



2-2 : 데이터베이스, ERD, SQL 개요

들어가기 전에



이번 시간에는...

데이터베이스

RDB

ERD

SQL

데이터베이스(DB)

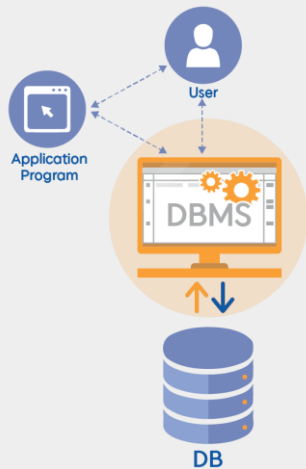
데이터베이스는 여러 사람이 공유하여 사용할 목적으로
체계화, 통합, 관리하는 **데이터의 집합**
데이터베이스 관리 시스템(DBMS)에 의해 제어된다



데이터베이스 관리 시스템(DBMS)

데이터베이스(DB)에 접근하여 **데이터베이스 관리를 지원**하는 소프트웨어

종류에는 MySQL, MariaDB, PostgreSQL, Oracle 등
유형에는 크게 관계형, 비관계형



관계형 데이터베이스



관계형 데이터베이스(RDBMS)

테이블 형태로 데이터를 관리
테이블 간의 **관계**를 매핑

컬럼 (Column) = 필드 (Field) = 속성 (Attribute)

번호	이름	아이디
1	Dave Lee	FunCoding
2	David	Honny
3	Andy	beyond

로우 (Row) = 레코드 (Record) = 튜플 (Tuple)

기본키 (Primary Key)				외부키 (Foreign Key)			기본키 (Primary Key)		
번호	이름	아이디	반				반	담당선생님	학생수
1	Dave Lee	FunCoding	1-1				1-1	Kate	10
2	David	Honny	1-2				1-2	Willy	20
3	Andy	beyond	1-3				1-3	Brown	30

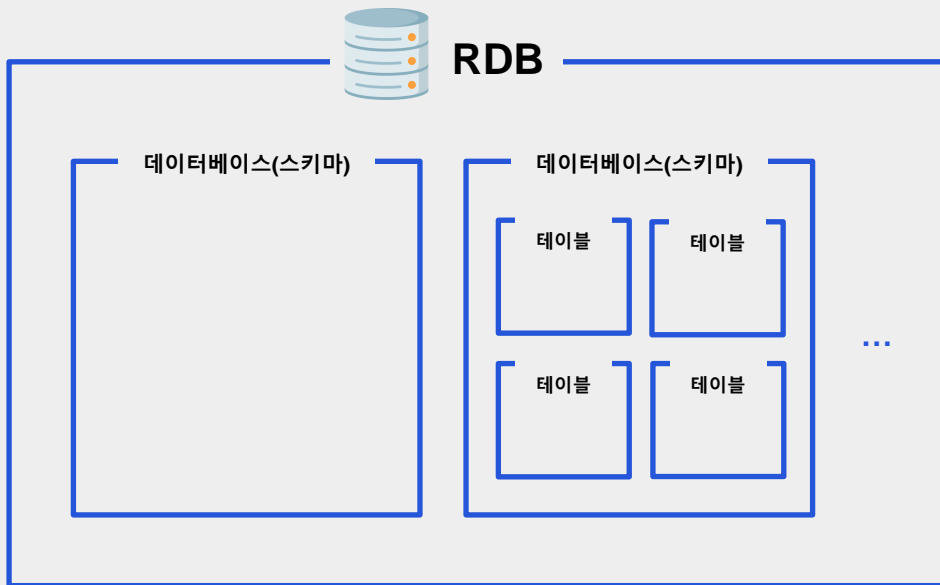
student 테이블

class 테이블

관계형 데이터베이스

관계형 데이터베이스 구조

RDB 서버에 여러 개의 데이터베이스
각 데이터베이스에 여러 개의 테이블이 있고, 관계를 맺고 있다



컬럼 (Column) = 필드 (Field) = 속성 (Attribute)

번호	이름	아이디
1	Dave Lee	FunCoding
2	David	Honny
3	Andy	beyond

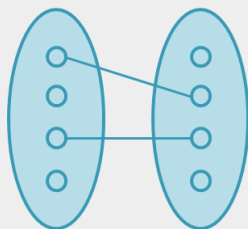
로우 (Row) = 레코드 (Record) = 튜플 (Tuple)

관계형 데이터베이스

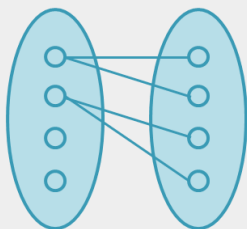


관계형 데이터베이스 관계

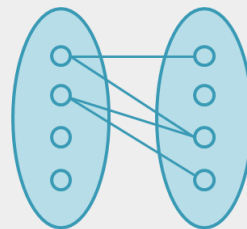
RDB의 관계에는 1:1, 1:N, N:M 관계가 있다.



