

목차

- 1.Data Base 의 정의와 종류
2. 관계형 데이터베이스 장점 및 단점 관련 제품
3. NoSQL 장점 및 단점 관련 제품
4. DB(제품)들의 장점 및 단점

www.quora.com 참조

1.Data Base 의 정의와 종류

1-1.Data Base : 여러 응용 시스템들의 통합된 정보들을 저장, 운용할 수 있는 공용 데이터들의 묶음

=> 데이터의 접근성 향상 - 데이터들이 관계와 함께 논리적으로 저장된 것

1) : 계층형 데이터 베이스 : 관계를 트리 구조로 정의 (부모 - 자식 형태) , 데이터의 중복문제 발생

2) : 네트워크형 데이터 베이스 : 계층형의 데이터 중복 문제를 해결, 그물 관계(거미줄) , 복잡한 구조로 인한 추후 유지보수의 어려움

3) : 관계형 데이터베이스 : 행과열로 구성된 Table 간의 관계시 사용, 이렇게 표현된 데이터를 SQL(Structured Query Language)사용하여 데이터 관리 및 접근

4) : NoSql 데이터베이스 : 관계형 데이터베이스보다 덜 제한적인 모델, 키와 값의 형태로 저장, 키를 사용하여 데이터 접근 및 관리

데이터 사이의 관계 유무에따라 SQL NoSQL를 쓰면됨

(join 또는 결합 구문은 디비내의 여러 테이블의 레코드를 조합하여 하나의 열로 표현한것 ex 현재 학생부 와 현재 성적을 join) (transaction 데이터베이스 관리 시스템 또는 이와 유사한 시스템간의 상호작용 단위이다)

2.관계형 데이터베이스(SQL) 장점 및 단점 관련 제품

장점 :

다양한 용도 사용 가능 ,
높은 성능,
데이터의 일관성 보증,
정규화에 따른 갱신 비용 최소화

단점:

대량의 데이터 입력 처리
갱신이 발생한 테이블의 인덱스 생성 및 스키마 변경
컬럼의 확장 어려움
단순히 빠른 결과

주요 제품 :

Oracle / Oracle
MS-SQL Server / MS
MySQL / Oracle
DB2 / IBM
InfoMix / IBM
Sybase / Sybase
Derby / APache
SQLite / Opensource

3.NoSQL 데이터베이스 장점 및 단점 관련 제품

관계형 데이터베이스가 아니며 SQL을 사용하지 않는다는 의미로 Not only SQL(개선 보안의 의미로 SQL 언어가 필요없다는 뜻)

장점 :

대용량 데이터
데이터 분산 처리
Cloud Computing
빠른 write & read speed
유연한 데이터 모델링

주요 제품:

key/value 휘발성 영속성
Memcached, Tokyo Tyrant, Flare, Roma, Redis

Document
스키마 정의 없음
MongoDB, CouchDB

Big Table(Column 형)DB
뛰어난 확장성 검색에 유리
Hbase, Casandara, Hypertable

4. DB(제품)들의 장점 및 단점

Rank			DBMS	Database Model
Jul 2018	Jun 2018	Jul 2017		
1.	1.	1.	Oracle +	Relational DBMS
2.	2.	2.	MySQL +	Relational DBMS
3.	3.	3.	Microsoft SQL Server +	Relational DBMS
4.	4.	4.	PostgreSQL +	Relational DBMS
5.	5.	5.	MongoDB +	Document store
6.	6.	6.	DB2 +	Relational DBMS
7.	7.	↑ 9.	Redis +	Key-value store
8.	8.	↑ 10.	Elasticsearch +	Search engine
9.	9.	↓ 7.	Microsoft Access	Relational DBMS
10.	10.	↓ 8.	Cassandra +	Wide column store
13.	↑ 14.	↑ 16.	Splunk	Search engine
14.	↓ 13.	↑ 18.	MariaDB +	Relational DBMS
22.	22.	22.	Amazon Redshift +	Relational DBMS
33.	33.	↑ 35.	Google BigQuery +	Relational DBMS
34.	34.	↓ 31.	Impala	Relational DBMS
35.	35.	↑ 37.	Spark SQL +	Relational DBMS
36.	36.	↑ 40.	InfluxDB +	Time Series DBMS
20.	20.	↓ 19.	SAP HANA +	Relational DBMS
21.	21.	↑ 22.	Amazon DynamoDB +	NoSQL
22.	22.	↓ 21.	Microsoft Azure Cosmos DB +	NoSQL

