

[오픈소스 프로젝트 소개]

라이브러리 형태의 멀티 모델 데이터베이스 엔진, GlueSQL

('26.6.22, 작성자 : 문태훈)

GlueSQL은 2019년 12월 첫 커밋을 시작으로 개발되어 온 Rust 기반 오픈소스 SQL 데이터베이스 엔진입니다. SQL 파서와 실행 계층, 다양한 저장소 옵션을 라이브러리 형태로 제공하며, 인메모리 저장소부터 파일 기반 저장소, key-value 및 임베디드 저장소, 커스텀 저장소까지 프로젝트의 데이터 계층에 SQL 인터페이스를 결합할 수 있도록 설계되었습니다.

GlueSQL의 가장 큰 특징은 SQL 처리 계층과 저장소 계층을 분리한 구조입니다. 이를 통해 정형 데이터뿐 아니라 스키마가 고정되지 않은 비정형·반정형 데이터에도 SQL을 적용할 수 있고, 개발자는 SQL 문자열뿐 아니라 GlueSQL의 AST Builder를 통해 쿼리를 직접 구성할 수도 있습니다. GlueSQL은 새로운 저장소 구현체를 만들 때 SQL 동작을 같은 기준으로 검증할 수 있도록, 재사용 가능한 공통 테스트 케이스를 라이브러리 형태로 제공합니다. 또한 핵심 코드의 라인 기준 테스트 커버리지를 약 99% 수준으로 유지하며, SQL 엔진의 기존 동작이 안정적으로 유지되는지 지속적으로 확인하고 있습니다.

2026년 6월 기준 GlueSQL은 GitHub에서 약 3.1천 개의 Star를 받으며, Rust 애플리케이션에 SQL 기반 데이터 처리 기능을 내장하거나 커스텀 저장소 위에 SQL 인터페이스를 제공하려는 개발자들에게 꾸준히 주목받고 있습니다. 프로젝트는 GitHub 저장소를 중심으로 이슈 제기, PR, 테스트 및 문서 개선 방식의 기여가 이루어지고 있습니다.

□ 오픈소스 허브 프로젝트 - GlueSQL

▶ 오픈소스 프로젝트(OpenSource Project) - GlueSQL

구분	설명	
프로젝트 개요	프로젝트 저장소	https://github.com/gluesql/gluesql
	홈페이지	https://gluesql.org/docs
	라이선스	Apache-2.0 license
	프로젝트 분야	Database, SQL Engine, Rust, Developer Tools
	프로젝트 소개	GlueSQL은 SQL 파서, 실행 계층, 저장소 추상화 계층을 제공하는 Rust 기반 오픈소스 SQL 데이터베이스 엔진입니다. 특정 저장소에 고정된 단일 데이터베이스 제품이 아니라, 인메모리 저장소, 파일 기반 저장소, key-value 및 임베디드 저장소, 사용자가 직접 구현한 커스텀 저장소 위에서 SQL 인터페이스를 사용할 수 있도록 설계된 라이브러리형 프로젝트입니다. 개발자는 GlueSQL을 애플리케이션에 통합해 별도의 데이터베이스 서버 없이 SQL 기반 조회, 필터링, 정렬, 집계 기능을 데이터 계층에 결합할 수 있습니다.