일대일(one-to-one)



일대다(one-to-many)



다대다(many-to-many)





관계형 데이터베이스

관계형 데이터베이스 관계

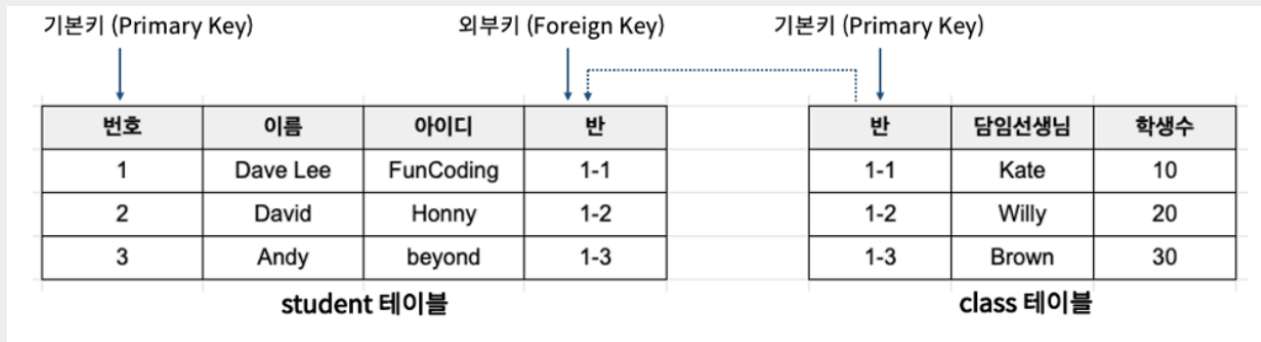
다음 예시의 관계를 한 번 생각해보자

- 학생과 사물함
- 작가와 책
- 게시물과 리뷰
- 게시물과 사진

관계형 데이터베이스

관계형 데이터베이스 키(Key)

테이블의 행을 식별할 수 있게 하는 속성
ex) 기본키, 외래키

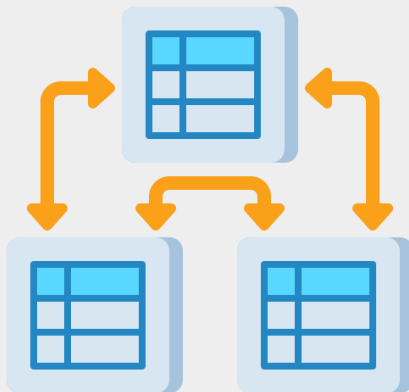


기본키(Primary Key): 튜플을 유일하게 구별할 수 있게 하는 속성(중복X, null X)

외래키(Foreign Key): 다른 테이블의 기본키를 참조하는 속성

관계형 데이터 모델링

관계형 데이터베이스는 정형화된 구조를 갖는다
테이블, 테이블 간의 관계를 설계하는 작업



개념적 데이터 모델링

현실의 업무를 분해하여, 데이터 상의 개념을 뽑아내는 일



논리적 데이터 모델링

만들어진 개념적 데이터 설계를 관계형 DB에 맞게 설계



물리적 데이터 모델링

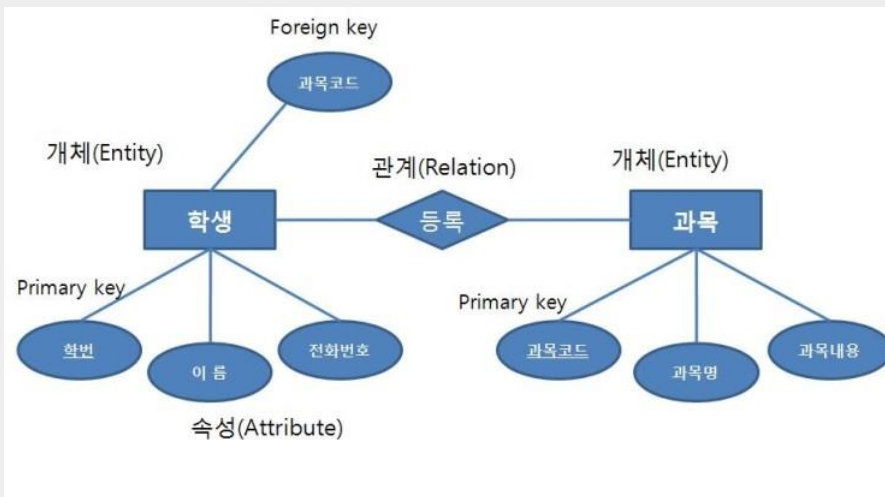
설계된 데이터를 바탕으로 실제 테이블을 생성하고 성능 향상 고려
서비스 운영 후 점진적으로 수정해 나가는 작업

모델링 과정

개념적 데이터 모델링

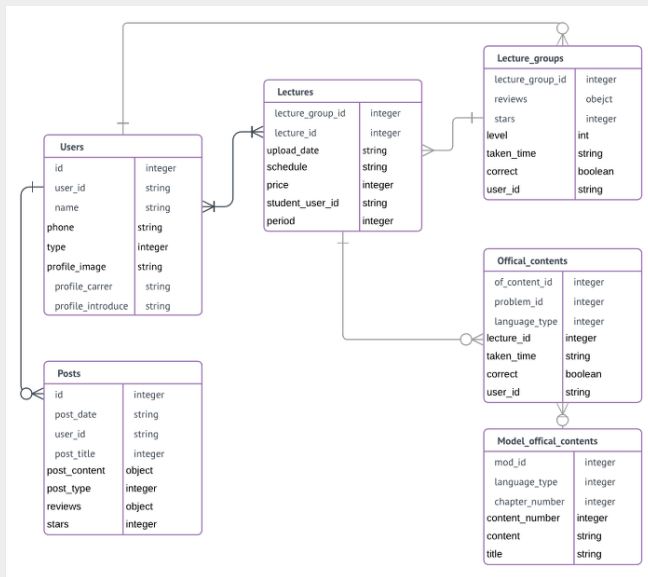
현실의 업무를 분해하여, 데이터 상의 개념을 뽑아내는 일
산출물: ER 다이어그램

