

4.4 데이터베이스의 종류

4.4.1 관계형 데이터베이스

관계형 데이터베이스(RDBMS)는 행과 열을 가지는 표 형식 데이터를 저장하는 형태의 데이터 베이스를 가리키며 SQL이라는 언어를 써서 조작합니다.

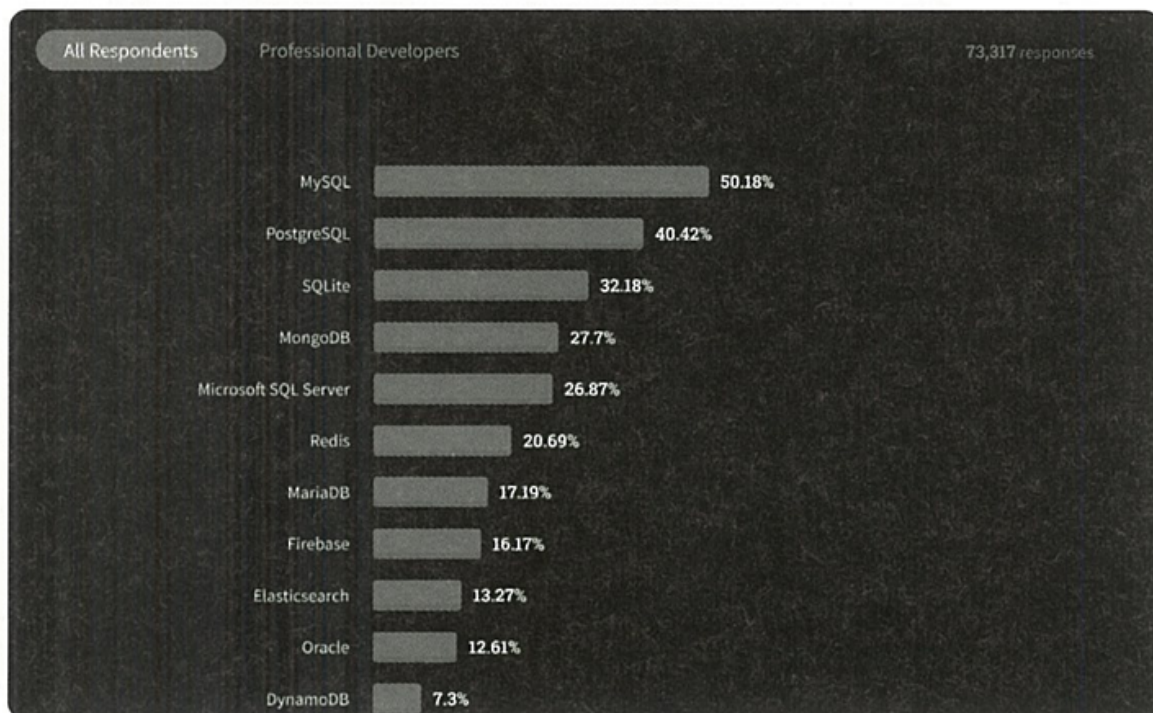
대표적으로 MySQL, PostgreSQL, 오라클, SQL Server, MSSQL 등이 있습니다.

관계형 데이터베이스의 경우 표준 SQL은 지키기는 하지만, 각각의 제품에 특화시킨 SQL을 사용합니다. 예를 들어 오라클의 경우 PL/SQL이라고 하며 SQL Server에서는 T-SQL, MySQL은 SQL을 씁니다.