구분	설명	
<p>핵심 가치</p>	<p>GlueSQL의 핵심 가치는 SQL 처리 계층과 저장소 계층을 분리해, 애플리케이션이나 서비스가 이미 가지고 있는 데이터 저장 방식 위에 SQL 인터페이스를 결합할 수 있게 하는 데 있습니다. 개발자는 특정 DBMS에 종속되지 않고 인메모리, 파일, key-value, 임베디드, NoSQL 성격의 저장소 등 각 환경에 맞는 저장소를 구현하거나 선택해 SQL 조회, 필터링, 정렬, 집계 기능을 재사용할 수 있습니다.</p> <p>특히 GlueSQL은 저장소마다 데이터 모델, 인덱스, 스캔 방식, 실행 계층이 다르다는 점을 고려해 쿼리 플래너를 저장소 특성에 맞게 확장할 수 있도록 설계되어 있습니다. 이질적인 저장소를 하나의 최저공통분모 방식으로만 다루면 성능과 기능이 제한될 수 있는데, GlueSQL은 공통 SQL 파서와 실행 계층을 제공하면서도 필요한 경우 저장소 구현체가 자신에게 맞는 쿼리 계획 단계를 구성할 수 있게 해 서로 다른 데이터 저장소를 SQL 인터페이스와 연결하는 기반을 제공합니다.</p>	
<p>주요 특징</p>	<p>등장배경</p>	<p>GlueSQL은 처음부터 전통적인 데이터베이스 서버를 새로 만들기 위한 프로젝트라기보다, 다양한 애플리케이션 환경에서 반복적으로 발생하는 데이터 관리 문제를 SQL이라는 익숙한 인터페이스로 해결해보려는 시도에서 출발했습니다.</p> <p>2019년 무렵 웹 프론트엔드 개발 환경은 점점 복잡한 상태와 데이터를 다루게 되었지만, 클라이언트 측 저장소나 상태 관리 방식은 여전히 애플리케이션마다 직접 구조를 만들고 쿼리 로직을 구현해야 하는 경우가 많았습니다. 백엔드에서는 SQL 데이터베이스를 통해 데이터 구조화와 조회 로직을 안정적으로 처리할 수 있는데, 브라우저나 임베디드 환경에서는 비교적 작은 데이터와 단순한 실행 환경임에도 SQL 기반 데이터 처리를 자연스럽게 활용하기 어려운 점이 문제의식이 되었습니다.</p> <p>초기에는 Rust와 WebAssembly를 활용해 웹에서도 동작할 수 있는 SQL 엔진을 만드는 것이 중요한 출발점이었습니다. 그러나 개발 과정에서 SQL 처리 계층과 저장소 계층을 분리하면 이 구조가 웹 환경에만 한정되지 않는다는 점이 분명해졌습니다. 읽기와 쓰기 인터페이스를 구현할 수 있는 저장소라면 메모리, 파일, key-value 저장소, 임베디드 저장소, 커스텀 저장소 등 다양한 환경에 SQL 계층을 결합할 수 있었고, 이 방향이 GlueSQL이라는 이름처럼 서로 다른 데이터 저장소와 SQL 인터페이스를 연결하는 프로젝트로 확장되는 계기가 되었습니다.</p>
	<p>핵심 기능</p>	<p>GlueSQL은 SQL 문장을 해석하고 실행 계획으로 변환한 뒤, 저장소 구현체를 통해 데이터를 조회·수정하는 SQL 파서와 실행 엔진을 제공합니다. 조건 검색, 정렬, 집계, 조인, primary key 및 index 기반 조회와 같은 데이터 처리 기능을 지원하며, 스키마가 있는 정형 데이터뿐 아니라 스키마가 고정되지 않은 데이터도 SQL 인터페이스로 다룰 수 있도록 설계되어 있습니다.</p> <p>또한 GlueSQL은 인메모리 저장소뿐 아니라 파일, JSON, CSV, Sled, Redb, MongoDB, Redis, Git, Parquet 등 다양한 저장소 구현체를 제공하며, 개발자가 직접 커스텀 저장소를</p>

구분	설명	
		<p>구현할 수 있도록 trait 기반의 스토리지 인터페이스를 제공합니다. SQL 문자열을 직접 사용하는 방식 외에도 Rust 코드에서 쿼리를 구성할 수 있는 AST Builder를 지원하고, 새로운 저장소 구현체가 동일한 SQL 동작을 만족하는지 검증할 수 있도록 공통 테스트 케이스를 라이브러리 형태로 제공합니다.</p>
	<p>차별화 요소</p>	<p>GlueSQL은 특정 저장소를 전제로 동작하는 단일 데이터베이스 제품이 아니라, SQL 처리 계층을 저장소와 분리해 재사용할 수 있도록 설계된 라이브러리형 SQL 엔진이라는 점에서 차별화됩니다. 기존에는 SQL이 필요하다면 별도의 DBMS를 도입하거나, 각 저장소에 맞는 조회 로직을 애플리케이션 코드에서 직접 구현해야 하는 경우가 많았습니다. GlueSQL은 저장소 구현체를 연결해 같은 SQL 파서와 플래너, 실행 계층을 활용할 수 있도록 하며, 저장소의 성격에 따라 쿼리 처리 방식을 조정할 수 있는 구조를 지향합니다.</p> <p>특히 GlueSQL은 저장소가 가진 실행 능력에 따라 SQL 처리 책임을 다르게 배분할 수 있는 방향으로 발전하고 있습니다. 자체 쿼리 플래너나 실행 계층을 가진 저장소와 결합할 때는 GlueSQL이 상대적으로 얇은 SQL 인터페이스와 계획 계층으로 동작하고, key-value 저장소나 파일 기반 저장소처럼 SQL 실행 기능이 없는 저장소에서는 GlueSQL이 더 많은 실행 책임을 담당하는 방식입니다. 이를 통해 서로 다른 저장소를 단순히 하나의 공통 인터페이스로 감싸는 데 그치지 않고, 각 저장소의 특성을 살리면서 SQL 인터페이스를 제공하는 것을 목표로 합니다.</p>
	<p>대상 사용자</p>	<p>GlueSQL의 주요 대상 사용자는 Rust 애플리케이션 안에 SQL 기반 데이터 처리 기능을 내장하려는 개발자뿐 아니라, 이미 MongoDB, Redis, 파일 기반 데이터, key-value 저장소 등 특정 저장소를 사용하고 있으면서 그 위에 익숙한 SQL 인터페이스를 제공하고 싶은 개발자와 기업입니다. 기존 저장소를 다른 DBMS로 교체하지 않고도 조건 검색, 정렬, 집계 등 SQL식 데이터 접근 방식을 결합하고 싶은 경우 GlueSQL을 활용할 수 있습니다.</p> <p>또한 문서형 데이터베이스나 key-value 저장소처럼 SQL을 기본 인터페이스로 제공하지 않는 저장소를 사용하면서도, 서비스 내부 도구나 관리 기능, 데이터 조회 API, 테스트 환경에서 SQL의 표현력을 활용하고 싶은 팀에도 적합합니다. 더 나아가 SQL 엔진, 쿼리 플래너, 저장소 추상화, Rust 기반 데이터베이스 시스템을 학습하거나 실험하려는 개발자와 오픈소스 기여자도 주요 대상입니다.</p>
	<p>운영 환경</p>	<p>GlueSQL은 Rust crate로 애플리케이션에 포함되어 동작하는 라이브러리형 SQL 엔진입니다. Rust 애플리케이션이 실행되는 서버, CLI 도구, 데스크톱 애플리케이션, 테스트 환경 등에 내장해 사용할 수 있으며, 인메모리 저장소나 파일 기반</p>