기 호	의 미
	개체
	속성
	기본키
	관계
	개체 타입과 속성을 연결
	개체간의 관계 타입



논리적 데이터 모델링

만들어진 개념적 데이터 설계를 관계형 DB에 맞게 설계
산출물: 테이블 관계도



물리적 데이터 모델링

설계된 데이터를 바탕으로 실제 테이블을 생성하고 성능 향상 고려
서비스 운영 후 점진적으로 수정해 나가는 작업

ex) 인덱스, 캐시, 역정규화

산출물: SQL 쿼리문

```
1  -- 데이터의 삽입, 수정, 삭제
2  SELECT * FROM emp;
3  INSERT INTO emp VALUES (7369, 'SMITH', 'CLERK', 7902, TO_DATE('80/12/17'), 800, NULL, 20);
4  SELECT * FROM dept;
5  INSERT INTO dept (deptno, dname) VALUES (10, 'ACCOUNTING');
6  INSERT INTO dept2 SELECT * FROM dept;
7  UPDATE emp SET deptno=20 WHERE empno=7902;
8  UPDATE emp SET sal=sal*1.1 WHERE deptno=20;
9  UPDATE emp SET hiredate=SYSDATE;
10 select hiredate from emp;
11 DELETE FROM emp WHERE empno=7902;
12 DELETE FROM emp WHERE sal<(SELECT AVG(sal) FROM emp);
13 DELETE FROM emp;
14 select * from emp;
15 rollback; --연습 후 rollback으로 데이터 원상복구
16 select * from emp; -- 재확인
```

모델링 예시



모델링 예제

유튜브 예시를 통한 모델링 과정 익히기

검색

전체 게임 음악 실시간 요리 축구 액션 어드벤처 게임 만화 영화 최근에 업로드된 동영상

"의원님들 양해 바랍니다" 간담회 중 뛰쳐나간 대원들
MBCNEWS
조회수 12만회 • 39분 전

90년대 인기발라드 모음 피아노 연속듣기
지니도비지영 Jinido bgm
조회수 33만회 • 8개월 전

[22/23 세리에A 14R] 나폴리 vs 엠틀리 김민재 활약상
SPOTV
조회수 10만회 • 1시간 전

[골라봐이지] "왜 이렇게 진지해?" 중이의 집 도요량 전허 탄...
JTBC Voyage
조회수 142만회 • 4개월 전

배민이 얼마나 떼가냐면요
배민 배민원 수수료 현실은 이렇습니다. 자영업자 공감 잃어버린...
진심인사들
조회수 31만회 • 2주 전

Recording
Playlist 어느덧 겨울이 오고 있는 걸
매일룩TAKE A LOOK
조회수 33만회 • 5일 전

한국의 실제 군사력
전쟁다면 밝혀지는 한국의 실제 군사력 수준 (유럽 가면 최강ㄷㄷ)
BODA
조회수 125만회 • 5일 전

Q&A 답변
Q&A 답변영상
회오리
조회수 1.2천회 • 1일 전

모델링 예시



모델링 예제 - 개념적

화면을 보고, 데이터 상의 개념을 추출

검색

전체 게임 음악 실시간 요리 축구 액션 어드벤처 게임 만화 영화 최근에 업로드된 동영상

"의원님들 양해 바랍니다"
간담회 중 뛰쳐나간 대원들 2:37
[오늘 이 뉴스] "의원님들 양해 바랍니다" 간담회 중 뛰쳐나간 대원들...
MBCNEWS
조회수 12만회 • 39분 전

90년대 인기발라드 모음
피아노 연속듣기 6:47:21
2nd) 당신이 사랑하는 1 인기가요,
발라드 1 감미로운 [11] 피아노 버...
지니도비지엠 Jinido bgm
조회수 33만회 • 8개월 전

[22/23 세리에A 14R] 나폴리 vs 영
폴리 김민재 활약상 4:22
SPOTV
[22/23 세리에A 14R] 나폴리 vs 영폴리 김민재 활약상
SPOTV
조회수 10만회 • 1시간 전

중서는 아직 예능 적응 중
진지한데 묘하게 웃긴 전종민 14:48
[골라봐야지] "왜 이렇게 진지해?" 종이의 집 도요량 전혀 딴...
JTBC Voyage
조회수 142만회 • 4개월 전

