목차

Database 종류와 비교 3
데이터베이스 유형: 관계형 데이터베이스 (RDBMS - Relational Database Management System) 3
특징 3
탄생 배경 3
주요 사용처 3
대표적인 예시 3
데이터베이스 유형: NoSQL (Not Only SQL) 4
특징 4
탄생 배경 4
주요 사용처 4
대표적인 예시 4
데이터베이스 유형: 키-값 저장소 (Key-Value Store) 4
특징 4
탄생 배경 5
주요 사용처 5
대표적인 예시 5
데이터베이스 유형: 문서 데이터베이스 (Document Database) 5
특징 5
탄생 배경 5
주요 사용처 5
대표적인 예시 6
데이터베이스 유형: 컬럼 패밀리 데이터베이스 (Column-Family Database) 6
특징 6
탄생 배경 6
주요 사용처 6
대표적인 예시 6
데이터베이스 유형: 그래프 데이터베이스 (Graph Database) 7
특징 7
탄생 배경 7
주요 사용처 7
대표적인 예시 7
데이터베이스 유형: 검색 엔진 데이터베이스 (Search Engine Database) 7
특징 7
탄생 배경 8
주요 사용처 8
대표적인 예시 8
데이터베이스 유형: 시계열 데이터베이스 (TSDB - Time Series Database) 8
특징 8
탄생 배경 8
주요 사용처 8
대표적인 예시 9
데이터베이스 유형: NewSQL 데이터베이스 9
특징 9
탄생 배경 9
주요 사용처 9
대표적인 예시 9
데이터베이스 유형: 공간 데이터베이스 (Spatial Database) 10
특징 10

탄생 배경 10
주요 사용처 10
대표적인 예시 10
데이터베이스 유형: 벡터 데이터베이스 (Vector Database) 10
특징 10
탄생 배경 10
주요 사용처 11
대표적인 예시 11
데이터베이스 유형: 객체 지향 데이터베이스 (OODBMS - Object-Oriented Database Management System) 11
특징 11
탄생 배경 11
주요 사용처 11
대표적인 예시 11
데이터베이스 유형: XML 데이터베이스 (XML Database) 12
특징 12
탄생 배경 12
주요 사용처 12
대표적인 예시 12
데이터베이스 유형: 인메모리 데이터베이스 (IMDB - In-Memory Database) 12
특징 12
탄생 배경 13
주요 사용처 13
대표적인 예시 13
데이터베이스 유형: 블록체인 데이터베이스 (Blockchain Database / 분산 원장 기술 - DLT) ... 13
특징 13
탄생 배경 13
주요 사용처 14
대표적인 예시 14
데이터베이스 유형: 클라우드 데이터베이스 (Cloud Database / 서비스형 데이터베이스 - DBaaS) 14
특징 14
탄생 배경 14
주요 사용처 14
대표적인 예시 15

Database 종류와 비교

— *이강우 2025/05/27 11:10*

데이터베이스는 현대 정보 시스템의 핵심 구성 요소로, 데이터의 효율적인 저장, 관리, 검색을 지원합니다. 전통적인 관계형 데이터베이스(RDBMS) 외에도 다양한 요구사항을 충족하기 위한 여러 유형의 데이터베이스가 등장했습니다. 이 문서에서는 RDBMS, NoSQL, 시계열 데이터베이스(TSDB)를 포함한 주요 데이터베이스 유형을 소개합니다.

데이터베이스 유형: 관계형 데이터베이스 (RDBMS - Relational Database Management System)

특징

- 데이터를 정해진 스키마에 따라 테이블(Table) 형태로 저장 (행과 열 구조)
- SQL(Structured Query Language)을 사용하여 데이터 조작 및 질의
- ACID(원자성, 일관성, 고립성, 지속성) 트랜잭션을 통한 데이터 무결성 보장
- 정규화(Normalization)를 통해 데이터 중복 최소화
- 명확한 데이터 구조 및 관계 정의 가능

탄생 배경

- 1970년대 에드거 F. 커드(Edgar F. Codd)에 의해 제안된 관계형 모델에 기반
- 기존 파일 시스템이나 계층형/네트워크형 데이터베이스의 복잡성과 데이터 종속성 문제를 해결하기 위해 등장
- 데이터의 논리적 구조와 물리적 저장 구조를 분리하여 유연성 증대