구분	설명	
		<p>저장소를 선택하면 별도의 외부 DB 서버 없이도 SQL 기반 데이터 처리를 구성할 수 있습니다.</p> <p>실제 운영 구성은 선택한 저장소 구현체에 따라 달라집니다. 로컬 환경에서는 메모리, 파일, JSON, CSV, Sled, Redb 등의 저장소를 사용할 수 있고, MongoDB나 Redis 저장소를 선택하면 기존 외부 저장소 위에 GlueSQL의 SQL 인터페이스를 결합할 수 있습니다. JavaScript 환경에서는 GlueSQL.js를 통해 브라우저와 Node.js에서도 사용할 수 있습니다.</p>
	<p>활용 분야</p>	<p>GlueSQL은 애플리케이션 내부에 SQL 기반 데이터 처리 기능을 내장해야 하는 분야에서 활용할 수 있습니다. 서버 애플리케이션, CLI 도구, 데스크톱 도구, 내부 운영 도구 등에서 별도의 데이터베이스 서버를 새로 운영하지 않고도 조건 검색, 정렬, 집계와 같은 데이터 조회 기능을 제공할 수 있습니다.</p> <p>또한 MongoDB, Redis, 파일 기반 데이터, key-value 저장소 등 기존 저장소를 유지하면서 그 위에 SQL 조회 API나 관리용 쿼리 인터페이스를 제공하는 용도로 활용할 수 있습니다. 웹 애플리케이션에서는 브라우저 환경에 맞는 저장소 구현체와 결합해 클라이언트 측 데이터를 SQL 방식으로 다루는 기반으로 사용할 수 있으며, 빠른 기능 검증과 초기 제품 개발 단계에서 데이터 처리 로직을 유연하게 구성하는 데에도 활용할 수 있습니다.</p>
<p>프로젝트 생태계</p>	<p>커뮤니티 현황</p>	<p>GlueSQL은 GitHub 저장소를 중심으로 이슈 제기, 버그 리포트, PR 제출, 코드 리뷰, 문서 및 테스트 개선이 이루어지는 공개 오픈소스 프로젝트입니다. 별도로 GlueSQL Discord 커뮤니티를 운영하며, 컨트리뷰션 아카데미 참여자와 기존 기여자들이 질문, 코드 읽기, 리뷰 대응, 오프라인 모각코 등을 통해 꾸준히 기여 경험을 이어가고 있습니다.</p>
<p>오픈소스 허브</p>	<p>https://oss.kr/opensource/hub/56922</p>	

