Google Developer Student Clubs

Backend 4th study

Google Developer Student Clubs

Contents

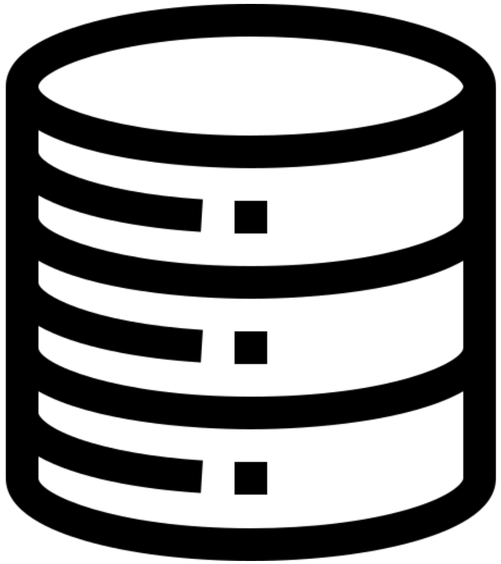
1. 데이터베이스

2. SQL

Google Developer Student Clubs

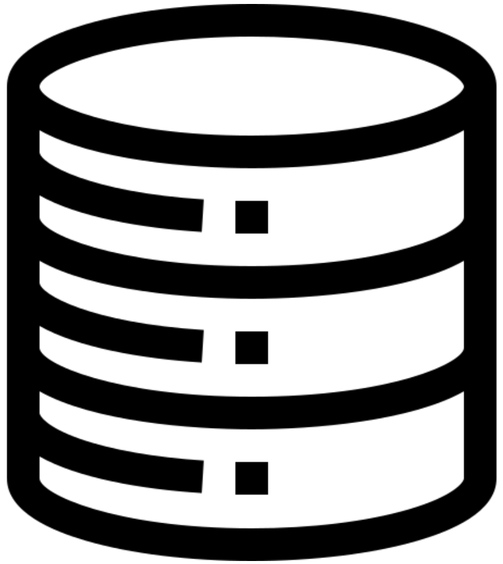
데이터베이스

데이터베이스



서버는 데이터베이스에 접근하여 데이터들을 Read/Write

데이터베이스



DATABASE

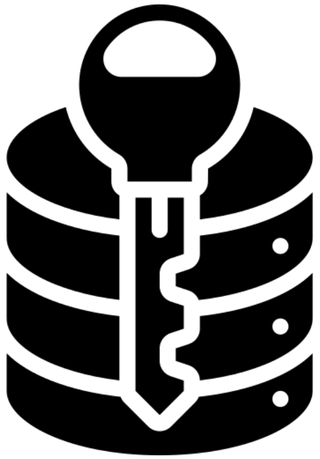
체계화된 데이터의 집합체

중복된 데이터를 없애고, 자료를 구조화하는 효율적인 처리

DBMS : 데이터베이스를 관리하는 미들웨어

SQL 사용 O -> RDB
SQL 사용 X -> NOSQL

데이터베이스



RDB

Key - Value들의 간단한 관계를 테이블화 시킨 데이터베이스

Id	Name	Email	Part
1	이제준	dlwpwns01@naver.com	server
2	박세열	parkseyeol@naver.com	server

데이터베이스

RDB
Relational Database

table

Row

Id	Name	Email	Part
1	이제준	dlwpwns01@naver.com	server
2	박세열	parkseyeol@naver.com	server