배민이 얼마나
폐가나면요 12:35
배민 배민원 수수료 현실은 이렇습니다. 자영업자 공감 잃어버린...
진심인사람들
조회수 31만회 • 2주 전

Recording
2:09:05
Playlist 어느덧 겨울이 오고 있는 걸
매일 Take a Look
조회수 33만회 • 5일 전

한국의 실제 군사력
9:10
전쟁나면 밝혀지는 한국의 실제 군사력 수준 (유럽 가면 최강ㄸ)
BODA 보다
조회수 125만회 • 5일 전

Q&A 답변
0:46
Q&A 답변영상
회오리
조회수 1.2천회 • 1일 전

영상
썸네일 사진
영상
영상 길이
영상 제목
조회수
업로드 시간
카테고리

채널
채널명
채널 사진

모델링 예시

모델링 예제 - 개념적

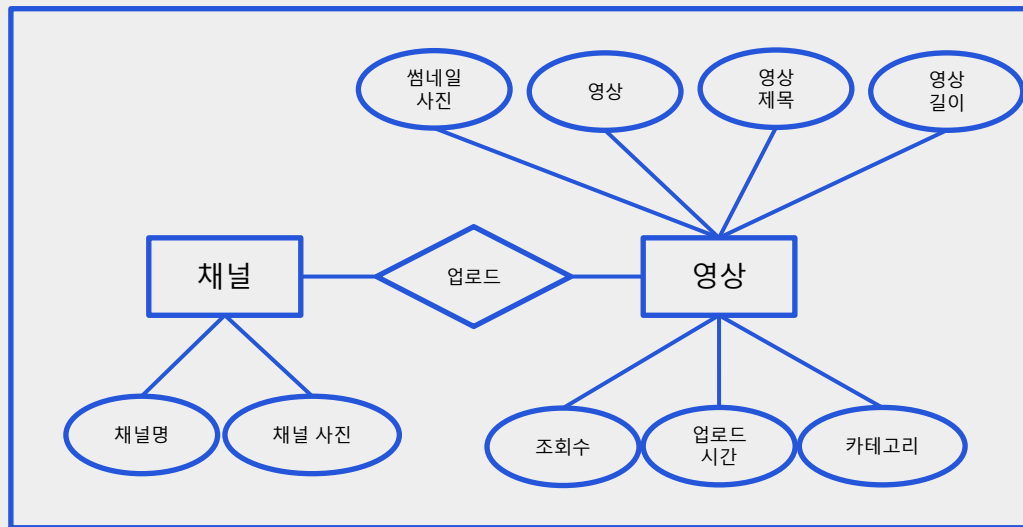
추출한 데이터를 바탕으로 ER다이어그램 작성

영상

썸네일 사진
영상
영상 길이
영상 제목
조회수
업로드 시간
카테고리

채널

채널명
채널 사진

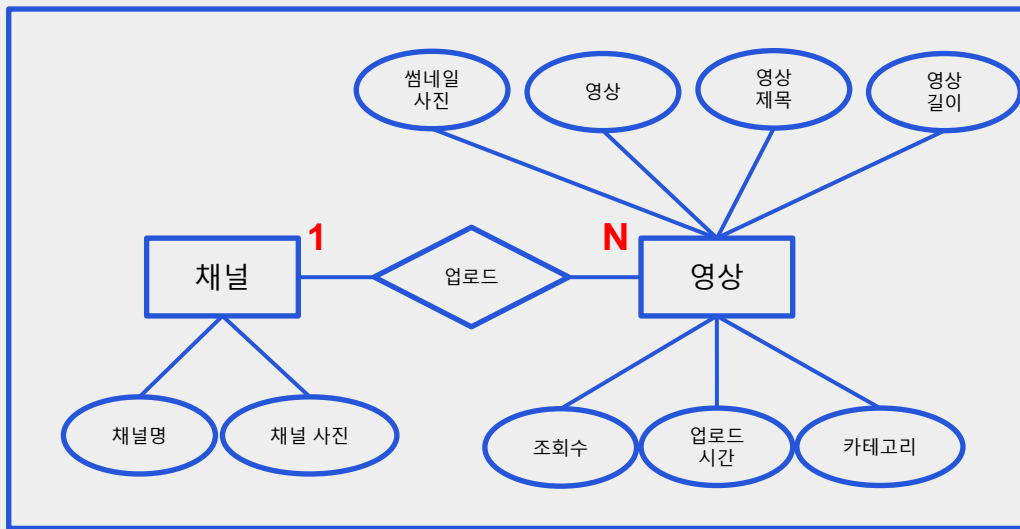


모델링 예시



모델링 예제 - 논리적

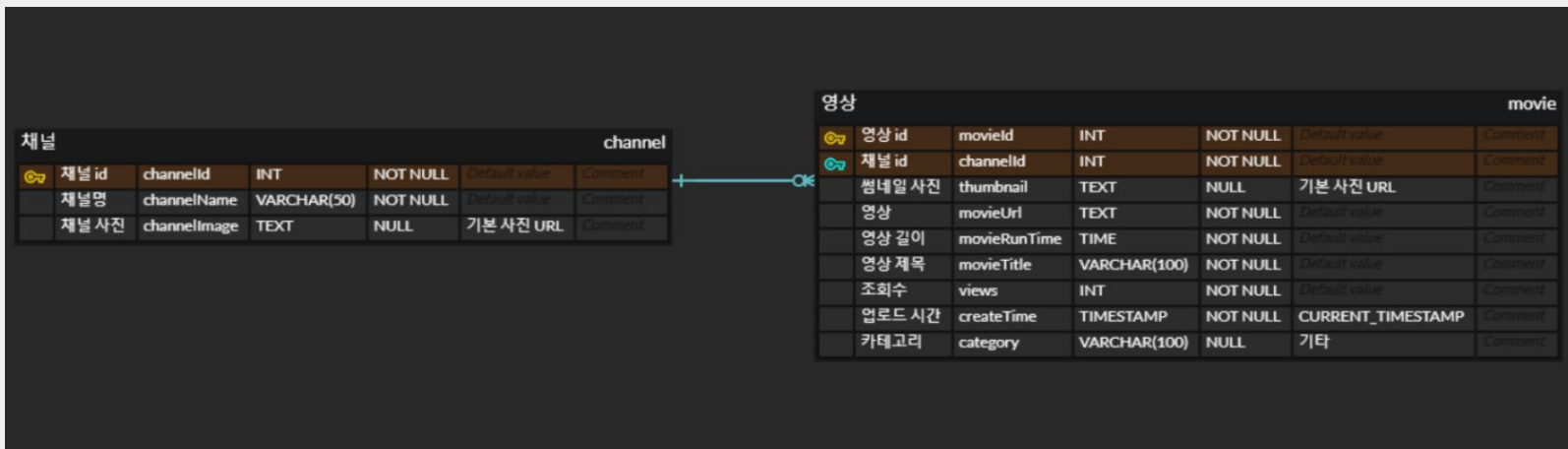
ER다이어그램을 바탕으로 각 객체 간의 관계 설정



모델링 예시

모델링 예제 - 논리적

ER다이어그램을 기반으로 테이블 관계도 작성






모델링 팁 - PK 값

모든 테이블에는 자동 생성되는 PK 컬럼을 별도로 둘 것
매핑 테이블도 동일하게 PK를 컬럼을 둘 것

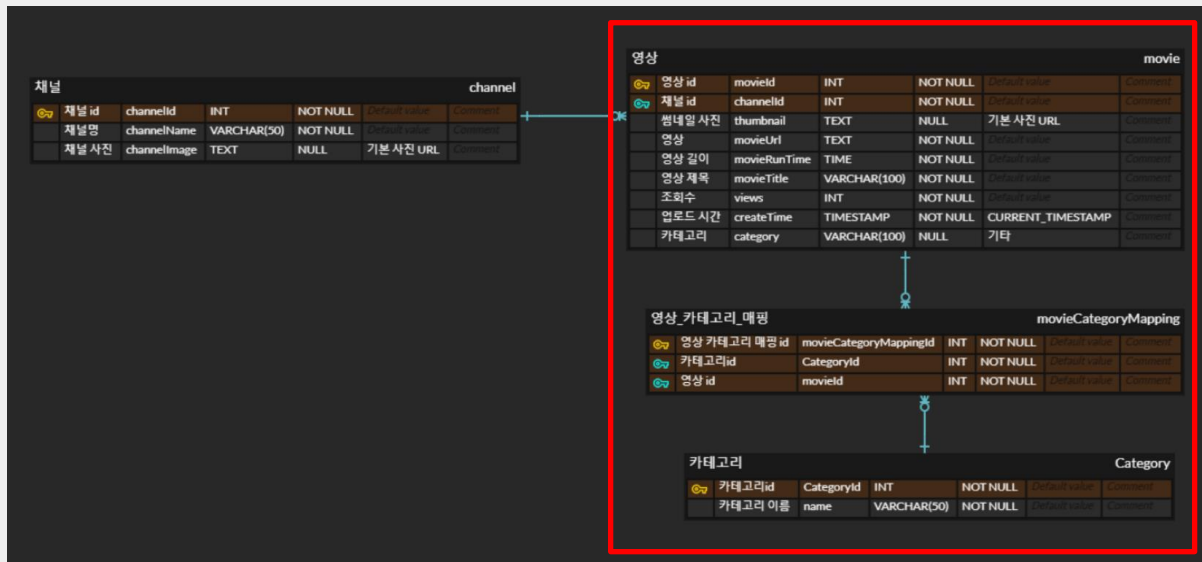
회원 번호	이름	성별	나이	이메일	전화번호
1	유재석	남	10	aaa@naver.com	010-1111-1111
2	정준하	남	20	bbb@naver.com	010-2222-2222
3	노홍철	남	30	ccc@naver.com	010-3333-3333
4	박명수	여	40	ddd@naver.com	010-4444-4444

영상_카테고리_매핑	
영상id	카테고리id
1	1
1	2
2	2
2	3
3	2
3	4

영상_카테고리_매핑		movieCategoryMapping				
	영상 카테고리 매핑 id	movieCategoryMappingId	INT	NOT NULL	Default value	Comment
	카테고리id	CategoryId	INT	NOT NULL	Default value	Comment