주요 사용처

- 금융 시스템 (은행, 증권 거래)
- 전사적 자원 관리 (ERP) 시스템
- 고객 관계 관리 (CRM) 시스템
- 재고 관리, 인사 관리 등 대부분의 전통적인 기업 애플리케이션
- 데이터 일관성과 무결성이 매우 중요한 시스템

대표적인 예시

- Oracle Database
- MySQL
- PostgreSQL
- Microsoft SQL Server
- MariaDB
- SQLite

데이터베이스 유형: NoSQL (Not Only SQL)

특징

- 관계형 모델을 사용하지 않는 데이터베이스들을 통칭 (스키마 제약이 적거나 없음)
- 수평적 확장성(Horizontal Scaling) 및 고가용성(High Availability)에 용이
- 대용량 데이터 및 분산 환경 처리에 적합
- 다양한 데이터 모델 지원:
    - 키-값 (Key-Value)
    - 문서 (Document)
    - 컬럼 패밀리 (Column-Family)
    - 그래프 (Graph)
- BASE(Basically Available, Soft state, Eventually consistent) 속성을 따르는 경우가 많음 (ACID를 완전하게 지원하지 않을 수 있음)

탄생 배경

- 2000년대 후반, 웹 2.0 시대의 도래와 함께 폭증하는 비정형/반정형 데이터, 높은 트래픽, 빠른 개발 속도 요구에 대응하기 위해 등장
- 기존 RDBMS의 엄격한 스키마와 수직적 확장(Scale-up)의 한계를 극복하고자 함

주요 사용처

- 대규모 웹 애플리케이션 (소셜 미디어, 콘텐츠 플랫폼)
- 실시간 데이터 처리 및 분석
- 빅데이터 저장 및 처리
- IoT 데이터 관리
- 캐싱 시스템

대표적인 예시

- *(각 모델별 상세는 아래 개별 항목 참조)*
- MongoDB (문서)
- Apache Cassandra (컬럼 패밀리)
- Redis (키-값)
- Neo4j (그래프)

데이터베이스 유형: 키-값 저장소 (Key-Value Store)

특징

- 고유한 키(Key)와 그에 해당하는 값(Value)의 쌍으로 데이터를 저장하는 단순한 모델
- 매우 빠른 읽기 및 쓰기 속도 제공
- 단순한 API (GET, PUT, DELETE 등)
- 확장성이 뛰어남

탄생 배경

- NoSQL의 한 형태로, 매우 간단한 데이터 모델을 통해 성능과 확장성을 극대화하려는 요구에서 출발
- 웹 애플리케이션의 세션 관리, 캐싱, 사용자 프로필 저장 등에 적합

주요 사용처

- 인메모리 캐싱
- 세션 관리
- 사용자 프로필 저장
- 실시간 추천 데이터 저장
- 메시지 큐잉 시스템의 일부

대표적인 예시

- Redis
- Amazon DynamoDB (기본 모델은 키-값)
- Memcached
- Riak KV

데이터베이스 유형: 문서 데이터베이스 (Document Database)

특징

- 데이터를 JSON, BSON, XML과 같은 문서(Document) 형태로 저장 및 질의
- 각 문서는 필드와 값의 쌍으로 구성되며, 스키마가 유연하여 문서마다 다른 구조를 가질 수 있음
- 객체 지향 프로그래밍의 객체와 유사한 형태로 데이터를 다룰 수 있어 개발 편의성 높음
- 문서 내 특정 필드에 대한 인덱싱 및 질의 지원

탄생 배경

- NoSQL의 한 형태로, 웹 애플리케이션에서 많이 사용되는 JSON과 같은 유연한 데이터 구조를 효율적으로 저장하고 관리하기 위해 개발
- 개발자가 RDBMS의 스키마 변경 어려움 없이 빠르게 애플리케이션을 개발하고 데이터를 저장할 수 있도록 지원

주요 사용처

- 콘텐츠 관리 시스템 (CMS)

- 블로그 및 사용자 생성 콘텐츠 플랫폼
- 전자상거래 상품 카탈로그
- 모바일 애플리케이션 백엔드
- 로그 데이터 저장 및 분석

대표적인 예시

- MongoDB
- Couchbase
- Amazon DocumentDB (MongoDB 호환)
- Elasticsearch (검색 엔진이지만 문서 저장 기능 강력)

