

데이터베이스응용 1강

하 석 재

CEO, 2HCUBE

sjha72@gmail.com

교과목정보

- 과목명 : 데이터베이스응용(온라인, 가반/나반)
- 강좌번호: 2150028101/2150028102
- 학점 : 3학점
- 비고 : 공학인증
- 선수과목: 데이터베이스

교과목 개요 및 목표

교과목 개요

- 관계형 데이터베이스의 활용 데이터베이스 설계와 실제 서비스 모델링 SQL, 상용 DBMS 도구 실습, 웹을 활용한 실습 및 과제

공학인증역량

- 이론 및 알고리즘의 검증 능력 문제해결을 위한 도구 활용 능력 설계 능력

출석규정

- 출석 1/3 이상 결석할 경우 자동적으로 F부여(교육부 가이드라인)
- 1학기동안 개인적인 사유로 결석 1회 인정
- 스마트캠퍼스출석 활용

평가기준

- 출석 : 10%
- 중간고사 : 35% -> 과제대체나 기말고사로 통합가능(별도공지)
- 기말고사 : 35%
- 과제 : 20%

수업일정

Week	Contents	
1	과정소개, 오픈소스 및 환경구성	
2	관계형 데이터베이스	
3	SQL - 기본 및 고급 질의	
4	JDBC	
5	인덱스와 트랜잭션(원자성)	
6	트랜잭션 고급(격리성/일관성/MVCC)	
7	MySQL Replication(복제) - DB이중화	
8	중간고사	
9	MySQL HA(고가용성)	
10	파티셔닝/백업	
11	샤딩	
12	MyBatis, Hibernate, JPA	
13	반정형데이터처리(JSON, NoSQL)	
14	비정형데이터처리	
15	기말고사	

오늘의 과제

- 실습환경구성 확인
 - 도커설치
 - 다음주 수업 전까지
 - sjha72@gmail.com로 설치화면 캡처해서 보내기

오픈소스

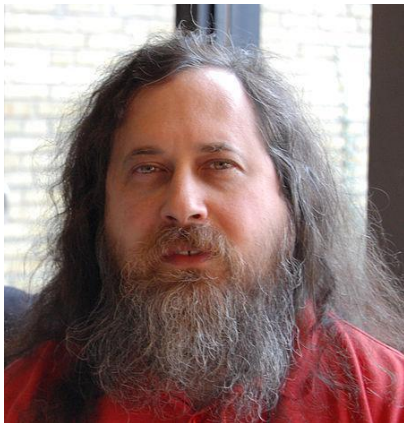


오픈소스의 이해

- 오픈소스는
 - 소스를 공개하고 라이선스로만 제약하는 방식
 - 무료이상의 개념
 - 개발자 커뮤니티에서 등장한 개념
 - 문제가 생기면 직접 해결할 수 있는 능력이 필요
 - 무료라 문제생기면 누구도 책임을 지지 않음
- 하지만, 프로그램 소스는 개발사의 가장 중용한 자산(IP)!
 - 이것을 무료로 한다니?
 - MS의 반응
 - 공산주의자 -> I love Linux ??

오픈소스의 시작

- 리처드 스톨만
 - GCC, Emacs
 - GPL라이선스
- 라이너스 토발즈
 - Linux
 - Git



왜 오픈소스인가?

- 오픈소스의 장점
 - 최고수준의 프로그래밍 소스가 무료
 - 능력이 있으면 직접 기여 가능, 단 능력있어야 인정해줌
- 오픈소스의 단점
 - 많이 쓰는 오픈소스를 사용해야
 - 문제 생기면 직접 해결
- 특징
 - 잘 나가는 오픈소스는 계속 파생 오픈소스가 생김(Fork)
 - 잘 안나가는 오픈소스는 업그레이드 정지
 - **업데이트가 일어나면 기존 오픈소스는 지원 종료**
 - 하이퍼렛저 패브릭 1.1(2018.3.20)이 나오면서
 - 1.0의 지원종료
 - 지속적인 Follow-up 필요하면?

