Полякова Татьяна 23КНТ-5

Лабораторная работа 2.

Отчет.

Описание предметной области:

Предметная область: музыкальная библиотека (управление музыкальной коллекцией).

БД предназначена для хранения и организации информации о музыкальных произведениях, обеспечивая удобный доступ к данным и возможности управления коллекцией и поддерживает следующие поля:

* ID: (int): Уникальный идентификатор записи. Гарантирует однозначную идентификацию каждой записи в базе данных.
* Title: (String): Название музыкального произведения.
* Artist: (String): Имя исполнителя или группы.
* Album: (String): Название альбома, к которому принадлежит произведение. Single - если произведение не входит в альбом.
* Genre: (String): Жанр музыкального произведения.
* Duration: (int): Продолжительность музыкального произведения в секундах.

База данных поддерживает следующие функции:

* Создание и открытие базы данных: Инициализация пустой базы данных (создание файла music\_database.json) или загрузка данных из существующего файла.
* Добавление музыкальных записей: Ввод информации о новом музыкальном произведении и добавление его в базу данных. Для сохранения композиции необходимо наличие всех полей. Проверка на уникальность ID.
* Редактирование музыкальных записей: Любое поле может быть изменено, но в случае id будет проверка на уникальность.
* Удаление музыкальных записей: Удаление записей по ID или по значению конкретного поля (название, исполнитель, альбом, жанр).
* Поиск музыкальных записей: Поиск записей по значению любого из полей (с использованием индексирования для полей для повышения производительности).
* Резервное копирование и восстановление базы данных: Сохранение копии базы данных и восстановление из резервной копии.
* Очистка базы данных: Удаление всех записей из базы данных.
* Удаление базы данных: Удаление файла music\_database.json базы данных.
* Экспорт в файл \*.xlsx: Выгрузка всех данных в табличный формат Excel.

Временная статистика.

| Операция: | 100 записей | 1000 записей | 10000 записей |
| --- | --- | --- | --- |
| Добавление записи | 1955400 нс = 1,95 мс | 3277100 нс = 3,27 мс | 17581800 нс = 17,5818 мс |
| Удаление записи по полю | 2173100 нс = 2,17 мс | 3213400 нс = 3,2134 мс | 16825800 нс = 16,8258 мс |
| Поиск записи по полю | 18500 нс = 0,0185 | 6200 нс = 0,0062 мс | 11800 нс = 0,0118 мс |

Алгоритмы являются эффективными, но запись на диск добавляет существенное замедление на больших объёмах. Основное время тратится на запись базы данных на диск, особенно при увеличении размера базы.

Анализ сложности-эффективности алгоритмов.

* Создание базы данных: Сложность O(1).
* Добавление записи: Вставка в HashMap имеет сложность O(1) в среднем, а индексация — O(log n) из-за использования TreeMap. В среднем сложность O(log n).
* Удаление по ключу: Удаление из HashMap — O(1), деиндексация — O(log n).
* В среднем сложность O(log n).
* Удаление по полю: Сложность O(r log n), где n — общее число записей, а r — количество записей, которые нужно удалить по заданному полю. Поиск по ключу в TreeMap — O(log n).
* Поиск по полю: Сложность O(log n).
* Очистка базы данных: Сложность O(1). Очистка HashMap и TreeMap — O(1).
* Удаление базы данных: Сложность O(1). Удаление файла — O(1).
* Индексирование и деиндексирование записей или полей: O(log n) - из-за использования TreeMap.