데이터베이스 유형: 컬럼 패밀리 데이터베이스 (Column-Family Database)

특징

- 데이터를 행(Row)이 아닌 컬럼(Column) 또는 컬럼 패밀리(Column Family) 단위로 저장
- 각 행은 서로 다른 컬럼을 가질 수 있어 스키마가 매우 유연함 (희소 행렬 저장에 유리)
- 대량의 데이터 쓰기 및 특정 컬럼 범위 읽기에 최적화
- 높은 확장성과 가용성을 제공하도록 설계

탄생 배경

- Google의 Bigtable 논문에 기반한 NoSQL 데이터베이스 유형
- 페타바이트급의 초대용량 데이터를 분산 환경에서 효율적으로 처리하기 위해 개발

주요 사용처

- 빅데이터 분석 애플리케이션
- 시계열 데이터 저장 (일부 경우)
- 로그 데이터, 이벤트 데이터 등 대량의 쓰기가 발생하는 시스템
- 추천 시스템의 사용자 활동 기록

대표적인 예시

- Apache Cassandra
- Apache HBase
- Google Cloud Bigtable
- ScyllaDB

데이터베이스 유형: 그래프 데이터베이스 (Graph Database)

특징

- 데이터를 노드(Node), 관계(Edge/Relationship), 속성(Property)으로 표현하고 저장
- 복잡한 관계를 가진 데이터의 질의 및 탐색에 최적화 (예: 최단 경로, 패턴 매칭)
- 유연한 스키마 구조
- 직관적인 데이터 모델링 가능

탄생 배경

- 소셜 네트워크, 추천 시스템 등 데이터 간의 '관계'가 중요한 애플리케이션에서 RDBMS의 한계를 극복하기 위해 등장
- 그래프 이론에 기반하여 연결된 데이터를 효율적으로 처리

주요 사용처

- 소셜 네트워크 분석
- 추천 엔진
- 사기 탐지 시스템 (FDS)
- 지식 그래프 구축 및 관리
- 네트워크 및 IT 인프라 모니터링
- 공급망 분석

대표적인 예시

- Neo4j
- Amazon Neptune
- ArangoDB (멀티 모델 DB)
- JanusGraph
- TigerGraph

데이터베이스 유형: 검색 엔진 데이터베이스 (Search Engine Database)

특징

- 전문 검색(Full-text search) 및 정보 검색에 최적화
- 역 인덱스(Inverted Index)를 사용하여 텍스트 데이터의 빠른 검색 지원
- 복잡한 검색 질의, 집계, 필터링, 관련도 순위 지정, 자동 완성 등 다양한 검색 기능 제공
- 비정형 및 반정형 데이터 처리 용이
- 대량의 문서 및 로그 데이터에 대한 수평적 확장성 우수

탄생 배경

- 웹 검색 엔진을 위해 개발된 기술에서 발전하여, 애플리케이션 내 검색 기능, 로그 분석 등 다양한 분야로 확장
- 기존 데이터베이스로는 복잡한 텍스트 검색 요건을 충족하기 어려웠음

주요 사용처

- 웹사이트 및 전자상거래 상품 검색
- 기업 내부 자료 검색 (Enterprise Search)
- 로그 수집, 분석 및 시각화 (ELK Stack 등)
- 애플리케이션 성능 모니터링 (APM) 데이터 분석
- 보안 정보 및 이벤트 관리 (SIEM)

대표적인 예시

- Elasticsearch
- Apache Solr
- OpenSearch (Elasticsearch의 포크)
- Algolia (서비스형)
- Meilisearch

데이터베이스 유형: 시계열 데이터베이스 (TSDB - Time Series Database)

특징

- 시간 순서대로 정렬된 데이터 포인트(시계열 데이터)의 저장, 처리, 분석에 최적화
- 고속 데이터 수집(High Ingestion Rate) 및 대량의 시계열 데이터 처리 능력
- 특정 시간 범위에 대한 빠른 질의 성능 (Windowing, Aggregation 등)
- 데이터 압축, 자동 롤업(Roll-up), 데이터 보존 정책(Retention Policy) 등의 기능 제공
- 태그(Tag)를 이용한 다차원 데이터 모델링 지원

탄생 배경

- IoT 센서 데이터, 시스템/애플리케이션 모니터링 데이터, 금융 시장 데이터 등 시간 축에 따라 발생하는 데이터가 급증하면서 기존 RDBMS나 범용 NoSQL로는 효율적인 처리가 어려워 등장