성당과 시장 모델

- 성당(Cathedral)과 시장(Market) 모델
 - 소수의 개발자가 개발하는 모델 vs. 다수의 개발자가 개발하는 모델

오픈소스의 에코시스템(**Eco-system**)

- 돈을 내고 유지보수해주는 업체 생김
 - 무료버전과 유료버전
 - **Community Edition vs. Enterprise Edition**
 - cf. Docker, MySQL, ...
- **Edge / Stable / LTS(Long-term Support)** -> Ubuntu/Linux 18.04(LTS) vs. 19.10
 - 최신기술 테스트(Edge)
 - 어느정도 사용하는 데 문제는 없는 버전(Stable)
 - 새 기술과 관계없이 지속적인 지원(LTS, 보통 3-5년)
 - MySQL에서는 **GA(Generally Available)**라고 부름
 - cf. Python 2.7 vs. 3.8
- 알파 -> 베타 -> RC(Release Candidate) -> 정식
- -> Service Pack(RedStone)
 - Microsoft

오픈소스의 에코시스템(**Eco-system**)

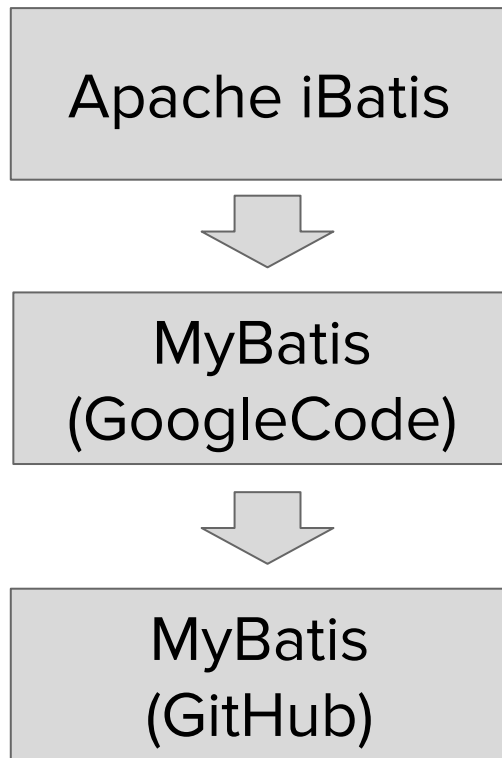
- 개발자는 어떤 장점이?
 - 별도의 보상을 받지 않음
 - 다른 일하면서 남는 시간에 기여
 - 회사에 소속되서 월급받으면서 개발
 - 해당 오픈소스가 유명해지면 지명도가 생김
 - 몸값이 높아짐
 - 리처드 스톨만 / 라이너스 토발스 / 더그 커팅 / ...
 - 해당 분야의 Guru가 됨

- > 오픈소스는 회사가 원하는 기술

오픈소스 프로젝트 현황

- 아파치 재단(Apache Foundation)
 - <http://www.apache.org>
 - 아파치(웹 서버)
 - 자카르타 프로젝트(Jakarta Project)
 - 톰캣(Tomcat), 메이븐(Maven), HBase, 하둡(Hadoop), 스파크(Spark), 루씬(Lucene), 주키퍼(Zookeeper), Kafka, ...
- 구글코드(GoogleCode)
 - 현재는 서비스 접음
- 깃허브(GitHub)
 - <http://github.com>
 - 마이바티스(MyBatis), Angular.js/구글, 텐서플로우(Tensorflow)/구글
 - 도커(Docker), D3.js, Node.js, JQuery.js, React.js
 - 스프링(Spring Framework)/Pivotal, 엘라스틱서치(Elasticsearch)/Elastic, 웹킷(Webkit)/Apple, v8/Google, blink/Google, ...