MySQL

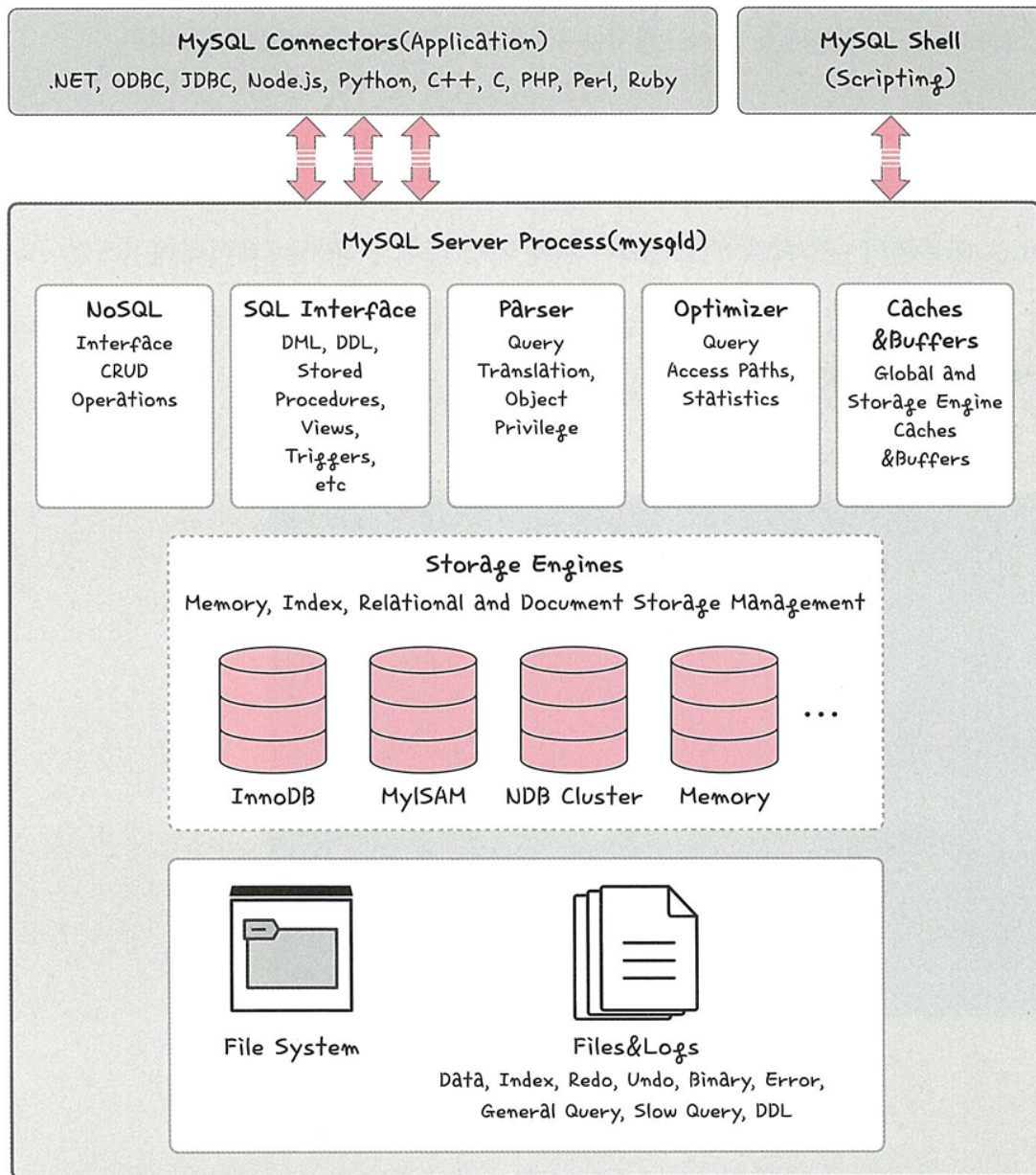
MySQL은 대부분의 운영체제와 호환되며 현재 가장 많이 사용하는 데이터베이스입니다.

▼ 그림 4-29 스택 오버플로우 조사 결과



C, C++로 만들어졌으며, MyISAM 인덱스 압축 기술, B-트리 기반의 인덱스, 스레드 기반의 메모리 할당 시스템, 매우 빠른 조인, 최대 64개의 인덱스를 제공합니다. 대용량 데이터베이스를 위해 설계되어 있고 롤백, 커밋, 이중 암호 지원 보안 등의 기능을 제공하며 많은 서비스에서 사용됩니다.

▼ 그림 4-30 MySQL 스토리지 엔진 아키텍처



<https://hoing.io/archives/2478>

PostgreSQL

PostgreSQL은 MySQL 다음으로 개발자들이 선호하는 데이터베이스 기술로 널리 인정받고 있습니다.

디스크 조각이 차지하는 영역을 회수할 수 있는 장치인 VACUUM이 특징입니다. 최대 테이블의 크기는 32TB이며 SQL뿐만 아니라 JSON을 이용해서 데이터에 접근할 수 있습니다. 지정시간에 복구하는 기능, 로깅, 접근 제어, 중첩된 트랜잭션, 백업 등을 할 수 있습니다.

