BonsaiDB :: User Manual

John Fajardo

Version: 1.1
Date: July 11, 2025

1 Introduction

Welcome to BonsaiDB!

BonsaiDB is a lightweight, high-performance database engine written in C++17. This project is designed as a practical demonstration of advanced backend concepts, including efficient data structures, disk storage management, and concurrent programming.

This manual will guide you through the process of compiling, installing, and using BonsaiDB via its Command Line Interface (CLI), as well as show you how to run the included tests and benchmarks.

2 Compilation and Installation

To compile BonsaiDB from source, you will need a C++ compiler compatible with the C++17 standard and CMake.

2.1 Requirements

- **CMake**: Version 3.10 or higher.
- **C++17 Compiler**: GCC 7+, Clang 5+, or MSVC 2017+.

2.2 Compilation Steps

1. Open a terminal and clone the project repository from GitHub.
2. Navigate to the project root and create a build directory (*out-of-source build*) to keep the source clean:

```
mkdir build
cd build
```

3. Run CMake to generate the build files and then compile the project:

```
cmake ..
make -j$(nproc)
```

Note: `$(nproc)` is a Linux command that returns the number of available processing cores, allowing for faster parallel compilation. On other systems, you can replace it with a number, e.g., `make -j4`.

4. Once finished, you will find the main executable `bonsaidb` along with the test executables inside the `build` directory.

---

3 Using the Command Line Interface (CLI)

BonsaiDB is managed through an interactive command line application.

3.1 Starting the Engine

To start the CLI, run the `bonsaidb` program from the `build` directory, specifying the database file you want to create or use:

```
./bonsaidb my_database.db
```

Once inside, you will see the prompt `bonsaidb>`.

3.2 Available Commands

Command	Syntax	Description
insert	insert <id> <name> <age> <balance>	Inserts a new record into the database. The name must not contain spaces.
select	select <id>	Searches for and displays a specific record by its `id`.
delete	delete <id>	Deletes a record from the database using its `id`.
dump	dump	Shows all records stored in the database in CSV format.
help	help	Displays this help message with the list of commands.
exit	exit	Closes the interactive session and exits the program.

---

4 Tests and Benchmarks

The project includes a set of tests to verify the engine's correctness and benchmarks to measure its performance.

4.1 Running the Tests

After compiling the project, the test executables will be available in the `build/test` directory. You can run them individually to verify the functionality of each component. For example:

```
./test/test_insert
./test/test_query
./test/test_concurrency
```

4.2 Running the Benchmarks

The repository includes a script to compile and run the stress tests automatically. From the project root, run:

```
./benchmarks/run_benchmarks.sh
```

This script will compile the project in a temporary directory (`build_bench`) and save the benchmark results in `benchmarks/results/bench.log`.