



Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ
КАФЕДРА СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ

РАБОТА ДЛЯ КЕЙС-ЧЕМПИОНАТА

«Разработка реляционной системы управления базами данных databasetopit»

ДИСЦИПЛИНА: «Программирование на основе классов и шаблонов»

Выполнили: студенты группы ИУ5-23Б

Ахметзянова Г. Т.

Коротенко А. А.

Миронов Н. П.

Немцов И. С.

Федяев А. С.

Москва, 2026 г.

Описание выполненных работ по шагам

Настоящий отчет содержит последовательное описание практических шагов, выполненных командой разработчиков в процессе создания системы управления базами данных (СУБД) databasetopit. Информация полностью основана на хронологии фиксации изменений в системе контроля версий Git. Текст изложен простым техническим языком, исключая избыточные теоретические описания.

- 1. Подготовка среды разработки и основы синтаксического разбора.** Была создана структура проекта и настроен конфигурационный файл сборки программного кода. Написан начальный программный модуль, который принимает текстовый запрос на языке SQL, делит его на отдельные минимальные составляющие части (слова, знаки и операторы) и строит из них структурированное дерево запроса для его дальнейшей обработки.
- 2. Создание бинарного хранилища данных.** Разработан механизм записи информации на жесткий диск. Данные сохраняются в файлы специального двоичного формата, разделенные на блоки фиксированного размера по 8 килобайт. Чтобы программа не перечитывала файлы постоянно, создана область буферной памяти, которая хранит наиболее востребованные блоки в оперативной памяти компьютера.
- 3. Разработка алгоритмов быстрого поиска (Индексы).** Для ускорения работы с таблицами была реализована древовидная структура данных (B+ дерево). Она позволяет находить, добавлять и удалять строки из таблицы не полным перебором всего файла, а путем быстрого точечного поиска по ключевому значению. Также добавлены механизмы создания дополнительных (вторичных) индексов для поиска по любым выбранным колонкам.
- 4. Реализация транзакций и защита от сбоев.** Для обеспечения надежности данных внедрены команды начала, фиксации и отмены группы операций (транзакций). Написан модуль упреждающего логирования: любые изменения сначала записываются в специальный журнал, и только потом

применяются к файлам данных. Это гарантирует, что при внезапном выключении компьютера данные не повредятся, а незавершенные операции будут корректно отменены.

5.Разработка сетевого сервера и разграничение прав. База данных переведена на сетевой режим работы. Настроен сервер, который принимает запросы от пользователей по локальной сети. Реализована система регистрации и входа по паролям с использованием безопасного шифрования. Добавлены команды выдачи и отзыва прав доступа к таблицам. Все данные о пользователях и их правах хранятся внутри скрытой системной таблицы.

6.Интеграция СУБД. Для интеграции СУБД с внешними аналитическими системами и быстрого наполнения таблиц разработан модуль пакетного импорта CSV-файлов. Функционал поддерживается как на уровне SQL-движка (оператор LOAD CSV), так и на уровне сетевого HTTP-API сервера (эндпоинт /import-csv).

7.Интеграция модуля перевода с естественного языка. Создан программный блок взаимодействия с искусственным интеллектом (моделью Mistral AI). Когда пользователь вводит запрос на обычном русском языке, сервер автоматически собирает информацию о структуре текущих таблиц, отправляет ее вместе с запросом в модель и получает обратно готовое, синтаксически верное выражение на языке SQL для мгновенного выполнения.

8.Создание графического интерфейса пользователя. На языке C# разработано полноценное оконное приложение для управления созданной СУБД. В приложении сделано окно текстового ввода запросов, обзорщик доступных таблиц, визуальные таблицы вывода результатов с возможностью постраничного листания, а также специальный чат-режим, где общение с базой данных выглядит как диалог в мессенджере с выводом результатов прямо в сообщения.

9.Разработка официального информационного веб-сайта проекта. Для презентации разработанной системы, размещения технической

документации и публикации руководства пользователя был создан и запущен информационный веб-сайт. Развернутая веб-страница проекта доступна в глобальной сети по адресу: <https://db-topiteleninglmsabcerybalkaigayana.pages.dev/>.

10. Оптимизация работы и автоматическое тестирование. Проведена финальная настройка внутренних алгоритмов: объединены методы связывания нескольких таблиц и улучшены механизмы сортировки. В репозитории настроен автоматический конвейер проверки: при каждом обновлении кода система сама собирает и тестирует программу на операционных системах Linux и Windows.

Использованные паттерны в проекте databasetopit:

Поведенческие паттерны:

1. Interpreter (Интерпретатор): Рекурсивное вычисление узлов AST дерева.
2. Strategy (Стратегия): Выбор алгоритмов (Index-Based Join, Hash Join, Index Scan).
3. Command / Compensation Action: Записи компенсации для обеспечения ROLLBACK.
4. Pipeline (Конвейер): Обработка запросов через цепочку (Lexer -> Parser -> Executor).

Паттерны конкурентности и параллелизма:

5. Thread Pool (Пул потоков): Сервер обрабатывает HTTP-запросы в пуле потоков.
6. Thread-Specific Storage: Изоляция контекста сессий через thread_local.
7. Readers-Writer Lock: Разделяемая блокировка для чтения, эксклюзивная для записи.

Структурные и паттерны работы с данными:

8. Null Object / Option Type: Использование std::optional для значений БД.
9. State (Состояние): Управление сессиями (session_id) поверх HTTP.
10. REPL (Read-Eval-Print Loop): Интерактивный цикл работы CLI-клиента.

Ранее выделенные (Архитектура и Хранилище):

11. Client-Server (Клиент-сервер): Базовая архитектура системы.

12. MVVM (Model-View-ViewModel): Разделение логики в GUI на C#.

13. Write-Ahead Logging (WAL): Журнал транзакций для ACID. 14. B+ Tree: Кластеризованные индексы.

15. Slotted Page: Организация записей внутри страниц.

16. LRU Cache: Кэширование страниц памяти.

Заключение

Все запланированные этапы разработки успешно выполнены. Созданный программный продукт является полностью автономным, не использует сторонних СУБД и реализует весь цикл обработки реляционных данных от физического сохранения байтов на диск до интеллектуальной генерации кода с помощью искусственного интеллекта, вывода результатов в удобном графическом виде и публикации справочных материалов на информационном веб-ресурсе.

