Обоснования использования данной структуры данных InMemorySimpleDB в контексте поставленных требований:

1. **Возможность добавления новых записей**. Метод addRecord(Record record) позволяет добавлять новые записи, добавляя каждую запись в три отдельных TreeMap, индексированных по разным полям (account, name, value).

2. **Возможность удаления записей**. Метод deleteRecord(long account) удаляет запись по ключу account. После удаления записи из accountIndex, дополнительно удаляются соответствующие записи из других индексов (nameIndex и valueIndex), что предотвращает наличие неконсистентных данных.

3. **Возможность изменения записей**. Метод updateRecord(Record record) позволяет обновлять записи. Сначала удаляются старые значения из nameIndex и valueIndex на основе данных из accountIndex, затем обновляются все три индекса. Это обеспечивает целостность данных при изменении.

4. **Получение записи по любому полю с одинаковой алгоритмической сложностью**. TreeMap обеспечивает доступ к элементам с логарифмической сложностью O(log n), благодаря чему метод getRecord(String key, String value) позволяет извлекать записи по ключу с требуемой производительностью для любого поля (account, name, value).

5. **Экономный способ хранения данных в памяти**. Хотя использование трех TreeMap для каждого поля записи увеличивает потребление памяти, такое решение обосновано необходимостью быстрого доступа к записям по разным полям с сохранением логарифмической сложности операций. Каждый TreeMap хранит ссылки на объекты Record, а не копии данных, что снижает избыточность хранения данных. В случае больших объемов данных можно рассмотреть альтернативные подходы, например, использование баз данных с поддержкой индексации по нескольким полям, что может быть более эффективно в плане использования памяти.