세 진영을 모두 거친 프로젝트



오픈소스 현황

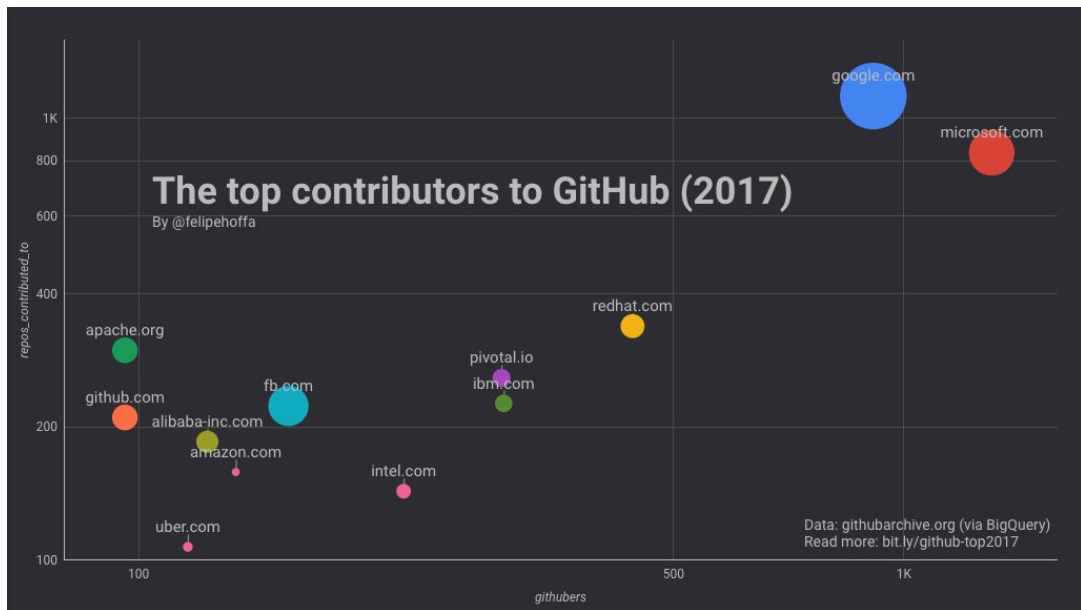
- **4차산업혁명의 주요분야**
 - 딥러닝(AI), 빅데이터, 블록체인 등에서 오픈소스가 주도
- **S/W 대기업의 적극적 참여및 경쟁**
 - 주도권 경쟁
 - 구글 vs. 페이스북
 - Tensorflow vs. Caffe2, PyTorch
 - Angular.js vs. React.js
 - Protocol Buffer vs. Thrift
 - MS(Github 인수)
 - Visual Studio Code, CNTK, Typescript
 - IBM
 - Hyperledger
 - Apple
 - WebKit

오픈소스 현황

- 인기있는 오픈소스프로젝트의 적극적 스피노프
 - 도커 -> 도커 컴포즈 -> Kubernetes(Google)
- 여러 개의 오픈소스를 사용하는 복잡한 오픈소스 등장
 - IBM의 프라이빗 블럭체인 하이퍼렛저(Hyperledger) Fabric
 - 컨테이너기반 가상화
 - Docker/ swarm/Kubernetes
 - 스마트 컨트랙트 Go/Node.js/Java
 - 자동화 쉘스크립트+도커+파이썬
 - 복제/저장 CouchDB(NoSQL)
 - 분산프로그래밍 gRPC+Protocol Buffer
 - 합의/가용성 Kafka+Zookeeper

오픈소스 기여정도

- Google > MS > Redhat > Apache.org > Pivotal > Facebook > IBM > Alibaba > Amazon > IBM > Intel > Uber



<https://medium.freecodecamp.org/the-top-contributors-to-github-2017-be98ab854e87>

오픈소스 기여정도

- 웹브라우저
 - Chrome/IE, IE -> Edge
- 오픈소스
 - Chromium -> Chrome
 - 동영상코덱(유료 라이선스/H.264, H.265) + 버그패치
- Edge -> Chromium 기반 Edge(2020.1)

-> 여러 회사가 같이 만들고 결과를 공유

오픈소스 기여정도

- Visual Studio(상용) vs. VSCode(에디터, 오픈소스)
- Visual Studio vs VSCode
 - IDE vs. Editor+컴파일러/인터프리터
- VSCode(3rd Party 플러그인)

