

## 40. DBMS ဆိုတဲ့ (၇)။

\* ဤအားလုံး၏

JDBC API

Database

[datetime]

(ii) yyyy-MM-dd hh:mm:ss

getDate()  $\Rightarrow$  java.sql.Date  $\Rightarrow$   
yyyy-MM-dd

getTimestamp()  $\Rightarrow$  java.sql.Timestamp  $\Rightarrow$   
yyyy-MM-dd hh:mm:ss,

# 41. 외부키(foreign key) 활용

myapp-users

user_id	name	email	pwd	created_at
1	aaa	-	-	-
2	bbb	-	-	-
3	ccc	-	-	-

myapp-projects

project_id	Title	description	startdate	enddate	members
101	p1	-	-	-	1,3
102	p2	-	-	-	1,2
103	p3	-	-	-	1,2,3

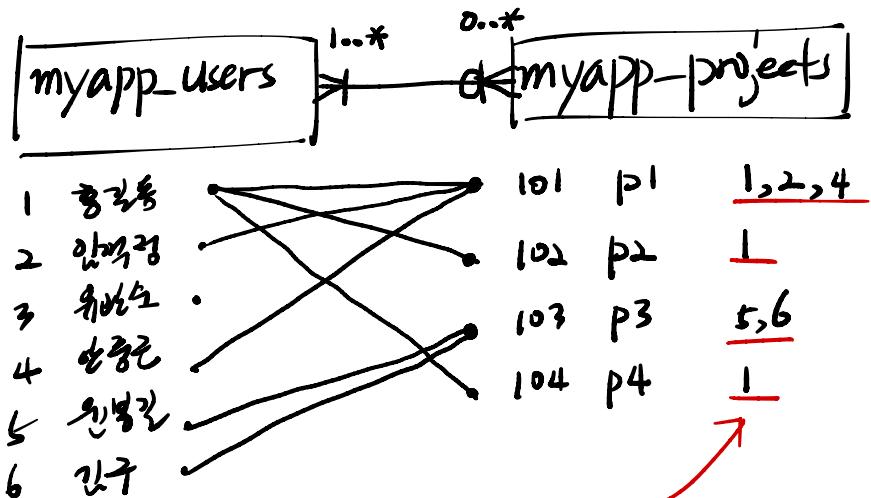
1번은 1번

프로젝트에 참여하는  
사람은 반드시 있는  
필수값입니다!

↑  
필수값이 까지다!  
↳ 필수값은!

## 41. 외부키(foreign key) 활용

1) 데이터의 데이터간에 관계를 짜는  
(cf 대 cf 관계)



(cf 대 cf)  
(cf 대 cf 관계를

외부 키와 함께 관계의 풍부함을

증가시킬 수 있어서 광범위한 활용 가능

Foreign key 활용  
+  
데이터의 관계  
관계형 데이터베이스에서 관계를

증가?

\* 데이터를 관리하는 틀

- 1 흐름
- 2 일정
- 3 주제

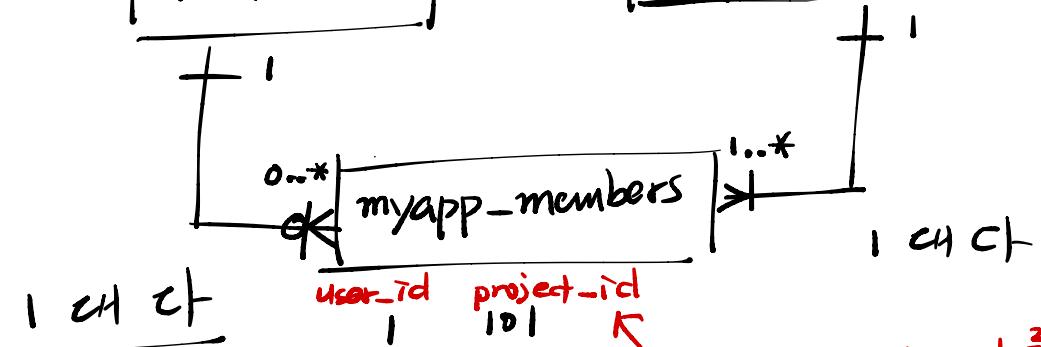
myapp-users	
1	2

- 1 흐름
- 2 일정
- 3 주제

myapp-projects	
101	p1
102	p2
103	p3

\* 외부키 (foreign key)

- 다른 테이블의 pk를 가리킨다
- 같은 pk를 사용할 수 있다



1 CH C1

	user_id	project_id
1	101	
1	103	
2		101
2		102
2		103

다른 테이블의 pk를 → 가리킨다  
"Foreign key"

\* DB의 특성 짚기

① 데이터가 구조화된



data + 구조  
구조화된  
데이터



구조화된

데이터구조가  
가장자리  
(제품 정보)



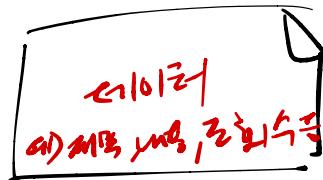
제품 번호, 제조사  
제작지는 10자리



제품을 알고 싶을 때  
마우스로 드래그

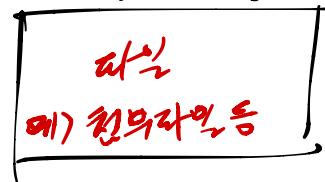
구조?  
→

데이터 구조의  
구조화



제품구조화  
제작국가

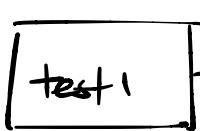
제품 구조화



제품 구조화

\* Foreign key

*Parent table*



*table = entity*



*pk*

no	title	content	rdt
----	-------	---------	-----

1	aaa	-	-
2	bbb	-	-
3	ccc	-	-
4	ddd	-	-
5	eee	-	-

*row = record*

*pk*

fno	filepath	bno
-----	----------	-----

101	a1.gif	1
102	a2.gif	1
103	a3.gif	1
104	b1.gif	5
105	b2.gif	5
106	x.gif	00

*fk*

→ test1 (no)

test1의 no는 5이므로 5번 행