<https://db-engines.com/en/ranking> 디비랭킹 및 정보제공 에서 가지고온 현재
디비점유율이다.

여기서 대표적으로 사용되는 Oracle, MySQL, MS SQL server, PostgreSQL, MonogoDB, DB2, Redis, Elasticsearch, Cassandra 와 그 외의 Maria DB, Google BigQuery, Sparky SQL, Amazon DynamoDB(key-value , Document store) 를 조사하였다.

Oracle : 충분한 예산과 복잡한 비즈니스 요구와 기업고객을 위해 설계 관계형 디비

장점 :

Management Server : 중앙 집중 방식 관리자 모니터링 가능, Multiple databases 튜닝 가능
다른 어드민과 공유 가능

Change Manager : 변경 플랜을 만들면 실제로 구현하기전에 미리 변경사항의 효과 측정 가능
생산 시스템을 방해하지 않음

Alerts : 오류가 발생하면 사용자에게 알려줌(메일 등등)

Planning : 병복현상을 파악가능 , 사용패턴 추적(업그레이드된 관리자의 계획을 돕기위해)

Optimizer : 쿼리 최적화 프로그램, sql 문을 실행하는 효율적인 방법을 선택
cost 비용을 최소화하기위해 테이블과 인덱스를 분석

단점 :

조금 비싼 가격, 많은 오버헤드 사용(메모리사용량 업업),
복잡성 (설치하는과정만 하더라도 인스톨이외의 특별한 기술을 요구함)

결론 : 사용자가 거대한 인프라를 다룰 경우거나 혹은 오라클을 사용할만큼 재정적 뒷받침이
된다면 사용하게 될 DBMS 의 한 종류이다.

MySQL : 데이터베이스 기반 웹 사이트 및 가벼운 어플리케이션에 시용되는 디비
관계형디비

장점 :

타 DBMS (데이버베이스 매니지먼트 서비스) 보다 사용하기 쉽다.

phpadmin 같은 무료 gui 툴이 많다

매우 적은 오버헤드 사용 (메모리 사용량이 작음)

무료버전에서도 사용가능한 고급기능

설치가 간편하며 상대적으로 빠름

단점 :

고질적 버그, 관계형 테이블의 단점이 들어난 비유연성(상황에따라 데이터모델을 수정못함)

상용 디비엠에스에 비해 낮은 안정성,
낮은 퍼포먼스(동시성의 작업 요구될시 성능의 저하 발생)

오라클이 인수한뒤로 낮은 성장률 및 오픈소스로서의 가치 저하

기능이 추가애드온에 너무 의지함 (기본기능이 상대적으로 부실),

결론 : 스케일이 작거나 소규모 프로젝트일 경우 사용하면 좋다. 작은 쓰기읽기 작업에
괜찮다. 허나 오라클에 인수당한 뒤로 오픈소스로서의 가치를 잃어버려 Maria DB 로
사용자들이 이동하는 추세이다.

Maria DB : mysql 이 오라클에 인수된뒤 불확실한 라이선스 문제를 해결하기 위해 나온
디비엠에스 , mysql 의 장점을 그대로 계승하며 단점은 보완한 케이스 대부분 마리아 디비로
갈아타는 중이다. 관계형디비 BSD 라이선스를 지원한다.

장점 : 가볍고 성능이 괜찮다. 라이선스 문제에서 자유롭다.

