

University Table					
	Univ ID	Name	Access Control	Default value	Comment
	1	서울여자대학교	1	Default value	Comment
	2	고려대학교	0	Default value	Comment
	3	하버드대학교	0	Default value	Comment

Department Table					
	Dep ID	Name	Access Control	Univ ID	Comment
	1	정보보호학과	1	1	Comment
	2	사회복지학과	0	1	Comment
	3	기독교학과	0	1	Comment
	4	아동학과	0	1	Comment
	5	디지털미디어학과	0	1	Comment

Grade Table					
	Grade ID	Grade	Access Control	Dep ID	Univ ID
	1	Freshman	0	1	1
	2	Sophomore	0	1	1
	3	Junior	0	1	1
	4	Senior	0	1	1

Professor Table					
	Prof ID	Prof	Open	Default value	Comment
	1	김형중	1	Default value	Comment
	2	박후린	1	Default value	Comment
	3	최은정	1	Default value	Comment
	4	오준형	0	Default value	Comment

PBL Table					
	Class ID	Domain	Prof ID	Default value	Comment
	1	Domain	1	Default value	Comment
	2	Domain	2	Default value	Comment
	3	Domain	3	Default value	Comment

Student Table #1					
	Name	MBTI ID	Dep ID	Class ID	Grade ID
	이경화	6	1	2	3
	손효림	3	0	2	3
	한별하	6	1	2	3

Student Table #2					
	Name	MBTI ID	Dep ID	Class ID	Grade ID
	김다은	9	1	1	4
	김민지	11	1	1	4
	김채영	13	1	1	4
	이승하	4	1	1	4
	오유진	3	1	2	4
	허희윤	9	1	2	4

MBTI Table					
	MBTI ID	MBTI	Type	Default value	Comment
	0	ISTJ	Type	Default value	Comment
	1	ISTP	Type	Default value	Comment
	2	INFJ	Type	Default value	Comment
	3	INTJ	Type	Default value	Comment
	4	ISFJ	Type	Default value	Comment
	5	ISFP	Type	Default value	Comment
	6	INFP	Type	Default value	Comment
	7	INTP	Type	Default value	Comment
	8	ESTJ	Type	Default value	Comment
	9	ESFP	Type	Default value	Comment
	10	ENFP	Type	Default value	Comment
	11	ENTP	Type	Default value	Comment
	12	ESFJ	Type	Default value	Comment
	13	ESTP	Type	Default value	Comment
	14	ENFJ	Type	Default value	Comment
	15	ENTJ	Type	Default value	Comment

데이터베이스 설명 상세 #1

데이터베이스 구축 시 고려한 요소: 가시성, 데이터 접근 속도, 접근 시 충돌 방지

본 데이터베이스는 하나의 테이블의 모든 정보를 모아놓기보다는, 각각 다른 특성을 가진 테이블로 분산한 뒤 이들을 연결하는 방식으로 구현되었습니다. 테이블들을 분할한 첫 번째 이유는 데이터들의 가시성을 확보하기 위함입니다. 큰 범위의 테이블부터 University, Department, Grade, Student Table이 있으며, Student 테이블과 직접적으로 연결되는 Professor Table 및 PBL Table, MBTI 테이블 또한 구현하였습니다.

테이블 다분화의 두 번째 이유는 바로 데이터 접근 속도 향상을 위함입니다. 데이터학적으로 하나의 테이블에 모든 데이터가 밀집되어있을 때 검색을 통해 찾는 것보다, 각각의 요소들을 분리하여 다방면으로 접근하는 것이 속도 측면에서 효율적이기에 위와 같이 구현하였습니다.

또한 사용자가 데이터에 접근할 때 충돌이 일어나는 것을 방지하기 위해, "Access Control" 이라는 키값을 주어 해당 데이터가 사용 중인지의 여부를 명확히 알 수 있도록 하였습니다. 이미 사용 중인 데이터의 경우는 1, 현 사용자가 사용 가능한 데이터의 경우는 0(=Default)으로 설정하였습니다.

이어 각 테이블들에 대한 설명입니다. 먼저 University Table의 경우, 각 대학명에 따라 ID를 부여하였습니다. 그 후 선택한 대학에 대해 Department 테이블을 구현하였으며, 각 학과별로 또 다른 ID들을 할당하였습니다. 이때 한 Department DB에는 당연히 같은 대학 내에서의 학과들에 대한 정보들만이 있으므로, Univ ID는 모두 같습니다.

데이터베이스 설명 상세 #2

학과까지 내려왔다면, 그 후 학년을 나타내는 Grade Table이 등장합니다. 1학년부터 4학년까지 고유한 Grade ID를 부여하였으며, 이를 통해 학생 테이블들을 전개합니다. 현재 데이터베이스에는 3,4학년만이 있으므로, 1학년과 2학년 테이블은 제외하였습니다.

Student Table에서 학생들은 이름(가나다) 순으로 정렬되며, 각각 속성값으로 MBTI, 학과, PBL 분반, 학년에 대한 ID를 할당받습니다. 이때 이름을 제외한 다른 ID 값들은 중복될 수 있게 구현하였습니다.

다음으로는 PBL 및 교수 Table 에 대한 설명입니다. PBL 테이블 구현에 앞서, 담당 교수를 명시하기 위한 Professor Table을 구현하였습니다. 각 교수님들께 Prof ID를 부여하고, PBL 분반 개설 여부를 int 형 값 Open으로 표현했습니다. PBL 반을 개설하신 교수님의 경우 1, 그렇지 않으신 분의 경우 0의 값이 들어갑니다.

최종적으로, PBL 반을 개설하신 교수님들에 한해 PBL Class ID가 부여됩니다. 구현한 예시에서는 인원이 적어 Class ID와 Prof ID가 일치하지만, 교수님들의 수가 많아질 경우 각 ID가 달라지는 경우가 발생할 수 있습니다.

마지막으로 MBTI Table에 대한 내용입니다. 해당 테이블에서는 총 16종류의 MBTI별로 MBTI ID를 할당했습니다. MBTI ID는 Student Table 속 학생의 MBTI ID 속성에 매칭됩니다. 학생끼리 MBTI가 동일할 수 있기에, n명 이상의 학생이 같은 MBTI ID를 사용할 수 있도록 구현하였습니다.