쿼뷰 : MySQL과 PostgreSQL 비교표

기능 및 특징	MySQL	PostgreSQL
ORDBMS와 RDBMS 비교	관계형 데이터베이스 관리 시스템 (RDBMS)	객체 관계형 데이터베이스 관리 시스템(ORDBMS)
ACID 규정 준수	대부분의 엔진은 ACID 규정 준수를 제공하지만 MyISAM은 ACID를 지원하지 않습니다.	완벽한 지원
백업 및 복구	백업 및 복구 기능을 제공합니다.	효율적인 백업 및 복구 기능으로 정평이 나 있습니다.
크로스 플랫폼	대응	UNIX 기반 시스템에 최적
확장 기능 및 플러그인	다양하게 있음	확장성으로 유명한 PostGIS 등
외부 키	지원되지만 MyISAM에서는 지원되지 않습니다.	완벽한 지원
인덱싱 기법	다양한 기법 사용 가능	GIN 및 GiST와 같은 고급 유형 제공
SQL 데이터 유형	표준 유형 사용 가능	더 다양하고, 배열, hstore 포함
저장 프로시저	지원 있음	PL/pgSQL 언어로 더욱 고급화
트리거	지원 있음	유연하고 다국어 지원
뷰	지원 있음	구체화된 뷰 제공

사용 사례	MySQL	PostgreSQL
웹 애플리케이션	속도와 안정성으로 널리 인정받고 있습니다.	특히 복잡한 사용 사례에서 인기가 높아지고 있습니다.

공간 데이터베이스	기본 공간 기능	PostGIS 확장 기능으로 사용할 수 있는 고급 공간 기능
엔터프라이즈 시스템	다양한 엔터프라이즈 애플리케이션에 적합	견고성과 확장성을 위해 사용
데이터 웨어하우스	데이터 웨어하우스에 사용되지만 맞춤형 솔루션이 필요할 수 있습니다.	고급 데이터 유형이 있는 데이터 웨어하우스에 대한 지원 강화
임베디드 시스템	경량 버전 사용 가능	흔하지는 않지만 가능

4.4.2 NOSQL 데이터베이스

NoSQL(Not only SQL)이라는 슬로건에서 생겨난 데이터베이스입니다. SQL을 사용하지 않는 데이터베이스를 말하며, 대표적으로 MongoDB와 redis 등이 있습니다.

MongoDB

MongoDB는 JSON을 통해 데이터에 접근할 수 있고, Binary JSON 형태(BSON)로 데이터가 저장되며 와이어드타이거 엔진이 기본 스토리지 엔진으로 장착된 키-값 데이터 모델에서 확장된 문서 기반의 데이터베이스입니다. 확장성이 뛰어나며 빅데이터를 저장할 때 성능이 좋고 고가용성과 샤딩, 레플리카셋을 지원합니다. 또한, 스키마를 정해놓지 않고 데이터를 삽입할 수 있기 때문에 다양한 도메인의 데이터베이스를 기반으로 분석하거나 로깅 등을 구현할 때 강점을 보입니다.

와이어드 타이거엔진 - <https://rastalion.me/mongodb의-wiredtiger-스토리지-엔진/>

샤딩이란? - 데이터를 조각내 분산 저장하는 데이터 처리 기법, 일괄적 관리가 힘든 거대 데이터베이스나 네트워크를 작게 나눠서 저장하고 관리하는 방법입니다.

샤딩의 장점

1. 데이터 처리 속도

샤딩의 가장 큰 장점은 데이터 처리 속도를 월등히 향상시키는 것입니다. 만약 100만건의 데이터가 존재하고 100개의 노드가 있다면, 기존 블록체인 방식에서는 100개의 노드가 100만건의 데이터를 모두 공유하여 처리합니다.

하지만 샤딩을 활용할 경우 데이터 100을 100개로 쪼개어 노드 1개당 1만건의 데이터만 처리합니다.

이로 인해 노드에 가해지는 데이터 부하는 작아지고 데이터 처리 속도는 향상됩니다.

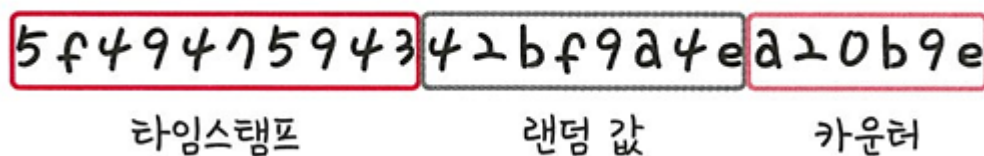
2. 탈중앙화 및 보안성 확보

샤딩의 두번째 장점은 탈중앙화 및 보안성 확보입니다.