Column

데이터베이스

RDB
Relational Database

Primary Key
기본키

Id	Name
1	이제준
2	박세열

다른 데이터와 구별할 수 있는 식별자

한 테이블에는 하나 혹은 그 이상의 primary key가 있어야 함

Primary key는 Not Null이며 Unique한 값이어야 함

데이터베이스

RDB
Relational Database

Index
인덱스

Id	Name
1	이제준
2	박세열

데이터의 색인 기능

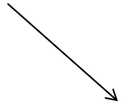
관계형 데이터베이스에서 검색 속도를 높이기 위한 도구

index는 Unique하면서 특정 데이터를 대표하는 데 사용
-> index를 주로 Primary Key로 사용

데이터베이스

RDB
Relational Database

Id	Name
1	이제준
2	박세열



Id	user_id	title
1	2	First_title
2	1	Second_title

Foreign Key
외래키

한 테이블의 Column 중

다른 테이블의 row를 식별할 수 있는 key

"Relational" Database

테이블과 테이블 간의 관계가 형성되어 있는 데이터 베이스

"Relational" Database

1:1 관계

1:N 관계

N:M 관계

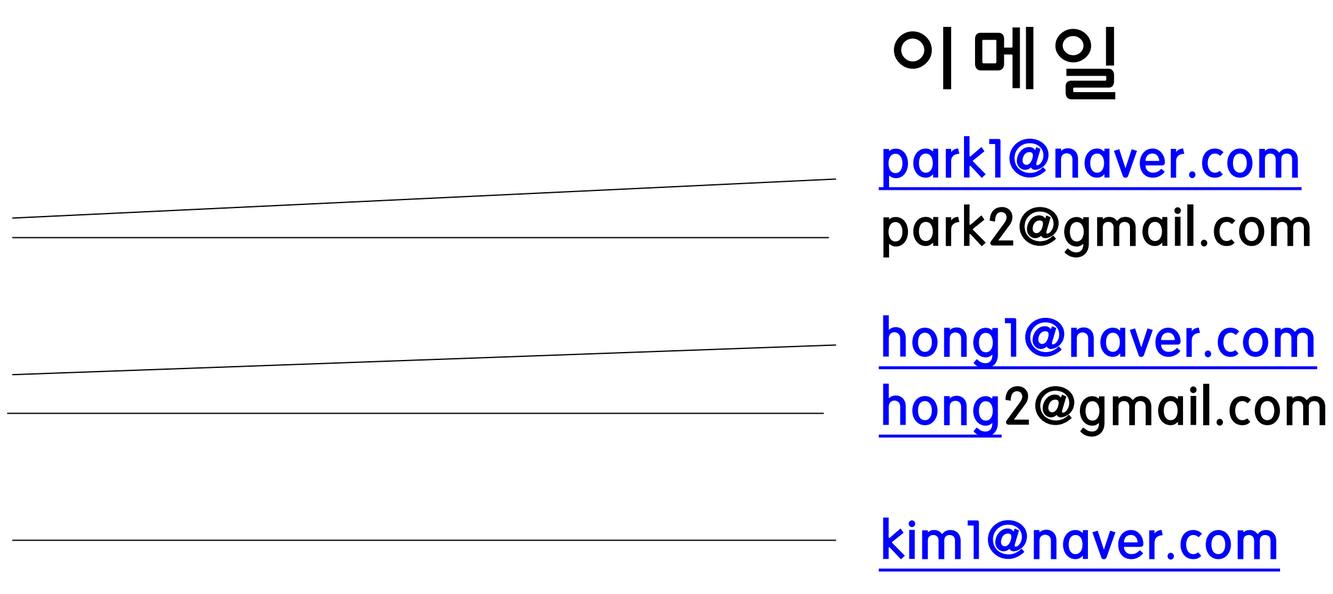
1:1 관계

이름		주민등록번호
박세열		123456-xxxxxxx
홍승현		123478-xxxxxxx
김유진		123490-xxxxxxx

1:1 관계

Id	Name	ssn
1	박세열	123456-xxxxxxx
2	홍승현	123478-xxxxxxx
3	김유진	123490-xxxxxxx

1:N 관계


이름
이메일
박세열
홍승현
김유진
park1@naver.com
park2@gmail.com
hong1@naver.com
hong2@gmail.com
kim1@naver.com

1:N 관계

Id	Name	email
1	박세열	park1@naver.com, park@gmail.com
2	홍승현	hong1@naver.com, hong2@gmail.com
3	김유진	kim1@naver.com

데이터베이스

1:N 관계

Id	Name	email
1	박세열	park1@naver.com,
2	박세열	park2@gmail.com
3	홍승현	hong1@naver.com
4	홍승현	hong2@naver.com
5	김유진	kim1@naver.com

이름중복발생

이거 괜찮을까..?