단점 : 아직 안정성 성능이 다른 대규모 사용자 기반 시스템에 비해선 부족하다

결론 : Mysql 의 장점을 그대로 전승한 상태에서 단점을 보완하였다. 그리고 오픈소스이므로
유저들의 수많은 커뮤니티 활동이 활발하다.

Mongo DB : no sql 로 document 형이다.

장점 :

no sql인만큼 유연한 데이터베이스. 여러가지 데이터타입의 문서 소유 가능, 자유롭게
데이터를 저장할 수 있다.(데이터타입과 무관하게)

큰 데이터를 여러개의 서버에 나누어(어플리케이션을 통해 연결한 상태에서) 저장할 수 있다.
만약 서버가 기능하지 않더라도 다른 서버가 이를 대체해준다.

다큐먼트 기반형이므로 빠른 액세스가 가능하며 빠른 쿼리 응답속도를 보여준다. 관계형보다 100배빠른 속도를 뽐내고디비는 보여준다.

들어오는 데이터에 맞게 디비를 확장시키기 편하다. ad hoc query support 때문에 차후 들어올 쿼리에 대해 걱정하지 않아도된다(호환성) 또한 타 디비엠에스에 비해 인스톨작업이 간편하다.

단점 : 관계형 디비처럼 조인 기능을 지원하지 않는다. (수동으로 추가해야하며 느리다) , 높은 메모리 사용량 , 제한된 데이터사이즈 (다큐먼트 형이라 16메가바이트) 제한된 nesting

결론 : no sql 형인만큼 유연한 DB와 최근에 나온 DB인만큼 다양한 기능을 지원한다. 다만 조금 무거운 프로그램과 비관계형에서 나오는 단점들이 보인다.

PostgreSQL : 관계형디비 BSD 라이선스 (소스를 변경 후 숨긴채 재배포해도 문제없음) 주로 자사 솔루션과 포함되는 무료 디비에 사용됨

장점 :

자유로운 소스 수정 , 비용 절감(소프트웨어비용의 절감),

데이터베이스 를 관리하기 위한 데이터베이스 관리시스템=> 넓은 확장성,

유닉스도 지원하고 윈도우도 지원하는 크로스플랫폼

gui 인터페이스 지원, 쉬운접근성, 많은 데이터를 다룰 수 있게 설계됨

단점 :

작업하기에 비효율적인 아키텍처, 비효율적인 데이터 레플리케이션

테이블 충돌 이슈

새로운 버전 업그레이드에 대한 불편함

결론 : mysql보단 뛰어난 기능이 많고 다양한 애드온을 지원하지만 아직 속도면에선 느리다는게 업계의 정설이다. 개인과 기업의 업무에 있어서 속도가 그렇게 중요하지 않다면 이 디비엠에스를 사용하는것도 나쁘지는 않을 것 같다.

MS SQL server : 관계형 디비 , ms 에서 만든 sql 이며 ms 답게 윈도우에서도 실행이 잘되는 모습을 보여준다.

장점:

엔터프라이즈와 프로페셔널 레벨의 기능을 제공하며 타사의 제품보다 훨씬 발전된 형태이다(mysql)

또한 충돌된 데이터가 있거나 갑작스런 상황에 데이터가 증발되는 경우가 있다면 이를 완벽하게 복구해주는 기술을 제공해준다. 로그파일을 캐싱및 백업하여 복원해준다.

단점:

가장 큰 단점은 역시 라이선스 비용이다. 비싸다.

오로지 윈도우 베이스 서버에서 돌아가도록 설계되어있으므로 윈도우 이외의 다른환경에서 실행할때 오류가 발생하지 않는다는 보장이 없다.

다른 디비엠에스들은 다양한 플랫폼을 지원하는데 반해서보면 참 아이러니하다.

결론: 작업환경이 윈도우 일 경우 윈도우 베이스에 최적화된 ms sql server 를 쓰는것을 추천

DB2 : 관계형 디비 ibm 에서 개발하였으며 빅 플랫폼 중 하나다.