	영상 id	movieId	INT	NOT NULL	Default value	Comment

모델링 팁 - N:M 관계

N:M 관계는 1:N 관계 2개로 표현



모델링 팁 - 네이밍

정해진 네이밍은 없지만, 하나의 ERD 안에서는 통일해서 사용
ex) 카멜 케이스, 파스칼 케이스, 스네이크 케이스

카멜 케이스 `movieName`

파스칼 케이스 `MovieName`

스네이크 케이스 `movie_name`

DB 전문가들은 대문자에 스네이크 케이스를 주로 사용
현 교육 과정에서는 카멜 케이스를 추천

모델링 팁



모델링 팁 - 생성, 업데이트 일자

모든 테이블에는 생성, 업데이트 일자를 추가해주자

채널						channel
채널 id	channelId	INT	NOT NULL	Default value	Comment	
채널명	channelName	VARCHAR(50)	NOT NULL	Default value	Comment	
채널 사진	channelImage	TEXT	NULL	기본 사진 URL	Comment	

채널						channel
채널 id	channelId	INT	NOT NULL	Default value	Comment	
채널명	channelName	VARCHAR(50)	NOT NULL	Default value	Comment	
채널 사진	channelImage	TEXT	NULL	기본 사진 URL	Comment	
생성일	created	TIMESTAMP	NOT NULL	Default value	Comment	
업데이트일	updated	TIMESTAMP	NOT NULL	Default value	Comment	

모델링 팁 - 논리 삭제

주요 테이블에는 상태 값(ACTIVE, INACTIVE)로 삭제 여부를 나타내자

채널						channel
채널 id	channelId	INT	NOT NULL	Default value	Comment	
채널명	channelName	VARCHAR(50)	NOT NULL	Default value	Comment	
채널 사진	channelImage	TEXT	NULL	기본 사진 URL	Comment	
생성일	created	TIMESTAMP	NOT NULL	Default value	Comment	
업데이트일	updated	TIMESTAMP	NOT NULL	Default value	Comment	