데이터베이스

1:N 관계

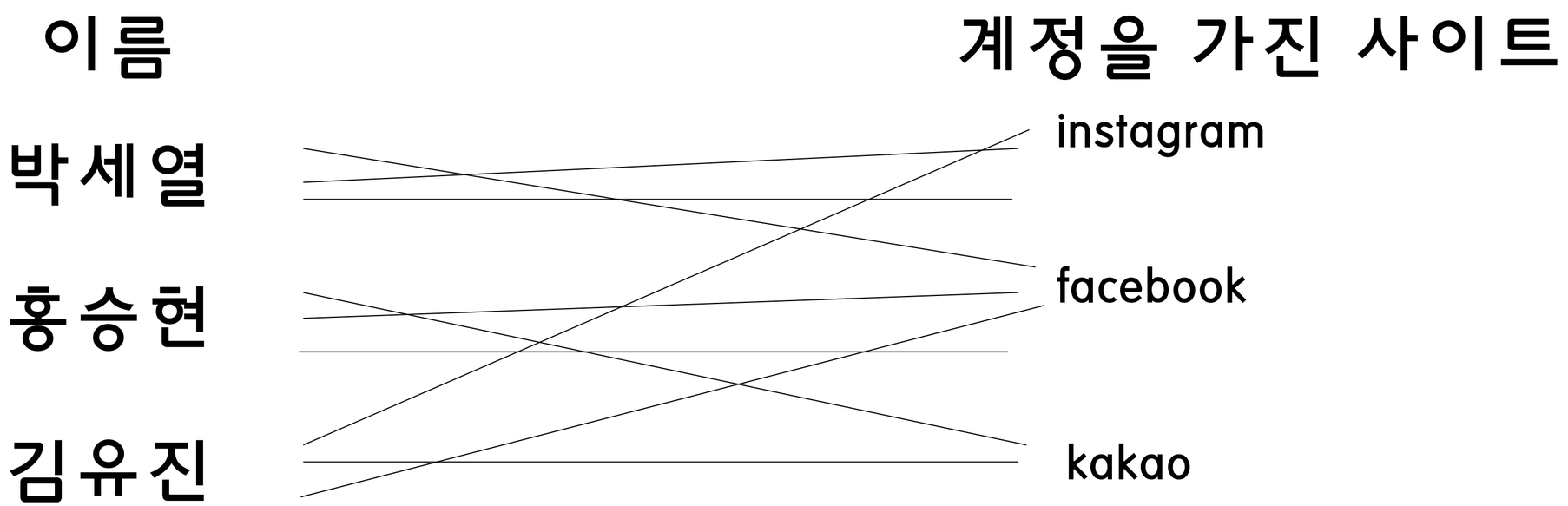
새로운 테이블!

Id	Name
1	박세열
2	홍승현
3	김유진

Id	user_id	email
1	1	park1@naver.com
2	1	park2@gmail.com
3	2	hong1@naver.com
4	2	hong2@gmail.com
5	3	kim1@naver.com

왜 분리 해야되는지는
정규화, 역정규화 때!

N:M 관계



데이터베이스

N:M 관계

Id	Name	site
1	박세열	Google,Instagram,Naver
2	홍승현	Instagram,Naver,Kakao
3	김유진	Instgram,AWS

이거 괜찮을까..?

데이터베이스

N:M 관계

Id	Name	site
1	박세열	Google
2	박세열	Instagram
3	박세열	Naver
4	홍승현	Instagram
5	홍승현	Naver
6	홍승현	Kakao
7	김유진	Instgram
8	김유진	AWS

이름과 site 둘 다 중복..
이거 괜찮을까..?

데이터베이스

N:M 관계

새로운 테이블!

Id	Name
1	박세열
2	홍승현
3	김유진

Id	User_id	Site_id
1	1	1
2	1	2
3	1	3
4	2	2
5	2	3
6	2	4
7	3	2
8	3	5

Id	Name	uri
1	Google	Google.com
2	Instagram	Instagram.com
3	naver	Naver.com
4	Kakao	Kakao.com
5	AWS	Aws.com

데이터베이스

정규화를 통해 데이터베이스를 설계해보아요!

정규화

"RDB를 논리적이고 직관적으로 만드는 과정"

불필요한 데이터 제거 -> **데이터의 중복 최소화**

데이터를 다루면서 생기는 **이상현상 방지**

개발 중 데이터의 변화가 생겨도 **설계를 재구성할 필요성 감소**

데이터베이스

정규화
이상현상?

삽입이상 : 새 데이터를 삽입하기 위해 불필요한 데이터도 함께 삽입해야 하는 문제

갱신이상 : 중복되는 데이터 중 일부만 변경하여 데이터가 불일치하게 되는 모순의 문제

삭제이상 : 데이터를 삭제하면 꼭 필요한 데이터까지 함께 삭제되는 데이터 손실의 문제

정규화

1차 정규화

2차 정규화

3차 정규화

...