장점 :

강력한 쿼리 언어를 ms sql 보다 가지고 있다.

db2 는 객체 테이블을 지원한다.

sql을 통해 다이렉트로 인터페이스를 이용하여 데이터에 접근 가능하다.

다른 언어로 작성된 어플리케이션을 추가할수도 있다. 그만큼 유연하고 강력하다. ms sql 와 다르게 여러 플랫폼을 지원한다.

STMM(self tuning memory management) 기능을 지원하며 데이터베이스 스스로 워크로드에 맞게 메모리를 할당 및 최적화 해주는 기술이다.

단점 :

타사 디비엠에스에 비해서 느린 프로젝트의 진행도 느린 업데이트,

부족한 모니터링 툴

비싼 라이선스 비용

결론 : ms sql 와 대비되는 성능과 유틸리티성을 보여준다. stmm 등 다양한 기술들로 디비를 지원하는 DBMS 이다.

Redis : key-value 형이며 no sql type 이다. BSD 라이선스를 지원한다. 다양한 데이터타입을 지원한다.no sql 이다.

장점 :

넓은 공간(512mb)까지 키와밸류값 저장을 지원한다.

독자적인 레디스 해싱이란 메커니즘을 사용하여 빠른 접근성을 확보,

데이터 레플리케이션 또한 제공,

또한 ram 을 데이터 저장에 사용하므로(캐쉬메모리) 빠른 스피드를 제공, 이를 이용하여 쿼리 속도가 빠르다.

캐쉬 오브젝트에 대한 접근 또한 좋다.

단점 :

내장되어있는 아키텍처 모델에 대한 버그가 몇개 존재한다.

결론 : 대세인 no sql 타입이므로 유연한 디비를 자랑하며 key-value 인데 반해 넓은 저장공간 또한 제공해준다. 또한 캐쉬메모리를 사용하여 빠른 스피드를 자랑한다.

Elasticsearch : 아파치 기반으로 개발한 실시간 분산 검색 엔진(서버) 서칭엔진이며 JSon기반 데이터를 지원

장점 :

설치와 서버확장이 편하다.

분산 시스템이기에 검색대상의 용량이 증가했을때 대응하기가 수월하다.

소규모부터 빅데이터까지 광범위한 데이터 검색과 분석에 활용될 수 있음. 동작중에 죽은 노드를 감지하고 삭제하는 등 데이터가 안전하게 접근가능하도록 유지.

검색 엔진 답게 자유로운 검색 full text search 를 지원하며 다양한 플러그인 기능을 지원한다 빠르다

단점 :

하지만 사이즈가 큰 장문의 데이터 검색시 속도가 저하되는 현상이 발생된다.

결론 : 다양한 데이터입력을 지원하며 현재 대세인 빅데이터와도 맞물려 웹사이트등 다양한 데이터를 검색하고 이를 분석하는데 쓰일때 유용할 것이다.

Cassandra : 자유-오픈소스분산형 no-sql 이다 수많은 서버간의 대용량 데이터를 관리하기 위해 설계된 디비엠에스이다. 링구조를사용(원형으로 된 강강수월래 데이터구조)

장점 :

각 노드 장비들을 추가하거나 제거할때 자유로우며 또한 데이터센터까지 고려할수있는 데이터 복제 정책을 사용하여 안정성 측면에서도 많은 장점을 가지고 있다.

master-slave 나 sharding 같은 정책없이도 장애에 대한 기능이 가능하고 장비를 늘리고 줄이는데 비용이 큰 비용이 들지 않는다.

단점 :

join이나 transaction을 지원하지않고 인덱스등 검색을 위한 기능이 부족하다.

키스페이스나 데이터테이블을 과도하게 생성할경우 오버플로우가 발생할수도있다.

결론 : 노드 추가 제거같은 데이터 관리에는 적합하지만 다른 기능(조인 트랜잭션 검색) 같은 기능은 부족한 면을 보여준다.

