

Web Development

Back-end

- users
- groups
- ACL
- SSO, OAuth, Authentication, Authorize, passport
- encrypt/decrypt
- ORM(file, db)
- routing
- socket
- rest
- push server
- board
- spatial
- token
- cookie, session
- middle ware
- file uploader
- parser
- pagination

Back-End Frameworks

- Django (Python)
- Ruby on Rails (Ruby)
- Flask (Python)
- Express (Node.js)
- Laravel (PHP)
- Spring (Java)

[public:computer:back-end-frameworks](#) · 2021/01/20 17:48 · alex

Fron-end

- templates
- components
 - head
 - foot
 - side
 - card
 - graph
- Chrome debugger
- Fiddler, Wireshark, Postman,...

Front-End Frameworks

- [Vue.js](#) [Vuetify.js](#) [Nuxt.js](#) [Vue.js](#)
- [React](#) [React](#) (+ [Redux](#))
- [MobX](#) ([React](#) + [MobX](#))
- [AngularJS](#) ([Angular 1](#))
- [Angular](#) (Formerly known as [Angular 2](#))
- [TypeScript](#) [TypeScript](#)
- [Ember](#)
- [Knockout](#)
- [Polymer](#)
- [Riot](#)

References

- [Vue.js와 다른 프레임워크와의 비교](#)
- [프론트엔드 성능 체크리스트](#)

[public:computer:front-end-frameworks](#) · 2022/04/28 15:29 · alex

Infrastructures

- [IaaS](#); [AWS](#), [Google Cloud](#), [Oracle Cloud](#)
- [PaaS](#); [heroku](#)

Databases

Database

RDBMS

- 데이터 [Data](#); 자료
- 테이블 [Table](#); 데이터를 표 형식으로 표현
- 데이터베이스 [Database](#); 테이블을 저장하는 공간 또는 테이블 집합
- [DBMS](#) [Database Management System](#); 데이터베이스들을 관리하는 소프트웨어
- 레코드 [Record](#) 또는 로우 [Row](#); 테이블의 행 = tuple
- 필드 [Field](#) 또는 컬럼 [Column](#); 테이블의 열
- 데이터 타입 [Data Type](#); 각 필드에 입력할 값의 타입(정수, 문자, 날짜 등)
- 필드 이름; 각 필드(열)의 이름
- 주 키 [Primary Key](#) 필드; 레코드를 식별하기 위한 유일한 값을 갖고 있으며 비어있지 않는 필드 (unique, not null)
- 외래 키 [Foreign Key](#) 필드; 다른 테이블의 주 키와 대응되는 필드
- [RDBMS](#) [Relational DBMS](#); 관계형 [DBMS](#)
- [SQL](#) [Structured Query Language](#); '구조화된 질의 언어'란 의미로 DB에서 정보를 얻거나, 생성하거나

나, 갱신하려고 정의한 표준 언어(규약)

SQL

- SHOW DATABASES;
- USE 데이터베이스이름;
- CREATE DATABASE 데이터베이스이름;
- DROP DATABASE 데이터베이스이름;
- SHOW TABLES;
- EXPLAIN 테이블이름; 또는 DESC 테이블이름;
- CREATE TABLE 테이블이름 (필드이름1 필드타입1, 필드이름2 필드타입2,);
- DROP TABLE 테이블이름;
- ALTER TABLE 옵션
- INSERT INTO 테이블이름 VALUES (값1, 값2,);
- DELETE FROM 테이블이름 WHERE 조건;
- UPDATE 테이블이름 SET 필드이름1=수정값1, 필드이름2=수정값2, ... WHERE 조건;
- SELECT 필드이름1, 필드이름2,... FROM 테이블이름 WHERE 조건;

References

- [\[MySQL\] Join 깔끔한 이해와 사용법](#)
- [SQL 문법 - SELECT문 및 함수](#)
- [\[MyBatis\] 동적 쿼리 foreach문 문법 총 정리](#)

Data Types

- VARCHAR(n)
- NVARCHAR(n)
- CHAR(n)
- NCHAR(n)
- INT
- FLOAT
- DATE
- TIME

NoSQL

- Key-Value

[public:computer:databases](#) · 2021/12/17 18:37 · alex

Server

- [linux 기초 사용법](#)
- CD/CI
- git; github, bitbucket, gitlab,

- docker
- Ansible / Chef / Puppet / terraform
- Kubernetes
- server monitorings
 - process
 - cpu
 - memory
 - network traffic
 - disk usages
 - logs
 - 장애 및 통계

Others

- backup Policy
- visual studio code
- vim
- regular expression
- git

From:
<http://dev.theta5912.net/> - **reth.dev**

Permanent link:
<http://dev.theta5912.net/doku.php?id=public:computer:web-development&rev=1629729516>

Last update: **2021/08/23 23:38**