1차 정규화

컬럼은 원자 값을 가져야한다.

Row마다 Column의 값이 1개씩만 있어야 한다는 뜻

Id	Name	email
1	박세열	park1@naver.com, park@gmail.com
2	홍승현	hong1@naver.com, hong2@gmail.com
3	김유진	kim1@naver.com

원자값이 아님
1차 정규화 만족하지 못함

1차 정규화

컬럼은 원자 값을 가져야한다.

Row마다 Column의 값이 1개씩만 있어야 한다는 뜻

Id	Name	email
1	박세열	park1@naver.com,
2	박세열	park2@gmail.com
3	홍승현	hong1@naver.com
4	홍승현	hong2@naver.com
5	김유진	kim1@naver.com

데이터베이스

예제 – 1차 정규화 후

삭제이상

번호가 1번인 학생이 수강신청을 취소하게 되면 과목 코드가 111인 DB의 담당 교수가 이제준 교수님이라는 사실까지 삭제됨

번호	학과	학생 이름	학년	과목코드	과목명	담당교수
1	소프트웨어학과	홍승현	2	111	DB	이제준
2	경제학과	박세열	4	222	경제원론	김동원
1	소프트웨어학과	김유진	3	111	DB	이제준
2	경제학과	서준원	3	333	객지프	안재욱
3	철학과	이상현	4	444	알고리즘	황솔희
3	철학과	이상현	4	111	DB	이제준

삽입이상
또 다른 교수인 김동원 교수가 담당한 과목을 학생들이 신청하지 않는 한, 김동원 교수가 삽입될 수 없음!

갱신이상
이상현 학생이 전과하게 되면 2개의 학과 튜플을 모두 갱신시켜야함

데이터베이스

2차 정규화

모든 컬럼은 완전함수적 종속을 만족해야한다.

기본키중에 특정 컬럼에만 종속된 컬럼(부분적 종속)이 없어야 한다.

번호	학과	학생 이름	학년	과목코드	과목명	담당교수	학점
1	소프트웨어학과	홍승현	2	111	DB	이제준	A
2	경제학과	박세열	4	222	경제원론	김동원	B
1	소프트웨어학과	김유진	3	111	DB	이제준	C
2	경제학과	서준원	3	333	객지프	안재욱	B
3	철학과	이상현	4	444	알고리즘	황솔희	B
3	철학과	이상현	4	111	DB	이제준	C

데이터베이스

2차 정규화

모든 컬럼은 완전함수적 종속을 만족해야한다.

기본키중에 특정 컬럼에만 종속된 컬럼(부분적 종속)이 없어야 한다.

학생 테이블

번호	학과	학생 이름	학년
1	소프트웨어학과	홍승현	2
2	경제학과	박세열	4
3	소프트웨어학과	김유진	3
4	경제학과	서준원	3
5	철학과	이상현	4

수강과목 테이블

과목코드	과목명	학과	담당교수
111	DB	소프트웨어학과	이제준
222	경제원론	경제학과	김동원
333	객지프	소프트웨어학과	안재욱
444	알고리즘	소프트웨어학과	황솔희

성적 테이블

번호	과목 코드	학점
1	111	A
2	222	B
3	111	C
4	333	B
5	111	C
5	444	B

데이터베이스

2차 정규화

2차 정규화 후 수강과목 테이블에서 삽입이상 발생!

삽입이상
또 다른 교수인 이상현교수가
개설한 과목이 없을 때 이상현
교수는 삽입될 수 없음

삭제이상
김동원 교수의 경제원론 강의가
없어진다면 김동원 교수의 정보가 사라짐

갱신이상
이제준 교수의 소속학과가 변경된다면,
이제준 교수에 해당하는 학과에 대한 모든
속성값을 바꿔줘야 함.

수강과목 테이블

과목코드	과목명	학과	담당교수
111	DB	소프트웨어학과	이제준
222	경제원론	경제학과	김동원
333	객지프	소프트웨어학과	안재욱
444	알고리즘	소프트웨어학과	황솔희
555	미적분	수학과	이제준

데이터베이스

3차 정규화

"이행적 함수 종속이 없어야 한다"
기본키 외 Column이 다른 Column을 결정할 수 없다는 뜻.