▶ 주요 개발자 (Main Developers) - 문태훈, 조형관

구분	내용	
개발자 소개	개발자 이름	문태훈
	개발자 소개	웹·앱 프론트엔드와 백엔드, 데이터베이스 개발 경험을 바탕으로 2019년 GlueSQL을 시작했으며, 현재 Rust 기반 오픈소스 SQL 데이터베이스 엔진 GlueSQL의 설계와 개발, 유지보수를 맡고 있습니다.
	이메일 또는 SNS	- Email: taehoon@gluesql.com - GitHub: panarch
	전문 분야	Frontend, Backend, Database, Rust, SQL Engine
	경력	- 웹·앱 프론트엔드, 백엔드, 데이터베이스 개발 17년차 - GlueSQL creator & maintainer (2019~)
참여 프로젝트	https://github.com/gluesql/gluesql	
기여 내용	GlueSQL의 creator이자 maintainer로서 저장소 독립적인 SQL 엔진 구조를 설계하고, 프로젝트의 핵심 개발과 유지보수를 주도하고 있습니다.	
오픈소스 참여 계기	GlueSQL을 시작하기 전부터 VexFlow 같은 오픈소스 프로젝트에 기여하며 협업 경험을 쌓아 왔습니다. 이후 여러 서비스와 제품을 개발하면서, 개발자들이 반복적으로 필요로 하지만 아직 충분히 제공되지 않는 기반 기술이 있다는 것을 느끼게 되었습니다. GlueSQL은 이러한 문제의식을 바탕으로, 다양한 저장소 위에서 SQL 인터페이스를 활용할 수 있는 오픈소스 기반 기술을 직접 만들고 싶다는 생각에서 시작되었습니다.	
기여 활동	<ul style="list-style-type: none"> - 2019년부터 GlueSQL 프로젝트의 핵심 구조 설계 및 개발 - SQL 실행 계층, 스토리지 trait, 플래너, AST Builder 등 주요 기능 설계와 구현 - GitHub 이슈 관리, PR 리뷰, 신규 기여자 온보딩을 진행하며 컨트리뷰션 아카데미 멘토링에도 참여 	
성장 포인트	처음부터 데이터베이스 엔진 개발을 전문적으로 해온 것은 아니었지만, Rust로 GlueSQL을 직접 만들면서 SQL 엔진과 데이터베이스 내부 구조를 하나씩 이해하게 되었습니다. 단순히 문서로 학습하는 데 그치지 않고, 직접 이해한 기능을 구현하고 테스트하며 프로젝트를 확장해 왔다는 점이 가장 큰 성장 포인트였습니다. GlueSQL은 만들면서 배우고, 배우면서 다시 설계하는 과정을 반복하게 해준 프로젝트입니다.	
후배들에게 조언	<p>오픈소스는 개발 역량을 압축적으로 성장시킬 수 있는 좋은 기회라고 생각합니다. 좋은 프로젝트의 코드를 읽고, 작은 기여를 하고, 경험 많은 메인테이너의 리뷰를 받는 과정은 회사 업무만으로는 만나기 어려운 학습 경험을 제공합니다. 특히 오픈소스 프로젝트는 외부 기여자가 이해하고 참여할 수 있어야 하기 때문에 코드 구조, 테스트, 문서화, 리뷰 문화에 더 신경 쓰게 되고, 이런 환경을 가까이에서 경험하는 것만으로도 개발자로서 큰 도움이 됩니다.</p> <p>AI가 개발 생산성을 높이는 시대에는 일자리가 사라질까를 걱정하기보다, 이 도구를 활용해 더 빠르게 배우고 성장하는 방법을 고민하는 것이 중요하다고 생각합니다. 예전에는 좋은 코드를 읽고 시행착오를 거치며 오랜 시간에 걸쳐 쌓아야 했던 경험들을, 이제는 AI의 도움을 받아 훨씬 짧은 시간 안에 분석하고 이해할 수 있습니다. 관심 있는 오픈소스 프로젝트를 정하고, AI를 활용해 코드를 읽고, 작은 기여를 반복해보면 생각보다 빠르게 성장할 수 있습니다.</p>	