IT 대기업의 관심도

- 구글
 - Tensorflow(Deep Learning), Angular(Front-end Framework), Kubernetes(Large-scale Virtualization), Android
- MS
 - VSCode, Electron.js(Cross platform platform), Node.js, Typescript, Docker, CNTK(Deep Learning)
- 아마존
 - Alexa(Deep Learning), AWS(Cloud)
- 레드햇(IBM에 인수)
 - Docker, Kubernetes, Eclipse, Hibernate
- IBM
 - istio(microservice), hyperledger fabric(private blockchain), open java

IT 대기업의 관심도

- 피보탈
 - Spring Framework
- 인텔
 - openCV(computer vision), BigDL(Deep Learning), sawtooth(blockchain/IoT)
- 페이스북
 - React.js(Front-end framework), atom(Editor), caffe2(Deep learning)
- 오라클
 - MySQL, Truffle Ruby

오픈소스에 대한 대응

- 오픈소스없이 프로그램 개발이 가능한가?
 - 웹 프로그래밍(서버/클라이언트)
 - 빅데이터, 딥러닝, 블록체인, ...
 - 버전컨트롤/이슈트래킹/Wiki/CI
 - OS/DBMS/네트워크/ ...
- 오픈소스에 대한 이해와 활용을 잘 해야 한다
 - 라이선스의 “정확”한 이해
 - 오픈소스가 무조건 공짜가 아님
 - 필요한 소프트웨어를 찾아 잘 활용해야 함
- 오픈소스 교육의 심화
 - 기업들의 요구에 대한 대응 / 취업연계

실습준비사항



실습환경 및 대상

- 도커기반의 MySQL
- <https://launchpad.net/test-db> 의 샘플 데이터베이스

도커 설치

- 도커 툴박스(버추얼박스기반) vs. 도커 데스크탑(Hyper-V 기반)
 - 윈도우 라이선스 무관 / 윈도우10 프로,엔터프라이즈 에디션

1. 구글에서 '**docker toolbox releases**' 입력
2. '**Releases**' 클릭
3. **assets -> 19.03.1-ce.exe** 다운로드하고 설치
4. '**Docker Quickstart Terminal**' 실행
5. **\$ 프롬프트**가 나오면 준비완료

도커 설치

- Quickstart Terminal 실행시 주의사항
 - CPU 가상화기능 요구(vt-x나 amd-v 요구)
 - Pentium/Celeron CPU를 사용하는 컴퓨터에서 실행되지 않음
 - 바이오스에서 (CPI virtualization) enabled 되어야 함
 - 버추얼박스나 Hyper-V/VMware와의 충돌가능성있음
 - 다른 SW는 삭제
- 트러블슈팅 방법
 - 1. 버추얼박스만 업데이트(virtualbox.org)해서 실행
 - 2. 먼저 도커와 버추얼박스 다지우고 리부팅한 후
도커만 설치해서 실행

도커 설치(리눅스)

- 버추얼박스 다운로드(virtualbox.org)
- 우분투 서버 다운로드(ubuntu.com)
 - .iso 파일 다운로드
- 가상머신 생성
 - Linux/Ubuntu 64bit
 - RAM 2G이상, 하드디스크 20GB이상 지정
 - 설정 - 저장소 - CD롬 아이콘 클릭(비어있음) - 오른쪽 CD롬 아이콘 클릭 - 다운받은 ISO 파일 지정
 - '시작' 버튼 클릭

도커 설치(리눅스)

- <https://vincetory.com/27> 참조
- 초반에 모두 엔터 -> 서버이름/아이디/비밀번호 모두 'ubuntu'로 설정
- 디스크 삭제 경고 - 'continue'로 변경 후
- 모두 탭을 눌러 'Done' 선택하면서 설치진행
- Reboot가 나올때까지 진행해서 리부팅한 후 로그인

- 도커설치

```
$ sudo apt update
```

```
$ sudo apt install docker.io
```

```
$ sudo apt install docker-compose
```

감사합니다