과목코드	과목명	학과	담당교수
111	DB	소프트웨어학과	이제준
222	경제원론	경제학과	김동원
333	객지프	소프트웨어학과	안재욱
444	알고리즘	소프트웨어학과	황솔희
555	미적분	수학과	이제준

현재 과목코드가 교수를 결정.
또한, 교수가 학과를 결정

데이터베이스

3차 정규화

3차 정규화 끝!

과목코드	과목명	담당교수
111	DB	이제준
222	경제원론	김동원
333	객지프	안재욱
444	알고리즘	황솔희
555	미적분	이제준

학과	담당교수
소프트웨어학과	이제준
경제학과	김동원
소프트웨어학과	안재욱
소프트웨어학과	황솔희
수학과	이제준

데이터베이스

역 정규화

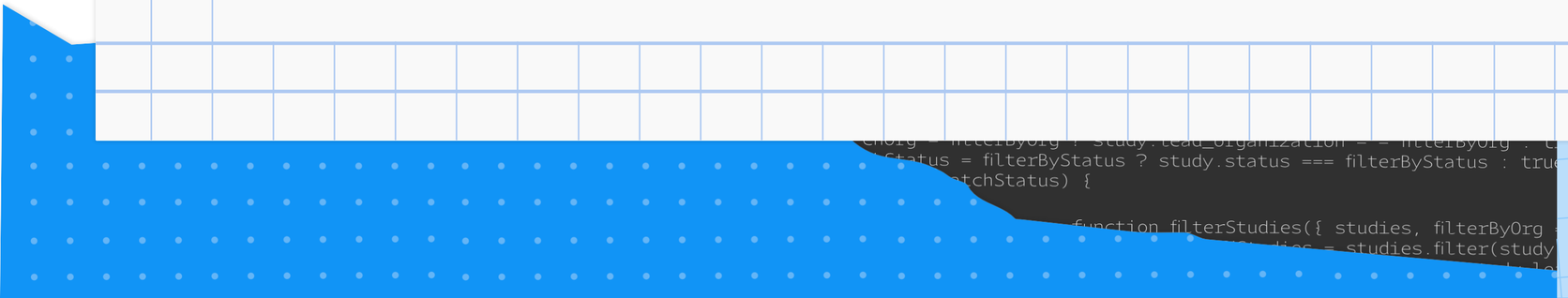
정규화된 테이블에 중복을 허용하고, 다시
통합하거나 분할하여 구조를 재조정하는 과정

정규화된 모델은 저장된 자료를 검색하는 시간을
증가시키기 때문에 성능을 저하

물리적 설계 과정에서 성능을 향상시키기 위해
역정규화를 실행

Google Developer Student Clubs

SQL



SQL

DB에서 테이블을 처리하는 방식

데이터 생성 CREATE -> INSERT
데이터 조회 READ -> SELECT
데이터 수정 UPDATE -> UPDATE
데이터 삭제 DELETE -> DELETE

이외에도 조회 중 join, where, order by 등 중요
-> 이건 한 번 찾아보세요! 중요합니다

SQL

SQL

INSERT

INSERT INTO 테이블 (컬럼1, 컬럼2) VALUES (값1, 값2)
INSERT INTO 테이블 VALUES (값1, 값2)
-- 컬럼 이름을 생략할 시, 테이블 컬럼 개수 및 위치가
-- VALUES와 일치해야함.

UPDATE

UPDATE 테이블 SET 컬럼1 = 값2, where = 조건;

DELETE

DELETE FROM 테이블 where 조건

SQL

SQL

SELECT

SELECT 컬럼1, 컬럼2, 컬럼3 FROM 테이블 where 조건;

SELECT DISTINCT 컬럼1 FROM 테이블 where 조건;

SELECT 컬럼1, 컬럼2, 컬럼3 FROM 테이블 where 조건
ORDER BY 정렬_기준이_될_컬럼 [ASC/DESC];
-- ASC 오름차순, DESC 내림차순

SELECT * FROM A_table
JOIN B_table ON A_table.동일_컬럼 = B_table.동일_컬럼
Where 조건;

그 밖에도 COUNT, SUM, GROUP BY도 있으니 한 번 찾아보시면 좋아요!

과제