

Daehyun Kim

Database engineer | Storage internals | Vector search | Systems tooling

kimdhuyng@gmail.com | vimkim.dev | github.com/vimkim | [LinkedIn](#) | [Technical blog](#)

SUMMARY

Database engineer focused on CUBRID engine development, storage internals, vector search, and systems programming. Recent work includes the OOS epic for out-of-line large-column storage and vector type/operator/function support, with side projects in CLI tooling and AI-assisted development workflows.

EDUCATION

Korea University — B.Eng. in Computer Science and Engineering, GPA 4.03 / 4.5 2019.03–2023.02

Coursework: operating systems, algorithms, computer architecture, databases, networks. Programming languages coursework included an interpreter, type checker, and garbage collector.

McGill University — B.A. Economics, Minor in Finance, GPA 3.11 / 4.0 2014.09–2018.08

CS coursework in software systems, computer science, and foundations of programming. Thesis: *Skewness in Individual U.S. Stock Log Returns at Different Investment Horizons*.

Naver Connect Foundation Boostcamp — Web/Mobile 8th cohort 2023.06–2023.12

SELECTED WORK

CUBRID Out-of-line Overflow Storage (OOS) CBRD-26357 / CUBRID engine

- Driving OOS, an engine feature that stores large column values out of line instead of always keeping every column inline in heap records.
- Designed around compact OOS references in heap records to reduce unnecessary I/O and improve heap page density for projection-heavy scans.
- Worked through MVCC, recovery, HA, and existing overflow-record consistency constraints; maintained [OOS design notes](#).

CUBRID Vector Type, Operators, and ANN Performance CUBVEC / vector search

- Implemented vector type support and vector-related SQL operators/functions in CUBRID.
- Investigated CUBVEC-99 performance gaps between 20-dimensional and 256-dimensional ANN queries.
- Applied SIMD/vectorization build options to the usearch distance metric path in CUBVEC-108.

CUBRID Development Tooling Side projects / Rust / Shell / AI automation

- Built supporting tools for CUBRID development: [CUBRID Doxygen](#), [cubrid-oos-search](#), [cubrid-test-ai-reporter](#), and [my-cubrid-skills](#).
- Built focused utilities including [cubrid-ha-oneclick](#), [cubrid-logtopbind-rs](#), [cubrid-memmondiff](#), and [cubrid-dev2-pr](#).

patchers Side project / Rust / terminal UI

- [Terminal-based interactive tool](#) for reviewing and slicing large diff/patch files into smaller, reviewable hunks.
- Designed around keyboard-driven Git workflows for cleaner commits, review preparation, and patch sharing.

Markdown to Jira/Confluence Side project / Rust / WebAssembly

- [Browser-based utility](#) for converting Markdown into Jira/Confluence-friendly markup.
- Related tooling includes [md-to-jira-uploader](#) for Jira upload workflows.

TECHNICAL SKILLS

Primary: Rust, C++, C, SQL

Systems: Linux, POSIX, gdb, perf, ufttrace, strace, valgrind, make, cmake, docker

Tooling: nushell, git, GitHub Actions, bash, zsh, Neovim, fzf, rg, jq, direnv, just

Platforms: Arch Linux, Rocky Linux, Ubuntu

Familiar: Python, Go, TypeScript, Java, OCaml, Node.js, MySQL