнения одной из операций над ней:

- добавление новой записи по ключу;
- чтение записи по её ключу;
- чтение набора записей с ключами из диапазона [*minbound*... *maxbound*];
- обновление данных для записи по ключу;
- удаление существующей записи по ключу.

Во время работы приложения возможно выполнение также следующих операций:

- добавление/удаление пулов данных;
- добавление/удаление схем данных для заданного пула данных;
- добавление/удаление коллекций данных для заданной схемы данных заданного пула данных.

Поток команд, выполняемых в рамках работы приложения, поступает из файла, путь к которому подаётся в качестве аргумента командной строки. Формат команд в файле определите самостоятельно.

Дополнительные задания:

Перед описанием задания в квадратных скобках указываются номера пререквизитов (заданий, которые необходимо выполнить для выполнения текущего). Частичная (неполная) реализация дополнительных заданий **не допускается** (за исключением заданий 6, 7).

1. [0] Реализуйте интерактивный диалог с пользователем. Пользователь при этом может вводить конкретные команды (формат ввода определите самостоятельно) и подавать на вход файлы с потоком команд.

2. [0] Реализуйте механизм персистентности данных в коллекциях данных, позволяющий выполнять запросы к данным в рамках коллекции данных на заданный момент времени (дата и время, для которых нужно вернуть актуальную версию данных, передаются как параметр). Для реализации механизма персистентности используйте поведенческие паттерны проектирования “Команда” и “Цепочка обязанностей”.

3. [0] Реализуйте механизм вторичных индексов, позволяющий выполнять эффективный поиск по различным отношениям порядка на пространстве данных (дублирование объектов данных в коллекциях данных, построенных по различным отношениям порядка на одном и том же

наборе объектов данных, при этом запрещается). Обеспечьте выбор индекса для поиска при помощи указания ключа отношения порядка (в виде строки, подаваемой как параметр поиска).

4. [0] Обеспечьте хранение объектов строк, размещённых в объектах данных, на основе структурного паттерна проектирования “Приспособленец”. Дублирования объектов строк для разных объектов (независимо от контекста хранения) при этом запрещены. Доступ к строковому пулу обеспечьте на основе порождающего паттерна проектирования “Одиночка”.

5. [0] Реализуйте механизмы сохранения состояния системы хранения данных в файловую систему и восстановления состояния системы хранения данных из файловой системы.

6. [0] Реализуйте возможность кастомизации (при создании) реализаций ассоциативных контейнеров, репрезентирующих коллекции данных: AVL-дерево, красно-чёрное дерево, косое дерево, scapegoat-дерево B-дерево, B^+ -дерево, B^* -дерево, B^{*+} -дерево.

7. Реализуйте функционал приложения в виде сервера, запросы на который поступают из клиентских приложений. При этом взаимодействие клиентских приложений с серверным должно быть реализовано посредством средств сетевого взаимодействия (FTP, HTTP, SOAP, REST API, gRPC, WCF, etc.) либо межпроцессного взаимодействия (IPC).

8. [0, 7] Реализуйте механизмы регистрации авторизации пользователя в системе (на клиентской стороне) и открытия пользовательской сессии (на серверной стороне) через пару значений <логин, пароль>, с дальнейшим взаимодействием клиентской стороны с серверной на основе передачи вместе с запросом токена аутентификации. Пароль при этом должен храниться и передаваться в виде хеша (используйте хеш-функцию SHA-256). Логин пользователей уникальны в рамках системы, могут содержать только символы букв и цифр в количестве [5..15] (обеспечьте валидацию на стороне сервера). Пароль должен содержать не менее 8 символов (обеспечьте валидацию на стороне клиента). Формат хранения и передачи данных для авторизации пользователей определите самостоятельно.

9. [0, 7, 8] На основе передаваемого в клиентские запросы токена аутентификации реализуйте различные роли, разграничивающие доступ к выполнению операций в рамках системы:

- администратор - имеет возможности создания новых пользователей; управления (выдача, блокировка) ролями других пользователей (кроме администраторов); управления доступом к схемам данных и доступом к операциям на уровне схем данных для заданных коллекций данных (режимы “только для чтения” и “чтение и модификация”);
- редактор - имеет возможности управления пулами (добавление/удаление), схемами (добавление/удаление) данных, коллекциями данных (добавление/удаление) в соответствии с предоставленными правами доступа;
- пользователь - имеет возможности взаимодействия с коллекциями данных в соответствии с предоставленными правами доступа.

10. [0, 7] Реализуйте комплекс серверных приложений (далее: кластер), одно из которых (далее: entrypoint) обрабатывает входящие пользовательские запросы и делегирует их остальным серверам (далее: storage), а storage-сервера хранят данные. Распределение данных по storage серверам должно быть примерно равномерным по памяти в произвольный момент времени. При реализации распределения данных предусмотрите реализацию механизмов горизонтального и вертикального шардирования.

11. [0, 7] Реализуйте механизм асинхронной обработки запросов (результатом запроса на выполнение операции должен являться идентификатор запроса (используйте формат GUID v4), по которому впоследствии должна иметься возможность получения результатов запроса).

12 [0, 7]. Реализуйте децентрализованную систему обработки запросов (entrypoint-сервер отсутствует; любой запрос может быть отправлен на любой из узлов кластера; результат асинхронного запроса также может быть получен из любого узла кластера).

13. [0] Реализуйте серверное приложение, собирающее логи клиентской (и, если есть, серверной) части приложения в файловые потоки вывода. Конфигурирование серверного логгера обеспечьте на основе файла со структурой JSON.