주요 사용처

- IT 인프라 및 애플리케이션 모니터링 (서버 메트릭, 로그)
- 산업 IoT (IIoT) 데이터 분석, 스마트 팩토리
- 금융 시장 데이터 분석 (주가, 거래량, 환율 변동)

- 센서 데이터 수집 및 분석 (스마트 홈, 헬스케어, 환경 모니터링)
- 실시간 이상 탐지 및 예측

대표적인 예시

- InfluxDB
- Prometheus
- TimescaleDB (PostgreSQL 확장)
- OpenTSDB
- Graphite
- QuestDB

데이터베이스 유형: NewSQL 데이터베이스

특징

- RDBMS의 ACID 트랜잭션과 SQL 인터페이스를 유지하면서 NoSQL의 수평적 확장성과 고가용성을 제공
- 온라인 트랜잭션 처리(OLTP) 워크로드에 적합하도록 설계
- 분산 환경에서도 강력한 데이터 일관성 보장 목표

탄생 배경

- 2010년대 초반, RDBMS의 확장성 문제와 NoSQL의 약한 일관성 문제를 동시에 해결하고자 등장
- 웹 스케일 애플리케이션에서 대량의 동시 트랜잭션을 데이터 무결성을 유지하며 처리해야 하는 요구에 부응

주요 사용처

- 미션 크리티컬한 금융 애플리케이션 (은행 코어 시스템, 증권 거래)
- 대규모 전자상거래 플랫폼의 주문 및 결제 시스템
- 높은 트랜잭션 처리량을 요구하는 온라인 게임
- 실시간 입찰 및 광고 시스템

대표적인 예시

- Google Spanner
- CockroachDB
- VoltDB (인메모리 특성도 강함)
- NuoDB
- TiDB
- Clustrix

데이터베이스 유형: 공간 데이터베이스 (Spatial Database)

특징

- 지리적 위치 정보, 공간 객체(점, 선, 면 등)를 저장, 질의, 관리
- 공간 인덱스(R-tree, Quadtree 등) 및 공간 연산(거리, 교차, 포함 관계 등)을 효율적으로 지원
- GIS(지리 정보 시스템) 표준(예: OGC 표준) 지원

탄생 배경

- 지도 서비스, 위치 기반 서비스(LBS), 도시 계획 등에서 공간 데이터를 효율적으로 처리하기 위해 개발
- 기존 데이터베이스는 이러한 유형의 데이터와 질의에 최적화되어 있지 않았음

주요 사용처

- GIS 애플리케이션 (지도 제작, 지리 분석)
- 내비게이션 시스템 및 위치 기반 서비스 (LBS)
- 도시 계획, 토지 관리, 환경 모니터링
- 물류 및 차량 경로 최적화, 자산 추적
- 부동산 정보 시스템

대표적인 예시

- PostGIS (PostgreSQL의 공간 데이터 확장)
- Oracle Spatial and Graph
- Microsoft SQL Server Spatial
- MySQL Spatial Extensions
- Esri ArcGIS 플랫폼의 지오데이터베이스

데이터베이스 유형: 벡터 데이터베이스 (Vector Database)

특징

- 고차원 벡터 임베딩(Vector Embedding)의 저장, 인덱싱, 검색에 특화
- 유사도 검색(Similarity Search)에 최적화 (예: k-최근접 이웃(k-NN), 근사 최근접 이웃(ANN) 검색)
- AI/ML 애플리케이션에서 생성된 벡터 데이터를 효율적으로 관리

탄생 배경

- 머신러닝, 특히 딥러닝의 발전으로 텍스트, 이미지, 오디오 등을 고차원 벡터로 변환하여 의미를 표현하는 방식이 일반화됨
- 이러한 벡터 데이터를 대규모로 저장하고 빠르게 유사성 검색을 수행할 필요성으로 등장

주요 사용처

- 의미 기반 검색 (Semantic Search) 시스템
- 이미지/오디오/비디오 검색 및 추천 시스템
- 자연어 처리(NLP) 기반의 질의응답, 텍스트 분류
- 이상 탐지, 사기 탐지 시스템
- 신약 개발, 유전체 분석 등 생명과학 연구