구분	내용	
개발자 소개	개발자 이름	조형관
	개발자 소개	DBA로 커리어를 시작한 경험을 바탕으로 GlueSQL에 합류했으며, Rust로 작성된 다양한 데이터베이스 관련 오픈소스 프로젝트에 기여하고 있습니다.
	이메일 또는 SNS	- Email: devgony@gmail.com - GitHub: devgony
	전문 분야	Database, AI Agent
	경력	- 한화시스템(DBA 7년, FA 3년) - GlueSQL, sea-orm-spanner
참여 프로젝트	https://github.com/gluesql/gluesql	
기여 내용	Review, 신규 contributor에 대한 멘토링, PR 기여	
오픈소스 참여 계기	DBA 경력을 살려 개발자로 전향하는 과정에서 잘 기여하고 배울 수 있는 오픈소스 커뮤니티를 알게 되었습니다.	
기여 활동	총 82개 PR 기여 - 기존 DB 운영 경험을 살려 상용 DB들의 기능 구현 - Mongo storage를 통해 MongoDB에서도 SQL 및 FK를 사용할 수 있도록 기여	
성장 포인트	익숙한 DB 도메인을 기반으로 생소한 Rust 언어를 통해 빠르게 개발자로서 성장할 수 있었습니다.	
후배들에게 조언	오픈소스는 현 AI 시대에 개발 역량을 증명할 수 있는 유일한 수단이라고 볼 수 있을 정도로 중요해졌습니다. 항상 관심을 가지고 기여하고, 없다면 직접 만들어서 공개해본다면 어느새 성장한 개발자가 되어 있을 것입니다.	

▶ 커뮤니티(Community) - GlueSQL

구분	내용	
커뮤니티 개요	커뮤니티 소개	GlueSQL Discord 커뮤니티는 컨트리뷰션 아카데미와 GitHub 기여를 계기로 연결된 개발자들이 온라인에서 소통하는 커뮤니티입니다. 코드 리뷰와 이슈, 학습 내용, 모임 일정을 공유하며, 프로젝트 기여 과정에서 생기는 질문과 진행 상황을 함께 나누는 기여자 중심 소통 채널로 운영되고 있습니다.
	홈페이지 및 SNS	Discord 커뮤니티(초대 기반)
	프로젝트 저장소	https://github.com/gluesql/gluesql
	커뮤니티 분야	Rust, Database, SQL, Storage Systems, Developer Tools
	전문분야	SQL parsing/execution, storage abstraction, Rust library design, test suite
	커뮤니티 연혁	2021년 7월 컨트리뷰션 아카데미 참여를 계기로 GlueSQL Discord 커뮤니티를 운영하기 시작했습니다. 이후 GitHub 기여 활동과 연계해 온라인 소통을 이어가고 있으며, 필요에 따라 오프라인 모각코 모임도 함께 진행하고 있습니다.
	커뮤니티 미션/비전	GlueSQL에 관심 있는 개발자들이 Rust, SQL 엔진, 저장소 시스템을 함께 학습하고 실제 오픈소스 기여 경험을 쌓을 수 있는 커뮤니티를 지향합니다. 프로젝트 기여를 통해 GlueSQL을 발전시키는 동시에, 참여자들이 코드 읽기, 테스트 작성, 리뷰 대응, 기능 구현을 경험하며 성장하는 것을 목표로 합니다.
글로벌 커뮤니티	GlueSQL은 GitHub 저장소를 중심으로 지역 구분 없이 이슈와 PR을 받고 있으며, 국내외 개발자들의 기여가 함께 이루어지고 있습니다. Discord 커뮤니티는 컨트리뷰션 아카데미와 기존 기여자를 중심으로 코드 리뷰, 질문, 모임 일정을 공유하는 소통 채널로 활용하고 있습니다.	
커뮤니티 활동	주요 활동	<ul style="list-style-type: none"> - Discord 기반 온라인 소통과 기여 방향 논의 - GitHub 이슈, PR 리뷰, 신규 기여자 온보딩 - 컨트리뷰션 아카데미 멘토링 및 월 1~2회 오프라인 모각코 모임
	최근 활동 사례	최근에는 컨트리뷰션 아카데미 참여자와 신규 기여자를 중심으로 GlueSQL 코드 구조를 함께 읽고, 이슈 분석, 테스트 작성, PR 리뷰를 진행하고 있습니다. 온라인에서는 Discord를 통해 질문과 진행 상황을 공유하고, 오프라인에서는 모각코 형태로 모여 각자 맡은 기여 작업을 진행합니다.
	커뮤니티 문화	작은 기여를 환영하고, 코드를 함께 읽으며 이해한 내용을 바탕으로 기여하는 문화를 지향합니다. 빠른 결과보다 재현 가능한 이슈, 테스트가 포함된 변경, 리뷰를 통한 점진적인 개선을 중요하게 생각합니다.

구분	내용
커뮤니티 참여 방법	<p>현재는 기여 활동을 계기로 참여하는 방식으로 운영하고 있습니다. 컨트리뷰션 아카데미 멘티로 참여하거나, GlueSQL maintainer/reviewer 또는 기존 커뮤니티 구성원에게 연락해 오프라인 모임이나 기여 활동부터 함께 시작할 수 있습니다. GitHub에서는 이슈 확인, 질문, PR 제출을 통해 누구나 프로젝트 기여를 시작할 수 있습니다.</p>