Google BigQuery : 구글에서 개발한 서버리스 저비용 기업용 데이터 웨어하우스이다. 특징은 클라우드 서비스란점이다.

장점 :

다른 디비엠에스와 달리 구글 클라우드 서비스를 통해 제공되는 빅데이터 저장 분석 서비스이다.

클릭 몇번으로 설정 및 유지보수가 가능하다.

물론 기존의 디비엠에스에서 지원하는 sql 언어 또한 지원한다.

클라우드 스케일의 인프라를 이용하기때문에 대용량의 지원과 빠른 성능을 자랑한다.

클라우드 서비스답게 기존 디비엠에스의 과금정책이 아니라 쿼리시 사용되는 트랜잭션 비용만큼만 과금된다. AWS와 유사하다.

쉽고 운영이 필요없고 인프라에 대한 투자없이 클라우드서비스를 통해 이용할 수 있다.

단점 : 클라우드시스템에서 돌아간다는 점이 기존의 사용자들이 꺼려한다는 점외엔 없는 것 같다.

결론 : 클라우드시스템에서 돌아가는 디비엠에스답게 아주 적은 인프라와 적은 비용으로 막대한 퍼포먼스와 기능을 사용할 수 있다는게 가장 큰 메리트인것 같다.

Amazon DynamoDB(key-value , Document store) :

아마존에서 개발한 nosql 디비엠에스이다. 아마존웹서비스에서 이 디비를 사용하고 있는 상태이며 현재 상용세이다. 구글과 마찬가지로 클라우드 기반의 디비엠에스이다.

장점 :

hdd 기반이아니라 ssd 기반이라 속도가 빠르며 nosql 사용시 경험이 적은 부분을 aws 에서 자동 서포팅 해준다.

또한 확장 용량 등을 아마존 웹서비스를 통해 간단하게 관리할 수 있으며 넓은 사용자층을 바탕으로 다양한 레퍼런스가 존재한다.

클라우드 서비스답게 트랜잭션한만크만 과금이 된다.

단점:

쿼리는 지정한 프라이머리 키만 가능하며 데이터 타입 오로지 string number array 만 지원하는 점이다.

결론: 현재 다양한 스타트업기업이나 벤처기업 등 많은 사용자를 다루고 싶지만 회사 규모 자체가 크지 않는 회사들이 아마존웹서비스를 이용하여 서비스를 하고 있다. 소중기업입장에선 더할나위 없이 좋은 클라우드 서비스인것같다.

Spark SQL : 범용적 목적의 분산 고성능 클러스터링 플랫폼으로서 관계형

디비엠에스이다. 아파치기반의 머신러닝 라이브러리나 스트리밍 데이터 핸들링들을 제공한다.

장점 :

우선 빠른속도의 장점이 있으며 스파크의 큰 장점이라하면 기존의 빅데이터 분석 플랫폼과 비교했을때 훨씬 간단하고 명료하다는 점이다.

다른 플랫폼은 빅데이터분석과정에서 여러 프레임워크를 사용해야하지만 스파크는 스파크 내에서 로딩 정제 디비에 넣고 리포팅 등 일련의 작업이 하나에 가능하며 한플랫폼안에 배치,스트리밍,머신러닝등 다양한 처리기능을 제공하기때문에 하나의 데아타를 여러가지형태로 처리할수있는 장점이 존재.

또한 다양한 언어를 지원합니다.

단점 :

unstructured data 를 분석할때 있어 매우 제한적이며, 반강제로 structed 데이터를
사용해야하는
단점이 있다.

사용하기 위해선 전문적인 지식(spark 에대한)이 필요하다. (엔진에 더 친숙) 스파크 자체
엔진에 친숙해지다보면 다른 툴과 병합하거나 스와핑하면서 사용할때 불편함이 발생한다.

결론 : 빅데이터관련 분야에서 다른 플랫폼에 비해 훨씬 사용하기 간편하며 다양한 기능또한
제공하고 속도 또한 빠르다는 등 빅데이터 분석에 있어 유용한 디비엠에스이다.