대표적인 예시

- Pinecone
- Weaviate
- Milvus
- Chroma DB
- Qdrant
- Vespa.ai (검색 엔진 및 벡터 DB 기능)
- Faiss (라이브러리, 직접적인 DB는 아니나 핵심 기술 제공)

데이터베이스 유형: 객체 지향 데이터베이스 (OODBMS - Object-Oriented Database Management System)

특징

- 데이터를 객체 지향 프로그래밍(OOP)에서와 같이 객체(Object) 형태로 표현하고 저장
- 클래스, 객체, 상속, 다형성, 캡슐화 등 OOP 개념 지원
- 프로그래밍 언어의 객체와 데이터베이스 객체 간의 임피던스 불일치(Impedance Mismatch) 해소
- 복잡한 관계 및 계층 구조를 가진 데이터 모델링 용이

탄생 배경

- 1980년대와 1990년대, 객체 지향 프로그래밍 패러다임의 확산과 함께 등장
- OOP로 개발된 애플리케이션의 데이터를 보다 자연스럽게 영속화(Persistence)하려는 시도

주요 사용처

- 복잡한 데이터 모델을 사용하는 공학 분야 (CAD/CAM, CAE)
- 과학 연구 데이터 관리 (생물정보학, 물리학)
- 통신 및 네트워크 관리 시스템
- 멀티미디어 데이터 관리
- *(상대적으로 RDBMS나 NoSQL에 비해 시장 점유율은 낮음)*

대표적인 예시

- ObjectStore (단종)

- GemStone/S
- Objectivity/DB
- InterSystems Caché (멀티 모델 DB, 객체 기능 포함)
- db4o (단종되었으나 영향력 있었음, 오픈소스)

데이터베이스 유형: XML 데이터베이스 (XML Database)

특징

- XML(eXtensible Markup Language) 문서를 네이티브(Native) 형식으로 저장 및 관리
- XML 데이터 모델(트리 구조, 요소, 속성 등)에 최적화된 질의 언어 지원 (XQuery, XPath)
- 유연한 스키마 또는 스키마리스(Schemaless) 운영 가능
- 문서 중심적 데이터 처리

탄생 배경

- 1990년대 후반 및 2000년대 초반, XML이 데이터 교환 및 문서 표현의 표준으로 널리 채택되면서 등장
- 기존 RDBMS는 계층적인 XML 구조를 저장하고 질의하는 데 비효율적이었음

주요 사용처

- 콘텐츠 관리 시스템(CMS), 특히 XML 기반 문서 관리
- 출판 및 디지털 아카이빙
- XML 기반의 데이터 통합 및 교환
- 기술 문서, 법률 문서 등 구조화된 문서 관리
- 웹 서비스의 메시지 저장

대표적인 예시

- BaseX
- eXist-db
- MarkLogic (멀티 모델 NoSQL 기능도 제공, XML 처리 강력)
- Oracle XML DB (Oracle Database 내 기능)

데이터베이스 유형: 인메모리 데이터베이스 (IMDB - In-Memory Database)

특징

- 모든 또는 주요 데이터를 디스크가 아닌 주기억장치(RAM)에 저장하여 운영

- 디스크 I/O 병목 현상을 제거하여 극도로 빠른 데이터 접근 및 처리 속도 제공
- 실시간 데이터 처리 및 분석에 매우 효과적
- 데이터 영속성을 위해 디스크 백업, 스냅샷, 로그 기록 등의 메커니즘 사용

탄생 배경

- RAM 가격 하락 및 용량 증가와 함께, 빠른 응답 속도를 요구하는 애플리케이션(금융 거래, 실시간 게임, 통신 등)의 증가로 발전
- 디스크 기반 DB의 성능 한계를 극복하고자 함

주요 사용처

- 실시간 분석 및 보고 시스템
- 고성능 캐싱 계층 (Caching Layer)
- 웹 애플리케이션 세션 관리
- 온라인 게임 서버 (빠른 상태 업데이트 및 동기화)
- 금융 거래 시스템 (고빈도 거래)
- 통신 네트워크 장비 (라우팅 테이블, 사용자 인증)

대표적인 예시

- SAP HANA
- Redis (주로 키-값 저장소/캐시지만, 인메모리 DB로 분류 가능)
- MemSQL (현 SingleStore)
- Oracle TimesTen
- VoltDB
- Apache Ignite

데이터베이스 유형: 블록체인 데이터베이스 (Blockchain Database / 분산 원장 기술 - DLT)