채널						channel
채널 id	channelId	INT	NOT NULL	Default value	Comment	
채널명	channelName	VARCHAR(50)	NOT NULL	Default value	Comment	
채널 사진	channelImage	TEXT	NULL	기본 사진 URL	Comment	
생성일	created	TIMESTAMP	NOT NULL	Default value	Comment	
업데이트일	updated	TIMESTAMP	NOT NULL	Default value	Comment	
상태	state	VARCHAR(10)	NOT NULL	Default value	Comment	

모델링 팁 – null, default 값

각 컬럼에 맞게 null, default 값을 채워주자
특히 생성일, 수정일, 상태 값

채널		channel			
채널 id	channelid	INT	NOT NULL	Default value	Comment
채널명	channelName	VARCHAR(50)	NOT NULL	Default value	Comment
채널 사진	channelImage	TEXT	NULL	기본 사진 URL	Comment
생성일	created	TIMESTAMP	NOT NULL	Default value	Comment
업데이트일	updated	TIMESTAMP	NOT NULL	Default value	Comment
상태	state	VARCHAR(10)	NOT NULL	Default value	Comment

채널		channel			
채널 id	channelid	INT	NOT NULL	Default value	Comment
채널명	channelName	VARCHAR(50)	NOT NULL	Default value	Comment
채널 사진	channelImage	TEXT	NULL	기본 사진 URL	Comment
생성일	created	TIMESTAMP	NOT NULL	CURRENT_TIMESTAMP	Comment
업데이트일	updated	TIMESTAMP	NOT NULL	CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP	Comment
상태	state	VARCHAR(10)	NOT NULL	ACTIVE	Comment

모델링 팁 - 데이터 타입

각 컬럼에 가장 알맞은 데이터 타입을 명시

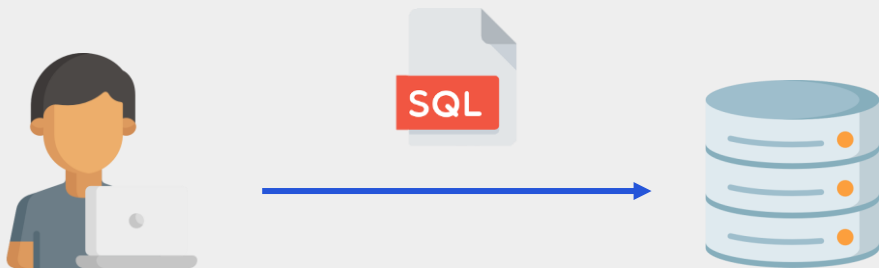
데이터 타입 고려 사항(MySQL 기준)

- 사용하는 DB엔진에 맞는 데이터 타입 확인하기
- 각 컬럼의 데이터를 다루기에 가장 쉬운 데이터 타입으로
 - URL은 TEXT 타입으로
 - 비밀번호는 길이 200이상으로
 - 양수 INT는 INT UNSIGNED 타입으로
- 데이터베이스의 ENUM 데이터 타입 사용은 지양

SQL

SQL

관계형 데이터베이스의 데이터를 관리하기 위해 만들어진 언어
DCL, DDL, DML 3종류가 있다.



DCL

데이터 제어 언어
데이터베이스에 접근하고 객체들을 사용하도록 **권한**을 주고 회수하는 명령어
ex) grant, revoke 등

■ GRANT

```
-- 사용자 권한 부여 명령어
GRANT ALL PRIVILEGES ON [dbname.table_name] TO [user@host] IDENTIFIED BY 'my_
password';

-- 설정한 권한 적용 명령어
FLUSH PRIVILEGES;
```

■ REVOKE

```
-- 권한 해제 명령어(INSERT, UPDATE, CREATE 권한 해제)
REVOKE insert,update,create ON[dbname.table_name] TO [user@host];
```

SQL

DDL

데이터 정의어

테이블과 같은 **데이터 구조를 정의**하는데 사용하는 언어
ex) create, alter, drop 등

```
-- 테이블 관련 DDL
CREATE TABLE [테이블명] -- 테이블 생성
DROP TABLE [테이블명] [CASCADE | RESTRICT] -- 테이블 삭제
ALTER TABLE [테이블명] ADD [컬럼명] [데이터타입] (제약조건) -- 컬럼 추가
ALTER TABLE [테이블명] DROP [컬럼명] -- 컬럼 삭제

-- 뷰 관련 DDL
CREATE VIEW [뷰이름] AS 조회쿼리 -- 뷰 생성
CREATE OR REPLACE VIEW [뷰이름] AS [조회쿼리] -- 뷰 수정
DROP VIEW [뷰이름] -- 뷰 삭제

-- 인덱스 관련 DDL
CREATE (UNIQUE) INDEX [인덱스명] ON 테이블명(컬럼1, 컬럼2 ...) -- 인덱스 생성
ALTER (UNIQUE) INDEX [인덱스명] ON 테이블명(컬럼1, 컬럼2 ...) -- 인덱스 수정
DROP INDEX [인덱스명] -- 인덱스 삭제
```