샤딩을 사용하면 노드가 많아지며 여러 저장소에 데이터가 나뉘어 저장됩니다. 중앙 데이터 저장소가 없는 탈중앙화 저장방식은 해킹 취약점을 낮춥니다. 그로 인해 바이러스 침투 및 해킹으로부터 보안성을 확보할 수 있습니다.

또한, MongoDB는 도큐먼트를 생성할 때마다 다른 컬렉션에서 중복된 값을 지니기 힘든 유니크한 값인 ObjectID가 생성됩니다.

▼ 그림 4-33 MongoDB ObjectID



이는 기본키로 유닉스 시간 기반의 타임스탬프(4바이트), 랜덤 값(5바이트), 카운터(3바이트)로 이루어져 있습니다.

redis

redis는 인메모리 데이터베이스이자 키-값 데이터 모델 기반의 데이터베이스입니다.

기본적인 데이터 타입은 문자열(string)이며 최대 512MB까지 저장할 수 있습니다. 이외에도 셋(set), 해시(hash) 등을 지원합니다.

pub/sub 기능을 통해 채팅 시스템, 다른 데이터베이스 앞단에 두어 사용하는 캐싱 계층, 단순한 키-값이 필요한 세션 정보 관리, 정렬된 셋(sorted set) 자료 구조를 이용한 실시간 순위표 서비스에 사용합니다.

주요 차이점: 관계형 데이터베이스와 비관계형 데이터베이스

관계형 데이터베이스와 비관계형 데이터베이스는 서로 매우 다른 방식으로 데이터를 저장하고 관리합니다. 다음 섹션에서는 구체적인 차이점에 대해 설명합니다.

구조

관계형 데이터베이스는 데이터를 테이블 형식으로 저장하며 데이터 변형 및 테이블 관계에 관한 엄격한 규칙을 따릅니다. 따라서 데이터 무결성과 일관성을 유지하면서 정형 데이터에 대한 복잡한 쿼리를 처리할 수 있습니다.

비관계형 데이터베이스는 더 유연하며, 요구 사항이 변화하는 데이터에 보다 유용합니다. 따라서 이미지, 비디오, 문서 및 기타 반정형 및 비정형 콘텐츠를 저장하는 데 사용할 수 있습니다.

데이터 무결성 메커니즘

원자성, 일관성, 격리 및 내구성(ACID)은 데이터 처리에서 오류나 중단이 발생하더라도 데이터 무결성을 유지할 수 있는 데이터베이스의 기능을 말합니다.

관계형 데이터베이스 모델은 엄격한 ACID 속성을 따릅니다. 즉, 일련의 후속 작업이 항상 함께 완료됩니다. 단일 작업이 실패하면 전체 작업 세트가 실패합니다. 따라서 데이터 정확성이 항상 보장됩니다.

반면 비관계형 데이터베이스는 기본적으로 가용성이 보장되고 소프트 상태이며 궁극적으로 일관된(BASE) 보다 유연한 모델을 제공합니다.

비관계형 데이터베이스는 가용성을 보장하지만 즉각적인 일관성은 보장하지 않습니다. 데이터베이스 상태는 시간이 지남에 따라 변할 수 있으며 결국 일관된 상태가 됩니다. 일부 비관계형 데이터베이스는 ACID 규정 준수와 성능 또는 기타 장단점을 제공할 수 있습니다.

성능

관계형 데이터베이스의 성능은 디스크 하위 시스템에 따라 달라집니다. SSD를 사용하고 디스크를 Redundant Array of Independent Disks(RAID)로 구성하여 디스크를 최적화하면 데이터베이스 성능을 높일 수 있습니다. 성능을 극대화하려면 인덱스, 테이블 구조 및 쿼리도 최적화해야 합니다.

반면 NoSQL 데이터베이스의 성능은 네트워크 지연 시간, 하드웨어 클러스터 크기 및 호출 애플리케이션에 따라 달라집니다. 비관계형 데이터베이스는 다음 몇 가지 방법으로 성능을 높일 수 있습니다.

- 클러스터 크기 증대
- 네트워크 지연 시간 최소화
- 인덱스 및 캐시

NoSQL 데이터베이스는 특정 사용 사례에서 관계형 데이터베이스보다 더 높은 성능과 확장성을 제공합니다.

<https://aws.amazon.com/ko/compare/the-difference-between-relational-and-non-relational-databases/>