특징

- 탈중앙화되고 분산된 원장(Ledger)에 데이터를 블록(Block) 단위로 저장하고, 각 블록을 암호학적으로 연결(Chain)
- 불변성(Immutability): 한번 기록된 데이터는 변경하거나 삭제하기 매우 어려움
- 투명성(Transparency): 참여자 간 데이터 공유 및 검증 가능 (설정에 따라 다름)
- 합의 메커니즘(Consensus Mechanism)을 통해 데이터의 유효성 확보
- 중앙 관리자 없이 P2P 네트워크 기반으로 운영 가능

탄생 배경

- 2008년 사토시 나카모토의 비트코인 논문에서 처음 소개된 기술로, 신뢰할 수 있는 제3자 없이 안전한 P2P 거래를 가능하게 하기 위해 고안
- 데이터의 위변조 방지 및 투명한 공유에 대한 요구 증가

주요 사용처

- 암호화폐 (Bitcoin, Ethereum 등)
- 공급망 관리 및 추적 (제품 이력, 원산지 증명)
- 스마트 계약 (Smart Contracts) 실행 플랫폼
- 디지털 신원 확인 및 관리
- 전자 투표 시스템
- 의료 기록 및 지적 재산권 관리 (보안성 및 투명성 강화)

대표적인 예시

- *(엄밀히 말해 DB 제품이라기보다는 플랫폼 또는 기술 스택에 가까움)*
- Hyperledger Fabric
- Ethereum (플랫폼)
- Corda
- Bitcoin Core (소프트웨어)
- Amazon Quantum Ledger Database (QLDB) (AWS의 관리형 원장 DB)

데이터베이스 유형: 클라우드 데이터베이스 (Cloud Database / 서비스형 데이터베이스 - DBaaS)

특징

- 클라우드 컴퓨팅 환경에서 서비스 형태로 제공되는 데이터베이스
- 사용자는 물리적 인프라 관리(설치, 유지보수, 패치, 백업 등) 부담 없이 데이터베이스 사용 가능
- 탄력적인 확장성(Scalability) 및 고가용성(High Availability) 제공
- 사용한 만큼 비용을 지불하는 종량제(Pay-as-you-go) 또는 구독 모델
- RDBMS, NoSQL 등 다양한 유형의 데이터베이스 엔진을 서비스로 제공

탄생 배경

- 클라우드 컴퓨팅의 발전과 함께 기업 및 개발자들이 인프라 관리 부담을 줄이고 핵심 비즈니스에 집중할 수 있도록 지원하기 위해 등장
- 초기 비용 절감 및 빠른 서비스 구축 요구 증가

주요 사용처

- 웹 및 모바일 애플리케이션 백엔드
- 개발 및 테스트 환경
- 빅데이터 분석 및 머신러닝 플랫폼
- 재해 복구(DR) 및 백업 솔루션
- 스타트업부터 대기업까지 다양한 규모의 조직에서 사용

대표적인 예시

- **Amazon Web Services (AWS):**
  - Amazon RDS (Relational Database Service - MySQL, PostgreSQL, Oracle, SQL Server 등 지원)
  - Amazon Aurora (MySQL 및 PostgreSQL 호환 고성능 RDBMS)
  - Amazon DynamoDB (NoSQL Key-Value 및 Document)
  - Amazon Redshift (데이터 웨어하우스)
  - Amazon Neptune (Graph DB)
- **Microsoft Azure:**
  - Azure SQL Database (RDBMS)
  - Azure Cosmos DB (Multi-model NoSQL)
  - Azure Database for MySQL/PostgreSQL/MariaDB
- **Google Cloud Platform (GCP):**
  - Cloud SQL (MySQL, PostgreSQL, SQL Server)
  - Cloud Spanner (글로벌 분산 RDBMS - NewSQL)
  - Cloud Bigtable (NoSQL Column-Family)
  - Firestore (NoSQL Document)
- **기타:**
  - MongoDB Atlas (MongoDB의 완전 관리형 서비스)
  - Oracle Cloud Database Services

From:
https://atl.kr/dokuwiki/ - **AllThatLinux!**

Permanent link:
https://atl.kr/dokuwiki/doku.php/database_%EC%A2%85%EB%A5%98%EC%99%80_%EB%B9%84%EA%B5%90

Last update: **2025/05/27 11:10**