SQL

DML

데이터 조작어

테이블에 데이터 입력, 수정, 삭제, 조회 시 사용
ex) insert, select, update, delete 등

```
-- CRUD
> INSERT INTO 테이블명 (속성1, 속성2, ...) VALUES (데이터1, 데이터2 ...)
> SELECT [ ALL : DISTINCT ] 속성1, 속성2 ... FROM [테이블명]
(WHERE 조건) (GROUP BY 속성1, ...) (HAVING 그룹 조건) (ORDER BY 속성)
> UPDATE 테이블명 SET 속성명 = 데이터, ... WHERE 조건
> DELETE FROM [테이블명] WHERE 조건
```

조인 연산, 집합 연산도 포함

SELECT



SELECT

SELECT문은 테이블의 데이터를 조회하는 문법이다.
SELECT, FROM, WHERE, GROUP BY, HAVING, ORDER BY 문법으로 구성된다

```
SELECT [ ALL : DISTINCT ] 속성1, 속성2 ...  
FROM [테이블명]  
(WHERE 조건)  
(GROUP BY 컬럼1, ...)  
(HAVING 그룹 조건)  
(ORDER BY 컬럼1 [ASC | DESC], ...)
```

FROM – WHERE – GROUP BY – HAVING – SELECT – ORDER BY

SQL 실습



SQL 실습

프로그래머스 SQL 고득점 Kit 풀이
(SELECT 부분 실습)

https://school.programmers.co.kr/learn/challenges?tab=sql_practice_kit

```
SELECT [ ALL : DISTINCT ] 속성1, 속성2 ...  
FROM [ 테이블명 ]  
(WHERE 조건)  
(GROUP BY 컬럼1, ...)  
(HAVING 그룹 조건)  
(ORDER BY 컬럼1 [ASC | DESC], ...)
```

FROM – WHERE – GROUP BY – HAVING – SELECT – ORDER BY

SELECT



SELECT WHERE절

```
SELECT *
FROM 테이블명

-- 일치값 검색
WHERE number = 10;
WHERE name = '홍길동';

-- 비교연산자 사용
WHERE salary >= 1500;
WHERE hiredate <= '2020-01-01';

-- 논리연산자 사용
WHERE name = '홍길동' AND number = 10;
WHERE NOT name = '홍길동';

-- BETWEEN: 사이인 값 검색(이상, 이하)
WHERE salary BETWEEN 1000 AND 2000;

-- IN: 하나라도 일치하면 참
WHERE number IN(10,20,30);

-- LIKE: 지정한 문자 패턴 검색
WHERE name LIKE '%길%';
WHERE name LIKE '홍_동';

-- NULL
WHERE name IS NULL; -- null만 조회
WHERE name IS NOT NULL; -- null이 아닌 것만 조회
SELECT IFNULL(name, 'No Name'); -- null인 것을 No Name으로 대체하여 조회
```

집계 함수

집계 함수

집계성 SQL에는 집계함수, 그룹함수, 윈도우함수 등이 있다.
집계함수에 대해 알아두자

함수명	함수표기	설 명
SUM	SUM()	합계를 구한다.
AVG	AVG()	평균을 구한다.
MIN	MIN()	최솟값을 구한다.
MAX	MAX()	최댓값을 구한다.
COUNT	COUNT()	행의 개수를 센다.
COUNT DISTINCT	COUNT(DISTINCT)	행의 개수를 센다. (중복은 1개만 인정)

o_num	o_id	o_g_code	o_amount	o_reg_date
1	id003	g006	2	2020-03-17 13:00:17
2	id004	g026	5	2020-03-17 14:55:17
3	id008	g052	10	2020-03-17 14:55:41
4	id009	g034	20	2020-03-17 14:56:05
5	id011	g052	40	2020-03-17 14:56:25
6	id014	g004	8	2020-03-17 14:56:49
7	id003	g008	1	2020-03-17 14:57:10
8	id005	g014	20	2020-03-17 14:57:26
9	id005	g027	50	2020-03-17 14:57:57
10	id013	g020	13	2020-03-17 14:58:26

```
SELECT
  o.o_id AS '구매자 아이디'
, sum(o.o_amount) AS '총 주문수량'
, AVG(o.o_amount) AS '평균 주문수량'
, ROUND(AVG(o.o_amount),1) AS '평균 주문수량'
, COUNT(o.o_amount) AS '총 주문횟수'
, MAX(o.o_amount) AS '최대 주문수량'
, MIN(o.o_amount) AS '최소 주문수량'
FROM
  tb_order AS o
GROUP BY o.o_id;
```

JOIN

JOIN은 두 개의 테이블을 서로 묶어서 하나의 결과를 만들어 내는 문법이다.
JOIN 문에는 INNER, OUTER, CROSS, SELF 조인 등이 있다.

INNER JOIN(내부 조인)은 두 테이블을 조인할 때, 두 테이블에 모두 지정한 열의 데이터가 있어야 한다.

OUTER JOIN(외부 조인)은 두 테이블을 조인할 때, 1개의 테이블에만 데이터가 있어도 결과가 나온다.

CROSS JOIN(상호 조인)은 한쪽 테이블의 모든 행과 다른 쪽 테이블의 모든 행을 조인하는 기능이다.

SELF JOIN(자체 조인)은 자신이 자신과 조인한다는 의미로, 1개의 테이블을 사용한다.

JOIN

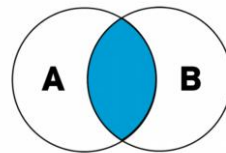


INNER JOIN

두 테이블의 공통 데이터를 묶어서 조회.
두 테이블 모두 지정한 열의 데이터가 있어야 결과가 나온다.

```
SELECT <열 목록>  
FROM <첫 번째 테이블>  
      INNER JOIN <두 번째 테이블>  
      ON <조인될 조건>  
[WHERE 검색 조건]
```

INNER JOIN



* INNER JOIN을 JOIN이라고만 써도 INNER JOIN으로 인식합니다.

JOIN



OUTER JOIN

두 테이블을 조건(LEFT, RIGHT, FULL)에 따라 묶어서 조회
두 테이블 중 한쪽에만 데이터가 있어도 결과가 출력된다

LEFT OUTER JOIN: 왼쪽 테이블의 모든 값이 출력되는 조인

RIGHT OUTER JOIN: 오른쪽 테이블의 모든 값이 출력되는 조인

FULL OUTER JOIN: 왼쪽 또는 오른쪽 테이블의 모든 값이 출력되는 조인

SELECT <열 목록>

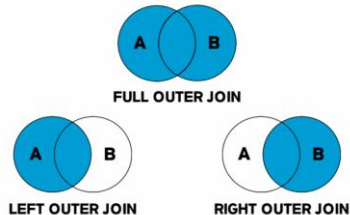
FROM <첫 번째 테이블(LEFT 테이블)>

<LEFT | RIGHT | FULL> OUTER JOIN <두 번째 테이블(RIGHT 테이블)>

ON <조인될 조건>

[WHERE 검색 조건]

OUTER JOIN





과제 안내

Mysql 강의 수강

Section1 ~ Section13 중에서 필요한 부분을 선택적으로 수강

테이블 생성 및 데이터 삽입, 제약조건

Section3 ~ Section4, Section12

기본 SQL 학습

Section5 ~ Section6

문자열 함수 및 세부 문법

Section7 ~ Section8, Section10 ~ Section11

집계함수

Section9

Join문

Section13 ~ Section14

쿠팡 ERD 설계

쿠팡의 핵심 서비스인 **상품 조회, 주문, 결제, 배송**과 관련된 기능 위주로 ERD 설계

카테고리
상품 리스트
상품 상세 페이지
장바구니
주문
배송
교환&반품
리뷰

멤버십(로켓프레시, 로켓배송)기획전, 여행/티켓, 댓글 도움이 돼요, 각 상품별 세부 리뷰 등 핵심 기능 외 부가 기능은 제외



과제 안내

쿠팡 ERD 설계

- 이용자는 크게 고객과 판매자로 구분된다.
- 카테고리는 대, 중, 소 카테고리로 설계한다.
- 하나의 상품은 여러 판매자가 판매할 수 있다.
- 하나의 상품은 다양한 옵션, 개수로 판매할 수 있다.
- 판매자는 상품 게시글을 여러 개 등록할 수 있다.
- 고객은 상품 게시글을 보고 상품 옵션을 여러 개 선택하여 주문할 수 있다.
- 고객이 주문한 이후 상품 내용이 바뀌더라도 주문 내용은 바뀌어서는 안된다.
 - 고객은 주문 이후 결제 정보를 확인할 수 있어야 한다.
 - 고객은 주문 이후 배송 정보를 확인할 수 있어야 한다.
 - 고객은 구매한 상품에 대해 상품평을 남길 수 있어야 한다.
 - 각각의 관계는 PK, FK로 표현되어야 한다.
- 쿠팡의 핵심 기능 외 복잡성을 증가시키는 기능은 설계에서 제외한다.

멤버십(로켓프레시, 로켓배송)기획전, 여행/티켓, 댓글 도움이 돼요, 각 상품별 세부 리뷰 등 핵심 기능 외 부가 기능은 제외



과제 안내

프로그래머스 고득점 Kit 풀이


프로그래머스 SQL 고득점 Kit 풀이 - 2레벨 문제까지 풀이(JOIN은 3레벨까지)

https://school.programmers.co.kr/learn/challenges?tab=sql_practice_kit

SQL 고득점 Kit

코딩 테스트에서 SQL문제를 만날 확률이 25%나 된다는 사실, 알고 계셨나요?
이제 프로그래머스에서 SQL 퀴리도 연습하세요!

기초적인 SELECT문부터 어려운 JOIN문까지, 실습 문제를 통해 차근차근 SQL 기본기를 다져보세요.



SELECT

조건에 맞는 레코드를, 원하는 순서대로, 기본기를 처음부터 탄탄히 다져보세요.

문제 번호	평균 점수	문제 세트
보통 ↘	높음 ↗	0 / 17

SUM, MAX, MIN

나이가 가장 많은 사람은 누구일까? 데이터를 뒤져 총계를 내어봅시다.

문제 번호	평균 점수	문제 세트
보통 →	높음 ↗	0 / 6

GROUP BY

나랑 이름이 같은 사람은 몇 명일까요? 데이터를 묶고 평균값을 계산해보세요.

문제 번호	평균 점수	문제 세트
보통 →	보통 →	0 / 10

IS NULL

값이 없음을 의미하는 NULL. NULL을 처리하는 방법을 배워봅시다.

문제 번호	평균 점수	문제 세트
보통 ↘	보통 →	0 / 5

JOIN

테이블 사이의 복잡한 관계를 파악해보아요. 이제부터 어렵습니다.

문제 번호	평균 점수	문제 세트
높음 ↗	보통 ↘	0 / 9

String, Date

숫자는 어떻게 쓰는 건지 알겠는데, 글자와 날짜는 어떻게 다루지?

문제 번호	평균 점수	문제 세트
보통 ↘	보통 ↘	0 / 8

Hello World!